Flight path planning for infrastructure inspection based on artificial potential fields

Goričanec, Jurica

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Jurica Goričanec

FLIGHT PATH PLANNING FOR INFRASTRUCTURE INSPECTION BASED ON ARTIFICIAL POTENTIAL FIELDS

DOCTORAL THESIS

Zagreb, 2024

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Jurica Goričanec

FLIGHT PATH PLANNING FOR INFRASTRUCTURE INSPECTION BASED ON ARTIFICIAL POTENTIAL FIELDS

DOCTORAL THESIS

Supervisor: Professor Stjepan Bogdan, PhD

Zagreb, 2024

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Jurica Goričanec

PLANIRANJE PUTANJE LETA ZA INSPEKCIJU INFRASTRUKTURE ZASNOVANO NA UMJETNIM POTENCIJALNIM POLJIMA

DOKTORSKI RAD

Mentor: Prof. dr. sc. Stjepan Bogdan

Zagreb, 2024.

Doctoral thesis is written at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering.

Supervisor: Professor Stjepan Bogdan, PhD

Doctoral thesis has 117 pages

Doktorski rad br.: _____

About the Supervisor

Stjepan Bogdan graduated, and got master and PhD degrees in the field of electrical engineering at the University of Zagreb, Faculty of Electrical Engineering and Computing (FER). Since June 1991, he has been working at the Department of Automation and Computer Engineering of FER. He spent the school year 1996/97 as a visiting researcher at the Automation and Robotics Research Institute, University of Texas at Arlington. In December 2016, he was elected to the position of full professor in a permanent position. From October 2022, he holds the position of vice dean for research and innovation at FER.

As a researcher, he participated in a dozen development research projects financed by the industry. He was the principal investigator of 2 projects financed by the Ministry of Education, Culture, Sports, Science and Technology, 1 project financed by the HRZZ, 2 bilateral Chinese-Croatian projects, 1 project financed by the Air Force Office of Scientific Research and 1 NATO SpS project. He participated in the research on 3 EU FP7 projects (ACROSS, EC-SAFEMOBIL, EuRoC), 7 H2020 projects (subCULTron, ENDORSE, CROBOHUB, ACROSS, RoboCom++, ENCORE, AeroWind), 4 ESIF projects (EKOKOMVOZ, HEKTOR, ASAP, VirtUAV), and was the leader of the work package of the EU FP7-FET project (ASSISI_bf). He currently participates in 2 Horizon2020 projects (WatchPlant, AerialCore), was the coordinator of the Horizon2020 project (AeRoTwin), and currently coordinates the HorizonEurope project AeroSTREAM. He has published 4 books, more than 50 journal papers and more than 160 papers in international conference proceedings in the field of intelligent control systems, multi-agent systems and autonomous drones.

Prof. Bogdan is a member of professional associations IEEE and KoREMA. He was the vice-president of the Croatian Society for Robotics and the president of the CS23 section of the Croatian IEEE section. He participates in dozens of program committees of scientific conferences and is a member of the editorial boards and reviewer of several scientific journals. He is editor-associate in 5 scientific journals. In 1993, he received the "Josip Lončar" silver plaque from FER for a particularly distinguished master's thesis, and in 2000 he received the award of the Association of University Teachers as the best young scientist. In 2013, he received the Science Award, awarded by the FER Faculty Council, in 2015, he received the Fran Bošnjaković Award, awarded by the University of Zagreb, and in 2021, he received the "Josip Lončar" gold plaque, awarded by the FER Faculty Council for scientific contribution in the field of autonomous systems, with an emphasis on the development of heterogeneous robotic systems.

O mentoru

Stjepan Bogdan diplomirao je, magistrirao i doktorirao u polju elektrotehnike na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva (FER). Od lipnja 1991. godine radi na Zavodu za automatiku i računalno inženjerstvo FER-a. Školsku godinu 1996/97 proveo je kao gostujući istraživač na Automation and Robotics Research Institute, Univesrity of Texas at Arlington. U prosincu 2016. godine izabran je u zvanje redovitog profesora u trajnom zvanju. Od listopada 2022. godine obnaša funkciju prodekana za istraživanje i inovacije FER-a.

Kao istraživač sudjelovao je u desetak razvojno istraživačkih projekata financiranih od strane industrije. Bio je voditelj 2 projekta financiranih od strane MZO-a, 1 projekta financiranog od strane HRZZ-a, 2 bilateralno Kinesko-Hrvatska projekta, 1 projekta financiranog od strane Air Force Office of Scientific Research te 1 NATO SpS projekta. Sudjelovao je u radu 3 EU FP7 projekta (ACROSS, EC-SAFEMOBIL, EuRoC), 7 H2020 projekata (subCULTron, ENDORSE, CROBOHUB, ACROSS, RoboCom++, ENCORE, AeroWind), 4 ESIF projekta (EKOKOMVOZ, HEKTOR, ASAP, VirtUAV), bio je voditelj radnog paketa EU FP7-FET projekta (ASSISI_bf). Trenutno sudjeluje u radu 2 Obzor2020 projekta (WatchPlant, AerialCore), bio je koordinator Obzor2020 projekta (AeRoTwin), a trenutno koordinira ObzorEurope projekt AeroSTREAM. Objavio je 4 knjige, više od 50 radova u časopisima i više od 160 radova u zbornicima međunarodnih konferencija u području inteligentnog upravljanja sustavima, višeagentskih sustava i autonomnih bespilotnih letjelica.

Prof. Bogdan član je stručnih udruga IEEE i KoREMA. Bio je potpredsjednik Hrvatskog društva za robotiku i predsjednik odjela CS23 hrvatske sekcije IEEE. Sudjeluje u više desetaka programskih odbora znanstvenih konferencija i član je uređivačkih odbora i recenzent više znanstvenih časopisa. Obavlja funkciju urednika-suradnika u 5 znanstvenih časopisa. Godine 1993. Primio je srebrnu plaketu „Josip Lončar“ FER-a za posebno istaknuti magistarski rad, a 2000. godine primio je nagradu Društva sveučilišnih nastavnika kao najbolji mladi znanstvenik. 2013. godine primio je nagradu Znanost koju mu je dodijelilo Fakultetsko vijeće FER-a, 2015. godine primio je nagradu Fran Bošnjaković koju dodjeljuje Sveučilište u Zagrebu, a 2021. godine primio je zlatnu plaketu „Josip Lončar“, koju dodijeljuje Fakultetsko vijeće FER-a, za znanstveni doprinos u području autonomnih sustava, s naglaskom na razvoj heterogenih robotskih sustava.

Acknowledgements

I am deeply grateful to my mentor, Prof. Stjepan Bogdan, for his unwavering guidance and support throughout my academic journey. His wisdom and belief in my abilities have profoundly influenced both my research endeavors and personal development. I am also indebted to Prof. Matko Orsag for his dedicated support and invaluable insights, without which this milestone would not have been possible.

I extend my sincere thanks to my colleagues at LARICS (Laboratory for Robotics and Intelligent Control Systems) for creating a collaborative and supportive atmosphere, making my time there both enjoyable and productive. I am also grateful to my friends outside the lab for their patience, motivation, and encouragement during the challenging phases of my PhD.

My deepest gratitude goes to my parents and sister for their unconditional love, encouragement, and belief in me. Their support has been a constant source of strength and inspiration. To all these remarkable individuals who have contributed to my academic and personal growth, I am profoundly thankful.

Abstract

The main focus of this thesis is the development of flight path planning algorithms and trajectory tracking methods for autonomous Unmanned Aerial Vehicle (UAV) inspection of infrastructure. These algorithms are based on the concept of artificial potential fields (APF) and are designed to be adaptable across various scenarios characterized by differing levels of complexity in surveyed targets and surrounding environments. The objective is to devise methodologies capable of generating safe and feasible flight paths for UAVs while ensuring the acquisition of comprehensive and high-quality datasets used for infrastructure assessment and damage detection. To be applicable in real-world conditions, the system must be capable of generating a map of the surrounding environment, detecting obstacles in real-time, and executing collision avoidance maneuvers while adhering to the originally planned path in unobstructed space.

The thesis begins with a comprehensive literature review, detailing the state-of-the-art in UAV path planning, APF-based navigation and inspection tasks, as well as the fundamental kinematic and dynamic models of quadcopters, UAV control structures, camera projection models, and methodologies for determining inspection parameters. Additionally, it introduces the UAV platforms and sensors utilized in this research.

Multiple inspection path planning methods are proposed within the thesis, depending on the complexity and existing knowledge about the surveyed target object. This knowledge may be available in the form of a complete 3D model, 2D layout, or only approximate location of the infrastructure object. Moreover, inspection parameters are determined based on the type and specifications of available sensors, as well as specific requirements such as overlapping area between consecutive images and size of features distinguishable in acquired datasets. The developed path planning algorithms are based on principles such as Huygens's wave propagation, construction of 3D envelopes, or the generation of closed isolines of APF.

Furthermore, the thesis introduces a path tracking method for LiDAR-equipped UAVs with real-time obstacle detection and collision avoidance based on augmented APF. The repulsive potential force, generated by detected obstacles, consists of two components - normal and rotational, ensuring that the UAV avoids getting trapped in local minima. The attractive potential forces are generated by the goal point on the initially planned path and obstacle vertex, utilized to minimize deviations during collision avoidance maneuvers. Additionally, the algorithm determines the feasibility of waypoints along the original path and discards unreachable ones to ensure UAV will adhere to planned path in unobstructed space.

The developed methods were extensively tested and validated through a series of realistic simulations, laboratory experiments, and real-world inspection scenarios, encompassing a range of structures from simple residential buildings to complex bridges and wind turbines, conducted across several projects.

The significant scientific contribution of this thesis include:

- *Inspection flight path planning for unknown infrastructures based on closed isolines of artificial potential fields*
- *Flight path tracking method in an unknown environment based on artificial potential fields*

Keywords: Unmanned Aerial Vehicle, Infrastructure Inspection, Artificial Potential Fields, Flight Path Planning, Obstacle Avoidance

Planiranje putanje leta za inspekciju infrastrukture zasnovano na umjetnim potencijalnim poljima

Glavni fokus ove doktorske disertacije je razvoj algoritama za planiranja putanje leta i metoda praćenja trajektorije za autonomnu inspekciju infrastrukture bespilotnim letjelicama na temelju umjetnih potencijalnih polja. Ovi algoritmi su osmišljeni da budu prilagodljivi na razne scenarije koji se međusobno razlikuju po razini složenosti infrastrukturnih objekata i okruženja u kojem se nalaze. Cilj je osmišljavanje metodologije kojom je moguće generirati sigurne i izvedive putanje leta za bespilotne letjelice te pritom osiguravati prikupljanje sveobuhvatnih i visokokvalitetnih podataka koji se u daljnjim koracima inspekcije koriste za procjenu stanja infrastrukture i otkrivanje mogućih oštećenja. Da bi razvijeni sustav bio primjenjiv u stvarnim uvjetima on mora biti sposoban generirati kartu okolnog prostora, detektirati prepreke u stvarnom vremenu te izvoditi manevre za izbjegavanja sudara uz uvjet praćenja izvorno planirane putanje u slobodnom prostoru.

Prvi dio disertacije obuhvaća pregled literature, detaljno opisujući najnovija dostignuća u planiranju putanja za bespilotne letjelice, navigaciji temeljenoj na umjetnim potencijalnim poljima i zadacima vezanim uz korištenje autonomnih sustava inspekcije. Uz to, u disertaciji su opisani osnovni kinematički i dinamički model višerotorske letjelice, strukture upravljanja bespilotnim letjelicama, modele projekcije kamere i metoda za određivanje relevantnih parametara inspekcije. Nadalje, predstavljene su bespilotne leteće platforme i senzori korišteni u ovom istraživanju.

Unutar disertacije predloženo je nekoliko metoda planiranja putanja letjelice korištene za autonomnu inspekciju, ovisno o složenosti i postojećem znanju o istraživanom objektu. Informacije o infrastrukturi mogu biti dostupne u obliku potpunog 3D modela, 2D tlocrta ili samo njene približne lokacije. Parametri inspekcije određuju se na temelju vrste i specifikacija korištenih senzora, kao i specifičnih zahtjeva poput područja preklapanja između susjednih slika te veličine oštećenja koje je moguće razlikovati u prikupljenim setovima podataka.

Razvijeni algoritmi planiranja putanja temelje se na principima poput Huygensove teorije širenja valova kojom se generira niz krugova oko poznatog tlocrta zgrade te se na temelju njihova sjecišta određuje putanja letjelice, konstrukcije 3D ovojnice za planiranje putanje kojom se osigurava da su sve točke na planiranoj putanji jednako udaljene od najbliže točke na zadanom modelu objekta ili generiranja zatvorenih izolinija umjetnih potencijalnih polja za slučaj kad su prilikom planiranja inspekcijske putanje dostupne informacije samo o lokaciji objekta.

Nadanje, u disertaciji je predstavljena metoda za praćenje putanje bespilotne letjelice opremljene LiDAR-om koja omogućuje detekciju prepreka u stvarnom vremenu te izbjegavanjem sudara na temelju modificiranih umjetnih potencijalnih polja. Odbojna potencijalna sila, generirana na temelju detektiranih prepreka, sastoji se od dvije komponente - normalne i rotacijske,

čime se osigurava to da bespilotna letjelica neće zapeti u točkama lokalnog minimuma. Generirani oblak točaka koji predstavlja trodimenzionalnu mapu okoline u kojoj se kreće bespilotna letjelica je podijeljen u više grupa sa zasebnim težištima. Svaka grupa predstavlja pojedinu prepreku čime je omogućen prolazak letjelice između dvije prepreke ako između njih postoji dovoljan razmak za siguran let.

Privlačne potencijalne sile generiraju se prema ciljnoj točki na izvorno planiranoj putanji te vrhu prepreke što se koristi kako bi se minimizirala odstupanja tijekom izvođenja manevra za izbjegavanja sudara. Osim toga, algoritam određuje je li moguće sigurno stići u pojedinu točku duž izvorne putanje te ukoliko se utvrdi da se točka nalazi preblizu prepreci ona se odbacuje, a letjelica kreće prema prvoj sljedećoj koja je slobodna.

Razvijene metode temeljito su testirane i validirane nizom realističnih simulacija, laboratorijskih eksperimenata i stvarnih inspekcijskih scenarija provedenih u nekoliko projekata, čime je obuhvaćen široki raspon struktura od jednostavnih stambenih zgrada do složenih mostova i vjetroelektrana.

Disertacija je podijeljena u sedam poglavlja.

U prvom poglavlju dan je uvod u područje bespilotnih letjelica te njihove primjene. Poboljšanjem modernih elektroničkih komponenata i senzora te razvojem različitih algoritama za upravljanje, lokalizaciju i navigaciju robota, omogućeno je korištenje bespilotnih letjelica u širokom spektru djelatnosti uključujući fotografiranje i snimanje iz zraka, nadzor nepristupačnih područja, pomoć u slučajevima katastrofe te mapiranja terena. Vrijedi istaknuti njihov značaj u revolucioniranju klasičnih metoda inspekcije infrastrukture, posebno u kontekstu povećanja efikasnosti i smanjenja rizika za ljude kod inspekcije kritičnih objekata kao što su mostovi, industrijska postrojenja, dalekovodi i vjetroelektrane.

U drugom poglavlju prikazan je pregled relevantne literature za ovu disertaciju. Prezentirani su radovi iz područja planiranja putanja za bespilotne letjelice, korištenje umjetnih potencijalnih polja u algoritmima navigacije te metode inspekcija infrastrukture autonomnim robotima. Uz to, dan je pregled prijašnjih radova na projektima vezanim uz temu disertacije - generiranje detaljnih trodimenzionalnih modela stambenih i povijesnih građevina koji služe kao podloga za arhitektonsko projektiranje, obnovu i urbanistički razvoj, razvoj sustava bespilotnih letjelica za autonomnu inspekciju vjetroelektrana s ciljem povećanja efikasnosti i sigurnosti kod prikupljanja podataka o stanju vjetroturbine i lopatica te poboljšanje i prilagodba sustava navigacije bespilotnih letjelica korištenih za inspekcije mostova.

U trećem poglavlju iznesen je detaljan pregled kompletnog sustava za autonomnu inspekciju infrastrukture korištenjem bespilotnih letjelica. Arhitektura sustava sastoji se od ulaznih podataka o cilju i parametrima inspekcije kao i ograničenjima na brzinu i akceleraciju letjelice

koja su postavljena od strane operatera, modula za planiranje putanje leta, generatora trajektorije te upravljačkog sustava bespilotne letjelice. Predstavljeni su osnovni kinematički i dinamički model višerotorske letjelice te njena upravljačka struktura koja se sastoji od estimatora stanja letjelice i kaskadnog kontrolera u dvije razine. Vanjska petlja kontrolera prima referentnu i estimiranu vrijednost pozicije te generira željene kuteve koji se koriste kao ulazni signali za unutarnju upravljačku petlju.

Budući da se u većini slučajeva kao senzor za prikupljanje podataka za inspekciju koriste različiti tipovi kamera, u ovom poglavlju je dan i model projekcije kamere kojim se definira veza između objekata u stvarnom svijetu i njihovog prikaza na slici. Uz to, definirane su i metode za odabir parametara inspekcije (maksimalni razmak među točkama putanje i željena horizontalna udaljenost bespilotne letjelice od objekta za vrijeme leta) prema zahtjevima za područje preklapanja između susjednih slika te veličine oštećenja koje je moguće razlikovati u prikupljenim setovima podataka.

U četvrtom poglavlju predstavljene su dvije metode za planiranje putanje bespilotne letjelice korištene u zadacima inspekcije inspirirane Huygensovim principom širenja valova. Dva pristupa razlikuju se prema dostupnim podacima u vrijeme planiranja - prvi algoritam temelji se na 2D tlocrtu objekta, dok je drugi baziran na potpunom 3D modelu ispitivane strukture. Prilikom planiranje putanje korištenjem 2D tlocrta, algoritam generira niz kružnih valića koji se međusobno sijeku i čiji centri se nalaze duž ruba zgrade. Sve točke na novoj valnoj fronti imaju jednaku udaljenost od najbliže točke zgrade te se mogu koristiti za održavanje željene udaljenosti letjelice od bočnih dijelova zgrade prilikom leta. Uz to, algoritam osigurava okomito poravnanje senzora za prikupljanje inspekcijskih podataka cijelom duljinom planirane putanje leta.

Pristup temeljen na 3D modelima objekta generira ovojnicu oko strukture po kojoj se nakon toga odabiru točke kroz koje mora proći bespilotna letjelica tijekom prikupljanja podataka čime se osigurava konstantna udaljenost između nje i snimanog objekta. Izgradnja ovojnice započinje generiranjem međusobno jednoliko razmaknutih točaka na površini sfera čiji centri se nalaze na svakoj pojedinoj poziciji definiranoj oblakom točaka kojim je predstavljen objekt. Na temelju tih novih točaka na sferama generira se konkavna ovojnica koja služi kao bazni set svih pozicija u kojima se letjelica može nalaziti prilikom inspekcije. Odabirom relevantnih pozicija koje letjelica mora posjetiti te planiranjem putanje između njih isključivo kroz točne na ovojnici moguće je generirati sigurnu inspekcijsku putanju kojom se održava željena udaljenost od objekta i prilikom inspekcije kompleksnih struktura kao što su mostovi ili vjetroelektrane.

U petom poglavlju dan je detaljan prikaz razvoja dva nova algoritma za planiranje putanje leta na temelju zatvorenih izolinija umjetnih potencijalnih polja. Metoda je osmišljena da bude

primjenjiva u oba scenarija kojima su obuhvaćeni slučajevi kad je model objekta prethodno poznat ili kad je dostupna samo njegova približna lokacija. U slučaju kaj je model objekta dostupan planiranje cijele putanje se izvodi prije samog leta, dok se u drugom slučaju ono provodi u stvarnom vremenu za vrijeme leta po segmentima generiranih izolinija umjetnih potencijalnih polja na temelju prikupljenog oblaka točaka. Time je omogućeno dinamičko generiranje izolinija na temelju novih podataka te prilagodbe putanje leta novim situacijama kako bi letjelica efikasno navigirala oko strukture koju snima.

Metoda obuhvaća nekoliko ključnih faza, uključujući obradu ulaznih podataka primjenom filtra za vokselizaciju dobivenog oblaka točaka, generiranje potencijalnog polja korištenjem prethodno definirane funkcije te projekciju točaka izolinije na ravninu kojim je predstavljena trenutna horizontalna brzina letjelice. Time je omogućeno zadržavanje konstantnog smjera leta oko objekta na traženoj udaljenosti i visini. Nakon što letjelica obiđe cijeli objekt te time generira zatvorenu izoliniju potencijalnog polja ona prelazi na sljedeću zadanu visinu te ponavlja proces. Razvijeni algoritam testiran je nizom simulacija u realnim uvjetima s nepoznatim objektima različite kompleksnosti te postavljanjem različitih ograničenja na brzinu i akceleraciju letjelice prilikom eksperimenta.

U šestom poglavlju prikazan je novi algoritam praćenja planiranje putanje leta u nepoznatom području temeljen na modificiranoj verziji umjetnih potencijalnih polja. Njime je omogućena detekcija nepoznatih prepreka u stvarnom vremenu, njihovo izbjegavanje te povratak letjelice na originalnu putanju u prvu slobodnu točku nakon obilaska prepreke. Time se povećava sigurnost same letjelice tijekom izvođenja autonomne misije, dok se zadržava praćenje izvorne putanje u slobodnom prostoru čime je osigurana bolja kvaliteta prikupljenih podataka.

Odbojna potencijalna sila generira se na temelju detektiranog oblaka točaka koji predstavlja prepreke u prostoru te se sastoji od dvije komponente - normalne koja gura letjelicu od prepreke u okomitom smjeru na samu prepreku i rotacijske koja djeluje tangencijalno niz prepreku. Time se rješava problem koji može nastati kod primjene potencijalnih polja za navigaciju letjelice a to je da ona ostane zarobljena u točkama lokalnih minimuma prije nego dođe do cilja. Temeljeno na gustoći detektiranih točaka, generirani oblak točaka je podijeljen u više grupa sa zasebnim težištima. Svaka grupa predstavlja pojedinu prepreku čime je omogućen prolazak letjelice između prepreka ako između njih postoji dovoljan razmak za siguran let.

Privlačna potencijalna sila generira se u najbližoj slobodnoj točki prepreke na originalno planiranoj putanji te na samom vrhu prepreke čime se minimizira generirano odstupanje od početne putanje prilikom obilaska prepreke.

Uz to, algoritam određuje je li moguće sigurno stići u pojedinu točku duž izvorne putanje te ukoliko se utvrdi da se točka nalazi preblizu prepreci ona se odbacuje, a letjelica kreće prema prvoj sljedećoj koja je slobodna. Time se povećava vrijeme leta koje letjelica provodi na

originalnoj putanji što pozitivno utječe na kvalitetu same inspekcije strukture bez nepotrebnog povećanja rizika za sigurnost same letjelice. Nizom simulacija u prostorima s različitim tipovima prepreka detaljno je analiziran utjecaj svake od četiri potencijalne sile na ponašanje letjelice te je primijenjenost algoritma potvrđena laboratorijskim eksperimentima.

Glavni znanstveni doprinos ove disertacije je:

- *Planiranje putanje leta za inspekciju nepoznate infrastrukture zasnovano na zatvorenim izolinijama umjetnih potencijalnih polja*
- *Metoda za slijeđenje putanje leta u nepoznatoj okolini zasnovana na umjetnim potencijalnim poljima*

Ključne riječi: bespilotna letjelica, inspekcija infrastrukture, umjetna potencijalna polja, planiranje putanje leta, izbjegavanje prepreka

Contents

CHAPTER 1

Introduction

The Unmanned Aerial Vehicles (UAVs) have gained significant attention within both the scientific community and the general public, thanks to their exceptional versatility, agility, and robust capabilities. These aerial platforms offer innovative solutions to an extensive array of challenges across various scientific disciplines, industries, and everyday applications. Among the wide range of UAV platforms available, multirotors have emerged as the most common choice due to their unique features, including vertical take-off and landing (VTOL) capability, exceptional maneuverability, simplicity of construction, and ease of maintenance [1].

The development of microelectronics and sensor technology, coupled with advancements in UAV control structures, has significantly increased the capabilities of UAVs to perform complex tasks with a high degree of autonomy. This has opened up opportunities for the use of UAVs in a wide range of of civil sectors [2]. These applications span a wide array of fields, including area surveillance [3], agriculture monitoring [4], search and rescue missions [5], terrain mapping [6], archaeology [7], and infrastructure inspection [8].

Given the need for regular inspections of infrastructure objects due to the harsh environmental conditions and high operational demands they often endure, the deployment of UAV systems, whether pilot-assisted or fully autonomous, has become increasingly common. These systems offer a compelling alternative to traditional inspection methods, reducing the risks posed to human operators while enhancing inspection accuracy and efficiency. Their ability to access hard-to-reach areas with ease, mitigating the need for costly and time-consuming setups of additional equipment, not only enhances the safety of inspection operations but also minimizes disruptions to the normal functioning of infrastructure assets. Already, UAV systems are being employed in the inspection of critical infrastructure such as wind turbines [9,10], power lines [11,12], industrial plants [13,14] and bridges [15,16,17]. Equipped with advanced

sensor payloads, UAVs are capable of capturing high-resolution images, Light Detection And Ranging (LiDAR) scans, thermal data, and multispectral datasets, among other data types. This acquired information allows inspectors and engineers to conduct detailed analyses of infrastructure components, identifying defects, deterioration, and potential hazards with great accuracy. Furthermore, the integration of UAVs with automated data processing algorithms, such as machine learning and computer vision, enables real-time analysis of great amounts of inspection data, allowing for rapid maintenance planning and prioritization of repairs, thereby extending the lifespan of infrastructure objects.

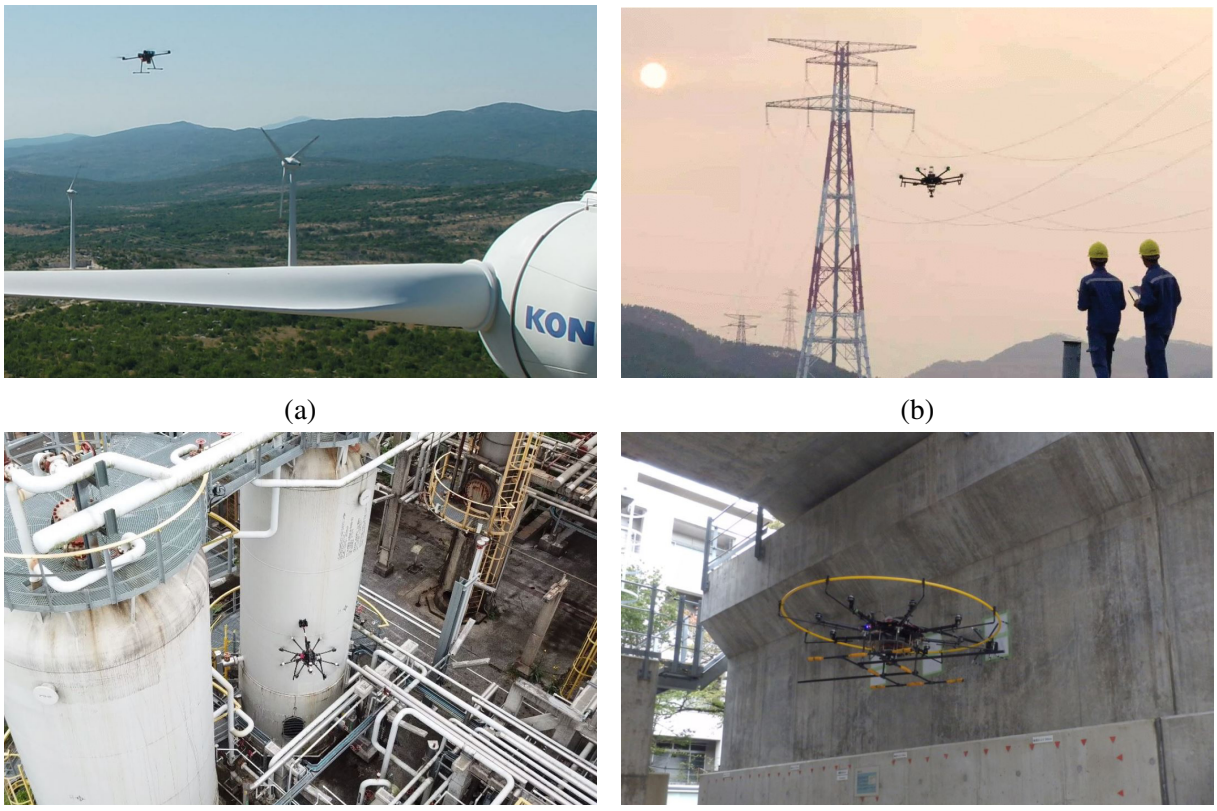


Figure 1.1: Images from the literature depicting UAV applications in infrastructure inspection. In (a), authors presented a LiDAR-equipped UAV utilized for pilot-assisted semi-autonomous wind turbine inspection [10]. In (b), a depiction of a UAV engaged in power line inspection [12]. (c) shows an octocopter during gas and oil plant inspection [14], and (d) shows a specialized UAV maintaining contact with a structure surface during bridge inspection [15].

Despite their numerous benefits, there are still challenges to widespread adoption of UAVs for infrastructure inspection. Regulatory constraints, airspace restrictions, as well as localization and navigation problems that UAVs encounter in GNSS-denied or high radio noise environments are just some of the obstacles in the development and deployment of autonomous inspection systems. Additionally, ensuring the reliability and accuracy of UAV-collected data remains an ongoing research effort that requires robust validation and verification processes to gain confidence in the results.

Central to the development of UAV-based autonomous infrastructure inspection systems is the integration of path planning [18,19,20] and collision avoidance [21,22] algorithms. These algorithms enable UAVs to autonomously navigate complex environments while adhering to predefined flight paths, avoiding obstacles, and ensuring the safety of both the aircraft and surrounding infrastructure. By utilizing real-time sensor data, UAVs can dynamically adjust their flight trajectories to account for changing environmental conditions, such as wind gusts, terrain variations, and unexpected obstacles.

Path planning algorithms play a crucial role in optimizing the efficiency and coverage of UAV inspection missions. By optimizing flight paths and avoiding redundant coverage areas, these algorithms enable UAVs to increase the amount of useful information gathered during each flight while minimizing energy consumption and mission duration. Moreover, path planning algorithms can incorporate specific constraints and objectives, such as maintaining a safe distance from sensitive infrastructure components, optimizing sensor coverage, and prioritizing areas of interest based on the operator's assessment of critical segments of the structure and inspection schedules.

In addition to path planning, collision avoidance algorithms are essential for ensuring the safe operation of UAVs in dynamic and unknown environments. By continuously gathering information about the surrounding environment, detecting potential obstacles, and executing evasive maneuvers when necessary, these algorithms mitigate the risk of collisions. Furthermore, collision avoidance algorithms can employ a sophisticated sensor fusion techniques [23], such as combining data from onboard cameras, LiDAR sensors, and radar systems, to enhance situational awareness and enable robust decision-making in real-time.

1.1Contributions

Original scientific contribution of this thesis is:

- *Inspection flight path planning for unknown infrastructures based on closed isolines of artificial potential fields* (Chapters4and5)
- *Flight path tracking method in an unknown environment based on artificial potential fields* (Chapter6)

1.2Thesis Outline

This thesis is structured into seven chapters, outlined as follows:

Chapter1: This chapter provides a broad introduction to the domain of UAVs in autonomous inspection missions, highlighting the contributions of the research and presenting an overview

of the thesis.

Chapter2: Here, a comprehensive review of the state-of-the-art in UAV path planning, navigation algorithms utilizing artificial potential fields, and infrastructure inspection methodologies is presented. Additionally, relevant prior work from various projects is discussed.

Chapter3: This chapter offers an overview of the proposed system for autonomous infrastructure inspection using UAVs. It includes discussions on fundamental kinematic and dynamic models of quadcopters, the general control structure of UAV systems, camera projection models, methodologies for determining inspection parameters, and introduces UAV platforms and sensors employed in the research.

Chapter4: In this chapter, two inspection path planning algorithms based on Huygens's wave propagation principle and construction of path planning envelope are introduced for infrastructure objects using either their 2D layout or 3D model.

Chapter5: This chapter presents two path planning algorithms based on closed isolines of artificial potential fields. The first algorithm facilitates offline path planning for objects with known layouts or models, while the second algorithm enables real-time planning for UAVs simultaneously scanning the side of unknown structures and generating new waypoints to traverse through.

Chapter6: Here, the thesis explores path tracking and real-time collision avoidance algorithms based on artificial potential fields for UAVs operating in unknown environments. The chapter provides detailed mathematical formulations of potential forces and their interactions, alongside simulation and real-world experiment results.

Chapter7: Finally, the thesis concludes with a summary of the presented work.

CHAPTER 2

State of the art and related work

2.1Path planning

Path planning is a critical aspect of autonomous UAV flight, alongside control and localization. It involves determining a sequence of waypoints in three-dimensional space that the UAV must navigate through during its mission. Generating feasible, collision-free, and time or energy optimized flight paths for autonomous vehicles is a significant area of research, as highlighted in recent studies [18]. Path planning techniques can be divided into two categories: sampling-based and artificial intelligence methods.

Among sampling-based planning techniques, the roadmap method is widely utilized. This approach involves constructing a network of feasible paths within the configuration space of the environment, enabling efficient trajectory generation for UAV navigation. Roadmap-based algorithms typically consist of two main steps: roadmap construction and query resolution. During roadmap construction, configurations are sampled from the free space of the environment and validated to ensure collision avoidance. Valid configurations are then connected based on proximity, forming a network of feasible paths. Query resolution involves searching for the shortest collision-free path between a start and a goal configuration on the constructed roadmap.

The Probabilistic Roadmap (PRM) consists of randomly sampled nodes connected with obstacle-free straight lines [24]. Another well-known algorithm based on roadmaps is the A* Algorithm, which computes a path based on the cost from the current node to both the starting and goal points, guaranteeing an optimal solution. Various modifications and enhancements of the A* Algorithm have been proposed in the literature for UAV flight path planning [25,26]. The Voronoi diagram (VD) is another tool utilized in path planning [27]. It divides the plane into regions based on proximity to a set of points, with each node in the diagram equidistant from

all points in the set representing the locations of obstacles. Paths generated using the Voronoi diagram are considered highly safe, as they ensure that obstacles are kept as far away as possible from all edges of the path, minimizing the risk of collision. Among the most popular roadmap techniques is Rapidly-exploring Random Trees (RRT), which constructs a tree-like structure from random nodes, starting from the initial node and searching for a connections towards the goal node. RRT is used to construct an efficient path for UAVs by the dense covering of free configuration space [28]. Several improvements of the original method have been proposed, including RRT-Connect, a more greedy variant used for collision avoidance among multiple UAVs [29], and RRT*, which optimizes generated path within a designated timeframe, rather than accepting the first obstacle-free path found [30].

Another sampling-based planning approach is cell decomposition, where the configuration space is divided into free and occupied cells, enabling easy computation of safe paths between points in the same cell or adjacent cells. Example of such method is presented in [31].

This thesis adopts a third approach to path planning, utilizing artificial potential fields to represent the environment. The subsequent section provides an extensive literature review outlining the various advantages, drawbacks, and modifications of this methodology.

2.2Potential fields in path planning and obstacle avoidance

The Artificial Potential Field (APF) method is widely used technique in both path planning and real-time collision avoidance for autonomous robots. The method is based on the concept of a repulsive and attractive forces that repels a robot from an obstacle while simultaneously drawing it towards a designated target. Initially devised as a solution for high-level planning issues in collision avoidance for robotic arms [32], the APF method has since undergone extensive adaptations for various applications. Due to the simple mathematical background and low computational complexity of the algorithm, it is suitable for implementation across diverse robotic platforms, including mobile robots [33], underwater [34], and aerial autonomous vehicles [35], as well as for controlling swarm formations [36,37] However, despite its widespread applicability, the APF method is primarily utilized in published literature for reaching target points rather than following predefined trajectories or exploration of the space.

The APF method does present several notable limitations, including issues such as local minima, oscillations within narrow corridors, jitter problems, and unreachable targets [38,39, 40]. These challenges arise when the sum of attractive and repulsive force vectors diminishes to a point where it becomes too small to generate any significant displacement of the robot before it reaches the target. To address these shortcomings, numerous enhanced approaches based on the APF method have been proposed for autonomous mobile robot path planning. These range from modifying or optimizing potential field functions to introducing external disturbances in

the form of additional forces or generating virtual obstacles and goal points.

In [35], authors addressed the non-reachable goal problem, which arises when the goal point does not coincide with the global minimum of the total potential field, meaning that when UAV should reach the target the attractive potential becomes zero, but repulsive potential from the obstacle in the vicinity of the goal point still exists. They proposed incorporating a relative distance factor between the UAV and the target into the repulsive function, effectively reducing repulsive force as the UAV approaches the goal. Additionally, the UAV's velocity term was integrated into the attractive potential function ensuring that target is considered reached only when the UAV's position aligns with the goal and its velocity reaches zero. In [41] authors are considering both the size of the robot and obstacles when adapting the weight of the Gaussian-like repulsive potential field function to facilitate the robot's escape from local minima.

As another solution for the problem of local minima authors proposed adding a rotational component to the repulsive potential field, as seen in [42] for single UAV scenarios and [43] for multi-UAV formation control. This rotational force acts perpendicular to the normal component of the repulsive force and activates when the algorithm detects the UAV's entrapment in a local minimum.

In cases when the artificial potential field method is utilized for multi-UAV formation control [37,44], in addition to attractive potential from goal points and repulsive potential from obstacles, potentials between agents in the system are introduced. These potentials ensure agents maintain desired distances and angles with respect to each other (or with respect to the swarm leader), functioning as both attractive or repulsive forces based on the difference between current and desired inter-agent distances. In [37], researchers presented a multi-agent control system utilizing a bell-shaped potential field function, along with an analysis of its stability properties. The facts that the form of elementary function is the same for both repulsive and attractive potential and defined by a single parameter, facilitated the synthesis of an adaptation algorithm ensuring agent arrival at the required formation to both static or moving targets. In [44] a virtual core of the UAV cluster is introduced with defined attractive force exerted upon the UAVs in the group ensuring that they can maintain the trend of convergence and inter-UAV repulsion force that drives UAVs in the same group to separate from each other to avoid collision between them.

In several studies, the introduction of virtual sources of potential fields was proposed to address local minima problems when encountering concave obstacles. In [44] and [45] authors proposed creating a virtual obstacle in the point of local minimum where UAV is trapped. This creates a new potential field that forces the UAV out of the local minimum and allows it to continue the movement towards a goal point. This process can be iteratively repeated when algorithm detects that UAV is trapped in the local minimum created by the environment. Authors in [40] presented similar idea, but instead of creating virtual obstacle they proposed generation

of the new virtual waypoint that attracts the UAV out of the local minimum trap.

Researchers in [46] discuss interesting case where artificial potential field is used for UAV to simultaneously avoid obstacles and follow and ground mobile robot. They proposed dividing the both attractive and repulsive potential forces into sub forces that are independently calculated and tuned based on the direction (Z axis is treated separately form forces in horizontal plane), position and velocity of the UAV.

In [47] authors provide a solution to the 2D collision fee path planning problem for an autonomous mobile robot utilizing an electrostatic potential field developed through a resistor network in unknown environment. Alongside this they also provide a stability analysis of the system using laws of electrostatic fields in terms that proposed approach generates an approximately optimal path in a real-time, as well as that a derived potential function can be considered a global navigation function free of all local minima.

Another approach, presented in [48] of working on APF method utilizes a global path planning method to find a desired path to the goal and concepts of APF methods for local path planning in highly dynamic environment. In order to improve the safety of the vehicle, the repulsive potential fields around moving obstacles are calculated with stochastic reachable sets. In [49], a large number of successful real-world experiments are reported using the global planner, which is based on the implementation of a Laplace equation that generates a potential function with a unique minimum at the target, while the local planner uses a modification of the conventional potential field method that takes into account the relative angles between the UAV and the target and obstacles.

2.3Inspection task using UAV

The rapid development of UAVs has significantly enhanced their capabilities to execute complex tasks with high degree of precision and autonomy, establishing them as indispensable tools in civil engineering for infrastructure inspection. Aerial platforms offer numerous advantages over conventional inspection methodologies, primarily because of their ability to access remote or hazardous locations effortlessly.

The efficacy of UAV-based inspections is based in the integration of advanced sensor technologies, such as LiDAR sensors, thermal or multispectral cameras. LiDAR systems provide precise three-dimensional mapping of infrastructure components, enabling accurate assessments of structural integrity and spatial configurations. In addition to standard cameras, which provide high-resolution images for surface anomaly detection, thermal and multispectral cameras capture detailed imagery across different wavelengths, enabling engineers to analyze material composition, identify defects, and monitor environmental factors such as corrosion, spreading of vegetation, or occurrence of water damages.

A significant number of UAV-based systems and frameworks have already been developed for civil infrastructure inspection and monitoring, with comprehensive evaluations documented in the literature [50], [51]. Noteworthy advancements include the development of a UAV system for autonomous inspection of enclosed industrial facilities in GPS-denied environments, utilizing a lightweight, vision-aided inertial navigation system [52]. Bridges represent a particularly interesting class of infrastructure for the deployment of UAV-based inspection methodologies, given their diverse and complex geometries, as well as sets of different components requiring distinct inspection approaches and data collection methods. In [53], researchers have introduced a UAV-based system employing image stitching and processing techniques for crack detection in concrete bridges. In [54], the significance of three-dimensional path planning for UAVs equipped with LiDAR sensors is emphasized to efficiently navigate and ensure coverage of potential defect locations on bridges. To address this, a Genetic Algorithm (GA) and A* algorithm were integrated to solve the Traveling Salesman Problem (TSP), optimizing flight time while maximizing visibility and ensuring perpendicular and overlapping viewpoints for comprehensive sampling. Researchers in [55] conducted laboratory and on-site experiments to measure plane horizontal displacement using UAVs, with results exhibiting excellent agreement with data obtained via standard Linear Variable Differential Transformer (LVDT) sensors, both in frequency and time domains. To measure the dynamic properties of bridge structures using UAV authors in [56] combined the convolution neural network (CNN) and the optical flow Kanade-Lucas-Tomas methods, highlighting the potential of advanced computational approaches in enhancing structural assessment methodologies.

Noteworthy innovations have been achieved in autonomous monitoring systems for electrical power infrastructure, traditionally reliant on optical sensors for power line detection in UAV-captured data [57], now enhanced through neural network implementation. In [58], an end-to-end convolutional neural network (CNN) was employed to extract complementary information from multilevel features and detect power lines with varying pixel widths and orientations, while in [59], power line detection and segmentation were based on transfer learning and an enhanced Mask Regional Convolutional Neural Network (Mask RCNN). Furthermore, authors in [60] went beyond visual inspection of power lines by developing and deploying a UAV-integrated tool for insulator cleaning and maintenance, accompanied by an autonomous system for landing on a charging pad.

In light of the increased interest in renewable energy sources, considerable research efforts are currently directed towards the development of autonomous systems designed for the automatic detection of damages in wind turbines. This focus arises from the inherent risks posed to human operators during inspection procedures, coupled with the considerable costs associated with maintenance downtime [61], [62]. In [10], a semi-autonomous method for wind turbine inspection utilizing LiDAR-equipped UAV was introduced. The UAV traverses parallel to the

blade tip and back, maintaining a constant relative distance and heading to the blade plane, thereby efficiently collecting visual dataset of wind turbine condition. In [63], the authors proposed an intelligent aerial manipulator capable not only of detecting damages but also executing repairs autonomously. Furthermore, the complexity of inspecting entire offshore wind farms requires a comprehensive approach, as explored in [64]. This study deals with the problem of employing multiple UAVs for inspection, requiring sophisticated optimization techniques for placement and routing to ensure thorough coverage and efficiency across expansive offshore wind installations.

2.3.1Residential structure inspection

Unmanned aerial vehicles, equipped with advanced LiDAR sensors and sophisticated camera systems, present a state-of-the-art approach for data acquisition concerning residential or historical buildings, improving architectural planning through the development of highly precise 3D models. This methodology enables the generation of dense point clouds, capturing detailed features of building exteriors, interiors, and surrounding environments with high accuracy and resolution. The integration of high-resolution visual data obtained from cameras mounted on UAV with LiDAR datasets further enhances the modeling process, allowing for the inclusion of detailed surface textures and the creation of comprehensive 3D representations. These models serve as assets for architectural design, renovation, and urban development projects.

As part of the ENCORE (ENergy-aware BIM Cloud Platform in a COst-effective Building REnovation Context) project [65], multiple surveys were conducted on model residential buildings, aiming to produce accurate and comprehensive 3D models to serve as the foundation for subsequent phases of architectural planning. This concept was planned to be adapted for use in historical buildings across Europe for architectural reconstruction.

The results from a specific experiment involving the utilization of UAVs in autonomous inspection missions of model residential buildings are illustrated in Figure2.1. Flight paths for the UAVs were planned based on satellite images of the structures, utilizing an algorithm based on Huygens's wave propagation principle, as detailed in Section4.1.

2.3.2Wind turbine inspection

Due to the frequently harsh environmental conditions in which they operate, both on-shore and off-shore wind turbines require regular blade inspections to assess for damages and ensure structural integrity. Using the UAVs for these tasks can significantly enhance the efficiency and safety of maintenance operations while also reducing costs. By providing high-resolution images, UAVs eliminate the need to endanger human operators or cause prolonged periods of turbine inactivity.

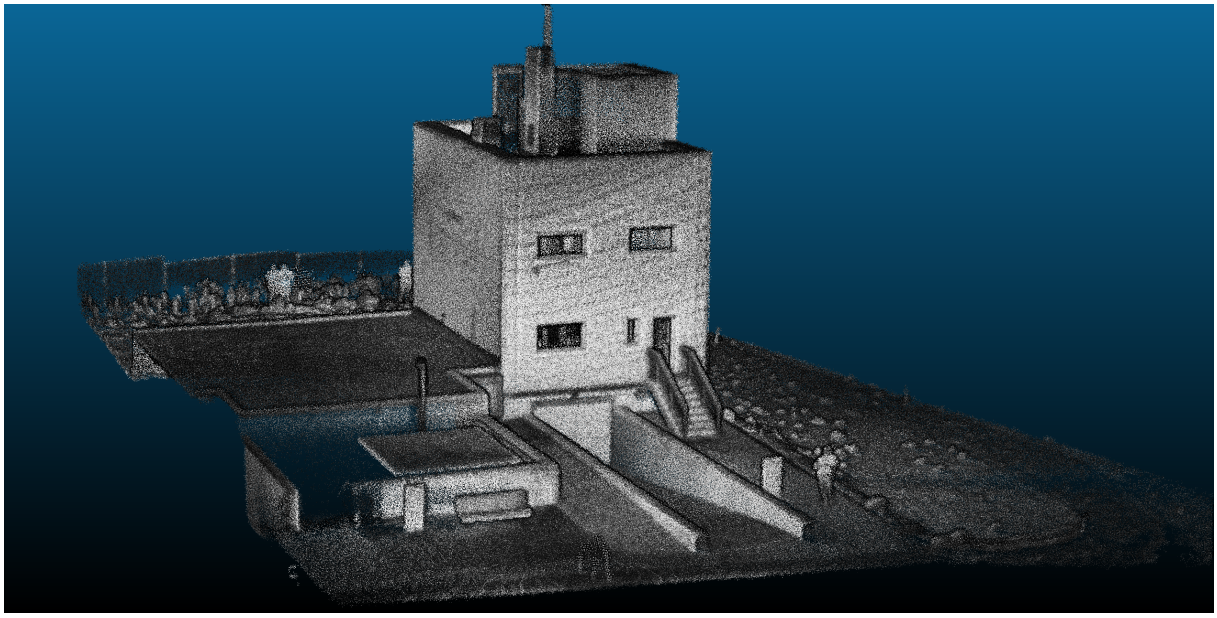


Figure 2.1: Resultant 3D model of residential building generated by UAV using a LiDAR sensor.

Wind turbines pose a unique challenge for autonomous UAV inspections due to their complex geometry and dynamic characteristics. Their complicated structure, combined with the absence of discernible environmental landmarks around them, presents significant hurdles for localization and navigation of UAVs assigned to inspect them. Even during inspections of stationary wind turbines, the precise orientation of the nacelle and the positioning of the blades cannot be reliably predicted in the global reference frame due to their adjustments aimed at minimizing stress induced by varying wind directions. Furthermore, the lack of distinctive features and the limited surface area available for sensor data acquisition, especially at the edges or tips of the blades, can compromise the reliability of conventional navigation methods relying solely on single GPS, camera, or LiDAR sensors.

A viable solution to these challenges involves integrating high-precision GPS systems with LiDAR sensors using advanced path planning algorithms and trajectory generators. The path and trajectory planning methods can be utilized to generate flight trajectories based on the generalised model of the wind turbine before flight. By incorporating data from both GPS and LiDAR sensors, these trajectories can then be dynamically adjusted to accommodate the actual position and orientation of the turbine in real-time.

This approach, developed during the *Autonomous UAV Inspection of Wind Turbine Blades* (AEROWIND) project [66], enhances the UAV's capability to navigate effectively around wind turbines and collect accurate inspection data. The UAV conducting autonomous inspections of stationary wind turbines, along with the resulting images acquired during the survey mission, are illustrated in Figure2.2. Detailed descriptions of the path planning algorithm and both simulation and real-world experiments conducted to validate it are provided in Section4.2.

Figure 2.2: Autonomous wind turbine inspection conducted by an UAV (left). High-resolution images acquired during the survey mission that are used for damage assessment on the wind turbine blades (right).

2.3.3Bridge inspection

Bridges, exposed to environmental influences and heavy traffic, require periodic monitoring due to material degradation over time. Additionally, following extraordinary events where damage is suspected or confirmed, timely and accurate damage assessment is essential for evaluating the condition of the bridge.

The inspection process often requires visual data for detecting cracks or corrosion in inaccessible areas. Using the UAVs provides a solution to enhance data acquisition efficiency without the need for complex equipment setups that could disrupt traffic or endanger human operators. However, visual data alone may not always suffice for a comprehensive assessment of bridge condition, so it is often required to obtain additional measurements such as frequency and intensity of vibrations in different bridge components.

Dynamic tests of suspension bridge structures are conducted to determine structural dynamic parameters, including frequencies and mode shapes. Changes in these parameters may indicate potential structural damage of the bridge components. Regular surveys of hanger cable tension involve measuring vibration frequencies using accelerometers attached to each cable when external force is applied. The axial force in the cable can be determined based on the theory of wire oscillation, using experimentally determined natural frequencies along with parameters such as cross-section, length, and material density.

To expedite this process and avoid the time-consuming determination of cable natural frequencies through experimental methods, a novel approach was developed within the Autonomous System for Assessment and Prediction of Infrastructure Integrity (ASAP) project [67]. This method, presented in [68], utilizes precise UAV positioning relative to the bridge to capture high-quality video from optimal distances and angles without disrupting bridge operations.

The algorithm estimates natural frequencies of vibrating cables by analyzing differences in cable positions between frames in the video. Straight lines representing cable edges are detected

in each frame, and the displacement of pixels constructing the line between consecutive frames is calculated. Filters based on predefined parameters, including region of interest, maximum displacement, and line slope, aid in accurate line detection. Specific regions of interest capture induced cable vibrations and static reference features. The sequence of pixel position differences across consecutive frames is then subjected to fast Fourier transform analysis to estimate vibration frequencies in the cable.

The developed method was extensively tested and validated through a series of simulations conducted in both 2D and 3D environments, laboratory experiments, and real-world case studies. Various levels of noise and disturbances were introduced during simulations, while different static features in video frames or sections of the oscillating cable were utilized in estimating vibration frequency during real-world experiments. In each instance, the obtained results were compared to those from accelerometers attached to the cables, thus confirming the algorithm's ability to accurately estimate vibration frequency. The results from the case study are illustrated in Figure2.3and summarized in Table2.1.

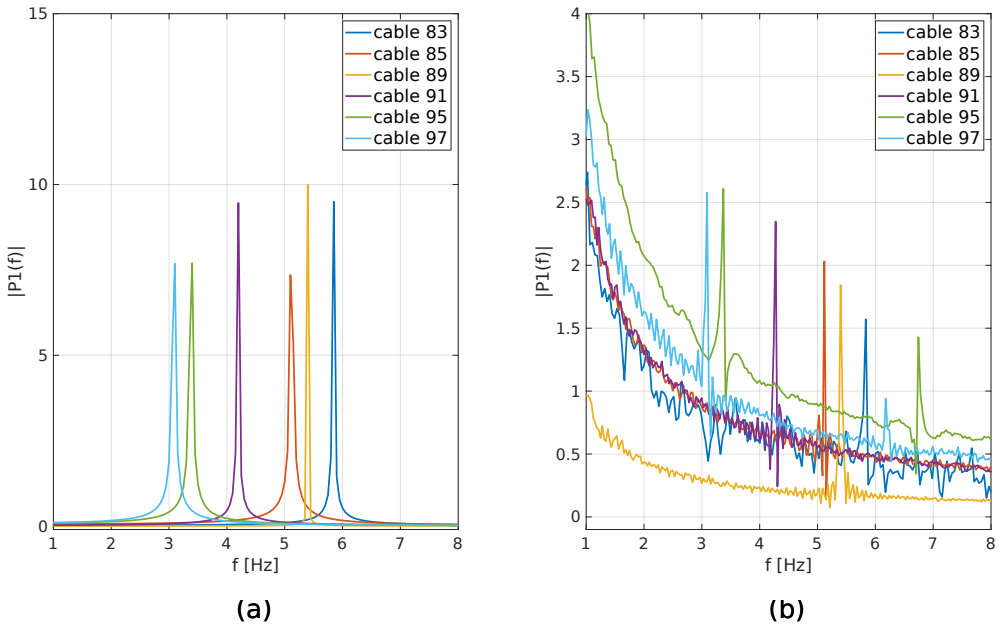


Figure 2.3: Power spectrum density function of signals measured by accelerometer attached to the cable (a) and by processing UAV captured video of cables oscillation (b)

Table 2.1: Comparison of natural frequency values of cables measured by traditional accelerometer (f^{acc}) and obtained by processing the video recorded by the UAV (f^{uav})

Cable number	(f^{acc}) $[Hz]$	(f^{uav}) $[Hz]$	$\lvert f^{acc} - f^{uav} \rvert$ $[Hz]$
83	5.84	5.90	0.04
85	5.12	5.19	0.07
89	5.40	5.40	0.00
91	4.29	4.29	0.00
95	3.38	3.37	0.04
97	3.08	3.09	0.01

CHAPTER 3

System overview

The primary objective of this research is the development of a comprehensive and robust system for autonomous inspection of infrastructure objects utilizing various UAV platforms. The proposed system aims to be adaptable to a wide range of infrastructure objects, varying in complexity and specific geometries. The planned flight paths and trajectories must ensure safety and feasibility in real-world conditions while providing high-quality data with complete coverage of surveyed objects. All parameters relevant to the inspection mission, including preferred distances from the sides of the object, flight altitudes, and UAV velocity and acceleration constraints, need to be integrated into the path planners or trajectory generators to ensure the safety and quality of autonomous inspection missions. General system architecture is depicted in Fig. 3.1.

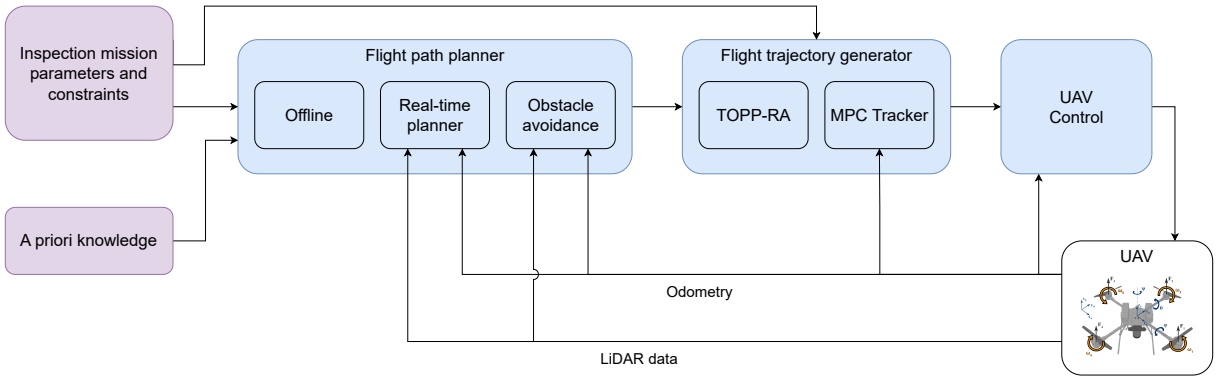


Figure 3.1: Architecture of the developed system for autonomous infrastructure inspections.

The path planning modules should accommodate different types of input data and levels of knowledge about surveyed structures, ranging from simple location information to detailed 2D layouts or complete 3D models. To ensure the broad applicability of the developed system, several methods and algorithms have been proposed based on a priori knowledge. In Section4.1,

a 2D path planning algorithm based on Huygens's wave propagation principle was presented, requiring a known layout of the building or satellite imagery for extrapolation. Another path planning algorithm, detailed in Section4.2, is utilized for generation of inspection paths for UAVs based on three-dimensional envelopes constructed around the structure model. Furthermore, a path planning algorithm based on artificial potential fields was presented in Section5.2 that is capable of working with either 2D layouts or 3D models of the structure as input data. These path planners are considered offline algorithms, generating a list of waypoints for the UAV to follow before actual flight.

To enable the system's usability in scenarios where only the location of the surveyed structure is available, a real-time inspection path planner based on the closed isolines of artificial potential field was developed. This planner generates segments of the inspection path during the flight for the UAV to traverse based on sensor measurements while considering inspection parameters and constraints. Depending on the object's complexity, the sensors employed on the UAV, and the inspection requirements, the acquired data from these surveys can serve as full inspection datasets or as bases for constructing models for more detailed inspections or different configurations of flight paths used in offline planners.

Since all planners only consider the target structure object in generating flight paths and not the entire environment, the system needed a reliable collision avoidance algorithm to be incorporated into the trajectory tracker for real-world usability. Given the uncertainty or complete lack of reliable data about the environment where inspection missions are executed, the proposed obstacle avoidance algorithm had to be robust and capable of dealing with a wide variety of unknown obstacles. The developed algorithm, based on artificial potential fields, not only provides a means to detect obstacles online and execute avoidance maneuvers but also minimizes deviations from the initially planned trajectory, ensuring that the UAV follows it in unobstructed space.

Inspection trajectories for the UAVs are generated based on the planned paths by incorporating velocity and acceleration constraints through the Time Optimal Path Parametrization by Reachability Analysis (TOPPRA) algorithm [69], or by utilizing the Model Predictive Control (MPC) tracker module detailed in Section3.1.1.

3.1UAV model and control

Although the primary focus of this work revolves around path planning algorithms and trajectory tracking methods rather than the control theory of UAVs, it is essential to comprehend the fundamental kinematic and dynamic models of the quadcopter, along with the general control structure employed in UAV systems. Furthermore, understanding the relationship between data-collecting sensors and the UAV's behavior during the inspection mission is crucial. For

more complex models of aerial manipulators and control structures, comprehensive details can be found in [70].

The kinematic model defines the position and orientation of the UAV in the desired coordinate system, while the dynamic model describes the influence of forces and torques produced by propellers, as well as external influences on the system. The UAV consists of a central body and four rotors positioned symmetrically at a distance l from the body (shown in Fig.3.2), generating the thrust and torques necessary for flight. The position and orientation of the UAV's body frame B (moving coordinate system) in the world reference frame W (inertial coordinate system) are defined as follows:

$$\mathbf{p}_W^B = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{\Theta}_W^B = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{3.1}$$

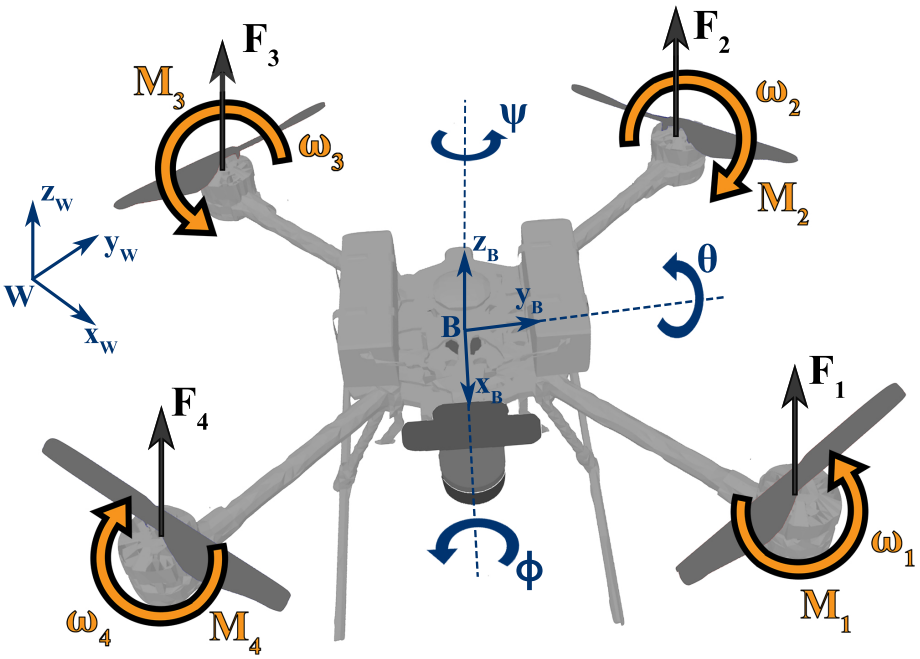


Figure 3.2: Model of the UAV

Using Euler angles as a representation for a UAV's orientation, a rotation matrix from the body frame B to the world frame W can be composed from three elementary rotation matrices. These matrices rotate the UAV body around the z-axis for the yaw angle ψ, followed by rotation

around the y-axis for the pitch angle θ, and finally around the x-axis for the roll angle ϕ:

$$\mathbf{R}_W^B = \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi)$$

$$\mathbf{R}_W^B = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \tag{3.2}$$

with notation $S_x = \sin x$ and $C_x = \cos x$. By combining the translation components from equation 3.1and rotation matrix defined with equation3.2a homogeneous transformation matrix can be constructed:

$$\mathbf{T}_W^B = \begin{bmatrix} \mathbf{R}_W^B & \mathbf{p}_W^B \\ \mathbf{0}_3^T & 1 \end{bmatrix} \tag{3.3}$$

Each of the four rotors of the quadcopter exhibits an angular velocity denoted as ω_i, generating a force F_i and a moment M_i, as illustrated in Figure3.2and defined by the equations:

$$F_{r_i} = c_T \omega_{r_i}^2, \quad \tau_{r_i} = \pm c_d \omega_{r_i}^2 \tag{3.4}$$

Here, c_T represents the thrust coefficient and c_d the drag coefficient, with the sign $\pm$ indicating the direction of the propeller's rotation (clockwise or counterclockwise). Notably, while the thrust and drag coefficients may vary for each propeller, multirotor UAVs typically utilize identical motor-propeller pairs, allowing for the assumption that these coefficients remain constant. The Euler angles used to define the rotation matrix of the UAV in equation3.2are generated by the thrust difference between motors M_2 and M_4 for roll, and motors M_1 and M_3 for pitch. The yaw angle arises from the difference in moments between pairs of motors with opposite direction of rotation.

Utilizing the Newton-Euler equations of motion, the forces and torques acting on the UAV in the world reference frame can be determined. Since the z-axes of the UAV body and rotors are aligned, the forces and moments defined in equation3.4can be represented in the UAV's body reference frame B as follows:

$$\mathbf{f}_B^{r_i} = c_T \omega_{r_i}^2 \mathbf{z}_B, \quad \boldsymbol{\tau}_B^{r_i} = \pm c_D \omega_{r_i}^2 \mathbf{z}_B \tag{3.5}$$

Considering the gravitational force acting on the UAV body with mass m in the world reference

frame W, defined by the vector $\mathbf{g} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$, total force can be expressed as:

$$m\ddot{\mathbf{p}}_W^B = -m\mathbf{g} + \sum_{i=1}^{n_r} \mathbf{R}_B^W \mathbf{f}_B^{r_i} \tag{3.6}$$

where $\ddot{\mathbf{p}}_W^B$ denotes the linear acceleration of the UAV in the world frame.

The angular velocities of the UAV in the world frame can be defined by utilizing the UAV's orientation Θ_W^B from equation3.1:

$$\boldsymbol{\omega}_W^B = \dot{\boldsymbol{\Theta}}_W^B = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \tag{3.7}$$

while in body frame B they are are usually denoted as $\boldsymbol{\omega}_B = \begin{bmatrix} p & q & r \end{bmatrix}^T$. According to [71], angular velocities determined in the body frame of the UAV can be expressed in the world frame by utilizing the transformation matrix $\mathbf{W}_\omega$:

$$\boldsymbol{\omega}_W^B = \mathbf{W}_\omega \boldsymbol{\omega}_B = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\theta}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \boldsymbol{\omega}_B \tag{3.8}$$

The Newton-Euler equation for the angular moments acting on the UAV is given by:

$$\mathbf{I}_B \dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (\mathbf{I}_B \boldsymbol{\omega}_B) = \boldsymbol{\tau}_B \tag{3.9}$$

where I is the diagonal inertia matrix for a symmetric UAV, and $\boldsymbol{\tau}_B$ is a vector of external torque expressed as the sum of moments acting on the UAV body produced by rotors:

$$\boldsymbol{\tau}_B = \sum_{i=1}^{n_r} (\mathbf{p}_B^{r_i} \times \mathbf{f}_B^{r_i}) + \sum_{i=1}^{n_r} \boldsymbol{\tau}_B^{r_i} \tag{3.10}$$

Here, the first term represents the moment produced by the force acting at some distance from the UAV body, while the second term represents the moment produced by the propeller, as stated in equation3.5.

Most commonly used quadcopters operate as underactuated systems, featuring four control inputs represented by the angular velocities of the four rotors, denoted as ω_i. These inputs are utilized for controlling the six states of the UAV, as defined in Equation3.1. For a comprehensive understanding of other types of multirotor systems and their configurations, as well as a theoretical exploration of fully actuated UAVs, detailed descriptions can be found in [70]. Since the direct control of the angular velocity of rotors is highly impractical, even for achieving sta-

ble hovering flight, quadcopter control is typically implemented by transmitting desired angles and thrust values as control inputs. These control inputs, denoted by the vector $\mathbf{u}$, are defined as follows:

$$\mathbf{u} = \begin{bmatrix} u_\phi & u_\theta & u_\psi & u_z \end{bmatrix}^T \tag{3.11}$$

where u_ϕ, u_θ and u_ψ represent the outputs of the attitude controller for the moments around the body frame axes, while u_z is the height controller output represented as the net thrust of all rotors. The relation between the controller outputs and rotor angular velocities can be expressed as:

$$\mathbf{u} = \begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u_t \end{bmatrix} = \mathbf{\Gamma} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{3.12}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{4\times4}$ is the control allocation matrix that depends on the configuration of the UAV.

Two of the most widely used multirotor UAV configurations are quadcopters, with four rotors symmetrically positioned around the body center in either a standard *plus* (+) or *cross* ($\times$) configuration. In the + configuration, the body x and y axes align with two neighboring arms of the quadcopter, as depicted in Figure3.2, while in the $\times$ configuration, they point towards the middle between two rotors (this configuration is used for vehicles described in Section3.3). Assuming symmetric rotor placement at a distance l from the center of the body, the allocation matrices for both configurations are given by:

$$\mathbf{\Gamma}_+ = \begin{bmatrix} 0 & lc_T & 0 & -lc_T \\ -lc_T & 0 & lc_T & 0 \\ c_D & -c_D & c_D & -c_D \\ c_T & c_T & c_T & c_T \end{bmatrix}, \quad \mathbf{\Gamma}_\times = \begin{bmatrix} -\frac{\sqrt{2}}{2}lc_T & \frac{\sqrt{2}}{2}lc_T & \frac{\sqrt{2}}{2}lc_T & -\frac{\sqrt{2}}{2}lc_T \\ -\frac{\sqrt{2}}{2}lc_T & -\frac{\sqrt{2}}{2}lc_T & \frac{\sqrt{2}}{2}lc_T & \frac{\sqrt{2}}{2}lc_T \\ c_D & -c_D & c_D & -c_D \\ c_T & c_T & c_T & c_T \end{bmatrix} \tag{3.13}$$

The use of the allocation matrix ensures that position and attitude control strategies can remain consistent across different multirotor configurations, even in cases with different numbers of rotors. While specific controller parameters may require tuning for each configuration, the functional schemes do not need be altered.

Based on equation3.6for the forces acting on the UAV, the linear motion equation can be

formulated as:

$$m\ddot{\mathbf{p}}_W^B = -m\mathbf{g} + \mathbf{R}_B^W \mathbf{u}_1$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{u_z}{m} \begin{bmatrix} C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\phi S_\theta S_\psi - S_\phi C_\psi \\ C_\phi C_\theta \end{bmatrix} \tag{3.14}$$

where $\mathbf{u}_1 = \begin{bmatrix} 0 & 0 & u_z \end{bmatrix}^T$ represents the controller output, which can be mapped to the rotor's angular velocities through the allocation matrix Γ. The angular acceleration equation, described by the Newton-Euler equation for the angular moments acting on the UAV (equation3.9), is as follows:

$$\mathbf{I}_B \dot{\boldsymbol{\omega}}_B = -\boldsymbol{\omega}_B \times (\mathbf{I}_B \boldsymbol{\omega}_B) + \mathbf{u}_2 \tag{3.15}$$

Here $\mathbf{u}_2 = \begin{bmatrix} u_\phi & u_\theta & u_\psi \end{bmatrix}^T$ contains the moment inputs for the UAV. Considering the symmetry of the UAV, where the inertia matrix contains only diagonal elements $\mathbf{I}_B = diag(I_{xx}, I_{yy}, I_{zz})$, equation3.15can be expanded to:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{I_{yy} - I_{zz}}{I_{xx}} qr \\ \frac{I_{zz} - I_{xx}}{I_{yy}} pr \\ \frac{I_{xx} - I_{yy}}{I_{zz}} pq \end{bmatrix} + \begin{bmatrix} \frac{u_\phi}{I_{xx}} \\ \frac{u_\theta}{I_{yy}} \\ \frac{u_\psi}{I_{zz}} \end{bmatrix} \tag{3.16}$$

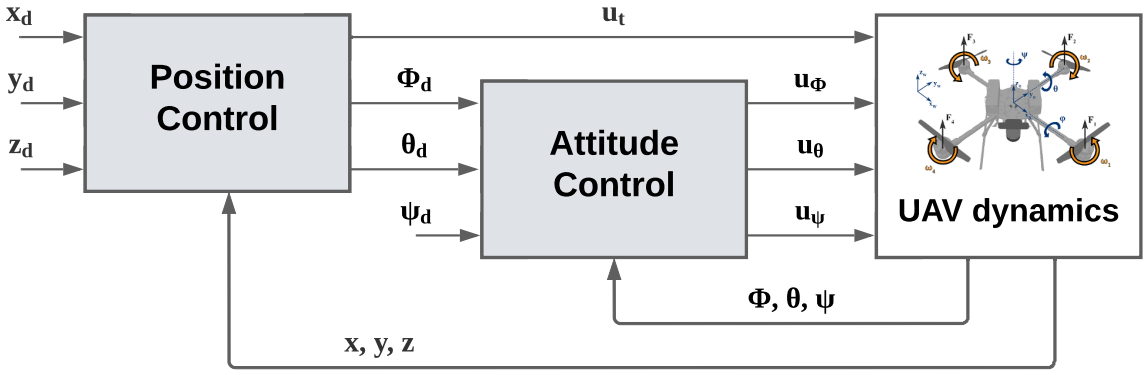


Figure 3.3: UAV cascade control structure with position controller (outer control loop) and attitude controller (inner control loop).

The control scheme for the UAV, defined by the equations for linear (3.14) and angular (3.16) accelerations, is illustrated in Figure3.3. This control structure consists of two PID controllers in a cascade configuration - the outer loop, comprising a position controller, and

the inner loop, containing an attitude controller. The position controller's inputs consist of the reference position vector $\mathbf{p}_d = \begin{bmatrix} x_d & y_d & z_d \end{bmatrix}^T$ and the UAV's estimated position. The outputs of the outer are the control signal u_z for the net thrust and the desired roll ϕ_d and pitch θ_d angles. In addition to the reference angles ϕ_d and θ_d generated by the position controller, the inputs to the attitude controller include the desired yaw angle ψ_d and the vector of estimated UAV angles. The inner loop's output comprises the control signals u_ϕ, u_θ, and u_ψ. This control structure allows position $\mathbf{p}_d$ and yaw angle ψ_d references for the system to be provided by the path planning or trajectory tracking algorithm, while the control inputs for motors are determined internally.

3.1.1MPC tracker

An MPC-based tracking method is chosen to generate UAV trajectory points along the corrected collision-free path obtained from the potential fields. The original implementation is presented in [72] while an adapted version of their work is used in this paper. The main motivation for using this tracking method is that it allows to quickly change and re-plan the UAV trajectory based on the current system state and model dynamics. Furthermore, the tracker enables safe and stable flight, regardless of the goal point resulting from the potential fields.

This tracking method employs a model predictive controller with a constant snap UAV model which controls a virtual UAV using snap commands. Snap commands are used as the input to the linear system, which predicts the next virtual UAV state based on the given model dynamics. The complete state of the virtual UAV is then sampled and used as a referent trajectory point for the real UAV at a rate of $100Hz$.

The definition of the MPC problem is as follows:

$$\begin{aligned} \min_{\mathbf{u}_0,\dots,\mathbf{u}_\mathrm{N}} \quad & \sum_{\mathrm{i}=0}^{\mathrm{N}} \left(\mathbf{e}_\mathrm{i}^\mathrm{T}\mathrm{Q}\mathbf{e}_\mathrm{i} + \mathbf{u}_\mathrm{i}^\mathrm{T}\mathrm{P}\mathbf{u}_\mathrm{i}\right) \\ \text{s.t.} \quad & \mathbf{x}_{\mathrm{k}+1} = \mathrm{A}\mathbf{x}_\mathrm{k} + \mathrm{B}\mathbf{u}_\mathrm{k}\,, \\ & \mathbf{x}_\mathrm{k} \leq \mathbf{x}_{max}\,, \\ & \mathbf{u}_\mathrm{k} \leq \mathbf{u}_{max}\,, \end{aligned} \tag{3.17}$$

where N is the horizon length. The error between the predicted virtual UAV state and the reference at the k-th horizon is defined as $\mathbf{e}_\mathrm{k} = \mathbf{x}_\mathrm{k} - \mathbf{r}_\mathrm{k}$. The state and input constraints are denoted by $\mathbf{x}_{max}$ and $\mathbf{u}_{max}$, respectively. The matrices A and B represent the well-known constant snap virtual UAV model. The weights Q and P are tuned for smooth trajectory generation with respect to the velocity and acceleration.

The CVXGEN solver is used to obtain the optimal snap input $\mathbf{u}_0^*$, which is used as an input to predict the next virtual UAV state and calculate the next referent trajectory point.

3.1.2Camera projection model

Understanding how the sensors utilized during inspection flights relate to the UAV's behavior can significantly enhance the quality and efficiency of missions. Unaccounted errors in UAV positioning or introduced sensor vibrations can greatly degrade data quality, leading to unfocused or blurred images of surveyed objects. Since inspection processes rely heavily on data captured by conventional high-resolution cameras or specialized thermal and multispectral imaging sensors, this section presents a general mathematical model for projecting a point in the world onto a two-dimensional image. This model incorporates both intrinsic and extrinsic camera parameters along with UAV dynamics.

According to the [73] projection of a point in the world with coordinates $\tilde{\mathbf{M}} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ to the 2D image pixel coordinates $\tilde{\mathbf{m}} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ can be defined with the equation:

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}} \tag{3.18}$$

Here, s is an arbitrary scale factor, and $\mathbf{P}$ is the perspective projection matrix of the camera, combining intrinsic and extrinsic parameters:

$$\mathbf{P} = \mathbf{A}\mathbf{P}_N\mathbf{D}_W^C \tag{3.19}$$

where $\mathbf{A}$ is camera intrinsic matrix containing with focal length f, axis scale parameters for pixel size k_u and k_v, skew angle α and optical center (the principal point) $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^T$:

$$\mathbf{A} = \begin{bmatrix} fk_u & fk_u \cot\alpha & u_0 \\ 0 & fk_v/\sin\alpha & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

normalization matrix $\mathbf{P}_N$:

$$\mathbf{P}_N = \begin{bmatrix} I_3 & \mathbf{0}_3 \end{bmatrix} \tag{3.21}$$

and extrinsic parameters matrix $\mathbf{D}_W^C$:

$$\mathbf{D}_W^C = \begin{bmatrix} \mathbf{R}_W^C & \mathbf{t}_W^C \\ \mathbf{0}_3^T & 1 \end{bmatrix} \tag{3.22}$$

The homogeneous transformation matrix $\mathbf{D}_W^C$ defines the camera reference frame C within the world frame W similar to how $\mathbf{T}_W^B$ defines the body frame of the UAV. If the camera is

rigidly connected to the UAV body, $\mathbf{D}_W^C$ can be expressed as $\mathbf{D}_W^C = \mathbf{T}_B^C \mathbf{T}_W^B$.

The homogeneous transformation matrix $\mathbf{D}_W^C$ defines the camera reference frame C within the world frame W in the same way as the matrix $\mathbf{T}_W^B$, defined with equation3.3, does for the body frame of the UAV. If the camera is connected to the AUV body by a rigid connection then by utilizing the static transformation $\mathbf{T}_B^C$ from the camera frame C to the UAV frame B matrix $\mathbf{D}_W^C$ can be expressed as:

$$\mathbf{D}_W^C = \mathbf{T}_B^C \mathbf{T}_W^B \tag{3.23}$$

In order to mitigate the vibrations caused by the imbalances in the construction or the propulsion system of the UAV, and enable independent control of the camera orientation it is often connected to the UAV by a gimbal with two or three degrees of freedom. By utilizing the 3-axis gimbal, rotation of the camera becomes decoupled from the rotation of the UAV. This means that in hover state, the rotation matrix $\mathbf{R}_W^C$ can be considered invariant over time and that the dynamics of the UAV are solely represented by changes in the translation vector $\mathbf{t}_W^C = \begin{bmatrix} x_c(t) & y_c(t) & z_c(t) \end{bmatrix}^T$. The projection equation3.18can then be written as follows:

$$s\tilde{\mathbf{m}} = \begin{bmatrix} X(r_{31}u_0 + fk_u r_{11}) + Y(r_{32}u_0 + fk_u r_{12}) + Z(r_{33}u_0 + fk_u r_{13}) + u_0(t_z(t)) + fk_u(x_c(t)) \\ X(r_{31}v_0 + fk_v r_{21}) + Y(r_{32}v_0 + fk_v r_{22}) + Z(r_{33}v_0 + fk_v r_{23}) + v_0(t_z(t)) + fk_v(y_c(t)) \\ Xr_{31} + Yr_{32} + Zr_{33} + z_c(t) \end{bmatrix} \tag{3.24}$$

where r_{ij} denotes a constant element of the rotation matrix $\mathbf{R}$.

During the development of the algorithm for estimation of the vibration frequency from the video captured by the UAV used in inspection of the suspended bridge, presented in [56], it was crucial to determine the amplitude and frequency range of the UAV vibrations. Since the external disturbances, such as wind effects and the position measurement errors generated by GPS, can be considered as a small amplitude white noise signal acting on the UAV system, which acts as a low-pass filter for the disturbance signal. The linearized model of the UAV based on the equations (3.14) and(3.16) is a second order filter, while the higher order of the overall system depends on the design of the position and attitude controllers. This enabled determination of the frequency range in which UAV vibrations affect captured video by performing spectral analysis of the position signal recorded during flight.

The signal of the estimated position of the UAV recorded during hover is illustrated in the Fig.3.4. Frequency domain analysis indicates that the oscillations of the UAV in the hover state are within the expected natural oscillation frequency range of the bridge cable. Therefore, these oscillations were were accounted for in the simulation validation of the algorithm by introducing a noise signal within this frequency range. The error between the desired and estimated position of the UAV is less than $0.2m$ throughout the flight. This amplitude is an order of magnitude

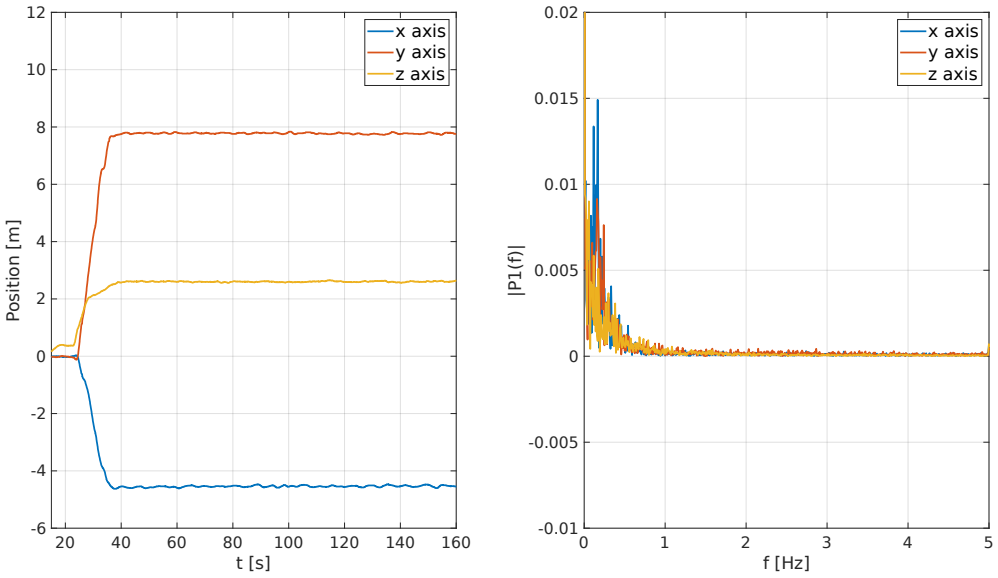


Figure 3.4: Estimated position of the UAV in the hover state (a) and power density of the estimated position signal in the frequency domain (b). Error between the desired and estimated position of the UAV in the hover state is less than 0.2 m throughout the flight. Frequency of the UAV oscillation is in the range of from 0 to 3 Hz which can be considered within the expected frequency range of the natural oscillations of the bridge cable.

smaller than the distance between the camera and the recorded objects, so this information can be used to select the amplitude of the noise signal in the simulations.

3.2Inspection parameters

In addition to generating safe and feasible flight trajectories for UAV systems engaged in autonomous infrastructure inspection, two specific criteria must be satisfied. Firstly, the system should produce quality inspection data, typically high-resolution images that can be utilized for assessing the structural condition and potentially identifying the presence and extent of damages. Secondly, it must ensure comprehensive coverage, particularly in areas where damages are likely to occur. This means ensuring that the UAV passes through every relevant viewpoint with precise heading, or alternatively, planning the orientation of the data acquisition sensor if it operates independently of the UAV's heading, such as a camera mounted on a gimbal with three degrees of freedom.

The typical configuration of the inspection path utilized in this work involves the UAV navigating around the surveyed target object, closely following its outer surface at a constant altitude while maintaining a predetermined distance from it. Upon completing a full pass around the object, the UAV adjusts its altitude to the next reference value and repeats the process.

Heading of the UAV is planned in every waypoint along the path in a way that inspection sensor maintains perpendicular orientation to the surface of the surveyed structure. However, certain exceptions to this pattern were made during experimental inspections of wind turbines due to their distinct geometries, which necessitated different trajectory designs in order to acquire all the relevant inspection data.

The quality of acquired images can be directly linked to the size of distinguishable features within the images, quantified by the spatial resolution of the image. The Ground Sampling Distance (GSD) principle from photogrammetry, which defines the distance between two consecutive pixel centers measured on the ground, is often utilized to calculate the distance (r_d) of the UAV from the surface of the surveyed object to meet the desired spatial resolution [74], [75].

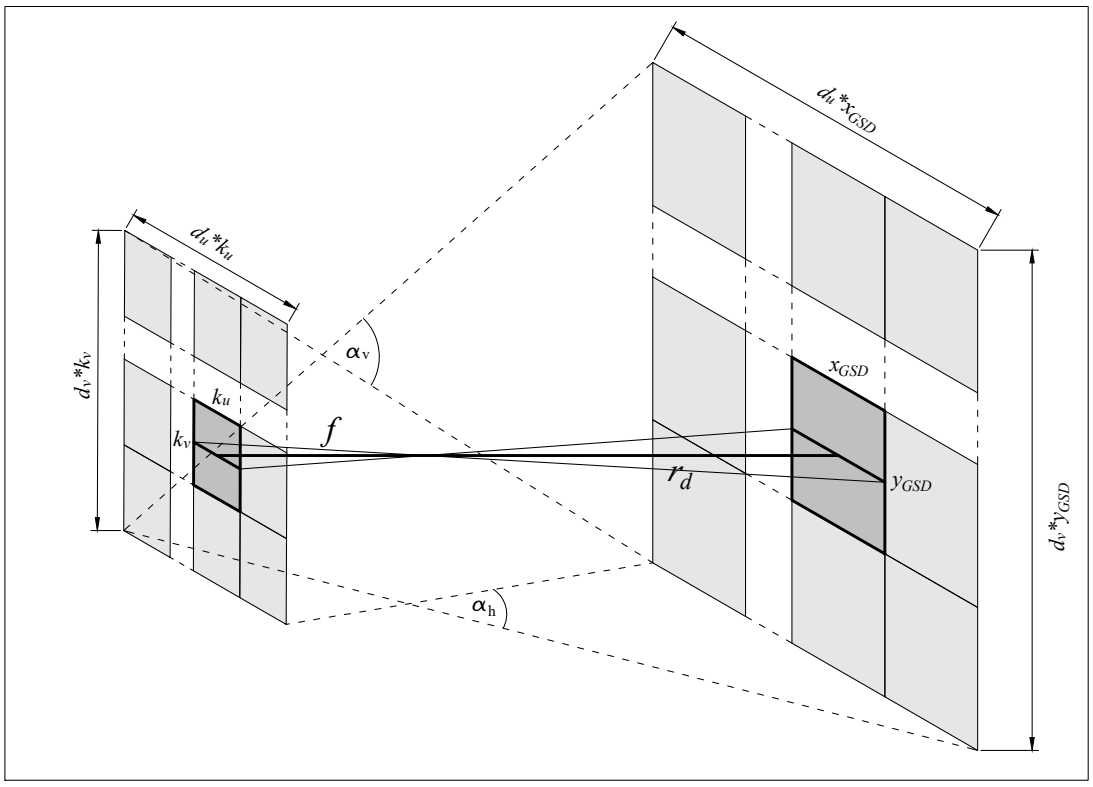


Figure 3.5: Illustration of the Ground Sampling Distance (GSD) principle where image is captured with camera placed at distance r_d from the object. Proprieties of the lens are focal length f, and horizontal (α_h) and vertical (α_v) field of view (FOV) angles, while image capture sensor specifications are size of the pixel denoted as $k_u \times k_v$ and dimensions of the image ($d_u \times d_v$) in pixels.

For a camera with a focal length (f) and pixel sizes (k_u and k_v), the ratio for horizontal and vertical GSD, denoted as x_{GSD} and y_{GSD} respectively, can be defined using the principle of

resemblance illustrated in Fig.3.5. These ratios can be expressed as:

$$\frac{x_{GSD}}{r_d} = \frac{k_u}{f} \tag{3.25}$$

$$\frac{y_{GSD}}{r_d} = \frac{k_v}{f} \tag{3.26}$$

As specifications for the parameters k_u and k_v are often unavailable, it is more practical to calculate the desired distance from the object using the dimensions of the captured images in pixels (d_u and d_v), and the horizontal or vertical field of view (FOV), denoted as α_h and α_v:

$$r_d = \frac{d_u x_{GSD}}{2\tan\left(\frac{\alpha_h}{2}\right)} = \frac{d_v y_{GSD}}{2\tan\left(\frac{\alpha_v}{2}\right)} \tag{3.27}$$

As previously emphasized, it is essential for the path planning system of the UAV engaged in inspection missions to ensure adequate coverage of the surveyed object. This can be achieved by treating relevant viewpoints as a minimal set of waypoints along the flight path that the UAV must traverse with correct orientation. Configuring the flight path as closed isolines around the surveyed object, maintained at a consistent distance and various altitude levels, enables the determination of discretization parameters, denoted as λ, based on the desired overlapping of images captured at neighboring waypoints.

As depicted in Figure3.6, parameter λ_v dictates the difference in altitude levels between successive passes of the UAV around the surveyed object, derived from the vertical overlapping area denoted as O_v. On the other hand, discretization parameter λ_h is determined based on the desired horizontal overlapping area, O_h, between two consecutive images captured at the same altitude during the inspection flight. Depending on the planning method employed, parameter λ_h is utilized either for discretization of the planned path or an area around the surveyed object or UAV, where the flight path is generated, as it determines the distance between two neighboring viewpoints. By using the parameters λ_v and λ_h, the path planning algorithm can generate a minimal set of waypoints that satisfy the conditions for overlapping areas in the acquired images. With values for O_v and O_h expressed as a percentage of the desired overlapping areas, the parameters λ_v and λ_h are calculated using the following equations:

$$\lambda_v = 2\,(1 - O_v)\, r_d \tan\left(\frac{\alpha_v}{2}\right), \quad O_v \in [0, 1\rangle \tag{3.28}$$

$$\lambda_h = 2\,(1 - O_h)\, r_d \tan\left(\frac{\alpha_h}{2}\right), \quad O_h \in [0, 1\rangle \tag{3.29}$$

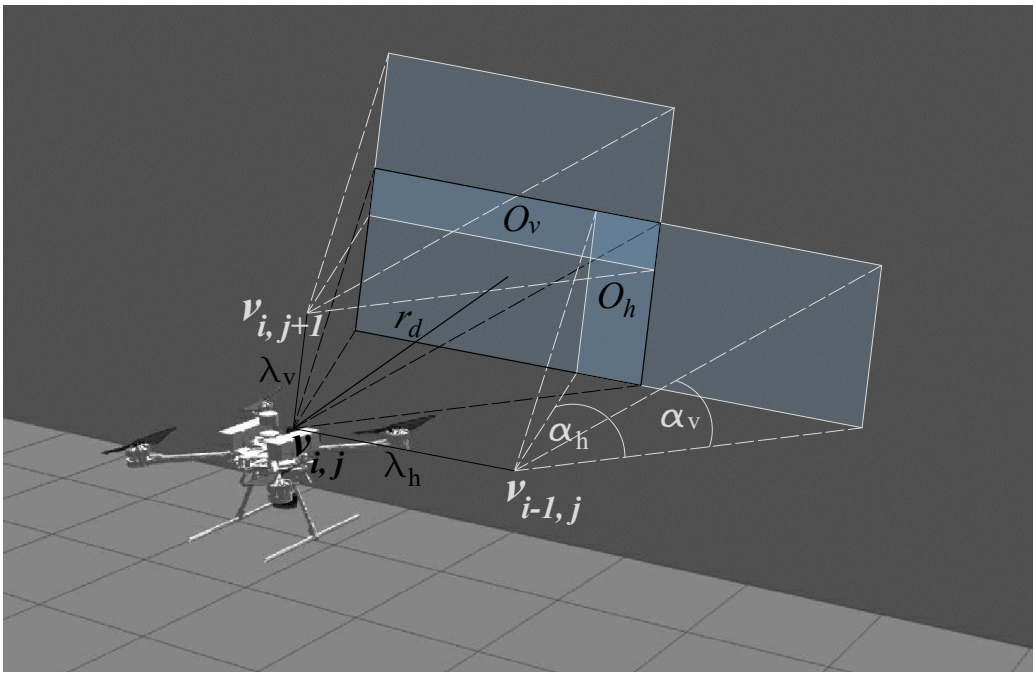


Figure 3.6: Current viewpoint, where the UAV is positioned, is denoted as $\mathbf{v}_{i,j}$. The distance from the previous viewpoint $\mathbf{v}_{i-1,j}$ at the same altitude is marked as λ_h, while the viewpoint denoted as $\mathbf{v}_{i,j+1}$ is the closest to the current one along the flight path on the next altitude level. Areas marked as O_h and O_v represent overlapping sections of the images captured at neighboring viewpoints.

3.3UAV testing platforms and sensors

3.3.1Multirotor platforms

To test and validate the developed algorithms and methods in this work, as well as to conduct experimental inspections of infrastructure, two UAV testing platforms were utilized. Both multirotors used in experiments are shown in Fig.3.7.

The *Kopterworx* UAV is a large-scale custom multirotor flying platform with a carbon-fiber frame featuring four *T-motor P60 KV170* brushless motors equipped with 22" folding propellers capable of operating in real-world conditions. Its dimensions measure $120cm \times 120cm$ in length from motor to motor and $55cm$ in height, with a total takeoff weight of $10kg$, including all electronics, sensors, and batteries. Power is provided by two large $6S$ LiPo batteries, each with a capacity of $10000mAh$, in series configuration and distributed through the *HEX-Kore* high-current power board to all electronic components and the UAV's propulsion system, ensuring 30 minutes of flight time for the UAV equipped with a full navigational and surveying sensor set comprising a GNSS module, LiDAR sensor, and multiple cameras. The GNSS module used in experiments was the *Drotek Sirius F9P*, a high-precision multi-band GNSS-based sensor coupled with the *RM3100* on-board magnetometer and capable of utilizing Real-Time Kinematics

(RTK) corrections received from the stationary base module for increased accuracy. The on-board flight controller, *Cube Orange*, with *Ardupilot* firmware, is utilized for attitude control and state estimation of the UAV based on the data provided by the internal Inertial Measurement Unit (IMU), previously mentioned GNSS and magnetometer sensors. Furthermore, an *Intel NUC 11TNHi5* on-board computer that was mounted on the UAV is capable of processing computationally demanding applications for mapping and planning, as well as running an autonomous navigation stack implemented on the Robot Operating System (ROS) framework. Communication between the flight controller and on-board computer was achieved through the *MAVLink* lightweight messaging protocol with the *MAVROS* extendable communication node for ROS.

For experimental evaluation of obstacle avoidance algorithms in an indoor flying area, a modified *Hexsoon EDU-450* quadcopter was employed. Its smaller size (with a length between motors of $36cm$) makes it more suitable for indoor flying than the larger *Kopterworx* UAV. The propulsion system consists of four *T-Motor* $2216880KV$ motors with $T1045$ self-locking propellers powered by a high-density $4S$ LiPo battery. Low-level attitude control is provided by the *Cube Orange* autopilot system with embedded IMU sensors running on *Ardupilot* firmware. Data processing from the *SLAMTEC RPLIDAR-A3* 2D LiDAR sensor, used for obstacle detection, and implementation of autonomous navigation flight stack with integrated collision avoidance algorithm, are performed on the *Intel NUC* onboard computer, equipped with an $i7-8650UCPU@1.90GHz \times 8$ processor. Communication between the autopilot module and the on-board computer is implemented using the same *MAVROS* node as on the *Kopterworx* UAV. Several infrared reflective markers are placed on the UAV to compute its precise position and orientation using an external *Optitrack* localization system installed in the flight arena.

Figure 3.7: Quadcopters used in experiments - *Kopterworx* UAV with equipped *Velodyne* VLP-16 LiDAR sensor and *ZED* stereo camera (left), and modified *Hexsoon EDU-450* with *SLAMTEC RPLIDAR-A3* 2D LiDAR used for obstacle detection (right).

3.3.2Sensors

Sensors play a critical role in UAV navigation, state estimation, and safety by providing vital environmental information for UAV operation and interaction. Many different array of sensor types is employed to enable precise flight maneuvers, ranging from inertial measurement sensors like accelerometers and gyroscopes, to precise positioning sensors reliant on the Global Navigation Satellite System (GNSS), and specialized marker detection systems. Additionally, LiDAR and optical sensors are utilized in Simultaneous Localization and Mapping (SLAM) algorithms and the generation of detailed environmental maps.

Inertial measurement unit - IMU

Nearly all multirotor UAVs utilized in research or practical applications use some form of inertial measurement unit (IMU) sensors as a core component of the flight controller. Typically, these IMUs are 6 Degrees of Freedom (6 DOF) sensors capable of measuring six distinct types of motion-related data: three axes each for acceleration and angular velocity. Depending on the operational environment, it is not uncommon to augment the IMU with an additional 3 DOF by integrating a magnetometer (electronic compass) for measuring magnetic orientation. However, the inclusion of a magnetometer may introduce sensitivity to external disturbances generated by power lines, electronic devices, or even nearby ferrous materials.

GNSS and Motion Capture

For outdoor localization and navigation tasks, high-precision positioning systems based on the GNSS network are most commonly used. These systems operate on the principle of triangulation, by calculating the distance from four or more satellites based on the difference between transmission and arrival times of signals. Enhanced positional accuracy can be achieved by incorporating Real-Time Kinematics (RTK) corrections from a stationary base module, which utilizes the carrier wave of satellite signals to refine the UAV's position without extracting embedded data. In environments where GNSS signals are unreliable, motion capture systems such as *Optitrack* are often employed for UAV pose estimation with a high degree of accuracy and frame rates. These systems detect specific markers with a known configuration on the vehicle, typically in the infrared light spectrum, to calculate the position and velocity of the UAV.

LiDAR

Modern LiDAR sensors are capable of generating highly detailed three-dimensional representations of the environment in the form of point cloud datasets. Due to their ability to provide information with high accuracy and frequency, they are indispensable tools for real-time object detection and collision avoidance applications. However, one drawback is their relatively high

demand for computational power to function effectively. LiDAR sensors operate by emitting pulsed laser light to measure distances to features in the environment. In this study, various 3D and 2D LiDAR systems were employed. The *Velodyne* VLP-16 is a 16-channel 3D LiDAR with a 360 horizontal and 30 vertical field of view, offering selectable rotation frequencies ranging from 5 to 20Hz. This sensor has a measurement range of up to 100m with an accuracy of $\pm 3cm$. Another rotating 3D LiDAR used is the *Ouster OS0-128*, which provides a slightly shorter measurement range of up to 75m, but far greater number of channels (128) distributed within a 90 vertical field of view. Both sensors are equipped with dual returns of the laser beam, enabling them to penetrate vegetation and generate accurate digital surface models. For indoor laboratory experiments, where weight limitations are a concern for smaller UAVs, a lighter 2D *SLAMTEC RPLIDAR-A3* LiDAR sensor proved to be the ideal solution. This sensor produces a 360 horizontal single-layer scan of the environment with a measurement range of up to 25m and a sampling rate of up to 16000 times per second. Detailed information regarding LiDAR data processing, including the application of different filters and clustering methods, is provided in sections discussing path planning and trajectory tracking algorithms.

Cameras

Optical imaging sensors form another essential category of sensors that find application both in data acquisition for inspections and as navigation tools for UAVs. A wide array of camera types is available depending on the use case: conventional RGB cameras capable of capturing high-quality images or videos suitable for various purposes, specialized thermal and multispectral cameras utilized for detecting water damages, material degradation, or assessing vegetation conditions, and stereo cameras primarily employed for the localization and navigation of autonomous robots. Numerous algorithms have been developed for tasks such as image-based visual servoing [76] of UAVs and target detection and tracking [77]. However, there are several limitations to using cameras as navigation tools in real-world conditions. These limitations include their reliance on lighting and weather conditions, limited range and field of view, and sensitivity to environmental variations such as shadows, reflections, or changes in texture and color. All of these factors can also affect the quality of the data gathered during infrastructure surveys, necessitating careful consideration during the planning of inspection missions.

CHAPTER 4

Path planning based on Huygens's wave propagation principle

4.1Path planning using envelope based on 2D layout

The presented path planning algorithm is primarily designed for UAVs engaged in the data acquisition of civil infrastructure [78]. The algorithm prioritizes two key objectives: maintaining a consistent desired distance from the surveyed structure throughout the entire flight trajectory and ensuring that the UAV's heading aligns the data acquisition sensor perpendicularly to the structure's sides. Diagram of a path planning algorithm is depicted in Fig.4.1.

To utilize this path planning method effectively, certain prerequisites must be met. Main prerequisite is access to a 2D layout representation of the surveyed structure, typically provided as a list of corner coordinates outlining the building's perimeter. This layout enables the interpolation of straight lines between corners, forming a complete outline of the structure. Obtaining this layout data can be achieved from various sources, such as georeferenced satellite images or cadastral databases.

A widely accepted standard format for representing cadastral databases is the Geography Markup Language (GML) [79], developed by the Open Geospatial Consortium (OGC). GML, an XML-based grammar, contains a set of primitives such as *Features*, *Geometry*, *Coordinate Reference System*, and *Topology*, designed for describing geographical structures, both natural and artificial. Utilizing the GML allows development of applications which use the cadastral data regardless of the country from which the data originates.

4.1.1Path planning method

The initial step in the path planning algorithm involves the generation of an accurate representation of a building layout. This is accomplished by utilizing a list of points, represented as $\mathcal{B} = \{b_1, \ldots, b_m\}$, where each point $b_i = (b_{x_i}, b_{y_i})$ corresponds to a corner in the building layout with Cartesian coordinates, and m denotes the total number of points. Ensuring that the list of points defining the building layout is ordered with a known orientation is important for subsequent stages of the planning algorithm. To sort the points counterclockwise, a comparator function is defined based on the angle between each point and a reference point. The reference point b_{ref} is selected from the set $\mathcal{B}$ as the point closest to the current position of the UAV q, determined by the shortest Euclidean distance:

$$b_{ref} = \min(\|b_i - q\|), \quad i \in \{1, \ldots, m\}. \tag{4.1}$$

With θ_i representing the angle of point b_i relative to reference point b_{ref}, the comparator function is defined as:

$$\theta_i = \text{atan2}(b_{y_i} - b_{y_{ref}}, b_{x_i} - b_{x_{ref}}). \tag{4.2}$$

Sorting the points based on θ_i results in an ordered set $\mathcal{B}$ with points arranged counterclockwise, where any point b_i satisfies the condition:

$$(b_i - b_{i-1}) \times (b_{i+1} - b_i) \leq 0, \quad i \in \{1, \ldots, m\} \tag{4.3}$$

with $b_0 = b_m$ and $b_{m+1} = b_1$. A sampled representation of a building outline is achieved through linear interpolation between adjacent corners, with a resolution parameter denoted as r_b, creating a new set of points $\mathcal{S} = \{s_1, \ldots, s_n\}$.

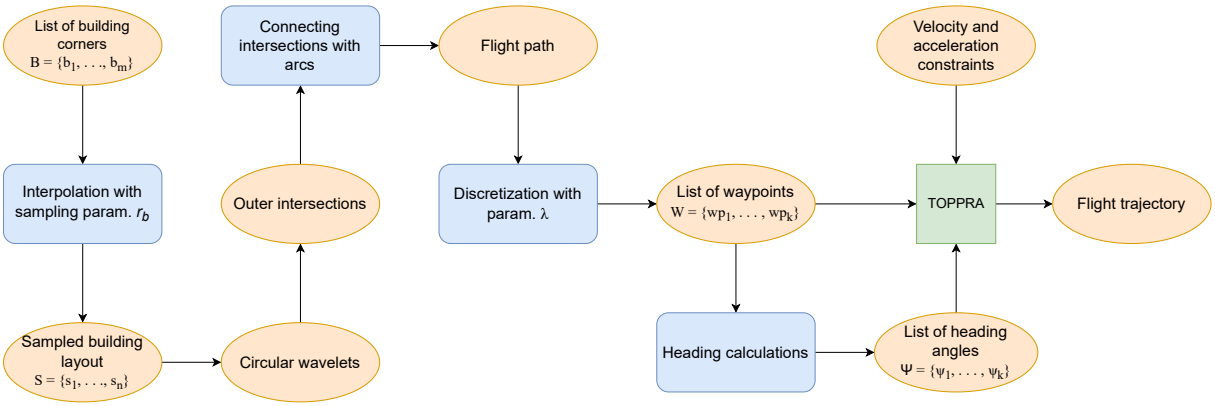


Figure 4.1: Trajectory planning diagram with path planning algorithm based on Huygens's wave propagation principle and known 2D layout of the structure

Following this, the algorithm proceeds to generate circles centered at each point s_i along the sampled building outline $\mathcal{S}$, with radii set to the parameter r_d representing the desired distance

of the flight path from the building. This methodology for flight path planning, involving the creation of a series of intersecting circles, draws inspiration from Huygens's wave propagation principle [80]. According to this principle, every point on a wave front can be considered a source of secondary spherical wavelets. By regarding the sampled building outline as the original wavefront and constraining the propagation of secondary wavelets to a radius of r_d, the resulting distance from every point on the new wavefront to the nearest point within the sampled building outline precisely equals r_d, as illustrated in Fig.4.2.

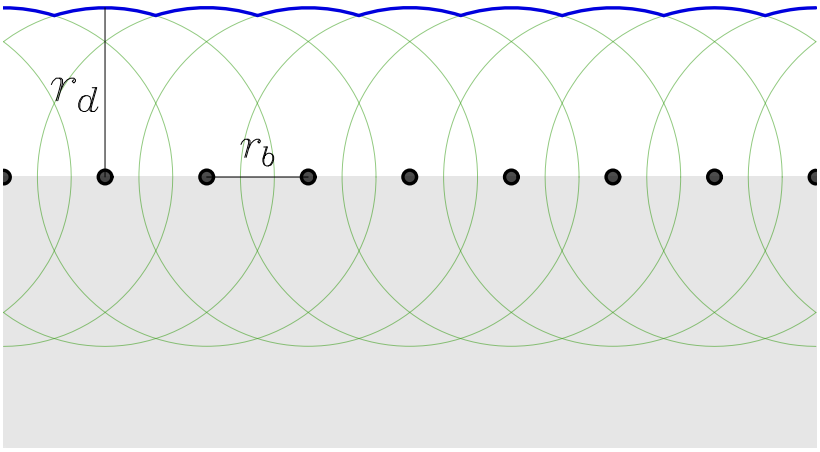


Figure 4.2: New wave front (blue line) generated by circular wavelets (green circles). Sources of the circular wavelets are points (black dots) interpolated between building corners with the resolution parameter r_b. Distance of the new wave front is given with the parameter r_d.

To define points on the new wavefront, the algorithm must identify all intersections between neighboring circles outside of the building layout and not contained within any generated circles. Given that the set of points $\mathcal{S}$ is ordered, point s_1 denotes the corner of the building closest to the current position of the UAV during path planning. Commencing with a circle centered at this point, intersections between adjacent circles in the counterclockwise direction are computed using the following equations:

$$r = \sqrt{(s_{x_{i+1}} - s_{x_i})^2 + (s_{y_{i+1}} - s_{y_i})^2} \tag{4.4}$$

$$\begin{aligned} v_{i_{1,2}} =& \frac{1}{2}(s_{x_i} + s_{x_{i+1}}) + \frac{(r_i^2 - r_{i+1}^2)}{2r^2}(s_{x_{i+1}} - s_{x_i}) \\ & \pm \frac{1}{2}\sqrt{2\frac{r_i^2 + r_{i+1}^2}{r^2} - \frac{(r_i^2 - r_{i+1}^2)^2}{r^4} - 1} \cdot (s_{y_{i+1}} - s_{y_i}) \end{aligned} \tag{4.5}$$

$$
\begin{aligned}
w_{i_{1,2}} =& \frac{1}{2}(s_{y_i} + s_{y_{i+1}}) + \frac{(r_i^2 - r_{i+1}^2)}{2r^2}(s_{y_{i+1}} - s_{y_i}) \\
& \pm \frac{1}{2}\sqrt{2\frac{r_i^2 + r_{i+1}^2}{r^2} - \frac{(r_i^2 - r_{i+1}^2)^2}{r^4} - 1} \cdot (s_{x_{i+1}} - s_{x_i}),
\end{aligned}
\tag{4.6}
$$

where (s_{x_i}, s_{y_i}) and $(s_{x_{i+1}}, s_{y_{i+1}})$ represent the centers of adjacent circles, (v_{i_1}, w_{i_1}) and (v_{i_2}, w_{i_2}) denote the intersection coordinates, and r_i and r_{i+1} are the radii of the circles. From the first intersection pair, one point is randomly selected as initial intersection. Subsequently, for all other intersection pairs, the algorithm chooses the point with the shorter arc length between the intersection points and the last point on the new wavefront, discarding the other. As the arc lengths are computed in the counterclockwise direction, consistent with the orientation of the circle centers in the ordered set $\mathcal{S}$, the endpoint of the shorter arc is always outside of the building layout. If the position of the selected point is not within any other generated circle, the intersection point is appended to the list of points on the new wavefront, and the algorithm proceeds to the next pair of circles. Given that the first intersection point was randomly selected from the two calculated for the first pair of circles, potentially placing it inside the building, the calculations for the first pair of circles are repeated at the conclusion of the process, and the initial intersection point is updated.

In constructing the flight path, which is composed of interconnected arcs between intersection points on the new wavefront, it is important to address deviations from the optimal trajectory. Optimal trajectory can be considered one on which the UAV constantly maintains desired distance throughout the whole flight. Illustrated in Fig.4.4, line l_d runs parallel to a segment of the building layout interpolated between two adjacent corners with an offset of r_d, representing a segment of the optimal flight path. Points on the generated flight path, coinciding with intersections of circles, have maximum deviation from line l_d. This maximum possible error e of any point on the computed flight path from the desired distance r_d from the building is defined by equation4.7.

$$
e = r_d - \sqrt{r_d^2 - \frac{r_b^2}{4}} \tag{4.7}
$$

By decreasing the resolution parameter r_b to $0.1r_d$, the resulting flight trajectory yields a maximum possible error of any point on the trajectory as $e = 1.25 \cdot 10^{-3} r_d$.

The path planner generates a list of waypoints by discretizing the flight path, where the resolution parameter λ_p denotes the arc length between adjacent waypoints. This parameter λ_p is determined based on the desired overlap area between two consecutive images and is calculated using equation3.29. It is noteworthy that the distance between two neighboring waypoints may

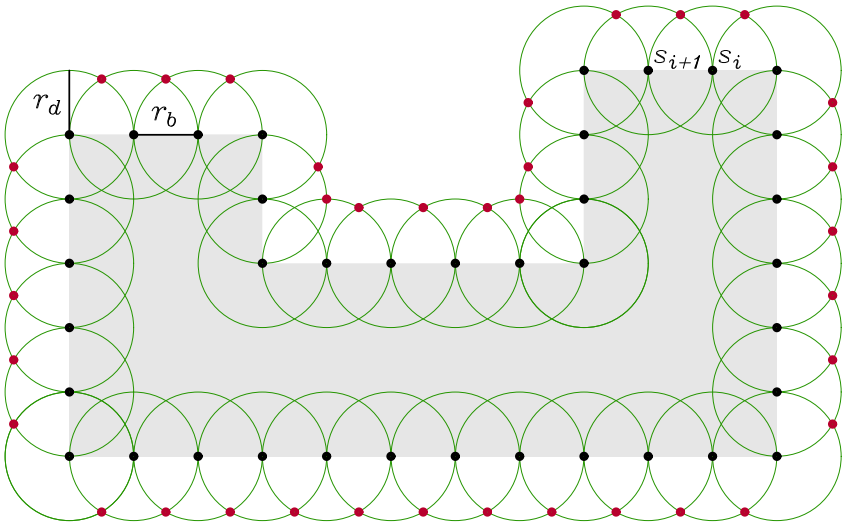


Figure 4.3: Set of points S represents a sampled building layout, illustrated by black markers, where the distance between adjacent points s_i and s_{i+1} is defined with parameter r_b. Green circles represent circular wavelets with radii r_d, while intersections between them positioned outside of a building layout are highlighted with red markers.

exceed the arc length of a single circular wavelet, implying that not every generated wavelet necessary contains a waypoint after discretization. Additionally, it is observed that around sharp corners of the object, a larger segment of the new wavefront is generated from a single wavelet, resulting in the inclusion of multiple waypoints within it. The resulting list represents a minimal waypoint set ensuring comprehensive object coverage while fulfilling overlap requirements.

Given that the primary purpose of the outlined path planning approach is for inspection trajectories, it is important for the UAV to maintain a perpendicular orientation to the surveyed object, assuming that the data acquisition sensor is aligned with the UAV's heading. The heading angle of the UAV ψ_i at the i-th waypoint is determined using equation4.8:

$$\psi_i = \frac{\text{atan2}(y_{i+1} - y_i, x_{i+1} - x_i)}{2} + \frac{\text{atan2}(y_{i-1} - y_i, x_{i-1} - x_i)}{2}, \tag{4.8}$$

where (x_i, y_i) represent the coordinates of the i-th waypoint in the list. The angle ψ can be easily adjusted for different configurations of the data-acquisition sensors by employing the static transformation matrix between the reference frame of the sensor and that of the UAV. Utilizing the generated list of waypoints, computed heading angles ψ, and velocity and acceleration constraints as inputs for the Time Optimal Path Parametrization by Reachability Analysis (TOPPRA) algorithm [69], the UAV flight trajectory is calculated. Employing the TOPPRA algorithm ensures that the planned trajectory passes through all waypoints while adhering to all

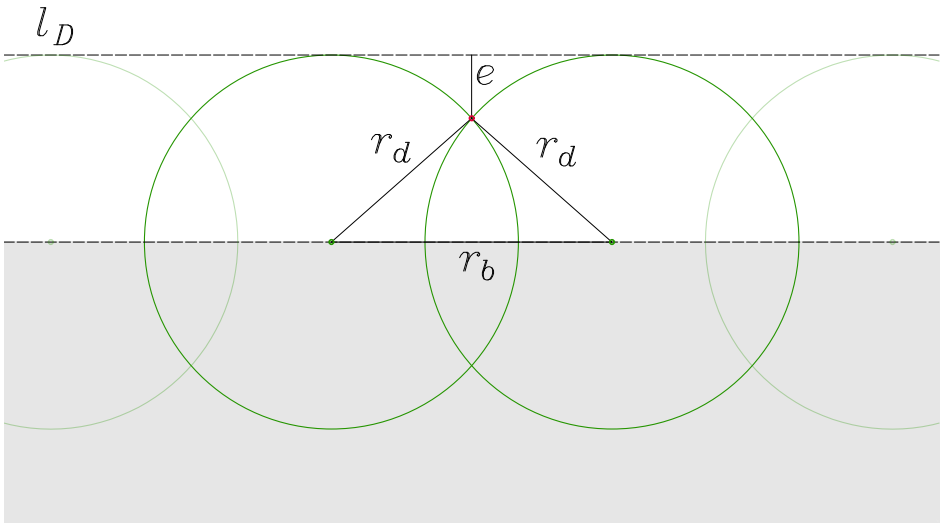


Figure 4.4: Maximum possible error e of any point on the calculated trajectory from desired distance from the building d

constraints, enhancing data quality and reducing the risk of collisions with obstacles surrounding the surveyed building.

Short computation time (within 5 seconds on *Intel NUC*, i7-8650U CPU @ 1.90GHz × 8) of the algorithm, including trajectory generation by TOPPRA, allows for a starting point of the flight path to be calculated with respect to the current position of the UAV, which allows the arbitrary starting point for the UAV in order to start executing the autonomous flight trajectory. Size and complexity of the surveyed building could significantly increase flight path length and flight time needed for the completion of the whole survey.

4.1.2Data acquisition and post-processing algorithm

In order to assess the effectiveness of path planning techniques for UAVs utilized in infrastructure inspection, a comprehensive procedures for data acquisition and post-processing was developed. and implemented. These methods were devised for implementation on both single and multi-UAV systems, and were integrated into simulation and real-world experimentation scenarios.

The data acquisition process involves the utilization of LiDAR sensors mounted on the UAVs to gather point cloud representation of the surveyed infrastructure objects. By sequentially stitching new time synchronized pairs of LiDAR point cloud scans and estimated UAV odometry obtained from GPS and IMU measurements a map of the environment was generated.

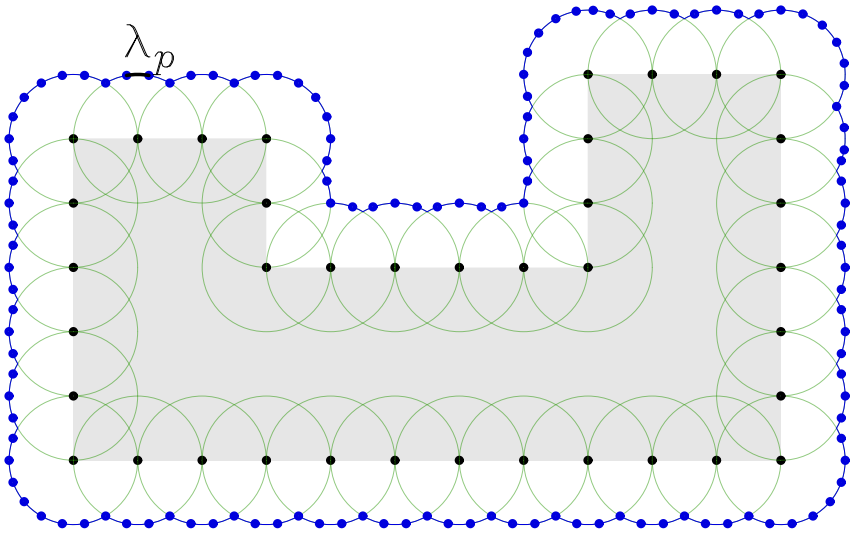


Figure 4.5: Waypoints (blue dots) generated by discretization of a flight path with resolution parameter λ_p.

Furthermore, the methodology was extended to a system with multiple UAVs, provided they share a common reference frame. This coordination was facilitated by synchronizing the home points of all UAVs, thereby aligning their respective world reference frames with that of the primary UAV. This synchronization process involved translating subsequent UAV origin points to coincide with that of the initial UAV. It is important to note that these methods were primarily devised to validate the efficacy of the trajectory planning algorithm and data acquisition procedures.

Algorithm1outlines a solution for generating an accumulated point cloud map from single UAV sensor data. To generalize this methodology for application with multiple UAVs, each dataset containing point cloud-odometry pairs is processed sequentially. In Step 1 of Algorithm 1, the accumulated point cloud map obtained from the previous dataset serves as the initial map before processing the subsequent dataset.a

Therefore, the requirements for successful mapping are twofold: i) East-North-Up (ENU) frame origin - base link - LiDAR transformation has to be known for each UAV; ii) ensuring that the initial point of each UAV is within an area observed and scanned by the other UAVs at some point during their mission.

The alignment of a single point cloud scan with the accumulated point cloud map is done through the iterative closest points (ICP) method at Step 7. This method iteratively estimates the transformation of a single point cloud $\mathcal{S} = \{x_1, x_2, ..., x_n\}$ with respect to the the accumulated point cloud map $\mathcal{T} = \{y_1, y_2, ..., y_m\}$. The transformation estimation is based on the Levenberg-

Algorithm 1: A sequential registration, based on iterative closest points (ICP), of newly obtained point cloud - estimated odometry data pairs to form an accumulated point cloud map of the environment .

Result: An accumulated point cloud map of the environment

1 ■ Initialize the accumulated point cloud map (referred to as the map) as empty;
2 **while** *more {point cloud, odometry} pairs exist* **do**
3 ○ Transform the current point cloud to the ENU frame using paired odometry data;
4 **if** *the map is empty* **then**
5 • Add the current transformed point cloud to the map;
6 **else**
7 • Using ICP obtain transformation **T** which best aligns the current transformed point cloud to the map;
8 • Apply $\mathbf{T}^{-1}$ transformation to the map;
9 • Add all points in the current transformed point cloud to the map;
10 • Apply Voxel Grid Filter to the map;
11 **end**
12 **end**

Marquadt algorithm, which solves a nonlinear least-squares problem:

$$\mathbf{t}, \gamma = \underset{\mathbf{t}\in\boldsymbol{R}^3, \gamma\in\boldsymbol{R}}{argmin} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[(\mathrm{R}_z(\gamma) x_i + \mathbf{t}) - y_j \right]^2 \tag{4.9}$$

$$\mathrm{R}_z(\alpha) = \begin{bmatrix} (cos)(\alpha) & -(sin)(\alpha) & 0 \\ (sin)(\alpha) & (cos)(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.10}$$

Here, γ represents the rotation angle around the z-axis, and **t** denotes translation. Roll and pitch are not considered in the problem formulation as their accurate measurements are derived from estimated UAV odometry, whereas yaw is heavily reliant on GPS quality. The transformation **T** at Steps 7 and 9 contains the optimal translation **t** and rotation around z-axis γ. The reason for using an inverse transformation on the entire accumulated point cloud map is to maintain a single East-North-Up (ENU) origin frame. Finally, at Step 10, a Voxel Grid Filter is applied to eliminate duplicate points after the accumulation step.

4.1.3Simulation results

To assess and validate the trajectory planning algorithm and the process of generating accumulated point cloud maps, simulations were conducted within the Gazebo environment. The civil infrastructure model was imported from Gazebo's building database, while models of UAVs were based on a custom made *Kopterworx* quadcopters that were later utilized in real-world ex-

periments. These UAV models were equipped with simulated *Velodyne VLP-16* LiDAR sensors to acquire point cloud data during the flight missions.

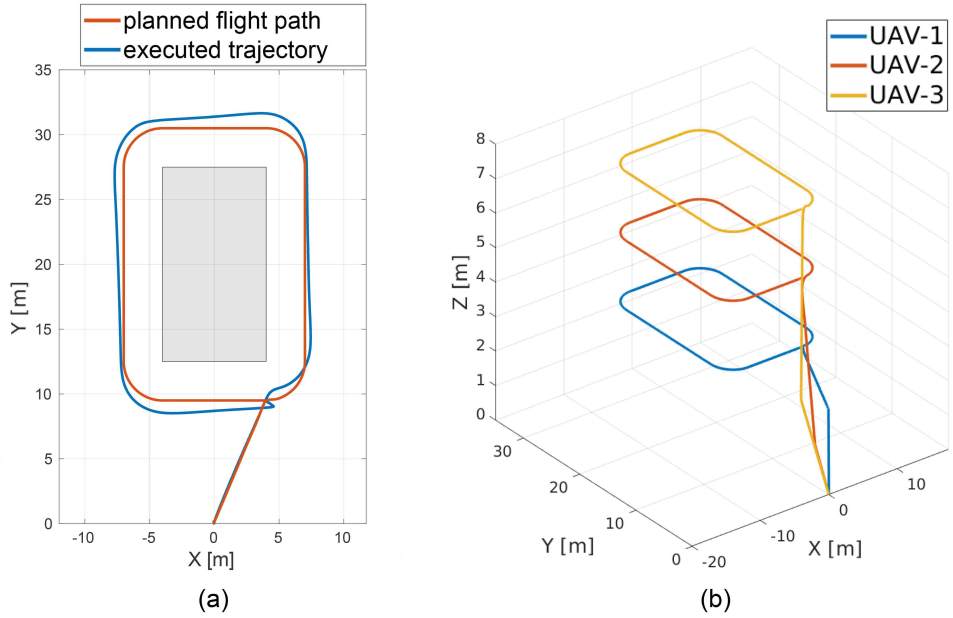
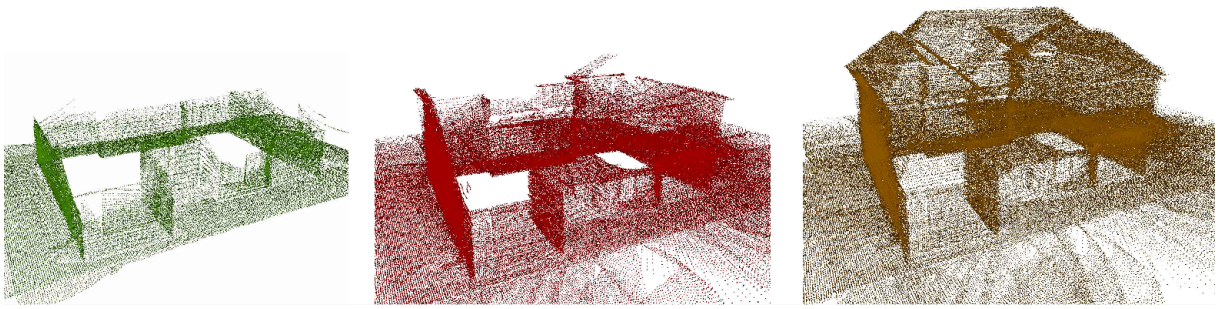


Figure 4.6: (a) Planned flight path and executed trajectory for UAV-1 in simulation of a building survey. (b) Planned flight paths for 3 UAVs in simaulation

The surveyed building dimensions were $8m \times 15m \times 8m$, with a building layout point resolution parameter of $r_b = 0.1m$. The flight trajectory was maintained at a distance of $3m$ from the building, with a waypoint discretization parameter of $\lambda_p = 0.3m$. The flight altitudes were set at $3m$ for UAV-1, $5m$ for UAV-2, and $7m$ for UAV-3. Velocity and acceleration constraints in the X, Y, and Z planes for all flight trajectories were set to $1m/s$ and $0.5m/s^2$ respectively. Angular velocity constraints were set to $1rad/s$, and angular acceleration to $0.3rad/s^2$ for rotation around the yaw axis. The generated trajectories are depicted in Fig.4.6. The maximum possible error of any point on the planned trajectory from the desired distance from the building, given the specified parameters, was $e = 4.17 \times 10^{-4}m$.

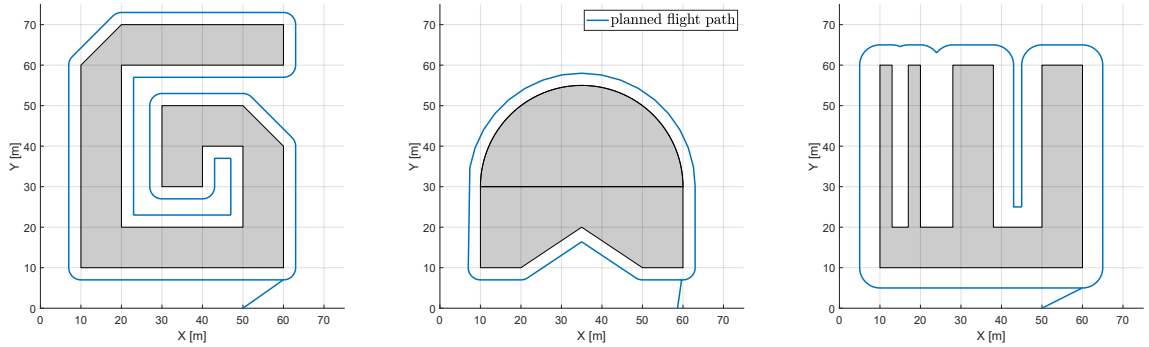
Point cloud maps obtained through post-processing the acquired data are displayed in Fig. 4.7. The complete point cloud map was generated by processing data acquired from multiple UAVs, utilizing the methods described in Section4.1.2. This process effectively validates the data acquisition in a simulation environment.

Furthermore, to explore the capabilities of the trajectory planning algorithm, it was tested on highly irregular and complex building shapes, as depicted in Fig.4.8. In the first case, flight paths were planned around a maze-like structure, while in the second case, the layout of the surveyed building included a semi-circular section. In the third scenario, the surveyed structure featured three narrow corridors, two of which were insufficiently wide for the UAV to pass through while maintaining the desired distance from the walls. In this case, the algorithm planned the flight path to enter only the wider corridor, ensuring UAV safety.



(a) (b) (c)

Figure 4.7: Three consecutive point cloud maps obtained by offline processing the acquired data from three separate UAVs. Each point cloud contains information from the previous map, i.e. the bottom image shows the cumulative map obtained from all three UAVs.



(a) (b) (c)

Figure 4.8: Simulation scenarios with complex structure shapes.

4.1.4Experimental results

Following the successful validation of trajectory planning and data processing algorithms in a simulated environment, real-world experiments were conducted at a designated test site. These experiments involved the utilization of two large-scale carbon-fiber custom-made *Kopterworx* quadcopters, both equipped with *Cube Black* autopilots running *Arducopter* firmware for low-level attitude control and an *Intel NUC* onboard computer for trajectory planning and execution. Point cloud data of the surveyed building was collected using *Velodyne VLP-16 LiDAR* sensors mounted on the UAVs.

The dimensions of the test site building significantly exceeded those of the simulated environment, with lengths of the east and north walls measuring $104m$ and $45m$, respectively. Trajectories around the surveyed structure were planned to maintain a $3m$ distance from the outer walls of the building, utilizing parameters of $r_b = 0.1m$ and $\lambda_p = 0.3m$. Constraints were set to $1m/s$ for velocity and $0.5m/s^2$ for acceleration in the X, Y, and Z planes, as well as $1rad/s$ and $0.3rad/s^2$ for yaw rotation. The layout of the structure, represented by a list of points cor-

Figure 4.9: Planned flight path, executed trajectory and UAV's heading vectors in experimental surveying of the test site building.

responding to the corners of the building, was generated based on satellite images using the API provided by *Google Maps*^{TM}.

Figure4.9illustrates the planned flight path, executed trajectory, and UAV heading vectors. In the initial phase of the experiment, a UAV executed a counterclockwise trajectory around the entire building, covering a trajectory length of 339m and maintaining a flight altitude above surrounding obstacles along the north wall. In the subsequent phase, the primary objective was to validate an algorithm designed for processing data obtained from multiple UAVs. Two UAVs executed planned trajectories around the U-shaped part of the infrastructure at lower altitudes to acquire relevant data necessary for generating a point cloud of the surveyed building.

The point cloud map depicted in Fig.4.10was generated through post-processing of data acquired by the two UAVs. The datasets utilized for map generation consisted of estimated UAV odometry provided by the *Cube Black* autopilot and point cloud scans from the *Velodyne VLP-16*. By utilizing data post-processing Algorithm1, a comprehensive map of the building was obtained, thereby validating the data acquisition procedure.

4.2Path planning using envelope based on 3D model

While much of civil infrastructure can be adequately described using a two-dimensional layout, there are scenarios where this approach falls short, particularly in planning optimal inspection trajectories for UAVs. When dealing with the inspection of complicated structures such as bridges, industrial facilities, or wind turbines, the limitations of a two-dimensional framework become apparent, necessitating the adoption of path planning methodologies based on three-

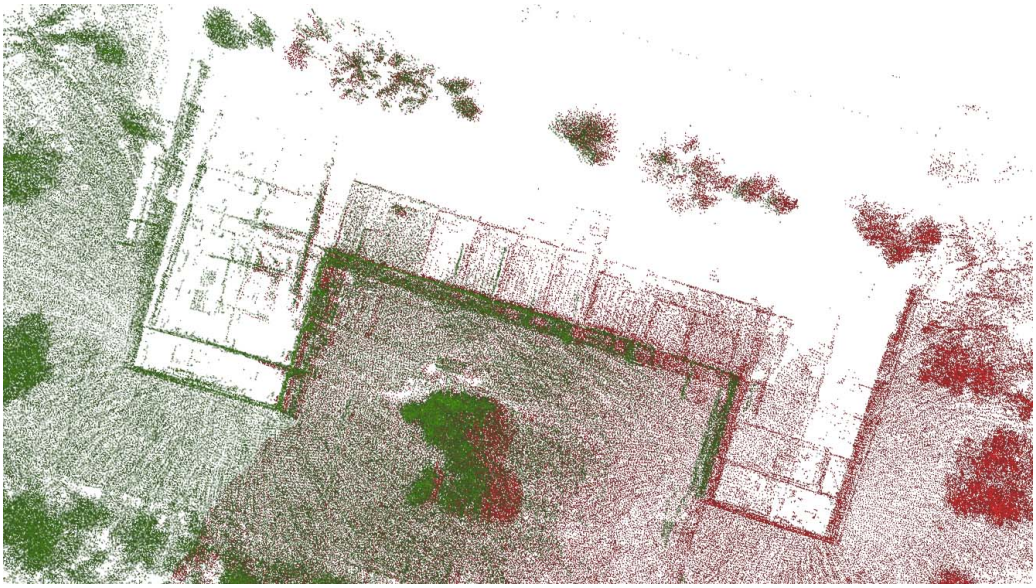


Figure 4.10: This image shows a top-down view of the map obtained using the acquired data from two different UAVs performing a data-acquisition scenario. Green and red points represent the map obtained by UAV-1 and UAV-2 respectively.

dimensional models of the surveyed object. This requirement emerges due to the inherently complex geometries of these structures, which do not allow usage of straightforward translation of planned trajectories along the vertical axis.

Analogous to the previously presented path planning algorithm, this method draws inspiration from Huygens's wave propagation principle. However, instead of generating secondary wavelets as circles emanating from the sampled 2D outline, a 3D model of the structure, represented as a point cloud, serves as the source for spherical wavelets. Around each point within this model, a sphere with a specified radius is constructed, comprising equidistant points. These spheres serve as the building blocks for generating an envelope in the form of a convex hull surrounding the model. The waypoints for the flight path are selected from this envelope, ensuring that the UAV maintains constant distance from the surveyed structure while enabling comprehensive coverage and efficient inspection trajectories. Diagram for generating path planning envelope is depicted in Fig4.11.

One of the key aspect of the proposed path planning method is its independence from precise geographical coordinates of the surveyed object for executing flight trajectories in real-world scenarios. This feature is particularly valuable for structures that are accurately represented by models but exhibit dynamic behavior in relation to the world frame. A prime example of such structures is wind turbines, where the blades can be stopped in arbitrary position to match the known model, while the orientation of the nacelle must be aligned with the prevailing wind direction to reduce the stress on the blades.

Through the utilization of real-time registration and matching techniques between the model

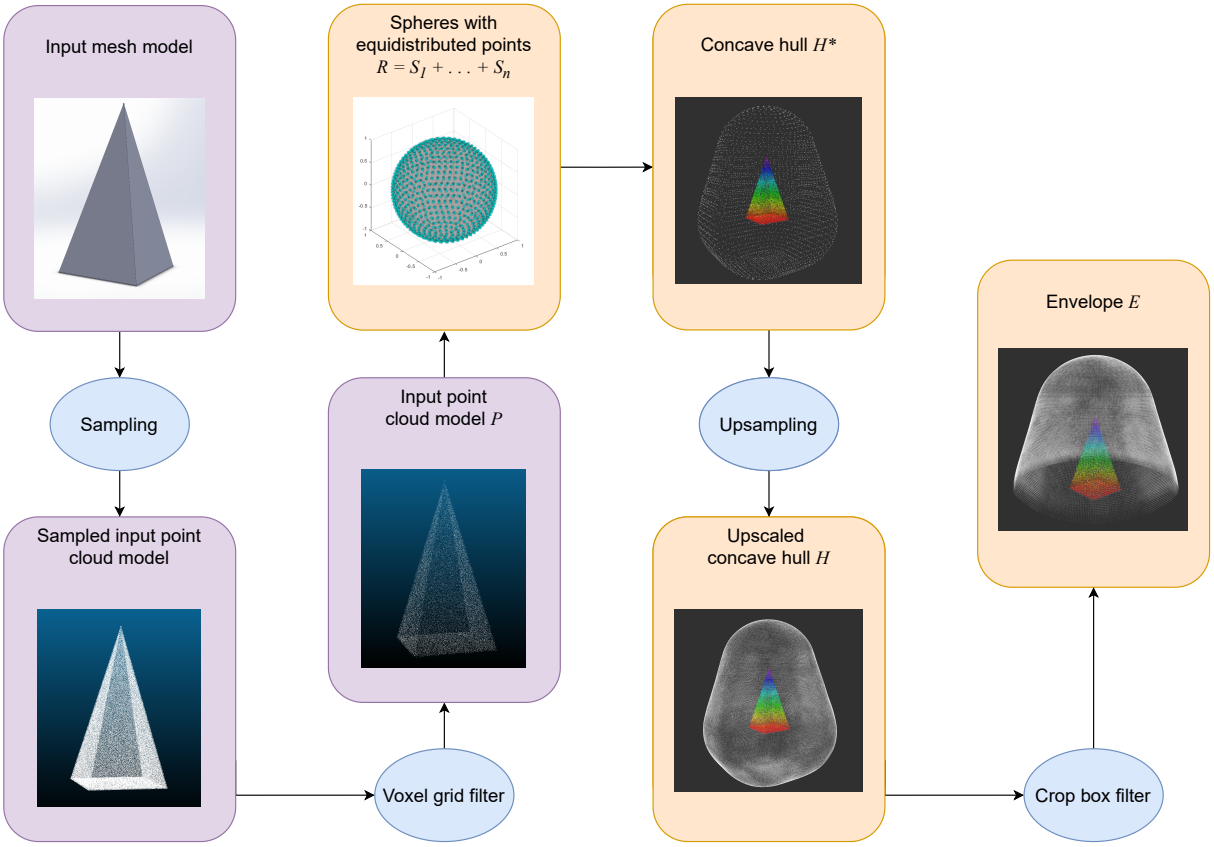


Figure 4.11: Diagram for generating path planning envelope based on a 3D model of the structure.

and the point cloud constructed during flight, the generated envelope and subsequently planned path can be aligned with the current configuration of the surveyed structure and accurately positioned in the real world. Achieving precise matching between point clouds ensures that the trajectory is executed with even greater precision in relation to the surveyed structure, without reliance on geographical coordinates that may exhibit lower reliability, especially when derived from satellite images. This capability enhances the adaptability and accuracy of the path planning method, enabling effective inspection and assessment of dynamic structures like wind turbines with great precision.

4.2.1Envelope generation

The primary prerequisite for employing the path planning method based on the envelope of the surveyed structure is a known 3D model of the structure, typically represented by a point cloud dataset or mesh model. The process of generating the point cloud from the mesh model can be accomplished through sampling techniques using various 3D modeling applications. The sampling parameters, which regulate the density of the point cloud, are determined by both the complexity of the model and the requirements to maintain the desired level of detail from the mesh model.

Since the input for the path planning method can encompass any point cloud dataset, not just one created by sampling the mesh model of the structure, it is advisable to ensure that the algorithm does not operate on an unnecessarily dense point cloud. This is achieved by applying a *VoxelGrid* filter from the PCL library [81]. This filtering technique partitions the point cloud data into voxels, forming a three-dimensional grid over the dataset, and approximates all points within each voxel by their centroid. This downsampling process effectively reduces the number of points while preserving the representation of the underlying surface. The result of the filtering process is a point cloud $\mathcal{P} = \{\mathbf{p_1}, \ldots, \mathbf{p}_k\}$, where k denotes the total number of points.

The initial step in constructing the envelope involves generating spheres centered around each point of the input dataset. Each sphere is composed of a predetermined number of equidistant points N_s, with a radius determined by the parameter r_s, representing the desired distance of the UAV from the structure during the surveying mission. Each point $\mathbf{s}_i$ on the individual sphere $\mathcal{S}_i$ can be represented by its spherical coordinates:

$$\mathbf{s}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = r_s \begin{bmatrix} \sin\theta_i \cos\phi_i \\ \sin\theta_i \sin\phi_i \\ \cos\theta_i \end{bmatrix} \tag{4.11}$$

with the inclination angle $\theta_i \in [0, \pi]$ and the azimuth angle $\phi \in [0, 2\pi\rangle$. Algorithm for generating evenly distributed points on the surface of a sphere is derived from the methodology presented in [82]. The idea for achieving regular equidistribution of points involves generating circles at a constant interval d_θ along the latitude of the sphere, with points on these circles positioned at intervals of d_ϕ. The values of these two intervals should be such that $d_\theta \approx d_\phi$ and that average area per point A equals:

$$A = d_\theta d_\phi = \frac{4\pi r_s^2}{N_s} \tag{4.12}$$

The number of latitude circles on a sphere, denoted by M_θ, is defined by the equation:

$$M_\theta = \left\lfloor \frac{\pi}{\sqrt{A}} \right\rfloor \tag{4.13}$$

which determines the values for intervals d_θ and d_ϕ as:

$$d_\theta = \frac{\pi}{M_\theta}, \quad d_\phi = \frac{A}{d_\theta} = \frac{A M_\theta}{\pi} \tag{4.14}$$

The inclination angle θ_i for every point on the sphere $\mathbf{s}_i$ can be calculated using:

$$\theta_i = \frac{\pi(m - 0.5)}{M_\theta}, \quad m \in \{1, \ldots, M_\theta\} \tag{4.15}$$

The number of points on the m-th latitude circle is determined by:

$$M_{\phi,m} = \left\lfloor \frac{2\pi \sin\theta_i}{d_\phi} \right\rceil \tag{4.16}$$

Finally, employing the equations4.15and4.16, the azimuth angle ϕ_i is calculated as:

$$\phi_i = \frac{2\pi n}{M_{\phi,m}} = \frac{2\pi n}{\left\lfloor \frac{2\pi \sin\theta_i}{d_\phi} \right\rceil} = \frac{2\pi n}{\left\lfloor \frac{2\pi \sin\left(\frac{\pi(m-0.5)}{M_\theta}\right)}{d_\phi} \right\rceil}, \quad m \in \{1,\ldots,M_\theta\}, n \in \{0,\ldots,M_{\phi,m}-1\} \tag{4.17}$$

Illustration of the latitude circles, average area per point A and distance intervals d_θ and d_ϕ, as well as an example of the result of the algorithm generating $N_s = 5000$ points on a sphere with radius $r_s = 1$, is depicted in the Fig.4.12. All constructed spheres are combined into a single point cloud $\mathcal{R}$:

$$\mathcal{R} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_k \tag{4.18}$$

where k is number of the points in input model of the structure $\mathcal{P}$. After generating the new point cloud $\mathcal{R}$, a *VoxelGrid* filter is applied to reduce its density, particularly in areas with a high number of overlapping spheres $\mathcal{S}_i$.

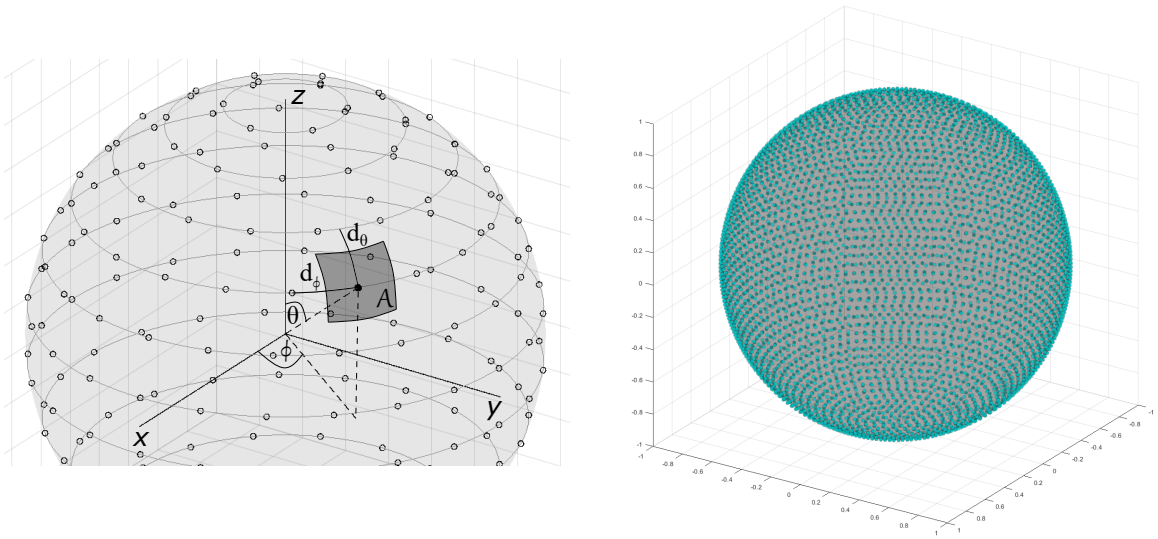


Figure 4.12: Illustration of the point on the sphere with denoted distance between latitude circles d_θ, spacing interval on the circle d_ϕ and average area per point A (left). Result of and algorithm with 5000 equidistant points on the sphere of radius $r_s = 1$ (right).

The subsequent stage in creating a path planning envelope involves constructing a concave hull $\mathcal{H}^*$ from the point cloud $\mathcal{R}$. This can be accomplished by utilizing the *ConcaveHull*

method provided by the PCL library. This method operates by processing Voronoi cells to extract boundaries from the input point cloud. The level of detail of the generated hull can be controlled using parameter α, which limits the size of the resulting hull segments.

Given the algorithm's focus on handling the complex geometries of structures, it's possible that certain features in the input point cloud are represented by a sparse number of points, such as the tip of a wind turbine blade. Consequently, some regions of the concave hull $\mathcal{H}^*$ may exhibit lower density than desired. While one approach could involve increasing the number of points N_s per sphere S_i, this might result in excessively long computation times for generating the path planning envelope, especially in cases with large input model $\mathcal{P}$. A more effective approach is to upsample the point cloud $\mathcal{H}^*$ to achieve uniform and predefined density, generating new point cloud $\mathcal{H}$.

Typically, upsampling involves sampling the local plane of each point in a circular manner with predefined parameters for radius r_{SLP} and step s_{SLP}. Alternatively, an uniform random distribution can be used to sample local plane of each point, ensuring constant point density throughout the cloud, determined by a specified parameter n_{RUD}. Smoothing of the resultant point cloud is executed by using the Moving Least Squares (MLS) algorithm, with adjustable parameters for region around point r_{MLS} and polynomial order p_{MLS}. This approach enhances the overall robustness of the algorithm by ensuring uniform density and consistent representation of features across the entire point cloud.

Once the point cloud $\mathcal{H}$ is generated encompassing the entire input model, inevitably, some points may reside below the surveyed structure. This problem is addressed through the utilization of the *CropBox* filter, which is available in the PCL library. Position of each point $\mathbf{h}_i$ in the point cloud $\mathcal{H}$ can be denoted by the vector $\mathbf{h}_i = \begin{bmatrix} h_{x,i} & h_{y,i} & h_{z,i} \end{bmatrix}^T \in \mathbb{R}^3$. The value $h_{z,min}$ is defined as the minimal value of the z-coordinate among all points:

$$h_{z,min} = \min(h_{z,i}), \quad i \in \{1, \ldots, m\} \tag{4.19}$$

where m represents the total number of points in $\mathcal{H}$. The parameters of the *CropBox* filter are configured such that every point $\mathbf{e}_i = \begin{bmatrix} e_{x,i} & e_{y,i} & e_{z,i} \end{bmatrix}^T \in \mathbb{R}^3$ within the constructed envelope $\mathcal{E}$ satisfies the condition:

$$e_{z,i} \geq h_{z,min} + r_s + a_{min}, \quad i \in \{1, \ldots, n\} \tag{4.20}$$

where n represents the total number of points in the point cloud $\mathcal{E}$, r_s is the desired trajectory distance from the model, and a_{min} denotes the minimum permitted flight altitude for the UAV with respect to the input model $\mathcal{P}$. It should be noted that condition4.20does not prohibit the generation of an envelope and consequently a flight path below the lowest point in the input cloud if parameter $a_{min} < 0$, which may be necessary for successful execution of a specific flight

tasks, such as a bridge inspection mission.

Given that the waypoints of the planned path will always be elements of the envelope $\mathcal{E}$, the same filter can be utilized to establish boundaries around the model, notably to restrict the maximum flight altitude above the surveyed structure.

The subsequent stage in the proposed algorithm involves determining the orientation of the UAV along the trajectory. To ensure that the UAV maintains a perpendicular heading to the surveyed structure, normal vectors to the surface of the envelope can be utilized. Given that the envelope was constructed from spheres with centers at points of the input point cloud, the direction of the normal vector at each point of the envelope points towards the center of the sphere and is perpendicular to the surface of the input model. The normal estimation method, employed for calculating the normal at every point of the envelope, is implemented in PCL library. This method calculates the normal at a selected point based on the estimated plane curve formed by neighboring points within a predefined radius determined with K-D Tree search method.

The points of the envelope $\mathcal{E}$ serve as nodes to construct a weighted graph, where key parameters for graph construction include the maximum number of neighboring points and the radius within which they are considered for connection to a selected point. Each edge of the graph is assigned a weight corresponding to its length, calculated as the Euclidean distance between points on the envelope.

The generation of a flight path for the UAV can be initiated by manually selecting several key points along the envelope that the UAV must visit. These key points serve as crucial locations essential for comprehensive coverage or targeted inspection. Utilizing the weighted graph derived from the envelope, the remaining waypoints between these selected key points can be determined. Dijkstra's algorithm can be applied to navigate this weighted graph, efficiently seeking the shortest path between nodes. As the algorithm iterates through the graph, it evaluates potential paths between the selected key points, progressively accumulating distances from the starting node to each reachable node. This process results in the generation of a comprehensive list of waypoints along the envelope, defining a safe and efficient flight path for the UAV.

This approach not only ensures precise navigation through the complex geometry of the surveyed structure but also optimizes the UAV's trajectory, minimizing flight time and energy consumption.

4.2.2Simulation results

To test the proposed path planning algorithm and validate its outcomes, extensive simulations were conducted, focusing particularly on the complex geometries encountered in bridges and wind turbines. These models were specifically chosen to simulate real-world scenarios en-

countered in projects like the *Autonomous System for Assessment and Prediction of Infrastructure Integrity* (ASAP) [67] and the *Autonomous UAV Inspection of Wind Turbine Blades* (AEROWIND) [66].

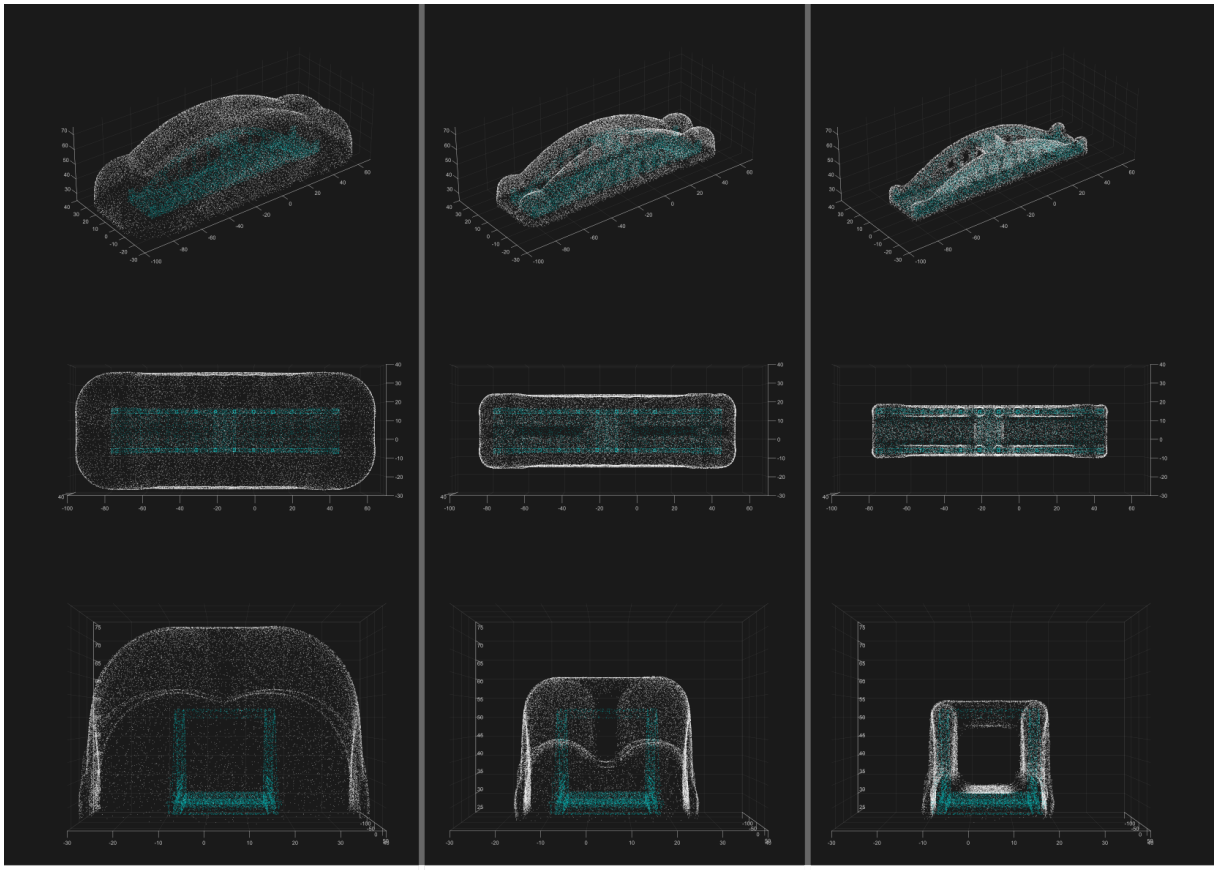


Figure 4.13: Simulations of the constructed path planning envelope around the bridge model with a different desired distance from it: 20m in case (a), 8m in case (b) and 2m in case (c).

The initial simulation experiment centered on generating a path planning envelope around a bridge model. The input point cloud was generated by sampling the mesh model, resulting in approximately 10000 points. Three distinct envelopes were created for this experiment, each with a different desired distance from the model: 20 meters, 8 meters, and 2 meters, denoted as cases (a), (b), and (c) respectively, as depicted in Figure4.13. Notably, all other parameters remained consistent across all cases, as detailed in Table4.1. In case (a), where the trajectory distance from the bridge is set to 20m, the envelope was generated only surrounding the bridge from the outside, restricting UAV flight paths from being planned within the bridge structure. In cases (b) and (c), designed envelope produced a corridor between the two sides of the bridge, with the latter scenario allowing for flight paths to even pass through narrow gaps between

columns on the side of the bridge.

Table 4.1: Parameters for generating path planning envelopes around the bridge.

Parameter	Value
Path distance from model r_s	$20m/8m/2m$
Input model voxel size	$1.0m$
Number of points per sphere N_s	1000
Spheres voxel size	$0.8m$
Concave hull α	4.0
MLS search radius r_{MLS}	$5.0m$
MLS polynomial order p_{MLS}	2
Upsampling method	SAMPLE_LOCAL_PLANE
Upsampling local plane radius r_{SLP}	$2.0m$
Upsampling local plane step size s_{SLP}	$0.3m$
Min. permitted flight altitude a_{min}	$0.0m$

The second simulation experiment entailed generating an envelope around the model of a wind turbine and planning various flight paths by manually selecting sets of key points for inspection. Figure4.14illustrates a flight path planned for inspecting all three blades of the wind turbine from both the leading and trailing edges. In addition to the marked positions, each waypoint is accompanied by a vector representing the desired heading of the UAV. A notable aspect of the depicted results is that the UAV maintains a consistent distance from the nearest point of the wind turbine. Furthermore, the UAV's heading remains perpendicular to the wind turbine model throughout the entire trajectory.

This aspect is crucial for maintaining a constant focus of the camera, enabling the acquisition of highly detailed photographs of the blades. Such detailed data is essential for detecting even minor damages, ensuring thorough coverage and precise inspection of the turbine blades. This approach enhances the reliability of the inspection process, allowing for early detection and mitigation of potential issues, thus contributing to the overall integrity and efficiency of wind turbine operations.

A detailed analysis of the simulation outcomes, including a comparison between flight paths planned using both the algorithm based on Huygens's wave propagation principle and the algorithm based on isolines generated by potential fields presented in Chapter5, is provided in Section5.3.3.

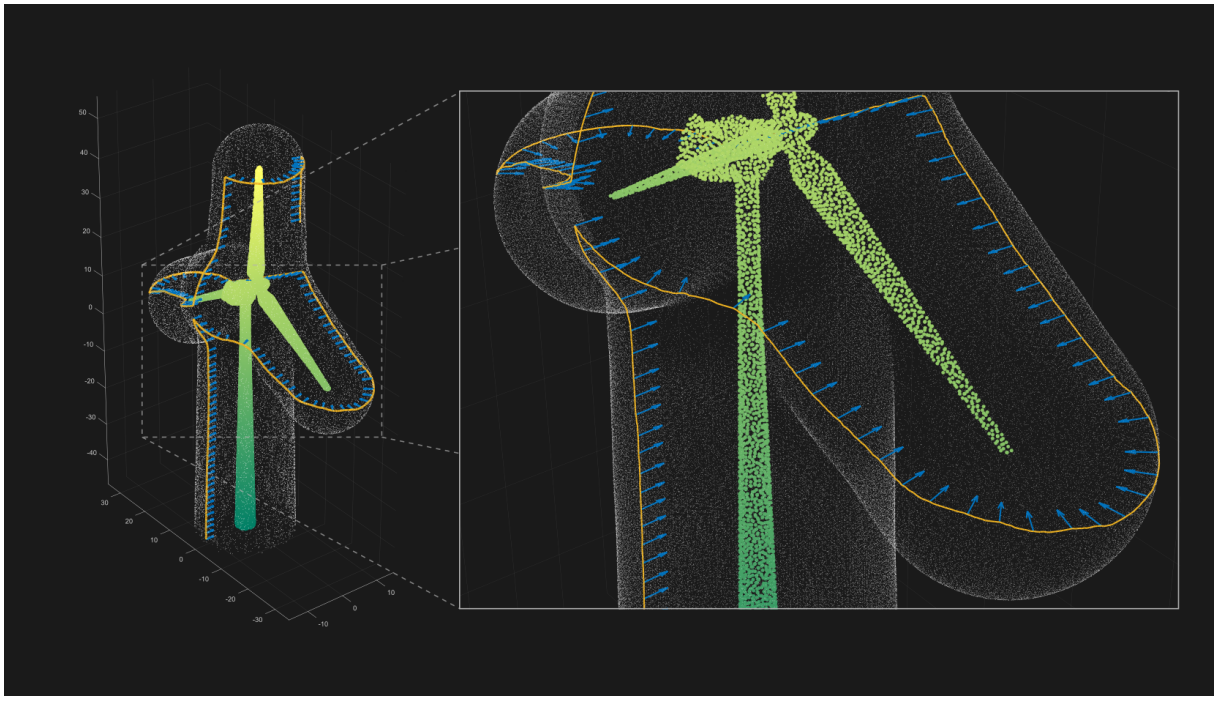


Figure 4.14: Simulation of the constructed envelope around the wind turbine model and planned inspection path with denoted orientation of the UAV along it.

4.2.3Experimental results

After validating the path planning method in simulation scenarios, real-world experiments were conducted on two complex structures: a wind turbine and a bridge. For these experiments, a large-scale carbon-fiber custom-made *Kopterworx* quadcopter was utilized. The quadcopter was equipped with a *Cube Orange* autopilot running *Arducopter* firmware for low-level attitude control, along with an *Intel NUC* onboard computer. Real-time model registration and matching were facilitated by data from an *Ouster OS0-128* LiDAR sensor mounted on the UAV.

The wind turbine used in the experiments had blades measuring 30 meters in length and a pole height of 60 meters. These dimensions provided sufficient data to define and construct a generalized model of the wind turbine for path planning purposes.

The primary challenge in planning an inspection trajectory for a wind turbine in real-world scenarios lies in the uncertainty surrounding the orientation of the nacelle. The blades of the turbine can be stopped at arbitrary positions, meaning it is possible to achieve that their configuration is aligned with the predefined model used for planning the flight trajectory. However, for technical reasons aimed at reducing strain on the blades during periods of turbine inactivity, the nacelle is typically positioned parallel to the current wind direction. This dynamic orientation prevents the static geographical coordinates of the wind turbine from serving as reliable markers for trajectory planning.

To address this challenge, a solution involves initially planning the inspection path using the envelope derived from the wind turbine model. Subsequently, the planned trajectory is

dynamically adjusted and aligned in the real world based on the current orientation of the wind turbine. This alignment is achieved through the real-time construction of a point cloud model of the wind turbine using scans from LiDAR sensors. These scans facilitate the process of model registration and matching with the input model, enabling the accurate placement of the planned trajectory within the world reference frame.

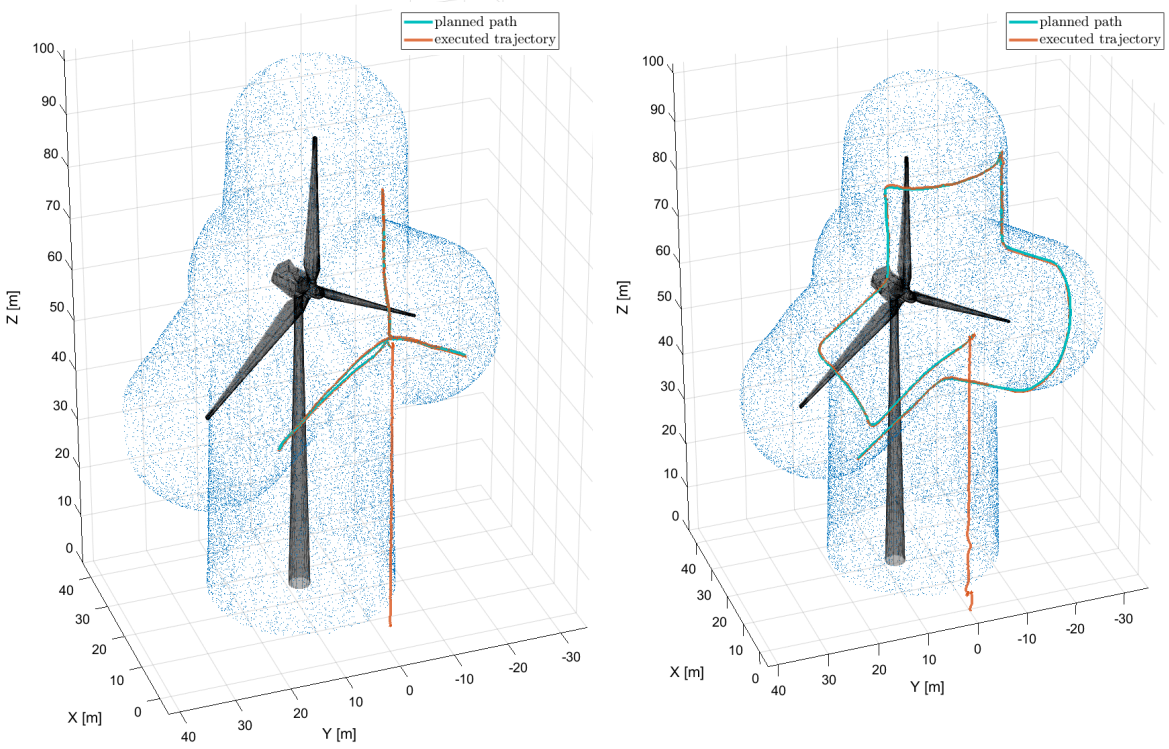


Figure 4.15: Planned path and executed trajectory in real-world experiments of autonomous wind turbine inspection. The first case depicts a simple inspection path in front of the wind turbine, while the second one illustrates a more complex trajectory that navigates around the turbine blades.

By correctly aligning the planned trajectory with the current orientation of the wind turbine, it is ensured that the UAV executes the inspection at the desired and safe distance from the turbine, while also collecting precise data necessary for the detection of any damages on the blades. This integration of real-time sensor data and model registration enhances the adaptability and effectiveness of the inspection process, enabling efficient and accurate assessment of condition and integrity of the wind turbines.

Results of the experiments are shown in the Fig4.15depicting model of the wind turbine used for path planning, generated envelope, as well as planned and executed flight trajectories. Inspection trajectory was planned only for front side of the wind turbine in case (a) at desired distance of 15m from the blades. In case (b) trajectory was designed for the UAV to be able to acquire data from top and bottom side of the blades.

The experimental results are presented in Fig.4.15, which illustrates the wind turbine model utilized for path planning, the generated envelope, as well as the planned and executed flight

trajectories. In case (a), the inspection trajectory was specifically planned for the front side of the wind turbine, maintaining a desired distance of 15m from the blades. Meanwhile, in case (b), the trajectory was designed to enable the UAV to capture data from both the top and bottom sides of the blades.

The second part of the experiments focused on a bridge designed for vehicular traffic. The bridge featured a traffic lane with a width of 5 meters and two side arcs supported by a series of columns. The arcs had a height of 6 meters and were connected at the top. While simulations demonstrated the feasibility of generating flight paths using the envelope within the bridge, safety considerations led to the decision to conduct flight trajectories exclusively for surveying the bridge from the outside during the real-world experiments. Figure4.16illustrates the results of the experiments, including the bridge model, the generated envelope, and the planned and executed flight trajectories.

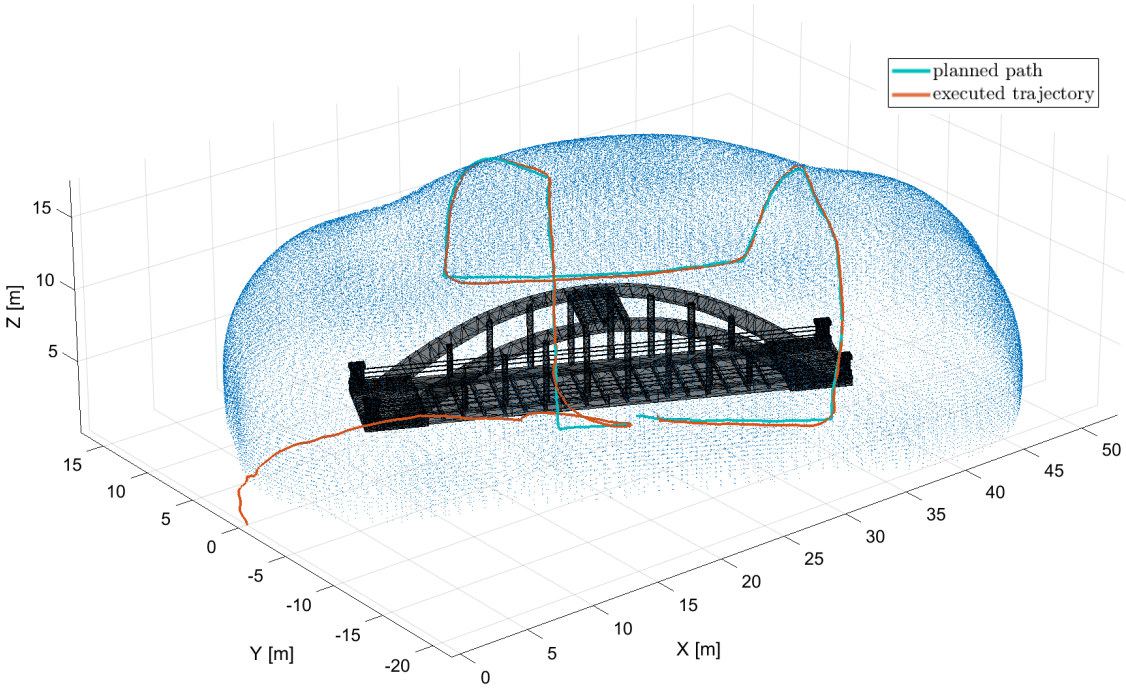


Figure 4.16: Results of algorithm testing in a realistic inspection scenario involving a bridge as the target object.

CHAPTER 5

Inspection flight path planning based on isolines

The primary objective is to develop a comprehensive and robust path planning method for UAV inspection missions applicable to both known and unknown structure layouts or models. The concept involves utilizing closed isolines of an artificial potential field generated by the surveyed structure to plan a flight path at the desired distance from it. The source of the potential field can either be available information about the structure's outline used for offline path planning or data acquired during the flight for real-time path generation during mission execution.

By selecting an appropriate function to generate the artificial potential field, the value of the field at any given point in space corresponds to a specific distance between that point and the source of the field. Generating a map of potential field isolines around the known surveyed structure for offline path planning or around the UAV for real-time planning allows the construction of a list of waypoints for the UAV to follow, ensuring it circumnavigates around the object while maintaining the desired distance and orientation with respect to it.

The development of the algorithm is divided into two stages based on the available information about the surveyed object. In the first case, the information about the structure is known and available in the form of an object layout or model. In the second case, the surveyed object is considered unknown, with only information about its location available, which can represent any vertex or point within or on the edge of the object.

5.1Basic concepts of potential fields in robot navigation

Potential fields are inspired by concepts of potential energy and force fields for guiding the autonomous robots through complex environments towards predefined goals while avoiding obstacles. In physics, a potential field is a scalar field where each point in space is associated with a scalar value representing the potential energy at that location. Under the influence of forces, objects tend to move towards regions of lower potential energy, following the gradient of the potential field. Using this principle of gradient descent, position of the object is iteratively adjusted in order to minimize its potential energy and reaching a state of equilibrium.

In the context of UAV navigation, potential fields are utilized to create artificial force fields that direct the UAV towards the goal destination while simultaneously repelling it from obstacles. The goal attraction field, $U_a(\cdot)$, generates a force that pulls the UAV towards the desired destination, while the repulsion field, $U_r(\cdot)$, creates repulsive forces around detected obstacles, pushing the UAV away from potential collisions. By combining these attractive and repulsive forces, potential fields provide a mechanism for autonomous navigation that corresponds to the behavior of physical objects moving through force fields. With position of the UAV represented by vector $\mathbf{q} = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$ total potential field $U_t(\cdot)$ can be defined with equation:

$$U_t(\mathbf{q}) = U_a(\mathbf{q}) + U_r(\mathbf{q}). \tag{5.1}$$

Potential fields are represented as scalar functions defined in the area occupied by the UAV and its surrounding environment. The potential field function assigns a scalar value to each point in space, indicating if that location repels or attracts the UAV. Typically, attractive potentials are assigned to regions near the goal destination, while repulsive potentials are assigned to regions surrounding obstacles. The resultant potential field is computed by combining the attraction and repulsion potentials, resulting in a composite field that guides the UAV towards the goal while avoiding collisions with obstacles. The entire environment can be represented as map of isolines of different levels of resultant potential field. An illustration of the potential field, as well as isoline map and gradient representation in an environment featuring one obstacle and one goal point is depicted in Fig.5.1.

Navigation in potential fields is achieved by following the gradient of the resultant potential field. The gradient represents the direction of steepest ascent or descent within the potential field, guiding the UAV towards regions of lower potential while steering it away from regions of high potential. By continuously adjusting its velocity based on the gradient of the potential field, the UAV can navigate through complex environments, dynamically adapting its trajectory to avoid obstacles and reach its target goal.

One of main advantages of navigation algorithms based on potential fields is their capacity

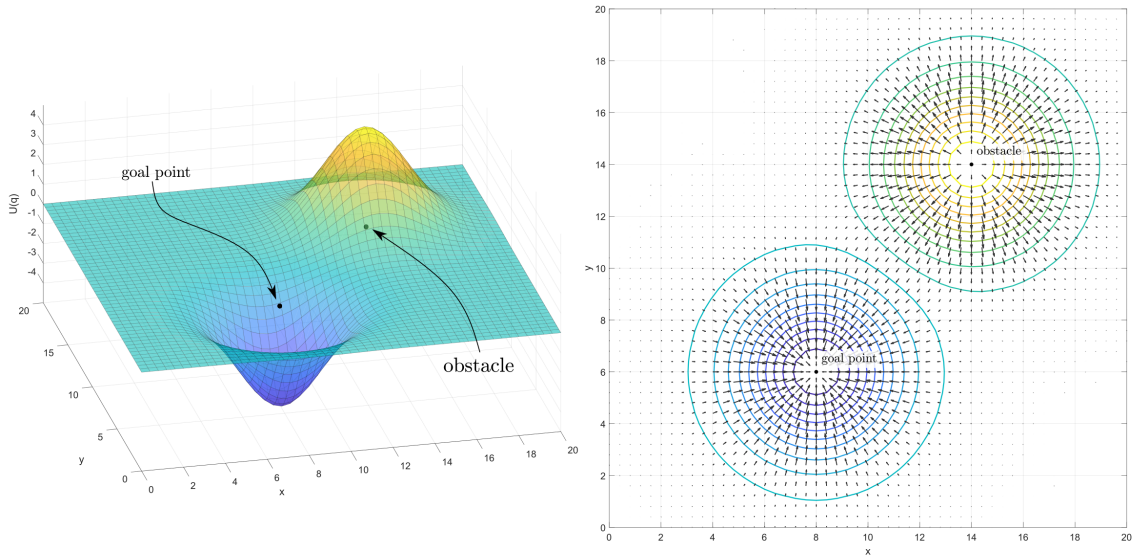


Figure 5.1: Artificial potential field defined with Gaussian distribution function and represented as 3D surface in area with one obstacle and one goal point (left). Map of isolines representing regions of the same potential level and gradient map of the resultant potential field (right).

to adapt to dynamic changes in the environment. As the UAV moves through the environment and encounters new obstacles or changes in the terrain, the potential fields can be dynamically updated to accurately represent environmental conditions. This enables real-time navigation and ensures that the UAV can respond effectively to unknown obstacles or disturbances.

The implementation of potential fields for UAV navigation involves several steps, including environment modeling, potential field generation and path or trajectory planning. Environment modeling is based on perception sensors such as LiDAR, cameras, or radar to detect obstacles and define their positions and shapes in the reference frame of the UAV. LiDAR sensors provide high-resolution point cloud data, enabling precise obstacle detection and localization, cameras offer visual information of the surrounding environment, while radar systems can be utilized in detecting obstacles even in unfavorable weather conditions. By processing sensor data, navigation algorithms generate a comprehensive environmental map representing obstacles, terrain features, and the goal destination.

Based on the environmental data provided by sensors, artificial potential fields are then modeled using the various mathematical functions, such as inverse distance functions or Gaussian distributions. Functions defining the potential fields must satisfy specific criteria, including continuity and differentiability. Values of the repulsive functions should approaching infinity in close proximity to obstacle and gradually diminish in magnitude as the distance from the obstacle increases.

While potential fields provide simple and effective navigation tools, they also encounter several challenges and limitations. One common limitation of potential field methods is their

susceptibility to local minima, where the UAV may become trapped in valleys of the potential field and fail to reach the goal destination. This can occur in complex environments with narrow passages or multiple obstacles, when the total force exerted by the potential field approaches zero or becomes sufficiently small, resulting in no significant displacement of the UAV before reaching the goal position.

Achieving optimal performance of potential field-based navigation systems often requires careful tuning of various parameters, including the shape and scale of the potential fields, obstacle avoidance thresholds, and control gains. Finding the right balance between attractive and repulsive forces is crucial to ensure stable and efficient navigation, particularly in complex environments where obstacles may move or change over time.

Another limiting factor to be considered in the implementation of real-time navigation algorithms is computation requirements, particularly for resource-constrained UAV platforms with limited onboard processing capabilities. This can present additional challenges in complex or cluttered environments by increasing the amount of data provided by perception sensors and potentially leading to delays in navigation control.

5.2Planning method based on closed PF isolines of known structure

The path planning algorithm based on closed isolines of the artificial potential field (CI-APF) was initially developed for scenarios where information about the surveyed structure is known before flight. The surveyed structure can be represented either by a two-dimensional layout or a three-dimensional model of the object. This scenario is considered simpler for path planning, as the entire inspection path can be planned and verified before flight, significantly reducing the risk of collision as well as ensuring the complete coverage of the object and the high quality of acquired data.

5.2.1Input data processing

In cases where the input data is the 2D layout outlining the object, similar procedures as those used for the path planning algorithm based on Huygens's wave propagation principle (described in Section4.1) can be applied. Each point $\mathbf{b}_i$ in the input set $\mathcal{B}$ corresponds to a corner of the structure with Cartesian coordinates. A sampled representation of the building layout can be achieved through linear interpolation between adjacent corners, with a resolution parameter denoted as r_b, resulting in a new set of points $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$. This set of points represents sources of potential fields used for flight path planning.

Another representation of the surveyed structure can be a 3D model in the form of a mesh

or a point cloud. The mesh model can be transformed into a point cloud using a sampling process through various 3D modeling applications. Since the input cloud can be in the form of unorganized points with uneven and unnecessarily high density, a *VoxelGrid* filter is applied to the input dataset.

To further reduce the computational complexity and execution time of the algorithm, the fact that the typical surveying mission pattern consists of the UAV passing around the structure at different altitudes several times, maintaining a constant altitude through each pass, can be exploited. Instead of calculating the potential field for every point in the input cloud $\mathcal{S}^*$, the calculation is limited to the vicinity Δ_h around the flight altitude of the current pass h_d. By filtering the input set $\mathcal{S}^*$ and creating slices of the original point cloud $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ along its z-axis at the desired altitude h_d and thickness determined by the non-negative parameter Δ_h, sources of potential fields are created. As a result of the filtering process, every point $\mathbf{s}_i = \begin{bmatrix} s_{x,i} & s_{y,i} & s_{z,i} \end{bmatrix}^T \in \mathbb{R}^3$ in the point cloud $\mathcal{S}$ satisfies the condition:

$$h_d - \Delta_h \leq s_{z,i} \leq h_d + \Delta_h, \quad i \in \{1, \ldots, n\} \tag{5.2}$$

5.2.2Isolines generation

The next step of the CI-APF algorithm involves creating a two-dimensional rasterized area $\mathcal{A} = \mathbf{a}_1, \ldots, \mathbf{a}_m$ around the outline of the object, represented by the set of points $\mathcal{S}$, with a defined resolution λ_a. The edges of area $\mathcal{A}$ define the boundaries within which the inspection flight path can be planned. Since planning the flight path within the outlined object is not permitted, these points must be filtered out. The outline of the object forms a closed area represented as a polygon with known vertices. A Ray Casting Algorithm, illustrated with Fig. 5.2, can be utilized to determine whether a point $\mathbf{a}_i$ is inside that area, choosing it to be excluded from later calculations of the potential field.

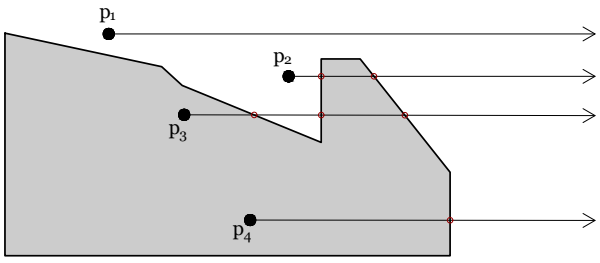


Figure 5.2: Illustration of the Ray Casting Algorithm used to determine if a point is inside a closed polygon. Rays originating from points p_1 and p_2 have an even number of intersections with the edges of the polygon, indicating that they are outside of it. Rays from points p_3 and p_4 have an odd number of intersections, identifying them as inside the polygon.

The Ray Casting Algorithm involves casting a ray from the point $\mathbf{a}_i$ and counting the number of intersections with the edges of the polygon. If the number of intersections is odd, the point is inside the polygon; otherwise, it is outside. Since the direction of the ray does not effect results of the algorithm it can be chosen arbitrarily. This provides a simple and efficient method for reducing the number of points in the set $\mathcal{A}$ for which a potential field has to be calculated.

For each point $\mathbf{a}_i$ outside the outline of the object, the algorithm identifies the closest point $\mathbf{s}_i^s$ using the Euclidean distance between them. Point $\mathbf{s}_i^s$ serves as the source of the potential field calculated at the point $\mathbf{a}_i$:

$$\mathbf{s}_i^s = \arg\min_j(\|\mathbf{s}_j - \mathbf{a}_i\|), \quad j \in \{1,\ldots,n\} \tag{5.3}$$

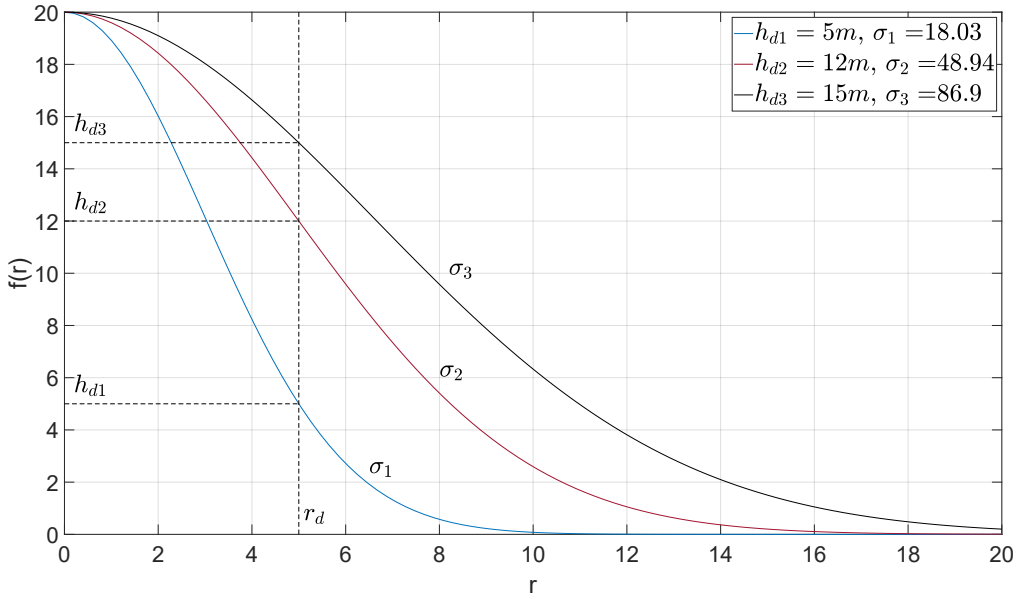


Figure 5.3: Illustration of the potential field functions $f(r)$ in the form of Gaussian distributions. The parameter σ_i, which determines the shape of each function, is calculated to ensure that each function attains a desired value h_{di} at the point indicated as r_d.

The selected function to define the artificial potential field U is a the form of a Gaussian distribution function. This function exhibits a bell-shaped curve and is characterized by parameters H and σ, with the variable r representing the Euclidean distance between the source of the potential field and the point at which it is calculated. The function is expressed as:

$$f(r) = He^{\frac{-r^2}{\sigma}} \tag{5.4}$$

The value of the potential field function $f(r)$ represents the relative flight altitude of the UAV with respect to the height of the surveyed object. The parameter H limits the maximum allowed altitude with respect to the altitude of the UAV during approach to the object, while σ deter-

mines the shape of the potential field function. σ is specifically chosen such that the value of the potential field function $f(r)$ equals the desired relative flight altitude of the UAV, denoted as h_d, at a desired horizontal distance $r = r_d$ from the object that the UAV should maintain while circumventing around it. Using the equation5.4 σ is calculated as:

$$\sigma = \frac{-r_d^2}{\ln\left(\frac{h_d}{H}\right)} \tag{5.5}$$

Utilizing the calculated σ, the value of the artificial potential field U at point $\mathbf{a}_i$, generated by the closest point in the object outline $\mathbf{s}_i^s$, is determined using the equation:

$$U(\mathbf{a}_i) = He^{\frac{-\|\mathbf{s}_i^s - \mathbf{a}_i\|^2}{\sigma}} \tag{5.6}$$

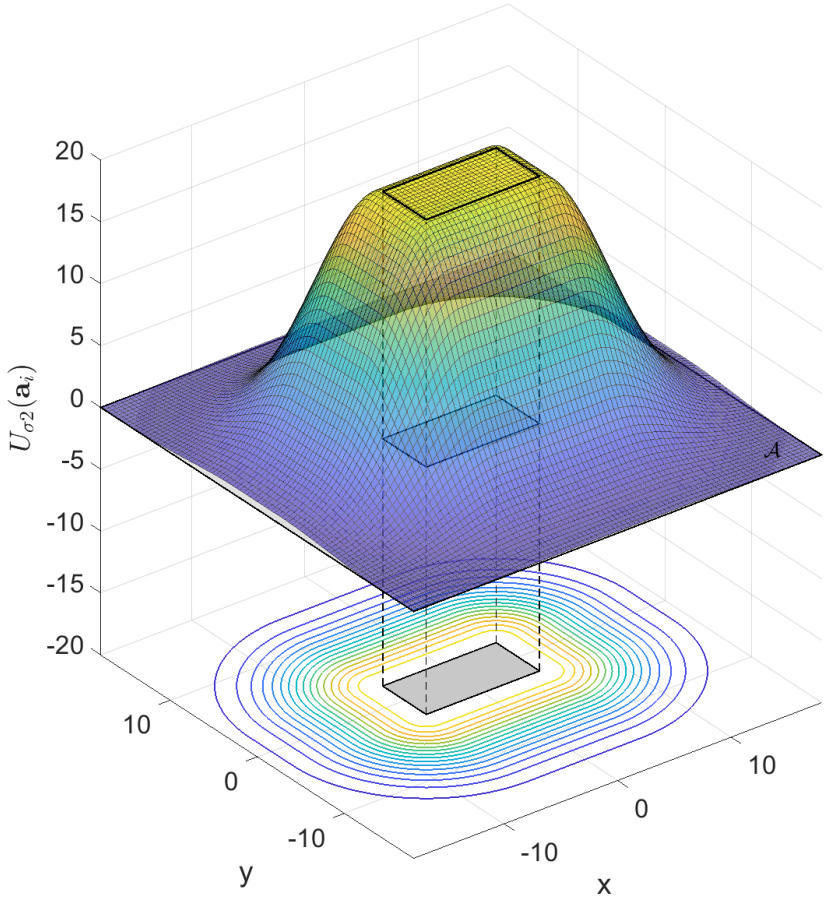


Figure 5.4: The potential field is defined by equation5.6with parameters $H = 20$ and $\sigma = 48.94$. This potential field is generated in the area around the rectangular object, with discrete value levels depicted through isolines.

To define closed isolines of the artificial potential field, the CI-APF algorithm uses the Marching Squares method [83] to generate contours for a two-dimensional scalar field. The initial step involves creating a binary field by applying a threshold to all the points in the set $\mathcal{A}$. The isovalue threshold corresponds to the value of the contour being calculated. By setting the threshold equal to h_d, a contour is generated at the value of the potential field U corresponding to the desired relative flight altitude h_d, thus forming the closed isoline around the object at the desired distance r_d in the horizontal plane. In the binary field, values are set to 1 where the data is above the isovalue and 0 where it is below.

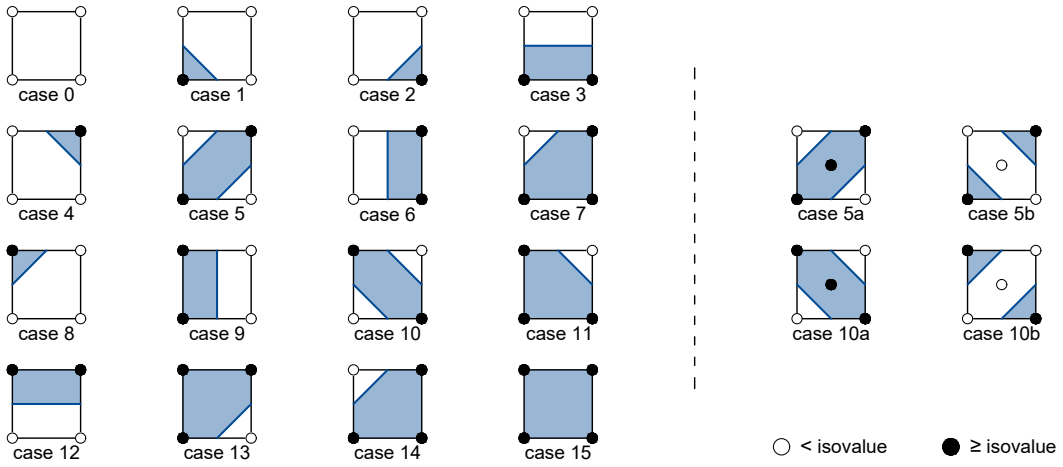


Figure 5.5: Lookup table for generating contour lines with marching squares algorithm.

Each 2×2 block of data in the binary field forms a cell for which its index is calculated. For each cell, the algorithm utilizes a pre-built lookup table, depicted in Fig.5.5, to select the correct isoline segment. Linear interpolation is then applied using the original data from the field $\mathcal{A}$ to determine the correct position of the endpoints of the isoline segment along the edges of the cell. This process ensures the accurate representation of the closed isolines of the artificial potential field around the object.

The orientation of the UAV during the inspection mission is crucial for effective data acquisition. Ensuring that the UAV maintains a perpendicular heading to the building layout enhances the quality and efficiency of the inspection process. To achieve this, the algorithm computes the angle ψ_i at each point $\mathbf{a}_i$ located outside the building outline, relative to the previously determined closest point $\mathbf{s}_i^s$:

$$\psi_i = \text{atan2}(\mathbf{a}_{y,i} - \mathbf{s}_{y,i}^s, \mathbf{a}_{x,i} - \mathbf{s}_{x,i}^s). \tag{5.7}$$

During the linear interpolation phase of the Marching Squares Algorithm, an additional data field for the angle ψ_i is included for each input point. This information is then utilized to

adjust the calculated heading for the UAV along the planned path, ensuring that it maintains a consistent perpendicular orientation to the building layout throughout the inspection mission.

The output of the CI-APF path planning algorithm is an ordered list of waypoints, which is subsequently transferred to the MPC tracker module integrated within the UAV controller. Each waypoint contains information regarding the precise position and orientation of the UAV within the global reference frame of the environment, while velocity and acceleration constraints are integrated trough MPC tracker generating fully defined flight trajectory. The Flight path, represented by discrete waypoints planned with CI-APF algorithm to circumnavigate around the object with known 2D layout is depicted in the Fig.5.6.

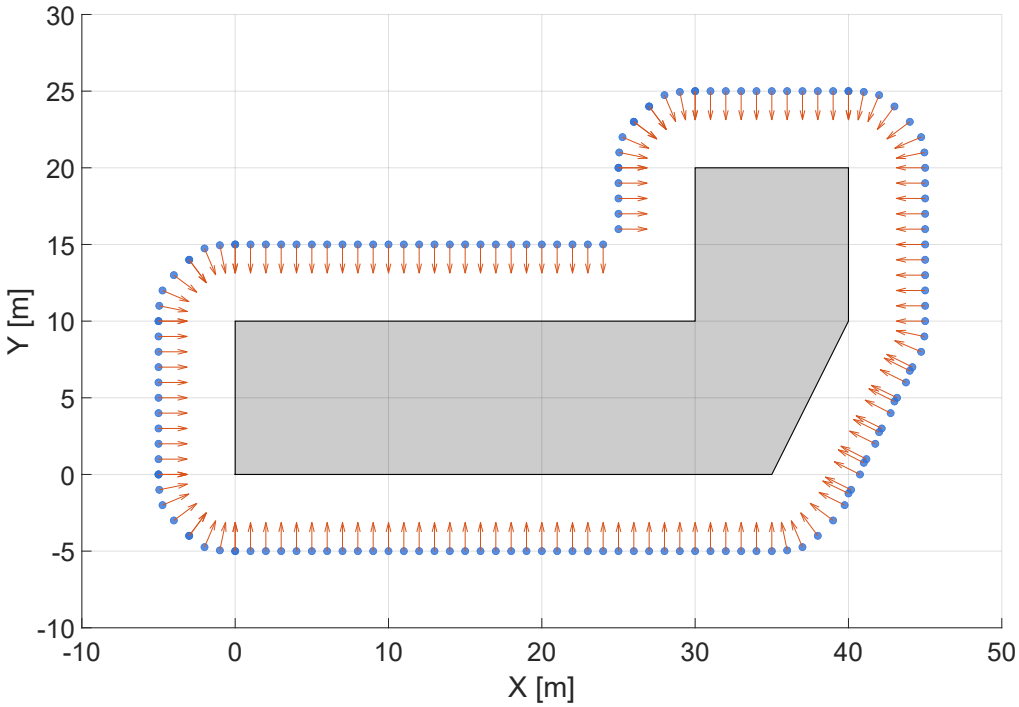


Figure 5.6: Flight path planned with CI-APF algorithm around known 2D layout of the object. Waypoints are visualized with blue markers, while the orientation of the UAV is denoted by orange arrows. The desired distance from the object is set at $r_d = 5m$, with a discretization resolution of $\lambda_a = 1m$ for the planning area.

5.3Planning method based on closed PF isolines of unknown structure

The second phase of developing a UAV path planning algorithm based on closed isolines of artificial potential fields (CI-APF) aims to eliminate the requirement for initial knowledge of the precise structure layout or model. Instead, the algorithm relies solely on approximate location data of the object and information provided by sensors onboard the UAV. The primary

objective is to enable real-time path planning while simultaneously scanning the structure. This algorithm ensures that the UAV maintains a constant distance from the surveyed object while circumventing its entire outline. Additionally, it aims to keep the UAV's orientation perpendicular to the sides of the structure throughout the entire flight path, optimizing data acquisition and inspection efficiency.

The approximate location of the structure serves as the starting point for the survey mission, allowing the UAV to approach the object and initiate data acquisition with the LiDAR sensor. This approach offers flexibility, as the take-off point for the UAV does not need to be in close proximity to the surveyed object, meaning that structure initially does not have to be within the scanning range of the LiDAR sensor.

Once the UAV acquires data from the LiDAR sensor, the point cloud dataset is utilized to generate an artificial potential field. By calculating the isolines of the potential field in the area surrounding the UAV, a local flight path is dynamically generated in real-time for a segment of the structure. As the UAV completes each pass around the structure, the algorithm continuously generates closed isolines of the potential field at the desired distance from the surveyed object, ensuring comprehensive coverage of the entire structure.

The system architecture overview is illustrated in Fig.5.7. It defines the user input for the survey mission, consisting of the object location, the desired distance from the object that the UAV should maintain while circumnavigating around it (r_d), the relative flight altitude concerning the surveyed structure for each pass (h_d), as well as the velocity and acceleration constraints for the MPC tracker module. The CI-APF path planning module generates a list of flight path points, with each waypoint containing information for the UAV's position and orientation, which serves as input for the MPC tracker. The output from the MPC tracker contains trajectory points that function as a reference for UAV control. Feedback from UAV sensors for both the planning and control modules includes estimated odometry and LiDAR point cloud scans.

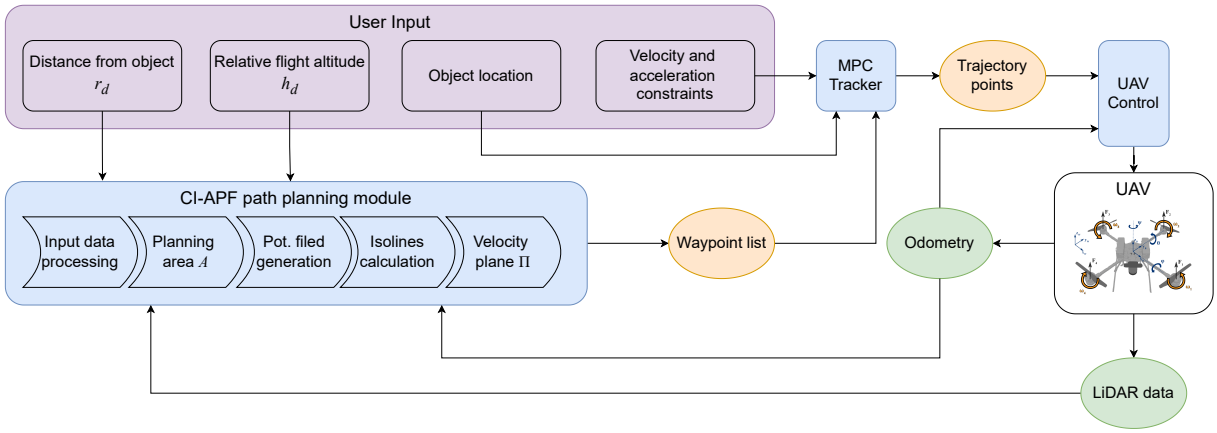


Figure 5.7: System architecture containing CI-APF module for online path planning.

5.3.1Input data processing

Depending on the type and parameters of the LiDAR sensor providing information for real-time path planning, its raw measurements in the form of a point cloud may contain noise or be excessively dense. To enable the implementation of the CI-APF algorithm on an onboard computer with limited computational power, filtering and downsampling of the input point cloud $\mathcal{P}^* = \{\mathbf{p}_1^*, \ldots, \mathbf{p}_k^*\}$ are performed.

Initially, all filters are applied to the point cloud scan obtained from the LiDAR sensor in its reference frame S. The *CropBox* filter, integrated into the PCL library, is used to select data points contained in a box-shaped region centered around the sensor origin and aligned with its axes. The conditions for each point $\mathbf{p}_i^*$, defined by its position vector $\begin{bmatrix} p_{x,i}^* & p_{y,i}^* & p_{z,i}^* \end{bmatrix}^T \in \mathbb{R}^3$, to avoid exclusion are:

$$|p_{x,i}^*| \leq l_h, \quad |p_{y,i}^*| \leq l_h, \quad |p_{z,i}^*| \leq l_v, \quad i \in \{1, \ldots, m\} \tag{5.8}$$

Here, l_h and l_v determine the size of the box region in the horizontal and vertical planes, respectively, and are defined by parameters of the survey mission such as the desired distance from the side of the structure r_d, the difference in altitude Δ_h between each pass of the UAV around it, and k_o, representing an overlap factor between LiDAR scans in the vertical direction:

$$l_h \geq 2r_d, \quad l_v \geq \frac{\Delta_h}{2 - k_o} \tag{5.9}$$

Subsequently, downsampling of the cropped input point cloud is performed using the *VoxelFilter* from the PCL library. The outcome of this process is a point cloud $\mathcal{P}$, where each point p_i represents the centroid of the input points within each voxel in the three-dimensional grid constructed over the point cloud $\mathcal{P}^*$.

Following the filtering and downsampling steps, the set of points $\mathcal{P}$ is in the sensor frame of reference S and can be denoted as $\mathbf{P}_S$. The objective of the path planning algorithm is to generate waypoints for the UAV in the world reference frame W, where the UAV is represented by a state vector $\mathbf{x} = \begin{bmatrix} \mathbf{q}^T & \psi \end{bmatrix}^T \in \mathbb{R}^4$. This vector consists of the position $\mathbf{q} = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^T \in \mathbb{R}^3$ and the yaw rotation angle ψ around the z-axis, constrained within the interval $[-\pi, \pi)$. To achieve this, each point within $\mathbf{P}_S$ is transformed to the world frame W using the static transform $\mathbf{T}_R^S$ from the sensor frame S to the UAV frame R and the transform matrix $\mathbf{T}_W^R$ derived from the estimated state vector $\hat{\mathbf{x}}$:

$$\mathbf{P}_W = \mathbf{T}_R^S \mathbf{T}_W^R \mathbf{P}_S \tag{5.10}$$

5.3.2Online path planing method

The online path planning method operates on similar principles as described in Section5.2. However, main difference is in that the potential field is calculated around the current position of the UAV based on the information about the segment of the structure gained from LiDAR scan, rather than around the entire surveyed object. As a result, only a portion of the closed isoline at the desired level is generated.

The area $\mathcal{A}$ in the horizontal plane around the current UAV position $\mathbf{q}$ is defined within the world reference frame W and rasterized with a parameter λ_a. This area is bounded by l_{Ax}^{+} and l_{Ax}^{-} along the x-axis and by l_{Ay}^{+} and l_{Ay}^{-} along the y-axis, ensuring that every point $\mathbf{a}_i = \begin{bmatrix} a_{x,i} & a_{y,i} & a_{z,i} \end{bmatrix}^T \in \mathbb{R}^3$ satisfies the conditions:

$$q_x + l_{Ax}^{-} \leq a_{x,i} \leq q_x + l_{Ax}^{+}, \quad q_y + l_{Ay}^{-} \leq a_{y,i} \leq q_y + l_{Ay}^{+}, \quad a_{z,i} = q_z, \quad \forall\, \mathbf{a}_i \in \mathcal{A} \tag{5.11}$$

The values of l_{Ax}^{+}, l_{Ax}^{-}, l_{Ay}^{+} and l_{Ay}^{-} are determined based on the estimated current velocity $\dot{\mathbf{q}}$ and acceleration $\ddot{\mathbf{q}}$ of the UAV, calculated through the following equations:

$$\begin{bmatrix} m_{Ax}^{+} \\ m_{Ax}^{-} \\ m_{Ay}^{+} \\ m_{Ay}^{-} \end{bmatrix} = \begin{bmatrix} l_{Ax0} \\ -l_{Ax0} \\ l_{Ay0} \\ -l_{Ay0} \end{bmatrix} + \begin{bmatrix} k_{Av} & 0 & 0 & 0 \\ 0 & k_{Av} & 0 & 0 \\ 0 & 0 & k_{Av} & 0 \\ 0 & 0 & 0 & k_{Av} \end{bmatrix} \begin{bmatrix} \dot{q}_x \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_y \end{bmatrix} + \begin{bmatrix} k_{Aa} & 0 & 0 & 0 \\ 0 & k_{Aa} & 0 & 0 \\ 0 & 0 & k_{Aa} & 0 \\ 0 & 0 & 0 & k_{Aa} \end{bmatrix} \begin{bmatrix} \ddot{q}_x \\ \ddot{q}_x \\ \ddot{q}_y \\ \ddot{q}_y \end{bmatrix} \tag{5.12}$$

$$\begin{bmatrix} l_{Ax}^{+} \\ l_{Ax}^{-} \\ l_{Ay}^{+} \\ l_{Ay}^{-} \end{bmatrix} = \begin{bmatrix} \max(l_{Ax0}, m_{Ax}^{+}) \\ \min(-l_{Ax0}, m_{Ax}^{-}) \\ \max(l_{Ay0}, m_{Ay}^{+}) \\ \min(-l_{Ay0}, m_{Ay}^{-}) \end{bmatrix} \tag{5.13}$$

Here, l_{Ax0} and l_{Ay0} are positive constants defining the minimum area around the UAV, while k_{Av} and k_{Aa} are predefined parameters that modify the influence of the UAV's velocity and acceleration on the boundaries of the planning area. By setting the boundaries in this manner, the area $\mathcal{A}$ extends in the direction of the UAV's current velocity, thereby expanding the horizon for path planning in the UAV's direction of movement. This approach is important as it allows the MPC tracker to achieve higher velocities which could not be possible with shorter path inputs while simultaneously reducing the area for potential field calculation.

For each point $\mathbf{a}_i$, the closest point $\mathbf{s}_i^s$ is selected as the source of the artificial potential field using Equation5.3. The artificial potential field, defined by the same function as in Equation 5.4, is then calculated at every point in the area $\mathcal{A}$. Utilizing the Marching Squares Algorithm described in Section5.2.2, an isoline at the desired distance from the structure is generated in the area around the UAV.

To determine and lock the direction (clockwise or counterclockwise) of the UAV flying around the object, only a portion of the generated isoline is utilized to create a segment of the flight path. The selection of this segment is achieved by constructing the velocity plane Π_{vp}, representing the velocity of the UAV in the horizontal plane, and projecting the points of the isoline onto it. Velocity plane Π_{vp}, as well as generated and projected points of isoline are depicted in the Fig.5.8.

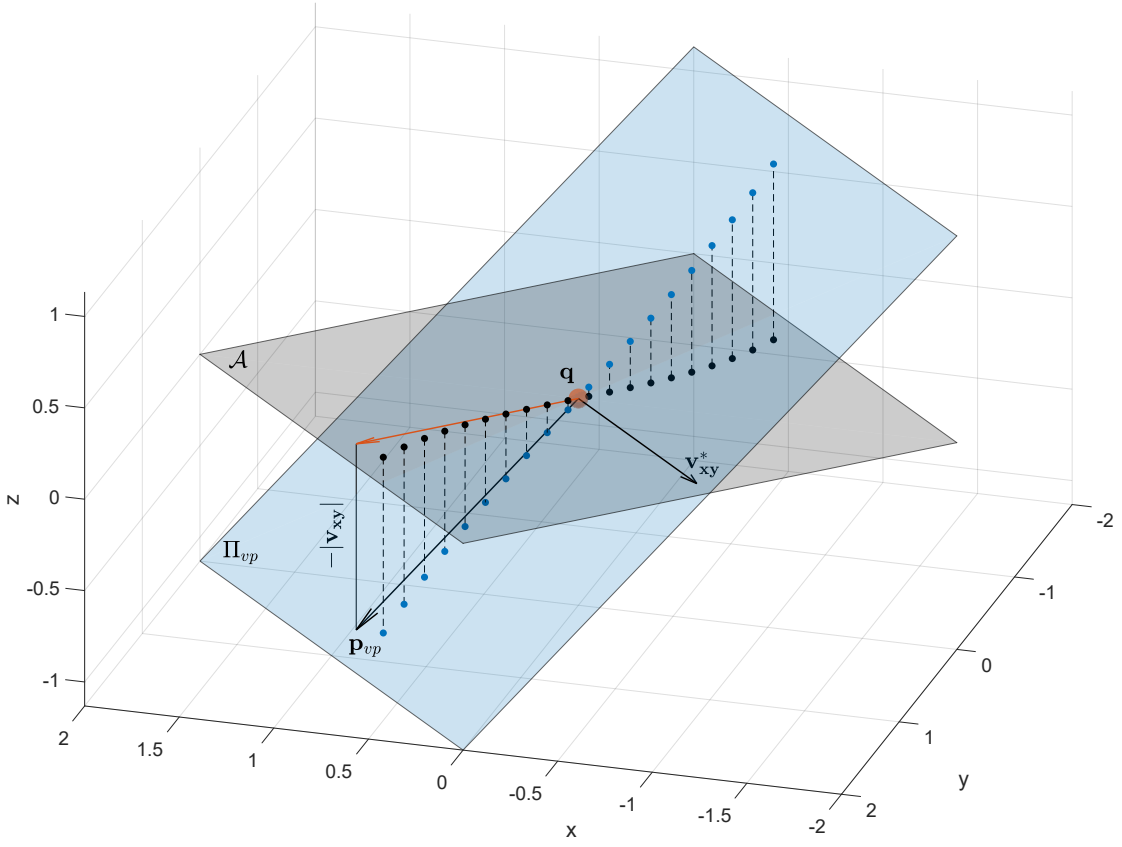


Figure 5.8: Visualization depicting the isoline points projected onto the velocity plane Π_{vp}.

The points on the isoline are denoted as $\mathbf{p}_i = \begin{bmatrix} p_{i_x} & p_{i_y} & p_{i_z} \end{bmatrix}^T \in \mathbb{R}^3$. With the position of the UAV denoted as $\mathbf{q}$, the linear velocity of the UAV in the world reference frame W is represented by the vector $\mathbf{v} = v_x\hat{i} + v_y\hat{j} + v_z\hat{k}$. The velocity of the UAV in the horizontal plane is defined by the vector $\mathbf{v_{xy}} = v_x\hat{i} + v_y\hat{j}$. Two vectors within the velocity plane Π_{vp} are defined: vector $\mathbf{v}^*_{\mathbf{xy}} = -v_y\hat{i} + v_x\hat{j}$, which is perpendicular to the velocity vector $\mathbf{v_{xy}}$, and vector $\mathbf{p_{vp}} = v_x\hat{i} + v_y\hat{j} - |\mathbf{v_{xy}}|\hat{k}$.

The normal of the velocity plane is obtained as the cross product of vectors $\mathbf{v}^*_{\mathbf{xy}}$ and **pvp**:

$$\mathbf{n_{vp}} = \mathbf{v}^*_{\mathbf{xy}} \times \mathbf{p_{vp}} \tag{5.14}$$

The velocity plane Π_{vp} is fully defined with the point $\mathbf{q}$ and the normal vector $\mathbf{n_{vp}}$. The isoline is projected onto the plane Π_{vp} with the direction of the parallel projection along $\hat{k}$ creating new set of points $\mathbf{p}^*_i = \begin{bmatrix} p^*_{i_x} & p^*_{i_y} & p^*_{i_z} \end{bmatrix}^T \in \mathbb{R}^3$. Only points $\mathbf{p}_i$ from the original set whose projection satisfies the condition $p^*_{i_z} < q_z$ are selected as waypoints for the flight path.

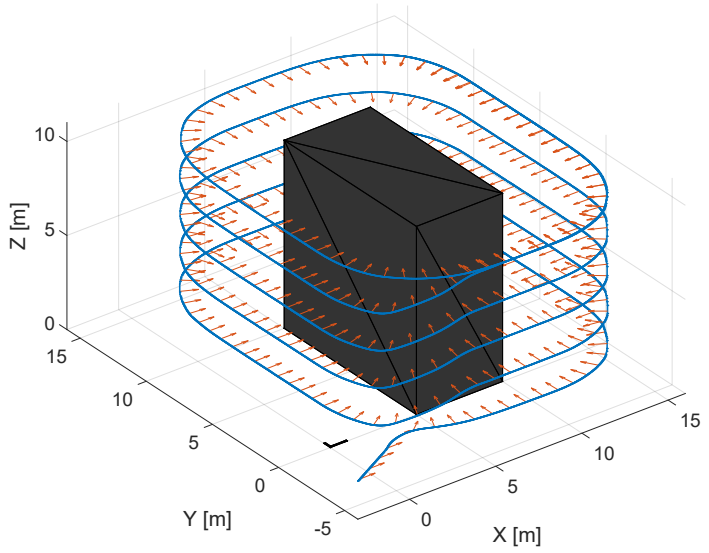
5.3.3Simulation results

To analyse and validate the efficacy of the proposed CI-APF algorithm for real-time path planning, a comprehensive series of simulation experiments was conducted. These experiments aimed to test the solution provided by the CI-APF algorithm across environments containing different survey objects unknown before flight. Additionally, they aimed to compare the performance of the CI-APF algorithm with that of path planners introduced in Chapter4and an offline path planner based on potential fields, as detailed in Section5.2. Surveyed objects varied in complexity, ranging from simple box-shaped models to realistic models of wind turbines. Different velocity and acceleration constraints were imposed on the UAV in the simulation cases.

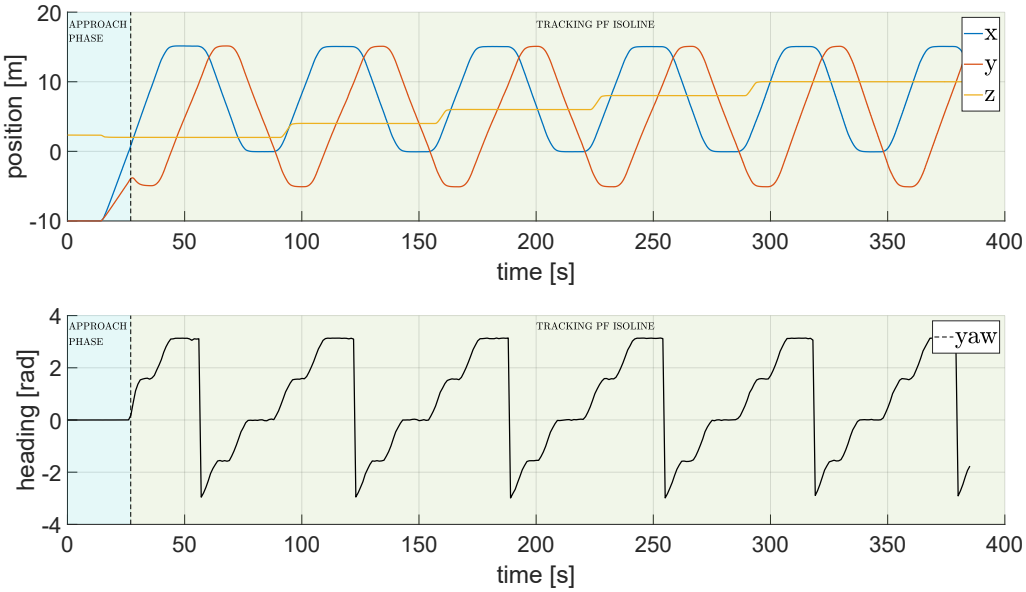
The experiments were executed within the Gazebo simulation environment, utilizing various models representing survey target structures. The simulation employed a model of the Kopterworx custom UAV, as outlined in Chapter3, equipped with a *Velodyne VLP-16* LiDAR sensor. To enhance the realism of the simulations, Gaussian noise was deliberately introduced to the LiDAR measurements, simulating real-world conditions more accurately.

The initial simulation scenario involved a simple box-shaped surveying object measuring $5m \times 10m \times 10m$. The desired distance r_d between the UAV and the sides of the object was set at $5m$, while the relative flight altitude, h_d, with respect to the surveyed object for the first pass was set to $2m$. Subsequent circumnavigation passes were conducted at increased altitudes, incremented by $2m$ each time. The simulation results, depicted in Figure5.9, illustrate the UAV's trajectory in 3D space, with its heading marked by orange arrows, alongside positional and yaw angle graphs in the time domain. The experiment started with an approach phase, during which the UAV navigated towards the predefined location of the object. As it neared the object, an isoline representing the potential field value equal to the parameter h_d was generated in the vicinity of the UAV. Subsequently, a new set of waypoints was computed, guiding the UAV to traverse along the generated isoline around the surveyed object while maintaining a distance of $5m$. Throughout the flight, the UAV kept a perpendicular heading relative to the object. Upon reaching the full height of the object the UAV maintained the same altitude for

the subsequent passes around the object.



(a) Executed UAV trajectory in 3D space around the unknown target object.



(b) Position and heading angle of the UAV in time domain with marked approach phase (blue) and APF isoline tracking phase (green).

Figure 5.9: Simulation results in a simple environment with box-shaped surveying target object.

The subsequent simulation experiments featured a different array of surveyed models, each presenting varying degrees of complexity. The trajectories executed by the UAV, guided by the CI-APF algorithm, around structures representing inspection targets, are visualized in Figure 5.10. In cases (a) and (b), the surveyed objects comprised a combination of geometric elements such as cylinder sections, slopes, and detached parts, demonstrating the algorithm's adaptabil-

ity to complex structures. Case (c) involved an inspection mission around a structure featuring a large dome. Notably, each pass of the UAV followed the curvature of the dome, with a diminishing radius, demonstrating the algorithm's ability to navigate along irregular surfaces. In case (d), the UAV traversed around three bridge pillars, while in case (e), it navigated through four environment with large poles. Case (f) presented a unique scenario—a simulation of inspecting the interior of a dome—using the same principles applied for generating isolines for circumventing surveyed objects. This demonstrates the versatility of the algorithm in addressing specialized inspection tasks within confined spaces.

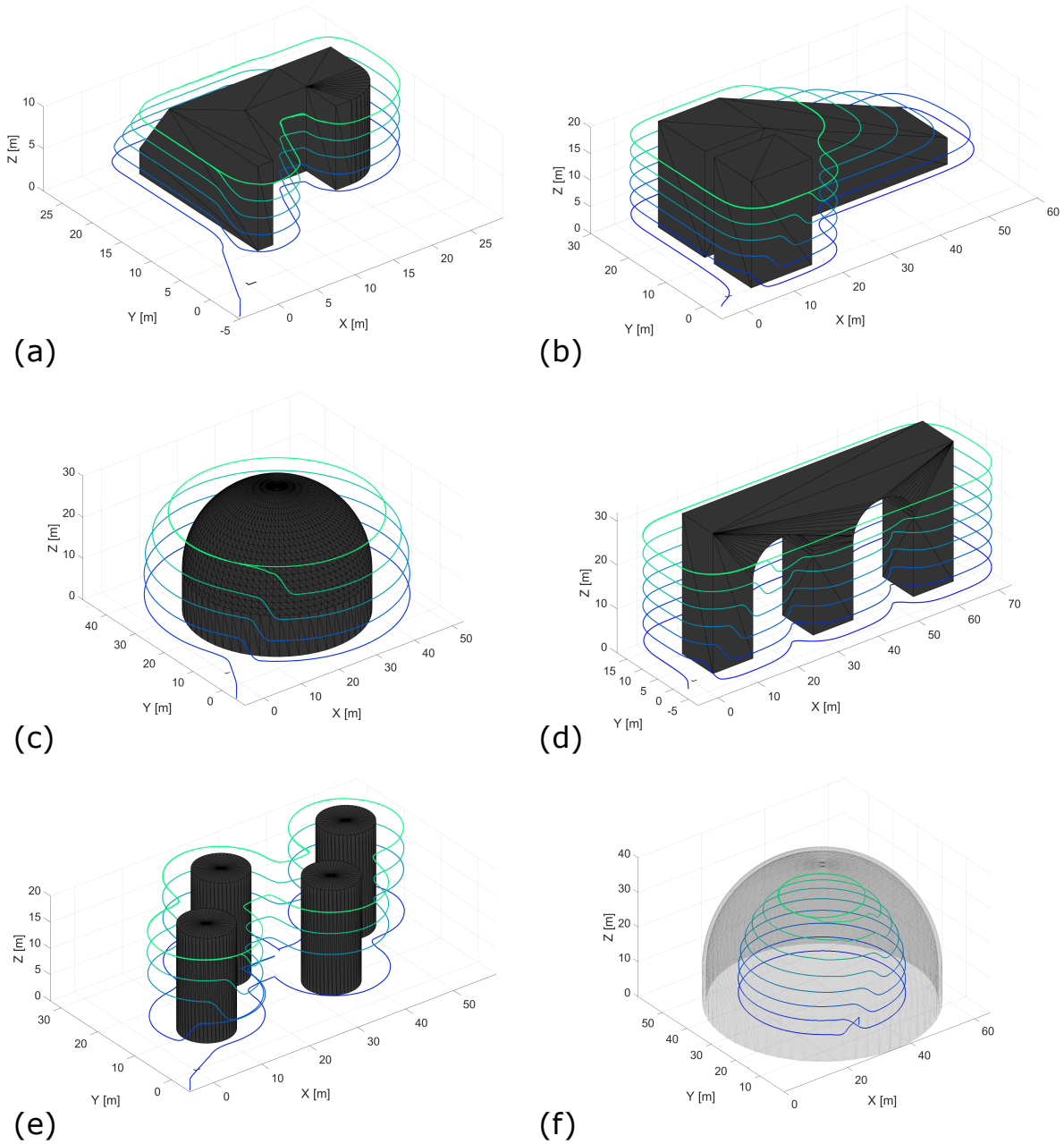


Figure 5.10: Executed trajectories of the UAV navigating with CI-APF method in the unknown environment with different target objects.

A particularly challenging scenario for testing the CI-APF algorithm involved an environment containing a model of a wind turbine, primarily due to its unique geometry. The model utilized for these experiments represented a generalized wind turbine configuration, featuring a pole height of 60m and blades extending 30m in an "inverted Y" configuration. Several specific constraints needed to be considered during the inspection mission. Given that the point at which the complete pass around the object is detected, and the flight altitude is changed to the next value, generally lies in close proximity (considering only x and y coordinates) to the point where the model is first detected, and the UAV starts traversing along the generated isoline, the approach vector had to be positioned either in front of or behind the wind turbine to avoid collisions with the blades while increasing flight altitude. This precaution was necessary due to the limited ability of the LiDAR sensor to detect obstacles directly above the UAV.

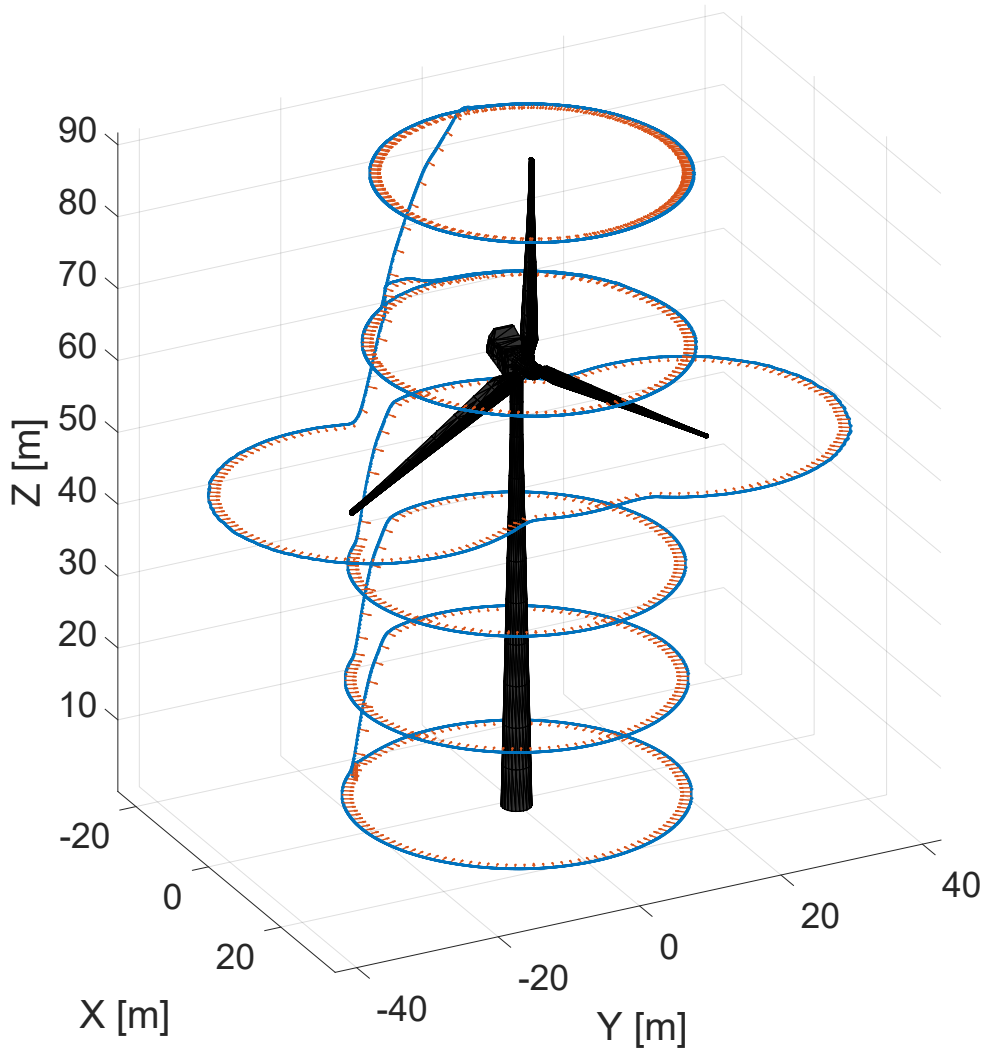


Figure 5.11: Simulation results of UAV navigating in environment with unknown windturbine model.

The second constraint involved maintaining a desired distance (r_d) from the object during

the inspection. This distance was set to $20m$ to prevent potential issues where a lesser value could result in the UAV becoming trapped in circumnavigation around one of the side blades. This scenario could occur when the UAV was navigating around the blade tip at a close distance, potentially failing to detect the pole or nacelle as the closest points and thus never reaching the starting point of the current pass to change altitude to the next level. Executed flight trajectory along with heading of the UAV around the wind turbine model is depicted in Fig.5.11.

Figure5.14presents a comparison between flight paths planned using methods requiring a priori knowledge of object layout based on Huygens's wave propagation principle (depicted in blue) and closed isolines of artificial potential field (illustrated in red), with the UAV's executed trajectory around an unknown object guided by the CI-APF algorithm (yellow line). In all three scenarios, the desired distance of the UAV from the object (r_d) is set to $5m$. The area discretization parameter (λ_a) in both scenarios utilizing the isolines of the potential field is $1/2\sqrt{2}m$, ensuring that the maximal distance between two neighboring waypoints is $0.5m$, equivalent to the path discretization parameter (λ_p) used in the case where the flight path was planned based on Huygens's wave propagation principle. The results demonstrate no significant difference between the paths planned using methods with known object layouts and minor deviations, particularly around sharp corners of the object, for the executed trajectory of the UAV generated with the real-time CI-APF planning algorithm.

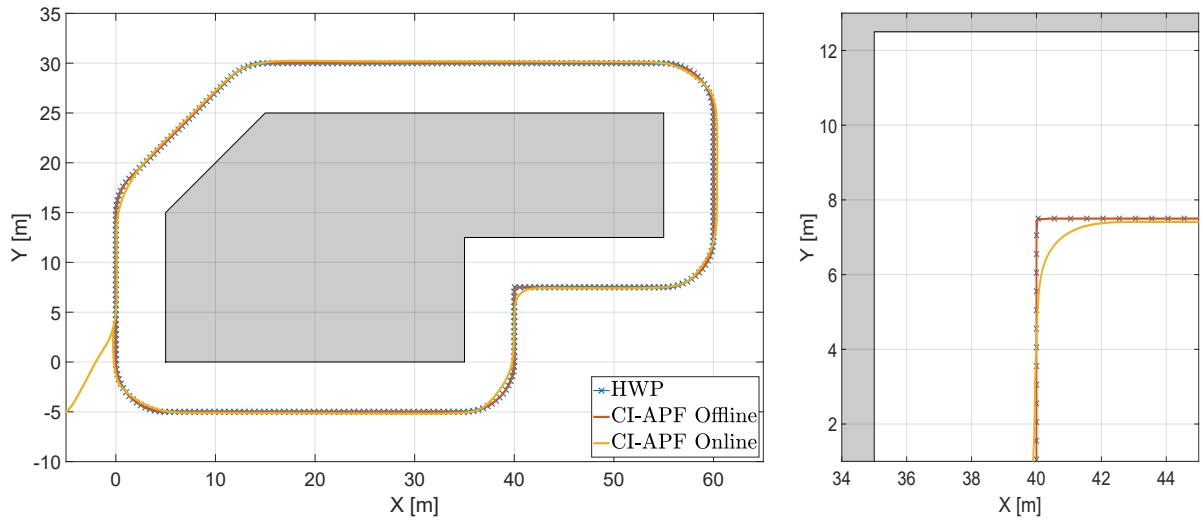
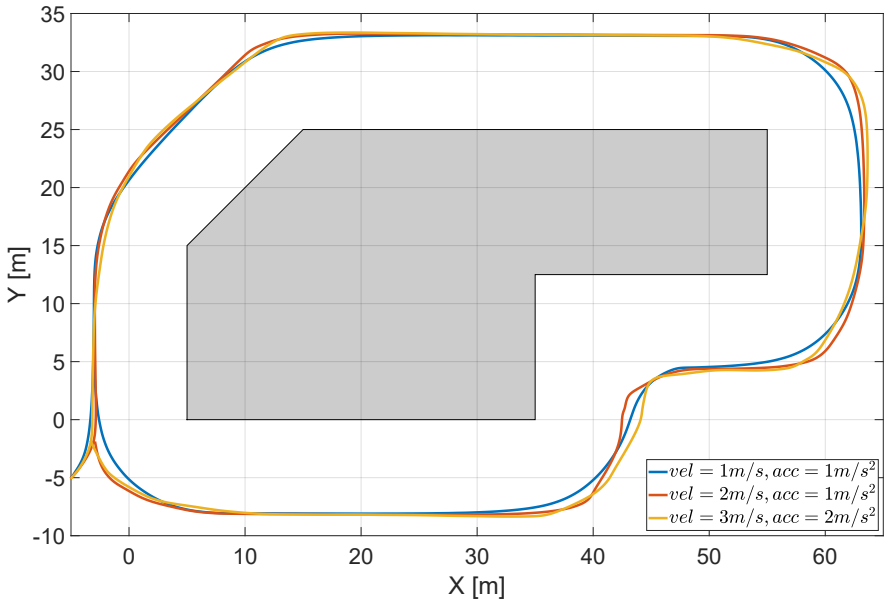


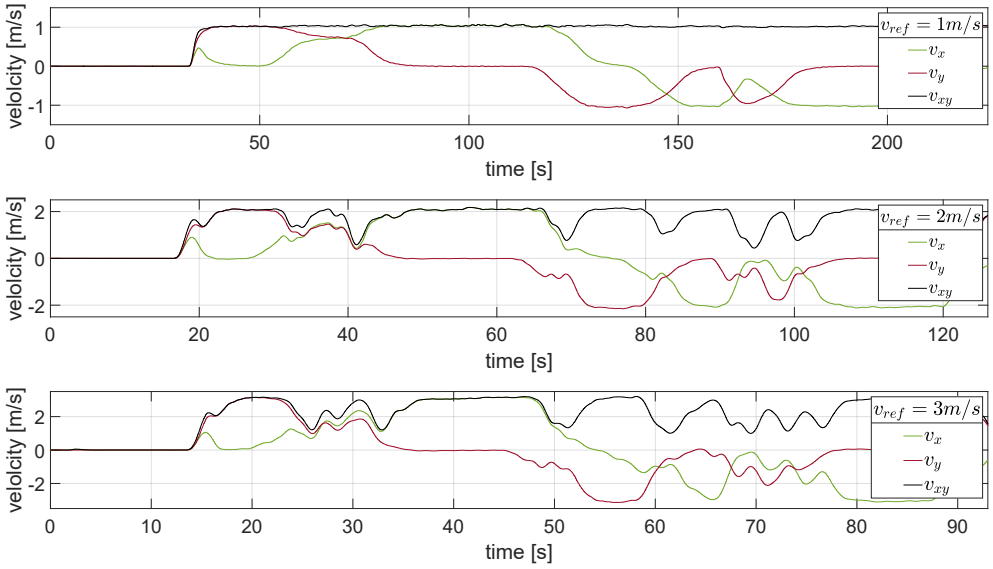
Figure 5.12: Comparison of the flight paths planned methods that require a priori knowlace of the object's layout based on Huygens's wave propagation principle (blue markers) and closed isolines of APF (red) with trajectory executed by UAV using real-time CI-APF algorithm for navigation around unknown structure (yellow).

The outcomes of experiments conducted with different velocity and acceleration constraints applied to the UAV are illustrated in Fig.5.13. The figure showcases both the trajectories executed in the horizontal plane and the temporal response of the UAV velocity along the X and Y axes, alongside the resultant velocity in the XY-plane. Analysis indicates that the UAV success-

fully attained a reference velocity, constrained by the imposed limitations, particularly while traversing along the straight segments of the object. Moreover, it effectively maintained the desired distance from the object within acceptable margins of error, even during flight maneuvers around sharp corners.



(a) Executed UAV trajectories with different reference velocities in XY-plane.



(b) Profiles of the UAV's velocity components in the X and Y directions, along with the magnitude of resultant velocity in the horizontal plane.

Figure 5.13: Simulation results depicting experiments conducted under three distinct scenarios, each featuring the same surveyed object but with varying reference velocities: $1m/s$, $2m/s$, and $3m/s$. The figure illustrates the impact of different reference velocities on the UAV's performance and trajectory outcomes.

Future research on the CI-APF algorithm should concentrate on modifying the algorithm for application in even more complex and challenging environments. This includes scenarios where detected features need to be separately clustered and treated as sources of multiple potential fields. Additionally, the algorithm should be adapted for situations where specific configurations of flight paths are required. A prime example of such a scenario is presented in [68], where each cable of the bridge should be treated as an individual inspection target, and the UAV needs to stop at specific points in space rather than continuously traversing through waypoints.

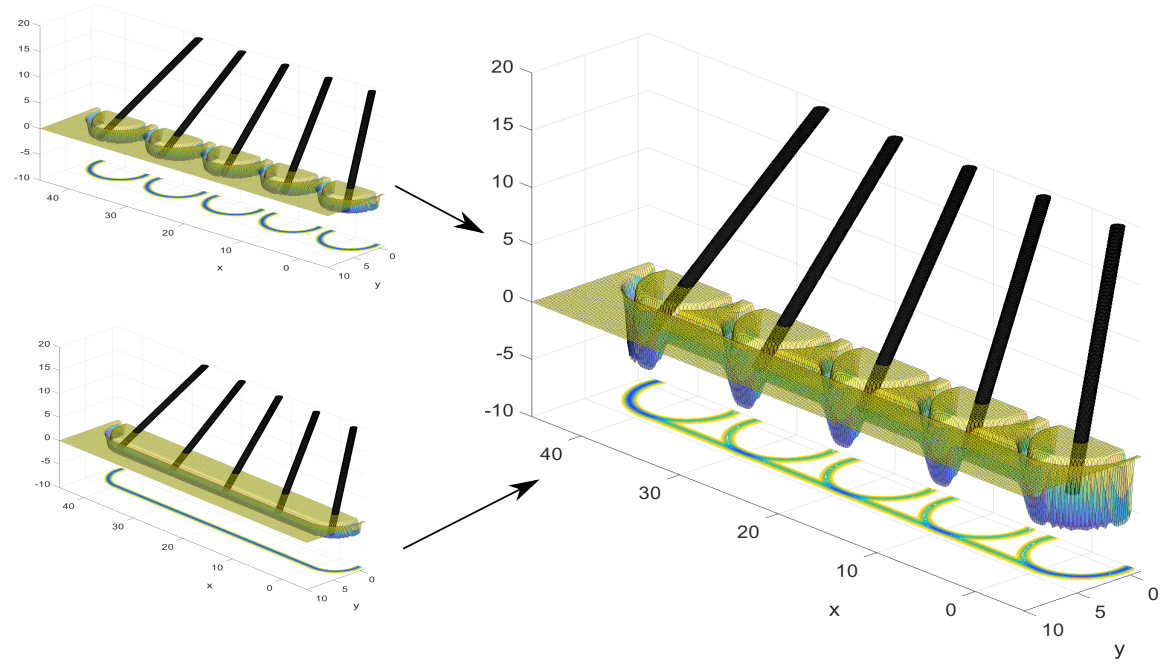


Figure 5.14: Initial concept for future modifications of the CI-APF algorithm, integrating multiple potential fields to generate a specific flight path as required in the case study presented in [68].

CHAPTER 6

Flight trajectory tracking in unknown environments based on APF

This chapter describes the development of the Augmented Artificial Potential Field (AAPF) algorithm, which aims to enhance the autonomous navigation capabilities of UAVs in unknown environments [84]. Inspection trajectories are often planned with consideration only for the structure of interest, neglecting potential obstacles along the flight path due to unreliable or completely unknown information about the surrounding environment. Therefore, it is essential for UAVs to have the capability to detect obstacles online and execute avoidance maneuvers while adhering to the initially planned trajectory in unobstructed space.

The AAPF algorithm builds upon the Modified Artificial Potential Field (MAPF) algorithm introduced in [85], which utilizes normal and rotational components of the potential field generated by detected obstacles for collision avoidance. In the AAPF algorithm, improvements are introduced by incorporating attractive forces generated by vertices of obstacles and the goal point on the initial trajectory, aiming to reduce unnecessary lengthy paths for obstacle avoidance. Furthermore, the AAPF algorithm seeks to identify a feasible waypoint on the planned path closest to the obstacle, where the UAV can safely return after circumventing the obstacle, by calculating repulsive potential field along the set of initial waypoints based on the sensor data from the UAV.

A comprehensive comparison between the MAPF and AAPF algorithms is conducted by analyzing simulation results across various scenarios and environment configurations. Additionally, the effects of all four forces implemented in the AAPF algorithm on the behavior of UAVs are examined in simulations. The feasibility of utilizing the AAPF algorithm in real-world scenarios is evaluated through experiments conducted in a flight arena.

6.1Augmented Artificial Potential Field Algorithm

The primary objective is to develop a robust autonomous tracking system for a UAVs operating in complex three-dimensional environments, utilizing sensors capable of generating extensive point cloud data, which is essential for effective obstacle avoidance. The autonomous navigation task is particularly challenging if the UAV operates in environments where prior knowledge is uncertain, potentially leading to encounters with obstacles that are unknown at the time of initial trajectory planning. A typical scenario illustrating this challenge is the execution of infrastructure inspection missions, where the structure being surveyed is known, but information regarding the surrounding environment is unreliable or completely absent. Consequently, initial flight trajectories are planned without considering the potential obstacles like trees or other structures in the vicinity which increases complexity to the navigation task. The main objective of the UAV is to follow the originally planned flight trajectory with minimal deviations while avoiding obstacles and reaching the final goal point. Flexibility is incorporated into the system, allowing deviations from the planned trajectory to circumvent previously unknown obstacles. The emphasis lies on minimizing deviations from the originally planned flight trajectory, enabling the UAV to navigate around significant obstacles and resume its trajectory at the nearest safe point determined by the algorithm. This approach ensures efficient trajectory tracking while accommodating dynamic environmental conditions encountered during flight.

6.1.1System overview

The system architecture integrates both global and local modules to enable efficient trajectory generation and execution. The global component is responsible for generating the initial trajectory by providing waypoints to the trajectory planner, while the local planner utilize the AAPF method to make trajectory corrections, ensuring obstacle avoidance. In addition to the waypoints, inputs for the trajectory planner include velocity and acceleration constraints, which are determined considering the physical limitations of the UAV and any constraints related to data acquisition.

During operation, the UAV executes the generated trajectory while continuously processing data from its onboard sensor system. Environmental information is represented as a point cloud generated by LIDAR, utilizing a wide Field of View (FOV) for comprehensive environmental modeling crucial for effective collision avoidance. The potential field algorithm calculates the total repulsive force F_r generated by the obstacles in the environment, which is described in detail in the next section. Constant K_{thr}^{UAV} regulates whether the algorithm follows the trajectory point or activates the collision avoidance adjustments on planned trajectory. The condition is described as follows:

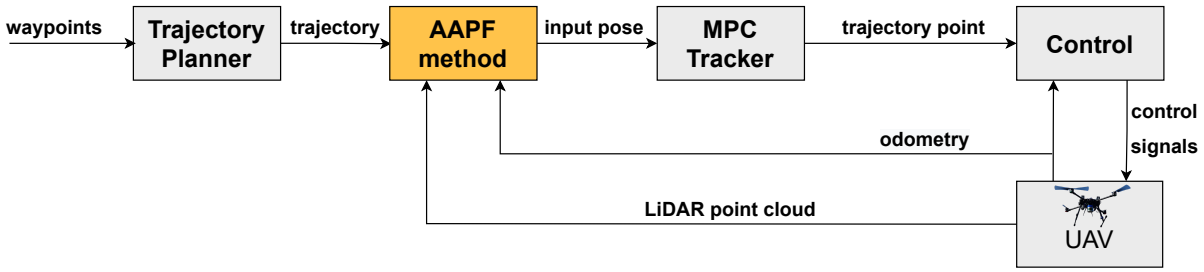


Figure 6.1: Overall schematic diagram of the trajectory planning and obstacle avoidance system. The trajectory generated by Trajectory Planner, the LiDAR point cloud, and the odometry data represent inputs to the AAPF method. The MPC tracker module generates a trajectory point to which the UAV navigates.

$$\text{Action} = \begin{cases} \text{Follow Trajectory} & \text{if } F_r < K_{thr}^{UAV}, \\ \text{Do APF Method} & \text{if } F_r \geq K_{thr}^{UAV}. \end{cases}$$

Figure6.1provides an overview of the proposed system architecture. When the AAPF method is not active, the global trajectory points are passed directly to the Model Predictive Control (MPC) tracker module. On the other hand, if the AAPF method is active, the modified path is generated and forwarded to the MPC tracker module. A standard PID cascade is used to control the UAV, with the inner loop controlling the velocity and the outer loop controlling the position. In this setup, the reference for the controller is a trajectory point.

The UAV is represented with a state vector $\mathbf{x} = \begin{bmatrix} \mathbf{q}^T & \psi \end{bmatrix}^T \in \mathbb{R}^4$ that consists of the position $\mathbf{q} = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$ and the yaw rotation angle around z axis $\psi \in [-\pi, \pi)$. Furthermore, the algorithm assumes a maximum linear velocity $\mathbf{v}_{max} \in \mathbb{R}^3$, a maximum angular velocity around z axis $\dot{\psi}_{max}$, a maximum linear acceleration $\mathbf{a}_{max} \in \mathbb{R}^3$ and maximum angular acceleration around z axis $\ddot{\psi}_{max}$. The algorithm relies on a maximum range of the sensor $R_{max} \in \mathbb{R}$ with horizontal and vertical FOV in range, α_h, $\alpha_v \in (0°, 360°]$, respectively. This allows the collision avoidance algorithm to work with point-cloud-producing sensors with various FOV, such as cameras with limited FOV and LiDARs with limited α_v.

Local path planning integrates the potential field algorithm to correct the planned global trajectory and generate a safe and collision-free path. The conventional artificial potential field method utilized in robot navigation is composed of two types of potential fields: attractive potential field and repulsive potential field [32]. Attractive potential field denoted as $U_a(\cdot)$ is typically formed by goal location $\mathbf{q}_g$, generating a force that pulls the UAV towards the desired destination. In contrast, repulsive potential field represented by $U_r(\cdot)$, originates from obstacle positions $\mathbf{q}_o$, creating a repulsive force that repels the UAV to evade potential collisions within a specified range d_0 around obstacles. The combination of these two potential fields results in a total potential field $U_t(\cdot)$ that guides the UAV toward the goal point while steering away of

obstacles:

$$U_t(\mathbf{q}) = U_a(\mathbf{q}) + U_r(\mathbf{q}). \tag{6.1}$$

Similarly to the conventional approach, AAPF algorithm consists of repulsive and attractive potential fields used for navigating the UAV in the environment. The repulsive field is expanded to include both normal and rotational components of the repulsive force, enabling complex deviations from the initially planned path and facilitating the resolution of local minima problems. The attractive potential field is generated from a goal point and obstacle vertex to minimize overshooting during obstacle avoidance maneuvers. A final aspect of the extended algorithm is the identification of waypoints along the originally planned trajectory closest to the obstacle, where the UAV can safely return to after executing a collision avoidance maneuver. By integrating these augmentations, the algorithm ensures that the UAV follows the planned trajectory while dynamically adjusting to avoid collisions with obstacles, thereby ensuring safe and efficient navigation through complex environments with minimal deviations from the initial flight path. Detailed formulations of the repulsive and attractive potential fields are provided in Sections6.1.3and6.1.4, respectively, while the selection method of the return point is outlined in Section6.1.5.

The result of the AAPF algorithm is a new pose point that serves as input for the MPC tracker module. This module generates a trajectory point used as reference input for the UAV controller. A detailed description of the implemented MPC tracker module is provided in Section3.1.1.

6.1.2Obstacle detection

The UAV utilizes a LIDAR sensor to perceive its surroundings during the flight mission, capturing environmental data in the form of a point cloud. Given the unpredictable geometrical configurations of real-world obstacles, each point within the resultant point cloud is interpreted as an origin of repulsive potential field. This strategy ensures the UAV's ability to navigate and avoid collisions with entire obstacles within its path. However, a drawback of this approach lies in the substantial computational demands imposed on the onboard computer, particularly in cluttered environments where the point cloud density is high. To mitigate this issue, the raw point cloud undergoes filtering and downsampling processes. Upon receiving new LIDAR scan $\mathbf{P}_S$ each point in the point cloud is transformed from the sensor frame of reference S to the UAV frame R using the static transform $\mathbf{T}_R^S$.

$$\mathbf{P}_R = \mathbf{T}_R^S \mathbf{P}_S \tag{6.2}$$

Furthermore, point cloud $\mathbf{P}_R$ is transformed to the world frame W using the transform matrix $\mathbf{T}_W^R$ derived from the UAV state vector $\mathbf{x}$.

$$\mathbf{P}_W = \mathbf{T}_W^R \mathbf{P}_R \tag{6.3}$$

Initially, the *CropBox* filter from the PCL library [81] is utilized. This filter selectively retains data within a specified box region, with the box centered on the current UAV position $\mathbf{q}$ and its axes aligned with the roll, pitch, and yaw angles of the UAV. The dimensions of the crop box l are configured to ensure that only points in proximity to the UAV, potentially requiring collision avoidance maneuvers, are perceived as part of the obstacles. All the individual points $\mathbf{p}_i$ of the cropped point cloud $\mathbf{P}_W^c$ satisfy condition:

$$\|\mathbf{p}_i - \mathbf{q}\| \leq l, \quad i \in \{1, \ldots, m\}, \tag{6.4}$$

where m is total number of points in $\mathbf{P}_W^c$.

Additionally, to further reduce the processing demands of the algorithm, the point cloud $\mathbf{P}_W^c$ is downsampled using the *VoxelGrid* filter from the PCL library. This process involves creating a three-dimensional voxel grid over the input point cloud data. Within each voxel, all points are approximated by their centroid, providing the accurate representation of the underlying surface. This downsampling significantly reduces the number of points in the resulting point cloud $\mathbf{P}_W^f$ without compromising the UAV's ability to detect obstacles in its environment.

While each point within the filtered point cloud model $\mathbf{P}_W^f$, derived from individual LiDAR scans, is considered a source of the repulsive potential field, clusters of points in the accumulated point cloud $\mathbf{P}_W^a$ are perceived as single obstacles within the algorithm. The accumulated point cloud $\mathbf{P}_W^a$ is generated by adding filtered point cloud $\mathbf{P}_W^f$ processed from the new LiDAR scan to the existing accumulated point cloud:

$$\mathbf{P}_W^a(k) = \mathbf{P}_W^a(k-1) + \mathbf{P}_W^f(k), \quad \mathbf{P}_W^a(0) = \emptyset \tag{6.5}$$

Applying the *VoxelGrid* filter to the accumulated point cloud $\mathbf{P}_W^a$ yields the resultant point cloud $\mathbf{P}_W^{af}$, effectively alleviating the potential issue of excessive point density arising from continual integration of data from the LiDAR sensor over time. This process ensures the preservation of essential environmental features while managing the accumulation of data. The filtered accumulated point cloud $\mathbf{P}_W^{af}$ serves as a comprehensive representation of the environmental map and it is utilized to accurately determine the position of centroids for large obstacles. A clustering method is used to divide a disorganized point cloud model of the environment into smaller parts that represent individual obstacles. The simplest way is to use the Euclidean clustering algorithm, which is a greedy growing region algorithm based on the nearest neighbor.

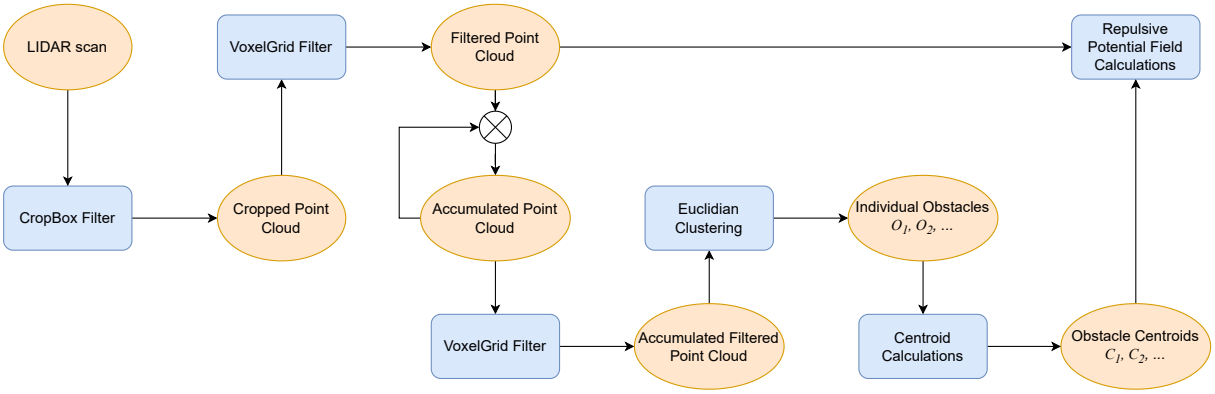


Figure 6.2: Point cloud processing diagram

The cluster affinity is based on the distance to each point of a cluster and is defined by the parameter called cluster tolerance c_t. If the Euclidean distance between the point $\mathbf{p}_l$ and any point $\mathbf{p}_k^i$ that is a member of the cluster $\mathcal{S}_i$ is below a certain threshold value c_t, then the point $\mathbf{p}_l$ is included in cluster $\mathcal{S}_i$. If that is not the case and the distance exceeds this threshold, point $\mathbf{p}_l$ is assigned to a different cluster $\mathcal{S}_j$. The choice of the c_t parameter is based on the dimensions of the UAV augmented by a safety margin. This ensures that the UAV can navigate between any two points of separate clusters. If the c_t value is set too low, an obstacle that should be considered as a single entity may be perceived as multiple clusters. For instance, a narrow gap in a wall that is too small for the UAV to pass through might be observed as two separate obstacles. On the other hand, if the chosen value is excessively high, multiple obstacles may be treated as a single cluster, thereby neglecting potential pathways between them. The 3D centroid $\mathbf{C}_i$ is computed for each cluster of the accumulated point cloud using the function implemented in PCL library.

6.1.3Repulsive Potential Field

Based on the principles outlined in the [32], the repulsive potential field $U_r(\cdot)$ is designed to establish a barrier at each point along the surface of an obstacle, gradually diminishing in magnitude as the distance from the obstacle increases. To satisfy these criteria, $U_r(\cdot)$ is formulated as a non-negative continuous and differentiable function, with values approaching infinity in close proximity to obstacle. Considering these requirements, the repulsive potential field, utilizing the Euclidean distance between the UAV and the obstacle, is defined as:

$$U_r(\mathbf{q}) = \begin{cases} \frac{1}{2}k_{rn}\left(\frac{1}{\|\mathbf{q}_o-\mathbf{q}\|}-\frac{1}{d_0}\right)^2 & \text{if } \|\mathbf{q}_o-\mathbf{q}\| \leq d_o, \\ 0 & \text{if } \|\mathbf{q}_o-\mathbf{q}\| > d_o, \end{cases} \tag{6.6}$$

where k_{rn} is repulsive gain coefficient, $\|\mathbf{q}_o-\mathbf{q}\|$ denotes the relative distance between the position of the UAV $\mathbf{q}$ and the obstacle $\mathbf{q}_o$ and d_o represents the boundary distance of the po-

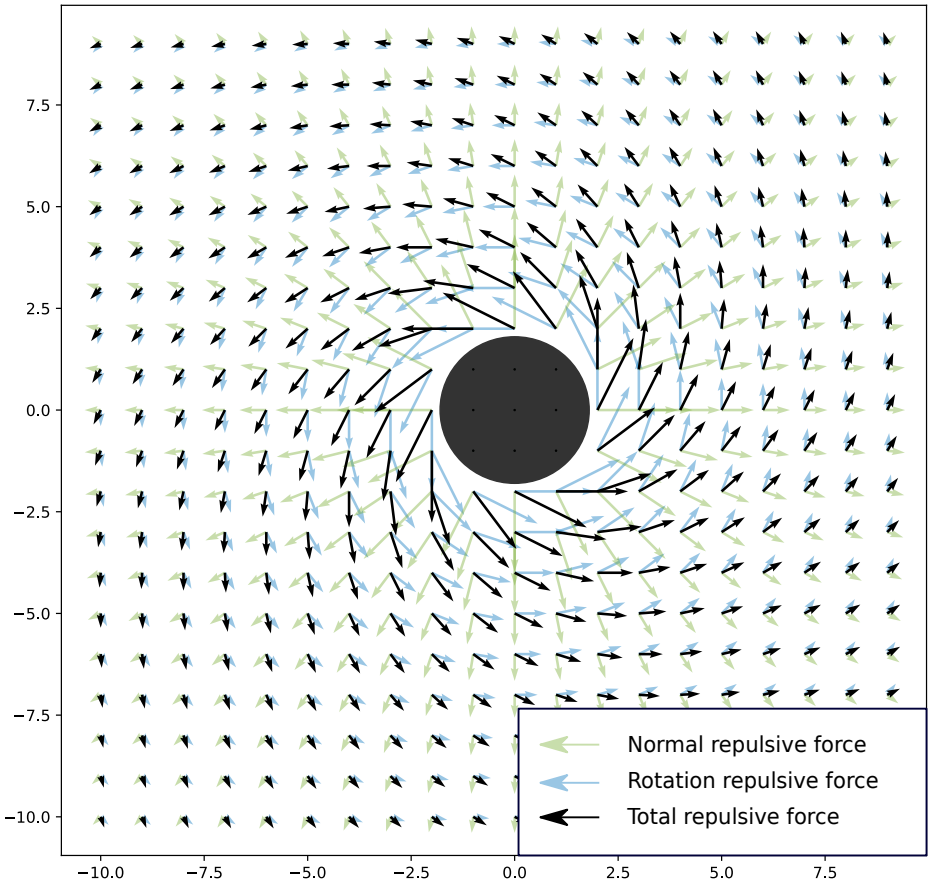


Figure 6.3: Repulsive force generated by the obstacle. The total repulsive force consists of the normal component vector $\mathbf{F}_{rn}$ and the rotational component vector $\mathbf{F}_{rr}$, where the rotational potential field is generated in the counterclockwise direction.

tential field's influence. The corresponding *normal repulsive potential field force*, is derived by computing the negative gradient of the repulsive potential function as follows:

$$\mathbf{F}_{rn}(\mathbf{q}) = -\nabla U_r(\mathbf{q}) = \begin{cases} k_{rn}\left(\frac{1}{\|\mathbf{q}_o-\mathbf{q}\|} - \frac{1}{d_0}\right)\frac{1}{\|\mathbf{q}_o-\mathbf{q}\|^3}(\mathbf{q}-\mathbf{q}_o) & \text{if } \|\mathbf{q}_o-\mathbf{q}\| \le d_o, \\ 0 & \text{if } \|\mathbf{q}_o-\mathbf{q}\| > d_o, \end{cases} \tag{6.7}$$

where k_{rn} is the gain of the normal force vector. To calculate the total normal repulsive field force $\mathbf{F}_{rn_i}$ for the i-th obstacle, considering each point $\mathbf{p}_k^i$ within the cluster $\mathcal{S}_i$ as a source of the potential field, the following equation is utilized:

$$\mathbf{F}_{rn_i}(\mathbf{q}) = \frac{\sum_{j=1}^{n} \mathbf{F}_{rn}^j(\mathbf{q})}{n} \tag{6.8}$$

Here, $\mathbf{F}_{rn}^j(\mathbf{q})$ is calculated using the Equation6.7, with $\mathbf{q}_o$ substituted by $\mathbf{p}_k^i$ for all k in the range $\{1,\ldots,n\}$, where n represents the total number of points in cluster $\mathcal{S}_i$.

The conventional artificial potential field method, while effective in path generation, encounters challenges such as oscillations around certain types of obstacles, difficulty traversing narrow passages, and susceptibility to getting stuck in local minima, resulting in incomplete paths [38], [85]. The local minima problem arises when the total force exerted by the potential field approaches zero or becomes sufficiently small, resulting in no significant displacement of the UAV before reaching the goal position. This occurs when the magnitudes of the attractive and repulsive potential field forces are equal but opposite in direction, effectively canceling each other out. Conventional APF methods typically rely on local information, such as obstacle distances and directions, for robot navigation or path planning. However, this local information may lead the robot towards a local minimum in the potential field, resulting in suboptimal or inefficient paths. Additionally, conventional APF methods are sensitive to initial conditions of the system and the starting position of the robot. If the robot starts near a local minimum, it is more likely to converge to that minimum and struggle to escape.

To address these challenges and mitigate local minima problems, an extension of the repulsive force, denoted as $\mathbf{F}_r$, is implemented to include both normal and rotational components. This modified repulsive force enhances UAV navigation in challenging environments, facilitating collision avoidance and escape from local minima. The modified repulsive force is defined as follows:

$$\mathbf{F}_r(\mathbf{q}) = \mathbf{F}_{rn}(\mathbf{q}) + \mathbf{F}_{rr}(\mathbf{q}), \tag{6.9}$$

where $\mathbf{F}_{rn}(\mathbf{q})$ represents the normal component of the repulsive force defined with equation 6.7, while $\mathbf{F}_{rr}(\mathbf{q})$ corresponds to the *rotational repulsive potential field force* illustrated in Fig. 6.3. The rotational component of the repulsive force is calculated only in the X-Y plane, with position vectors $\mathbf{q} = \begin{bmatrix} \mathbf{r} & z \end{bmatrix}^T \in \mathbb{R}^3$ for the UAV and $\mathbf{q}_o = \begin{bmatrix} \mathbf{r}_o & z_o \end{bmatrix}^T \in \mathbb{R}^3$ for the obstacle. It is designed such that the curl of the potential field $U_{rr}(\mathbf{q})$ equals zero:

$$\nabla \times U_{rr}(\mathbf{q}) = 0 \tag{6.10}$$

The rotational repulsive force $\mathbf{F}_{rr}(\mathbf{q})$ is defined as:

$$\mathbf{F}_{rr}(\mathbf{q}) = \begin{cases} k_{rr}\left(\frac{1}{\|\mathbf{q}_o - \mathbf{q}\|} - \frac{1}{d_0}\right) \cdot \frac{1}{\|\mathbf{q}_o - \mathbf{q}\|^3} R(\mathbf{r} - \mathbf{r}_o) & \text{if } \|\mathbf{q}_o - \mathbf{q}\| \leq d_o, \\ 0 & \text{if } \|\mathbf{q}_o - \mathbf{q}\| > d_0, \end{cases} \tag{6.11}$$

where k_{rr} represents the gain of the rotational force vector and $\mathbf{R}$ is the rotation matrix. The choice of the matrix $\mathbf{R}$ depends on the direction of the generated rotating repulsive field, which can be either clockwise or counterclockwise. The angle ϕ is defined as the angle of the UAV on the trajectory, while ρ indicates the angle of the vector from the position of the UAV to

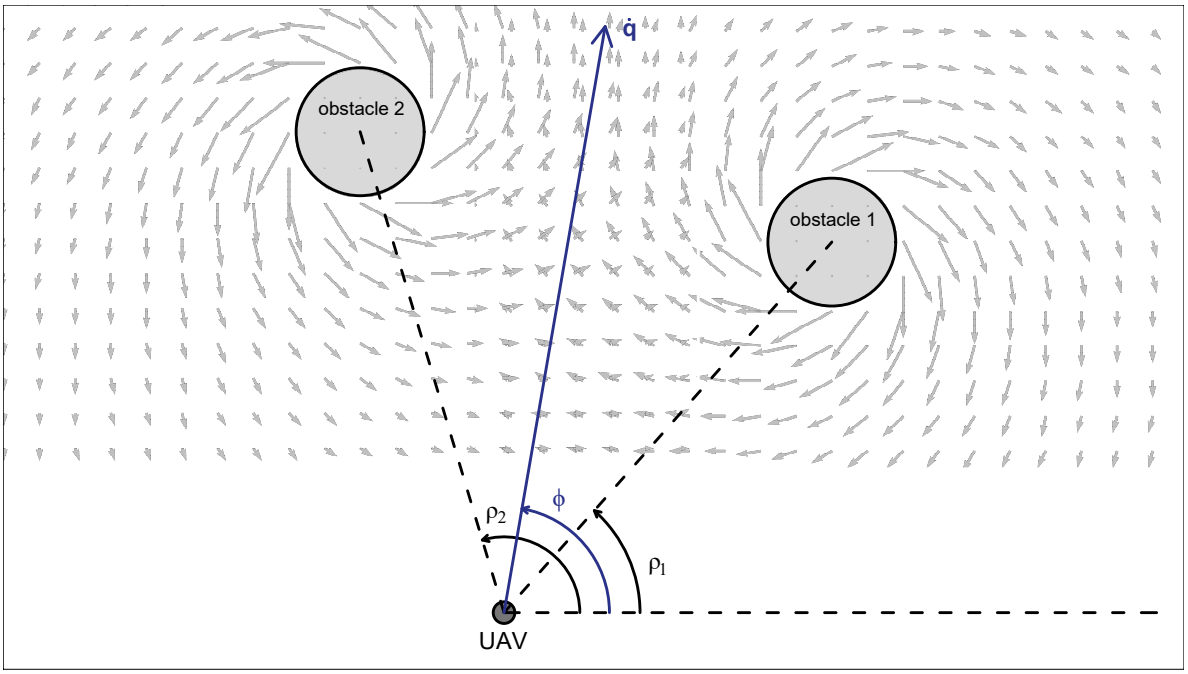


Figure 6.4: Direction of the rotational potential field generated by the obstacle i in the environment is defined by the difference between the angle of the trajectory ϕ and the angle ρ_i, which is an angle of a vector from the UAV to the centroid of the obstacle i.

the centroid $\mathbf{C}_i$ of the obstacle in the environment (Fig.6.4). The angle θ is defined as the difference between these two angles:

$$\theta = \phi - \rho \tag{6.12}$$

The sign of θ determines the direction of the potential field around the obstacle. The matrix $\mathbf{R}$ is defined with respect to θ as follows:

$$\mathbf{R} = \begin{cases} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \text{if } \theta \geq 0, \\ \\ \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} & \text{if } \theta < 0. \end{cases} \tag{6.13}$$

Applying the same principle as that used for the normal repulsive force, where each point $\mathbf{p}_k^i$ within the cluster $\mathcal{S}_i$, containing a total of n points, is regarded as a source of the repulsive potential field, the total rotational repulsive field force $\mathbf{F}_{rr_i}(\mathbf{q})$ is determined by the following equation:

$$\mathbf{F}_{rr_i}(\mathbf{q}) = \frac{\sum_{j=1}^{n} \mathbf{F}_{rr}^{j}(\mathbf{q})}{n} \tag{6.14}$$

Since the repulsive potential field is produced for each obstacle, the total potential field for m obstacles is expressed as the sum of the repulsive forces generated by encountered obstacles:

$$\mathbf{F}_{rt}(\mathbf{q}) = \sum_{1}^{m} (\mathbf{F}_{rn_i}(\mathbf{q}) + \mathbf{F}_{rr_i}(\mathbf{q})). \tag{6.15}$$

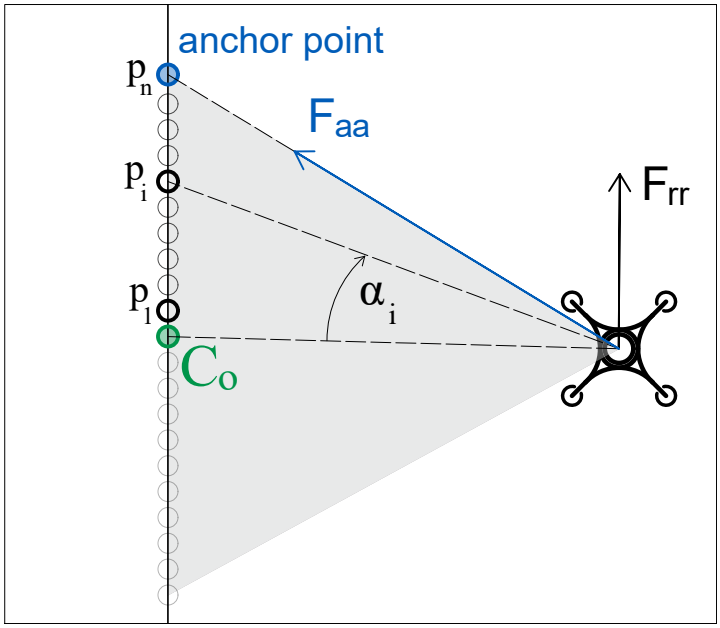
This combination of forces ensures that the UAV consistently moves away from obstacles and aligns with the intended trajectory, preventing it from becoming trapped in oscillatory motion around local minima created by normal forces or orbiting the obstacle at the same distance induced by rotational forces.

6.1.4Attractive Potential Field

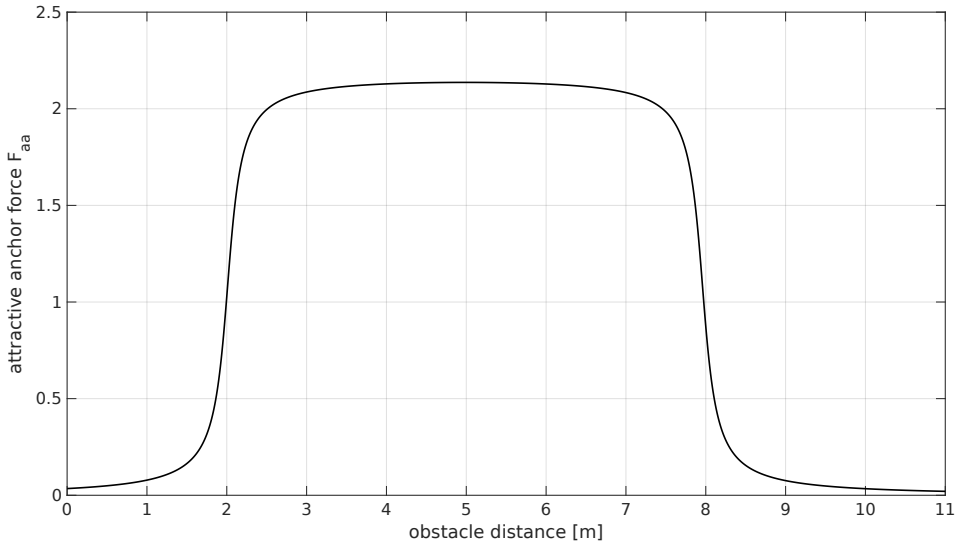
As demonstrated in [85], the Modified Artificial Potential Field algorithm (MAPF), which integrates both normal and rotational components of the force for collision avoidance, exhibits efficacy in navigating complex environments with unknown obstacles. However, this algorithm shows significant deficiencies in optimal trajectory tracking, often resulting in large deviations from the planned path. Consequently, the return point for the UAV can be substantially distant distant from the obstacle compared to the point at which the UAV initiates the collision avoidance maneuver, thereby disregarding segments of the planned trajectory that could be safely traversed. With the attractive forces set to zero, the repulsive forces drive the UAV to the edge of the field before it initiates the return to the planned trajectory. To mitigate the excessive deviations incurred during obstacle avoidance, two attractive forces based on the detected obstacle points and the goal point were implemented in Augmented Artificial Potential Field algorithm.

The first attractive force, named the *attractive anchor force* $\mathbf{F}_{aa}$, draws inspiration from the mechanical dynamics of a mass-spring system, where the spring imposes a maximum deviation of an object from the anchor point to which it is tethered in the environment. Within the AAPF algorithm framework, one vertex of each detected obstacle serves as an anchor point (Fig.6.5.a). The anchor point is being separately computed for every obstacle represented by a cluster of detected LiDAR points. To determine the anchor point, a specific portion of the point cloud cluster is selected based on the rotational potential field generated around the obstacle. Within this selected portion, the angle α_i is computed for each LiDAR point $\mathbf{p}_i$, representing the angle between the vector from the UAV to the centroid of the obstacle and the vector from the UAV to the point $\mathbf{p}_i$. The anchor point $\mathbf{q}_a$ is then identified as the point with the maximum angle α_i.

Since $\mathbf{F}_{aa}$ attracts the UAV towards the obstacle, a bell-shaped function is selected to generate the force (Fig.6.5.b). This choice is made to ensure that the force achieves maximum



(a) Set of detected LiDAR points $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ are selected as candidates for the *anchor point* $\mathbf{q}_a$ based on the direction of the rotational repulsive force and position of the obstacle centroid C_o. Angle a_i is calculated for each point $\mathbf{p}_i$ and the *anchor point* $\mathbf{q}_a$ is then chosen from the set $\mathcal{P}$ such that it has the highest value of angle.



(b) Anchor attractive force generated with a bell-shaped function and parameters $k_{aa} = 0.7$, $b_1 = b_2 = 7.5$, $k_1 = 4.8$, $k_2 = 19.0$. The center of the function is at the $5m$ distance from the obstacle.

Figure 6.5: Selection process of the *anchor point* (a) and example of the bell function used to calculate *attractive anchor force* $\mathbf{F}_{aa}$ (b).

magnitude at the preferred distance from the obstacle for maneuvering around it, while minimizing the magnitude when the UAV is either too close or too far from the obstacle. The mathematical formulation for generating the anchor attractive force is expressed as:

$$\mathbf{F}_{aa}(\mathbf{q}) = k_{aa}(\arctan(b_1\|\mathbf{q}_a - \mathbf{q}\| - k_1\pi) - \arctan(b_2\|\mathbf{q}_a - \mathbf{q}\| - k_2\pi))(\mathbf{q}_a - \mathbf{q}). \quad (6.16)$$

where parameter k_{aa} determines the magnitude of the force $\mathbf{F}_{aa}$, and b_1, k_1 along with b_2, k_2 define the rising and falling edges of the force function, respectively.

The *attractive goal force* $\mathbf{F}ag$ represents the second attractive force component operational within the AAPF algorithm. Its direction is oriented towards the next waypoint along the predefined trajectory, effectively serving as the current goal point analogous to conventional APF algorithms. The magnitude of the $\mathbf{F}_{ag}$ force depends on the distance between the UAV and the current goal point. The main difference between conventional APF attractive force and $\mathbf{F}_{ag}$ is that $\mathbf{F}_{ag}$ is exclusively active in proximity to obstacles, specifically when the repulsive force $\mathbf{F}_r$ exceeds a predefined threshold K_{thr}^{UAV}. The attractive goal potential field force is defined as follows:

$$\mathbf{F}_{ag}(\mathbf{q}) = (k_{ag}\|\mathbf{q}_g - \mathbf{q}\| + c_{ag})(\mathbf{q}_g - \mathbf{q}), \quad (6.17)$$

where k_{ag} is the magnitude gain, c_{ag} is a constant and $\|\mathbf{q}_g - \mathbf{q}\|$ represents the relative distance between the position of the UAV $\mathbf{q}$ and the current goal point $\mathbf{q}_g$.

All the potential field forces acting on the UAV in the vicinity of the obstacle are shown in Fig.6.6. Normal and rotational repulsive forces are defined with polynomial functions, which are dependent on the distance of the UAV from the obstacle (Eq.6.7and6.11). Additionally, the anchor attractive force is a bell-shaped function (Eq.6.16), while the attractive goal force is a linear function of distance between the UAV and current goal function (Eq.6.17). Detection of local minima is possible when the velocity of the UAV falls below a predetermined threshold, the sum of all forces approaches zero and the UAV is not close enough to a current goal position. By periodically checking these three conditions and confirming their validity, rotational repulsive force can progressively increase its value at each step, leading to exponential growth of force $\mathbf{F}_{rr}$. This ensures that the rotational repulsive force will surpass all other forces, leading to the displacement of the UAV out of the local minima while maintaining the safe distance from the obstacle.

6.1.5Return to Initially Planned Path

The Modified Artificial Potential Field algorithm (MAPF), as outlined in [85], implemented a simple mechanism to realign the UAV with its initially planned trajectory following obstacle avoidance maneuvers. Upon activation of the MAPF method for collision avoidance at time t_k, the algorithm monitored the duration required for deactivation, denoted as t_o, which signified successful obstacle avoidance. Upon reaching a point where the total repulsive force acting on

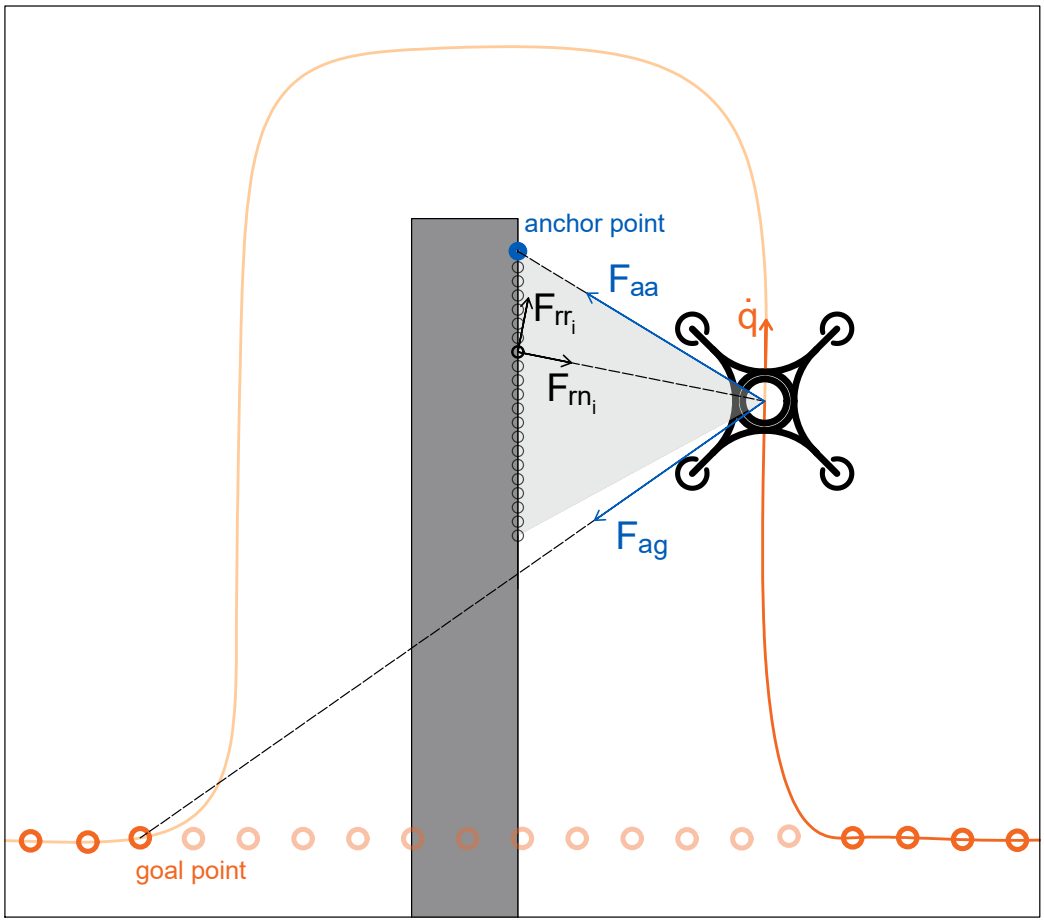


Figure 6.6: When the UAV is in the vicinity of obstacles the AAPF algorithm generates 4 forces that navigate the UAV around the obstacles and back to the planned path. Two repulsive forces are the normal repulsive force F_{rn}, which pushes the UAV away from the obstacle, and the rotational repulsive force F_{rr}, which guides the UAV around the obstacle. Two attractive forces are goal attractive force F_{ag}, which attracts the UAV to the next waypoint on the originally planned path, and the anchor attractive force F_{aa}, which drives the UAV to the vertex of the currently detected point cloud of the obstacle. Planned waypoints are shown with orange markers (solid markers are feasible, transparent ones are not) and the anchor point $\mathbf{q}_a$ is marked with a blue circle.

the UAV fell below a predefined threshold K_{thr}^{UAV}, indicating successful obstacle avoidance, the UAV started to approach back to its original trajectory. Specifically, it resumed following the trajectory point corresponding to state $\mathbf{x}(t_k + t_o)$.

However, while this strategy proved effective for scenarios involving minor deviations from the planned path, it exhibited limitations in cases where the UAV deviates from its original trajectory for an extended period, such as when navigating around large obstacles. This limitations posed challenges for the UAV to effectively realign with and resume following the originally planned trajectory.

To enhance trajectory following and maintain the UAV on the planned path for as long as possible, an online waypoint feasibility assessment mechanism based on the artificial potential

field generated by obstacles was integrated into the AAPF algorithm. This mechanism identifies a waypoint as feasible if it can be safely reached without encountering obstacles in close proximity. The determination of the currently feasible waypoint, or the current goal point, uses the same principle as method calculating the repulsive force vectors acting on the UAV. Specifically, a force acting on the current goal point is computed based on UAV sensor data. If the magnitude of this force is below a predefined threshold K_{thr}^{goal}, the current goal point is marked as feasible. Otherwise, it implies that the current goal point is too close to an obstacle, prompting the selection of the npext waypoint along the planned trajectory as the new current goal point. This iterative process enables the UAV to navigate back to the closest feasible waypoint that can be reached safely, ensuring the continuation of its optimal and originally planned trajectory. Potential field U_{gp} acting on the current goal point is defined by:

$$U_{gp}(\mathbf{q}_g) = \begin{cases} \frac{1}{2}k_{gp}(\frac{1}{\|\mathbf{q}_o-\mathbf{q}_g\|} - \frac{1}{d_0})^2 & \text{if } \|\mathbf{q}_o-\mathbf{q}_g\| \leq d_o, \\ 0 & \text{if } \|\mathbf{q}_o-\mathbf{q}_g\| > d_o. \end{cases} \tag{6.18}$$

where $\mathbf{q}_g$ is position of the current goal point in the world reference frame W, parameter k_{gp} defines the magnitude gain and $\|\mathbf{q}_o-\mathbf{q}_g\|$ is Euclidian distance between the current goal point and obstacle detected by the UAV. The corresponding *goal point potential field force* $\mathbf{F}_{gp}$ is derived by computing the negative gradient of the potential function U_{gp} as follows:

$$\mathbf{F}_{gp}(\mathbf{q}_g) = -\nabla U_{gp}(\mathbf{q}_g) = \begin{cases} k_{gp}(\frac{1}{\|\mathbf{q}_o-\mathbf{q}_g\|} - \frac{1}{d_0}) \cdot \frac{1}{\|\mathbf{q}_o-\mathbf{q}_g\|^3}(\mathbf{q}_g-\mathbf{q}_o) & \text{if } \|\mathbf{q}_o-\mathbf{q}_g\| \leq d_o, \\ 0 & \text{if } \|\mathbf{q}_o-\mathbf{q}_g\| > d_o, \end{cases} \tag{6.19}$$

The goal point potential field force is recalculated for each new sensor measurement acquired by the UAV, ensuring continual assessment of waypoint feasibility during flight.

6.2Simulation results

Simulation-based evaluation of the AAPF algorithm is performed in the Gazebo environment using Robot Operating System (ROS) and a model of the *Kopterworx* quadcopter. The quadcopter is equipped with a Velodyne VLP-16 LiDAR sensor to detect obstacles in the environment. All simulations were executed on an Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz $\times$ 12.

The AAPF algorithm was used to generate a collision-free path for the UAV, with the simulation results illustrated in Fig.6.7. The planned trajectory consists of a sequence of goal points for the UAV to follow, denoted by blue markers. These waypoints are provided as input to the MPC Tracker when the UAV operates in an unobstructed environment. However, when

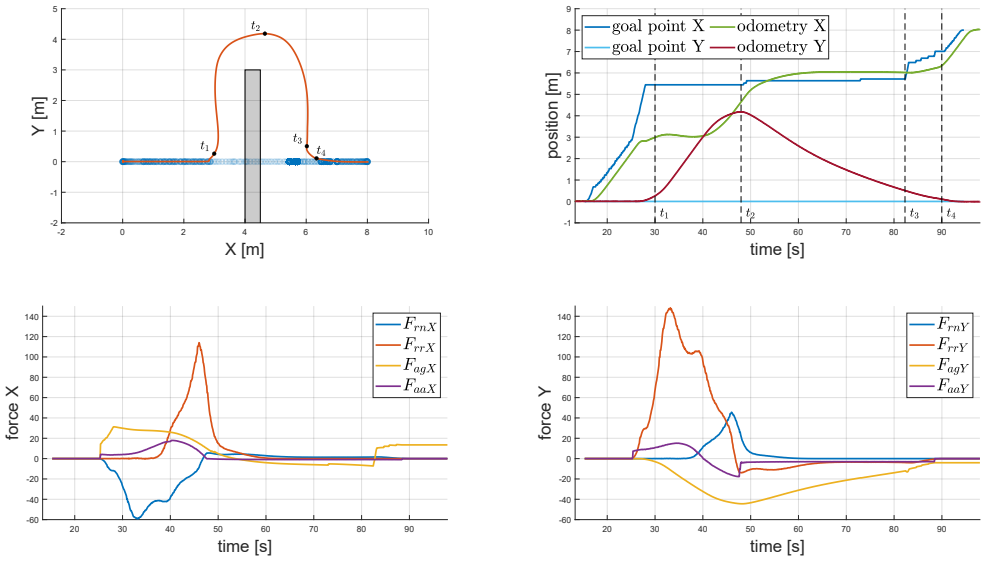


Figure 6.7: The collision-free path for the UAV was generated using artificial potential field forces. The top right graph shows the planned path as a series of goal points (marked in blue) for the UAV to reach, as well as the executed trajectory (orange line). The top left graph represents the time responses from the goal points and the position of the UAV during the simulation. The magnitudes of the forces generated by the AAPF algorithm in the X and Y axes are depicted in the bottom left and right graphs, respectively.

an obstacle is detected in close proximity to the UAV, the AAPF algorithm computes the total potential force acting on the UAV by summing the force vectors generated by the potential forces, as depicted in the graphs on the bottom row of of Fig.6.7. The resultant force vector is utilized to generate a new input pose in world reference frame W for the MPC Tracker. This causes the UAV to deviate from the planned trajectory to circumvent the obstacle, as evidenced at time point t_1.

Goal points in close proximity to the obstacle, identified by transparent markers in the top left graph of Fig.6.7, are discarded as unreachable based on the calculated force $\mathbf{F}_{gp}$. The position of the current goal point remains unchanged until time t_2, when the UAV detects the right side of the wall. At time point t_3, the distance between the UAV and the current goal point is less than the defined parameter ε, indicating that the current goal point has been reached. Finally, at time t_4, the repulsive forces $\mathbf{F}_{rn}$ and $\mathbf{F}_{rr}$ diminish below the defined threshold K_{thr}^{UAV}, signaling that the obstacle has been cleared and allowing the UAV to resume its trajectory as initially planned.

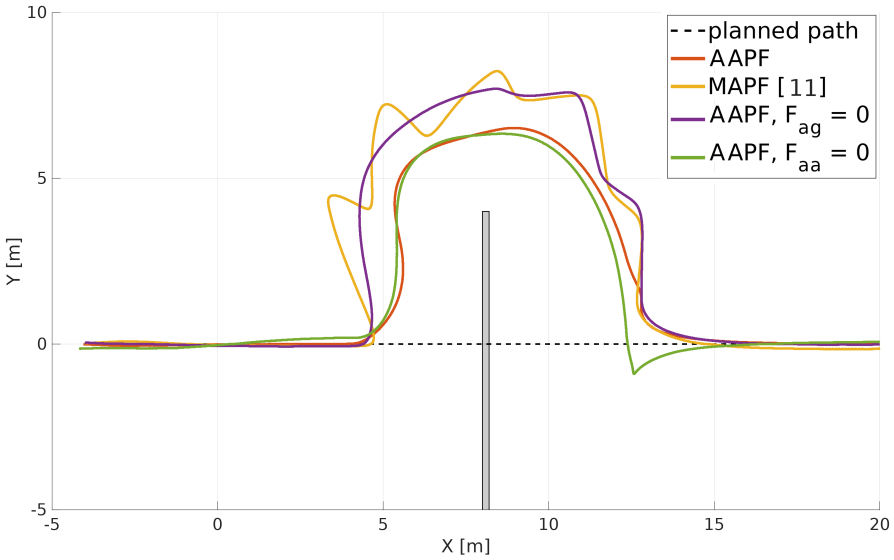
To further evaluate the efficacy of the AAPF algorithm,a comparative analysis was conducted against the MAPF algorithm [85], Additionally, the impact of each of the four potential field forces employed in the AAPF algorithm on the UAV's behavior during collision avoidance maneuvers was demonstrated across five distinct simulation scenarios varying in complexity, with subsequent analysis of the obtained results.

Table 6.1: The results of the experiments in Fig.6.8-6.9. The simulations were divided into two scenarios - the first with a wall-like obstacle and the second with a cylindrical obstacle. In each scenario, the MAPF and AAPF collision avoidance algorithms were tested. Path-tracking success is marked with ✓for successfully avoiding the obstacle and reaching the end of the planned path, × for algorithms that got stuck in local minima, and × for experiments that ended with a collision with the obstacle. In addition, for each experiment data was provided for the length of the path during execution of the collision avoidance maneuver l_{dev}, the time interval the UAV deviated from the planned path t_{dev}, and the minimum d_{min} and average d_{avg} distance between the UAV and the obstacle. All experiments were repeated 10 times and results are presented as tuples of mean and standard deviation.

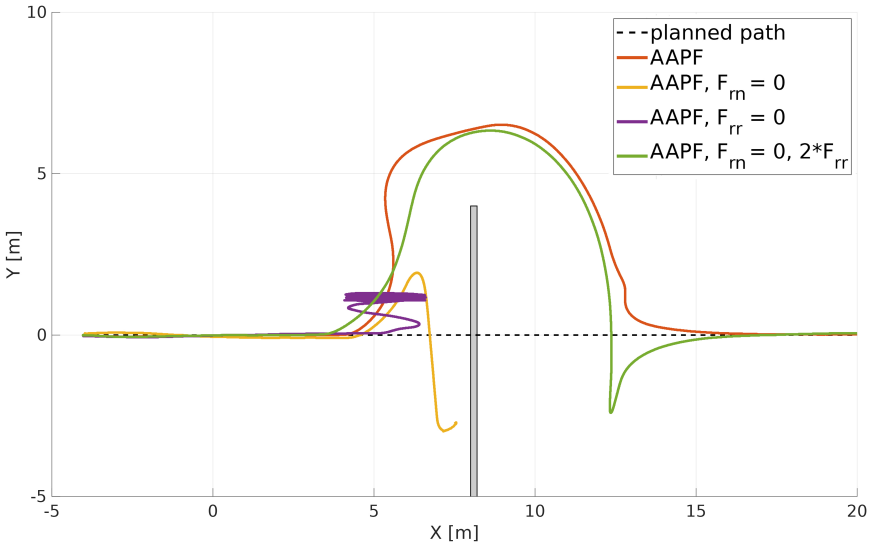
Scenario	Algorithm	Success	$l_{dev}[m]$	$t_{dev}[s]$	$d_{min}[m]$	$d_{avg}[m]$
wall obstacle	MAPF [85]	✓	(37.09, 0.90)	(74.72, 3.01)	(4.12, 0.07)	(5.23, 0.05)
	AAPF	✓	(24.28, 0.11)	**(48.84, 1.34)**	(1.81, 0.07)	(4.78, 0.41)
	AAPF, $F_{ag} = 0$	✓	(29.01, 0.52)	(61.33, 3.35)	(2.75, 0.01)	(4.76, 0.04)
	AAPF, $F_{aa} = 0$	✓	(26.16, 0.23)	(50.31, 1.36)	(1.89, 0.03)	(5.47, 0.10)
	AAPF, $F_{rn} = 0$	×	-	-0.0-		
	AAPF, $F_{rr} = 0$	×	-	-	(1.01, 0.01)	(3.62, 0.84)
	AAPF, $F_{rn} = 0, 2 \times F_{rr}$	✓	**(23.34, 0.14)**	(55.36, 1.36)	(0.99, 0.01)	(4.28, 0.09)
cylindrical obstacle	MAPF [85]	✓	(35.71, 0.42)	(48.58, 0.37)	(2.02, 0.07)	(3.79, 0.03)
	AAPF	✓	**(27.16, 0.07)**	**(41.26, 0.36)**	(2.22, 0.01)	(3.62, 0.04)
	AAPF, $F_{ag} = 0$	✓	(33.10, 0.49)	(43.08, 0.63)	(2.09, 0.07)	(3.90, 0.03)
	AAPF, $F_{aa} = 0$	✓	(27.93, 0.09)	(49.19, 0.36)	(2.16, 0.02)	(4.06, 0.05)
	AAPF, $F_{rn} = 0$	×	-	-	(1.63, 0.06)	(3.25, 0.12)
	AAPF, $F_{rr} = 0$	✓	(62.25, 0.93)	(80.45, 1.56)	(1.07, 0.05)	(4.68, 0.04)
	AAPF, $F_{rn} = 0, 2 \times F_{rr}$	×	-	-	(1.92, 0.08)	(3.48, 0.18)

The initial phase of the simulation experiments focused on testing path tracking and obstacle avoidance algorithms within two scenarios featuring simple obstacles, a wall-like structure and a cylindrical obstacle. In these scenarios, the path was initially planned as a straight line. The primary objectives included the comparison of outcomes between the AAPF and MAPF algorithms, as well as the assessment of the individual effects of each potential force integrated into the AAPF. To achieve this, simulations were executed with all forces active in the AAPF algorithm, followed by deactivation of one force at a time for sequential analyses. This iterative process was repeated for both scenarios to comprehensively evaluate the algorithm performance across varying obstacle geometries.

The comparative analysis between the MAPF and AAPF algorithms revealed notable enhancements in obstacle avoidance efficiency and path optimization with the AAPF algorithm. Notably, the AAPF algorithm demonstrates improvements in both the duration required to circumvent obstacles and the length of the avoidance path, all while maintaining comparable average and minimum distances between the UAV and the obstacle during flight.. These advancements are evident from the data depicted in Fig.6.8and Fig.6.9, indicating effective mitigation

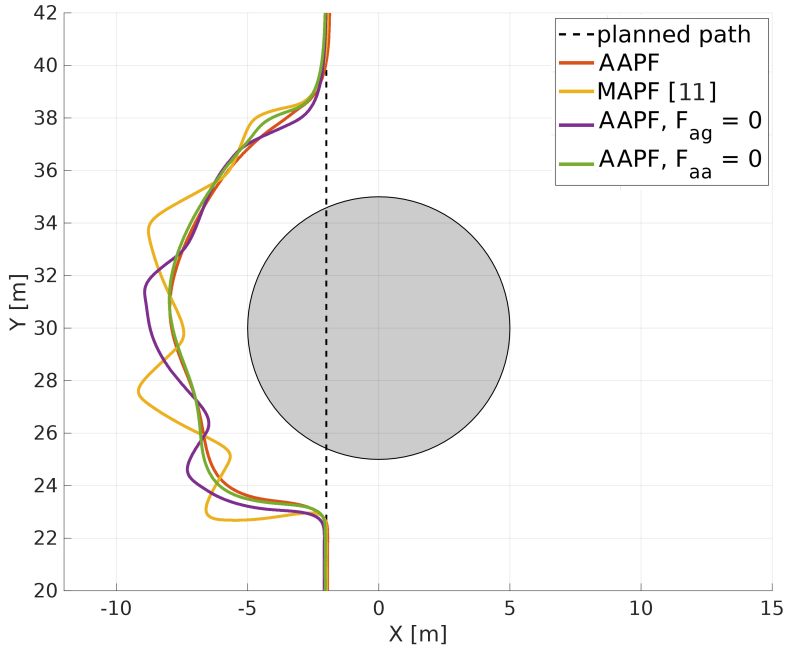


(a) While the UAV successfully avoided the obstacle using all methods and returned to the planned path, MAPF (yellow) and AAPF without the attractive force towards the goal point $\mathbf{F}_{ag}$ (purple) generated a much longer collision avoidance path. AAPF (red) and AAPF with the anchor attractive force $\mathbf{F}_{aa}$ set to zero (green) generated a similar obstacle avoidance path without oscillations. In the case where $\mathbf{F}_{aa}$ is zero, the UAV overshoots slightly when returning to the planned path.

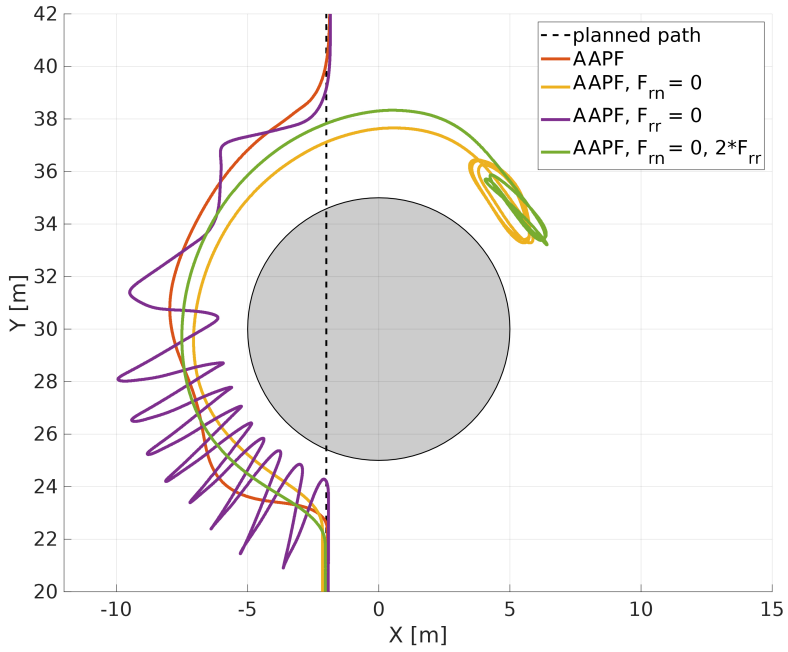


(b) In a case where there is no rotational repulsive force $\mathbf{F}_{rr}$, the UAV gets stuck in the local minima in front of the wall (purple). This method produces very similar results as the conventional potential field. In a case without the normal repulsive force $\mathbf{F}_{rn}$, the UAV could not successfully avoid the obstacle and the experiment ended with a collision (yellow). To compensate for the lack of $\mathbf{F}_{rn}$, the gain for the force $\mathbf{F}_{rr}$ was set to twice the default value (green). This generated a collision-free path for the UAV, but with a significant overshoot when returning to the planned path.

Figure 6.8: Executed flight trajectories in a simple simulation scenario with a wall-like obstacle.



(a) Results are similar to the scenario with a wall-like obstacle - in all 4 cases UAV successfully avoided the obstacle and completed the planned path. MAPF (yellow) and AAPF without the $\mathbf{F}_{ag}$ (purple) generated avoidance path with significant oscillations.



(b) Without the normal force component, the UAV got stuck in a local minima in both cases (yellow and green), regardless of the value of the $\mathbf{F}_{rr}$ gain. These are the expected results with a circular obstacle and a rotational repulsive force. In the case without the rotational component, the UAV avoided the obstacle, but with highly oscillating and longer path (purple) than in the case where all forces are active (red).

Figure 6.9: Executed flight trajectories in simulation scenario with a single circular obstacle.

of path oscillations encountered at the edges of the potential field.

Both attractive forces within the AAPF algorithm contributed positively to resolving unnecessarily large deviations from the planned path when navigating around simple obstacles. In both simulation scenarios, activating all four potential forces resulted in the shortest duration for circumnavigating the obstacle and the shortest avoidance path, particularly evident in the scenario featuring a cylindrical obstacle. In the scenario with a wall-like obstacle, the shortest avoidance path was observed when the normal repulsive force component $\mathbf{F}_{rn}$ was disabled, and the gain for the rotational repulsive force $\mathbf{F}_{rr}$ was doubled. However, this configuration led to a significantly smaller minimum distance between the UAV and the obstacle compared to the case with all forces active.

Analysis of individual force deactivation scenarios underscores the critical role of each potential force in ensuring effective obstacle avoidance. In the first scenario, deactivating the rotational repulsive force caused the UAV to become stuck in the local minima, while in the second scenario, local minima entrapment occurred when the normal repulsive force was disabled, regardless of the gain value for the rotational repulsive force. Notably, the only case in which a collision with the obstacle occurred was when the normal repulsion force was deactivated in the scenario with the wall-like obstacle.

The simulation results revealed that the distance between the obstacle and the point at which the UAV initiates collision avoidance closely matches the distance between the obstacle and the feasible waypoint to which the UAV returns to resume the planned trajectory. The precise position of the return waypoint is determined by the predefined parameter K_{thr}^{goal}. A larger value of this parameter signifies a closer proximity of the feasible waypoint to the obstacle. However, in cases where the return waypoint is too close for the UAV to reach due the effect of the repulsive potential field, the UAV will become trapped in local minima or fly in an indefinite loop around the obstacle.

In the third scenario of the simulation experiments, the goal was to navigate the UAV through a narrow L-shaped corridor while following the planned trajectory. Narrow corridors pose significant challenges for collision avoidance algorithms based on artificial potential fields, as highlighted in prior literature [86]. Despite the initially planned path did not collide with the obstacle along the first part of the trajectory, the AAPF algorithm generated a small deviation to keep the UAV equidistant from both walls in the corridor. As illustrated in Fig.6.10, the AAPF algorithm successfully generated an oscillation-free flight path for the UAV through the corridor and guided it through a sharp turn where the planned trajectory would have intersected with the obstacle.

In the fourth scenario, a simulation environment featuring a large concave obstacle was generated, which is widely recognized as a challenging problem for artificial potential field-based algorithms due to the potential for UAVs to become trapped in local minima within the

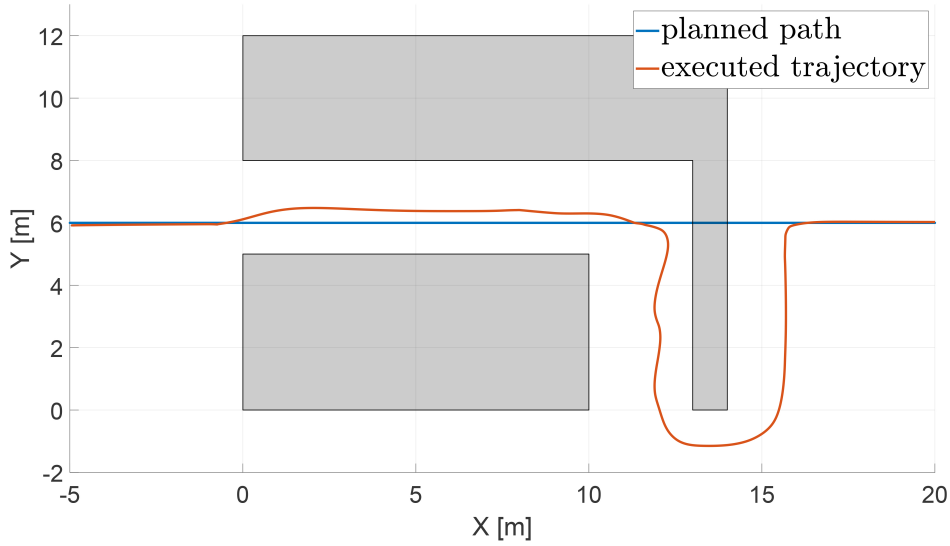


Figure 6.10: Results of a simulation experiment in a narrow L-shaped corridor, where the AAPF algorithm successfully guided the UAV through the middle of the $3m$ wide corridor and avoided collision with the obstacle at the location where the initially planned trajectory would have collided with the wall.

obstacle. Generated obstacle was intentionally larger than the scanning range of the LiDAR, making it impossible for the UAV to detect the whole obstacle within a single scan with the sensor.

The outcomes of the simulation experiment involving the large concave obstacle are presented in Fig.6.11across four distinct time points. At the initial time point $t_1 = 38$ seconds, the UAV detected only a small portion of the obstacle and initiated a collision avoidance maneuver using a clockwise direction of the rotational potential field. During the flight, the environment was mapped by adding each scan from the LiDAR into the accumulated point cloud. This allowed for the constant updating of the centroid position of the detected obstacle with new information. After a larger portion of the obstacle was detected and the UAV reached the position marked as *turning point* on the top right graph of in Fig6.11angle ρ (defined as angle of the vector from the position of the UAV to the obstacle centroid) became larger than angle ϕ (angle of the vector from the position of the UAV to the current goal point on the planned path). Consequently, the sign of angle θ (defined by equation6.12) changed, prompting a counterclockwise switch in rotational potential field direction. By $t_3 = 120$ seconds, the UAV had successfully navigated out of the concave obstacle without getting stuck in a local minimum. Finally, the results of the experiment, shown in the lower right graph of Fig.6.11at $t_4 = 141$ seconds confirm that the UAV had successfully returned to its planned flight path.

In Fig.6.12the results of the simulation are presented in a complex maze-like environment containing several unknown obstacles with different shapes. The original path was planned as a realistic scenario (e.g., inspection of a known infrastructure object in an unknown environment), where the UAV has to fly around the designated object of interest. The results show that the

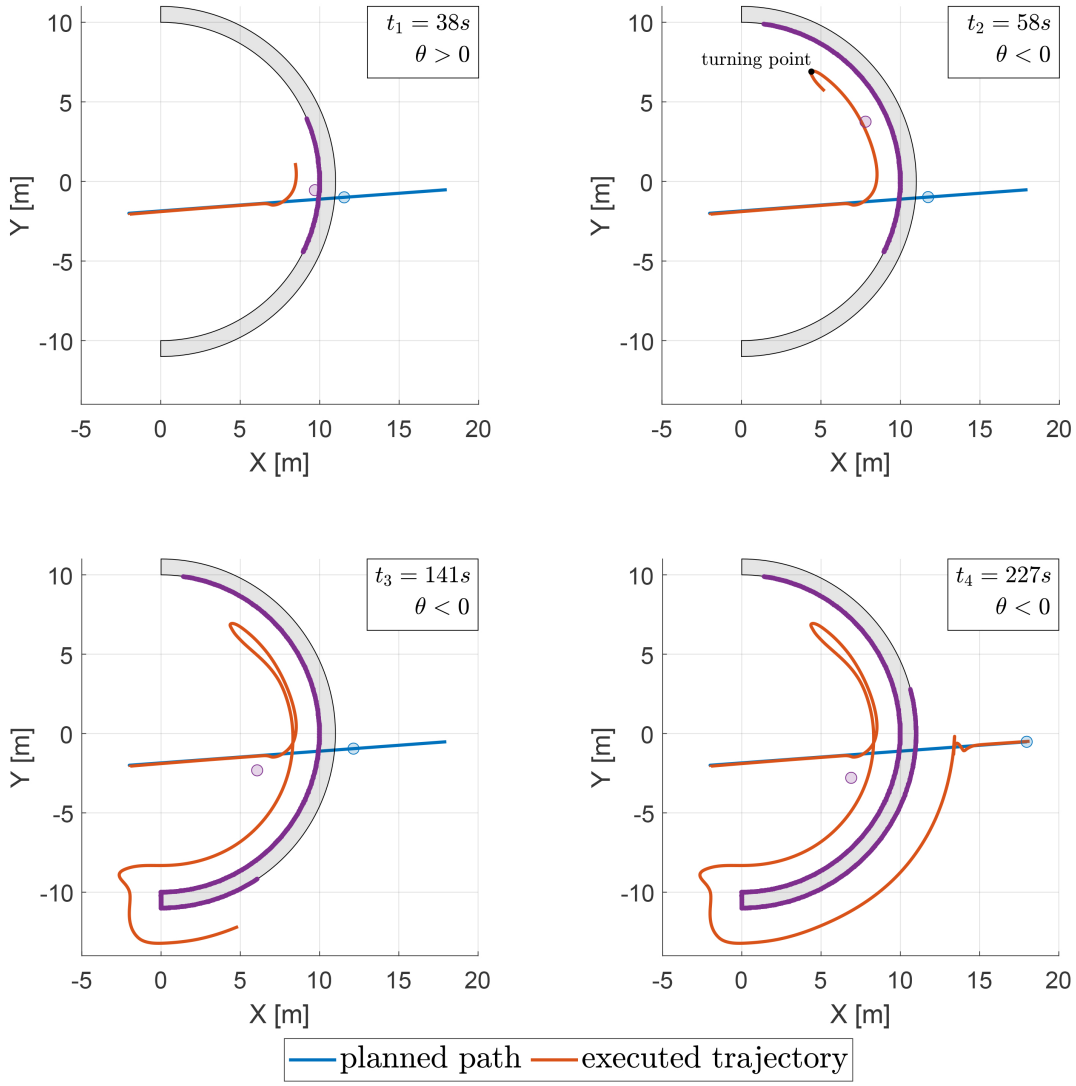


Figure 6.11: Results of a simulation experiment with a concave obstacle are displayed at four distinct time points. The planned flight path for the UAV is depicted as a blue line, while the executed trajectory up to time t_i is represented by an orange line. The purple points show the accumulated point cloud at time t_i. A blue marker indicates the current goal point on the planned path at t_i, while a purple marker represents the position the centroid of the obstacle. Angle θ is defined with the equation6.12.

UAV successfully avoided all obstacles on the path using the AAPF algorithm, including the narrow corridor and the concave obstacle, which are considered extremely difficult obstacles for algorithms based on artificial potential fields. In addition to successfully generating and executing a collision-free path, the AAPF algorithm forced the UAV to return to the planned path at the nearest safe waypoint after avoiding the obstacle.

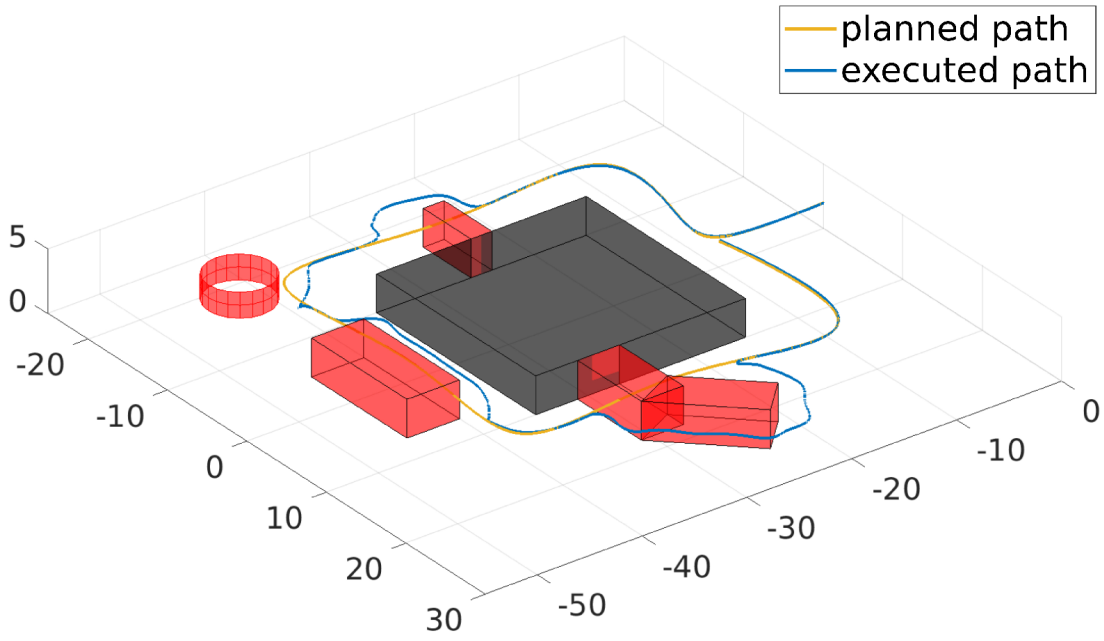


Figure 6.12: Simulation experiment in a complex, maze-like environment. Realistic path is planned around the known object of interest (black object). Red objects are unknown obstacles during the planning phase. Results show successful execution of a flight mission

6.3Experimental results

The flight experiments were conducted within a $10m \times 7m \times 3m$ flight arena, equipped with an *Optitrack* localization system for precise tracking. A *Hexsoon EDU-450* quadcopter served as the test platform, outfitted with the *Cube Orange* autopilot system responsible for low-level attitude control, and an *Intel NUC* onboard computer capable of processing data from the *SLAMTEC RPLIDAR-A3* 2D LiDAR sensor utilized for obstacle detection (Fig.6.13). Detailed description of the *Hexsoon EDU-450* quadcopter can be found in Section3.3. The implementation of the AAPF algorithm was integrated into autonomous navigation flight stack using the Robot Operating System (ROS) framework. To evaluate the efficacy of the AAPF algorithm, three distinct scenarios were designed, each featuring different configurations of obstacles and initial flight trajectories.

In the initial experimental scenario, two wall-like obstacles were positioned within the arena, with the flight trajectory of the UAV intentionally directed towards a collision course with one of them. To establish the initial set of parameters for the AAPF algorithm within this constructed environment and to gather data for subsequent analysis and algorithm evaluation, identical scenarios were initially simulated. The planned and executed trajectories, both in simulation and real-world experiments, are illustrated in Fig.6.14. Results of the experiment demonstrate that the UAV managed to successfully avoid both walls, navigating safely through the gap between them, return to and finish the initially planned trajectory.

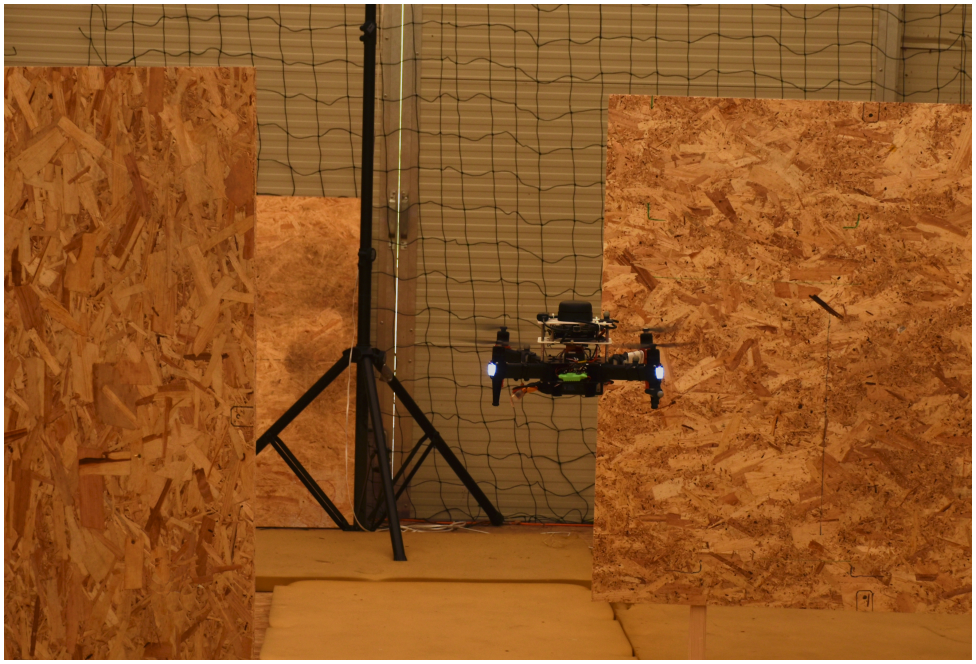


Figure 6.13: *Hexsoon EDU-450* UAV equiped with 2d LiDAR used in experimental evaluation of the AAPF algorithm.

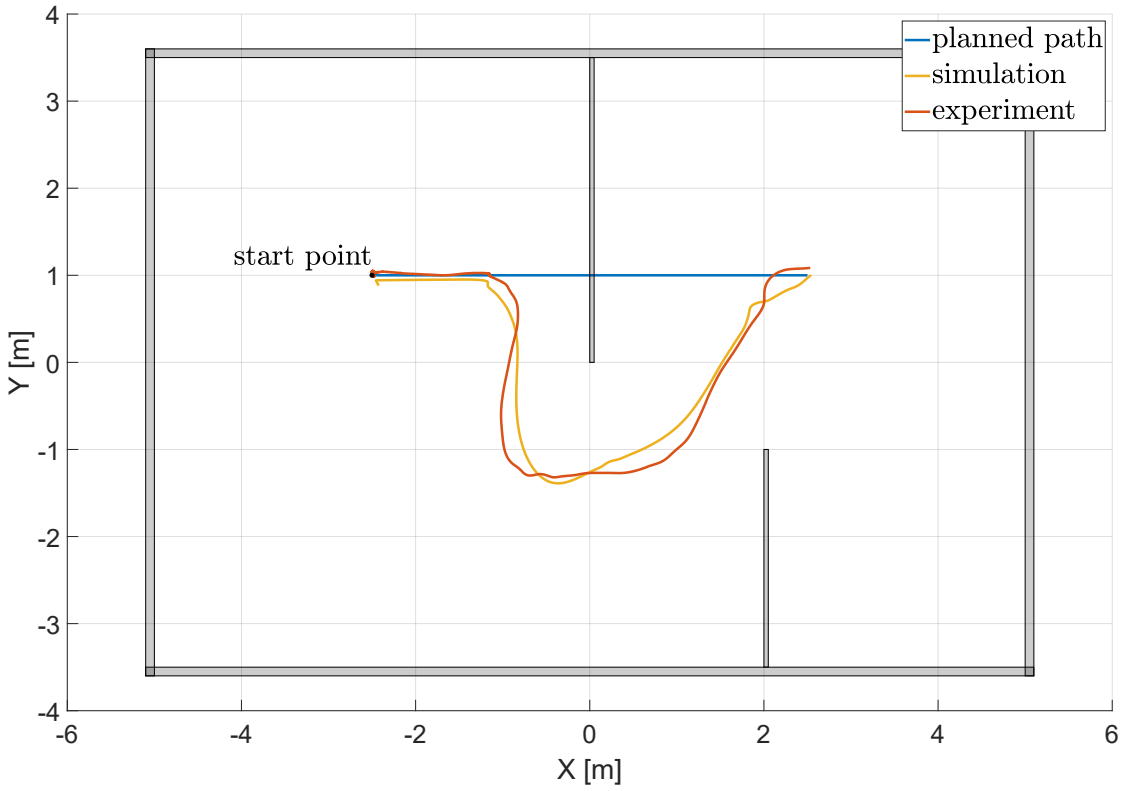


Figure 6.14: The results of the simulation and real-world experiment in the arena with two wall-like obstacles. On the graph are depicted planned trajectory (blue) and executed flight paths in simulation (yellow) and real-world experiment (orange) in the XY-plane.

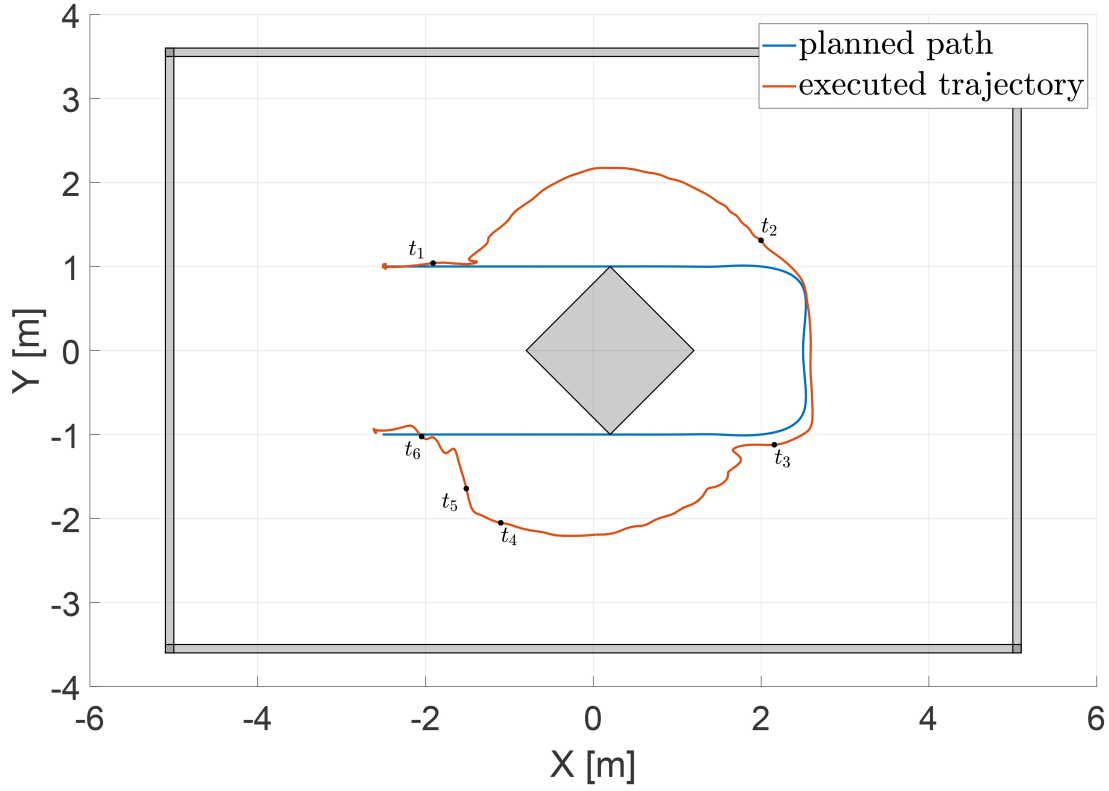
The second experimental scenario involved setting up a central obstacle, around which the initial flight trajectory was planned. The experimental results, depicted in Fig.6.15, illustrate the behavior of the UAV throughout the course of the flight. Initially, the UAV followed the planned trajectory until the repulsive forces generated by the obstacle exceeded a predetermined threshold of $K_{thr}^{UAV} = 3$, denoted at time t_1. From time point t_1 to t_2, the UAV followed the points generated by the AAPF algorithm to navigate around the obstacle. Upon successfully navigating around the obstacle, as indicated by a decrease in the magnitude of repulsive forces below K_{thr}^{UAV}, the UAV resumed its initial trajectory until encountering the obstacle for a second time at t_3. Between time points t_4 and t_5, the UAV was sufficiently far from the obstacle to move towards the current goal point on the planned trajectory. After time point t_6, the UAV resumed tracking the planned trajectory in unobstructed space until it reached the end of the planned path.

In the final experimental scenario, an initial trajectory for the UAV was designed to enter a narrow concave obstacle represented by a closed corridor. To enhance the challenge, the range of the LiDAR sensor was intentionally constrained, preventing the UAV from detecting the end of the corridor from its entrance. As a result, upon entering the corridor, the UAV perceived two parallel walls as separate obstacles, each with its own centroid. However, as the UAV progressed and detected the closed end of the corridor, all three walls were perceived as a single concave obstacle with a single centroid. This triggered a change in the direction of the rotational potential field, guiding the UAV out of the corridor.

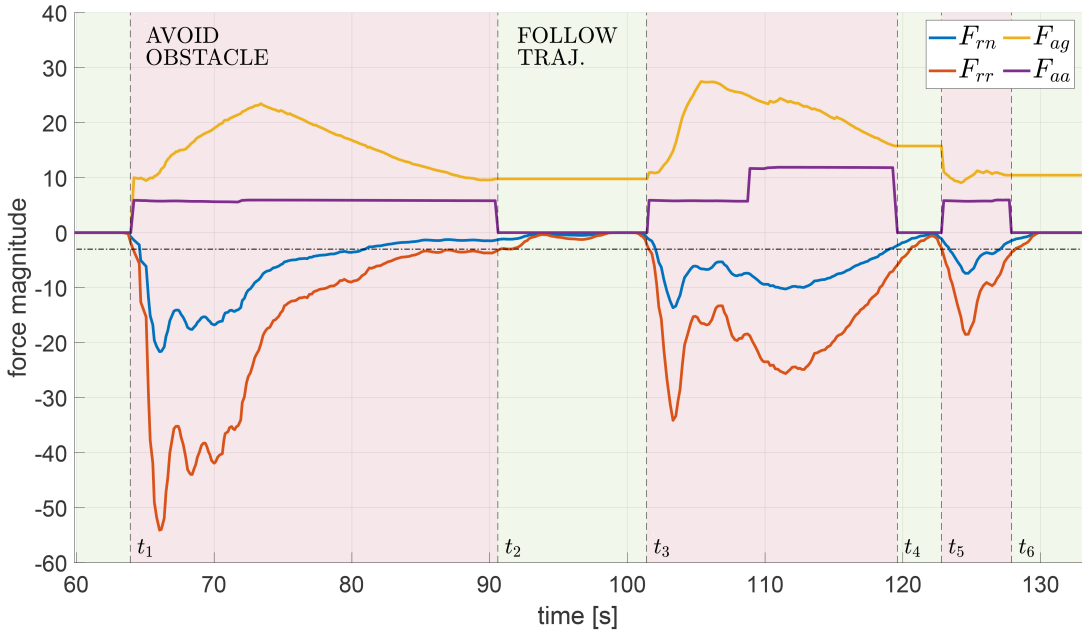
Table 6.2: Parameter settings for the potential field forces in experiments with a narrow concave obstacle.

Experiment num.	d_0	k_{rn}	k_{rr}	k_{ag}	c_{ag}	k_{aa}	b_1	k_1	b_2	k_2	k_{gp}	$K_{treshold}^{goal}$
1	1.8	250	550	8.5	6.0	2.0	6.0	1.2	1.7	5.0	200	5.0
2-5	1.8	200	500	8.5	6.0	2.0	6.0	1.2	1.7	5.0	200	5.0

To assess the repeatability of the AAPF algorithm, the same test was conducted five times without making significant changes to the potential force field parameters. In the first test, visible oscillations were observed as the UAV exited the concave obstacle. To resolve this problem, slight adjustments to the gain coefficients of the repulsive potential forces were implemented, resulting in a reduction of oscillations in subsequent tests. Furthermore, during the fifth test, the algorithm produced different clustering results compared to the other tests. This disparity was a result of changes in the environment, specifically the presence of open doors in the flight arena, which the algorithm detected as a gap in the outer walls. Consequently, the outer walls were perceived as separate objects instead of a single obstacle, leading to a slightly altered trajectory for the UAV as it returned to its initially planned path. As illustrated in Fig.6.16, the UAV successfully navigated out of the narrow corridor and resumed its planned trajectory in all five tests, demonstrating the robustness and adaptability of the AAPF algorithm in overcoming complex obstacles.



(a) Planned and executed trajectory for the experimental scenario with central obstacle in XY-plane.



(b) Magnitudes of potential field forces during the experiment with central obstacle.

Figure 6.15: Results of the experimental scenario where the initial flight trajectory for the UAV was planned close to the obstacle in the center of the area. In the graph (b), sections where the UAV is following the initially planned trajectory are highlighted with a green background, while sections where the UAV is avoiding the obstacle, are highlighted with a red background.

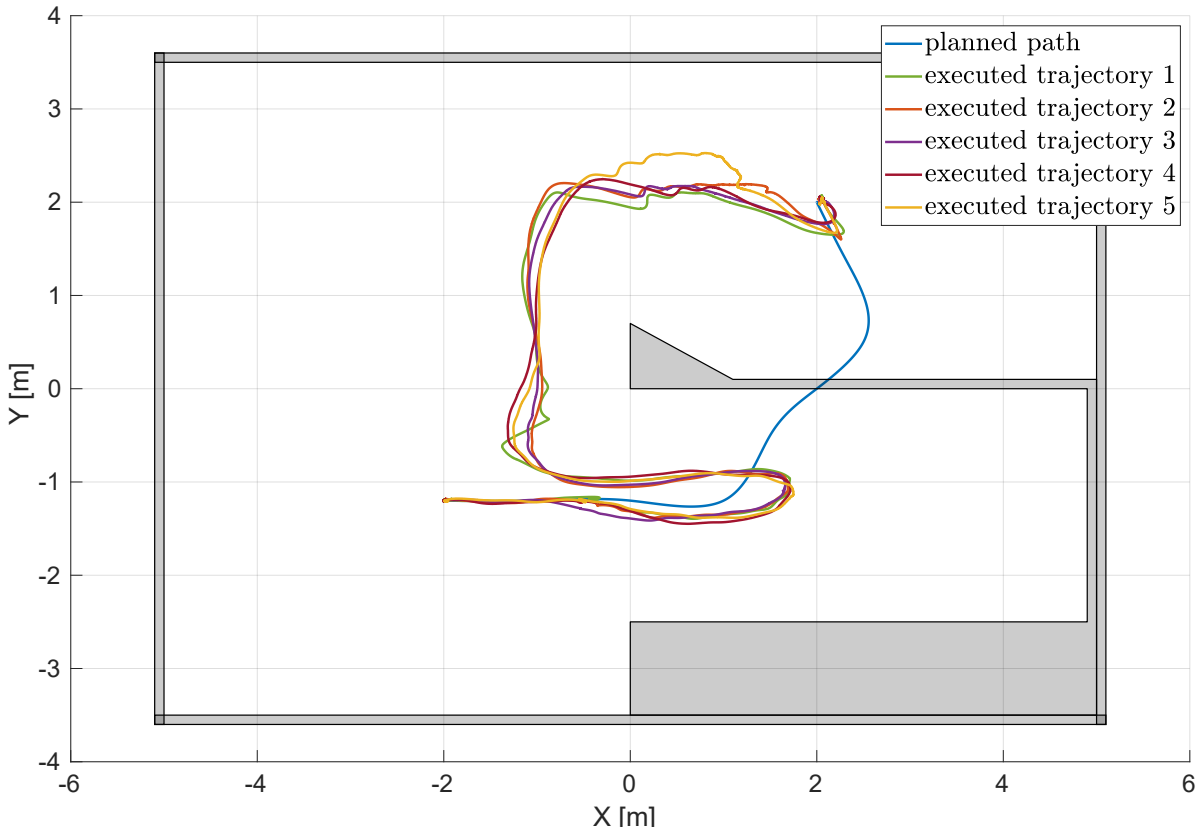


Figure 6.16: The results of the experiment with a narrow concave obstacle, showing the trajectory of the UAV as it followed the planned path into the narrow corridor where obstacles are detected as the two parallel walls. When the UAV detected the closed end of the corridor it changed the perception of the environment to a single concave obstacle with a new centroid, leading to a change in the direction of the rotational potential force. As a result, the UAV was able to successfully exit the corridor and return to the planned trajectory outside of the concave obstacle.

With these experiments, the possibility of the AAPF algorithm to be run with limited onboard resources and in real-time in order to navigate the UAV around obstacles in the environment that were unknown in the planning phase of the mission was demonstrated. This showcases the algorithm's potential for practical implementation in autonomous UAV navigation systems, especially in scenarios where real-time obstacle avoidance is crucial for mission success.

Future work concerning the AAPF algorithm should prioritize conducting a stability analysis of the system. This entails providing a mathematical formulation to demonstrate that the algorithm generates a global navigation function, ensuring that the UAV does not become trapped in local minima but consistently returns to the originally planned path. This analysis will contribute to a deeper understanding of the algorithm's behavior and its robustness in various environmental conditions, enabling its further optimization and application in real-world scenarios.

CHAPTER 7

Conclusion

The primary focus of this thesis revolves around flight path planning algorithms and trajectory tracking methods for autonomous UAV inspection of infrastructure, applicable across various scenarios characterized by differing levels of complexity in surveyed targets and surrounding environments. The work begins with an exhaustive literature overview covering the state-of-the-art in UAV path planning, navigation algorithms based on artificial potential fields, and infrastructure inspection methodologies. Subsequently, fundamental kinematic and dynamic models of the quadcopter are provided, along with an overview of the general control structure utilized in UAV systems, the camera projection model, and methodologies for determining inspection parameters to ensure comprehensive coverage and quality of acquired data.

Comprehensive descriptions and mathematical models of the developed algorithms are then presented, accompanied by the outcomes of simulations and real-world experiments. Furthermore, the results obtained from both simulated and real scenarios are thoroughly analyzed, offering insights into the performance and efficacy of the proposed methodologies across different environmental conditions and target structures.

Inspection flight path planning for unknown infrastructures based on closed isolines of artificial potential fields

This thesis introduces several path planning methods designed for autonomous UAV inspection of infrastructure, adaptable to various levels and types of prior knowledge about the surveyed structure. Throughout the development of these path planning algorithms, considerations for the feasibility and safety of the planned flight path, as well as requirements for the inspection datasets and constraints on UAV velocity and acceleration, were carefully taken into account.

The first proposed method is based on the Huygens's wave propagation principle and relies

on a known two-dimensional layout of the surveyed structure. The output of this algorithm is a minimal set of waypoints, positioned at a constant distance from the structure's side, with heading information ensuring the UAV remains perpendicular to the structure. These waypoints form the trajectory along which the UAV must traverse to gather a comprehensive inspection dataset.

The second algorithm constructs a three-dimensional envelope around a known model of the surveyed structure, comprised of points at a desired distance from the object. This envelope enables operators to select waypoints and design a comprehensive inspection path, even for structures with complex geometry.

The third proposed algorithm, designed for offline path planning, involves generating closed isolines of artificial potential fields based on the 2D layout or 3D model of the structure. The resulting path takes the form of a closed line with a desired offset from the structure's surface, which the UAV follows at a specified altitude and orientation.

Furthermore, a real-time flight path planning algorithm was developed for navigating around unknown infrastructure objects using closed isolines of artificial potential fields. In this approach, the UAV simultaneously scans the structure's side surface while planning a segment of the closed isoline for navigation around the object. It maintains a constant altitude and predetermined distance from the object's surface, adjusting altitude between passes and ensuring the inspection sensor maintains a perpendicular orientation to the structure's surface.

These algorithms underwent extensive testing and validation in various simulation scenarios under realistic conditions. Subsequently, their suitability for real-world applications was evaluated through carefully designed inspection experiments on different infrastructure objects such as residential buildings, bridges, and wind turbines.

Flight path tracking method in an unknown environment based on artificial potential fields

The second part of this thesis focuses on tracking the planned flight path within an unknown environment while simultaneously detecting and avoiding collisions with obstacles. In addition to generating a safe flight path, the developed algorithm aims to minimize deviations and closely follow the originally planned path in unobstructed space.

The flight path tracking and collision avoidance method, denoted as the Augmented Artificial Potential Field (AAPF) algorithm, operates on the principle of repulsive potential forces that repel the UAV from detected obstacles and attractive potential forces that guide it back to the initially planned path after executing avoidance maneuvers. The repulsive potential field comprises two components: a normal component and a rotational one, designed to prevent the UAV from becoming trapped in local minima created by various obstacle configurations in the environment. Similarly, the attractive potential force is also composed of two components: an

anchor force aimed at minimizing unnecessary deviations during collision avoidance, and a goal force responsible for steering the UAV back to its original path.

Newely created waypoint serves as input for the MPC tracker module that incorporates predefied velocity and acceleration constraints for the UAV thus correcting the UAV trajectory based on the current system state and the model dynamics.

The result of the potential forces acting on the UAV is a newly derived waypoint, which then serves as input for the MPC tracker module, which integrates predefined velocity and acceleration constraints for the UAV. This module functions to adjust the UAV trajectory in real-time, based on the current system state and model dynamics.

Furthermore, the algorithm assesses the feasibility of waypoints along the initially planned path to increase trajectory following efficiency. It accomplishes this by calculating the repulsive potential obstacles exert on waypoints in their vicinity. If the repulsive potential exceeds a predefined threshold, the selected waypoint is determined unreachable and is discarded from the flight path.

The developed algorithm was validated through a series of simulations and laboratory experiments, followed by a detailed analysis of the results produced. During the simulations, the effect of each of the four potential forces was individually assessed on the behavior of the UAV, and the executed flight trajectories were compared for evaluation purposes.

Bibliography

[1]Quan, Q., Introduction to multicopter design and control. Springer, 2017.

[2]Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N. S., Khreishah, A., Guizani, M., "Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges", IEEE Access, Vol. 7, 2019, str. 48 572-48 634.

[3]Semsch, E., Jakob, M., Pavlicek, D., Pechoucek, M., "Autonomous uav surveillance in complex urban environments", in 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Vol. 2, 2009, str. 82-85.

[4]Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., Sousa, J. J., "Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry", Remote Sensing, Vol. 9, No. 11, 2017, dostupno na: https://www.mdpi.com/2072-4292/9/11/1110

[5]Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I. L., Ruess, F., Suppa, M., Burschka, D., "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue", IEEE Robotics Automation Magazine, Vol. 19, No. 3, 2012, str. 46-56.

[6]Popovic, M., Vidal-Calleja, T., Hitz, G., Chung, J., Sa, I., Siegwart, R., Nieto, J., "An informative path planning framework for uav-based terrain monitoring", Autonomous Robots, Vol. 44, 07 2020.

[7]Campana, S., "Drones in archaeology. state-of-the-art and future perspectives", Archaeological Prospection, Vol. 24, No. 4, 2017, str. 275–296.

[8]Ellenberg, A., Branco, L., Krick, A., Bartoli, I., Kontsos, A., "Use of unmanned aerial vehicle for quantitative infrastructure evaluation", Journal of Infrastructure Systems, Vol. 21, No. 3, 2015, str. 04014054.

[9]Stokkeland, M., Klausen, K., Johansen, T. A., "Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection", in 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 2015, str. 998-1007.

[10]Car, M., Markovic, L., Ivanovic, A., Orsag, M., Bogdan, S., "Autonomous wind-turbine blade inspection using lidar-equipped unmanned aerial vehicle", IEEE Access, Vol. 8, 2020, str. 131 380-131 387.

[11]Luque-Vega, L. F., Castillo-Toledo, B., Loukianov, A., Gonzalez-Jimenez, L. E., "Power line inspection via an unmanned aerial system based on the quadrotor helicopter", in MELECON 2014-2014 17th IEEE Mediterranean electrotechnical conference. IEEE, 2014, str. 393–397.

[12]Yang, L., Fan, J., Liu, Y., Li, E., Peng, J., Liang, Z., "A review on state-of-the-art power line inspection techniques", IEEE Transactions on Instrumentation and Measurement, Vol. 69, No. 12, 2020, str. 9350-9365.

[13]Tognon, M., Chávez, H. A. T., Gasparin, E., Sablé, Q., Bicego, D., Mallet, A., Lany, M., Santi, G., Revaz, B., Cortés, J. *et al.*, "A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants", IEEE Robotics and Automation Letters, Vol. 4, No. 2, 2019, str. 1846–1851.

[14]Trujillo, M. Á., Martínez-de Dios, J. R., Martín, C., Viguria, A., Ollero, A., "Novel aerial manipulator for accurate and robust industrial ndt contact inspection: A new tool for the oil and gas inspection industry", Sensors, Vol. 19, No. 6, 2019, str. 1305.

[15]Ikeda, T., Yasui, S., Fujihara, M., Ohara, K., Ashizawa, S., Ichikawa, A., Okino, A., Oomichi, T., Fukuda, T., "Wall contact by octo-rotor uav with one dof manipulator for bridge inspection", in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, str. 5122–5127.

[16]Metni, N., Hamel, T., "A uav for bridge inspection: Visual servoing control law with orientation limits", Automation in construction, Vol. 17, No. 1, 2007, str. 3–10.

[17]Ereiz, S., Bartolac, M., Gori čanec, J., Orsag, M., "Application of uavs for assessment of bridge infrastructure", Građevinar, Vol. 73, No. 11., 2021, str. 1095–1106.

[18]Aggarwal, S., Kumar, N., "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges", Computer communications, Vol. 149, 2020, str. 270–299.

[19]Yang, L., Qi, J., Xiao, J., Yong, X., "A literature review of uav 3d path planning", in Proceeding of the 11th world congress on intelligent control and automation. IEEE, 2014, str. 2376–2381.

[20]Sathyaraj, B. M., Jain, L. C., Finn, A., Drake, S., "Multiple uavs path planning algorithms: a comparative study", Fuzzy Optimization and Decision Making, Vol. 7, 2008, str. 257–267.

[21]Yasin, J. N., Mohamed, S. A. S., Haghbayan, M.-H., Heikkonen, J., Tenhunen, H., Plosila, J., "Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches", IEEE Access, Vol. 8, 2020, str. 105 139-105 155.

[22]Wang, X., Yadav, V., Balakrishnan, S. N., "Cooperative uav formation flying with obstacle/collision avoidance", IEEE Transactions on Control Systems Technology, Vol. 15, No. 4, 2007, str. 672-679.

[23]Markovi ć, L., Kovač, M., Milijas, R., Car, M., Bogdan, S., "Error state extended kalman filter multi-sensor fusion for unmanned aerial vehicle localization in gps and magnetometer denied indoor environments", in 2022 International Conference on Unmanned Aircraft Systems (ICUAS), 2022, str. 184-190.

[24]Geraerts, R., Overmars, M. H., "A comparative study of probabilistic roadmap planners", in Algorithmic foundations of robotics V. Springer, 2004, str. 43–57.

[25]Chen, J., Li, M., Yuan, Z., Gu, Q., "An improved a* algorithm for uav path planning problems", in 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Vol. 1, 2020, str. 958-962.

[26]Chengjun, Z., Xiuyun, M., "Spare a* search approach for uav route planning", in 2017 IEEE International Conference on Unmanned Systems (ICUS), 2017, str. 413-417.

[27]Chen, X., Li, G.-y., Chen, X.-m., "Path planning and cooperative control for multiple uavs based on consistency theory and voronoi diagram", in 2017 29th Chinese Control And Decision Conference (CCDC), 2017, str. 881-886.

[28]"Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan", Algorithmic and computational robotics, 2001, str. 303–307.

[29]Kuffner, J., LaValle, S., "Rrt-connect: An efficient approach to single-query path planning", in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), Vol. 2, 2000, str. 995-1001 vol.2.

[30]Noreen, I., Khan, A., Habib, Z., "Optimal path planning using rrt* based approaches: a survey and future directions", International Journal of Advanced Computer Science and Applications, Vol. 7, No. 11, 2016.

[31]Samaniego, F., Sanchis, J., Garcia-Nieto, S., Simarro, R., "Uav motion planning and obstacle avoidance based on adaptive 3d cell decomposition: Continuous space vs discrete space", in 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM). IEEE, 2017, str. 1–6.

[32]Khatib, O., "Real-time obstacle avoidance for manipulators and mobile robots", in Proceedings. 1985 IEEE International Conference on Robotics and Automation, Vol. 2, 1985, str. 500-505.

[33]Ren, J., McIsaac, K., Patel, R., "Modified newton's method applied to potential field-based navigation for mobile robots", IEEE Transactions on Robotics, Vol. 22, No. 2, 2006, str. 384-391.

[34]Saravanakumar, S., Asokan, T., "Multipoint potential field method for path planning of autonomous underwater vehicles in 3d space", Intelligent Service Robotics, Vol. 6, 10 2013, str. 211-224.

[35]Mac, T. T., Copot, C., De Keyser, R., Ionescu, C. M., "The development of an autonomous navigation system with optimal control of an uav in partly unknown indoor environment", Mechatronics, Vol. 49, 2018, str. 187–196.

[36]Paul, T., Krogstad, T. R., Gravdahl, J. T., "Modelling of uav formation flight using 3d potential field", Simulation Modelling Practice and Theory, Vol. 16, No. 9, 2008, str. 1453–1462.

[37]Hengster-Movri ć, K., Bogdan, S., Draganjac, I., "Multi-agent formation control based on bell-shaped potential functions", Journal of Intelligent and Robotic Systems, Vol. 58, 2010, str. 165–189.

[38]Ma'Arif, A., Rahmaniar, W., Vera, M. A. M., Nuryono, A. A., Majdoubi, R., Çakan, A., "Artificial potential field algorithm for obstacle avoidance in uav quadrotor for dynamic environment", in 2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), 2021, str. 184-189.

[39]Sun, J., Tang, J., Lao, S., "Collision avoidance for cooperative uavs with optimized artificial potential field algorithm", IEEE Access, Vol. 5, 2017, str. 18 382-18 390.

[40]Liu, Y., Zhao, Y., "A virtual-waypoint based artificial potential field method for uav path planning", in 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), 2016, str. 949-953.

[41]Zhou, L., Li, W., "Adaptive artificial potential field approach for obstacle avoidance path planning", in 2014 Seventh International Symposium on Computational Intelligence and Design, Vol. 2, 2014, str. 429-432.

[42]Choi, D., Lee, K., Kim, D., "Enhanced potential field-based collision avoidance for unmanned aerial vehicles in a dynamic environment", in AIAA scitech 2020 forum, 2020, str. 0487.

[43]Pan, Z., Zhang, C., Xia, Y., Xiong, H., Shao, X., "An improved artificial potential field method for path planning and formation control of the multi-uav systems", IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 69, No. 3, 2022, str. 1129-1133.

[44]Wu, E., Sun, Y., Huang, J., Zhang, C., Li, Z., "Multi uav cluster control method based on virtual core in improved artificial potential field", IEEE Access, Vol. 8, 2020, str. 131 647-131 661.

[45]Iswanto, I., Ma'arif, A., Wahyunggoro, O., Cahyadi, A. I., "Artificial potential field algorithm implementation for quadrotor path planning", International Journal of Advanced Computer Science and Applications, Vol. 10, No. 8, 2019.

[46]Jayaweera, H. M., Hanoun, S., "A dynamic artificial potential field (d-apf) uav path planning technique for following ground moving targets", IEEE Access, Vol. 8, 2020, str. 192 760-192 776.

[47]Valavanis, K., Hebert, T., Kolluru, R., Tsourveloudis, N., "Mobile robot navigation in 2-d dynamic environments using an electrostatic potential field", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol. 30, No. 2, 2000, str. 187-196.

[48]Chiang, H.-T., Malone, N., Lesser, K., Oishi, M., Tapia, L., "Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments", in 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, str. 2347-2354.

[49]Scherer, S., Singh, S., Chamberlain, L., Elgersma, M., "Flying fast and low among obstacles: Methodology and experiments", The International Journal of Robotics Research, Vol. 27, No. 5, 2008, str. 549–574.

[50]Nooralishahi, P., Ibarra-Castanedo, C., Deane, S., López, F., Pant, S., Genest, M., Avdelidis, N. P., Maldague, X. P. V., "Drone-based non-destructive inspection of industrial sites: A review and case studies", Drones, Vol. 5, No. 4, 2021, dostupno na: https://www.mdpi.com/2504-446X/5/4/106

[51]Halder, S., Afsari, K., "Robots in inspection and monitoring of buildings and infrastructure: A systematic review", Applied Sciences, Vol. 13, No. 4, 2023, str. 2304.

[52]Nikolic, J., Burri, M., Rehder, J., Leutenegger, S., Huerzeler, C., Siegwart, R., "A uav system for inspection of industrial facilities", in 2013 IEEE Aerospace Conference, 2013, str. 1-8.

[53]Yu, H., Yang, W., Zhang, H., He, W., "A uav-based crack inspection system for concrete bridge monitoring", in 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017, str. 3305-3308.

[54]Bolourian, N., Hammad, A., "Lidar-equipped uav path planning considering potential locations of defects for bridge inspection", Automation in Construction, Vol. 117, 2020, str. 103250, dostupno na:https://www.sciencedirect.com/science/article/pii/S0926580519309355

[55]Ribeiro, D., Santos, R., Cabral, R., Saramago, G., Montenegro, P., Carvalho, H., Correia, J., Calçada, R., "Non-contact structural displacement measurement using unmanned aerial vehicles and video-based systems", Mechanical Systems and Signal Processing, Vol. 160, 2021, str. 107869, dostupno na:https://www.sciencedirect.com/science/article/pii/S0888327021002648

[56]Yan, Z., Jin, Z., Teng, S., Chen, G., Bassir, D., "Measurement of bridge vibration by uavs combined with cnn and klt optical-flow method", Applied Sciences, Vol. 12, No. 10, 2022, dostupno na:https://www.mdpi.com/2076-3417/12/10/5181

[57]Jenssen, R., Roverso, D. *et al.*, "Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning", International Journal of Electrical Power & Energy Systems, Vol. 99, 2018, str. 107–120.

[58]Xu, C., Li, Q., Zhou, Q., Zhang, S., Yu, D., Ma, Y., "Power line-guided automatic electric transmission line inspection system", IEEE Transactions on Instrumentation and Measurement, Vol. 71, 2022, str. 1-18.

[59]Vemula, S., Frye, M., "Mask r-cnn powerline detector: A deep learning approach with applications to a uav", in 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), 2020, str. 1-6.

[60]Lopez Lopez, R., Batista Sanchez, M. J., Perez Jimenez, M., Arrue, B. C., Ollero, A., "Autonomous uav system for cleaning insulators in power line inspection and maintenance", Sensors, Vol. 21, No. 24, 2021, dostupno na:https://www.mdpi.com/1424-8220/21/24/8488

[61]Wang, L., Zhang, Z., "Automatic detection of wind turbine blade surface cracks based on uav-taken images", IEEE Transactions on Industrial Electronics, Vol. 64, No. 9, 2017, str. 7293-7303.

[62]Kanellakis, C., Fresk, E., Mansouri, S. S., Kominiak, D., Nikolakopoulos, G., "Towards visual inspection of wind turbines: A case of visual data acquisition using autonomous aerial robots", IEEE Access, Vol. 8, 2020, str. 181 650-181 661.

[63]Kocer, B. B., Orr, L., Stephens, B., Kaya, Y. F., Buzykina, T., Khan, A., Kovac, M., "An intelligent aerial manipulator for wind turbine inspection and repair", in 2022 UKACC 13th International Conference on Control (CONTROL), 2022, str. 226-227.

[64]Chung, H.-M., Maharjan, S., Zhang, Y., Eliassen, F., Strunz, K., "Placement and routing optimization for automated inspection with unmanned aerial vehicles: A study in offshore wind farm", IEEE Transactions on Industrial Informatics, Vol. 17, No. 5, 2021, str. 3032-3043.

[65]"Energy aware bim cloud platform in a cost-effective building renovation context - encore project",https://encorebim.eu/, accessed: 2024-04-22.

[66]"Autonomous uav inspection of wind turbine blades - aerowind project",https://www.esmera-project.eu/welcome/, accessed: 2024-04-22.

[67]"Autonomous system for assessment and prediction of infrastructure integrity - asap project",https://asap-project.com/en/, accessed: 2024-04-22.

[68]Goricanec, J., Ereiz, S., Orsag, M., Duvnjak, I., "Identification of the dynamic parameters of bridge elements using unmanned aerial vehicle", Journal of Sound and Vibration, Vol. 566, 2023, str. 117901.

[69]Pham, H., Pham, Q., "A new approach to time-optimal path parameterization based on reachability analysis", IEEE Transactions on Robotics, Vol. 34, No. 3, June 2018, str. 645-659.

[70]Orsag, M., Korpela, C., Oh, P., Bogdan, S., Ollero, A., Aerial manipulation. Springer, 2018.

[71]Luukkonen, T., "Modelling and control of quadcopter", Independent research project in applied mathematics, Espoo, Vol. 22, No. 22, 2011.

[72]Baca, T., Hert, D., Loianno, G., Saska, M., Kumar, V., "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles", in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, str. 6753-6760.

[73]Zhang, Z., "Camera parameters (intrinsic, extrinsic)", in Computer Vision: A Reference Guide, Ikeuchi, K., (ur.). Boston, MA: Springer US, 2014, str. 81–85.

[74]Orych, A., "Review of methods for determining the spatial resolution of uav sensors", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 40, 2015, str. 391–395.

[75]Lee, J., Sung, S., "Evaluating spatial resolution for quality assurance of uav images", Spatial Information Research, Vol. 24, 2016, str. 141–154.

[76]Guenard, N., Hamel, T., Mahony, R., "A practical visual servo control for an unmanned aerial vehicle", IEEE Transactions on Robotics, Vol. 24, No. 2, 2008, str. 331-340.

[77]Fabresse, F. R., Caballero, F., Maza, I., Ollero, A., "Localization and mapping for aerial manipulation based on range-only measurements and visual markers", in 2014 IEEE international conference on robotics and automation (ICRA). IEEE, 2014, str. 2100–2106.

[78]Gori čanec, J., Ivanović, A., Marković, L., Pavlak, I., Bogdan, S., "Civil infrastructure data acquisition in urban environments based on multi-uav mission", in 2021 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2021, str. 542–551.

[79]"Geography markup language",https://www.ogc.org/standards/gml, accessed: 2021-02-09.

[80]Miller, D. A. B., "Huygens's wave propagation principle corrected", Opt. Lett., Vol. 16, No. 18, Sep 1991, str. 1370–1372, dostupno na:http://ol.osa.org/abstract.cfm?URI=ol-16-18-1370

[81]Rusu, R. B., Cousins, S., "3D is here: Point Cloud Library (PCL)", in IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 2011.

[82]Deserno, M., "How to generate equidistributed points on the surface of a sphere", If Polymerforshung (Ed.), Vol. 99, No. 2, 2004, str. 1.

[83]Maple, C., “Geometric design and space planning using the marching squares and marching cube algorithms”, in 2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings, 2003, str. 90-95.

[84]Goricanec, J., Milas, A., Markovic, L., Bogdan, S., “Collision-free trajectory following with augmented artificial potential field using uavs”, IEEE Access, 2023.

[85]Batinovic, A., Goricanec, J., Markovic, L., Bogdan, S., “Path planning with potential field-based obstacle avoidance in a 3d environment by an unmanned aerial vehicle”, in 2022 International Conference on Unmanned Aircraft Systems (ICUAS), 2022, str. 394-401.

[86]Schleich, D., Behnke, S., “Predictive angular potential field-based obstacle avoidance for dynamic uav flights”, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022.

Biography

Jurica Goričanec obtained his master's degree in 2013 from the University of Zagreb Faculty of Electrical Engineering and Computing (UNIZG-FER). Following his graduation, he worked as an electrical engineer at Geoarheo d.o.o., where he focused on utilizing unmanned aerial vehicles for mapping and surveying archaeological sites and civil infrastructure. In 2020, he joined the Laboratory for Robotics and Intelligent Control Systems (LARICS) at UNIZG-FER as a research assistant under the mentorship of Prof. Stjepan Bogdan. During his undergraduate studies, he was awarded the Rector's Award for his contribution on the "Spincopter" project. His research is primarily focused on robotics, aerial manipulation and the practical application of unmanned aerial vehicles. As a Ph.D. student, he actively participated in various projects including ENCORE (ENergy aware BIM Cloud Platform in a COst-effective Building REnovation Context), AEROWIND (Autonomous UAV inspection of wind turbine blades), ASAP (Autonomous System for Assessment and Prediction of Infrastructure Integrity) and HEKTOR (Heterogeneous autonomous robotic system in viticulture and mariculture). He was part of the winning team in the MBZIRC2023 (Mohamed Bin Zayed International Robotics Challenge) robotics competition. So far, he has authored and co-authored 4 conference papers and 5 journal papers.

List of publications

Journal papers

1.Gori čanec, J., Ereiz, S., Orsag, M., Duvnjak, I., "Identification of the dynamic parameters of bridge elements using unmanned aerial vehicle", Journal of Sound and Vibration, Vol. 566, 2023., 117901

2.Markovi ć, L., Petric, F., Ivanović, A., Goričanec, J., Car, M., Orsag, M., Bogdan, S., "Towards A Standardized Aerial Platform: ICUAS'22 Firefighting Competition" Journal of intelligent & robotic systems, Vol. 108, Issue 3, 2023., 52

3.Gori čanec, J., Milas, A., Marković, L. and Bogdan, S., "Collision-Free Trajectory Following With Augmented Artificial Potential Field Using UAVs", IEEE Access, Vol. 11, 2023., pp. 83492-83506

4.Kapetanovi ć, N., Goričanec, J., Vatavuk, I., Hrabar, I., Stuhne, D., Vasiljević, G., Kovačić, Z., Mišković, N., Antolović, N., Anić, M., Kozina, B., "Heterogeneous autonomous robotic system in viticulture and mariculture: Vehicles development and systems integration", Sensors, Vol. 22, Issue 8, 2022., 2961

5.Ereiz, S., Bartolac, M., Gori čanec, J. and Orsag, M., "Application of UAVs for assessment of bridge infrastructure" Građevinar, Vol. 73, Issue 11, 2021., pp.1095-1106

Conference papers

1.Batinovi ć, A., Goričanec, J., Marković, L., Bogdan, S., "Path planning with potential field-based obstacle avoidance in a 3D environment by an unmanned aerial vehicle", 2022 International Conference on Unmanned Aircraft Systems (ICUAS), 2022., pp. 394-401

2.Gori čanec, J., Kapetanović, N., Vatavuk, I., Hrabar, I., Vasiljević, G., Gledec, G., Stuhne, D., Bogdan, S., Orsag, M., Petrović, T., Mišković, N., "Heterogeneous autonomous robotic system in viticulture and mariculture-project overview", 16th international conference on telecommunications (ConTEL), 2021., pp. 181-188

3.Gori čanec, J., Ivanović, A., Marković, L., Pavlak, I., Bogdan, "Civil infrastructure data acquisition in urban environments based on multi-UAV mission", 2021 International Conference on Unmanned Aircraft Systems (ICUAS), 2021., pp. 542-551

4.Hrabar, I., Gori čanec, J., Kovačić, Z., "Towards autonomous navigation of a mobile robot in a steep slope vineyard", 44th International Convention on Information, Communication and Electronic Technology (MIPRO), 2021., pp. 1119-1124

Životopis

Jurica Goričanec je 2013. godine diplomirao na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu (UNIZG-FER). Nakon diplome, radio je kao inženjer elektrotehnike u tvrtki Geoarheo d.o.o., gdje se fokusirao na korištenje bespilotnih letjelica području kartiranja i istraživanja arheoloških nalazišta te inspekcije građevinske infrastrukture. Godine 2020. pridružio se Laboratoriju za Robotiku i Inteligentne Sustave Upravljanja (LARICS) na UNIZG-FER-u kao istraživački asistent pod mentorstvom profesora Stjepana Bogdana. Tijekom preddiplomskog studija, nagrađen je Rektorovom nagradom za svoj doprinos u projektu "Samorotirajuća letjelica". Njegova istraživanja primarno su usmjerena na robotiku, zračnu manipulaciju i praktičnu primjenu bespilotnih letjelica. Kao doktorand, aktivno je sudjelovao u različitim projektima, uključujući ENCORE (ENergy aware BIM Cloud Platform in a COst-effective Building REnovation Context), AEROWIND (Autonomous UAV inspection of wind turbine blades), ASAP (Autonomous System for Assessment and Prediction of Infrastructure Integrity) i HEKTOR (Heterogeneous autonomous robotic system in viticulture and mariculture). Bio je dio pobjedničkog tima na natjecanju u robotici MBZIRC2023 (Mohamed Bin Zayed International Robotics Challenge). Do sada je autor ili koautor četiri konferencijska rada i pet radova u časopisima.