



NATIONAL TECHNICAL UNIVERSITY OF ATHENS

PHD DISSERTATION

**Motion control strategies for the
autonomous sampling, transportation
and release of loads suspended to
multicopters**

Author:

Fotios PANETSOS

Supervisor:

Prof. Kostas J.
KYRIAKOPOULOS

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Control Systems Laboratory
School of Mechanical Engineering

May 2024



NATIONAL TECHNICAL UNIVERSITY OF ATHENS

PHD DISSERTATION

**Motion control strategies for the
autonomous sampling, transportation
and release of loads suspended to
multicopters**

Author:

Fotios PANETSOS

Supervisor:

Prof. Kostas J.
KYRIAKOPOULOS

Examination Committee:

Prof. K. J. Kyriakopoulos

Prof. E. Papadopoulos

Assoc. Prof. G. C. Karras

Prof. I. Antoniadis

Assoc. Prof. C. Tzafestas

Assoc. Prof. I. Poulakakis

Assist. Prof. P. Vartholomeos

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Control Systems Laboratory
School of Mechanical Engineering

May 2024



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Μεθοδολογίες Ελέγχου για
Αυτόνομη Δειγματοληψία, Μεταφορά
και Εναπόθεση Φορτίων
Αναρτημένων επί Ρομποτικών
Πολυκοπτέρων

Συγγραφέας:

Φώτιος ΠΑΝΕΤΣΟΣ

Επιβλέπων:

Καθ. Κώστας ΚΥΡΙΑΚΟΠΟΥΛΟΣ

Η διατριβή υποβάλλεται εκπληρώνοντας τις απαιτήσεις για την απονομή
του Διδακτορικού Διπλώματος

σύμφωνα με το

Εργαστήριο Αυτομάτου Ελέγχου
Σχολή Μηχανολόγων Μηχανικών

Μάιος 2024

NATIONAL TECHNICAL UNIVERSITY OF ATHENS

Abstract

School of Mechanical Engineering

Doctor of Philosophy

**Motion control strategies for the autonomous sampling, transportation
and release of loads suspended to multicopters**

by Fotios PANETSOS

The field of multirotor Unmanned Aerial Vehicles (UAVs) has risen in popularity throughout the last years, since they are recognized as efficient robotic platforms for a variety of applications. More recently, the incorporation of loads onto multirotors has expanded the boundaries of UAVs, enhancing their capabilities and enabling their use in a wider range of tasks. Particularly the suspension of loads through a cable has captivated the interest of the research community as it is a ready-to-use solution, as opposed to the rigid attachment of loads, and alleviates the need for complicated designs. However, the challenges arising from the existence of the cable impede the autonomous deployment of multirotors with cable-suspended loads in real-world environments.

In this dissertation, we aim to bridge this gap by equipping multirotors with the necessary autonomy, thereby enabling their safe operation within the context of 3 distinct applications: (a) swing-free transportation of the cable-suspended load, (b) tracking of a moving ground vehicle with the goal of releasing the load towards the target, and (c) water sampling from aquatic environments. More precisely, we design innovative state estimation techniques which leverage the measurements provided by onboard sensors and reliably estimate the swinging motion and the tension of the cable. In addition, we develop robust control schemes that handle the complicated coupled dynamics of the system and fulfill the objectives of each application. Finally, we demonstrate the efficacy of the proposed methodologies in outdoor settings and practical scenarios. Consequently, we believe that our work can contribute to the ongoing research concerning multirotors with cable-suspended loads, thus broadening their applicability in real-world environments.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Περίληψη

Σχολή Μηχανολόγων Μηχανικών

Διδακτορική Διατριβή

Μεθοδολογίες Ελέγχου για Αυτόνομη Δειγματοληψία, Μεταφορά και Εναπόθεση Φορτίων Αναρτημένων επί Ρομποτικών Πολυκοπτέρων

Φώτιος ΠΑΝΕΤΣΟΣ

Το πεδίο των μη επανδρωμένων εναέριων οχημάτων (UAVs) έχει αποκτήσει μεγαλύτερη απήχηση τα τελευταία χρόνια, καθώς αναγνωρίζονται ως αποτελεσματικές ρομποτικές πλατφόρμες για ποικίλες εφαρμογές. Πιο πρόσφατα, η ενσωμάτωση φορτίων στα πολυκόπτερα έχει επεκτείνει τα όρια των UAVs, ενισχύοντας τις δυνατότητές τους και επιτρέποντας τη χρήση τους σε ένα ευρύτερο φάσμα εφαρμογών. Ιδιαίτερα η ανάρτηση φορτίων μέσω σχοινιού έχει μαγνητίσει το ενδιαφέρον της ερευνητικής κοινότητας καθώς είναι μια έτοιμη προς χρήση λύση, σε αντίθεση με την σταθερή προσάρτηση των φορτίων, και μετριάζει την ανάγκη για πολύπλοκους σχεδιασμούς. Ωστόσο, οι προκλήσεις που προκύπτουν από την ύπαρξη του σχοινιού εμποδίζουν την αυτόνομη ανάπτυξη των πολυκοπτέρων με αναρτημένα μέσω σχοινιού φορτία σε πραγματικά περιβάλλοντα.

Σε αυτή τη διατριβή, στοχεύουμε να γεφυρώσουμε αυτό το χάσμα εξοπλίζοντας τα πολυκόπτερα με την απαραίτητη αυτονομία, επιτρέποντας έτσι την ασφαλή λειτουργία τους στο πλαίσιο 3 διαφορετικών εφαρμογών: (α) μεταφορά του αναρτημένου μέσω σχοινιού φορτίου χωρίς ταλάντωση του σχοινιού, (β) παρακολούθηση ενός κινούμενου επίγειου οχήματος με σκοπό την εναπόθεση του φορτίου στο στόχο και (γ) δειγματοληψία νερού από υδάτινα περιβάλλοντα. Πιο συγκεκριμένα, σχεδιάζουμε καινοτόμες τεχνικές εκτίμησης κατάστασης που αξιοποιούν τις μετρήσεις που παρέχονται από αισθητήρες, ενσωματωμένους στο UAV, και εκτιμούν αξιόπιστα την ταλαντωτική κίνηση και την τάση του σχοινιού. Επιπλέον, αναπτύσσουμε εύρωστα σχήματα ελέγχου που αντιμετωπίζουν την περίπλοκη δυναμική του συστήματος και εκπληρώνουν τους στόχους κάθε εφαρμογής. Τέλος, επιδεικνύουμε την αποτελεσματικότητα των προτεινόμενων μεθοδολογιών σε εξωτερικούς χώρους και πρακτικά σενάρια. Ως εκ τούτου, πιστεύουμε ότι η εργασία μας μπορεί να συμβάλει στην τρέχουσα έρευνα πάνω σε πολυκόπτερα με αναρτημένα μέσω σχοινιού φορτία, διευρύνοντας έτσι την εφαρμοσιμότητά τους σε πραγματικά περιβάλλοντα.

Acknowledgements

This Ph.D. Dissertation could not have been completed without the support of a lot of people.

Foremost, I am immensely grateful to my PhD advisor Prof. Kostas J. Kyriakopoulos for believing in me and providing me with the chance to delve into a demanding yet interesting topic. His guidance has been invaluable and I greatly appreciate the freedom he gave me to explore my own research paths.

I am also grateful to my committee member Prof. Evangelos Papadopoulos. Additionally, I would like to thank Assoc. Prof. George C. Karras for his priceless contributions. Besides serving as a committee member, he played a crucial role in keeping me on track through our discussions and in shaping my mindset to overcome the obstacles that I encountered.

I would also like to thank my friends and colleagues from the Control Systems Laboratory, Panagiotis Rousseas and Sotiris Aspragkathos, who filled each day of this journey with funny moments, creating memories that I will never forget. They not only made my daily life easier but also provided practical assistance in conducting experiments.

I am particularly thankful to the other members of the Control Systems Lab, Michalis Drossakis and Assoc. Prof. Charalambos Bechlioulis, who were always there to offer their help whenever it was needed.

Needless to say, I am especially grateful to my family for their support and their belief in my abilities, and to my close friends, who always found the time to encourage me and listen to my anxieties.

Contents

Abstract	iii
Περίληψη	v
Acknowledgements	vii
1 Introduction	1
1.1 Motivation	1
1.2 State-of-the-Art and Related Work	3
1.3 Outline	11
2 Preliminaries	15
2.1 Introduction	15
2.2 Simulation Environment	15
2.3 Low-Level Control	17
2.4 Notation	18
2.5 Equations of Motion	18
2.5.1 Equations of Motion during Load Transportation	18
2.5.2 Equations of Motion during Water Sampling	21
3 Aerial Transportation of Cable-Suspended Loads	23
3.1 Introduction	23
3.2 Problem Formulation	24
3.3 Estimation of the Cable's State	25
3.3.1 Frame-based Estimation of the Cable's State	25
3.3.2 Event-based Estimation of the Cable's State	28
3.4 Deep Reinforcement Learning Control	36
3.4.1 Reinforcement Learning Background	37
3.4.2 Structure of Networks	37
3.4.3 Reward Function	38
3.4.4 Training Procedure	40
3.5 Robust Deep Reinforcement Learning Control	41
3.5.1 Domain Randomization	42
3.5.2 Training Procedure	42
3.6 Nonlinear Model Predictive Control	43

3.6.1	Nominal NMPC	44
3.6.2	Augmented Dynamics	45
3.6.3	Gaussian Process Regression	46
3.6.4	GP-based NMPC	50
3.7	Experimental Results	51
3.7.1	Experimental Setup	51
3.7.2	Deep RL Control Experiments	52
3.7.3	Robust Deep RL Control Experiments	54
3.7.4	Nominal NMPC Experiments	57
3.7.5	GP-based NMPC Experiments	61
4	Tracking and Release of a Suspended Load to a Ground Target	67
4.1	Introduction	67
4.2	Problem Formulation	68
4.3	Estimation of the Target State	69
4.3.1	Detection of the Target	69
4.3.2	Target Motion Prediction	70
4.4	Nonlinear Model Predictive Control for Target Tracking	72
4.5	Release Condition	73
4.6	Experimental Results	74
4.6.1	Experimental Setup	74
4.6.2	Real-World Experiments	74
5	Water Sampling from Aquatic Environments	77
5.1	Introduction	77
5.2	Problem Formulation	78
5.3	Disturbance Estimation	79
5.3.1	Sensor-based Disturbance Estimation	79
5.3.2	Gaussian Process Regression	81
5.3.3	Extended Kalman Filter	81
5.4	Position Control	82
5.4.1	Geometric Control	82
5.4.2	Nonlinear Model Predictive Control	82
5.5	Experimental Results	83
5.5.1	Experimental Setup	83
5.5.2	Geometric Control with Gaussian Processes	84
5.5.3	NMPC with EKF	87
5.5.4	Autonomous Water Sampling Mission	91
6	Conclusions and Future Work	93
	Appendices	97
A	List of Publications	99

Bibliography	103
---------------------	------------

Extended Greek Summary	115
-------------------------------	------------

List of Figures

1.1	(a) Aerial transportation of cable-suspended loads. (b) Tracking of a moving ground vehicle and release of the load towards the target. (c) Water sampling in aquatic environments.	4
2.1	The quadrotor with the cable-suspended load in the Gazebo simulation environment.	16
2.2	Capture of a Gazebo environment which simulates the water sampling procedure.	16
2.3	The octorotor vehicle utilized throughout the real-world experiments.	17
2.4	The cascaded PID control architecture of Ardupilot.	18
2.5	(a) The octorotor UAV with the cable-suspended load. The world \mathbf{W} and body-fixed \mathbf{B} coordinate systems, the positions \mathbf{p} and \mathbf{p}_L of the UAV and the load respectively, and the cable unit vector \mathbf{n} are also depicted. (b) The cable angles $\boldsymbol{\eta}_L = [\phi_L \ \theta_L]^T$	19
2.6	The octorotor with the cable-suspended sampling mechanism during the sampling procedure in a river.	22
3.1	The octorotor with the cable-suspended load deployed in the field. . .	24
3.2	(a) The input RGB image obtained by the downward-looking frame-based camera. (b) The output segmented image produced by the CNN. (c) The RGB image combined with the output of CNN. The red pixels correspond to the pixels which the CNN identifies as cable. The green cross represents the edge of the cable, i.e., the load. The convex hull of the CNN output is also illustrated.	26
3.3	The octorotor UAV with the onboard computer and the DVS camera (DAVIS 346). The body-fixed \mathbf{B} and the camera \mathbf{C} frames are also depicted. The red, green, blue colored arrows correspond to the x,y,z axes of each frame.	28
3.4	The grayscale images captured by the DVS in an (a) indoor and (c) outdoor environment at the current time instant t_0 . A top-down view (u - v plane) of the corresponding “event” point clouds \mathcal{P} for $T = 25$ ms in the (b) indoor and (d) outdoor environment. The purple and red colored points of \mathcal{P} correspond to events with polarities $p = 1$ and $p = 0$ respectively.	30

3.5	(a), (b), (c) The processing algorithm applied to the “event” point cloud \mathcal{P} of Figure 3.4d: (a) A top-down view (u - v plane) of the point cloud \mathcal{P}' generated after the downsampling of \mathcal{P} . The purple and red colored centroids of \mathcal{P}' correspond to centroids with polarities $p = 1$ and $p = 0$ respectively while the remaining centroids have polarities $p' \in (0, 1)$. (b) A top-down view of the point cloud \mathcal{P}'' after the filtering of \mathcal{P}' based on the polarity of each centroid. (c) A top-down view of the cable cluster \mathcal{C} and the respective blue colored line fitted to \mathcal{C} . (d) The Bézier curve (orange color) which approximates the measurements of the lower edge of the cable (yellow color).	32
3.6	The current “event” point cloud, obtained by the DVS, and a sequence of cable clusters, extracted according to the proposed method, throughout the last 2 s in an outdoor environment. The u - v axes denote the image plane and the t axis represents the time. The purple and red colored points of the “event” point cloud correspond to events with polarities $p = 1$ and $p = 0$ respectively.	33
3.7	The cable angles, (a) ϕ_L and (b) θ_L , computed according to the event-based algorithm and compared against the accurate measurements provided by the Vicon system during the indoor experiment.	35
3.8	Structure of Neural Networks.	39
3.9	The cumulative reward during the learning procedure.	40
3.10	The cumulative reward and the corresponding moving average throughout the training procedure, including domain randomization.	43
3.11	The position \mathbf{p} of the vehicle compared to the reference 3D waypoints \mathbf{p}_{ref} , while using the trained control policy, and the angle β during the simulation scenario.	53
3.12	The deep RL-based control scheme.	53
3.13	The position \mathbf{p} of the vehicle, compared to the reference 3D waypoints \mathbf{p}_{ref} , while using the trained control policy, and the angle β during the experimental scenario.	54
3.14	The reference positions \mathbf{p}_{ref} and the actual trajectory \mathbf{p} of the UAV in the Gazebo for $m_L = 0.5\text{kg}$ and $l = 1.0\text{m}$	55
3.15	The angle β for various load masses m_L and constant cable length l during the 1 st comparative study in the Gazebo.	55
3.16	Block diagram of the proposed RL-based strategy including domain randomization.	56
3.17	The reference positions \mathbf{p}_{ref} and the actual trajectory \mathbf{p} of the UAV during the real experiment for $m_L = 0.5\text{kg}$	57
3.18	The angle β for various load masses m_L during the comparative study in the real-world environment.	57

3.19	An overview of the deployed nominal NMPC scheme for the swing-free load transportation based on feedback provided by a frame-based camera.	58
3.20	The actual position of the UAV \mathbf{p} and the reference waypoints \mathbf{p}_{ref} during the load transportation experiment inside the NTUA campus. (a) Position x. (b) Position y. (c) Position z. (d) The 3D path of the vehicle compared to the reference 3D waypoints.	59
3.21	The estimates of the cable state, computed by the CNN and the EKF, during the load transportation experiment inside the NTUA campus. (a) Cable angles $\boldsymbol{\eta}_L$. (b) Cable angular velocity $\boldsymbol{\omega}_L$	59
3.22	The actual position of the UAV \mathbf{p} and the reference waypoints \mathbf{p}_{ref} during the load transportation experiment in a beach location. (a) Position x. (b) Position y. (c) Position z. (d) The 3D path of the vehicle compared to the reference 3D waypoints.	60
3.23	The estimates of the cable state, computed by the CNN and the EKF, during the load transportation experiment in a beach location. (a) Cable angles $\boldsymbol{\eta}_L$. (b) Cable angular velocity $\boldsymbol{\omega}_L$	60
3.24	(a) The position \mathbf{p} of the UAV and the reference positions \mathbf{p}_{ref} in the x - y plane, (b) the cable angles $\boldsymbol{\eta}_L$, and (c) velocity $\boldsymbol{\omega}_L$ during the outdoor experiment with the event camera.	62
3.25	The norm of the cable's angle $\ \boldsymbol{\eta}_L\ $ during the comparative study between the event-based method (DVS) and the frame-based solution (ZED).	62
3.26	(a) The error $\ \mathbf{p} - \mathbf{p}_{ref}\ $ between the actual position of the UAV and the reference ones, and (b) the cost function of the NMPC (y -axis displayed in logarithmic scale) during the simulation comparative study.	64
3.27	Overview of the GP-based NMPC architecture including the DVS camera.	64
3.28	(a) The error $\ \mathbf{p} - \mathbf{p}_{ref}\ $ between the actual position of the UAV and the reference ones (a zoomed-in view near zero is illustrated in the upper right corner), and (b) the cost function of the NMPC (y -axis displayed in logarithmic scale) during the comparative study in a outdoor environment.	65
4.1	The multirotor with the cable-suspended load and the UGV deployed in the field.	68
4.2	The proposed control scheme for tracking the moving ground vehicle.	69
4.3	Detection of the target. The green cross corresponds to the center of the bounding box.	70

4.4	The red arrows at the upper part of the figure illustrate the trajectory of the UAV while the red and green colored lines at the lower part represent the past and future trajectory of the target respectively, as approximated by the Bézier curve. The ballistic trajectory of the load, assuming that is released, is depicted by the line with the cyan hue. Finally, the blue colored sphere indicates that the release condition is satisfied.	73
4.5	The position \mathbf{p} and the velocity \mathbf{v} of the UAV during the first experiment compared to the UGV motion, i.e., \mathbf{p}_T and \mathbf{v}_T , as estimated by the Bézier regression. (a) Position x. (b) Position y. (c) Velocity x. (d) Velocity y. (e) The cable angles $\boldsymbol{\eta}_L = [\phi_L \ \theta_L]^T$	75
4.6	The successful release of the load towards the target during the second experiment. (a) The position of the load $\mathbf{p}_L(t_h)$, assuming that is released, compared to the position of the target $\mathbf{p}_T(t_h)$ in the x-y plane. The green circle indicates that the release condition (Equation (4.8)) is satisfied. (b) The gripper opens. (c) The load follows a ballistic trajectory. (d) The load is placed inside the box.	76
5.1	Water Sampling with an octorotor in a river.	78
5.2	Dense flow fields computed by the DIS algorithm. The green arrows illustrate the direction of the flow while the length of the arrow indicates the magnitude of the displacement.	80
5.3	The position of the additional sensors required during the water sampling procedure.	84
5.4	The geometric controller of Section 5.4.1 augmented with the predictions of the cable tension according to Section 5.3.2.	85
5.5	The position error $\mathbf{p} - \mathbf{p}_{ref}$ and the estimated disturbances \mathbf{T} during the simulation scenario.	86
5.6	The position error $\mathbf{p} - \mathbf{p}_{ref}$ and the estimated disturbances \mathbf{T} during the experimental scenario in a river.	87
5.7	An overview of the deployed NMPC scheme (Section 5.4.2), which incorporates the estimates of the cable tension according to Section 5.3.3 and achieves the precise stabilization of the vehicle during water sampling.	88
5.8	The actual position of the UAV \mathbf{p} compared to the reference sampling location \mathbf{p}_{ref} and the cable tension estimated by the additional sensors and the Extended Kalman Filter during the stabilization experiment in a beach location. (a) Position x. (b) Position y. (c) Position z. (d) Estimated cable tension.	89

5.9	The actual position of the UAV \mathbf{p} compared to the reference sampling location \mathbf{p}_{ref} and the cable tension estimated by the additional sensors and the Extended Kalman Filter during the stabilization experiment in the Ladonas river. (a) Position x. (b) Position y. (c) Position z. (d) Estimated cable tension.	90
5.10	Autonomous water sampling mission in a beach location. (a) The mission commanded via the Ground Control Station. (b) The 3D path of the vehicle during the mission.	92

List of Abbreviations

UAV	U n m anned A erial V ehicle
DoF	D egree of F reedom
RL	R einforcement L earning
DDPG	D eep D eterministic P olicy G radient
TD3	T win D elayed DDPG
RGB	R ed G reen B lue
MPC	M odel P redictive C ontrol(ler)
NMPC	N onlinear M odel P redictive C ontrol(ler)
GP	G aussian P rocess
KCF	K ernelized C orrelation F ilter
EKF	E xtended K alman F ilter
UKF	U nscented K alman F ilter
DVS	D ynamic V ision S ensor
NN	N eural N etwork
CNN	C onvolutional N eural N etwork
OCP	O ptimal C ontrol P roblem
CoM	C enter of M ass
ROS	R obot O perating S ystem
PID	P roportional I ntegral D erivative
PCL	P oint C loud L ibrary
UGV	U n m anned G round V ehicle
DIS	D ense I nverse S earch

Chapter 1

Introduction

1.1 Motivation

The field of multirotor Unmanned Aerial Vehicles (UAVs) has seen widespread development throughout the last decade due to their versatile nature, their low cost, and their applicability to a variety of tasks such as infrastructure inspection, environmental monitoring, search and rescue missions, surveying and mapping, and agriculture. More recently, the integration of loads onto aerial platforms has broadened the limits of UAVs, extended their capabilities, and expanded their utilization to additional scenarios. For instance, UAVs may significantly contribute to search and rescue missions by delivering necessary supplies, medicine, equipment or other goods. Additionally, multirotors can potentially be deployed in case of environmental disasters where the access of first responders to the field is hazardous or even infeasible, e.g., water sampling in contaminated aquatic areas.

According to the existing approaches, UAVs with loads may be classified into two categories based on the attachment of the load to the vehicle's main body. Specifically, loads are commonly either rigidly attached to the vehicle's body or suspended through a cable. In the former case, rigorous mechanical design, including considerations such as mounting points tailored to the structure of each multirotor, is a prerequisite for the successful integration of the load. Moreover, the release of the load entails the landing of the vehicle, which may be infeasible in emergency situations. Contrastingly, in the latter case, the usage of a cable is a ready-to-use and non-complex solution and, hence, has captivated the interest of the research community.

However, the autonomous deployment of multirotors with cable-suspended loads in real-world environments is quite challenging. From a control point of view, several issues arise from the underactuated nature of the system dynamics, considering the increased number of degrees of freedom (DoFs). More precisely, during the aerial transportation of suspended loads, the arbitrary motion of the vehicle may excite the swinging motion of the cable owing to the complicated coupled dynamics that govern the whole system. Substantial angular displacements of the cable can result in unpredictable scenarios, posing risks to both the vehicle and the load's safety, e.g.,

the cable may dangerously approach the arms and the motors of the UAV. To this end, the need arises towards the design of robust control schemes that minimize the oscillations of the cable and ensure the secure transportation of the load towards reference positions.

Various controllers exist in the literature that aim to cope with the cable angular motion [92]; however, they are typically validated in indoor environments, thus neglecting the presence of external disturbances. Furthermore, conducting experiments in indoor conditions facilitates the estimation of internal state variables, e.g., by using motion capture systems. In contrast, when multirotors with cable-suspended loads are deployed in outdoor environments and real-world conditions, estimating the cable's state based on onboard sensors remains a challenging and unresolved issue. This is crucial as the majority of controllers require reliable and real-time state feedback. To this end, multiple sensor options can be considered towards estimating the cable angular motion in real-world environments; nevertheless, the most straightforward choice is exploiting a camera, since almost every UAV is equipped with a camera mounting and, thus, complex mechanical interventions on the platform, e.g., addition of encoders [7], [73], are avoided. Hence, appropriately designed vision-based techniques, adapted to the specific task of transporting loads, are required in order to provide robust feedback of the cable's state.

Furthermore, the aerial transportation of loads can be expanded in order to consider the tracking of arbitrarily moving targets, with the ultimate aim of releasing the load towards them. This not only broadens the applicability of UAVs but also introduces them to new operational scenarios. Vision-based target tracking with multirotors has been extensively treated in the literature and constitutes a well-studied problem [9], [10], [29], [36]; however, without considering the existence of a cable-suspended load. The incorporation of a load and, consequently, its release towards the target imply various modifications to existing frameworks. Even for a highly experienced drone pilot, achieving the successful release of the load can be particularly challenging owing to the oscillations of the cable and the random motion of the target. Towards this direction, a control scheme, that handles both the predicted trajectory of the target and the cable angular motion, is indispensable so as to fulfill this task.

Moreover, UAVs have also proven to be a promising robotic platform for collecting samples from aquatic environments during a water contamination incident. The deployment of a UAV, instead of manually collecting samples, reduces the exposure of responders to danger, accelerates the disaster response, and, thus, contributes significantly to the emergency management. Two approaches are commonly deployed in order to conduct water sampling with a UAV: (i) integration of a floating base [1], [43], [58], customized for a specific vehicle, which allows the UAV to float and land on the water surface, and (ii) suspension of the sampling mechanism through a cable.

Evidently, the latter solution offers seamless integration in terms of mechanical complexity and preserves the UAV's versatility across various tasks beyond water-related applications. However, when the UAV with the suspended sampling mechanism operates in a fast-flowing aquatic environment, e.g., a river, considerable underwater disturbances act on the device and indirectly on the vehicle through the rope. Hence, to guarantee the secure execution of the sampling procedure, it is essential to employ an appropriate sensor fusion technique that estimates the cable tension using onboard sensors, along with a control strategy that achieves disturbance rejection and precise stabilization of the vehicle.

Briefly, UAVs with cable-suspended loads can be exploited for a wide range of applications and, thus, offer solutions to real-world problems. However, the challenges, stemming from the presence of the suspended load, hinder their widespread utilization. To this end, our research endeavors to bridge this gap by supporting the platform with the required software and hardware and, thus, enable the autonomous operation of UAVs within the context of three specific tasks (Figure 1.1): (a) swing-free and safe transportation of suspended loads, (b) tracking of a moving ground target with the goal of releasing the load towards the former, and (c) water sampling from aquatic environments. Our approach involves refining existing methodologies and developing novel state estimation techniques and controllers which address the challenges posed by the cable while simultaneously meeting the objectives of each application. The proposed methodologies are exhaustively validated in outdoor conditions to demonstrate their effectiveness and potential in practical scenarios. Through this dissertation, we aim to contribute to the field of multirotors with cable-suspended loads, thereby expanding their utilization in real-world environments.

1.2 State-of-the-Art and Related Work

In the existing literature, there are various approaches which aim to control the swinging motion of the cable during the transportation of the suspended load, but may be distinguished into two main categories: generation of swing-free or aggressive trajectories, according to the requirements of the studied task, and direct control of the load state [92].

In [64], [66], dynamic programming was applied, so as to compute optimal swing-free trajectories. In order to track the desired trajectories, a feedback linearization controller was employed combined with an adaptive part which accounts for changes in the center of gravity [65]. The authors experimentally verified the dynamic programming part while using a simpler controller for trajectory tracking. However, model inaccuracies led the same authors to a Reinforcement Learning (RL) approach [16]. By implementing the approximate value iteration algorithm, they estimated the value function based on a reward which penalizes the distance from the desired waypoint and the swinging motion of the vehicle. During the training procedure, a

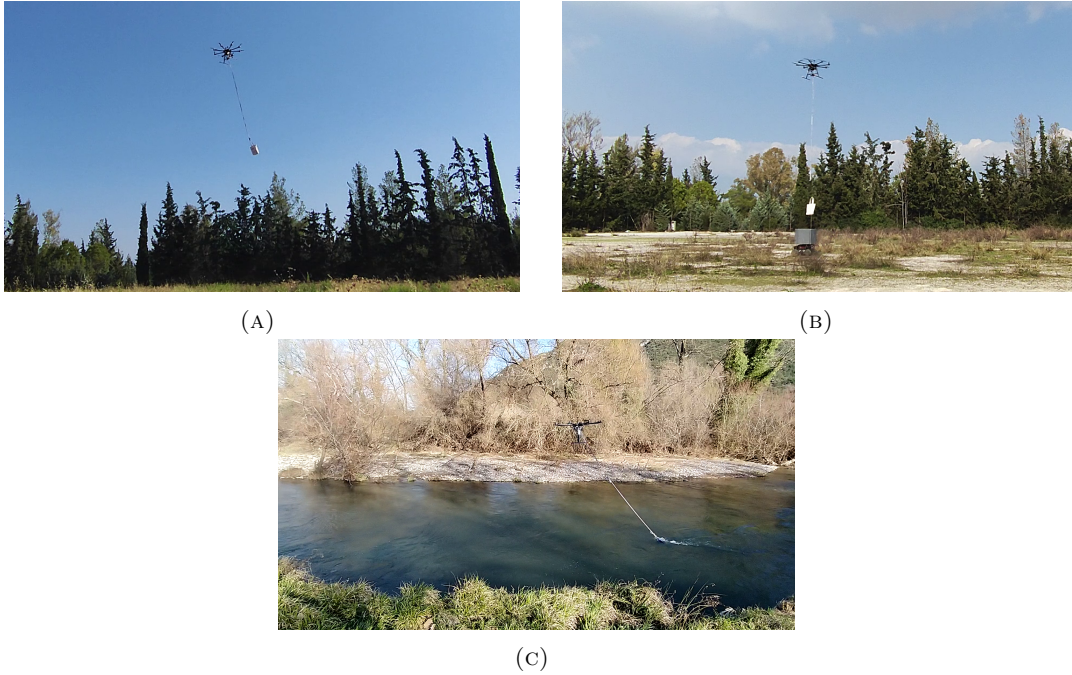


FIGURE 1.1: (a) Aerial transportation of cable-suspended loads. (b) Tracking of a moving ground vehicle and release of the load towards the target. (c) Water sampling in aquatic environments.

simplified model of the quadrotor-load system was utilized in order to obtain the state transition function. Once the value function was estimated, a greedy policy produced linear accelerations which maximize the approximated value function. However, the policy relied on the aforementioned simplified model of the system and, additionally, actions were sampled from a discrete space, which typically results in action oscillations and, hence, non-smooth control.

In [82], a controller was presented which directly controls the position of the load, considering not only minimum swinging motion but also aggressive maneuvers, thereby exploiting the entire range of the cable's motion. More precisely, two dynamic models of the whole system were derived depending on whether the cable is taut or not. The system was proven to be hybrid differentially flat and this property was exploited to generate trajectories. Moreover, a geometric controller was designed which tracks one of the following states: vehicle attitude, load attitude or load position [81]. The performance of the controller was validated experimentally for the planar case only. In a later work [86], the efficiency of the aforementioned geometric controller was demonstrated experimentally in the full three-dimensional workspace and at significant swing angles, while using an on-board camera and an Extended Kalman Filter in order to estimate the state of the load. A similar approach was followed in [25], where, additionally, the cable was modeled as a serially connected link in order to take into account its flexibility. The stabilization of the links was achieved experimentally through nonlinear geometric control. The hybrid dynamic system, including the case when the cable is not taut, and the differential flatness property of

the system were also exploited in [85], where appropriate trajectories were generated in the flat space of the system using Mixed Integer Quadratic Programming. These trajectories guaranteed waypoint tracking, obstacle avoidance and switching between the non-zero and the zero cable tension subsystem when necessary. This approach was validated experimentally in two tasks, pick-up and release of the load and maneuvering the vehicle with the load through a window. However, in both cases, the system was approximated as planar.

A different model was used in [18], where the load was considered to be attached to the base of the vehicle via two revolute joints and a prismatic one. The trajectory generation problem was formulated as a Mathematical Program with Complementarity Constraints, where the constraints take into account the limitation of the cable length and impose bounds on the input, the load angle and the distance between the platform and the surrounding obstacles. According to the authors, this approach reduces the computational cost compared to relative works. A variety of tasks, including waypoint navigation, obstacle avoidance and payload throwing, were presented to validate the generated trajectories.

Additional non-linear control techniques are proposed in the literature in order to minimize the swinging motion of the load. In [27], an Interconnection and Damping Assignment-Passivity Based Control (IDA-PBC) methodology was developed for navigating the vehicle with simultaneous minimization of the load oscillations. Experimental results were presented for planar maneuvers, i.e., longitudinal and transverse motions, and, hence, the payload was connected to the quadrotor with an 1 DoF rigid rod. Furthermore, in [96], the backstepping technique was applied to the quadrotor-slung load system in order to address the problem of load trajectory tracking. Swing suppression was also studied in [51]. More precisely, a nonlinear hierarchical control law, which exploits the cascaded property of the whole system, was designed so as to achieve accurate positioning of the quadrotor with simultaneous elimination of the swing angle, as presented in experimental results under perturbations.

The majority of the aforementioned control schemes require robust and high-speed feedback of the complete cable's state. As a result, they were tested in indoor settings with precise state feedback. Despite the significant number of controllers throughout the literature, the estimation of the cable's state remains a challenging and open issue, especially when deploying UAVs with cable-suspended loads in outdoor environments and real-world conditions, where motion capture systems or markers cannot be used. A few studies have attempted to tackle the estimation of the cable's state based on onboard sensors. More specifically, a monocular fish-eye camera and a novel encoder-based device were utilized in [73] and the respective measurements were fused with the aid of a Gaussian fusion-based estimation algorithm in order to obtain an estimate of the payload state. Moreover, in [7], a load transportation device (LTD), composed of a two-axis cardan joint with two magnetic encoders attached to each axis, was designed and integrated into a helicopter. However, the oscillations

of the LTD and the rope severely affected the measurements and, hence, the authors were forced to develop an observer. In [69], the authors aimed to estimate the state of a cable-suspended load by utilizing only common on-board sensors, e.g., IMU, and an Extended Kalman Filter. However, increased estimation errors were observed in the experimental results, compared to the simulation ones, probably due to the assumed model. IMU measurements were also exploited in [45], along with a load cell sensor and a disturbance observer algorithm [46], and, hence, a swing-angle estimation technique was introduced. Knowledge of the plant model of the UAV including the inner attitude and thrust controller is a prerequisite for the implementation of this method.

Returning to the control of the multirotor with the cable-suspended load, deep Reinforcement Learning (RL) emerges as an appropriate tool given the complexity inherent in the coupled dynamics of the platform. More specifically, modern model-free deep RL algorithms, e.g., deep deterministic policy gradient (DDPG) [52], twin delayed deep deterministic policy gradient (TD3) [19], and proximal policy optimization (PPO) [77], mitigate the necessity for exhaustive system identification and have the potential to handle complicated dynamic models and challenging tasks. Hence, various works in the literature have leveraged deep RL algorithms to learn policies capable of controlling multirotors, albeit without incorporating a cable-suspended load. For instance, in [34], the authors trained a neural network policy which directly maps the state of a quadrotor with rotor thrusts. The performance of the policy was experimentally validated in a waypoint tracking scenario while also recovery from manual upside-down throws was demonstrated. In [42], a policy was learned, based on the PPO algorithm, enabling a real quadrotor to land on an inclined surface – a challenging maneuver which results in a non-equilibrium final state, as the attitude of the vehicle must align with the slope of the surface. Finally, in [80], deep RL was exploited for the purposes of autonomous drone racing. Specifically, a policy was trained, using the PPO algorithm and relative gate observations, that can compute near-time-optimal trajectories and adapt to environmental changes. The authors validated the generated trajectory with an actual quadrotor and achieved aggressive flight at considerable speeds.

Despite the remarkable recent advances in deep RL, the transferability of the learned policy from the simulation environment to real robotic platforms remains a matter of concern. Specifically, the training procedure is commonly realized exclusively within simulation environments, thus, ensuring access to an abundance of data. The current architecture of deep RL algorithms precludes training in real-world settings, as the actions produced by the policy, prior to the convergence of the learning procedure, can lead to unpredictable situations, thereby posing the safety of the platform to risk. Additionally, restrictions are imposed due to the time required for the training to converge. Despite the model-free nature of the aforementioned RL algorithms, the performance of the policy heavily relies on the dynamics that govern

the simulator. Consequently, the unavoidable inconsistencies between the simulator and the real-world environment could negatively impact the effectiveness of the policy, when applied to the actual robot. Furthermore, the robustness of the policy to observation noise or different dynamic properties of the system is not guaranteed.

In this regard, domain randomization has been proposed as an effective means for bridging the gap when transferring policies learned in simulation to real robots. By introducing variations and randomness into the simulator, the goal is to learn policies that exhibit enhanced robustness and can generalize better to real-world conditions. More precisely, the randomization of the simulator's properties, e.g., sensor noise and variations in dynamics, during the training procedure, results in policies that can adjust to a diverse range of conditions. Thus, the real-world environment may be perceived as merely another sample of the varied simulator and the policy can be successfully deployed on the real robot.

Various works in the literature have exploited domain randomization in the context of real-world robotic applications. More specifically, in [88], an object detector is trained using low-fidelity rendered images with random camera and object positions, lighting conditions, and non-realistic textures. Afterwards, the detector is successfully transferred to real images featuring high localization accuracy during grasping. In [76], a vision-based collision avoidance policy for indoor flight was learned entirely in simulation using single RGB images and randomized rendered scenes and was successfully deployed in a real flight. Additionally, domain randomization was implemented in a pushing task with a real robotic arm [67] where a variety of dynamic parameters, e.g., link mass, and properties of the object to be pushed, were randomized throughout the training procedure in the simulation in order to achieve a comparable performance during the real experiments. Finally, the applicability of domain randomization is extended to a variety of other scenarios and applications such as dexterous in-hand manipulation [4] and locomotion control of bipedal robots [50].

Considering the drawbacks of deep RL techniques, namely reward engineering, lack of performance guarantees, and the necessity for a realistic simulator, nonlinear Model Predictive Control (NMPC) appears as an appealing alternative approach for controlling the multirotor with the cable-suspended load. NMPC has been proven to be an efficient tool for controlling UAVs in general, owing to the inherent properties stemming from the nature of MPC scheme, i.e., exploitation of the known system dynamics, optimization over a future horizon, and incorporation of input and state constraints. Additionally, recent advances in embedded computers and optimization libraries enable the solution of optimal control problems within a few milliseconds and, hence, the real-time performance of MPC is feasible. Consequently, NMPC has been extensively exploited throughout the literature to effectively control multirotors for a variety of tasks.

More precisely, MPC schemes for multirotors have been proposed in order to address the problem of trajectory generation and tracking. In [62], SLQ-MPC was employed so as to solve a nonlinear optimal control problem for simultaneous trajectory optimization and tracking control. Perception objectives were considered in [14], and, thus, a perception-aware model predictive control (PAMPC) algorithm for quadrotors was developed. The work in [22], incorporated the existence of obstacles into the optimal control problem and presented an NMPC technique that generates safe trajectories. Also, in [78], NMPC was utilized in order to achieve collision-free and autonomous navigation of UAVs in obstructed environments with obstacles of arbitrary non-convex shape.

Moreover, fault-tolerant NMPC strategies have been successfully implemented so as to control UAVs during motor failure. Specifically, in [5], the authors applied NMPC to a hexacopter with three failed rotors and achieved the control of the vehicle, as demonstrated via simulations. In [91], the authors presented an EKF for monitoring the health of each actuator along with an NMPC which considers the full dynamics of a real hexarotor and controls the vehicle under actuator failure. In [61], an NMPC framework for quadrotor fault-tolerant flight was proposed which can effectively stabilize a quadrotor subjected to the complete failure of a single rotor, even if the latter occurs during an agile flight.

The applicability of UAVs is extended to a wide range of other scenarios such as trajectory generation for obstacle avoidance considering the existence of a slung load [79], aerial pick-and-place with an integrated manipulator [21], and aggressive cable-suspended payload trajectory tracking including field of view constraints [48]. Based on the several aforementioned successful attempts to exploit NMPC, it is reasonable to investigate its utilization for controlling the multirotor with the cable-suspended load.

However, model-based controllers, and particularly NMPC schemes, are susceptible to the adopted model and, thus, possible mismatches between the nominal and actual system dynamics may degrade the control performance. This is mainly visible when the UAV with the cable-suspended load operates in outdoor environments, where external disturbances, arising primarily from the prevailing environmental conditions, act on the platform. The aforementioned disturbances, combined with the assumptions throughout the formulation of the nominal dynamics, result in model discrepancies between the adopted model and the real system dynamics, thereby impacting the efficacy of NMPC schemes. Gaussian Processes (GPs) have been proven efficient tools for estimating unknown dynamics and, hence, various works in the literature have exploited GP regression in the context of NMPC schemes in order to bridge the gap between real and nominal robot dynamics and further enhance the control performance. More specifically, in [32], [38], the authors exploited GP regression in order to learn the residual dynamics of a vehicle's nominal model and incorporated the respective predictions into an NMPC scheme so as to improve online the tracking

performance of an autonomous racing car. In [89], the authors extended [32], [38] to three-dimensional GP predictions for a quadrotor in order to learn the aerodynamic effects acting on the platform at high speeds. By utilizing an NMPC, augmented with the approximated residual dynamics, a considerable reduction in trajectory tracking error was observed. Finally, in [94], a path following algorithm was presented, which merges GP-based learning, feedback linearization, and MPC into a single framework and achieved the operation of a ground mobile robot in off-road terrains.

Moving on to discuss the topic of aerial target tracking, various works address the problem of following a moving target with a multicopter using visual feedback. More specifically, in [10], the authors presented a visual scheme for tracking a maneuvering target using a UAV with an onboard gimbal camera. The Kernelized Correlation Filter (KCF) tracker [30] was utilized in order to detect the target while the full state of the target was estimated by the Interacting Multi-Model Extended Kalman Filtering (IMM-EKF) algorithm [36] assuming a Singer model with an adaptive mean. Considering the aforementioned estimates, both the gimbal and the vehicle were controlled in order to achieve the tracking of the moving target. However, the IMM-EKF estimator predicts solely the state of the target at the next time instant while also requiring a target motion model, which is a priori unknown.

In [9], the authors proposed a framework for real-time target tracking in cluttered environments. More precisely, the trajectory of the target, observed with the aid of markers, was approximated and predicted for a short time horizon using polynomial regression. Based on the predicted target trajectory and the detected obstacles, the A^* method was utilized in order to find a grid-based collision-free flight corridor for the UAV. Eventually, a dynamically feasible for the UAV trajectory was generated using quadratic programming. The authors validated their approach experimentally in an indoor environment.

Polynomial regression for predicting the target motion was also considered in [29], where a Bézier curve was utilized in order to describe the target trajectory. Additionally, the authors assumed that the target velocity and acceleration were bounded and, thus, constraints were imposed on the quadratic programming problem. Given the target trajectory, a kinodynamic searching method was implemented so as to find an appropriate flight corridor consisting of obstacle-free grids and, afterwards, an optimal polynomial trajectory was generated within the safe flight corridor. The authors experimentally verified their approach in both indoor and outdoor environments. In their following paper [95], the visibility of the target was also integrated into the trajectory planning method so as to avoid target loss. However, the works mentioned above did not consider the incorporation of a cable-suspended load and, hence, the release of the load towards the target.

As for the task of conducting water sampling using a UAV with a suspended

mechanism, we previously highlighted the need for a disturbance rejection technique which attains the accurate stabilization of the vehicle, especially in fast-floating aquatic environments, despite the cable tension. In general, multirotor platforms are now more commonly deployed in real-world outdoor environments so as to perform a variety of desired tasks. However, due to their interaction with the environment, UAVs are subject to external disturbances, e.g., wind gusts, ground and wall effects, and payload disturbances. Since the majority of applications require accurate position control, i.e., stabilization, path following or trajectory tracking, the rejection of the disturbances during the flight is crucial for the successful completion of the commanded task.

In the existing literature, various disturbance rejection methodologies are demonstrated, albeit not specifically applied to water sampling. These methodologies may be distinguished into two main categories (i) robust or adaptive control strategies with unknown disturbances and (ii) online identification and compensation of the disturbances.

In [2], a robust MPC scheme was designed for robust trajectory tracking under unknown but bounded disturbances. The authors experimentally validated their approach under turbulence effects, slung load disturbances and collisions with the environment. Special attention was given to the computation of the optimization problem and to conservativeness, which is a typical issue in robust control strategies. A different scheme was presented in [68] where a \mathcal{L}_1 adaptive controller was augmented with an iterative learning control (ILC) framework so as to reduce trajectory tracking errors under disturbances. The aforementioned controller was employed to a quadrotor with an added suspended mass and featured an improved performance compared to a non-adaptive PD-ILC approach.

Instead of designing controllers capable of handling unknown disturbances, another approach is to directly observe the disturbance in real-time and incorporate this estimation into a controller. Hence, issues such as conservativeness and adaptation performance are avoided. The authors in [97] proposed a deterministic nonlinear wrench observer, which is capable of approximating the external forces applied to the Center of Gravity (CoG) of the vehicle, and exploited this estimate by utilizing an Interconnection and Damping Assignment Passivity Based Controller (IDA-PBC) in order to control the interaction of the UAV with the environment, as presented in simulation results. The aforementioned observer relies on the dynamic model of the vehicle and assumes that the disturbances are not rapidly varying.

Alternatively, stochastic filters, i.e., Kalman Filters, were implemented in [55], [31] so as to mitigate sensor and process noise. More specifically, the authors designed Unscented and Extended Kalman Filters which estimate the disturbances, modeled as random walks, based on an accurate dynamic model of the vehicle. Since the aforementioned filters require knowledge of the UAV's dynamics, thrust and torque

maps, i.e., relationship between thrust/torques and motor commands, were identified so as to approximate the control input produced by the motors.

Revisiting the problem of water sampling, the sensor-based estimation of the disturbance, i.e., the tension of the cable, is a challenging task. Purely vision-based techniques are unsuitable for this scenario since the mechanism is submerged, rendering its detection extremely demanding, if not entirely unfeasible, and highly contingent on water purity.

1.3 Outline

The organization of this dissertation is as follows.

Chapter 3:

In this chapter, we present appropriate state estimation techniques and controllers that address the transportation of a cable-suspended load with a multicopter in outdoor environments. To guarantee the safe transportation of the load, we develop control schemes that handle the complicated coupled dynamics of the platform and minimize the swinging motion of the cable. Additionally, considering the necessity for feedback of the complete cable's state throughout the autonomous operation of the platform, we design techniques that reliably estimate the cable angular motion based on visual information and feed closed-loop controllers.

More precisely, to avoid the integration of multiple sensors, we exploit a single downward-looking camera for estimating the cable's state. Almost every UAV includes a camera mount and, hence, its integration is a plug-and-play solution. Towards this direction, we present 2 vision-based methodologies for estimating the cable angular motion with the aid of either a conventional RGB camera or a dynamic vision sensor (DVS).

In the former case, a trained Convolutional Neural Network (CNN) is applied to the RGB image obtained by the camera and, thus, the robust detection of the cable is achieved despite the challenging lighting conditions, commonly observed in outdoor environments. The complete cable's state is estimated by a Kalman Filter, formulated for the nonlinear dynamics of the load, which fuses the output of the CNN and the measurements of a load cell.

In the latter case, we design a novel event processing method, tailored to the specific application, that leverages the asynchronous nature and the advantageous properties of DVS and considerably accelerates the estimation of the cable's state. To the best of our knowledge, the concept of utilizing event cameras for aerial transportation is original. More specifically, the events, acquired from the DVS, are represented in the form of a point cloud, where both spatiotemporal information and polarity are stored. Afterward, a computationally inexpensive method is developed in order to process the input point cloud and to robustly distinguish the events that correspond to the

cable in a quasi-continuous manner. The extracted information is approximated by a Bézier curve, solving a constrained optimization problem, and, eventually, both the cable angles and angular velocity are obtained within a much shorter time period compared to the aforementioned frame-based algorithm. We remark that high-speed feedback can enhance the performance of closed-loop control schemes.

To attain the transportation of the load towards reference positions with simultaneous minimization of the cable angular motion, we develop both model-free and model-based control schemes. Particularly, building upon the model-free nature of deep Reinforcement Learning (RL) algorithms, we design a data-driven control policy without any assumption about the model of the system. This is accomplished through the utilization of Twin Delayed Deep Deterministic Policy Gradient (TD3), a deep RL algorithm for continuous action spaces. The RL framework is reformulated according to the requirements of the application, to train an appropriate neural network policy that fulfills the objectives of the task. Afterward, we adopt domain randomization during the training procedure, realized in a simulation environment, in order to learn a policy which is more robust to varying properties of the system, e.g., load mass, cable length, and observation noise, as well as system dynamics that differ from the ones encountered during the training. By employing this technique the gap between simulation and real-world conditions is bridged and the successful transfer of the policy, trained exclusively in simulation, to the real vehicle is achieved.

Furthermore, we develop a nonlinear Model Predictive Control (NMPC) scheme, based on the coupled dynamics of the platform, that produces optimal control inputs and, thus, attains the swing-free transportation of the load. Subsequently, considering that the performance of NMPC highly depends on the adopted model, we correct the nominal system dynamics using a non-parametric learning-based approach. More precisely, we employ Gaussian Processes (GPs) to learn online the model discrepancies between the actual and nominal system dynamics as well as the disturbances, stemming from the prevailing environmental conditions, that act on the platform in outdoor environments. Considering the dimensionality of the addressed task, sparse GP regression is exploited in order to reduce the computational demands of GPs. We also introduce a weighted version of sparse GP regression with the ultimate goal of assigning importance to more recent measurements and, thus, accomplishing both real-time and rapid estimation of the time-varying disturbances encountered in outdoor environments. Ultimately, the learned dynamics are appropriately integrated into a GP-based NMPC scheme so as to improve the control performance compared to the nominal case.

Chapter 4:

In this chapter, we present an NMPC scheme for tracking a ground target using a multirotor with a cable-suspended load and a downward-looking camera. The NMPC framework relies on the dynamic model of the UAV with the suspended load and, therefore, an estimate of the cable angular motion is obtained using the frame-based

methodology outlined in Chapter 3. Regarding the detection of the target, a CNN, along with a Kernelized Correlation Filter (KCF) tracker, is employed. This hybrid approach ensures reliable target identification even in scenarios where the target may be visually occluded by the cable. Since the NMPC entails propagating the system's behavior over a predicted horizon, the past observations of the target are approximated by a Bézier curve. Specifically, a constrained Bézier regression problem is appropriately formulated to predict the trajectory of the arbitrarily moving target. The future target motion is directly incorporated into the predicted horizon of the NMPC scheme, which computes optimal inputs to the inner attitude control loop of the autopilot, thus achieving the following of the aggressively moving target. The ultimate goal of the proposed framework is to release the suspended load to the ground target and, consequently, a condition is checked at each time instant. Once the predicted ballistic trajectory of the load intersects with the future target motion, the condition is satisfied, triggering the opening of a gripper, located at the lower edge of the cable.

Chapter 5:

In this chapter, we address the problem of conducting water sampling using a multicopter with a cable-suspended sampling mechanism. We present sensor fusion techniques for identifying the external forces applied to the vehicle during the sampling procedure and we develop appropriate control strategies that reject the disturbances and accurately stabilize the UAV. This is mainly applicable in fast-floating aquatic environments, such as rivers, where considerable underwater drag forces, owing to the high water velocity, act on the mechanism and are transferred to the vehicle through the cable.

More precisely, as the mechanism is submerged and solely vision-based techniques cannot be applied, we integrate low-cost sensors, namely an ultrasonic sensor, a load cell, and a depth sensor, besides the downward-looking camera, in order to estimate the cable tension. Since the corresponding measurements are subject to noise, we design sensor fusion techniques that alleviate the noise effects. Specifically, GPs are exploited to learn online and approximate the disturbances, transferred through the cable, in a non-parametric manner. Additionally, a Kalman Filter is developed as a more computationally inexpensive approach for fusing the measurements provided by the onboard sensors. An Extended Kalman Filter (EKF) is employed owing to the nonlinear measurement model.

To attain the precise position control of the vehicle above the sampling location, we employ a geometric controller which incorporates the estimates of the cable tension. Furthermore, we develop an NMPC scheme and impose state and input constraints on the respective Optimal Control Problem (OCP) so as to avoid aggressive maneuvers of the vehicle above the water surface and, thereby, increase the safety level of the sampling procedure.

Eventually, we present a fully autonomous framework which unifies the methodologies of Chapters 3 and 5 and addresses all the phases of a water sampling mission, namely the swing-free transportation of the platform to the sampling position, the stabilization of the UAV during the sampling procedure, and the transportation of the platform to a final position for sample collection.

Chapter 2

Preliminaries

2.1 Introduction

This chapter establishes the necessary preliminary background about UAVs with cable-suspended loads, serving as the basis for the subsequent methodologies. We introduce the UAV as well as the required variables and fundamental equations that govern the motion of the platform in the context of the studied applications, i.e., the swing-free transportation of the cable-suspended load, the tracking and the release of the load to a ground target, and the water sampling from an aquatic environment. The control algorithms and sensor fusion techniques, presented in the entire dissertation, heavily rely on the aforementioned mathematical formulations.

The outline of this chapter is as follows. Section 2.2 presents the simulation environment where the developed algorithms are initially tested prior to conducting real-world experiments. In Section 2.3, the low-level control, handled by the autopilot, is discussed. Finally, Sections 2.4 and 2.5 introduce the notation and the dynamic models of the platform during the addressed tasks correspondingly.

2.2 Simulation Environment

The majority of the proposed control schemes and state estimation techniques are initially validated in simulation environments before being deployed in real settings. Hence, a considerable amount of time is conserved since conducting real-world experiments highly depends on the external weather conditions. Additionally, the safety of the actual octorotor, utilized throughout the outdoor experiments, is ensured especially during the early stages of software testing.

More precisely, environments were set up in the high-fidelity Gazebo robotics simulator [41], based on the needs of the studied application. During the simulation experiments, a quadrotor vehicle with a cable-suspended load, depicted in Figure 2.1, is considered. The quadrotor is equipped with the Ardupilot firmware [26] and, hence, Software in the Loop (SITL) simulations are feasible. The control of the vehicle is achieved using the Robot Operating System (ROS) [71] and, particularly,

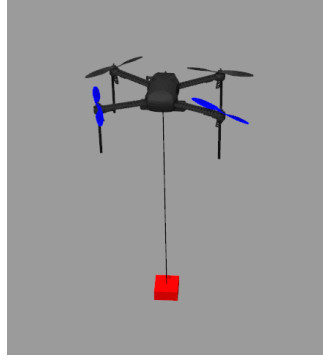


FIGURE 2.1: The quadrotor with the cable-suspended load in the Gazebo simulation environment.

through the MAVROS node [70] which is an intermediate communication protocol between the flight controller and ROS. It is mentioned that the real octorotor vehicle is also equipped with the Ardupilot firmware and, thus, a seamless integration of the developed control algorithms is assured. Regarding the cable-suspended load, the cable is modeled as a rigid link and its upper edge is attached to the center of mass (CoM) of the quadrotor with the aid of a spherical joint.

Various sensors, e.g., downward-looking camera, load cell, and depth sensor, are incorporated into the quadrotor besides the common navigation ones, e.g., GPS, IMU, and altimeter. The aforementioned sensors are appropriately integrated either into the quadrotor's body or upon the cable according to the requirements of the addressed task. For the simulation of the water sampling procedure, visual effects are also included in the simulation environment, as illustrated in Figure 2.2, while buoyancy and drag forces, related to the water flow, are applied to the submerged load, in order to model the dynamics during the sampling.

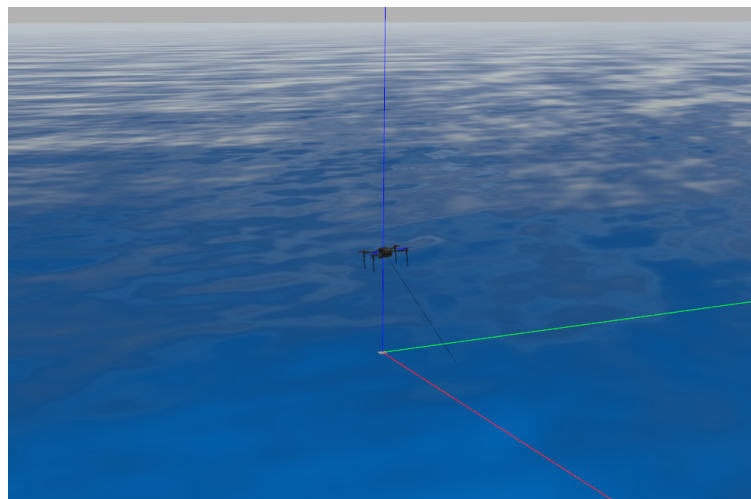


FIGURE 2.2: Capture of a Gazebo environment which simulates the water sampling procedure.

2.3 Low-Level Control

As aforementioned, both the quadrotor and the real octorotor of Figure 2.3 are equipped with the Ardupilot firmware, which is capable of providing state feedback and controlling the vehicles based on a cascaded PID control architecture.



FIGURE 2.3: The octorotor vehicle utilized throughout the real-world experiments.

More precisely, the estimation of the internal state variables of the UAV is achieved through an Extended Kalman Filter (EKF), which fuses measurements provided by common navigation sensors, such as IMU, GPS, compass etc. Regarding the control of the vehicle, the cascaded PID control architecture of Figure 2.4 is employed, following standard practice in multirotor autopilots, due to the underactuated nature of the system. Specifically, an outer PID position controller commands an inner attitude one based on the reference position, velocity, and heading of the vehicle. Subsequently, the inner attitude loop converts the required orientation and vertical velocity into appropriate inputs, i.e., Pulse Width Modulation (PWM) values, for each motor of the UAV.

It should be noted that the control schemes, presented in the dissertation, substitute the default PID position controller of the autopilot and command directly the inner attitude one due to the addition of the cable-suspended load, which alters the translational dynamics of the whole system, and the need for controlling the cable's state. Consequently, the inner attitude loop of the autopilot is incorporated into the proposed control schemes while the outer position one is disregarded since it does not consider the presence of a cable-suspended load.

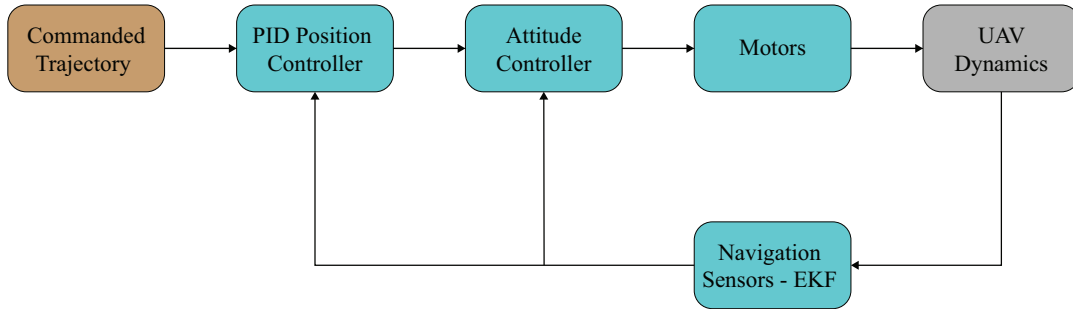


FIGURE 2.4: The cascaded PID control architecture of Ardupilot.

2.4 Notation

The necessary notation, which will be utilized consistently throughout the dissertation, is listed below in order to aid the reader's comprehension of the subsequent concepts:

- Vectors and matrices are generally represented by bold lowercase and uppercase symbols respectively, while non-bold letters denote scalars.
- Vectors are defined with respect to the world frame \mathbf{W} unless a superscript is utilized.
- Subscripts are used in order to distinguish the variables, e.g., the subscript L refers to variables of the cable-suspended load.
- $\mathbf{I}_{n \times n}$ and $\mathbf{0}_{n \times m}$ denote the $n \times n$ identity and $n \times m$ null matrix correspondingly.
- $\text{diag}()$ denotes a diagonal matrix whose non-diagonal elements are zero.
- $\|\mathbf{x}\|_{\mathbf{A}}^2$, where $\mathbf{x} \in \mathbb{R}^n$ defines a vector and $\mathbf{A} \in \mathbb{R}^{n \times n}$ a symmetric positive definite matrix, is equal to $\mathbf{x}^T \mathbf{A} \mathbf{x}$.

2.5 Equations of Motion

In this section, the equations of motion for the multirotor with the cable-suspended load are presented. It should be highlighted that the forces, arising from the water flow and acting on the submerged load during the sampling from aquatic environments, alter the dynamics of the system compared to the transportation of the load and the tracking of the ground target.

2.5.1 Equations of Motion during Load Transportation

Consider the multirotor with the cable-suspended load, and let \mathbf{W} be a world (inertial) coordinate system, i.e., East-North-Up (ENU) frame, and \mathbf{B} a body-fixed one, as depicted in Figure 2.5a. The origins of \mathbf{W} and \mathbf{B} coincide with the position where the multirotor takes off and the center of mass (CoM) of the vehicle correspondingly. Under the assumptions that the cable is taut, has a constant length l , and

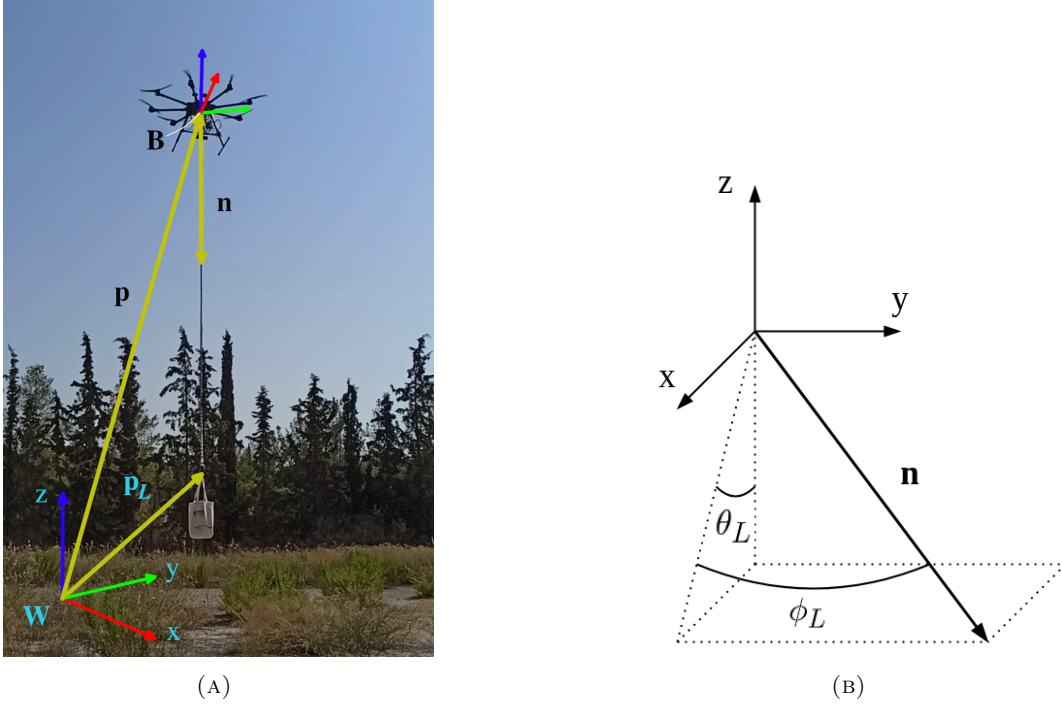


FIGURE 2.5: (a) The octorotor UAV with the cable-suspended load. The world \mathbf{W} and body-fixed \mathbf{B} coordinate systems, the positions \mathbf{p} and \mathbf{p}_L of the UAV and the load respectively, and the cable unit vector \mathbf{n} are also depicted. (b) The cable angles $\boldsymbol{\eta}_L = [\phi_L \ \theta_L]^T$.

its upper edge is exactly attached to the CoM of the vehicle, the cable unit vector $\mathbf{n} = [n_x \ n_y \ n_z]^T \in \mathbb{R}^3$ is related to the positions $\mathbf{p}_L = [x_L \ y_L \ z_L]^T \in \mathbb{R}^3$ and $\mathbf{p} = [x \ y \ z]^T \in \mathbb{R}^3$ of the load and the UAV respectively, according to the equation:

$$\mathbf{n} = \frac{1}{l} (\mathbf{p}_L - \mathbf{p}) \quad (2.1)$$

Since the cable unit vector has 2 degrees of freedom (DoFs), it can also be defined based on the cable angles $\boldsymbol{\eta}_L = [\phi_L \ \theta_L]^T$, as illustrated in Figure 2.5b, where ϕ_L and θ_L denote consecutive rotations around the x and y axes of the world frame \mathbf{W} correspondingly, i.e.:

$$\begin{aligned} \mathbf{n} &= \mathbf{R}_y(\theta_L) \mathbf{R}_x(\phi_L) [0 \ 0 \ -1]^T \\ \Rightarrow \mathbf{n} &= [-s\theta_L c\phi_L \ s\phi_L \ -c\theta_L c\phi_L]^T \end{aligned} \quad (2.2)$$

By differentiating successively Equation (2.2), the following relationships are produced for the cable velocity $\dot{\mathbf{n}}$ and acceleration $\ddot{\mathbf{n}}$:

$$\dot{\mathbf{n}} = \mathbf{J}\boldsymbol{\omega}_L \quad (2.3)$$

$$\ddot{\mathbf{n}} = \mathbf{J}\dot{\boldsymbol{\omega}}_L + \begin{bmatrix} c\theta_L s\phi_L \\ 0 \\ -s\theta_L s\phi_L \end{bmatrix} 2\dot{\theta}_L \dot{\phi}_L + \begin{bmatrix} s\theta_L c\phi_L \|\boldsymbol{\omega}_L\|^2 \\ -s\phi_L \dot{\phi}_L^2 \\ c\theta_L c\phi_L \|\boldsymbol{\omega}_L\|^2 \end{bmatrix} \quad (2.4)$$

where $\boldsymbol{\omega}_L = \dot{\boldsymbol{\eta}}_L$ is the cable angular velocity and

$$\mathbf{J} = \frac{\partial \mathbf{n}}{\partial \boldsymbol{\eta}_L} = \begin{bmatrix} s\theta_L s\phi_L & -c\theta_L c\phi_L \\ c\phi_L & 0 \\ c\theta_L s\phi_L & s\theta_L c\phi_L \end{bmatrix} \quad (2.5)$$

By multiplying Equation (2.4) with the matrix $(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$, the cable angular acceleration $\dot{\boldsymbol{\omega}}_L$ is derived:

$$\dot{\boldsymbol{\omega}}_L = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \ddot{\mathbf{n}} + \mathbf{c}(\boldsymbol{\eta}_L, \boldsymbol{\omega}_L)$$

$$\dot{\boldsymbol{\omega}}_L = \begin{bmatrix} s\theta_L s\phi_L & c\phi_L & c\theta_L s\phi_L \\ -\frac{c\theta_L}{c\phi_L} & 0 & \frac{s\theta_L}{c\phi_L} \end{bmatrix} \ddot{\mathbf{n}} + \begin{bmatrix} -c\phi_L s\phi_L \dot{\theta}_L^2 \\ 2 \tan(\phi_L) \dot{\theta}_L \dot{\phi}_L \end{bmatrix} \quad (2.6)$$

The above equation is valid for $|\phi_L| \neq \pi/2$, which is a reasonable assumption in case of swing-free load transportation. The dependency on the cable acceleration $\ddot{\mathbf{n}}$ can be eliminated in Equation (2.6) by differentiating twice Equation (2.1) and by exploiting the Newton-Euler equations of motion for the load, i.e.:

$$\dot{\mathbf{v}}_L = -g\mathbf{e}_z - \frac{\|\mathbf{T}\|}{m_L} \mathbf{n} \quad (2.7)$$

where \mathbf{v}_L is the load velocity, $g = 9.81 \text{ m/s}^2$ is the gravitational acceleration, \mathbf{e}_z is the z axis of \mathbf{W} , m_L is the load mass, and \mathbf{T} is the cable tension. It is mentioned that other external forces acting on the load, e.g., air drag forces, are neglected. Eventually, considering that $\mathbf{J}^T \mathbf{n} = \mathbf{0}_{2 \times 1}$, the cable angular dynamics can be extracted by rewriting Equation (2.6) as follows:

$$\dot{\boldsymbol{\omega}}_L = \frac{1}{l} (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (-g\mathbf{e}_z - \dot{\mathbf{v}}) + \mathbf{c}(\boldsymbol{\eta}_L, \boldsymbol{\omega}_L) \quad (2.8)$$

with $\mathbf{v} = [v_x \ v_y \ v_z]^T$ denoting the velocity of the UAV.

Regarding the dynamics of the multicopter, the rigid body equations of motion are exploited in order to describe the translational dynamics of the vehicle subject to external disturbances [54]. Concerning the rotational dynamics, the attitude subsystem is controlled by the fast embedded autopilot and a first-order model is utilized so as to represent the inner-loop attitude controller due to the efficiency of the onboard autopilot [39], [63]. Consequently, the following equations describe the behavior of

the multirotor platform:

$$\begin{aligned}
\dot{\mathbf{p}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= -g\mathbf{e}_z + F\mathbf{R}_{WB}\mathbf{e}_z + \frac{\mathbf{f}_{ext}}{m} \\
\dot{\phi} &= \frac{1}{\tau_\phi} (K_\phi\phi_d - \phi) \\
\dot{\theta} &= \frac{1}{\tau_\theta} (K_\theta\theta_d - \theta) \\
\dot{\psi} &= \frac{1}{\tau_\psi} (K_\psi\psi_d - \psi)
\end{aligned} \tag{2.9}$$

where m is the mass of the vehicle, F represents the thrust, normalized with respect to mass, produced by the motors, \mathbf{R}_{WB} denotes the rotation matrix from \mathbf{B} to \mathbf{W} derived by the Euler roll, pitch, yaw angles, i.e., ϕ , θ , ψ respectively and \mathbf{f}_{ext} are the external forces acting on the vehicle. Additionally, the parameters K_ϕ , K_θ , K_ψ and τ_ϕ , τ_θ , τ_ψ are the gains and time constants of the inner attitude controller which is identified according to the attitude response of the octorotor. Finally, the angles ϕ_d , θ_d , ψ_d represent the desired roll, pitch, yaw angles, which are directly sent to the inner attitude controller, while the actual thrust is computed based on the desired vertical velocity v_{z_d} and is scaled so as to consider nonzero roll and pitch angles according to the relationship:

$$F = \frac{g + (K_z v_{z_d} - v_z) / \tau_z}{c\theta c\phi} \tag{2.10}$$

with K_z and τ_z denoting the gain and the time constant of the inner thrust controller.

Finally, the complete dynamic model of the system during the aerial transportation of the load, assuming that the only external disturbance acting on the vehicle is the cable tension and, thus, neglecting the air drag forces, is:

$$f : \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\boldsymbol{\eta}}_L \\ \dot{\boldsymbol{\omega}}_L \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -g\mathbf{e}_z + F\mathbf{R}_{WB}\mathbf{e}_z + \frac{\|\mathbf{T}\|}{m}\mathbf{n} \\ \frac{1}{\tau_\phi} (K_\phi\phi_d - \phi) \\ \frac{1}{\tau_\theta} (K_\theta\theta_d - \theta) \\ \frac{1}{\tau_\psi} (K_\psi\psi_d - \psi) \\ \boldsymbol{\omega}_L \\ \frac{1}{I} (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (-g\mathbf{e}_z - \dot{\mathbf{v}}) + \mathbf{c}(\boldsymbol{\eta}_L, \boldsymbol{\omega}_L) \end{bmatrix} \tag{2.11}$$

2.5.2 Equations of Motion during Water Sampling

Consider the case in which a UAV is conducting water sampling with a suspended sampling mechanism, as depicted in Figure 2.6. During the water sampling procedure, the equations which describe the dynamics of the multirotor, i.e., Equation (2.9), are identical. However, as far as the load is concerned, Equation (2.7) does not hold since additional forces are acting on the submerged mechanism, besides the gravitational



FIGURE 2.6: The octorotor with the cable-suspended sampling mechanism during the sampling procedure in a river.

force and the tension of the cable, due to the interaction of the load with the fluid. More specifically, the Newton-Euler equations for the mechanism during the sampling are:

$$m_L \dot{\mathbf{v}}_L = -m_L g \mathbf{e}_z - \|\mathbf{T}\| \mathbf{n} + \mathbf{f}_b + \mathbf{f}_d \quad (2.12)$$

where:

- $\mathbf{f}_b = \rho V g \mathbf{e}_z$ is the buoyancy force with V denoting the volume of the submerged load and ρ the density of the water and
- $\mathbf{f}_d = \mathbf{D}_v (\mathbf{v}_w - \mathbf{v}_L) + \mathbf{D}_{|v|v} |\mathbf{v}_w - \mathbf{v}_L| (\mathbf{v}_w - \mathbf{v}_L)$ is the drag force induced by the water with $\mathbf{D}_v, \mathbf{D}_{|v|v} \in \mathbb{R}^{3 \times 3}$ the linear and quadratic drag coefficient matrices correspondingly, and $\mathbf{v}_w = [v_{w,x} \ v_{w,y} \ 0]^T$ the water velocity expressed to the world frame \mathbf{W} .

It is evident, from the above equations, that during the sampling procedure some parameters are related to the shape of the mechanism and to the fluid itself, and thus are difficult to be identified.

However, due to the coupled dynamics of the whole system, the underwater disturbances, applied to the submerged mechanism, are transferred to the UAV through the cable. Consequently, solely an estimate of the cable tension is required for the control of the vehicle during the sampling procedure.

Chapter 3

Aerial Transportation of Cable-Suspended Loads

3.1 Introduction

In this chapter, we address the task of transporting a cable-suspended load with a multirotor UAV in outdoor environments. The suspension of loads through cables has the advantage of minimum mechanical intervention in the UAV platform and alleviates the need for complicated designs as opposed to the rigid attachment of loads to UAVs. However, the existence of the cable-suspended load entails control schemes that handle the increased number of degrees of freedom (DoFs) and the swinging motion of the cable and, thus, guarantee the safety of the platform. Additionally, considering that a reliable estimate of the cable's state is a prerequisite for feedback controllers, the need arises towards the development of appropriate techniques for estimating the cable's angular motion based on onboard sensors.

To this end, we present 2 methodologies for estimating the cable's state with the aid of either a standard camera or a dynamic vision sensor (DVS). In the former case, the detection of the cable relies on a Convolutional Neural Network (CNN) and the complete cable's state is obtained by a Kalman Filter. In the latter case, based on a point cloud representation for the incoming event streams, a computationally lightweight event processing method is designed which achieves the fast detection of the cable while the respective measurements are afterwards fitted to a Bézier curve in order to approximate the complete cable's state, within a much shorter time period compared to the frame-based solution.

Regarding the control of the platform, both model-free and model-based approaches are investigated in order to achieve the swing-free transportation of the load. Initially, based on deep Reinforcement Learning (RL), a model-free policy is learned in the simulation environment which accomplishes the addressed task. Afterwards, domain randomization is included into the training procedure so as to improve the transferability of the learned policy to real-world conditions and enhance its robustness to varying properties of the system. Eventually, a model-based approach,

particularly a nonlinear Model Predictive Controller (NMPC), is formulated for the system dynamics. Since the multirotor with the cable-suspended load operates in outdoor environments under the influence of external disturbances, Gaussian Processes (GPs) are exploited in order to approximate online the model errors between the nominal and actual dynamics. Subsequently, the corrected dynamics are fed into a GP-based NMPC scheme which fulfills the swing-free transportation of the load regardless of the existence of disturbances and the conditions, prevailing in the outdoor environment where the UAV is deployed.

The outline of this chapter is as follows. In Section 3.2, the problem of the aerial transportation of cable-suspended loads is formulated. Section 3.3 introduces 2 algorithms for estimating the cable's state based on visual data acquired by a conventional or an event camera. In Section 3.4, a model-free policy is trained using deep RL in order to achieve the swing-free transportation of the load while, in Section 3.5, domain randomization is incorporated so as to learn a more generalized policy. Section 3.6 presents an NMPC scheme utilizing GPs to estimate in real-time the unmodeled dynamics and the external disturbances acting on the platform. Finally, Section 3.7 demonstrates the efficacy of the proposed state estimation techniques and control schemes through real-world experiments.

3.2 Problem Formulation



FIGURE 3.1: The octorotor with the cable-suspended load deployed in the field.

We consider the motion control problem of transporting a cable-suspended load towards reference waypoints using a multirotor, as depicted in Figure 3.1. In order to accomplish the safe transportation of the load, it is essential to take the swinging motion of the cable into account during the autonomous navigation of the platform. Otherwise, the arbitrary motion of the UAV may provoke severe oscillations of the

cable and lead to large angular displacements which may jeopardize the safety of the whole system. To this end, our ultimate goal is to develop appropriate control schemes, either model-free or model-based, that handle the complicated coupled dynamics of the platform and compute suitable commands $\phi_d, \theta_d, \psi_d, v_{z_d}$ for the inner attitude controller of the autopilot in order to:

- minimize the position error $\|\mathbf{p} - \mathbf{p}_{ref}\|$ between the actual position of the UAV and the reference one $\mathbf{p}_{ref} \in \mathbb{R}^3$.
- minimize the cable velocity $\dot{\mathbf{n}}$, or equivalently the angular velocity $\boldsymbol{\omega}_L$, as well as the cable angles $\boldsymbol{\eta}_L$. The latter implies that the cable should be maintained as close as possible to a vertical configuration, i.e., parallel to the z axis of the world frame \mathbf{W} , and, hence, the components n_x, n_y of the cable unit vector \mathbf{n} should be minimized.

However, the aforementioned closed-loop controllers require robust and high-speed estimation of the cable's state besides the vehicle's state, which is directly provided by the autopilot. Considering that the octorotor operates in outdoor environments, where motion capture systems cannot be utilized, we design state estimation algorithms which, based on visual information obtained by a single standard camera or a dynamic vision sensor (DVS), fulfill the following objectives:

- Robust and real-time identification of the cable despite the challenging outdoor conditions.
- Estimation of the complete state of the cable, i.e., both $\boldsymbol{\eta}_L$ and $\boldsymbol{\omega}_L$, required for any feedback controller.

3.3 Estimation of the Cable's State

In this section, we present 2 methodologies for estimating the state of the cable during the aerial transportation of the load. In order to avoid the integration of multiple sensors, e.g., encoders, which results in complex mechanical interventions in the UAV platform, or/and the development of appropriate observers [7], [45], [73], we aim to exploit a single downward-looking camera, either a standard frame-based or an event camera. Consequently, given that almost every UAV comes with a mounting for a camera, the proposed algorithms can be implemented with minimal prerequisites.

3.3.1 Frame-based Estimation of the Cable's State

CNN-based Cable Detection

Consider the RGB image obtained by the downward-looking camera, located near the vehicle's CoM, as depicted in Figure 3.2a. Our objective is to process the aforementioned image in order to identify the cable and, eventually, detect the load in the image plane. Various methods and algorithms exist in order to achieve object

detection, such as utilization of markers, histogram of oriented gradients (HOG) etc. However, since the UAV with the cable-suspended load is operating in outdoor environments, where various lighting conditions, visual occlusions and shadows are observed, a neural network approach is chosen due to its robustness and versatility.

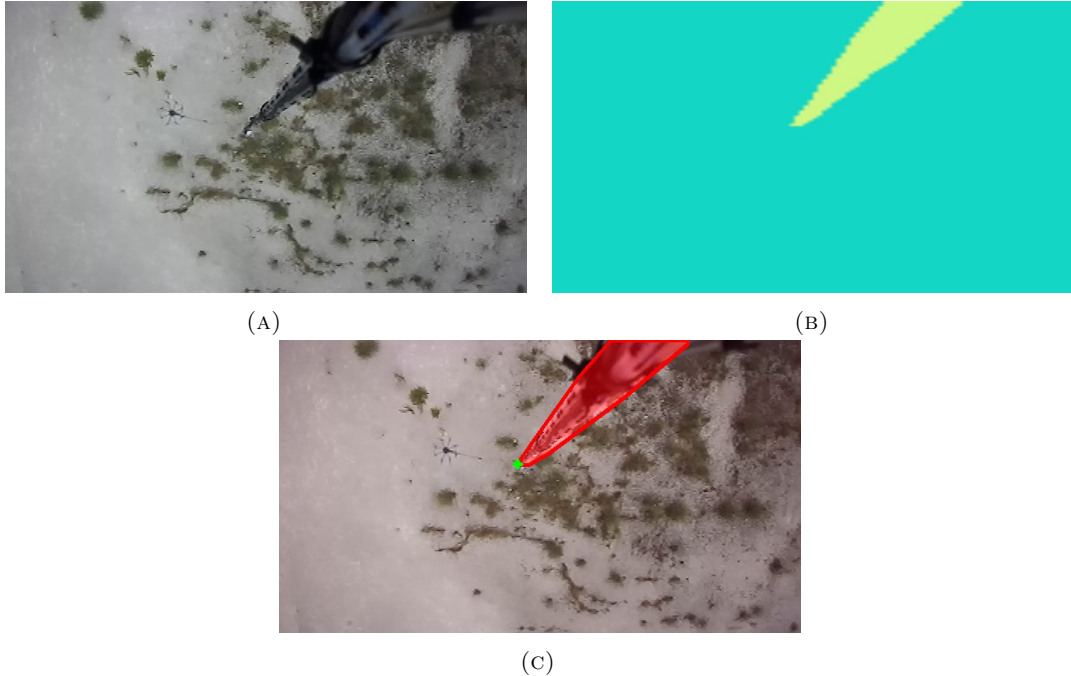


FIGURE 3.2: (a) The input RGB image obtained by the downward-looking frame-based camera. (b) The output segmented image produced by the CNN. (c) The RGB image combined with the output of CNN. The red pixels correspond to the pixels which the CNN identifies as cable. The green cross represents the edge of the cable, i.e., the load. The convex hull of the CNN output is also illustrated.

More precisely, the *mobilenet_segnet* model, which consists of multiple convolutional and pooling layers, was employed from the Keras image segmentation framework [28] due to its low computational cost. Specifically, we found that the average time for the inference of the Convolutional Neural Network (CNN) was 33 ms on the embedded computer of the vehicle, namely the Jetson AGX Xavier [35], when the VGA resolution is used for the ZED 2 camera [98]. Regarding the training procedure, a dataset of 1000 images were captured by manually flying the vehicle with the suspended load in various environments. The dataset was then divided into training and validation set according to a 90:10 split. The training dataset was labelled using the *labelme* software [93] and afterwards the size of the dataset was increased to 4000 images by applying multiple transformations on the images with the *imgaug* image augmentation tool [37]. Eventually, the CNN was trained via the back-propagation method and converged to over 98% accuracy on the validation set after 15 epochs. An indicative example of the output of the trained CNN is illustrated in Figure 3.2b.

Based on the output of the trained CNN, a region of interest is extracted by means of a convex hull, the edge (u_L, v_L) of which denotes the pixel coordinates of

the load in the image plane as depicted in Figure 3.2c. According to the typical pinhole camera model, the position of the load, i.e., the lower edge of the cable, ${}^C\mathbf{p}_L \in \mathbb{R}^3$ with respect to the camera frame \mathbf{C} is extracted from the pixel coordinates $u_L \in [-W/2, W/2]$, $v_L \in [-H/2, H/2]$, as follows:

$${}^C\mathbf{p}_L = \begin{bmatrix} {}^Cz_L u_L / f & {}^Cz_L v_L / f & {}^Cz_L \end{bmatrix}^T \quad (3.1)$$

where $W \times H$ is the resolution of the camera, f is the focal length, and Cz_L is the cartesian coordinate of the cable's edge w.r.t. the z axis of \mathbf{C} . Since the distance Cz_L is unknown, the constant cable length l is exploited so as to retrieve the complete 3D information about the position of the load. Specifically, given that the cable is assumed to be taut and attached to the origin of the body-fixed frame \mathbf{B} , the following equation is derived:

$$\|{}^B\mathbf{p}_L\| = \|\mathbf{R}_{BC} {}^C\mathbf{p}_L + \mathbf{t}_{BC}\| = l \quad (3.2)$$

where ${}^B\mathbf{p}_L$ is the position of the cable's edge w.r.t. \mathbf{B} , and \mathbf{R}_{BC} and \mathbf{t}_{BC} are the known rotation matrix and translation vector from \mathbf{C} to \mathbf{B} . Solving Equation (3.2) results in a function $h : \mathbb{R}^2 \mapsto \mathbb{R}$ that maps the coordinates (u_L, v_L) – and known parameters, namely, the elements of \mathbf{R}_{BC} and \mathbf{t}_{BC} , the intrinsic parameters of the camera, and the cable's length l – with the distance Cz_L , i.e., ${}^Cz_L = h(u_L, v_L)$. The analytical form of the function h is omitted since it relies on the specific relative pose between the camera and the body-fixed frame. Eventually, the cable unit vector ${}^B\mathbf{n}$ is computed:

$${}^B\mathbf{n} = \frac{{}^B\mathbf{p}_L}{\|{}^B\mathbf{p}_L\|} = \frac{\mathbf{R}_{BC} {}^C\mathbf{p}_L + \mathbf{t}_{BC}}{l} \quad (3.3)$$

and is then transformed to the world frame \mathbf{W} ($\mathbf{n} = \mathbf{R}_{WB} {}^B\mathbf{n}$).

Kalman Filter

Besides the cable unit vector \mathbf{n} , an estimate of the cable velocity is essential for the subsequent control schemes. Although direct Euler differentiation of the measurement \mathbf{n} is a possible solution, it typically results in a considerably noisy estimate and, hence, in degraded control performance.

Consequently, a Kalman Filter is selected since it is a widely adopted framework for state estimation [15], [49]. The Kalman Filter is implemented based on the dynamic model of Equation (2.7). More precisely, defining $\mathbf{x}_L = [\mathbf{n}^T \quad \dot{\mathbf{n}}^T]^T \in \mathbb{R}^6$ as the state vector, the following process and measurement models are considered in the Kalman Filter formulation:

$$\dot{\mathbf{x}}_L = \begin{bmatrix} \dot{\mathbf{n}} \\ \ddot{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{n}} \\ \frac{1}{l} \left(-g\mathbf{e}_z - \frac{\|\mathbf{T}\|}{m_L} \mathbf{n} - \dot{\mathbf{v}} \right) \end{bmatrix} + \mathbf{w} \quad (3.4)$$

$$\mathbf{y}_L = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{n} \\ \dot{\mathbf{n}} \end{bmatrix} + \boldsymbol{\omega} \quad (3.5)$$

where $\mathbf{w} \in \mathbb{R}^6$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ represent additive process and observation Gaussian noises. An Extended or Unscented Kalman Filter (EKF, UKF) is exploited due to the nonlinear process model. It is mentioned that the acceleration of the UAV $\dot{\mathbf{v}}$ and the tension of the cable $\|\mathbf{T}\|$ are directly measured by the IMU of the vehicle and a load cell, placed upon the cable, correspondingly. Once the cable unit vector \mathbf{n} and the cable velocity $\dot{\mathbf{n}}$ are estimated, the cable angles $\boldsymbol{\eta}_L$ and angular velocity $\boldsymbol{\omega}_L$ are computed according to Equations 2.2 and 2.3.

3.3.2 Event-based Estimation of the Cable's State

In order to achieve the estimation of the cable state within a much shorter time period compared to the above frame-based methodology, we investigate the integration of a Dynamic Vision Sensor (DVS), i.e., an event camera, into the vehicle, as illustrated in Figure 3.3. Event cameras offer multiple advantages compared to standard ones, including low latency, low power consumption, high dynamic range, and robustness against motion blur [20]. Various works in the literature have successfully demonstrated the deployment of event cameras in outdoor environments for a wide range of applications, such as powerline inspection [12], pole tracking [90], visual guidance for ornithopter robot flight [13], intruder monitoring [74], autonomous quadrotor flight despite loss of a single rotor [83], or event-based odometry [53]. However, due to the asynchronous nature of event cameras and the novel ways of acquiring visual data, each work is unique and highly dependent on the object to be identified and the task to be addressed. Hence, a novel solution adapted to the specific problem of identifying the cable and estimating its state during aerial transportation is designed.

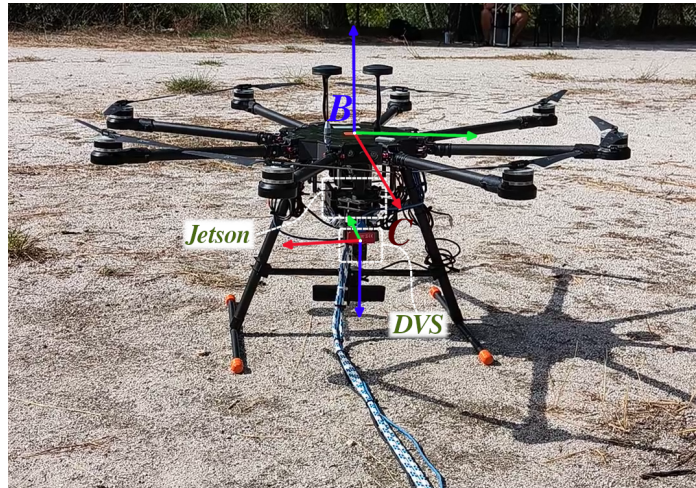


FIGURE 3.3: The octorotor UAV with the onboard computer and the DVS camera (DAVIS 346). The body-fixed \mathbf{B} and the camera \mathbf{C} frames are also depicted. The red, green, blue colored arrows correspond to the x, y, z axes of each frame.

Event Representation

In contrast to frame-based conventional cameras, an event camera is a bio-inspired sensor which responds to luminance changes asynchronously and independently for each pixel. Hence, the output of a DVS camera is an asynchronous stream of “events” triggered by brightness changes in the scene. More precisely, an event $e_k = (u_k, v_k, t_k, p_k)$ is triggered at pixel coordinates (u_k, v_k) and at time t_k as soon as a change in log luminance above a predefined threshold is sensed compared to the last event at the same pixel. The polarity $p_k \in \{0, 1\}$ indicates the direction of luminance change.

According to the proposed algorithm, the events generated by the DVS are pre-processed in order to identify the ones that correspond to the cable and extract later significant information about its state. More precisely, as soon as an event e_i is triggered, it is stored in a point cloud $\mathcal{P} = \{e_i | t_i \in [t_0 - T, t_0]\}$, which consists of the most recent events up to the current time instant t_0 . It is mentioned that both spatiotemporal information (u, v, t) and polarity p are stored in the point cloud. Indicative examples of “event” point clouds, along with the respective grayscale images captured by the DVS, are illustrated in Figure 3.4, where an increased number of events, corresponding to the background or to noise, is observed in outdoor environments. The Point Cloud Library (PCL) [75], which distinguishes itself for its computational efficiency, is exploited for the representation of the “event” point cloud and the subsequent processing algorithms. Regarding the representation, a point cloud with points of type XYZI is selected from the PCL, where the X-Y coordinates are used for the pixel values, the Z coordinate for the time, and the intensity value I for the polarity of the events.

Event Point Cloud Processing

The aforementioned point cloud \mathcal{P} is downsampled by applying a 3D voxel grid filter within a spatiotemporal neighborhood of each event, defined by an $n_u \times n_v$ pixel window by n_t ms volume. Hence, the events that belong to each voxel are approximated by their centroid. It is highlighted that, besides the pixel coordinates and the timestamps, the polarity of the events in each voxel is also averaged. Consequently, the newly generated point cloud $\mathcal{P}' = \{e'_j | t'_j \in [t_0 - T, t_0]\}$ consists of centroids e'_j with real-valued polarities $p'_j \in [0, 1]$, as illustrated in Figure 3.5a.

Afterward, the point cloud \mathcal{P}' is filtered based on the average polarity p'_j of each centroid e'_j . More specifically, only the centroids with polarity $p'_j \in (0, 1)$, i.e., $p'_j \neq 0$ and $p'_j \neq 1$, are considered as candidate events to belong to the cable, as depicted in Figure 3.5b. This filtering step essentially implies that there should be at least 2 events e_i in each voxel of the point cloud \mathcal{P} with opposite polarities in order to promote the respective centroid e'_j to a candidate event. This filtering approach is derived from the observation that, due to either the motion of the vehicle or the

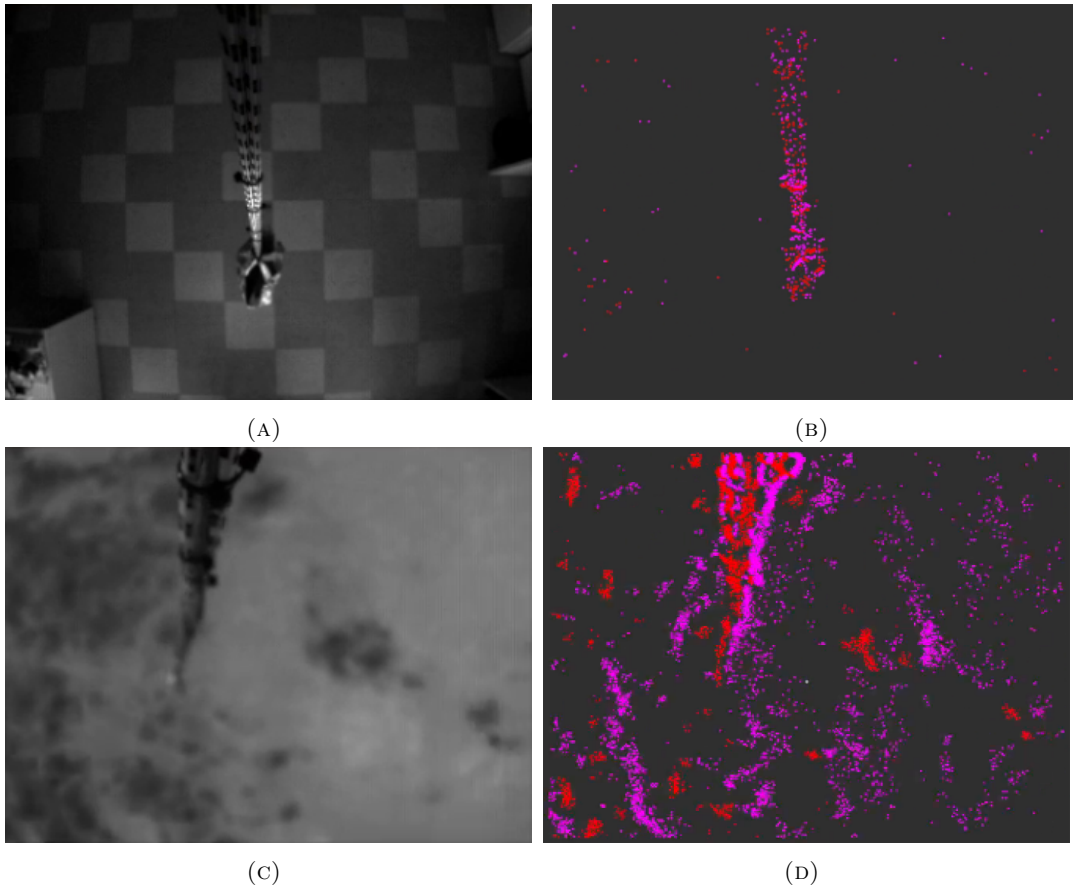


FIGURE 3.4: The grayscale images captured by the DVS in an (a) indoor and (c) outdoor environment at the current time instant t_0 . A top-down view ($u-v$ plane) of the corresponding “event” point clouds \mathcal{P} for $T = 25$ ms in the (b) indoor and (d) outdoor environment. The purple and red colored points of \mathcal{P} correspond to events with polarities $p = 1$ and $p = 0$ respectively.

cable itself, an adequate number of events of opposite polarities is generated on a ms scale during the transportation of the suspended load. Additionally, when the UAV hovers at a specific position, both the UAV and the cable cannot be firmly stable due to existence of external disturbances, arising mainly from the wind in outdoor conditions. Another advantage of this filter is that the majority of noisy and spurious events are disregarded. More specifically, after an event is triggered, additional events of the same polarity and at the same pixel location are sometimes generated within a certain time interval [12]. Consequently, by applying the aforementioned filtering step, these same-polarity events are not promoted.

However, there exist some centroids e'_j that pass the above-mentioned filter but correspond to either noise or objects that lie on the terrain/background and not to the cable. Thus, an Euclidean Clustering algorithm in the spatiotemporal space is performed using the effective Kd-tree representation for the input point cloud $\mathcal{P}'' = \{e'_j | t'_j \in [t_0 - T, t_0], p'_j \in (0, 1)\}$ and subsequently employing the nearest neighbor search algorithm. It is highlighted that the cluster tolerance should be properly selected in order to avoid the cable being detected as multiple sub-clusters. Finally, the cluster with the maximum size is promoted to the cable cluster \mathcal{C} since the cable is closer to the DVS than any other object on the terrain and is assumed to have a sufficient diameter. Thus, the events that represent the cable typically occupy most of the image plane, as depicted in Figure 3.5c. Further checks are also employed between consecutive extracted cable clusters by comparing their relative position and, thus, ensuring the proper selection of the cable cluster when objects of similar shape lie on the terrain. Eventually, a sequence of cable clusters is obtained in a quasi-continuous manner, as illustrated in Figure 3.6.

As soon as the cable cluster \mathcal{C} is extracted, it is essential to find the event that corresponds to the load, i.e., the lower edge of the cable. Based on the specific configuration and relative pose of the DVS with respect to the cable, a naive approach would be to pick the event $e'_j \in \mathcal{C}$ with the maximum coordinate v'_j ; however, this approach degrades the importance of the remaining events. Consequently, a 2D line fitting is performed considering the pixel coordinates (u'_j, v'_j) of the events $e'_j \in \mathcal{C}$, as depicted in Figure 3.5c.

Approximation by a Bézier Curve

By employing the aforementioned processing pipeline, the pixel coordinates (u_L, v_L) of the lower edge of the cable are eventually obtained. However, the lower part of the cable might not generate an adequate number of events during some sparse iterations, which results in a noisy estimate of the cable's state and degrades the control performance. Consequently, Bézier curves are exploited in order to both smoothly approximate the coordinates (u_L, v_L) and obtain the pixel velocities (\dot{u}_L, \dot{v}_L) , required for the estimation of the complete state of the cable.

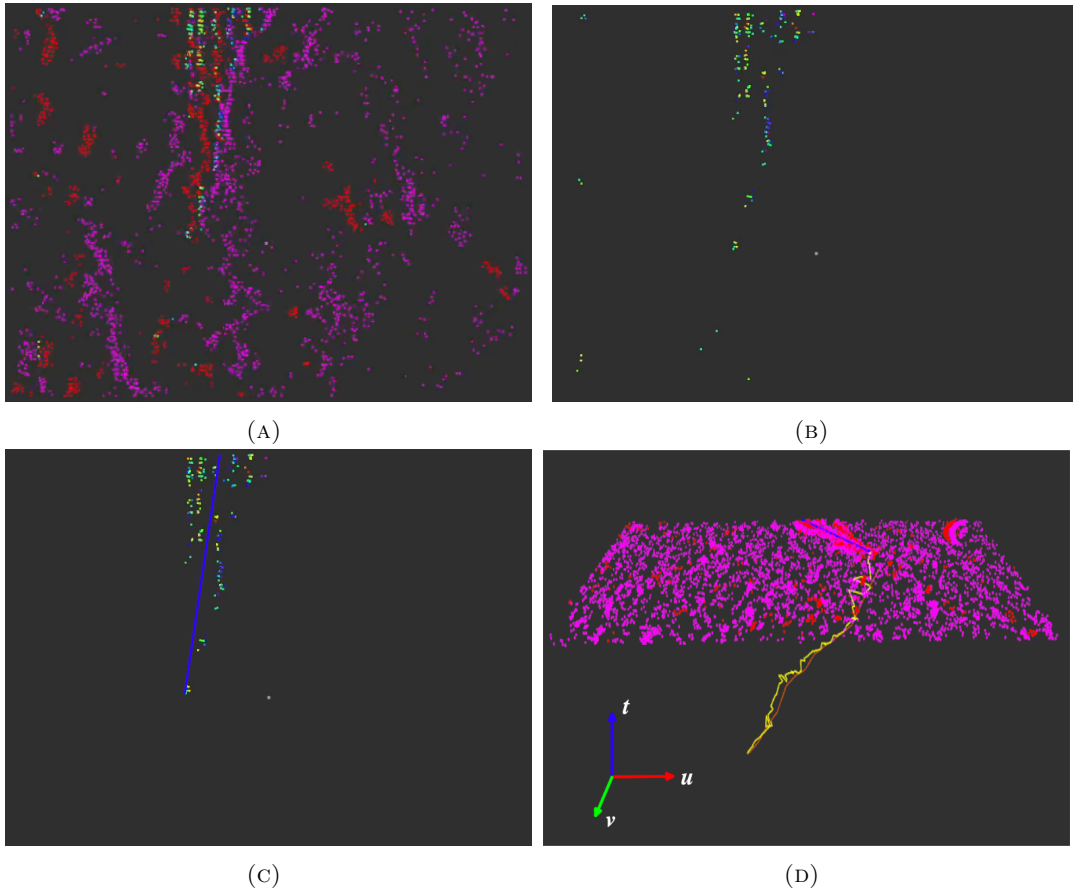


FIGURE 3.5: (a), (b), (c) The processing algorithm applied to the “event” point cloud \mathcal{P} of Figure 3.4d: (a) A top-down view (u - v plane) of the point cloud \mathcal{P}' generated after the downsampling of \mathcal{P} . The purple and red colored centroids of \mathcal{P}' correspond to centroids with polarities $p = 1$ and $p = 0$ respectively while the remaining centroids have polarities $p' \in (0,1)$. (b) A top-down view of the point cloud \mathcal{P}'' after the filtering of \mathcal{P}' based on the polarity of each centroid. (c) A top-down view of the cable cluster \mathcal{C} and the respective blue colored line fitted to \mathcal{C} . (d) The Bézier curve (orange color) which approximates the measurements of the lower edge of the cable (yellow color).

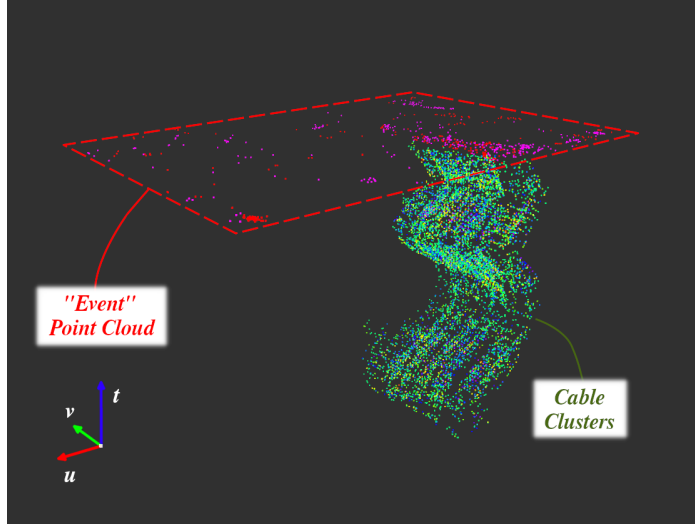


FIGURE 3.6: The current “event” point cloud, obtained by the DVS, and a sequence of cable clusters, extracted according to the proposed method, throughout the last 2 s in an outdoor environment. The u - v axes denote the image plane and the t axis represents the time. The purple and red colored points of the “event” point cloud correspond to events with polarities $p = 1$ and $p = 0$ respectively.

More specifically, a Bézier curve is given, according to the order n of the curve, as follows:

$$\mathbf{B}(\tau) = \sum_{i=0}^n b_{i,n}(\tau) \mathbf{c}_i \quad (3.6)$$

where $\tau \in [0, 1]$, $\mathbf{B}(\tau) \in \mathbb{R}^2$ (for the u - v image plane) is the Bézier curve, $b_{i,n}(\tau)$, $i = 0, \dots, n$ are the Bernstein basis polynomials of degree n , and $\mathbf{c}_i = [c_{u,i} \ c_{v,i}]^T \in \mathbb{R}^2$ are the $n + 1$ control points.

In order to compute the suitable set of control points $\mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n]$ and, thus, approximate the trajectory of the cable in the image plane, a dataset $D = \{(t_{N-1}, u_{L,N-1}, v_{L,N-1}), \dots, (t_0, u_{L,0}, v_{L,0})\}$ is constructed, consisting of the last N measurements, up to the current time instant t_0 , of the pixel coordinates (u_L, v_L) along with the respective timestamp, as obtained by the above processing pipeline. The dataset has a constant length and is updated once a new measurement is available.

After performing a mapping between the parameter $\tau \in [0, 1]$ and the time interval $t \in [t_{N-1}, t_0]$, the optimal set of control points \mathbf{c} is the solution of the optimization problem:

$$\min_{\mathbf{c}} \sum_{i=0}^{N-1} \left(w_{t_i} \left\| \mathbf{B}(\tau_i) - [u_{L,i}, v_{L,i}]^T \right\|_2^2 \right) \quad (3.7)$$

which essentially minimizes the distance between the Bézier curve and the measurements, while the weights w_{t_i} boost the more recent measurements and impose penalties on the previous ones [29].

Moreover, constraints, related to the minimum and maximum velocity and acceleration of the cable in the image plane, are incorporated into the optimization problem, so as to find solely feasible solutions. More precisely, by differentiating the Bézier curve, the constraints are formulated as follows for the u axis (similarly for the v one):

$$\begin{aligned} n \cdot (c_{u,i} - c_{u,i-1}) / (t_0 - t_{N-1}) &\in [-\dot{u}_{max}, \dot{u}_{max}] \\ \frac{n \cdot (n-1) \cdot (c_{u,i} - 2c_{u,i-1} + c_{u,i-2})}{(t_0 - t_{N-1})^2} &\in [-\ddot{u}_{max}, \ddot{u}_{max}] \end{aligned} \quad (3.8)$$

The above-mentioned constrained Quadratic Programming problem (Equation (3.7)) with $2 \cdot (n+1)$ primal variables, i.e., the control points, and $2 \cdot n + 2 \cdot (n-1)$ constraints (Equation (3.8)) is solved with the aid of the OOQP software [24] and, eventually, an estimate of the pixel coordinates (u_L, v_L) and velocities (\dot{u}_L, \dot{v}_L) at each current time instant t_0 is derived from the Bézier curve, illustrated in Figure 3.5d.

Afterward, given the Bézier curve, the complete state of the cable is computed. More precisely, similarly to the frame-based method, the cable unit vector ${}^B \mathbf{n}$ is computed according to Equations 3.2 and 3.3. Regarding the cable velocity, it is obtained by differentiating Equation (3.3) according to the following relationships:

$${}^B \dot{\mathbf{n}} = \frac{\mathbf{R}_{BC}}{l} {}^C \dot{\mathbf{p}}_L = \frac{\mathbf{R}_{BC}}{l} \frac{1}{f} \begin{bmatrix} {}^C \dot{z}_L u_L + {}^C z_L \dot{u}_L \\ {}^C \dot{z}_L v_L + {}^C z_L \dot{v}_L \\ f {}^C \dot{z}_L \end{bmatrix} \quad (3.9)$$

$${}^B \dot{\mathbf{n}} = \frac{\mathbf{R}_{BC}}{l \cdot f} \begin{bmatrix} \left(\frac{\partial h}{\partial u_L} \dot{u}_L + \frac{\partial h}{\partial v_L} \dot{v}_L \right) u_L + {}^C z_L \dot{u}_L \\ \left(\frac{\partial h}{\partial u_L} \dot{u}_L + \frac{\partial h}{\partial v_L} \dot{v}_L \right) v_L + {}^C z_L \dot{v}_L \\ f \left(\frac{\partial h}{\partial u_L} \dot{u}_L + \frac{\partial h}{\partial v_L} \dot{v}_L \right) \end{bmatrix} \quad (3.10)$$

with $\frac{\partial h}{\partial u_L}$, $\frac{\partial h}{\partial v_L}$ denoting the partial derivatives of the function ${}^C z_L = h(u_L, v_L)$. Consequently, given the output of the Bézier curve, i.e., the coordinates (u_L, v_L) and the velocities (\dot{u}_L, \dot{v}_L) , both the cable unit vector ${}^B \mathbf{n}$ and the cable velocity ${}^B \dot{\mathbf{n}}$ can be computed and then transformed to the world frame \mathbf{W} so as to calculate the angles $\boldsymbol{\eta}_L$ and angular velocity $\boldsymbol{\omega}_L$ (Equations 2.2 and 2.3).

Accuracy of the Method

The accuracy of the proposed method was initially evaluated through indoor experiments, where the output of the aforementioned event-based solution was compared against accurate measurements provided by a motion capture system. More precisely, the DVS and the upper edge of the cable were rigidly mounted on a ceiling with a relative pose similar to the one on the UAV. It is mentioned that in this case, the world frame coincides with the body-fixed one. Ground truth was obtained by the Vicon motion capture system based on markers that were cautiously placed around the cable to ensure the continuous tracking of its swinging motion. An image of the

indoor environment, acquired from the frame-based camera of the DAVIS 346, and the respective “event” point cloud are illustrated in Figures 3.4a and 3.4b respectively.

Throughout the experiments, data from both the DVS and the Vicon system were gathered by varying the initial angle of the load and releasing it with zero velocity. Hence, the load moved freely under the influence of gravity. Indicative results for an indoor experiment are demonstrated in Figure 3.7 where the cable angles $\eta_L = [\phi_L \ \theta_L]^T$ obtained by the processing of the event stream are compared against the ground truth captured by the Vicon system. It is evident that our method successfully distinguishes the cable and captures the entire range of its swinging motion. The Root Mean Square Errors (RMSE) for the angles ϕ_L and θ_L were 2.204° and 2.639° correspondingly during the indoor tests. Finally, the success rate, i.e., the number of detections divided by the total number of iterations, was equal to 96.67%.

However, it should be highlighted that although our method captures the frequency of the cable's swinging motion, the magnitude of the cable angles is overestimated, especially when the angles considerably move away from the zero degrees. This behavior is mainly due to the fact that the lowest part of the cable may not generate an adequate number of events and, hence, our method may identify as the edge of the cable (u_L, v_L) a centroid that does not actually correspond to the edge but instead to a point located a few centimeters above. Nevertheless, the RMSEs for the angles are relatively small while also the ultimate goal of our work is the aerial transportation with minimum swing angles, where the aforementioned behavior is alleviated.

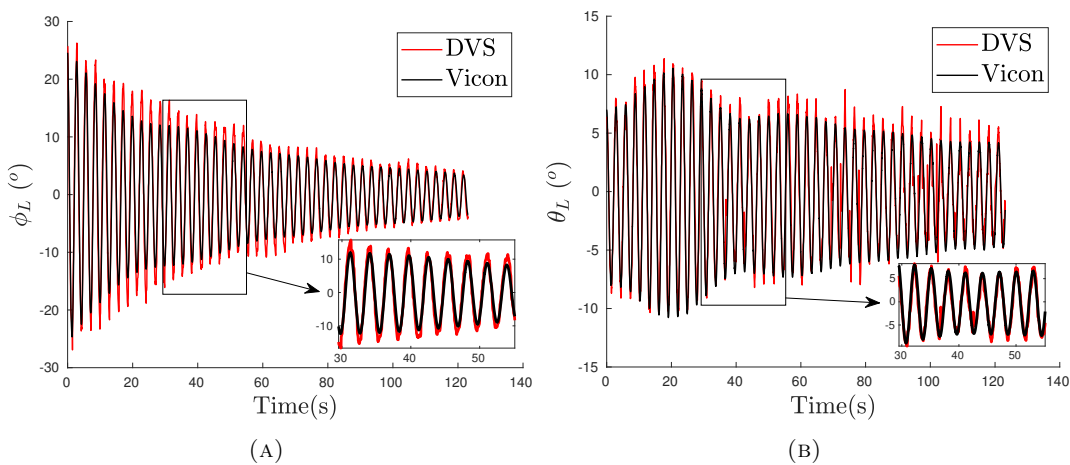


FIGURE 3.7: The cable angles, (a) ϕ_L and (b) θ_L , computed according to the event-based algorithm and compared against the accurate measurements provided by the Vicon system during the indoor experiment.

Computational Cost

In order to quantify the computational cost reduction achieved with this method compared to the frame-based solution, the execution time of the individual processing steps of the event-based algorithm is measured on the Jetson AGX Xavier in the case of both indoor and outdoor environments. The mean processing time for each of the steps that comprise the overall method, i.e., the processing of the “event” point cloud, and the approximation by a Bézier curve, as well as the parameters of the proposed algorithm are summarized in Tables 3.1 and 3.2 respectively. It is mentioned that the computational cost of the point cloud processing is directly related to the number of incoming events. Consequently, during the indoor experiments, where the DVS is not moving and the surrounding environment is static, the execution time of this step is reduced. In any case, the mean processing time of the proposed method is significantly lower compared to a standard frame-based detection algorithm with CNNs, where the average time for the inference of the CNN is equal to 33 ms.

Step	Time[ms]	
	Indoor	Outdoor
Point Cloud Processing	3.424	8.091
Bézier curve ($N = 100$)	0.302	0.337

TABLE 3.1: The computational cost of the individual processing steps that comprise the event-based method.

Parameter	Indoor	Outdoor
T [ms]	25	25
$n_u \times n_v \times n_t$ [px \times px \times ms]	$5 \times 5 \times 10$	$10 \times 3 \times 5$

TABLE 3.2: The parameters of the event-based method.

3.4 Deep Reinforcement Learning Control

In this section, we present a deep Reinforcement Learning (RL) strategy for controlling the multirotor with the cable-suspended load. Specifically, a policy, represented by a deep neural network (NN), is trained in a model-free manner with the ultimate goal to produce appropriate continuous control inputs which navigate the vehicle towards reference waypoints while, simultaneously, compensating for the cable oscillations.

Modern model-free deep RL algorithms alleviate the need for system identification and have been proven to be an efficient tool for learning policies capable of addressing complicated tasks and controlling agents with complex system dynamics, including UAVs [34], [42]. However, the concept of leveraging deep RL for the aerial transportation of suspended loads is original.

3.4.1 Reinforcement Learning Background

According to the classic Reinforcement Learning Theory, the agent's environment is formulated as a Markov Decision Process (MDP). At each time step t , a MDP is described by the tuple $(s_t, a_t, p(s_{t+1}|s_t, a_t), r_t)$, where s_t is the current state of the agent, a_t is the chosen action, $p(s_{t+1}|s_t, a_t)$ is the transition probability or function, depending on whether stochastic or deterministic dynamics are considered, from the current state s_t to the next one s_{t+1} when action a_t is applied, and $r_t = r_t(s_t, s_{t+1}, a_t)$ is the reward function which quantifies the performance of the agent.

Assuming an episodic setting, the aim of the RL problem is to find an optimal policy which maximizes the cumulative reward or return over the episode. More specifically, an episode is terminated after T interactions of the agent with the environment, $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$, or when a termination criterion is reached. The return is computed as the discounted sum of the rewards obtained during an episode, $R(\tau) = \sum_{t=0}^{T-1} \gamma^t r_t(s_t, s_{t+1}, a_t)$, where $\gamma \in (0, 1)$ is a discount factor which favors current rewards over future ones. In deep RL, the objective is to find a policy neural network π_θ , parameterized by some weights θ , which given the current state s_t of the agent produces actions $a_t = \pi_\theta(s_t)$, in the deterministic case, that maximize the expected return $\mathbb{E}_{\tau \sim \pi} [R(\tau)]$.

According to the existing literature [6], the model-free RL algorithms, which are used for training agents, may be classified into three main categories: policy optimization, value function learning and actor-critic methods. In summary, policy optimization methods improve directly the parameterized policy π_θ by applying gradient ascent on the expected return. In contrary, value function methods rely on approximating the optimal Q function, defined as $Q^*(s, a) = \max_{\pi} Q(s, a) = \max_{\pi} \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s, a_0 = a]$, with a deep neural network $Q_\phi(s, a)$, and, then, choosing actions greedily, $a(s) = \underset{a}{\operatorname{argmax}} Q_\phi(s, a)$.

Actor-critic methods combine the advantages of policy optimization and value-function learning methods. The actor network is the policy π_θ whereas the critic network represents the Q function $Q_\phi(s, a)$. Actor-critic methods are an appropriate choice for continuous action spaces and distinguish for their sample efficiency. Consequently, the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, an actor-critic method proposed by [19], is selected in order to train the UAV with the slung load.

3.4.2 Structure of Networks

Following the common actor-critic architecture, two neural networks are considered, in particular the policy network $\pi_\theta(s)$ and the Q function network $Q_\phi(s, a)$. The input layer of the policy is the state of system, which specifically for the case of an octorotor with a cable-suspended load consists of the position error $\mathbf{p} - \mathbf{p}_{ref}$, the velocity \mathbf{v} , the orientation of the vehicle $\boldsymbol{\eta}$, i.e., the roll ϕ and pitch θ , the respective angular rates,

the x, y elements of the cable vector \mathbf{n} , n_x and n_y , and the corresponding elements of cable velocity $\dot{\mathbf{n}}$. It is noted that only the position error is taken into account, instead of both the current position of the vehicle and the goal position, in order to reduce the dimension of the state vector. Additionally, the state space is augmented with the last action a_{t-1} taken by the agent, in order to penalize action oscillations in the subsequent reward function and learn a smooth control policy [60]. The action vector, i.e. the output of the policy network, is composed of the desired commands $\phi_d, \theta_d, v_{z_d}$ for the inner attitude controller of the autopilot. It is mentioned that the yaw angle is maintained constant to zero. Briefly, the state and action vectors are given by:

$$s = [\mathbf{p} - \mathbf{p}_{ref} \quad \mathbf{v} \quad \boldsymbol{\eta} \quad \dot{\boldsymbol{\eta}} \quad \mathbf{n} \quad \dot{\mathbf{n}} \quad a_{t-1}]^T \quad (3.11)$$

$$a = [\phi_d \quad \theta_d \quad v_{z_d}]^T \quad (3.12)$$

The policy network is a fully connected multi-layer deep neural network, as shown in Figure 3.8a, with 2 hidden layers of 64 nodes and tanh activation function. The presence of the tanh function restricts each output of the policy network to the range $[-1, 1]$ and, hence, the actions are afterwards multiplied by the maximum roll, pitch and climb rate respectively.

Regarding the structure of the Q function network, the inputs are the states s and the actions a (Figure 3.8b). The state and the action vectors pass through 2 separate layers of 16 and 32 nodes correspondingly, before concatenating them to a common tensor which is then inputted to 2 hidden layers of 256 nodes each. Rectified Linear Unit (ReLU) activation function is used in all layers. The output of the Q function network is a real number which represents the Q value.

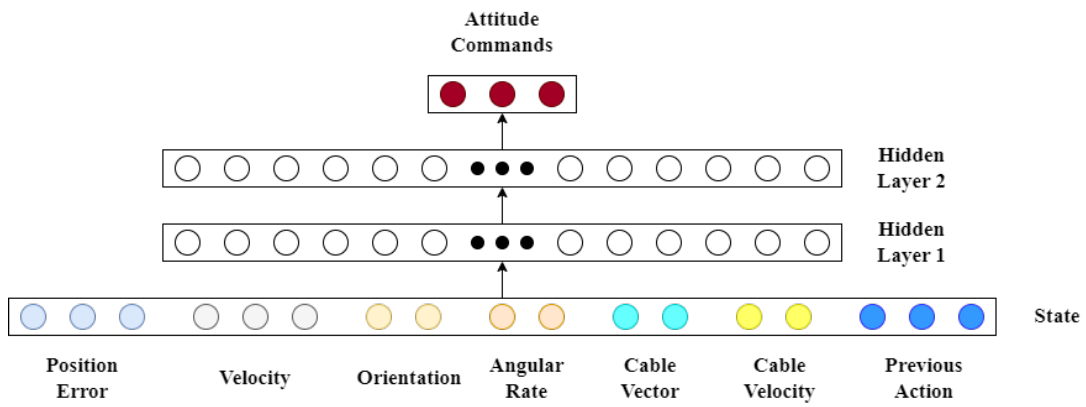
3.4.3 Reward Function

The reward is a user-defined function, which is of vital importance for the convergence of the learning procedure and should satisfy the requirements of the application. To this end, considering the studied task, i.e., the swing-free transportation of the load towards reference positions, the reward is computed at each interaction of the agent with the environment as the weighted linear combination of the following terms:

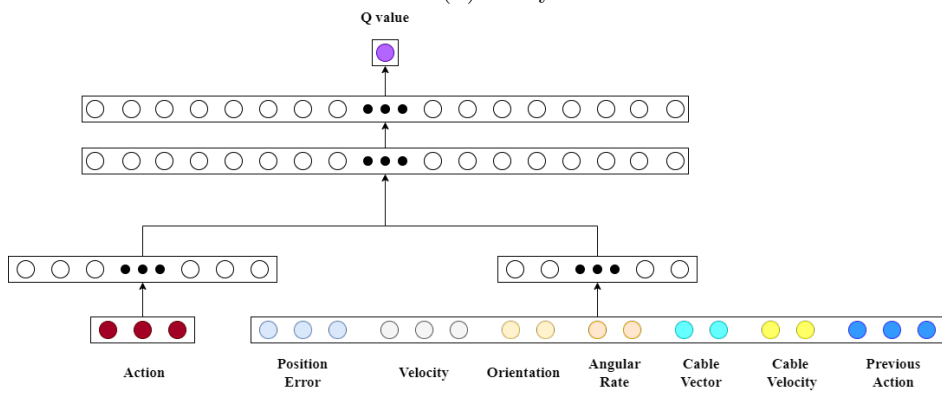
$$r_t = \mathbf{w}^T \boldsymbol{\rho}_t \quad (3.13)$$

$$r_t = \mathbf{w}^T [\|\mathbf{p}_t - \mathbf{p}_{ref}\|, \|\mathbf{v}_t\|, \|\boldsymbol{\eta}_t\|, \|\dot{\boldsymbol{\eta}}_t\|, \|\mathbf{n}_t\|, \|\dot{\mathbf{n}}_t\|, \|a_t\|, \|a_t - a_{t-1}\|]^T \quad (3.14)$$

where \mathbf{w} is a vector of weights assigned to each term of $\boldsymbol{\rho}_t$ at any given time step t . Negative weights are considered since the objective lies in the maximization of the return. By employing the aforementioned reward, the minimization of the position error and the reduction in the swinging motion of the cable are encouraged. Additionally, the jerky motion of the vehicle, originating from unnecessary sudden changes in the control signals, is avoided while also the energy consumption is reduced.



(A) Policy Network



(B) Q function network

FIGURE 3.8: Structure of Neural Networks.

Parameter	Value
Buffer size	10^6
Mini-batch size	64
Exploration noise ϵ	$\sim N(0, 0.1^2)$
Smoothing noise ϵ'	$\sim N(0, 0.2^2)$
Discount factor γ	0.99
Target networks update τ	0.05

TABLE 3.3: Hyperparameters for TD3 algorithm.

3.4.4 Training Procedure

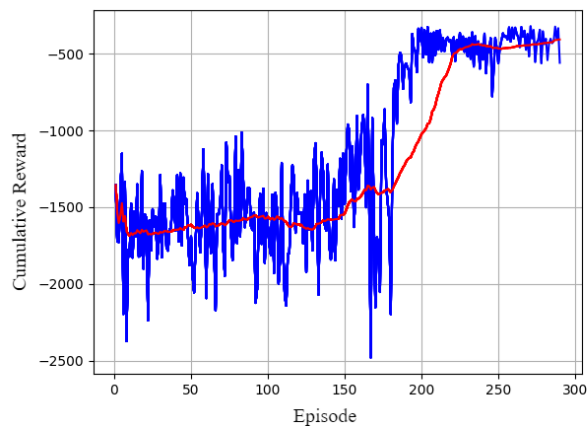


FIGURE 3.9: The cumulative reward during the learning procedure.

The agent is trained using the TD3 algorithm, which employs the structure of actor-critic algorithms and is appropriate for continuous action spaces. The TD3 algorithm is considered as an extension of the Deep Deterministic Policy Gradient (DDPG) [52] algorithm and aims to reduce the overestimation of the Q value function. In comparison with the DDPG algorithm, TD3 uses a pair of neural networks, Q_{ϕ_1} and Q_{ϕ_2} , in order to approximate the Q function and the minimum one is selected during the update of the critic networks. Additionally, in TD3, the policy and the target networks are updated less frequently than the Q function networks. One policy and target networks update is proposed for every two Q function updates. Ultimately, noise is added to the target action and, hence, the Q function estimate is smoothed along similar actions. Thorough details about the TD3 algorithm are presented in [19].

Briefly, the critic networks are updated based on the mean-squared Bellman error, while the policy by applying gradient ascent with respect to θ in order to maximize the Q function approximation. Moreover, target networks, with the same structure as the main ones, are considered so as to provide stability in the training procedure. The weights of the target networks are slowly synchronized to the weights of the main ones by a small update.

The UAV with the cable-suspended load, illustrated in Figure 2.1, was trained on the Gazebo simulation environment of Section 2.2 according to the hyperparameters of the Table 3.3. The ADAM optimizer [40] was used in order to update the weights of the policy and the Q function networks with a learning rate equal to 0.001. The agent was trained in an episodic setting, where each episode consists of 512 time steps. During each episode, the agent starts at a random position, sampled within a bounded 3D space, while the reference position is fixed. The episode is terminated when the maximum number of time steps is reached or the position error exceeds a certain threshold. The learning procedure was terminated when the cumulative reward, i.e. the sum of all rewards during an episode, and the respective moving average, converged, as depicted in Figure 3.9. It should be highlighted that, since the policy is trained based on exclusively simulation data, the existence of a highly realistic simulation environment is a crucial factor for the successful deployment of the learned policy onto the real vehicle at a later stage.

3.5 Robust Deep Reinforcement Learning Control

Although deep RL-based policies can adequately control the dynamics of robotic platforms, a key point for discussion revolves around the gap between simulation and real-world conditions. More precisely, policies are typically trained in exclusively simulation environments, where an unlimited amount of data is available, since training in real-world environments can be both time-consuming and critically unsafe, especially at the initial stages of the learning process. As a result, possible mismatches between the simulator and the real-world environment may impact the performance of the learned policy when transferred to the real robot. Moreover, another crucial issue is the robustness of the policy to varying dynamic parameters of the system. Considering the case of aerial transportation of loads, it is highly impractical to train a policy for each load mass or cable length. The aforementioned parameters may dynamically change even during the same flight. For instance, the UAV may release part of the load or pick an object throughout the same mission. Similarly, the cable length may vary when a pulley is attached to the vehicle [99], e.g., for navigating through narrow spaces.

To this end, we modify the learning procedure of Section 3.4 and incorporate domain randomization [50], [67] in order to bridge the “sim2real” gap and develop a policy which is robust to varying model parameters, e.g., load mass and cable length. The core concept of this technique lies in the randomization of the physical properties of the simulated environment, called the source domain, during the training procedure with the ultimate goal of learning a policy that performs satisfactorily across all of the variations of the simulator. Hence, the learned policy can be successfully transferred to the real-world environment, i.e. the target domain, since the latter is anticipated to be a sample of the variable source domain.

3.5.1 Domain Randomization

As aforementioned, in order to bridge the gap between the Gazebo simulation environment of Section 2.2 and the real conditions and enhance the robustness of the policy, the randomization of multiple parameters is enforced during training. Towards this direction, the following aspects of the source domain are randomized: (i) mass m_L of the load, (ii) cable length l , (iii) observation noise in \mathbf{n} , (iv) observation noise in $\dot{\mathbf{n}}$, and (v) forces \mathbf{F}_Q applied to the UAV. The additive observation noise in the components of the cable vector \mathbf{n} and velocity $\dot{\mathbf{n}}$ models the uncertainty in the sensor-based estimation of the cable state while the forces $\mathbf{F}_Q \in \mathbb{R}^3$ represent the disturbances acting on the vehicle when deployed in outdoor environments. The ranges of the parameters are summarized in Table 3.4.

Parameter	Range
Load Mass m_L [kg]	[0.25,1.0]
Cable Length l [m]	[0.5,2.0]
Noise in \mathbf{n}	[-0.05,0.05]
Noise in $\dot{\mathbf{n}}$	[-0.1,0.1]
Force \mathbf{F}_Q [N]	[-1.0,1.0]

TABLE 3.4: Ranges of parameter randomizations.

3.5.2 Training Procedure

In order to introduce the domain randomization into the training procedure, various modifications in the simulation environment are required compared to Section 3.4. Specifically, a prismatic joint is integrated into the quadrotor of Figure 2.1 so as to modify the cable length during the simulation, while a plugin is designed in order to dynamically change the load mass at runtime. Additionally, although the state and input vectors as well as the reward function are defined similarly to Equations 3.11, 3.12, and 14, the deep neural networks, i.e., the policy $\pi_\theta(s)$ and the Q function network $Q_\phi(s, a)$, are revised. More precisely, the input of the policy contains the state vector, including also the noise in the cable state. In contrast, the ground truth state, i.e., without the additive noise, and the action are concatenated and comprise the input to the Q-function. Moreover, since the Q-function is solely used throughout the simulation, the known randomized parameters of Table 3.4 are also provided as input in order to boost the training procedure, as proposed in [67].

Regarding the learning strategy, fixed parameters are utilized at the early stages of the training. More specifically, the observation noise and the external forces are not initially involved in the simulation while the load mass and the cable length are equal to 0.5kg and 1.0m respectively. As the training progresses, the parameters are randomized over time and sampled according to a uniform distribution with bounds that approach gradually the ranges of Table 3.4. By incrementally incorporating randomization into the training procedure, the adoption of conservative policies is avoided

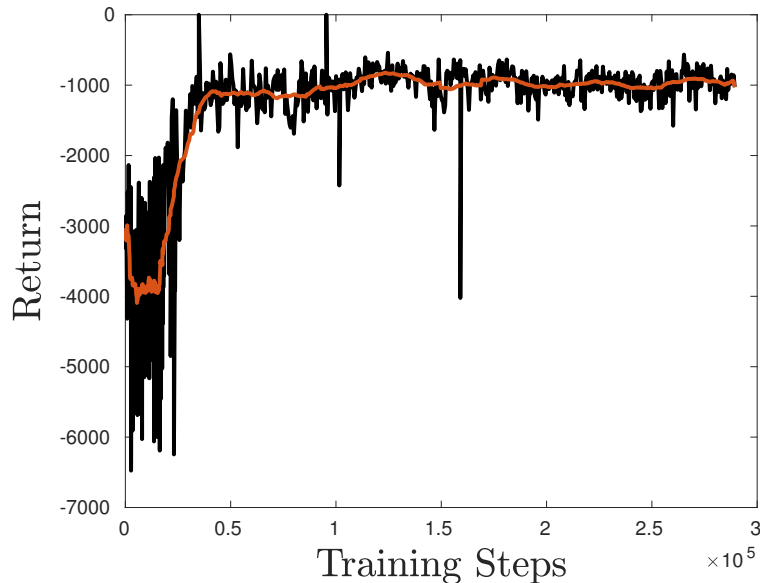


FIGURE 3.10: The cumulative reward and the corresponding moving average throughout the training procedure, including domain randomization.

[50]. The randomized parameters are sampled at the beginning of each episode and are maintained constant throughout it, except for the observation noise in the cable state which is added at every time step. Eventually, the evolution of the cumulative reward throughout the training, based again on the TD3 algorithm, is demonstrated in the learning curve of Figure 3.10.

3.6 Nonlinear Model Predictive Control

In this section, we introduce a model-based approach, as opposed to the model-free strategies of Sections 3.4 and 3.5, for controlling the UAV with the cable-suspended load. More specifically, according to the optimal control theory, we develop a nonlinear Model Predictive Control (NMPC) scheme, formulated for the system dynamics, in order to achieve the swing-free transportation of the suspended load. In general, NMPC has been proven to be an efficient tool for controlling multirotor UAVs in the context of various applications, e.g., [14], [61], and [62].

However, especially when the UAV with the cable-suspended load operates in outdoor environments, external disturbances, mainly arising from the encountered environmental conditions, are acting on the platform. The aforementioned disturbances, along with the assumptions in the system modelling, result in mismatches between the nominal and actual system dynamics which affect the performance of model-based controllers, particularly NMPC schemes. To this end, we leverage Gaussian Processes (GPs) in order to learn online and in a non-parametric manner the model discrepancies that exist between the actual and nominal system dynamics [32], [38]. In order to guarantee real-time performance, given the computational demands

of GPs, and also ensure adaptability to the conditions prevailing in the outdoor environment, a weighted version of sparse GP regression is exploited. Eventually, the approximated dynamics are integrated into a learning-based NMPC scheme so as to enhance the control performance.

3.6.1 Nominal NMPC

In order to transport the load safely and with minimum oscillations, an NMPC is initially formulated for the nonlinear system dynamics described in Equation (2.11). Hence, the implementation of the NMPC scheme requires, besides the state of the vehicle and the tension $\|\mathbf{T}\|$ of the cable, an estimate of the cable state, i.e., angles $\boldsymbol{\eta}_L$ and angular velocity $\boldsymbol{\omega}_L$, and a reference position \mathbf{p}_{ref} . As mentioned in Section 2.3, the NMPC scheme is deployed as a cascade position control which computes attitude commands for the inner attitude controller of the autopilot.

Since the objective of the NMPC scheme is to suppress cable oscillations and to minimize the distance between the actual position of the UAV and the reference one while simultaneously satisfying input constraints, the following Optimal Control Problem (OCP) is formulated:

$$\begin{aligned} \min_{\mathbf{v}} \int_{t_0}^{t_0+T} (\|\mathbf{x}(t) - \mathbf{x}_{ref}(t)\|_{\mathbf{Q}}^2 + \|\mathbf{v}(t)\|_{\mathbf{R}}^2) dt + \|\mathbf{x}(t_0 + T) - \mathbf{x}_{ref}(t_0 + T)\|_{\mathbf{P}}^2 \\ \text{subject to: } \mathbf{x}(t_0) = \mathbf{x}_0 \\ \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{v}) \text{ (Equation (2.11))} \\ \mathbf{v} \in \mathbb{U} \end{aligned} \quad (3.15)$$

where $\mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T \ \phi \ \theta \ \psi \ \boldsymbol{\eta}_L^T \ \boldsymbol{\omega}_L^T]^T \in \mathbb{R}^{13}$, $\mathbf{v} = [\phi_d \ \theta_d \ \psi_d \ v_{z_d}]^T \in \mathbb{R}^4$ are the state and input vectors, $\mathbf{x}_{ref} = [\mathbf{p}_{ref}^T \ \mathbf{0}_{1 \times 3} \ 0 \ 0 \ \psi_{ref} \ \mathbf{0}_{1 \times 2} \ \mathbf{0}_{1 \times 2}]^T \in \mathbb{R}^{13}$ is the reference state vector, T is the time horizon of the NMPC optimization problem, \mathbf{Q} is the state error cost matrix, \mathbf{R} is the control input cost matrix, \mathbf{P} is the terminal state error cost matrix, and \mathbb{U} is the set of input constraints related to roll, pitch and vertical velocity limits. The cost matrices \mathbf{Q} , \mathbf{R} , \mathbf{P} are positive-definite and block-diagonal and, for instance, the state error cost matrix \mathbf{Q} is structured as follows:

$$\mathbf{Q} = \text{diag} \left(\left[\mathbf{Q}_p \ \mathbf{Q}_v \ \mathbf{Q}_\eta \ \mathbf{Q}_{\eta_L} \ \mathbf{Q}_{\omega_L} \right] \right) \quad (3.16)$$

where \mathbf{Q}_p , \mathbf{Q}_v , \mathbf{Q}_η , \mathbf{Q}_{η_L} , \mathbf{Q}_{ω_L} are block-diagonal weight matrices that penalize the deviation of the vehicle's position, velocity, orientation, and cable angles, and angular velocity respectively from the reference state vector. The terminal state error cost matrix \mathbf{P} is defined in a similar way.

The aforementioned OCP is solved at each iteration and an optimal control sequence is computed. Only the first control input of the derived sequence is applied to the UAV and the whole optimization process is repeated according to the receding

horizon control principle. In order to design the MPC problem and achieve real-time performance, the ACADO Toolkit [33] is utilized combined with the qpOASES [17] solver. Based on experiments conducted on the embedded computer of the UAV, namely the Jetson AGX Xavier, it was found that the ACADO Toolkit with the qpOASES solver provides a solution to the OCP after 3.4 ms on average and, hence, a highly satisfactory control frequency is attained.

3.6.2 Augmented Dynamics

In the above section, the nominal dynamics of the system $\hat{\mathbf{x}}_{k+1} = \mathbf{f}_{nom}(\mathbf{x}_k, \mathbf{v}_k)$ were exploited, where \mathbf{f}_{nom} the discrete-time equivalent model of Equation (2.11), and $\hat{\mathbf{x}}_{k+1}$ the estimate of the state at the next time instant t_{k+1} . Although the aforementioned model is adequate for the control of the system, the performance can be further improved since the nominal model does not capture the entire range of the actual system dynamics. Consequently, discrepancies occur between the measured and the estimated next state, and an error $\mathbf{e}_k = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}$ is observed originating from the following reasons:

- The drag forces, induced by the air and applied to both the vehicle and the load, are neglected in the nominal dynamics.
- The flexibility of the cable may violate the assumption of tautness (Equation (2.2)) and, thus, the subsequent analysis of Section 2.5.1.
- The upper edge of the cable is assumed to coincide with the CoM of the vehicle; however, this is practically infeasible due to the limitations that emerge from the mechanical structure of the UAV.
- The integration of an embedded computer and sensors, e.g., a camera, onto the UAV may result in a displacement of the CoM of the vehicle.
- Other discrepancies may exist, e.g., errors in the calibration of the load cell that measures the cable tension.

The incorporation of the aforementioned discrepancies into the nominal dynamics not only increases the complexity of the model but also entails exhaustive and possibly expensive system identification procedures, e.g., aerodynamic testing in wind tunnels. Additionally, some sources of error are subject to the conditions prevailing in the outdoor environment where the UAV operates, e.g. wind velocity, and, hence, cannot be predicted.

To this end, we aim to exploit GPs to estimate the unmodeled dynamics in a non-parametric manner. Inspired by [32], [38], the actual system dynamics can be represented by the equation:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{v}_k) \\ &= \mathbf{f}_{nom}(\mathbf{x}_k, \mathbf{v}_k) + \mathbf{B}_d(\mathbf{d}(\mathbf{x}_k, \mathbf{v}_k) + \mathbf{w}_k) \end{aligned} \quad (3.17)$$

where $\mathbf{w}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}^w)$ is i.i.d. Gaussian noise with a diagonal variance matrix $\boldsymbol{\Sigma}^w$, \mathbf{d} denotes the unknown dynamic part to be approximated by the GPs, and \mathbf{B}_d defines the subspace of the system dynamics affected by the model discrepancies. It is assumed that model errors exist solely in the second-order system dynamics and, thus, $\mathbf{B}_d = [\mathbf{0}_{3 \times 3}; \mathbf{I}_{3 \times 3}; \mathbf{0}_{3 \times 3}; \mathbf{0}_{2 \times 2}; \mathbf{I}_{2 \times 2}]$. Consequently, the ultimate goal is to estimate the model errors $\mathbf{d}_v \in \mathbb{R}^3$ and $\mathbf{d}_\omega \in \mathbb{R}^2$ that impact on the translational dynamics of the UAV and the angular motion of the cable respectively.

3.6.3 Gaussian Process Regression

In the following section, GP regression is briefly introduced according to the thorough description in [72]. In general, GPs can be exploited for the approximation of an unknown function $g(\mathbf{z}) : \mathbb{R}^{n_z} \rightarrow \mathbb{R}$, based on a collection of M noisy observations y_j of the true function $g(\mathbf{z})$ measured at the inputs \mathbf{z}_j , with:

$$y_j = g(\mathbf{z}_j) + w_j, \quad w_j \sim \mathcal{N}(0, \sigma^2) \quad (3.18)$$

More specifically, a dataset D is constructed, i.e.:

$$D = \left\{ \begin{aligned} \mathbf{Z} &= [\mathbf{z}_1, \dots, \mathbf{z}_M]^T \in \mathbb{R}^{M \times n_z}, \\ \mathbf{y} &= [y_1, \dots, y_M]^T \in \mathbb{R}^M \end{aligned} \right\} \quad (3.19)$$

which can be exploited in order to predict the posterior distribution at any query input \mathbf{z} . Conditioning on the dataset D , the posterior distribution is specified by the following mean and covariance:

$$\mu(\mathbf{z}) = \mathbf{k}_{zZ}(\mathbf{K}_{ZZ} + \mathbf{I}_{M \times M} \sigma^2)^{-1} \mathbf{y} \quad (3.20)$$

$$\Sigma(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) - \mathbf{k}_{zZ}(\mathbf{K}_{ZZ} + \mathbf{I}_{M \times M} \sigma^2)^{-1} \mathbf{k}_{Zz} \quad (3.21)$$

where $k(\mathbf{z}_i, \mathbf{z}_j) = \sigma_f^2 \exp(-0.5(\mathbf{z}_i - \mathbf{z}_j)^T \mathbf{L}(\mathbf{z}_i - \mathbf{z}_j)) \in \mathbb{R}$ is the squared exponential kernel function with positive diagonal length scale matrix $\mathbf{L} \in \mathbb{R}^{n_z \times n_z}$ and variance σ_f^2 , $[\mathbf{K}_{ZZ}]_{i,j} = k(\mathbf{z}_i, \mathbf{z}_j)$ is the i, j element of the matrix $\mathbf{K}_{ZZ} \in \mathbb{R}^{M \times M}$ with $\mathbf{z}_i, \mathbf{z}_j \in \mathbf{Z}$, $[\mathbf{k}_{zZ}]_j = k(\mathbf{z}, \mathbf{z}_j)$ is the j element of the row vector $\mathbf{k}_{zZ} \in \mathbb{R}^M$ with $\mathbf{z}_j \in \mathbf{Z}$, and $\mathbf{k}_{Zz} = \mathbf{k}_{zZ}^T$.

However, the computational cost of GP regression, mainly determined by the inversion of the \mathbf{K}_{ZZ} matrix, is highly dependent on the size M of the dataset and renders its usage impractical for real-time applications where a significant amount of data is considered. Additionally, it should be noted that GPs can approximate solely one-dimensional functions and, hence, 5 independent GPs are required to estimate the model errors $\mathbf{d}_v \in \mathbb{R}^3$ and $\mathbf{d}_\omega \in \mathbb{R}^2$ for the purposes of the studied application. Consequently, since the dataset should be updated online – thus, pre-computations are infeasible – and real-time performance is a prerequisite for the successful control

of the vehicle, an approximation technique should be employed instead so as to learn the model discrepancies.

Sparse Gaussian Process Regression

In order to alleviate the computational demands of GP regression and achieve real-time performance, sparse GP regression is employed. The concept of sparse GP regression lies in the usage of a subset of the original dataset, consisting of m inducing points with $m \ll M$, in order to increase the computational efficiency of GP regression for large datasets while simultaneously providing reasonable predictions. Based on [87], assuming a set of inducing points $\mathbf{Z}_{ind} = [\mathbf{z}_1, \dots, \mathbf{z}_m]^T \in \mathbb{R}^{m \times n_z}$, the posterior distribution is approximated by the following mean and covariance:

$$\mu(\mathbf{z}) = \mathbf{k}_{zZ_{ind}} \mathbf{K}_{Z_{ind}Z_{ind}}^{-1} \mu_m \quad (3.22)$$

$$\Sigma(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) - \mathbf{k}_{zZ_{ind}} \mathbf{K}_{Z_{ind}Z_{ind}}^{-1} \mathbf{k}_{Z_{ind}z} + \mathbf{k}_{zZ_{ind}} \Sigma \mathbf{k}_{Z_{ind}z} \quad (3.23)$$

where $\Sigma = (\mathbf{K}_{Z_{ind}Z_{ind}} + \sigma^{-2} \mathbf{K}_{Z_{ind}Z} \mathbf{K}_{ZZ_{ind}})^{-1}$, $\mu_m = \sigma^{-2} \mathbf{K}_{Z_{ind}Z_{ind}} \Sigma \mathbf{K}_{Z_{ind}Z} \mathbf{y}$, $[\mathbf{K}_{Z_{ind}Z_{ind}}]_{i,j} = k(\mathbf{z}_i, \mathbf{z}_j)$ is the i, j element of the matrix $\mathbf{K}_{Z_{ind}Z_{ind}} \in \mathbb{R}^{m \times m}$ with $\mathbf{z}_i, \mathbf{z}_j \in \mathbf{Z}_{ind}$, $[\mathbf{K}_{Z_{ind}Z}]_{i,j} = k(\mathbf{z}_i, \mathbf{z}_j)$ is the i, j element of the matrix $\mathbf{K}_{Z_{ind}Z} \in \mathbb{R}^{m \times M}$ with $\mathbf{z}_i \in \mathbf{Z}_{ind}$ and $\mathbf{z}_j \in \mathbf{Z}$, $\mathbf{K}_{ZZ_{ind}} = \mathbf{K}_{Z_{ind}Z}^T$, $[\mathbf{k}_{zZ_{ind}}]_j = k(\mathbf{z}, \mathbf{z}_j)$ is the j element of the row vector $\mathbf{k}_{zZ_{ind}} \in \mathbb{R}^m$ with $\mathbf{z}_j \in \mathbf{Z}_{ind}$, and $\mathbf{k}_{Z_{ind}z} = \mathbf{k}_{zZ_{ind}}^T$.

By employing the above approximation technique, the computational complexity scales as $\mathcal{O}(Mm^2)$, thus rendering the application of sparse GP models feasible for real-time systems and online learning, in contrast to the intractable standard GP regression (Equations 3.20 and 3.21) with complexity $\mathcal{O}(M^3)$.

Weighted & Sparse Gaussian Process Regression

However, throughout the experiments, the UAV operates in non-stationary outdoor environments with diverse and variable environmental conditions, e.g., wind speed, which influence the system dynamics and, hence, result in the time variation of the model discrepancies \mathbf{d}_v and \mathbf{d}_ω . Consequently, during the learning of the model errors, adaptability to the outdoor conditions is required in order to capture the temporal variation of the approximated functions. To this end, weighted GP regression is employed according to [11], but appropriately adjusted to the case of sparse GP regression, thus ensuring both adaptability and computational efficiency.

More precisely, a weight $w_i \in \mathbb{R}^+$ is assigned to each measurement y_i with the ultimate goal of boosting the more recent measurements during the GP regression. Towards this direction, the matrices $\mathbf{W} = \text{diag}(\sqrt{w_1}, \dots, \sqrt{w_M}) \in \mathbb{R}^{M \times M}$ and $\mathbf{W}_{ind} = \text{diag}(\sqrt{w_1}, \dots, \sqrt{w_m}) \in \mathbb{R}^{m \times m}$, related to the whole dataset and the inducing points respectively, are introduced. By properly modifying the kernel vectors and matrices, the posterior distribution for the case of weighted and sparse GP regression

is characterized by a structure similar to Equations 3.22 and 3.23:

$$\mu(\mathbf{z}) = \tilde{\mathbf{k}}_{zZ_{ind}} \tilde{\mathbf{K}}_{Z_{ind}Z_{ind}}^{-1} \tilde{\mu}_m \quad (3.24)$$

$$\Sigma(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) - \tilde{\mathbf{k}}_{zZ_{ind}} \tilde{\mathbf{K}}_{Z_{ind}Z_{ind}}^{-1} \tilde{\mathbf{k}}_{Z_{ind}z} + \tilde{\mathbf{k}}_{zZ_{ind}} \tilde{\Sigma} \tilde{\mathbf{k}}_{Z_{ind}z} \quad (3.25)$$

where $\tilde{\Sigma} = \left(\tilde{\mathbf{K}}_{Z_{ind}Z_{ind}} + \sigma^{-2} \tilde{\mathbf{K}}_{Z_{ind}Z} \tilde{\mathbf{K}}_{ZZ_{ind}} \right)^{-1}$, $\tilde{\mathbf{y}} = \mathbf{W}\mathbf{y}$, $\tilde{\mu}_m = \sigma^{-2} \tilde{\mathbf{K}}_{Z_{ind}Z_{ind}} \tilde{\Sigma} \tilde{\mathbf{K}}_{Z_{ind}Z} \tilde{\mathbf{y}}$, $\tilde{\mathbf{k}}_{zZ_{ind}} = \mathbf{k}_{zZ_{ind}} \mathbf{W}_{ind}$, $\tilde{\mathbf{K}}_{Z_{ind}Z_{ind}} = \mathbf{W}_{ind} \mathbf{K}_{Z_{ind}Z_{ind}} \mathbf{W}_{ind}^T$, and $\tilde{\mathbf{K}}_{Z_{ind}Z} = \mathbf{W}_{ind} \mathbf{K}_{Z_{ind}Z} \mathbf{W}$. It is mentioned that if all the weights are equal to 1, the aforementioned mean and covariance are identical to the corresponding values of Equations 3.22 and 3.23.

Practical Implementation

Herein, a more in-depth explanation is provided regarding the implementation of weighted and sparse GP regression for the case of aerial transportation of cable-suspended loads.

1) *Dataset Update*: Throughout each experiment, the dataset D is initially either empty or filled with offline observations from previous experiments. Once a new measurement is collected during the operation of the UAV, D is updated online; however, the size of the dataset is restricted in order to guarantee the computational feasibility of GP regression over time. Hence, when the dataset D reaches a maximum size of M data points, the incoming measurement $\{\mathbf{z}', y'\}$ replaces a data point currently in D according to the following criterion:

$$\min_{i \in D} \left(\alpha \|\mathbf{z}' - \mathbf{z}_i\|^2 + \beta w_i^2 \right) \quad (3.26)$$

where $\alpha, \beta \in \mathbb{R}^+$ are tuning parameters. By employing the aforementioned strategy, the substitution of older measurements, similar to the incoming one, is encouraged, thus ensuring that the dataset is up-to-date and covers adequately the input space.

2) *Weight Selection*: Each measurement $\{\mathbf{z}_i, y_i\}$ in the dataset is associated with a weight w_i computed according to the time instant t_i the measurement was collected:

$$w_i = k_0 + c \cdot \tanh(k_1 t_i) \quad (3.27)$$

with $k_0, c, k_1 \in \mathbb{R}^+$. The choice of the above function satisfies the requirement to boost more recent measurements and enhance the adaptability of the learning procedure to varying conditions.

3) *Selection of inducing points*: The choice of the set \mathbf{Z}_{ind} may be critical for the accuracy of the sparse GP regression. Given a test input \mathbf{z} , a naive approach would be to pick the closest points as inducing ones, i.e., the m points $\mathbf{z}_i \in D$ with the minimum distance $\|\mathbf{z} - \mathbf{z}_i\|$. However, this approach may lead to numerical instability in the inversion of the matrix $\tilde{\mathbf{K}}_{Z_{ind}Z_{ind}}$. Hence, a probability is assigned

to each measurement instead, with lower values of the distance $\|\mathbf{z} - \mathbf{z}_i\|$ corresponding to a higher probability, and m points are selected randomly given this distribution.

4) *Derivative Approximation:* Another advantage inherent in GPs is the possibility to make predictions about the derivatives of the learned function $g(\mathbf{z})$ with respect to the input variables $\mathbf{z} = [z_1, \dots, z_{n_z}]^T$. The calculation of the derivatives solely entails the differentiability of the kernel function. Consequently, considering that the selected squared exponential kernel function is infinitely differentiable, the gradient $\nabla\mu(\mathbf{z}) \in \mathbb{R}^{n_z}$ and hessian $\mathbf{H}(\mathbf{z}) \in \mathbb{R}^{n_z \times n_z}$ of the function g at a test input \mathbf{z} can be approximated by the following equations:

$$\nabla\mu(\mathbf{z}) = \left[\frac{\partial\mu}{\partial z_1}, \dots, \frac{\partial\mu}{\partial z_{n_z}} \right]^T = \left[\frac{\partial\tilde{\mathbf{k}}_{zZ_{ind}}}{\partial z_1}, \dots, \frac{\partial\tilde{\mathbf{k}}_{zZ_{ind}}}{\partial z_{n_z}} \right]^T \tilde{\mathbf{K}}_{Z_{ind}Z_{ind}}^{-1} \tilde{\boldsymbol{\mu}}_m \quad (3.28)$$

$$\mathbf{H}(\mathbf{z}) = \begin{bmatrix} \frac{\partial^2\tilde{\mathbf{k}}_{zZ_{ind}}}{\partial z_1^2} & \dots & \frac{\partial^2\tilde{\mathbf{k}}_{zZ_{ind}}}{\partial z_1 \partial z_{n_z}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2\tilde{\mathbf{k}}_{zZ_{ind}}}{\partial z_1 \partial z_{n_z}} & \dots & \frac{\partial^2\tilde{\mathbf{k}}_{zZ_{ind}}}{\partial z_{n_z}^2} \end{bmatrix} \tilde{\mathbf{K}}_{Z_{ind}Z_{ind}}^{-1} \tilde{\boldsymbol{\mu}}_m \quad (3.29)$$

It is noted that $\tilde{\mathbf{K}}_{Z_{ind}Z_{ind}}^{-1}$ and $\tilde{\boldsymbol{\mu}}_m$ are already calculated during the inference of GP regression, and thus, the computational cost of Equations 3.28 and 3.29 is minor. Higher-order derivatives can also be computed; however, the approximation error is increased considering solely the measurements \mathbf{y} of the function, i.e., without the addition of derivative observations in the GP regression.

5) *Measurements:* As aforementioned, the ultimate goal is to approximate the model discrepancies \mathbf{d}_v and \mathbf{d}_ω using 5 independent GPs. However, the unmodelled disturbances, e.g., drag forces, are physically expressed w.r.t. the body-fixed frame \mathbf{B} . To this end, the training procedure is adjusted so as to learn the model errors ${}^B\mathbf{d}_v$ and ${}^B\mathbf{d}_\omega$, defined in \mathbf{B} . Hence, consistency in the collected dataset is ensured, the dependency on the orientation of the vehicle, especially the yaw angle, is eliminated, and the dimension of the input space, and hence, the computational complexity, is reduced.

Towards this direction, we introduce the velocity of the UAV w.r.t. \mathbf{B} , i.e., ${}^B\mathbf{v} = \mathbf{R}_{WB}^T \mathbf{v} = [{}^Bv_x \quad {}^Bv_y \quad {}^Bv_z]^T$, as well as the cable angles ${}^B\boldsymbol{\eta}_L = [{}^B\phi_L \quad {}^B\theta_L]^T$ and angular velocity ${}^B\boldsymbol{\omega}_L = {}^B\dot{\boldsymbol{\eta}}_L$ computed by the cable unit vector ${}^B\mathbf{n} = \mathbf{R}_{WB}^T \mathbf{n}$ and velocity ${}^B\dot{\mathbf{n}} = \mathbf{R}_{WB}^T \dot{\mathbf{n}}$ (defined similarly to Equations 2.2 and 2.3 respectively). During the deployment of the UAV, measurements are collected using the nominal model of the system, presented in Section 2.5.1, while the inputs \mathbf{z} for each component of the approximated vectors ${}^B\mathbf{d}_v$ and ${}^B\mathbf{d}_\omega$ are selected based on physical observations, as summarized in Table 3.5. It is mentioned that the selection of appropriate inputs is critical for successfully learning unknown functions with GP models. Once the predictions of ${}^B\mathbf{d}_v$ and ${}^B\mathbf{d}_\omega$ are available, the transformation in \mathbf{W} is conversely realized. Finally, it should be noted that some model discrepancies share the same

input vector and, thus, certain kernel matrix calculations are avoided resulting in the further reduction of the computational cost.

Model Error		Measurement y	Input \mathbf{z}
${}^B\mathbf{d}_v$	${}^B d_{v,x}$	$\frac{{}^B\mathbf{v}_{k+1} - {}^B\hat{\mathbf{v}}_{k+1}}{dt}$	${}^B v_x, {}^B\theta_L, {}^B\dot{\theta}_L$
	${}^B d_{v,y}$		${}^B v_y, {}^B\phi_L, {}^B\dot{\phi}_L$
	${}^B d_{v,z}$		${}^B v_z$
${}^B\mathbf{d}_\omega$	${}^B d_{\omega,\phi}$	$\frac{{}^B\boldsymbol{\omega}_{L,k+1} - {}^B\hat{\boldsymbol{\omega}}_{L,k+1}}{dt}$	${}^B v_y, {}^B\phi_L, {}^B\dot{\phi}_L$
	${}^B d_{\omega,\theta}$		${}^B v_x, {}^B\theta_L, {}^B\dot{\theta}_L$

TABLE 3.5: The structure of the dataset D . Measurements are obtained using the difference between the actual next state and the predicted one, according to the nominal system dynamics, divided by the time step dt . The inputs \mathbf{z} for each component of the approximated vectors ${}^B\mathbf{d}_v$ and ${}^B\mathbf{d}_\omega$ are also depicted.

6) *Computational Cost*: Regarding the computational cost of the weighted and sparse GP regression, the mean execution time for the inference of the 5 GPs, considering a dataset of $M = 1000$ measurements and $m = 30$ inducing points for each GP, is equal to 30.9 ms on the embedded Jetson AGX Xavier.

3.6.4 GP-based NMPC

In order to achieve the transportation of the multirotor towards reference positions $\mathbf{p}_{ref} \in \mathbb{R}^3$ with simultaneous minimization of the cable angular motion, despite the disturbances acting on the platform, an NMPC is formulated for the augmented system dynamics which includes the – approximated by the weighted and sparse GP regression – model discrepancies \mathbf{d}_v and \mathbf{d}_ω . To this end, the following OCP is defined:

$$\begin{aligned}
& \min_{\mathbf{v}_0, \dots, \mathbf{v}_{N-1}} \sum_{k=0}^{N-1} \left((\mathbf{x}_k - \mathbf{x}_{ref})^T \mathbf{Q} (\mathbf{x}_k - \mathbf{x}_{ref}) + \mathbf{v}_k^T \mathbf{R} \mathbf{v}_k \right) \\
& \quad + (\mathbf{x}_N - \mathbf{x}_{ref})^T \mathbf{P} (\mathbf{x}_N - \mathbf{x}_{ref}) \\
& \text{s.t. } \mathbf{x}_{k+1} = \mathbf{f}_{nom}(\mathbf{x}_k, \mathbf{v}_k) + \mathbf{B}_d \mathbf{d}(\mathbf{x}_k, \mathbf{v}_k) \text{ (Equation (3.17))} \\
& \quad \mathbf{v}_k \in \mathbb{U}, \quad k = 0, \dots, N-1 \\
& \quad \mathbf{x}_0 = \mathbf{x}(0)
\end{aligned} \tag{3.30}$$

where N is the length of the discretized predicted horizon, and the remaining variables are defined according to Section 3.6.1. The aforementioned OCP is again solved within approximately 4 ms using the ACADO Toolkit and the qpOASES solver.

Regarding the unmodelled dynamics, the discrepancies $\mathbf{d}_{v,0}$ and $\mathbf{d}_{\omega,0}$ are predicted at each time instant t_0 using Equation (3.24), for each component of the vectors, and the corresponding input \mathbf{z}_0 , extracted by the current state \mathbf{x}_0 . A key

point of discussion revolves around the propagation of the learned dynamics throughout the predicted horizon of the NMPC. Possible solutions are either the evaluation of GPs throughout the whole horizon or keeping $\mathbf{d}_{v,0}$ and $\mathbf{d}_{\omega,0}$ constant during the optimization procedure. However, the former solution is computationally intractable whereas the latter implies lower accuracy. Towards this direction, a computationally lightweight second-order local approximation is employed around the current time instant t_0 :

$$\mu_k(\mathbf{z}_k) = \mu_0(\mathbf{z}_0) + \nabla\mu_0^T(\mathbf{z}_0)(\mathbf{z}_k - \mathbf{z}_0) + \frac{1}{2}(\mathbf{z}_k - \mathbf{z}_0)^T \mathbf{H}_0(\mathbf{z}_0)(\mathbf{z}_k - \mathbf{z}_0) \quad (3.31)$$

where the gradient and the Hessian matrix are computed according to Equations 3.28 and 3.29 respectively. Consequently, at each iteration of the NMPC, only one evaluation of the individual GPs is required.

3.7 Experimental Results

In this section, the performance of the proposed control methodologies, namely (i) the deep RL-based policy, (ii) the robust deep RL-based policy, (iii) the nominal NMPC, and (iv) the GP-based NMPC, is evaluated through simulation and real experiments in outdoor environments. During the real-world experiments, the feedback of the cable state is provided by either the frame-based or the event-based solution, presented in Sections 3.3.1 and 3.3.2 respectively.

3.7.1 Experimental Setup

Throughout the outdoor experiments, the octorotor vehicle of Figure 2.3 was utilized. All the developed algorithms, implemented in either Python or C++, are based on the Robot Operating System (ROS) and run on the onboard Jetson AGX Xavier computer, which stands out for its high performance and is suitable for drone applications where size, weight and power consumption are crucial. As mentioned in Sections 2.2 and 2.3, the embedded computer interfaces with the flight controller using the MAVLink protocol, while the communication between ROS and the autopilot is achieved through the MAVROS node. Additionally, a 5 kg load cell, located upon the cable, along with the HX711 amplifier, are integrated in order to measure the norm of the cable tension $\|\mathbf{T}\|$, which is required for the frame-based estimation of the cable state and the NMPC schemes. Moreover, an Arduino Mega [57] is mounted on the UAV in order to transfer the measurements provided by the load cell to the onboard computer through serial communication. Finally, the vehicle is equipped with the ZED 2 stereocamera or the DAVIS 346, depending on whether the frame-based or the event-based estimation algorithm is employed.

3.7.2 Deep RL Control Experiments

Simulation Experiments

Initially, a comparative study is presented in the Gazebo simulator between the default PID position controller of the autopilot (Figure 2.4), which is not designed for minimizing cable oscillations, and the control policy $\pi_\theta(s)$, trained according to Section 3.4. In both cases, the same 8 waypoints, $\mathbf{p}_{ref|i} = [x_{ref} \ y_{ref} \ z_{ref}]^T | i \in \mathbb{R}^3$ for $i = 1, \dots, 8$, were sent successively as reference to the controllers. In order to quantify the swinging motion of the cable during the navigation of the vehicle, the deviation of the cable unit vector \mathbf{n} from the desired vertical configuration, was computed by the angle:

$$\beta = \cos^{-1}(|n_z|) \quad (3.32)$$

As illustrated in Figure 3.11d, the PID controller excited the swinging motion of the cable during the simulation scenario and, hence, notable values of the angle β were observed, which in case of a real-world experiment would jeopardize the safety of both the vehicle and the load. In contrast, the trained control policy successfully navigated the UAV towards the reference 3D waypoints, as depicted in Figures 3.11a, 3.11b and 3.11c, while maintaining the angle β at significantly lower levels (Figure 3.11d).

Real-World Experiments

Afterwards, a comparative experimental study between the default PID position controller of the autopilot and the control scheme of Figure 3.12 was conducted with the octorotor in order to evaluate the efficacy of the trained control policy $\pi_\theta(s)$ in real-world conditions. In this preliminary experiment, a box of 1.5 kg, suspended through a cable of 2.3 m length, was considered, while the detection of the load by the ZED camera was achieved with the aid of four ArUco markers [23], placed at the upper surface of the box. The same 8 waypoints were commanded consecutively as reference to both controllers. It is highlighted that the experiments were realized successively, at the same location and hour, and, thus, the environmental conditions, especially the wind speed, could be assumed as identical.

Initially, the default PID position controller provided by the Ardupilot side, which does not compensate for the cable oscillations, was utilized. Once the vehicle moved towards the second waypoint, the angle β exceeded the 60° and, hence, the load was not maintained inside the camera's field of view, as shown in the relevant video. For safety reasons, the experiment was terminated since the cable approached the vehicle's arms and, therefore, the safe completion of the target mission was not guaranteed.

In contrast, the control policy $\pi_\theta(s)$ efficiently drives the vehicle towards the reference 3D waypoints, as illustrated in Figure 3.13, while simultaneously a swing-free motion is achieved. More precisely, the angle β does not exceed the value of 15°

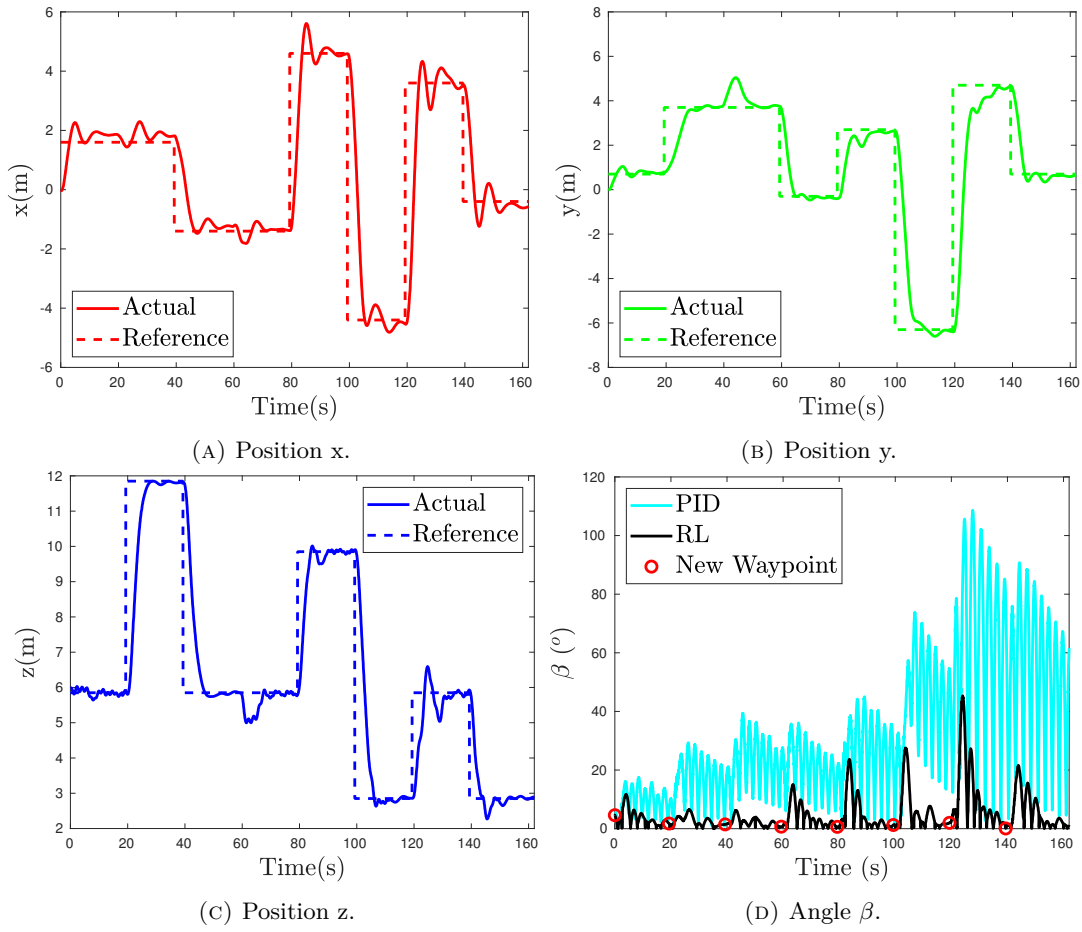


FIGURE 3.11: The position \mathbf{p} of the vehicle compared to the reference 3D waypoints \mathbf{p}_{ref} , while using the trained control policy, and the angle β during the simulation scenario.

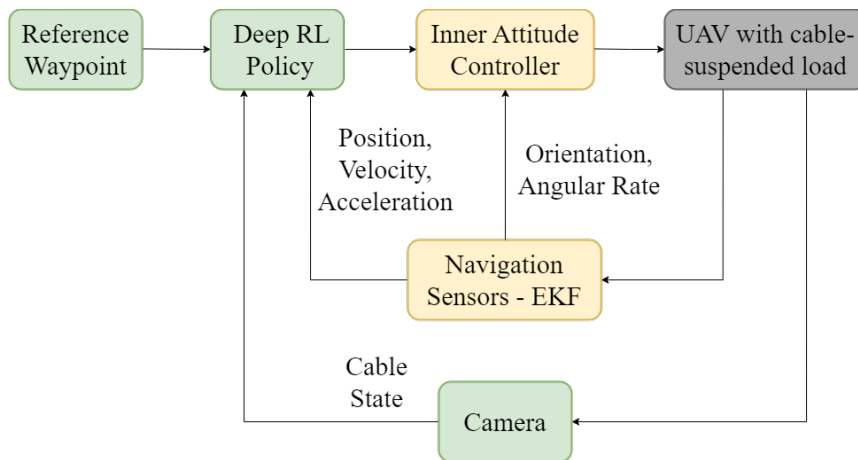


FIGURE 3.12: The deep RL-based control scheme.

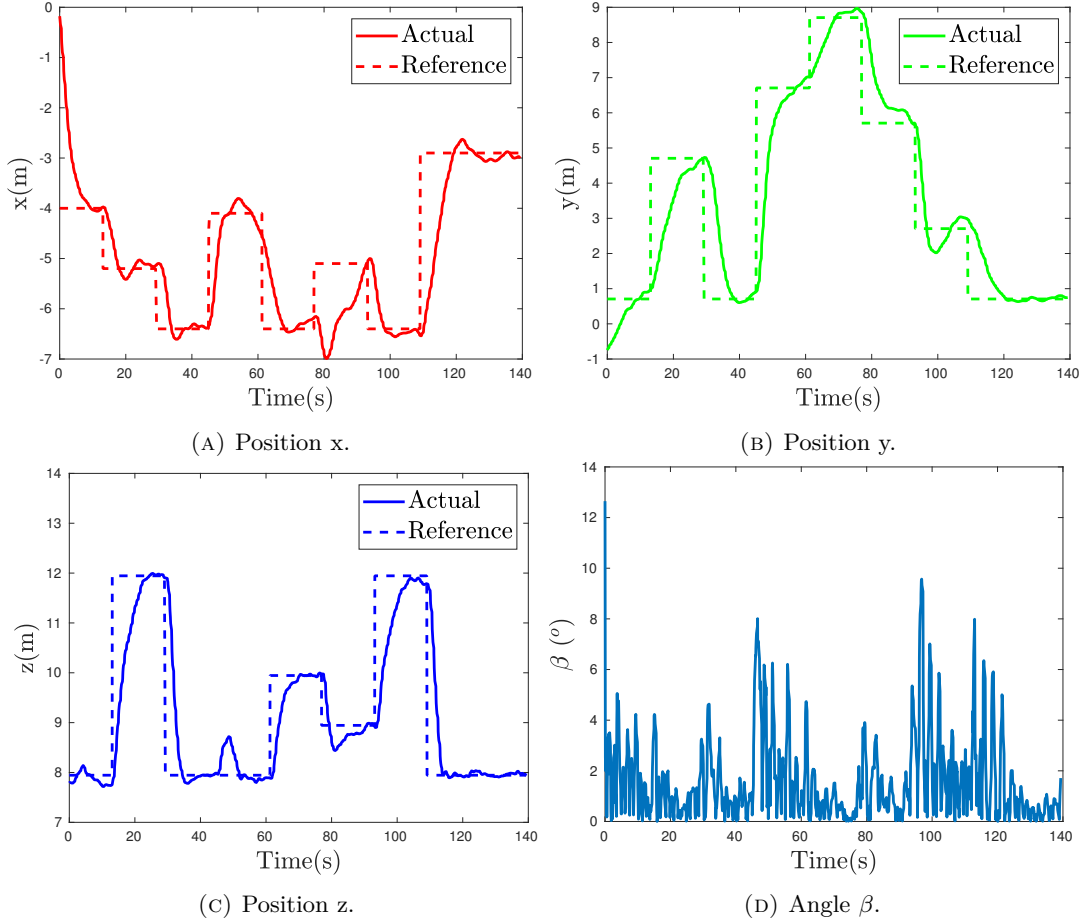


FIGURE 3.13: The position \mathbf{p} of the vehicle, compared to the reference 3D waypoints \mathbf{p}_{ref} , while using the trained control policy, and the angle β during the experimental scenario.

during the experiment (Figure 3.13d). The higher values of the angle are noticed when a new waypoint is commanded, which is a reasonable observation given the fact that, initially, the position error $\mathbf{p} - \mathbf{p}_{ref}$ is dominant in the learned Q function network. As the vehicle approaches the reference position, the swinging motion of the load, and specifically the angle β , is minimized, as depicted in Figure 3.13d.

The aforementioned comparative studies and an additional real-world experiment are better illustrated in the following link <https://youtu.be/7C843hts83E>.

3.7.3 Robust Deep RL Control Experiments

Simulation Experiments

Following the convergence of the training procedure in Section 3.5, where domain randomization was included, the learned policy $\pi_{\theta}(s)$ was extensively tested in the Gazebo Simulator in order to validate its robustness to varying cable lengths l , load masses m_L , and external disturbances \mathbf{F}_Q . More precisely, comparative studies were conducted considering the mission of Figure 3.14, which consists of 8 reference positions \mathbf{p}_{ref} . The ability of the policy to minimize the swinging motion of the cable

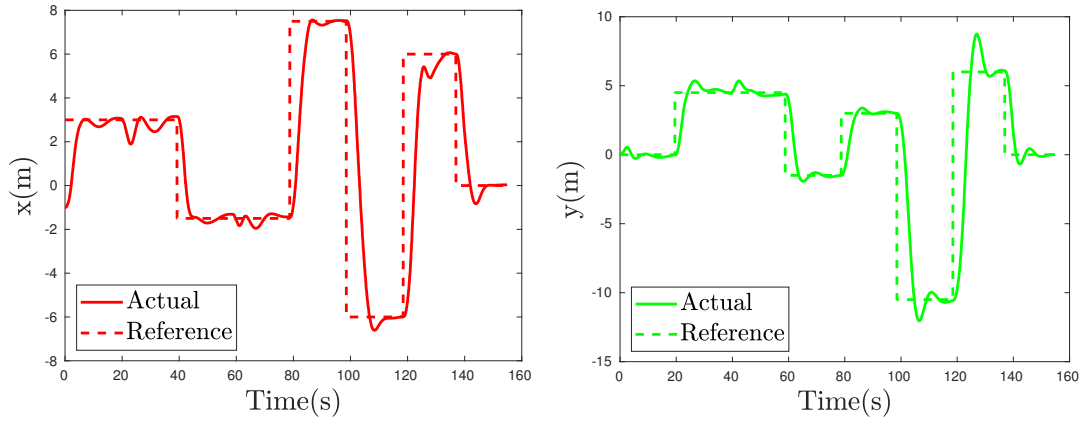


FIGURE 3.14: The reference positions \mathbf{p}_{ref} and the actual trajectory \mathbf{p} of the UAV in the Gazebo for $m_L = 0.5\text{kg}$ and $l = 1.0\text{m}$.

	Load Mass [kg]		
	$m_L = 0.25$	$m_L = 0.5$	$m_L = 1.0$
$\ \dot{\mathbf{n}}\ [s^{-1}]$	0.069	0.062	0.064
$\beta [^\circ]$	2.578	2.407	2.237

TABLE 3.6: The mean cable velocity $\|\dot{\mathbf{n}}\|$ and angle β for $l = 1.0\text{m}$ and different load masses m_L .

was measured based on the norm of the cable velocity $\|\dot{\mathbf{n}}\|$ and the angle β , defined in Equation (3.32). The following scenarios were examined:

1) *Robustness to load mass*: In this comparative study, the cable length was held constant at 1.0m and 3 different load masses, specifically $m_L = 0.25$, 0.5, and 1.0kg, were considered during each mission. The time evolution of the angle β throughout each case is demonstrated in Figure 3.15, while the mean velocity $\|\dot{\mathbf{n}}\|$ and angle β for the individual missions are depicted in Table 3.6.

2) *Robustness to cable length*: Conversely, in the subsequent scenario, the load mass was maintained fixed to 0.5kg and the cable length was set to $l = 0.5$, 1.5, and

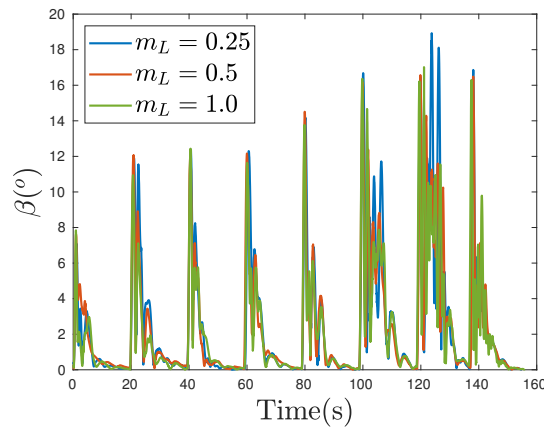


FIGURE 3.15: The angle β for various load masses m_L and constant cable length l during the 1st comparative study in the Gazebo.

	Cable Length [m]		
	$l = 0.5$	$l = 1.5$	$l = 2.0$
$\ \dot{\mathbf{n}}\ [s^{-1}]$	0.070	0.065	0.068
$\beta [^\circ]$	2.392	2.623	2.791

TABLE 3.7: The mean cable velocity $\|\dot{\mathbf{n}}\|$ and angle β for $m_L = 0.5\text{kg}$ and different cable lengths l .

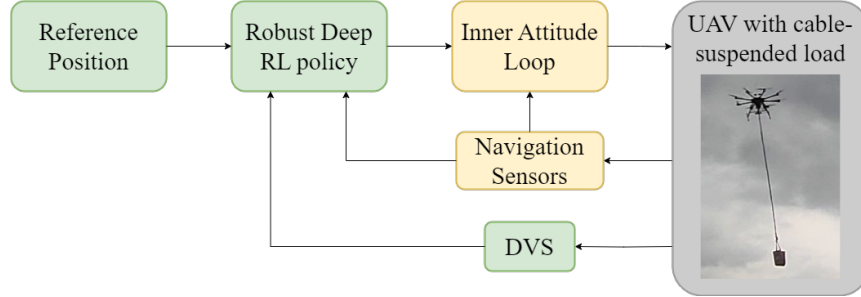


FIGURE 3.16: Block diagram of the proposed RL-based strategy including domain randomization.

2.0m. The swinging motion of the cable during each mission is characterized by the mean values of Table 3.7.

3) *Robustness to external forces*: In the previous studies, the quadrotor operated in the Gazebo environment without the existence of external disturbances. To this end, the last scenario involved the default values for the cable length and load mass, namely 1.0m and 0.5kg respectively, and random external forces \mathbf{F}_Q , with each component of the vector being uniformly sampled from the range $[-1.0, 1.0]$ at each time step. The mean values for the velocity $\|\dot{\mathbf{n}}\|$ and the angle β , according to 10 simulation experiments, were equal to $0.086s^{-1}$ and 2.574° correspondingly.

The above numerical results validate that the performance of the learned policy is both acceptable and consistent across different values of the system properties owing to the domain randomization.

Real-World Experiments

After the analysis in the Gazebo Simulator, the control scheme of Figure 3.16 was directly deployed on the octorotor of Figure 2.3 in order to evaluate performance of the policy, trained via domain randomization, in a real-world outdoor environment. The vehicle was equipped with the DAVIS 346 camera for estimating the cable state according to Section 3.3.2. A cable of $l = 3.8\text{m}$, which is outside of the training range, was integrated onto the UAV while loads were placed within a bag, located at the lower edge of the cable.

More precisely, a comparative study was conducted between 3 different load masses, namely $m_L = 0.3, 0.5, \text{ and } 0.9\text{kg}$. The same mission of 4 reference positions

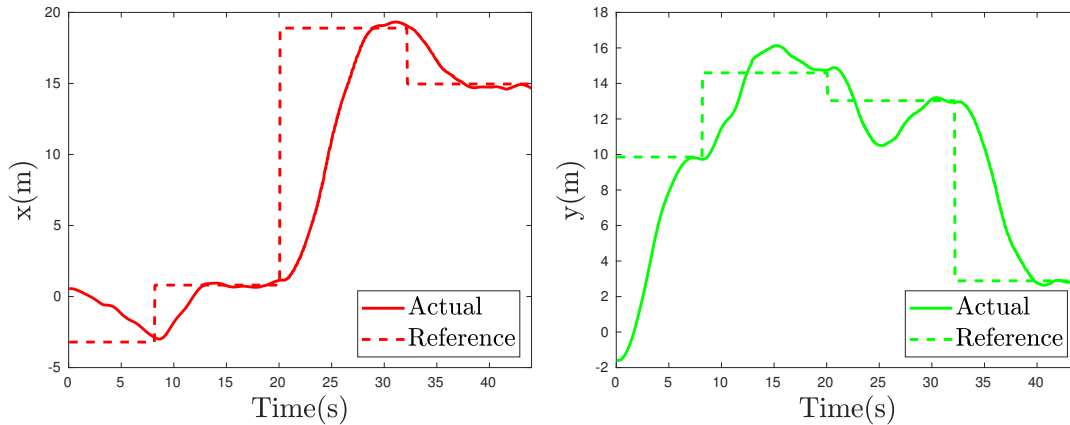


FIGURE 3.17: The reference positions \mathbf{p}_{ref} and the actual trajectory \mathbf{p} of the UAV during the real experiment for $m_L = 0.5$ kg.

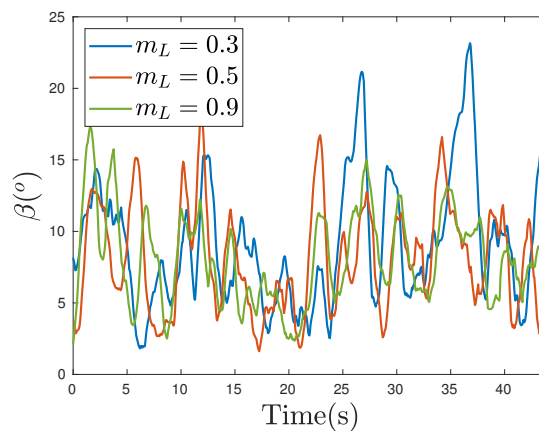


FIGURE 3.18: The angle β for various load masses m_L during the comparative study in the real-world environment.

was considered, as illustrated in Figure 3.17. Despite the external disturbances acting on the platform and the observation noise in the sensor-based estimation of the cable state, the learned policy minimized both the position error, e.g., Figure 3.17, and the cable angular motion, featuring also similar performance levels across the different masses, as depicted in Figure 3.18, where the angle β is demonstrated for each case. Consequently, the policy was able to successfully generalize to a real-world setting due to the domain randomization applied throughout the training procedure. The aforementioned experiments with the octorotor are better demonstrated in the following video <https://youtu.be/kbPSdWZiRAo>.

3.7.4 Nominal NMPC Experiments

Nominal NMPC with Frame-Based Feedback

Initially, the capability of the nominal NMPC scheme, presented in Section 3.6.1, to efficiently navigate the multicopter to the reference 3D waypoints, without exciting the swinging motion of the cable-suspended load, was investigated. As for the load, a sampling mechanism – that will be later exploited for conducting water sampling

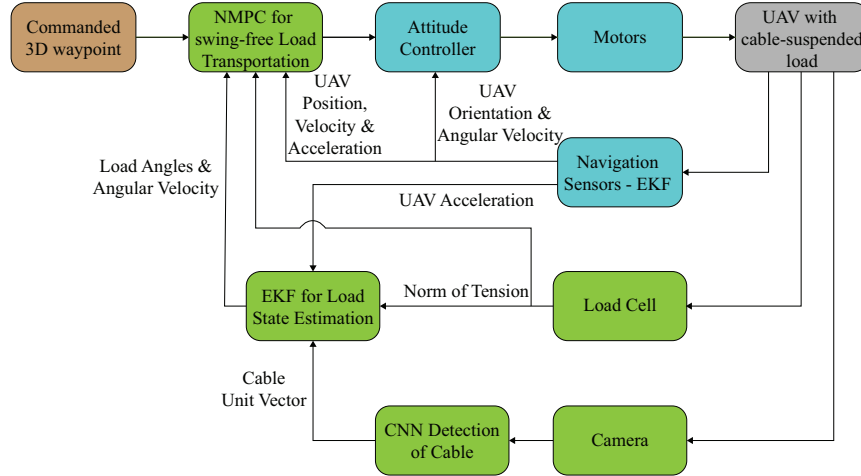


FIGURE 3.19: An overview of the deployed nominal NMPC scheme for the swing-free load transportation based on feedback provided by a frame-based camera.

in aquatic environments – was considered. The mass of the mechanism is equal to $m_L = 0.25$ kg, and it was suspended through a cable of length $l = 3.8$ m. Additionally, the ZED 2 stereocamera was integrated onto the vehicle and, thus, the frame-based algorithm of Section 3.3.1 was employed so as to provide feedback of the cable’s state, as depicted in Figure 3.19. In order to evaluate the performance of both the nominal NMPC and the CNN, 2 experiments were realized in different locations, particularly the NTUA campus and a beach location.

In both cases, a mission consisting of multiple waypoints was commanded. The deployed NMPC successfully minimized the error, $\mathbf{p}_{ref} - \mathbf{p}$, between the reference locations and the actual position of the UAV (Figures 3.20, 3.22), while simultaneously the cable angles $\boldsymbol{\eta}_L$ and angular velocity $\boldsymbol{\omega}_L$, computed by the CNN and the EKF, were maintained below critical levels (Figures 3.21, 3.23). Consequently, the safe transportation of the vehicle and the load was achieved and both missions were successfully completed, as illustrated in Figures 3.20d, 3.22d. It is also highlighted that the trained CNN successfully detects the cable and exhibits robustness to the different environments. Additionally, it is mentioned that, during both missions, the vehicle moved at a speed of up to 4 m/s.

Both experiments are better illustrated in the following videos <https://youtu.be/G-7rAuAFxHM> and <https://youtu.be/Km6nJYrVGoo>.

Nominal NMPC with Event-Based Feedback

Herein, the DAVIS 346 was incorporated into the octorotor platform with a $m_L = 0.5$ kg load and a $l = 3.8$ m cable, as depicted in Figure 3.3, in order to test the performance of the event-based estimation method of Section 3.3.2 against real-world and challenging conditions, commonly encountered in outdoor environments. Despite the increased number of events, generated by the motion of the vehicle and corresponding

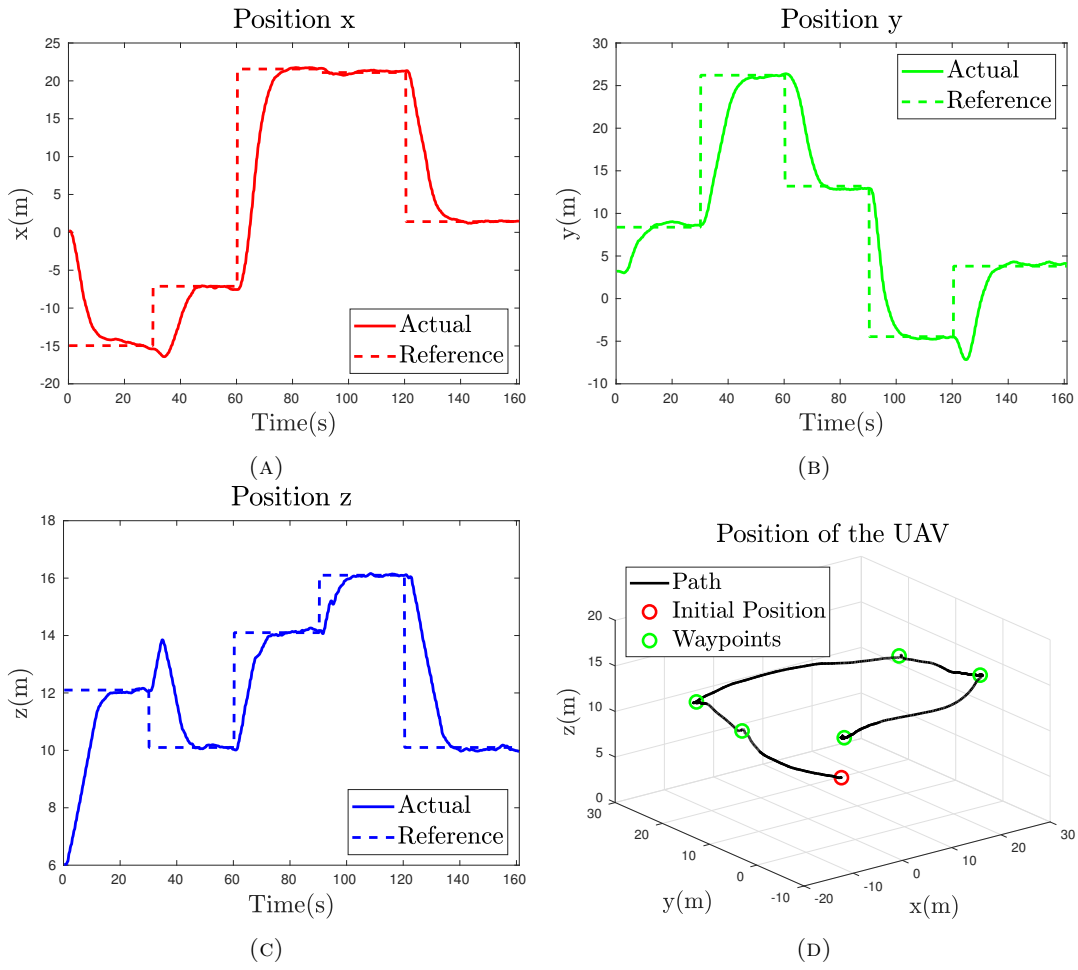


FIGURE 3.20: The actual position of the UAV \mathbf{p} and the reference waypoints \mathbf{p}_{ref} during the load transportation experiment inside the NTUA campus. (a) Position x. (b) Position y. (c) Position z. (d) The 3D path of the vehicle compared to the reference 3D waypoints.

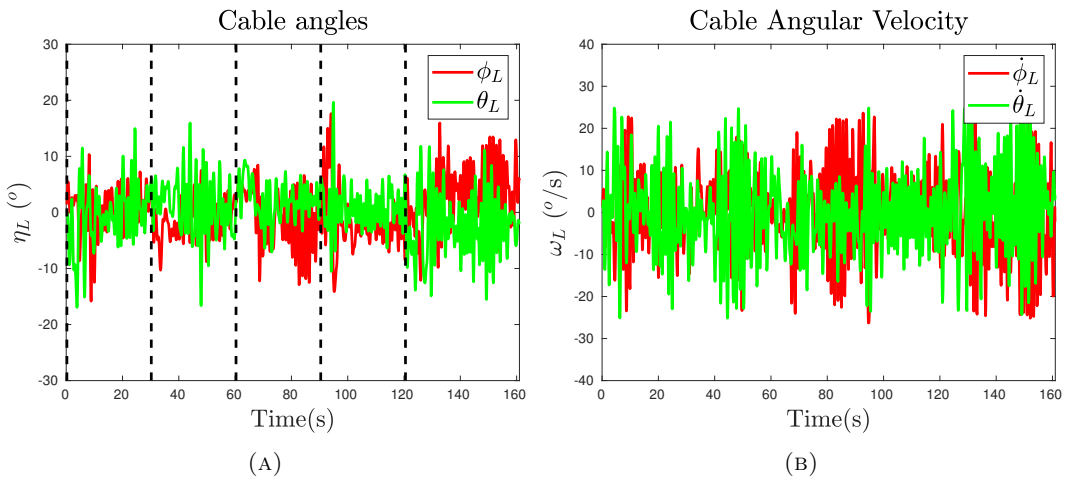


FIGURE 3.21: The estimates of the cable state, computed by the CNN and the EKF, during the load transportation experiment inside the NTUA campus. (a) Cable angles η_L . (b) Cable angular velocity ω_L .

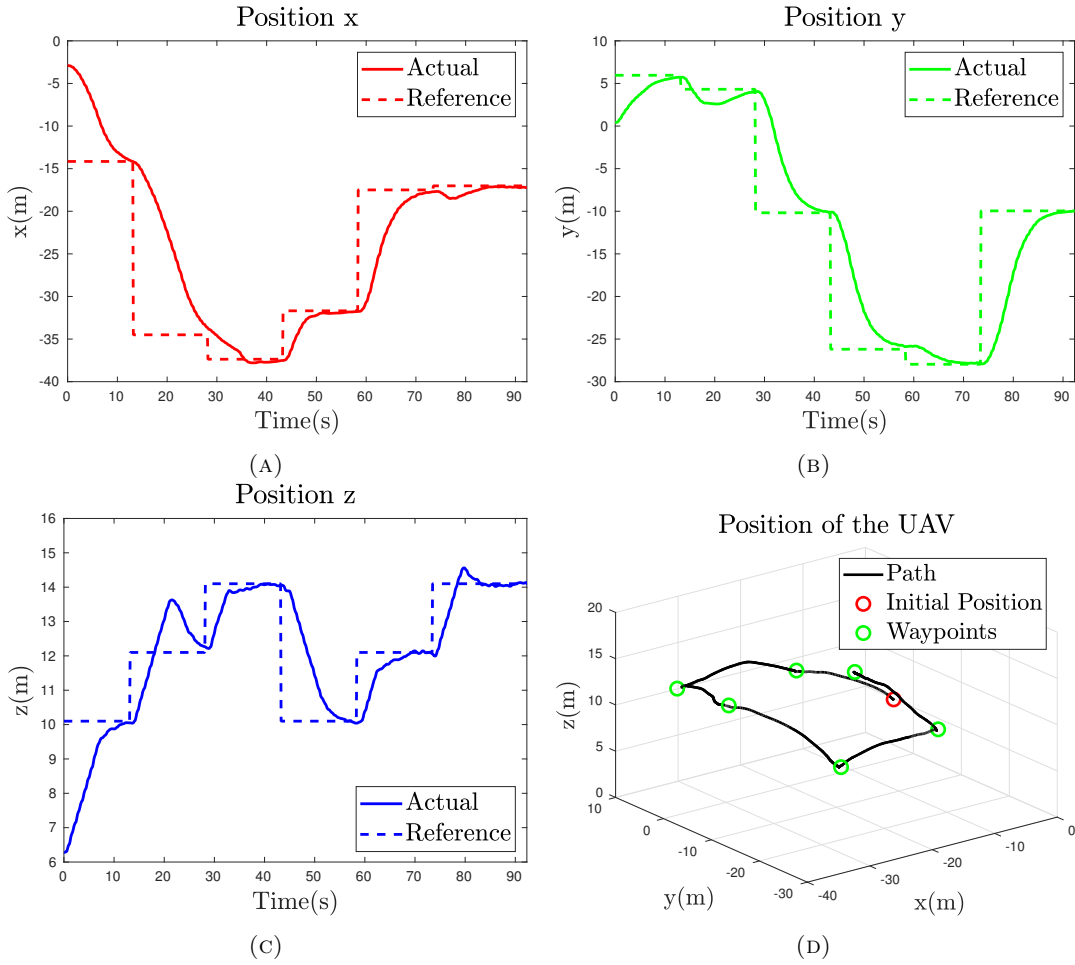


FIGURE 3.22: The actual position of the UAV \mathbf{p} and the reference waypoints \mathbf{p}_{ref} during the load transportation experiment in a beach location. (a) Position x. (b) Position y. (c) Position z. (d) The 3D path of the vehicle compared to the reference 3D waypoints.

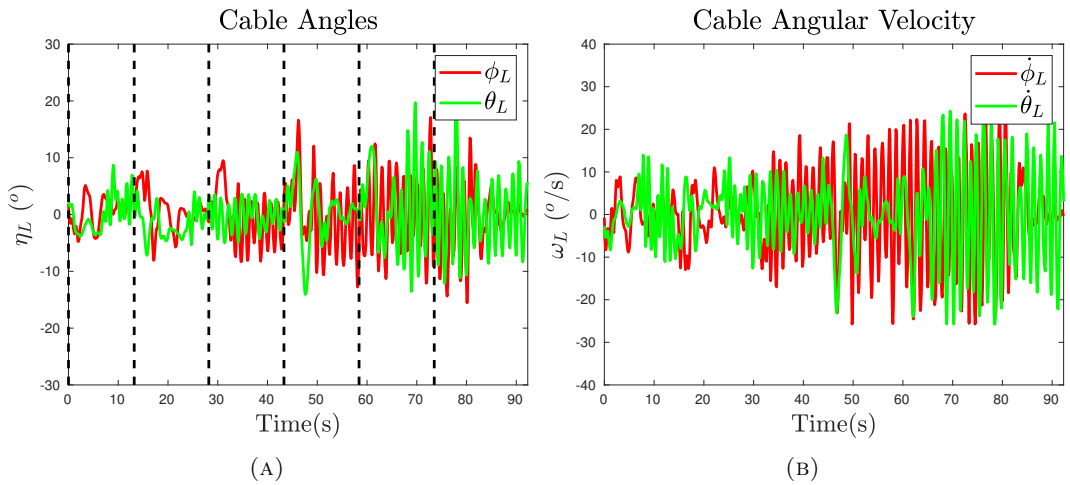


FIGURE 3.23: The estimates of the cable state, computed by the CNN and the EKF, during the load transportation experiment in a beach location. (a) Cable angles η_L . (b) Cable angular velocity ω_L .

to the surrounding environment, the proposed method reliably identified the events that were related to the swinging motion of the cable and provided robust feedback of the complete cable's state, fed into the nominal NMPC.

Given a set of reference 3D positions \mathbf{p}_{ref} and the cable's state, i.e., $\boldsymbol{\eta}_L$ and $\boldsymbol{\omega}_L$, obtained by the processing of the event stream, the nominal NMPC scheme computed desired setpoints for the inner attitude subsystem of the autopilot in order to minimize the error $\mathbf{p}_{ref} - \mathbf{p}$ and the vehicle fulfilled the mission, as depicted in Figure 3.24a. Additionally, throughout the experiment, the swinging motion of the cable, estimated by the event-based method, was maintained below critical values (Figure 3.24), since the upper part of the cable did not approach the structure of the UAV, specifically the legs of the platform, and the safety of the system was guaranteed.

In order to further investigate the reliability of the proposed algorithm, a second experiment was conducted where a mission consisting of more reference positions \mathbf{p}_{ref} was commanded. Similarly, the robust identification of the cable during the mission and the transportation of the suspended load with minimum swinging motion of the cable were attained. Both experiments are better illustrated in the following video <https://youtu.be/jFUQcZ00hoM>.

Afterwards, in order to demonstrate the advantages of using an event camera for the task of aerial transportation of cable-suspended loads, a comparative study was conducted between the event-based and the frame-based method. Throughout the comparative study, the same mission, composed of 4 reference positions, was commanded and the feedback of the cable's state, provided by the two methods, was fed into the nominal NMPC scheme of Section 3.6.1. It is highlighted that the parameters of the controller, i.e., the cost matrices and the input limits, were identical. Additionally, the experiments were realized successively in order to ensure similar environmental conditions and, hence, a fair comparison.

The two methods were compared based on the efficiency of the controller and, more specifically, its ability to minimize the norm of the cable's angle $\|\boldsymbol{\eta}_L\|$. As illustrated in Figure 3.25, the rapid and reliable feedback provided by the DVS resulted in a better performance of the controller, whereas greater swing angles were observed with the 30 FPS frame-based approach. More precisely, the maximum cable's angle $\|\boldsymbol{\eta}_L\|$ for the event-based method and the frame-based one were equal to 14.13° and 21.95° respectively. Therefore, in the aerial transportation of cable-suspended loads, high-speed feedback can significantly impact the performance of the controller and, thus, the utilization of an event camera can further improve the efficiency of existing control schemes.

3.7.5 GP-based NMPC Experiments

In this section, we demonstrate the advantages of leveraging weighted and sparse GP regression according to Section 3.6.3, when the UAV with the cable-suspended

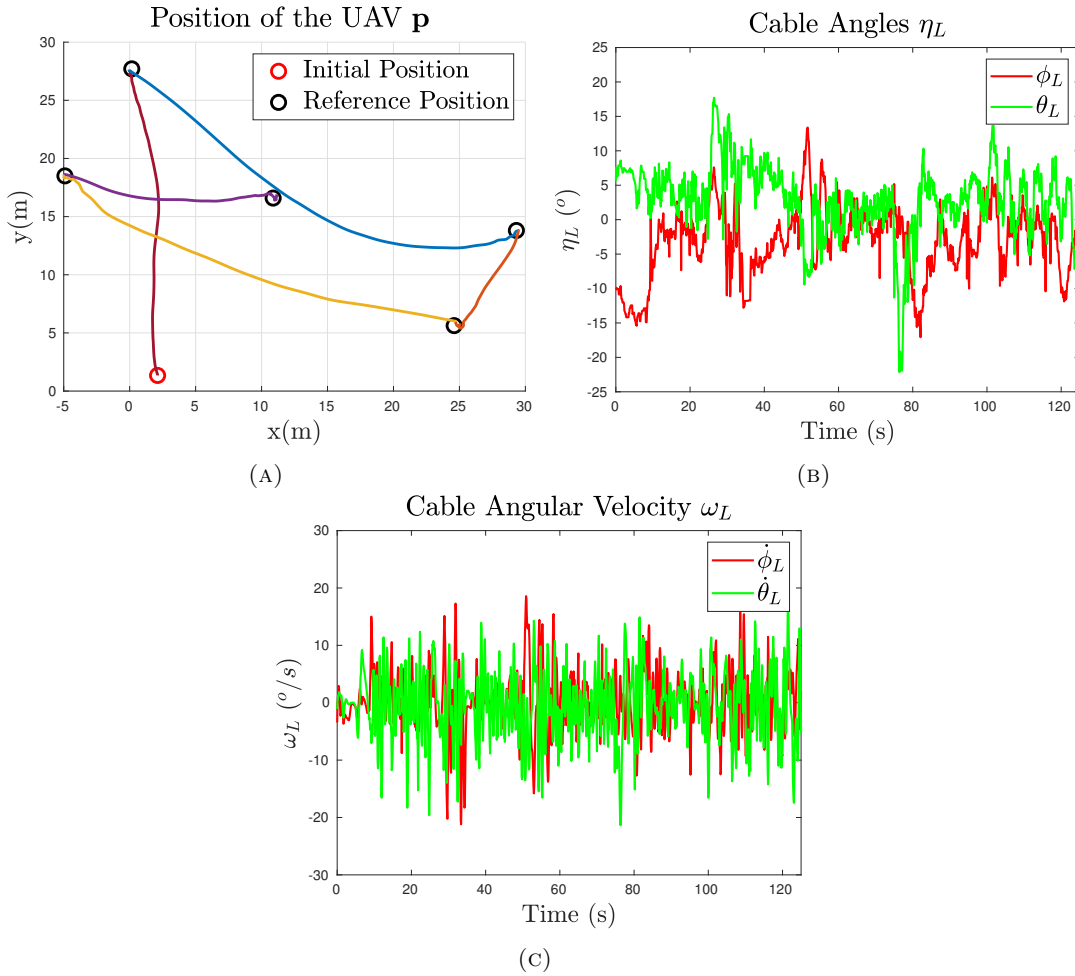


FIGURE 3.24: (a) The position \mathbf{p} of the UAV and the reference positions \mathbf{p}_{ref} in the x - y plane, (b) the cable angles η_L , and (c) velocity ω_L during the outdoor experiment with the event camera.

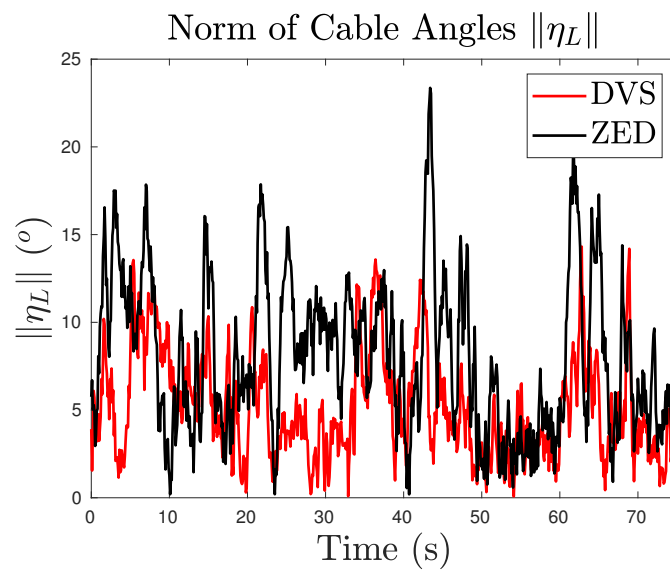


FIGURE 3.25: The norm of the cable's angle $\|\eta_L\|$ during the comparative study between the event-based method (DVS) and the frame-based solution (ZED).

load operates in the presence of considerable external disturbances and, hence, non-negligible model discrepancies exist between the nominal and actual system dynamics.

Simulation Experiments

Prior to deploying the GP-based NMPC (Section 3.6.4) on the real octorotor, experiments were realized in the Gazebo simulation environment to evaluate our method under controlled conditions. In order to validate the improvement in the control performance with the addition of GPs, a comparative study was conducted between the following scenarios: (a) NMPC with the nominal model (Section 3.6.1), (b) NMPC with sparse GP regression, (c) NMPC with weighted and sparse GP regression. The same mission, consisting of 10 different $\mathbf{p}_{ref} \in \mathbb{R}^3$, was commanded in all cases.

During the comparative study, the following external forces ${}^B\mathbf{F}_Q = [-1.2, 1.8, 0]^T - 0.3 \cdot {}^B\mathbf{v} |{}^B\mathbf{v}|$ and ${}^B\mathbf{F}_L = -0.13 \cdot {}^B\mathbf{v}_L |{}^B\mathbf{v}_L|$ were applied to the quadrotor and the load respectively w.r.t. \mathbf{B} . It is mentioned that during scenarios (b) and (c), where GPs are incorporated, the dataset was initially filled with offline measurements collected during a previous simulation experiment *without* the existence of the aforementioned constant and drag forces. As illustrated in Figure 3.26, where the position error and the cost function of the NMPC are depicted, the GP models significantly enhanced the behavior of the system under the influence of the external forces and improved the performance of the control scheme compared to the nominal case, i.e., scenario (a). Additionally, in scenario (c), the incorporation of weights, which prioritize more recent measurements, accelerated the adaptation of the regression problem to the newly encountered environment and, thus, resulted in a faster reduction of the position error and the cost function compared to the scenario (b). Finally, it should be noted that, during scenarios (b) and (c), the cost of the NMPC does not converge to zero since the vehicle hovers at the reference positions with non-zero roll and pitch angles.

Real-World Experiments

Following the validation in the Gazebo simulator, real-world experiments with the octorotor, including a $m_L = 0.5\text{kg}$ load suspended through a $l = 3.8\text{m}$ cable, and the DVS camera were carried out in an outdoor environment in order to test the efficacy of the GP-based NMPC scheme of Figure 3.27 against challenging conditions and unknown external disturbances. Similarly, a comparative study was conducted between the different NMPC schemes, i.e., including (a) the nominal model, (b) sparse GP regression, and (c) weighted and sparse GP regression. It should be noted that the experiments were realized sequentially, at the same location, and, thus, the outdoor conditions, particularly the air speed and direction, can be considered approximately identical despite the unavoidable existence of turbulence. Finally, the same 7 reference positions \mathbf{p}_{ref} were provided as input to the controllers so as to ensure a fair comparison.

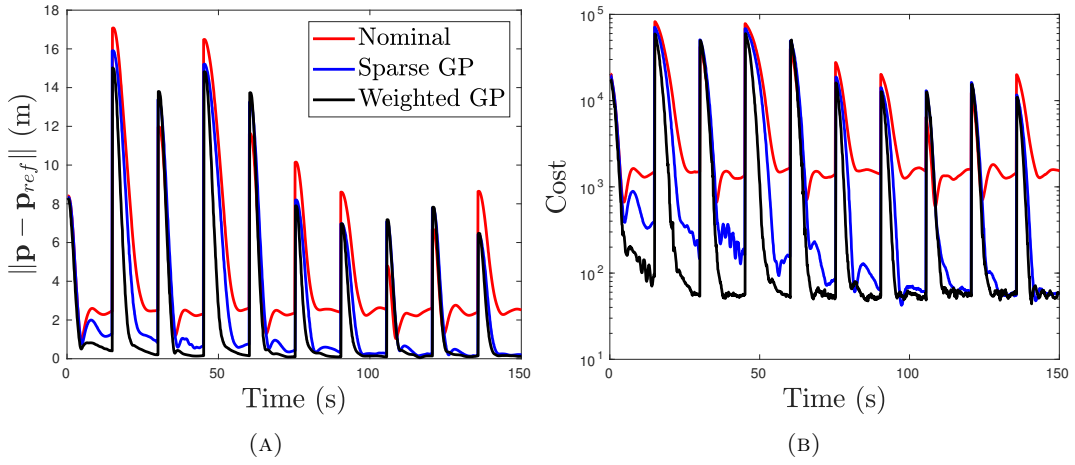


FIGURE 3.26: (a) The error $\|\mathbf{p} - \mathbf{p}_{ref}\|$ between the actual position of the UAV and the reference ones, and (b) the cost function of the NMPC (y-axis displayed in logarithmic scale) during the simulation comparative study.

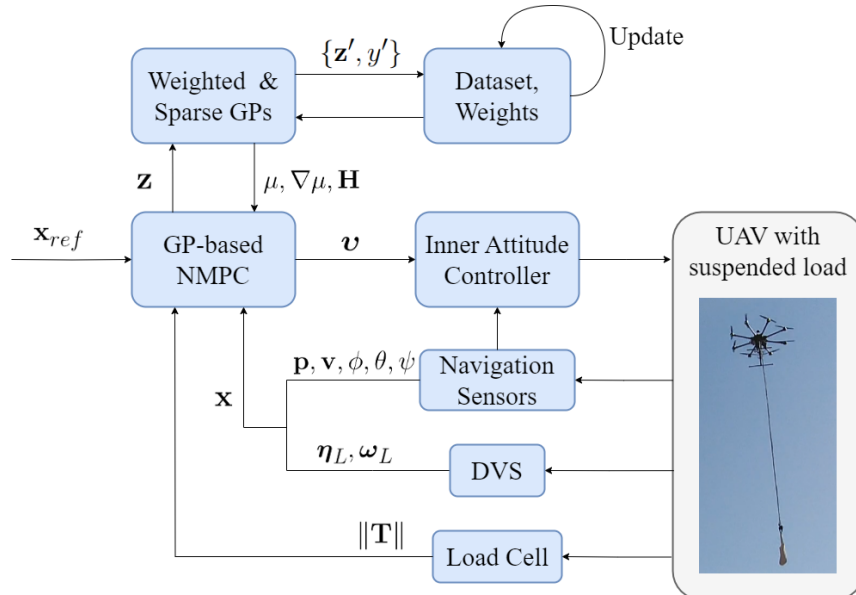


FIGURE 3.27: Overview of the GP-based NMPC architecture including the DVS camera.

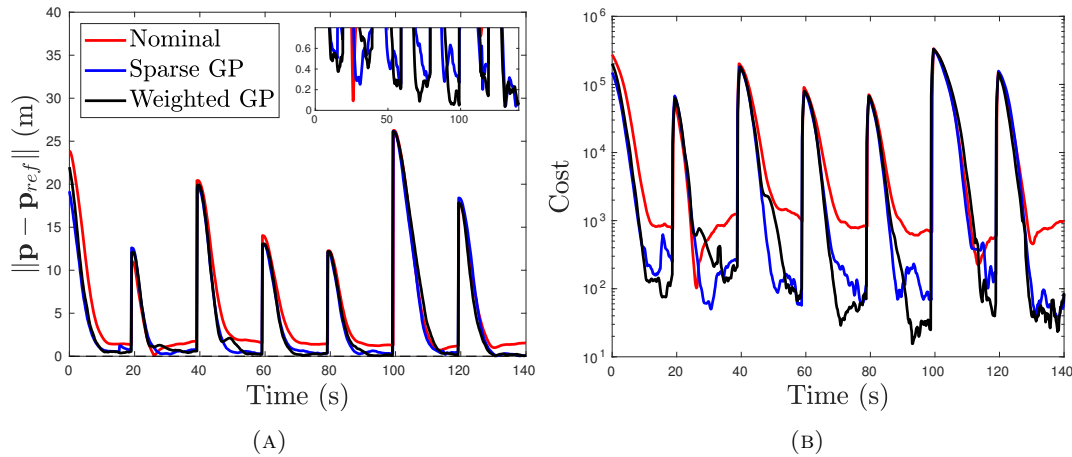


FIGURE 3.28: (a) The error $\|\mathbf{p} - \mathbf{p}_{ref}\|$ between the actual position of the UAV and the reference ones (a zoomed-in view near zero is illustrated in the upper right corner), and (b) the cost function of the NMPC (y-axis displayed in logarithmic scale) during the comparative study in a outdoor environment.

In both scenarios (b) and (c), the dataset was initially composed of measurements gathered during a previous day with completely different environmental conditions. As depicted in Figure 3.28, a considerable steady-state position error and an increased cost were observed when the NMPC formulated for the nominal model, i.e., scenario (a), was deployed. The aforementioned behavior mainly originated from the 4 beaufort mean wind speed, prevailing in the area according to meteorological data, that resulted in a deviation between the nominal and actual system dynamics. In contrast, the incorporation of GP models throughout scenarios (b) and (c) led to a significantly reduced position error and cost which proves the ability of GPs to approximate the model discrepancies and the external disturbances. Moreover, the weighted version of sparse GP regression during scenario (c) highlighted the importance of the newly collected measurements and, hence, a faster improvement in the system behavior and a generally better performance of the control scheme were attained compared to scenario (b). The aforementioned experiments are better illustrated in the following video <https://youtu.be/4bm3JLo0d5U>.

Chapter 4

Tracking and Release of a Suspended Load to a Ground Target

4.1 Introduction

In this chapter, we address the task of tracking a ground vehicle using a multirotor with a cable-suspended load and a downward-looking camera, with the ultimate goal of successfully releasing the load towards the target. Aerial target tracking based on visual feedback is a well studied field throughout the literature, e.g., [9], [10], [29], and [36]; however without considering the existence of a cable-suspended load and, hence, its release towards the target.

To this end, a Nonlinear Model Predictive Control (NMPC) scheme, formulated for the system dynamics, is developed in order to accomplish the following of the arbitrarily moving target. Since the NMPC relies on the future behavior of the system, the trajectory of the ground target throughout the predicted time horizon of the NMPC, is required. Towards this direction, a Convolutional Neural Network (CNN), accompanied by the Kernelized Correlation Filter (KCF) tracker, is exploited for uninterruptedly detecting the target and, subsequently, the aforementioned measurements are integrated into a constrained Bézier regression problem in order to predict its future trajectory [29]. Eventually, the load is released towards the ground target, by opening a gripper, as soon as a condition, which relates the predicted ballistic trajectory of the load and the future target motion, is met. It is mentioned that the swinging motion of the cable during the tracking of the target renders the successful release of the load quite challenging even for an experienced human operator.

This chapter is organized as follows. Section 4.2 describes the problem of tracking and releasing a suspended load to a ground vehicle using a multirotor equipped with a camera. In Sections 4.3 and 4.4, the vision-based estimation of the future target motion and the deployed NMPC scheme are presented respectively. Finally, Section 4.5 discusses the release condition while the chapter concludes with Section 4.6, where

the performance of the proposed NMPC framework is validated through real-world experiments.

4.2 Problem Formulation

In this section, we formulate the motion control problem of tracking a ground target using a multirotor with a cable-suspended load, as illustrated in Figure 4.1. More precisely, consider the case in which an unmanned ground vehicle (UGV) is moving arbitrarily on the ground without sharing any information about its state, i.e., its position \mathbf{p}_T and velocity \mathbf{v}_T w.r.t. the world frame \mathbf{W} , and, hence, the UAV relies only on the visual information, obtained by the onboard downward-looking camera, so as to continuously track the target and ultimately release the suspended load towards it.



FIGURE 4.1: The multirotor with the cable-suspended load and the UGV deployed in the field.

Towards this direction, the control scheme, depicted in Figure 4.2, is deployed in order to fulfill the requested task. The aforementioned control strategy consists of the following parts:

1. CNN-based detection of the cable and estimation of the full cable's state, i.e., $\boldsymbol{\eta}_L$ and $\boldsymbol{\omega}_L$, using a Kalman Filter which exploits the output of the CNN and the measurements provided by a load cell, according to Section 3.3.1,
2. Detection of the target using a KCF tracker and a CNN, which indicates an initial region of interest (ROI) for the KCF tracker, and prediction of the future target motion, i.e. the position \mathbf{p}_T and the velocity \mathbf{v}_T w.r.t. \mathbf{W} , based on Bézier curves,

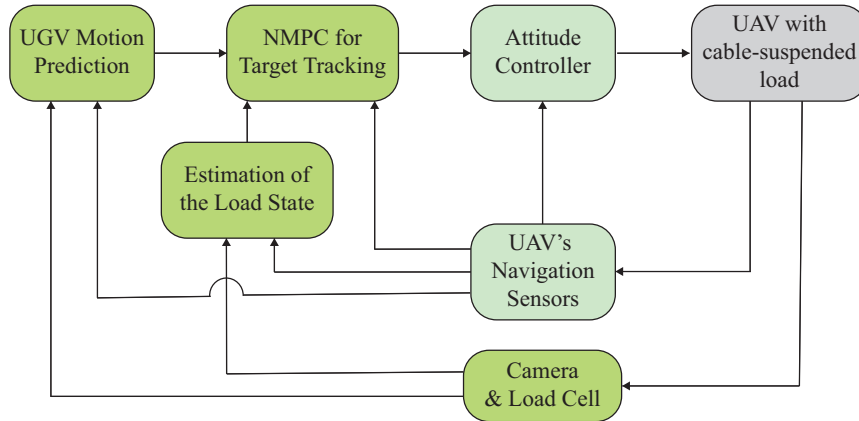


FIGURE 4.2: The proposed control scheme for tracking the moving ground vehicle.

3. Deployment of an NMPC, formulated for the nonlinear dynamics of the system (Equation (2.11)), which integrates the future trajectory of the target into the predicted horizon of the NMPC and produces appropriate commands for the inner attitude control loop of the autopilot so as to minimize the tracking error between the target and the load and
4. Check of a condition which, based on the predicted trajectory of the target and the load, triggers the opening of a gripper and the release of the load towards the target.

The above individual parts, besides the previously presented frame-based estimation of the cable's state (Section 3.3.1), are explicitly analyzed in the following sections.

4.3 Estimation of the Target State

4.3.1 Detection of the Target

Besides the cable's state, an estimate of the target state is a prerequisite to achieve the tracking of the ground target and the successful release of the load towards it. Similarly to Section 3.3.1, a CNN is trained in order to detect the target which moves in an unknown manner.

However, when the load is located above the target, a part of the area, which corresponds to the target, is visually occluded by the load and, as a result, the CNN fails to detect the target consecutively. Hence, a tracker is additionally exploited so as to achieve the robust identification of the target despite the visual occlusions while also ensuring a faster detection rate. Among the various trackers presented in the literature, the KCF tracker [30] is selected due to its high speed and accuracy and its ability to track the target in the presence of partial occlusions.

The pipeline of the detection is as follows:



FIGURE 4.3: Detection of the target. The green cross corresponds to the center of the bounding box.

1. Extract a ROI by utilizing the trained CNN,
2. Based on the aforementioned ROI, deploy the KCF tracker and
3. Re-detect the target using the CNN after some iterations in order to reduce the accumulated error or when the tracker fails.

Based on the aforementioned pipeline, a ROI, which corresponds to the target, is extracted by means of a bounding box, as depicted in Figure 4.3. The center (u_T, v_T) of the bounding box defines the pixel coordinates of the target in the image plane. Since the target moves on the ground, its known height $z_T = h$ is exploited in order to compute its 3D position. Consequently, after suitable transformations between the camera and the world frame, an estimate of the target position $\mathbf{p}_T = [x_T \ y_T \ z_T]^T$ is obtained at each time instant.

4.3.2 Target Motion Prediction

The estimate of the target position is not adequate for the deployed control scheme, since the NMPC requires both the position and velocity of the target not only at the current time instant but also throughout the predicted time horizon. To this end, Bézier curves are exploited, as proposed in [29], similarly to Section 3.3.2. However, since the objective is to predict the future trajectory of the target, the regression problem is reformulated compared to Section 3.3.2, where the Bézier curve smoothed the measurements up to the current time instant.

More precisely, as mentioned in Section 3.3.2, a Bézier curve is defined, based on the order n of the curve, as follows:

$$\mathbf{B}(\tau) = \sum_{i=0}^n b_{i,n}(\tau) \mathbf{c}_i \quad (4.1)$$

where $\tau \in [0, 1]$, $\mathbf{B}(\tau) \in \mathbb{R}^2$ (since the target is moving on the x-y plane) is the Bézier curve, $b_{i,n}(\tau), i = 0, \dots, n$ are the Bernstein basis polynomials of degree n and $\mathbf{c}_i = [c_{x,i} \ c_{y,i}]^T \in \mathbb{R}^2$ are the $n + 1$ control points.

In order to estimate and predict the target motion, it is essential to find the suitable set of control points $\mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n]$. Towards this direction, assuming that N estimates of the target position \mathbf{p}_T are available according to Section 4.3.1, along with the corresponding timestamps t , a set $D = \{(t_1, x_{T,1}, y_{T,1}), \dots, (t_N, x_{T,N}, y_{T,N})\}$ of constant length is constructed, where the timestamp t_N corresponds to the current time. The aforementioned set D is updated once a new measurement is available. Since the aim is to predict the motion of the target up to the future timestamp t_p , the following mapping is performed between the timestamps $t \in [t_1, t_p]$ and the parameter $\tau \in [0, 1]$ of the Bézier curve:

$$\tau = \frac{t - t_1}{t_p - t_1} \quad (4.2)$$

The optimal set of control points is computed according to the following optimization problem:

$$\min_{\mathbf{c}} \sum_{i=1}^N \left(w_{t_i} \|\mathbf{B}(\tau_i) - \mathbf{p}_T(t_i)\|_2^2 \right) \quad (4.3)$$

where the goal is to minimize the distance between the measurements and the Bézier curve with w_{t_i} denoting weights which penalise past measurements [29].

Additionally, in order to identify feasible solutions, constraints are imposed on the above optimization problem. The aforementioned constraints arise from the dynamic limits of the target according to which the predicted velocity and acceleration should lie inside $[-u_{max}, u_{max}]$ and $[-a_{max}, a_{max}]$ respectively. By differentiating the Bézier curve, the following constraints are defined for the x axis (similarly for the y dimension):

$$\begin{aligned} -u_{max} &\leq n \cdot (c_{x,i} - c_{x,i-1}) / (t_p - t_1) \leq u_{max} \\ -a_{max} &\leq n \cdot (n - 1) \cdot (c_{x,i} - 2c_{x,i-1} + c_{x,i-2}) / (t_p - t_1)^2 \leq a_{max} \end{aligned} \quad (4.4)$$

The above constrained Quadratic Programming problem is solved using the OQPP software [24], similarly to Section 3.3.2, and, eventually, an estimate of the future trajectory of the target, i.e., $\mathbf{p}_T(t)$ and $\mathbf{v}_T(t)$, is available throughout the predicted horizon $[t_N, t_p]$.

4.4 Nonlinear Model Predictive Control for Target Tracking

In order to efficiently track the ground target, an NMPC is formulated for the nonlinear system dynamics defined by Equation (2.11). Since the objective of the NMPC scheme is to track the ground target and eventually release the cable-suspended load towards the target, the following cost functions are defined:

$$\begin{aligned}
 J_p(t) &= \left\| \mathbf{p}(t) + \mathbf{ln}(t) - \mathbf{p}_T(t) - \begin{bmatrix} 0 & 0 & z_{ref} \end{bmatrix}^T \right\|_{\mathbf{Q}_p}^2 \\
 J_v(t) &= \|\mathbf{v}(t) + \mathbf{ln}(t) - \mathbf{v}_T(t)\|_{\mathbf{Q}_v}^2 \\
 J_\eta(t) &= \left\| \begin{bmatrix} \phi(t) & \theta(t) & \psi(t) - \psi_{ref} \end{bmatrix}^T \right\|_{\mathbf{Q}_o}^2 \\
 J_L(t) &= \left\| \begin{bmatrix} \boldsymbol{\eta}_L^T(t) & \boldsymbol{\omega}_L^T(t) \end{bmatrix}^T \right\|_{\mathbf{Q}_L}^2
 \end{aligned} \tag{4.5}$$

where $\mathbf{Q}_p \in \mathbb{R}^{3 \times 3}$, $\mathbf{Q}_v \in \mathbb{R}^{3 \times 3}$, $\mathbf{Q}_o \in \mathbb{R}^{3 \times 3}$ and $\mathbf{Q}_L \in \mathbb{R}^{4 \times 4}$ are diagonal weighting matrices and J_p , J_v , J_η and J_L are task-specific cost functions which penalize the distance between the load and the target, the relative velocity between the load and the target, the error between the current orientation of the vehicle and the reference one, and the swinging motion of the cable. It is mentioned that the UAV should track the ground target while maintaining a certain altitude above it, defined by the variable z_{ref} . The reference yaw orientation ψ_{ref} is maintained constant. Additionally, the swinging motion of the cable is minimized so as to avoid unnecessary oscillations and ensure the safety of the load to be delivered.

Considering the total cost function $J_t = J_p + J_v + J_\eta + J_L$ and the input constraints, the following Optimal Control Problem (OCP) is formulated throughout the horizon $[t_N, t_p]$:

$$\begin{aligned}
 \min_{\mathbf{v}} \int_{t_N}^{t_p} (J_t(t) + \|\mathbf{v}(t)\|_{\mathbf{R}}^2) dt + J_t(t_p) \\
 \text{s.t.: } \mathbf{x}(t_N) = \mathbf{x}_N, \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{v}) \text{ (Equation (2.11))}, \mathbf{v} \in \mathbb{U}
 \end{aligned} \tag{4.6}$$

where $\mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T \ \phi \ \theta \ \psi \ \boldsymbol{\eta}_L^T \ \boldsymbol{\omega}_L^T]^T \in \mathbb{R}^{13}$, $\mathbf{v} = [\phi_d \ \theta_d \ \psi_d \ v_{z_d}]^T \in \mathbb{R}^4$ are the state vector and the commands for the inner attitude controller of the autopilot respectively, \mathbf{R} is the control input cost matrix and, \mathbb{U} is the set of input constraints specified by the roll, pitch and vertical velocity limits.

The aforementioned OCP is solved at each iteration t_N , given the predicted target trajectory, and the first control input of the derived control sequence is sent to the inner attitude control loop of the autopilot. The ACADO Toolkit [33], along with the qpOASES solver [17], are again utilized for designing and solving in real time the presented OCP.

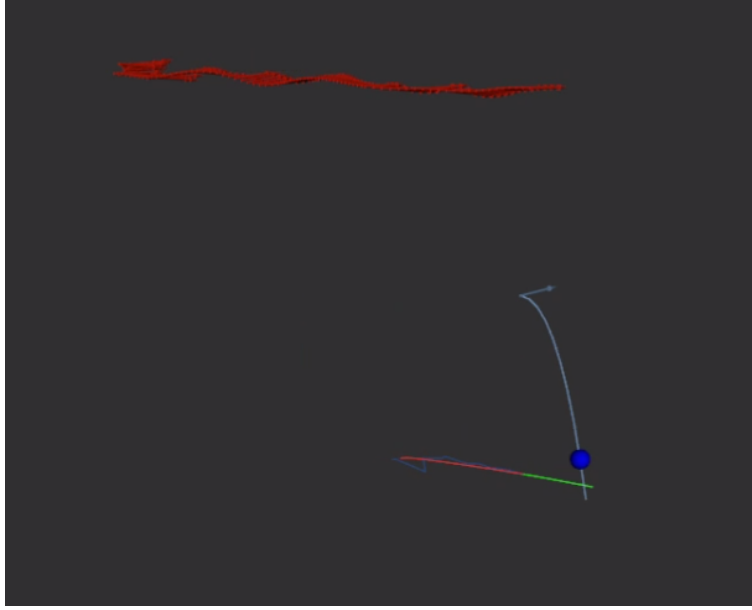


FIGURE 4.4: The red arrows at the upper part of the figure illustrate the trajectory of the UAV while the red and green colored lines at the lower part represent the past and future trajectory of the target respectively, as approximated by the Bézier curve. The ballistic trajectory of the load, assuming that is released, is depicted by the line with the cyan hue. Finally, the blue colored sphere indicates that the release condition is satisfied.

4.5 Release Condition

Regarding the release condition, at each current timestamp t_N , the trajectory of the load is computed assuming that is released. In this case, the load follows a ballistic trajectory, i.e. moves only under the influence of gravity, according to the following equation:

$$\mathbf{p}_L(t) = \mathbf{p}_L(t_N) + \mathbf{v}_L(t_N) \cdot t - g\mathbf{e}_z \frac{t^2}{2}, \quad (4.7)$$

where \mathbf{p}_L and \mathbf{v}_L are given by Equation (2.1). Given the known target's height z_T and the above equation, the timestamp t_h at which the trajectory of the load intersects with the z_T is computed, i.e., $z_L(t_h) = z_T$. Consequently, the release condition is formulated as follows:

$$\|\mathbf{p}_L(t_h) - \mathbf{p}_T(t_h)\| \leq d \quad (4.8)$$

where d is the maximum horizontal distance between the load and the target. It is mentioned that the condition implies that the timestamp t_h lies inside the predicted horizon of the target motion, i.e., $t_h \in [t_N, t_p]$, and, hence, an estimate of the target motion is available. Otherwise, the release condition is not checked. Figure 4.4 offers a visual explanation of the release condition.

4.6 Experimental Results

4.6.1 Experimental Setup

The proposed scheme is evaluated experimentally in an outdoor environment while using the octorotor of Figure 4.1 with the embedded computer Jetson AGX Xavier and the ZED 2 camera. The vehicle is additionally equipped with a load cell, which measures the cable tension $\|\mathbf{T}\|$, and a servo motor which controls a gripper and, thus, the release of the load. A base is 3D printed in order to assemble the load cell and the gripper, both located exactly above the load. Both the load cell and the servo motor are directly connected with an Arduino Uno, which communicates serially with the Jetson. As for the load, a bag containing a $m_L = 0.5$ kg load is considered. Finally, the ground target is a UGV Robotnik Summit which is manually driven towards random directions through joystick teleoperation.

As for the computational cost of the proposed framework, the mean processing time, on the powerful Jetson, of the individual parts, that comprise the control scheme of Figure 4.2, is demonstrated in Table 4.1. It is evident that a real-time performance is achieved during the tracking of the UGV.

Step	Time[ms]
CNN	32.41
KCF	5.27
Bézier regression ($N = 20$)	0.93
NMPC	3.33

TABLE 4.1: The computational cost of the individual parts that comprise the proposed framework for tracking the ground target.

4.6.2 Real-World Experiments

In order to validate the ability of the proposed control strategy to uninterruptedly track the UGV, an outdoor experiment was initially conducted without considering the release of the load. Throughout the first experiment, the UGV moved with an average horizontal velocity of $1.5m/s$ and a maximum equal to $2.5m/s$, as estimated by the Bézier regression (Figures 4.5c, 4.5d). The deployed vision-based NMPC scheme, which communicates directly with the inner attitude subsystem of the UAV, enables the vehicle to respond quickly to the unknown maneuvers of the target, performed by the operator, and, hence, the continuous tracking of the target was achieved, according to Figures 4.5a, 4.5b, where the trajectories of both vehicles in the horizontal plane are depicted. Additionally, despite the agile motion of the UGV, the swinging motion of the cable, estimated according to Section 3.3.1, was not excited and the cable angles were maintained below 20° , as illustrated in Figure 4.5e.

Afterwards, a second experiment was realized in order to evaluate the ability of the proposed scheme to successfully release the suspended load towards the target. The UAV initially tracked the vehicle until the release condition of Equation (4.8)

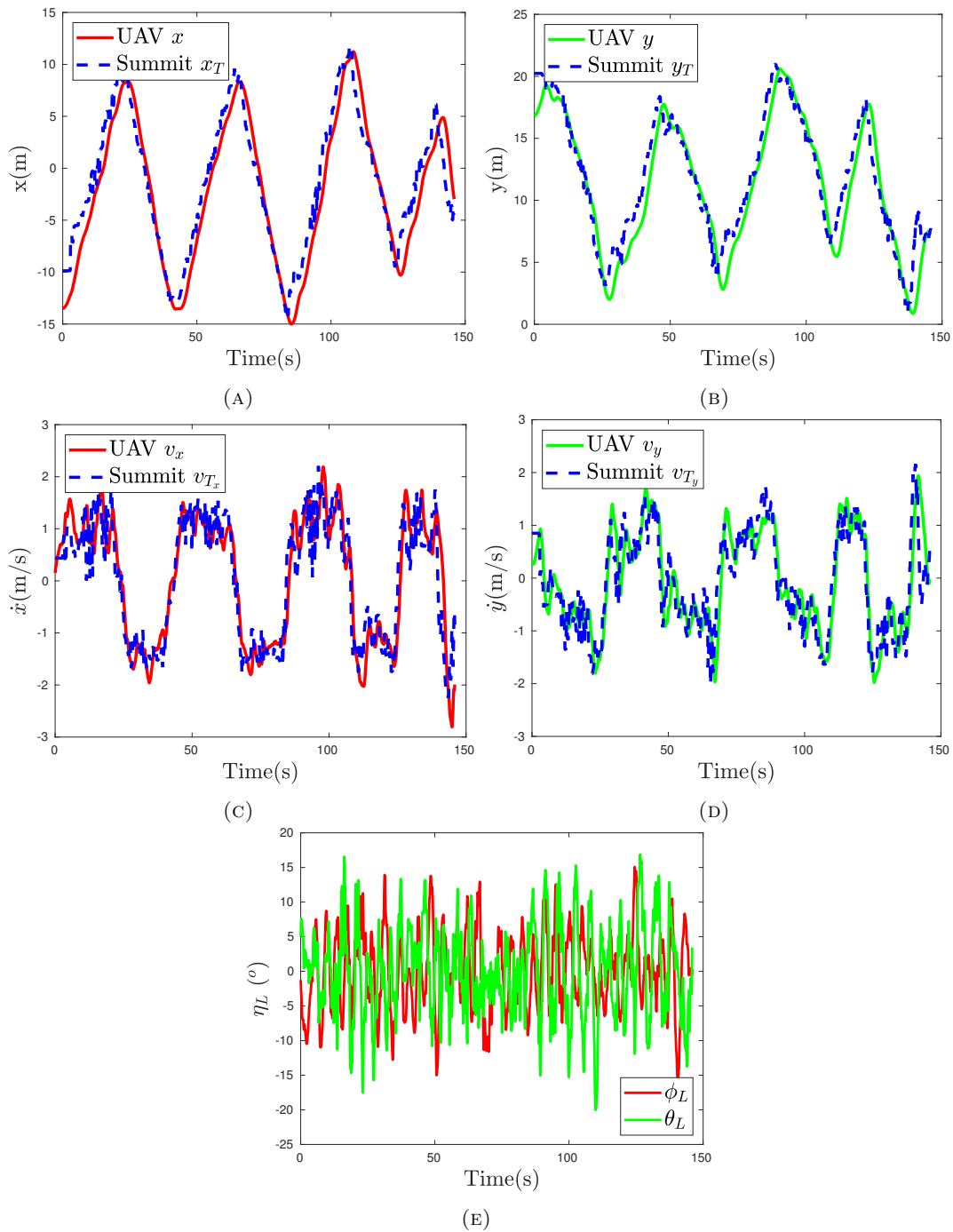


FIGURE 4.5: The position \mathbf{p} and the velocity \mathbf{v} of the UAV during the first experiment compared to the UGV motion, i.e., \mathbf{p}_T and \mathbf{v}_T , as estimated by the Bézier regression. (a) Position x . (b) Position y . (c) Velocity x . (d) Velocity y . (e) The cable angles $\boldsymbol{\eta}_L = [\phi_L \ \theta_L]^T$.

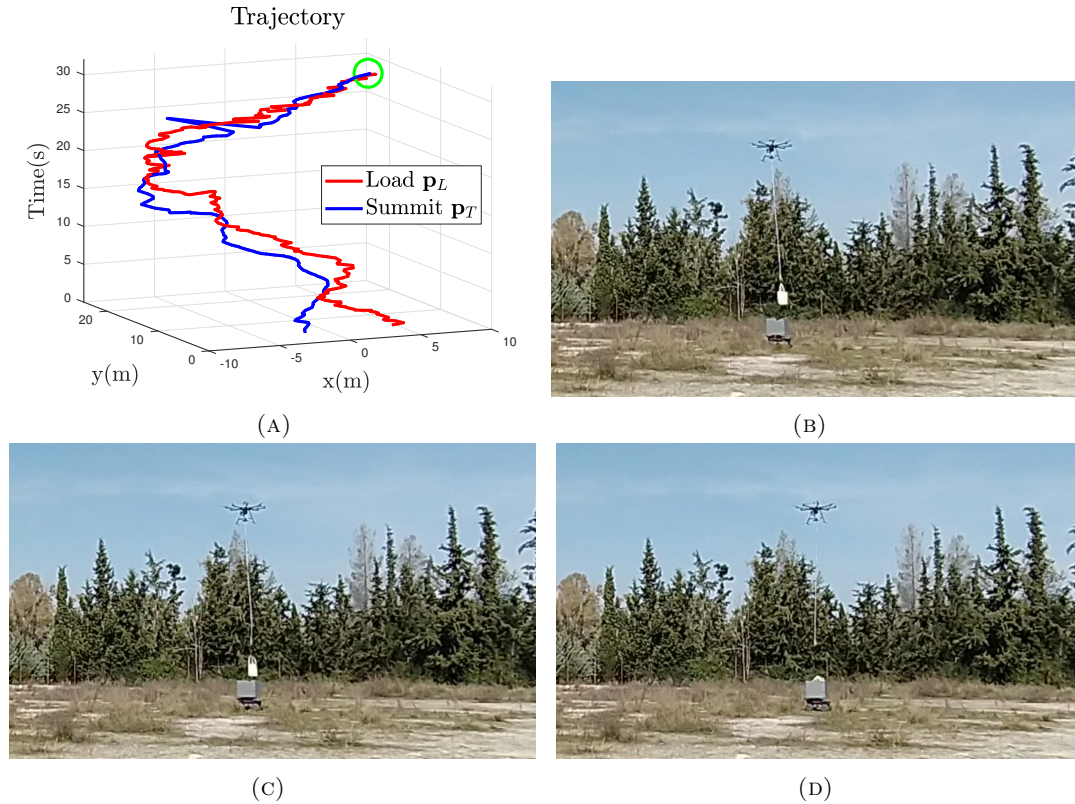


FIGURE 4.6: The successful release of the load towards the target during the second experiment. (a) The position of the load $\mathbf{p}_L(t_h)$, assuming that is released, compared to the position of the target $\mathbf{p}_T(t_h)$ in the x-y plane. The green circle indicates that the release condition (Equation (4.8)) is satisfied. (b) The gripper opens. (c) The load follows a ballistic trajectory. (d) The load is placed inside the box.

was satisfied, as depicted in Figure 4.6a. At that moment, the opening of the gripper was triggered and the load was successfully released and placed into a box, carried by the UGV, as illustrated in Figure 4.6.

The aforementioned experiments, including also an additional one, are better illustrated in the following video <https://youtu.be/6aleTdb71Sc>.

Chapter 5

Water Sampling from Aquatic Environments

5.1 Introduction

Throughout this chapter, we address the task of conducting water sampling from an aquatic environment with the aid of a multirotor. In order to avoid the integration of a carefully designed floating base at the bottom of the UAV's body, which enables the vehicle to land and float upon the water surface [1],[43],[58], we prefer the usage of a cable-suspended mechanism since it constitutes a more reproducible and plug-and-play solution with regard to mechanical complexity and does not restrict the applicability of the UAV to tasks related solely to water sampling.

However, the drag forces applied to the submerged mechanism, due to its interaction with the water flow, lead to an inclined configuration of the cable, especially in river ecosystems where a considerably high average water velocity is usually observed. The tension of the inclined cable impedes the UAV from precisely stabilizing above the commanded sampling position and, thus, has to be estimated based on onboard sensors. Exclusively vision-based techniques are not suitable in this case, since the mechanism is submerged and, hence, its detection is extremely challenging, if not infeasible, and highly dependent on the purity of the water. To this end, the platform's sensor suite is augmented with additional low cost sensors, namely an ultrasonic sensor, a load cell, and a depth sensor, besides the downward-looking camera. In order to fuse the respective measurements, we design two methodologies: (a) Gaussian Process (GP) regression and (b) Extended Kalman Filter (EKF). Eventually, the effect of the disturbance is mitigated by incorporating the estimate of the cable tension into a feedback controller and, hence, precise stabilization of the UAV above the sampling position is achieved. Two control schemes are investigated for this task, specifically (a) a geometric controller and (b) a Nonlinear Model Predictive Control (NMPC) strategy.

The outline of this chapter is as follows. Section 5.2 defines the control problem of conducting water sampling with a multirotor while Section 5.3 presents two sensor

fusion techniques, based on GP regression and Kalman Filtering, which combine the measurements of low cost sensors and estimate the cable tension during the sampling procedure. In Section 5.4, a geometric controller and an NMPC scheme are developed in order to reject the disturbances and attain the accurate position control of the vehicle. Finally, in Section 5.5, the proposed methodologies are evaluated through real-world experiments while also a unified framework for autonomous water sampling missions is presented.

5.2 Problem Formulation

In this section, the formulation of the control problem of water sampling with a multirotor is presented. Consider the case in which a UAV with a cable-suspended mechanism is performing water sampling in an aquatic environment, as illustrated in Figure 5.1. Assuming that the velocity of the water flow \mathbf{u}_w is not negligible, the induced drag forces \mathbf{f}_d (Equation (2.12)) acting on the submerged mechanism, result in an inclined configuration of the cable, i.e., $|n_z| < 1$. Consequently, the tension of the inclined cable $\mathbf{T} = \|\mathbf{T}\|\mathbf{n}$, applied to the airframe, affects the position control of the UAV and, hence, an increased position error, compared to the reference sampling location \mathbf{p}_{ref} , is observed when the external force is not considered.



FIGURE 5.1: Water Sampling with an octorotor in a river.

For this purpose, a suitable control scheme is required in order to achieve the precise stabilization of the vehicle above the reference sampling position despite the underwater disturbances transferred to the UAV through the cable. The following parts are implemented:

1. Direct estimation of the cable tension $\hat{\mathbf{T}}$ based on measurements provided by appropriately integrated onboard sensors i.e. an ultrasonic sensor, a downward-looking camera, a load cell and a depth sensor,
2. Approximation of the cable tension with the aid of an Extended Kalman Filter (EKF) or Gaussian Processes (GPs) in order to mitigate the sensor noise effects and
3. Deployment of a geometric controller or a Nonlinear Model Predictive Controller (NMPC) which considers the estimates of the tension and produces appropriate desired commands for the inner attitude controller of the autopilot so as to minimize the position error $\mathbf{p} - \mathbf{p}_{ref}$.

5.3 Disturbance Estimation

5.3.1 Sensor-based Disturbance Estimation

In order to achieve the robust stabilization of the vehicle above the reference sampling location, an estimate of the cable tension $\mathbf{T} = \|\mathbf{T}\|\mathbf{n}$ is essential. For this purpose, the downward-looking camera, the load cell, the ultrasonic sensor and the depth sensor are exploited.

More precisely, a load cell is added at the middle of the cable in such a manner that it measures directly the tension. It is mentioned that the load cell provides reliable measurements of the tension with low noise. However, only the norm of the tension $\|\mathbf{T}\|$ is acquired using the load cell and, hence, additional sensors are required to obtain the direction of the disturbance, i.e., the cable unit vector \mathbf{n} .

Consequently, a depth/pressure sensor is placed at the end of the cable in order to measure the depth d up to which the sampling mechanism is submerged, i.e., the vertical distance between the lower edge of the cable and the water surface level. Since the cable length l is known, an estimate of the component \hat{n}_z is derived, under the assumption that the cable is taut, based on the similar triangle method:

$$\hat{n}_z = -\frac{h+d}{l} \quad (5.1)$$

where h denotes the height of the vehicle above the water surface level, directly obtained by the ultrasonic sensor and corrected for non-zero roll and pitch angles. Since ultrasonic sensors are prone to noise above water surfaces, a moving average filter is applied while outliers are disregarded.

Furthermore, the downward-facing camera is exploited in order to estimate the projection of the cable unit vector \mathbf{n} to the x-y plane of the world frame \mathbf{W} . More specifically, the Dense Inverse Search (DIS) [44], an optical flow algorithm which distinguishes itself for its low computational complexity and its competitive accuracy, is employed so as to compute a dense flow field, i.e., the optical flow for all the points

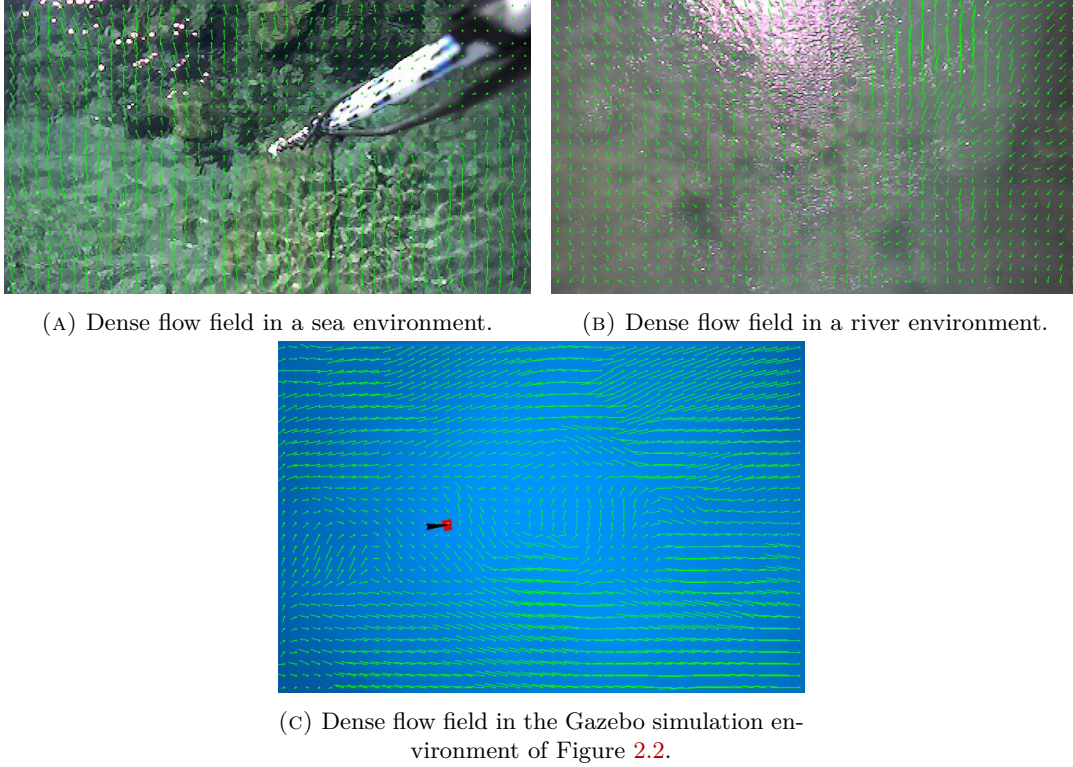


FIGURE 5.2: Dense flow fields computed by the DIS algorithm. The green arrows illustrate the direction of the flow while the length of the arrow indicates the magnitude of the displacement.

in the frame. The output of the DIS algorithm is demonstrated in Figure 5.2 for both simulation and real-world environments.

By averaging the dense flow field, the dominant water flow velocity ${}^C\hat{\mathbf{v}}_w$ is estimated with respect to the camera frame after converting the pixel velocities to cartesian ones and removing the odometry of the vehicle. Afterwards, an appropriate transformation to the world frame \mathbf{W} is performed, by utilizing the constant rotation matrix from the camera frame to the body-fixed one \mathbf{B} and the matrix \mathbf{R}_{WB} , and, thus, the water velocity $\hat{\mathbf{v}}_w$ with respect to \mathbf{W} is extracted.

According to Equation (2.12), the projection $\mathbf{n}_{xy} = [n_x \ n_y \ 0]^T$ of the cable unit vector to the x-y plane of \mathbf{W} , i.e., the water surface, and the drag force \mathbf{f}_d , which acts on the mechanism, are co-directional vectors. Since the drag force \mathbf{f}_d and the velocity of the water flow \mathbf{v}_w are also co-directional vectors, the following equation is derived:

$$\text{atan2}(\hat{n}_y, \hat{n}_x) = \text{atan2}(\hat{u}_{w,y}, \hat{u}_{w,x}) = \theta_w \quad (5.2)$$

where θ_w is the direction of the water flow. It is mentioned that the above equation holds under the assumption that the submerged mechanism is approximately stabilized, i.e., $\|\mathbf{v}_L\| \approx 0$ and $\|\dot{\mathbf{v}}_L\| \approx 0$.

5.3.2 Gaussian Process Regression

To cope with the noise, associated with the sensor measurements, a non-parametric learning-based estimation of the cable tension is designed based on Gaussian Processes. The concept of GP regression is thoroughly described in Section 3.6.3.

Considering the requirements of the specific application, the GP regression problem is formulated in such a manner that the cable tension is approximated. Hence, the training set D consists of the cable tension $\hat{\mathbf{T}} = \|\mathbf{T}\|\hat{\mathbf{n}}$, measured according to Section 5.3.1, as the output and the state of the vehicle $\mathbf{x} = [\mathbf{p} \ \mathbf{v}]^T$ as the input data:

$$D = \left\{ \mathbf{Z} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times 6}, \mathbf{Y} = [\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_N]^T \in \mathbb{R}^{N \times 3} \right\} \quad (5.3)$$

As mentioned in Section 3.6.3, GPs model functions with a scalar value. Consequently, 2 independent Gaussian Processes are utilized in order to approximate the tension force vector, since the norm of the tension $\|\mathbf{T}\|$ is measured by the load cell and the cable unit vector \mathbf{n} has 2 degrees of freedom. Once a new estimate of the cable tension $\hat{\mathbf{T}}$ is available, the training set D and the GPs are updated online during the sampling procedure.

5.3.3 Extended Kalman Filter

Alternatively, we present a more computationally efficient approach based on a Kalman Filter which fuses the measurements provided by the onboard sensors and approximates the cable tension \mathbf{T} . More precisely, given the known norm of the tension $\|\mathbf{T}\|$, the ultimate goal of the Kalman Filter is to estimate the cable unit vector \mathbf{n} by exploiting Equations 5.1 and 5.2. To this end, defining $\mathbf{x}_L \equiv \mathbf{n} = [n_x \ n_y \ n_z]^T \in \mathbb{R}^3$ as the state vector, the following process and observation models are formulated:

$$\dot{\mathbf{x}}_L = \dot{\mathbf{n}} = \mathbf{w}_L \quad (5.4)$$

$$\mathbf{y}_L = \begin{bmatrix} \|\mathbf{n}\| \\ \text{atan2}(\hat{n}_y, \hat{n}_x) \\ \hat{n}_z \end{bmatrix} + \mathbf{v}_L \quad (5.5)$$

where $\mathbf{w}_L \in \mathbb{R}^3$ and $\mathbf{v}_L \in \mathbb{R}^3$ are additive process and observation Gaussian noises with diagonal covariance matrices. According to the above formulation, the cable unit vector is modeled as a random walk while the equality state constraint $\|\mathbf{n}\| = 1$ is incorporated into the measurement model as an approximately noiseless pseudo-measurement [3]. Owing to the nonlinear observation model, an Extended Kalman Filter (EKF) is deployed.

5.4 Position Control

5.4.1 Geometric Control

Following the approximation of the cable tension \mathbf{T} , a position controller is developed in order to reject the estimated disturbance and accurately stabilize the vehicle above the reference sampling position \mathbf{p}_{ref} . Initially, the geometric controller, presented in [47], [56], which is built upon the differential flatness property of the multicopter dynamics, is implemented with an extension, so as to incorporate the external force [8], [84], particularly the cable tension.

More precisely, the following position and velocity errors are defined:

$$\mathbf{e}_p = \mathbf{p} - \mathbf{p}_{ref} \quad (5.6)$$

$$\mathbf{e}_v = \mathbf{v} - \mathbf{v}_{ref} \quad (5.7)$$

Since the aim is the stabilization of the vehicle to a given position, the reference velocity is equal to zero, $\mathbf{v}_{ref} = [0 \ 0 \ 0]^T$. According to the dynamics of the vehicle, presented in Equation (2.9), the desired attitude of the UAV, expressed by the rotation matrix \mathbf{R}_d , is computed, given a desired yaw angle ψ_d , as follows:

$$\mathbf{b}_{3,d} = \frac{-k_p \mathbf{e}_p - k_v \mathbf{e}_v + m g \mathbf{e}_z - \mathbf{T}}{\|-k_p \mathbf{e}_p - k_v \mathbf{e}_v + m g \mathbf{e}_z - \mathbf{T}\|} \quad (5.8)$$

$$\mathbf{b}_{2,d} = [-s\psi_d \quad c\psi_d \quad 0]^T \quad (5.9)$$

$$\mathbf{b}_{1,d} = \frac{\mathbf{b}_{2,d} \times \mathbf{b}_{3,d}}{\|\mathbf{b}_{2,d} \times \mathbf{b}_{3,d}\|} \quad (5.10)$$

$$\mathbf{R}_d = [\mathbf{b}_{1,d} \quad \mathbf{b}_{3,d} \times \mathbf{b}_{1,d} \quad \mathbf{b}_{3,d}] \quad (5.11)$$

where $k_p, k_v \in \mathbb{R}^+$ are positive gains. The rotation matrix \mathbf{R}_d is eventually converted into desired Euler angles ϕ_d, θ_d, ψ_d for the inner attitude controller of the autopilot. It is mentioned that other external forces, acting on the platform, e.g., air drag forces, are neglected.

5.4.2 Nonlinear Model Predictive Control

Additionally, a Nonlinear Model Predictive Controller (NMPC) is designed as an alternative control strategy in order to both accomplish the precise hovering of the vehicle during the sampling procedure and impose input and state constraints. Thus, aggressive maneuvers of the vehicle above the water surface are avoided and the safety level of the sampling procedure is increased.

The above mentioned NMPC scheme relies on the vehicle's dynamics, i.e., Equation (2.9), with the estimated cable tension \mathbf{T} considered as the only source of external disturbances. Hence, the state vector consists of the vehicle's position, velocity and orientation, i.e., $\mathbf{x} = [\mathbf{p}^T \quad \mathbf{v}^T \quad \phi \quad \theta \quad \psi]^T \in \mathbb{R}^9$, while the input vector

is defined by the desired commands for the inner attitude loop of the autopilot, $\mathbf{v} = [\phi_d \ \theta_d \ \psi_d \ u_{zd}]^T \in \mathbb{R}^4$. The Optimal Control Problem (OCP) to be solved by the NMPC is expressed as:

$$\begin{aligned} \min_{\mathbf{v}} \int_{t_0}^{t_0+T} (\|\mathbf{x}(t) - \mathbf{x}_{ref}(t)\|_{\mathbf{Q}}^2 + \|\mathbf{v}(t)\|_{\mathbf{R}}^2) dt + \|\mathbf{x}(t_0+T) - \mathbf{x}(t_0+T)\|_{\mathbf{P}}^2 \\ \text{s.t.: } \mathbf{x}(t_0) = \mathbf{x}_0, \text{ Equation 2.9, } \mathbf{v} \in \mathbb{U} \end{aligned} \quad (5.12)$$

where $\mathbf{x}_{ref} = [\mathbf{p}_{ref}^T \ \mathbf{0}_{1 \times 3} \ 0 \ 0 \ \psi_{ref}]^T \in \mathbb{R}^9$ is the reference state vector. Similarly to Section 3.6.1, the cost matrices \mathbf{Q} , \mathbf{R} and \mathbf{P} are positive-definite and block-diagonal and, in this case, the state error cost matrix \mathbf{Q} is defined by the following equation:

$$\mathbf{Q} = \text{diag}([\mathbf{Q}_p \ \mathbf{Q}_v \ \mathbf{Q}_o]) \quad (5.13)$$

where the block-diagonal weight matrices \mathbf{Q}_p , \mathbf{Q}_v , and \mathbf{Q}_o penalize the vehicle's position, velocity and orientation error respectively.

In addition to the input bounds, related to the set \mathbb{U} , state constraints are also included in the OCP, so as to restrict the motion of the UAV in the x-y plane, according to the following inequalities:

$$-v_{x,max} \leq v_x \leq v_{x,max} \quad (5.14a)$$

$$-v_{y,max} \leq v_y \leq v_{y,max} \quad (5.14b)$$

By enforcing input and state constraints, the smooth behavior of the NMPC strategy and, thus, the safe execution of the sampling procedure are guaranteed. It should be highlighted that safety is a critical factor for the specific task of sampling given the fact that the UAV operates in an aquatic environment and above the water surface. Consequently, aggressive maneuvers may jeopardize the electric components that comprise the vehicle.

The ACADO Toolkit and the qpOASES solver are again exploited for solving the aforementioned constrained optimization problem and a real-time performance, similar to Section 3.6.1, is achieved.

5.5 Experimental Results

5.5.1 Experimental Setup

In order to demonstrate the capabilities of the above methodologies to approximate the cable tension and stabilize the UAV at the reference sampling location, experiments were carried out with the octorotor of Figure 2.3. The vehicle is equipped with a downward-looking camera, namely the ZED 2 stereocamera, required for the estimation of the water flow. Additionally, a 5 kg load cell, along with the HX711



FIGURE 5.3: The position of the additional sensors required during the water sampling procedure.

amplifier, is utilized in order to measure the norm of the cable tension $\|\mathbf{T}\|$ during the water sampling procedure. The load cell is suitably placed at the middle of the cable. Moreover, the Bar30 High-Resolution 300m Depth/Pressure Sensor, accompanied with an I2C Level Converter, provides the depth d at which the sampling mechanism is submerged. The aforementioned pressure sensor can measure up to 30 Bar with a depth resolution of 2 mm. The pressure sensor is fixed to the lower edge of the cable and, hence, water samples are collected at desired depths. Finally, the A01NYUB ultrasonic ranging sensor is located near the vehicle's body to measure the distance h between the UAV and the water surface level. The measurement range of the ultrasonic sensor is 28-750 cm which evidently implies that the cable length should be less than 750 cm. The exact location of the aforementioned sensors in the assembly is better illustrated in Figure 5.3. All the additional sensors, i.e. the ultrasonic, the pressure sensor and the load cell, are directly connected with an Arduino Uno [57] and the respective measurements are transferred to the embedded Jetson AGX Xavier computer via serial communication.

5.5.2 Geometric Control with Gaussian Processes

Simulation Experiments

Initially, a comparative study was conducted in the simulation environment of Figure 2.2 between the geometric controller of Section 5.4.1 with and without the term of the cable tension, estimated with the aid of GPs (Section 5.3.2). The deployed control scheme is demonstrated in Figure 5.4. As depicted in Figures 5.5a, 5.5b, the position controller augmented with the prediction of the GPs was capable of stabilizing

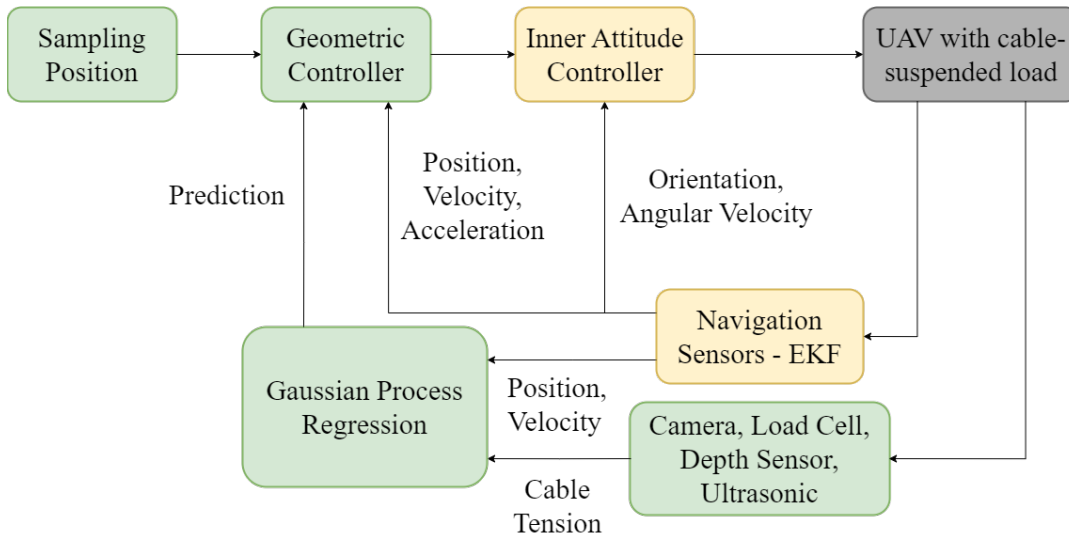


FIGURE 5.4: The geometric controller of Section 5.4.1 augmented with the predictions of the cable tension according to Section 5.3.2.

accurately the vehicle above the sampling position, whereas significant steady-state position errors $\mathbf{p} - \mathbf{p}_{ref}$ were observed during the same scenario when the disturbance term \mathbf{T} was not considered. The estimation of the disturbance, using Gaussian Processes, was successful since the actual tension of the cable, obtained directly by the Gazebo Simulator, lies inside the confidence bounds of the Gaussian Processes, as illustrated in Figures 5.5c, 5.5d. It is mentioned that throughout the simulation scenario, the mass of the mechanism was increased in order to simulate the water sampling procedure.

Real-World Experiments

Following the validation in the Gazebo simulation environment, a comparative experimental study between the geometric controller with and without the term, related to the disturbance, was conducted with the octorotor in real-world conditions. More specifically, the experiments were realized in the Ladonas river, where significant drag forces were applied to the submerged load, due to the velocity of the water flow, and hence an inclined configuration of the cable was observed as depicted in Figure 5.1. It is mentioned that the experiments were conducted successively and, thus, the external conditions, and especially the wind forces, could be considered similar. Additionally, the same positive gains k_p, k_v were utilized and, consequently, a fair comparison between the two controllers was ensured. Lastly, a 1 liter bottle of $m_L = 1.2\text{kg}$ was used as a substitute for a sampling mechanism.

Initially, the position controller, which does not account for the tension of the cable, was employed. As shown in Figure 5.6a, a steady-state error of approximately 1 meter was noticed in the y-axis of the world frame \mathbf{W} , which is a reasonable observation since the direction of the main water flow is almost parallel to the y-axis. In contrast, when the disturbance term was estimated with the aid of GPs (Figures

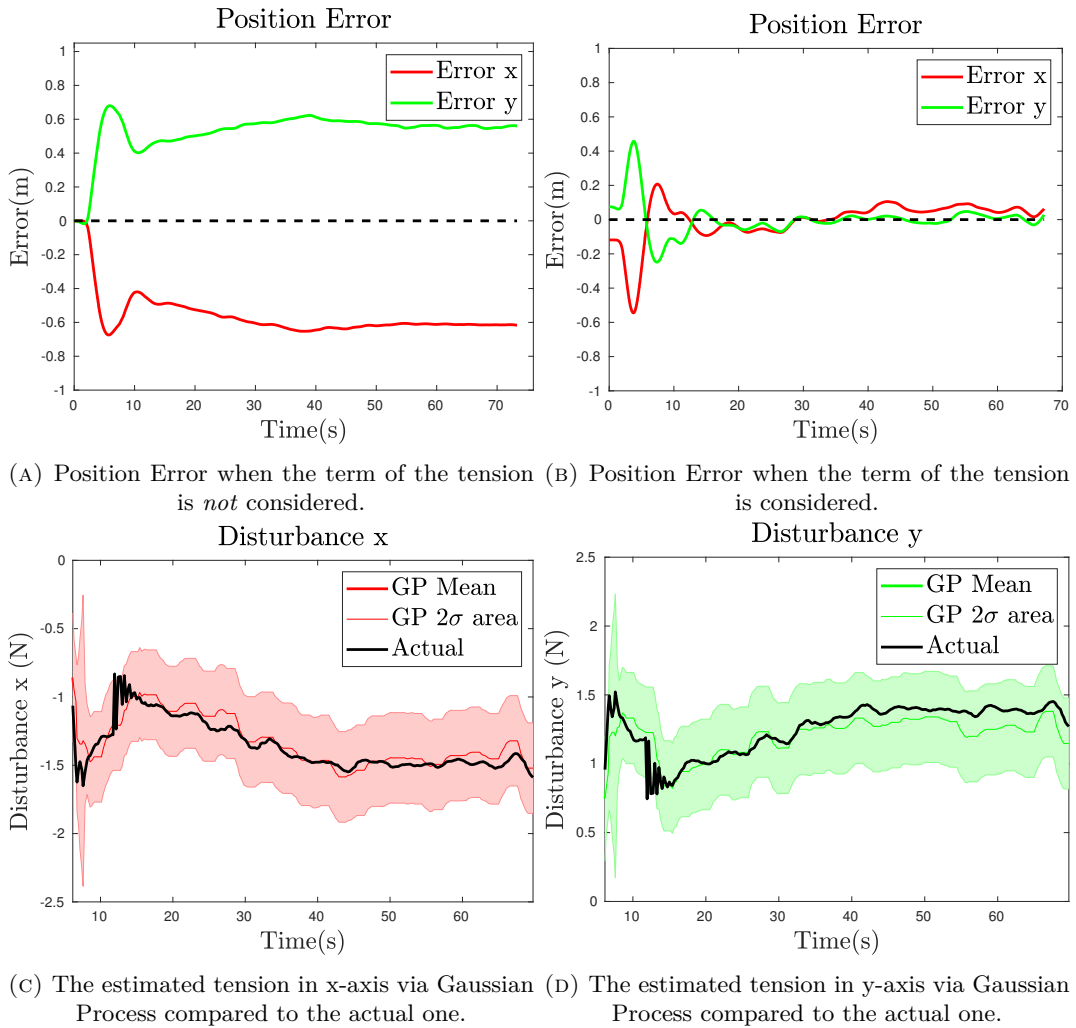


FIGURE 5.5: The position error $\mathbf{p} - \mathbf{p}_{ref}$ and the estimated disturbances \mathbf{T} during the simulation scenario.

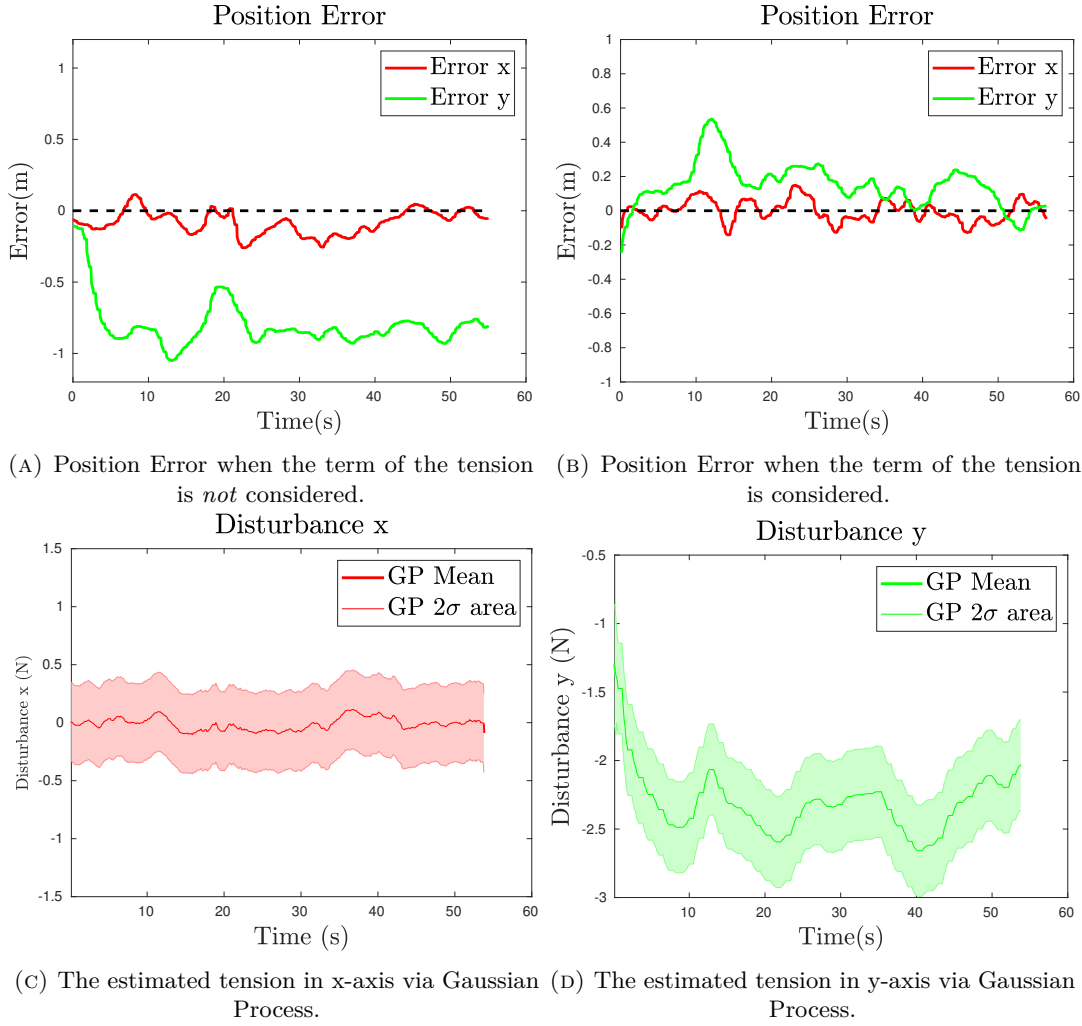


FIGURE 5.6: The position error $\mathbf{p} - \mathbf{p}_{ref}$ and the estimated disturbances \mathbf{T} during the experimental scenario in a river.

5.6c, 5.6d), the position error was significantly reduced and the vehicle was stabilized to the reference position, as depicted in Figure 5.6b. Consequently, the disturbance estimation technique and the performance of the position controller can be considered successful.

The aforementioned comparative studies that validate the efficacy of the geometric controller, augmented with the GP predictions, are also illustrated in the following video <https://youtu.be/4BynPU0ec0k>.

5.5.3 NMPC with EKF

Afterward, we examined the performance of the NMPC scheme of Section 5.4.2 to stabilize the vehicle at the reference sampling position, given the estimates of the cable tension according to the Kalman Filter of Section 5.3.3. Throughout the subsequent experiments, a real sampling device of $m_L = 0.25\text{kg}$, suspended through a $l = 3.8\text{m}$ cable, was considered. The mechanism accomplishes water sampling through the use of a solenoid which reveals a hole and enables the water flow inside a glass tube.

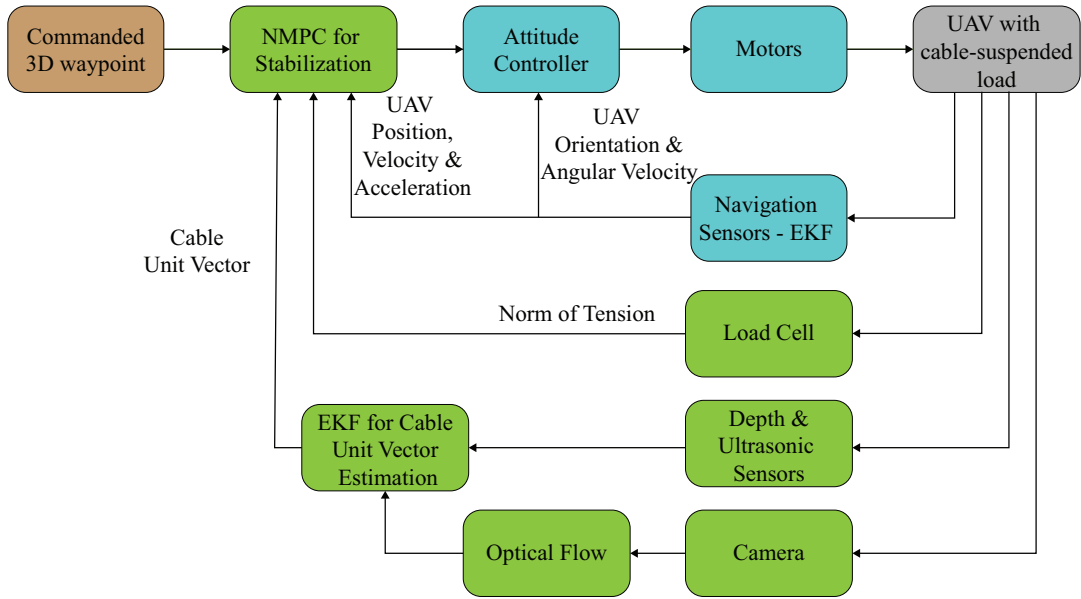


FIGURE 5.7: An overview of the deployed NMPC scheme (Section 5.4.2), which incorporates the estimates of the cable tension according to Section 5.3.3 and achieves the precise stabilization of the vehicle during water sampling.

An overall view of the control architecture is demonstrated in Figure 5.7. In order to validate the identification of the cable tension and the accurate position control of the vehicle during the sampling procedure, experiments were realized in the most common aquatic environments, namely in a sea and in a river.

In the former case, despite the existence of waves, the drag forces, generated by the water flow and acting on the mechanism, were negligible. Consequently, the cable was maintained parallel to the z axis of the world frame during the whole experiment and, hence, the z component of the cable tension $\mathbf{T} = \|\mathbf{T}\|\mathbf{n}$ was the dominant disturbance acting on the UAV, as properly estimated by the additional sensors and the deployed EKF (Figure 5.8d). Regarding the performance of the NMPC, the UAV was accurately stabilized above the desired 3D sampling location, as depicted in Figure 5.8.

In the latter case, the drag forces, induced by the river flow, were considerable and led to an inclined configuration of the cable. As a result, horizontal disturbances were also estimated by the onboard sensors and the EKF, as illustrated in Figure 5.9d. Despite the tension of the cable, the deployed NMPC successfully stabilized the vehicle above the desired position, as depicted in Figure 5.9.

The experiments conducted in the sea and in the Ladonas river are better presented in the relevant videos <https://youtu.be/IB7a72ZvBQc> and <https://youtu.be/EFHL8ckaWpM>.

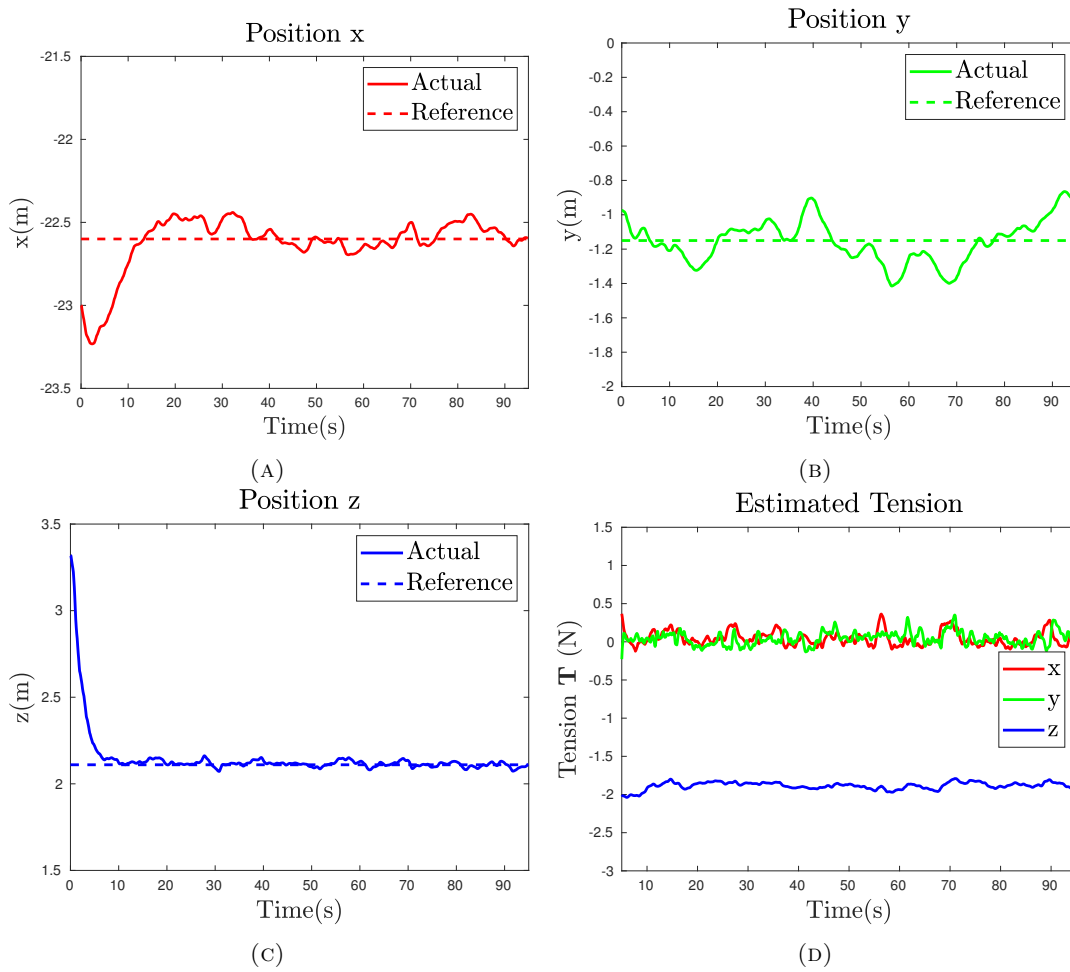


FIGURE 5.8: The actual position of the UAV \mathbf{p} compared to the reference sampling location \mathbf{p}_{ref} and the cable tension estimated by the additional sensors and the Extended Kalman Filter during the stabilization experiment in a beach location. (a) Position x. (b) Position y. (c) Position z. (d) Estimated cable tension.

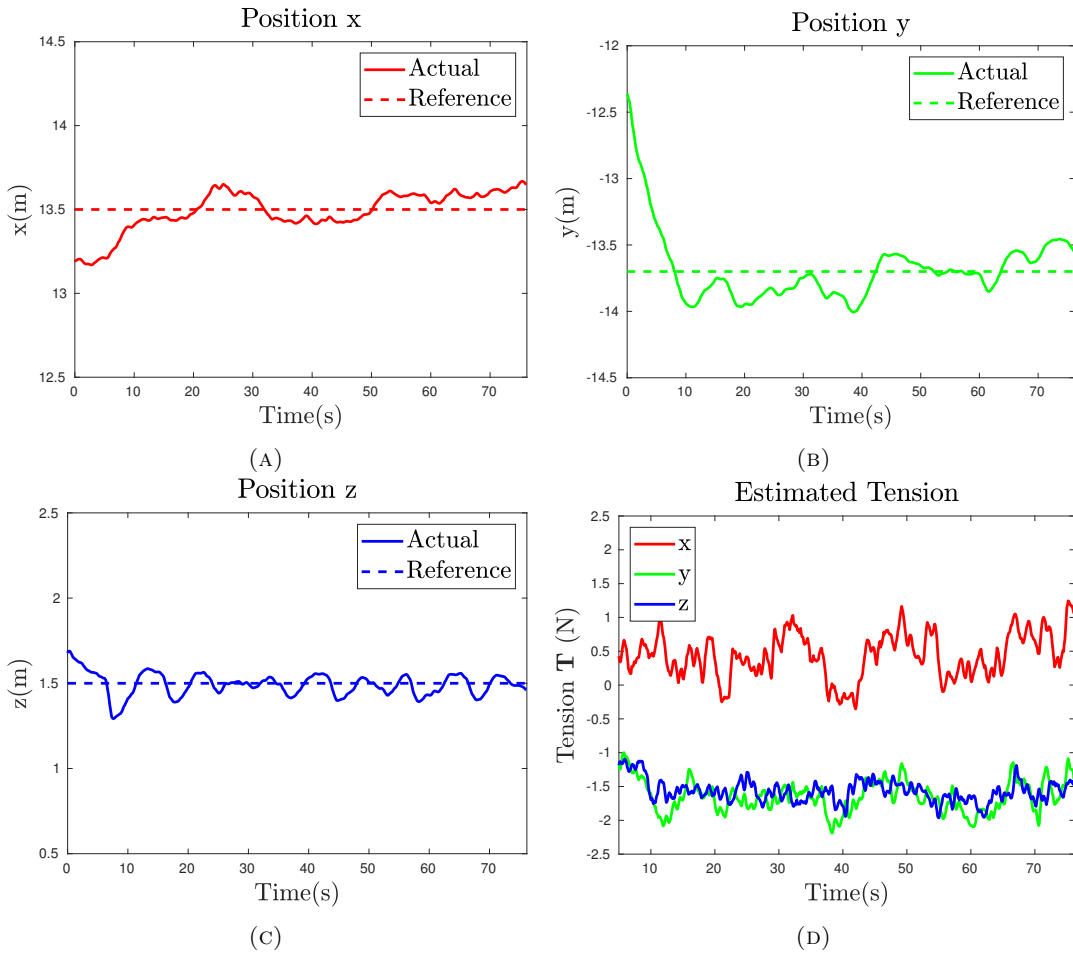


FIGURE 5.9: The actual position of the UAV \mathbf{p} compared to the reference sampling location \mathbf{p}_{ref} and the cable tension estimated by the additional sensors and the Extended Kalman Filter during the stabilization experiment in the Ladonas river. (a) Position x. (b) Position y. (c) Position z. (d) Estimated cable tension.

5.5.4 Autonomous Water Sampling Mission

In this section we merge the outcomes of Chapters 3 and 5 into a fully autonomous solution for water sampling missions by using a UAV with a cable-suspended mechanism equipped with the necessary sensors, perception and control algorithms. Towards this goal, we design a completely autonomous and unified framework which addresses all of the aspects that comprise such a mission with a UAV, particularly, the control of the vehicle during the sampling procedure and the safe transportation of the sample. Consequently, in case of a water contamination incident, the proposed framework can automate the water sampling mission and protect the field personnel since contact with the contaminated area and, thus, with possible pathogenic microorganisms is avoided.

More precisely, consider the case in which an emergency alert occurs regarding a possible contamination in an aquatic environment. In order to identify the existence or not of pathogenic microorganisms, a location is indicated where water sampling should be conducted. Since safeguarding the health of field personnel is of utmost importance, a UAV with a cable-suspended mechanism is deployed instead so as to perform sampling. More precisely, the following steps comprise the water sampling mission:

- The field personnel, i.e., the First Responders, point out a reference sampling location \mathbf{p}_{ref} via a Ground Control Station (a system which enables users to monitor the flight status and interact with the UAV). Additionally, they indicate a final position \mathbf{p}_f where the collection of the sample should be realized for post-processing (typically the position where the UAV takes off). It is mentioned that the water sampling mission can be easily extended for multiple sampling locations or other positions where the UAV should be transferred with minimum swing, e.g., for monitoring.
- The UAV should navigate from its initial position \mathbf{p}_0 to the reference sampling location \mathbf{p}_{ref} while minimizing the cable angles $\boldsymbol{\eta}_L$ and the cable angular velocity $\boldsymbol{\omega}_L$, in order to guarantee the safe transportation of both the vehicle and the mechanism.
- As the UAV approaches the sampling location \mathbf{p}_{ref} , the disturbances acting on the submerged mechanism and transferred through the cable, i.e., the tension $\|\mathbf{T}\|\mathbf{n}$, should be identified and rejected so as to achieve accurate position control during the sampling procedure.
- Eventually, upon the collection of the sample, the UAV should similarly navigate towards the final position \mathbf{p}_f in a safe manner.

In order to fulfill the aforementioned objectives the NMPC schemes, presented in Sections 3.6.1 and 5.4.2, are exploited to achieve the swing-transportation of the sampling mechanism and the precise stabilization of the UAV during the sampling

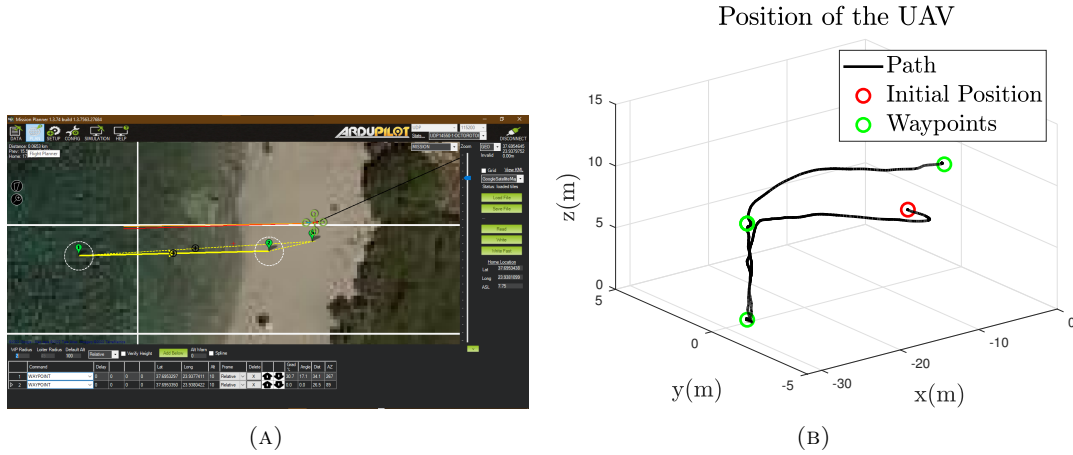


FIGURE 5.10: Autonomous water sampling mission in a beach location. (a) The mission commanded via the Ground Control Station. (b) The 3D path of the vehicle during the mission.

respectively. The state of the cable and the tension are available with the aid of the Kalman Filters of Sections 3.3.1 and 5.3.3.

Real-World Experiments

The efficacy of the individual methodologies, i.e., Kalman Filters and NMPC schemes, to estimate the cable state and tension and control the vehicle throughout the whole water sampling mission was validated with the octorotor vehicle in a beach location.

According to the above analysis, 2 waypoints were drawn via the Mission Planner [59] (a Ground Control Station compatible with the autopilot of the vehicle), as depicted in Figure 5.10a. The first waypoint is located at a certain height above the reference sampling position while the second one indicates the final position where the sample is collected by the human operators. Firstly, the NMPC scheme of Section 3.6.1 navigated the vehicle from its initial position to the first waypoint while minimizing the oscillations of the mechanism, estimated according to Section 3.3.1. Once the UAV approached the first waypoint, the vehicle descended and hovered above the desired sampling location by deploying the position controller of Section 5.4.2. Afterwards, the UAV ascended towards the first waypoint and, eventually, was safely transported to the second waypoint. The 3D path of the UAV is depicted in Figure 5.10b, where the 2 waypoints and the sampling position are demonstrated.

The aforementioned water sampling mission is illustrated in the following video <https://youtu.be/l3XI6a0Btp8>. Additional missions, conducted in the water treatment plant of Thessaloniki and in the Axios river correspondingly, are available on <https://youtu.be/WyGah4iFNl0> and <https://youtu.be/5wHLSPGcLC0>.

Chapter 6

Conclusions and Future Work

In this dissertation, we explored a wide variety of control methodologies and state estimation techniques tailored specifically for Unmanned Aerial Vehicles (UAVs) with cable-suspended loads that operate in real-world environments. Our ultimate goal was to equip such robotic platforms with the necessary autonomy and enable their safe deployment within the context of 3 distinct applications: (a) swing-free transportation of the load, (b) tracking of a moving ground target with the goal of releasing the load towards the former, and (c) water sampling from aquatic environments. Through rigorous analysis, we elucidated the challenges for each task, arising from the existence of the cable-suspended load, and we emphasized the importance of reliable state estimation techniques that, based on onboard sensors, provide feedback to appropriately designed controllers, which eventually handle the coupled system dynamics and fulfill the desired tasks.

Particularly, we designed a frame-based methodology, using a conventional camera, to estimate the cable angular motion during the aerial transportation of loads. We employed a Convolutional Neural Network (CNN) for the robust detection of the cable in outdoor environments and we developed a Kalman Filter to estimate its complete state.

Considering the computational demands of CNNs, we alternatively leveraged a dynamic vision sensor. Building upon the asynchronous nature of DVS, we designed a computationally lightweight event processing pipeline that persistently identifies the cable and approximates its trajectory in the image plane in a quasi-continuous manner with the aid of a Bézier curve. Thus, we considerably expedited the estimation of the cable state, compared to the frame-based solution, which results in improved control performance.

In order to close the loop and attain the swing-free transportation of the load towards reference positions, we developed both model-free and model-based control schemes. Initially, we investigated a data-driven approach, reformulated for the addressed task, based on deep Reinforcement Learning (RL) due to its model-free nature and its ability to cope with complex dynamics. Afterwards, we incorporated domain

randomization into the RL framework so as to produce a more robust and generalized policy.

Given the drawbacks of RL-based schemes, e.g., reward engineering, training in simulation environments, and lack of performance guarantees, we conversely designed a nonlinear Model Predictive Controller (NMPC). Since the performance of model-based controllers is susceptible to the adopted dynamic model, we employed Gaussian Processes (GPs) to approximate the discrepancies between the nominal and actual dynamics. Weighted and sparse GP regression was exploited to achieve both real-time performance and adaptability to the outdoor conditions. The GP predictions were appropriately integrated into the NMPC in order to handle the unmodelled dynamics.

Furthermore, we explored the following of a randomly moving ground vehicle which does exchange any information related to its state. We developed an NMPC-based framework, which relies on a downward-looking camera, in order to predict the future target motion based on a CNN, a tracker, and a Bézier curve. The future trajectory of the target was incorporated directly into an NMPC scheme, which minimizes the tracking error between the ground vehicle and the load. Ultimately, once a condition, related to the predicted trajectories of the load and the target, was satisfied, the opening of a gripper and the release of the load towards the ground vehicle were triggered.

Afterward, we successfully addressed the task of conducting water sampling in aquatic environments, including fast-floating ones, i.e., rivers. We augmented the platform's sensor suite with additional low-cost sensors and fused the respective measurements using 2 sensor fusion techniques, namely GP regression and Kalman Filter, in order to mitigate the noise effects and estimate the cable tension. We employed a geometric controller which rejects the disturbance of the cable and accurately stabilizes the vehicle during the sampling procedure. In addition, we presented an NMPC scheme in order to impose state constraints and guarantee the safety of the vehicle above the water surface. Finally, we combined the methodologies presented throughout the dissertation to completely automate sampling missions during emergency situations.

In conclusion, all the above methodologies were thoroughly validated in real-world environments under the influence of challenging conditions. It is also mentioned that the state estimation algorithms involve the integration of cameras and low-cost sensors, commonly encountered in robotics, and, thus can be seamlessly implemented. Moreover, the control schemes are built upon an inner attitude controller provided by the majority of the recent autopilots. Hence, the proposed methodologies feature minimal prerequisites and can be easily deployed. Consequently, we believe that this dissertation contributes to the field of multirotors with cable-suspended loads and constitutes a step forward towards the autonomous deployment of these robotic platforms for practical and real-world applications.

Looking ahead, a general topic for discussion revolves around the incorporation of the presented or similar methodologies directly into the autopilots, without the need of an external embedded computer. This integration can significantly accelerate the commercial and operational usage of multirotors with suspended loads, resulting in their widespread deployment in the industry.

Additionally, it should be noted that the applicability of these platforms is not limited to the studied tasks and can be extended to other applications. For instance, a possible scenario is the operation of the UAV in GPS-denied environments, where the visual information obtained by the camera should be exploited in order to estimate both the cable state and the odometry of the vehicle. Another task is the navigation of the multirotor in cluttered environments, perceived with the aid of a forward-looking camera or a LIDAR, where the volume of the whole platform should be considered and appropriately integrated into constrained control schemes to avoid collisions with the surrounding obstacles. Finally, a pulley system can be integrated into the upper edge of the cable, which enables the UAV to dynamically alter the cable length and, thus, navigate through constrained pathways, but also entails control schemes that handle the additional degree of freedom.

Regarding the aerial transportation of loads, presented in Chapter 3, future work could extend to exploring control schemes that do not restrict the cable angular motion but attain more agile maneuvers of the load so as to achieve obstacle avoidance, navigation through narrow passages, or interaction of the load with surrounding objects. However, field-of-view constraints are required for this case to guarantee the continuous detection of the cable. Moreover, future studies might focus on incorporating the covariances of the GP posterior distributions into the GP-based NMPC of Section 3.6.4 in order to investigate safety-critical applications.

As for the tracking of the ground target, addressed in Chapter 4, a possible future direction is the substitution of the conventional camera with a dynamic vision sensor so as to expedite both the estimation of the cable state and the prediction of the future target motion. However, the transition to this configuration is not straightforward since the simultaneous identification of both the cable and the target can be extremely challenging owing to the novel acquisition of visual data with event cameras. Another interesting scenario involves considering a target which does not move randomly but intentionally attempts to evade the UAV, thereby leading to an adversarial setting. Consequently, more aggressive maneuvers of the multirotor and the load are required in order to achieve the following of the target.

In Chapter 5, where the water sampling is discussed, future research endeavors may consider a sampling device which provides real-time measurements, related to the water quality. In this case, the multirotor must navigate safely, instead of being stabilized, so as to seek and eventually locate the source of pollution. Finally, in the

event that the mechanism becomes lodged in the seabed, onboard sensors should detect this occurrence and command the release of the cable through a suitably designed mechanism.

Appendices

Appendix A

List of Publications

My research has resulted to 7 peer-reviewed papers for international conferences world-wide and 4 journal papers. The list of publications up to date is as follows:

Journal Papers

[J1] **Panetsos F**, Rousseas P, Karras G, Bechlioulis C, Kyriakopoulos KJ. A Vision-Based Motion Control Framework for Water Quality Monitoring Using an Unmanned Aerial Vehicle. *Sustainability*. 2022; 14(11):6502. <https://doi.org/10.3390/su14116502>

[J2] **Panetsos, F.**, Karras, G.C., Kyriakopoulos, K.J., Oikonomides, O., Kolios, P., Eliades, D. et al. (2023) A motion control framework for autonomous water sampling and swing-free transportation of a multicopter UAV with a cable-suspended mechanism. *Journal of Field Robotics*, 40, 1209–1230. <https://doi.org/10.1002/rob.22182>

[J3] **F. Panetsos**, G. C. Karras and K. J. Kyriakopoulos, "Aerial Transportation of Cable-Suspended Loads With an Event Camera," in *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 231-238, Jan. 2024, doi: 10.1109/LRA.2023.3333245.

[J4] **F. Panetsos**, G. C. Karras and K. J. Kyriakopoulos, "GP-based NMPC for Aerial Transportation of Cable-Suspended Loads", in *IEEE Robotics and Automation Letters*. (**under review**)

Conference Papers

[C1] **F. Panetsos**, G. C. Karras and K. J. Kyriakopoulos, "A Deep Reinforcement Learning Motion Control Strategy of a Multi-rotor UAV for Payload Transportation with Minimum Swing," 2022 30th Mediterranean Conference on Control and Automation (MED), Vouliagmeni, Greece, 2022, pp. 368-374, doi: 10.1109/MED54222.2022.9837220.

[C2] **F. Panetsos**, G. C. Karras, S. N. Aspragkathos and K. J. Kyriakopoulos, "Precise Position Control of a Multi-rotor UAV with a Cable-suspended Mechanism During Water Sampling," 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 1780-1786, doi: 10.1109/IROS47612.2022.9981057.

[C3] S. N. Aspragkathos, M. Sinani, G. C. Karras, **F. Panetsos** and K. J. Kyriakopoulos, "An Event-triggered Visual Servoing Predictive Control Strategy for the Surveillance of Contour-based Areas using Multirotor Aerial Vehicles," 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 2203-2210, doi: 10.1109/IROS47612.2022.9981176.

[C4] A. Mitakidis, S. N. Aspragkathos, **F. Panetsos**, G. C. Karras and K. J. Kyriakopoulos, "A Deep Reinforcement Learning Visual Servoing Control Strategy for Target Tracking Using a Multirotor UAV," 2023 9th International Conference on Automation, Robotics and Applications (ICARA), Abu Dhabi, United Arab Emirates, 2023, pp. 219-224, doi: 10.1109/ICARA56516.2023.10125971.

[C5] **F. Panetsos** et al., "A Nonlinear Model Predictive Control Strategy for Water Sampling Using a UAV with a Slung Mechanism," 2023 31st Mediterranean Conference on Control and Automation (MED), Limassol, Cyprus, 2023, pp. 158-163, doi: 10.1109/MED59994.2023.10185801.

[C6] **F. Panetsos**, G. C. Karras and K. J. Kyriakopoulos, "An NMPC Framework for Tracking and Releasing a Cable-suspended Load to a Ground Target Using a Multirotor UAV," 2024 IEEE International Conference on Robotics and Automation (ICRA) (accepted for publication)

[C7] **F. Panetsos**, G. C. Karras and K. J. Kyriakopoulos, "Robust Deep RL-based Aerial Transportation of Suspended Loads," 2024 32nd Mediterranean Conference on Control and Automation (MED). **(accepted for publication)**

Bibliography

- [1] Pankaj Agarwal and Mukesh Kr. Singh. “A multipurpose drone for water sampling amp; video surveillance”. In: *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*. 2019, pp. 1–5. DOI: [10.1109/ICACCP.2019.8883017](https://doi.org/10.1109/ICACCP.2019.8883017).
- [2] Kostas Alexis et al. “Robust Model Predictive Flight Control of Unmanned Rotorcrafts”. In: *Journal of Intelligent & Robotic Systems* 81 (June 2015). DOI: [10.1007/s10846-015-0238-7](https://doi.org/10.1007/s10846-015-0238-7).
- [3] A.T. Alouani and W.D. Blair. “Use of a kinematic constraint in tracking constant speed, maneuvering targets”. In: *IEEE Transactions on Automatic Control* 38.7 (1993), pp. 1107–1111. DOI: [10.1109/9.231465](https://doi.org/10.1109/9.231465).
- [4] OpenAI: Marcin Andrychowicz et al. “Learning dexterous in-hand manipulation”. In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20. DOI: [10.1177/0278364919887447](https://doi.org/10.1177/0278364919887447). eprint: <https://doi.org/10.1177/0278364919887447>. URL: <https://doi.org/10.1177/0278364919887447>.
- [5] Yusuke Aoki et al. “Nonlinear Model Predictive Control of Position and Attitude in a Hexacopter with Three Failed Rotors**This work was partly supported by JSPS KAKENHI Grant Number 15H02257”. In: *IFAC-PapersOnLine* 51.20 (2018). 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018, pp. 228–233. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2018.11.018>. URL: <https://www.sciencedirect.com/science/article/pii/S2405896318326727>.
- [6] Kai Arulkumaran et al. “Deep Reinforcement Learning: A Brief Survey”. In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 26–38. DOI: [10.1109/MSP.2017.2743240](https://doi.org/10.1109/MSP.2017.2743240).
- [7] Markus Bernard et al. “Autonomous transportation and deployment with aerial robots for search and rescue missions”. In: *Journal of Field Robotics* 28.6 (2011), pp. 914–931. DOI: <https://doi.org/10.1002/rob.20401>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20401>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20401>.
- [8] Mahdis Bisheban and Taeyoung Lee. “Geometric Adaptive Control for a Quadrotor UAV with Wind Disturbance Rejection”. In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 2816–2821. DOI: [10.1109/CDC.2018.8619390](https://doi.org/10.1109/CDC.2018.8619390).

- [9] Jing Chen, Tianbo Liu, and Shaojie Shen. “Tracking a moving target in cluttered environments using a quadrotor”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 446–453. DOI: [10.1109/IROS.2016.7759092](https://doi.org/10.1109/IROS.2016.7759092).
- [10] Hui Cheng et al. “An autonomous vision-based target tracking system for rotorcraft unmanned aerial vehicles”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 1732–1738. DOI: [10.1109/IROS.2017.8205986](https://doi.org/10.1109/IROS.2017.8205986).
- [11] Yuntian Deng et al. “Weighted Gaussian Process Bandits for Non-stationary Environments”. In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, 2022, pp. 6909–6932. URL: <https://proceedings.mlr.press/v151/deng22b.html>.
- [12] Alexander Dietsche et al. “Powerline Tracking with Event Cameras”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 6990–6997. DOI: [10.1109/IROS51168.2021.9636824](https://doi.org/10.1109/IROS51168.2021.9636824).
- [13] A. Gómez Eguíluz et al. “Why fly blind? Event-based visual guidance for ornithopter robot flight”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 1958–1965. DOI: [10.1109/IROS51168.2021.9636315](https://doi.org/10.1109/IROS51168.2021.9636315).
- [14] Davide Falanga et al. “PAMPC: Perception-Aware Model Predictive Control for Quadrotors”. In: Apr. 2018. DOI: [10.1109/IROS.2018.8593739](https://doi.org/10.1109/IROS.2018.8593739).
- [15] Huazhen Fang et al. “Nonlinear Bayesian estimation: from Kalman filtering to a broader horizon”. In: *IEEE/CAA Journal of Automatica Sinica* 5.2 (2018), pp. 401–417. DOI: [10.1109/JAS.2017.7510808](https://doi.org/10.1109/JAS.2017.7510808).
- [16] Aleksandra Faust et al. “Learning swing-free trajectories for UAVs with a suspended load”. In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 4902–4909. DOI: [10.1109/ICRA.2013.6631277](https://doi.org/10.1109/ICRA.2013.6631277).
- [17] Joachim Ferreau et al. “qpOASES: A parametric active-set algorithm for quadratic programming”. In: *Mathematical Programming Computation* 6 (Dec. 2014). DOI: [10.1007/s12532-014-0071-1](https://doi.org/10.1007/s12532-014-0071-1).
- [18] P. Foehn et al. “Fast Trajectory Optimization for Agile Quadrotor Maneuvers with a Cable-Suspended Payload”. In: *Robotics: Science and Systems*. 2017.
- [19] Scott Fujimoto, Herke van Hoof, and David Meger. “Addressing Function Approximation Error in Actor-Critic Methods”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 1587–1596. URL: <https://proceedings.mlr.press/v80/fujimoto18a.html>.

- [20] G. Gallego et al. “Event-Based Vision: A Survey”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 44.01 (2022), pp. 154–180. ISSN: 1939-3539. DOI: [10.1109/TPAMI.2020.3008413](https://doi.org/10.1109/TPAMI.2020.3008413).
- [21] Gowtham Garimella and Marin Kobilarov. “Towards model-predictive control for aerial pick-and-place”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 4692–4697. DOI: [10.1109/ICRA.2015.7139850](https://doi.org/10.1109/ICRA.2015.7139850).
- [22] Gowtham Garimella, Matthew Sheckells, and Marin Kobilarov. “Robust obstacle avoidance for aerial platforms using adaptive model predictive control”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 5876–5882. DOI: [10.1109/ICRA.2017.7989692](https://doi.org/10.1109/ICRA.2017.7989692).
- [23] Sergio Garrido-Jurado et al. “Automatic generation and detection of highly reliable fiducial markers under occlusion”. In: *Pattern Recognition* 47 (June 2014), 2280–2292. DOI: [10.1016/j.patcog.2014.01.005](https://doi.org/10.1016/j.patcog.2014.01.005).
- [24] E. Michael Gertz and Stephen J. Wright. “Object-Oriented Software for Quadratic Programming”. In: *ACM Trans. Math. Softw.* 29.1 (2003), 58–81. ISSN: 0098-3500. DOI: [10.1145/641876.641880](https://doi.org/10.1145/641876.641880). URL: <https://doi.org/10.1145/641876.641880>.
- [25] Farhad Goodarzi, Daewon Lee, and and Lee. “Geometric Control of a Quadrotor UAV Transporting a Payload Connected via Flexible Cable”. In: *International Journal of Control Automation and Systems* (July 2014). DOI: [10.1007/s12555-014-0304-0](https://doi.org/10.1007/s12555-014-0304-0).
- [26] ArduPilot Partners group. *ArduPilot Documentation*. 2016. URL: <https://ardupilot.org/ardupilot/>.
- [27] M. Eusebia Guerrero-Sánchez et al. “Swing-attenuation for a quadrotor transporting a cable-suspended payload”. In: *ISA Transactions* 68 (2017), pp. 433–449. ISSN: 0019-0578. DOI: <https://doi.org/10.1016/j.isatra.2017.01.027>. URL: <https://www.sciencedirect.com/science/article/pii/S0019057817301635>.
- [28] Divam Gupta. *A Beginner’s guide to Deep Learning based Semantic Segmentation using Keras*. 2019. URL: <https://divamgupta.com/image-segmentation/2019/06/06/deep-learning-semantic-segmentation-keras.html>.
- [29] Zhichao Han et al. “Fast-Tracker: A Robust Aerial System for Tracking Agile Target in Cluttered Environments”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 328–334. DOI: [10.1109/ICRA48506.2021.9561948](https://doi.org/10.1109/ICRA48506.2021.9561948).
- [30] João F. Henriques et al. “High-Speed Tracking with Kernelized Correlation Filters”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.3 (2015), pp. 583–596. DOI: [10.1109/TPAMI.2014.2345390](https://doi.org/10.1109/TPAMI.2014.2345390).
- [31] Daniel Hentzen et al. “Disturbance Estimation and Rejection for High-Precision Multirotor Position Control”. In: *2019 IEEE/RSJ International Conference on*

- Intelligent Robots and Systems (IROS)*. 2019, pp. 2797–2804. DOI: [10.1109/IROS40897.2019.8968471](https://doi.org/10.1109/IROS40897.2019.8968471).
- [32] Lukas Hewing, Juraj Kabzan, and Melanie N. Zeilinger. “Cautious Model Predictive Control Using Gaussian Process Regression”. In: *IEEE Transactions on Control Systems Technology* 28.6 (2020), pp. 2736–2743. DOI: [10.1109/TCST.2019.2949757](https://doi.org/10.1109/TCST.2019.2949757).
- [33] B. Houska, H.J. Ferreau, and M. Diehl. “ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization”. In: *Optimal Control Applications and Methods* 32.3 (2011), pp. 298–312.
- [34] Jemin Hwangbo et al. “Control of a Quadrotor With Reinforcement Learning”. In: *IEEE Robotics and Automation Letters* 2.4 (2017), pp. 2096–2103. DOI: [10.1109/LRA.2017.2720851](https://doi.org/10.1109/LRA.2017.2720851).
- [35] Jetson Xavier. <https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit>.
- [36] L.A. Johnston and V. Krishnamurthy. “An improvement to the interacting multiple model (IMM) algorithm”. In: *IEEE Transactions on Signal Processing* 49.12 (2001), pp. 2909–2923. DOI: [10.1109/78.969500](https://doi.org/10.1109/78.969500).
- [37] Alexander B. Jung et al. *imgaug*. <https://github.com/aleju/imgaug>. Online; accessed 01-Feb-2020. 2020.
- [38] Juraj Kabzan et al. “Learning-Based Model Predictive Control for Autonomous Racing”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3363–3370. DOI: [10.1109/LRA.2019.2926677](https://doi.org/10.1109/LRA.2019.2926677).
- [39] Mina Samir Kamel, Michael Burri, and Roland Siegwart. “Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles”. In: vol. 50. July 2017. DOI: [10.1016/j.ifacol.2017.08.849](https://doi.org/10.1016/j.ifacol.2017.08.849).
- [40] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2015).
- [41] N. Koenig and A. Howard. “Design and use paradigms for Gazebo, an open-source multi-robot simulator”. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3. 2004, 2149–2154 vol.3. DOI: [10.1109/IROS.2004.1389727](https://doi.org/10.1109/IROS.2004.1389727).
- [42] Jacob E. Kooi and Robert Babuška. “Inclined Quadrotor Landing using Deep Reinforcement Learning”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 2361–2368. DOI: [10.1109/IROS51168.2021.9636096](https://doi.org/10.1109/IROS51168.2021.9636096).
- [43] Cengiz Koparan et al. “In Situ Water Quality Measurements Using an Unmanned Aerial Vehicle (UAV) System”. In: *Water* 10.3 (2018). ISSN: 2073-4441. DOI: [10.3390/w10030264](https://doi.org/10.3390/w10030264). URL: <https://www.mdpi.com/2073-4441/10/3/264>.
- [44] Till Kroeger et al. “Fast Optical Flow Using Dense Inverse Search”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 471–488. ISBN: 978-3-319-46493-0.

- [45] Seung Jae Lee and H. Jin Kim. “Autonomous swing-angle estimation for stable slung-load flight of multi-rotor UAVs”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 4576–4581. DOI: [10.1109/ICRA.2017.7989532](https://doi.org/10.1109/ICRA.2017.7989532).
- [46] Seung Jae Lee et al. “Robust acceleration control of a hexarotor UAV with a disturbance observer”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. 2016, pp. 4166–4171. DOI: [10.1109/CDC.2016.7798901](https://doi.org/10.1109/CDC.2016.7798901).
- [47] Taeyoung Lee, Melvin Leok, and N. Harris McClamroch. “Geometric tracking control of a quadrotor UAV on SE(3)”. In: *49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 5420–5425. DOI: [10.1109/CDC.2010.5717652](https://doi.org/10.1109/CDC.2010.5717652).
- [48] Guanrui Li, Alex Tunchez, and Giuseppe Loianno. *PCMP: Perception-Constrained Model Predictive Control for Quadrotors with Suspended Loads using a Single Camera and IMU*. July 2021.
- [49] Xue Li et al. “A Novel Dynamic Watermarking-Based EKF Detection Method for FDIAs in Smart Grid”. In: *IEEE/CAA Journal of Automatica Sinica* 9.7 (2022), pp. 1319–1322. DOI: [10.1109/JAS.2022.105704](https://doi.org/10.1109/JAS.2022.105704).
- [50] Zhongyu Li et al. “Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 2811–2817. DOI: [10.1109/ICRA48506.2021.9560769](https://doi.org/10.1109/ICRA48506.2021.9560769).
- [51] Xiao Liang et al. “Nonlinear Hierarchical Control for Unmanned Quadrotor Transportation Systems”. In: *IEEE Transactions on Industrial Electronics* 65.4 (2018), pp. 3395–3405. DOI: [10.1109/TIE.2017.2752139](https://doi.org/10.1109/TIE.2017.2752139).
- [52] T. Lillicrap et al. “Continuous control with deep reinforcement learning”. In: *CoRR* abs/1509.02971 (2016).
- [53] Florian Mahlknecht et al. “Exploring Event Camera-Based Odometry for Planetary Robots”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 8651–8658. DOI: [10.1109/LRA.2022.3187826](https://doi.org/10.1109/LRA.2022.3187826).
- [54] R. Mahony, V. Kumar, and P. Corke. “Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor”. In: *IEEE Robotics Automation Magazine* 19.3 (2012), pp. 20–32. DOI: [10.1109/MRA.2012.2206474](https://doi.org/10.1109/MRA.2012.2206474).
- [55] Christopher D. McKinnon and Angela P. Schoellig. “Unscented external force and torque estimation for quadrotors”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 5651–5657. DOI: [10.1109/IROS.2016.7759831](https://doi.org/10.1109/IROS.2016.7759831).
- [56] Daniel Mellinger and Vijay Kumar. “Minimum snap trajectory generation and control for quadrotors”. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 2520–2525. DOI: [10.1109/ICRA.2011.5980409](https://doi.org/10.1109/ICRA.2011.5980409).
- [57] David Mellis and Massimo Banzi. “Arduino: An Open Electronics Prototyping Platform”. In: 2007.
- [58] Shatadal Mishra et al. “Autonomous Vision-Guided Object Collection From Water Surfaces With a Customized Multirotor”. In: *IEEE/ASME Transactions*

- on Mechatronics* 26.4 (2021), pp. 1914–1922. DOI: [10.1109/TMECH.2021.3080701](https://doi.org/10.1109/TMECH.2021.3080701).
- [59] *Mission Planner*. <https://ardupilot.org/planner/>.
- [60] Siddharth Mysore et al. “How to Train your Quadrotor: A Framework for Consistently Smooth and Responsive Flight Control via Reinforcement Learning”. In: *CoRR* abs/2012.06656 (2020). arXiv: [2012.06656](https://arxiv.org/abs/2012.06656). URL: <https://arxiv.org/abs/2012.06656>.
- [61] Fang Nan et al. “Nonlinear MPC for Quadrotor Fault-Tolerant Control”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5047–5054. DOI: [10.1109/LRA.2022.3154033](https://doi.org/10.1109/LRA.2022.3154033).
- [62] Michael Neunert et al. “Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1398–1404. DOI: [10.1109/ICRA.2016.7487274](https://doi.org/10.1109/ICRA.2016.7487274).
- [63] Huan Nguyen et al. “Model Predictive Control for Micro Aerial Vehicles: A Survey”. In: *2021 European Control Conference (ECC)*. 2021, pp. 1556–1563. DOI: [10.23919/ECC54610.2021.9654841](https://doi.org/10.23919/ECC54610.2021.9654841).
- [64] Ivana Palunko, Patricio Cruz, and Rafael Fierro. “Agile Load Transportation : Safe and Efficient Load Manipulation with Aerial Robots”. In: *IEEE Robotics Automation Magazine* 19.3 (2012), pp. 69–79. DOI: [10.1109/MRA.2012.2205617](https://doi.org/10.1109/MRA.2012.2205617).
- [65] Ivana Palunko and Rafael Fierro. “Adaptive Control of a Quadrotor with Dynamic Changes in the Center of Gravity”. In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 2626–2631. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20110828-6-IT-1002.02564>. URL: <https://www.sciencedirect.com/science/article/pii/S1474667016440097>.
- [66] Ivana Palunko, Rafael Fierro, and Patricio Cruz. “Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach”. In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 2691–2697. DOI: [10.1109/ICRA.2012.6225213](https://doi.org/10.1109/ICRA.2012.6225213).
- [67] Xue Bin Peng et al. “Sim-to-Real Transfer of Robotic Control with Dynamics Randomization”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 3803–3810. DOI: [10.1109/ICRA.2018.8460528](https://doi.org/10.1109/ICRA.2018.8460528).
- [68] Karime Pereida, Rikky R. P. R. Duivenvoorden, and Angela P. Schoellig. “High-precision trajectory tracking in changing environments through L1 adaptive feedback and iterative learning”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 344–350. DOI: [10.1109/ICRA.2017.7989044](https://doi.org/10.1109/ICRA.2017.7989044).
- [69] Vicko Prkačin, Ivana Palunko, and Ivan Petrović. “Extended Kalman filter for payload state estimation utilizing aircraft inertial sensing”. In: *2021 Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*. 2021, pp. 1–6. DOI: [10.1109/AIRPHARO52252.2021.9571038](https://doi.org/10.1109/AIRPHARO52252.2021.9571038).

- [70] MAVLink Micro Air Vehicle Protocol. *MAVLink to ROS gateway with proxy for Ground Control Station*. 2013. URL: <https://github.com/mavlink/mavros>.
- [71] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.
- [72] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006, pp. I–XVIII, 1–248. ISBN: 026218253X.
- [73] Yunfan Ren et al. “Vision-encoder-based Payload State Estimation for Autonomous MAV With a Suspended Payload”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 9632–9638. DOI: [10.1109/IROS51168.2021.9636466](https://doi.org/10.1109/IROS51168.2021.9636466).
- [74] J.P. Rodríguez-Gomez et al. “Asynchronous event-based clustering and tracking for intrusion monitoring in UAS”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 8518–8524. DOI: [10.1109/ICRA40945.2020.9197341](https://doi.org/10.1109/ICRA40945.2020.9197341).
- [75] Radu Bogdan Rusu and Steve Cousins. “3D is here: Point Cloud Library (PCL)”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, 2011.
- [76] Fereshteh Sadeghi and Sergey Levine. “CAD2RL: Real Single-Image Flight Without a Single Real Image”. In: *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*. Ed. by Nancy M. Amato et al. 2017. DOI: [10.15607/RSS.2017.XIII.034](https://doi.org/10.15607/RSS.2017.XIII.034). URL: <http://www.roboticsproceedings.org/rss13/p34.html>.
- [77] John Schulman et al. “Proximal Policy Optimization Algorithms”. In: *ArXiv abs/1707.06347* (2017). URL: <https://api.semanticscholar.org/CorpusID:28695052>.
- [78] Elias Small et al. “Aerial navigation in obstructed environments with embedded nonlinear model predictive control”. In: *2019 18th European Control Conference (ECC)*. 2019, pp. 3556–3563. DOI: [10.23919/ECC.2019.8796236](https://doi.org/10.23919/ECC.2019.8796236).
- [79] Clark Youngdong Son et al. “Model Predictive Control of a Multi-Rotor with a Suspended Load for Avoiding Obstacles”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 5233–5238. DOI: [10.1109/ICRA.2018.8460749](https://doi.org/10.1109/ICRA.2018.8460749).
- [80] Yunlong Song et al. “Autonomous Drone Racing with Deep Reinforcement Learning”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 1205–1212. DOI: [10.1109/IROS51168.2021.9636053](https://doi.org/10.1109/IROS51168.2021.9636053).
- [81] Koushil Sreenath, Taeyoung Lee, and Vijay Kumar. “Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load”. In: *52nd IEEE Conference on Decision and Control*. 2013, pp. 2269–2274. DOI: [10.1109/CDC.2013.6760219](https://doi.org/10.1109/CDC.2013.6760219).

- [82] Koushil Sreenath, Nathan Michael, and Vijay Kumar. “Trajectory generation and control of a quadrotor with a cable-suspended load - A differentially-flat hybrid system”. In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 4888–4895. DOI: [10.1109/ICRA.2013.6631275](https://doi.org/10.1109/ICRA.2013.6631275).
- [83] Sihao Sun et al. “Autonomous Quadrotor Flight Despite Rotor Failure With Onboard Vision Sensors: Frames vs. Events”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 580–587. DOI: [10.1109/LRA.2020.3048875](https://doi.org/10.1109/LRA.2020.3048875).
- [84] James Svacha, Kartik Mohta, and Vijay Kumar. “Improving quadrotor trajectory tracking by compensating for aerodynamic effects”. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2017, pp. 860–866. DOI: [10.1109/ICUAS.2017.7991501](https://doi.org/10.1109/ICUAS.2017.7991501).
- [85] Sarah Tang and Vijay Kumar. “Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 2216–2222. DOI: [10.1109/ICRA.2015.7139492](https://doi.org/10.1109/ICRA.2015.7139492).
- [86] Sarah Tang, Valentin Wüest, and Vijay Kumar. “Aggressive Flight With Suspended Payloads Using Vision-Based Control”. In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 1152–1159. DOI: [10.1109/LRA.2018.2793305](https://doi.org/10.1109/LRA.2018.2793305).
- [87] Michalis Titsias. “Variational Learning of Inducing Variables in Sparse Gaussian Processes”. In: *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*. Ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, 2009, pp. 567–574. URL: <https://proceedings.mlr.press/v5/titsias09a.html>.
- [88] Josh Tobin et al. “Domain randomization for transferring deep neural networks from simulation to the real world”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 23–30. DOI: [10.1109/IROS.2017.8202133](https://doi.org/10.1109/IROS.2017.8202133).
- [89] Guillem Torrente et al. “Data-Driven MPC for Quadrotors”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3769–3776. DOI: [10.1109/LRA.2021.3061307](https://doi.org/10.1109/LRA.2021.3061307).
- [90] Florian Tschopp et al. “Hough²Map – Iterative Event-Based Hough Transform for High-Speed Railway Mapping”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2745–2752. DOI: [10.1109/LRA.2021.3061404](https://doi.org/10.1109/LRA.2021.3061404).
- [91] Dimos Tzoumanikas, Qingyue Yan, and Stefan Leutenegger. “Nonlinear MPC with Motor Failure Identification and Recovery for Safe and Aggressive Multicopter Flight”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 8538–8544. DOI: [10.1109/ICRA40945.2020.9196690](https://doi.org/10.1109/ICRA40945.2020.9196690).
- [92] Daniel Villa, A.s Brandao, and Mário Sarcinelli-Filho. “A Survey on Load Transportation Using Multirotor UAVs”. In: *Journal of Intelligent & Robotic Systems* 98 (May 2020), pp. 1–30. DOI: [10.1007/s10846-019-01088-w](https://doi.org/10.1007/s10846-019-01088-w).

-
- [93] Kentaro Wada. *labelme: Image Polygonal Annotation with Python*. <https://github.com/wkentaro/labelme>. 2016.
- [94] Jie Wang, Michael T. H. Fader, and Joshua A. Marshall. “Learning-based model predictive control for improved mobile robot path following using Gaussian processes and feedback linearization”. In: *Journal of Field Robotics* 40.5 (2023), pp. 1014–1033. DOI: <https://doi.org/10.1002/rob.22165>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.22165>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22165>.
- [95] Qianhao Wang et al. “Visibility-aware Trajectory Optimization with Application to Aerial Tracking”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 5249–5256. DOI: [10.1109/IROS51168.2021.9636753](https://doi.org/10.1109/IROS51168.2021.9636753).
- [96] Gan Yu et al. “Nonlinear Backstepping Control of a Quadrotor-Slung Load System”. In: *IEEE/ASME Transactions on Mechatronics* 24.5 (2019), pp. 2304–2315. DOI: [10.1109/TMECH.2019.2930211](https://doi.org/10.1109/TMECH.2019.2930211).
- [97] Burak Yüksel et al. “A nonlinear force observer for quadrotors and application to physical interactive tasks”. In: *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2014, pp. 433–440. DOI: [10.1109/AIM.2014.6878116](https://doi.org/10.1109/AIM.2014.6878116).
- [98] *ZED Stereo Camera*. <https://www.stereolabs.com/zed/>.
- [99] Jun Zeng, Prasanth Kotaru, and Koushil Sreenath. “Geometric Control and Differential Flatness of a Quadrotor UAV with Load Suspended from a Pulley”. In: *2019 American Control Conference (ACC)*. 2019, pp. 2420–2427. DOI: [10.23919/ACC.2019.8815173](https://doi.org/10.23919/ACC.2019.8815173).

Εκτενής Ελληνική Περίληψη

Εισαγωγή

Κίνητρο

Το πεδίο των μη επανδρωμένων εναέριων οχημάτων, γνωστά και ως UAVs, έχει σημειώσει ευρεία ανάπτυξη κατά την τελευταία δεκαετία λόγω της ευέλικτης φύσης τους, του χαμηλού κόστους τους και της δυνατότητας εφαρμογής τους σε μια ποικιλία εργασιών όπως επιθεώρηση υποδομών, περιβαλλοντική παρακολούθηση, αποστολές έρευνας και διάσωσης, αποτύπωση και χαρτογράφηση, και γεωργία. Πιο πρόσφατα, η ενσωμάτωση φορτίων στις εναέριες πλατφόρμες διεύρυνε τα όρια των UAVs, ενίσχυσε τις δυνατότητές τους και επέκτεινε τη χρήση τους σε επιπρόσθετα σενάρια. Για παράδειγμα, τα UAVs μπορούν να συμβάλλουν σημαντικά σε αποστολές έρευνας και διάσωσης μεταφέροντας απαραίτητες προμήθειες, φάρμακα, εξοπλισμό ή άλλα αγαθά. Επιπλέον, τα πολυκόπτερα μπορούν ενδεχομένως να χρησιμοποιηθούν σε περίπτωση μιας περιβαλλοντικής καταστροφής όπου η πρόσβαση των πρώτων ανταποκριτών στο πεδίο είναι επικίνδυνη ή ακόμη και ανέφικτη, π.χ. δειγματοληψία νερού σε μολυσμένες υδάτινες περιοχές.

Σύμφωνα με τις υπάρχουσες προσεγγίσεις, τα UAVs με φορτία μπορούν να ταξινομηθούν σε δύο κατηγορίες με βάση την προσάρτηση του φορτίου στο κύριο σώμα του οχήματος. Συγκεκριμένα, τα φορτία συνήθως είτε είναι στερεά συνδεδεμένα στο σώμα του οχήματος είτε κρέμονται μέσω ενός σχοινιού. Στην πρώτη περίπτωση, ο ενδεδειγμένος μηχανικός σχεδιασμός, συμπεριλαμβανομένων παραμέτρων όπως τα σημεία στήριξης στη δομή του εκάστοτε πολυκοπτέρου, αποτελεί προϋπόθεση για την επιτυχή ενσωμάτωση του φορτίου. Επιπλέον, η απελευθέρωση του φορτίου συνεπάγεται την προσγείωση του οχήματος, η οποία μπορεί να είναι αδύνατη σε καταστάσεις έκτακτης ανάγκης. Αντίθετα, στην δεύτερη περίπτωση, η χρήση ενός σχοινιού είναι μια έτοιμη προς χρήση και μη σύνθετη λύση και, ως εκ τούτου, έχει μαγνητίσει το ενδιαφέρον της ερευνητικής κοινότητας.

Ωστόσο, η αυτόνομη ανάπτυξη πολυκοπτέρων με αναρτημένα μέσω σχοινιού φορτία σε πραγματικά περιβάλλοντα παρουσιάζει ιδιαίτερες προκλήσεις. Από άποψη ελέγχου, πολλά ζητήματα προκύπτουν από την underactuated φύση της δυναμικής του συστήματος, λαμβάνοντας υπόψη τον αυξημένο αριθμό βαθμών ελευθερίας. Πιο συγκεκριμένα, κατά την εναέρια μεταφορά αναρτημένων φορτίων, η αυθαίρετη κίνηση του οχήματος μπορεί να διεγείρει την ταλαντωτική κίνηση του σχοινιού λόγω της περίπλοκης δυναμικής που διέπει ολόκληρο το σύστημα. Σημαντικές γωνιακές μετατοπίσεις του σχοινιού μπορεί να οδηγήσουν σε απρόβλεπτα σενάρια, θέτοντας σε κίνδυνο την ασφάλεια τόσο του οχήματος όσο και του φορτίου, π.χ. το σχοινί μπορεί να πλησιάσει επικίνδυνα τους

βραχίονες και τους κινητήρες του UAV. Για το λόγο αυτό, ανακύπτει η ανάγκη σχεδιασμού εύρωστων σχημάτων ελέγχου που ελαχιστοποιούν τις ταλαντώσεις του σχοινιού και διασφαλίζουν την ασφαλή μεταφορά του φορτίου προς τις θέσεις αναφοράς.

Διάφοροι ελεγκτές υπάρχουν στη βιβλιογραφία που στοχεύουν στην αντιμετώπιση της γωνιακής κίνησης του σχοινιού [92]. Ωστόσο, επικυρώνονται κατά βάση σε εσωτερικά περιβάλλοντα, παραβλέποντας έτσι την παρουσία εξωτερικών διαταραχών. Επιπλέον, η διεξαγωγή πειραμάτων σε εσωτερικές συνθήκες διευκολύνει την εκτίμηση των μεταβλητών κατάστασης, π.χ., χρησιμοποιώντας συστήματα καταγραφής κίνησης. Αντίθετα, όταν τα πολυκόπτερα με αναρτημένα μέσω σχοινιού φορτία αναπτύσσονται σε εξωτερικά περιβάλλοντα και πραγματικές συνθήκες, η εκτίμηση της κατάστασης του σχοινιού με βάση αισθητήρες ενσωματωμένους στο όχημα παραμένει ένα απαιτητικό και άλυτο ζήτημα. Το γεγονός αυτό είναι ιδιαίτερα κρίσιμο, καθώς η πλειοψηφία των ελεγκτών απαιτεί αξιόπιστη και σε πραγματικό χρόνο ανατροφοδότηση κατάστασης. Προς αυτήν την κατεύθυνση, πολλές επιλογές αισθητήρων μπορούν να ληφθούν υπόψη για την εκτίμηση της γωνιακής κίνησης του σχοινιού σε πραγματικά περιβάλλοντα. Ωστόσο, η πιο απλή επιλογή είναι η αξιοποίηση μιας κάμερας, καθώς σχεδόν κάθε UAV είναι εξοπλισμένο με μία βάση κάμερας και, κατά συνέπεια, πολύπλοκες μηχανικές παρεμβάσεις στην πλατφόρμα, π.χ. προσθήκη περιστροφικών κωδικοποιητών [7], [73], αποφεύγονται. Ως εκ τούτου, απαιτούνται κατάλληλα σχεδιασμένες τεχνικές βασισμένες στην όραση, και προσαρμοσμένες στη συγκεκριμένη εφαρμογή της μεταφοράς φορτίων, προκειμένου να παρέχεται εύρωστη ανάδραση της κατάστασης του σχοινιού.

Επιπλέον, η εναέρια μεταφορά φορτίων μπορεί να επεκταθεί ώστε να συμπεριλάβει την παρακολούθηση αυθαίρετα κινούμενων στόχων, με απώτερο σκοπό την εναπόθεση του φορτίου σε αυτούς. Αυτό όχι μόνο διευρύνει την εφαρμοσιμότητα των UAVs αλλά και τα εισάγει σε νέα επιχειρησιακά σενάρια. Η παρακολούθηση στόχων βάσει όρασης με πολυκόπτερα έχει αντιμετωπιστεί εκτενώς στη βιβλιογραφία και αποτελεί ένα καλά μελετημένο πρόβλημα [9], [10], [29], [36]: χωρίς ωστόσο να λαμβάνεται υπόψη η ύπαρξη ενός αναρτημένου μέσω σχοινιού φορτίου. Η ενσωμάτωση ενός φορτίου και, κατά συνέπεια, η εναπόθεσή του στον στόχο συνεπάγονται διάφορες τροποποιήσεις στα υπάρχοντα σχήματα. Ακόμη και για έναν έμπειρο πιλότο drone, η επιτυχής εναπόθεση του φορτίου μπορεί να είναι ιδιαίτερα απαιτητική λόγω των ταλαντώσεων του σχοινιού και της τυχαίας κίνησης του στόχου. Προς αυτή την κατεύθυνση, ένα σχήμα ελέγχου, που χειρίζεται τόσο την προβλεπόμενη τροχιά του στόχου όσο και τη γωνιακή κίνηση του σχοινιού, είναι απαραίτητο για να εκπληρώσει αυτήν την αποστολή.

Επιπλέον, τα UAVs έχουν επίσης αποδειχθεί ότι είναι μια πολλά υποσχόμενη ρομποτική πλατφόρμα για τη συλλογή δειγμάτων από υδάτινα περιβάλλοντα κατά τη διάρκεια ενός περιστατικού μόλυνσης νερού. Η χρήση ενός UAV, αντί της χειροκίνητης συλλογής δειγμάτων, μειώνει την έκθεση των ανταποκριτών σε κίνδυνο, επιταχύνει την απόκριση σε μια καταστροφή και, ως εκ τούτου, συμβάλλει σημαντικά στη διαχείριση της έκτακτης ανάγκης. Δύο προσεγγίσεις αναπτύσσονται συνήθως για τη διεξαγωγή δειγματοληψίας νερού με ένα UAV: (α) ενσωμάτωση μίας βάσης που επιπλέει [1], [43], [58], ειδικά

σχεδιασμένη για ένα συγκεκριμένο όχημα, η οποία επιτρέπει στο UAV να επιπλέει και να προσγειώνεται στην επιφάνεια του νερού και (β) ανάρτηση του μηχανισμού δειγματοληψίας μέσω ενός σχοινιού. Προφανώς, η τελευταία λύση συνεπάγεται ανεμπόδιση ενσωμάτωση από πλευράς μηχανικής πολυπλοκότητας και διατηρεί την πολυχρηστικότητα του UAV σε διάφορες άλλες εργασίες πέρα από εφαρμογές που σχετίζονται με το νερό. Ωστόσο, όταν το UAV με τον αναρτημένο μηχανισμό δειγματοληψίας επιχειρεί σε ένα υδάτινο περιβάλλον με ταχεία ροή νερού, π.χ. σε ένα ποτάμι, σημαντικές υποβρύχιες διαταραχές επιδρούν στη συσκευή και έμμεσα στο όχημα μέσω του σχοινιού. Ως εκ τούτου, για να διασφαλιστεί η ασφαλής εκτέλεση της διαδικασίας δειγματοληψίας, είναι απαραίτητο να χρησιμοποιηθεί μια κατάλληλη τεχνική σύντηξης αισθητήρων που εκτιμά την τάση του σχοινιού χρησιμοποιώντας αισθητήρες ενσωματωμένους στο όχημα, μαζί με μια στρατηγική ελέγχου που επιτυγχάνει την απόρριψη των διαταραχών και την ακριβή σταθεροποίηση του οχήματος.

Εν συντομία, τα UAVs με αναρτημένα μέσω σχοινιού φορτία μπορούν να αξιοποιηθούν για ένα ευρύ φάσμα εφαρμογών και, έτσι, να προσφέρουν λύσεις σε πραγματικά προβλήματα. Ωστόσο, οι προκλήσεις, που απορρέουν από την παρουσία του αναρτημένου φορτίου, εμποδίζουν την ευρεία χρήση τους. Για το σκοπό αυτό, η έρευνά μας προσπαθεί να γεφυρώσει αυτό το χάσμα υποστηρίζοντας την πλατφόρμα με το απαιτούμενο λογισμικό και υλικό και, κατά συνέπεια, να επιτρέψει την αυτόνομη λειτουργία των UAVs στο πλαίσιο τριών συγκεκριμένων εφαρμογών (Εικόνα 1.1): (α) ασφαλή μεταφορά αναρτημένων φορτίων με ελαχιστοποίηση της ταλαντωτικής κίνησης του σχοινιού, (β) παρακολούθηση κινούμενου επίγειου στόχου με σκοπό την εναπόθεση του φορτίου σε αυτόν και (γ) δειγματοληψία νερού από υδάτινα περιβάλλοντα. Η προσέγγισή μας περιλαμβάνει τη βελτίωση των υπάρχουσών μεθοδολογιών και την ανάπτυξη νέων τεχνικών εκτίμησης κατάστασης και ελεγκτών που αντιμετωπίζουν τις προκλήσεις που θέτει το σχοινί ενώ ταυτόχρονα πετυχαίνουν τους στόχους κάθε εφαρμογής. Οι προτεινόμενες μεθοδολογίες επικυρώνονται εξαντλητικά σε εξωτερικές συνθήκες για να αναδειχθεί η αποτελεσματικότητά τους και οι δυνατότητές τους σε πρακτικά σενάρια. Μέσω αυτής της διατριβής, στοχεύουμε να συμβάλουμε στο πεδίο των πολυκοπτέρων με αναρτημένα μέσω σχοινιού φορτία, επεκτείνοντας έτσι τη χρήση τους σε πραγματικά περιβάλλοντα.

Εναέρια Μεταφορά Αναρτημένων Φορτίων

Εισαγωγή

Σε αυτό το κεφάλαιο, εξετάζουμε την μεταφορά ενός αναρτημένου μέσω σχοινοῦ φορτίου χρησιμοποιώντας ένα πολυκόπτερο UAV σε πραγματικά περιβάλλοντα. Η ανάρτηση φορτίων μέσω σχοινοῦν έχει το πλεονέκτημα της ελάχιστης μηχανικής παρέμβασης στην ρομποτική πλατφόρμα και μετριάζει την ανάγκη για περίπλοκους σχεδιασμούς σε αντίθεση με την στερεά σύνδεση των φορτίων στα UAVs. Ωστόσο, η ύπαρξη του αναρτημένου μέσω σχοινοῦ φορτίου απαιτεί σχήματα ελέγχου τα οποία χειρίζονται τον αυξημένο αριθμό βαθμῶν ελευθερίας και την ταλαντωτική κίνηση του σχοινοῦ και, ως εκ τούτου, εγγυώνται την ασφάλεια της πλατφόρμας. Επιπλέον, λαμβάνοντας υπόψη ότι η αξιόπιστη εκτίμηση της κατάστασης του σχοινοῦ είναι απαραίτητη προϋπόθεση για την χρήση ελεγκτῶν ανάδρασης, ανακύπτει η ανάγκη για ανάπτυξη κατάλληλων τεχνικῶν οι οποίες εκτιμούν τη γωνιακή κίνηση του σχοινοῦ με βάση τους ενσωματωμένους στο όχημα αισθητήρες.

Για το σκοπὸ αυτό, παρουσιάζουμε 2 μεθοδολογίες για την εκτίμηση της κατάστασης του σχοινοῦ με τη βοήθεια είτε μιας συμβατικῆς κάμερας είτε ενός Δυναμικοῦ Αισθητήρα Ὁρασης (DVS). Στην πρώτη περίπτωση, η ανίχνευση του σχοινοῦ βασίζεται σε ένα Συνελικτικό Νευρωνικό Δίκτυο (CNN) και η συνολικὴ κατάστασή του λαμβάνεται ἀπὸ ἕνα φίλτρο Kalman. Στην δεύτερη περίπτωση, αναπαριστώντας τις εισερχόμενες ροές γεγονότων (events) με ἕνα νέφος σημείων (point cloud), σχεδιάζεται μια υπολογιστικὰ ελαφριά μέθοδος επεξεργασίας των events που επιτυγχάνει τη γρήγορη ανίχνευση του σχοινοῦ, και στη συνέχεια οι αντίστοιχες μετρήσεις προσεγγίζονται με μια καμπύλη Bézier προκειμένου να εκτιμηθεῖ τελικὰ η συνολικὴ κατάσταση του σχοινοῦ, σε πολὺ μικρότερο χρονικὸ διάστημα σε σύγκριση με τη λύση που βασίζεται στην συμβατικὴ κάμερα.

Ὅσον ἀφορὰ τον ἔλεγχο της ρομποτικῆς πλατφόρμας, διερευνώνται προσεγγίσεις είτε βασισμένες στο μοντέλο του συστήματος είτε ὄχι, οι οποίες αποσκοποῦν στην μεταφορά του φορτίου χωρίς ταλάντωση του σχοινοῦ. Αρχικὰ, με τη βοήθεια βαθιάς Ενισχυτικῆς Μάθησης (RL), εκπαιδεύεται στο περιβάλλον προσομοίωσης μια πολιτικὴ, η οποία δεν στηρίζεται στο μοντέλο του συστήματος και εκπληρώνει την επιθυμητὴ εφαρμογή. Στη συνέχεια, η μέθοδος domain randomization εισάγεται κατὰ τη διαδικασία

της εκμάθησης, ώστε να βελτιωθεί η μεταφερσιμότητα της εκπαιδευμένης πολιτικής σε πραγματικές συνθήκες και να ενισχυθεί η ευρωστία της ως προς τις μεταβλητές ιδιότητες του συστήματος. Στο τέλος, μια προσέγγιση βασισμένη στο μοντέλο του συστήματος, συγκεκριμένα ένας μη γραμμικός Προβλεπτικός Ελεγκτής (NMPC), διατυπώνεται για τη δυναμική της πλατφόρμας. Δεδομένου όμως ότι το πολυκόπτερο με το αναρτημένο μέσω σχοινιού φορτίο επιχειρεί σε υπαίθρια περιβάλλοντα υπό την επίδραση εξωτερικών διαταραχών, αξιοποιούνται Gaussian Processes (GPs) προκειμένου να προσεγγιστούν σε πραγματικό χρόνο τα σφάλματα μεταξύ του ονομαστικού και του πραγματικού δυναμικού μοντέλου του συστήματος. Στη συνέχεια, το διορθωμένο δυναμικό μοντέλο τροφοδοτείται σε ένα σχήμα GP-based NMPC το οποίο επιτυγχάνει τη μεταφορά του φορτίου χωρίς ταλάντωση ανεξάρτητα από την ύπαρξη διαταραχών και των συνθηκών που επικρατούν στο εξωτερικό περιβάλλον όπου χρησιμοποιείται το UAV.

Ορισμός του Προβλήματος

Εξετάζουμε το πρόβλημα ελέγχου της μεταφοράς ενός αναρτημένου μέσω σχοινιού φορτίου προς σημεία αναφοράς χρησιμοποιώντας ένα πολυκόπτερο, όπως απεικονίζεται στην Εικόνα 3.1. Προκειμένου να επιτευχθεί η ασφαλής μεταφορά του φορτίου, είναι απαραίτητο να ληφθεί υπόψη η ταλαντωτική κίνηση του σχοινιού κατά την αυτόνομη πλοήγηση της πλατφόρμας. Διαφορετικά, η αυθαίρετη κίνηση του UAV μπορεί να προκαλέσει έντονες ταλαντώσεις του σχοινιού και να οδηγήσει σε μεγάλες γωνιακές μετατοπίσεις που μπορεί να θέσουν σε κίνδυνο την ασφάλεια ολόκληρου του συστήματος. Για το σκοπό αυτό, ο απώτερος στόχος μας είναι να αναπτύξουμε κατάλληλα σχήματα ελέγχου, είτε βασισμένα στο μοντέλο του συστήματος είτε όχι, τα οποία χειρίζονται την περίπλοκη δυναμική της πλατφόρμας και υπολογίζουν κατάλληλες εντολές $\phi_d, \theta_d, \psi_d, v_{zd}$ για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου προκειμένου να:

- ελαχιστοποιηθεί το σφάλμα $\|\mathbf{p} - \mathbf{p}_{ref}\|$ μεταξύ της πραγματικής θέσης του UAV και της θέσης αναφοράς $\mathbf{p}_{ref} \in \mathbb{R}^3$.
- ελαχιστοποιηθεί η ταχύτητα του σχοινιού $\dot{\mathbf{n}}$, ή ισοδύναμα η γωνιακή ταχύτητα ω_L , καθώς και οι γωνίες του σχοινιού η_L . Το τελευταίο συνεπάγεται ότι το σχοινί θα πρέπει να διατηρείται όσο το δυνατόν πιο κοντά σε μια κατακόρυφη θέση, δηλαδή παράλληλα με τον άξονα z του συστήματος αναφοράς \mathbf{W} και, ως εκ τούτου, οι συνιστώσες n_x, n_y του μοναδιαίου διανύσματος του σχοινιού \mathbf{n} θα πρέπει να ελαχιστοποιηθούν.

Ωστόσο, οι προαναφερθέντες ελεγκτές κλειστού βρόχου απαιτούν εύρωστη και γρήγορη ανατροφοδότηση της κατάστασης του σχοινιού πέρα από την κατάσταση του οχήματος, που παρέχεται απευθείας από τον αυτόματο πιλότο. Λαμβάνοντας υπόψη ότι το octorotor επιχειρεί σε εξωτερικά περιβάλλοντα, όπου τα συστήματα καταγραφής κίνησης δεν μπορούν να χρησιμοποιηθούν, σχεδιάζουμε αλγόριθμους εκτίμησης κατάστασης οι οποίοι, με βάση την οπτική πληροφορία που λαμβάνεται είτε από μια απλή συμβατική

κάμερα είτε από έναν δυναμικό αισθητήρα όρασης (DVS), εκπληρώνουν τους ακόλουθους στόχους:

- Εύρωστη και σε πραγματικό χρόνο αναγνώριση του σχοινιού παρά τις αντίξοες εξωτερικές συνθήκες.
- Εκτίμηση της πλήρους κατάστασης του σχοινιού, δηλαδή η_L και ω_L , απαραίτητη για οποιονδήποτε ελεγκτή ανάδρασης.

Εκτίμηση της Κατάστασης του Σχοινιού

Σε αυτή την ενότητα, παρουσιάζουμε 2 μεθοδολογίες για την εκτίμηση της κατάστασης του σχοινιού κατά την εναέρια μεταφορά του φορτίου. Προκειμένου να αποφευχθεί η ενσωμάτωση πολλών αισθητήρων, π.χ. περιστροφικών κωδικοποιητών, που προαπαιτούν περίπλοκες μηχανικές παρεμβάσεις στο UAV, ή/και η ανάπτυξη κατάλληλων παρατηρητών [7], [45], [73], στοχεύουμε να αξιοποιήσουμε μια μόνο κάμερα με προσανατολισμό προς τα κάτω, και συγκεκριμένα είτε μια συμβατική κάμερα είτε έναν αισθητήρα DVS. Επομένως, δεδομένου ότι σχεδόν κάθε UAV διαθέτει μια βάση για κάμερα, οι προτεινόμενοι αλγόριθμοι μπορούν να υλοποιηθούν με ελάχιστες προϋποθέσεις.

Εκτίμηση της Κατάστασης του Σχοινιού με Συμβατική Κάμερα

Ανίχνευση του Σχοινιού με CNN

Θεωρούμε την εικόνα RGB που λαμβάνεται από την κάμερα που βρίσκεται κοντά στο κέντρο μάζας του οχήματος, όπως απεικονίζεται στην Εικόνα 3.2α. Στόχος μας είναι να επεξεργαστούμε την προαναφερθείσα εικόνα προκειμένου να αναγνωρίσουμε το σχοινί και, τελικά, να ανιχνεύσουμε το φορτίο στο επίπεδο της εικόνας. Διάφορες μέθοδοι υπάρχουν για ανίχνευση αντικειμένων, όπως η χρήση marker, το ιστόγραμμα προσανατολισμένων κλίσεων (HOG) κ.λπ. Ωστόσο, δεδομένου ότι το UAV με το αναρτημένο μέσω σχοινιού φορτίο πλοηγείται σε εξωτερικά περιβάλλοντα, όπου παρατηρούνται διάφορες συνθήκες φωτισμού και σκιές, επιλέγεται ένα νευρωνικό δίκτυο λόγω της ευρωστίας και της ευελιξίας του.

Πιο συγκεκριμένα, χρησιμοποιήθηκε το μοντέλο *mobilenet_segnet* από το Keras image segmentation framework [28] λόγω του χαμηλού υπολογιστικού του κόστους. Ειδικότερα, διαπιστώσαμε ότι ο μέσος χρόνος για την εξαγωγή συμπερασμάτων από το Συνελικτικό Νευρωνικό Δίκτυο (CNN) ήταν 33 ms στον ενσωματωμένο υπολογιστή του οχήματος, δηλαδή στο Jetson AGX Xavier [35], όταν χρησιμοποιείται η VGA ανάλυση για την κάμερα ZED 2 [98]. Όσον αφορά τη διαδικασία της εκπαίδευσης, συλλέχθηκε ένα σύνολο 1000 εικόνων πετώντας χειροκίνητα το όχημα με το αναρτημένο φορτίο σε διάφορα περιβάλλοντα. Στη συνέχεια, τα δεδομένα χωρίστηκαν σε ένα σετ εκπαίδευσης και ένα σετ επικύρωσης σύμφωνα με έναν διαχωρισμό 90:10. Το σύνολο των δεδομένων εκπαίδευσης επεξεργάστηκε χρησιμοποιώντας το λογισμικό *labelme* [93] και στη συνέχεια το μέγεθος του συνόλου αυξήθηκε στις 4000 εικόνες εφαρμόζοντας

πολλαπλούς μετασχηματισμούς στις εικόνες με το εργαλείο *imgaug* [37]. Τελικά, το CNN εκπαιδεύτηκε μέσω της μεθόδου back-propagation και συνέκλινε σε ακρίβεια άνω του 98% στο σύνολο επικύρωσης μετά από 15 εποχές. Ένα ενδεικτικό παράδειγμα της εξόδου του εκπαιδευμένου CNN απεικονίζεται στην Εικόνα 3.2b.

Με βάση την έξοδο του εκπαιδευμένου CNN, εξάγεται μια περιοχή ενδιαφέροντος με τη μορφή ενός convex hull, το άκρο (u_L, v_L) του οποίου υποδηλώνει τις συντεταγμένες pixel του φορτίου στο επίπεδο της εικόνας όπως απεικονίζεται στην Εικόνα 3.2c. Σύμφωνα με το τυπικό pinhole μοντέλο κάμερας, η θέση του φορτίου, δηλαδή η κάτω άκρη του σχοινιού, ${}^C \mathbf{p}_L \in \mathbb{R}^3$ σε σχέση με το σύστημα συντεταγμένων της κάμερας \mathbf{C} εξάγεται από τις συντεταγμένες pixel $u_L \in [-W/2, W/2]$, $v_L \in [-H/2, H/2]$, ως εξής:

$${}^C \mathbf{p}_L = \begin{bmatrix} c_{zL} u_L / f & c_{zL} v_L / f & c_{zL} \end{bmatrix}^T \quad (1)$$

όπου $W \times H$ είναι η ανάλυση της κάμερας, f είναι η εστιακή απόσταση και c_{zL} είναι η καρτεσιανή συντεταγμένη της άκρης του σχοινιού ως προς τον άξονα z του \mathbf{C} . Εφόσον η απόσταση c_{zL} είναι άγνωστη, αξιοποιείται το σταθερό μήκος l του σχοινιού ώστε να ανακτηθεί η τρισδιάστατη θέση του φορτίου. Συγκεκριμένα, δεδομένου ότι το σχοινί θεωρείται τεντωμένο και προσαρτημένο στην αρχή του συστήματος συντεταγμένων του οχήματος \mathbf{B} , προκύπτει η ακόλουθη εξίσωση:

$$\|{}^B \mathbf{p}_L\| = \|\mathbf{R}_{BC} {}^C \mathbf{p}_L + \mathbf{t}_{BC}\| = l \quad (2)$$

όπου ${}^B \mathbf{p}_L$ είναι η θέση της άκρης του σχοινιού ως προς το \mathbf{B} και \mathbf{R}_{BC} και \mathbf{t}_{BC} είναι ο γνωστός πίνακας περιστροφής και το διάνυσμα μετατόπισης από το \mathbf{C} στο \mathbf{B} . Η επίλυση της Εξίσωσης 3.2 έχει ως αποτέλεσμα μια συνάρτηση $h : \mathbb{R}^2 \mapsto \mathbb{R}$ που αντιστοιχίζει τις συντεταγμένες (u_L, v_L) – και γνωστές παραμέτρους, δηλαδή, τα στοιχεία των \mathbf{R}_{BC} και \mathbf{t}_{BC} , τις εσωτερικές παραμέτρους της κάμερας και το μήκος l του σχοινιού – με την απόσταση c_{zL} , δηλαδή, $c_{zL} = h(u_L, v_L)$. Η αναλυτική μορφή της συνάρτησης h παραλείπεται καθώς βασίζεται στην εκάστοτε σχετική στάση μεταξύ της κάμερας και του συστήματος συντεταγμένων του οχήματος. Τελικά, υπολογίζεται το μοναδιαίο διάνυσμα του σχοινιού ${}^B \mathbf{n}$:

$${}^B \mathbf{n} = \frac{{}^B \mathbf{p}_L}{\|{}^B \mathbf{p}_L\|} = \frac{\mathbf{R}_{BC} {}^C \mathbf{p}_L + \mathbf{t}_{BC}}{l} \quad (3)$$

και στη συνέχεια μετασχηματίζεται στο σύστημα αναφοράς \mathbf{W} ($\mathbf{n} = \mathbf{R}_{WB} {}^B \mathbf{n}$).

Kalman Φίλτρο

Εκτός από το μοναδιαίο διάνυσμα \mathbf{n} , μια εκτίμηση της ταχύτητας του σχοινιού είναι επίσης απαραίτητη για τα επόμενα σχήματα ελέγχου. Αν και η αριθμητική παραγωγή Euler της μέτρησης \mathbf{n} είναι μια πιθανή λύση, συνήθως οδηγεί σε μια πολύ θορυβώδη εκτίμηση και, επομένως, σε υποβαθμισμένη απόδοση του σχήματος ελέγχου.

Κατά συνέπεια, επιλέγεται ένα φίλτρο Kalman, καθώς είναι ένα ευρέως διαδεδομένο φίλτρο για εκτίμηση κατάστασης [15], [49]. Το φίλτρο Kalman υλοποιείται με βάση το

δυναμικό μοντέλο της Εξίσωσης 2.7. Πιο συγκεκριμένα, ορίζοντας $\mathbf{x}_L = [\mathbf{n}^T \quad \dot{\mathbf{n}}^T]^T \in \mathbb{R}^6$ ως το διάνυσμα κατάστασης, προκύπτουν τα ακόλουθα μοντέλα συστήματος και μέτρησης:

$$\dot{\mathbf{x}}_L = \begin{bmatrix} \dot{\mathbf{n}} \\ \ddot{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{n}} \\ \frac{1}{l} \left(-g\mathbf{e}_z - \frac{\|\mathbf{T}\|}{m_L} \mathbf{n} - \dot{\mathbf{v}} \right) \end{bmatrix} + \mathbf{w} \quad (4)$$

$$\mathbf{y}_L = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{n} \\ \dot{\mathbf{n}} \end{bmatrix} + \boldsymbol{\omega} \quad (5)$$

όπου $\mathbf{w} \in \mathbb{R}^6$ και $\boldsymbol{\omega} \in \mathbb{R}^3$ αντιπροσωπεύουν αθροιστικούς γκαουσιανούς θορύβους. Ένα Extended ή Unscented Kalman Filter (EKF, UKF) χρησιμοποιείται λόγω του μη γραμμικού μοντέλου του συστήματος. Σημειώνεται ότι η επιτάχυνση του UAV $\dot{\mathbf{v}}$ και η τάση του σχοινιού $\|\mathbf{T}\|$ μετρώνται απευθείας από το IMU του οχήματος και μια δυναμοκυψέλη, τοποθετημένη πάνω στο σχοινί, αντίστοιχα. Μόλις εκτιμηθεί το μοναδιαίο διάνυσμα του σχοινιού \mathbf{n} και η ταχύτητά του $\dot{\mathbf{n}}$, υπολογίζονται οι γωνίες η_L και η γωνιακή ταχύτητα ω_L σύμφωνα με τις Εξισώσεις 2.2 και 2.3.

Εκτίμηση της Κατάστασης του Σχοινιού με DVS

Προκειμένου να επιτευχθεί η εκτίμηση της κατάστασης του σχοινιού σε πολύ μικρότερο χρονικό διάστημα σε σύγκριση με την παραπάνω μεθοδολογία με συμβατική κάμερα, διερευνούμε την ενσωμάτωση ενός αισθητήρα DVS στο όχημα, δηλαδή μιας event κάμερας, όπως απεικονίζεται στην Εικόνα 3.3. Οι DVS παρουσιάζουν πολλά πλεονεκτήματα σε σχέση με τις τυπικές κάμερες, όπως χαμηλό latency, χαμηλή κατανάλωση ενέργειας, υψηλό δυναμικό εύρος και ευρωστία ως προς motion blur [20]. Διάφορες δουλειές στη βιβλιογραφία έχουν επιδείξει με επιτυχία την χρήση DVS σε εξωτερικά περιβάλλοντα για ένα ευρύ φάσμα εφαρμογών [12], [90], [13], [74], [83], [53]. Ωστόσο, λόγω της ασύγχρονης φύσης αυτών των καμερών και του καινοτόμου τρόπου απόκτησης οπτικών δεδομένων, κάθε δουλειά είναι μοναδική και εξαρτάται σε μεγάλο βαθμό από το αντικείμενο που πρέπει να ταυτοποιηθεί και την εφαρμογή που αντιμετωπίζεται. Ως εκ τούτου, σχεδιάζεται μια νέα λύση προσαρμοσμένη στο συγκεκριμένο πρόβλημα της αναγνώρισης του σχοινιού και της εκτίμησης της κατάστασής του κατά την εναέρια μεταφορά φορτίων.

Αναπαράσταση των Γεγονότων

Σε αντίθεση με τις συμβατικές κάμερες, η event κάμερα είναι ένας αισθητήρας εμπνευσμένος από την βιολογία που ανταποκρίνεται στις αλλαγές φωτεινότητας ασύγχρονα και ανεξάρτητα για κάθε pixel. Ως εκ τούτου, η έξοδος μιας κάμερας DVS είναι μια ασύγχρονη ροή γεγονότων (events) που προκαλούνται από αλλαγές φωτεινότητας στο περιβάλλον. Πιο συγκεκριμένα, ένα event $e_k = (u_k, v_k, t_k, p_k)$ παρατηρείται στις συνεταγμένες pixel (u_k, v_k) και τη στιγμή t_k μόλις ανιχνευθεί μια αλλαγή στη φωτεινότητα πάνω από ένα προκαθορισμένο όριο σε σχέση με το τελευταίο event στο ίδιο pixel. Η πολικότητα (polarity) $p_k \in \{0, 1\}$ υποδεικνύει την κατεύθυνση της αλλαγής της φωτεινότητας.

Σύμφωνα με τον προτεινόμενο αλγόριθμο, τα events που παράγονται από την κάμερα DVS υποβάλλονται σε προεπεξεργασία προκειμένου να εντοπιστούν αυτά που αντιστοιχούν στο σχοινί και να εξαχθούν αργότερα σημαντικές πληροφορίες για την κατάστασή του. Πιο συγκεκριμένα, μόλις εμφανιστεί ένα event e_i , αποθηκεύεται σε ένα νέφος σημείων (point cloud) $\mathcal{P} = \{e_i | t_i \in [t_0 - T, t_0]\}$, που αποτελείται από τα πιο πρόσφατα events μέχρι την τρέχουσα χρονική στιγμή t_0 . Σημειώνεται ότι τόσο η χωροχρονική πληροφορία (u, v, t) όσο και η πολικότητα p αποθηκεύονται στο point cloud. Ενδεικτικά παραδείγματα «event» point clouds, μαζί με τις αντίστοιχες grayscale εικόνες που καταγράφονται από την DVS, απεικονίζονται στην Εικόνα 3.4, όπου παρατηρείται ένας αυξημένος αριθμός events, που αντιστοιχεί στο background ή σε θόρυβο, σε εξωτερικούς χώρους. Η Βιβλιοθήκη Point Cloud Library (PCL) [75], η οποία διακρίνεται για την υπολογιστική της αποτελεσματικότητα, αξιοποιείται για την αναπαράσταση του event point cloud και για τους επακόλουθους αλγορίθμους επεξεργασίας. Όσο για την αναπαράσταση, επιλέγεται από την PCL ένα point cloud με σημεία τύπου XYZI, όπου οι συντεταγμένες X-Y χρησιμοποιούνται για τις τιμές των pixel, η συντεταγμένη Z για το χρόνο και η τιμή I για την πολικότητα των events.

Επεξεργασία του Event Point Cloud

Το προαναφερθέν point cloud \mathcal{P} υποβάλλεται σε downsampling με την εφαρμογή ενός 3D voxel φίλτρου εντός μιας χωροχρονικής γειτονιάς κάθε event, που ορίζεται από τον όγκο $n_u[\text{pixels}] \times n_v[\text{pixels}] \times n_t[\text{ms}]$. Ως εκ τούτου, τα events που ανήκουν σε κάθε voxel προσεγγίζονται από το κέντρο τους. Επισημαίνεται ότι, εκτός από τις συντεταγμένες pixel και τις χρονικές στιγμές, υπολογίζεται επίσης ο μέσος όρος της πολικότητας των events σε κάθε voxel. Συνεπώς, το νέο point cloud που δημιουργείται $\mathcal{P}' = \{e'_j | t'_j \in [t_0 - T, t_0]\}$ αποτελείται από κέντρα e'_j με πραγματικές πολικότητες $p'_j \in [0, 1]$, όπως απεικονίζεται στην Εικόνα 3.5a.

Στη συνέχεια, το point cloud \mathcal{P}' φιλτράρεται με βάση τη μέση πολικότητα p'_j κάθε κέντρου e'_j . Πιο συγκεκριμένα, μόνο τα κέντρα με πολικότητα $p'_j \in (0, 1)$, δηλαδή $p'_j \neq 0$ και $p'_j \neq 1$, θεωρούνται ως υποψήφια events για να ανήκουν στο σχοινί, όπως απεικονίζεται στην Εικόνα 3.5b. Το φίλτρο αυτό ουσιαστικά υποδηλώνει ότι θα πρέπει να υπάρχουν τουλάχιστον 2 events e_i σε κάθε voxel του point cloud \mathcal{P} με αντίθετες πολικότητες προκειμένου να προωθηθεί το αντίστοιχο κέντρο e'_j σε υποψήφιο event. Αυτή η προσέγγιση προέρχεται από την παρατήρηση ότι, είτε λόγω της κίνησης του οχήματος είτε του ίδιου του σχοινού, δημιουργείται ένας επαρκής αριθμός events με αντίθετες πολικότητες σε κλίμακα ms κατά τη μεταφορά του αναρτημένου φορτίου. Επιπλέον, ακόμη κι όταν το UAV σταθεροποιείται σε μια συγκεκριμένη θέση, τόσο το UAV όσο και το σχοινί δεν μπορούν να είναι απολύτως σταθερά λόγω της ύπαρξης εξωτερικών διαταραχών, που προκύπτουν κυρίως από τον άνεμο. Ένα άλλο πλεονέκτημα αυτού του φίλτρου είναι ότι η πλειονότητα των θορυβωδών και ψευδών events αγνοούνται. Αναλυτικότερα, μετά την εμφάνιση ενός event, παρατηρούνται μερικές φορές πρόσθετα events ίδιας πολικότητας και στην ίδια θέση pixel μέσα σε ένα συγκεκριμένο χρονικό διάστημα

[12]. Κατά συνέπεια, με την εφαρμογή του προαναφερθέντος φιλτραρίσματος, αυτά τα events ίδιας πολικότητας δεν προωθούνται.

Ωστόσο, υπάρχουν μερικά κέντρα e'_j που περνούν το προαναφερθέν φίλτρο αλλά αντιστοιχούν είτε σε θόρυβο είτε σε αντικείμενα που βρίσκονται στο έδαφος/background και όχι στο σχοινί. Έτσι, εκτελείται ένας αλγόριθμος Euclidean Clustering χρησιμοποιώντας την αποτελεσματική αναπαράσταση Kd-tree για το εισερχόμενο point cloud $\mathcal{P}'' = \{e'_j | t'_j \in [t_0 - T, t_0], p'_j \in (0, 1)\}$ και στη συνέχεια τον αλγόριθμο nearest neighbor search. Το cluster με το μεγαλύτερο μέγεθος προωθείται ως το cluster του σχοινοῦ \mathcal{C} καθώς το σχοινί είναι πιο κοντά στη DVS από οποιοδήποτε άλλο αντικείμενο στο έδαφος και θεωρείται ότι έχει επαρκή διάμετρο. Συνεπώς, τα events που αντιπροσωπεύουν το σχοινί συνήθως καταλαμβάνουν το μεγαλύτερο μέρος του επιπέδου της εικόνας, όπως απεικονίζεται στην Εικόνα 3.5c. Περαιτέρω έλεγχοι πραγματοποιούνται επίσης μεταξύ των διαδοχικών clusters σχοινοῦ, συγκρίνοντας τη σχετική τους θέση και, επομένως, διασφαλίζοντας τη σωστή επιλογή cluster όταν αντικείμενα παρόμοιου σχήματος βρίσκονται στο έδαφος. Τελικά, μια ακολουθία clusters σχοινοῦ εξάγεται με σχεδόν συνεχή τρόπο, όπως φαίνεται στην Εικόνα 3.6.

Μόλις εξαχθεί το cluster του σχοινοῦ \mathcal{C} , είναι απαραίτητο να βρεθεί το event που αντιστοιχεί στο φορτίο, δηλαδή το κάτω άκρο του σχοινοῦ. Με βάση τον προσανατολισμό και τη σχετική θέση της DVS σε σχέση με το σχοινί, μια απλή προσέγγιση θα ήταν η επιλογή του event $e'_j \in \mathcal{C}$ με τη μεγαλύτερη συντεταγμένη v'_j . Ωστόσο, αυτή η προσέγγιση υποβαθμίζει τη σημασία των υπόλοιπων events. Κατά συνέπεια, προσαρμόζεται μια 2D ευθεία λαμβάνοντας υπόψη τις συντεταγμένες pixel (u'_j, v'_j) των events $e'_j \in \mathcal{C}$, όπως φαίνεται στην Εικόνα 3.5c.

Προσέγγιση από Καμπύλη Bézier

Εφαρμόζοντας τα προαναφερθέντα βήματα επεξεργασίας, λαμβάνονται τελικά οι συντεταγμένες pixel (u_L, v_L) του κάτω άκρου του σχοινοῦ. Ωστόσο, το κάτω μέρος του σχοινοῦ ενδέχεται να μην δημιουργήσει επαρκή αριθμό events κατά τη διάρκεια ορισμένων λιγοστών επαναλήψεων, γεγονός που οδηγεί σε μια θορυβώδη εκτίμηση της κατάστασης του σχοινοῦ και υποβαθμίζει την απόδοση του σχήματος ελέγχου. Συνεπώς, αξιοποιείται μία καμπύλη Bézier για να προσεγγίσει ομαλά τις συντεταγμένες (u_L, v_L) αλλά και τις ταχύτητες (\dot{u}_L, \dot{v}_L) , που απαιτούνται για την εκτίμηση της πλήρους κατάστασης του σχοινοῦ.

Πιο συγκεκριμένα, μια καμπύλη Bézier ορίζεται, με βάση το βαθμό n της καμπύλης, ως εξής:

$$\mathbf{B}(\tau) = \sum_{i=0}^n b_{i,n}(\tau) \mathbf{c}_i \quad (6)$$

όπου $\tau \in [0, 1]$, $\mathbf{B}(\tau) \in \mathbb{R}^2$ (για το u - v επίπεδο της εικόνας) είναι η καμπύλη Bézier, $b_{i,n}(\tau), i = 0, \dots, n$ είναι τα πολυώνυμα βάσης Bernstein βαθμού n και $\mathbf{c}_i = [c_{u,i} \quad c_{v,i}]^T \in \mathbb{R}^2$ είναι τα $n + 1$ σημεία ελέγχου.

Για να υπολογιστεί το κατάλληλο σύνολο σημείων ελέγχου $\mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n]$ και, επομένως, να προσεγγιστεί η τροχιά του σχοινιού στο επίπεδο της εικόνας, ένα σετ δεδομένων $D = \{(t_{N-1}, u_{L,N-1}, v_{L,N-1}), \dots, (t_0, u_{L,0}, v_{L,0})\}$ δημιουργείται, το οποίο αποτελείται από τις τελευταίες N μετρήσεις, μέχρι την τρέχουσα χρονική στιγμή t_0 , των συντεταγμένων pixel (u_L, v_L) μαζί με τις αντίστοιχες χρονικές στιγμές, όπως προκύπτουν από την επεξεργασία του event point cloud. Το σετ δεδομένων έχει σταθερό μέγεθος και επικαιροποιείται μόλις είναι διαθέσιμη μια νέα μέτρηση.

Αφού αντιστοιχηθεί η παράμετρος $\tau \in [0, 1]$ με το χρονικό διάστημα $t \in [t_{N-1}, t_0]$, υπολογίζεται το βέλτιστο σύνολο σημείων ελέγχου \mathbf{c} από τη λύση του προβλήματος βελτιστοποίησης:

$$\min_{\mathbf{c}} \sum_{i=0}^{N-1} \left(w_{t_i} \left\| \mathbf{B}(\tau_i) - [u_{L,i}, v_{L,i}]^T \right\|_2^2 \right) \quad (7)$$

που ουσιαστικά ελαχιστοποιεί την απόσταση μεταξύ της καμπύλης Bézier και των μετρήσεων, ενώ τα βάρη w_{t_i} ενισχύουν τις πιο πρόσφατες μετρήσεις και επιβάλλουν κυρώσεις στις παλαιότερες [29].

Επιπλέον, περιορισμοί, που σχετίζονται με την ελάχιστη και μέγιστη ταχύτητα και επιτάχυνση του σχοινιού στο επίπεδο της εικόνας, ενσωματώνονται στο πρόβλημα βελτιστοποίησης, έτσι ώστε να βρεθούν αποκλειστικά εφικτές λύσεις. Πιο συγκεκριμένα, παραγωγίζοντας την καμπύλη Bézier, διατυπώνονται οι εξής περιορισμοί για τον άξονα u (ομοίως για τον v):

$$\begin{aligned} n \cdot (c_{u,i} - c_{u,i-1}) / (t_0 - t_{N-1}) &\in [-\dot{u}_{max}, \dot{u}_{max}] \\ \frac{n \cdot (n-1) \cdot (c_{u,i} - 2c_{u,i-1} + c_{u,i-2})}{(t_0 - t_{N-1})^2} &\in [-\ddot{u}_{max}, \ddot{u}_{max}] \end{aligned} \quad (8)$$

Το προαναφερθέν πρόβλημα Τετραγωνικού Προγραμματισμού με περιορισμούς (Εξίσωση 3.7) που χαρακτηρίζεται από $2 \cdot (n+1)$ μεταβλητές, δηλαδή τα σημεία ελέγχου, και $2 \cdot n + 2 \cdot (n-1)$ περιορισμούς (Εξίσωση 3.8), επιλύεται με τη βοήθεια του λογισμικού OOQP [24] και, τελικά, μια εκτίμηση των συντεταγμένων pixel (u_L, v_L) και των ταχυτήτων (\dot{u}_L, \dot{v}_L) σε κάθε τρέχουσα χρονική στιγμή t_0 λαμβάνεται μέσω της καμπύλης Bézier, η οποία απεικονίζεται στην Εικόνα 3.5d.

Στη συνέχεια, δεδομένης της καμπύλης Bézier, εκτιμάται η συνολική κατάσταση του σχοινιού. Πιο συγκεκριμένα, παρόμοια με τη μέθοδο που βασίζεται στη συμβατική κάμερα, το μοναδιαίο διάνυσμα ${}^B \mathbf{n}$ υπολογίζεται σύμφωνα με τις Εξισώσεις 3.2 και 3.3. Όσον αφορά την ταχύτητα του σχοινιού, αυτή προκύπτει παραγωγίζοντας το Εχυσιαστικό (3.3) σύμφωνα με τις ακόλουθες σχέσεις:

$${}^B \dot{\mathbf{n}} = \frac{\mathbf{R}_{BC}}{l} C \dot{\mathbf{p}}_L = \frac{\mathbf{R}_{BC}}{l} \frac{1}{f} \begin{bmatrix} {}^C \dot{z}_L u_L + {}^C z_L \dot{u}_L \\ {}^C \dot{z}_L v_L + {}^C z_L \dot{v}_L \\ f {}^C \dot{z}_L \end{bmatrix} \quad (9)$$

$${}^B \dot{\mathbf{n}} = \frac{\mathbf{R}_{BC}}{l \cdot f} \begin{bmatrix} \left(\frac{\partial h}{\partial u_L} \dot{u}_L + \frac{\partial h}{\partial v_L} \dot{v}_L \right) u_L + {}^C z_L \dot{u}_L \\ \left(\frac{\partial h}{\partial u_L} \dot{u}_L + \frac{\partial h}{\partial v_L} \dot{v}_L \right) v_L + {}^C z_L \dot{v}_L \\ f \left(\frac{\partial h}{\partial u_L} \dot{u}_L + \frac{\partial h}{\partial v_L} \dot{v}_L \right) \end{bmatrix} \quad (10)$$

με $\frac{\partial h}{\partial u_L}$, $\frac{\partial h}{\partial v_L}$ να δηλώνουν τις μερικές παραγώγους της συνάρτησης ${}^C z_L = h(u_L, v_L)$. Συνεπώς, με βάση την έξοδο της καμπύλης Bézier, δηλαδή τις συντεταγμένες (u_L, v_L) και τις ταχύτητες (\dot{u}_L, \dot{v}_L) , τόσο το μοναδιαίο διάνυσμα του σχοινοῦ ${}^B \mathbf{n}$ όσο και η ταχύτητά του ${}^B \dot{\mathbf{n}}$ μπορούν να υπολογιστούν και στη συνέχεια να μετασχηματιστούν στο σύστημα αναφοράς \mathbf{W} έτσι ώστε να υπολογιστούν οι γωνίες $\boldsymbol{\eta}_L$ και η γωνιακή ταχύτητα ω_L (Εξισώσεις 2.2 και 2.3).

Ακρίβεια της Μεθόδου

Η ακρίβεια της προτεινόμενης μεθόδου αξιολογήθηκε αρχικά μέσω πειραμάτων σε εσωτερικούς χώρους, όπου η έξοδος της προαναφερθείσας λύσης που βασίζεται στα events συγκρίθηκε με ακριβείς μετρήσεις που παρέχονται από ένα σύστημα καταγραφής κίνησης. Πιο συγκεκριμένα, η DVS και το άνω άκρο του σχοινοῦ στερεώθηκαν στο ταβάνι ενός δωματίου με σχετική στάση παρόμοια με αυτή που έχουν πάνω στο UAV. Σημειώνεται ότι σε αυτή την περίπτωση το σύστημα αναφοράς συμπίπτει με το \mathbf{B} . Το ground truth λήφθηκε από το σύστημα καταγραφής κίνησης Vicon το οποίο βασίζεται στη χρήση markers. Για να εξασφαλιστεί η συνεχής παρακολούθηση της ταλαντωτικής κίνησης του σχοινοῦ, markers τοποθετήθηκαν προσεκτικά γύρω από το σχοινί. Μια εικόνα του εσωτερικού χώρου, από τη συμβατική κάμερα της DAVIS 346, και το αντίστοιχο event point cloud απεικονίζονται στις Εικόνες 3.4a και 3.4b αντίστοιχα.

Κατά τη διάρκεια των πειραμάτων, συγκεντρώθηκαν δεδομένα τόσο από τη DVS όσο και από το σύστημα Vicon μεταβάλλοντας την αρχική γωνία του φορτίου και απελευθερώνοντάς το με μηδενική ταχύτητα. Ως εκ τούτου, το φορτίο κινήθηκε ελεύθερα υπό την επίδραση της βαρύτητας. Ενδεικτικά αποτελέσματα ενός πειράματος παρουσιάζονται στην Εικόνα 3.7 όπου οι γωνίες του σχοινοῦ $\boldsymbol{\eta}_L = [\phi_L \ \theta_L]^T$ που υπολογίζονται από την επεξεργασία της ροής events συγκρίνονται με το ground truth που καταγράφεται από το σύστημα Vicon. Είναι προφανές ότι η μέθοδός μας διακρίνει με επιτυχία το σχοινί και καταγράφει όλο το εύρος της ταλαντωτικής του κίνησης. Τα σφάλματα RMSE για τις γωνίες ϕ_L και θ_L ήταν 2.204° και 2.639° αντίστοιχα κατά τις δοκιμές στον εσωτερικό χώρο. Τέλος, το ποσοστό επιτυχούς ανίχνευσης, δηλαδή ο αριθμός των ανιχνεύσεων προς τον συνολικό αριθμό επαναλήψεων, ήταν ίσος με 96.67%.

Ωστόσο, θα πρέπει να τονιστεί ότι αν και η μέθοδός μας συλλαμβάνει τη συχνότητα της ταλαντωτικής κίνησης του σχοινοῦ, το μέτρο των γωνιών του σχοινοῦ υπερεκτιμάται, ειδικά όταν οι γωνίες απομακρύνονται σημαντικά από τις μηδέν μοίρες. Αυτή η συμπεριφορά οφείλεται κυρίως στο γεγονός ότι το χαμηλότερο τμήμα του σχοινοῦ ενδέχεται να μην δημιουργήσει επαρκή αριθμό events και, ως εκ τούτου, η μέθοδός μας μπορεί να ταυτοποιήσει ως την άκρη του σχοινοῦ (u_L, v_L) ένα κέντρο το οποίο στην πραγματικότητα δεν αντιστοιχεί στην άκρη αλλά σε ένα σημείο που βρίσκεται λίγα

εκατοστά πιο πάνω. Παρόλα αυτά, σε κάθε περίπτωση, τα σφάλματα RMSE για τις γωνίες είναι σχετικά μικρά ενώ επίσης απώτερος στόχος της δουλειάς μας είναι η εναέρια μεταφορά με ελάχιστες γωνίες του σχοινοίου, όπου μετριάζεται η προαναφερθείσα συμπεριφορά.

Υπολογιστικό Κόστος

Προκειμένου να ποσοτικοποιήσουμε τη μείωση του υπολογιστικού κόστους που επιτυγχάνεται με αυτήν τη μέθοδο σε σύγκριση με τη λύση που βασίζεται στη συμβατική κάμερα, μετράται στο Jetson AGX Xavier ο χρόνος εκτέλεσης των επιμέρους βημάτων του προτεινόμενου αλγορίθμου τόσο σε εσωτερικά όσο και σε εξωτερικά περιβάλλοντα. Ο μέσος χρόνος επεξεργασίας για καθένα από τα βήματα που απαρτίζουν τη συνολική μέθοδο, δηλαδή η επεξεργασία του event point cloud και η προσέγγιση με μια καμπύλη Bézier, καθώς και οι παράμετροι του προτεινόμενου αλγορίθμου συνοψίζονται στους Πίνακες 3.1 και 3.2 αντίστοιχα. Τονίζεται ότι το υπολογιστικό κόστος της επεξεργασίας του point cloud σχετίζεται άμεσα με τον αριθμό των εισερχόμενων events. Κατά συνέπεια, κατά τη διάρκεια των πειραμάτων σε εσωτερικό χώρο, όπου η DVS δεν κινείται και το περιβάλλον είναι στατικό, ο χρόνος εκτέλεσης αυτού του βήματος μειώνεται. Σε κάθε περίπτωση, ο μέσος χρόνος επεξεργασίας της προτεινόμενης μεθόδου είναι σημαντικά χαμηλότερος σε σύγκριση με έναν τυπικό αλγόριθμο ανίχνευσης με συμβατική κάμερα και χρήση CNN, όπου ο μέσος χρόνος για την εξαγωγή συμπερασμάτων από το CNN είναι ίσος με 33 ms.

Έλεγχος με Βαθιά Ενισχυτική Μάθηση

Σε αυτή την ενότητα, παρουσιάζουμε μια στρατηγική βασισμένη στη βαθιά Ενισχυτική Μάθηση (deep RL) για τον έλεγχο του πολυκοπτέρου με το αναρτημένο μέσω σχοινοίου φορτίο. Συγκεκριμένα, εκπαιδεύεται μια πολιτική, που αντιπροσωπεύεται από ένα βαθύ νευρωνικό δίκτυο (NN), χωρίς να βασίζεται στο μοντέλο του συστήματος με απώτερο σκοπό να παράγει κατάλληλες συνεχείς εισόδους που πλοηγούν το όχημα προς σημεία αναφοράς ενώ, ταυτόχρονα, αντισταθμίζουν τις ταλαντώσεις του σχοινοίου.

Οι σύγχρονοι αλγόριθμοι βαθιάς Ενισχυτικής Μάθησης, που δεν βασίζονται στο μοντέλο του συστήματος, δεν απαιτούν την αναγνώριση της δυναμικής του συστήματος και έχουν αποδειχθεί αποτελεσματικά εργαλεία για την εκμάθηση πολιτικών ικανών να αντιμετωπίζουν δύσκολες εφαρμογές και να ελέγχουν πράκτορες (agents) με πολύπλοκη δυναμική, όπως τα UAVs [34], [42]. Ωστόσο, η ιδέα της αξιοποίησης deep RL για την εναέρια μεταφορά αναρτημένων φορτίων είναι πρωτότυπη.

Υπόβαθρο Ενισχυτικής Μάθησης

Σύμφωνα με την κλασική Θεωρία της Ενισχυτικής Μάθησης, το περιβάλλον ενός πράκτορα ορίζεται ως ένα Markov Decision Process (MDP). Σε κάθε χρονικό βήμα t , ένα MDP περιγράφεται από την πλειάδα $(s_t, a_t, p(s_{t+1}|s_t, a_t), r_t)$, όπου s_t είναι η τρέχουσα

κατάσταση του πράκτορα, a_t είναι η επιλεγμένη ενέργεια (action), $p(s_{t+1}|s_t, a_t)$ είναι η πιθανότητα ή η συνάρτηση μετάβασης, ανάλογα με το εάν λαμβάνεται υπόψη στοχαστική ή ντετερμινιστική δυναμική του συστήματος, από την τρέχουσα κατάσταση s_t στην επόμενη s_{t+1} όταν εφαρμόζεται η ενέργεια a_t και $r_t = r_t(s_t, s_{t+1}, a_t)$ είναι η ανταμοιβή (reward) που ποσοτικοποιεί την απόδοση του πράκτορα.

Υποθέτοντας ότι ο πράκτορας λειτουργεί σε επεισόδια, ο στόχος του προβλήματος RL είναι να βρει μια βέλτιστη πολιτική που μεγιστοποιεί την αθροιστική ανταμοιβή ή απόδοση (return) στο επεισόδιο. Πιο συγκεκριμένα, ένα επεισόδιο τερματίζεται μετά από T αλληλεπιδράσεις του πράκτορα με το περιβάλλον, $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$, ή όταν ικανοποιηθεί ένα κριτήριο τερματισμού. Η απόδοση υπολογίζεται ως το άθροισμα των ανταμοιβών που λαμβάνονται κατά τη διάρκεια ενός επεισοδίου, $R(\tau) = \sum_{t=0}^{T-1} \gamma^t r_t(s_t, s_{t+1}, a_t)$, όπου $\gamma \in (0, 1)$ είναι ένας παράγοντας που ευνοεί τις τρέχουσες ανταμοιβές έναντι των μελλοντικών. Στη Βαθιά Ενισχυτική Μάθηση, ο στόχος είναι να βρεθεί μία πολιτική υπό τη μορφή ενός νευρωνικού δικτύου π_θ , παραμετροποιημένη από κάποια βάρη θ , η οποία δεδομένης της τρέχουσας κατάστασης s_t του πράκτορα παράγει ενέργειες $a_t = \pi_\theta(s_t)$, στην ντετερμινιστική περίπτωση, που μεγιστοποιούν την αναμενόμενη απόδοση $\mathbb{E}_{\tau \sim \pi} [R(\tau)]$.

Σύμφωνα με την υπάρχουσα βιβλιογραφία [6], οι αλγόριθμοι RL μπορούν να ταξινομηθούν σε τρεις κύριες κατηγορίες: βελτιστοποίηση πολιτικής (policy optimization), εκμάθηση value function και actor-critic μεθοδολογίες. Συνοπτικά, οι μέθοδοι policy optimization βελτιώνουν άμεσα την παραμετροποιημένη πολιτική π_θ εφαρμόζοντας gradient ascent στην αναμενόμενη απόδοση. Αντίθετα, οι μέθοδοι value function βασίζονται στην προσέγγιση της βέλτιστης συνάρτησης Q , που ορίζεται ως $Q^*(s, a) = \max_{\pi} Q(s, a) = \max_{\pi} \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s, a_0 = a]$, με ένα βαθύ νευρωνικό δίκτυο $Q_\phi(s, a)$ και, στη συνέχεια, επιλέγουν ενέργειες ως εξής: $a(s) = \underset{a}{argmax} Q_\phi(s, a)$.

Οι μέθοδοι actor-critic συνδυάζουν τα πλεονεκτήματα των μεθόδων policy optimization και value function learning. Το δίκτυο actor είναι η πολιτική π_θ ενώ το critic αντιπροσωπεύει τη συνάρτηση Q , $Q_\phi(s, a)$. Οι μέθοδοι actor-critic παράγουν συνεχώς ενέργειες, σε αντίθεση με άλλες μεθόδους που παράγουν διακριτές, και διακρίνονται επίσης για την αποτελεσματικότητά τους ως προς τον απαιτούμενο αριθμό δεδομένων. Κατά συνέπεια, επιλέγεται ο αλγόριθμος Twin Delayed Deep Deterministic Policy Gradient (TD3), μια μέθοδος actor-critic που προτείνεται από το [19], για την εκπαίδευση του UAV με το αναρτημένο φορτίο.

Δομή Δικτύων

Ακολουθώντας την τυπική αρχιτεκτονική των μεθόδων actor-critic, εξετάζονται δύο νευρωνικά δίκτυα, ιδιαίτερα το δίκτυο της πολιτικής $\pi_\theta(s)$ και το δίκτυο της συνάρτησης Q , $Q_\phi(s, a)$. Το επίπεδο της εισόδου (input layer) για την πολιτική είναι η κατάσταση του συστήματος, η οποία ειδικά για την περίπτωση του octorotor με το αναρτημένο μέσω σχοινιού φορτίο αποτελείται από το σφάλμα στη θέση $\mathbf{p} - \mathbf{p}_{ref}$, την ταχύτητα \mathbf{v} , τον

προσανατολισμό του οχήματος $\boldsymbol{\eta}$, δηλαδή, τις γωνίες roll ϕ και pitch θ , τις αντίστοιχες γωνιακές ταχύτητες, τις συνιστώσες x, y του μοναδιαίου διανύσματος του σχοινιού \mathbf{n} , n_x και n_y , και τις αντίστοιχες συνιστώσες της ταχύτητάς του $\dot{\mathbf{n}}$. Σημειώνεται ότι λαμβάνεται υπόψη μόνο το σφάλμα θέσης, αντί τόσο της τρέχουσας θέσης του οχήματος όσο και της επιθυμητής, ώστε να μειωθεί η διάσταση του διανύσματος κατάστασης. Επιπλέον, ο χώρος κατάστασης προσανυξάνεται με την τελευταία ενέργεια a_{t-1} που έλαβε ο πράκτορας, προκειμένου να επιβληθούν κυρώσεις στις απότομες εναλλαγές ενεργειών στην συνάρτηση ανταμοιβής [60]. Το διάνυσμα ενεργειών (action vector), δηλαδή η έξοδος του δικτύου της πολιτικής, αποτελείται από τις εντολές $\phi_d, \theta_d, v_{z_d}$ για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου. Τονίζεται ότι η γωνία yaw διατηρείται σταθερή και ίση με μηδέν. Συνοπτικά, τα διανύσματα κατάστασης και ενεργειών ορίζονται ως:

$$s = [\mathbf{p} - \mathbf{p}_{ref} \quad \mathbf{v} \quad \boldsymbol{\eta} \quad \dot{\boldsymbol{\eta}} \quad \mathbf{n} \quad \dot{\mathbf{n}} \quad a_{t-1}]^T \quad (11)$$

$$a = [\phi_d \quad \theta_d \quad v_{z_d}]^T \quad (12)$$

Το δίκτυο της πολιτικής είναι ένα πλήρως συνδεδεμένο, πολυεπίπεδο και βαθύ νευρωνικό δίκτυο, όπως φαίνεται στην Εικόνα 3.8a, με 2 κρυφά επίπεδα 64 κόμβων και συνάρτηση ενεργοποίησης tanh. Η παρουσία της συνάρτησης tanh περιορίζει κάθε έξοδο του δικτύου της πολιτικής στο εύρος $[-1, 1]$ και, ως εκ τούτου, οι ενέργειες πολλαπλασιάζονται στη συνέχεια με τις μέγιστες τιμές roll, pitch και κάθετης ταχύτητας αντίστοιχα.

Όσον αφορά τη δομή του δικτύου της συνάρτησης Q , το επίπεδο της εισόδου αποτελείται από την κατάσταση s και τις ενέργειες a (Εικόνα 3.8b). Τα διανύσματα κατάστασης και ενεργειών περνούν από 2 ξεχωριστά επίπεδα των 16 και 32 κόμβων αντίστοιχα, πριν ενωθούν σε έναν κοινό ταυστή, ο οποίος στη συνέχεια εισάγεται σε 2 κρυφά επίπεδα των 256 κόμβων το καθένα. Η συνάρτηση ενεργοποίησης Rectified Linear Unit (ReLU) χρησιμοποιείται σε όλα τα επίπεδα. Η έξοδος του δικτύου της συνάρτησης Q είναι ένας πραγματικός αριθμός που αντιπροσωπεύει την τιμή Q .

Συνάρτηση Ανταμοιβής

Η ανταμοιβή είναι μια συνάρτηση ζωτικής σημασίας για τη σύγκλιση της διαδικασίας εκμάθησης, η οποία καθορίζεται από τον χρήστη και θα πρέπει να ικανοποιεί τις απαιτήσεις της εφαρμογής. Για το σκοπό αυτό, λαμβάνοντας υπόψη την προς μελέτη εφαρμογή, δηλαδή τη μεταφορά του φορτίου χωρίς ταλάντωση του σχοινιού προς θέσεις αναφοράς, η ανταμοιβή υπολογίζεται σε κάθε αλληλεπίδραση του πράκτορα με το περιβάλλον ως ο σταθμισμένος γραμμικός συνδυασμός των ακόλουθων όρων:

$$r_t = \mathbf{w}^T \boldsymbol{\rho}_t \quad (13)$$

$$r_t = \mathbf{w}^T [\|\mathbf{p}_t - \mathbf{p}_{ref}\|, \|\mathbf{v}_t\|, \|\boldsymbol{\eta}_t\|, \|\dot{\boldsymbol{\eta}}_t\|, \|\mathbf{n}_t\|, \|\dot{\mathbf{n}}_t\|, \|a_t\|, \|a_t - a_{t-1}\|]^T \quad (14)$$

όπου \mathbf{w} είναι ένα διάνυσμα βαρών που αντιστοιχίζονται στους όρους του διανύσματος ρ_t σε κάθε δεδομένο χρονικό βήμα t . Χρησιμοποιούνται αρνητικά βάρη αφού ο στόχος έγκειται στη μεγιστοποίηση της απόδοσης. Με βάση την προαναφερθείσα ανταμοιβή, ενθαρρύνεται η ελαχιστοποίηση του σφάλματος στη θέση και η μείωση της ταλαντωτικής κίνησης του σχοινιού. Επιπλέον, αποφεύγεται η σπασμωδική κίνηση του οχήματος, λόγω άσκοπων και απότομων αλλαγών στα σήματα ελέγχου, ενώ μειώνεται και η κατανάλωση ενέργειας.

Διαδικασία Εκμάθησης

Ο πράκτορας εκπαιδεύεται χρησιμοποιώντας τον αλγόριθμο TD3, ο οποίος επιστρατεύει τη δομή των αλγορίθμων actor-critic και παράγει συνεχείς ενέργειες. Ο αλγόριθμος TD3 θεωρείται επέκταση του αλγορίθμου Deep Deterministic Policy Gradient (DDPG) [52] και στοχεύει στη μείωση της υπερεκτίμησης της συνάρτησης Q . Σε σύγκριση με τον αλγόριθμο DDPG, ο TD3 χρησιμοποιεί ένα ζεύγος νευρωνικών δικτύων, Q_{ϕ_1} και Q_{ϕ_2} , προκειμένου να προσεγγίσει τη συνάρτηση Q και επιλέγεται αυτό με την μικρότερη τιμή κατά την ενημέρωση των δικτύων critic. Επιπλέον, στον TD3, η πολιτική και τα target δίκτυα ενημερώνονται λιγότερο συχνά από τα δίκτυα της συνάρτησης Q . Προτείνεται μία ενημέρωση της πολιτικής και των target δικτύων για κάθε δύο ενημερώσεις της συνάρτησης Q . Τέλος, προστίθεται θόρυβος στην ενέργεια που παράγεται από το target δίκτυο actor και, ως εκ τούτου, η εκτίμηση της συνάρτησης Q εξομαλύνεται για παρόμοιες ενέργειες. Εκτενείς λεπτομέρειες σχετικά με τον αλγόριθμο TD3 παρουσιάζονται στο [19].

Εν συντομία, τα δίκτυα critic ενημερώνονται με βάση το mean-squared Bellman error, ενώ η πολιτική εφαρμόζοντας gradient ascent ως προς τα βάρη θ με στόχο τη μεγιστοποίηση της εκτιμώμενης συνάρτησης Q . Επιπλέον, συμπεριλαμβάνονται target δίκτυα, με ίδια δομή με τα κύρια, έτσι ώστε να διασφαλίζεται η ευστάθεια της διαδικασίας εκπαίδευσης. Τα βάρη των target δικτύων συγχρονίζονται με τα βάρη των κύριων με αργό ρυθμό.

Το UAV με το αναρτημένο μέσω σχοινιού φορτίο, που απεικονίζεται στην Εικόνα 2.1, εκπαιδεύτηκε στο περιβάλλον προσομοίωσης Gazebo της Εικόνας 2.2 σύμφωνα με τις παραμέτρους του Πίνακα 3.3. Το εργαλείο βελτιστοποίησης ADAM [40] χρησιμοποιήθηκε για την ενημέρωση των βαρών της πολιτικής και των δικτύων της συνάρτησης Q με ρυθμό εκμάθησης ίσο με 0.001. Κατά τη διάρκεια κάθε επεισοδίου, ο πράκτορας ξεκινά από μια τυχαία θέση, εντός ενός περιορισμένου τρισδιάστατου χώρου, ενώ η θέση αναφοράς είναι σταθερή. Το επεισόδιο τερματίζεται όταν επιτευχθεί ο μέγιστος αριθμός χρονικών βημάτων (512) ή όταν το σφάλμα θέσης υπερβεί ένα συγκεκριμένο όριο. Η διαδικασία εκμάθησης τερματίστηκε όταν το άθροισμα όλων των ανταμοιβών κατά τη διάρκεια ενός επεισοδίου, και ο αντίστοιχος κινούμενος μέσος όρος, συνέκλιναν, όπως απεικονίζεται στην Εικόνα 3.9. Θα πρέπει να τονιστεί ότι, δεδομένου ότι η πολιτική

εκπαιδεύεται με χρήση αποκλειστικά δεδομένων προσομοίωσης, η ύπαρξη ενός εξαιρετικά ρεαλιστικού περιβάλλοντος προσομοίωσης είναι κρίσιμος παράγοντας για την επιτυχή μεταφορά σε μεταγενέστερο στάδιο της εκπαιδευμένης πολιτικής στο πραγματικό όχημα.

Εύρωστος Έλεγχος με Βαθιά Ενισχυτική Μάθηση

Οι πολιτικές, που έχουν εκπαιδευτεί με βαθιά Ενισχυτική Μάθηση, έχουν αποδειχθεί ικανές να ελέγχουν ικανοποιητικά τη δυναμική διαφόρων ρομποτικών πλατφορμών. Ωστόσο, σημαντικό θέμα προς συζήτηση αποτελεί το χάσμα μεταξύ του περιβάλλοντος προσομοίωσης και των πραγματικών συνθηκών. Πιο συγκεκριμένα, οι πολιτικές συνήθως εκπαιδεύονται αποκλειστικά σε προσομοιωτές, όπου απεριόριστος όγκος δεδομένων είναι διαθέσιμος, καθώς η εκπαίδευση σε πραγματικά περιβάλλοντα μπορεί να είναι τόσο χρονοβόρα όσο και επικίνδυνη, ειδικά στα αρχικά στάδια της διαδικασίας εκμάθησης. Ως αποτέλεσμα, πιθανές αναντιστοιχίες μεταξύ του προσομοιωτή και του πραγματικού περιβάλλοντος μπορεί να επηρεάσουν την απόδοση της εκπαιδευμένης πολιτικής όταν μεταφερθεί στο πραγματικό ρομπότ. Επιπλέον, ένα άλλο κρίσιμο ζήτημα είναι η ευρωστία της πολιτικής ως προς τις δυναμικές παραμέτρους του συστήματος. Λαμβάνοντας υπόψη την περίπτωση της εναέριας μεταφοράς φορτίων, δεν είναι καθόλου πρακτικό να εκπαιδεύεται μια πολιτική για κάθε τιμή της μάζας του φορτίου ή του μήκους του σχοινιού. Οι προαναφερθείσες παράμετροι ενδέχεται να αλλάζουν ακόμη και κατά τη διάρκεια της ίδιας πτήσης. Για παράδειγμα, το UAV μπορεί να απελευθερώσει μέρος του φορτίου ή να συλλέξει ένα αντικείμενο κατά την ίδια αποστολή. Ομοίως, το μήκος του σχοινιού μπορεί να μεταβάλλεται όταν μια τροχαλία είναι προσαρτημένη στο όχημα [99], π.χ., για πλοήγηση σε στενούς χώρους.

Για το σκοπό αυτό, τροποποιούμε τη διαδικασία εκμάθησης της Ενότητας 3.4 και ενσωματώνουμε τη μέθοδο domain randomization [50], [67] προκειμένου να γεφυρωθεί το χάσμα μεταξύ του προσομοιωτή και του πραγματικού περιβάλλοντος και να αναπτυχθεί μία πολιτική που είναι εύρωστη ως προς τις διάφορες παραμέτρους του μοντέλου, π.χ. μάζα φορτίου και μήκος σχοινιού. Η βασική ιδέα πίσω από αυτή την τεχνική έγκειται στην τυχαιοποίηση των φυσικών ιδιοτήτων του περιβάλλοντος προσομοίωσης, που ονομάζεται source domain, κατά τη διάρκεια της εκπαίδευσης με απώτερο στόχο την εκμάθηση μιας πολιτικής που αποδίδει ικανοποιητικά σε όλες τις παραλλαγές του προσομοιωτή. Ως εκ τούτου, η εκπαιδευμένη πολιτική μπορεί να μεταφερθεί επιτυχώς στο πραγματικό περιβάλλον (target domain) καθώς αυτό αναμένεται να είναι ένα δείγμα του μεταβλητού source domain.

Μέθοδος Domain Randomization

Όπως προαναφέρθηκε, προκειμένου να γεφυρωθεί το χάσμα μεταξύ του περιβάλλοντος προσομοίωσης Gazebo της Ενότητας 2.2 και των πραγματικών συνθηκών και να ενισχυθεί η ευρωστία της πολιτικής, επιβάλλεται η τυχαιοποίηση πολλών παραμέτρων κατά τη διάρκεια της εκπαίδευσης. Προς αυτή την κατεύθυνση, τυχαιοποιούνται οι ακόλουθες

πτυχές του source domain: (α) μάζα m_L του φορτίου, (β) μήκος σχοινιού l , (γ) θόρυβος στην παρατήρηση του μοναδιαίου διανύσματος \mathbf{n} , (δ) θόρυβος στην παρατήρηση της ταχύτητας $\dot{\mathbf{n}}$, και (ε) δυνάμεις \mathbf{F}_Q που ασκούνται στο UAV. Ο προσθετικός θόρυβος παρατήρησης στις συνιστώσες του μοναδιαίου διανύσματος του σχοινιού \mathbf{n} και στην ταχύτητα $\dot{\mathbf{n}}$ μοντελοποιεί την αβεβαιότητα στην εκτίμηση της κατάστασης του σχοινιού με χρήση αισθητήρων, ενώ οι δυνάμεις $\mathbf{F}_Q \in \mathbb{R}^3$ αντιπροσωπεύουν τις διαταραχές που επιδρούν στο όχημα όταν επιχειρεί σε εξωτερικούς χώρους. Τα εύρη των παραμέτρων συνοψίζονται στον Πίνακα 3.4.

Διαδικασία Εκμάθησης

Προκειμένου να εισαχθεί η μέθοδος domain randomization στη διαδικασία εκπαίδευσης, απαιτούνται διάφορες τροποποιήσεις στο περιβάλλον προσομοίωσης σε σύγκριση με την Ενότητα 3.4. Συγκεκριμένα, μια πρισματική άρθρωση ενσωματώνεται στο quadrotor της Εικόνας 2.1 ώστε να τροποποιείται το μήκος του σχοινιού κατά τη διάρκεια της προσομοίωσης, ενώ σχεδιάζεται ένα plugin για να μεταβάλλεται δυναμικά η μάζα του φορτίου. Επιπλέον, αν και τα διανύσματα κατάστασης και ενεργειών καθώς και η συνάρτηση ανταμοιβής ορίζονται παρόμοια με τις Εξισώσεις 3.11, 3.12 και 14, τα βαθιά νευρωνικά δίκτυα, δηλαδή η πολιτική $\pi_\theta(s)$ και το δίκτυο της συνάρτησης Q $Q_\phi(s, a)$, αναθεωρούνται. Πιο συγκεκριμένα, η είσοδος της πολιτικής περιέχει το διάνυσμα κατάστασης, συμπεριλαμβανομένου και του θορύβου στην κατάσταση του σχοινιού. Αντίθετα, η πραγματική κατάσταση του συστήματος, χωρίς τον πρόσθετο θόρυβο, και οι ενέργειες συνενώνονται και αποτελούν την είσοδο στη συνάρτηση Q . Επιπλέον, δεδομένου ότι η συνάρτηση Q χρησιμοποιείται αποκλειστικά στην προσομοίωση, οι γνωστές τυχαίοποιημένες παράμετροι του Πίνακα 3.4 παρέχονται επίσης ως είσοδοι προκειμένου να υποστηριχθεί η διαδικασία της εκπαίδευσης, όπως προτείνεται στο [67].

Όσον αφορά τη στρατηγική εκμάθησης, χρησιμοποιούνται σταθερές παράμετροι στα αρχικά στάδια της εκπαίδευσης. Πιο συγκεκριμένα, ο θόρυβος παρατήρησης και οι εξωτερικές δυνάμεις δεν εμπεριέχονται αρχικά στην προσομοίωση ενώ η μάζα του φορτίου και το μήκος του σχοινιού παραμένουν ίσα με 0.5kg και 1.0m αντίστοιχα. Καθώς προχωρά η εκπαίδευση, οι παράμετροι τυχαίοποιούνται με την πάροδο του χρόνου και δειγματοληπτούνται σύμφωνα με μια ομοιόμορφη κατανομή με όρια που πλησιάζουν σταδιακά τα εύρη του Πίνακα 3.4. Με την σταδιακή ενσωμάτωση της τυχαίοποίησης στη διαδικασία εκπαίδευσης, αποφεύγεται η υιοθέτηση συντηρητικών πολιτικών [50]. Οι τυχαίοποιημένες παράμετροι δειγματοληπτούνται στην αρχή κάθε επεισοδίου και διατηρούνται σταθερές σε όλο το επεισόδιο, εκτός από τον θόρυβο παρατήρησης στην κατάσταση του σχοινιού που προστίθεται σε κάθε χρονικό βήμα. Τελικά, η εξέλιξη του αθροίσματος των ανταμοιβών, με βάση και πάλι τον αλγόριθμο TD3, καταδεικνύεται στην καμπύλη της Εικόνας 3.10.

Μη Γραμμικός Προβλεπτικός Έλεγχος

Σε αυτήν την ενότητα, παρουσιάζουμε μια προσέγγιση που βασίζεται στο μοντέλο του συστήματος, σε αντίθεση με τις προηγούμενες στρατηγικές των Ενότητων 3.4 και 3.5, για τον έλεγχο του UAV με το αναρτημένο μέσω σχοινιού φορτίο. Πιο συγκεκριμένα, σύμφωνα με τη θεωρία βέλτιστου ελέγχου, αναπτύσσουμε ένα σχήμα μη γραμμικού Προβλεπτικού Ελέγχου (NMPC), διατυπωμένο για τη δυναμική του συστήματος, προκειμένου να επιτευχθεί η μεταφορά του αναρτημένου φορτίου χωρίς ταλάντωση. Γενικά, ο NMPC έχει αποδειχθεί ένα αποτελεσματικό εργαλείο για τον έλεγχο των UAVs στο πλαίσιο διαφόρων εφαρμογών, π.χ., [14], [61] και [62].

Ωστόσο, ειδικά όταν το UAV με το αναρτημένο μέσω σχοινιού φορτίο επιχειρεί σε υπαίθριους χώρους, εξωτερικές διαταραχές, που προέρχονται κυρίως από τις επικρατούσες περιβαλλοντικές συνθήκες, επιδρούν στην πλατφόρμα. Οι προαναφερθείσες διαταραχές, μαζί με τις υποθέσεις κατά τη μοντελοποίηση του συστήματος, οδηγούν σε αναντιστοιχίες μεταξύ της ονομαστικής και της πραγματικής δυναμικής του συστήματος, οι οποίες επηρεάζουν την απόδοση των σχημάτων NMPC. Για το σκοπό αυτό, αξιοποιούμε Gaussian Processes (GPs) για να μάθουμε σε πραγματικό χρόνο και με μη παραμετρικό τρόπο τις αποκλίσεις που υπάρχουν μεταξύ του πραγματικού και του ονομαστικού μοντέλου του συστήματος [32], [38]. Προκειμένου να διασφαλιστεί επίδοση σε πραγματικό χρόνο παρά τις υπολογιστικές απαιτήσεις των GPs, και επίσης προσαρμοστικότητα στις συνθήκες που επικρατούν στο εξωτερικό περιβάλλον, αξιοποιείται μια σταθμισμένη (weighted) έκδοση της sparse GP παλινδρόμησης. Τελικά, η εκτιμώμενη δυναμική ενσωματώνεται σε ένα σχήμα learning-based NMPC έτσι ώστε να βελτιωθεί η απόδοση του ελέγχου.

Μη Γραμμικός Προβλεπτικός Έλεγχος με βάση το Ονομαστικό Μοντέλο

Προκειμένου να μεταφερθεί το φορτίο με ασφάλεια και με ελάχιστες ταλαντώσεις, διατυπώνεται αρχικά ένας NMPC για τη μη γραμμική δυναμική του συστήματος που περιγράφεται από την Εξίσωση 2.11. Ως εκ τούτου, η εφαρμογή του σχήματος NMPC απαιτεί, εκτός από την κατάσταση του οχήματος, την τάση $\|\mathbf{T}\|$ του σχοινιού, μια εκτίμηση της κατάστασης του σχοινιού, δηλαδή τις γωνίες η_L και τη γωνιακή ταχύτητα ω_L , και μια θέση αναφοράς \mathbf{p}_{ref} . Όπως αναφέρθηκε στην Ενότητα 2.3, το σχήμα NMPC υπολογίζει εντολές για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου.

Δεδομένου ότι ο στόχος του σχήματος NMPC είναι να περιορίσει τις ταλαντώσεις του σχοινιού και να ελαχιστοποιήσει την απόσταση μεταξύ της πραγματικής θέσης του UAV και της θέσης αναφοράς ενώ ταυτόχρονα ικανοποιούνται περιορισμοί στην είσοδο,

διατυπώνεται το ακόλουθο Πρόβλημα Βέλτιστου Ελέγχου (OCP):

$$\begin{aligned} \min_{\mathbf{v}} \int_{t_0}^{t_0+T} (\|\mathbf{x}(t) - \mathbf{x}_{ref}(t)\|_{\mathbf{Q}}^2 + \|\mathbf{v}(t)\|_{\mathbf{R}}^2) dt + \|\mathbf{x}(t_0 + T) - \mathbf{x}_{ref}(t_0 + T)\|_{\mathbf{P}}^2 \\ \text{subject to: } \mathbf{x}(t_0) = \mathbf{x}_0 \\ \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{v}) \text{ (Εξίσωση 2.11)} \\ \mathbf{v} \in \mathbb{U} \end{aligned} \quad (15)$$

όπου $\mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T \ \phi \ \theta \ \psi \ \boldsymbol{\eta}_L^T \ \boldsymbol{\omega}_L^T]^T \in \mathbb{R}^{13}$, $\mathbf{v} = [\phi_d \ \theta_d \ \psi_d \ v_{zd}]^T \in \mathbb{R}^4$ είναι τα διανύσματα κατάστασης και εισόδου, $\mathbf{x}_{ref} = [\mathbf{p}_{ref}^T \ \mathbf{0}_{1 \times 3} \ 0 \ 0 \ \psi_{ref} \ \mathbf{0}_{1 \times 2} \ \mathbf{0}_{1 \times 2}]^T \in \mathbb{R}^{13}$ είναι το επιθυμητό διάνυσμα κατάστασης, T είναι ο χρονικός ορίζοντας του προβλήματος βελτιστοποίησης, \mathbf{Q} είναι ο πίνακας στάθμισης για το σφάλμα κατάστασης, \mathbf{R} είναι ο πίνακας στάθμισης για την είσοδο ελέγχου, \mathbf{P} είναι ο πίνακας στάθμισης για το τερματικό σφάλμα κατάστασης και \mathbb{U} είναι το σύνολο των περιορισμών στην είσοδο που σχετίζονται με τα όρια των γωνιών roll, pitch και της κατακόρυφης ταχύτητας. Οι πίνακες στάθμισης \mathbf{Q} , \mathbf{R} , \mathbf{P} είναι θετικά ορισμένοι και διαγώνιοι και, για παράδειγμα, ο πίνακας \mathbf{Q} δομείται ως εξής:

$$\mathbf{Q} = \text{diag} \left([\mathbf{Q}_p \ \mathbf{Q}_v \ \mathbf{Q}_\eta \ \mathbf{Q}_{\eta_L} \ \mathbf{Q}_{\omega_L}] \right) \quad (16)$$

όπου \mathbf{Q}_p , \mathbf{Q}_v , \mathbf{Q}_η , \mathbf{Q}_{η_L} , \mathbf{Q}_{ω_L} είναι διαγώνια μητρώα στάθμισης που επιβάλλουν κυρώσεις στα σφάλματα της θέσης, της ταχύτητας, του προσανατολισμού του οχήματος και των γωνιών και της γωνιακής ταχύτητας του σχοινιού αντίστοιχα σε σχέση με το επιθυμητό διάνυσμα κατάστασης. Ο πίνακας \mathbf{P} ορίζεται με παρόμοιο τρόπο.

Το προαναφερθέν OCP επιλύεται σε κάθε επανάληψη υπολογίζοντας μια βέλτιστη ακολουθία εντολών ελέγχου. Μόνο η πρώτη είσοδος της παραγόμενης ακολουθίας εφαρμόζεται στο UAV και ολόκληρη η διαδικασία βελτιστοποίησης επαναλαμβάνεται. Προκειμένου να σχεδιαστεί το πρόβλημα MPC και να επιτευχθεί επίδοση σε πραγματικό χρόνο, χρησιμοποιείται το ACADO Toolkit [33] σε συνδυασμό με τον επιλύτη qpOASES [17]. Με βάση πειράματα που έγιναν στον ενσωματωμένο υπολογιστή του UAV, συγκεκριμένα στο Jetson AGX Xavier, διαπιστώθηκε ότι το ACADO Toolkit με τον επιλύτη qpOASES παρέχουν λύση στο OCP μετά από 3.4 ms κατά μέσο όρο και, ως εκ τούτου, επιτυγχάνεται μια εξαιρετικά ικανοποιητική συχνότητα ελέγχου.

Επαυξημένη Δυναμική

Στην παραπάνω ενότητα, χρησιμοποιήθηκε η ονομαστική δυναμική του συστήματος $\hat{\mathbf{x}}_{k+1} = \mathbf{f}_{nom}(\mathbf{x}_k, \mathbf{v}_k)$, όπου \mathbf{f}_{nom} είναι το ισοδύναμο μοντέλο διακριτού χρόνου της Εξίσωσης 2.11 και $\hat{\mathbf{x}}_{k+1}$ η εκτίμηση της κατάστασης την επόμενη χρονική στιγμή t_{k+1} . Αν και το προαναφερθέν μοντέλο είναι επαρκές για τον έλεγχο του συστήματος, η απόδοση μπορεί να βελτιωθεί περαιτέρω καθώς το ονομαστικό μοντέλο δεν συλλαμβάνει ολόκληρο το εύρος της πραγματικής δυναμικής του συστήματος. Κατά συνέπεια, προκύπτουν

αποκλίσεις μεταξύ της μετρούμενης και της εκτιμώμενης επόμενης κατάστασης του συστήματος και παρατηρείται ένα σφάλμα $\mathbf{e}_k = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}$ για τους ακόλουθους λόγους:

- Οι δυνάμεις οπισθέλκουσας, που προκαλούνται από τον αέρα και ασκούνται τόσο στο όχημα όσο και στο φορτίο, παραμελούνται στο ονομαστικό μοντέλο.
- Η ευκαμψία του σχοινιού μπορεί να παραβιάσει την υπόθεση ότι είναι τεντωμένο (Εξίσωση 2.2) και, ως εκ τούτου, την επακόλουθη ανάλυση της Ενότητας 2.5.1.
- Το άνω άκρο του σχοινιού θεωρείται ότι συμπίπτει με το κέντρο μάζας του οχήματος. Ωστόσο, αυτό είναι πρακτικά ανέφικτο λόγω των περιορισμών που προκύπτουν από την κατασκευή του UAV.
- Η ενσωμάτωση ενός υπολογιστή και αισθητήρων, π.χ. μιας κάμερας, στο UAV μπορεί να οδηγήσει σε μετατόπιση του κέντρου μάζας του οχήματος.
- Άλλες αποκλίσεις μπορεί να υπάρχουν, π.χ. σφάλματα στη βαθμονόμηση της δυναμοκυψέλης που μετρά την τάση του σχοινιού.

Η ενσωμάτωση των προαναφερθέντων αποκλίσεων στο ονομαστικό μοντέλο του συστήματος όχι μόνο αυξάνει την πολυπλοκότητά του αλλά συνεπάγεται επίσης εξαντλητικές και πιθανώς δαπανηρές διαδικασίες για την αναγνώριση του συστήματος, π.χ. αεροδυναμικές δοκιμές σε σήραγγες. Επιπλέον, ορισμένες πηγές σφαλμάτων υπόκεινται στις συνθήκες που επικρατούν στο εξωτερικό περιβάλλον όπου λειτουργεί το UAV, π.χ. ταχύτητα του ανέμου και, ως εκ τούτου, δεν μπορούν να προβλεφθούν.

Για το σκοπό αυτό, στοχεύουμε να εκμεταλλευτούμε GPs για την μη παραμετρική εκτίμηση της μη μοντελοποιημένης δυναμικής. Εμπνευσμένοι από τα [32], [38], η πραγματική δυναμική του συστήματος μπορεί να αναπαρασταθεί από την εξίσωση:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ &= \mathbf{f}_{nom}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{B}_d(\mathbf{d}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k) \end{aligned} \quad (17)$$

όπου $\mathbf{w}_k \sim \mathcal{N}(0, \Sigma^w)$ είναι γκαουσιανός θόρυβος με διαγώνιο πίνακα διακύμανσης Σ^w , \mathbf{d} υποδηλώνει το άγνωστο δυναμικό μέρος που πρέπει να προσεγγιστεί από τις GPs και ο πίνακας \mathbf{B}_d ορίζει τον υποχώρο της δυναμικής του συστήματος που επηρεάζεται από τις αποκλίσεις στο μοντέλο. Υποτίθεται ότι σφάλματα στο μοντέλο υπάρχουν αποκλειστικά στη δεύτερης τάξης δυναμική του συστήματος και, επομένως, $\mathbf{B}_d = [\mathbf{0}_{3 \times 3}; \mathbf{I}_{3 \times 3}; \mathbf{0}_{3 \times 3}; \mathbf{0}_{2 \times 2}; \mathbf{I}_{2 \times 2}]$. Κατά συνέπεια, ο απώτερος στόχος είναι να εκτιμηθούν τα σφάλματα $\mathbf{d}_v \in \mathbb{R}^3$ και $\mathbf{d}_\omega \in \mathbb{R}^2$ που επηρεάζουν τη μεταφορική δυναμική του UAV και τη γωνιακή κίνηση του σχοινιού αντίστοιχα.

Παλινδρόμηση με Gaussian Processes

Στην ακόλουθη ενότητα, παρουσιάζεται εν συντομία η παλινδρόμηση με GPs σύμφωνα με τη λεπτομερή περιγραφή στο [72]. Γενικά, οι GPs μπορούν να αξιοποιηθούν για την προσέγγιση μιας άγνωστης συνάρτησης $g(\mathbf{z}) : \mathbb{R}^{n_z} \rightarrow \mathbb{R}$, με βάση μια συλλογή

M θορυβωδών παρατηρήσεων y_j της αληθινής συνάρτησης $g(\mathbf{z})$ που μετρώνται στις εισόδους \mathbf{z}_j , όπου:

$$y_j = g(\mathbf{z}_j) + w_j, \quad w_j \sim \mathcal{N}(0, \sigma^2) \quad (18)$$

Πιο συγκεκριμένα, δημιουργείται ένα σετ δεδομένων D :

$$D = \left\{ \begin{aligned} \mathbf{Z} &= [\mathbf{z}_1, \dots, \mathbf{z}_M]^T \in \mathbb{R}^{M \times n_z}, \\ \mathbf{y} &= [y_1, \dots, y_M]^T \in \mathbb{R}^M \end{aligned} \right\} \quad (19)$$

το οποίο μπορεί να αξιοποιηθεί προκειμένου να προβλεφθεί η εκ των υστέρων (posterior) κατανομή σε οποιαδήποτε είσοδο \mathbf{z} . Με βάση το σετ δεδομένων D , η εκ των υστέρων κατανομή καθορίζεται από τις ακόλουθες μέση τιμή και συνδιακύμανση:

$$\mu(\mathbf{z}) = \mathbf{k}_{zZ}(\mathbf{K}_{ZZ} + \mathbf{I}_{M \times M} \sigma^2)^{-1} \mathbf{y} \quad (20)$$

$$\Sigma(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) - \mathbf{k}_{zZ}(\mathbf{K}_{ZZ} + \mathbf{I}_{M \times M} \sigma^2)^{-1} \mathbf{k}_{zZ} \quad (21)$$

όπου $k(\mathbf{z}_i, \mathbf{z}_j) = \sigma_f^2 \exp(-0.5(\mathbf{z}_i - \mathbf{z}_j)^T \mathbf{L}(\mathbf{z}_i - \mathbf{z}_j)) \in \mathbb{R}$ είναι η squared exponential kernel συνάρτηση με $\mathbf{L} \in \mathbb{R}^{n_z \times n_z}$ και διακύμανση σ_f^2 , $[\mathbf{K}_{ZZ}]_{i,j} = k(\mathbf{z}_i, \mathbf{z}_j)$ είναι το i, j στοιχείο του πίνακα $\mathbf{K}_{ZZ} \in \mathbb{R}^{M \times M}$ με $\mathbf{z}_i, \mathbf{z}_j \in \mathbf{Z}$, $[\mathbf{k}_{zZ}]_j = k(\mathbf{z}, \mathbf{z}_j)$ είναι το j στοιχείο του διανύσματος γραμμής $\mathbf{k}_{zZ} \in \mathbb{R}^M$ με $\mathbf{z}_j \in \mathbf{Z}$, και $\mathbf{k}_{zZ} = \mathbf{k}_{zZ}^T$.

Ωστόσο, το υπολογιστικό κόστος της παλινδρόμησης με GPs, που καθορίζεται κυρίως από την αντιστροφή του πίνακα \mathbf{K}_{ZZ} , εξαρτάται σε μεγάλο βαθμό από το μέγεθος M του σετ δεδομένων και καθιστά τη χρήση της μη πρακτική σε εφαρμογές πραγματικού χρόνου όπου απαιτείται σημαντικός όγκος δεδομένων. Επιπλέον, θα πρέπει να σημειωθεί ότι οι GPs μπορούν να προσεγγίσουν μονοδιάστατες συναρτήσεις και, ως εκ τούτου, απαιτούνται 5 ανεξάρτητες GPs για την εκτίμηση των σφαλμάτων στο μοντέλο $\mathbf{d}_w \in \mathbb{R}^3$ και $\mathbf{d}_\omega \in \mathbb{R}^2$. Κατά συνέπεια, δεδομένου ότι το σετ δεδομένων πρέπει να ενημερώνεται συνεχώς – επομένως, οι εκ των προτέρων υπολογισμοί είναι αδύνατοι – και η επίδοση σε πραγματικό χρόνο είναι προϋπόθεση για τον επιτυχή έλεγχο του οχήματος, θα πρέπει να χρησιμοποιηθεί μια προσεγγιστική τεχνική για την εκμάθηση των αποκλίσεων στο μοντέλο.

Παλινδρόμηση με Sparse Gaussian Processes

Προκειμένου να μειωθούν οι υπολογιστικές απαιτήσεις της παλινδρόμησης με GPs και να επιτευχθεί επίδοση σε πραγματικό χρόνο, αξιοποιείται η μέθοδος sparse GP regression. Η μέθοδος αυτή έγκειται στη χρήση ενός υποσυνόλου του αρχικού σετ δεδομένων, που αποτελείται από m σημεία (inducing points) με $m \ll M$, προκειμένου να αυξηθεί η υπολογιστική αποδοτικότητα των GPs για μεγάλα σετ δεδομένων, παρέχοντας ταυτόχρονα ικανοποιητικές προβλέψεις. Με βάση το [87], υποθέτοντας ένα σετ inducing points $\mathbf{Z}_{ind} = [\mathbf{z}_1, \dots, \mathbf{z}_m]^T \in \mathbb{R}^{m \times n_z}$, η εκ των υστέρων κατανομή προσεγγίζεται με

τις ακόλουθες μέση τιμή και συνδιακύμανση:

$$\mu(\mathbf{z}) = \mathbf{k}_{zZ_{ind}} \mathbf{K}_{Z_{ind}Z_{ind}}^{-1} \boldsymbol{\mu}_m \quad (22)$$

$$\Sigma(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) - \mathbf{k}_{zZ_{ind}} \mathbf{K}_{Z_{ind}Z_{ind}}^{-1} \mathbf{k}_{Z_{ind}z} + \mathbf{k}_{zZ_{ind}} \boldsymbol{\Sigma} \mathbf{k}_{Z_{ind}z} \quad (23)$$

όπου $\boldsymbol{\Sigma} = (\mathbf{K}_{Z_{ind}Z_{ind}} + \sigma^{-2} \mathbf{K}_{Z_{ind}Z} \mathbf{K}_{ZZ_{ind}})^{-1}$, $\boldsymbol{\mu}_m = \sigma^{-2} \mathbf{K}_{Z_{ind}Z_{ind}} \boldsymbol{\Sigma} \mathbf{K}_{Z_{ind}Z} \mathbf{y}$, $[\mathbf{K}_{Z_{ind}Z_{ind}}]_{i,j} = k(\mathbf{z}_i, \mathbf{z}_j)$ είναι το i, j στοιχείο του πίνακα $\mathbf{K}_{Z_{ind}Z_{ind}} \in \mathbb{R}^{m \times m}$ με $\mathbf{z}_i, \mathbf{z}_j \in \mathbf{Z}_{ind}$, $[\mathbf{K}_{Z_{ind}Z}]_{i,j} = k(\mathbf{z}_i, \mathbf{z}_j)$ είναι το i, j στοιχείο του πίνακα $\mathbf{K}_{Z_{ind}Z} \in \mathbb{R}^{m \times M}$ με $\mathbf{z}_i \in \mathbf{Z}_{ind}$ και $\mathbf{z}_j \in \mathbf{Z}$, $\mathbf{K}_{ZZ_{ind}} = \mathbf{K}_{Z_{ind}Z}^T$, $[\mathbf{k}_{zZ_{ind}}]_j = k(\mathbf{z}, \mathbf{z}_j)$ είναι το j στοιχείο του διανύσματος γραμμής $\mathbf{k}_{zZ_{ind}} \in \mathbb{R}^m$ με $\mathbf{z}_j \in \mathbf{Z}_{ind}$, και $\mathbf{k}_{Z_{ind}z} = \mathbf{k}_{zZ_{ind}}^T$.

Χρησιμοποιώντας την παραπάνω προσεγγιστική τεχνική με υπολογιστική πολυπλοκότητα $\mathcal{O}(Mm^2)$, καθίσταται εφικτή η χρήση sparse GPs για συστήματα πραγματικού χρόνου και online εκμάθηση, σε αντίθεση με τη δυσεπίλυτη τυπική παλινδρόμηση με GPs (Εξισώσεις 3.20 και 3.21) πολυπλοκότητας $\mathcal{O}(M^3)$.

Παλινδρόμηση με Weighted & Sparse Gaussian Processes

Ωστόσο, κατά τη διάρκεια των πειραμάτων, το UAV λειτουργεί σε μη στατικά εξωτερικά περιβάλλοντα με ποικίλες και μεταβλητές περιβαλλοντικές συνθήκες, π.χ. ταχύτητα ανέμου, που επηρεάζουν τη δυναμική του συστήματος και, ως εκ τούτου, έχουν ως αποτέλεσμα τη χρονική διακύμανση των αποκλίσεων στο μοντέλο \mathbf{d}_v και \mathbf{d}_ω . Κατά συνέπεια, κατά την εκμάθηση των σφαλμάτων στο μοντέλο, απαιτείται προσαρμοστικότητα στις εξωτερικές συνθήκες προκειμένου να αποτυπωθεί η χρονική διακύμανση των συναρτήσεων που προσεγγίζονται από τις GPs. Για το σκοπό αυτό, χρησιμοποιείται σταθμισμένη (weighted) παλινδρόμηση με GPs σύμφωνα με το [11], αλλά κατάλληλα προσαρμοσμένη στην περίπτωση των sparse GPs, διασφαλίζοντας έτσι τόσο προσαρμοστικότητα όσο και υπολογιστική αποτελεσματικότητα.

Πιο συγκεκριμένα, αντιστοιχίζεται ένα βάρος $w_i \in \mathbb{R}^+$ σε κάθε μέτρηση y_i με απώτερο στόχο την ενίσχυση των πιο πρόσφατων μετρήσεων κατά τη διάρκεια της παλινδρόμησης με GPs. Έτσι, εισάγονται οι πίνακες $\mathbf{W} = \text{diag}(\sqrt{w_1}, \dots, \sqrt{w_M}) \in \mathbb{R}^{M \times M}$ και $\mathbf{W}_{ind} = \text{diag}(\sqrt{w_1}, \dots, \sqrt{w_m}) \in \mathbb{R}^{m \times m}$, που σχετίζονται με ολόκληρο το σετ δεδομένων και τα inducing points αντίστοιχα. Τροποποιώντας κατάλληλα τα kernels, η εκ των υστέρων κατανομή για την περίπτωση της παλινδρόμησης με weighted & sparse GPs χαρακτηρίζεται από τις εξισώσεις:

$$\mu(\mathbf{z}) = \tilde{\mathbf{k}}_{zZ_{ind}} \tilde{\mathbf{K}}_{Z_{ind}Z_{ind}}^{-1} \tilde{\boldsymbol{\mu}}_m \quad (24)$$

$$\Sigma(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) - \tilde{\mathbf{k}}_{zZ_{ind}} \tilde{\mathbf{K}}_{Z_{ind}Z_{ind}}^{-1} \tilde{\mathbf{k}}_{Z_{ind}z} + \tilde{\mathbf{k}}_{zZ_{ind}} \tilde{\boldsymbol{\Sigma}} \tilde{\mathbf{k}}_{Z_{ind}z} \quad (25)$$

όπου $\tilde{\boldsymbol{\Sigma}} = (\tilde{\mathbf{K}}_{Z_{ind}Z_{ind}} + \sigma^{-2} \tilde{\mathbf{K}}_{Z_{ind}Z} \tilde{\mathbf{K}}_{ZZ_{ind}})^{-1}$, $\tilde{\mathbf{y}} = \mathbf{W} \mathbf{y}$, $\tilde{\boldsymbol{\mu}}_m = \sigma^{-2} \tilde{\mathbf{K}}_{Z_{ind}Z_{ind}} \tilde{\boldsymbol{\Sigma}} \tilde{\mathbf{K}}_{Z_{ind}Z} \tilde{\mathbf{y}}$, $\tilde{\mathbf{k}}_{zZ_{ind}} = \mathbf{k}_{zZ_{ind}} \mathbf{W}_{ind}$, $\tilde{\mathbf{K}}_{Z_{ind}Z_{ind}} = \mathbf{W}_{ind} \mathbf{K}_{Z_{ind}Z_{ind}} \mathbf{W}_{ind}^T$, και $\tilde{\mathbf{K}}_{Z_{ind}Z} = \mathbf{W}_{ind} \mathbf{K}_{Z_{ind}Z} \mathbf{W}$.

Σημειώνεται ότι εάν όλα τα βάρη είναι ίσα με 1, η μέση τιμή και η συνδιακύμανση είναι ίδιες με τις αντίστοιχες τιμές των Εξισώσεων 3.22 και 3.23.

Πρακτική Υλοποίηση

Σε αυτήν την ενότητα, παρουσιάζεται λεπτομερώς η υλοποίηση της παλινδρόμησης με weighted & sparse GPs για την περίπτωση της εναέριας μεταφοράς φορτίων αναρτημένων μέσω σχοινιού.

1) *Επικαιροποίηση σετ δεδομένων*: Σε κάθε πείραμα, το σετ δεδομένων D είναι αρχικά είτε κενό είτε γεμάτο με μετρήσεις από προηγούμενα πειράματα. Μόλις συλλεχθεί μια νέα μέτρηση κατά τη λειτουργία του UAV, το σετ D ενημερώνεται σε πραγματικό χρόνο. Ωστόσο, το μέγεθος του σετ δεδομένων περιορίζεται προκειμένου να διασφαλιστεί από υπολογιστικής πλευράς η δυνατότητα υλοποίησης της παλινδρόμησης με την πάροδο του χρόνου. Ως εκ τούτου, όταν το σετ δεδομένων D φτάσει σε ένα μέγιστο μέγεθος M δεδομένων, η νέα μέτρηση $\{\mathbf{z}', y'\}$ αντικαθιστά ένα δεδομένο από το τρέχον σετ D σύμφωνα με το ακόλουθο κριτήριο:

$$\min_{i \in D} (\alpha \|\mathbf{z}' - \mathbf{z}_i\|^2 + \beta w_i^2) \quad (26)$$

όπου $\alpha, \beta \in \mathbb{R}^+$ είναι παράμετροι προς ρύθμιση. Με την εφαρμογή της προαναφερθείσας στρατηγικής, ενθαρρύνεται η αντικατάσταση παλαιότερων μετρήσεων, παρόμοιων με τις εισερχόμενες, διασφαλίζοντας έτσι ότι το σετ δεδομένων είναι επικαιροποιημένο και καλύπτει επαρκώς τον χώρο των εισόδων.

2) *Επιλογή Βαρών*: Κάθε μέτρηση $\{\mathbf{z}_i, y_i\}$ στο σετ δεδομένων σχετίζεται με ένα βάρος w_i που υπολογίζεται σύμφωνα με τη χρονική στιγμή t_i κατά την οποία συλλέχθηκε η μέτρηση:

$$w_i = k_0 + c \cdot \tanh(k_1 t_i) \quad (27)$$

με $k_0, c, k_1 \in \mathbb{R}^+$. Η επιλογή της παραπάνω συνάρτησης ικανοποιεί την απαίτηση για ενίσχυση των πιο πρόσφατων μετρήσεων και βελτιώνει την προσαρμοστικότητα της διαδικασίας εκμάθησης στις μεταβλητές συνθήκες.

3) *Επιλογή των inducing points*: Η επιλογή του συνόλου \mathbf{Z}_{ind} είναι κρίσιμη για την ακρίβεια της παλινδρόμησης με sparse GPs. Δεδομένης μιας εισόδου \mathbf{z} , μια απλοϊκή προσέγγιση θα ήταν η επιλογή των πλησιέστερων σημείων ως inducing points, δηλαδή των m σημείων $\mathbf{z}_i \in D$ με την μικρότερη απόσταση $\|\mathbf{z} - \mathbf{z}_i\|$. Ωστόσο, αυτή η προσέγγιση μπορεί να οδηγήσει σε αριθμητική αστάθεια κατά την αντιστροφή του πίνακα $\tilde{\mathbf{K}}_{\mathbf{Z}_{ind}\mathbf{Z}_{ind}}$. Επομένως, αντιστοιχίζεται σε κάθε μέτρηση μια πιθανότητα, με χαμηλότερες τιμές της απόστασης $\|\mathbf{z} - \mathbf{z}_i\|$ να αντιστοιχούν σε υψηλότερη πιθανότητα, και m σημεία επιλέγονται τυχαία με βάση αυτή την κατανομή.

4) *Προσέγγιση Παραγώγων*: Ένα άλλο πλεονέκτημα των GPs είναι η δυνατότητα πρόβλεψης των παραγώγων της συνάρτησης $g(\mathbf{z})$ ως προς τις μεταβλητές εισόδου

$\mathbf{z} = [z_1, \dots, z_{n_z}]^T$. Ο υπολογισμός των παραγώγων προϋποθέτει αποκλειστικά τη διαφόριση της συνάρτησης kernel. Συνεπώς, λαμβάνοντας υπόψη ότι η επιλεγμένη squared exponential kernel συνάρτηση είναι απείρως διαφορίσιμη, η κλίση $\nabla \mu(\mathbf{z}) \in \mathbb{R}^{n_z}$ και η εσσιανή $\mathbf{H}(\mathbf{z}) \in \mathbb{R}^{n_z \times n_z}$ της συνάρτησης g σε μια δοκιμαστική είσοδο \mathbf{z} μπορούν να προσεγγιστούν από τις παρακάτω εξισώσεις:

$$\nabla \mu(\mathbf{z}) = \left[\frac{\partial \mu}{\partial z_1}, \dots, \frac{\partial \mu}{\partial z_{n_z}} \right]^T = \left[\frac{\partial \tilde{\mathbf{k}}_{z Z_{ind}}}{\partial z_1}, \dots, \frac{\partial \tilde{\mathbf{k}}_{z Z_{ind}}}{\partial z_{n_z}} \right]^T \tilde{\mathbf{K}}_{Z_{ind} Z_{ind}}^{-1} \tilde{\boldsymbol{\mu}}_m \quad (28)$$

$$\mathbf{H}(\mathbf{z}) = \begin{bmatrix} \frac{\partial^2 \tilde{\mathbf{k}}_{z Z_{ind}}}{\partial z_1^2} & \dots & \frac{\partial^2 \tilde{\mathbf{k}}_{z Z_{ind}}}{\partial z_1 \partial z_{n_z}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \tilde{\mathbf{k}}_{z Z_{ind}}}{\partial z_1 \partial z_{n_z}} & \dots & \frac{\partial^2 \tilde{\mathbf{k}}_{z Z_{ind}}}{\partial z_{n_z}^2} \end{bmatrix} \tilde{\mathbf{K}}_{Z_{ind} Z_{ind}}^{-1} \tilde{\boldsymbol{\mu}}_m \quad (29)$$

Σημειώνεται ότι τα $\tilde{\mathbf{K}}_{Z_{ind} Z_{ind}}^{-1}$ και $\tilde{\boldsymbol{\mu}}_m$ έχουν ήδη υπολογιστεί κατά την παλινδρόμηση, και επομένως, το υπολογιστικό κόστος των Εξισώσεων 3.28 και 3.29 είναι ασήμαντο. Μπορούν επίσης να υπολογιστούν παράγωγοι υψηλότερης τάξης. Ωστόσο, το σφάλμα προσέγγισης αυξάνεται λαμβάνοντας υπόψη μόνο τις μετρήσεις \mathbf{y} της συνάρτησης, δηλαδή, χωρίς την προσθήκη παρατηρήσεων των παραγώγων κατά την παλινδρόμηση.

5) *Μετρήσεις*: Όπως προαναφέρθηκε, ο απώτερος στόχος είναι να προσεγγιστούν οι αποκλίσεις \mathbf{d}_v και \mathbf{d}_ω χρησιμοποιώντας 5 ανεξάρτητες GPs. Ωστόσο, οι μη μοντελοποιημένες διαταραχές, π.χ. δυνάμεις οπισθέλκουσας, εκφράζονται συνήθως σε σχέση με το σταθερό σύστημα συντεταγμένων του οχήματος \mathbf{B} . Συνεπώς, η διαδικασία της εκπαίδευσης προσαρμόζεται έτσι ώστε να προσεγγίζονται τα σφάλματα στο μοντέλο ${}^B \mathbf{d}_v$ και ${}^B \mathbf{d}_\omega$, που ορίζονται ως προς το \mathbf{B} . Ως εκ τούτου, διασφαλίζεται η συνέπεια στο σετ δεδομένων, εξαλείφεται η εξάρτηση από τον προσανατολισμό του οχήματος, ειδικά τη γωνία yaw, και μειώνονται η διάσταση του χώρου εισόδου, και κατά συνέπεια, η υπολογιστική πολυπλοκότητα.

Προς αυτή την κατεύθυνση, εισάγουμε την ταχύτητα του UAV ως προς το \mathbf{B} , ${}^B \mathbf{v} = \mathbf{R}_{WB}^T \mathbf{v} = [{}^B v_x \quad {}^B v_y \quad {}^B v_z]^T$, καθώς και τις γωνίες του σχοινιού ${}^B \boldsymbol{\eta}_L = [{}^B \phi_L \quad {}^B \theta_L]^T$ και τη γωνιακή ταχύτητα ${}^B \boldsymbol{\omega}_L = {}^B \dot{\boldsymbol{\eta}}_L$ που υπολογίζονται από το μοναδιαίο διάνυσμα του σχοινιού ${}^B \mathbf{n} = \mathbf{R}_{WB}^T \mathbf{n}$ και την ταχύτητα ${}^B \dot{\mathbf{n}} = \mathbf{R}_{WB}^T \dot{\mathbf{n}}$ (ορίζονται παρόμοια με τις Εξισώσεις 2.2 και 2.3 αντίστοιχα). Κατά την λειτουργία του UAV, μετρήσεις συλλέγονται χρησιμοποιώντας το ονομαστικό μοντέλο του συστήματος, που παρουσιάζεται στην Ενότητα 2.5.1, ενώ οι είσοδοι \mathbf{z} για κάθε συνιστώσα των προσεγγιζόμενων διανυσμάτων ${}^B \mathbf{d}_v$ και ${}^B \mathbf{d}_\omega$ επιλέγονται με βάση φυσικές παρατηρήσεις, όπως συνοφίζεται στον Πίνακα 3.5. Σημειώνεται ότι η επιλογή των κατάλληλων εισόδων είναι κρίσιμη για την επιτυχή εκμάθηση άγνωστων συναρτήσεων μέσω GPs. Μόλις είναι διαθέσιμες οι προβλέψεις των ${}^B \mathbf{d}_v$ και ${}^B \mathbf{d}_\omega$, ο μετασχηματισμός τους ως προς το σύστημα αναφοράς \mathbf{W} πραγματοποιείται αντίστροφα. Τέλος, πρέπει να σημειωθεί ότι ορισμένες αποκλίσεις μοιράζονται το ίδιο διάνυσμα εισόδου και, επομένως, κάποιοι υπολογισμοί πινάκων αποφεύγονται με αποτέλεσμα την περαιτέρω μείωση του υπολογιστικού κόστους.

6) **Υπολογιστικό Κόστος:** Όσον αφορά το υπολογιστικό κόστος της παλινδρόμησης με weighted & sparse GPs, ο μέσος χρόνος εκτέλεσης για την εξαγωγή συμπερασμάτων από τις 5 GPs, θεωρώντας ένα σετ δεδομένων $M = 1000$ μετρήσεων και $m = 30$ inducing points για κάθε GP, είναι ίσος με 30.9 ms στο Jetson AGX Xavier.

Μη Γραμμικός Προβλεπτικός Έλεγχος με Gaussian Processes

Για να επιτευχθεί η μεταφορά του πολυκοπτότερου προς τις θέσεις αναφοράς $\mathbf{p}_{ref} \in \mathbb{R}^3$ με ταυτόχρονη ελαχιστοποίηση της γωνιακής κίνησης του σχοινιού, παρά τις διαταραχές που ασκούνται στην πλατφόρμα, ένα NMPC σχήμα διατυπώνεται για την επαυξημένη δυναμική του συστήματος που συμπεριλαμβάνει τις αποκλίσεις \mathbf{d}_v και \mathbf{d}_ω , όπως αυτές προσεγγίζονται από την παλινδρόμηση με weighted & sparse GPs. Για το σκοπό αυτό, ορίζεται το ακόλουθο OCP:

$$\begin{aligned} \min_{\mathbf{v}_0, \dots, \mathbf{v}_{N-1}} \quad & \sum_{k=0}^{N-1} \left((\mathbf{x}_k - \mathbf{x}_{ref})^T \mathbf{Q} (\mathbf{x}_k - \mathbf{x}_{ref}) + \mathbf{v}_k^T \mathbf{R} \mathbf{v}_k \right) \\ & + (\mathbf{x}_N - \mathbf{x}_{ref})^T \mathbf{P} (\mathbf{x}_N - \mathbf{x}_{ref}) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_{nom}(\mathbf{x}_k, \mathbf{v}_k) + \mathbf{B}_d \mathbf{d}(\mathbf{x}_k, \mathbf{v}_k) \quad (\text{Εξίσωση 3.17}) \\ & \mathbf{v}_k \in \mathbb{U}, \quad k = 0, \dots, N-1 \\ & \mathbf{x}_0 = \mathbf{x}(0) \end{aligned} \quad (30)$$

όπου N είναι ο διακριτοποιημένος ορίζοντας πρόβλεψης ενώ οι υπόλοιπες μεταβλητές ορίζονται σύμφωνα με την Ενότητα 3.6.1. Το προαναφερθέν OCP επιλύεται και πάλι μέσα σε περίπου 4 ms χρησιμοποιώντας το ACADO Toolkit και τον επιλύτη qpOASES.

Όσον αφορά τη μη μοντελοποιημένη δυναμική, οι αποκλίσεις $\mathbf{d}_{v,0}$ και $\mathbf{d}_{\omega,0}$ προβλέπονται σε κάθε χρονική στιγμή t_0 χρησιμοποιώντας την Εξίσωση 3.24 για κάθε συνιστώσα των διανυσμάτων και την αντίστοιχη είσοδο \mathbf{z}_0 , που εξάγεται από την τρέχουσα κατάσταση \mathbf{x}_0 . Ένα σημαντικό θέμα προς συζήτηση αποτελεί η πρόβλεψη της μη μοντελοποιημένης δυναμικής σε όλο τον χρονικό ορίζοντα του NMPC. Πιθανές λύσεις είναι είτε η συνεχής προσέγγιση των \mathbf{d}_v και \mathbf{d}_ω σε ολόκληρο τον ορίζοντα είτε η διατήρηση των $\mathbf{d}_{v,0}$ και $\mathbf{d}_{\omega,0}$ σταθερών κατά τη διαδικασία της βελτιστοποίησης. Ωστόσο, η πρώτη λύση είναι υπολογιστικά δυσεπίλυτη ενώ η δεύτερη συνεπάγεται χαμηλότερη ακρίβεια. Προς αυτήν την κατεύθυνση, χρησιμοποιείται μια υπολογιστικά ελαφριά προσέγγιση δεύτερης τάξης γύρω από την τρέχουσα χρονική στιγμή t_0 :

$$\mu_k(\mathbf{z}_k) = \mu_0(\mathbf{z}_0) + \nabla \mu_0^T(\mathbf{z}_0) (\mathbf{z}_k - \mathbf{z}_0) + \frac{1}{2} (\mathbf{z}_k - \mathbf{z}_0)^T \mathbf{H}_0(\mathbf{z}_0) (\mathbf{z}_k - \mathbf{z}_0) \quad (31)$$

όπου η κλίση και η εσσιανή υπολογίζονται σύμφωνα με τις Εξισώσεις 3.28 και 3.29 αντίστοιχα. Συνεπώς, σε κάθε επανάληψη του NMPC, απαιτείται μόνο μία αξιολόγηση των επιμέρους GPs.

Πειραματικά Αποτελέσματα

Σε αυτήν την ενότητα, η απόδοση των προτεινόμενων μεθοδολογιών ελέγχου, δηλαδή (α) ο έλεγχος με βαθιά ενισχυτική μάθηση, (β) ο εύρωστος έλεγχος με βαθιά ενισχυτική μάθηση, (γ) το NMPC σχήμα διατυπωμένο για το ονομαστικό μοντέλο του συστήματος και (δ) το NMPC σχήμα που βασίζεται στις GPs, αξιολογούνται μέσω προσομοιώσεων και πραγματικών πειραμάτων σε υπαίθρια περιβάλλοντα. Κατά τη διάρκεια των πειραμάτων στον πραγματικό κόσμο, η ανατροφοδότηση της κατάστασης του σχοινοίου παρέχεται είτε από τη λύση που βασίζεται στην συμβατική κάμερα είτε από τη λύση που στηρίζεται στην DVS, που παρουσιάζονται στις Ενότητες 3.3.1 και 3.3.2 αντίστοιχα.

Πειραματική Διάταξη

Κατά τη διάρκεια των πειραμάτων σε εξωτερικούς χώρους, χρησιμοποιήθηκε το octotor όχημα της Εικόνας 2.3. Όλοι οι αλγόριθμοι, υλοποιημένοι είτε σε Python είτε σε C++, βασίζονται στο Robot Operating System (ROS) και τρέχουν στον ενσωματωμένο υπολογιστή Jetson AGX Xavier, ο οποίος ξεχωρίζει για την υψηλή του απόδοση και είναι κατάλληλος για εφαρμογές με drones όπου το μέγεθος, το βάρος και η κατανάλωση ρεύματος παίζουν σημαντικό ρόλο. Όπως αναφέρθηκε στις Ενότητες 2.2 και 2.3, ο ενσωματωμένος υπολογιστής συνδέεται με τον αυτόματο πιλότο χρησιμοποιώντας το πρωτόκολλο MAVLink, ενώ η επικοινωνία μεταξύ ROS και αυτόματου πιλότου επιτυγχάνεται μέσω του κόμβου MAVROS. Επιπλέον, ενσωματώνεται μια δυναμοκυψέλη 5 kg πάνω στο σχοινί, μαζί με τον ενισχυτή HX711, για τη μέτρηση της νόρμας της τάσης του σχοινοίου $\|\mathbf{T}\|$, καθώς είναι απαραίτητη για την εκτίμηση της κατάστασης του σχοινοίου μέσω της συμβατικής κάμερας και για τα σχήματα NMPC. Επιπλέον, ένα Arduino Mega [57] τοποθετείται στο UAV προκειμένου να μεταφέρει στον ενσωματωμένο υπολογιστή τις μετρήσεις που παρέχονται από την δυναμοκυψέλη μέσω σειριακής επικοινωνίας. Τέλος, το όχημα είναι εξοπλισμένο με τη κάμερα ZED 2 ή τη DAVIS 346, ανάλογα με το αν χρησιμοποιείται ο αλγόριθμος εκτίμησης βάσει RGB εικόνων ή events.

Πειράματα με Βαθιά Ενισχυτική Μάθηση

Πειράματα σε Προσομοιωτή

Αρχικά, παρουσιάζεται μια συγκριτική μελέτη στον προσομοιωτή Gazebo μεταξύ του default PID ελεγκτή θέσης του αυτόματου πιλότου (Ενότητα 2.4), ο οποίος δεν λαμβάνει υπόψη τις ταλαντώσεις του σχοινοίου, και της πολιτικής $p_\theta(s)$, που εκπαιδεύτηκε σύμφωνα με την Ενότητα 3.4. Και στις δύο περιπτώσεις, τα ίδια 8 waypoints, $\mathbf{p}_{ref}|_i = [x_{ref} \ y_{ref} \ z_{ref}]^T |_i \in \mathbb{R}^3$ φορ $i = 1, \dots, 8$, στάλθηκαν διαδοχικά ως θέσεις αναφοράς στους ελεγκτές. Προκειμένου να ποσοτικοποιηθεί η ταλαντωτική κίνηση του σχοινοίου κατά την πλοήγηση του οχήματος, υπολογίστηκε η απόκλιση του μοναδιαίου διανύσματος του σχοινοίου \mathbf{n} από την επιθυμητή κατακόρυφη θέση σύμφωνα με τη γωνία:

$$\beta = \cos^{-1}(|n_z|) \quad (32)$$

Όπως φαίνεται στην Εικόνα 3.11d, ο PID ελεγκτής διέγειρε την ταλαντωτική κίνηση του σχοινιού κατά τη διάρκεια του σεναρίου προσομοίωσης και, ως εκ τούτου, παρατηρήθηκαν αξιοσημείωτες τιμές της γωνίας β , οι οποίες στην περίπτωση ενός πραγματικού πειράματος θα έθεταν σε κίνδυνο την ασφάλεια τόσο του οχήματος όσο και του φορτίου. Αντίθετα, η εκπαιδευμένη πολιτική οδήγησε με επιτυχία το UAV προς τα τρισδιάστατα σημεία αναφοράς, όπως απεικονίζεται στις Εικόνες 3.11a, 3.11b και 3.11c, ενώ η γωνία β διατηρήθηκε σε σημαντικά χαμηλότερα επίπεδα (Εικόνα 3.11d).

Πειράματα σε Πραγματικό Περιβάλλον

Στη συνέχεια, πραγματοποιήθηκε μια συγκριτική πειραματική μελέτη μεταξύ του default PID ελεγκτή θέσης του αυτόματου πιλότου και του σχήματος ελέγχου της Εικόνας 3.12 με το octorotor προκειμένου να αξιολογηθεί η αποτελεσματικότητα της εκπαιδευμένης πολιτικής $\pi_{\theta}(s)$ σε πραγματικές συνθήκες. Σε αυτό το προκαταρκτικό πείραμα, χρησιμοποιήθηκε ένα κουτί 1.5 kg, αναρτημένο μέσω ενός σχοινιού μήκους 2.3 m, ενώ η ανίχνευση του φορτίου από την κάμερα ZED επιτεύχθηκε με τη βοήθεια τεσσάρων ArUco markers [23], τοποθετημένων στην πάνω επιφάνεια του κουτιού. Τα ίδια 8 waypoints στάλθηκαν διαδοχικά ως θέσεις αναφοράς στους δύο ελεγκτές. Επισημαίνεται ότι τα πειράματα πραγματοποιήθηκαν διαδοχικά, στην ίδια τοποθεσία και ώρα, και, ως εκ τούτου, οι περιβαλλοντικές συνθήκες, και ιδιαίτερα η ταχύτητα του ανέμου, θα μπορούσαν να θεωρηθούν πανομοιότυπες.

Αρχικά, χρησιμοποιήθηκε ο default PID ελεγκτής θέσης του Ardupilot, ο οποίος δεν αντισταθμίζει τις ταλαντώσεις του σχοινιού. Μόλις το όχημα κινήθηκε προς το δεύτερο waypoint, η γωνία β ξεπέρασε τις 60° και, ως εκ τούτου, το φορτίο δεν διατηρήθηκε μέσα στο οπτικό πεδίο της κάμερας, όπως φαίνεται στο σχετικό βίντεο. Για λόγους ασφαλείας, το πείραμα τερματίστηκε αφού το σχοινί πλησίασε τους βραχίονες του οχήματος και, συνεπώς, δεν ήταν εγγυημένη η ασφαλής ολοκλήρωση της αποστολής.

Αντίθετα, η πολιτική $\pi_{\theta}(s)$ οδηγεί αποτελεσματικά το όχημα προς τα τρισδιάστατα σημεία αναφοράς, όπως φαίνεται στην Εικόνα 3.13, ενώ ταυτόχρονα επιτυγχάνεται η ελαχιστοποίηση της ταλαντωτικής κίνησης του σχοινιού. Πιο συγκεκριμένα, η γωνία β δεν υπερβαίνει την τιμή των 15° κατά τη διάρκεια του πειράματος (Εικόνα 3.13d). Οι υψηλότερες τιμές της γωνίας παρατηρούνται όταν δίνεται εντολή για ένα νέο σημείο, μια λογική παρατήρηση δεδομένου ότι, αρχικά, το σφάλμα θέσης $\mathbf{p} - \mathbf{p}_{ref}$ είναι κυρίαρχο στο εκπαιδευμένο δίκτυο της συνάρτησης Q . Καθώς το όχημα πλησιάζει προς τη θέση αναφοράς, η ταλαντωτική κίνηση του σχοινιού, και συγκεκριμένα η γωνία β , ελαχιστοποιείται, όπως απεικονίζεται στην Εικόνα 3.13d.

Οι προαναφερθείσες συγκριτικές μελέτες καθώς και ένα πρόσθετο πραγματικό πείραμα παρουσιάζονται καλύτερα στον ακόλουθο σύνδεσμο <https://youtu.be/7C843hts83E>.

Πειράματα με Εύρωστη Βαθιά Ενισχυτική Μάθηση

Πειράματα σε Προσομοιωτή

Μετά τη σύγκλιση της διαδικασίας εκμάθησης στην Ενότητα 3.5, όπου συμπεριλήφθηκε η μέθοδος domain randomization, η εκπαιδευμένη πολιτική $\pi_{\theta}(s)$ δοκιμάστηκε εκτενώς στον προσομοιωτή Gazebo προκειμένου να επικυρωθεί η ευρωστία της ως προς τα διάφορες τιμές του μήκους σχοινιού l και της μάζας φορτίου m_L αλλά και τις εξωτερικές διαταραχές \mathbf{F}_Q . Πιο συγκεκριμένα, πραγματοποιήθηκαν συγκριτικές μελέτες λαμβάνοντας υπόψη την αποστολή της Εικόνας 3.14, η οποία αποτελείται από 8 θέσεις αναφοράς \mathbf{p}_{ref} . Η ικανότητα της πολιτικής να ελαχιστοποιεί την ταλαντωτική κίνηση του σχοινιού μετρήθηκε με βάση την νόρμα της ταχύτητας του σχοινιού $\|\dot{\mathbf{n}}\|$ και τη γωνία β , που ορίζεται από την Εξίσωση 3.32. Εξετάστηκαν τα ακόλουθα σενάρια:

1) *Ευρωστία ως προς τη μάζα του φορτίου:* Σε αυτή τη συγκριτική μελέτη, το μήκος του σχοινιού διατηρήθηκε σταθερό και ίσο με 1.0m ενώ λήφθηκαν υπόψη 3 διαφορετικές μάζες φορτίου, συγκεκριμένα $m_L = 0.25, 0.5, \text{ και } 1.0\text{kg}$ κατά τη διάρκεια κάθε αποστολής. Η χρονική εξέλιξη της γωνίας β σε κάθε περίπτωση παρουσιάζεται στην Εικόνα 3.15, ενώ η μέση τιμή της ταχύτητας $\|\dot{\mathbf{n}}\|$ και της γωνίας β για κάθε μεμονωμένη αποστολή απεικονίζονται στον Πίνακα 3.6.

2) *Ευρωστία ως προς το μήκος του σχοινιού:* Αντίθετα, στο ακόλουθο σενάριο, η μάζα του φορτίου διατηρήθηκε σταθερή στα 0.5kg ενώ το μήκος του καλωδίου ορίστηκε στα $l = 0.5, 1.5, \text{ και } 2.0\text{m}$. Η ταλαντωτική κίνηση του σχοινιού κατά τη διάρκεια κάθε αποστολής χαρακτηρίζεται από τις μέσες τιμές του Πίνακα 3.7.

3) *Ευρωστία ως προς τις εξωτερικές διαταραχές:* Στις προηγούμενες μελέτες, το quadrotor επιχειρούσε στο περιβάλλον του Gazebo χωρίς την ύπαρξη εξωτερικών διαταραχών. Για το σκοπό αυτό, το τελευταίο σενάριο περιλάμβανε τις default τιμές για το μήκος του σχοινιού και τη μάζα φορτίου, δηλαδή 1.0m και 0.5kg αντίστοιχα, και τυχαίες εξωτερικές δυνάμεις \mathbf{F}_Q , με κάθε συνιστώσα του διανύσματος να δειγματοληπτείται ομοιόμορφα από το εύρος $[-1.0, 1.0]$ σε κάθε χρονικό βήμα. Οι μέσες τιμές για την ταχύτητα $\|\dot{\mathbf{n}}\|$ και τη γωνία β , σύμφωνα με 10 πειράματα προσομοίωσης, ήταν ίσες με 0.086s^{-1} και 2.574° αντίστοιχα.

Τα παραπάνω αριθμητικά αποτελέσματα αποδεικνύουν ότι η απόδοση της εκπαιδευμένης πολιτικής είναι τόσο αποδεκτή όσο και συνεπής στις διάφορες τιμές των ιδιοτήτων του συστήματος χάρη στη μέθοδο domain randomization.

Πειράματα σε Πραγματικό Περιβάλλον

Μετά την ανάλυση στον προσομοιωτή Gazebo, το σχήμα ελέγχου της Εικόνας 3.16 εφαρμόστηκε απευθείας στο octorotor της Εικόνας 2.3 προκειμένου να αξιολογηθεί η επίδοση της πολιτικής, που εκπαιδεύτηκε μέσω της μεθόδου domain randomization, σε ένα πραγματικό υπαίθριο περιβάλλον. Το όχημα ήταν εξοπλισμένο με την κάμερα DAVIS 346 με σκοπό την εκτίμηση της κατάστασης του σχοινιού σύμφωνα με την Ενότητα 3.3.2.

Ένα καλώδιο με μήκος $l = 3.8\text{m}$, το οποίο βρίσκεται εκτός του εύρους των τιμών που χρησιμοποιήθηκαν κατά την εκπαίδευση, ενσωματώθηκε στο UAV ενώ διάφορα φορτία τοποθετήθηκαν μέσα σε μια τσάντα, που βρίσκεται στο κάτω άκρο του σχοινιού.

Πιο συγκεκριμένα, μια συγκριτική μελέτη διεξήχθη μεταξύ 3 διαφορετικών μαζών φορτίου, συγκεκριμένα $m_L = 0.3, 0.5, \text{ και } 0.9\text{kg}$. Εξετάστηκε η ίδια αποστολή αποτελούμενη από 4 θέσεις αναφοράς, όπως φαίνεται στην Εικόνα 3.17. Παρά τις εξωτερικές διαταραχές που δρουν στην πλατφόρμα και τον θόρυβο στην εκτίμηση της κατάστασης του σχοινιού, η εκπαιδευμένη πολιτική ελαχιστοποίησε τόσο το σφάλμα θέσης, π.χ. Εικόνα 3.17, όσο και τη γωνιακή κίνηση του σχοινιού, παρουσιάζοντας μάλιστα παρόμοια επίπεδα απόδοσης για τις 3 διαφορετικές μάζες, όπως απεικονίζεται στην Εικόνα 3.18, όπου παρουσιάζεται η γωνία β για κάθε περίπτωση. Κατά συνέπεια, η πολιτική μπόρεσε να γενικευτεί με επιτυχία σε ένα πραγματικό περιβάλλον χάρη στη μέθοδο domain randomization που εφαρμόστηκε κατά την εκπαίδευση. Τα προαναφερθέντα πειράματα με το octorotor φαίνονται καλύτερα στο παρακάτω βίντεο <https://youtu.be/kbPSdWZiRAo>.

Πειράματα NMPC με βάση το Ονομαστικό Μοντέλο

Πειράματα NMPC βάσει Ονομαστικού Μοντέλου και Συμβατική Κάμερα

Αρχικά, διερευνήθηκε η ικανότητα του σχήματος NMPC, που βασίζεται στο ονομαστικό μοντέλο του συστήματος όπως παρουσιάζεται στην Ενότητα 3.6.1, να οδηγεί αποτελεσματικά το πολυκόπτερο σε τρισδιάστατα σημεία αναφοράς, χωρίς να διεγείρει την ταλαντωτική κίνηση του αναρτημένου μέσω σχοινιού φορτίου. Όσον αφορά το φορτίο, χρησιμοποιήθηκε ένας μηχανισμός δειγματοληψίας, που αργότερα θα αξιοποιηθεί για τη διεξαγωγή δειγματοληψίας σε υδάτινα περιβάλλοντα. Η μάζα του μηχανισμού είναι ίση με $m_L = 0.25\text{ kg}$, ενώ αναρτήθηκε μέσω ενός σχοινιού μήκους $l = 3.8\text{ m}$. Επιπλέον, η κάμερα ZED 2 ενσωματώθηκε στο όχημα και, ως εκ τούτου, χρησιμοποιήθηκε ο αλγόριθμος της Ενότητας 3.3.1 για την ανατροφοδότηση της κατάστασης του σχοινιού, όπως απεικονίζεται στην Εικόνα 3.19. Προκειμένου να αξιολογηθεί η απόδοση τόσο του NMPC όσο και του CNN, πραγματοποιήθηκαν 2 πειράματα σε διαφορετικές τοποθεσίες, συγκεκριμένα στην πανεπιστημιούπολη του ΕΜΠ και σε μια παραλία.

Και στις δύο περιπτώσεις, εξετάστηκε μια αποστολή αποτελούμενη από πολλαπλά σημεία. Ο NMPC ελαχιστοποίησε επιτυχώς το σφάλμα $\mathbf{p}_{ref} - \mathbf{p}$ μεταξύ των θέσεων αναφοράς και της πραγματικής θέσης του UAV (Εικόνες 3.20, 3.22), ενώ ταυτόχρονα οι γωνίες η_L και η γωνιακή ταχύτητα ω_L του σχοινιού, που εκτιμήθηκαν από το CNN και το EKF, διατηρήθηκαν μακριά από επικίνδυνες τιμές (Εικόνες 3.21, 3.23). Έτσι, επιτεύχθηκε η ασφαλής μεταφορά του οχήματος και του φορτίου και ολοκληρώθηκαν με επιτυχία και οι δύο αποστολές, όπως φαίνεται στις Εικόνες 3.20d, 3.22d. Τονίζεται επίσης ότι το εκπαιδευμένο CNN ανιχνεύει με επιτυχία το σχοινί και επιδεικνύει ευρωστία στα διαφορετικά περιβάλλοντα. Επιπρόσθετα, αναφέρεται ότι και στις δύο αποστολές το όχημα κινήθηκε με ταχύτητες έως και 4 m/s .

Τα δύο πειράματα απεικονίζονται καλύτερα στα παρακάτω βίντεο <https://youtu.be/G-7rAuAFxHM> και <https://youtu.be/Km6nJYrVGoo>.

Πειράματα NMPC βάσει Ονομαστικού Μοντέλου και DVS

Σε αυτή την ενότητα, ενσωματώθηκε η κάμερα DAVIS 346 στο octorotor με φορτίο $m_L = 0.5\text{kg}$ και σχοινί $l = 3.8\text{m}$, όπως απεικονίζεται στην Εικόνα 3.3, προκειμένου να διερευνηθεί η απόδοση της μεθόδου εκτίμησης της κατάστασης του σχοινοῦ βάσει events (Ενότητα 3.3.2) έναντι δύσκολων συνθηκών, που συνήθως συναντώνται σε υπαίθρια περιβάλλοντα. Παρά τον αυξημένο αριθμό events, που δημιουργούνται από την κίνηση του οχήματος και αντιστοιχούν στον περιβάλλοντα χώρο, η προτεινόμενη μέθοδος ανίχνευσε αξιόπιστα τα events που σχετίζονταν με την ταλαντωτική κίνηση του σχοινοῦ και παρείχε εύρωστη ανατροφοδότηση της συνολικής κατάστασής του στο NMPC σχήμα.

Με δεδομένο ένα σύνολο 3D θέσεων αναφοράς \mathbf{p}_{ref} και την κατάσταση του σχοινοῦ, δηλαδή $\boldsymbol{\eta}_L$ και $\boldsymbol{\omega}_L$, που εκτιμάται από την επεξεργασία της εισερχόμενης ροής events, το σχήμα NMPC, διατυπωμένο για το ονομαστικό μοντέλο του συστήματος, υπολόγισε κατάλληλες εισόδους για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου προκειμένου να ελαχιστοποιηθεί το σφάλμα $\mathbf{p}_{ref} - \mathbf{p}$ και, έτσι, το όχημα εκπλήρωσε την αποστολή, όπως απεικονίζεται στην Εικόνα 3.24a. Επιπλέον, καθ' όλη τη διάρκεια του πειράματος, η ταλαντωτική κίνηση του σχοινοῦ, που εκτιμήθηκε μέσω της DVS, διατηρήθηκε κάτω από κρίσιμες τιμές (Εικόνα 3.24), καθώς το σχοινί δεν πλησίασε το UAV και, επομένως, διασφαλίστηκε η ασφάλεια του συστήματος.

Προκειμένου να διερευνηθεί περαιτέρω η αξιοπιστία του προτεινόμενου αλγορίθμου, διεξήχθη ένα δεύτερο πείραμα όπου εξετάστηκε μια αποστολή με περισσότερες θέσεις αναφοράς \mathbf{p}_{ref} . Ομοίως, επιτεύχθηκαν τόσο η εύρωστη αναγνώριση του σχοινοῦ κατά τη διάρκεια της αποστολής όσο και η μεταφορά του αναρτημένου φορτίου με ελάχιστη ταλαντωτική κίνηση του σχοινοῦ. Και τα δύο πειράματα παρουσιάζονται καλύτερα στο παρακάτω βίντεο <https://youtu.be/jFUQcZ00hoM>.

Στη συνέχεια, προκειμένου να αποδείξουμε τα πλεονεκτήματα που απορρέουν από την χρήση DVS κατά την εναέρια μεταφορά φορτίων, πραγματοποιήθηκε μία συγκριτική μελέτη μεταξύ της μεθόδου που βασίζεται στα events και της μεθόδου που στηρίζεται στη συμβατική κάμερα. Κατά τη συγκριτική μελέτη, θεωρήθηκε η ίδια αποστολή, αποτελούμενη από 4 θέσεις αναφοράς, και η ανατροφοδότηση της κατάστασης του σχοινοῦ, που εκτιμήθηκε από τις δύο μεθόδους, τροφοδοτήθηκε στο σχήμα NMPC της Ενότητας 3.6.1. Επισημαίνεται ότι οι παράμετροι του ελεγκτή, δηλαδή οι πίνακες κόστους και οι περιορισμοί στην είσοδο, ήταν πανομοιότυπες. Επιπλέον, τα πειράματα πραγματοποιήθηκαν διαδοχικά προκειμένου να διασφαλιστούν παρόμοιες περιβαλλοντικές συνθήκες και, ως εκ τούτου, μια δίκαιη σύγκριση.

Οι δύο μέθοδοι συγκρίθηκαν με βάση την αποτελεσματικότητα του ελεγκτή και, πιο συγκεκριμένα, την ικανότητά του να ελαχιστοποιεί την νόρμα της γωνίας του σχοινοῦ

$\|\eta_L\|$. Όπως φαίνεται στην Εικόνα 3.25, η γρήγορη και αξιόπιστη ανατροφοδότηση που παρέχεται από την DVS είχε ως αποτέλεσμα την καλύτερη επίδοση του ελεγκτή, ενώ αντιθέτως παρατηρήθηκαν μεγαλύτερες γωνίες όταν χρησιμοποιήθηκε η συμβατική κάμερα. Πιο συγκεκριμένα, η μέγιστη γωνία του σχοινιού $\|\eta_L\|$ για τη μέθοδο που βασίζεται στα events και για τη μέθοδο που στηρίζεται στις RGB εικόνες ήταν ίση με 14.13° και 21.95° αντίστοιχα. Επομένως, κατά την εναέρια μεταφορά αναρτημένων φορτίων, η γρήγορη ανατροφοδότηση μπορεί να επηρεάσει σημαντικά την επίδοση του ελεγκτή και, επομένως, η χρήση μιας κάμερας DVS μπορεί να βελτιώσει περαιτέρω την αποτελεσματικότητα των υπαρχόντων σχημάτων ελέγχου.

Πειράματα NMPC με GPs

Σε αυτήν την ενότητα, παρουσιάζουμε τα πλεονεκτήματα της χρήσης παλινδρόμησης με weighted και sparse GPs σύμφωνα με την Ενότητα 3.6.3, όταν το UAV με το αναρτημένο μέσω σχοινιού φορτίο επιχειρεί υπό την παρουσία σημαντικών εξωτερικών διαταραχών και, ως εκ τούτου, υπάρχουν μη αμελητέες αποκλίσεις μεταξύ της ονομαστικής και της πραγματικής δυναμικής του συστήματος.

Πειράματα σε Προσομοιωτή

Πριν από την εφαρμογή του σχήματος NMPC που βασίζεται στις GPs (Ενότητα 3.6.4) στο πραγματικό octotoror, πραγματοποιήθηκαν πειράματα στο περιβάλλον προσομοίωσης Gazebo για την αξιολόγηση της μεθόδου υπό ελεγχόμενες συνθήκες. Προκειμένου να επικυρωθεί η βελτίωση της επίδοσης του σχήματος ελέγχου με την προσθήκη GPs, πραγματοποιήθηκε μια συγκριτική μελέτη μεταξύ των ακόλουθων σεναρίων: (α) NMPC με βάση το ονομαστικό μοντέλο (Ενότητα 3.6.1), (β) NMPC με sparse GPs, (γ) NMPC με weighted και sparse GPs. Η ίδια αποστολή, αποτελούμενη από 10 διαφορετικά σημεία $\mathbf{p}_{ref} \in \mathbb{R}^3$, εξετάστηκε σε όλες τις περιπτώσεις.

Κατά τη συγκριτική μελέτη, οι ακόλουθες εξωτερικές δυνάμεις ${}^B\mathbf{F}_Q = [-1.2, 1.8, 0]^T - 0.3 \cdot {}^B\mathbf{v} |{}^B\mathbf{v}|$ και ${}^B\mathbf{F}_L = -0.13 \cdot {}^B\mathbf{v}_L |{}^B\mathbf{v}_L|$ ασκήθηκαν στο quadrotor και στο φορτίο αντίστοιχα ως προς το σύστημα συντεταγμένων \mathbf{B} . Αναφέρεται ότι κατά τα σενάρια (β) και (γ), όπου ενσωματώνονται οι GPs, το σετ δεδομένων ήταν αρχικά γεμάτο με μετρήσεις που συλλέχθηκαν κατά τη διάρκεια ενός προηγούμενου πειράματος προσομοίωσης χωρίς την ύπαρξη των προαναφερθέντων εξωτερικών δυνάμεων. Όπως φαίνεται στην Εικόνα 3.26, όπου απεικονίζονται το σφάλμα θέσης και η συνάρτηση κόστους του NMPC, τα μοντέλα GPs βελτίωσαν σημαντικά τη συμπεριφορά του συστήματος υπό την επίδραση των εξωτερικών δυνάμεων και την απόδοση του σχήματος ελέγχου σε σύγκριση με την ονομαστική περίπτωση, δηλαδή το σενάριο (α). Επιπλέον, στο σενάριο (γ), η ενσωμάτωση βαρών, που δίνουν προτεραιότητα στις πιο πρόσφατες μετρήσεις, επιτάχυνε την προσαρμογή του προβλήματος παλινδρόμησης στο νέο περιβάλλον και, ως εκ τούτου, είχε ως αποτέλεσμα την ταχύτερη μείωση του σφάλματος θέσης και της συνάρτησης κόστους σε σύγκριση με το σενάριο (β). Τέλος, πρέπει να σημειωθεί ότι,

κατά τα σενάρια (β) και (γ), το κόστος του NMPC δεν συγκλίνει στο μηδέν αφού το όχημα σταθεροποιείται στις θέσεις αναφοράς με μη μηδενικές γωνίες roll και pitch.

Πειράματα σε Πραγματικό Περιβάλλον

Μετά την επικύρωση στον προσομοιωτή Gazebo, πραγματοποιήθηκαν πειράματα με το πραγματικό octocopter, συμπεριλαμβανομένου ενός φορτίου $m_L = 0.5\text{kg}$ που αναρτήθηκε μέσω ενός σχοινιού $l = 3.8\text{m}$ και της κάμερας DVS, σε ένα εξωτερικό περιβάλλον, προκειμένου να ελεγχθεί η αποτελεσματικότητα του σχήματος NMPC με GPs της Εικόνας 3.27 έναντι δύσκολων συνθηκών και αγνώστων εξωτερικών διαταραχών. Ομοίως, διεξήχθη μια συγκριτική μελέτη μεταξύ των διαφορετικών σχημάτων NMPC, δηλαδή, συμπεριλαμβανομένου (α) του ονομαστικού μοντέλου, (β) των sparse GPs και (γ) των weighted και sparse GPs. Θα πρέπει να σημειωθεί ότι τα πειράματα πραγματοποιήθηκαν διαδοχικά, στην ίδια τοποθεσία, και, επομένως, οι εξωτερικές συνθήκες, ιδιαίτερα η ταχύτητα και η κατεύθυνση του αέρα, μπορούν να θεωρηθούν πανομοιότυπες παρά την αναπόφευκτη ύπαρξη τύρβης. Τέλος, οι ίδιες 7 θέσεις αναφοράς \mathbf{p}_{ref} δόθηκαν ως είσοδοι στους ελεγκτές, ώστε να εξασφαλιστεί μια δίκαιη σύγκριση.

Και στα δύο σενάρια (β) και (γ), το σετ δεδομένων αρχικά απαρτιζόταν από μετρήσεις που συγκεντρώθηκαν κατά τη διάρκεια μιας προηγούμενης ημέρας με εντελώς διαφορετικές περιβαλλοντικές συνθήκες. Όπως απεικονίζεται στην Εικόνα 3.28, ένα σημαντικό σφάλμα μόνιμης κατάστασης στη θέση και ένα αυξημένο κόστος παρατηρήθηκαν όταν εφαρμόστηκε ο NMPC διατυπωμένος για το ονομαστικό μοντέλο, δηλαδή το σενάριο (α). Η προαναφερθείσα συμπεριφορά προήλθε κυρίως από τη μέση ταχύτητα ανέμου, περίπου 4 μποφόρ, που επικρατούσε στην περιοχή σύμφωνα με τα μετεωρολογικά δεδομένα, και είχε ως αποτέλεσμα αποκλίσεις μεταξύ της ονομαστικής και της πραγματικής δυναμικής του συστήματος. Αντίθετα, η ενσωμάτωση μοντέλων GPs στα σενάρια (β) και (γ) οδήγησε σε σημαντικά μειωμένο σφάλμα θέσης και κόστος, γεγονός που αποδεικνύει την ικανότητα των GPs να προσεγγίζουν τις αποκλίσεις στο μοντέλο και τις εξωτερικές διαταραχές. Επιπλέον, η weighted εκδοχή της παλινδρόμησης με sparse GPs κατά το σενάριο (γ) τόνισε τη σημασία των πιο πρόσφατων μετρήσεων και, ως εκ τούτου, επιτεύχθηκε ταχύτερη βελτίωση στη συμπεριφορά του συστήματος και γενικά καλύτερη επίδοση του σχήματος ελέγχου σε σύγκριση με το σενάριο (β). Τα προαναφερθέντα πειράματα απεικονίζονται καλύτερα στο παρακάτω βίντεο <https://youtu.be/4bm3JLo0d5U>.

Παρακολούθηση και Εναπόθεση ενός Αναρτημένου Φορτίου σε Επίγειο Στόχο

Εισαγωγή

Σε αυτό το κεφάλαιο, εξετάζουμε την παρακολούθηση ενός επίγειου οχήματος χρησιμοποιώντας ένα πολυκόπτερο με αναρτημένο μέσω σχοινιού φορτίο και μια κάμερα με προσανατολισμό προς τα κάτω, με απώτερο σκοπό την επιτυχή εναπόθεση του φορτίου στον στόχο. Η εναέρια παρακολούθηση στόχων με βάση οπτική ανατροφοδότηση είναι ένα καλά μελετημένο πεδίο στη βιβλιογραφία, π.χ. [9], [10], [29] και [36]· χωρίς ωστόσο να λαμβάνεται υπόψη η ύπαρξη αναρτημένου μέσω σχοινιού φορτίου και, ως εκ τούτου, η εναπόθεσή του στον στόχο.

Για το σκοπό αυτό, αναπτύσσεται ένα σχήμα Μη Γραμμικού Προβλεπτικού Ελέγχου (NMPC), διατυπωμένο για τη δυναμική του συστήματος, προκειμένου να επιτευχθεί η παρακολούθηση του αυθαίρετα κινούμενου στόχου. Δεδομένου ότι ο NMPC βασίζεται στη μελλοντική συμπεριφορά του συστήματος, απαιτείται η τροχιά του επίγειου στόχου σε όλο τον χρονικό ορίζοντα του NMPC. Προς αυτή την κατεύθυνση, ένα Συνελικτικό Νευρωνικό Δίκτυο (CNN), συνοδευόμενο από έναν Kernelized Correlation Filter (KCF) tracker, αξιολογείται για την αδιάλειπτη ανίχνευση του στόχου και, στη συνέχεια, οι προαναφερθείσες μετρήσεις ενσωματώνονται σε ένα πρόβλημα παλινδρόμησης Bézier προκειμένου να προβλεφθεί η μελλοντική του τροχιά [29]. Τελικά, το φορτίο απελευθερώνεται προς τον επίγειο στόχο, ανοίγοντας μια αρπάγη, μόλις ικανοποιηθεί μια συνθήκη που συσχετίζει την προβλεπόμενη βαλλιστική τροχιά του φορτίου με τη μελλοντική κίνηση του στόχου. Σημειώνεται ότι η ταλαντωτική κίνηση του σχοινιού κατά την παρακολούθηση του στόχου καθιστά την επιτυχή εναπόθεση του φορτίου αρκετά δύσκολη ακόμη και για έναν έμπειρο χειριστή.

Ορισμός του Προβλήματος

Σε αυτήν την ενότητα, διατυπώνουμε το πρόβλημα ελέγχου κίνησης της παρακολούθησης ενός επίγειου στόχου χρησιμοποιώντας ένα πολυκόπτερο με αναρτημένο μέσω σχοινιού φορτίο, όπως φαίνεται στην Εικόνα 4.1. Πιο συγκεκριμένα, εξετάζουμε την περίπτωση κατά την οποία ένα μη επανδρωμένο επίγειο όχημα (UGV) κινείται αυθαίρετα στο έδαφος

χωρίς να μοιράζεται πληροφορίες σχετικά με την κατάσταση του, δηλαδή τη θέση του \mathbf{p}_T και την ταχύτητά του \mathbf{v}_T σε σχέση με το σύστημα αναφοράς \mathbf{W} , και, ως εκ τούτου, το UAV βασίζεται μόνο στις οπτικές πληροφορίες, που λαμβάνονται από την κάμερα, ώστε να παρακολουθεί συνεχώς τον στόχο και τελικά να εναποθέσει το αναρτημένο φορτίο σε αυτόν.

Προς αυτήν την κατεύθυνση, αναπτύσσεται το σχήμα ελέγχου, που απεικονίζεται στην Εικόνα 4.2, προκειμένου να εκπληρωθεί η ζητούμενη εφαρμογή. Η προαναφερθείσα στρατηγική ελέγχου αποτελείται από τα ακόλουθα μέρη:

1. Ανίχνευση του σχοινοῦ βάσει CNN και εκτίμηση της συνολικής κατάστασης του σχοινοῦ, δηλαδή η_L και ω_L , χρησιμοποιώντας ένα φίλτρο Kalman που αξιοποιεί την έξοδο του CNN και τις μετρήσεις που παρέχονται από δυναμοκυψέλη, σύμφωνα με την Ενότητα 3.3.1,
2. Ανίχνευση του στόχου χρησιμοποιώντας έναν KCF tracker και ένα CNN, το οποίο υποδεικνύει μια αρχική περιοχή ενδιαφέροντος (ROI) για τον KCF tracker, και πρόβλεψη της μελλοντικής κίνησης του στόχου, δηλαδή της θέσης \mathbf{p}_T και της ταχύτητας \mathbf{v}_T σε σχέση με το σύστημα αναφοράς \mathbf{W} , χρησιμοποιώντας καμπύλες Bézier,
3. Ανάπτυξη ενός NMPC, διατυπωμένου για τη μη γραμμική δυναμική του συστήματος (Εξίσωση 2.11), ο οποίος ενσωματώνει τη μελλοντική τροχιά του στόχου στον χρονικό ορίζοντα του NMPC και παράγει κατάλληλες εντολές για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου που ελαχιστοποιούν το σφάλμα παρακολούθησης μεταξύ του στόχου και του φορτίου και
4. Έλεγχος μιας συνθήκης η οποία, βάσει της προβλεπόμενης τροχιάς του στόχου και του φορτίου, ενεργοποιεί το άνοιγμα μιας αρπάγης και την απελευθέρωση του φορτίου προς τον στόχο.

Τα παραπάνω μεμονωμένα μέρη, εκτός από την εκτίμηση της κατάστασης του σχοινοῦ μέσω κάμερας που παρουσιάστηκε προηγουμένως (Ενότητα 3.3.1), αναλύονται σχολαστικά στις ακόλουθες ενότητες.

Εκτίμηση της Κατάστασης του Στόχου

Ανίχνευση του Στόχου

Εκτός από την κατάσταση του σχοινοῦ, η εκτίμηση της κατάστασης του στόχου είναι απαραίτητη προϋπόθεση για την επίτευξη της παρακολούθησής του και την επιτυχή εναπόθεση του φορτίου σε αυτόν. Με ανάλογο τρόπο με την Ενότητα 3.3.1, ένα CNN εκπαιδεύεται για να ανιχνευθεί ο στόχος που κινείται προς άγνωστες κατευθύνσεις.

Ωστόσο, όταν το φορτίο βρίσκεται πάνω από τον στόχο, ένα μέρος της περιοχής της εικόνας, που αντιστοιχεί στον στόχο, καλύπτεται από το φορτίο και, ως αποτέλεσμα, το CNN αποτυγχάνει να ανιχνεύσει τον στόχο συνεχόμενα. Συνεπώς, ένας tracker

αξιοποιείται επιπρόσθετα έτσι ώστε να επιτευχθεί η εύρωστη αναγνώριση του στόχου παρά τα οπτικά εμπόδια, ενώ παράλληλα διασφαλίζεται ταχύτερος ρυθμός ανίχνευσης. Μεταξύ των διαφόρων tracker που παρουσιάζονται στη βιβλιογραφία, επιλέγεται ο KCF tracker [30] λόγω της υψηλής ταχύτητας και ακρίβειάς του και της ικανότητάς του να παρακολουθεί στόχους παρουσία μερικής οπτικής παρεμπόδισης.

Η διαδικασία της ανίχνευσης είναι ως εξής:

1. Εξαγωγή ROI χρησιμοποιώντας το εκπαιδευμένο CNN,
2. Με βάση την προαναφερθείσα περιοχή ενδιαφέροντος (ROI), χρήση ενός KCF tracker και
3. Ανίχνευση ξανά του στόχου χρησιμοποιώντας το CNN μετά από μερικές επαναλήψεις, προκειμένου να μειωθεί το συσσωρευμένο σφάλμα ή όταν αποτύχει ο tracker.

Με βάση την προαναφερθείσα διαδικασία, εξάγεται μια περιοχή ενδιαφέροντος (ROI), που αντιστοιχεί στον στόχο, με τη μορφή ενός ορθογωνίου, όπως απεικονίζεται στην Εικόνα 4.3. Το κέντρο (u_T, v_T) του ορθογωνίου ορίζει τις συντεταγμένες pixel του στόχου στο επίπεδο της εικόνας. Εφόσον ο στόχος κινείται στο έδαφος, το γνωστό του ύψος $z_T = h$ χρησιμοποιείται για να υπολογιστεί η τρισδιάστατη θέση του. Συνεπώς, μετά από κατάλληλους μετασχηματισμούς μεταξύ του συστήματος της κάμερας και του συστήματος αναφοράς, λαμβάνεται μια εκτίμηση της θέσης του στόχου $\mathbf{p}_T = [x_T \ y_T \ z_T]^T$ σε κάθε χρονική στιγμή.

Πρόβλεψη της Κίνησης του Στόχου

Η εκτίμηση της θέσης του στόχου δεν είναι επαρκής για το αναπτυσσόμενο σχήμα ελέγχου, καθώς ο NMPC απαιτεί τόσο τη θέση όσο και την ταχύτητα του στόχου όχι μόνο την τρέχουσα χρονική στιγμή αλλά και σε όλο τον μελλοντικό χρονικό ορίζοντα. Για το σκοπό αυτό, αξιοποιούνται οι καμπύλες Bézier, όπως προτείνεται στο [29], παρόμοια με την Ενότητα 3.3.2. Ωστόσο, δεδομένου ότι σκοπός είναι η πρόβλεψη της μελλοντικής τροχιάς του στόχου, το πρόβλημα παλινδρόμησης επαναδιατυπώνεται σε σύγκριση με την Ενότητα 3.3.2, όπου η καμπύλη Bézier εξομάλυνσε τις μετρήσεις μέχρι την τρέχουσα χρονική στιγμή.

Πιο συγκεκριμένα, όπως αναφέρεται στην Ενότητα 3.3.2, μια καμπύλη Bézier ορίζεται, με βάση το βαθμό n της καμπύλης, ως εξής:

$$\mathbf{B}(\tau) = \sum_{i=0}^n b_{i,n}(\tau) \mathbf{c}_i \quad (33)$$

όπου $\tau \in [0, 1]$, $\mathbf{B}(\tau) \in \mathbb{R}^2$ (αφού ο στόχος κινείται στο $x - y$ επίπεδο) είναι η καμπύλη Bézier, $b_{i,n}(\tau)$, $i = 0, \dots, n$ είναι τα πολυώνυμα βάσης Bernstein βαθμού n και $\mathbf{c}_i = [c_{x,i} \ c_{y,i}]^T \in \mathbb{R}^2$ είναι τα $n + 1$ σημεία ελέγχου.

Προκειμένου να εκτιμηθεί και να προβλεφθεί η κίνηση του στόχου, είναι απαραίτητο να βρεθεί το κατάλληλο σύνολο σημείων ελέγχου $\mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n]$. Προς αυτήν την κατεύθυνση, υποθέτοντας ότι N εκτιμήσεις της θέσης του στόχου \mathbf{p}_T είναι διαθέσιμες σύμφωνα με την Ενότητα 4.3.1, μαζί με τις αντίστοιχες χρονικές στιγμές t , ένα σετ δεδομένων $D = \{(t_1, x_{T,1}, y_{T,1}), \dots, (t_N, x_{T,N}, y_{T,N})\}$ σταθερού μεγέθους δημιουργείται, όπου η χρονική στιγμή t_N αντιστοιχεί στην τρέχουσα στιγμή. Το προαναφερθέν σετ δεδομένων D επικαιροποιείται μόλις είναι διαθέσιμη μια νέα μέτρηση. Δεδομένου ότι ο σκοπός είναι να προβλέψουμε την κίνηση του στόχου μέχρι τη μελλοντική χρονική στιγμή t_p , η ακόλουθη αντιστοίχιση πραγματοποιείται μεταξύ των χρονικού διαστήματος $t \in [t_1, t_p]$ και της παραμέτρου $\tau \in [0, 1]$ της καμπύλης Bézier:

$$\tau = \frac{t - t_1}{t_p - t_1} \quad (34)$$

Το βέλτιστο σύνολο σημείων ελέγχου υπολογίζεται σύμφωνα με το ακόλουθο πρόβλημα βελτιστοποίησης:

$$\min_{\mathbf{c}} \sum_{i=1}^N (w_{t_i} \|\mathbf{B}(\tau_i) - \mathbf{p}_T(t_i)\|_2^2) \quad (35)$$

όπου ο στόχος είναι να ελαχιστοποιηθεί η απόσταση μεταξύ των μετρήσεων και της καμπύλης Bézier με w_{t_i} να δηλώνουν βάρη που επιβάλλουν κυρώσεις στις παλαιότερες μετρήσεις [29].

Επιπλέον, για τον εντοπισμό μόνο εφικτών λύσεων, επιβάλλονται περιορισμοί στο παραπάνω πρόβλημα βελτιστοποίησης. Οι περιορισμοί αυτοί πηγάζουν από τα δυναμικά όρια του στόχου, σύμφωνα με τα οποία η προβλεπόμενη ταχύτητα και επιτάχυνσή του θα πρέπει να βρίσκονται στο εσωτερικό των $[-u_{max}, u_{max}]$ και $[-a_{max}, a_{max}]$ αντίστοιχα. Παραγωγίζοντας την καμπύλη Bézier, ορίζονται οι ακόλουθοι περιορισμοί για τον άξονα x (ομοίως για τη διάσταση y):

$$\begin{aligned} -u_{max} &\leq n \cdot (c_{x,i} - c_{x,i-1}) / (t_p - t_1) \leq u_{max} \\ -a_{max} &\leq n \cdot (n-1) \cdot (c_{x,i} - 2c_{x,i-1} + c_{x,i-2}) / (t_p - t_1)^2 \leq a_{max} \end{aligned} \quad (36)$$

Το παραπάνω πρόβλημα Τετραγωνικού Προγραμματισμού με περιορισμούς επιλύεται χρησιμοποιώντας το λογισμικό QOQP [24], παρόμοια με την Ενότητα 3.3.2 και, τελικά, μια εκτίμηση της μελλοντικής τροχιάς του στόχου, δηλαδή $\mathbf{p}_T(t)$ και $\mathbf{v}_T(t)$, είναι διαθέσιμη σε όλο τον μελλοντικό ορίζοντα $[t_N, t_p]$.

Μη Γραμμικός Προβλεπτικός Έλεγχος για Παρακολούθηση Στόχου

Με σκοπό την αποτελεσματική παρακολούθηση του επίγειου στόχου, ένας NMPC διατυπώνεται για τη μη γραμμική δυναμική του συστήματος που ορίζεται από την Εξίσωση 2.11. Δεδομένου ότι το σχήμα NMPC επιδιώκει την παρακολούθηση του επίγειου στόχου και τελικά την εναπόθεση του αναρτημένου φορτίου σε αυτόν, ορίζονται οι ακόλουθες συναρτήσεις κόστους:

$$\begin{aligned}
 J_p(t) &= \left\| \mathbf{p}(t) + \mathbf{ln}(t) - \mathbf{p}_T(t) - \begin{bmatrix} 0 & 0 & z_{ref} \end{bmatrix}^T \right\|_{\mathbf{Q}_p}^2 \\
 J_v(t) &= \|\mathbf{v}(t) + \mathbf{l}\dot{\mathbf{n}}(t) - \mathbf{v}_T(t)\|_{\mathbf{Q}_v}^2 \\
 J_\eta(t) &= \left\| \begin{bmatrix} \phi(t) & \theta(t) & \psi(t) - \psi_{ref} \end{bmatrix}^T \right\|_{\mathbf{Q}_o}^2 \\
 J_L(t) &= \left\| \begin{bmatrix} \boldsymbol{\eta}_L^T(t) & \boldsymbol{\omega}_L^T(t) \end{bmatrix}^T \right\|_{\mathbf{Q}_L}^2
 \end{aligned} \tag{37}$$

όπου $\mathbf{Q}_p \in \mathbb{R}^{3 \times 3}$, $\mathbf{Q}_v \in \mathbb{R}^{3 \times 3}$, $\mathbf{Q}_o \in \mathbb{R}^{3 \times 3}$ και $\mathbf{Q}_L \in \mathbb{R}^{4 \times 4}$ είναι διαγώνιοι πίνακες στάθμισης και J_p , J_v , J_η και J_L είναι συναρτήσεις κόστους, ειδικά σχεδιασμένες για την συγκεκριμένη εφαρμογή, που σχετίζονται με την απόσταση μεταξύ του φορτίου και του στόχου, τη σχετική ταχύτητα μεταξύ του φορτίου και του στόχου, το σφάλμα μεταξύ του τρέχοντος και του επιθυμητού προσανατολισμού του οχήματος, και την ταλαντωτική κίνηση του σχοινιού. Σημειώνεται ότι το UAV θα πρέπει να παρακολουθεί τον επίγειο στόχο διατηρώντας ένα συγκεκριμένο ύψος πάνω από αυτόν, που ορίζεται από τη μεταβλητή z_{ref} . Η γωνία αναφοράς yaw ψ_{ref} διατηρείται σταθερή. Επιπλέον, η γωνιακή κίνηση του σχοινιού ελαχιστοποιείται, ώστε να αποφεύγονται περιττές ταλαντώσεις και να διασφαλίζεται η ασφάλεια του προς παράδοση φορτίου.

Λαμβάνοντας υπόψη τη συνάρτηση του συνολικού κόστους $J_t = J_p + J_v + J_\eta + J_L$ και τους περιορισμούς στην είσοδο του συστήματος, το ακόλουθο Πρόβλημα Βέλτιστου Ελέγχου (OCP) διατυπώνεται στον χρονικό ορίζοντα $[t_N, t_p]$:

$$\begin{aligned}
 \min_{\mathbf{v}} \int_{t_N}^{t_p} (J_t(t) + \|\mathbf{v}(t)\|_{\mathbf{R}}^2) dt + J_t(t_p) \\
 \text{s.t.: } \mathbf{x}(t_N) = \mathbf{x}_N, \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{v}) \text{ (Εξίσωση 2.11)}, \mathbf{v} \in \mathbb{U}
 \end{aligned} \tag{38}$$

όπου $\mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T \ \phi \ \theta \ \psi \ \boldsymbol{\eta}_L^T \ \boldsymbol{\omega}_L^T]^T \in \mathbb{R}^{13}$, $\mathbf{v} = [\phi_d \ \theta_d \ \psi_d \ v_{zd}]^T \in \mathbb{R}^4$ είναι το διάνυσμα κατάστασης και οι εντολές για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου αντίστοιχα, \mathbf{R} είναι ο πίνακας στάθμισης για την είσοδο και, \mathbb{U} είναι το σύνολο των περιορισμών στην είσοδο που καθορίζονται από τα όρια των γωνιών roll, pitch και της κατακόρυφης ταχύτητας.

Το προαναφερθέν OCP επιλύεται σε κάθε χρονική επανάληψη t_N , δεδομένης της προβλεπόμενης τροχιάς του στόχου, και η πρώτη είσοδος από την παραγόμενη ακολουθία

εισόδων αποστέλλεται στον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου. Το ACADO Toolkit [33], μαζί με τον επιλύτη qpOASES [17], χρησιμοποιούνται και πάλι για το σχεδιασμό και την επίλυση σε πραγματικό χρόνο του OCP.

Συνθήκη Απελευθέρωσης του Φορτίου

Όσον αφορά τη συνθήκη για την απελευθέρωση του φορτίου, σε κάθε τρέχουσα χρονική στιγμή t_N , υπολογίζεται η τροχιά του φορτίου υποθέτοντας ότι απελευθερώνεται. Στην περίπτωση αυτή, το φορτίο ακολουθεί μια βαλλιστική τροχιά, δηλαδή κινείται μόνο υπό την επίδραση της βαρύτητας, σύμφωνα με την ακόλουθη εξίσωση:

$$\mathbf{p}_L(t) = \mathbf{p}_L(t_N) + \mathbf{v}_L(t_N) \cdot t - g\mathbf{e}_z \frac{t^2}{2}, \quad (39)$$

όπου \mathbf{p}_L και \mathbf{v}_L δίνονται από την Εξίσωση 2.1. Δεδομένου του γνωστού ύψους του στόχου z_T και της παραπάνω εξίσωσης, υπολογίζεται η χρονική στιγμή t_h κατά την οποία η τροχιά του φορτίου τέμνεται με το z_T , δηλαδή $z_L(t_h) = z_T$. Συνεπώς, η συνθήκη για την απελευθέρωση του φορτίου διαμορφώνεται ως εξής:

$$\|\mathbf{p}_L(t_h) - \mathbf{p}_T(t_h)\| \leq d \quad (40)$$

όπου d είναι η μέγιστη οριζόντια απόσταση μεταξύ του φορτίου και του στόχου. Σημειώνεται ότι η συνθήκη υποδηλώνει ότι η χρονική στιγμή t_h βρίσκεται εντός του οριζοντα της προβλεπόμενης κίνησης του στόχου, δηλαδή $t_h \in [t_N, t_p]$, και, ως εκ τούτου, μια εκτίμηση της κίνησης του στόχου είναι διαθέσιμη. Διαφορετικά, η συνθήκη δεν ελέγχεται. Η Εικόνα 4.4 προσφέρει μια οπτική αναπαράσταση της συνθήκης για την απελευθέρωση του φορτίου.

Πειραματικά Αποτελέσματα

Πειραματική Διάταξη

Το προτεινόμενο σχήμα αξιολογείται πειραματικά σε ένα εξωτερικό περιβάλλον χρησιμοποιώντας το octorotor της Εικόνας 4.1 με τον ενσωματωμένο υπολογιστή Jetson AGX Xavier και την κάμερα ZED 2. Το όχημα είναι επιπλέον εξοπλισμένο με μία δυναμοκυψέλη, η οποία μετρά την τάση του σχοινιού $\|\mathbf{T}\|$, και έναν σερβοκινητήρα που ελέγχει μια αρπάγη και, επομένως, την απελευθέρωση του φορτίου. Μια βάση εκτυπώνεται για τη συναρμολόγηση της δυναμοκυψέλης και της αρπάγης, που βρίσκονται και οι δύο ακριβώς πάνω από το φορτίο. Τόσο η δυναμοκυψέλη όσο και ο σερβοκινητήρας συνδέονται απευθείας με ένα Arduino Uno, το οποίο επικοινωνεί σειριακά με το Jetson. Όσον αφορά το φορτίο, λαμβάνεται υπόψη μία τσάντα που περιέχει φορτίο $m_L = 0,5$ kg. Τέλος, ο επίγειος στόχος είναι το UGV Robotnik Summit που οδηγείται χειροκίνητα προς τυχαίες κατευθύνσεις μέσω τηλεχειρισμού με joystick.

Όσον αφορά το υπολογιστικό κόστος του προτεινόμενου σχήματος, ο μέσος χρόνος επεξεργασίας, στο υψηλών δυνατοτήτων Jetson, των μεμονωμένων μερών, που συνιστούν το σχήμα ελέγχου της Εικόνας 4.2, παρουσιάζεται στον Πίνακα 4.1. Είναι προφανές ότι επιτυγχάνεται επίδοση σε πραγματικό χρόνο κατά τη διάρκεια της παρακολούθησης του UGV.

Πειράματα σε Πραγματικό Περιβάλλον

Προκειμένου να επικυρωθεί η ικανότητα της προτεινόμενης στρατηγικής ελέγχου να παρακολουθεί αδιάκοπα το UGV, διεξήχθη αρχικά ένα πείραμα σε εξωτερικό χώρο χωρίς να ληφθεί υπόψη η απελευθέρωση του φορτίου. Κατά τη διάρκεια του πρώτου πειράματος, το UGV κινήθηκε με μέση οριζόντια ταχύτητα $1.5m/s$ και μέγιστη $2.5m/s$, όπως εκτιμήθηκε μέσω της καμπύλης Bézier (Εικόνες 4.5c, 4.5d). Το αναπτυσσόμενο σύστημα NMPC που βασίζεται στην όραση και επικοινωνεί απευθείας με το εσωτερικό βρόχο ελέγχου του UAV, επιτρέπει στο όχημα να ανταποκρίνεται γρήγορα στους άγνωστους ελιγμούς του στόχου, που εκτελούνται από τον χειριστή και, κατά συνέπεια, η συνεχής παρακολούθηση του στόχου επιτεύχθηκε, σύμφωνα με τις Εικόνες 4.5a, 4.5b, όπου απεικονίζονται οι τροχιές και των δύο οχημάτων στο οριζόντιο επίπεδο. Επιπλέον, παρά τις απότομες κινήσεις του UGV, η ταλαντωτική κίνηση του σχοινιού, που εκτιμάται σύμφωνα με την Ενότητα 3.3.1, δεν διεγέρθηκε και οι γωνίες του σχοινιού διατηρήθηκαν κάτω από 20° , όπως φαίνεται στην Εικόνα 4.5e.

Στη συνέχεια, πραγματοποιήθηκε ένα δεύτερο πείραμα προκειμένου να αξιολογηθεί η ικανότητα του προτεινόμενου σχήματος να εναποθέσει επιτυχώς το αναρτημένο φορτίο στον στόχο. Το UAV αρχικά παρακολούθησε το όχημα μέχρι να ικανοποιηθεί η συνθήκη για την απελευθέρωση του φορτίου (Εξίσωση 4.8), όπως απεικονίζεται στην Εικόνα 4.6a. Εκείνη τη στιγμή, ενεργοποιήθηκε το άνοιγμα της αρπάγης και το φορτίο απελευθερώθηκε με επιτυχία και τοποθετήθηκε εντός κουτιού, που μετέφερε το UGV, όπως φαίνεται στην Εικόνα 4.6.

Τα προαναφερθέντα πειράματα, συμπεριλαμβανομένου και ενός επιπλέον, απεικονίζονται καλύτερα στο σχετικό βίντεο <https://youtu.be/6aleTdb71Sc>.

Δειγματοληψία Νερού από Υδάτινα Περιβάλλοντα

Εισαγωγή

Σε αυτό το κεφάλαιο, εξετάζουμε τη διεξαγωγή δειγματοληψίας νερού από ένα υδάτινο περιβάλλον με τη βοήθεια ενός πολυκοπτέρου. Προκειμένου να αποφευχθεί η ενσωμάτωση μιας ειδικά σχεδιασμένης πλωτής βάσης στο κάτω μέρος του σώματος του UAV, η οποία επιτρέπει στο όχημα να προσγειωθεί και να επιπλέει στην επιφάνεια του νερού [1],[43],[58], προτιμούμε την ανάρτηση ενός μηχανισμού μέσω σχοινιού, καθώς αποτελεί μια έτοιμη προς χρήση λύση από πλευράς μηχανικής πολυπλοκότητας και δεν περιορίζει την εφαρμοσιμότητα του UAV σε εργασίες που σχετίζονται αποκλειστικά με τη δειγματοληψία νερού.

Ωστόσο, η οπισθέλκουσα δύναμη που ασκείται στον βυθισμένο μηχανισμό, λόγω της αλληλεπίδρασής του με τη ροή του νερού, οδηγεί σε μια επικλινή θέση του σχοινιού, κυρίως σε ποτάμια όπου συνήθως παρατηρείται μια αρκετά υψηλή μέση ταχύτητα νερού. Η τάση του επικλινούς σχοινιού εμποδίζει το UAV να σταθεροποιηθεί επακριβώς πάνω από την επιθυμητή θέση δειγματοληψίας και, επομένως, πρέπει να εκτιμηθεί με βάση τους ενσωματωμένους πάνω στο όχημα αισθητήρες. Τεχνικές που βασίζονται αποκλειστικά στην όραση δεν είναι κατάλληλες σε αυτή την περίπτωση, καθώς ο μηχανισμός είναι βυθισμένος και, ως εκ τούτου, η ανίχνευσή του είναι εξαιρετικά δύσκολη, αν όχι ανέφικτη, και εξαρτάται σε μεγάλο βαθμό από την καθαρότητα του νερού. Για το σκοπό αυτό, το σετ αισθητήρων της ρομποτικής πλατφόρμας επαυξάνεται με πρόσθετους αισθητήρες χαμηλού κόστους, συγκεκριμένα έναν αισθητήρα υπερήχων, μια δυναμοκυψέλη και έναν αισθητήρα βάθους, πέρα φυσικά από την κάμερα με προς τα κάτω προσανατολισμό. Με σκοπό τη σύντηξη των αντίστοιχων μετρήσεων, σχεδιάζουμε δύο μεθοδολογίες: (α) παλινδρόμηση με Gaussian Processes (GPs) και (β) Extended Kalman Filter (EKF). Τελικά, η επίδραση της διαταραχής μετριάζεται ενσωματώνοντας την εκτίμηση της τάσης του σχοινιού σε ένα σχήμα ελέγχου και, έτσι, επιτυγχάνεται η ακριβής σταθεροποίηση του UAV πάνω από τη θέση δειγματοληψίας. Δύο σχήματα ελέγχου διερευνώνται για αυτήν την εφαρμογή, συγκεκριμένα: (α) ένα σχήμα γεωμετρικού ελέγχου (geometric controller) και (β) μια στρατηγική μη γραμμικού προβλεπτικού ελέγχου (NMPC).

Ορισμός του Προβλήματος

Στην ενότητα αυτή, διατυπώνουμε το πρόβλημα ελέγχου της δειγματοληψίας νερού με πολυκόπτερο. Συγκεκριμένα, εξετάζουμε την περίπτωση κατά την οποία ένα UAV με αναρτημένο μέσω σχοινιού μηχανισμό πραγματοποιεί δειγματοληψία νερού σε ένα υδάτινο περιβάλλον, όπως φαίνεται στην Εικόνα 5.1. Υποθέτοντας ότι η ταχύτητα της ροής του νερού \mathbf{u}_w δεν είναι αμελητέα, η επαγόμενη οπισθέλκουσα δύναμη \mathbf{f}_d (Εξίσωση 2.12) που δρα στον βυθισμένο μηχανισμό, έχει ως αποτέλεσμα την επικλινή θέση του σχοινιού, δηλαδή $|n_z| < 1$. Συνεπώς, η τάση του επικλινούς σχοινιού $\mathbf{T} = \|\mathbf{T}\|\mathbf{n}$, που ασκείται στο όχημα, επηρεάζει τον έλεγχο της θέσης του UAV και, κατά συνέπεια, όταν δεν λαμβάνεται υπόψη η εξωτερική δύναμη παρατηρείται αυξημένο σφάλμα, σε σύγκριση με την επιθυμητή θέση δειγματοληψίας \mathbf{p}_{ref} .

Για το σκοπό αυτό, απαιτείται ένα κατάλληλο σχήμα ελέγχου προκειμένου να επιτευχθεί η ακριβής σταθεροποίηση του οχήματος πάνω από τη θέση δειγματοληψίας παρά τις υποβρύχιες διαταραχές που μεταφέρονται στο UAV μέσω του σχοινιού. Τα ακόλουθα μέρη υλοποιούνται:

1. Άμεση εκτίμηση της τάσης του σχοινιού $\hat{\mathbf{T}}$ με βάση τις μετρήσεις που παρέχονται από κατάλληλα ενσωματωμένους επί του οχήματος αισθητήρες, συγκεκριμένα έναν αισθητήρα υπερήχων, μια κάμερα, μια δυναμοκυψέλη και έναν αισθητήρα βάθους,
2. Προσέγγιση της τάσης του σχοινιού με τη βοήθεια ενός Extended Kalman Filter ή Gaussian Processes (GPs) προκειμένου να μετριάσει ο θόρυβος στις μετρήσεις των αισθητήρων και
3. Ανάπτυξη ενός γεωμετρικού ελεγκτή ή ενός μη γραμμικού προβλεπτικού ελεγκτή (NMPC) που λαμβάνει υπόψη τις εκτιμήσεις της τάσης και παράγει τις κατάλληλες εντολές για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου, ώστε να ελαχιστοποιηθεί το σφάλμα θέσης $\mathbf{p} - \mathbf{p}_{ref}$.

Εκτίμηση της Διαταραχής

Εκτίμηση της Διαταραχής βάσει Αισθητήρων

Προκειμένου να επιτευχθεί η εύρωστη σταθεροποίηση του οχήματος πάνω από την επιθυμητή θέση δειγματοληψίας, είναι απαραίτητη μια εκτίμηση της τάσης του σχοινιού $\mathbf{T} = \|\mathbf{T}\|\mathbf{n}$. Για το σκοπό αυτό, αξιοποιούνται μία κάμερα που κοιτάζει προς τα κάτω, μία δυναμοκυψέλη, ένας αισθητήρας υπερήχων και ένας αισθητήρας βάθους.

Πιο συγκεκριμένα, προστίθεται μια δυναμοκυψέλη στη μέση του σχοινιού με τέτοιο τρόπο ώστε να μετρά άμεσα την τάση. Σημειώνεται ότι η δυναμοκυψέλη παρέχει αξιόπιστες μετρήσεις της τάσης με χαμηλό θόρυβο. Ωστόσο, μόνο η νόρμα της τάσης $\|\mathbf{T}\|$ μετρείται από την δυναμοκυψέλη και, ως εκ τούτου, απαιτούνται πρόσθετοι αισθητήρες για να εκτιμηθεί η κατεύθυνση της διαταραχής, δηλαδή το μοναδιαίο διάνυσμα του σχοινιού \mathbf{n} .

Κατά συνέπεια, τοποθετείται ένας αισθητήρας βάρους στο κάτω άκρο του σχοινιού ο οποίος υπολογίζει το βάθος d μέχρι το οποίο είναι βυθισμένος ο μηχανισμός δειγματοληψίας, δηλαδή την κατακόρυφη απόσταση μεταξύ της επιφάνειας του νερού και του κάτω άκρου του σχοινιού. Εφόσον το μήκος του σχοινιού l είναι γνωστό, προκύπτει μια εκτίμηση της συνιστώσας \hat{n}_z , υπό την προϋπόθεση ότι το καλώδιο είναι τεντωμένο, με βάση την ομοιότητα τριγώνων:

$$\hat{n}_z = -\frac{h+d}{l} \quad (41)$$

όπου το h υποδηλώνει το ύψος του οχήματος πάνω από την επιφάνεια του νερού, το οποίο μετρείται απευθείας από τον αισθητήρα υπερήχων και διορθώνεται για μη μηδενικές γωνίες roll και pitch. Δεδομένου ότι οι αισθητήρες υπερήχων είναι επιρρεπείς σε θόρυβο πάνω από το νερό, εφαρμόζεται ένα φίλτρο κινούμενου μέσου όρου, ενώ οι ακραίες τιμές αγνοούνται.

Επιπλέον, για να εκτιμηθεί η προβολή του μοναδιαίου διανύσματος του σχοινιού \mathbf{n} στο επίπεδο $x-y$ του συστήματος αναφοράς \mathbf{W} , αξιοποιείται η κάμερα. Πιο συγκεκριμένα, χρησιμοποιείται ο αλγόριθμος Dense Inverse Search (DIS) [44], ένας αλγόριθμος οπτικής ροής (optical flow) που διακρίνεται για τη χαμηλή υπολογιστική πολυπλοκότητά του και την ακρίβειά του, με σκοπό τον υπολογισμό ενός πυκνού πεδίου ροής (dense flow field), δηλαδή της οπτικής ροής για όλα τα σημεία της εικόνας. Το αποτέλεσμα του αλγόριθμου DIS παρουσιάζεται στην Εικόνα 5.2 τόσο για το περιβάλλον προσομοίωσης όσο και για πραγματικά περιβάλλοντα.

Υπολογίζοντας τον μέσο όρο του πεδίου ροής, εξάγεται η κύρια ταχύτητα ροής του νερού ${}^C\hat{\mathbf{v}}_w$ σε σχέση με το σύστημα συντεταγμένων της κάμερας αφού μετατραπούν οι ταχύτητες των pixels σε καρτεσιανές και αφαιρεθεί η οδομετρία του οχήματος. Στη συνέχεια, εκτελείται ένας κατάλληλος μετασχηματισμός στο σύστημα αναφοράς \mathbf{W} , χρησιμοποιώντας τον σταθερό πίνακα περιστροφής από το σύστημα συντεταγμένων της κάμερας στο σύστημα του οχήματος \mathbf{B} και τον πίνακα \mathbf{R}_{WB} και, τελικά, υπολογίζεται η ταχύτητα του νερού $\hat{\mathbf{v}}_w$ σε σχέση με το \mathbf{W} .

Σύμφωνα με την Εξίσωση 2.12, η προβολή $\mathbf{n}_{xy} = [n_x \ n_y \ 0]^T$ του μοναδιαίου διανύσματος του σχοινιού στο επίπεδο $x-y$ του \mathbf{W} , δηλαδή στην επιφάνεια του νερού, και η οπισθέλκουσα δύναμη \mathbf{f}_d , που δρα στον μηχανισμό, είναι ομόρροπα διανύσματα. Δεδομένου ότι η οπισθέλκουσα δύναμη \mathbf{f}_d και η ταχύτητα της ροής του νερού \mathbf{v}_w είναι επίσης ομόρροπα διανύσματα, προκύπτει η ακόλουθη εξίσωση:

$$\text{atan2}(\hat{n}_y, \hat{n}_x) = \text{atan2}(\hat{u}_{w,y}, \hat{u}_{w,x}) = \theta_w \quad (42)$$

όπου θ_w είναι η κατεύθυνση της ροής του νερού. Σημειώνεται ότι η παραπάνω εξίσωση ισχύει υπό την προϋπόθεση ότι ο βυθισμένος μηχανισμός είναι περίπου σταθεροποιημένος, δηλαδή $\|\mathbf{v}_L\| \approx 0$ και $\|\dot{\mathbf{v}}_L\| \approx 0$.

Παλινδρόμηση με Gaussian Processes

Για την αντιμετώπιση του θορύβου, που συνοδεύει τις μετρήσεις των αισθητήρων, σχεδιάζεται μια μη παραμετρική εκτίμηση της τάσης του σχοινοῦ βασισμένη στη μηχανική μάθηση και, συγκεκριμένα, στις Gaussian Processes. Η έννοια της παλινδρόμησης με GPs περιγράφεται διεξοδικά στην Ενότητα 3.6.3.

Λαμβάνοντας υπόψη τις απαιτήσεις της συγκεκριμένης εφαρμογής, το πρόβλημα της παλινδρόμησης διατυπώνεται με τέτοιο τρόπο ώστε να προσεγγίζεται η τάση του σχοινοῦ. Επομένως, το σετ δεδομένων D αποτελείται από την τάση του σχοινοῦ $\hat{\mathbf{T}} = \|\mathbf{T}\|\hat{\mathbf{n}}$, μετρούμενη σύμφωνα με την Ενότητα 5.3.1, ως έξοδο, και την κατάσταση του οχήματος $\mathbf{x} = [\mathbf{p} \ \mathbf{v}]^T$ ως δεδομένο εισόδου:

$$D = \left\{ \mathbf{Z} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times 6}, \mathbf{Y} = [\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_N]^T \in \mathbb{R}^{N \times 3} \right\} \quad (43)$$

Όπως αναφέρθηκε στην Ενότητα 3.6.3, τα GPs μοντελοποιούν βαθμωτές συναρτήσεις. Συνεπώς, χρησιμοποιούνται 2 ανεξάρτητες Gaussian Processes για να προσεγγιστεί το διάνυσμα της τάσης, δεδομένου ότι η νόρμα της τάσης $\|\mathbf{T}\|$ μετριέται απευθείας από την δυναμοκυψέλη και ότι το μοναδιαίο διάνυσμα του σχοινοῦ \mathbf{n} έχει 2 βαθμούς ελευθερίας. Μόλις είναι διαθέσιμη μια νέα μέτρηση της τάσης του σχοινοῦ $\hat{\mathbf{T}}$, το σύνολο εκπαίδευσης D και οι GPs επικαιροποιούνται σε πραγματικό χρόνο κατά τη διαδικασία της δειγματοληψίας.

Extended Kalman Filter

Εναλλακτικά, παρουσιάζουμε μια υπολογιστικά πιο αποδοτική προσέγγιση που βασίζεται σε ένα φίλτρο Kalman, το οποίο συνδυάζει τις μετρήσεις που παρέχονται από τους αισθητήρες του οχήματος και προσεγγίζει την τάση του καλωδίου \mathbf{T} . Πιο συγκεκριμένα, δεδομένης της γνωστής νόρμας της τάσης $\|\mathbf{T}\|$, ο απώτερος στόχος του φίλτρου Kalman είναι να εκτιμήσει το μοναδιαίο διάνυσμα του σχοινοῦ \mathbf{n} αξιοποιώντας τις Εξισώσεις 5.1 και 5.2. Για το σκοπό αυτό, ορίζοντας $\mathbf{x}_L \equiv \mathbf{n} = [n_x \ n_y \ n_z]^T \in \mathbb{R}^3$ ως το διάνυσμα κατάστασης, διατυπώνονται τα ακόλουθα μοντέλα για το σύστημα και τις μετρήσεις:

$$\dot{\mathbf{x}}_L = \dot{\mathbf{n}} = \mathbf{w}_L \quad (44)$$

$$\mathbf{y}_L = \begin{bmatrix} \|\mathbf{n}\| \\ \text{atan2}(\hat{n}_y, \hat{n}_x) \\ \hat{n}_z \end{bmatrix} + \mathbf{v}_L \quad (45)$$

όπου $\mathbf{w}_L \in \mathbb{R}^3$ και $\mathbf{v}_L \in \mathbb{R}^3$ είναι αθροιστικοί γκαουσιανοί θόρυβοι με διαγώνιους πίνακες συμμεταβλητότητας. Σύμφωνα με την παραπάνω διατύπωση, το μοναδιαίο διάνυσμα του σχοινοῦ μοντελοποιείται ως random walk ενώ ο περιορισμός ισότητας $\|\mathbf{n}\| = 1$ ενσωματώνεται στο μοντέλο μέτρησης σαν μια σχεδόν αθόρυβη ψευδομέτρηση [3]. Λόγω του μη γραμμικού μοντέλου των μετρήσεων, αναπτύσσεται ένα Extended Kalman Filter (EKF).

Έλεγχος Θέσης

Γεωμετρικός Έλεγχος

Μετά την προσέγγιση της τάσης του σχοινιού \mathbf{T} , αναπτύσσεται ένας ελεγκτής θέσης προκειμένου να απορριφθεί η εκτιμώμενη διαταραχή και να σταθεροποιηθεί με ακρίβεια το όχημα πάνω από την επιθυμητή θέση δειγματοληψίας \mathbf{p}_{ref} . Αρχικά, υλοποιείται ο γεωμετρικός ελεγκτής, που παρουσιάζεται στα [47], [56], και ο οποίος βασίζεται στην ιδιότητα differential flatness της δυναμικής του πολυκόπτερου, με μια επέκταση, έτσι ώστε να συμπεριληφθεί η εξωτερική δύναμη [8], [84], συγκεκριμένα η τάση του σχοινιού.

Πιο αναλυτικά, ορίζονται τα ακόλουθα σφάλματα θέσης και ταχύτητας:

$$\mathbf{e}_p = \mathbf{p} - \mathbf{p}_{ref} \quad (46)$$

$$\mathbf{e}_v = \mathbf{v} - \mathbf{v}_{ref} \quad (47)$$

Εφόσον ο στόχος είναι η σταθεροποίηση του οχήματος σε μια δεδομένη θέση, η ταχύτητα αναφοράς είναι ίση με μηδέν, $\mathbf{v}_{ref} = [0 \ 0 \ 0]^T$. Σύμφωνα με τη δυναμική του οχήματος, που παρουσιάζεται στην Εξίσωση 2.9, ο επιθυμητός προσανατολισμός του UAV, εκφρασμένος μέσω του πίνακα περιστροφής \mathbf{R}_d , υπολογίζεται, δεδομένης μιας επιθυμητής γωνίας yaw ψ_d , ως εξής:

$$\mathbf{b}_{3,d} = \frac{-k_p \mathbf{e}_p - k_v \mathbf{e}_v + m g \mathbf{e}_z - \mathbf{T}}{\| -k_p \mathbf{e}_p - k_v \mathbf{e}_v + m g \mathbf{e}_z - \mathbf{T} \|} \quad (48)$$

$$\mathbf{b}_{2,d} = [-s\psi_d \ c\psi_d \ 0]^T \quad (49)$$

$$\mathbf{b}_{1,d} = \frac{\mathbf{b}_{2,d} \times \mathbf{b}_{3,d}}{\| \mathbf{b}_{2,d} \times \mathbf{b}_{3,d} \|} \quad (50)$$

$$\mathbf{R}_d = [\mathbf{b}_{1,d} \ \mathbf{b}_{3,d} \times \mathbf{b}_{1,d} \ \mathbf{b}_{3,d}] \quad (51)$$

όπου $k_p, k_v \in \mathbb{R}^+$ είναι θετικά κέρδη. Ο πίνακας περιστροφής \mathbf{R}_d τελικά μετατρέπεται στις επιθυμητές γωνίες Euler ϕ_d, θ_d, ψ_d για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου. Σημειώνεται ότι άλλες εξωτερικές δυνάμεις, που δρουν στην ρομποτική πλατφόρμα, π.χ. αεροδυναμικές δυνάμεις οπισθέλκουσας, αμελούνται.

Μη Γραμμικός Προβλεπτικός Έλεγχος

Επιπλέον, αναπτύσσεται ένας μη γραμμικός προβλεπτικός ελεγκτής (NMPC) ως εναλλακτική στρατηγική ελέγχου προκειμένου να επιτευχθεί η ακριβής σταθεροποίηση του οχήματος κατά τη διάρκεια της δειγματοληψίας και να επιβληθούν περιορισμοί στην κατάσταση του οχήματος και στην είσοδο. Έτσι, αποφεύγονται απότομοι ελιγμοί του οχήματος πάνω από την επιφάνεια του νερού και αυξάνεται το επίπεδο ασφάλειας της διαδικασίας της δειγματοληψίας.

Το παραπάνω σχήμα NMPC βασίζεται στη δυναμική του οχήματος, δηλαδή στην Εξίσωση 2.9, με την εκτιμώμενη τάση του σχοινιού \mathbf{T} να θεωρείται ως η μόνη πηγή εξωτερικών διαταραχών. Επομένως, το διάνυσμα κατάστασης αποτελείται από τη θέση, την ταχύτητα και τον προσανατολισμό του οχήματος, δηλαδή $\mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T \ \phi \ \theta \ \psi]^T \in \mathbb{R}^9$, ενώ το διάνυσμα εισόδου ορίζεται από τις επιθυμητές εντολές για τον εσωτερικό βρόχο ελέγχου του αυτόματου πιλότου, $\mathbf{v} = [\phi_d \ \theta_d \ \psi_d \ u_{z_d}]^T \in \mathbb{R}^4$. Το πρόβλημα βέλτιστου ελέγχου (OCP) που πρέπει να λυθεί από τον NMPC εκφράζεται ως:

$$\begin{aligned} \min_{\mathbf{v}} \int_{t_0}^{t_0+T} (\|\mathbf{x}(t) - \mathbf{x}_{ref}(t)\|_{\mathbf{Q}}^2 + \|\mathbf{v}(t)\|_{\mathbf{R}}^2) dt + \|\mathbf{x}(t_0+T) - \mathbf{x}(t_0+T)\|_{\mathbf{P}}^2 \\ \text{s.t.: } \mathbf{x}(t_0) = \mathbf{x}_0, \text{ Εξίσωση 2.9, } \mathbf{v} \in \mathbb{U} \end{aligned} \quad (52)$$

όπου $\mathbf{x}_{ref} = [\mathbf{p}_{ref}^T \ \mathbf{0}_{1 \times 3} \ 0 \ 0 \ \psi_{ref}]^T \in \mathbb{R}^9$ είναι το επιθυμητό διάνυσμα κατάστασης. Ομοίως με την Ενότητα 3.6.1, οι πίνακες κόστους \mathbf{Q} , \mathbf{R} και \mathbf{P} είναι θετικά ορισμένοι και διαγώνιοι και, σε αυτήν την περίπτωση, ο πίνακας κόστους για το σφάλμα κατάστασης \mathbf{Q} ορίζεται από την ακόλουθη εξίσωση:

$$\mathbf{Q} = \text{diag} \left([\mathbf{Q}_p \ \mathbf{Q}_v \ \mathbf{Q}_o] \right) \quad (53)$$

όπου τα διαγώνια μητρώα στάθμισης \mathbf{Q}_p , \mathbf{Q}_v και \mathbf{Q}_o επιβάλλουν κυρώσεις στα σφάλματα της θέσης, της ταχύτητας και του προσανατολισμού αντίστοιχα.

Εκτός από του περιορισμούς εισόδου, που ορίζονται από το σύνολο \mathbb{U} , περιλαμβάνονται επίσης στο OCP περιορισμοί στην κατάσταση του συστήματος, έτσι ώστε να περιοριστεί η κίνηση του UAV στο επίπεδο $x-y$, σύμφωνα με τις ακόλουθες ανισότητες:

$$-v_{x,max} \leq v_x \leq v_{x,max} \quad (54\alpha')$$

$$-v_{y,max} \leq v_y \leq v_{y,max} \quad (54\beta')$$

Με την επιβολή περιορισμών στην είσοδο και στην κατάσταση, διασφαλίζεται η ομαλή συμπεριφορά του NMPC σχήματος και, επομένως, η ασφαλής εκτέλεση της διαδικασίας της δειγματοληψίας. Θα πρέπει να τονιστεί ότι η ασφάλεια είναι ένας κρίσιμος παράγοντας για τη συγκεκριμένη εφαρμογή δεδομένου ότι το UAV επιχειρεί σε υδάτινο περιβάλλον και πάνω από την επιφάνεια του νερού. Κατά συνέπεια, απότομοι ελιγμοί μπορεί να θέσουν σε κίνδυνο τα ηλεκτρικά μέρη που αποτελούν το όχημα.

Το ACADO Toolkit και ο επιλύτης qpOASES αξιοποιούνται ξανά για την επίλυση του προαναφερθέντος προβλήματος βελτιστοποίησης με περιορισμούς και, έτσι, επιτυγχάνεται επίδοση σε πραγματικό χρόνο, παρόμοια με την Ενότητα 3.6.1.

Πειραματικά Αποτελέσματα

Πειραματική Διάταξη

Προκειμένου να αποδειχθεί η ικανότητα των παραπάνω μεθοδολογιών να προσεγγίζουν την τάση του σχοινιού και να σταθεροποιούν το UAV στην επιθυμητή θέση δειγματοληψίας, πραγματοποιήθηκαν πειράματα με το octorotor της Εικόνας 2.3. Το όχημα είναι εξοπλισμένο με μια κάμερα που κοιτάζει προς τα κάτω, συγκεκριμένα τη stereocamera ZED 2, που αξιοποιείται για την εκτίμηση της ροής του νερού. Επιπλέον, χρησιμοποιείται μια δυναμοκυψέλη 5 kg, μαζί με τον ενισχυτή HX711, για τη μέτρηση της νόρμας της τάσης του σχοινιού $\|\mathbf{T}\|$ κατά τη διάρκεια της δειγματοληψίας νερού. Η δυναμοκυψέλη τοποθετείται κατάλληλα στη μέση του σχοινιού. Επιπλέον, ο Bar30 High-Resolution 300m αισθητήρας βάθους, συνοδευόμενος από έναν I2C Level Converter, παρέχει το βάθος d στο οποίο βρίσκεται ο μηχανισμός δειγματοληψίας. Ο προαναφερθείς αισθητήρας βάθους μπορεί να μετρήσει έως και 30 Bar με διακριτότητα 2 mm. Ο αισθητήρας βάθους είναι στερεωμένος στο κάτω άκρο του σχοινιού και, έτσι, συλλέγονται δείγματα νερού στα επιθυμητά βάθη. Τέλος, ο αισθητήρας υπερήχων A01NYUB βρίσκεται κοντά στο σώμα του οχήματος έτσι ώστε να μετράει την απόσταση h μεταξύ του UAV και της επιφάνειας του νερού. Το εύρος μέτρησης του αισθητήρα υπερήχων είναι 28-750 cm, γεγονός το οποίο σημαίνει ότι το μήκος του σχοινιού θα πρέπει να είναι μικρότερο από 750 cm. Η ακριβής θέση των προαναφερθέντων αισθητήρων στην πλατφόρμα απεικονίζεται καλύτερα στην Εικόνα 5.3. Όλοι οι πρόσθετοι αισθητήρες, δηλαδή ο αισθητήρας υπερήχων, ο αισθητήρας βάθους και η δυναμοκυψέλη, συνδέονται απευθείας με ένα Arduino Uno [57] και οι αντίστοιχες μετρήσεις μεταφέρονται στον ενσωματωμένο υπολογιστή Jetson AGX Xavier μέσω σειριακής επικοινωνίας.

Γεωμετρικός Έλεγχος με Gaussian Processes

Πειράματα σε Προσομοιωτή

Αρχικά, διεξήχθη μια συγκριτική μελέτη στο περιβάλλον προσομοίωσης της Εικόνας 2.2 μεταξύ του γεωμετρικού ελεγκτή της Ενότητας 5.4.1 με και χωρίς τον όρο της τάσης του σχοινιού, εκτιμώμενη με τη βοήθεια των GPs (Ενότητα 5.3.2). Το σχήμα ελέγχου παρουσιάζεται στην Εικόνα 5.4. Όπως απεικονίζεται στις Εικόνες 5.5a, 5.5b, ο ελεγκτής θέσης που συμπεριλαμβάνει την πρόβλεψη των GPs ήταν ικανός να σταθεροποιήσει με ακρίβεια το όχημα πάνω από τη θέση δειγματοληψίας, ενώ σημαντικά σφάλματα μόνιμης κατάστασης στη θέση $\mathbf{p} - \mathbf{p}_{ref}$ παρατηρήθηκαν κατά το ίδιο σενάριο όταν δεν ελήφθη υπόψη ο όρος της διαταραχής \mathbf{T} . Η εκτίμηση της διαταραχής, χρησιμοποιώντας τις GPs, ήταν επιτυχής καθώς η πραγματική τάση του σχοινιού, που λαμβάνεται απευθείας από τον προσομοιωτή Gazebo, βρίσκεται μέσα στα όρια εμπιστοσύνης των GPs, όπως φαίνεται στα σχήματα 5.5c, 5.5d. Σημειώνεται ότι κατά τη διάρκεια του σεναρίου, η μάζα του μηχανισμού αυξήθηκε για να προσομοιωθεί η διαδικασία της δειγματοληψίας νερού.

Πειράματα σε Πραγματικό Περιβάλλον

Μετά την επαλήθευση στο περιβάλλον προσομοίωσης Gazebo, μια συγκριτική πειραματική μελέτη μεταξύ του γεωμετρικού ελεγκτή με και χωρίς τον όρο, που σχετίζεται με τη διαταραχή, διεξήχθη με το `octotoro1` σε πραγματικές συνθήκες. Πιο συγκεκριμένα, τα πειράματα πραγματοποιήθηκαν στον ποταμό Λάδωνα, όπου ασκήθηκε σημαντική οπισθέλκουσα δύναμη στον βυθισμένο μηχανισμό, λόγω της ταχύτητας της ροής του νερού, και ως εκ τούτου παρατηρήθηκε μια επικλινή θέση του σχοινιού όπως απεικονίζεται στην Εικόνα 5.1. Σημειώνεται ότι τα πειράματα έγιναν διαδοχικά και, έτσι, οι εξωτερικές συνθήκες, και ιδιαίτερα οι αεροδυναμικές δυνάμεις, μπορούν να θεωρηθούν παρόμοιες. Επιπλέον, χρησιμοποιήθηκαν τα ίδια θετικά κέρδη k_p , k_v και, κατά συνέπεια, εξασφαλίστηκε μια δίκαιη σύγκριση μεταξύ των δύο ελεγκτών. Τέλος, ένα μπουκάλι 1 λίτρου και μάζας $m_L = 1.2\text{kg}$ χρησιμοποιήθηκε ως υποκατάστατο ενός πραγματικού μηχανισμού δειγματοληψίας.

Αρχικά, χρησιμοποιήθηκε ο ελεγκτής θέσης, ο οποίος δεν λαμβάνει υπόψη την τάση του σχοινιού. Όπως φαίνεται στην Εικόνα 5.6a, παρατηρήθηκε ένα σφάλμα μόνιμης κατάστασης περίπου 1m στον άξονα y του συστήματος αναφοράς \mathbf{W} , το οποίο είναι μια λογική παρατήρηση αφού πράγματι η κατεύθυνση της κύριας ροής του νερού είναι σχεδόν παράλληλη με τον άξονα y . Αντίθετα, όταν ο όρος διαταραχής εκτιμήθηκε με τη βοήθεια των GPs (Εικόνες 5.6c, 5.6d), το σφάλμα θέσης μειώθηκε σημαντικά και το όχημα σταθεροποιήθηκε στην επιθυμητή θέση, όπως απεικονίζεται στην Εικόνα 5.6b. Κατά συνέπεια, η εκτίμηση της διαταραχής και η επίδοση του ελεγκτή θέσης μπορούν να θεωρηθούν επιτυχημένες.

Οι προαναφερθείσες συγκριτικές μελέτες που επαληθεύουν την αποτελεσματικότητα του γεωμετρικού ελεγκτή, επαυξημένου με τις προβλέψεις των GPs, απεικονίζονται επίσης στο παρακάτω βίντεο <https://youtu.be/4BynPU0ec0k>.

NMPC με EKF

Στη συνέχεια, εξετάσαμε την ικανότητα του σχήματος NMPC της Ενότητας 5.4.2 να σταθεροποιήσει το όχημα στη θέση δειγματοληψίας, δεδομένων των εκτιμήσεων της τάσης του σχοινιού σύμφωνα με το φίλτρο Kalman της Ενότητας 5.3.3. Σε όλα τα παρακάτω πειράματα, χρησιμοποιήθηκε μια πραγματική συσκευή δειγματοληψίας μάζας $m_L = 0,25\text{kg}$, αναρτημένη μέσω ενός σχοινιού μήκους $l = 3,8\text{m}$. Ο μηχανισμός επιτυγχάνει τη δειγματοληψία νερού μέσω της χρήσης μιας ηλεκτρομαγνητικής βαλβίδας που αποκαλύπτει μια οπή και επιτρέπει τη ροή του νερού μέσα σε έναν γυάλινο σωλήνα. Μια σύνοψη της αρχιτεκτονικής ελέγχου παρουσιάζεται στην Εικόνα 5.7. Προκειμένου να επικυρωθεί η αναγνώριση της τάσης του σχοινιού και ο ακριβής έλεγχος θέσης του οχήματος κατά τη διαδικασία της δειγματοληψίας, πραγματοποιήθηκαν πειράματα στα πιο κοινά υδάτινα περιβάλλοντα, δηλαδή σε θάλασσα και ποτάμι.

Στην πρώτη περίπτωση, παρά την ύπαρξη κυμάτων, η οπισθέλκουσα δύναμη, που προκαλείται από τη ροή του νερού και δρα στον μηχανισμό, ήταν αμελητέα. Συνεπώς,

το σχοινί διατηρήθηκε παράλληλο με τον άξονα z του συστήματος αναφοράς κατά τη διάρκεια ολόκληρου του πειράματος και, ως εκ τούτου, η συνιστώσα z της τάσης του σχοινιού $\mathbf{T} = \|\mathbf{T}\|\mathbf{n}$ ήταν η κυρίαρχη διαταραχή που ασκήθηκε στο UAV, όπως σωστά εκτιμήθηκε από τους πρόσθετους αισθητήρες και το αναπτυγμένο EKF (Εικόνα 5.8d). Όσον αφορά την επίδοση του NMPC, το UAV σταθεροποιήθηκε με ακρίβεια πάνω από την επιθυμητή θέση δειγματοληψίας, όπως απεικονίζεται στην Εικόνα 5.8.

Στην δεύτερη περίπτωση, η οπισθέλκουσα δύναμη, λόγω της ροής του ποταμού, ήταν ιδιαίτερα σημαντική και οδήγησε σε επικλινή θέση του σχοινιού. Ως αποτέλεσμα, οι οριζόντιες διαταραχές εκτιμήθηκαν επίσης από τους ενσωματωμένους αισθητήρες και το EKF, όπως φαίνεται στην Εικόνα 5.9d. Παρά την τάση του σχοινιού, το NMPC σχήμα σταθεροποίησε με επιτυχία το όχημα πάνω από την επιθυμητή θέση, όπως απεικονίζεται στην Εικόνα 5.9.

Τα πειράματα που έγιναν στη θάλασσα και στον ποταμό Λάδωνα παρουσιάζονται καλύτερα στα σχετικά βίντεο <https://youtu.be/IB7a72ZvBQc> και <https://youtu.be/EFHL8ckaWpM>.

Αυτόνομες Αποστολές Δειγματοληψίας Νερού

Σε αυτήν την ενότητα ενοποιούμε τα αποτελέσματα των Κεφαλαίων 3 και 5 σε μια πλήρως αυτόνομη λύση για αποστολές δειγματοληψίας νερού χρησιμοποιώντας ένα UAV με αναρτημένο μέσω σχοινιού μηχανισμό και εξοπλισμένο με τους απαραίτητους αισθητήρες, αλγορίθμους εκτίμησης κατάστασης και σχήματα ελέγχου. Προς αυτόν τον στόχο, σχεδιάζουμε ένα εντελώς αυτόνομο και ενοποιημένο πλαίσιο που αντιμετωπίζει όλες τις πτυχές που συνιστούν μια τέτοια αποστολή με ένα UAV, συγκεκριμένα τον έλεγχο του οχήματος κατά τη διαδικασία της δειγματοληψίας και την ασφαλή μεταφορά του δείγματος. Κατά συνέπεια, σε περίπτωση ενός περιστατικού μόλυνσης νερού, το προτεινόμενο πλαίσιο μπορεί να αυτοματοποιήσει την αποστολή δειγματοληψίας νερού και να προσταλεί τους πρώτους ανταποκριτές αφού αποφεύγεται η επαφή τους με τη μολυσμένη περιοχή και, επομένως, με πιθανούς παθογόνους μικροοργανισμούς.

Πιο συγκεκριμένα, εξετάζουμε την περίπτωση κατά την οποία εμφανίζεται μία κατάσταση έκτακτης ανάγκης σχετικά με πιθανή μόλυνση σε υδάτινο περιβάλλον. Προκειμένου να ταυτοποιηθεί η ύπαρξη ή όχι παθογόνων μικροοργανισμών, υποδεικνύεται μια τοποθεσία όπου θα πρέπει να γίνει δειγματοληψία νερού. Δεδομένου ότι η διαφύλαξη της υγείας των πρώτων ανταποκριτών είναι υψίστης σημασίας, χρησιμοποιείται αντ' αυτού ένα UAV με αναρτημένο μέσω σχοινιού μηχανισμό για να πραγματοποιήσει τη δειγματοληψία. Αναλυτικότερα, τα ακόλουθα βήματα συνθέτουν την αποστολή δειγματοληψίας νερού:

- Οι πρώτοι ανταποκριτές επισημαίνουν μια επιθυμητή τοποθεσία δειγματοληψίας \mathbf{p}_{ref} μέσω ενός σταθμού εδάφους (ένα σύστημα που επιτρέπει στους χρήστες να παρακολουθούν την κατάσταση της πτήσης και να αλληλεπιδρούν με το UAV). Επιπλέον, υποδεικνύουν μια τελική θέση \mathbf{p}_f όπου πρέπει να πραγματοποιηθεί η

συλλογή του δείγματος για μετεπεξεργασία (συνήθως η θέση από την οποία απογειώνεται το UAV). Σημειώνεται ότι η αποστολή δειγματοληψίας νερού μπορεί εύκολα να επεκταθεί για πολλαπλές θέσεις δειγματοληψίας ή άλλες θέσεις όπου το UAV θα πρέπει να μεταφερθεί με ελάχιστη ταλάντωση, π.χ. για συλλογή εικόνων.

- Το UAV θα πρέπει να πλοηγηθεί από την αρχική του θέση \mathbf{p}_0 στην επιθυμητή θέση δειγματοληψίας \mathbf{p}_{ref} ελαχιστοποιώντας παράλληλα τις γωνίες η_L και τη γωνιακή ταχύτητα ω_L του σχοινιού, ώστε να διασφαλιστεί η ασφαλής μεταφορά τόσο του οχήματος όσο και του μηχανισμού.
- Καθώς το UAV πλησιάζει τη θέση δειγματοληψίας \mathbf{p}_{ref} , οι διαταραχές που δρουν στον βυθισμένο μηχανισμό και μεταφέρονται μέσω του σχοινιού στο UAV, δηλαδή η τάση $\|\mathbf{T}\|\mathbf{n}$, θα πρέπει να αναγνωριστούν και να απορριφθούν ώστε να επιτευχθεί ακριβής έλεγχος της θέσης του οχήματος κατά τη διαδικασία της δειγματοληψίας.
- Τελικά, μετά τη συλλογή του δείγματος, το UAV θα πρέπει παρομοίως να πλοηγηθεί προς την τελική θέση \mathbf{p}_f με ασφαλή τρόπο.

Προκειμένου να εκπληρωθούν οι προαναφερθέντες στόχοι, αξιοποιούνται τα σχήματα NMPC, που παρουσιάστηκαν στις Ενότητες 3.6.1 και 5.4.2, για την μεταφορά του μηχανισμού με ελάχιστη ταλαντωτική κίνηση του σχοινιού και την ακριβή σταθεροποίηση του UAV κατά τη δειγματοληψία αντίστοιχα. Η γωνιακή κίνηση και η τάση του σχοινιού είναι διαθέσιμα με τη βοήθεια των φίλτρων Kalman των Ενότητων 3.3.1 και 5.3.3.

Πειράματα σε Πραγματικό Περιβάλλον

Η αποτελεσματικότητα των επιμέρους μεθοδολογιών, δηλαδή των Kalman φίλτρων και των σχημάτων NMPC, για την εκτίμηση της κατάστασης και της τάσης του σχοινιού και τον έλεγχο του οχήματος κατά την αποστολή δειγματοληψίας νερού επαληθεύτηκε με το octocopter σε μία παραλία.

Σύμφωνα με την παραπάνω ανάλυση, 2 waypoints σχεδιάστηκαν μέσω του Mission Planner [59] (ένας σταθμός εδάφους συμβατός με τον αυτόματο πιλότο του οχήματος), όπως απεικονίζεται στην Εικόνα 5.10a. Το πρώτο σημείο βρίσκεται σε ένα ορισμένο ύψος πάνω από την επιθυμητή θέση δειγματοληψίας, ενώ το δεύτερο υποδεικνύει την τελική θέση όπου το δείγμα συλλέγεται από τους χειριστές. Αρχικά, το σχήμα NMPC της Ενότητας 3.6.1 οδήγησε το όχημα από την αρχική του θέση στο πρώτο waypoint ελαχιστοποιώντας παράλληλα τις ταλαντώσεις του σχοινιού, εκτιμώμενες σύμφωνα με την Ενότητα 3.3.1. Μόλις το UAV πλησίασε το πρώτο waypoint, το όχημα κατέβηκε και σταθεροποιήθηκε πάνω από την επιθυμητή θέση δειγματοληψίας, αναπτύσσοντας τον ελεγκτή θέσης της Ενότητας 5.4.2. Στη συνέχεια, το UAV ανέβηκε προς το πρώτο waypoint και τελικά μεταφέρθηκε με ασφάλεια στο τελικό σημείο. Η τρισδιάστατη διαδρομή του UAV απεικονίζεται στην Εικόνα 5.10b, όπου παρουσιάζονται τα 2 waypoints και η θέση δειγματοληψίας.

Η προαναφερθείσα αποστολή δειγματοληψίας νερού παρουσιάζεται στο παρακάτω βίντεο <https://youtu.be/13XI6a0Btp8>. Πρόσθετες αποστολές, που πραγματοποιήθηκαν στις εγκαταστάσεις της ΕΥΑΘ και στον Αξιό ποταμό αντίστοιχα, είναι διαθέσιμες στα <https://youtu.be/WyGah4iFNLO> και <https://youtu.be/5wHLSPGcLC0>.