Università degli studi di Modena e Reggio Emilia

Dipartimento di Scienze e Metodi dell'Ingegneria

Corso di Laurea Magistrale in Digital Automation Engineering

IMU-Only Bump-Resilient Control System for Quadrotors

Relatore:
Prof. Lorenzo Sabattini

Tesi di Laurea di:
Croci Nicolò

Anno Accademico 2024/2025

Summary

This thesis addresses the problem of preserving the stability and attitude of an unmanned aerial vehicle (UAV) when the vehicle is subject to undesired physical interactions with the surrounding environment relying exclusively on onboard inertial sensing provided by an inertial measurement unit (IMU). The primary objective is to design, implement, and validate a control architecture capable of detecting contact events, estimating the external wrench acting on the vehicle, and maintaining stability during and after the interaction, without resorting to external sensors such as LiDAR, motion capture systems, or force sensors.

Many UAV platforms lack on-board sensing capable of reliably detecting and characterizing unexpected collisions with small, thin or low-reflectivity obstacles (e.g. branches, thin wires or reflective surfaces), which prevents them from safely and effectively restoring equilibrium and the commanded attitude; this thesis addresses that gap by proposing an IMU-only control strategy that detects contact, preserves stability in confined spaces, and—through controlled sliding along the obstacle—enables local mapping of the environment so the vehicle can resume its mission once the obstacle is no longer present.

A practical motivation for this work is that common payload and cost constraints on small UAVs often rule out LiDAR, stereo vision and force sensors, while an IMU is present on virtually every platform; a robust IMU-only solution therefore increases deployability and reduces hardware requirements.

In cluttered and GPS-denied environments, small or elongated obstacles frequently go undetected by standard navigation sensors, causing loss of stability or mission aborts; an on-board contact-aware controller improves mission success rate and safety in these scenarios.

By enabling controlled sliding along obstacles, the proposed approach also supports opportunistic local mapping and environment reconstruction, which can be exploit to update path plans and avoid repeated collisions.

A further motivation is the need to operate safely in confined spaces (e.g., indoor inspection, collapsed buildings, forests), where the probability of contact is high and the ability to recover without external assistance is critical. Finally, reducing the dependency

on external sensing increases robustness to sensor failures and to degraded conditions such as dust, smoke or reflective surfaces that impair cameras and LiDAR.

To this end, a two-stage control strategy is proposed. In the first stage, referred to as *alignment*, the vehicle is rapidly reoriented so that the resultant contact force acts in a direction that minimizes destabilizing moments, in particular around the yaw axis. In the second stage, denoted as *sliding*, the controller enables a controlled interaction with the surface by regulating the tangential motion while enforcing a stiff attitude behavior and an impedance-like response in the contact plane.

At the core of the proposed architecture lies an external wrench estimator that exploits the UAV dynamic model together with accelerometer and gyroscope measurements from the IMU to infer external forces and moments acting on the vehicle body. The estimator is formulated as a model-based observer equipped with adaptive filtering techniques to cope with sensor noise and the fast dynamics associated with impulsive contact events. Low-latency contact detection mechanisms based on adaptive thresholds and confidence measures are employed to reliably trigger transitions between control modes.

The control logic is implemented as a high-level node within the MRS UAV framework running on ROS. The node processes IMU data and internally estimated attitude information, computes the external wrench estimate, manages mode transitions, and publishes thrust and body moment references to the underlying attitude controller. The proposed approach is validated in a high-fidelity simulation environment using the MRS simulator, incorporating rigid contact models. A comprehensive set of scenarios is considered, including single collisions at different approach velocities and different contact angles.

II

Simulation results demonstrate that the combined use of external wrench estimation and a modular control strategy significantly reduces post-impact attitude oscillations and shortens recovery times compared to a baseline controller designed for free-flight conditions. Stable behavior is maintained even in the presence of significant

disturbance moments, particularly around the yaw axis. The analysis also highlights critical dependencies of the approach, including the accuracy of the vehicle dynamic model and IMU calibration, as well as the influence of unmodeled aerodynamic effects in confined environments.

Overall, this thesis shows that a substantial increase in UAV resilience to undesired physical interactions can be achieved using IMU-only sensing, provided that robust estimation and contact-aware control strategies are employed. The results indicate the potential of the proposed approach for applications such as inspection, exploration, and navigation in confined or cluttered environments.

Summary - Italian Version

Questa tesi affronta il problema della preservazione della stabilità e dell'assetto di un veicolo aereo senza pilota (UAV) quando il mezzo è soggetto a interazioni fisiche indesiderate con l'ambiente circostante, facendo affidamento esclusivamente sulla sensoristica inerziale a bordo fornita da una unità di misura inerziale (IMU). L'obiettivo principale è progettare, implementare e validare un'architettura di controllo capace di rilevare eventi di contatto, stimare il wrench esterno agente sul veicolo e mantenere la stabilità durante e dopo l'interazione, senza ricorrere a sensori esterni quali LiDAR, sistemi di motion capture o sensori di forza.

Molte piattaforme UAV non dispongono di una sensoristica di bordo in grado di rilevare e caratterizzare in modo affidabile collisioni impreviste con ostacoli piccoli, sottili o a bassa riflettività (ad esempio rami, fili sottili o superfici riflettenti), impedendo così al veicolo di ristabilire in sicurezza ed efficacemente l'equilibrio e l'assetto comandato; questa tesi colma tale lacuna proponendo una strategia di controllo IMU-only che rileva il contatto, preserva la stabilità in spazi ristretti e — attraverso uno scorrimento controllato lungo l'ostacolo — consente la mappatura locale dell'ambiente, così che il veicolo possa riprendere la missione una volta che l'ostacolo non sia più presente.

Una motivazione pratica di questo lavoro è che i vincoli comuni di carico utile e di costo sui piccoli UAV spesso escludono LiDAR, visione stereo e sensori di forza, mentre un'IMU è presente su praticamente ogni piattaforma; una soluzione robusta basata solo su IMU aumenta quindi la possibilità di impiego e riduce i requisiti hardware.

In ambienti congestionati e privi di GPS, ostacoli piccoli o allungati vengono spesso rilevati con difficoltà dai sensori di navigazione standard, causando perdita di stabilità o aborti della missione; un controllore a bordo consapevole dei contatti migliora il tasso di successo della missione e la sicurezza in questi scenari.

Abilitando lo scorrimento controllato lungo gli ostacoli, l'approccio proposto supporta inoltre la mappatura locale opportunistica e la ricostruzione dell'ambiente, che possono essere sfruttate per aggiornare i piani di traiettoria ed evitare collisioni ripetute.

Un'ulteriore motivazione è la necessità di operare in sicurezza in spazi confinati (ad esempio ispezioni indoor, edifici crollati, foreste), dove la probabilità di contatto è elevata e la capacità di recuperare senza assistenza esterna è critica. Infine, ridurre la dipendenza

da sensori esterni aumenta la robustezza ai guasti sensoristici e a condizioni degradate quali polvere, fumo o superfici riflettenti che compromettono telecamere e LiDAR.

A tal fine, viene proposta una strategia di controllo in due fasi. Nella prima fase, denominata *alignment*, il veicolo viene rapidamente riorientato in modo che la forza di contatto risultante agisca lungo una direzione che minimizzi i momenti destabilizzanti, in particolare attorno all'asse di imbardata. Nella seconda fase, denominata *sliding*, il controllore consente un'interazione controllata con la superficie regolando il moto tangenziale ed imponendo un comportamento d'assetto rigido e una risposta di tipo impedenza nel piano di contatto.

Il cuore dell'architettura proposta è un estimatore del wrench esterno che sfrutta il modello dinamico dell'UAV insieme alle misure di accelerometro e giroscopio fornite dall'IMU per inferire le forze e i momenti esterni agenti sul corpo del veicolo. L'estimatore è formulato come un osservatore basato sul modello, dotato di tecniche di filtraggio adattativo per far fronte al rumore dei sensori e alle rapide dinamiche associate a eventi di contatto impulsivi. Meccanismi di rilevamento del contatto a bassa latenza, basati su soglie adattive e misure di confidenza, sono impiegati per attivare in modo affidabile le transizioni tra le modalità di controllo.

La logica di controllo è implementata come nodo di alto livello nel framework MRS UAV in esecuzione su ROS. Il nodo elabora i dati dell'IMU e le informazioni di assetto stimate internamente, calcola la stima del wrench esterno, gestisce le transizioni di modalità e pubblica riferimenti di spinta e momenti di corpo al controllore d'assetto sottostante. L'approccio proposto è validato in un ambiente di

simulazione ad alta fedeltà tramite il simulatore MRS, includendo modelli di contatto rigido. Viene considerato un insieme completo di scenari, incluse collisioni singole a diverse velocità di approccio e con diversi angoli di contatto.

I risultati di simulazione mostrano che l'uso combinato dell'estimazione del wrench esterno e di una strategia di controllo modulare riduce significativamente le oscillazioni d'assetto successive all'impatto e accorcia i tempi di recupero rispetto a un controllore

di riferimento progettato per il volo libero. Un comportamento stabile è mantenuto anche in presenza di momenti disturbanti significativi, in particolare attorno all'asse di imbardata. L'analisi evidenzia inoltre dipendenze critiche dell'approccio, tra cui l'accuratezza del modello dinamico del veicolo e della calibrazione dell'IMU, nonché l'influenza di effetti aerodinamici non modellati in ambienti confinati.

Nel complesso, questa tesi mostra che è possibile ottenere un notevole aumento della resilienza di un UAV verso interazioni fisiche indesiderate utilizzando esclusivamente la sensoristica IMU, purché si adottino strategie di stima robuste e un controllo consapevole del contatto. I risultati indicano il potenziale dell'approccio proposto per applicazioni quali ispezione, esplorazione e navigazione in ambienti confinati o congestionati.

Contents

CONTENTS

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) have become increasingly widespread in applications such as inspection, exploration, search and rescue, and infrastructure monitoring. Many of these missions require operation in complex, cluttered, and often GPS-denied environments, where the vehicle must navigate safely in close proximity to obstacles. In such scenarios, unexpected physical interactions with the environment—such as collisions with thin structures, branches, wires, or poorly detectable surfaces—are difficult to avoid and can severely compromise the stability and safety of the platform.

Traditional UAV navigation and control systems rely heavily on exteroceptive sensors such as LiDAR, cameras, or motion capture systems to perceive the environment and avoid obstacles. However, these sensing modalities present several limitations: they may fail in the presence of reflective or transparent surfaces, degrade under adverse conditions such as dust or smoke, and are often constrained by payload, power, and cost considerations, particularly in small aerial platforms. As a result, many UAVs operate without reliable means of detecting and characterizing contact events, leading to loss of stability, degraded performance, or mission failure when unexpected interactions occur.

In contrast, inertial measurement units (IMUs) are lightweight, low-cost, and ubiquitously available on virtually all UAV platforms. While typically used for state

estimation, IMU measurements also implicitly contain information about external disturbances acting on the vehicle. This observation motivates the development of control strategies that rely exclusively on onboard inertial sensing to detect contact events and react accordingly, without requiring additional hardware.

This thesis addresses the problem of preserving UAV stability and attitude in the presence of undesired physical interactions with the environment using IMUonly sensing. The core idea is to exploit the vehicle's dynamic model together with inertial measurements to estimate the external wrench acting on the UAV, and to use this information to drive a contact-aware control strategy. In particular, the proposed approach enables the vehicle not only to withstand collisions, but also to maintain controlled behavior during contact and recover its nominal flight condition afterward.

To this end, a two-stage control architecture is developed. Upon detection of a contact event, the controller first enters an *alignment* phase, in which the vehicle is reoriented to minimize destabilizing moments induced by the contact force, especially around the yaw axis. Subsequently, a *sliding* phase is activated, allowing the UAV to interact with the environment in a controlled manner by regulating tangential motion while maintaining a stiff attitude response and an impedance-like behavior in the contact plane. This strategy enables safe interaction with surfaces and supports local environment exploration through controlled sliding.

At the core of the proposed framework lies a model-based external wrench estimator that leverages IMU measurements to infer forces and moments acting on the vehicle. The estimator is designed to handle sensor noise and fast transient dynamics associated with impulsive contacts, and is coupled with a low-latency contact detection mechanism that triggers appropriate control mode transitions.

The proposed approach is implemented within the MRS UAV framework and validated in a high-fidelity simulation environment. The results demonstrate that incorporating contact awareness and external wrench estimation significantly improves post-impact stability and reduces recovery time compared to conventional controllers designed for free-flight operation. Moreover, the method enhances the UAV's capability to operate in confined and cluttered environments, increasing robustness and mission success rate without requiring additional sensing hardware.

Overall, this work contributes toward more resilient and adaptable UAV systems, capable of safely operating in real-world environments where contact is not an exception but an inevitable event.

1.1 Scenario

This thesis addresses the design, implementation, and experimental validation of a contact-aware control framework for small multirotor UAVs. The framework's goals are threefold: detect and survive unplanned impacts with previously unknown obstacles, intentionally exploit contact to traverse and trace an obstacle surface for mapping, and safely return to and resume the interrupted mission once the obstacle is cleared. A central, deliberate constraint of this work is minimal sensing: the vehicle is assumed to carry only an Inertial Measurement Unit (IMU) and a position source (e.g., GPS or motion capture position). Active proximity sensors or vision systems (LiDAR, cameras) are not used. This assumption is chosen to rigorously evaluate the external-wrench estimator and to demonstrate a control strategy that relies on force/moment information rather than direct geometric sensing.

The experimental validation is staged on linear impacts against a vertical, planar surface (a "wall"). The wall scenario yields repeatable, well-defined test conditions (incidence angle, approach speed) while keeping the presented algorithms general enough to be extended to curved or segmented surfaces in future work.

1.2 Problem Statement

Let the UAV state be $x(t)$ and the control input be $u(t)$. An external-wrench estimator provides the estimated wrench in the body frame,

$$\hat{w}_{\text{ext}}(t) = \begin{bmatrix} \hat{F}_x & \hat{F}_y & \hat{F}_z & \hat{M}_x & \hat{M}_y & \hat{M}_z \end{bmatrix}^{\top},$$

where $\hat{F}$ denotes force components and $\hat{M}$ denotes moment components.

The control and mode-management problem addressed in this thesis is: design a policy (estimator + controller + mode manager) such that, for any unmodeled collision event with a rigid obstacle, the system ensures

1. reliable collision detection with bounded latency;

2. a safe transition to a contact-aware control mode (Alignment and/or Sliding) that stabilizes attitude and enforces desired contact behavior;

3. controlled traversal (sliding) along the surface to acquire a spatial trace suitable for mapping;

4. detection of obstacle termination and a staged, safe recovery back to nominal flight with bounded deviation from the pre-impact mission.

1.3. VEHICLE MODEL

1.3 Vehicle Model

There exists a wide variety of commercial UAVs with very different mechanical designs and mission profiles. For clarity and reproducibility, this thesis restricts the considered class to *octocopter*. Consequently, throughout the document the generic term “drone” refers specifically to a quadrotor UAV. Octocopter commonly adopt either a “+” or “×” motor configuration depending on the chosen body-frame orientation relative to the motors. In this work all controller derivations, coordinate definitions and experimental descriptions are given with respect to the “+” configuration; results for the “×” configuration are equivalent up to a 45° rotation of the body frame.

The specific vehicle used in experiments is described in Table 1.1 where mass m, inertia matrix J, arm length d, drone geometry, and numerical parameters are reported.

Table 1.1: Main physical parameters of the UAV platform.

Parameter	Symbol	Value
Number of motors	n_{motors}	8
Mass	m	7.5 kg
Arm length	d	0.20 m
Body height	h	0.20 m

The vehicle is modeled as a rigid body actuated by eight coaxial propellers. Let I denote the inertial frame and B the body-fixed frame attached to the vehicle center of

mass. The translational and rotational dynamics used in controllers are the standard octocopter rigid-body model:

$$m\ddot{p} = mge_3 + R_B^I(-Te_3) + F_{ext}, \quad (1.1)$$

$$J\dot{\omega} + \omega \times J\omega = \tau_{ctrl} + M_{ext}, \quad (1.2)$$

where $p \in \mathrm{R}^3$ is the vehicle position in I, $R_B^I \in SO(3)$ is the rotation matrix from body to inertial frame, $T = \sum_{i=1}^{8} f_i$ is the total body-axis thrust generated by propellers (positive along the negative body z-axis in this sign convention), $\tau_{ctrl} \in \mathrm{R}^3$ is the control torque vector produced by differential thrust, and F_{ext}, M_{ext} are external force and moment applied to the body (contact + aerodynamic disturbances). The

mapping from motor inputs to T and τ_{ctrl} follows the standard allocation matrix for a octocopter and is detailed in Equation 2.4.

The above equations provide the reference dynamical model used by the estimator and the controllers; additional modeling of the protective cage and distributed contact is described in the simulation and implementation chapters.

For the theoretical development and experimental setup the following assumptions are made and held throughout the thesis:

1. Identical actuators and negligible actuator dynamics. All propellers and motors are assumed identical and the actuator dynamics are considered sufficiently fast that they can be approximated as instantaneous with respect to the control loop (i.e., thrust commands are implemented without significant lag). This simplifies control allocation and is consistent with common octocopter modeling in the literature.

2. Rigid-body assumption. The vehicle is modeled as a rigid body. Elastic deformations of the airframe, arms or cage are considered negligible for the control timescales of interest and are not included in the main dynamical model.

3. Post-impact functionality. It is assumed that after the considered impacts the vehicle remains electrically functional and that mechanical components are not irreversibly damaged: motors, propellers and sensing electronics operate

nominally and the vehicle is capable of actuating the recovery maneuvers. This assumption is enforced experimentally by using a protective cage that absorbs a portion of the collision energy and prevents structural breakage.

4. Thrust independence from relative airspeed. The thrust produced by each propeller is assumed to be independent of the local airspeed around the propeller (i.e., aerodynamic inflow effects due to vehicle motion are neglected in the mapping from motor command to thrust). Under this assumption thrust is a direct function of motor command only. Any aerodynamic effects induced by proximity to the wall or by induced flow are treated as part of the external

1.3. VEHICLE MODEL

wrench $F_{\text{ext}}, M_{\text{ext}}$ and estimated by the external-wrench estimator.

These assumptions are stated explicitly to delimit the scope of the results and to facilitate reproducibility.

1.4 Mission pipeline and mode-management overview

The system is organized as a linear pipeline: *estimator → mode manager → contact-aware controllers → nominal mission controller*.

The mode manager implements thresholding, hysteresis and minimal dwell times to ensure robust switching.

A concise operational description:

1. Nominal mission. The UAV follows the planned trajectory under the standard mission controller (MPC or geometric SO(3) controller).

2. Collision detection. The estimator produces $\hat{w}_{\text{ext}}$. A collision is declared when conditions on $\|\hat{F}\|$ and/or $\|\hat{M}\|$ persist for a detection window t_{det}.

3. Alignment entry. On confirmed detection the manager transitions to ALIGNMENT. The nominal controller is demoted and the alignment controller regulates attitude and contact force.

4. Alignment. Stabilize attitude, enforce a reference normal force $F_{n,\text{ref}}$, and minimize destabilizing moments using rate-limited reference shaping to avoid jerk.

5. Sliding. If traversal is desired or tangential wrench components exceed thresholds, the manager activates SLIDING. Sliding permits controlled tangential motion while compensating destabilizing moments (notably $\hat{M}_z$).

6. End-of-obstacle detection. Monitor wrench trends or position to detect loss of contact or surface termination.

7. Recovery. Gradually release contact constraints (progressive stiffness reduction), restore hover, and return control to the nominal mission controller.

1.5 Limitations, generalizability and roadmap

This scenario deliberately restricts sensing to emphasize the estimator and control logic. The absence of direct geometric sensing imposes limits on mapping fidelity: contact-based mapping relies on dead-reckoning and is subject to drift. Near-wall aerodynamic phenomena are treated implicitly by the estimator and are not explicitly modeled by the control laws; this choice simplifies controller design but is a documented limitation.

Nevertheless, the proposed architecture (external-wrench estimator → mode manager → Alignment/Sliding/Recovery controllers) is modular and extensible. The restriction to planar vertical obstacles is a pragmatic experimental choice that does not prevent the methodology from being generalized to curved surfaces or multi-contact scenarios.

The estimation of the external wrench is presented in Section 2.1, the collision detection strategy and equilibrium recovery in Section 2.2, the alignment and sliding phase in Section 2.3 and 2.4, the safety logic in Section 2.5, and the end of obstacle detection in Section 2.6.

The scenario defined in this chapter provides a reproducible experimental playground to evaluate contact-aware control under challenging, sensor-poor conditions. By constraining experiments to controlled linear impacts against a vertical surface, the thesis can precisely quantify detection, stabilization, traversal and recovery performance while preserving methodological extensibility. The following chapters develop the estimator, software architecture and detailed control strategies introduced here, and present a rigorous experimental validation against the metrics and test cases summarized above.

1.6 Related Works

The literature relevant to this thesis spans several intersecting areas: IMU-based collision identification and external-wrench estimation, collision reaction and recovery strategies, wall-interaction and wall-climbing methods, foundational geometric control on $\mathrm{SE}(3)$, and the UAV software platforms used for real-world validation.

Collision detection and external-wrench estimation. A number of works address the estimation of external forces and moments from onboard sensors and the consequent detection of collisions. As shown in [1], Tomic propose a comprehensive external-wrench estimator and reflex reaction scheme for flying robots, providing useful design patterns for IMU-driven detection. Earlier, Tomić and Haddadin [2] present techniques to separate aerodynamic and contact forces, which are important when distinguishing contact-induced signals from environmental disturbances. Wang et al. [3] introduce the “Air Bumper” framework for collision detection and reactive control on MAVs, offering a recent practical baseline. Hughes and Lyons [4] demonstrate IMU-only wall detection via data classification, directly relevant to this work’s sensor-constrained scenario.

Reaction and recovery. Patnaik et al. [5] describe a collision recovery controller for a foldable octocopter, highlighting recovery strategies that combine mechanical design and control to restore stable flight after impacts. Such approaches inform the design tradeoffs between mechanical mitigation and control effort.

Wall interaction and climbing. A separate line of research explores active wallsticking and climbing, often via mechanical modifications (tilt rotors, rotor offsets) to sustain vertical contact [6, 7, 8]. Although these methods are outside the hardware scope of this thesis, they provide alternative paradigms for maintaining contact and attaining traversal, which are useful for comparative discussion.

Geometric control and platform integration. The foundational results on geometric tracking control on $\mathrm{SE}(3)$ by Lee et al. [9] underpin the attitude and position controllers used as nominal baselines. The MRS UAV software stack [10] serves as the integration platform for the estimator and mode-manager developed here, providing reproducibility and real-world deployment capabilities.

Gap analysis. While the cited works cover elements of detection, reaction, or wall interactions, none presents a complete, experimentally validated pipeline that (i) performs IMU-only collision identification, (ii) engages a contact-aware alignment/sliding behavior to map the obstacle surface, and (iii) executes a staged recovery and resume of the pre-impact mission. This thesis addresses that gap by developing and evaluating a modular estimator + mode-manager + contact controllers pipeline

tailored to IMU-only operation.

Table 1.2: Summary of key related work and relevance to this thesis.

Reference	Core contribution	Relevance / limitation
Tomić et al. (2017)	External-wrench estimation	Strong methodological baseline for IMU-based wrench estimation; used as primary reference.
Tomić & Haddadin (2015)	Separation of aerodynamic and contact forces	Useful techniques for disentangling disturbance vs contact; may require additional modelling.
Wang et al. (2023)	Air Bumper: collision detection + reaction	Recent practical framework for MAV collision reaction; check sensor assumptions.
Hughes & L ɔns (2021)	IMU-based wall detection (classification)	Directly relevant to IMUonly detection; often MLbased—generalization issues.
Patnaik et al. (2021)	Collision recovery for foldable quadrotor	Recovery control strategies; hardware-specific assumptions.
Myeong et al. (2015, 2019), Lee et al. (2021)	Wall-sticking / climbing robots	Mechanical solutions for sustained contact; outside hardware scope but useful for comparison.
Lee et al. (2010)	Geometric tracking on SE(3)	Theoretical foundation for attitude control used as nominal baseline.
Baca et al. (2021)	MRS UAV system	Chosen software/hardware platform for integration and reproducibility.

Chapter 2

Implementation of a Contact-Aware Control Framework

One of the primary objectives of this work is to detect collisions with previously unknown objects using only onboard inertial sensing. For this reason, the proposed approach follows the methodology commonly adopted in the literature on IMU-based contact detection and estimation [1].

In practice, the onboard inertial measurement unit provides ten scalar measurements, together with their associated uncertainty (covariance): a four-element unit quaternion representing the vehicle orientation, three linear accelerations expressed in the body frame, and three angular velocity components.

These measurements, combined with an explicit noise and bias model, constitute the only direct inputs to both the external-wrench estimator and the collision detection logic developed in this thesis.

$$y(t) = \begin{bmatrix} q(t) \\ a(t) \\ \omega(t) \end{bmatrix} \in \mathbb{R}^{10}, \qquad q(t) \in \mathbb{S}^3,\ a(t), \omega(t) \in \mathbb{R}^3, \tag{2.1}$$

with measurement covariance

$$R_y(t) = \mathrm{diag}(R_q, R_a, R_\omega), \tag{2.2}$$

where $R_q \in \mathbb{R}^{4\times4}$, $R_a, R_\omega \in \mathbb{R}^{3\times3}$.

A collision typically produces a rapid change not only in linear acceleration but also in angular rate, a signature that is particularly evident during lateral contacts with the vehicle structure. For this reason, relying solely on raw IMU measurements is insufficient to reliably discriminate between contact-induced disturbances and variations caused by nominal control actions or environmental

effects. Instead, the inertial measurements must be compared against the accelerations and angular-rate evolution expected from the control inputs applied to the vehicle.

To this end, the actuator allocation matrix is employed. This matrix defines the mapping between the squared rotor angular velocities and the resulting total thrust and control moments acting on the UAV.

The allocation matrix, of dimension $4\times N_{\text{motors}}$, encodes the thrust and torque coefficients of the propulsion system, as well as the geometric configuration of the platform. Given the vector of rotor angular speeds $\Omega = [\omega_1,\ldots,\omega_N]^\top$, the predicted control input vector is obtained as

$$\mathbf{u}_{\text{pred}} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \\ T \end{bmatrix} = A\Omega^{\circ 2},\ A \in \mathrm{R}^{4\times N}. \quad (2.3)$$

where $\Omega^{\circ 2}$ denotes the element-wise square of the rotor speeds and $\mathbf{u}_{\text{pred}}$ collects the total thrust and body moments.

Using the predicted control inputs and the current attitude estimate, the expected translational and rotational dynamics of the UAV can be reconstructed through the rigid-body equations of motion. The discrepancy between these modelpredicted quantities and the actual IMU measurements constitutes a residual signal, which represents the effect of unmodeled external forces and moments acting on the vehicle. Large and abrupt residuals, inconsistent with model uncertainty and sensor noise, are therefore interpreted as the presence of an external contact and are exploited by the external wrench estimator and the collision detection logic developed in this work.

Let $\Omega = [\omega_1,\ldots,\omega_N]^\top$ be the vector of rotor angular speeds and denote $\Omega^{\circ 2} = \left[\omega_1^2,\ldots,\omega_N^2\right]^\top$ as the elementwise square, for a standard octocopter in "+" configuration (8 rotors, numbered 1..8), a typical allocation is:

$$A = \begin{bmatrix} k_f l\sigma_{x1}c_1 & k_f l\sigma_{x2}c_2 & k_f l\sigma_{x3}c_3 & k_f l\sigma_{x4}c_4 & k_f l\sigma_{x5}c_5 & k_f l\sigma_{x6}c_6 & k_f l\sigma_{x7}c_7 & k_f l\sigma_{x8}c_8 \\ k_f l\sigma_{y1}s_1 & k_f l\sigma_{y2}s_2 & k_f l\sigma_{y3}s_3 & k_f l\sigma_{y4}s_4 & k_f l\sigma_{y5}s_5 & k_f l\sigma_{y6}s_6 & k_f l\sigma_{y7}s_7 & k_f l\sigma_{y8}s_8 \\ k_m k_f \sigma_{z1} & k_m k_f \sigma_{z2} & k_m k_f \sigma_{z3} & k_m k_f \sigma_{z4} & k_m k_f \sigma_{z5} & k_m k_f \sigma_{z6} & k_m k_f \sigma_{z7} & k_m k_f \sigma_{z8} \\ k_f & k_f & k_f & k_f & k_f & k_f & k_f & k_f \end{bmatrix}$$

(2.4)

In (2.4), τ_x, τ_y, and τ_z denote the roll, pitch, and yaw moments expressed in the body frame, while F represents the total thrust force acting along the body z axis.

The coefficient k_f is the thrust constant mapping the squared motor angular velocity ω_i^2 to a force contribution, while k_m is the torque constant relating thrust to the aerodynamic reaction torque generated by each propeller. The parameter l denotes the distance between each motor and the vehicle center of mass.

The terms c_i and s_i are geometry-dependent coefficients that encode the planar placement of the i-th motor with respect to the body frame axes and are typically expressed as direction cosines. The sign coefficients $\sigma_{xi},\sigma_{yi} \in \{-1,+1\}$ determine the direction of the roll and pitch moments generated by each motor thrust.

The yaw contribution is governed by the sign coefficient $\sigma_{zi} \in \{-1,+1\}$, which depends on the rotation direction of the i-th propeller and captures the sign of the aerodynamic reaction torque.

Finally, the last row reflects the fact that all motors contribute positively to the total thrust magnitude. This symbolic formulation mirrors the onboard implementation, where a geometry-dependent allocation template is scaled by the physical parameters of the propulsion system.

So that $T = k_f \sum_i \omega_i^2$ and $\tau_z = k_m(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$, etc. The predicted translational acceleration follows from the translational dynamics:

$$m\hat{a}_{\text{pred}} = mge_3 + R_{\text{BI}}(-Te_3), \tag{2.5}$$

and the rotational dynamics give the predicted torque input:

$$J\dot{\omega}_{\text{pred}} = \tau_{\text{ctrl}} + M_{\text{ext}} - \omega \times J\omega, \tag{2.6}$$

where $\tau_{\text{ctrl}} = [\tau_x,\tau_y,\tau_z]^\top$ is taken from $\mathbf{u}_{\text{pred}}$. Define the residuals

$$r_a(t) = a_{\text{meas}}(t) - \hat{a}_{\text{pred}}(t), \qquad r_\omega(t) = \omega_{\text{meas}}(t) - \omega_{\text{pred}}(t), \tag{2.7}$$

which are the signals fed to the external-wrench estimator / collision detector. Statistically grounded detection uses the residuals together with their covariances (see Chapter X on estimation).

2.1 External Wrench Estimation

At this point, it is necessary to clarify the estimation strategy adopted for reconstructing the external wrench acting on the vehicle. The literature does not present mutually exclusive methodologies, but rather two complementary formulations of the same estimation problem, each relying on different measured quantities and system assumptions.

The first formulation is based on the observation of the system generalized momentum and exploits velocity information to reconstruct the external wrench through a residual-based observer. This momentum-based approach provides a filtered estimate of the external forces and torques and exhibits favorable stability properties. However, it requires accurate knowledge or estimation of the vehicle translational and angular velocities, which for flying robots typically demands exteroceptive sensors and sensor-fusion algorithms.

An alternative formulation expresses the external wrench directly as a function of the measured accelerations and known control inputs. This acceleration-based approach is particularly attractive for aerial robots, since linear accelerations and angular velocities are readily available from onboard inertial measurement units. Nevertheless, the direct use of acceleration signals introduces sensitivity to measurement noise and numerical differentiation, especially in the estimation of external torques.

Due to these complementary characteristics, neither formulation alone is fully suitable for reliable external wrench estimation using proprioceptive sensing only. For this reason, a hybrid estimation scheme is adopted, combining the accelerationbased formulation for external force estimation with the momentum-based formulation for external torque estimation. This choice allows the reconstruction of the full external wrench while relying exclusively on onboard inertial sensing.

We consider the rigid-body dynamics of the multicopter in compact form

$$M\,\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g = J^{\top}\tau + \tau_e, \tag{2.8}$$

where $\nu = [v^{\top}, \omega^{\top}]^{\top}$ is the generalized velocity (translational velocity v and angular rate ω in the body frame), M is the (constant) mass–inertia matrix, $C(\nu)$ and $D(\nu)$ collect

Coriolis/centrifugal and damping terms, g is gravity, $J^{\top}\tau$ are the generalized inputs produced by the rotors, and $\tau_e = [f_e^{\top},\ m_e^{\top}]^{\top}$ is the unknown external wrench (force f_e and moment m_e, expressed in the body frame). For compactness denote

$$N(\nu) := C(\nu)\nu + D(\nu)\nu + g. \tag{2.9}$$

In the following, the two estimator formulations commonly used in the literature are presented, and the hybrid estimator adopted in this work is derived.

Momentum-Based Estimation We start from the rigid-body dynamics written in generalized form:

$$M\dot{\nu} + C(\nu)\nu + g = J^{\top}\tau + \tau_e, \tag{2.10}$$

where:

- $\nu = \begin{bmatrix} v \\ \omega \end{bmatrix}$ is the generalized velocity (translational velocity v and angular rate ω, expressed in the body frame);
- $M \in \mathrm{R}^{6\times6}$ is the (constant) mass–inertia matrix of the UAV;
- $C(\nu)$ collects Coriolis/centrifugal terms, g is gravity;
- $J^{\top}\tau$ denotes the generalized input produced by the rotors (known from motor commands and the allocation matrix);
- $\tau_e = \begin{bmatrix} f_e \\ m_e \end{bmatrix}$ is the unknown external wrench (force and moment) to be estimated.

Introduce the compact notation for the known nonlinear terms $N(\nu) := C(\nu)\nu + g.$ The *generalized momentum* of the vehicle is defined as (2.11)

$$p(t) = M\,\nu(t). \tag{2.12}$$

Because M is assumed constant for the rigid UAV, differentiation yields

$$\dot{p} = M\dot{\nu} \tag{2.13}$$

Substituting (2.13) into the dynamics (2.10) gives the momentum evolution

$$\dot{p} = J^{\top}\tau + \tau_e - N(\nu). \tag{2.14}$$

Equations (2.14) and (2.11) make explicit the relation between the measurable/known terms $J^{\top}\tau$, $N(\nu)$ and the unknown wrench τ_e. The goal of the momentumbased observer is to reconstruct τ_e while avoiding explicit numerical differentiation of noisy signals.

Integrating (2.14) from 0 to t provides an equation that relates time-integrals of measurable terms to the integral of the unknown wrench:

$$\int_0^t \tau_e(\sigma)\,d\sigma \;=\; p(t) - p(0) \;-\; \int_0^t \big(J^{\top}\tau - N(\nu)\big)(\sigma)\,d\sigma. \tag{2.15}$$

The left-hand side contains the integral of the unknown wrench and is therefore not computable directly. The right-hand side, however, is composed of terms that can be formed from available quantities (state estimate, motor commands and the known model).

To obtain a causal estimator that avoids differentiating p, we introduce an integral residual $\rho(t) \in \mathrm{R}^6$ defined by

$$\rho(t) \;=\; K_I\left[\, p(t) \;-\; \int_0^t \big(J^{\top}\tau - N(\nu) + \rho(\sigma)\big)\,d\sigma \;-\; p(0) \right], \tag{2.16}$$

where $K_I \in \mathrm{R}^{6\times 6}$ is a diagonal, positive-definite gain matrix. Two comments on (2.16):

1. the term ρ appears inside the integral; this self-referential (autonomous) definition is a standard device that yields a simple linear residual dynamics after differentiation;

2. the presence of $p(0)$ preserves consistency with the integral identity $\int_0^t \dot{p}\,d\sigma = p(t) - p(0)$ and must be handled at initialization (see implementation notes below).

Differentiate (2.16) with respect to time. Using (2.14) and the Fundamental Theorem of Calculus we obtain:

$$\begin{aligned}\dot{\rho}(t) &= K_I\Big[\dot{p}(t) - \big(J^\top\tau(t) - N(\nu(t)) + \rho(t)\big)\Big] \\ &= K_I\Big[\big(J^\top\tau + \tau_e - N(\nu)\big) - \big(J^\top\tau - N(\nu) + \rho\big)\Big] \\ &= K_I\big(\tau_e - \rho\big).\end{aligned} \tag{2.17}$$

Equation (2.17) is a linear, exponentially stable, first-order filter driven by the true external wrench τ_e. Therefore:

- $\rho(t)$ is a low-pass filtered reconstruction of $\tau_e(t)$;
- it is natural to define the momentum-based estimate of the external wrench as

$$\widehat{\tau}_e^{(p)}(t) \;:=\; \rho(t). \tag{2.18}$$

Remarks and practical points.

- Do not set $p = \tau_e$. The physical momentum p and the external wrench τ_e are distinct quantities. The residual ρ (not p) is the filtered estimator that converges to τ_e.
- Initialization: the term $p(0)$ in (2.16) introduces an initialization bias. In practice either (i) $p(0)$ is measured/estimated at filter start or (ii) a short initialization transient is discarded when reporting steady-state estimates.
- Sampling / discrete-time implementation: implement the integral in (2.16) with a numerical quadrature consistent with your control/estimation timestep. It is usually preferable to implement the equivalent differential form (2.17) in discrete time:

  $$\rho_{k+1} = \rho_k + T_s\, K_I\big(\tau_{e,k} - \rho_k\big),$$

  where $\tau_{e,k}$ is replaced by the computable term obtained from $p^{\cdot} + N - J^\top\tau$ if available, or by a predictor. In practice, since τ_e is unknown, one implements (2.17) in the form $\rho^{\cdot} = K_I(-\rho)+K_I$ (measurable driving term) consistent with

(2.16).

Acceleration-Based Estimation We consider again the rigid-body dynamics in generalized form:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g = J^{\top}\tau + \tau_e, \tag{2.19}$$

where $\nu = [v^{\top}, \omega^{\top}]^{\top}$ is the generalized velocity, M is the constant mass–inertia matrix, $C(\nu)$ and $D(\nu)$ collect Coriolis/centrifugal and linear damping terms respectively, g is gravity, $J^{\top}\tau$ denotes the generalized input from the rotors and $\tau_e = [f_e^{\top},\ m_e^{\top}]^{\top}$ is the unknown external wrench.

For compactness define the known nonlinear term

$$N(\nu) := C(\nu)\nu + D(\nu)\nu + g. \tag{2.20}$$

Rearranging (2.19) yields the algebraic (raw) expression for the external wrench:

$$\bar{\tau}_e = M\dot{\nu} + N(\nu) - J^{\top}\tau. \tag{2.21}$$

The quantity $\bar{\tau}_e$ is an instantaneous (but noisy) estimate of the external wrench obtained by substituting measured/estimated values in the dynamic equation. In particular, the translational inertial term $M_t\dot{v}$ can be related to the accelerometer measurement after appropriate rotation and gravity compensation (see below).

Let a_{meas} be the accelerometer measurement expressed in the body frame. Neglecting sensor bias for brevity, the ideal relation is

$$a_{\text{meas}} = R_{\mathcal{I}}^{\mathcal{B}}(\ddot{r} - ge_3), \tag{2.22}$$

where R_{I}^{B} is the rotation from the inertial to the body frame and $\ddot{r}$ is the inertial translational acceleration. Thus, the translational component of $\bar{\tau}_e$ can be formed as

$$\bar{f}_e = M_t\, a_{\text{meas}} + \underbrace{(\text{compensation terms})}_{\text{if } N \text{ retained}} - f_{\text{cmd}}, \tag{2.23}$$

where M_t is the translational mass (the appropriate block of M), f_{cmd} is the expected thrust produced by the rotors (thrust component of $J^\top \tau$), and "compensation terms" denote contributions from $N(\nu)$ if these are explicitly accounted for (see discussion below).

The raw quantity τ_e contains high-frequency noise due to sensor measurements and numerical differentiation (if $\dot{\nu}$ is computed numerically). To obtain a robust estimate we apply a first-order stable filter to τ_e:

$$\dot{\hat{\tau}}_e = K_I\left(\tau_e - \hat{\tau}_e\right), \tag{2.24}$$

where $K_I \in \mathrm{R}^{6\times 6}$ is a positive-definite diagonal gain matrix. In practice, implement the translational and rotational blocks separately and choose different gains for force and torque (e.g. $K_I = \mathrm{diag}(K_{I,f}, K_{I,m})$).

Substituting the algebraic expression (2.21) into the filter dynamics (2.24) yields the implementation-ready form of the acceleration-based estimator:

$$\dot{\hat{\tau}}_e = K_I\left(M\dot{\nu} + N(\nu) - J^\top \tau - \hat{\tau}_e\right). \tag{2.25}$$

The acceleration-based estimator provides a physically interpretable and practically implementable route to recover the translational component of the external wrench, exploiting directly the onboard accelerometer after appropriate rotation and gravity compensation. Its performance depends critically on (i) the treatment of the nonlinear term $N(\nu)$ (retain and compute it from state estimates in dynamic flight, or approximate/neglect it near hover), (ii) accurate bias calibration of the accelerometer, and (iii) tight timestamp alignment between IMU, attitude/pose estimates and motor-command readings.

Hybrid Estimation The unknown external wrench acting on the vehicle is $\tau_e = \begin{bmatrix} f_e \\ m_e \end{bmatrix}$, containing the translational force f_e and the rotational moment m_e. The estimator design is driven by a pragmatic observation: different parts of the wrench are best reconstructed from different measurable quantities available onboard. In particular:

- the translational acceleration is measured directly (with gravity compensation) by the IMU accelerometer, hence an acceleration-based estimator is naturally suited to reconstruct f_e with good bandwidth;

- the rotational moment is more robustly reconstructed by exploiting the rotational momentum $p_\omega = I\omega$ (momentum-based observer),since numerical differentiation of ω to obtain $\dot{\omega}$ is typically too noisy.

For these reasons we design a hybrid estimator that uses the acceleration-based method for the force block and the momentum-based method for the moment block. Below we derive each block (aligned with the notation of Section 2) and show the final estimator form and implementation notes.

The translational component of $\overline{\tau}_e$ is obtained from the accelerometer (after rotation to the body frame and gravity compensation). Denote the body-frame accelerometer measurement by a_{meas} so that (ideal)

$$M_t \dot{v} \approx M_t a_{\text{meas}},$$

where M_t is the translational mass (block of M). The instantaneous (translational) raw force becomes

$$f_e = M_t a_{\text{meas}} + N_f(v) - f_{\text{cmd}},$$

with f_{cmd} the thrust component of $J^\top \tau$ computed from motor commands via the allocation matrix, and $N_f(v)$ the translational part of $N(v)$ if retained.

Because $\overline{\tau}_e$ is noisy, we apply a first-order stable filter:

$$\dot{\hat{\tau}}_e = K_I\left(\overline{\tau}_e - \hat{\tau}_e\right), \tag{2.26}$$

and partition $K_I = \text{diag}(K_{I,f}, K_{I,m})$. The force block (component) is therefore implemented as

$$\dot{\hat{f}}_e = K_{I,f}\left(M_t a_{\text{meas}} + N_f(\nu) - f_{\text{cmd}} - \hat{f}_e\right), \tag{2.27}$$

or, equivalently, in integral form

$$\hat{f}_e(t) = \hat{f}_e(0) + \int_0^t K_{I,f}\left(M_t a_{\text{meas}}(\sigma) + N_f(\nu(\sigma)) - f_{\text{cmd}}(\sigma) - \hat{f}_e(\sigma)\right) d\sigma. \tag{2.28}$$

The external torque is estimated using a momentum-based residual. As shown previously, the momentum-based estimator output for the rotational channel is denoted ρ_m and the external-torque estimate is defined as

$$\widehat{m}_e := \rho_m. \tag{2.29}$$

Recall the general integral residual previously defined for the generalized momentum (Eq. (2.16)):

$$\rho = K_I \Big[p - \int_0^t \big(J^\top \tau - N(\nu) + \rho \big)\, d\sigma - p(0) \Big].$$

We now specialize this expression to the rotational (angular-momentum) component. This simply replaces the generalized momentum p with the rotational momentum p_ω.

For a rigid body the rotational momentum is

$$p_\omega = I\,\omega, \tag{2.30}$$

hence $p_\omega(0) = I\omega(0)$. The rotational residual is defined as

$$\rho_m(t) = K_{I,m} \Big[p_\omega(t) - \int_0^t \big(\tau_{\text{ctrl,rot}} - N_\omega(\nu) + \rho_m(\sigma) \big)\, d\sigma - p_\omega(0) \Big], \tag{2.31}$$

where $K_{I,m} \in \mathrm{R}^{3\times3}$ is the diagonal gain of the torque block and $N_\omega(\nu)$ denotes the rotational portion of $N(\nu)$. (Equation (2.31) is the integral definition of the residual specialized to the angular channel.)

Although the integral expression formally contains the initial angular momentum term $p_\omega(0) = I\omega(0)$, this term is typically omitted or compensated in practice since it only produces an initialization-dependent bias and does not change the residual dynamics or the estimator convergence properties. Differentiate (2.31) and use the rotational dynamics

$$\dot{p}_\omega = \tau_{\text{ctrl,rot}} + m - N_\omega(\nu), \tag{2.32}$$

where m denotes the external torque. Substituting into the derivative of the residual yields

$$\begin{aligned}\dot{\rho}_m(t) &= K_{I,m}\Big[\dot{p}_\omega(t) - \big(\tau_{\text{ctrl,rot}} - N_\omega(\nu) + \rho_m(t)\big)\Big] \\ &= K_{I,m}\Big[\big(\tau_{\text{ctrl,rot}} + m - N_\omega(\nu)\big) - \big(\tau_{\text{ctrl,rot}} - N_\omega(\nu) + \rho_m\big)\Big] \\ &= K_{I,m}\big(m(t) - \rho_m(t)\big).\end{aligned} \tag{2.33}$$

Equation (2.33) shows that ρ_m acts as a first-order low-pass reconstruction of the true external torque m.

To make model terms explicit, write the typical decomposition of the rotational nonlinearity:

$$N_\omega(\nu) = (I\omega) \times \omega + m_g + \tau_d(\omega), \tag{2.34}$$

where m_g is the gravitational torque due to a centre-of-gravity offset and $\tau_d(\omega)$ collects rotational damping/aerodynamic moments if present.

Substituting (2.34) into (2.31) and using $p_\omega = I\omega$ gives the explicit integral representation:

$$\rho_m(t) \;=\; K_{I,m}\Big[\, I\omega(t) - \int_0^t \big(\tau_{\text{ctrl,rot}} - \big((I\omega)\times\omega + m_g + \tau_d(\omega)\big) + \rho_m(\sigma)\big)\, d\sigma - I\omega(0)\Big]. \tag{2.35}$$

Here $\tau_{\text{ctrl,rot}}$ is known (computed from motor-speeds and the allocation matrix), so the known terms inside the integrand can be compensated.

Rearranging the integrand to group the unknown torque and the modelled rotational terms yields the implementation-friendly expression

$$\rho_m(t) \;=\; K_{I,m}\Big[\, I\omega(t) - \int_0^t \big(m(\sigma) + m_g(\sigma) + (I\omega(\sigma)) \times \omega(\sigma) - \rho_m(\sigma)\big)\, d\sigma\Big], \tag{2.36}$$

where we have adopted the short symbol m for the external torque.

Finally, substituting $\rho_m = \widehat{m}_e$ (definition (2.29)) gives the final integral form used in this chapter (signs matched exactly to the thesis convention):

$$\widehat{m}_e(t) \;=\; K_{I,m}\Big[\, I\omega(t) \;-\; \int_0^t \big(m(\sigma) + m_g(\sigma) + (I\omega(\sigma)) \times \omega(\sigma) - \widehat{m}_e(\sigma)\big)\, d\sigma\Big]. \tag{2.37}$$

Differentiating (2.37) recovers the residual dynamics

$$\dot{\widehat{m}}_e(t) \;=\; K_{I,m}\big(m(t) - \widehat{m}_e(t)\big),$$

but, as explained above, the integral/residual implementation (or its discrete-time update) is preferred to avoid direct numerical differentiation of ω.

Therefore, the final form of the external-wrench estimator implemented in this work is given in block form by

$$\widehat{\tau}_e(t) \;=\; \begin{bmatrix} \widehat{f}_e(t) \\ \widehat{m}_e(t) \end{bmatrix} = \begin{bmatrix} \displaystyle\int_0^t K_{I,f}\big(M_t a(\sigma) - f(\sigma) - \widehat{f}_e(\sigma)\big)\, d\sigma \\ K_{I,m}\Big(I\omega(t) - \displaystyle\int_0^t \big(m(\sigma) + m_g(\sigma) + (I\omega(\sigma)) \times \omega(\sigma) - \widehat{m}_e(\sigma)\big)\, d\sigma\Big) \end{bmatrix} \tag{2.38}$$

,

where:

- $a = R^\top(\ddot{r} - ge_3)$ is the accelerometer reading in the body frame (including gravity),
- M_t is the translational mass block of M,
- f denotes the thrust contribution (translational part of $J^\top\tau$),
- m is the control torque,
- m_g is the torque due to a center-of-gravity offset,
- $K_I = \mathrm{diag}(K_{I,f}, K_{I,m})$ splits the estimator gains into the force and moment diagonal blocks.

The two blocks have different roles and motivations:

The first (upper) block integrates the difference between the "expected" translational inertial force $M_t a - f$ and the current estimate $\widehat{f}_e$. It is the integral form of the differential filter

$$\dot{\widehat{f}}_e \;=\; K_{I,f}\big(M_t a - f - \widehat{f}_e\big),$$

and leverages the direct availability of the accelerometer measurement to produce a fast estimate of external translational forces without requiring drift-free translational velocity estimates.

The second (lower) block builds the external moment estimate from the integral of known rotational terms, exploiting the angular momentum $p_\omega = I\omega$. This avoids numerical differentiation of ω (which is typically very noisy) and yields a robust torque estimate. In differential form the corresponding residual dynamics are

$$\dot{\hat{m}}_e = K_{I,m}\left(m - \hat{m}_e\right),$$

but the integral/residual implementation is preferred in practice.

Advantages of the hybrid design

- Force block: the acceleration-based block uses the IMU accelerometer directly → it does not rely on drift-free translational velocity (the main weakness of momentum-only approaches). It therefore provides a fast and responsive estimate of translational force.

- Moment block: the momentum-based block avoids numerical differentiation of ω and relies on $p_\omega = I\omega$ (readily available). This produces cleaner moment/torque estimates.

- Decoupled tuning: splitting K_I into $K_{I,f}$ and $K_{I,m}$ (diagonal blocks) allows independent tuning of force and moment bandwidths (e.g. larger $K_{I,f}$ when accelerometer noise is low; smaller $K_{I,m}$ to limit amplification of angular-rate noise).

2.2 Collision Detection and Equilibrium Recovery

Collision detection is performed using the norms of the estimated external wrench components, namely the translational force $\|\widehat{f}_e\|$ and the rotational moment $\|\widehat{m}_e\|$.

These scalar quantities provide a compact and robust measure of disturbance intensity, reducing the complexity of the decision process while retaining sufficient information for reliable detection.

A contact is declared whenever either norm exceeds a predefined threshold. In particular, two separate thresholds are used (one for force, one for moment), and a debounce logic based on a sliding window is applied: an event is confirmed only if the corresponding threshold is exceeded for N consecutive samples. The estimates themselves are produced by the hybrid observer introduced in (2.38): the translational estimate $\widehat{f}_e$ is supplied by the acceleration-based filter and the rotational estimate $\widehat{m}_e$ by the momentum-based filter. Both observer outputs are further smoothed by first-order filtering; therefore the signals compared against the thresholds are damped and less sensitive to high-frequency noise.

Static thresholds were selected experimentally by characterizing sensor noise during steady hover and by observing the estimator response during representative manoeuvres. This pragmatic choice simplifies implementation and ensures predictable behavior in practical experiments.

2.2.1 Collision-point estimation

To locate the contact point on the vehicle surface (so that the obstacle can later be identified and negotiated), the information provided by the external-wrench estimator is exploited. Several modeling assumptions and simplifications are adopted to keep the procedure robust and implementable with the available onboard sensors.

Assumptions and simplifications

- The vehicle geometry is known a priori; in the present implementation the vehicle is enclosed in a cubic (or rectangular-prism) protective frame. Knowledge of the geometric model is required to localize the contact on a specific face of the cage.

- The collision is treated as *point-like* and *instantaneous*. This simplification facilitates the algebraic inversion used to obtain a candidate contact location from the estimated wrench.

- Only the first estimator sample associated with the detected contact is used to compute the contact point. Subsequent wrench samples are ignored because post-impact vehicle dynamics (which produce additional inertial forces and moments) are not observable with sufficient accuracy using only a single IMUbased estimator.

- For the purpose of the contact-point calculation we reduce the problem from full 3D to a 2D planar problem in the horizontal plane. This excludes vertical impacts or collisions with highly oblique / strongly curved surfaces; this choice is justified by the fact that most impacts observed in our experiments are frontal (in the horizontal plane). The full 3D procedure follows the same algebraic steps and can be implemented analogously if required.

- A high-pass filter is applied to the measured wrench to attenuate slowlyvarying disturbances (e.g. wind) and model errors; the filtered result is used for the contact-point computation.

Let $\mathrm{b}\mathbf{f}_e$ and $\widehat{\mathrm{m}}_e$ denote the estimated external force and moment expressed in the body frame (as in (2.38)). For the 2D planar reduction we define

$$\mathrm{b}\mathbf{f}_{e,xy} = \begin{bmatrix} f_{\mathrm{b}e,x} \\ f_{\mathrm{b}e,y} \\ 0 \end{bmatrix}, \qquad m_{e,z} \in \mathrm{R\,b}$$

(the z-component of the moment about the body frame). The contact point in the body frame is denoted by $\mathbf{r}_c$ and the inertial pose of the vehicle by $(\mathbf{r}, R)$ so that a body-frame point $\mathbf{r}_c$ maps to the world frame as $\mathbf{r}_p = \mathbf{r} + R\mathbf{r}_c$.

From the wrench relation for a point contact we have

$$\widehat{\mathbf{m}}_e = \mathbf{r}_c \times \widehat{\mathbf{f}}_e.$$

Because the component of $\mathbf{r}_c$ parallel to the force is not observable, the set of admissible solutions for $\mathbf{r}_c$ lies on a line. A convenient particular solution is

$$\mathbf{p}_0 = \frac{\widehat{\mathbf{f}}_{e,xy} \times \left(0, 0, \widehat{m}_{e,z}\right)^\top}{\widehat{\mathbf{f}}_{e,xy}^\top \widehat{\mathbf{f}}_{e,xy}},$$

and the general solution (restricted to the plane) can be written as

$$\mathbf{r}_c(\alpha) = \mathbf{p}_0 + \alpha\, {}_b\mathbf{f}_{e,xy}, \qquad \alpha \in \mathbb{R},$$

i.e. a line passing through $\mathbf{p}_0$ and directed along ${}_b\mathbf{f}_{e,xy}$. Every point on this line satisfies the moment equation. The scalar parameter α is determined by imposing that $\mathbf{r}_c$ belongs to one of the planar faces of the vehicle model.

Consider a candidate face of the vehicle geometry defined by a reference point $\mathbf{v}_0$ (on the face plane) and a unit normal $\mathbf{n}_{\text{face}}$ (expressed in the body frame). The intersection condition between the line $\mathbf{r}_c(\alpha)$ and the face plane is

$$\mathbf{n}_{\text{face}}^\top \left(\mathbf{r}_c(\alpha) - \mathbf{v}_0\right) = 0.$$

Substituting $\mathbf{r}_c(\alpha)$ and solving for α yields

$$\alpha = -\frac{\mathbf{n}_{\text{face}}^\top (\mathbf{p}_0 - \mathbf{v}_0)}{\mathbf{n}_{\text{face}}^\top \widehat{\mathbf{f}}_{e,xy}}. \tag{2.39}$$

The denominator $\mathbf{n}_{\text{face}}^\top \widehat{\mathbf{f}}_{e,xy}$ may be close to zero when the force direction is nearly parallel to the face plane (i.e. nearly perpendicular to the face normal). In such cases the intersection is ill-conditioned and should be treated with caution (see fallback strategies below).

Having computed α, the candidate contact point on the face is $\mathbf{r}_c(\alpha)$. One must then verify whether this point lies inside the face polygon:

- for rectangular faces: project $\mathbf{r}_c(\alpha)$ onto local face coordinates and check the bounds;

- for generic polygons: use a point-in-polygon test or barycentric coordinates.

We accept intersections that satisfy both:

1. physical consistency: $\alpha < 0$ (we assume the measured force pushes toward the vehicle interior, hence the component along $\mathrm{b}\mathbf{f}_{e,xy}$ pointing inward corresponds to a negative α); and

2. geometric validity: the intersection point lies inside the face polygon.

If several faces provide admissible intersections, we select the face with the smallest $|\alpha|$ (i.e. the intersection closest to $\mathbf{p}_0$ in the inward direction) or, alternatively, the candidate with minimum distance to the face boundary (a practical heuristic that depends on the cage geometry).

If no face yields a valid intersection (this may occur due to coarse geometric modelling or estimator noise), we adopt a fallback strategy:

- project the candidate line onto the vehicle hull and take the closest point between the line and the hull surface, or

- perform a numerical ray/mesh intersection between the candidate line and a more detailed mesh representation of the vehicle (this is more robust but computationally heavier).

Once a valid body-frame contact point $\mathbf{r}_c$ has been selected, the contact position in the world frame is

$$\mathbf{r}_p = \mathbf{r} + R\mathbf{r}_c.$$

A robust estimate of the collision-plane normal in the world frame is obtained from the measured force direction:

$$\mathbf{n}_p = \frac{R\mathbf{f}_e}{\|\mathbf{f}_e\|},$$

and the collision plane can be written as $\mathbf{n}_p^\top(\mathbf{x} - \mathbf{r}_p) = 0$. In practice the normal direction is typically more reliable than the absolute position, whose accuracy depends on the fidelity of the geometric prism approximation and on estimator noise.

This approach yields a simple, sensor-light method to localize an impact using only the IMU-based external-wrench estimator and a coarse geometric model. Its main limitations are the point-contact/instantaneous impact assumption and the planar reduction: vertical impacts or collisions with strongly oblique surfaces will not be handled accurately. The contact estimate is intentionally only approximate — it serves as a reference for subsequent *controlled* contact manoeuvres, where the point estimate can be refined once the vehicle establishes continuous, constrained contact with the obstacle (and additional sensor information or controlled motions become available).

2.2.2 Attitude stabilization after collision

Once a collision between the UAV and an obstacle is detected, the control objective switches from trajectory tracking to *attitude stabilization*. The primary goal in this phase is to recover a safe and controllable flight condition by suppressing undesired rotational motions that could lead to secondary impacts or loss of control.

In particular, the controller aims at:

- minimizing undesired yaw motion, which may cause uncontrolled lateral movements and further collisions,
- driving roll and pitch angles towards zero so that the thrust vector is aligned with the vertical direction,
- maintaining sufficient thrust to keep the vehicle airborne.

During this phase, position control is intentionally deprioritized; the main objective is to restore a stable attitude configuration.

The desired acceleration is defined in order to ensure vertical stability:

$$\mathbf{a}_{\text{des}} = \begin{bmatrix} 0 \\ 0 \\ a_z \end{bmatrix}, \tag{2.40}$$

where a_z is chosen to maintain altitude. The corresponding desired force is

$$\mathbf{f}_{\text{des}} = m\left(g\mathbf{e}_3 + \mathbf{a}_{\text{des}}\right), \tag{2.41}$$

from which the desired body z-axis is obtained as

$$\mathbf{z}_{\text{des}b} = \frac{\mathbf{f}_{\text{des}}}{\|\mathbf{f}_{\text{des}}\|}. \tag{2.42}$$

The desired attitude is then constructed by aligning the body z-axis with $\mathbf{z}^{\text{des}}{}_b$ while keeping a bounded yaw rotation.

Let $R \in SO(3)$ be the current rotation matrix and R_{des} the desired one. The attitude error is defined as

$$e_R = \frac{1}{2}\left(R_{\text{des}}^{\top} R - R^{\top} R_{\text{des}}\right)^{\vee}, \tag{2.43}$$

while the angular velocity error is

$$e_\omega = \omega - R^{\top} R_{\text{des}} \omega_{\text{des}}. \tag{2.44}$$

A standard control law on $SO(3)$ is adopted:

$$\tau_{\text{ctrl}} = -K_R e_R - K_\omega e_\omega + \omega \times I\omega, \tag{2.45}$$

where K_R and K_ω are positive-definite gain matrices.

Immediately after the collision, the UAV may exhibit high angular velocities. To enhance robustness, an additional damping action is introduced by penalizing angular rates:

$$\tau_{\text{damp}} \propto -\omega, \tag{2.46}$$

which acts as a viscous term that accelerates the decay of rotational motion.

Altitude is maintained through a simple PD control law:

$$u_z = K_{p,z}(z_{\text{des}} - z) + K_{d,z}(\dot{z}_{\text{des}} - \dot{z}), \tag{2.47}$$

leading to the required thrust:

$$T = mg + u_z. \tag{2.48}$$

This ensures that the UAV remains airborne while attitude stabilization is being achieved.

The system is considered stabilized when both the attitude and the angular rates are sufficiently small. This can be expressed as

$$\|e_R(t)\| < \epsilon_R, \qquad \|\omega(t)\| < \epsilon_\omega, \tag{2.49}$$

for a sufficiently long time interval $T_{\min}$. The introduction of this temporal condition prevents premature switching due to measurement noise.

The described stabilization strategy prioritizes robustness over tracking performance. By decoupling attitude recovery from position control and by introducing strong damping of rotational dynamics, the controller ensures a fast return to a safe configuration after impact. This approach is particularly effective in scenarios where external disturbances are large and poorly predictable, such as physical interactions with the environment.

2.3 Alignment Phase

2.3.1 Recovery translation: moving away from the contact

After the vehicle has recovered a controllable attitude, the next objective is to create safe spatial separation from the contact point before engaging any wall-interaction behavior. This stage aims at (i) translating the vehicle along a direction approximately normal to the contact surface to a pre-defined recovery locus, (ii) keeping altitude and rotational stability, and (iii) detecting and reacting to situations in which the vehicle is not progressing away ("stuck").

The following paragraphs present the conceptual elements of this stage and the associated control/decision methodology.

Let $\mathbf{p}_c \in \mathrm{R}^3$ denote the estimated contact point and let $\mathbf{n}_c \in \mathrm{R}^3$ be the corresponding unit contact normal, pointing outward from the obstacle surface. The recovery position is defined by moving a fixed safety distance $d_s > 0$ along the normal direction:

$$\mathbf{p}_r = \mathbf{p}_c - d_s \mathbf{n}_c.$$

This means that the recovery position is not chosen arbitrarily, but lies along the direction perpendicular to the contact surface, at a prescribed distance from the contact point.

To evaluate the progress of the vehicle away from the obstacle, define the relative vector from the recovery locus to the current vehicle position as

$$\mathbf{r} = \mathbf{p} - \mathbf{p}_r)_{xy},$$

where $(\cdot)_{xy}$ denotes the projection onto the horizontal plane. The signed progress along the recovery direction is then obtained by projecting $\mathbf{r}$ onto the recovery unit direction

$$\mathbf{d}_r = -\mathbf{n}_c,$$

namely

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{d}_r.$$

A positive value of s indicates that the vehicle is moving away from the contact point, whereas a small or negative value indicates that the vehicle is still too close to the obstacle.

To robustly judge progress and to guide motion, the controller forms a composite error that mixes a longitudinal term (progress along the recovery direction) and a lateral/perpendicular term (distance from the recovery point). Denote the longitudinal error

$$e_{\parallel} = \mathbf{r} \cdot \mathbf{d}_r$$

and the perpendicular error

$$e_{\perp} = \|(\mathbf{p}_r - \mathbf{p})_{xy}\|.$$

A weighted composite error is defined as

$$e_c = w_{\parallel} e_{\parallel} + w_{\perp} e_{\perp},$$

with positive weights $w_{\parallel}, w_{\perp}$ chosen to prioritise forward progress (typically $w_{\perp} \geq w_{\parallel}$ if accurate centring is more important than raw distancing). The composite error provides a single scalar measure used for derivative-based approach detection and for simple monotonicity tests.

Altitude is regulated independently (as in the stabilization phase) using a vertical PD:

$$u_z = K_{p,z}(z_r - z) + K_{d,z}(\dot{z}_r - \dot{z}),$$

and the resulting vertical force F_z is combined with the horizontal force to obtain the total desired force vector in the inertial frame:

$$\mathbf{F}_{\text{des}} = \begin{bmatrix} \mathbf{F}_{xy} \\ F_z + mg \end{bmatrix}.$$

This total force determines the desired thrust direction and magnitude used for attitude reference generation (see previous section).

The desired vehicle attitude is constructed so that the body thrust axis aligns with $\mathbf{F}_{\text{des}}$; the corresponding desired body z-axis is $\mathbf{z}^{\text{des}}{}_b = \mathbf{F}_{\text{des}}/\|\mathbf{F}_{\text{des}}\|$. A desired yaw angle is chosen to orient the vehicle relative to the wall. A practical design is to select the yaw among a small finite set of orientations parallel to the wall surface (e.g. four candidates separated by 90°) and to pick the candidate closest to the current heading. Since the UAV is equipped with an approximately cubic protective cage, it is more convenient for

the vehicle to approach and subsequently slide along the obstacle with one of its faces. For this reason, and in order to simplify the interaction phase, the vehicle is preliminarily oriented so that one face is aligned perpendicularly to the estimated obstacle surface.So this discrete choice provides simple, robust alignment with the wall geometry and simplifies the subsequent approach stage.

To effect the yaw correction smoothly, the yaw-rate reference can be generated with a saturating smooth law of the form

$$\dot{\psi}_{\mathrm{des}} = \mathrm{sgn}(\psi_{\mathrm{err}})\,\dot{\psi}_{\mathrm{max}}\left(1 - e^{-k|\psi_{\mathrm{err}}|}\right),$$

where ψ_{err} is the yaw error and $\dot{\psi}_{\mathrm{max}}, k > 0$ control the maximum rate and convergence shape. This yields small commanded rates for small errors and bounded commands for large errors, avoiding aggressive yaw motions.

Transition to the approach mode is allowed only when a set of conservative conditions is met simultaneously for a sufficient duration:

- forward progress beyond a threshold: $s \geq s_{\mathrm{th}}$,
- low rotational activity around the yaw axis: $|\omega_z| < \omega_{\mathrm{th}}$,
- low planar body velocity magnitude: $\|\mathbf{v}_{B,xy}\| < v_{\mathrm{th}}$,
- yaw error within tolerance: $|\psi_{\mathrm{err}}| < \psi_{\mathrm{th}}$.

Each condition should persist for N_{hold} samples (or for a time T_{hold}) to prevent chattering. Only after all conditions are met the finite-state machine advances to the next stage.

2.3.2 Obstacle Approach

Once the UAV has been aligned with the wall, the next stage consists of bringing the vehicle into controlled contact with the obstacle and aligning the selected face of the protective cage with the wall surface. This phase is particularly delicate, since the first impact with the obstacle may induce a new loss of stability if not properly managed. At

the same time, a correct geometric alignment greatly simplifies the subsequent interaction phase, especially the transition to sliding.

For this reason, the approach phase is explicitly separated from the sliding phase, so that the initial contact between the UAV and the wall can be handled in a controlled and conservative manner. As discussed previously, the contact point is estimated only through the IMU-based external wrench estimation process; consequently, its position is not perfectly accurate, and the selected face of the vehicle is unlikely to be exactly parallel to the obstacle at first contact. The purpose of the approach stage is therefore to exploit the obstacle itself to progressively refine the alignment, establish a more geometrically consistent contact configuration, and collect the information required to make the following stages more reliable and easier to execute.

Because the cage geometry is approximately symmetric with respect to 90° rotations around the vertical axis, the wall can be approached by selecting one of a small number of equivalent yaw configurations. In practice, this means that the vehicle chooses the body face whose normal is closest to the estimated obstacle normal. This discrete selection simplifies the contact problem: once the correct face is chosen, the wall interaction can be described as a motion in which one body axis remains aligned with the wall normal, while the orthogonal horizontal axis defines the tangential direction of motion.

Due to these complementary characteristics, neither formulation alone is fully suitable for reliable external wrench estimation using proprioceptive sensing only.For this reason, a hybrid estimation scheme is adopted, combining the acceleration-based formulation for external force estimation with the Its objective is to maintain a consistent sliding motion along an obstacle while progressively refining the geometric description of the wall. From a theoretical standpoint, the procedure combines contact-frame identification, kinematic consistency verification, and online leastsquares estimation.

Let $\mathbf{R}^{W_B} \in SO(3)$ denote the rotation matrix mapping vectors from the body frame B to the world frame W, and let $\mathbf{d}_c \in \mathrm{R}^3$ be the unit vector pointing toward the contact direction. The relative orientation between the UAV and the obstacle is evaluated by computing the alignment between $\mathbf{d}_c$ and the body axes expressed in the world frame:

$$d_x = \hat{\mathbf{b}}_x^W \cdot \mathbf{d}_c, \qquad d_y = \hat{\mathbf{b}}_y^W \cdot \mathbf{d}_c, \tag{2.50}$$

where $\hat{\mathbf{b}}^{W}_{x}$ and $\hat{\mathbf{b}}^{W}_{y}$ are the first two columns of $\mathbf{R}^{W}_{B}$. The corresponding angular misalignments are given by

$$\theta_x = \arccos(d_x), \qquad \theta_y = \arccos(d_y) \tag{2.51}$$

When one of these angles is sufficiently small, the controller infers that the UAV is approximately orthogonal to one of its body axes. This allows the contact geometry to be expressed in a locally consistent frame.

In order to consistently interpret the contact interaction, the controller identifies which body axis of the UAV is most aligned with the direction of the obstacle. This is achieved by evaluating the alignment between the contact direction $\mathbf{d}_c$ and the body axes expressed in the world frame:

$$d_i = \hat{\mathbf{b}}_i^W \cdot \mathbf{d}_c, \quad i \in \{x, y\}. \tag{2.52}$$

The axis exhibiting the strongest alignment (in absolute value) is selected as the primary contact axis. This choice determines which face of the UAV is effectively interacting with the obstacle and allows the controller to define a consistent local contact frame.

Depending on the selected axis and on the sign of the alignment, a discrete reorientation matrix $\mathbf{R}_{\text{adj}}$ is applied to the body-frame quantities. The purpose of this transformation is to map the measured velocities into a frame in which one axis is aligned with the wall normal and another with the tangential sliding direction.

This step is essential to ensure that subsequent computations, such as velocity projection and contact validation, are performed in a geometrically consistent reference frame, independently of the specific face of the UAV involved in the interaction.

Once the dominant orientation has been identified, the wall normal and tangential directions are extracted from the body frame. If the contact is associated with the body x-axis, the wall normal is defined as

$$\mathbf{n}^W = \hat{\mathbf{b}}_x^W, \qquad \mathbf{t}^W = \hat{\mathbf{b}}_y^W, \tag{2.53}$$

whereas if the contact is associated with the body y-axis, the roles of the two axes are exchanged:

$$\mathbf{n}^W = \hat{\mathbf{b}}_y^W, \qquad \mathbf{t}^W = \hat{\mathbf{b}}_x^W. \tag{2.54}$$

This construction defines a physically meaningful local frame in which one axis is normal to the obstacle and the other is aligned with the desired sliding direction.

To ensure that the UAV is actually performing a sliding motion, the body-frame velocity $\mathbf{v}_B$ is mapped into an adjusted frame:

$$\mathbf{v}_{\text{adj}} = \mathbf{R}_{\text{adj}}\mathbf{v}_B, \tag{2.55}$$

where $\mathbf{R}_{\text{adj}}$ is a discrete reorientation matrix selected according to the detected contact configuration.

The tangential motion is considered valid only if its magnitude exceeds a threshold and its sign is consistent with the commanded motion:

$$|\mathbf{v}_{\text{adj},y}| > \alpha_v v_{\text{ref}}, \tag{2.56}$$

$$\text{sign}(\mathbf{v}_{\text{adj},y}) = \text{sign}(v_{\text{ref},y}), \tag{2.57}$$

with $\alpha_v \in (0,1)$ and v_{ref} the nominal sliding velocity. These conditions prevent the controller from interpreting non-tangential or unstable motion as valid wall-following behavior.

The wall is modeled in the horizontal plane as a straight line,

$$y = m_{\text{wall}}x + q_{\text{wall}}, \tag{2.58}$$

where m_{wall} is the slope and q_{wall} is the intercept. For a set of collected wall-contact points $\mathbf{p}_i = (x_i, y_i, z_i)$, the scalar quantity

$$q_i = y_i - m_{\text{wall}}x_i \tag{2.59}$$

is used to evaluate the relative position of each point with respect to the estimated wall.

The line parameters m_{wall} and q_{wall} are updated online based on the set of collected wall points. In the considered implementation, the estimation is not explicitly recomputed in this block, but the parameters are assumed to be available from a previous estimation stage and are directly used for geometric consistency checks and switching conditions.

Because the initial contact point cannot be estimated with high precision and the exact orientation of the obstacle face is not known a priori, the first impact between the UAV and the wall is generally not perfectly aligned. The underlying idea is therefore to exploit the obstacle itself as a reference to progressively infer its orientation.

When the UAV intercepts the wall, any mismatch between the assumed obstacle orientation and the true one, or any small error in the motion execution, leads to an approximately point-like contact rather than a fully distributed one. This occurs because the UAV may still exhibit a non-negligible pitch or roll angle, depending on the selected face used for the interaction, and because the contacted face is not exactly parallel to the obstacle surface.

As a consequence, the contact force generates a rotational moment that tends to reorient the UAV until its surface becomes more parallel to the wall. In first approximation, the rotational effect about the yaw axis can be described as

$$\tau_z = (\mathbf{r}_c \times \mathbf{f}_c)_z, \tag{2.60}$$

where $\mathbf{r}_c$ is the contact lever arm and $\mathbf{f}_c$ is the contact force. The resulting yaw dynamics can be approximated by

$$I_z \dot{r} = \tau_z, \tag{2.61}$$

where I_z is the yaw inertia and r is the yaw rate. Therefore, repeated contact naturally produces a corrective yaw motion that tends to align the UAV with the obstacle.

At the same time, the initial rebounds caused by the impact generate small translations in the direction normal to the obstacle. These oscillations progressively decay as the interaction stabilizes, and the UAV becomes effectively “stuck” to the wall.

From a physical point of view, this corresponds to a transition from transient impact dynamics to a quasi-static contact regime.

To evaluate the reliability of the contact condition, a scalar confidence metric is introduced. This metric is designed to capture whether the UAV has reached a quasi-static interaction with the wall, by combining both rotational and translational indicators of stability.

The overall confidence is defined as

$$w = w_r w_n, \tag{2.62}$$

where w_r accounts for rotational stability and w_n accounts for translational stability along the wall normal.

The rotational component is defined as

$$w_r = \frac{1}{1 + k_r |\omega_z|}, \tag{2.63}$$

where ω_z is the yaw rate and $k_r > 0$ is a tuning parameter. This term penalizes high angular velocities, assigning higher confidence to conditions in which the UAV orientation is stable.

The translational component is defined as

$$w_n = \frac{1}{1 + \exp(k_n (|v_n| - v_0))}, \tag{2.64}$$

where $v_n = \mathbf{v} \cdot \mathbf{n}$ is the velocity component along the wall normal, v_0 is a threshold value, and $k_n > 0$ controls the sharpness of the transition. This term behaves as a decreasing sigmoid function, assigning high confidence when the normal velocity is small and penalizing large oscillations perpendicular to the wall.

The use of the product $w = w_r w_n$ ensures that the overall confidence is high only when both rotational and translational dynamics are sufficiently stable. In particular, high values of w correspond to a regime in which the UAV exhibits negligible yaw motion and minimal normal oscillations, which is consistent with a stable and well-established contact with the wall.

This metric is therefore used both to weight measurements for wall estimation and to determine whether the system can transition to subsequent control phases.

Stable sliding is enabled only after sufficient information has been accumulated. Let the total confidence be

$$W_{\text{tot}} = \sum_{i=1}^{N} w_i \quad . \tag{2.65}$$

The system is considered ready for reliable wall-following once

$$W_{\text{tot}} > W_{\text{min}}. \tag{2.66}$$

At that point, the wall yaw can be estimated as a weighted average:

$$\psi_{\text{wall}} = \frac{\sum_{i=1}^{N} w_i \psi_i}{\sum_{i=1}^{N} w_i} \quad , \tag{2.67}$$

where ψ_i denotes the yaw angle measured from each valid sample.

The transition to the fully active sliding phase is permitted only when the contact geometry is sufficiently reliable and the motion is sufficiently stable. Conceptually, this requires three conditions:

- the vehicle face is approximately aligned with the wall normal,
- the UAV is moving in the correct tangential direction,
- the estimate of the wall orientation has accumulated enough confidence.

Once these conditions are satisfied, the system can safely exploit the wall as a geometric reference and begin the sliding phase with a much lower risk of secondary impacts or unstable reorientation.

Overall, this phase can be interpreted as a hybrid estimator-controller for wallcontact locomotion. The controller first infers the contact orientation, then defines a consistent local contact frame, verifies the tangential sliding motion, and finally updates the wall model through online regression and confidence-weighted averaging.

This structure provides a principled way to transition from transient contact to stable wall-following behavior.

2.4 Sliding Phase

Once the alignment phase is completed, the UAV is required to translate along the obstacle while maintaining contact, with the objective of bypassing the obstacle and resuming the previously assigned mission. As discussed in the previous sections, the obstacle is modeled as a vertical planar surface without concavities, in order to simplify the geometric description of the interaction.

Once the alignment phase has been completed, the UAV enters the sliding phase, whose objective is to move tangentially along the obstacle while maintaining continuous contact. In this phase, the obstacle is idealized as a vertical planar surface without concavities, which allows the interaction to be described through a simplified geometric model. Under this assumption, the motion of the UAV can be decomposed into a component normal to the surface, which must be regulated to preserve contact, and a component tangential to the surface, which generates the desired sliding motion.

During the sliding phase, the UAV continuously acquires contact-related measurements in order to refine the geometric model of the obstacle. Only samples associated with sufficiently reliable interaction conditions are retained, so that the wall estimate is updated exclusively when the contact is considered meaningful. In this way, the estimation process is constrained to the regime in which the vehicle is effectively sliding along the surface and the measured state can be interpreted consistently.

2.4.1 Online Wall Estimation

The UAV is assumed to be in contact with a vertical wall and must move along the wall while maintaining a stable interaction. Let $\mathbf{p}_W \in \mathrm{R}^3$ denote the position of the UAV in the world frame, and let $\mathbf{v}_W \in \mathrm{R}^3$ denote its linear velocity. The attitude of the vehicle is represented by the rotation matrix $\mathbf{R}^{W}_{B} \in SO(3)$, which maps body-frame vectors into the world frame.

The wall is modeled locally as a straight line in the horizontal plane. This approximation is sufficient for the short-range contact motion considered here and enables an efficient online estimation of the wall orientation. In the simplest form, the wall can be represented by the relation

$$y = m_{\text{wall}}x + q_{\text{wall}}, \tag{2.68}$$

where m_{wall} is the wall slope and q_{wall} is the intercept. The estimated wall direction determines the tangential sliding direction, while the wall normal determines the direction in which contact forces are regulated.

To simplify the control design, it is useful to define two orthogonal directions in the plane:

- the unit normal vector $\mathbf{n}_W$, oriented orthogonally to the wall surface;
- the unit tangent vector $\mathbf{t}_W$, aligned with the desired direction of motion along the wall.

These vectors define a local contact frame that evolves as the wall estimate is updated. A third direction is given by the binormal vector

$$\mathbf{b}_W = \mathbf{n}_W \times \mathbf{t}_W, \tag{2.69}$$

which completes a right-handed basis and is useful for constructing a visual representation of the contact region and for interpreting the geometry of the interaction.

The control problem can therefore be formulated as follows: given the current vehicle state and the estimated wall geometry, generate a force command that (i) stabilizes the normal interaction with the wall, (ii) produces the desired tangential sliding motion, (iii) compensates gravity and disturbances, and (iv) yields an attitude command compatible with the UAV actuation limits.

A key component of the proposed controller is the online estimation of the wall orientation. Since the UAV is not assumed to know the exact wall geometry in advance, the wall must be reconstructed during contact using the positions of valid contact points. The algorithm maintains a set of points collected while the UAV is in the sliding phase. Each point stores the UAV position, a confidence weight, and a scalar coordinate used to sort the samples along the wall.

Not every position sample is suitable for wall estimation. The controller accepts new samples only when the system is sufficiently confident that it is actually in contact or in close proximity to the wall. This confidence is represented by a wall estimation weight. In the implementation considered here, only samples with sufficiently high confidence are used to enrich the wall point set. This prevents spurious measurements from corrupting the geometric estimate during transients or low-contact situations.

Each accepted point is projected onto a scalar coordinate

$$q(\mathbf{p}) = y - m_{\text{wall}}x, \tag{2.70}$$

where the current estimate of the wall slope is used to define an ordering along the wall. This scalar quantity is not a geometric distance in the Euclidean sense; rather, it is a convenient monotonic quantity used to determine whether the vehicle is moving forward or backward along the wall. The point cloud is also limited in size to avoid unbounded memory growth and to preserve computational efficiency.

The wall point set is updated incrementally. When the number of stored samples is still small, new points are accepted more permissively. Once the set reaches a sufficient size, new points are validated based on their consistency with the current motion direction. More precisely, the controller compares the scalar coordinate of the current UAV position with that of the most recent stored point. If the UAV is moving in the expected direction along the wall, the point is accepted; otherwise, it is discarded.

This mechanism has two advantages. First, it prevents points collected during reversals or local oscillations from contaminating the estimate. Second, it ensures that the regression is performed on a spatially consistent set of samples, which improves the robustness of the wall slope estimate.

In addition to motion consistency, the update is rate-limited. The controller only performs a regression update when either a new valid point has been added or a certain amount of time has elapsed since the last update. This prevents excessive recomputation and avoids reacting too aggressively to short-term fluctuations.

During the initial phases of the sliding motion, the UAV has not yet collected a sufficient number of points and therefore does not possess an accurate estimate of the wall position. In fact, small oscillations or slight deviations of the wall, which are typical in the early stages of the transition to motion parallel to the wall, may generate points that are not fully consistent with the actual geometry of the obstacle. For this reason, the first estimate is initialized from the weighted average of the UAV yaw about its vertical axis, obtained at the final stage of the wall approach according to the weighted sum in (2.67), and it is subsequently updated until enough valid points have been collected.

Once enough valid points have been collected, the wall parameters are refined by linear regression. Given a set of N points (x_i,y_i), the slope is estimated as

$$m_{\text{wall}} = \frac{N\sum_i x_i y_i - \left(\sum_i x_i\right)\left(\sum_i y_i\right)}{N\sum_i x_i^2 - \left(\sum_i x_i\right)^2}, \tag{2.71}$$

provided that the denominator is not too small. The intercept can then be recovered as

$$q_{\text{wall}} = \frac{1}{N}\left(\sum_i y_i - m_{\text{wall}}\sum_i x_i\right). \tag{2.72}$$

This least-squares estimate gives a compact and efficient representation of the wall in the horizontal plane. If the regression denominator becomes too small, which would happen for nearly vertical data or degenerate sample distributions, the update is skipped to avoid numerical instability.

The regression output is used not only for estimating the wall direction but also for sorting the stored points along the sliding direction. This ordering is later useful for deciding which points should define the current subpath of motion and for diagnosing whether the vehicle is progressing correctly along the wall.

The estimated wall slope can be converted into a yaw angle. For a wall represented by the line $y = m_{\text{wall}}x + q_{\text{wall}}$, the orientation of the wall normal in the horizontal plane is given by

$$\psi_n = \arctan(m_{\text{wall}}), \tag{2.73}$$

up to the usual ambiguity introduced by the choice of normal direction. This estimate is compared against the yaw obtained from the perception layer or from the initial

alignment stage. The controller later uses this information to determine the desired yaw and to decide whether the UAV is still aligned with the wall within acceptable limits.

2.4.2 Control in the Normal Direction

The normal direction is the most critical component of the contact controller. It governs how strongly the vehicle presses against the wall and whether the UAV maintains the desired interaction without losing contact or overloading the structure.

In general, two main control strategies can be employed to maintain contact with such a surface: impedance control and admittance control. Impedance control defines a desired dynamic relationship between the interaction force and the resulting motion, as if the robot were characterized by virtual mass, damping, and stiffness. A common Cartesian impedance formulation can be written as

$$\mathbf{F}_{ext} = \mathbf{M}_d\ddot{\mathbf{x}} + \mathbf{D}_d\dot{\mathbf{x}} + \mathbf{K}_d(\mathbf{x} - \mathbf{x}_d), \tag{2.74}$$

where $\mathbf{F}_{ext}$ is the external interaction force, $\mathbf{x}$ is the end-effector position, and $\mathbf{M}_d$, $\mathbf{D}_d$, and $\mathbf{K}_d$ are the desired inertia, damping, and stiffness matrices, respectively. In this framework, the system behaves like a virtual mechanical impedance and reacts compliantly to external disturbances without rigidly enforcing a prescribed trajectory. This approach is particularly suitable for robots capable of directly regulating forces or torques.

Admittance control, on the other hand, measures the external force acting on the robot and converts it into a desired motion. A basic admittance model can be expressed as

$$\mathbf{M}_a\ddot{\mathbf{x}}_d + \mathbf{D}_a\dot{\mathbf{x}}_d + \mathbf{K}_a(\mathbf{x}_d - \mathbf{x}_{d,0}) = \mathbf{F}_{ext}, \tag{2.75}$$

where $\mathbf{x}_d$ is the desired motion generated by the controller, $\mathbf{x}_{d,0}$ is its nominal reference, and $\mathbf{M}_a$, $\mathbf{D}_a$, and $\mathbf{K}_a$ define the virtual admittance dynamics. In this case, the measured force is used as the input that generates a position or velocity command.

The main difference between the two approaches lies in the controlled quantity: impedance control regulates the mechanical response of the system to an imposed motion, whereas admittance control computes the motion reference from the applied

force. In a simplified form, admittance control can also be interpreted in the Laplace domain as

$$\mathbf{X}_d(s) = \mathbf{Y}(s)\mathbf{F}_{ext}(s), \tag{2.76}$$

where $\mathbf{Y}(s)$ is the desired admittance transfer function.

In the present context, the force is not measured directly through a dedicated force or pressure sensor, but rather estimated indirectly from the quantities provided by the IMU. As a consequence, the resulting estimate is inherently affected by uncertainty and by external disturbances that cannot be fully observed or compensated for. For this reason, admittance control was selected, since it provides a simpler and more robust way of managing the interaction between the UAV and the obstacle during the sliding phase.

The controller generates a normal reference velocity whose magnitude depends on the current phase of motion and on the yaw alignment. A smooth modulation is used to reduce the normal motion when the vehicle is significantly misaligned with the wall. In the implementation, this modulation is performed through a logistic function of the yaw error:

$$v_{n\text{ref}} = \frac{}{1 + \exp(5(|e_\psi| - \psi_0))} v_{\text{adm}}, \tag{2.77}$$

where v_{adm} is an admissible velocity scale, e_ψ is the yaw error, and ψ_0 is a threshold angle. This formulation ensures that the normal motion is reduced when the UAV is not sufficiently aligned, thereby avoiding destabilizing contact forces.

The reference normal velocity is transformed from the local contact frame to the world frame and integrated over the sampling interval to obtain a normal reference position. These quantities are then expressed both in the world frame and in the body frame for subsequent control computations.

The normal control law is based on the position and velocity errors along the wall normal. Let $\mathbf{n}_W$ denote the estimated wall normal. The position and velocity errors are

$$e_{p,n} = (\mathbf{p}_{\text{ref}W} - \mathbf{p}_W)^\top \mathbf{n}_W, \tag{2.78}$$

$$e_{v,n} = (\mathbf{v}_{W\text{ref}} - \mathbf{v}_W)^\top \mathbf{n}_W. \tag{2.79}$$

A force command is then generated as

$$f_n = k_n e_{p,n} + d_n e_{v,n}, \tag{2.80}$$

where k_n and d_n are the normal stiffness and damping gains, respectively. The resulting force vector is

$$\mathbf{F}_n = f_n \mathbf{n}_W. \tag{2.81}$$

The key idea is that position and velocity deviations normal to the wall are converted into a force along the wall normal, which stabilizes the interaction and regulates the contact condition.

The normal force term performs two tasks simultaneously. First, it maintains the UAV at a desired proximity to the wall by resisting deviations in the contact direction. Second, it contributes to the overall force vector used to generate the attitude command. Since the body axis is aligned with the total force, the normal component influences the tilt angle of the vehicle and thereby shapes the resultant motion.

2.4.3 Tangential Sliding Control

While the normal controller ensures that the UAV remains in controlled contact with the wall, the tangential controller generates the sliding motion along the surface. This component is especially important because it must provide motion persistence without disrupting contact stability.

At the beginning of the sliding phase, the controller determines whether the vehicle should move in the positive or negative tangential direction. This is done by comparing the current UAV position with the trajectory points or with a subpath extracted from the planned motion. The closest point on the reference path is identified, and the remaining portion of the path is used to determine which direction is more appropriate. The final direction is encoded as a sign variable, typically equal to $+1$ or -1.

This step is important because the same wall geometry can support two opposite sliding motions. Selecting the correct direction ensures consistency with the mission objective and prevents the vehicle from moving backward relative to the intended route.

The desired sliding speed is not constant. It is modulated by safety terms that reduce the commanded motion when the yaw error is large or when the motion normal to the wall is still significant. A representative form is

$$v_t^{\mathrm{ref}} = \mathrm{sat}\left(s\, v_{\mathrm{slide}}\, \alpha_\psi\, \alpha_n\right), \tag{2.82}$$

where $s \in \{-1,+1\}$ is the sliding direction, v_{slide} is the nominal tangential speed, α_ψ is a yaw-dependent reduction factor, α_n is a normal-velocity reduction factor, and sat(·) denotes a saturation function used to keep the command within reasonable limits.

The yaw reduction factor is typically a sigmoid-like function that decreases the sliding speed when the heading error exceeds a threshold. The normal-velocity reduction factor penalizes tangential motion when the vehicle is still moving too strongly toward or away from the wall. In this way, the controller naturally prioritizes stabilization of the contact before allowing fast motion along the wall.

To avoid abrupt changes in the sliding speed, the tangential command is filtered. A simple first-order filter is sufficient for this purpose:

$$v_{tf}[k] = \alpha v_t^{\mathrm{ref}}[k] + (1-\alpha) v_{tf}[k-1], \tag{2.83}$$

with $\alpha \in (0,1)$. This filtered command is used only when the magnitude increases; when the command decreases, the controller may allow a faster reduction to improve responsiveness. This asymmetric filtering strategy reduces oscillations while preserving the ability to slow down rapidly in potentially unsafe situations.

In addition, the tangential command is forcibly set to zero when the product of the yaw and normal reduction terms falls below a safety threshold. This represents a conservative safeguard that prevents motion from continuing when the system is not sufficiently well aligned.

Once the desired tangential speed is known, a tangential reference position is obtained by integrating the reference velocity. The tangential position and velocity errors are then computed with respect to the current UAV state. Let $\mathbf{t}_W$ denote the sliding tangent. The errors can be written as

$$e_{p,t} = (\mathbf{p}_{\text{ref}W} - \mathbf{p}_W)^\top \mathbf{t}_W, \tag{2.84}$$

$$e_{v,t} = (\mathbf{v}_{W\text{ref}} - \mathbf{v}_W)^\top \mathbf{t}_W. \tag{2.85}$$

Since tangential motion is less critical than normal contact stability, the controller limits the magnitude of the tangential position error before using it in the force law. This avoids excessive growth of the command if the vehicle temporarily lags behind the reference. The corresponding force is then

$$f_t = k_t e_{p,t} + d_t e_{v,t}, \tag{2.86}$$

where k_t and d_t are the tangential gains. The force vector in the world frame is

$$\mathbf{F}_t = f_t \mathbf{t}_W. \tag{2.87}$$

This term produces the actual sliding action along the wall.

2.4.4 Force Composition and Command generation

The total force command is obtained by summing the individual contributions related to gravity, wall contact, tangential sliding, disturbance compensation, and altitude regulation. The resulting expression is

$$\mathbf{F}_{\text{tot}} = \mathbf{F}_g + \mathbf{F}_n + \mathbf{F}_t - \mathbf{F}_d + \mathbf{F}_z. \tag{2.88}$$

The gravity compensation term is simply

$$\mathbf{F}_g = \begin{bmatrix} 0 & 0 & mg \end{bmatrix}^\top, \tag{2.89}$$

where m is the total mass and g is the gravitational acceleration.

The disturbance compensation term $\mathbf{F}_d$ is included to counteract external effects estimated by a disturbance observer or wrench estimator. This helps the vehicle preserve the desired contact behavior despite unmodeled forces.

The altitude correction term $\mathbf{F}_z$ is based on a vertical PD controller:

$$\mathbf{F}_z = \begin{bmatrix} 0 \\ 0 \\ k_z e_z + d_z e_{z^\cdot} \end{bmatrix}, \tag{2.90}$$

where

$$e_z = z_{\text{ref}} - z, \qquad e_{z^\cdot} = -z^\cdot \tag{2.91}$$

This term is used to maintain the desired altitude while the vehicle is sliding along the wall. Altitude regulation is necessary because the contact motion is not purely planar; the UAV must remain within a vertical band that is compatible with the mission and with the environment.

The total force is finally used as the input for the attitude generation stage. In particular, the direction of $\mathbf{F}_{\text{tot}}$ defines the desired thrust axis of the body.

Multirotor vehicles do not directly track force commands; instead, the force must be converted into attitude and thrust. The proposed controller performs this conversion by aligning the body z-axis with the total force vector, while the yaw is selected according to the wall geometry and the current heading. The desired body z-axis is given by the normalized total force:

$$\mathbf{z}_{\text{des}B} = \frac{\mathbf{F}_{\text{tot}}}{\|\mathbf{F}_{\text{tot}}\|}. \tag{2.92}$$

This choice ensures that the thrust produced by the UAV is directed along the force needed to realize the contact and sliding behavior.

To complete the attitude, a desired heading direction in the horizontal plane is selected:

$$\mathbf{x}_c = \begin{bmatrix} \cos\psi_d \\ \sin\psi_d \\ 0 \end{bmatrix}, \tag{2.93}$$

where ψ_d is the desired yaw. The desired body y-axis is then constructed as

$$\mathbf{y}_{B\text{des}} = \left\| \frac{\mathbf{z}_{\text{des}} \times \mathbf{x}}{\mathbf{z}_{B\text{des}B} \times \mathbf{x}_{cc}} \right\|, \tag{2.94}$$

and the desired body x-axis is obtained from

$$\mathbf{x}_{B\text{des}} = \mathbf{y}_{B\text{des}} \times \mathbf{z}_{B\text{des}}. \tag{2.95}$$

The resulting attitude matrix is

$$\mathbf{R}_{WB,\text{des}} = \mathrm{h}\mathbf{x}_{B\text{des}} \quad \mathbf{y}_{B\text{des}} \quad \mathbf{z}_{B\text{des}} \quad . \tag{2.96}$$

The thrust magnitude is extracted by projecting the total force onto the desired body z-axis:

$$T = \mathbf{F}_{\text{tot}}^{\top} \mathbf{z}_{B}^{\text{des}}. \tag{2.97}$$

This scalar value is then passed through the thrust model of the platform to obtain the corresponding throttle command. In practice, the mapping is nonlinear and depends on the specific motor and propeller characteristics of the UAV. A quadratic or otherwise calibrated thrust model is typically used for this conversion.

Yaw control is treated separately from the main attitude synthesis. This is useful because the wall-sliding task often requires the vehicle to keep a heading aligned with the wall or with a specific direction in the environment. The controller first determines a desired yaw based on the wall estimate. If the UAV is not yet ready for sliding, the current yaw is held. If the wall estimate is still uncertain, the desired yaw is set to the estimated wall yaw with a safety margin. Otherwise, multiple candidate yaw angles parallel to the wall are generated, and the one closest to the current heading is selected.

This strategy avoids unnecessary yaw flips and helps preserve smooth behavior during contact. In addition, if the current yaw deviates too much from the initial alignment, the controller resets the wall estimation and re-enters the preparation phase. This protects the vehicle from accumulating orientation errors that could destabilize the sliding motion.

A desired yaw rate can also be computed from the yaw error and the measured angular velocity. A simple PD structure is used:

$$\dot{\psi}_{\text{des}} = k_{\psi} e_{\psi} - d_{\psi} \omega_z - \tau_{\text{ext}}, \tag{2.98}$$

where ω_z is the measured yaw rate and τ_{ext} is an estimate of the external torque about the vertical axis. The resulting command is saturated to preserve safe actuation limits.

2.5 Safety Logic and Reset Conditions

Because the controller operates during physical contact, safety logic is essential. Several mechanisms are introduced to prevent unstable behavior and to recover from unfavorable conditions.

2.5.1 Yaw Deviation Reset

If the yaw error becomes too large, the controller considers the current configuration no longer reliable. In that case, the wall point set is cleared, the sliding state is disabled, and the controller returns to the initialization stage. This reset mechanism ensures that the wall estimate is not used outside its validity range and that the UAV does not continue sliding under a badly misaligned attitude.

2.5.2 Confidence-Based Sliding Activation

The sliding command is only activated when the system has a sufficient amount of wall information and the confidence in the estimation is high. If the vehicle is not yet ready to slide, the tangential force command is set to zero. This prevents the controller from generating motion along the wall before the contact geometry has been established.

2.5.3 Bounded Commands

Several internal variables are saturated to avoid excessive commands. For example, the tangential position error is limited to a finite interval, and the desired yaw rate is clipped to a bounded range. These constraints reduce the likelihood of sudden spikes in the output and improve the numerical stability of the implementation.

2.6 Detection of Wall Passing Condition

During the sliding phase, it is necessary to determine when the UAV has effectively passed the obstacle, so that the control logic can transition to the next stage of the mission. This decision is not based on a single instantaneous measurement, but on a geometric condition that combines the estimated wall equation, the current UAV position, the motion direction, and a safety margin. The purpose of this logic is to avoid premature switching caused by small oscillations around the wall boundary and to ensure that the transition occurs only when the vehicle has clearly moved beyond the obstacle.

The wall is locally represented by the estimated line

$$y = m_{\text{wall}}x + q_{\text{wall}}, \tag{2.99}$$

where m_{wall} and q_{wall} are obtained from the online wall estimation procedure. To increase robustness, the passing condition is not evaluated exactly on the estimated line, but on two parallel boundaries displaced by a margin Δq. In particular, two auxiliary conditions are defined:

$$y_{\text{ahead}} > m_{\text{wall}}x + q_{\text{wall}} + \Delta q, \tag{2.100}$$

$$y_{\text{behind}} < m_{\text{wall}}x + q_{\text{wall}} - \Delta q. \tag{2.101}$$

These inequalities define two regions that lie sufficiently far from the wall on opposite sides of the estimated boundary. The margin Δq is introduced to avoid false detections due to estimation noise, local wall irregularities, or short-term oscillations of the UAV during contact.

The switching condition is evaluated only when the system is already in the sliding phase and the vehicle is effectively moving. In this way, the obstacle-passing event is not triggered during initialization, transient stabilization, or when the UAV is momentarily stopped.

A further distinction is required depending on the wall orientation. When the estimated slope has a moderate value, the sign of the motion along the wall can be inferred directly from the geometry of the wall line. In this case, the controller evaluates whether the UAV is moving forward with respect to the contact direction.

2.6. DETECTION OF WALL PASSING CONDITION

If the vehicle is moving forward, the passing event is detected when the UAV is found sufficiently ahead of the wall; conversely, if it is moving backward, the event is detected when the UAV is sufficiently behind the wall.

For nearly vertical walls, the slope m_{wall} becomes very large and the previous geometric interpretation becomes numerically less convenient. In this case, a special handling is introduced. The forward direction is inferred directly from the sign of the component along the horizontal axis, and the passing condition is checked using a modified expression based on the absolute value of the wall slope. This avoids ambiguity in the orientation of the wall and improves numerical robustness when the wall is close to parallel to the y-axis.

More formally, the switching logic can be summarized as follows. Let

$$\mathrm{C}_{\text{ahead}} \equiv y > m_{\text{wall}}x + q_{\text{wall}} + \Delta q, \tag{2.102}$$

$$\mathrm{C}_{\text{behind}} \equiv y < m_{\text{wall}}x + q_{\text{wall}} - \Delta q. \tag{2.103}$$

Then, if the UAV is moving forward, the obstacle is considered passed when $\mathrm{C}_{\text{ahead}}$ becomes true; otherwise, the transition is triggered when $\mathrm{C}_{\text{behind}}$ becomes true. In both cases, the margin ensures that the decision is only made once the vehicle has clearly overcome the estimated wall boundary.

This detection strategy provides a simple yet effective way to identify the end of the wall-sliding maneuver. It is computationally lightweight, compatible with realtime execution, and robust to small errors in wall estimation. Most importantly, it allows the controller to decide the transition to the next phase using purely local geometric information, without requiring a complete map of the environment.

Chapter 3

Simulation and Experimental Analysis

3.1 Experimental Setup

This section presents the simulation environment, vehicle model, obstacle configuration, initial conditions, and controller setup used to evaluate the proposed contactaware control framework. The aim is to define a clear and reproducible experimental setup for performance assessment.

3.1.1 Simulation Environment

All experiments are conducted in a physics-based simulation environment, designed to emulate the dynamics of a multirotor aerial vehicle interacting with external obstacles. The simulator integrates rigid-body dynamics with a simplified contact model, allowing the reproduction of collision events and sustained contact interactions.

The simulation framework operates within a ROS-based architecture and is built upon the MRS UAV system, which provides a complete pipeline for multirotor simulation, state estimation, and control. This framework enables modular integration of estimation, control, and supervision components, and reflects a realistic

deployment scenario for aerial robotic systems.

The control loop runs at a fixed frequency, ensuring consistent timing between sensing, estimation, and command generation. Sensor feedback, including odometry

and inertial measurements, is assumed to be available with negligible delay and noise, unless otherwise specified.

The simulation environment is chosen to allow controlled and repeatable experiments, while preserving the essential physical characteristics required to study contact-aware control behaviors.

3.1.2 Vehicle Model and Control Interface

The aerial vehicle is modeled as a rigid-body multirotor system with standard translational and rotational dynamics. The platform considered is an octocopter configuration, equipped with eight rotors, which provides increased thrust authority and redundancy compared to quadrotor platforms.

The total mass of the vehicle is set to $m = 7.5$ kg. The geometric structure is characterized by an arm length of 0.20 m and a body height of 0.20 m. Propulsion is modeled using a quadratic relationship between rotor speed and generated thrust, such that the thrust produced by each motor is proportional to the square of its rotational speed. The corresponding force constant is set to 5.7658×10^{-7}, while the torque generated by each propeller is modeled as proportional to the produced thrust, with a torque constant of 0.07. The propeller radius is 0.13 m.

Figure 3.1: Drone 3D model

In addition, the vehicle is equipped with an external protective structure designed to mitigate the effects of physical interaction with the environment. This

structure enables safer contact with obstacles, preventing direct damage to the propellers and main body, and allowing controlled interaction during collision and sliding phases.

Control inputs are provided at the level of desired attitude and collective thrust. The proposed controller generates force-based commands, which are subsequently mapped to attitude and thrust references compatible with the underlying flight control interface. This abstraction reflects common practice in aerial robotics, where high-level controllers operate in the force domain and rely on an inner-loop attitude controller for stabilization.

Actuation limits are explicitly considered, including bounds on maximum thrust and allowable attitude angles. These constraints are enforced in the command generation stage to ensure physically feasible inputs.

3.1.3 Obstacle and Contact Scenario

The environment includes a planar obstacle representing a vertical wall. The wall is modeled as a rigid, immovable surface with known geometric properties but without prior explicit knowledge of its pose available to the controller.

The interaction between the vehicle and the wall is governed by a simplified contact model, which allows the generation of reaction forces upon impact and during sustained contact. This setup enables the study of collision recovery, alignment with the surface, and controlled sliding along the wall.

The chosen scenario captures the essential features of contact-rich navigation tasks, where aerial vehicles must interact with the environment rather than strictly avoid it.

3.1.4 Initial Conditions

Each experiment is initialized with the vehicle positioned at a predefined distance from the wall, with a given orientation and velocity profile. The initial conditions are selected to induce a controlled interaction with the obstacle, typically involving an approach phase that leads to contact.

Unless otherwise specified, the vehicle starts with zero angular velocity and a nominal attitude aligned with the world frame. The initial linear velocity is directed toward the wall to trigger the collision and subsequent contact phases.

Multiple runs are performed with consistent initial conditions to ensure repeatability. In some cases, variations in initial position or approach velocity are introduced to evaluate robustness.

3.1.5 Controller Configuration

The experiments are performed using the complete contact-aware control framework described in Chapter 2. The control architecture includes the following phases: collision detection and recovery, alignment with the obstacle, and sliding along the surface.

The controller operates by combining force components in the normal and tangential directions relative to the estimated wall. The normal component regulates the contact force, while the tangential component enables controlled motion along the surface. These contributions are merged and converted into attitude and thrust commands through a dedicated command generation module.

Key parameters, such as control gains, stiffness and damping coefficients, and thresholds for mode transitions, are selected empirically to achieve stable and responsive behavior. The same nominal parameter set is used across most experiments, unless explicitly varied for comparative analysis.

3.1.6Test Protocol

A series of simulation runs is conducted to evaluate the performance of the proposed framework under consistent and controlled conditions. Each run follows the same sequence: approach to the wall, collision, recovery, alignment, and sliding phase.

The duration of each experiment is sufficient to observe the full evolution of the interaction, including steady-state behavior during sliding. Performance is assessed based on the system response over time, with particular attention to stability,

convergence, and robustness during contact.

Unless otherwise stated, all tests are performed under identical conditions to ensure fair comparison between different configurations and control strategies.

3.2 External Wrench Estimation Results

3.2.1 Filtered vs Unfiltered Estimation

The results reported in Fig. 3.2 highlight the importance of filtering in the external wrench estimation process. In principle, the estimator is expected to provide a reliable indication of the external forces acting on the vehicle; however, in practice, several undesired disturbances may introduce spurious contributions in the estimate. In particular, aerodynamic drag, modelling inaccuracies, parameter uncertainties, and unmodeled effects can generate non-negligible variations in the estimated wrench even when the UAV is not in contact with any obstacle.

This issue becomes especially critical during abrupt maneuvers, where the estimator may temporarily interpret dynamic transients as external interactions with the environment. As a consequence, the system may produce false positive detections of an obstacle, thereby compromising the reliability of the contact detection logic. The use of an appropriate filter is therefore essential to attenuate these undesired components and to make the estimated external wrench more consistent with the actual interaction state of the vehicle.

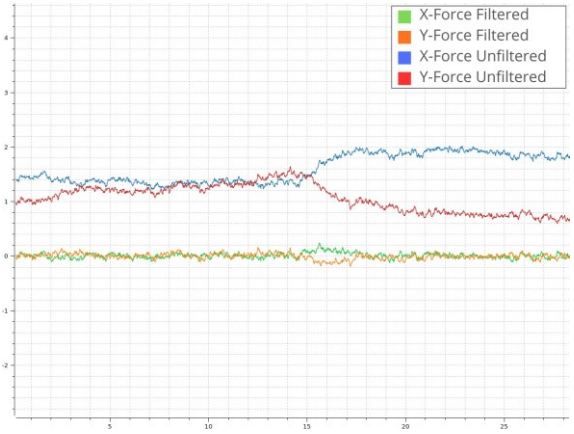

Figure 3.2: Unfiltered vs filtered external wrench estimation

The plotted results show that, in nominal flight conditions and in the absence of any physical contact, the raw estimator may still indicate a non-zero external force. This behaviour confirms that the estimated wrench is affected by disturbances unrelated to

obstacle interaction. After filtering, these effects are significantly reduced, and the estimator becomes more robust against spurious force components, allowing

a clearer separation between genuine contact events and internal or environmental disturbances.

The same considerations also justify the choice not to adopt a classical impedance control strategy in this context. Impedance control typically relies on a meaningful estimate of the interaction force between the vehicle and the environment. However, when the wrench estimate is affected by persistent disturbances such as aerodynamic drag or modelling errors, it becomes difficult to distinguish actual contact forces from artificial force components generated by the estimator itself.

Moreover, a high-pass filter is not suitable for sustained contact interactions. While such a filter may help in emphasizing transient events, it progressively attenuates low-frequency and constant components of the signal. As a result, a prolonged and continuous contact with an obstacle would be partially removed by the filter, eventually leading to an underestimation of the actual interaction force. This would make the control action unreliable and, in the case of sliding motion, could prevent the system from maintaining a consistent contact condition.

For this reason, the proposed framework does not rely on impedance control, but instead adopts a contact-aware strategy based on filtered wrench estimation, statebased supervision, and explicit handling of the interaction phases. This approach allows the controller to remain robust both in free flight and during sustained contact with the environment.

3.2.2 Contact Identification in Different Impact Configurations

The performance of the contact detection strategy is evaluated by analyzing the estimated external wrench under different collision configurations. In particular, Figures 3.3 and 3.4 report two representative cases that highlight how the interaction with the obstacle manifests in different components of the wrench depending on the relative orientation between the UAV and the wall.

In the first scenario (3.3), the UAV approaches the wall with a configuration approximately perpendicular to the surface, while being rotated by about 45° with

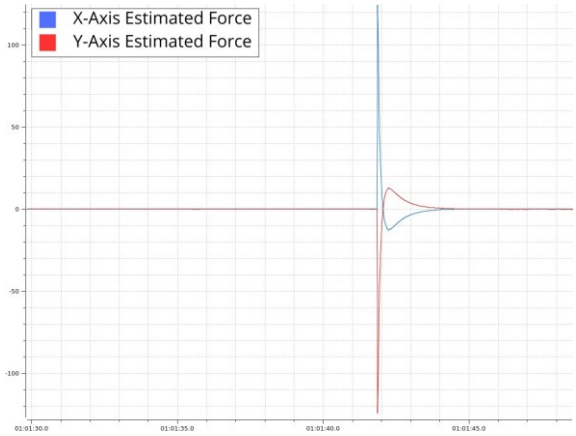

Figure 3.3: Estimated external wrench during a perpendicular impact with the wall, with the UAV rotated by approximately 45°. The contact is primarily reflected in the force components along the x and y axes, which exhibit a clear increase at the moment of impact. The moment around the z-axis remains negligible. This result shows that, in this configuration, translational force components are sufficient to reliably detect the contact event.

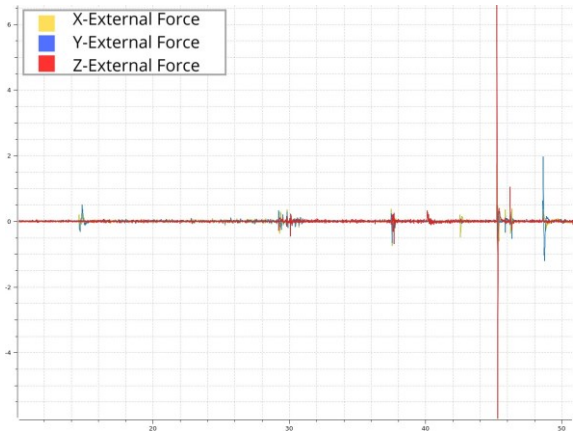

Figure 3.4: Estimated external wrench during a nearly parallel interaction with the wall. In this configuration, the force components along the x and y axes remain small and do not provide a clear indication of contact. Instead, a significant increase in the moment around the z-axis is observed, which enables the correct identification of the interaction.

This highlights the importance of considering rotational components in contact detection.

respect to it. In this condition, the contact primarily excites the translational force components along the x and y axes. As shown in the plot, these components exhibit a clear increase at the moment of impact, providing a reliable indication of contact. Conversely, the moment around the z-axis remains negligible and does not contribute significantly to the detection process. Despite this, the contact is correctly identified, demonstrating that force components alone can be sufficient when the interaction geometry predominantly generates translational effects.

A complementary behavior is observed in the second scenario (3.4), where the UAV interacts with the wall in a nearly parallel configuration. In this case, the force components along the x and y axes are relatively small and do not provide a clear signature of the interaction. Instead, the dominant contribution arises from a significant moment around the z-axis. The plot shows a marked increase in the yaw moment at the time of contact, which enables a correct identification of the interaction despite the weak translational components.

These results emphasize that the manifestation of contact in the estimated wrench strongly depends on the relative orientation between the UAV and the obstacle. As a consequence, relying on a single component of the wrench is not sufficient for robust detection. The proposed approach therefore considers both force and moment contributions, allowing the system to correctly identify contact events across different impact configurations.

This analysis validates the adopted contact detection logic, showing that it remains effective even when the dominant signature of the interaction shifts between translational forces and rotational moments. Such robustness is essential for handling unstructured contact scenarios, where the orientation of the UAV with respect to the environment cannot be assumed a priori.

3.3 Contact Recovery Results

The response of the system after contact is analyzed in two representative scenarios: a perpendicular impact with the wall and an oblique impact occurring with the UAV rotated with respect to the obstacle. In both cases, the plots show a sudden variation in the attitude variables immediately after the interaction, followed by a recovery toward the equilibrium configuration.

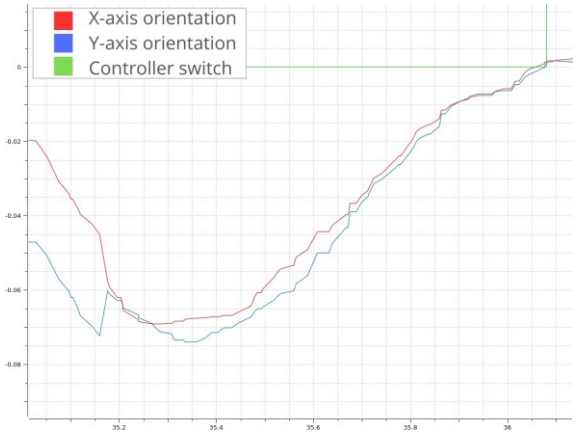

Figure 3.5: Attitude response of the UAV after a perpendicular impact with the wall. The contact induces a sudden variation in the attitude variables, but roll and pitch are rapidly driven back toward zero by the control action. The yaw angle shows a larger excursion and a slower recovery, due to the intrinsic rotational dynamics of the vehicle.

In particular, roll and pitch exhibit a rapid variation after impact, but their excursion is effectively limited by the controller action. As a result, both angles converge again to values close to zero, indicating that the vehicle is able to regain a stable attitude despite the disturbance induced by contact. This behavior confirms the effectiveness of the control law in counteracting the destabilizing effect of the collision and in restoring the vehicle to a balanced condition.

The yaw dynamics are more challenging. In both contact configurations, the yaw angle grows significantly immediately after the impact and then slowly returns to nominal values. This slower recovery can be explained by the intrinsic rotational dynamics of the UAV and by the spatial distribution of the motors, which makes yaw correction less direct than roll and pitch stabilization. Among the considered cases, this

represents the most demanding situation to handle, since the contact generates a stronger rotational disturbance around the vertical axis. Nevertheless, the system is

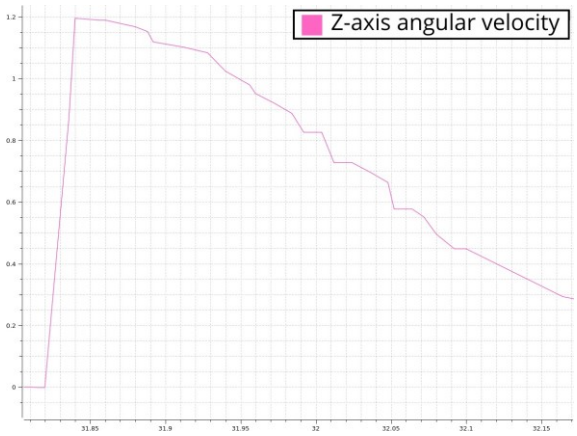

Figure 3.6: Attitude response of the UAV after an oblique impact with the wall. As in the perpendicular case, roll and pitch are effectively stabilized after the disturbance, while yaw exhibits a slower and more pronounced transient before converging back to nominal values. This represents the most challenging recovery case, yet equilibrium is successfully restored.

still able to return to a stable equilibrium, demonstrating robustness of the proposed control strategy even in the presence of severe attitude perturbations.

Overall, the results show that the controller is able to limit the attitude excursion induced by contact and to restore the equilibrium configuration in both cases. The recovery is faster for roll and pitch, while yaw exhibits a slower settling behavior due to the vehicle dynamics, but without preventing convergence to a stable state.

3.4 Sliding Phase Results

The next stage of the analysis focuses on the behavior of the system during the transition from contact recovery to sliding, and on its ability to resume the original mission once the obstacle is no longer present. This phase is particularly relevant, since it verifies whether the controller is not only capable of handling the contact event itself, but also

of maintaining continuity with the pre-planned trajectory after the interaction has ended.

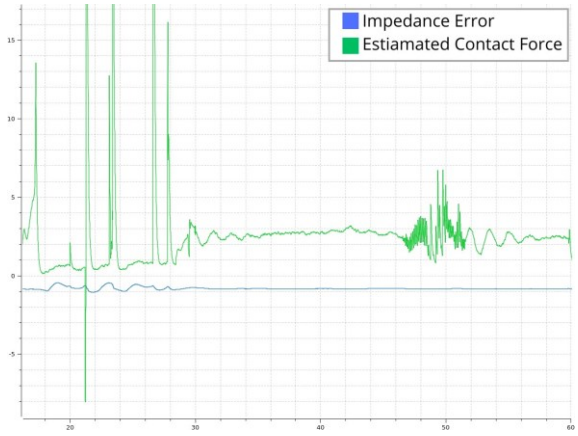

Figure 3.7: Comparison between impedance-based and admittance-based formulations during the sliding phase. The impedance approach produces a velocity reference that leads to a persistent tracking error, effectively keeping the UAV attached to the wall. The admittance approach generates a response driven by the external wrench estimator, but its reliability is limited by estimation imperfections, which affect the consistency of the resulting command.

The first comparison, reported in 3.7, contrasts the use of impedance and admittance-based approaches in the sliding context. The results show that the impedance formulation leads to a velocity reference that produces a persistent tracking error. In practice, this behaviour corresponds to a condition in which the UAV remains effectively “stuck” to the wall, since the generated reference does not allow a proper detachment or a smooth transition along the obstacle surface. This is consistent with the fact that the control action remains strongly constrained by the contact interaction, preventing the vehicle from recovering a fully autonomous motion.

On the other hand, the admittance-based approach produces a response that is driven by an external wrench estimator. Although this strategy is conceptually

suitable for contact-rich interaction, in the considered case the estimator is not perfectly accurate and its output depends only partially on the actual external force. As a result, the corresponding reference cannot be considered fully reliable. In particular, estimation imperfections may lead to undesired variations in the generated command, which affects

the consistency of the sliding motion and limits the robustness of the overall behavior. This comparison supports the decision to avoid a purely force-driven formulation and motivates the adoption of a more structured contact-aware control strategy.

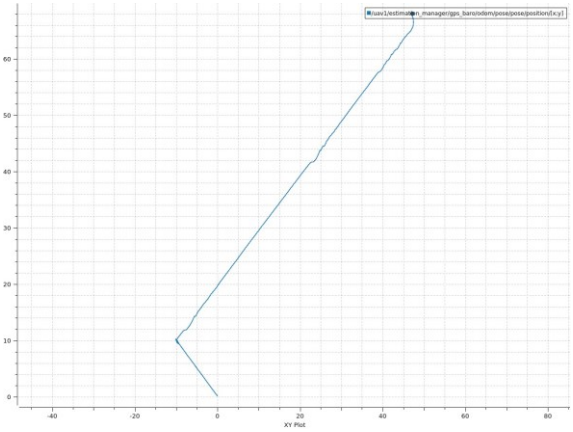

Figure 3.8: Cartesian trajectory of the UAV during the complete mission. The vehicle first follows the nominal path, then encounters the obstacle, recovers equilibrium, performs lateral sliding along the wall, and finally resumes the original mission after the obstacle is no longer present. This result demonstrates the correct integration of the contact phase within the overall navigation pipeline.

The second plot, reported in 3.8, shows the Cartesian trajectory of the UAV during the complete mission. The vehicle initially follows a straight path corresponding to the main task. After the encounter with the obstacle, the UAV successfully recovers equilibrium, starts sliding laterally along the wall, and finally detects that the obstacle is no longer present. At this point, the controller enables a smooth return to the original mission, demonstrating that the contact phase is properly integrated within the overall navigation pipeline.

This result is particularly important because it confirms that the proposed framework does not treat contact as an isolated event, but rather as one phase of a broader mission execution process. The ability to recover from impact, maintain controlled sliding, and then resume the nominal trajectory shows that the system

3.4. SLIDING PHASE RESULTS

can manage the obstacle interaction without losing the continuity of the task. In other words, the UAV is able to preserve the overall mission objective despite the temporary disruption caused by the wall.

To further validate this behavior, twenty experimental trials were carried out under identical conditions. In all cases, the system successfully completed the mission, yielding a success rate of 100%. This outcome confirms the repeatability of the proposed solution and highlights its robustness with respect to the considered contact scenario.

Chapter 4

Conclusion and Future Work

4.1 Conclusions

The objective of this research was to develop a contact-aware control framework that enables a UAV equipped with a minimal onboard sensing suite, and in particular relying on an IMU-based estimation pipeline, to correctly detect an unexpected collision with an unknown obstacle, recover stability after the impact, approach the obstacle in a safe and controlled manner, and finally slide along its surface until the end of the contact region is identified.

The results obtained in simulation show that the proposed external wrench estimation strategy is able to correctly detect impacts of sufficient magnitude and different directions. In particular, the estimator provides a reliable indication of collision events when the interaction generates a clear dynamic signature, allowing the supervisory logic to distinguish actual contact from nominal flight transients. At the same time, the reaction of the control system after impact is immediate and effective: once the collision is detected, the recovery logic prevents the vehicle from diverging into unstable configurations and drives it back toward an equilibrium condition. This behavior is especially important because a delayed response after impact could easily lead to attitude loss and make recovery impossible.

A second key contribution of the proposed framework is the controlled approach

phase. This phase makes it possible to establish a first safe interaction with the obstacle and to transition smoothly from free flight to contact handling. Beyond its operational role, the approach phase also provides useful information about the spatial position of the obstacle, which becomes fundamental for the subsequent sliding stage. In this way, contact is not treated as a failure mode, but as an informative event that can be exploited to improve the overall mission execution.

The sliding phase represents the most distinctive part of the proposed pipeline. During this stage, the UAV remains in sustained contact with the wall, which allows the system to continuously verify the presence of the obstacle and to maintain a consistent interaction with it. This behavior also makes it possible to collect geometric information about the environment that can later be exploited for navigation or mapping purposes. Moreover, the vehicle is able to translate along the obstacle with limited oscillations, maintaining a stable contact condition and recovering equilibrium whenever perturbations occur. This confirms that the control strategy is not only capable of resisting the contact disturbance, but also of actively supporting a controlled motion along the obstacle surface.

Overall, the work provides a relevant contribution in the context of contactaware UAV control. A vehicle equipped with a basic inertial sensing infrastructure can, at least under the assumptions considered in this thesis, survive an unintended impact, recover stability, negotiate the obstacle, and continue its mission after the contact phase. Without such a framework, the same event could force the vehicle to abort the mission, remain unstable after the collision, or continue flying in a compromised state, with a high risk of repeated impacts and structural damage. The proposed approach is also particularly promising for narrow environments, where even a small trajectory error may lead to accidental contact with nearby surfaces. In addition, the same principle can be exploited intentionally for inspection tasks, for instance when the UAV needs to interact with walls or other structures in a controlled manner.

4.2 Future Work

Several directions can be identified for extending and improving the proposed framework. A first natural step concerns the external wrench estimator. Although the

current solution is effective in detecting relevant impact events, it is still influenced by undesired disturbances such as aerodynamic effects, modelling errors, and transient dynamics. Future work should therefore focus on improving the estimator robustness, reducing false positives, and increasing the consistency between the estimated wrench and the actual physical interaction. A more accurate estimator would also strengthen the reliability of the subsequent contact logic and make the overall system easier to tune.

A second important extension concerns the interaction model used during the sliding phase. The current approach is designed for planar surfaces, and therefore its effectiveness may decrease in the presence of more complex geometries, such as persistent concavities, curved surfaces, or highly irregular obstacles. A promising direction would be the development of a geometry-adaptive controller capable of handling different types of surfaces while maintaining stable contact and coherent motion along the obstacle. In this sense, a more general surface-tracking or local shape-adaptation mechanism could significantly broaden the applicability of the method.

Further improvements could also target the transition logic between the different mission phases. In particular, more advanced criteria for switching between recovery, alignment, sliding, and mission resumption could reduce sensitivity to noisy measurements and increase robustness in cluttered environments. Similarly, a richer representation of the obstacle, possibly derived from contact-induced sensing information, could support online environment reconstruction and improve the decision-making capabilities of the vehicle.

From an experimental perspective, the framework should be validated beyond simulation, by testing it on a real aerial platform under realistic sensing noise, actuator saturation, and unmodelled disturbances. Real-world experiments would provide a more complete assessment of the practical limits of the approach and would help identify additional issues related to contact compliance, mechanical tolerances, and estimator tuning.

Finally, future work may also consider deliberate contact-based inspection tasks. In such scenarios, the vehicle would intentionally interact with the environment to explore, map, or inspect surfaces while preserving stability and mission continuity. This would

open the way to applications in infrastructure inspection, confined-space navigation, and robotic exploration in partially known environments.

In conclusion, the proposed framework demonstrates that contact-aware control can significantly improve the resilience of UAVs operating in constrained environments. By combining contact detection, post-collision stabilization, safe approach, and controlled sliding, the system provides a foundation for more robust aerial robots capable of handling unexpected interactions with the environment without losing mission continuity.

Ringraziamenti

Ringrazio profondamente i miei genitori, Silvia ed Enrico, per il costante supporto che mi hanno sempre dimostrato, per essermi stati accanto anche nei momenti più difficili e per aver sponsorizzato questo percorso. Grazie per tutti i sacrifici fatti per permettermi di arrivare fin qui e per i valori che mi avete trasmesso: questo traguardo è anche vostro.

Un grazie a mio fratello Samuele per la sua presenza costante e per aver sempre trovato il modo di tirarmi su il morale nei momenti no.

Ringrazio i miei nonni, Lucia e Graziano, per l'affetto con cui mi hanno sempre fatto sentire speciale, per i loro sorrisi, per il "Crepi" prima di ogni esame e per i saggi consigli che mi hanno accompagnato lungo il mio percorso.

Ai miei nonni, Dirce e Valter, sempre nei miei pensieri, che da lassù mi proteggono e mi aiutano a prendere le decisioni più difficili: questo è anche vostro.

Un ringraziamento speciale a Giada, la mia metà da oltre un decennio, per avermi sempre sostenuto e spronato a superare ogni ostacolo, per aver saputo comprendere anche i miei silenzi e per essermi stata accanto, vicina o lontana, in ogni momento.

Ringrazio i miei zii, Titti, Gabri, Rita e Fede, per la loro allegria, per avere sempre una battuta pronta a strapparmi un sorriso, per il supporto lungo questo percorso e per aver sempre creduto in me.

Un grazie ai miei cugini, Giuli, Edo, Lula, Simo e Pippo, per i momenti di spensieratezza, le risate e per aver reso questo percorso ancora più leggero.

Ringrazio i miei amici, Kev, Ronda, Paglia, Roman, Sofy, Bozz, Briga e Giuli, per avermi fatto vivere momenti indimenticabili, per esserci sempre quando ho bisogno, per capirmi con uno sguardo e, soprattutto, per concordare che ho sempre ragione.

Un ringraziamento a tutte le persone che, anche con un piccolo gesto o una parola, hanno contribuito a rendere questo percorso possibile.

Un sentito ringraziamento al mio relatore, il Prof. Lorenzo Sabattini, per la disponibilità, il supporto e i preziosi consigli che hanno contribuito in modo significativo alla realizzazione di questo lavoro.

Bibliography

[1] T. Tomić, C. Ott, and S. Haddadin. “External Wrench Estimation, Collision Detection, and Reflex Reaction for Flying Robots”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017.

[2] T. Tomić and S. Haddadin. “Simultaneous Estimation of Aerodynamic and Contact Forces in Flying Robots: Applications to Metric Wind Estimation and Collision Detection”. In: *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 5290–5296.

[3] R. Wang et al. *Air Bumper: A Collision Detection and Reaction Framework for Autonomous MAV Navigation*. arXiv preprint. 2023.

[4] J. Hughes and D. Lyons. *Wall Detection Via IMU Data Classification In Autonomous Quadcopters*. arXiv preprint. 2021.

[5] K. Patnaik et al. “Collision Recovery Control of a Foldable Quadrotor”. In: *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2021, pp. 418–423.

[6] W. C. Myeong et al. “Development of a Drone-Type Wall-Sticking and Climbing Robot”. In: *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. 2015, pp. 386–389.

[7] H. Lee et al. “CAROS-Q: Climbing Aerial RObot System Adopting Rotor Offset With a Quasi-Decoupling Controller”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 8490–8497.

[8] W. Myeong and H. Myung. “Development of a Wall-Climbing Drone Capable of Vertical Soft Landing Using a Tilt-Rotor Mechanism”. In: *IEEE Access* 7 (2019), pp. 4868–4879.

[9] T. Lee, M. Leok, and N. H. McClamroch. “Geometric Tracking Control of a Quadrotor UAV on SE(3)”. In: *49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 5420–5425.

[10] T. Baca et al. “The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-World Deployment, and Education with Autonomous Unmanned Aerial Vehicles”. In: *Journal of Intelligent & Robotic Systems* 102.1 (2021), p. 26.