Model-Based Control of Flying Robots for Robust Interaction under Wind Influence

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades
Doktor-Ingenieur (abgekürzt: Dr.-Ing.)
genehmigte Dissertation

von

M. Sc. Teodor Tomić

geboren am 4. August 1986
in Požega, Kroatien

2018

Vorsitzender	Prof. Dr.-Ing. Axel Mertens
1. Referent	Prof. Dr.-Ing. Sami Haddadin
2. Referent	Prof. Dr.-Ing. Holger Blume
Tag der Promotion	25. September 2018 *)

*) Datum der mündlichen Doktorprüfung

Kurzfassung

Modellbasierte Regelung von Flugrobotern für robuste physikalische Interaktion unter Windeinfluss

Das Hauptziel dieser Arbeit ist es, die Lücke zwischen Trajektorienverfolgung und Interaktionsregelung für Flugroboter zu schließen, um eine möglichst exakte physikalische Interaktion sogar unter Windeinfluss zu ermöglichen. Von zentraler Bedeutung ist es, dass die Roboter Störungen, Interaktionen und Fehler, die auf sie einwirken, erkennen können und zielgerichtet auf diese reagieren. Konkret wird dies dadurch erreicht, dass extern auf den Roboter einwirkende Kraftwinder (Kraft und Drehmoment), modellbasiert gefolgert wird und zwischen den kausalen Ursachen Wind, Interaktionen und Kollisionen diskriminiert (unterschieden) wird. Dieses grundsätzliche Ziel wirft folgende Forschungsfragen auf. Erstens, ist eine simultane Schätzung und Unterscheidung zwischen externen Kraftwinderkomponenten für die Nutzung in der Regelung kontinuierlich und in Echtzeit möglich? Zweitens, was sind geeignete effektive Regelungsmethoden für die Interaktion und Trajektorienverfolgung unter bestimmten Windbedingungen und physikalischer Interaktion? Drittens: wie können unerwartete Fehler, wie Kollisionen mit der Umwelt, effizient und effektiv erkannt und behandelt werden? Im Rahmen der ersten Frage kann eine vierte Frage gestellt werden: Ist es grundsätzlich möglich, eine vom externen Kraftwinder unabhängige Messung bzw. Schätzung der Windgeschwindigkeit zu erhalten? In dieser Arbeit werden modellbasierte Methoden angewendet, um spezifische Antworten auf diese Fragen zu finden. Dies erfordert sowohl ein gutes Dynamikmodell des Roboters sowie genau identifizierte Parameter. Daher wird ein in dieser Güte bisher unerreichtes neues systematisches Parameteridentifizierungsverfahren für Flugroboter entwickelt und angewendet. Ferner werden etablierte Methoden aus der Robotik zu Schätzung des externen Kraftwinders für Flugroboter geeignet erweitert, ohne dass externe Geschwindigkeitsmessungen erforderlich sind. Basierend auf der Schätzung des externen Kraftwinders werden Methoden der Interaktionsregelung (Impedanz- und Admittanzregelung) auf Flugroboter erweitert und ein gründlicher Stabilitätsbeweis gegeben. Die Schätzung wird auch in einem geometrischen Trajektorienverfolgungsregler angewendet, um externe Störungen zu kompensieren, um so den Schleppfehler unter Windeinfluss zu minimieren, ohne Notwendigkeit eines Integralterms. Die genannten Regler werden zu einem neuartigen Impedanzregler kombiniert. Ferner wird die Kollisionserkennung auf fliegende Roboter erweitert, wodurch eine Reflexreaktion zur Erhöhung der Sicherheit dieser autonomen Roboter erreicht wurde. Um aerodynamische Modelle zur Windgschätzung zu identifizieren und vergleichen, wurden Flugversuche in einem dreidimensionalen Windkanal mit einem Hexacopter durchgeführt. Es wird gezeigt, dass eine gute Modellgüte bereits mit relativ einfachen linearen Regressionsmodellen erhalten werden kann. Ferner wird das Propeller-Aerodynamikmodell verwendet, um aus verfügbaren Motorleistungsmesswerten die Windgeschwindigkeit zu schätzen. Darüber hinaus wurde eine neuartige, optimierungsbasierte Methode entwickelt, die das Propeller-Aerodynamikmodell nutzt, um die Windgeschwindigkeit anhand der aerodynamischen Leistung zu schätzen. Im Wesentlichen verwenden diese beiden Verfahren die Propeller als Windgeschwindigkeitssensoren, so dass eine Messung bereitgestellt wird, die unabhängig von der externen Kraft ist. Schließlich wird in dieser Arbeit erstmals das neuartige Thema der Diskriminierung zwischen aerodynamischen, Interaktions- und Fehlerkraftwindern systematisch behandelt. Dies ermöglicht schließlich die Implementierung eines Reglers, der bei physikalischen Interaktion gewollt nachgiebiges Verhalten erzeugt, Windstörungen jedoch ausgleicht.

Schlagwörter: Flugroboter, Interaktionsregelung, Fehlererkennung und -isolierung

Abstract

Model-Based Control of Flying Robots for Robust Interaction under Wind Influence

The main goal of this thesis is to bridge the gap between trajectory tracking and interaction control for flying robots in order to allow *physical interaction under wind influence* by making aerial robots aware of the disturbance, interaction, and faults acting on them. This is accomplished by reasoning about the external wrench (force and torque) acting on the robot, and *discriminating* (distinguishing) between wind, interactions, and collisions. This poses the following research questions. First, is discrimination between the external wrench components even possible in a continuous real-time fashion for control purposes? Second, given the individual wrench components, what are effective control schemes for interaction and trajectory tracking control under wind influence? Third, how can unexpected faults, such as collisions with the environment, be detected and handled efficiently and effectively? In the interest of the first question, a fourth can be posed: is it possible to obtain a measurement of the wind speed that is independent of the external wrench? In this thesis, model-based methods are applied in the pursuit of answers to these questions. This requires a good dynamics model of the robot, as well as accurately identified parameters. Therefore, a systematic parameter identification procedure for aerial robots is developed and applied. Furthermore, external wrench estimation techniques from the field of robot manipulators are extended to be suitable for aerial robots without the need of velocity measurements, which are difficult to obtain in this context. Based on the external wrench estimate, interaction control techniques (impedance and admittance control) are extended and applied to flying robots, and a thorough stability proof is provided. Similarly, the wrench estimate is applied in a geometric trajectory tracking controller to compensate external disturbances, to provide zero steady-state error under wind influence without the need of integral control action. The controllers are finally combined into a novel *compensated impedance controller*, to facilitate the main goal of the thesis. Collision detection is applied to flying robots, providing a low level reflex reaction that increases safety of these autonomous robots. In order to identify aerodynamic models for wind speed estimation, flight experiments in a three-dimensional wind tunnel were performed using a custom-built hexacopter. This data is used to investigate wind speed estimation using different data-driven aerodynamic models. It is shown that good performance can be obtained using relatively simple linear regression models. In this context, the propeller aerodynamic power model is used to obtain information about wind speed from available motor power measurements. Leveraging the wind tunnel data, it is shown that power can be used to obtain the wind speed. Furthermore, a novel optimization-based method that leverages the propeller aerodynamics model is developed to estimate the wind speed. Essentially, these two methods use the propellers as wind speed sensors, thereby providing an additional measurement independent of the external force. Finally, the novel topic of simultaneously discriminating between aerodynamic, interaction, and fault wrenches is opened up. This enables the implementation of novel types of controllers that are e.g. compliant to physical interaction, while compensating wind disturbances at the same time. The previously unexplored force discrimination topic has the potential to even open a new research avenue for flying robots.

Keywords: flying robots, interaction control, fault detection and identification

Acronyms

Acronym	**Description**
AIBC	Adaptive Integral Backstepping Control
ADRC	Active Disturbance Rejection Control
DO	Disturbance Observer
DMP	Dynamic Movement Primitive
FDI	Fault Detection, Isolation and Identification
FDIR	Fault Detection, Isolation, Identification and Reaction
GPS	Global Positioning System
IMU	Inertial Measurement Unit
ISS	Input-to-State Stability
ISM	Integral Sliding Mode
LMI	Linear Matrix Inequality
LQR	Linear Quadratic Regulator
MAV	Micro Aerial Vehicle
MPC	Model Predictive Control
NLS	Nonlinear Least Squares
PD	Proportional-Derivative control
PID	Proportional-Integral-Derivative control
SIR	Sequential Importance Resampling
VIO	Visual-inertial odometry
UAV	Unmanned Aerial Vehicle

Notation

Symbol	Dim.	Units	Description
x	$\mathbb{R}$	-	Scalar
$\boldsymbol{x}$	$\mathbb{R}^N$	-	N-dimensional vector
$\dot{\boldsymbol{x}}$	$\mathbb{R}^N$	-	Time derivative of $\boldsymbol{x}$, $\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{x}(t)}{\partial t}$
$\lVert\boldsymbol{x}\rVert$	$\mathbb{R}$	-	Norm of vector $\boldsymbol{x}$
$\boldsymbol{A}$	$\mathbb{R}^{N\times M}$	-	$N \times M$ matrix
$\hat{\cdot}$	$\mathbb{R}^N$	-	Estimate of quantity represented by $\cdot$
$\tilde{\cdot}$	$\mathbb{R}^N$	-	Error of quantity represented by $\cdot$
$\mathbf{0}$	$\mathbb{R}^N$	-	Zero vector of size N
$\mathbf{0}_{N\times N}$	$\mathbb{R}^{N\times N}$	-	Zero matrix
A	$\mathbb{R}$	m^2	Propeller surface area
$\boldsymbol{A}_d$	$\mathbb{R}^{3\times 3}$	s/m	Blade flapping and induced drag matrix
$\boldsymbol{a}$	$\mathbb{R}^3$	$\mathrm{m/s}^2$	Acceleration
C_T	$\mathbb{R}$	-	Propeller thrust coefficient
C_Q	$\mathbb{R}$	-	Propeller torque coefficient
$\boldsymbol{C}(\boldsymbol{\nu})$	$\mathbb{R}^{6\times 6}$	-	Matrix of centripetal and Coriolis terms
D	$\mathbb{R}$	m	Propeller diameter
D_r	$\mathbb{R}$	Nm	Rotor drag torque
$\boldsymbol{e}_i$	$\mathbb{R}^3$	-	Unit vector for axis i, i.e. $\boldsymbol{e}_2 = [0,1,0]^T$
η	$\mathbb{R}$	-	Scalar part of quaternion
$\boldsymbol{\epsilon}$	$\mathbb{R}^3$	-	Vector part of quaternion
$\boldsymbol{f}$	$\mathbb{R}^3$	N	Force; control force in the body frame
$\boldsymbol{f}_c$	$\mathbb{R}^3$	N	Collision force
$\boldsymbol{f}_d$	$\mathbb{R}^3$	N	Aerodynamic (drag) force
$\boldsymbol{f}_e$	$\mathbb{R}^3$	N	External force
$\boldsymbol{f}_i$	$\mathbb{R}^3$	N	Interaction force
FM	$\mathbb{R}$	-	Propeller figure of merit, $FM \in (0,1)$
Γ	$\mathbb{R}^{3\times 3}$	-	Ratio of actual and virtual inertia $\mu = \boldsymbol{\mathcal{I}}\boldsymbol{\mathcal{I}}_v^{-1}$
$\boldsymbol{I}$	$\mathbb{R}^{N\times N}$	-	Identity matrix
i_a	$\mathbb{R}$	A	Motor current
$\boldsymbol{\mathcal{I}}$	$\mathbb{R}^{3\times 3}$	$\mathrm{kg\,m}^2$	Rigid body inertia

I_r	$\mathbb{R}$	$\mathrm{kg\,m^2}$	Rotor inertia
$\boldsymbol{J}$	$\mathbb{R}^{N\times M}$	-	Jacobian (depends on context)
$\boldsymbol{K}$	$\mathbb{R}^{N\times N}$	-	Gain matrix (depends on context)
K_q	$\mathbb{R}$	Nm/A	Motor torque constant
$\lambda_{\max}\{\boldsymbol{A}\}$	$\mathbb{R}$	-	Maximum eigenvalue of matrix $\boldsymbol{A}$
$\lambda_{\min}\{\boldsymbol{A}\}$	$\mathbb{R}$	-	Minimum eigenvalue of matrix $\boldsymbol{A}$
$\mathcal{M}$	$\mathbb{R}$	kg	Actual rigid body mass
$\mathcal{M}_0$	$\mathbb{R}$	kg	Nominal mass
$\mathcal{M}_v$	$\mathbb{R}$	kg	Virtual mass
μ	$\mathbb{R}$	-	Ratio of actual and virtual mass $\mu = \mathcal{M}\mathcal{M}_v^{-1}$
$\boldsymbol{M}$	$\mathbb{R}^{6\times 6}$	kg, $\mathrm{kg\,m^2}$	Generalized rigid body inertia tensor
$\boldsymbol{m}$	$\mathbb{R}^3$	Nm	Torque; control torque in the body frame
$\boldsymbol{m}_d$	$\mathbb{R}^3$	Nm	Aerodynamic (drag) torque
$\boldsymbol{m}_e$	$\mathbb{R}^3$	Nm	External torque
$\boldsymbol{m}_i$	$\mathbb{R}^3$	Nm	Interaction torque
$\boldsymbol{\omega}$	$\mathbb{R}^3$	rad/s	Angular velocity of a rigid body
ϖ	$\mathbb{R}$	s^{-1}	Angular velocity of a propeller
$\boldsymbol{\varpi}$	$\mathbb{R}^{N_p}$	s^{-1}	Angular velocity of N_p propellers
$\boldsymbol{\omega}_d$	$\mathbb{R}^3$	rad/s	Desired angular velocity
p	$\mathbb{R}$	Pa	Air pressure
$\boldsymbol{p}$	$\mathbb{R}^6$	J	Generalized momentum $\boldsymbol{p} = \boldsymbol{M}\boldsymbol{\nu}$
P_m	$\mathbb{R}$	W	Motor power
P_a	$\mathbb{R}$	W	Aerodynamic power of a propeller
$\hat{P}_a$	$\mathbb{R}$	W	Estimated aerodynamic power of a propeller
P_h	$\mathbb{R}$	W	Aerodynamic power of a propeller in hover
Q	$\mathbb{R}$	Nm	Propeller torque
$\underline{\boldsymbol{q}}$	$SO(3)$	-	Unit quaternion $\underline{\boldsymbol{q}} = [\eta\,\boldsymbol{\epsilon}^T]^T$
$\boldsymbol{r}$	$\mathbb{R}^3$	m	Position in the inertial frame
$\boldsymbol{r}_d$	$\mathbb{R}^3$	m	Desired position in the inertial frame
$\boldsymbol{R}$	$SO(3)$	-	Rotation matrix from body to inertial frame
$\boldsymbol{R}_{ab}$	$SO(3)$	-	Rotation matrix from frame B to frame A
R_{air}	$\mathbb{R}$	J/kg K	Ideal gas constant of dry air
ρ	$\mathbb{R}$	$\mathrm{kg/m^3}$	Air density
$\boldsymbol{\rho}$	$\mathbb{R}^6$	J	Residual
$\boldsymbol{s}$	$\mathbb{R}^3$	-	Tracking variable
$\mathcal{S}(\cdot)$	$\mathbb{R}^{3\times 3}$	-	Skew-symmetric matrix operator, $\mathcal{S}(\boldsymbol{a})\,\boldsymbol{b} = \boldsymbol{a}\times\boldsymbol{b}$
$\boldsymbol{\theta}$	$\mathbb{R}^{N_p}$	-	Parameter vector
T	$\mathbb{R}$	N	Propeller thrust
$\boldsymbol{\tau}$	$\mathbb{R}^6$	N, Nm	Force; control force in the body frame
$\boldsymbol{\tau}_c$	$\mathbb{R}^6$	N, Nm	Collision wrench
$\boldsymbol{\tau}_d$	$\mathbb{R}^6$	N, Nm	Aerodynamic (drag) wrench
$\boldsymbol{\tau}_e$	$\mathbb{R}^6$	N, Nm	External wrench
$\boldsymbol{\tau}_i$	$\mathbb{R}^6$	N, Nm	Interaction wrench

$\boldsymbol{\tau}_f$	$\mathbb{R}^6$	N, Nm	Fault wrench
τ_m	$\mathbb{R}$	Nm	Motor torque
U	$\mathbb{R}$	m/s	Velocity of the propeller slipstream
V	$\mathbb{R}$	-	Lyapunov function
v_h	$\mathbb{R}$	m/s	Induced velocity of a propeller, in hover
v_i	$\mathbb{R}$	m/s	Induced velocity of a propeller
$\boldsymbol{v}_r$	$\mathbb{R}^3$	m/s	Relative airspeed in the inertial frame
$\boldsymbol{v}_w$	$\mathbb{R}^3$	m/s	Wind velocity in the inertial frame
$\boldsymbol{v}_\infty$	$\mathbb{R}^3$	m/s	Propeller freestream velocity
$\boldsymbol{\nu}$	$\mathbb{R}^6$	m/s, rad/s	Generalized velocity $\boldsymbol{\nu} = [\dot{\boldsymbol{r}}^T \, \boldsymbol{\omega}^T]^T$
$\boldsymbol{\nu}_d$	$\mathbb{R}^6$	m/s, rad/s	Desired generalized velocity $\boldsymbol{\nu} = [\dot{\boldsymbol{r}}_d^T \, \boldsymbol{\omega}_d^T]^T$
$\boldsymbol{w}$	$\mathbb{R}^3$	m/s	Propeller slipstream velocity
$\boldsymbol{x}$	$\mathbb{R}^N$	-	State vector (depending on context)
$\boldsymbol{Y}$	$\mathbb{R}^{M \times N_p}$	-	Regression matrix for N_p parameters and M measurements
$\boldsymbol{z}$	$\mathbb{R}^M$	-	Measurement vector

Contents

1 Introduction 1
- 1.1 Problem statement . . . 2
- 1.2 Related Work . . . 3
- 1.3 Contributions . . . 11

2 Modeling 17
- 2.1 Dynamics model . . . 18
  - 2.1.1 Rigid body dynamics . . . 18
  - 2.1.2 Propulsion wrench . . . 18
  - 2.1.3 Propeller aerodynamics . . . 20
  - 2.1.4 Simplified drag model . . . 22
  - 2.1.5 Reduced brushless DC motor model . . . 22
- 2.2 Parameter identification . . . 23
- 2.3 External wrench estimation . . . 26
  - 2.3.1 Momentum-based estimation . . . 27
  - 2.3.2 Acceleration-based estimation . . . 28
  - 2.3.3 Hybrid estimation . . . 28

3 Tracking and interaction control 31
- 3.1 Trajectory tracking control . . . 31
  - 3.1.1 Attitude tracking control . . . 31
  - 3.1.2 Position tracking control . . . 35
  - 3.1.3 Evaluation of trajectory tracking . . . 37
- 3.2 Interaction control . . . 42
  - 3.2.1 Impedance control with inertia shaping . . . 42
  - 3.2.2 Compensated impedance control . . . 48
  - 3.2.3 Admittance control . . . 49
  - 3.2.4 Discussion and practical considerations . . . 49
  - 3.2.5 Experimental validation of impedance control . . . 50
  - 3.2.6 ISS numerical example . . . 50

4 Wind estimation 55
- 4.1 Inversion-based metric wind estimation . . . 55
- 4.2 Wind tunnel experiments . . . 56
- 4.3 Aerodynamic model evaluation . . . 58
  - 4.3.1 Propeller aerodynamic power . . . 60
  - 4.3.2 Aerodynamic models . . . 61
  - 4.3.3 Model performance . . . 63

4.3.4 Model generalization 65
4.3.5 Aerodynamic torque models 69
4.4 Physics model based wind estimation 69
4.4.1 Comparison to data-driven estimation 73
4.4.2 Choosing optimal measurements 75
4.4.3 Jacobian of the optimization problem 77

5 Force discrimination **79**
5.1 Problem statement 79
5.2 Collision detection under wind influence 80
5.3 Contact location 86
5.4 Contact detection under wind influence 86
5.5 Modified model-checking force discrimination scheme 89
5.6 Interaction force at known contact position 91
5.7 Particle filter-based force discrimination 93
5.8 Power-based discrimination 97
5.9 Kalman filter-based force discrimination 99
5.10 Summary 101

6 Applications and outlook **103**
6.1 Collision detection applications 103
6.1.1 Collision reaction and location 103
6.1.2 Tactile mapping 106
6.2 Interaction, disturbance, and fault aware flying robot swarms 107
6.2.1 Swarm system dynamics 109
6.2.2 Awareness pipeline and interaction control 110
6.2.3 Disturbance awareness pipeline 111
6.2.4 Conclusions 116

7 Conclusion **119**

Bibliography **123**

Appendix **141**

Publications **141**

Curriculum Vitae **143**

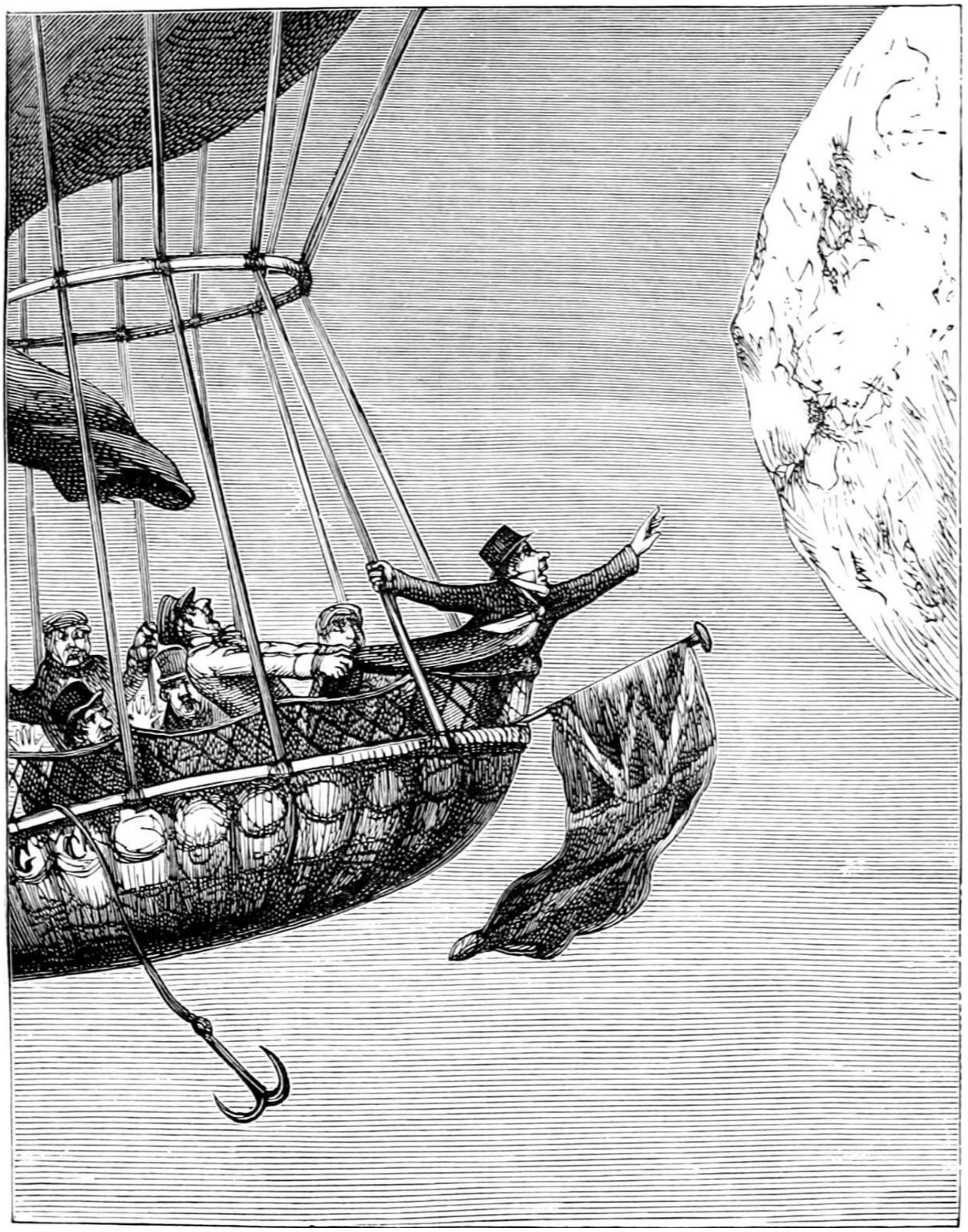
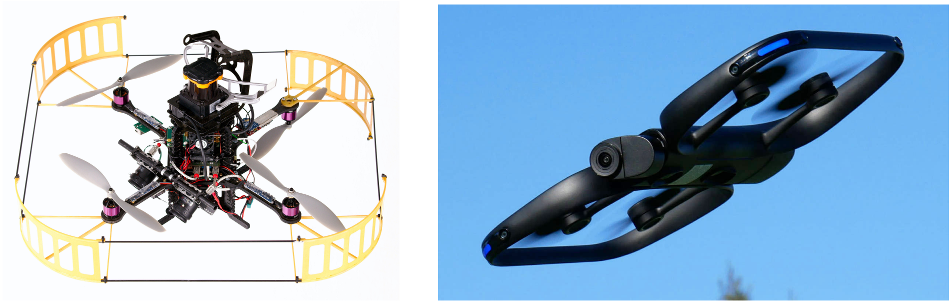


Illustration: G. Cruikshank: Monstre Balloon; Book: The Ingoldsby legends, New York: Scribner and Welford, n.d.

CHAPTER 1

Introduction



(a) DLR's AscTec Pelican (2011). Copyright © 2012 IEEE [143].

(b) Skydio R1 (2018) in flight [128]

Figure 1.1: The techniques developed in the framework of this thesis have been applied and experimentally verified on the depicted research and commercial multirotor flying robots.

The continuation of Moore's law and resulting proliferation of cheap and powerful computational hardware in a small form factor, coupled with advances in sensors and state estimation algorithms, has dramatically increased the autonomy of small-scale flying robots in recent years. Autonomy in the context of this thesis is understood as the ability of robots to operate in unknown, uncertain and cluttered environments without the intervention of a human operator. This includes estimating the robot's pose, mapping the environment, trajectory generation, and decision-making. The key to autonomy is the fusion of exteroceptive sensors, such as cameras and laser range finders, with proprioceptive sensors such as an inertial measurement unit (IMU). The combination provides a high-rate estimate of the robot's motion using the IMU, while stabilizing the pose and mapping can be done at a lower rate. Processing large amounts of data from exteroceptive sensors such as cameras in real time during flight is a computationally demanding task. For example, state of the art in 2011, see Tomić et al. [143] and Figure 1.1(a) and further developments in Schmid et al. [123], was a 2 kg quadcopter carrying a laser range finder and one sterao pair, while computation was divided among three single-core ARM computers, and one Atom computer. This was enough to run a low-resolution stereo algorithm, visual odometry, laser odometry, and an extended Kalman filter for state estimation. Due to the total system weight of the prototyping platform, the flight time was less than 10 minutes. The price of hardware alone (quadrotor platform, laser sensor, cameras, computers) was around 10,000 USD. Several years later, the Skydio R1 [128] in 2018, see Figure 1.1(b), contains an Nvidia Jetson TX1 computer with four ARM cores, and a 256-core GPU, which allows the real-time computation of 6 stereo pairs, occupancy grid mapping, motion planning, visual odometry, state estimation, and person tracking (using deep learning). This comes in a portable 1.1 kg quadrotor platform, largely due to the weight of the computational hardware. The price of this consumer robot is 2500 USD. Less capable (i.e.with limited or nonexistent obstacle avoidance) aerial robots are in the price range of 200 to 1000 USD and weight several hundred grams. So, even though powerful computational capabilities are coming in smaller packages, there is still a long way before miniature (<250 g) and affordable (50 USD) fully autonomous flying robots become ubiquitous. The limit of 250 g is due to the lack of regulation requirements below this mass, as the robots are deemed safe in the event of a crash [1].

It is notable that even though resource-hungry perception and planning algorithms have improved substantially, low-level control algorithms have remained in the domain of microcontroller-level computational demands. Despite significant advances in e.g. Model Predictive Control (MPC) based controllers for flying robots, they have not yet been widely deployed due to their computational complexity. Furthermore, control algorithms have mainly focused on two opposing areas: trajectory tracking and interaction control. The aim of the former is to follow a desired trajectory as close as possible under the influence of external disturbances and model uncertainties. The latter controls a compliant behavior towards external interaction forces, or exerts a desired force on the environment. These paradigms contain implicit assumptions. Trajectory tracking assumes that all external forces and torques acting on the robot are disturbances, and therefore have to be rejected – any deviation from the reference trajectory is undesirable. This is true if the robot is operating outdoors in free space, under moderate wind influence, and is carrying a previously unknown payload. Conversely, interaction control schemes assume that the only external forces acting on the robot come from physical interaction. The robot will therefore be compliant, i.e. deviate from the reference trajectory, even to wind forces, which is undesirable in most scenarios.

Despite these advances in sensing and computation, some obstacles still remain elusive due to unfavorable sensing conditions. Examples are reflective and transparent surfaces (glass), and thin structures that cannot be sensed due to limited resolution of the sensors. Furthermore, computational power limits the sensing and planning frequency. This can cause a delay in reaction to dynamic obstacles, or lack of perception thereof. Failure of the obstacle avoidance system may then lead to contact or collision of the flying robot with its environment, causing damage to the robot, property, or people. In these cases, it is necessary to have a low-level collision detection and reflex strategy to minimize harm to the robot and its environment. Therefore, the four essential phases of collision Fault Detection, Identification, Isolation and Recovery (FDIR) have to be handled, see Haddadin et al. [54, 56]. *Collision detection* provides binary information whether a collision with the environment has occurred. *Collision classification* provides information about the collision type. The location of the collision is obtained through *collision isolation.* Lastly, appropriate *collision recovery* mitigates danger.

1.1 Problem statement

This thesis aims to make aerial robots aware of the disturbance, interaction, and faults acting on them, see Figure 1.2, which yields the following objectives. The first objective is to make flying robots *aware* of contacts and collisions, by providing a low-level detection and reaction framework. This requires the reasoning about the external wrench (force and torque) acting on the robot, and *discriminating* (distinguishing) between wind, interactions, and collisions. The second goal of this thesis is to provide an estimate of the wind

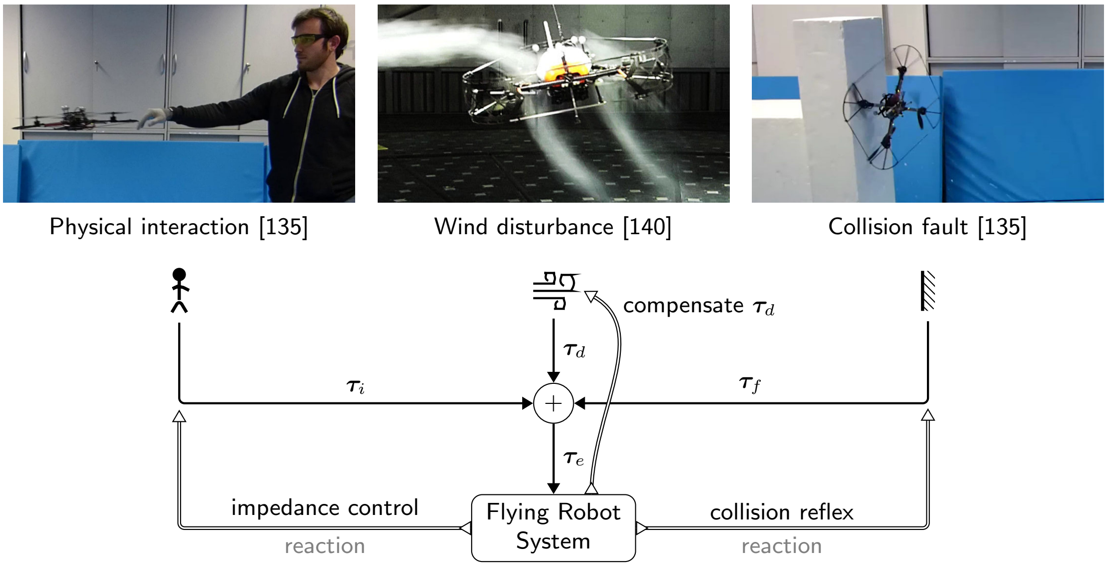

Physical interaction [135]
Wind disturbance [140]
Collision fault [135]

Figure 1.2: An interaction, disturbance, and fault aware flying robot system understands the source and context of its simultaneous inputs and reacts accordingly.

velocity that is independent of the external wrench. This allows a motion planner to reason about future external forces based on the aerodynamics model, and obtain energy-efficient and saturation-aware trajectories. The final goal of this thesis is to bridge the gap between trajectory tracking and interaction control, to allow *physical interaction under wind influence*, which is made possible by the discrimination of these forces.

These goals pose following research questions. First, is discrimination between the external wrench components for control purposes possible in real time? Second, given the individual wrench components, what are effective control schemes for interaction and trajectory tracking control under wind influence? Third, how can unexpected faults, such as collisions with the environment, be detected and handled efficiently and effectively? In the interest of the first question, a fourth can be posed: is it possible to obtain a measurement of the wind speed that is independent of the external wrench? In this thesis, model-based methods are developed and applied in the pursuit of answers to these questions.

1.2 Related Work

The following literature overview touches the most relevant topics covered in this thesis: trajectory tracking control for disturbance compensation, physical interaction control, collision fault detection and identification, wind estimation, and force discrimination.

Trajectory tracking control for disturbance compensation

Trajectory tracking. A large body of literature has been written about trajectory tracking control of aerial vehicles, see Table 1.1. The goal of trajectory tracking control is to make the vehicle track a time-dependent spatial trajectory. For multirotor flying robots, the trajectory is most commonly defined by the Cartesian position $\boldsymbol{r}_d(t)$ and yaw angle $\psi_d(t)$, and their time derivatives. It has been shown that quadrotors without aerodynamic effects are differentially flat, see Mellinger et al. [88], which means that all feedforward control signals may be obtained from the spatial trajectory alone. Earlier works used cascaded proportional-integral-derivative (PID) [60], linear quadratic regulator (LQR) [28], and feedback linearization controllers [147] to provide adequate trajectory following

Table 1.1: An overview of work related to trajectory tracking and disturbance compensation.

Topic	**References**
Trajectory tracking	Achtelik et al. [3], Cowling et al. [28], Faessler et al. [38], Goodarzi et al. [49, 48], Hoffmann et al. [60], Kamel et al. [70, 71], Lee et al. [77], Mellinger et al. [88], Omari et al. [107], Svacha et al. [131], Voos [146]
Disturbance compensation	Besnard et al. [20], Madani and Benallegue [82], [81], [21], [115], [7, 8], [35, 90], [153, 39], Leonard et al. [79], Gong et al. [47], Ruggiero et al. [118], Mellinger et al. [90], Hancer et al. [57], Park et al. [67], Kondak et al. [74], [75, 66, 65]

performance about hover conditions. Outside hover conditions, trajectory tracking is based on geometric tracking control on SE(3), see Lee et al. [77] and Goodarzi et al. [49], direct inversion based position control, see Achtelik et al. [3], as well as nonlinear model predictive control in Kamel et al. [70, 71] and unified model predictive trajectory generation and control in Neunert et al. [99]. An aerodynamics model may be used to improve trajectory tracking, as done e.g. by Omari et al. [107], Svacha et al. [131] and Faessler2018 et al. [**.**] The dominant attitude representations in literature are rotation matrices and quaternions [51, 50] due to being singularity free.

Disturbance compensation. During operation, aerial robots may be subject to the disturbances listed in Table 1.2. They are discussed in more detail in Section 6.2.3. The most obvious disturbances are due to aerodynamic effects, i.e.wind. Modeling and parameter uncertainties can also be viewed as disturbances. The system parameters, such as mass, inertia, or center of gravity, may change if a payload is attached during flight. In the case of partial propeller failure, its thrust coefficient changes. All these effects must be compensated by the trajectory tracking controller. Some of the techniques used in literature for dealing with disturbances and uncertainties are backstepping [81], sliding mode [21] control, H_∞ control [115], model predictive control [7, 8], and adaptive control [35, 90]. Explicit estimation of the uncertainties has been shown to improve flight performance. Sliding mode disturbance observers have been applied to quadrotors by Besnard et al. [20] and Madani and Benallegue [82]. However, higher-order sliding modes are computationally expensive for embedded systems due to noninteger powers. The disturbance estimate must be filtered to avoid chattering, thereby sacrifing robustness. Adaptive integral backstepping control (AIBC), which estimates and compensates model uncertainties, has also been applied to quadrotors [153, 39]. The controller has been shown to be effective for varying system inertia. Due to the large number of parameters it is not easy to tune. In Escareño et al. [36] a two-dimensional horizontal wind disturbance is estimated and compensated. Extended state observers have been applied for helicopter disturbance observation in Leonard et al. [79] and Gong et al. [47], in the context of active disturbance rejection control (ADRC). Momentum-based observers of the external wrench have been applied by Ruggiero et al. [118] to compensate disturbances. Kalman-filter-based disturbance observation has been shown for quadrotors in Mellinger et al. [90] for near-hover conditions.

Disturbance observer based control has been shown to improve tracking performance of various systems by explicitly estimating and compensating general uncertainties in nonlinear systems [80]. For example, acceleration-based observers have been applied to tilt-wing quadrotors in Hancer et al. [57] and underwater vehicles in Park et al. [67]. In Kondak et al. [74], the authors showed performance improvements when using a disturbance observer for quadrotor attitude control. In this paper, no significant improvements in

Table 1.2: A list of the most common disturbances acting on flying robots.

Type	**Description**	**Sensing**
Aerodynamic wrench	Aerodynamic drag wrench due to relative airspeed of the flying robot, caused by form drag and blade flapping of the propellers	Model [140, 149], Pitot tubes [152], Anemometer [26]
Propeller thrust	Change of propeller thrust (*i.e.* control input) with airspeed [14, 16]	Motor power [16], Force sensing [29]
Payload	Payload inertial parameters, aerodynamic drag wrench on the payload	Online identification [89]
Payload aerodynamic wrench	Aerodynamic drag wrench on the payload during cooperative transport	Online identification (*open problem*)

tracking performance have been observed when applied the observer to position control. Investigation of disturbance observers in quadrotors has also been done in [75, 66, 65] and more recent works. Acceleration-based disturbance observation is well-suited for small UAVs because acceleration measurements are provided by the Inertial Measurement Unit (IMU). Attitude control typically runs at high update rate (1 kHz in the case of an AscTec Hummingbird), so the angular acceleration signal required for the observer may be computed from the angular velocity by finite differences. Therefore, this approach is suitable in the context of the goals defined above.

Physical interaction control

Physical interaction control aims to prescribe a desired dynamics between the robot and an external interaction force. Widely applied in robot manipulator control, the two main approaches to achieve this are admittance control and impedance control [61, 108]. To implement these, the robot must have information about the interaction force. This can be measured (e.g. by a force-torque sensor (FTS)), or estimated. These topics have been thoroughly investigated in the context of robotic manipulators; see e.g. Haddadin et al. [56] and Ott et al. [108] and references therein. Our goal is to transfer the original concepts from manipulators and extend them to flying robots. Due to limited payload constraints of flying robots, in this work we estimate the external force. Note that this problem is closely related to disturbance observation, however we view the external force as a desired interaction, as opposed to an unwanted disturbance. Table 1.3 lists some state of the art developments in physical interaction control of flying robots, while Table 1.4 lists some example applications of flying robots physically interacting with their environment. So far, aerodynamic effects due to wind are largely ignored in the interaction control literature. As of today, this significantly limits the applicability of physical interaction methods in harsh environments.

External wrench estimation and impedance control. In the context of hybrid pose/wrench control, Bellens et al. [17] mapped offline control inputs to forces and torques generated by the UAV while being fixed to a base. The estimated wrench is then a function of the control input. This approach is only valid while the robot is in contact. The work of Ruggiero et al. [118, 119] is related to the wrench estimation and interaction control aspects of this thesis. The authors used a momentum-based external wrench estimator with second-order estimation dynamics and implemented an impedance controller based on this estimate. A sketch of the controller's stability proof was given. However, their

Table 1.3: An overview of work related to physical interaction control.

Topic	References
External wrench estimation	Bellens et al. [17], Ruggiero et al. [118, 119], Yüksel et al. [155], Yüksel et al. [154], McKinnon et al. [86], Augugliaro et al. [11]
Physical interaction control	Nguyen et al. [100], Jung et al. [69], Fumagalli et al. [42, 44], Ruggiero et al. [118, 119], Yüksel et al. [155], McKinnon et al. [86], Albers et al. [5], Forte et al. [40], Augugliaro et al. [11], Mersha et al. [92]

Table 1.4: Application examples of physical interaction between flying robots and environment.

Application	Related work
Individual robots	
Interaction control	Impedance control [92, 117, 118, 135, 138], Admittance control [11, 120, 135, 138], Force control [17, 110], Contact inspection [6]
Load transport	Slung load transport [19, 129], Aerial manipulators [4, 43]
Reconfiguration	Whole-body manipulation [156], Full controllability [24, 121, 120]
Groups of robots	
Manipulation	Transport [91, 93], Grasping [45], Aerial Manipulators [13, 95]
Assembly / Construction	Building structures [12]
Formation control	Energy tank based [41]
Physical docking	Distributed flight array [109]

force estimator requires translational velocity information, which is not readily available on flying systems, and is subject to failure cases of exteroceptive sensing and state estimation schemes. Yüksel et al. [155, 154] use a Lyapunov-based design of an nonlinear external wrench observer including also inertia shaping. A port-based Hamiltonian approach is used to design an Interconnection and Damping Assignment Passivity-Based Controller (IDA-PBC), which allows to reshape the physical properties of a quadrotor. Only simulation results are presented. McKinnon et al. [86] implemented an Unscented Kalman Filter (UKF) based estimator of the external wrench. This allows the explicit incorporation of sensor noise properties into the filter. However, the experimental results show overshoot of the wrench estimate, which we consider undesirable when using the estimate for feedback control.

Physical interaction control. A dedicated propeller for horizontal force control has been used by Albers et al. [5]. However, only a feedforward signal was used to exert a contact force. More recently, force control at a tooltip rigidly attached to a flying robot was investigated in Nguyen et al. [100], Jung et al. [69], and Fumagalli et al. [42]. The force was assumed to be measured by a force sensor. Impedance control was applied to UAVs for contact inspection by Forte et al. [40] and Fumagalli et al. [42, 44]. An external force measurement was not required because of reliance on the passivity properties of impedance control. Augugliaro and D'Andrea [11] extend the Kalman filter used for quadcopter state estimation with an external wrench state. They implement an admittance controller for interaction control, using the estimated external force. Mersha et al. [92] developed a variable impedance controller for aerial manipulation using a simplified model. A force controller is implemented as the outer loop of the variable impedance controller. It is assumed that the environment can be modeled as a linear spring. The external force is therefore estimated using position information and the environment stiffness, making it dependent on the used controller and state estimation method. The effectiveness of this approach is shown in flight experiments in contact scenarios, with ground truth provided by a force-torque sensor.

Collision detection and reaction

In order to protect a flying robot from unintended collisions with the environment due to e.g. sensing failures, these have to be detected and appropriate reaction has to be taken. Collision detection can be viewed as Fault Detection, Identification, Isolation and Recovery problem [32, 127], and has been extensively researched in the context of robotic manipulators in Haddadin et al. [54, 56]. *Collision detection* provides binary information whether a collision with the environment has occurred. *Collision classification* provides information about the collision type. The location of the collision is obtained through *collision isolation*. Lastly, appropriate *collision recovery* mitigates danger. Collision detection and reaction literature is briefly reviewed next. Table 1.5 show an overview thereof.

A control loop supervisor for contact detection was implemented by Naldi et al. [96]. If the error of the path following controller is above a threshold, contact is assumed and the path is interpolated backwards until there is no collision anymore. Here, no wrench information is used and absolute thresholds on the position error must be set. The position error acts as a proxy to the external force, so the threshold then depends on the controller gains. Furthermore, the duration required for the signal to be above the threshold is absolute. This makes the method trajectory and controller dependent. In Mersha et al. [92] contacts are detected through a threshold on the external force, making the implementation equivalent to Naldi et al. [96]. Onboard accelerometers and small force sensors attached to elastic springs were used by Briod et al. [25] to detect collisions with the environment. The acceleration-based approach detects collisions when the acceleration magnitude is above a pre-defined threshold. The threshold needs to be chosen larger than nominal accelerations occuring during normal flight. Therefore, the sensitivity of the method depends significantly on the controller and the trajectory. Upon collision, the motors are turned off and the robot relies on its robust mechanical design to land safely. It resumes flight opposite of the detected collision direction. This information is used for tactile exploration. More recently, Dicker et al. [31] used the accelerometer signal to detect collisions, and coupled this with a fuzzy logic system to obtain a reflex attitude command reaction for collision recovery.

Aerodynamic modeling and wind estimation

Table 1.6 shows that wind speed estimation for multirotor vehicles is a well established topic, and wind tunnel experiments are increasingly being carried out. Publications relating to applications of wind estimates show that there is a need for methods that provide this information. The aerodynamics of a flying robot (e.g. quadrotor) during fast flight and under external wind influence are significantly nonlinear.

Model-based wind speed estimation. A large body of literature shows that it is possible to obtain the freestream velocity by using an aerodynamics model and onboard measurements in flight. The methods fall broadly into two categories. The first set uses a physical modeling based approach. The exploited effects are usually blade flapping and propeller induced drag, which produce a horizontal force that can be measured by the onboard accelerometer. The second set of methods uses a data-driven approach, where a

Table 1.5: Overview of work related to collision detection and handling.

Topic	**References**
Collision handling	Briod et al. [25], Tomić and Haddadin [135, 136], Mersha et al. [92], Naldi et al. [96]

Table 1.6: An overview of related work in the field of wind estimation and force discrimination.

Topic	**References**
Model-based wind speed estimation	Waslander et al. [149], Huang et al. [64], Martin et al. [85], Neumann et al. [97], Omari et al. [107], Abeywardena et al. [2], Neumann et al. [98], Tomic et al. [136], Sikkel et al. [126], Ware et al. [148]
Wind tunnel experiments	Marino et al. [84], Schiano et al. [122], Neumann et al. [98], Planckaert et al. [113], Jung et al. [68], Prudden et al. [114], Bruschi et al. [26], Sikkel et al. [126], Tomić et al. [140]
Applications	Bangura et al. [15], Sydney et al. [133], Guerrero et al. [52], Bangura et al. [14], Ware et al. [148], Bangura et al. [16], Bennetts et al. [18]
Force discrimination	Manuelli et al. [83], Rajappa et al. [117], Tomić and Haddadin [136, 142],

regressor between a measured variable and the freestream velocity is found.

Physical modeling based estimation. The freestream velocity of multicopter MAVs is commonly obtained from accelerometer measurements. This can be done because the propeller induced drag and blade flapping produce a horizontal force that can be measured by the onboard accelerometer. However, only the horizontal velocity components can be obtained in this way. Huang et al. [64] used the dependence of thrust on freestream velocity to add a feedforward term to their position controller in order to improve tracking performance during aggressive maneuvers. Escareño et al. [37] estimated a slowly-varying 2D wind vector using a kinematics-based observer. The trajectory was modified on-line to compensate for the wind effects. This work is also related to disturbance compensation. Schiano et al. [122] used a wind tunnel to measure the forces and torques acting on a static quadrotor under varying conditions.

Another body of related work deals with using propeller induced drag to estimate the horizontal freestream velocity components. Waslander and Wang [149] use the propeller model from Huang et al. [64] and a linear drag model to estimate the metric wind velocity. This information is used for feedforward compensation of wind effects to improve position tracking accuracy. To reduce complexity for thrust calculation, a zero-order-hold is applied to the estimated freestream velocity, which results in rather limited estimation of the vertical wind component. This paper shows the feasibility of model-based wind estimation. The same effect was used by Martin et al. [85] and Abeywardena et al. [2] to estimate the relative airspeed of a quadcopter. Martin et al. [85] derive how the blade flapping effect can be used to estimate the relative airspeed and provide a simple compensation scheme of the resulting aerodynamic forces. Abeywardena et al. [2] incorporated a linear induced-drag model into a visual SLAM scheme. The filter state is augmented by the horizontal wind velocity, which is estimated from the accelerometer measurements. The approach improves convergence of accelerometer biases. However, the vertical wind component is not estimated. Omari et al. [107] incorporate blade flapping and induced drag as a feedforward term into a nonlinear control scheme. However, they assume that the wind velocity and yaw rates of the vehicle are negligible, and the thrust is independent of the freestream velocity. More recently, Sikkel et al. [126] identified parameters of the blade flapping model by flying in a wind tunnel and used it for wind speed estimation. It may be concluded that related work, especially Waslander et al. [149], shows the feasibility of model-based wind estimation.

Data-driven estimation. Another type of aerodynamics models uses generalized regression to fit experimental data to wind speeds. Neumann et al. [97, 98] and Ware et al. [148] related the quadrotor pitch angle to the wind speed. The assumption is that under static conditions the aerodynamic and control forces are in equilibrium, hence the aerodynamic force is estimated indirectly from the position controller output. As such, this method depends heavily on the controller and system parameters and is ill-suited to wind estimation during aggressive flight. It can be concluded that model-based wind speed estimation is well established in literature, albeit with significant limitations.

Sensor-based wind speed estimation. Alternatively, airspeed probes have been used by Sydney et al. [133] and Yeo et al. [151, 152] to measure the freestream velocity of a quadcopter. They used this measurement to create a probabilistic map of the wind field and improve controller performance. A successful evaluation of a small MEMS anemometers has also been performed by Bruschi et al. [26]. The anemometer was mounted 22 cm above the propellers of the quadrotor and tested in a wind tunnel, and performed well under different airflow conditions. However, force sensors are discretely localized, and a reliable, lightweight wind sensor for multirotor UAV does not yet exist.

Wind tunnel measurements. Wind tunnel measurements on multirotor vehicles have been increasingly carried out in recent years. Schiano et al. [122] and Planckaert et al. [113] measured the forces and torques acting on a static quadrotor under varying conditions for model identification purposes. Similarly, Jung et al. [68] performed comprehensive wind tunnel tests on commercial multicopter vehicles, measuring the vehicle drag and thrust under varying conditions. The purpose of the tests was to assess performance of multicopter systems. Marino et al. [84] measured the motor power in steady-state wind conditions, and related it to the wind velocity for estimation purposes. They found that the mapping of power to wind velocity is not unique, and the solution quality varies with the flow conditions. However, no online estimation scheme was proposed. Ware et al. [148] flew a quadcopter in a horizontal wind tunnel to identify the power used for flight under varying wind conditions and used it for path planning in an urban wind field. Bruschi et al. [26] evaluated the performance of a small anemometer mounted on a quadrotor. Prudden et al. [114] evaluated flow conditions around a quadcopter to find a feasible flow sensor mounting location where rotor influence is minimal. They found that such a sensor would have to be mounted at least 2.5 rotor radii in front of the hub axis to compensate for induced flow effects. Sikkel et al. [126] flew a quadcopter in a wind tunnel to estimate an aerodynamic model based on blade flapping, and used it to estimate the wind velocity.

Applications. The estimated wind speed is commonly used in literature to improve control performance or plan time- or energy-optimal paths through a wind field. Bangura et al. [15, 14] have shown that propeller power can be used to estimate and control its thrust. They used momentum theory [78] to estimate and control the propeller aerodynamic power, which is directly related to thrust. The estimated aerodynamic power to estimate the propeller thrust with known freestream velocity. Aerodynamic power control was applied to a quadrotor in Bangura et al. [16] in order to improve flight control performance. Sydney et al. [133] used the estimated wind speed and an aerodynamics model to improve flight performance. This was extended by an experimental validation in Sydney et al. [134]. Guerrero et al. [52] used a kinematic model to plan time-optimal quadrotor trajectories in known wind fields. Ware et al. [148] planned energy-optimal trajectories in a planar urban wind field, which was estimated using a fast CFD solver and a known map. Bennetts et al. [18] use the onboard estimated wind speed for probabilistic air flow modeling. Clearly, an online estimate of the wind speed can be used to improve overall performance of a flying robot.

Open problems. The methods available in literature are mainly based on first principles models and a relatively small dataset to fit parameters of those models. While this is sufficiently accurate to improve tracking performance and obtain information about wind speed, there is a lack of thorough exploration and analysis of aerodynamic models using large datasets. Second, data-driven methods map steady state values, like the pitch angle, to an airspeed. While this may work in a limited set of conditions, it is a proxy for the external wrench and depends on the controller parameters. This means that the model has to be reidentified when the controller parameters change. Furthermore, the generalization of aerodynamic models is not explored in literature, nor applications other than feedforward of the aerodynamic force and wind speed estimation. Lastly, due to complex flow conditions, there are no lightweight and reliable onboard airspeed sensors for aerial robots as of this writing.

Discrimination between aerodynamic and contact forces

This topic is crucial for the goals posed by this thesis, however it has obtained the least attention in literature, as can be seen from Table 1.7. Table 1.8 shows some sensors that could be used to for this purpose. However, the problem of distinguishing between aerodynamic, contact, and collision forces has not been formally defined in literature. Instead, solutions are predominantly application-specific. In physical robot-environment interaction scenarios, it is common to assume the use of a dedicated force sensor, see Nguyen et al. [100], Jung et al. [69], Fumagalli et al. [42], and Alexis et al. [6]. If the external force is estimated, it is assumed that aerodynamic forces are non-existent or negligible during the interaction, see Fumagalli et al. [44], Yüksel et al. [154]. Other authors have exploited different assumptions to discriminate between forces acting on the robot. For handling collision scenarios, Briod et al. [25] have used an Euler spring and force sensor to determine the collision force and direction. This adds additional sensors and weight, which is undesirable for a flying robot. Neither of these methods can distinguish between *slow* contact and aerodynamic forces from the *estimated* external wrench. More recently, Rajappa et al. [117] proposed a discrimination scheme that employs a sensor ring around a flying robot to separate human interaction force from additional disturbances. This scheme relies on adding localized sensors to the robot. Related to this problem, Manuelli et al. [83] developed a Contact Particle Filter to obtain contact positions on an Atlas humanoid robot using external wrench information. Discrimination between interaction, disturbance, and fault wrenches is possible in a localized fashion using sensors for specific applications, see Alexis et al. [6], Briod et al. [25], and Rajappa et al. [117]. The interaction input may further be discriminated from other inputs with the help of sensors as the ones listed in Table 1.8. However, force sensors are discretely localized, and a reliable, lightweight wind sensor for multirotor UAV does not yet exist. Approaching this from a different perspective, the known wind speed and an aerodynamics model may be used to obtain the aerodynamic wrench. This requires a wind sensing method that is independent of estimating the external wrench, such as onboard pitot tubes [152], anemometers [26], or motor power and an aerodynamics model [140]. Lastly, uncertainty of the control wrench due to the propeller force varying with airspeed may result in modeling errors, reducing the accuracy any of input discrimination. Recent advances have been made to estimate [16] or directly measure [29] the control input in flying robots.

Open problems. The problem as such has not been formally defined in literature yet,

Table 1.7: Work related to discriminating between aerodynamic and interaction forces.

Topic	**References**
Force discrimination	Manuelli et al. [83], Rajappa et al. [117], Tomić and Haddadin [136, 142],

Table 1.8: Most typical sensors and models that enable the discrimination of contact forces acting on flying robots.

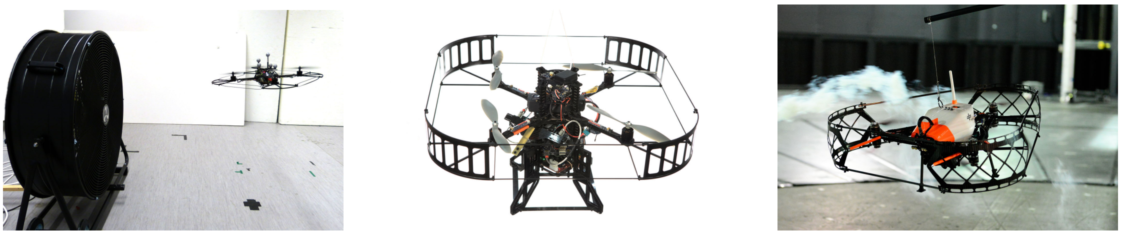
Sensor	**Measures**	**Description**
IMU	Acceleration, angular rate	Commonly used for navigation (strapdown). May be used for external wrench estimation [135].
Force / torque sensor	Force / Torque	Directly measures the wrench at certain locations. Provides direct measurement of the interaction wrench at the sensor's position [117]. Useful e.g. for contact inspection tasks [6], measuring payload properties [19], or forces generated by propellers [29].
Tactile sensor	Touch	Can be installed on the robot hull as input device for physical human-robot interaction.
Bumper	Touch	Detect collisions and contacts at specific hull points [25].
Surface humidity sensor	Humidity	Can detect and isolate environmental conditions that are potentially harmful for the robot, for example a water avoidance reflex in caves.
Strain gauge	Deformation	Measures bending of flexible structures. May be used to directly measure the propeller wrench, or detect structural failure.
Wind		
Anemometer	Wind speed	Measures wind speed directly. Difficult to apply to multirotor vehicles due to complex airflow.
Pitot tube	Dynamic pressure	Can be used to estimate the wind velocity. However, application to multirotor vehicles is limited by the complex airflow.
Motor current	Current	May be used to estimate the wind speed from the propeller aerodynamics model [14, 140]. Collisions with objects in propellers may be detected from the motor external torque.

and therefore not approached in a systematic manner. Isolated, application-specific solutions using dedicated sensors can be found in literature. Purely model based discrimination between slow aerodynamic and contact forces is therefore still an open research problem. This presents a good research opportunity, as such methods would allow novel applications of flying robots developed in the literature, without the need for additional sensing, which could potentially greatly enhance their usefulness.

1.3 Contributions

In the context of the goals of this thesis, the following contributions to the state of the art are made by means of our publications. They have been experimentally verified on the systems depicted in Figure 1.1 and Figure 1.3.

First, a systematic parameter identification procedure for aerial robots is developed [139]. The identification procedure is split into three stages: identifying the propulsion (motor and propeller) parameters first; followed by rigid body parameters; and finally identifying aerodynamic models. This makes it easier to isolate specific parameter sets as it reduces



(a) AscTec Hummingbird. Copyright © 2014 IEEE [141, 137, 138, 141].

(b) AscTec Pelican with modular stereo vision based navigation system. Copyright © 2013 IEEE [123, 138].

(c) DLR Ardea coaxial hexacopter custom developed and built for the wind tunnel experiments in this thesis. Copyright © 2016 IEEE [140, 139].

Figure 1.3: Different multirotor flying robot systems were used to develop and experimentally verify the techniques developed in this thesis.

the number of unknown parameters in each subsequent stage.
Second, external wrench estimation techniques are extended to be suitable for aerial robots without the need of velocity measurements [135, 138]. Accelerations and angular velocities are easy to measure by means of an IMU, however translational velocity is difficult to obtain on flying robots. It requires exteroceptive sensors like cameras and lasers, coupled with computationally expensive odometry and state estimation algorithms. Using only the IMU makes it possible to implement the wrench estimator on embedded platforms knowing only the control input, IMU measurements, and the system model.
Third, the wrench estimator is applied in a geometric trajectory tracking controller to compensate external disturbances [141]. It is shown that this greatly improves flight performance in wind without the need for integral or adaptive action.
Fourth, interaction control techniques (impedance and admittance control) are extended to flying robots, based on the external wrench estimate [135, 138]. An input-to-state stability proof is provided for the impedance controller, considering the wrench estimation dynamics. The impedance controller is further extended into a *compensated impedance controller*, that is compliant to the interaction wrench, but compensates the disturbance wrench. This assumes that the two wrenches can be discriminated online.
Fifth, wind speed estimation is investigated, using different data-driven aerodynamic models [140, 139]. To train and evaluate the models, measurements obtained by flying a custom-built hexacopter in a 3D wind tunnel were used, see Figure 1.3(c). The output of the models is the relative airspeed, i.e. the velocity of the robot w.r.t. the surrounding air. Different inputs are investigated, such as the external force, external wrench, and motor power. Conclusions are made about the model complexity required to accurately capture the aerodynamics model of a flying robot.

The *sixth* contribution is related to estimating wind speed using measured motor power [140, 139]. As this was also measured during the wind tunnel flights, it could then be used to estimate the aerodynamic power of each coaxial rotor pair during flight, and then the wind speed. Two novel methods were developed for this purpose. The first uses a data-driven approach to build nonlinear regression models from aerodynamic power to airspeed. The second is a first-principles model driven approach that builds an optimization problem based on the propeller aerodynamics model, and online motor power measurements. A sensitivity analysis of the optimization problem is provided and solutions are suggested to make it practically applicable. The sensitivity of this novel optimization problem also opens new avenues of research. This contribution essentially allows the propellers to be

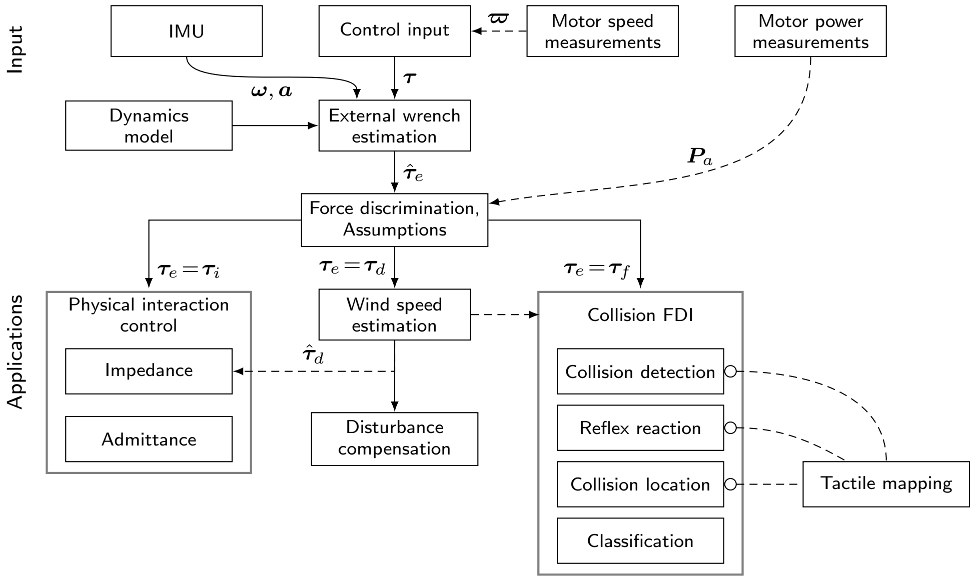


Figure 1.4: Applications of the external wrench estimate developed in this thesis. We consider the control input, IMU and motor power measurements as inputs into our estimation methods. External wrench information is used for disturbance compensation, interaction control, collision detection and wind speed estimation. Collision detection is further applied to tactile mapping. Wind speed estimation is improved by a propeller aerodynamics model and may be used to make collision detection more robust.

used as wind speed sensors, and provide a measurement that is independent of the IMU. *Seventh*, the topic of discriminating between aerodynamic, interaction, and fault wrenches is introduced [142, 139]. This topic has gained the least attention in literature so far. The methods developed in this thesis allow the external wrench to continuously, in real-time, be decomposed into the constituent terms. Collisions are detected based on the frequency content of the external force. Upon detection, a reflex reaction can be taken. The geometry of the problem can be used to obtain the collision location on the robot's convex hull. Experimental verification and applications of collision detection for flying robots are provided. Next, slow contact forces can be distinguished from wind through a residual of the aerodynamic torque model. Discrimination can then be performed by simultaneously estimating the wind speed, and slowing down wind estimation when contact is detected. Due of the failure cases of this method, an alternative method that uses aerodynamic power based wind estimation is developed. This relies on other aerodynamic models to distinguish between the constituent terms. In summary, this contribution presents a starting point of a quite new research field for aerial robots, as it also opens interesting new questions.
Eighth, a synopsis of recent developments in physically interacting flying robots is provided in Chapter 6 [142]. These are put in a larger taxonomical context, and a generalized concept of interacting flying robots at scale, such as in a swarm, is developed. An awareness pipeline is introduced that unifies the representation of interaction, disturbance, and fault awareness for a single flying robot as well as flying robot swarms. A synopsis of already developed individual elements of the pipeline from the literature is provided, and future research directions that would validate the proposed approach are speculated.

The overall contributions of the thesis are summarized in Figure 1.4. First, the external wrench τ_e acting on the robot is estimated. It is based on a dynamics model of the robot, acceleration a and angular velocity ω measurements from an inertial measurement unit (IMU) and speed measurements ϖ of all the propellers on the robot to obtain the

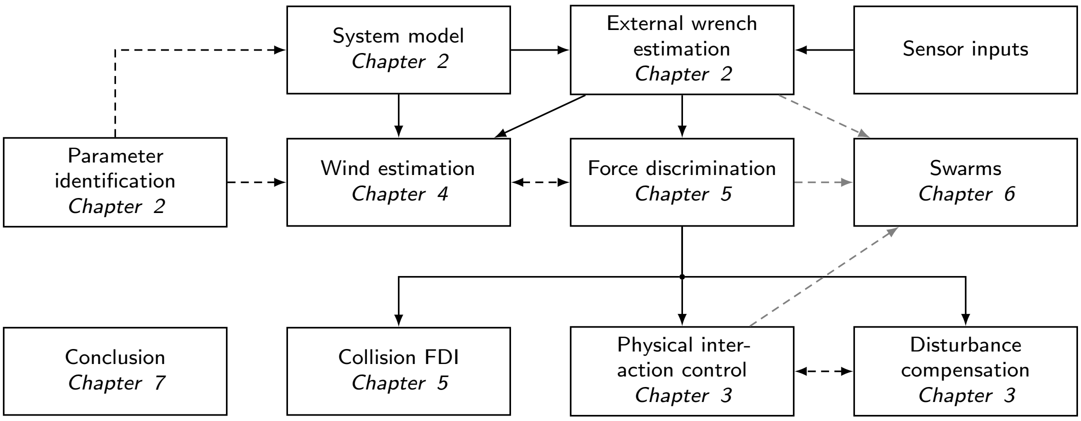


Figure 1.5: Overview of the objectives of this thesis.

applied control wrench $\boldsymbol{\tau}$. The estimated external wrench $\hat{\boldsymbol{\tau}}_e$ is then discriminated into the interaction wrench $\boldsymbol{\tau}_i$, the aerodynamic drag wrench $\boldsymbol{\tau}_d$, and the fault wrench $\boldsymbol{\tau}_f$. Force discimination and wind estimation may use motor power measurements and an aerodynamics model in order to use propellers as wind sensors. The following applications are first developed under the assumption that only one of the wrenches is acting at a time. Physical interaction control assumes that only the interaction control is acting on the robot. Wind speed estimation assumes that only the aerodynamic wrench is acting on the robot, and uses an aerodynamic model to estimate the wind speed. Lastly, collision fault detection and isolation (FDI) is able to detect and react to collisions under wind influence based on frequency characteristics of the wrenches. This allows for a tactile mapping application, that transparently supplements exteroceptive methods by adding obstacles into a map when collisions with the environment are detected.

The thesis is organized as depicted in Figure 1.5. Chapter 2 covers relevant modeling, parameter identification, and external wrench estimation. Chapter 3 covers trajectory tracking and physical interaction control. Wind tunnel experiments, evaluation of aerodynamic models, and wind estimation are covered in Chapter 4. Force discrimination techniques, including collision detection, are dealt with in Chapter 5. Applications and an outlook towards interaction and disturbance aware robot swarms is presented in Chapter 6. Finally, conclusions are made in Chapter 7.

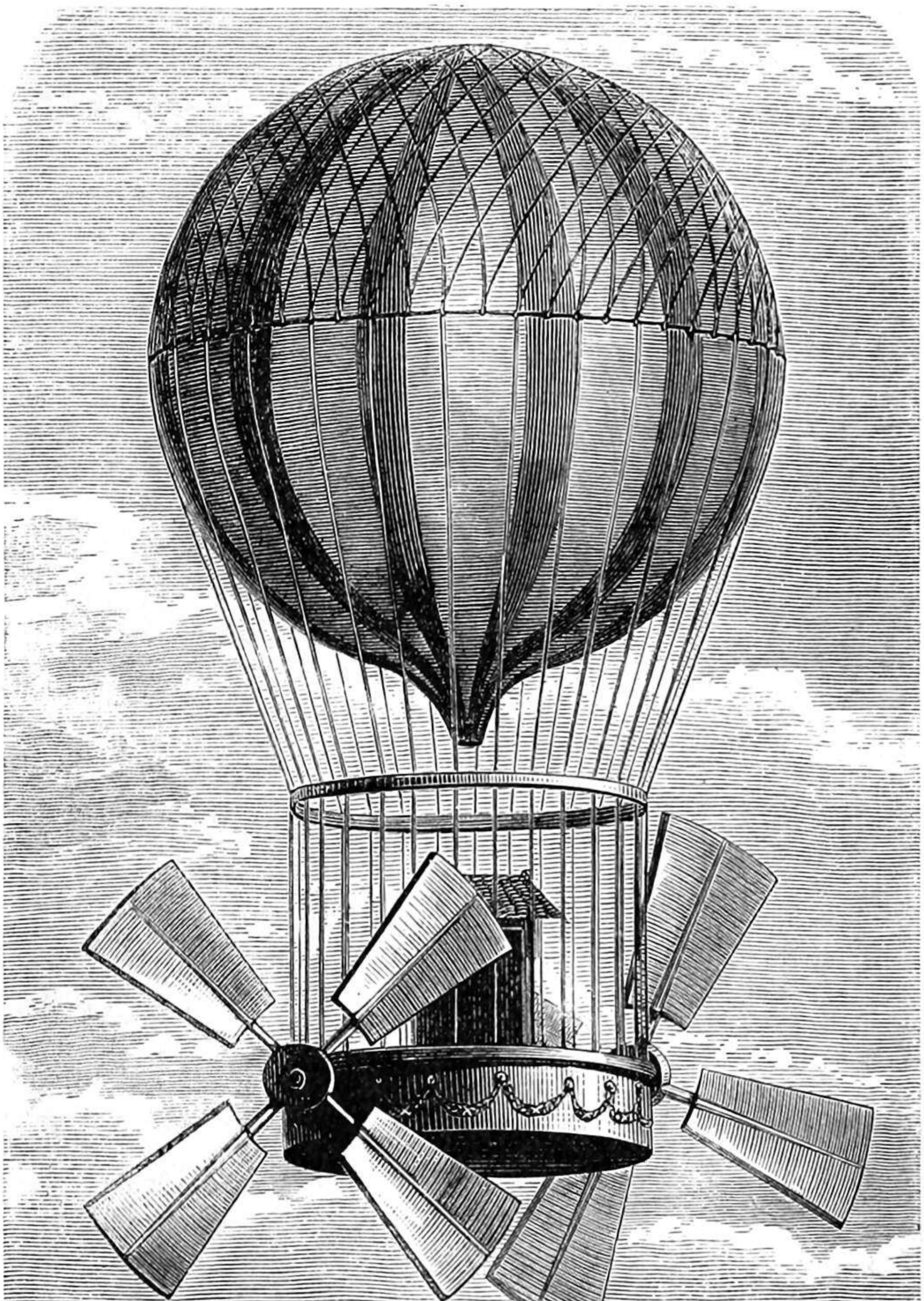


Illustration: L. Figuier: Les aérostats; Paris: Furne, Jouvet et Cie, 1887

CHAPTER 2

Modeling

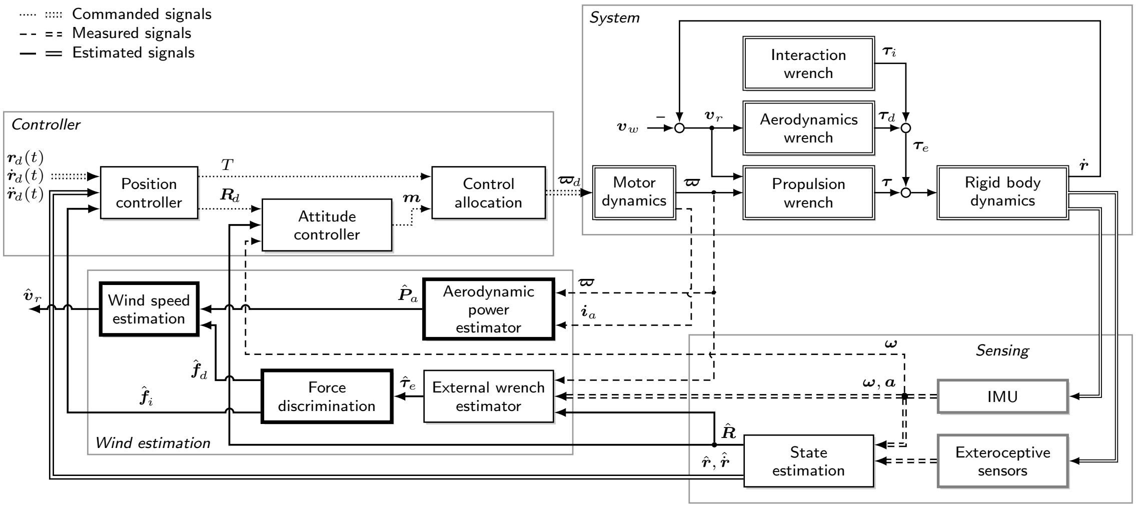


Figure 2.1: High-level overview of subsystems of a mutirotor flying robot.

Figure 2.1 depicts an overview of a mutirotor flying robot. On a high level, in the context of this thesis, it can be decomposed into the Controller, System, Sensing, and Wind estimation. The *Sensing* modules provide the controller with an estimate of the vehicle's state (position, orientation, velocity). We use this, and the motor feedback, in the *Wind estimation* blocks. The *aerodynamic power estimator* uses a propeller aerodynamics model and motor feedback to obtain the aerodynamic power $\hat{\boldsymbol{P}}_a$, as described in Section 2.1.5. *Force discrimination* uses the external wrench, the robot's convex hull and an aerodynamics module to simultaneously estimate the interaction force $\boldsymbol{f}_i$ and the aerodynamic force $\boldsymbol{f}_d$, as described in Chapter 5. The interaction force $\boldsymbol{f}_i$ can be used for interaction control, as presented in Section 3.2.2. The aerodynamic force $\boldsymbol{f}_d$, along with $\hat{\boldsymbol{P}}_a$, may then used to perform *wind velocity estimation*, as described in Chapter 4.

The controller, discussed in detail in Chapter 3, is typically a cascade of a position and an attitude controller, as depicted here. The *control allocation* module takes the desired control wrench and outputs the desired motor angular velocities. The *attitude controller* tracks the desired orientation $\boldsymbol{R}_d$ through a desired control torque. The *position controller* tracks the desired spatial trajectory $\boldsymbol{r}_d(t)$ and derivatives, through a desired attitude and thrust. The controller subsystem is discussed in detail in Chapter 3.

In this chapter, the dynamics of the flying robot, represented by the *System* subblock, are modeled. The *System* consists of a rigid body, with the aerodynamics wrench $\boldsymbol{\tau}_d$, interaction wrench $\boldsymbol{\tau}_i$ and propulsion wrench $\boldsymbol{\tau}$ acting on it through the external wrench $\boldsymbol{\tau}_e$. Rigid body dynamics is presented in Section 2.1.1, while the propulsion wrench is described in Section 2.1.2, along with relevant propeller aerodynamics in Section 2.1.3. The propulsion wrench and motor dynamics, modeled in Section 2.1.5, are connected through the propeller speed $\boldsymbol{\varpi}$. Lastly, the external wrench $\boldsymbol{\tau}_e$ is estimated as discussed in Section 2.3.

2.1 Dynamics model

2.1.1 Rigid body dynamics

Free-body diagrams of a quadrotor and a coaxial hexacopter is depicted in Figure 2.2. The equations of motion about the center of mass of such robots can be written as

$$\mathcal{M}\ddot{\boldsymbol{r}} = \mathcal{M}g\boldsymbol{e}_3 + \boldsymbol{R}\boldsymbol{f} + \boldsymbol{R}\boldsymbol{f}_e \tag{2.1}$$

$$\boldsymbol{\mathcal{I}}\dot{\boldsymbol{\omega}} = \mathcal{S}(\boldsymbol{\mathcal{I}}\boldsymbol{\omega})\,\boldsymbol{\omega} + \boldsymbol{m} + \boldsymbol{m}_e \tag{2.2}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\mathcal{S}(\boldsymbol{\omega}) \tag{2.3}$$

where $\mathcal{M}$ is the robot mass, $\boldsymbol{r} = [x, y, z]^T$ is its position in the fixed North-East-Down (NED) inertial frame, $\boldsymbol{R} \in SO(3)$ is the rotation matrix from the body to the inertial frame and equivalent to $\boldsymbol{R} \equiv \boldsymbol{R}_{ib}$, $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{3\times 3}$ is its moment of inertia, $\mathcal{S}(\cdot)$ is skew-symmetric matrix operator, g is the acceleration of gravity, $\boldsymbol{\omega}$ is the body angular velocity, $\boldsymbol{e}_3$ is the z-axis unit vector, $\boldsymbol{f}$ and $\boldsymbol{f}_e$ are the body-frame propulsion and external forces, and $\boldsymbol{m}$ and $\boldsymbol{m}_e$ are the control and external torques, respectively. The propulsion wrench is denoted as $\boldsymbol{\tau} = [\boldsymbol{f}^T\ \boldsymbol{m}^T]^T$, and the external wrench as $\boldsymbol{\tau}_e = [\boldsymbol{f}_e^T\ \boldsymbol{m}_e^T]^T$. In our case, the body frame is located at the center of propellers. To account for the offset center of mass, the control torque $\boldsymbol{m}$ contains the correction term $\boldsymbol{m}_g = \mathcal{M}g\mathcal{S}(\boldsymbol{r}_g)\,\boldsymbol{R}^T\boldsymbol{e}_3$, where $\boldsymbol{r}_g$ is the position of the center of gravity expressed in the body frame. By writing the generalized velocity as $\boldsymbol{\nu} = [\dot{\boldsymbol{r}}^T\ \boldsymbol{\omega}^T]^T$, the equations of motion can be rewritten in Lagrange form as

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g} = \boldsymbol{J}^T\boldsymbol{\tau} + \boldsymbol{\tau}_e, \tag{2.4}$$

where

$$\boldsymbol{g} = -\begin{bmatrix}\boldsymbol{I}_{3\times3}\\ \mathcal{S}(\boldsymbol{r}_g)\,\boldsymbol{R}^T\end{bmatrix}\mathcal{M}g\boldsymbol{e}_3, \qquad \boldsymbol{J} = \begin{bmatrix}\boldsymbol{R}^T & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3}\end{bmatrix},$$

$$\boldsymbol{M} = \begin{bmatrix}\mathcal{M}\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{\mathcal{I}}\end{bmatrix}, \qquad \boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix}\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & -\mathcal{S}(\boldsymbol{\mathcal{I}}\boldsymbol{\omega})\end{bmatrix}.$$

External wrench. For the purposes of this thesis, the external wrench may be defined as

$$\boldsymbol{\tau}_e = \boldsymbol{\tau}_i + \boldsymbol{\tau}_d(\boldsymbol{v}_r) + \boldsymbol{\tau}_f, \tag{2.5}$$

where $\boldsymbol{\tau}_i$ is the interaction wrench, $\boldsymbol{\tau}_d(\boldsymbol{v}_r)$ is the aerodynamic drag wrench that depends on the relative airspeed $\boldsymbol{v}_r = \dot{\boldsymbol{r}} - \boldsymbol{v}_w$, where $\boldsymbol{v}_w$ is the wind speed in the inertial frame. The fault wrench $\boldsymbol{\tau}_f$ consists of the collision wrench $\boldsymbol{\tau}_c$ and model faults $\boldsymbol{\tau}_{f,m}$, which stem from parameter uncertainties [32].

2.1.2 Propulsion wrench

Control of a flying robot's motion is achieved through the forces and torques generated by the propellers. These can be represented in the propulsion wrench $\boldsymbol{\tau}$, which also acts as the control input, As $\boldsymbol{\tau}$ can not be directly measured during flight, an accurate model is

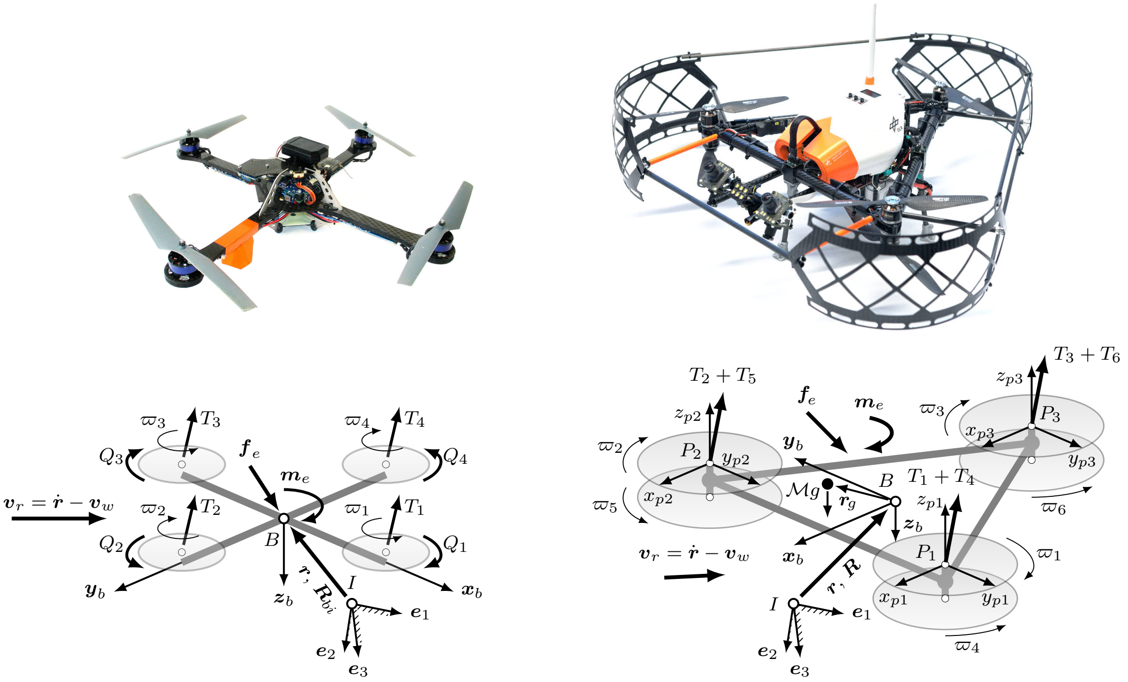


Figure 2.2: Free-body diagram of a quadrotor (left), and the coaxial hexacopter used in the wind tunnel experiments (right). The body frame B is located at position $\boldsymbol{r}$ and orientation $\boldsymbol{R}$ in the inertial frame I and subject to wind velocity $\boldsymbol{v}_w$. This causes the external wrench $\boldsymbol{\tau}_e = [\boldsymbol{f}_e^T \ \boldsymbol{m}_e^T]^T$ due to aerodynamic forces dependent on the airspeed $\boldsymbol{v}_r$. The propellers rotating at angular velocities $\boldsymbol{\varpi} = [\varpi_1 \ \ldots \ \varpi_6]^T$ generate the control wrench $\boldsymbol{\tau} = [\boldsymbol{f}^T \ \boldsymbol{m}^T]^T$ through the thrusts T_i and drag torques Q_i. The propeller frames $P_{1,2,3}$ are depicted in blue. Free-body diagrams Copyright © 2015, 2016 IEEE [136, 140]. Hexacopter photo Copyright © 2016 DLR (CC-BY 3.0).

required. The control wrench generated by the propellers about the center of mass for N propellers is given by

$$\boldsymbol{\tau} = \begin{bmatrix} \sum_{i=1}^{N} T_i \boldsymbol{n}_i \\ \sum_{i=1}^{N} \left(T_i(\boldsymbol{r}_i + \boldsymbol{r}_g) \times \boldsymbol{n}_i + \delta_i Q_i \boldsymbol{n}_i\right) \end{bmatrix} = \boldsymbol{B}\boldsymbol{u} \tag{2.6}$$

where $\boldsymbol{n}_i = \boldsymbol{R}_{bp,i}\boldsymbol{e}_3$ is the axis of rotation of propeller i located at $\boldsymbol{r}_i$ in the body frame, $\boldsymbol{R}_{pb,i}$ the rotation matrix from the body to the propeller frame, $\delta_i \in \{-1,1\}$ is the propeller rotational sense, and $\boldsymbol{B} \in \mathbb{R}^{6\times N}$ is the control allocation matrix. In classical designs where propellers are coplanar, $\boldsymbol{R}_{bp,i}$ is an identity matrix. In more recent multirotor designs, propellers are tilted to obtain more yaw control authority. In those cases, $\boldsymbol{R}_{bp,i}$ will be the appropriate transformation. The desired propeller velocities $\boldsymbol{u} = [\varpi_1^2 \ldots \varpi_N^2]^T$ can be obtained for control purposes by (pseudo-)inverting the matrix $\boldsymbol{B}$. The rotor thrust and torque in hover may be obtained by

$$T_{h,i} = \rho C_T D^4 \varpi_i^2, \tag{2.7}$$

$$Q_{h,i} = \rho C_Q D^5 \varpi_i^2 + I_r \dot{\varpi}_i, \tag{2.8}$$

where C_T and C_Q are the nondimensional rotor thrust and torque coefficients, respectively, and are typically obtained from static thrust measurements; D is the propeller diameter, and ϖ is the propeller speed. Additionally, ρ is the air density, D is the propeller diameter,

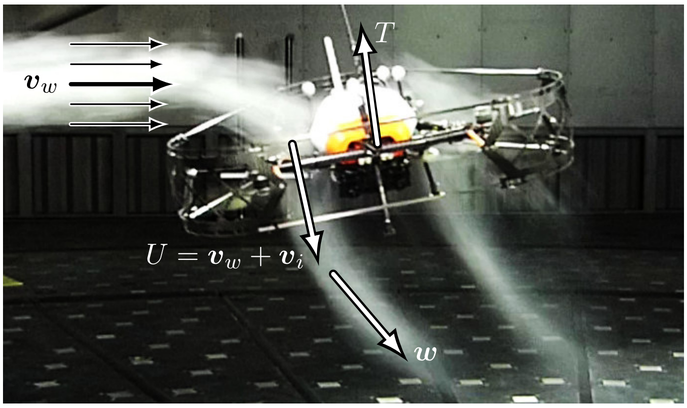


Figure 2.3: Thrust T is generated by increasing the wind velocity $\boldsymbol{v}_w$ by the propeller induced velocity v_i, which goes through the propeller normal. The propeller slipstream finally merges into the wind flow to produce $\boldsymbol{w}$. Copyright © 2016 IEEE [140].

and I_r is the combined inertia of the rotor and propeller. Next, we show how the thrust and torque change under wind influence. Of these values, D is known from the geometry, and ϖ is measured online. Note that the thrust and torque change with airspeed, so (2.7) and (2.8) are only valid in hover conditions. The air density $\rho = p/(R_{\text{air}}T)$ depends on the air pressure p, absolute temperature T and specific gas constant of air $R_{\text{air}} = 287.05$ J/kg K, and may be estimated online or computed from onboard barometer measurements. This leaves C_T, C_Q, and $\mathcal{I}_r$ to be identified. The coefficients C_T and C_Q are commonly obtained from bench tests relating propeller angular speed to measured force and torque. The rotor inertia $\mathcal{I}_r$ may be obtained by inertia estimation methods like swing tests, or dynamic identification on a motor. In Section 2.2, we identify these parameters on a force-torque sensor in the full hexacopter configuration.

2.1.3 Propeller aerodynamics

The forces exerted by a propeller depend on its freestream velocity (relative wind velocity). The freestream velocity of the k-th propeller expressed in the propeller frame is

$$\boldsymbol{v}_{\infty,k} = \boldsymbol{R}_{bp,k}^T \left(\boldsymbol{R}^T \boldsymbol{v}_r + \boldsymbol{\omega} \times \boldsymbol{r}_k\right), \tag{2.9}$$

where $\boldsymbol{v}_r = \dot{\boldsymbol{r}} - \boldsymbol{v}_w$ is the true airspeed, $\boldsymbol{v}_w$ is the wind velocity, and $\boldsymbol{r}_k$ is the location of the propeller relative to the center of gravity. The thrust acts in positive z-direction of the propeller frame P_k, see Figure 2.2. According to momentum theory [78] it can be written as

$$T = 2\rho A v_i U, \tag{2.10}$$

where A is the rotor disk surface area, and $U = \|v_i \boldsymbol{e}_3 + \boldsymbol{v}_\infty\|$ is the velocity of the propeller slipstream. The induced velocity v_i can be obtained using

$$v_i = \frac{v_h^2}{\sqrt{v_{xy}^2 + (v_i - v_z)^2}}, \tag{2.11}$$

which may be solved by several Newton-Raphson iterations with known v_h and v_∞ [78]. A flow visualization of thrust generation and the relevant velocities is depicted in Figure 2.3. The horizontal and vertical components of the freestream velocity are $\boldsymbol{v}_{xy} = \boldsymbol{v}_\infty - \boldsymbol{v}_z$ and $\boldsymbol{v}_z = \boldsymbol{e}_3^T \boldsymbol{v}_\infty$, respectively. Their norms are $v_{xy} = \|\boldsymbol{v}_{xy}\|$ and $v_z = \|\boldsymbol{v}_z\|$. In hover conditions the induced velocity is $v_h = \sqrt{T_h/2\rho A}$, where the hover thrust is obtained from (2.7). The

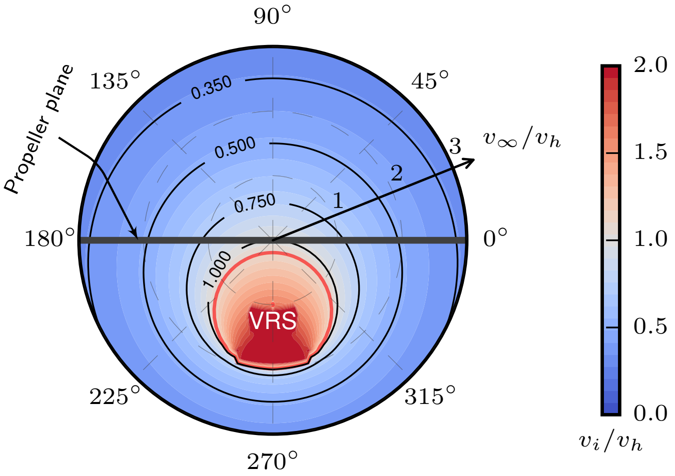


Figure 2.4: Relative induced velocity v_i/v_h in forward flight, depending on angle of attack and relative airspeed v_∞/v_h, depicted on the radial axis. Unmodified momentum theory is invalid in area delineated in red, where the propeller is in the Vortex Ring State (VRS) [78]. The contour lines depict different values of v_i/v_h. Copyright © 2016 IEEE [140].

propeller ideal aerodynamic power is

$$P_a = 2\rho A v_i U (v_i - v_z). \tag{2.12}$$

Furthermore, the aerodynamic power in forward flight is related to the hovering power following

$$\frac{P_a}{P_h} = \frac{v_i - v_z}{v_h}, \tag{2.13}$$

with $P_h = 2\rho A v_h^3$. Nonidealities can be included through the figure of merit (FM), between 0 and 1, such that $P_a = P_m \cdot FM$, where P_m is the motor power. The theory must be applied in the valid domain. Unmodified momentum theory does not apply in the unsteady Vortex Ring State (VRS) [78], as depicted in Figure 2.4.

Note that we may also model the aerodynamic power P_a as a general function of P_m, $P_a := g(P_m)$ which can be found in a parameter identification step. This will also include effects neglected by first-principles physics modeling, such as motor losses or fluid-structure interaction effects.

Coaxial rotors. In the case of coaxial rotors, we can consider one propeller pair as a single propeller. In that case, the hover induced velocity is

$$v_h = \sqrt{\frac{T_1 + T_2}{2\rho A}} = D\sqrt{\frac{2}{\pi}}\sqrt{C_{T,1}\varpi_1^2 + C_{T,2}\varpi_2^2}, \tag{2.14}$$

where $C_{T,1}$ and $C_{T,2}$ are thrust coefficients of the upper and lower propeller, respectively. They are obtained using an identification procedure of the propulsion system, as described in Section 2.2. Having obtained v_h, we use other aerodynamic quantities as described above. Note that the torque coefficients (and thereby power) will also differ between the upper and lower propeller.

2.1.4 Simplified drag model

Next, a widely used form of the aerodynamic wrench model is derived. Without loss of generality, the aerodynamic drag wrench may be written as

$$\boldsymbol{\tau}_d(\boldsymbol{\nu}_r) = -\boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r, \tag{2.15}$$

where $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_w$, with $\boldsymbol{\nu}_w$ being the generalized wind velocity. Based on physical models, it is composed of viscous (linear) drag and parasitic (quadratic) drag. As such, the model may be written as

$$\boldsymbol{D}(\boldsymbol{\nu}_r) = \boldsymbol{D}_L + \boldsymbol{D}_{i.d.} + \boldsymbol{D}_Q|\boldsymbol{\nu}_r| \tag{2.16}$$

with viscous damping coefficients $\boldsymbol{D}_L$, induced drag coefficients $\boldsymbol{D}_{i.d.}$ presented below in (2.19), and quadratic damping coefficients $\boldsymbol{D}_Q$. Viscous drag is principally the result of propeller induced drag and blade flapping, which are dominant at low velocities [106]. The coefficients in $\boldsymbol{D}_Q$ represent parasitic form drag, the components of which can be written as

$$d_{q,ij} = \tfrac{1}{2}\rho A_{\text{ref}} C_{D,ij},$$

where C_D is the drag coefficient, and A_{ref} is the reference surface area. A common assumption for small UAVs is that the drag forces are decoupled ($d_{q,ij} = 0 \; \forall \; i \neq j$) and act in a principal coordinate frame which usually coincides with the body frame. However, this is not a requirement for analyses performed in this thesis.

Following [107], the blade flapping and induced drag forces $\boldsymbol{f}_{i.d}$ acting on a propeller for small advance ratios — when the translational velocity is significantly smaller than blade tip speed — may be written as

$$\boldsymbol{f}_{i.d} = T\boldsymbol{A}_d\boldsymbol{v}_\infty, \tag{2.17}$$

where the blade flapping and induced drag matrix $\boldsymbol{A}_d$ is

$$\boldsymbol{A}_d = \begin{bmatrix} c_a + c_{d,x} & -c_b & 0 \\ c_b & c_a + c_{d,y} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.18}$$

where c_a and c_b are the longitudinal and lateral flapping coefficients, and $c_{d,x}$ and $c_{d,y}$ are induced drag coefficients. The induced drag wrench may now be defined as $\boldsymbol{\tau}_{i.d} = \boldsymbol{D}_i\boldsymbol{\nu}_r$, with

$$\boldsymbol{D}_{i.d.} = \begin{bmatrix} \sum_i T_i\boldsymbol{R}_{pb}^T\boldsymbol{A}_d & \boldsymbol{0}_{3\times3} \\ \sum_i \boldsymbol{r}_i \times \left(T_i\boldsymbol{R}_{pb}^T\boldsymbol{A}_d\right) & \boldsymbol{0}_{3\times3} \end{bmatrix}. \tag{2.19}$$

Notice that the location of the propellers may produce a drag torque. Notice that through propeller induced drag, the drag model is also dependent on the control input. This model is used for inversion-based wind estimation in Chapter 4. Other data-driven aerodynamic models based on general regressors are defined in Chapter 4.

2.1.5 Reduced brushless DC motor model

In order to estimate the propeller aerodynamic power, we employ the BLDC motor model from [14]. The mechanical part of motor dynamics can be represented by

$$\tau_m = (K_{q,0} - K_{q,1}i_a)i_a, \tag{2.20}$$

$$I_r\dot{\varpi} = \tau_m - D_r, \tag{2.21}$$

where i_a is the current through the motor, and ϖ is the rotor angular velocity. The motor torque is τ_m, with the torque constant modeled as $K_q(i_a) = (K_{q,0} - K_{q,1}i_a)$. The parameter I_r is the rotor inertia, and D_r is the aerodynamic drag torque acting on the rotor. The total motor mechanical power is $P_m = P_a/FM + P_r$, where the mechanical power P_m and power consumed by rotor acceleration P_r are used to estimate the aerodynamic power using

$$\begin{aligned} P_m &= \tau_m \varpi = (K_{q,0} - K_{q,1} i_a) i_a \varpi, && (2.22)\\ P_r &= I_r \varpi \dot{\varpi}, && (2.23)\\ \hat{P}_a &= FM\Big((K_{q,0} - K_{q,1} i_a) i_a - I_r \dot{\varpi} \varpi\Big). && (2.24) \end{aligned}$$

Note that, in general, the figure of merit FM can be a nonlinear function. In summary, we need to estimate or measure the motor current i_a, rotor speed ϖ and rotor acceleration $\dot{\varpi}$. The measurements i_a and ϖ can be obtained from modern ESCs, and $\dot{\varpi}$ can be estimated [14].

Current measurement on the speed controller allows us to directly relate the motor torque (2.20) to the aerodynamic torque of the propeller (2.8) through (2.21) as

$$\begin{aligned} \tau_m &= Q, \\ (K_{q0} - K_{q1} i_a) i_a &= I_r \dot{\varpi} + \rho C_Q D^5 \varpi_i^2, \end{aligned} \qquad (2.25)$$

i.e. the motor current can be related to the torque applied to the rotor, without the need of numerical differentiation to obtain $\dot{\varpi}$. By using this measurement to get the propulsion wrench (2.6), and using it in the external wrench estimator, see Section 2.3, an accurate estimate of the external yaw torque may be obtained even under wind influence. This relation is further investigated in Section 2.2.

2.2 Parameter identification

Problem formulation. The methods presented in this paper are model-based, and as such require model parameters to be identified. The rigid body and propulsion models can be represented as the linear regression model

$$\boldsymbol{Y}\boldsymbol{\theta} = \boldsymbol{u}, \qquad (2.26)$$

where $\boldsymbol{Y} \in \mathbb{R}^{N \times M}$ is the regression matrix, $\boldsymbol{\theta} \in \mathbb{R}^M$ is the vector of unknown parameters, and $\boldsymbol{u} \in \mathbb{R}^N$ is the known input. Here, N is the number of measurement samples, and M is the number of parameters. For the full hexacopter model, the following parameters have to identified: the mass $\mathcal{M}$ (1 parameter), inertia $\boldsymbol{\mathcal{I}}$ (6 parameters), center of gravity $\boldsymbol{r}_g$ (3 parameters), thrust and torque coefficients of the coaxial propellers (4 parameters), propeller inertia (1 parameter) and the motor torque constants (2 parameters), making in total 18 parameters to be identified. By using only diagonal inertia terms and known mass, the number of unknown parameters is reduced to 14. In order to further reduce the search space, the parameter estimation is performed in three stages, as depicted in Figure 2.5.

Identification methods. Two methods to obtain the estimated parameters $\hat{\boldsymbol{\theta}}$ are compared next. *Batch least squares.* First, the *batch least squares* [102] solution is obtained by

$$\hat{\boldsymbol{\theta}}_{\mathrm{LS}} = \big(\boldsymbol{Y}^T \boldsymbol{Y}\big)^{-1} \boldsymbol{Y}^T \boldsymbol{u}, \qquad (2.27)$$

which minimizes the ℓ_2 norm of the estimation error.

Iteratively reweighted least squares (IRLS). Second, the ℓ_1 norm of the model residuals is minimized by means of IRLS [27]. This provides robustness to outliers and creates a

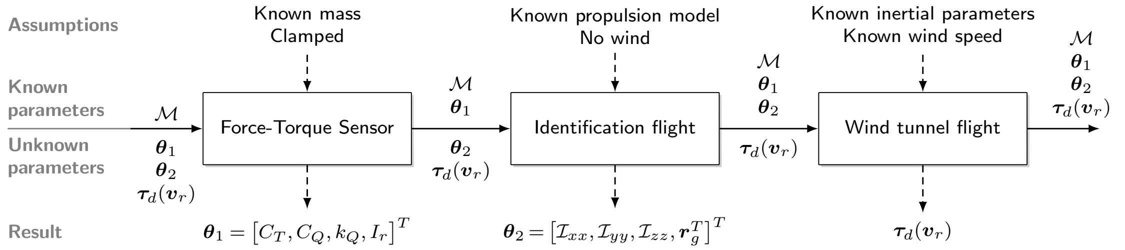

Assumptions
Known parameters
Unknown parameters
Result
Known mass
Clamped
Known propulsion model
No wind
Known inertial parameters
Known wind speed
Force-Torque Sensor
Identification flight
Wind tunnel flight
$\mathcal{M}$
$\boldsymbol{\theta}_1$
$\boldsymbol{\theta}_2$
$\boldsymbol{\tau}_d(\boldsymbol{v}_r)$
$\boldsymbol{\theta}_1 = [C_T, C_Q, k_Q, I_r]^T$
$\boldsymbol{\theta}_2 = [\mathcal{I}_{xx}, \mathcal{I}_{yy}, \mathcal{I}_{zz}, \boldsymbol{r}_g^T]^T$
$\boldsymbol{\tau}_d(\boldsymbol{v}_r)$

Figure 2.5: Parameter identification procedure. To minimize coupling effects in the high-dimensional parameter space, the identification is performed in three steps. Using the known vehicle mass $\mathcal{M}$, first the propulsion parameters $\boldsymbol{\theta}_1$ are estimated on a force-torque sensor. Inertia and the center of gravity $\boldsymbol{\theta}_2$ are identified from an identification flight without wind. Lastly, aerodynamic models $\boldsymbol{\tau}_d(\boldsymbol{v}_r)$ are identified from wind tunnel experiments.

sparse model by driving some parameters to zero. The estimate at step k is obtained by solving the weighted least squares problem

$$\hat{\boldsymbol{\theta}}_{\mathrm{IRLS}}^{(k)} = \left(\boldsymbol{Y}^T \boldsymbol{W}^{(k)} \boldsymbol{Y}\right)^{-1} \boldsymbol{Y}^T \boldsymbol{W}^{(k)} \boldsymbol{u} \tag{2.28}$$

where $\boldsymbol{W}^{(k)} = \operatorname{diag}\{w_1^{(k)}, w_2^{(k)}, \ldots, w_N^{(k)}\}$ is the weight matrix. Minimization of an ℓ_p-norm, $0 \leq p \leq 1$, is obtained by setting the weights to $w_i^{(k)} = \left(\boldsymbol{r}^{(k-1)2} + \varepsilon\right)^{(p/2)-1}$, where $\boldsymbol{r}^{(k)} = \boldsymbol{Y}\boldsymbol{\theta}^{(k)} - \boldsymbol{u}$ is the estimation residual, and $\varepsilon \ll 1$ is a regularization parameter obtained as described in [27].

Identification procedure. A difficulty with identifying all parameters from flight data is the lack of ground truth measurements of the total torque acting on the robot. It was also found that estimating the thrust and torque coefficients from flight data is sensitive to time delay in the measurements (on the order of 20 ms), and can lead to physically meaningless parameters, such as negative thrust coefficients. Identification of the system parameters is therefore split into three parts as depicted in Figure 2.5:

1. propeller and motor parameters are obtained using measurements on a force-torque sensor and the known mass,
2. rigid body parameters are obtained from an identification flight,
3. aerodynamic models are obtained by flying in a 3D wind tunnel.

The propulsion model is treated as ground truth for the rigid body identification. In the last step, the external wrench is estimated based on the previously identified models. In our experiments only the aerodynamic wrench acts on the robot. The estimated external wrench is therefore used to identify aerodynamic models. The procedure and results are covered in depth in Section 4.3.

Propulsion system parameters. The propulsion parameter vector for our coaxial hexacopter is

$$\boldsymbol{\theta}_1 := [C_{T,1},\, C_{T,2},\, C_{Q,1},\, C_{Q,2},\, \boldsymbol{K}_{q,1}^T,\, \boldsymbol{K}_{q,2}^T,\, I_r]^T, \tag{2.29}$$

and the regression matrix $\boldsymbol{Y}_1$ contains the rotor rates and motor current. The motor torque coefficients $\boldsymbol{K}_{q,i} = [K_{q,0i}, K_{q,1i}]^T$ are split for upper and lower motors ($\boldsymbol{K}_{q,1}$ and $\boldsymbol{K}_{q,2}$ respectively) because of the aerodynamic interaction between the propellers. For this step in the identification procedure, the hexacopter was fixed to an ATI 85 Mini force-torque sensor as depicted in Figure 2.6. The wrench measured by the sensor is concatenated in $\boldsymbol{u}_1 := \boldsymbol{\tau}_{\mathrm{FTS}}$. The pose of the hexacopter and the force-torque sensor were obtained by a

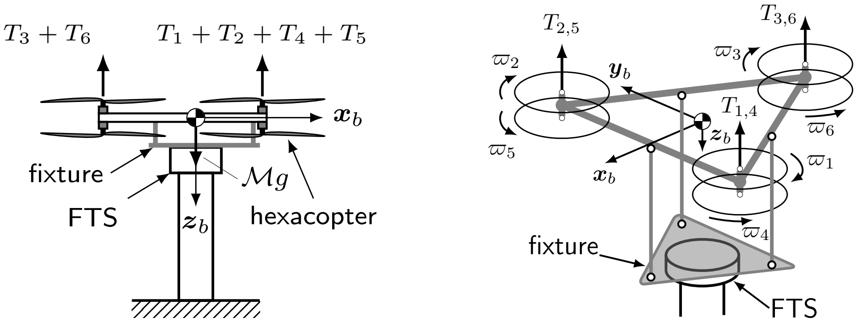


Figure 2.6: Setup of the force-torque sensor experiment. The hexacopter with coaxial propeller pairs was fixed to a force-torque-sensor (FTS). The lower and upper propellers have different thrust and torque coefficients due to interaction effects. Copyright © 2017 IEEE [138].

motion capture system at 250 Hz, while the onboard attitude controller ran at 500 Hz. The following signals were logged: pose, IMU, motor speed and current as measured by the speed controllers, the commanded control input, and the force and torque. The relative orientation of the force-torque sensor to the IMU was calibrated beforehand. The resulting parameter estimates are listed in Table 2.1. Note the different motor constants between upper and lower propellers. In comparison to the lower propellers, the upper propellers generate less thrust ($C_{T,1} < C_{T,2}$) and require more power ($C_{Q,1} > C_{Q,2}$).

Comparison of the identified model to the force-torque sensor measurements is shown in Figure 2.7. It can be seen that the identified propulsion model closely matches the force-torque sensor measurements. In this case, using the measured motor speeds to obtain the control wrench shows only a minor improvement over using the commanded speeds.

Figure 2.8 shows the yaw torque estimation using different measurements. The model most widely used in literature uses only the motor speed, and is shown as $\hat{m}_{e,z}^{\varpi}$. This simple model does not capture fast transitions well because the rotor acceleration torque is not modeled. Adding also the rotor acceleration ($\hat{m}_{e,z}^{\varpi,\dot{\varpi}}$) as in (2.8) improves accuracy during fast changes of the desired torque, but requires estimation of the rotor acceleration. Lastly, the motor torque may be obtained directly from the measured motor current as in (2.25), shown in Figure 2.8 as $\hat{m}_{e,z}^{i_a}$. Note that the motor torque is used to obtain the yaw component of the propulsion wrench (2.6). The measured current is also depicted for illustrative purposes. In this case, a propeller model is not needed, while the accuracy is similar to the model using rotor acceleration. Note that in the case of actuator failure (e.g.

Table 2.1: System parameters identified in the first identification step, using data from a force-torque sensor. Coaxial propeller pair coefficients are written as [upper, lower].

Parameter	Value
$\mathcal{M}$	2.445 kg
D	0.254 m
C_T	$[5.1137, 7.8176] \cdot 10^{-2}$
C_Q	$[7.5183, 4.7597] \cdot 10^{-3}$ -
$K_{q,0}$	$[2.9404, 1.4545] \cdot 10^{-2}$ N m/A
$K_{q,1}$	$[-1.4099, -3.3360] \cdot 10^{-3}$ N m/A^2
I_r	$2.1748 \cdot 10^{-4}$ kg m^2

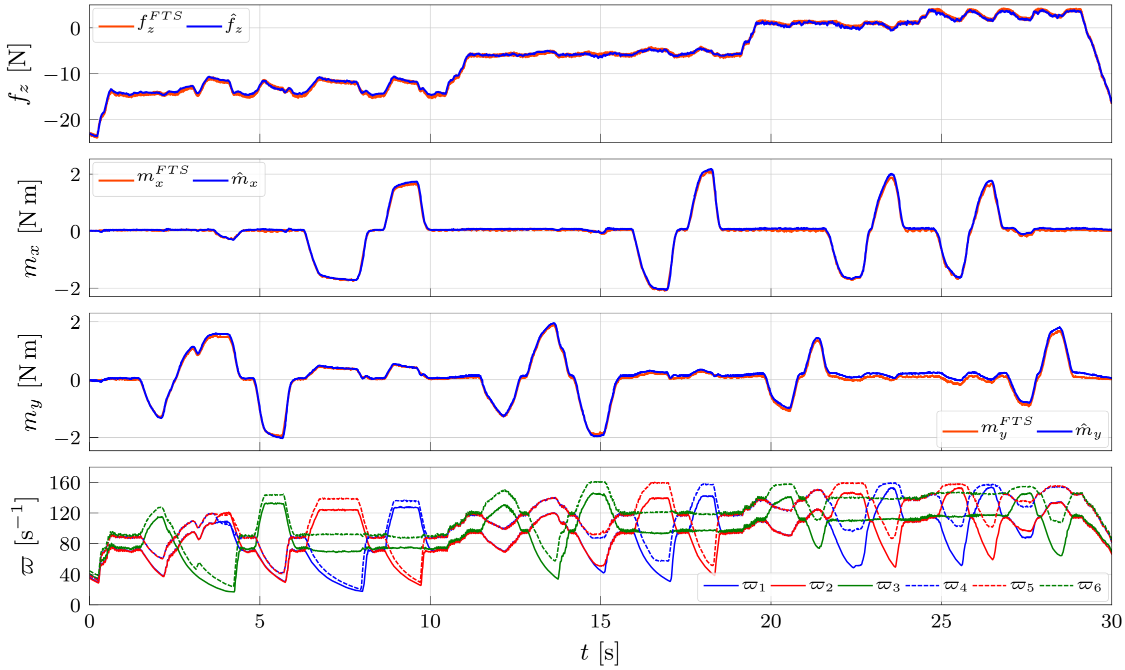


Figure 2.7: Validation of the identified propulsion model (—) on the setup depicted in Figure 2.6, compared to force-torque sensor measurements (—). For clarity only show the thrust f_z and torques m_x and m_y are shown. The propulsion forces and torques are obtained using the measured motor speeds, shown in the bottom plot. The measured motor speeds $\boldsymbol{\varpi}$ of the upper propellers are shown as solid lines (—, —, —), and lower propellers are shown as dashed lines (--, --, --). Copyright © 2017 IEEE [138].

partially losing a propeller), using the motor current will provide a better estimate of the yaw torque, as the method does not explicitly model the propeller drag torque.

Rigid body parameters. For a diagonal inertia tensor, the rigid body parameter vector is

$$\boldsymbol{\theta}_2 := [\mathcal{I}_{xx}, \mathcal{I}_{yy}, \mathcal{I}_{zz}, \boldsymbol{r}_g^T]^T. \tag{2.30}$$

On the right-hand side, $\boldsymbol{u}_2$ is obtained from the identified propulsion model and the known mass $\mathcal{M}$ as

$$\boldsymbol{u}_2 = \boldsymbol{Y}_1\boldsymbol{\theta}_1 - \mathcal{M}\boldsymbol{y}_{\mathcal{M}}, \tag{2.31}$$

where $\boldsymbol{y}_{\mathcal{M}}$ is the regression matrix column associated with the mass. Furthermore, the off-diagonal inertia terms are more than an order of magnitude lower than the diagonal terms, which allows us to simplify the model to diagonal inertia. The identified parameters are listed in Table 2.2. Figure 2.9 compares the propulsion model torque to the torque predicted by the identified rigid body model. The ℓ_1-identified parameters are shown, as the predicted torque is almost indistinguishable from ℓ_2. The result confirms correctness of the identified dynamics model.

Identification of aerodynamic models was done through wind tunnel experiments, which are described next.

2.3 External wrench estimation

Knowledge of the system model and control laws can be used to estimate the external wrench acting on the robot [30]. Two schemes are investigated – the *momentum-based*

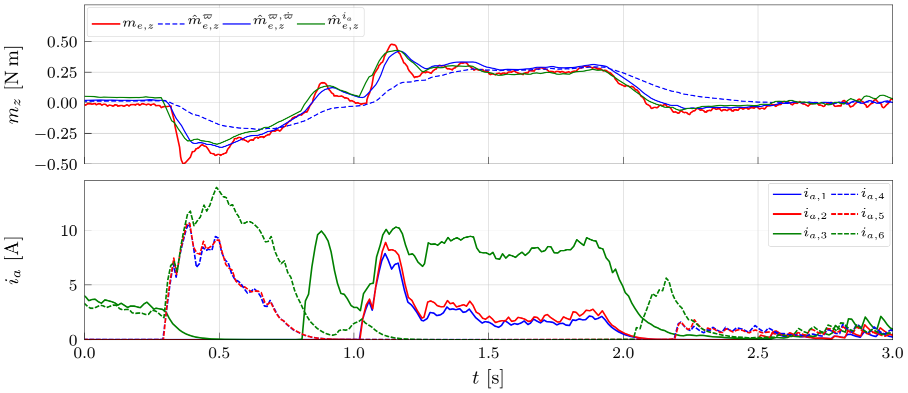


Figure 2.8: Estimation of the external yaw torque, using the measured motor speed only ($m^{\varpi}_{e,z}$), using the motor speed and rotor acceleration ($m^{\varpi,\dot{\varpi}}_{e,z}$) and using motor current ($m^{i_a}_{e,z}$). The estimator gain is $\boldsymbol{K}^m_I = 36$ to make the signals more discrenible.

Table 2.2: Results of the rigid body parameter identification step for the DLR Ardea hexacopter, using data from an identification flight, and the identified propulsion model. Results obtained by batch least squares (ℓ_2) and IRLS (ℓ_1) do not differ significantly.

ℓ_2	$\boldsymbol{\mathcal{I}}$	$\mathrm{diag}\{2.58, 2.46, 4.32\} \cdot 10^{-2}$ kg m^2
	$\boldsymbol{r}_g$	$[4.28, -1.11, -11.1]^T \cdot 10^{-3}$ m
ℓ_1	$\boldsymbol{\mathcal{I}}$	$\mathrm{diag}\{2.54, 2.58, 5.46\} \cdot 10^{-2}$ kg m^2
	$\boldsymbol{r}_g$	$[4.01, -1.05, 8.64]^T \cdot 10^{-3}$ m

method which uses velocity information, and the *acceleration-based method*. Finally, we combine the two to obtain a practical choice for a flying robot, using already available sensors.

2.3.1 Momentum-based estimation

The first method relies on observing the robot's generalized momentum $\boldsymbol{p} = \boldsymbol{M}\boldsymbol{\nu}$. Rewriting (2.4) in terms of $\boldsymbol{p}$ gives

$$\dot{\boldsymbol{p}} = \boldsymbol{M}\dot{\boldsymbol{\nu}} = \boldsymbol{J}^T\boldsymbol{\tau} + \boldsymbol{\tau}_e - \boldsymbol{N}, \tag{2.32}$$

where $\boldsymbol{N} := \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}$. Following [30], we define a residual vector

$$\boldsymbol{\rho} = \boldsymbol{K}_I \left[\boldsymbol{p} - \int \left(\boldsymbol{J}^T\boldsymbol{\tau} - \boldsymbol{N} + \boldsymbol{\rho}\right) \mathrm{d}t - \boldsymbol{p}(0)\right] \tag{2.33}$$

with positive definite diagonal observer matrix $\boldsymbol{K}_I \in \mathbb{R}^{6\times 6}$. By differentiating (2.33), we obtain the residual dynamics

$$\dot{\boldsymbol{\rho}} = \boldsymbol{K}_I\boldsymbol{\tau}_e - \boldsymbol{K}_I\boldsymbol{\rho}. \tag{2.34}$$

Note that (2.34) represents a linear exponentially stable system, driven by the true external wrench $\boldsymbol{\tau}_e$. Hence, $\boldsymbol{\rho}$ is the first-order lowpass filtered reconstruction of $\boldsymbol{\tau}_e$. Therefore, the estimated external wrench is denoted $\hat{\boldsymbol{\tau}}_e := \boldsymbol{\rho}$. Note that this method requires the measurement or estimation of the generalized velocity $\boldsymbol{\nu}$.

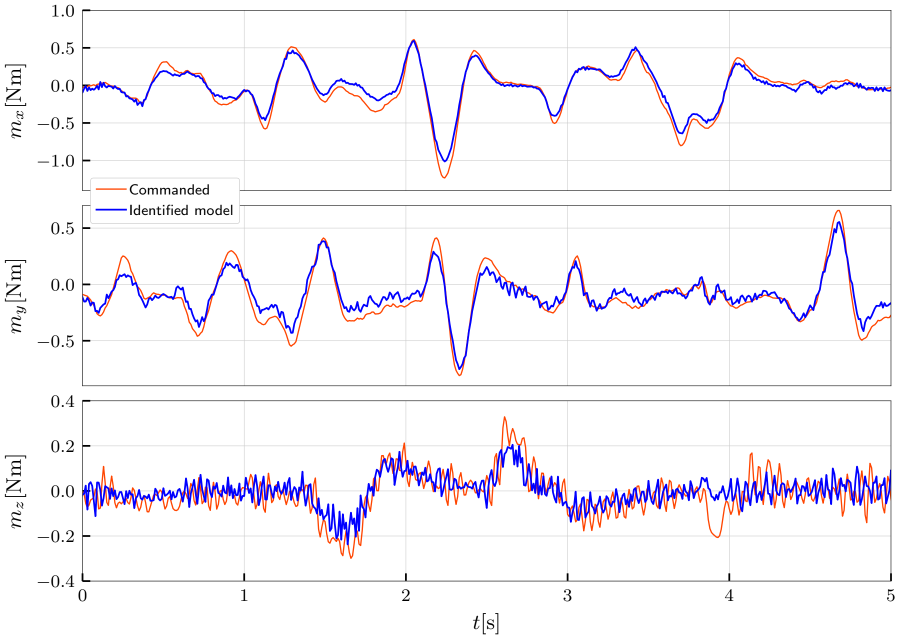


Figure 2.9: Rigid body torque prediction using parameters obtained by the IRLS method, minimizing the ℓ_1-norm. Overall the model shows a good match to the ground truth.

2.3.2 Acceleration-based estimation

In general, the external wrench could be obtained directly from full acceleration information. By rearranging the terms in (2.4) we may algebraically calculate the external wrench as

$$\bar{\boldsymbol{\tau}}_e = \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g} - \boldsymbol{J}^T\boldsymbol{\tau}, \tag{2.35}$$

where the acceleration $\dot{\boldsymbol{\nu}}$ contains significant sensor noise. A first-order stable filter is therefore applied to obtain the estimation dynamics

$$\dot{\hat{\boldsymbol{\tau}}}_e = \boldsymbol{K}_I\bar{\boldsymbol{\tau}}_e - \boldsymbol{K}_I\hat{\boldsymbol{\tau}}_e. \tag{2.36}$$

By lumping the nonlinear terms in $\boldsymbol{N} := \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}$, we finally obtain

$$\dot{\hat{\boldsymbol{\tau}}}_e = \boldsymbol{K}_I\left(\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{N} - \boldsymbol{J}^T\boldsymbol{\tau} - \hat{\boldsymbol{\tau}}_e\right). \tag{2.37}$$

In contrast to robot manipulators, this method is suitable for observing the translational dynamics of flying robots, as the translational acceleration $\ddot{\boldsymbol{r}}$ and acceleration of gravity $g\boldsymbol{e}_3$ are directly measured by the onboard inertial measurement unit (IMU). The angular acceleration $\dot{\boldsymbol{\omega}}$ may be obtained by numerical differentiation.

2.3.3 Hybrid estimation

Practical considerations for flying robots. Table 2.3 shows measurement requirements for the two wrench estimation methods. Obtaining a drift-free *translational* velocity requires exteroceptive sensors and a fusion algorithm. This greatly limits the applicability of the momentum based method for force estimation. Similarly, the *angular* acceleration can only be obtained through numerical differentiation. This reduces the quality of the torque estimate by the acceleration based method.

By considering the directly measurable values, it is intuitively clear that a combination of the two methods solves each one's shortcomings — (2.37) is used for external force

Table 2.3: Motivation for the hybrid estimation scheme — the translational velocity required by the momentum observer would have to be estimated. Alternatively, the readily available accelerometer signal can be used for estimation of the external force. The table shows relevant measurements provided by different state of the art sensor suites: translational velocity $\dot{\boldsymbol{r}}$, translational acceleration $\ddot{\boldsymbol{r}}$, angular velocity $\boldsymbol{\omega}$, and angular acceleration $\dot{\boldsymbol{\omega}}$. Measured: •, obtained numerically: ∘, or estimated: △.

Sensor	$\dot{\boldsymbol{r}}$	$\boldsymbol{\omega}$	$\ddot{\boldsymbol{r}}$	$\dot{\boldsymbol{\omega}}$
Accelerometer	-	-	•	-
Gyroscope	-	•	-	∘
PX4FLOW [62]	△	•	-	∘
Skybotix VI [101]	△	•	•	∘
IMU–odometry fusion [124, 123]	△	•	•	∘

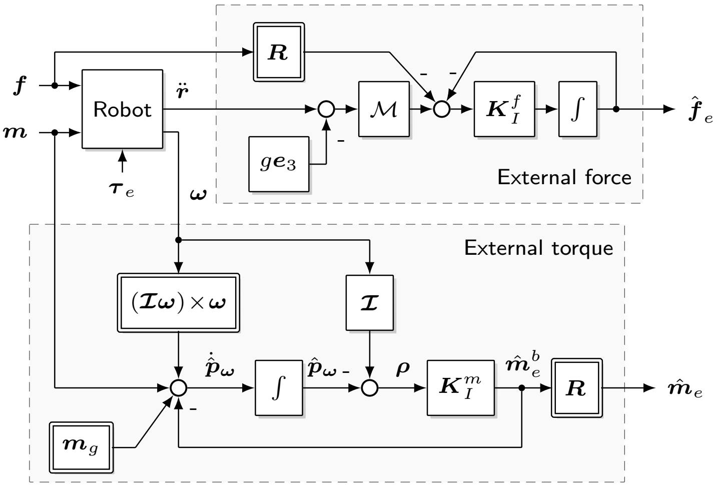


Figure 2.10: Structure of the hybrid external wrench estimator near hovering conditions. It uses the robot acceleration $\ddot{\boldsymbol{r}}$ and angular velocity $\boldsymbol{\omega}$, both measured directly by the onboard IMU. Copyright © 2017 IEEE [138].

estimation and (2.33) for external torque estimation. The resulting estimator structure is depicted in Figure 2.10. The rotational part of the momentum $\boldsymbol{p}$ is denoted $\boldsymbol{p}_\omega$. The estimator gain $\boldsymbol{K}_I$ is split into its diagonal blocks $\boldsymbol{K}_I^f$ for the force and $\boldsymbol{K}_I^m$ for the torque components. By combining both methods we can estimate the external wrench using proprioceptive sensors only. The estimated external wrench $\hat{\boldsymbol{\tau}}_e = [\hat{\boldsymbol{f}}_e \, \hat{\boldsymbol{m}}_e]^T$ is obtained by

$$\hat{\boldsymbol{\tau}}_e = \begin{bmatrix} \int_0^t \boldsymbol{K}_{I,f}\left(\mathcal{M}\boldsymbol{a} - \boldsymbol{f} - \hat{\boldsymbol{f}}_e\right)\mathrm{d}t \\ \boldsymbol{K}_{I,m}\left(\boldsymbol{\mathcal{I}}\boldsymbol{\omega} - \int_0^t \left(\boldsymbol{m} + (\boldsymbol{\mathcal{I}}\boldsymbol{\omega})\times\boldsymbol{\omega} - \hat{\boldsymbol{m}}_e\right)\mathrm{d}t\right) \end{bmatrix} \tag{2.38}$$

where $\boldsymbol{a} = \boldsymbol{R}^T(\ddot{\boldsymbol{r}} - g\boldsymbol{e}_3)$ is the acceleration measured by an accelerometer in the body frame, and $\hat{\boldsymbol{m}}_e$ is the estimated external torque, also expressed in the body frame. Note that an accelerometer also measures the acceleration of gravity. The estimator dynamics are linear and decoupled in both methods.

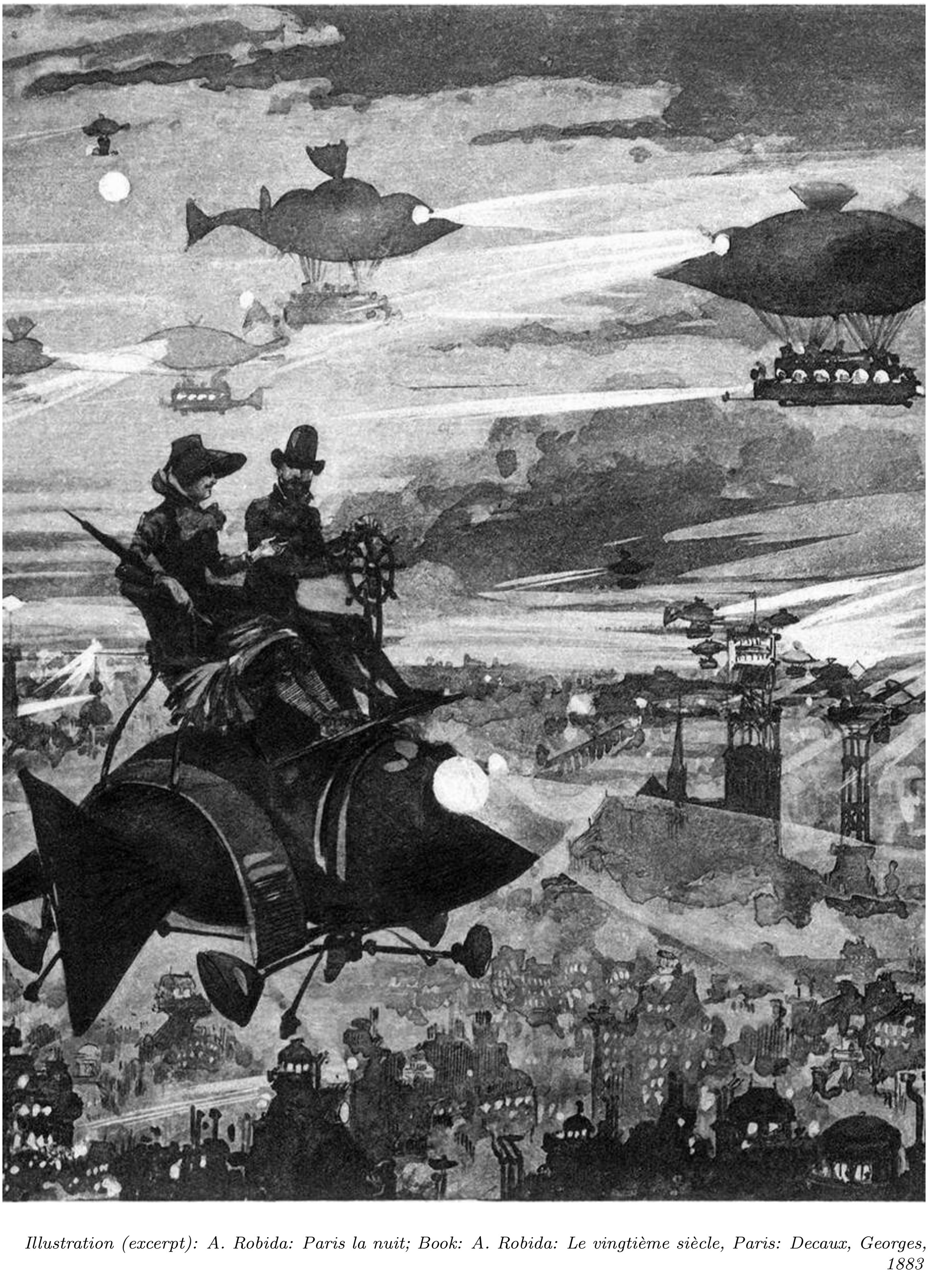


Illustration (excerpt): A. Robida: Paris la nuit; Book: A. Robida: Le vingtième siècle, Paris: Decaux, Georges, 1883

CHAPTER 3

Tracking and interaction control

This chapter covers disturbance-observer based trajectory tracking control and interaction control for multirotor flying robots. The geometric trajectory tracking controller in Section **??** is based on a disturbance observer approach, while interaction control, Section 3.2 is based on impedance and admittance control. Both are based on the external wrench estimator from Chapter 2. For trajectory tracking, the complete wrench is compensated, while for interaction inertia shaping is performed for compliant behavior. For interaction control under wind influence, the two methods are combined into the aerodynamics compensated impedance controller in Section 3.2.2. Based on discriminated aerodynamic and interaction wrenches, the controller compensates the aerodynamic wrench, and is compliant to the interaction wrench.

3.1 Trajectory tracking control

3.1.1 Attitude tracking control

Kinematics

In this following, three coordinate frames are considered: the non-moving inertial frame I, the body-fixed frame B, and the desired frame D. The goal of the attitude tracking controller is to align frame B with frame D. Unit quaternions are used for singularity-free attitude representation. The quaternion $\underline{\boldsymbol{q}} = [\eta \ \boldsymbol{\epsilon}^T]^T$ consists of the scalar part η and the vector part $\boldsymbol{\epsilon}$. To prevent notational ambiguity with vectors, we denote quaternions with an underbar. Unit quaternion attitude representation is related to the angle-axis representation through the half of the rotation angle:

$$\eta = \cos\tfrac{\varphi}{2}, \qquad \boldsymbol{\epsilon} = \boldsymbol{k}\sin\tfrac{\varphi}{2}, \tag{3.1}$$

where $\boldsymbol{k}$ is the rotation axis, and φ the rotation angle. The norm of a unit quaternion is always unity such that $\|\underline{\boldsymbol{q}}\| = \sqrt{\underline{\boldsymbol{q}}^T\underline{\boldsymbol{q}})} = 1$. The conjugate of a unit quaternion $\underline{\boldsymbol{q}}^* = [\eta \ -\boldsymbol{\epsilon}^T]^T$ represents inverse rotation. The rotation matrix $\boldsymbol{R}(\underline{\boldsymbol{q}})$ can be obtained by using the Euler-Rodriguez formula [103]. The kinematic differential equation of a unit quaternion is

$$\dot{\underline{\boldsymbol{q}}} = \tfrac{1}{2}\boldsymbol{U}(\underline{\boldsymbol{q}})\boldsymbol{\omega} = \tfrac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}^T \\ \eta\boldsymbol{I}_{3\times3} + \mathcal{S}(\boldsymbol{\epsilon}) \end{bmatrix}\boldsymbol{\omega}, \tag{3.2}$$

where $\boldsymbol{\omega} = [p\,q\,r]^T$ is the body angular velocity.

Attitude error

The attitude error $\underline{\tilde{\boldsymbol{q}}}$ between the desired attitude $\underline{\boldsymbol{q}}_{id}$ and the body attitude $\underline{\boldsymbol{q}}_{ib}$ is next defined geometrically [103, 76]. It is equivalent to $\tilde{\boldsymbol{R}} = \boldsymbol{R}(\underline{\tilde{\boldsymbol{q}}}) = \boldsymbol{R}_{db} = \boldsymbol{R}_{id}^T \boldsymbol{R}_{ib}$. The error represents the rotation from frame B to frame D. Using quaternion multiplication it can be defined as

$$\begin{aligned} \underline{\tilde{\boldsymbol{q}}} &= \underline{\boldsymbol{q}}_{db} = \underline{\boldsymbol{q}}_{di}^{*} \otimes \underline{\boldsymbol{q}}_{bi} \\ \underline{\tilde{\boldsymbol{q}}} &= \begin{bmatrix} \tilde{\eta} \\ \tilde{\boldsymbol{\epsilon}} \end{bmatrix} = \begin{bmatrix} \eta_{di} & \boldsymbol{\epsilon}_{di}^T \\ -\boldsymbol{\epsilon}_{di} & \eta_{di}\boldsymbol{I}_{3\times 3} - \mathcal{S}(\boldsymbol{\epsilon}_{di}) \end{bmatrix} \begin{bmatrix} \eta_{bi} \\ \boldsymbol{\epsilon}_{bi} \end{bmatrix}, \end{aligned} \tag{3.3}$$

where $\otimes$ represents quaternion multiplication. The angular velocity error transforms the desired angular velocity from the D-frame to the B-frame and is defined as

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} - \tilde{\boldsymbol{R}}^T \boldsymbol{\omega}_d, \qquad \dot{\tilde{\boldsymbol{\omega}}} = \dot{\boldsymbol{\omega}} - \tilde{\boldsymbol{R}}^T \dot{\boldsymbol{\omega}}_d - \dot{\tilde{\boldsymbol{R}}}^T \boldsymbol{\omega}_d, \tag{3.4}$$

where $\dot{\boldsymbol{R}} = \boldsymbol{R}\mathcal{S}(\boldsymbol{\omega})$ is the time derivative of the rotation matrix.

Attitude stabilization

To better explain quaternion attitude control, let us first consider the attitude stabilization problem without model uncertainties. Consider the Lyapunov function candidate

$$V = \tfrac{1}{2}\boldsymbol{\omega}^T \boldsymbol{\mathcal{I}} \boldsymbol{\omega} + 2cH(\tilde{\varepsilon}), \tag{3.5}$$

where the inertia $\boldsymbol{\mathcal{I}}$ is constant, c is positive and $H(\tilde{\varepsilon})$ is the quaternion error function. Because the error is a unit quaternion, the error function must be a Lipschitz function defined on the range $[-1, 1]$ and vanish at ± 1, since $\tilde{\eta} = \pm 1$ represents aligned B and D frames. Several common choices for $H(\tilde{\varepsilon})$ can be found in [103]. After expansion, the derivative of the Lyapunov function candidate is

$$\dot{V} = \boldsymbol{\omega}^T \left[\boldsymbol{m} + \mathcal{S}(\boldsymbol{\mathcal{I}}\boldsymbol{\omega})\,\boldsymbol{\omega} + \boldsymbol{m}_e\right] - c\,\boldsymbol{\omega}^T \tilde{\boldsymbol{e}},$$

where $\tilde{\boldsymbol{e}} = -\frac{\partial H}{\partial \tilde{\eta}} \tilde{\boldsymbol{\epsilon}}$ has been introduced. By assuming $\boldsymbol{h}_m = \boldsymbol{0}$ and taking the control law $\boldsymbol{m} = -\boldsymbol{K}_v \boldsymbol{\omega} - c\tilde{\boldsymbol{e}}$, with gain $\boldsymbol{K}_v \geq \boldsymbol{0}_{3\times 3}$, and using the skew-symmetry property of the Coriolis term in the dynamics, $\dot{V}$ becomes negative semidefinite

$$\dot{V} = -\boldsymbol{\omega}^T \boldsymbol{K}_v \boldsymbol{\omega} < 0, \qquad \forall \boldsymbol{\omega} \neq \boldsymbol{0}.$$

The equilibrium points depend on the chosen potential function of the quaternion error. Due to the structure of the $SO(3)$ group, at least two equilibrium points exist In this thesis, $H(\tilde{\eta}) = \tilde{\boldsymbol{\epsilon}}^T \tilde{\boldsymbol{\epsilon}}$ is chosen, which yields $\tilde{\boldsymbol{e}} = 2\tilde{\eta}\tilde{\boldsymbol{\epsilon}}$. When using this error function, the asymptotically stable equilibrium points are $\tilde{\eta} = \pm 1$, and an unstable equilibrium point exists at $\tilde{\eta} = 0$.

Robust attitude tracking control

In the following, an integral sliding mode (ISM) controller [145] is combined with disturbance observer based control [80]. The disturbance is estimated using the external wrench estimator from Chapter 2. Consider the desired attitude error dynamics to be

$$\dot{\tilde{\boldsymbol{\omega}}} + \boldsymbol{K}_v \dot{\tilde{e}} + c\tilde{e} = \boldsymbol{0}, \tag{3.6}$$

where instead of the angular velocity error, the time derivative of the geometric attitude error has been taken. For control design, an ideal model without disturbances is chosen to

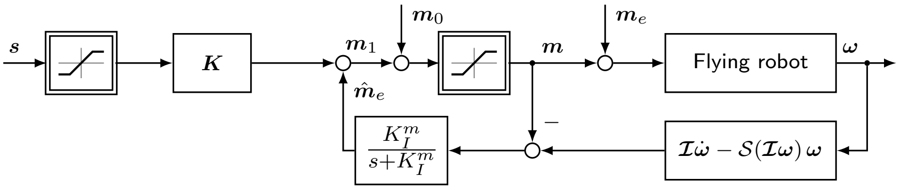


Figure 3.1: Block diagram of the attitude controller and disturbance observer. Copyright © 2014 IEEE [141].

be followed, such that

$$\boldsymbol{\mathcal{I}}_0\dot{\boldsymbol{\omega}}_0 = \boldsymbol{m}_0 + \mathcal{S}(\boldsymbol{\mathcal{I}}_0\boldsymbol{\omega}_0)\,\boldsymbol{\omega}_0. \tag{3.7}$$

In integral sliding mode controller design, the tracking (sliding) variable is expanded by a model-based integral term $\boldsymbol{z}$ which is designed such that sliding mode starts immediately without a reaching phase [145]. If the sliding variable $\boldsymbol{s}$ is defined

$$\boldsymbol{s} = \boldsymbol{s}_0 + \boldsymbol{z}, \quad \boldsymbol{s}_0 = \tilde{\boldsymbol{\omega}} + \boldsymbol{K}_v\tilde{\boldsymbol{e}}, \tag{3.8}$$

from the sliding condition $\dot{\boldsymbol{s}} = \boldsymbol{0}$, one obtains $\boldsymbol{z}$ to be

$$\dot{\boldsymbol{z}} = -\dot{\boldsymbol{s}}_0 = -\dot{\tilde{\boldsymbol{\omega}}}_0 - \boldsymbol{K}_v\dot{\tilde{\boldsymbol{e}}}, \quad \boldsymbol{z}(0) = -\boldsymbol{s}_0(0), \tag{3.9}$$

where the ideal error dynamics $\dot{\tilde{\boldsymbol{\omega}}}_0$ are obtained from the nominal model (3.7), using (3.4). The actual error dynamics can be obtained from (2.3) to be

$$\begin{aligned} \dot{\tilde{\boldsymbol{\omega}}}_0 &= \boldsymbol{\mathcal{I}}_0^{-1}\boldsymbol{m}_0 + \boldsymbol{\mathcal{I}}_0^{-1}\mathcal{S}(\boldsymbol{\mathcal{I}}_0\boldsymbol{\omega})\,\boldsymbol{\omega} - \tilde{\boldsymbol{R}}^T\dot{\boldsymbol{\omega}}_d - \dot{\tilde{\boldsymbol{R}}}^T\boldsymbol{\omega}_d, \\ \dot{\tilde{\boldsymbol{\omega}}} &= \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{m} + \boldsymbol{\mathcal{I}}^{-1}\mathcal{S}(\boldsymbol{\mathcal{I}}\boldsymbol{\omega})\,\boldsymbol{\omega} + \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{m}_e - \tilde{\boldsymbol{R}}^T\dot{\boldsymbol{\omega}}_d - \dot{\tilde{\boldsymbol{R}}}^T\boldsymbol{\omega}_d. \end{aligned} \tag{3.10}$$

The control law $\boldsymbol{m}$ consists of a nominal control $\boldsymbol{m}_0$ and a robust control $\boldsymbol{m}_1$. The nominal control law is obtained such that the nominal model (3.7) follows the desired error dynamics (3.6), while the robust control is determined from the following stability analysis. The atttiude controller is

$$\begin{aligned} \boldsymbol{m} &= \boldsymbol{m}_0 + \boldsymbol{m}_1 \\ \boldsymbol{m}_0 &= \boldsymbol{\mathcal{I}}_0\Big(\tilde{\boldsymbol{R}}^T\dot{\boldsymbol{\omega}}_d + \dot{\tilde{\boldsymbol{R}}}^T\boldsymbol{\omega}_d - \boldsymbol{K}_v\dot{\tilde{\boldsymbol{e}}} - c\tilde{\boldsymbol{e}}\Big) - \mathcal{S}(\boldsymbol{\mathcal{I}}_0\boldsymbol{\omega})\,\boldsymbol{\omega}. \end{aligned} \tag{3.11}$$

Next, the robust control $\boldsymbol{m}_1$ is derived. By applying (3.11), the derivative of the sliding variable (3.8) is

$$\begin{aligned} \dot{\boldsymbol{s}} &= \dot{\boldsymbol{s}}_0 + \dot{z} = \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2\boldsymbol{m}_0 + \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{m}_1 + \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{m}_e, \\ \boldsymbol{\zeta}_1 &= \boldsymbol{\mathcal{I}}^{-1}\mathcal{S}(\boldsymbol{\mathcal{I}}\boldsymbol{\omega})\,\boldsymbol{\omega} - \boldsymbol{\mathcal{I}}_0^{-1}\mathcal{S}(\boldsymbol{\mathcal{I}}_0\boldsymbol{\omega})\,\boldsymbol{\omega} \\ \boldsymbol{\zeta}_2 &= \boldsymbol{\mathcal{I}}^{-1} - \boldsymbol{\mathcal{I}}_0^{-1} \end{aligned} \tag{3.12}$$

The disturbance $\boldsymbol{m}_e$ can be estimated by the external wrench estimator from Section 2.3. This will include both modeling errors and external disturbance.

Next take $\tilde{\boldsymbol{m}}_e = \boldsymbol{m}_e - \hat{\boldsymbol{m}}_e$ and the Lyapunov function

$$V = \tfrac{1}{2}\boldsymbol{s}^T\boldsymbol{\mathcal{I}}\boldsymbol{s} + \tfrac{1}{2}\tilde{\boldsymbol{m}}_e^T\boldsymbol{K}_I^{m-1}\tilde{\boldsymbol{m}}_e, \tag{3.13}$$

whose derivative is

$$\dot{V} = \boldsymbol{s}^T \boldsymbol{\mathcal{I}}\Big(\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 \boldsymbol{m}_0\Big) + \boldsymbol{s}^T \boldsymbol{m}_1 + \boldsymbol{s}^T \boldsymbol{m}_e + \tilde{\boldsymbol{m}}_e^T \boldsymbol{K}_I^{m-1} (\dot{\boldsymbol{m}}_e - \dot{\hat{\boldsymbol{m}}}_e). \tag{3.14}$$

By choosing the robust control law $\boldsymbol{m}_1$ and disturbance estimation dynamics to be

$$\boldsymbol{m}_1 = -\boldsymbol{K}_w \mathrm{sgn}(\boldsymbol{s}) - \hat{\boldsymbol{m}}_e, \qquad \dot{\hat{\boldsymbol{m}}}_e = \boldsymbol{K}_I^m \tilde{\boldsymbol{m}}_e + \xi \, \mathrm{sgn}(\tilde{\boldsymbol{m}}_e) \tag{3.15}$$

the derivative of the Lyapunov function becomes

$$\dot{V} = \boldsymbol{s}^T \boldsymbol{\mathcal{I}}\Big(\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 \boldsymbol{m}_0\Big) - \boldsymbol{s}^T \boldsymbol{K}_w \mathrm{sgn}(\boldsymbol{s}) - \boldsymbol{s}^T \tilde{\boldsymbol{m}}_e - \tilde{\boldsymbol{m}}_e^T \tilde{\boldsymbol{m}}_e + \tilde{\boldsymbol{m}}_e^T \boldsymbol{K}_I^{m-1} \Big(\dot{\boldsymbol{m}}_e - \xi \, \mathrm{sign}(\tilde{\boldsymbol{m}}_e)\Big). \tag{3.16}$$

For (3.16) to be negative definite, it must hold that

$$\begin{aligned} \boldsymbol{K}_w > \frac{1}{\gamma} \|\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 \boldsymbol{m}_0 + \boldsymbol{\mathcal{I}}^{-1} \tilde{\boldsymbol{m}}_e\|, \quad & \gamma < \lambda_{\min}(\boldsymbol{\mathcal{I}}^{-1}) \\ & \xi > \sup \|\dot{\boldsymbol{m}}_e\| \end{aligned} \tag{3.17}$$

where it is assumed that $\boldsymbol{m}_e$ is Lipschitz. It can be seen that the sliding gain $\boldsymbol{K}_w$ depends on the modeling error $\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 \boldsymbol{m}_0$ as well as the disturbance estimation error $\tilde{\boldsymbol{m}}_e$. If the disturbance had not been estimated, the gain would also have to be larger than the maximum amplitude of the external disturbance, as well as the modeling errors. Thus, by incorporating an explicit disturbance estimator into the controller, the sliding gain can be smaller, which leads to improved robustness. The gain ξ in the disturbance estimator compensates for the rate of change of the disturbance. Note that here the external torque is interpreted as a disturbance. For purposes of the stability proof, it was expanded with a sign term. However, under the assumption of quasistatic disturbances, ξ can practically be chosen to be zero. In that case the sliding gain has to be higher in order to compensate $\boldsymbol{m}_e$. The equivalent control, or averaged motion, of the sliding mode term is

$$\boldsymbol{m}_{1,\mathrm{eq}} = \boldsymbol{\mathcal{I}}\Big(\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 \boldsymbol{m}_0\Big) + \tilde{\boldsymbol{m}}_e \tag{3.18}$$

Since using a signum function in the control leads to chattering, a boundary-layer approach is adpoted, which is equivalent to a lowpass-filtered signal of the signum function. By taking

$$\boldsymbol{m}_1 = -\boldsymbol{K}_w \mathrm{sat}\,(\boldsymbol{s}/\varepsilon) - \hat{\boldsymbol{m}}_e \tag{3.19}$$

with a small constant $\varepsilon > 0$, the system behaves as in (3.16) outside the boundary layer. The system will therefore not converge asymptotically to the tracking variable, but to its ε-vicinity. Inside the boundary layer an equivalent gain $\boldsymbol{K} = \boldsymbol{K}_w/\epsilon$ can be defined and $\xi = 0$ can be chosen to obtain

$$\begin{aligned} \dot{V} &= \boldsymbol{s}^T \boldsymbol{\mathcal{I}}\Big(\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 \boldsymbol{m}_0\Big) + \tilde{\boldsymbol{m}}_e^T \boldsymbol{K}_I^{m-1} \dot{\boldsymbol{m}}_e - W, \\ W &= \boldsymbol{s}^T \boldsymbol{K} \boldsymbol{s} + \boldsymbol{s}^T \tilde{\boldsymbol{m}}_e + \tilde{\boldsymbol{m}}_e^T \boldsymbol{K}_I^m \tilde{\boldsymbol{m}}_e \\ &\geq \lambda_{\min}\{\boldsymbol{K}\} \|\boldsymbol{s}\|^2 + \|\boldsymbol{s}\| \|\tilde{\boldsymbol{m}}_e\| + \lambda_{\min}\{\boldsymbol{K}_I^m\} \|\tilde{\boldsymbol{m}}_e\|^2. \end{aligned} \tag{3.20}$$

The condition for positive definiteness of W is

$$4\lambda_{\min}\{\boldsymbol{K}\} \lambda_{\min}\{\boldsymbol{K}_I^m\} > 1. \tag{3.21}$$

Hence, the system must be sufficiently damped inside the boundary layer to compensate the estimation error dynamics. An upper limit on $\boldsymbol{K}$ is imposed by the propeller dynamics and sensor noise. The Lyapunov function inside the boundary layer is not negative definite, but is dominated by the modeling and estimation errors. The error dynamics inside the boundary layer in case of $\boldsymbol{\mathcal{I}} \neq \boldsymbol{\mathcal{I}}_0$ can be obtained as

$$\begin{aligned} \dot{\tilde{\boldsymbol{\omega}}} + \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{\mathcal{I}}_0\boldsymbol{K}_v\tilde{\boldsymbol{\omega}} + \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{\mathcal{I}}_0 c\tilde{\boldsymbol{e}} &= \boldsymbol{\zeta}_1 - \boldsymbol{\mathcal{I}}^{-1}\tilde{\boldsymbol{m}}_e \\ \dot{\hat{\boldsymbol{m}}}_e + \boldsymbol{K}_I^m\hat{\boldsymbol{m}}_e &= \boldsymbol{K}_I^m\tilde{\boldsymbol{m}}_e \end{aligned} \tag{3.22}$$

which shows that the error dynamics inside the boundary layer is excited by the disturbance estimation error.

The boundary-layer integral sliding mode controller behaves like a saturated PID controller. Therefore, an anti-windup method must be applied [130]. During saturation of the sliding mode term in $\boldsymbol{m}_1$, the sliding surface is reset such that $\boldsymbol{z} = -\boldsymbol{s}_0$ and $\dot{\boldsymbol{z}} = \boldsymbol{0}$. Furthermore, saturated control inputs are used for disturbance observation, as depicted in Figure 3.1.

3.1.2 Position tracking control

The position controller is designed to track a desired position $\boldsymbol{r}_d = [x_d \; y_d \; z_d]^T$, velocity $\dot{\boldsymbol{r}}_d$ and acceleration $\ddot{\boldsymbol{r}}_d$. Control of the yaw angle ψ about the inertial z-axis is independent of the position. This is achieved by designing a controller in the inertial frame which calculates an inertial control force. The force is then used to generate an attitude and thrust reference for the underlying attitude controller.

Virtual control force

The design closely follows that of the attitude controller, therefore most details are omitted. The controller calculates a required force in the inertial frame based on the desired error dynamics

$$\ddot{\tilde{\boldsymbol{r}}} + \boldsymbol{K}_{v,p}\dot{\tilde{\boldsymbol{r}}} + \boldsymbol{K}_{p,p}\tilde{\boldsymbol{r}} = \boldsymbol{0}, \tag{3.23}$$

where $\tilde{\boldsymbol{r}} = \boldsymbol{r} - \boldsymbol{r}_d$ is the position error. We follow the ideal disturbance-free dynamics based on (2.3) without the external force to obtain

$$\mathcal{M}_0\ddot{\boldsymbol{r}}_0 = \mathcal{M}_0 g\boldsymbol{e}_3 - T\boldsymbol{R}\boldsymbol{e}_3, \tag{3.24}$$

where $\mathcal{M}_0$ is the nominal mass. A quaternion reference attitude is then generated that aligns the thrust vector with the desired control force. Equivalent to the attitude controller, the position control input $\boldsymbol{f} = \boldsymbol{f}_0 + \boldsymbol{f}_1$ consists of a nominal control $\boldsymbol{f}_0$, based on (3.24) and (3.23), and a robust control $\boldsymbol{f}_1$ such that

$$\begin{aligned} \boldsymbol{f}_0 &= \mathcal{M}_0\left(\ddot{\boldsymbol{r}}_d - \boldsymbol{K}_{v,p}\dot{\tilde{\boldsymbol{r}}} - \boldsymbol{K}_{p,p}\tilde{\boldsymbol{r}}\right) \\ \boldsymbol{f}_1 &= -\boldsymbol{R}\hat{\boldsymbol{f}}_e - \boldsymbol{K}_f \operatorname{sat}(\boldsymbol{s}_p/\boldsymbol{\varepsilon}_p) \end{aligned} \tag{3.25}$$

where $\ddot{\boldsymbol{r}}_d$ includes the gravity compensation term. By using the same integral sliding mode design process as in the attitude controller, the sliding variable is obtained to be

$$\boldsymbol{s}_p = \dot{\tilde{\boldsymbol{r}}} + \boldsymbol{K}_{v,p}\tilde{\boldsymbol{r}} + \int \boldsymbol{K}_{p,p}\tilde{\boldsymbol{r}}\,\mathrm{d}t - \dot{\tilde{\boldsymbol{r}}}(0) - \boldsymbol{K}_{v,p}\tilde{\boldsymbol{r}}(0) \tag{3.26}$$

Hence, the controller behaves locally as a PID controller. The thrust is equal to the norm of the desired force, i.e. $T = \|\boldsymbol{f}\|$.

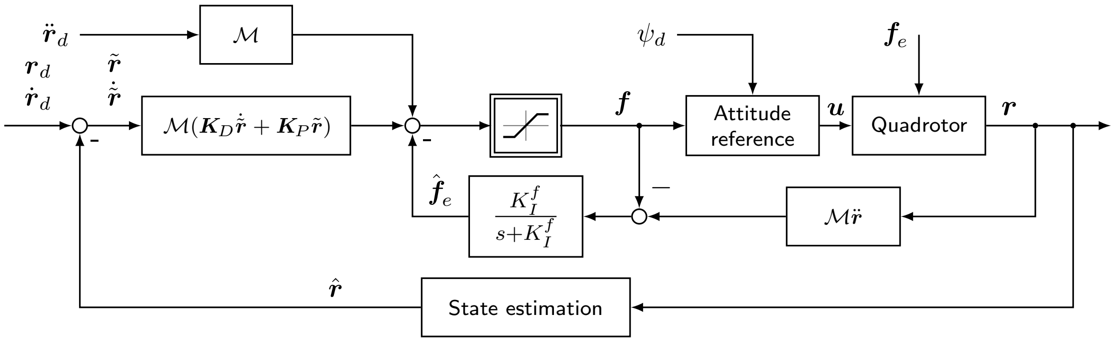
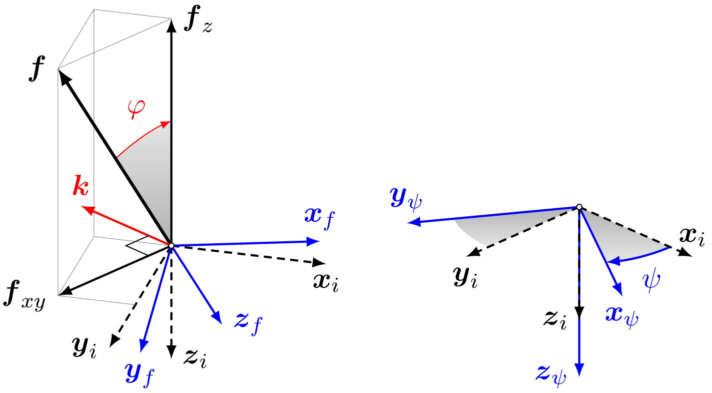


Figure 3.2: Block diagram of the position controller with disturbance observer.



(a) Thrust transformation $\underline{\boldsymbol{q}}_f$ (b) Yaw transformation $\underline{\boldsymbol{q}}_\psi$

Figure 3.3: Transformations involved in creating an attitude reference from the virtual control force $\boldsymbol{f}$. Copyright © 2014 IEEE [141].

Attitude reference generation

The attitude reference can be obtained from $\boldsymbol{f}$ through two transformations, depicted in Figure 3.3. The thrust transformation $\underline{\boldsymbol{q}}_f$ aligns the z_f-axis to the desired thrust vector, and is obtained from an angle-axis representation. The yaw transformation $\underline{\boldsymbol{q}}_\psi$ rotates about the inertial z-axis by angle ψ. The thrust vector points in the negative $\boldsymbol{z}_i$ direction in hover, so a rotation axis $\boldsymbol{k}$ can be obtained as the cross product between the desired inertial force and the negative $\boldsymbol{z}_i$ vector. The transformation between axes $\boldsymbol{z}_i$ and $\boldsymbol{z}_f$ can be obtained by normalizing the non-unit quaternion $\underline{\boldsymbol{q}}_f = [\eta_f \ \boldsymbol{\epsilon}_f]^T$

$$\eta_f = -\boldsymbol{z}_i^T\boldsymbol{f} + \sqrt{1 + \boldsymbol{f}^T\boldsymbol{f}}, \quad \boldsymbol{\epsilon}_f = -\boldsymbol{z}_i \times \boldsymbol{f} \tag{3.27}$$

The reference quaternion $\underline{\boldsymbol{q}}_d$ from the position controller is then obtained by transforming the intermediate yaw coordinate system by the thrust transformation as

$$\underline{\boldsymbol{q}}_d = \underline{\boldsymbol{q}}_f^0 \otimes \underline{\boldsymbol{q}}_\psi, \quad \underline{\boldsymbol{q}}_\psi = \left[\cos\tfrac{\psi}{2},\ 0,\ 0,\ \sin\tfrac{\psi}{2}\right]^T. \tag{3.28}$$

The transformation is free of singularities. Here, the yaw transformation represents the angle about the inertial z-axis, and not the Euler yaw angle.

Commanding an angular velocity improves attitude tracking performance. The angular velocity command can be generated by discretizing the kinematics of the rotation matrix and calculating delta rotations of the position controller between two time steps. The angular velocity at time step k is then

$$\boldsymbol{\omega}_{d,k} = {}^{\vee}\big(\boldsymbol{R}_{bi,k-1}^{T}\boldsymbol{R}_{bi,k}\big)\, t_s^{-1} \tag{3.29}$$

where t_s is the sampling time of the position controller, and ${}^{\vee}(\cdot)$ is the inverse of the skew-symmetric matrix operator, which extracts a vector from the matrix $\mathcal{S}(\cdot)$. In this way, angular velocity is generated from the feedback signal, rather than the feedforward signal from a precomputed trajectory.

3.1.3 Evaluation of trajectory tracking

In this section, behavior of the controller with disturbance observation is first shown when model uncertainties are applied in simulation. The transient response is compared to a PID controller and influence of the integral term is shown. Three sets of experiments are presented, where four controllers are compared – PID, Adaptive Integral Backstepping Controller (AIBC), Integral Sliding Mode with Disturbance Observation (ISM+DO) and PD with disturbance observation (PD+DO). First, hovering performance with and without turbulent wind influence is compared across position and attitude controllers. Second, trajectory tracking between waypoints is shown for situations where large angles must be applied. Lastly, behavior of the disturbance observation method is shown for cases when a constant force in the inertial frame is applied, and the response is compared with the AIBC. For presentation clarity, the ZYX Euler angles are shown, while quaternions are used for control.

For a fair comparison, all controllers have been tuned to have the same local closed-loop gains as a standard PID controller $f_{\mathrm{PID},x} = \mathcal{M}_0(\ddot{x}_d - K_{v,x}\dot{\tilde{x}} - K_{p,x}\tilde{x} - K_{i,x}\int\tilde{x}\mathrm{d}\tau)$ with gains $\boldsymbol{K}_{v,p} = 2\omega_c$, $\boldsymbol{K}_{v,p} = \omega_c^2$, $\boldsymbol{K}_{i,p} = \frac{1}{4}\omega_c$. This applies for the sliding mode boundary layer and AIBC controller. Disturbance observers were tuned separately. The AIBC position controller was implemented as three decoupled controllers of the form [153, 39] as

$$f_x = m_0\left(\ddot{x}_d - (1 - c_1^2 + \lambda)\tilde{x} - (c_1 + c_2)\dot{\tilde{x}} + c_1\lambda\xi_x - \hat{h}_x\right),$$

where $\xi_x = \int\tilde{x}\mathrm{d}\tau$ for the x-axis and equivalent for other axes, with the disturbance estimation as $\dot{\hat{h}}_x = \gamma\dot{e}$. Since the mass is constant and is not adapted in other controllers, the adaptation has not been implemented for the AIBC.

Simulation results

The simulation model includes sensor noise, propeller modeling and actuator discretization. Angular accelerations are obtained through finite-difference differentiation. Figure 3.4 shows the step position disturbance response for different controllers, which is applied between 1 s and 8 s, with an ISM+DO attitude controller. The PD+DO approach shows faster convergence than the PID controller. The combined ISM+DO scheme inside the boundary layer has a faster response and a smaller absolute error, however the transient has an undershoot due to integrator influence.

The response to time-varying disturbances is shown next in Figure 3.5. The quadrotor is commanded to hover at a constant position, while time-varying disturbances are applied, as a torque about the x-axis (roll), and a force in the inertial y-axis. The force is counteracted by changing the roll angle, and is thereby coupled with the torque disturbance. It can be seen that the disturbance is counteracted very closely despite being time-varying. The

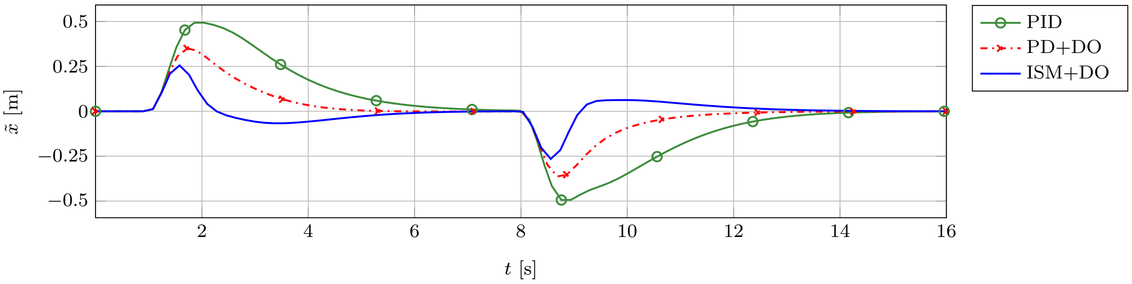


Figure 3.4: Simulation comparison of position controller errors for setpoint control and disturbance. A constant force disturbance of 2 N is applied in the inertial x-direction between 1 s and 8 s. The controller parameters for the x-axis are $\mathcal{M}_0 = \mathcal{M} = 0.5\,\text{kg}$, $\omega_{c,x} = 1.5\,\text{rad/s}$, $K_{sf,x} = 1.5m_0$, $\varepsilon = 1$, $K^f_{I,x} = 5$. Copyright © 2014 IEEE [141].

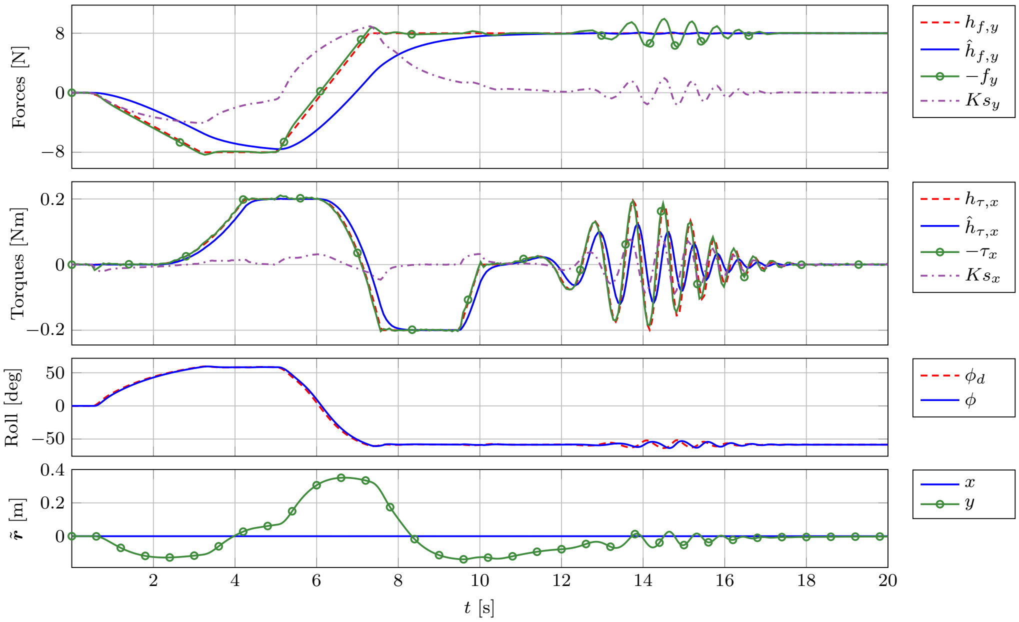
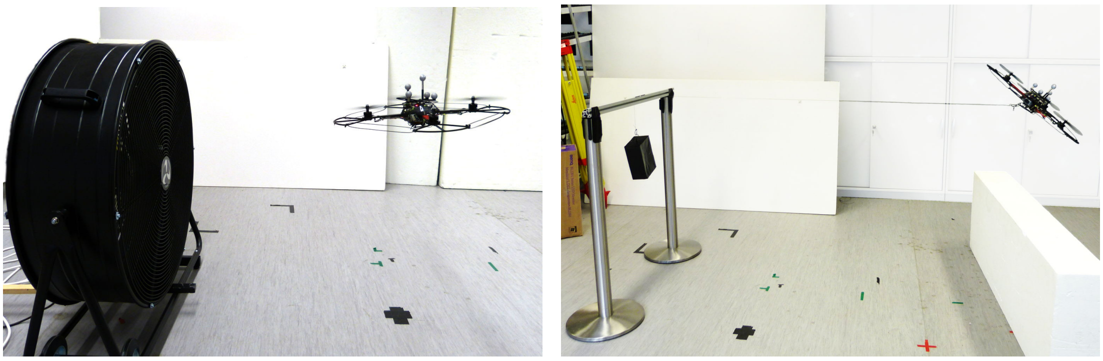


Figure 3.5: Estimation and compensation of time-varying disturbances in simulation, using the ISM+DO method. The quadrotor is commanded to hover in place while the disturbances are applied on the roll axis and in the inertial y-direction. The disturbance compensation is faster due to ISM, and the tracking variable s (term Ks_y) converges to zero when the disturbance is constant. Copyright © 2014 IEEE [141].

slow component is identified by the acceleration-based disturbance estimator, while the faster component is counteracted through the integral sliding mode term $\boldsymbol{Ks}$. A chirp torque disturbance with varying amplitude can be compensated, even though the lowpass disturbance estimate has a considerable phase delay. The system is in the boundary layer, and the sliding variable $\boldsymbol{s}$ goes to zero once the constant disturbance is counteracted by the disturbance estimate. Position and attitude are of course coupled, as a change in the attitude due to the disturbance causes a change in position, so the quadrotor must fly back to the hover position. Note that here the disturbance amplitude is larger than the quadrotor weight, so the roll angle must be held at 60 degrees in hover.

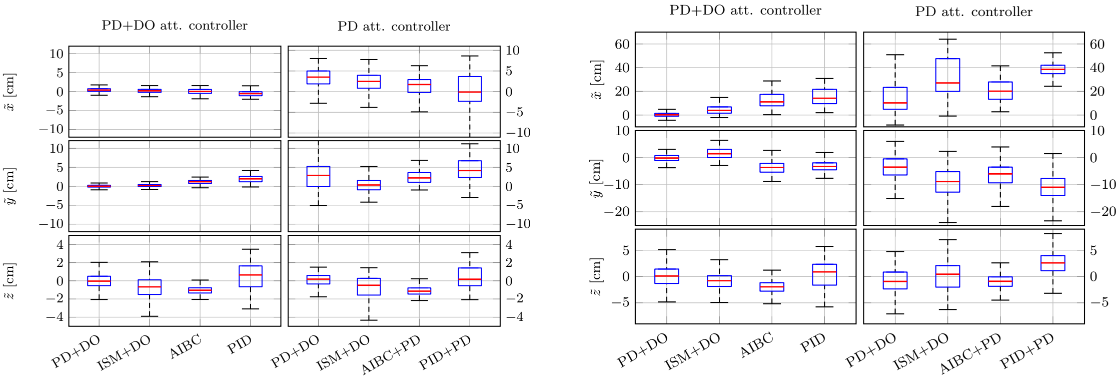
Experimental validation

The experiments are carried out using an AscTec Hummingbird that runs the quaternion attitude controller with disturbance observer onboard at 1 kHz. A strapdown algorithm



(a) Hovering in a 3 m/s turbulent wind stream, generated by a fan with 70 cm diameter (b) Hovering with a constant side force produced by a suspended 6 N weight

Figure 3.6: Experimental setups for hovering tests with external disturbance investigated in this thesis. Copyright © 2014 IEEE [141].



(a) Hovering without wind influence (b) Hovering in 3 m/s turbulent wind stream

Figure 3.7: Hovering performance using different position and attitude controllers. Controller parameters were constant through the experiments. The attitude controller was tuned as $\omega_{c,a} = 12\,\mathrm{rad/s}$ and $\boldsymbol{K}_I^m = 8$. All position controllers had $\omega_{c,p} = 1.5\,\mathrm{rad/s}$, and the mass was constant at $\mathcal{M} = \mathcal{M}_0 = 0.63\,\mathrm{kg}$. Position DO parameters were $\boldsymbol{K}_I^f = \mathrm{diag}\{[3\ 3\ 1.5]\}$ and AIBC $\boldsymbol{\gamma} = \mathrm{diag}\{[4\ 4\ 2]\}$. Copyright © 2014 IEEE [141].

integrates the onboard gyroscopes to obtain the attitude estimate that is used for the control. The attitude drift is corrected from motion tracking measurements. Position and attitude measurement are provided by an A.R.T. motion tracking system at 60 Hz. The position controller runs in Simulink, and sends attitude and thrust reference commands to the quadrotor via a wireless XBee link. The angular acceleration is obtained onboard by numerically differentiating the gyro signals. Translational velocity and acceleration are obtained by differentiating and filtering raw position measurements from the motion tracking system, hence a small delay is introduced.

First, the influence of a disturbance observer in hover conditions is investigated. All experiments have been carried out with the same hardware and under same conditions. Since it is common to use a PD attitude controller on the Hummingbird platform, the quaternion controller in PD form is compared to a controller with disturbance observation. The quadrotor parameters are not ideally known, so modeling errors exists. Therefore, as no external disturbances are present, these will dominate the error dynamics. Figure 3.7(a)

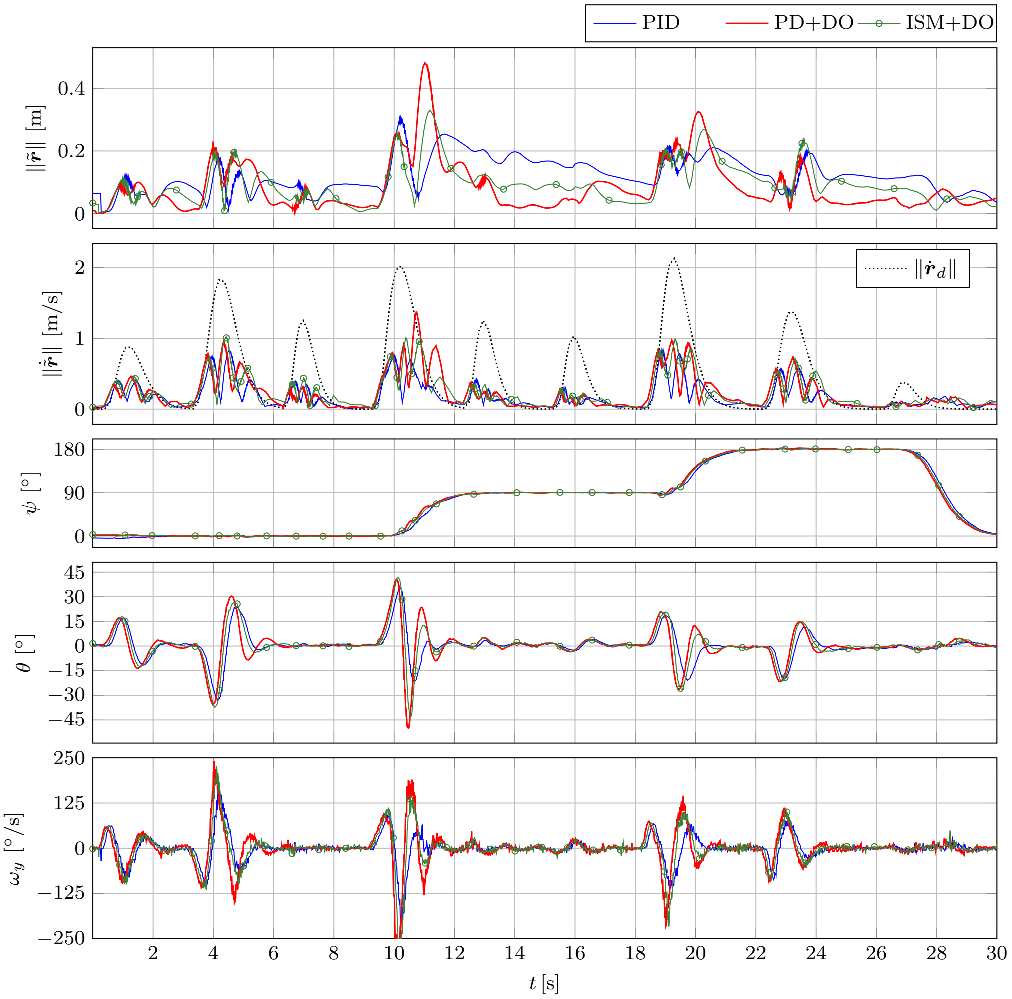
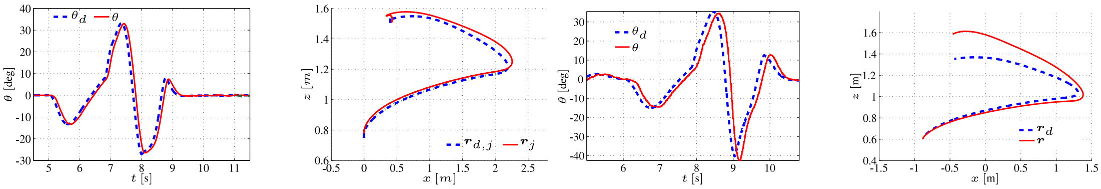


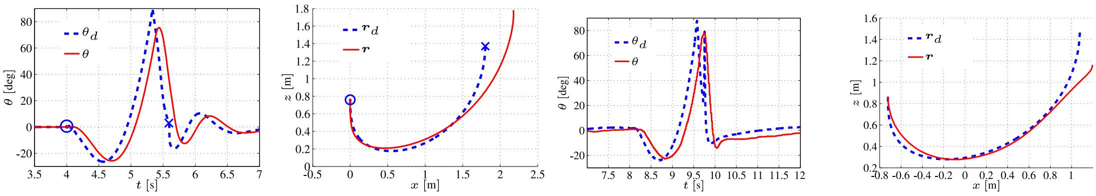
Figure 3.8: Position and velocity tracking during aggressive waypoint trajectory tracking. The maximum commanded velocity is 2 m/s. Due to overshoot, the max. reached velocity was 3 m/s. The roll and pitch angles reach 50°, and the commanded angular velocity reaches the limit of 250°/s. Copyright © 2014 IEEE [141].

shows box plots for the four position controllers and attitude controllers with and without disturbance observation. It can be seen that only using a disturbance observer in the attitude controller improves performance in the horizontal plane. It does not significantly influence altitude control. In hover, the ISM controller is always inside the boundary layer, so the integral term adds a low frequency component that spreads the error distribution when combined with the disturbance observer.

Next, the influence of turbulent wind on hovering performance is analyzed. The setup is depicted in Figure 3.6(a). The fan generates a turbulent wind stream with a velocity of 3 m/s, resulting in a ∼15° hover. Figure 3.7(b) shows very consistent performance of the PD+DO position control approach when combined with the attitude disturbance observer. It can be seen that the PD attitude controller performs much worse in turbulent conditions. It can be concluded that the position controller performance can be significantly improved in turbulent wind conditions by simply adding a disturbance observer to the attitude controller. Adding a disturbance observer to the position control loop further improves performance. However, the ISM integral term does not further improve the hovering performance.



(a) Slow trajectory simulation (b) Slow trajectory experiment



(c) Aggressive trajectory simulation (d) Aggressive trajectory experiment

Figure 3.9: Simulation and experimental results for a Hummingbird quadrotor executing Dynamic Movement Primitive (DMP) trajectories [137]. The quadrotor is running a PD+DO geometric attitude controller, and a PD position controller. Copyright © 2014 IEEE [137].

Figure 3.8 shows position and velocity tracking errors for a trajectory tracking experiment. The velocity and acceleration are generated by filtering the position between waypoints. Here, the maximal reference velocity was about 2 m/s. The ISM+DO controller shows fast and very oscillatory behavior. It can be seen that the PD+DO controller shows the fastest convergence of the position error, however it also shows large velocity overshoot. This is due to lumping the unmodeled drag force in the disturbance. Here, performance can be improved by modeling the velocity-dependent drag forces. The obtained pitch angle and commanded angular velocity show that the angles reach 45°, The onboard gyroscopes are rated to 300°/s, so the commanded signal is limited to avoid sensor saturation. This leads to lower tracking performance in periods of high acceleration. Yaw tracking remains good throughout the flights. The roll angle and angular velocity are not shown.

Trajectory tracking of aggressive trajectories represented by Dynamic Movement Primitives (DMPs) [137] is depicted in Figure 3.9. It can be observed that the attitude controller can track very aggressive trajectories, up to 80 ° in pitch, and stabilize rapidly. The main difference between simulation and experimental results is additional delay in experiments, possibly due to unmodeled motor dynamics.

Lastly, the transient response to a step disturbance in the inertial x-direction is compared. The setup is shown in Figure 3.6(b). A weight of 6 N (610 g) is suspended and bound to the quadrotor with a string. The quadrotor, with a mass of 630 g, is then commanded to hover and the weight is dropped. This produces a constant horizontal force in the x-direction on the quadrotor. Figure 3.10 shows that the DO identifies the weight quickly. The quantitative error indicates errors in modeling of the quadrotor thrust. The hover position is reached quickly. The ISM+DO approach shows faster convergence of the error to zero, as shown in simulations. In equilibrium, the quadrotor hovers at approx. 45°. It is importantl to note that releasine the weight does not cause an overshoot. The velocity-based disturbance observation of the AIBC does not correctly identify this disturbance, since the weight falls to the ground and the quadrotor velocity reaches zero. Therefore, the disturbance estimate converges to a constant value. The integral term is too slow to drive the error to zero. Hence, the DO-based approach is well suited for applications where relatively large external forces are present because these are estimated directly from the

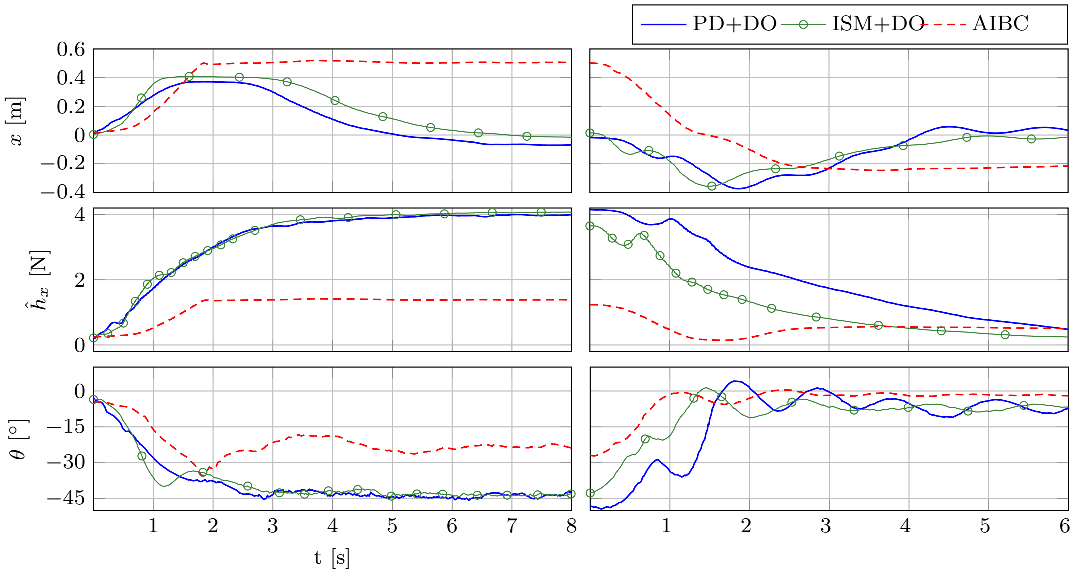


Figure 3.10: Position, pitch and disturbance estimate in the suspended weight experiment with PD+DO (—), ISM+DO (-⊖-) and AIBC (- -) controllers. The left column shows transients when the force is applied to the quadrotor, and the right column when it is released. For the DO-based approaches, the quadrotor hovers at about 45° in equilibrium. The AIBC does not correctly identify the external force, as the weight falls to the ground. Position DO parameters were $\boldsymbol{K}_I^f = \mathrm{diag}\{[7\ 1.5\ 1.5]\}$ and AIBC $\boldsymbol{\gamma} = \mathrm{diag}\{[8\ 3\ 2]\}$. Copyright © 2014 IEEE [141].

acceleration.

3.2 Interaction control

In this section, the integration of the external wrench estimate $\hat{\boldsymbol{\tau}}_e$ into the robot control feedback loop is considered. The goal is to enable the system to purposefully interact with its surroundings. Suitable interaction behaviors can be obtained through *impedance control* and *admittance control* [108]. In the impedance control approach, the interaction properties of the flying system are directly controlled. The resulting controller replaces the existing position control loop of a flying robot. Admittance control, on the other hand, is implemented around an existing position control loop [11]. However, is more difficult to select the gains to obtain the desired interaction behavior than in the case of a clean impedance controller. In the following, both interaction control approaches are discussed.

3.2.1 Impedance control with inertia shaping

For interaction control, a robot should ideally represent an impedance, as was argued in the seminal work of Hogan [61]. Consider the target closed loop dynamics to be

$$\boldsymbol{M}_v\dot{\tilde{\boldsymbol{\nu}}} + \boldsymbol{D}_v\tilde{\boldsymbol{\nu}} + \boldsymbol{K}_v\tilde{\boldsymbol{x}} = \boldsymbol{\tau}_e, \tag{3.30}$$

where $\boldsymbol{M}_v \in \mathbb{R}^{6\times 6}$ is the desired positive definite apparent inertia matrix, $\boldsymbol{D}_v \in \mathbb{R}^{6\times 6}$ is the desired positive definite diagonal damping matrix, and $\boldsymbol{K}_v = \mathrm{blockdiag}\{\boldsymbol{K}_{v,t},\ c\,\boldsymbol{I}_{3\times 3}\}$ is the desired positive definite diagonal stiffness matrix. Here, $\boldsymbol{K}_v \in \mathbb{R}^{3\times 3}$ is the translational stiffness, and c is the rotational stiffness about all three axes. The state and velocity tracking errors are $\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{r}}\ \tilde{\eta}\tilde{\boldsymbol{\epsilon}}]^T$ and $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - \boldsymbol{\nu}_d$, respectively. Here, $\underline{\tilde{\boldsymbol{q}}} := [\tilde{\eta}\ \tilde{\boldsymbol{\epsilon}}]^T$ is the quaternion representation of the geometric orientation error, with $\tilde{\eta}$ being the scalar, and

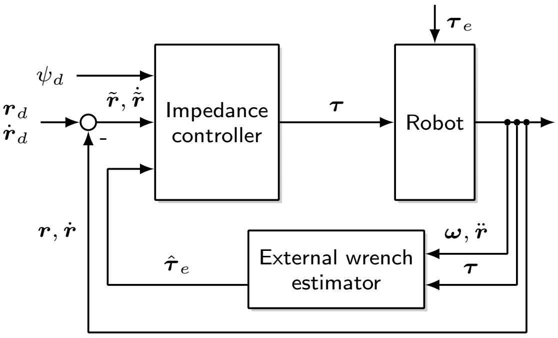

$\boldsymbol{\tau}_e$
ψ_d
$\boldsymbol{r}_d$
$\dot{\boldsymbol{r}}_d$
$\tilde{\boldsymbol{r}}, \dot{\tilde{\boldsymbol{r}}}$
-
Impedance controller
$\boldsymbol{\tau}$
Robot
$\boldsymbol{r}, \dot{\boldsymbol{r}}$
$\boldsymbol{\omega}, \ddot{\boldsymbol{r}}$
$\hat{\boldsymbol{\tau}}_e$
External wrench estimator
$\boldsymbol{\tau}$

Figure 3.11: Structure of an impedance controller (3.31) if it were possible to apply forces in every direction. This would be the case for a UAV with fully actuated degrees of freedom, where $\mathrm{rank}(\boldsymbol{B}) = 6$. Copyright © 2017 IEEE [135, 138].

$\tilde{\boldsymbol{\epsilon}}$ the vector part. By inserting (3.30) into (2.4) the required control input becomes

$$\boldsymbol{J}^T\boldsymbol{\tau} = (\boldsymbol{M}\boldsymbol{M}_v^{-1} - \boldsymbol{I})\boldsymbol{\tau}_e + \boldsymbol{M}\dot{\boldsymbol{\nu}}_d - \boldsymbol{M}\boldsymbol{M}_v^{-1}(\boldsymbol{D}_v\tilde{\boldsymbol{\nu}} + \boldsymbol{K}_v\tilde{\boldsymbol{x}}) + \boldsymbol{N}. \tag{3.31}$$

Since a measurement of $\boldsymbol{\tau}_e$ is not available, its estimate $\hat{\boldsymbol{\tau}}_e$ is used in (3.31). The resulting control structure for a fully actuated flying robot ($\mathrm{rank}(\boldsymbol{B}) = 6$) is shown in Figure 3.11. The impedance control structure for a nonholonomic flying robot (e.g. quadcopter) is shown in Figure 3.12. The resulting system response along the inertial x-direction is shown in Figure 3.13. The impedance controller has a simpler overall structure and it is easier to select its gains than for the admittance controller.

Remark 1 (Closed-loop dynamics). By applying $\hat{\boldsymbol{\tau}}_e$ in (3.31), the closed-loop dynamics of controller (3.31) for system (2.4) is

$$\boldsymbol{M}_v\dot{\tilde{\boldsymbol{\nu}}} + \boldsymbol{D}_v\tilde{\boldsymbol{\nu}} + \boldsymbol{K}_v\tilde{\boldsymbol{x}} = \boldsymbol{\tau}_e + (\boldsymbol{M}_v\boldsymbol{M}^{-1} - \boldsymbol{I})\tilde{\boldsymbol{\tau}}_e, \tag{3.32}$$

where $\tilde{\boldsymbol{\tau}}_e := \boldsymbol{\tau}_e - \hat{\boldsymbol{\tau}}_e$ is the external wrench estimation error. Equation (3.32) shows that controller (3.31) achieves the desired closed-loop dynamics up to the scaled wrench estimation error. The steady state ($\dot{\boldsymbol{\nu}} = \boldsymbol{0}$, $\boldsymbol{\nu} = \boldsymbol{0}$, $\tilde{\boldsymbol{\tau}}_e = \boldsymbol{0}$, $\dot{\tilde{\boldsymbol{\tau}}}_e = \boldsymbol{0}$) equilibrium point is obviously $\tilde{\boldsymbol{x}} = \boldsymbol{K}_v^{-1}\boldsymbol{\tau}_e$. Systems (3.32) and (2.34) may be rewritten using $\boldsymbol{z} := [\tilde{\boldsymbol{x}}\ \tilde{\boldsymbol{\nu}}\ \tilde{\boldsymbol{\tau}}_e]^T$, in matrix form

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{g}(t,\boldsymbol{z}), \tag{3.33}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{T} & \boldsymbol{0} \\ -\boldsymbol{M}_v^{-1}\boldsymbol{K}_v & -\boldsymbol{M}_v^{-1}\boldsymbol{D}_v & \boldsymbol{M}^{-1} - \boldsymbol{M}_v^{-1} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{K}_I \end{bmatrix}, \tag{3.34}$$

$$\boldsymbol{g}(t) = [\boldsymbol{0} \quad \boldsymbol{M}_v^{-1}\boldsymbol{\tau}_e \quad \dot{\boldsymbol{\tau}}_e]^T. \tag{3.35}$$

Here $\boldsymbol{T}$ is the kinematic transformation matrix with $\dot{\boldsymbol{x}} = \boldsymbol{T}\boldsymbol{\nu}$ and $\det(\boldsymbol{T}) = 1$. The nominal (unperturbed) system $\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z}$ will be asymptotically stable if $\mathrm{Re}(\boldsymbol{A}) < 0$. Note that in that case the coupling term $\left(\boldsymbol{M}^{-1} - \boldsymbol{M}_v^{-1}\right)$ in (3.34) does not affect stability due to zero submatrices in $\boldsymbol{A}$. Next, passivity and stability of the impedance controller are shown.

We note that with most flying systems (3.31) can not be obtained in the general case. Flying systems such as quadrotors are underactuated and therefore cannot generate horizontal forces in the body frame. The target impedance dynamics (3.30) can only be obtained in the in the direction of the rotor thrust, which is typically the body z-axis. Therefore, to obtain a Cartesian impedance, the robot must turn its body z-axis in the

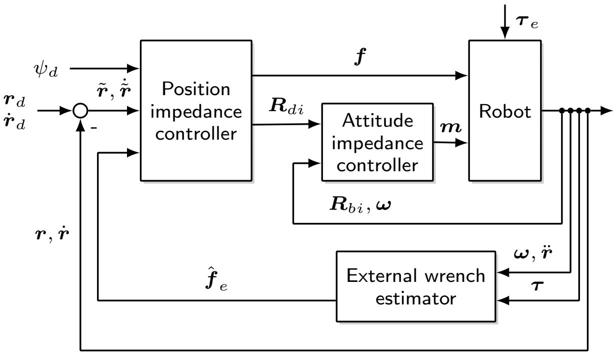

$\boldsymbol{\tau}_e$
ψ_d
$\boldsymbol{r}_d$
$\dot{\boldsymbol{r}}_d$
$\tilde{\boldsymbol{r}}, \dot{\tilde{\boldsymbol{r}}}$
-
Position impedance controller
$\boldsymbol{f}$
$\boldsymbol{R}_{di}$
Attitude impedance controller
$\boldsymbol{m}$
Robot
$\boldsymbol{R}_{bi}, \boldsymbol{\omega}$
$\boldsymbol{r}, \dot{\boldsymbol{r}}$
$\boldsymbol{\omega}, \ddot{\boldsymbol{r}}$
$\boldsymbol{\tau}$
External wrench estimator
$\hat{\boldsymbol{f}}_e$

Figure 3.12: Control structure of an impedance controller for an underactuated UAV, like a typical quadrotor with rank$(\boldsymbol{B}) < 6$. Position impedance is only achievable in the thrust direction. Therefore the attitude controller must be used to obtain the desired impedance in Cartesian translational directions. The controller calculates a virtual control force in the inertial frame $\boldsymbol{f}_c = T\boldsymbol{R}_d\boldsymbol{e}_3$. This force is then decomposed into the desired thrust T and attitude $\boldsymbol{R}_d$ using state of the art methods. Copyright © 2017 IEEE [135, 138].

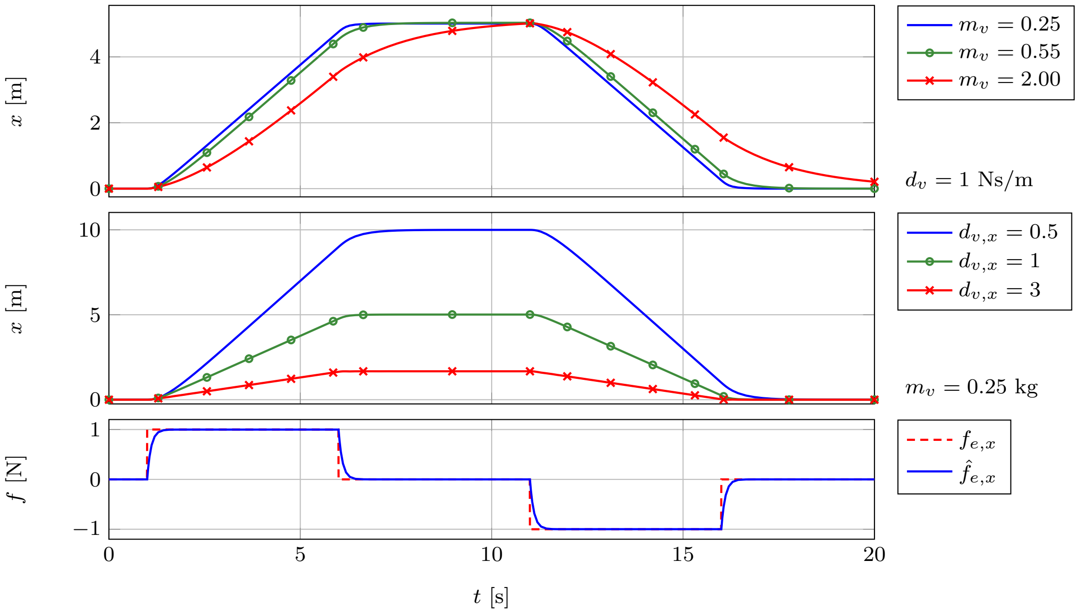


Figure 3.13: Simulated response of the position impedance controller to an external force in the inertial $x-$direction, for varying virtual inertia and damping parameters. The system is in damping mode — the virtual stiffness is $\boldsymbol{K}_v = \boldsymbol{0}$. The real system mass is $m = 0.55$ kg. Copyright © 2017 IEEE [135, 138].

direction of the desired force. This is achieved by using a cascaded control structure as depicted in Figure 3.12, which will be the subject of our analysis. By introducing $\mu := \mathcal{M}\mathcal{M}_v^{-1}$ and $\boldsymbol{\Gamma} := \boldsymbol{\mathcal{I}}\boldsymbol{\mathcal{I}}_v^{-1}$, the position and attitude impedance controllers can be written as

$$\boldsymbol{R}_d\boldsymbol{f} = \mathcal{M}\ddot{\boldsymbol{r}}_d + (\mu - 1)\,\hat{\boldsymbol{f}}_e - \mu\big(\boldsymbol{D}_1\dot{\tilde{\boldsymbol{r}}} + \boldsymbol{K}_1\tilde{\boldsymbol{r}}\big) - \boldsymbol{g}, \tag{3.36}$$

$$\boldsymbol{m} = \boldsymbol{\mathcal{I}}\dot{\boldsymbol{\omega}}_d + (\boldsymbol{\Gamma} - \boldsymbol{I})\,\hat{\boldsymbol{m}}_e - \boldsymbol{\Gamma}(\boldsymbol{D}_2\tilde{\boldsymbol{\omega}} + c_v\,\tilde{\boldsymbol{e}}) - \boldsymbol{n}, \tag{3.37}$$

where $\boldsymbol{g} = \mathcal{M}g\boldsymbol{e}_3$, $\tilde{\boldsymbol{e}} = \tilde{\eta}\tilde{\boldsymbol{\varepsilon}}$, $\boldsymbol{n} = \mathcal{S}(\boldsymbol{\mathcal{I}}\boldsymbol{\omega})\,\boldsymbol{\omega} - \boldsymbol{m}_g$. The overall structure of the described system is depicted in Figure 3.14. In order to incorporate the attitude error dynamics into the error dynamics of the translation subsystem, we define the *control force error*.

Definition 1 (Control force error)**.** Suppose that the desired thrust T can be obtained instantaneously, i. e. that the thrust dynamics can be neglected w. r. t. the attitude dynamics. This assumption is valid due to the motor dynamics and speed control being faster than the attitude control loop. The desired control force $\boldsymbol{f}_c := T\boldsymbol{R}_d\boldsymbol{e}_3$ is then achieved perfectly, whereas the actual control force depends on the actual attitude as $\boldsymbol{f} = T\boldsymbol{R}\boldsymbol{e}_3$. The control force error $\tilde{\boldsymbol{f}} := \boldsymbol{f} - \boldsymbol{f}_c$ is

$$\begin{aligned} \tilde{\boldsymbol{f}} &= T(\tilde{\boldsymbol{R}} - \boldsymbol{I})\boldsymbol{R}_d\boldsymbol{e}_3, \\ &= 2T\big((\tilde{\eta}\boldsymbol{I} + \mathcal{S}(\tilde{\varepsilon}))\mathcal{S}(\tilde{\varepsilon})\big)\boldsymbol{R}_d\boldsymbol{e}_3, \end{aligned} \tag{3.38}$$

from which we obtain the norm estimates $\left\|\tilde{\boldsymbol{f}}\right\| \leq 2T\left\|\tilde{\varepsilon}\right\|$ and $\left\|\dot{\tilde{\boldsymbol{f}}}\right\| \leq 2\dot{T}\left\|\tilde{\varepsilon}\right\| + T\left\|\tilde{\boldsymbol{\omega}}\right\|$. By limiting the thrust to the range $0 < T < T_{\max}$, we obtain the upper bound $\left\|\tilde{\boldsymbol{f}}\right\| < 2T_{\max}\left\|\tilde{\varepsilon}\right\|$.

The error dynamics of the impedance controllers is

$$\mathcal{M}_v\ddot{\tilde{\boldsymbol{r}}} + \boldsymbol{D}_1\dot{\tilde{\boldsymbol{r}}} + \boldsymbol{K}_1\tilde{\boldsymbol{r}} = \boldsymbol{f}_e + (\bar{\mu} - 1)\tilde{\boldsymbol{f}}_e + \bar{\mu}\tilde{\boldsymbol{f}}, \tag{3.39}$$

$$\boldsymbol{\mathcal{I}}_v\dot{\tilde{\boldsymbol{\omega}}} + \boldsymbol{D}_2\tilde{\boldsymbol{\omega}} + c_v\tilde{\eta}\tilde{\varepsilon} = \boldsymbol{m}_e + (\bar{\boldsymbol{\Gamma}} - \boldsymbol{I})\tilde{\boldsymbol{m}}_e, \tag{3.40}$$

where $\bar{\mu} = \mu^{-1} = \mathcal{M}_v\mathcal{M}^{-1}$, $\bar{\boldsymbol{\Gamma}} = \boldsymbol{\mathcal{I}}_v\boldsymbol{\mathcal{I}}^{-1}$, $\tilde{\boldsymbol{f}}_e := \boldsymbol{f}_e - \hat{\boldsymbol{f}}_e$ and $\tilde{\boldsymbol{m}}_e := \boldsymbol{m}_e - \hat{\boldsymbol{m}}_e$ are the force and torque estimation errors, respectively, and $\tilde{\boldsymbol{f}}$ is the control force error resulting from the attitude tracking error.

Proposition 1. *The attitude impedance controlled subsystem* (3.40) *is passive w.r.t. the pair* $(\tilde{\boldsymbol{\omega}}, \boldsymbol{m}_e + (\bar{\boldsymbol{\Gamma}} - \boldsymbol{I})\tilde{\boldsymbol{m}}_e)$*.*

Proof. Take the storage function

$$V_1 = \tfrac{1}{2}\tilde{\boldsymbol{\omega}}^T\boldsymbol{\mathcal{I}}_v\tilde{\boldsymbol{\omega}} + \tfrac{1}{2}c_v\,\tilde{\varepsilon}^T\tilde{\varepsilon}.$$

The derivative of V_1 is

$$\begin{aligned} \dot{V}_1 &= \tilde{\boldsymbol{\omega}}^T\big(\boldsymbol{m}_e + (\bar{\boldsymbol{\Gamma}} - \boldsymbol{I})\tilde{\boldsymbol{m}}_e - \boldsymbol{D}_2\tilde{\boldsymbol{\omega}} - c_v\,\tilde{\eta}\tilde{\varepsilon}\big) + c_v\,\tilde{\eta}\tilde{\boldsymbol{\omega}}^T\tilde{\varepsilon} \\ \dot{V}_1 &= -\tilde{\boldsymbol{\omega}}^T\boldsymbol{D}_2\tilde{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}^T\left(\boldsymbol{m}_e + (\bar{\boldsymbol{\Gamma}} - \boldsymbol{I})\tilde{\boldsymbol{m}}_e\right), \end{aligned}$$

which shows passivity of the subsystem. Here, the estimator error $\tilde{\boldsymbol{m}}_e$ is an additional scaled input. □

Proposition 2. *The position impedance controlled subsystem* (3.52) *is passive w.r.t. the pair* $(\dot{\tilde{\boldsymbol{r}}}, \boldsymbol{f}_e + (\bar{\mu} - 1)\tilde{\boldsymbol{f}}_e + \bar{\mu}\tilde{\boldsymbol{f}})$*.*

Proof. Take the storage function

$$V_2 = \tfrac{1}{2}\dot{\tilde{\boldsymbol{r}}}^T\mathcal{M}_v\dot{\tilde{\boldsymbol{r}}} + \tfrac{1}{2}\tilde{\boldsymbol{r}}^T\boldsymbol{K}_1\tilde{\boldsymbol{r}}.$$

The derivative of V_2 is

$$\dot{V}_2 = -\dot{\tilde{\boldsymbol{r}}}^T\boldsymbol{D}_1\dot{\tilde{\boldsymbol{r}}} + \dot{\tilde{\boldsymbol{r}}}^T\left(\boldsymbol{f}_e + (\bar{\mu} - 1)\tilde{\boldsymbol{f}}_e + \bar{\mu}\tilde{\boldsymbol{f}}\right),$$

which shows passivity of the subsystem. □

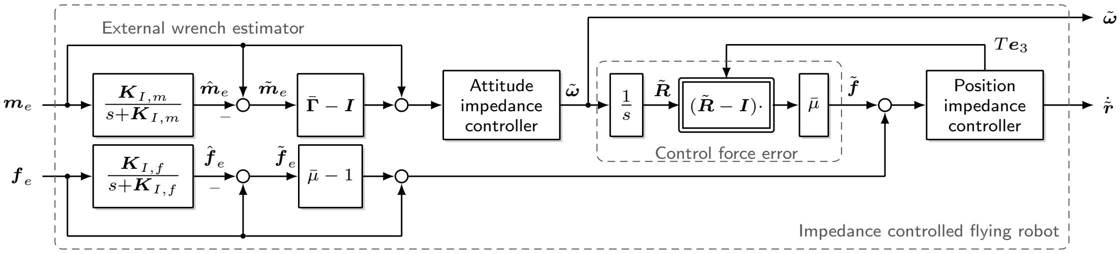

External wrench estimator
$\boldsymbol{m}_e$
$\frac{\boldsymbol{K}_{I,m}}{s+\boldsymbol{K}_{I,m}}$
$\hat{\boldsymbol{m}}_e$
$\tilde{\boldsymbol{m}}_e$
$-$
$\bar{\boldsymbol{\Gamma}}-\boldsymbol{I}$
Attitude impedance controller
$\tilde{\boldsymbol{\omega}}$
$\frac{1}{s}$
$\tilde{\boldsymbol{R}}$
$(\tilde{\boldsymbol{R}}-\boldsymbol{I})\cdot$
$\bar{\mu}$
$\tilde{\boldsymbol{f}}$
$T\boldsymbol{e}_3$
Position impedance controller
$\tilde{\boldsymbol{\omega}}$
$\dot{\boldsymbol{r}}$
Control force error
$\boldsymbol{f}_e$
$\frac{\boldsymbol{K}_{I,f}}{s+\boldsymbol{K}_{I,f}}$
$\hat{\boldsymbol{f}}_e$
$\tilde{\boldsymbol{f}}_e$
$-$
$\bar{\mu}-1$
Impedance controlled flying robot

Figure 3.14: Blocks considered in the passivity proof of the cascaded impedance controlled flying robot, considering the external wrench estimator and attitude error dynamics. In order for the complete system to be passive, the interconnected subsystems must be passive or lossless. The external wrench estimator has first-order error dynamics and is therefore passive. We show passivity of the attitude impedance controller w.r.t. the pair $(\tilde{\boldsymbol{\omega}}, \boldsymbol{m}_e + (\bar{\boldsymbol{\Gamma}} - \boldsymbol{I})\tilde{\boldsymbol{m}}_e)$. Furthermore, the control force error $\tilde{\boldsymbol{f}}$ is obtained by a lossless transformation. Lastly, we show passivity of the position impedance controller w.r.t. the pair $(\dot{\boldsymbol{r}}, \boldsymbol{f}_e + (\bar{\mu}-1)\tilde{\boldsymbol{f}}_e + \bar{\mu}\tilde{\boldsymbol{f}})$. Copyright © 2017 IEEE [138].

Note that in the case without inertia shaping, the interaction ports of the attitude controller correspond to the physical power ports. For the position controller, this only holds for $\tilde{\boldsymbol{f}} = 0$. Note that for a full passivity proof for a passive environment w.r.t $(\boldsymbol{f}_e, \boldsymbol{m}_e)$ and $(\dot{\boldsymbol{r}}, \tilde{\boldsymbol{\omega}})$ further steps would be necessary.

However, in the following we show local input-to-state stability (ISS) [72] of the complete impedance controlled system, taking into account the external wrench estimation dynamics. In the analysis we assume that the external wrench is bounded and differentiable, but do not impose any specific form on the bounds. In this way, we allow the analysis to be applied to more specific assumptions on the external wrench, such as e.g. [63]; As an example of bounds on the external force, one may use $\|\boldsymbol{f}_e\| \leq c_{f1} + c_{f2}\|\dot{\boldsymbol{r}} - \boldsymbol{v}_w\|$, see [63]. First, we recall the definition of ISS and Theorem 4.19 from [72].

Definition 2. The system $\dot{x} = f(t,x,u)$ is said to be input-to-state stable if there exist a class $\mathcal{KL}$ function β and a class $\mathcal{K}$ class function γ such that for any initial state $x(t_0)$ and any bounded input $u(t)$, the solution $x(t)$ exists for all $t \geq t_0$ and satisfies

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0) + \gamma\left(\sup_{t_0 \leq \tau \leq t} \|u(\tau)\|\right).$$

Theorem 1. *Let $V : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that*

$$\alpha_1(\|\boldsymbol{x}\|) \leq V(t,x) \leq \alpha_2(\|\boldsymbol{x}\|),$$

$$\frac{\partial V}{\partial t} + \left(\frac{\partial V}{\partial \boldsymbol{x}}\right)^T \boldsymbol{f}(t,\boldsymbol{x},\boldsymbol{u}) \leq -W_3(\boldsymbol{x}), \quad \forall \|\boldsymbol{x}\| \geq \rho(\|\boldsymbol{u}\|) > 0$$

$\forall(t,\boldsymbol{x},\boldsymbol{u}) \in [0,\infty) \times \mathbb{R}^n \times \mathbb{R}^m$, where α_1, α_2 are class $\mathcal{K}_\infty$ functions, ρ is a class $\mathcal{K}$ function, and $W_3(\boldsymbol{x})$ is a continuous positive definite function on $\mathbb{R}^n$. Then, the system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t,\boldsymbol{x},\boldsymbol{u})$$

is input-to-state stable with $\gamma = \alpha_1^{-1} \circ \alpha_2 \circ \rho$. ◇

Proposition 3. *The impedance controlled flying robot as depicted in Figure 3.14 is locally input-to-state stable for inputs $\boldsymbol{u}^T = [\boldsymbol{f}_e^T\ \dot{\boldsymbol{f}}_e^T\ \boldsymbol{m}_e^T\ \dot{\boldsymbol{m}}_e]$ and states $\boldsymbol{x}^T = [\tilde{\boldsymbol{r}}^T\ \dot{\boldsymbol{r}}^T\ \tilde{\boldsymbol{f}}_e^T\ \tilde{\boldsymbol{\varepsilon}}^T\ \tilde{\boldsymbol{\omega}}^T\ \tilde{\boldsymbol{m}}_e^T]$*

for all $\mathrm{eig}(\boldsymbol{P}) \succ 0$*,* $\mathrm{eig}(\boldsymbol{N}) \succ 0$*, and* $\boldsymbol{Q} \succ 0$*, with* $\boldsymbol{P}$ *defined in* (3.42)*,* $\boldsymbol{N}$ *defined in* (3.45)*, and* $\boldsymbol{Q}$ *defined in* (3.46)*.*

Proof. Take the candidate function for the complete system

$$V = \tfrac{1}{2}\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}, \tag{3.41}$$

where

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{K}_1 & c_1\boldsymbol{I} & \cdot & \cdot & \cdot & \cdot \\ c_1\boldsymbol{I} & \mathcal{M}_v\boldsymbol{I} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \boldsymbol{K}_{I,f}^{-1} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & c_v\boldsymbol{I} & c_2\boldsymbol{I} & \cdot \\ \cdot & \cdot & \cdot & c_2\boldsymbol{I} & \boldsymbol{\mathcal{I}}_v & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \boldsymbol{K}_{I,m}^{-1} \end{bmatrix}, \tag{3.42}$$

with $\cdot \equiv \boldsymbol{0}_{3\times 3}$ used for readability, and $c_1 > 0$, $c_2 > 0$ chosen such that $\mathrm{eig}(\boldsymbol{P}) \succ 0$ (see e.g. [77]). Then, V is bounded by

$$\tfrac{1}{2}\lambda_{\min}\{\boldsymbol{P}\}\,\|\boldsymbol{x}\|_2^2 \le V \le \tfrac{1}{2}\lambda_{\max}\{\boldsymbol{P}\}\,\|\boldsymbol{x}\|_2^2\,, \tag{3.43}$$

$$\begin{aligned} \alpha_1(\|\boldsymbol{x}\|) &= \tfrac{1}{2}\lambda_{\min}\{\boldsymbol{P}\}\,\|\boldsymbol{x}\|_2^2\,, \\ \alpha_2(\|\boldsymbol{x}\|) &= \tfrac{1}{2}\lambda_{\max}\{\boldsymbol{P}\}\,\|\boldsymbol{x}\|_2^2\,, \end{aligned}$$

and after algebraic manipulation its derivative can be found as

$$\dot{V} < -\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{N}\boldsymbol{v}, \tag{3.44}$$

where

$$\begin{aligned} \boldsymbol{y}^T &= \left[\|\tilde{\boldsymbol{r}}\|\ \|\dot{\tilde{\boldsymbol{r}}}\|\ \left\|\tilde{\boldsymbol{f}}_e\right\|\ \|\tilde{\boldsymbol{\varepsilon}}\|\ \|\tilde{\boldsymbol{\omega}}\|\ \|\tilde{\boldsymbol{m}}_e\|\right], \\ \boldsymbol{v}^T &= \left[\|\boldsymbol{f}_e\|\ \left\|\dot{\boldsymbol{f}}_e\right\|\ \|\boldsymbol{m}_e\|\ \|\dot{\boldsymbol{m}}_e\|\right] \end{aligned}$$

contain the state as $\boldsymbol{y}(\|\boldsymbol{x}\|)$ and input bounds as $\boldsymbol{v}(\|\boldsymbol{u}\|)$, and their coupling matrix $\boldsymbol{N} \in \mathbb{R}^{6\times 4}$ is

$$\boldsymbol{N} = \begin{bmatrix} c_{\mathcal{M}} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \lambda_{\min}^{-1}\{\boldsymbol{K}_{I,f}\} & 0 & 0 \\ 0 & 0 & c_{\mathcal{I}} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \lambda_{\min}^{-1}\{\boldsymbol{K}_{I,m}\} \end{bmatrix} \tag{3.45}$$

where $c_{\mathcal{M}} = c_1\mathcal{M}_v^{-1}$ and $c_{\mathcal{I}} = c_2\lambda_{\min}^{-1}\{\boldsymbol{\mathcal{I}}_v\}$ for $\boldsymbol{v} > 0$. The matrix $\boldsymbol{Q} \in \mathbb{R}^{6\times 6}$ can be written as

$$\boldsymbol{Q} = \begin{bmatrix} c_{\mathcal{M}}\lambda_{\min}\{\boldsymbol{K}_1\} & \frac{1}{2}c_{\mathcal{M}}\lambda_{\min}\{\boldsymbol{D}_1\} & -\frac{1}{2}c_{\mathcal{M}}(\bar{\mu}-1) & c_{\mathcal{M}}\bar{\mu}T_{\max} & 0 & 0 \\ \frac{1}{2}c_{\mathcal{M}}\lambda_{\min}\{\boldsymbol{D}_1\} & \lambda_{\min}\{\boldsymbol{D}_1\} - c_1 & -\frac{1}{2}(\bar{\mu}-1) & \bar{\mu}T_{\max} & 0 & 0 \\ -\frac{1}{2}c_{\mathcal{M}}(\bar{\mu}-1) & -\frac{1}{2}(\bar{\mu}-1) & 1 & 0 & 0 & 0 \\ c_{\mathcal{M}}\bar{\mu}T_{\max} & \bar{\mu}T_{\max} & 0 & c_v c_{\mathcal{I}} & \frac{1}{2}c_{\mathcal{I}}\lambda_{\min}\{\boldsymbol{D}_2\} & -\frac{1}{2}c_{\mathcal{I}}(\lambda_{\min}\{\bar{\boldsymbol{\Gamma}}\}-1) \\ 0 & 0 & 0 & \frac{1}{2}c_{\mathcal{I}}\lambda_{\min}\{\boldsymbol{D}_2\} & \lambda_{\min}\{\boldsymbol{D}_2\} - \frac{1}{2}c_2 & \frac{1}{2}(\lambda_{\min}\{\bar{\boldsymbol{\Gamma}}\}-1) \\ 0 & 0 & 0 & -\frac{1}{2}c_{\mathcal{I}}(\lambda_{\min}\{\bar{\boldsymbol{\Gamma}}\}-1) & \frac{1}{2}(\lambda_{\min}\{\bar{\boldsymbol{\Gamma}}\}-1) & 1 \end{bmatrix}. \tag{3.46}$$

Take $\theta:\ 0 < \theta < 1$ and rewrite $\dot{V}$ as

$$\dot{V} < -\theta\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{N}\boldsymbol{v} - (1-\theta)\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y}. \tag{3.47}$$

The quadratic term in $\dot{V}$ is negative definite iff $\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y} > 0$, i.e. $\boldsymbol{Q}$ must be positive definite. Then, $\dot{V}$ will be negative definite at least in the region where the quadratic term is larger than the linear term, i.e where $\theta \boldsymbol{y}^T \boldsymbol{Q} \boldsymbol{y} > \boldsymbol{y}^T \boldsymbol{N} \boldsymbol{v}$. From

$$\theta\, \lambda_{\min}\{\boldsymbol{Q}\}\, \|\boldsymbol{y}\|^2 > \lambda_{\max}\{\boldsymbol{N}\}\, \|\boldsymbol{y}\|\, \|\boldsymbol{v}\| \tag{3.48}$$

we obtain

$$\dot{V} < -(1-\theta)\boldsymbol{y}^T \boldsymbol{Q} \boldsymbol{y}, \quad \forall\, \|\boldsymbol{y}\| > \rho(\|\boldsymbol{u}\|) = \frac{\|\boldsymbol{v}\|}{\theta}\, \frac{\lambda_{\max}\{\boldsymbol{N}\}}{\lambda_{\min}\{\boldsymbol{Q}\}}, \tag{3.49}$$

for all $\lambda_{\max}\{\boldsymbol{N}\} > 0$. It follows that the complete system is locally input-to-state stable with

$$\gamma = \alpha_1^{-1}(\alpha_2(\rho(\|\boldsymbol{u}\|))) = \frac{\|\boldsymbol{v}\|}{\theta}\, \frac{\lambda_{\max}\{\boldsymbol{N}\}}{\lambda_{\min}\{\boldsymbol{Q}\}}\, \sqrt{\frac{\lambda_{\max}\{\boldsymbol{P}\}}{\lambda_{\min}\{\boldsymbol{P}\}}}, \tag{3.50}$$

where $\rho(\|\boldsymbol{u}\|)$ defines the region of local ISS. ☐

Remark 2. Notice that the matrix $\boldsymbol{Q}$ can be subdivided into the position control subsystem with force estimation dynamics in $\boldsymbol{Q}_p \in \mathbb{R}^{3\times 3}$, the attitude control subsystem with torque estimation dynamics in $\boldsymbol{Q}_a \in \mathbb{R}^{3\times 3}$, and the coupling matrix $\boldsymbol{Q}_{pa} \in \mathbb{R}^{3\times 1}$ as

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_p & \boldsymbol{Q}_{pa} \\ \boldsymbol{Q}_{pa}^T & \boldsymbol{Q}_a \end{bmatrix}, \boldsymbol{Q}_p = \begin{bmatrix} \boldsymbol{Q}_p^0 & \boldsymbol{Q}_{p1}^0 \\ \boldsymbol{Q}_{p1}^{0,T} & 1 \end{bmatrix}, \boldsymbol{Q}_a = \begin{bmatrix} \boldsymbol{Q}_a^0 & \boldsymbol{Q}_{a1}^0 \\ \boldsymbol{Q}_{a1}^{0,T} & 1 \end{bmatrix} \tag{3.51}$$

and $\boldsymbol{Q}_p$ and $\boldsymbol{Q}_a$ can be further subdivided into the respective error dynamics $\boldsymbol{Q}_p^0$ and $\boldsymbol{Q}_a^0$, estimator dynamics, and coupling matrices $\boldsymbol{Q}_{p1}^0$ and $\boldsymbol{Q}_{a1}^0$. Therefore, showing positive definiteness of $\boldsymbol{Q}$ can be shown by showing positive definiteness – and therefore local input-to-state stability – of the respective subsystems. Due to space constraints and complicated analytical expressions, we omit the particular conditions for the positive definiteness of $\boldsymbol{Q}$ and $\boldsymbol{P}$. This can be also seen as a linear matrix inequality (LMI) problem [22] and solved numerically.

Remark 3. The following qualitative thoughts can be given from analyzing (3.50). First, the bounds will obviously grow with the external inputs $\|\boldsymbol{v}\|$, as well as a smaller θ. Increasing the maximum eigenvalue of $\boldsymbol{Q}$ will decrease the ISS bounds. This can be achieved by increasing the minimum damping, stiffness, and virtual inertia. Analyzing the eigenvalues of $\boldsymbol{N}$, one finds that $c_{\mathcal{M}}$ is the only non-zero eigenvalue. Therefore, increasing the virtual mass will reduce the ISS bounds in this term. Lastly, decreasing the eigenvalues of $\boldsymbol{P}$ (the bounds of the Lyapunov function), and improving the conditioning of $\boldsymbol{P}$ will reduce the bounds further. This can be interpreted as increasing the observer gains, and reducing the virtual inertia and stiffness, and them being of the same order of magnitude as the inverse of the observer gains. This is clearly a compromise with tuning $\lambda_{\min}\{\boldsymbol{Q}\}$. Lastly, a numerical example applied to one of the experiments is given in Section 3.2.6.

3.2.2 Compensated impedance control

In the foregoing analysis we implicitly assumed that the external wrench is caused by interaction forces only. However, aerial robots are typically subject to wind influence. In that case, we would ideally compensate for the wind influence, and apply an impedance only to the interaction wrench. Suppose now that we are able to distinguish between the external wrench components $\boldsymbol{\tau}_e = \boldsymbol{\tau}_d + \boldsymbol{\tau}_i$, consisting of the aerodynamic drag wrench $\boldsymbol{\tau}_d$ and interaction wrench $\boldsymbol{\tau}_i$. It is then possible to compensate the aerodynamic wrench and apply an impedance on the interaction wrench $\boldsymbol{\tau}_i$. To analyze the stability of such a controller, we focus on the position controller for simplicity. The desired error dynamics

$$\mathcal{M}_v \ddot{\tilde{\boldsymbol{r}}} + \boldsymbol{K}_d \dot{\tilde{\boldsymbol{r}}} + \boldsymbol{K}_p \tilde{\boldsymbol{r}} = \boldsymbol{f}_i \tag{3.52}$$

results in the *compensated impedance controller*

$$\boldsymbol{R}_d \boldsymbol{f} = \mathcal{M}\ddot{\boldsymbol{r}}_d - \hat{\boldsymbol{f}}_d + (\mu - 1)\,\hat{\boldsymbol{f}}_i - \mu\big(\boldsymbol{D}_1\dot{\tilde{\boldsymbol{r}}} + \boldsymbol{K}_1\tilde{\boldsymbol{r}}\big) - \boldsymbol{g}, \tag{3.53}$$

where $\hat{\boldsymbol{f}}_d$ is the estimated aerodynamic force and $\hat{\boldsymbol{f}}_i$ is the estimated interaction force. We do not assume any properties on $\hat{\boldsymbol{f}}_d$ and $\hat{\boldsymbol{f}}_i$ other than $\hat{\boldsymbol{f}}_e = \hat{\boldsymbol{f}}_d + \hat{\boldsymbol{f}}_i$. The previous stability analysis may be applied to this controller by expanding the state $\tilde{\boldsymbol{f}}_e$ into $\tilde{\boldsymbol{f}}_i$ and $\tilde{\boldsymbol{f}}_d$, and setting the appropriate coupling matrix $\boldsymbol{Q}_{p1}^0$. The stability bounds will in this case depend on the quality of discrimination between the external force components. Furthermore, the estimaton error $\tilde{\boldsymbol{f}}_i$ must be bounded, and will have the same effect as the attitude control error $\tilde{\boldsymbol{f}}$. Note that in the case of $\boldsymbol{f}_i = \boldsymbol{0}$ and $\mu = 1$ we obtain a disturbance observer (DO) based controler discussed earlier in this chapter. Interestingly, the DO controller is equivalent to setting $(\mathcal{M}\mathcal{M}_v^{-1} - 1)\boldsymbol{f}_e = -\boldsymbol{f}_e$ in the impedance controller, i.e. setting the virtual mass to $\mathcal{M}_v = \infty$.

3.2.3 Admittance control

Essentially, admittance control generates velocity commands as a function of the contact wrench. The resulting trajectory may then be tracked using a position and attitude controller. In effect, this adds an additional cascade around the position control loop. The reference velocity and position are typically generated via the virtual dynamical system

$$\boldsymbol{M}_a\ddot{\tilde{\boldsymbol{x}}}_v + \boldsymbol{D}_a\dot{\tilde{\boldsymbol{x}}}_v + \boldsymbol{K}_a\tilde{\boldsymbol{x}}_v = \boldsymbol{\tau}_v, \tag{3.54}$$

where $\boldsymbol{x}_v = [\boldsymbol{r}_d \ \psi_d]$ is the resulting position trajectory, $\boldsymbol{M}_a = \text{blockdiag}\{\mathcal{M}_a\boldsymbol{I}_{3\times 3} \ \mathcal{I}_{a,z}\}$ with $m_a > 0$ and $\mathcal{I}_{a,z} > 0$ is the virtual admittance inertia matrix, $\boldsymbol{D}_a \in \mathbb{R}^{4\times 4}$ is the positive definite diagonal virtual damping gain matrix, $\boldsymbol{K}_a \in \mathbb{R}^{4\times 4}$ is the positive diagonal virtual spring gain matrix, and $\boldsymbol{\tau}_v = [\hat{\boldsymbol{f}}_e \ \hat{m}_{e,z}]^T$ is the estimated external wrench, with $\hat{m}_{e,z}$ being the torque about the z-axis. The torque information is used to implement an admittance on the yaw angle. The desired roll and pitch angles are commanded via the position controller. Similarly to the compensated impedance controller, a compensated admittance controller may be achieved by using a disturbance observer based position tracking controller, and applying the interaction wrench in (3.54) as $\boldsymbol{\tau}_v = [\hat{\boldsymbol{f}}_i \ \hat{m}_{e,z}]^T$.

In the case $\boldsymbol{K}_a = \boldsymbol{0}_{4\times 4}$, a human operator can move the robot freely in space. The steady state velocity will then be $\dot{\boldsymbol{x}}_{v,ss} = \boldsymbol{D}_a^{-1}\boldsymbol{M}_a^{-1}\boldsymbol{\tau}_v$, which in turn could be used to design the desired behavior of the admittance system. The admittance control structure is shown in Figure 3.15, and the response of (3.54) in Figure 3.16. The admittance controller adds another cascade into the control loop. This makes it simple to implement the method on a system where a position controller is already available. Note that now there is a fourth order system between the input $\boldsymbol{\tau}_v$ and the system states, as opposed to the impedance controller which imposes a second order system.

3.2.4 Discussion and practical considerations

Fig. 3.13 and 3.16 show responses of impedance and admittance controllers under the same external force, using various parameters. It can be seen that the behavior of the impedance controller is intuitively more consistent, e.g. a different virtual mass does not change the steady state position. It is thus easier to select the appropriate gains than for an admittance controller. An impedance controller replaces the position control loop of a flying robot. In contrast, the admittance controller can be easily implemented on existing systems around an existing position control loop.

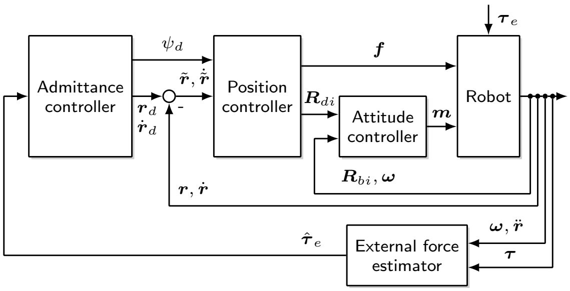


Figure 3.15: Structure of the admittance controller (3.54). The controller generates a position and yaw reference, which is then tracked by a position controller. The external wrench causes a velocity. Copyright © 2017 IEEE [135, 138].

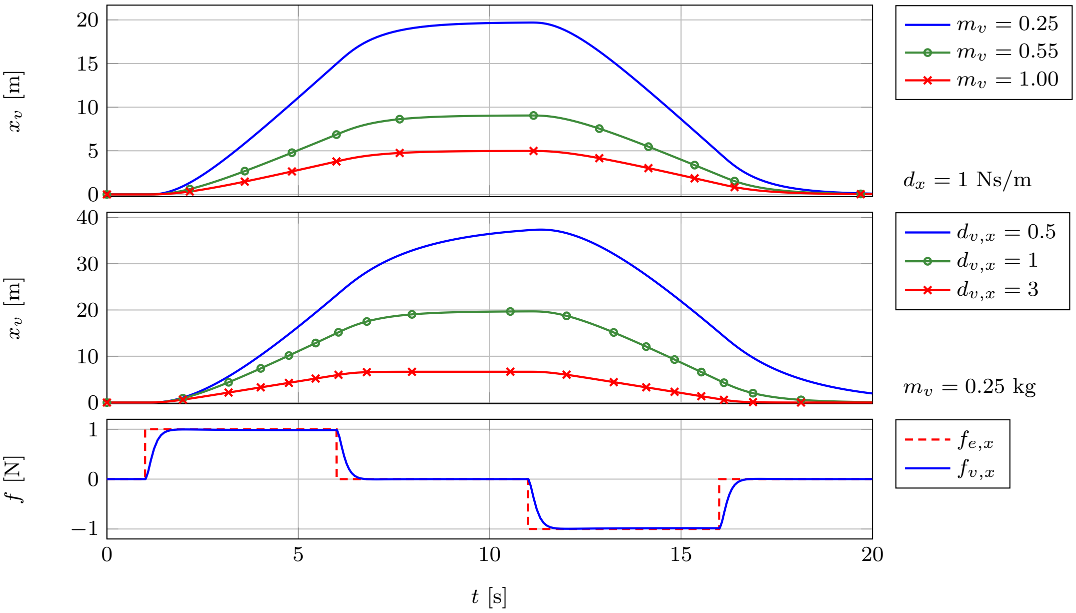


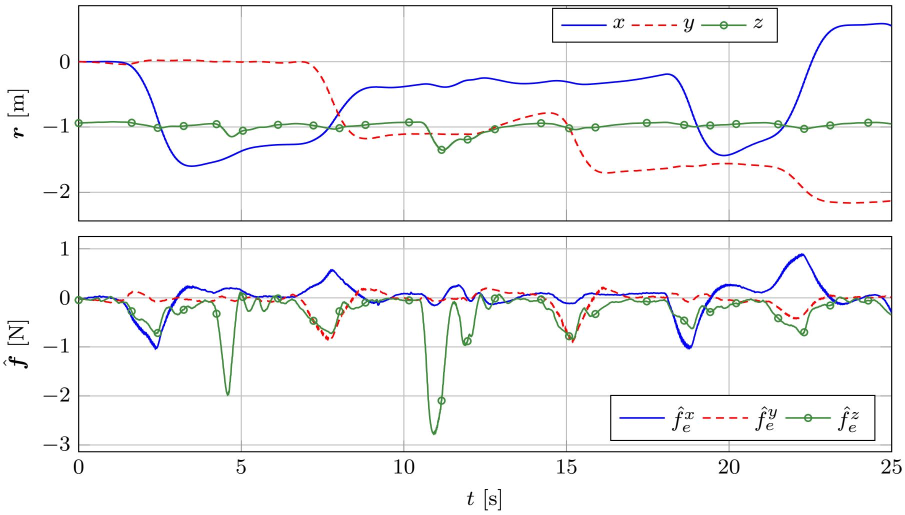
Figure 3.16: Admittance controller response. The parameters determine the steady-state velocity of the virtual system. The system is in damping mode — the virtual stiffness is $\boldsymbol{K}_v = \boldsymbol{0}$, and the real system mass is $m = 0.55$ kg. Copyright © 2017 IEEE [135, 138].

3.2.5 Experimental validation of impedance control

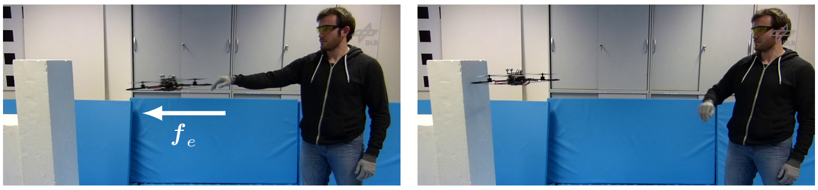
An impedance controller for the position subsystem as depicted Figure 3.12 was experimentally verified, using parameters shown in Table 3.1. The rotational inertia is unchanged, i.e. $\boldsymbol{\mathcal{I}}_v = \boldsymbol{\mathcal{I}}$. During the experiment, the controller was in damping mode in the inertial (x,y) plane, and a human applied a force to the robot. Figure 3.17 depicts the position response and estimated force during interaction. Due to the spring term, the altitude is controlled to 1 m, while the external force causes a proportional altitude error as expected. The robot may be freely moved in the horizontal axes by applying a force, as expected. The energy injected by a collision is dissipated by the impedance controller's damping term. A numerical analysis of the local input-to-state stability conditions for this experiment is given in Appendix 3.2.6.

3.2.6 ISS numerical example

The following is a numerical example of the local input-to-state stability conditions for the impedance control experiment shown in Figure 3.17, parameters from Table **??**, and $c_1 = 0.1$, $c_2 = 0.01$ taken numerically such that $\boldsymbol{P}$ and $\boldsymbol{Q}$ are positive definite. Because the



(a) Position response and estimated external force of the impedance controller during interaction without collisions.



(b) Video stills of an interaction experiment. The human pushes the robot, which causes a collision. The secondary collision can be recovered from safely because the interaction controller amplifies the collision force; as a result, the robot bounces away from the obstacle.

Figure 3.17: Trajectory and estimated external force in the impedance controller experiment with $m_v = 0.25$ kg, $\boldsymbol{D}_v = \mathrm{diag}\{1, 1, 1\}$ N/s, $\boldsymbol{K}_v = \mathrm{diag}\{0, 0, 2\}$ N/m, see Figure 3.17(a). The quadrotor can be freely moved along the inertial x and y axes, see Figure 3.17(b). Copyright © 2017 IEEE [135, 138].

$x-$ and $y-$ position stiffness is zero, these position coordinates are obviously unbounded. We therefore omit them from the state for this analysis. Note that the analysis includes the $x-$ and $y-$ translational velocities nonetheless. With $K_{i,m} = 30\boldsymbol{I}$, the following numerical values are obtained:

$$\boldsymbol{P} = \begin{bmatrix} 2 & 0.1 & \cdot & \cdot & \cdot & \cdot \\ 0.1 & 0.25 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0.1\,\boldsymbol{I} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & c_v\,\boldsymbol{I} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \boldsymbol{\mathcal{I}} & 0.01\,\boldsymbol{I} \\ \cdot & \cdot & \cdot & \cdot & 0.01\,\boldsymbol{I} & 30\,\boldsymbol{I} \end{bmatrix}, \tag{3.55}$$

$$\boldsymbol{Q} = \begin{bmatrix} 0.80 & 0.20 & 0.12 & 1.97 & 0 & 0 \\ 0.20 & 0.90 & 0.30 & 4.92 & 0 & 0 \\ 0.12 & 0.30 & 1.00 & 0 & 0 & 0 \\ 1.97 & 4.92 & 0 & 853.33 & 53.33 & 0 \\ 0 & 0 & 0 & 53.33 & 32 & 0 \\ 0 & 0 & 0 & -0 & -0 & 1 \end{bmatrix}. \tag{3.56}$$

Table 3.1: System and control parameters used in the impedance control experiments with the AscTec Hummingbird.

Parameter	Value	Unit
$\mathcal{M}$	0.55	kg
$\boldsymbol{K}_1$	diag{2.25, 2.25, 5.06}	s^{-2}
$\boldsymbol{D}_1$	diag{3, 3, 4.5}	s^{-1}
$\boldsymbol{\mathcal{I}}$	diag{3, 3, 4}	$10^{-3}\ \mathrm{kg\,m}^2$
c_v	256	-
$\boldsymbol{D}_2$	diag{32, 32, 32}	$\mathrm{rad}^{-1}\,\mathrm{s}^{-1}$

Their eigenvalues (with multiplicity in braces) are

$$\begin{aligned}\mathrm{eig}(\boldsymbol{P}) &= \big(256^{(3)}, 30^{(3)}, 2.006, 0.244, 0.1^{(3)}, 0.004, 0.003^{(2)}\big),\\ \mathrm{eig}(\boldsymbol{Q}) &= \big(856.81, 28.55, 1.32, 1, 0.75, 0.59\big),\end{aligned}$$

and are obviously positive, confirming the local input-to-state stability conditions from Proposition 3.

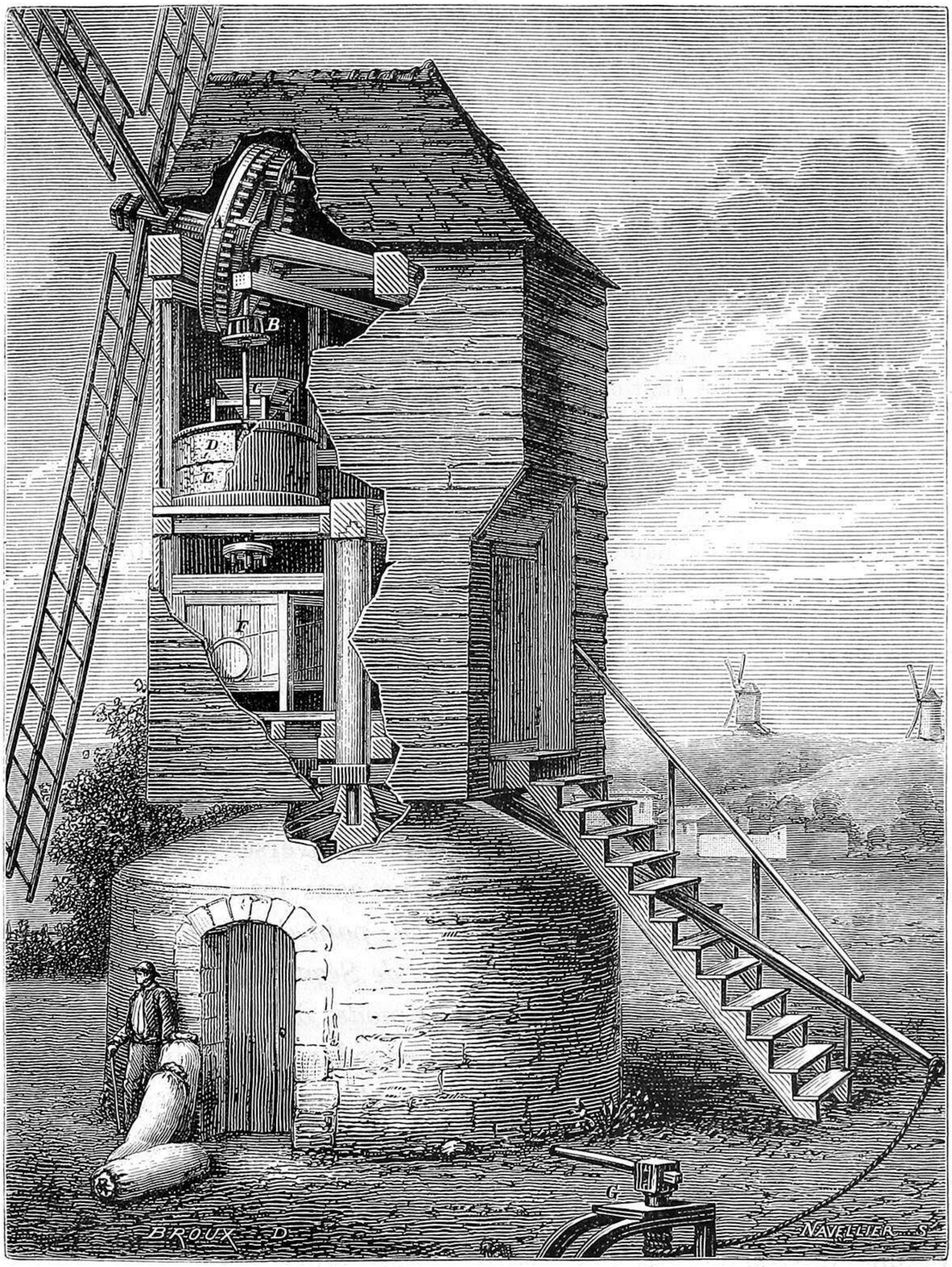


Illustration: Broux, P.; Book: Figuier, Louis: Les merveilles de l'industrie, vol. 4, Paris, Furne, Jouvet et Cie, 1877

CHAPTER 4

Wind estimation

In this chapter, the problem of estimating the wind speed from onboard measurements of a flying robot is investigated. The chapter is organized as follows. In Section 4.1, a novel method for inverting a nonlinear drag wrench model to obtain the wind speed is presented [136]. In order to obtain accurate aerodynamics models, a custom-built hexacopter was flown in a 3D wind tunnel. The experimental setup and resulting dataset are presented in Section 4.2. Next, the dataset is used to evaluate the model complexity and feasible measurements to directly obtain a good estimate of the airspeed, without the need of model inversion. It is shown that even simple models may be used to accurately obtain relative airspeed. It is also shown that the propeller aerodynamic power can be used to directly regress to the airspeed. Lastly, a novel physics model based wind estimation method is developed in Section 4.4. Starting from the momentum theory model of a propeller, it builds a nonlinear least squares problem to obtain the airspeed. Due to its sensitivity, a novel combined wind estimator is developed and evaluated. Finally, a method to obtain optimal measurements for this problem is formulated.

4.1 Inversion-based metric wind estimation

The external wrench and aerodynamic model (2.15) can be used to determine the surrounding wind velocity. It is assumed that the external wrench is purely due to aerodynamic forces, i.e. $\hat{\boldsymbol{\tau}}_e \approx \boldsymbol{\tau}_d$. The aerodynamic model may be iteratively linearized and inverted about the current estimate, as done in Waslander et al. [149]. By linearizing the nonlinear model (2.16) about $\boldsymbol{\nu}_r$ at step k, we obtain

$$\boldsymbol{\nu}_{r,k+1} = -\boldsymbol{D}_k^{-1}\boldsymbol{\tau}_d \tag{4.1}$$

where $\boldsymbol{D}_k = \boldsymbol{D}_L + \boldsymbol{D}_{i.d.} + \boldsymbol{D}_Q|\boldsymbol{\nu}_{r,k}|$. The relative velocity $\boldsymbol{\nu}_r$ is obtained in the body frame, and the wind velocity is then $\boldsymbol{\nu}_w = \boldsymbol{\nu} - \boldsymbol{\nu}_r$. At low velocities, the viscous (linear) effects are dominant. Therefore linearizing about the current velocity simplifies the problem [149].

Alternatively, using Newton-Raphson iteration, quadratic convergence may be achieved. The update is

$$\boldsymbol{\nu}_{r,k+1} = \boldsymbol{\nu}_{r,k} - \left(\frac{\partial \boldsymbol{D}_k}{\partial \boldsymbol{\nu}_{r,k}}\right)^{-1} \left(\boldsymbol{D}_k\boldsymbol{\nu}_{r,k} + \boldsymbol{\tau}_d\right), \tag{4.2}$$

where

$$\frac{\partial \boldsymbol{D}_k}{\partial \boldsymbol{\nu}_{r,k}} = \boldsymbol{D}_L + \boldsymbol{D}_{i.d.} + \operatorname{diag}\big\{|\boldsymbol{\nu}_{r,k}|\big\}\boldsymbol{D}_Q + \operatorname{diag}\big\{\boldsymbol{D}_Q|\boldsymbol{\nu}_{r,k}|\big\}. \tag{4.3}$$

Note that the matrix inversion is easy to compute if $\boldsymbol{D}_k$ is diagonal. Figure 4.1 shows

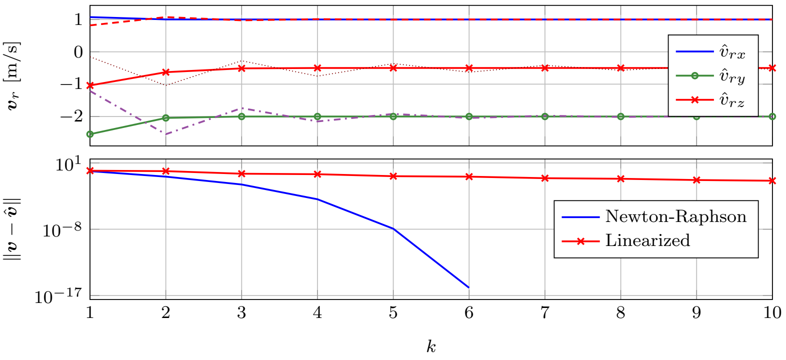


Figure 4.1: Convergence of the Newton-Raphson (4.2) (solid) and iterative linearization (4.1) (dashed) methods for obtaining the wind velocity from a force by model inversion, for $\boldsymbol{v}_r = [1, -2, -0.5]^T$ m/s with initial estimate $\boldsymbol{v}_{r,0} = [0, 0, 0]^T$ m/s. The Newton-Raphson method coverges to the exact value in 6 iterations. Sufficient relative or absolute tolerance may be reached in less. By initializing with the previous solution in an online implementation, one iteration is sufficient. Copyright © 2015 IEEE [136].

a comparison of the convergence of both methods for a simple simulation model. The Newton-Raphson method converges to the exact value in 6 iterations. However, in general there is no guarantee for real-time convergence. In an online implementation the result from the previous time step may be used as initialization. In that case, under the assumption that the result has not changed much, convergence is typically be obtained in one iteration.

In order for this method to work, the drag model has to be known, i.e. elements of the drag model matrices have to be identified. Furthermore, it has to be well-structured and invertible, in order for the inversion to converge. Alternatively, the airspeed can be fitted directly from flight data, for example as a function of the external wrench

$$\boldsymbol{\nu}_r := \boldsymbol{\nu}_r(\boldsymbol{\tau}_e), \tag{4.4}$$

where $\boldsymbol{\nu}_r(\boldsymbol{\tau}_e)$ is an nonlinear "inverse" drag model. This has the advantage of potentially being computationally cheaper than inverting a nonlinear drag model online. In the following, we investigated this approach by first collecting data in a wind tunnel and exploring nonlinear regression methods to fit aerodynamics models.

4.2 Wind tunnel experiments

Experiments to identify aerodynamic models were carried out at the Wind Engineering, Energy and Environment (WindEEE) Dome, see Hangan [58], in London, ON, Canada. It is the world's first 3D wind chamber, consisting of a hexagonal test area 25 m in diameter and an outer return dome 40 m in diameter. Mounted on the peripheral walls and on top of the test chamber are a total of 106 individually controlled fans and 202 louver systems. Additional subsystems, including an active boundary layer floor and "guillotine" allow for further manipulation of the flow. These are integrated via a sophisticated control system which allows dynamic manipulation with thousands of degrees of freedom to produce various time and spatially dependent flows including straight uniform, atmospheric boundary layer, shear gusts, downbursts and tornados at multiple scales. A pair of 5 m diameter turntables allow for a wide variety of objects to be tested inside and outside the facility.

For this project WindEEE was configured to produce straight flow closed-loop and downburst flows concurrently. In this configuration the test area was restricted to a 4.5 m diameter, 3.8 m tall region at the centre of the facility. See Figure 4.2 for a schematic

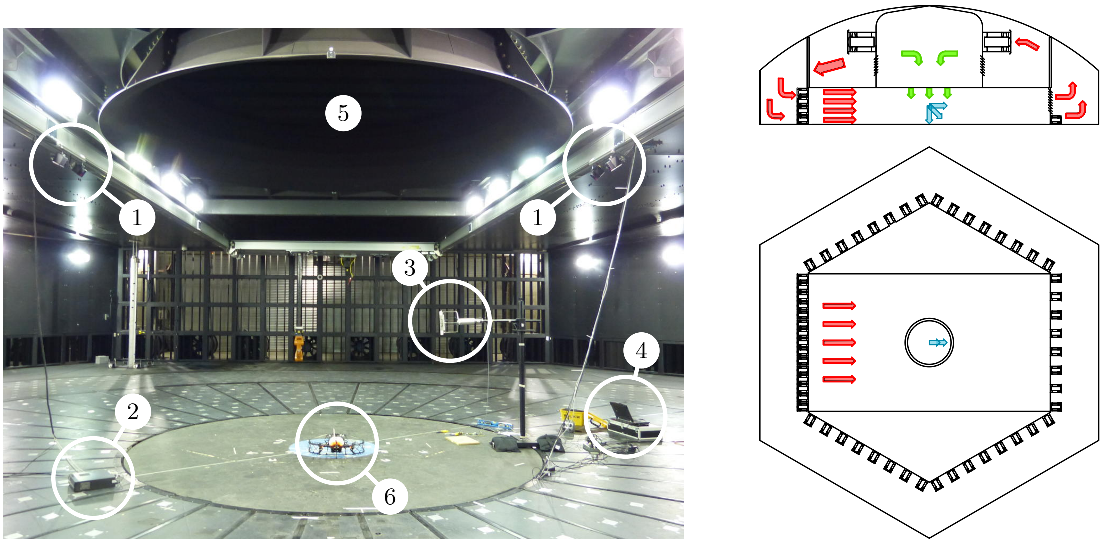


Figure 4.2: Left: experimental setup inside the wind tunnel: four ART Tracking cameras (1) and ART controller (2), RM Young Model 81000 Sonic Anemometer (3), groundstation laptop (4), vertical wind component inlet (5). The flying robot (6) is located in the center of the flying area. Right: schematic layout of the wind tunnel test. Red arrows show horizontal flow component, green arrows show vertical flow component, blue arrows show net wind vector. Copyright © 2016 IEEE [140].

drawing of the layout. A rectangular array of 36 fans (9 wide by 4 high) located on the south chamber wall were used to produce horizontal flow and 6 large fans above the test chamber were used to generate the downward flow. The respective flow rates from the horizontal and vertical component fans were manipulated individually to generate net wind vectors ranging in velocity from 1–5 m/s and vertical plane angularity from 0–90°. In some cases both the velocity and vertical plane angularity were manipulated dynamically to produce time-dependent wind vectors that either varied in speed or angularity over a given test run.

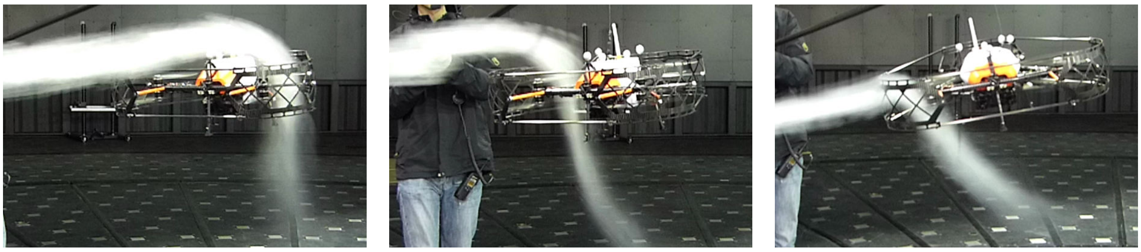
Motion capture noise. Before the experiments the effect of the wind tunnel on motion capture noise was investigated by increasing the horizontal speed to 6 m/s, and keeping the hexacopter stationary. It was found that the wind tunnel did not have a noticable effect on noise in the position and orientation measurements of the marker attached to the hexacopter.

Flow visualization. For illustrative purposes, flow visualization at various wind speeds is shown in Figure 4.3. In steady-state hover, each coaxial rotor pair has to provide a third of the total robot weight. Therefore, the induced velocity of a coaxial rotor pair at hover is

$$v_h = \sqrt{\frac{\frac{1}{3}\mathcal{M}g}{\frac{1}{2}\rho D^2 \pi}} = \sqrt{\frac{2 \cdot 2.445 \cdot 9.81}{3 \cdot 1.182 \cdot 0.254^2 \pi}} = 8.17 \text{ m/s},$$

where the air density in the wind tunnel was measured to be 1.182 kg/m^3, based on ambient temperature and pressure. From the flow visualization, deformation of the airflow becomes noticable at 8 m/s, or about one induced velocity at hover ($v_\infty/v_h \approx 1$). Note also that due to high v_h and limited range of tested airspeeds, the changes in the propeller aerodynamic power are expected to be small.

Dataset. For training aerodynamic models, the hexacopter was flown in horizontal, vertical and combined airflows with varying wind speeds. The robot was hovering in



(a) 3 m/s (b) 6 m/s (c) 8 m/s

Figure 4.3: Flow visualization in the wind tunnel at various horizontal wind speeds. Effects due to wind become visibly noticeable around 8 m/s. At lower speeds, propellers dominate the airflow.

position-controlled mode and was yawing during the experiments. The concatenated dataset, shown in Figure 4.4, contains over 37 minutes of flight data under horizontal wind speed up to 8 m/s, and vertical wind speeds up to 5 m/s. Individual flights are delimited by a dotted vertical line. Larger vertical wind speeds were not possible due to actuator saturation resulting in loss of yaw control authority. The relative airspeed is depicted in the body frame. We varied the yaw angle throughout the flights. The following data were logged: pose from the external tracking system and the onboard visual-inertial navigation system; IMU data (accelerometer, gyroscope); control input; motor speed, current, and voltage; anemometer data (wind velocity, direction, temperature, speed of sound). This is then used to compute the external wrench, relative airspeed, and aerodynamic power. The plot shows the sum of powers of the three coaxial motor pairs. The hover power was obtained from $P_h = 2\rho A v_h^3$, where v_h is obtained by (2.14). The expected aerodynamic power is P_a, with induced velocity calculated from (2.11) and (2.12), using the relative airspeed obtained from the anemometer data and external tracking system. Aerodynamic power is fitted as described in Section 4.3.1. Figure 4.4 also shows the individual motor speeds of two coaxial pairs. It can be seen that the motors were saturating for come flow conditions, limiting the maximum achievable wind speed.

Figure 4.5 depicts the distribution of relative airspeeds $\boldsymbol{v}_r$ achieved in the dataset. The shape is an inclined oblate spheroid. The implication of this shape is that our models will be valid only within this shape, i.e. the models will have to extrapolate (generalize) beyond the measured airspeeds. Most notably, the data does not include downward motion, which would lead to the vortex ring state, but is dominated by horizontal and upward relative airspeeds. Ideally, the dataset would have been a sphere, however only downward vertical wind speeds were possible in the wind tunnel. This dataset is next used to identify and evaluate aerodynamic models, and later used in Chapter 5 for simulation studies on force discrimination.

4.3 Aerodynamic model evaluation

In this section, the wind tunnel dataset is used to evaluate aerodynamic models for wind speed estimation. The evaluation is carried out for combinations of model types and input sets. Furthermore, quality of fit is investigated depending on regularization, as well as model generalization. These findings are then applied to models for simulation and force discrimination. Here, an attempt to answer the following questions is made:

- What is the minimal set of inputs required?

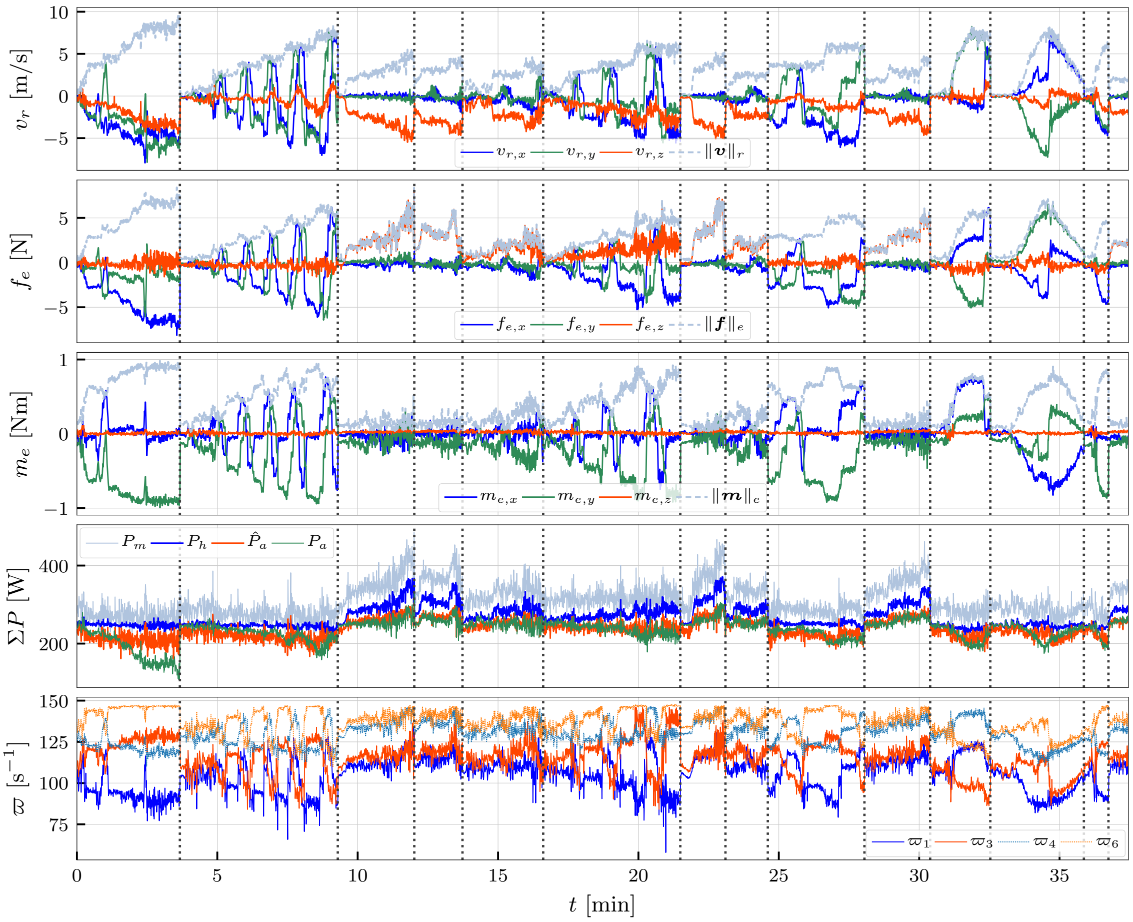


Figure 4.4: The training and validation set for aerodynamics models contains over 37 minutes of flight data under horizontal wind speed up to 8 m/s, and vertical wind speeds up to 5 m/s. Individual flights are delimited by a dotted vertical line. From top to bottom: relative airspeed in the body frame; external force in the body frame; external torque; sum of coaxial motor and rotor power, motor speeds of two coaxial pairs.

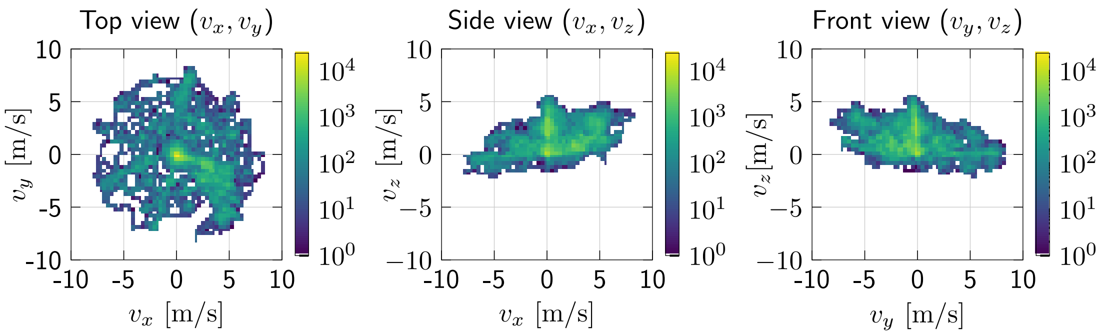


Figure 4.5: Flight data distribution. The intensity shows the number of data points collected at a certain relative airspeed. The data is represented here in 80 horizontal and vertical bins. Ideally, the training data would be a sphere. However, due to the limited space in the wind tunnel and no possibility to get wind from below, we were only able to cover a limited set of z-velocities. The data is shown here with the z-axis pointing upwards.

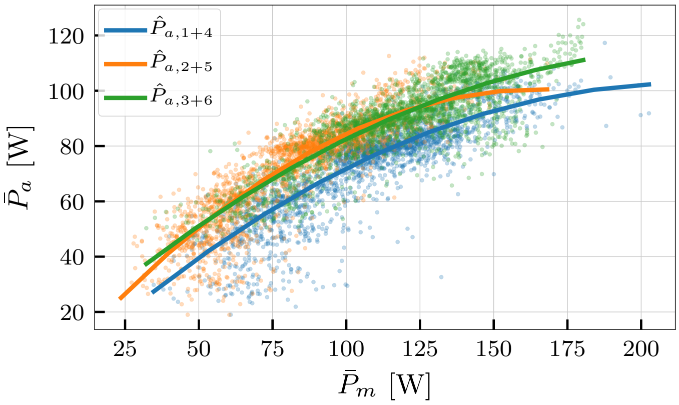


Figure 4.6: Fit of aerodynamic power for individual propeller pairs. The points represent the individual data points for each propeller pair, while the line shows a quadratic fit.

- Is is possible to estimate the airspeed using motor power measurements only?
- Which model/method works/generalizes best?

First, the mapping $\boldsymbol{v}_r(\boldsymbol{u})$ is investigated, as well as the effect of different inputs $\boldsymbol{u}$ on the model quality, i.e. the quality of the fit to existing data. Qualitative extrapolation properties of the models are also investigated. To estimate the relative airspeed $\boldsymbol{v}_r \in \mathbb{R}^3$, a combination of the following inputs was chosen:

- external force $\boldsymbol{f}_e \in \mathbb{R}^3$,
- external wrench $\boldsymbol{\tau}_e \in \mathbb{R}^6$,
- individual propeller aerodynamic power $\boldsymbol{P}_a \in \mathbb{R}^6$, $\boldsymbol{P}_h \in \mathbb{R}^6$,
- coaxial propeller aerodynamic power $\bar{\boldsymbol{P}}_a \in \mathbb{R}^3$, $\bar{\boldsymbol{P}}_h \in \mathbb{R}^3$,
- propeller rotational speed $\boldsymbol{\varpi} \in \mathbb{R}^6$.

Note that e.g. $\boldsymbol{P}_h$ is a nonlinear map of $\boldsymbol{\varpi}$ and is therefore redundant to that input. Equation (2.14) was used to obtain the aerodynamic power of the coaxial pairs $\bar{\boldsymbol{P}}_a \in \mathbb{R}^3$ and $\bar{\boldsymbol{P}}_h \in \mathbb{R}^3$ by treating each pair as a single propeller. Because of the complex shape of our robot, we hypothesize that simple models (e.g. blade flapping based) are not be expressive enough to predict the airspeed. Models with increasing complexity are therefore investigated to provide insights for other researchers. Lastly, all forces and airspeeds are treated in the body frame. This implicitly captures aerodynamic angles in the vector components, as they are projected to the body frame.

4.3.1 Propeller aerodynamic power

Using individual motor powers for coaxial propellers leads physically meaningless results, where the figure of merit of the lower motors can be above 1, whereas it should be around 0.5–0.6 (see [78]). In order to use the propeller aerodynamics model presented in Section 2.1.3, the motor power measurements (2.22) must be fit to the aerodynamic power (2.12). Figure 4.6 shows the fit of each propeller pair data to the quadratic model

$$\bar{P}_{a,j+k} = P_0 + \beta_1 \bar{P}_{m,j+k} + \beta_2 \bar{P}_{m,j+k}^2, \tag{4.5}$$

where the mechanical power $\bar{P}_m$ of each coaxial pair $j+k$ is the sum of individual motors' mechanical power, i.e.

$$\bar{P}_{m,j+k} := \hat{P}_{m,j} + \hat{P}_{m,k} \tag{4.6}$$

The mechanical power of motor i without the rotor acceleration term is obtained by

$$\hat{P}_{m,i} = \Big(\big(K_{q,0i} - K_{q,1i} i_{a,i} \big) i_{a,i} - I_r \dot{\varpi}_i \Big) \varpi_i. \tag{4.7}$$

The model fit uses IRLS and shows good agreement with the data. Helicopter aerodynamics literature (e.g. Leishman [78]) proposes a linear fit from shaft power to aerodynamic power, in the form of a figure of merit FM. However, only the current can be measured, and the shaft torque cannot be measured directly. The quadratic model therefore also captures losses in the conversion of electrical to mechanical power. Now the motor power can be related to momentum theory, and as such may be used in further analyses.

Momentum theory may be used to predict the ratio of aerodynamic power in forward flight to the aerodynamic power in flight. This is visualized for our dataset in Figure 4.7, for the motor power, and the fitted aerodynamic power. Note that momentum theory predicts a decline in power at oblique angles of attack (high horizontal and vertical airspeed components). However, we could not observe this effect during the combined airflow experiment (67° inflow angle), which is the first flight in Figure 4.4. Notably, the motor power saturates on the low side. It can be postulated that that this may be due to motor speed saturations from compensating the simultaneously large external force and torque during the flight. This produces more losses in the motor, and information about the wind speed cannot be deduced. The fit is better in the remainder of the flights, where motors do not saturate for large parts of the flight. Based on this, it can be concluded that any method that uses motor power as input will not be valid in periods of motor input saturation.

4.3.2 Aerodynamic models

Next, various models used in the evaluation are presented. The problem considered here is regressing from a set of inputs $\boldsymbol{u}$ to the vector of body-frame relative airspeed $\boldsymbol{v}_r$ (three velocity components). Data fitting was done using a 67%–33% train–test split of the complete wind tunnel dataset. For efficient training the inputs and outputs were normalized to a range of $[0,1]$. The Python package scikit-learn [112] was used for its machine learning functions.

Physics-based models

Physics-based models are obtained from first-principles modeling, such as conservation of momentum. As such, they provide insight into the structure of the problem. However, some effects are usually neglected due to assumptions contained therein.

Blade flapping. It is well established in multirotor literature that the dominant horizontal force in multicopters is due to propeller induced drag [149, 107]. The widely used induced-drag and blade flapping model [107] can be written as

$$\boldsymbol{f}_d(\boldsymbol{v}_r) = \boldsymbol{D}_l \; \boldsymbol{v}_r \sum_i \varpi_i, \tag{4.8}$$

where $\boldsymbol{D}_l$ is the matrix of coefficients, and ϖ_i is the speed of the i–th propeller. The reader is referred to the cited literature for the derivation of the model.

Parasitic drag. The blade flapping model may be extended with a quadratic parasitic

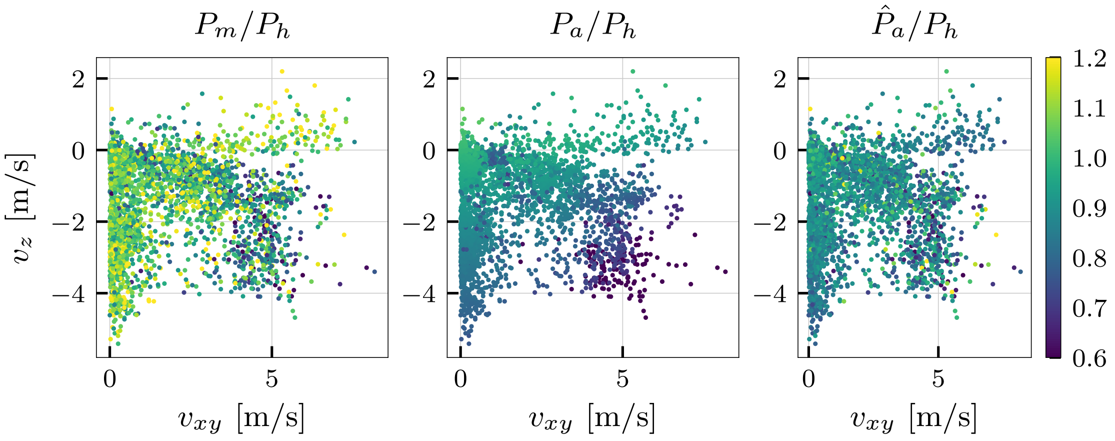


Figure 4.7: Power ratio at different airspeeds, relative to aerodynamic power in hover. From left to right: motor power, aerodynamic power, fitted aerodynamic power.

term due to form drag such that

$$\boldsymbol{f}_d(\boldsymbol{v}_r) = \boldsymbol{D}_l\, \boldsymbol{v}_r \sum \varpi_i + \boldsymbol{D}_q\, \boldsymbol{v}_r |\boldsymbol{v}_r|, \tag{4.9}$$

where $\boldsymbol{D}_q$ is the parasitic drag matrix. These models are used as a starting point to design more general regression models.

Linear regression models

Physics-based models are considered in the general framework of linear regression. The linear regression model can be written as

$$\boldsymbol{y}(\boldsymbol{u}) = \boldsymbol{X}(\boldsymbol{u})\, \boldsymbol{w}, \tag{4.10}$$

where $\boldsymbol{y} \in \mathbb{R}^{N_y}$ is the model output, $\boldsymbol{u} \in \mathbb{R}^{N_u}$ is the input, $\boldsymbol{X}(\boldsymbol{u}) \in \mathbb{R}^{N_y \times N_w}$ is the regression matrix, and $\boldsymbol{w} \in \mathbb{R}^{N_w}$ are the model weights. The weights are obtained by solving the minimization problem

$$\min_{\boldsymbol{w}} \frac{1}{2N} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \alpha_1 \|\boldsymbol{w}\|_1,$$

where N is the number of samples, α_1 is the regularization factor, and $\|\boldsymbol{w}\|_1$ is the ℓ_1 norm of the model weights. This will drive some of the model weights to zero, leading to sparse models depending on α_1.

To apply this model to physics-based models, $\boldsymbol{X}$ contains the linear and quadratic terms of the input, while $\boldsymbol{w}$ contains the matrix elements. In this evaluation, the linear model and quadratic model are considered. The *linear model* can be written as

$$\boldsymbol{y}(\boldsymbol{u}) = \boldsymbol{W}\boldsymbol{u}, \tag{4.11}$$

where $\boldsymbol{W}$ is the matrix of weights, and $\boldsymbol{u}$ is the input vector. This gives a maximum of $N_u \times N_y$ nonzero parameters. Similarly, the *quadratic model* can be written as

$$\boldsymbol{y}(\boldsymbol{u}) = \boldsymbol{W}_1\boldsymbol{u} + \boldsymbol{W}_2\boldsymbol{u}|\boldsymbol{u}|, \tag{4.12}$$

where $\boldsymbol{W}_1$ is the matrix of linear weights, $\boldsymbol{W}_2$ is the matrix of quadratic weights, This gives a maximum of $2 \times N_u \times N_y$ nonzero parameters.

Multilayer perceptron model

To compare a purely data-driven approach to physically based modeling, nonlinear regression is applied to the data. For this purpose a multilayer perceptron with one hidden layer is used, with the tanh activation function and a linear output layer. Experimentation on the available data has found that using more layers did not improve the fit, and using other activation functions did not change the results significantly. The model can be summarized as

$$\boldsymbol{y}(\boldsymbol{u}) = \boldsymbol{W}_2\Big(\sigma\big(\boldsymbol{W}_1\boldsymbol{u} + \boldsymbol{b}_1\big) + \boldsymbol{b}_2\Big) \tag{4.13}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{N_h \times N_u}$, $\boldsymbol{W}_2 \in \mathbb{R}^{N_y \times N_h}$ are the neuron weight matrices, $\boldsymbol{b}_1 \in \mathbb{R}^{N_h}$ and $\boldsymbol{b}_2 \in \mathbb{R}^{N_y}$ are the bias vectors, and $\sigma(\cdot) = \tanh(\cdot)$ is the activation function. This gives a maximum of $(N_u + N_y + 1) \times N_h + N_y$ nonzero parameters. For the perceptron, ℓ_2 regularization of the weights is used, with α_2 as regularization parameter. Training is performed by stochastic gradient descent [112].

4.3.3 Model performance

Figure 4.8 shows error histograms of the test data set for combinations of models and inputs as listed in Tables 4.1 and 4.2, respectively. Simple blade flapping based models have a problem predicting the body-vertical relative airspeed. The plot therefore contains histograms of individual relative airspeed components. From the results in Figure 4.8, the following conclusions can be drawn:

(1) Even the linear model (column 1) obtains good results for all input sets, for low regularization. As the model becomes sparser (i.e. contains fewer nonzero parameters) with increased regularization (column 2), it cannot predict the airspeed accurately anymore.

(2) Comparing the wrench input (row C) to the force input (row B), it can be seen that the external torque does not add significant information to the external force. In fact, the histograms are almost identical.

(3) Multiplying the wrench by the inverse sum of rotor speeds $\sum_i \varpi_i$ improves the results, see linear and quadratic models with higher regularization (B2, B4, C2, C4). This indicates that this is a more accurate representation of the underlying physics (blade flapping and

Table 4.1: Summary of tested regression models.

Model	**Formulation**
Linear model	$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{u}$
Quadratic model	$\boldsymbol{y} = \boldsymbol{W}_1\boldsymbol{u} + \boldsymbol{W}_2\boldsymbol{u}\lvert\boldsymbol{u}\rvert$
Perceptron	$\boldsymbol{y} = \boldsymbol{W}_2\Big(\tanh\big(\boldsymbol{W}_1\boldsymbol{u} + \boldsymbol{b}_1\big) + \boldsymbol{b}_2\Big)$

Table 4.2: Summary of tested inputs for predicting relative airspeed.

ID	**Label**	**Input data**	N_u
A	External force	$\boldsymbol{u}_A = \boldsymbol{f}_e$	3
B	Rotor speed normalized $\boldsymbol{f}_e$	$\boldsymbol{u}_B = \boldsymbol{f}_e / \sum_i \varpi_i$	3
C	Rotor speed normalized $\boldsymbol{\tau}_e$	$\boldsymbol{u}_C = \boldsymbol{\tau}_e / \sum_i \varpi_i$	6
D	Aerodynamic power	$\boldsymbol{u}_D = \hat{\boldsymbol{P}}_a$	6
E	Aerodynamic and hover power	$\boldsymbol{u}_E = \big[\hat{\boldsymbol{P}}_a^T \boldsymbol{P}_h^T\big]^T$	12
F	Coaxial aero. power	$\boldsymbol{u}_F = \bar{\boldsymbol{P}}_a$	3
G	Coaxial aero. and hover power	$\boldsymbol{u}_G = \big[\bar{\boldsymbol{P}}_a^T \bar{\boldsymbol{P}}_h^T\big]^T$	6

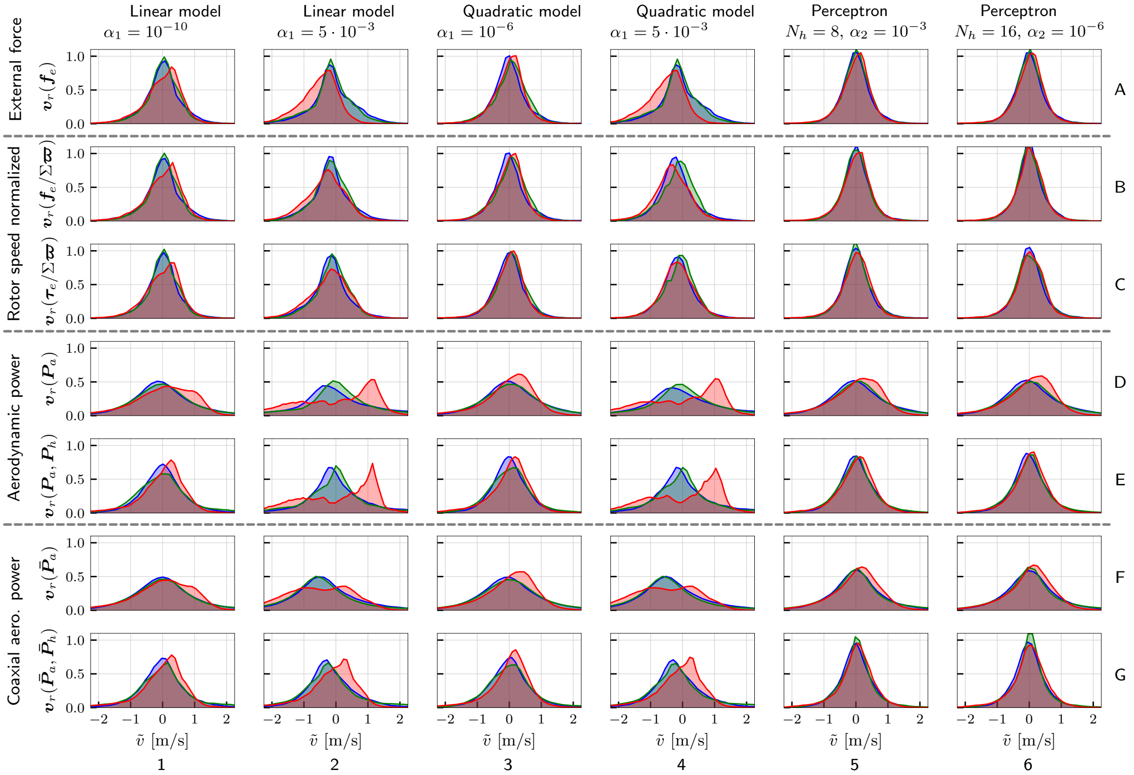


Figure 4.8: Normed error histograms of predicted airspeed as a function of different input sets, for different models. The histograms for x (blue), y (green) and z (red) airspeed are overlaid to better depict per-axis model performance. The letters on the right (rows) and numbers on the bottom (columns) define a grid for easier reference in the text.

induced drag). However, it does not significantly improve the nonlinear perceptron models, nor models with low regularization.

(4) The best regression results were achived by perceptron with $N_h = 16$ hidden neurons for the $\boldsymbol{f}_e / \sum_i \varpi_i$ input (B6). The difference is most notable in the z-axis, where the offset present in most histograms is absent. This may be used as a benchmark of the best possible regression for this dataset, and may compare simpler models to this one. In comparison, the preceptron with $N_h = 8$ hidden neurons (column 5) is almost as good with a lot fewer parameters.

(5) Increasing the perceptron model complexity does not significantly improve the airspeed regression after just several neurons in the hidden layer. This indicates that the underlying model has low complexity. Similarly, it was found that multiple hidden layers did not improve the regression (not shown in the histogram).

(6) The relative airspeed may be predicted by using only the aerodynamic power as input. Adding the aerodynamic power in hover (which is directly related to the control input, i.e. motor speeds) improves the fit. This is due to the fact that in steady state, the control input is equal to the external wrench, $\boldsymbol{\tau} = \boldsymbol{\tau}_e$. The aerodynamic power in hover P_h is computed from the motor speeds $\boldsymbol{\varpi}$, which relate to the control input through the control allocation matrix $\boldsymbol{B}$ as $\boldsymbol{\varpi} = \boldsymbol{B}^{+}\boldsymbol{\tau}$. Therefore, the aerodynamic power in hover is a function of the external wrench, i.e. $\boldsymbol{P}_h := \boldsymbol{P}_h(\boldsymbol{\tau}_e)$. This result must therefore be taken with caution, as the model essentially learns this nonlinear transformation. Our dataset does *not* include external forces other than wind, and the robot was hovering in place. The desired result of using this regression as an independent measurement of the airspeed is

therefore not attained.

(7) The offset in the z-velocity for power inputs indicates difficulty in modeling the aerodynamic power from motor power measurements (without the hover power).

(8) Aerodynamic power of coaxial propeller pairs is a more accurate representation than considering each propeller individually, as it is a better representation of underlying physics.

In summary, two of the questions posed at the beginning of this section may be answered:

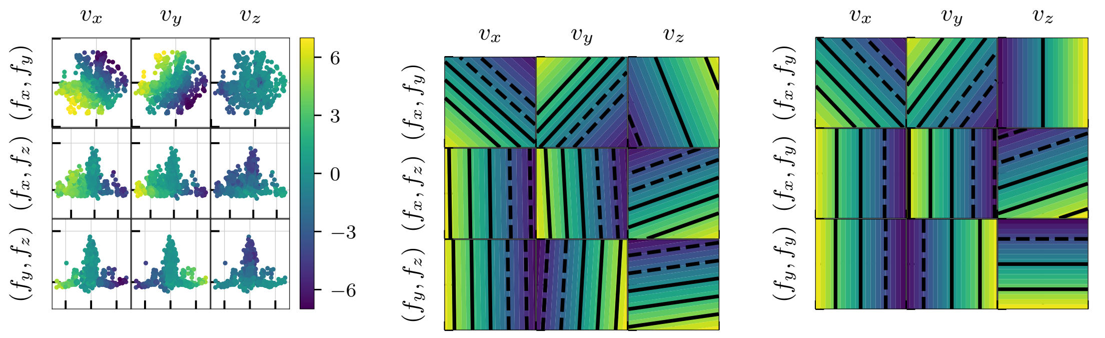
1. The minimum set of inputs is the external force $\boldsymbol{f}_e$.
2. It is possible to estimate the airspeed using motor power measurements. However, this will be noisier than when using the external force.

The last question about model quality (i.e. model selection) requires more investigation. To answer, the influence of regularization on the prediction quality of the models is investigated next.

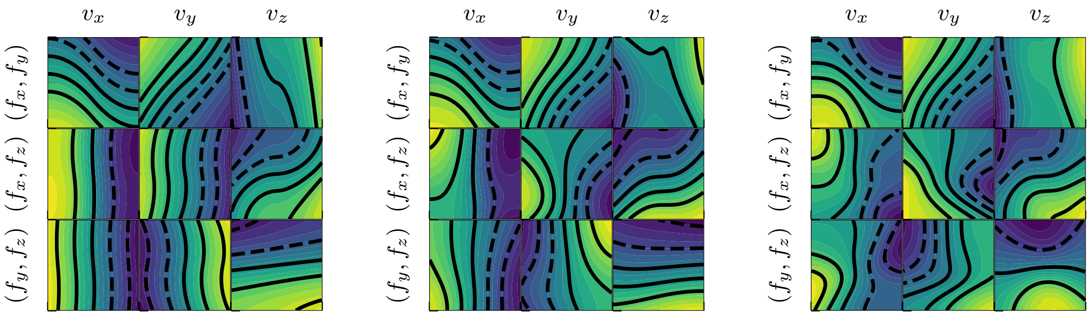
4.3.4 Model generalization

In order to reason about model fit and generalization, the dataset is visualized in Figure 4.9(a). Therein, the mapping from external force (input) to relative airspeed (output) is considered. Rows show three planar sections of the three-dimensional input force samples in the dataset. The upper-left plot (f_x,f_y) shows the x,$y-$plane, where f_z is close to zero. From top to bottom the data points are $(f_x, f_y, 0)$, $(f_x, 0, f_z)$, and $(0, f_y, f_z)$. In each plot, the color visualizes the output intensity – lighter colors indicate higher airspeed values. The columns represent different views of the model outputs, i.e. the relative airspeeds v_x, v_y, and v_z, respectively. Figure 4.9(b) and Figure 4.9(f) depict how the dataset was fitted by different models. Every contour line in the plots show a 3 m/s increase in airspeed (dashed lines for negative values). Figure 4.9(a) depicts the distribution of the training points, simplifying visually reasoning about the range of validity of the fitted models. As expected from the shape of training airspeeds in Figure 4.4, horizontally the input is a circle. However, the total shape is conic, because the number of combined airflow datapoints with substantial horizontal airspeed is limited. Clearly, fitted models will extrapolate in the regions where airspeed data has not been sampled.

The difference in expressivity between a linear and a perceptron model can also be seen from this visualization. Obviously, the perceptron model is more expressive. However, given the distribution of the training dataset, it is prone to overfitting to the training data. Outside of the training data distribution, the output may be physically meaningless. Therefore, a regularization parameter must be found for each model that both fits the training distribution, and generalizes outside of it. To furthter visualize the change in model outputs, Figure 4.10 shows the shape of $v_{r,z}$ for a quadratic model and a perceptron with 8 hidden neurons, with varying regularization values. Obviously, both models converge to a linear fit as regularization increases. However, the visualization alone does not tell us which model performs best. The following procedure is adopted to choose a model regularization parameter for good generalization. First, the complete dataset is split into training (70%) and validation (30%) datasets. The data points are therefore drawn from the same distribution. However, training never sees data from the validation set (holdout). K-fold cross-validation is then performed. The model is trained on a fraction of the training data, and a validation score is obtained on the remaining data. The procedure is repeated K times to get a mean and standard deviation of the training error and training loss (which includes regularization). Finally, in order to obtain the validation error, a model is trained on the complete training dataset. The validation error is computed between



(a) Dataset distribution (b) Linear model (c) Quadratic model, $\alpha = 10^{-2}$



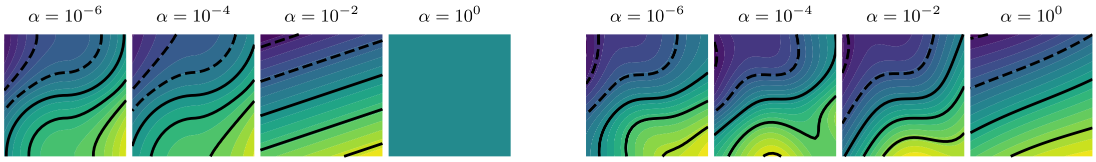
(d) Quadratic model, $\alpha = 10^{-4}$ (e) Perceptron model with 8 hidden tanh neurons, $\alpha = 10^{-2}$ (f) Perceptron model with 8 hidden tanh neurons, $\alpha = 10^{-6}$

Figure 4.9: Distribution of external force samples used for training and validation of the mapping *force (input) to relative airspeed (output)*. The color indicates the airspeed component v_x, v_y or v_z per column, brighter shades depict larger values. The distribution shows where learned models will be valid. Each row shows a different plane of the forces. The first row shows (f_x, f_y) for $|f_z| < 0.1$ N, the second (f_x, f_z) for $|f_y| < 0.1$ N, and the third one (f_y, f_z) for $f_x < 0.1$ N.

the model-predicted airspeed and the ground truth airspeed in the holdout set. To reason about generalization, the training loss will show when the regularization term in the cost function starts to dominate the loss, i.e. the model starts becoming linear.

Figure 4.11 shows the training loss, cross-validation training mean squared error (MSE) with standard deviation, and validation MSE for different models mapping external force to airspeed. The following can be concluded. First, the underlying structure of the data is simple. This is indicated by the fact that increasing the number of neurons in the perceptron's hidden layer does not significantly improve the fit. Second, both training and validation errors increase with regularization, indicating that overfitting is unlikely. This trend could also be caused by insufficient model complexity (expressiveness). However, given the simple structure of the data, this is unlikely to be the case. Lastly, for this particular model input, the quadratic model is only slightly better than the linear model, as they start converging around $\alpha_1 = 10^{-3}$.

Next, different inputs to predict airspeed are compared. Figure 4.12 shows a comparison for a perceptron with 8 hidden neurons, and Figure 4.13 depicts a comparison for a quadratic model. The results show that for the external force input, i.e. model $\boldsymbol{v}_r(\boldsymbol{f}_e)$, the difference between the models is very small. Therefore, it is recommended to use the simpler quadratic model over a perceptron. Second, the effect of incorporating the sum



(a) Quadratic model prediction of $v_{r,z}$ for $(0, f_y, f_z)$.

(b) Perceptron with 8 hidden neurons prediction of $v_{r,z}$ for $(0, f_y, f_z)$.

Figure 4.10: Change of model shape with increasing regularization.

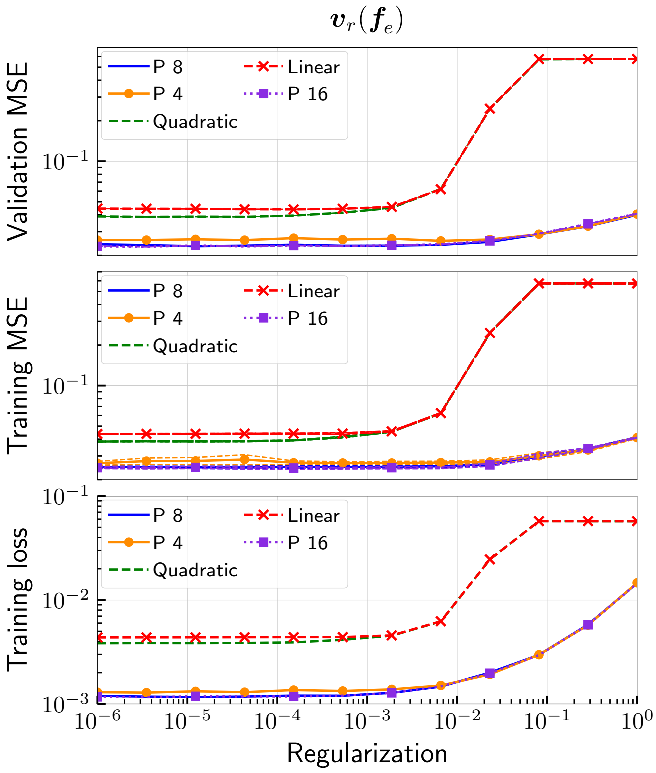


Figure 4.11: Training and validation mean square errors for 10-fold cross-validation of different models fitting $\boldsymbol{v}_r(\boldsymbol{f}_e)$, with varying regularization factors. Perceptron models with 4, 8, and 16 neurons in the hidden layer are labeled as P 4, P 8, and P 16, respectively.

of rotor rates into the input has been compared, motivated by the blade flapping model [107]. For modeling airspeed as a function of force, it is expected that a division by the rotor rates is necessary, due to the structure of this physical model. The results in Figure 4.12 and Figure 4.13 confirm this. Namely, multiplying the external force $\boldsymbol{f}_e$ by the sum of rotor rates $\sum \boldsymbol{\varpi}$ increases the MSE over just using the external force. Converesely, using $\boldsymbol{f}_e / \sum \boldsymbol{\varpi}$ as input does not have this effect. The MSE for this model does not increase as fast with regularization, however this may be attributed to scaling in the inputs.

Lastly, the use of coaxial aerodynamic power $\bar{\boldsymbol{P}}_a$ to estimate the airspeed was investigated. The MSE is higher than when using the external force. This can be attributed to the measurement noise of the motor current, i.e. motor power, which will limit the model performance. The training and validation MSE show the same behavior as with the external force models, however somewhat flatter for this particular range of the regularization parameter. This also indicates a slight scaling issue in the input. Here, the perceptron

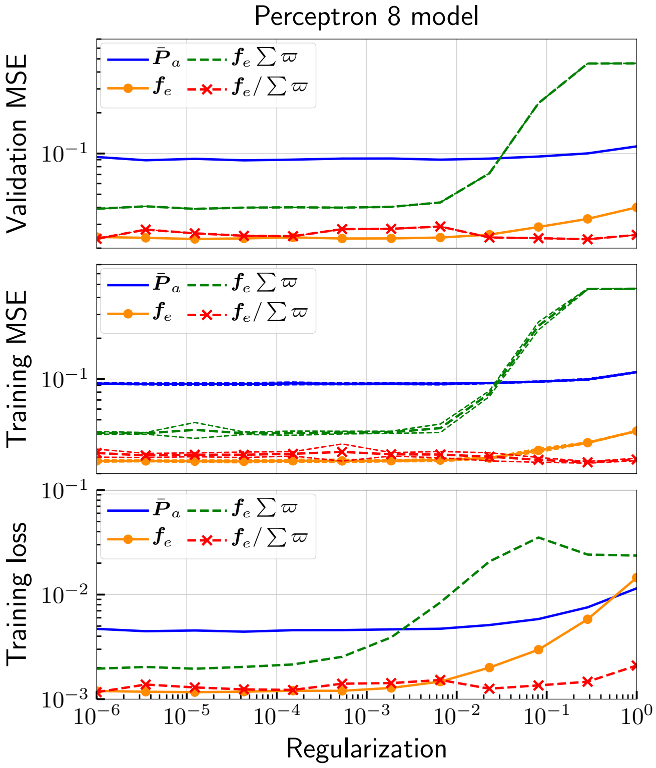


Figure 4.12: Training and validation mean square errors for 10-fold cross-validation of a Perceptron model with $N_h = 8$ fitting $\boldsymbol{v}_r(\boldsymbol{u})$, with varying $\boldsymbol{u}$ and regularization factors

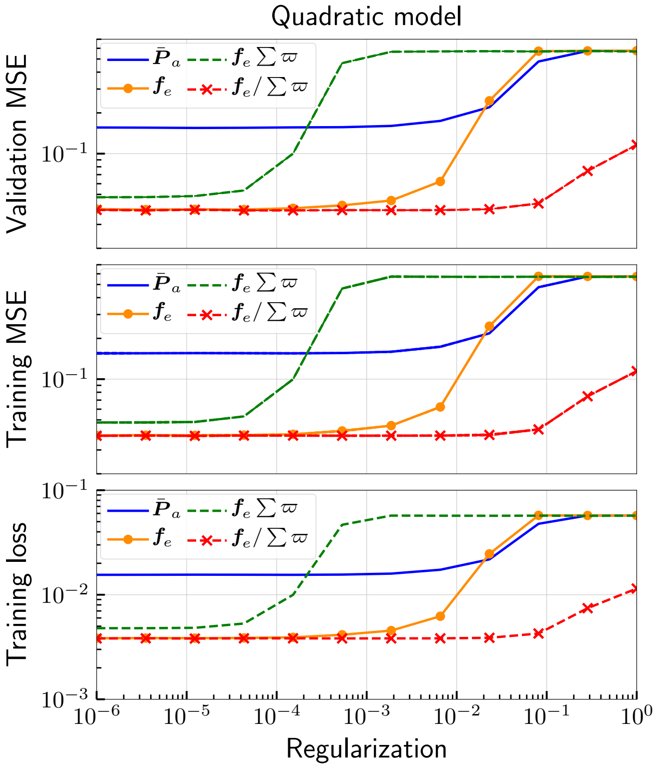


Figure 4.13: Training and validation mean square errors for 10-fold cross-validation of a quadratic model fitting $\boldsymbol{v}_r(\boldsymbol{u})$, with varying $\boldsymbol{u}$ and regularization factors

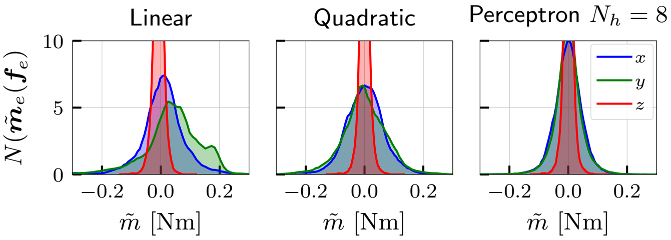


Figure 4.14: Error histograms for models predicting the aerodynamic torque as a function of the aerodynamic force $\boldsymbol{m}_d(\boldsymbol{f}_d)$, for the validation dataset.

model (validation MSE $\approx$ 0.009 m/s) performs almost twice as good as the quadratic model (validation MSE $\approx$ 0.016 m/s), since it can fit an underlying nonlinear mapping. When using aerodynamic power to estimate airspeed, therefore using a perceptron model is recommended.

To summarize, in this subsection K-fold cross-validation was used to analyze the effect of regularization on model generalization. It was concluded that the model training was set up properly, and that the models are not overfitting to the data. When using the external force as input, the underlying structure is simple, so a simple quadratic model is sufficient to represent it. When using aerodynamic power as input, the increased expressivity of perceptron models tends itself better to model the underlying nonlinear physics. It is postulated that current sensing with smaller measurement noise would improve the performance of the latter model, however this cannot be confirmed using the available data.

4.3.5 Aerodynamic torque models

In Section 4.3.3, using the complete external wrench as input did not significantly improve the fit when compared to only using the force. This indicated that there is a functional relationship between the aerodynamic torque and force. Physically, this can be understood as the aerodynamic force acting at the center of pressure. In other words, the torque and force are related by the point of action of the aerodynamic force.

Modeling the aerodynamic torque as a function of the force was therefore investigated. Figure 4.14 shows the error histogram for a linear model $\boldsymbol{m}_d := \boldsymbol{D}\boldsymbol{f}_d$, a quadratic model $\boldsymbol{m}_d := \boldsymbol{D}_l\boldsymbol{f}_d + \boldsymbol{D}_q|\boldsymbol{f}_d|\boldsymbol{f}_d$, and a perceptron with 8 hidden neurons. Because the geometric shape of the used robot is asymmetrical, it is expected that the perceptron model performs best. The results indicate that even the linear model can describe the relationship for our dataset, albeit with a bimodal error distribution for the y-axis. The quadratic model gets rid of this offset and performs significantly better. Finally, he perceptron shows the best performance, as it can capture the orientation-dependent relationship. To find the best model, cross-validation should be performed. In the interest of brevity, this step is skipped for torque models. Note that for other flying robots with simpler geometry, even a linear model could describe the relationship with good accuracy. These models are the basis of force discrimination methods in Chapter 5.

4.4 Physics model based wind estimation

In this section, a novel method to obtain wind velocity from aerodynamic power measurements based on momentum theory is presented. The aim is is to provide a measurement that is independent of the external force, in order to discriminate between aerodynamic and physical interaction forces. First, the aerodynamics of one propeller (2.11), (2.12) and (2.13) are rewritten as a system of nonlinear equations $\boldsymbol{F}(v_i, v_z, v_{xy}, v_h, P_a) = \boldsymbol{0}$, with $\boldsymbol{F} = [F_1, F_2, F_3]^T$, where

$$\begin{aligned} F_1 &= v_i^4 - 2v_i^3 v_z + v_i^2(v_z^2 + v_{xy}^2) - v_h^4 = 0, \\ F_2 &= v_i U(v_i - v_z) - P_a/(2\rho A) = 0, \\ F_3 &= v_h^2(v_i - v_z) - P_a/(2\rho A) = 0. \end{aligned} \tag{4.14}$$

Here, $P_a/(2\rho A)$ and v_h are considered to be known inputs, and want to determine $\boldsymbol{x} = [v_x,\, v_y,\, v_z,\, v_i]^T$. This system of nonlinear equations is underdetermined, as it has two knowns and three unknowns, since v_x and v_y are coupled in v_{xy}. Due to this mapping, the solution of (4.14) will be a manifold, and depends on the initial guess. Hence, (4.14) cannot be used to uniquely determine the unknowns. To solve this problem, the system of equations is expanded to include multiple measurements. A transformation of (4.14) into a common frame is then introduced. This allows us to estimate all three wind velocity components and the propeller induced velocities by solving a nonlinear least squares (NLS) problem.

Multiple measurements. Let us assume a constant wind velocity $\boldsymbol{v}_w = [v_x, v_y,\, v_z]^T$ through N measurements. This assumption holds in several cases. First, instantaneous measurements from multiple propellers that are rigidly attached (e.g. quadcopter) can be combined. These may also be rotated w. r. t. the body frame. Second, measurements from multiple poses at different time instants in a small time window can be combined. Third, if the flight is not aggressive, i. e. the orientation does not change significantly, the body-frame freestream velocity may be estimated. In effect, information gained from N measurements may be used to obtain the wind velocity components.

The state may be extended to N measurements

$$\boldsymbol{x}|_N = [v_x,\, v_y,\, v_z,\, v_{i,2},\, v_{i,2},\, \dots\, v_{i,N}]^T, \tag{4.15}$$

and solve the extended system of equations

$$\begin{aligned} &\boldsymbol{F}|_N(v_x, v_y, v_z, v_{i,1}, v_{h,1}, P_{a,1}, \dots, v_{i,N}, v_{h,N}, P_{a,N}) = \boldsymbol{0}, \\ &\boldsymbol{F}|_N = [F_{1,1}, F_{2,1}, F_{3,1},\, \dots,\, F_{1,N},\, F_{2,N},\, F_{3,N}]^T, \end{aligned} \tag{4.16}$$

where $F_{1,k}$, $F_{2,k}$ and $F_{3,k}$ are evaluations of (4.14) for the k-th measurement. A Jacobian is needed to solve (4.16). The Jacobian for the k-th measurement is defined as

$$\boldsymbol{J}_k = \begin{bmatrix} J_{11,k} & J_{12,k} & J_{13,k} & J_{14,k} \\ J_{21,k} & J_{22,k} & J_{23,k} & J_{24,k} \\ J_{31,k} & J_{32,k} & J_{33,k} & J_{34,k} \end{bmatrix}, \tag{4.17}$$

where $J_{ij,k} = \partial F_{i,k}/\partial x_{j,k}$. Now the extended Jacobian $\boldsymbol{J}|_N \in \mathbb{R}^{3N \times N+3}$ can be constructed.

For three measurements we have $\boldsymbol{x}|_3 = [v_x, v_y,\, v_z,\, v_{i,1}, v_{i,2}, v_{i,3}]^T$ and

$$\boldsymbol{J}|_3 = \left[\begin{array}{cccc:c:c} J_{11,1} & J_{12,1} & J_{13,1} & J_{14,1} & 0 & 0 \\ J_{21,1} & J_{22,1} & J_{23,1} & J_{24,1} & 0 & 0 \\ J_{31,1} & J_{32,1} & J_{33,1} & J_{34,1} & 0 & 0 \\ \hdashline J_{11,2} & J_{12,2} & J_{13,2} & 0 & J_{14,2} & 0 \\ J_{21,2} & J_{22,2} & J_{23,2} & 0 & J_{24,2} & 0 \\ J_{31,2} & J_{32,2} & J_{33,2} & 0 & J_{34,2} & 0 \\ \hdashline J_{11,3} & J_{12,3} & J_{13,3} & 0 & 0 & J_{14,3} \\ J_{21,3} & J_{22,3} & J_{23,3} & 0 & 0 & J_{24,3} \\ J_{31,3} & J_{32,3} & J_{33,3} & 0 & 0 & J_{34,3} \end{array}\right],$$

which is straightforward to extend to N measurements. Notice that the first three columns are due to the three airspeed components, which are assumed equal across measurements. The other columns are due to the induced velocity, which is different between measurements.

Transformed formulation. When combining measurements from different poses, the wind velocity has to be transformed into a common coordinate frame. Otherwise, the constant wind velocity assumption will not hold. Define the freestream velocity of propeller k as

$$\boldsymbol{v}_k = \begin{bmatrix} v_{x,k} \\ v_{y,k} \\ v_{z,k} \end{bmatrix} = \boldsymbol{R}_k \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \boldsymbol{v}_{0,k} = \boldsymbol{R}_k \boldsymbol{v} + \boldsymbol{v}_{0,k}, \tag{4.18}$$

and use the transformed velocities when calculating (4.14) and (4.17). It is assumed that the robot is moving between measurements. Therefore, the offset velocity $\boldsymbol{v}_{0,k}$ can be obtained from a pose estimation system as the relative velocity of the robot between two measurements. The propeller offset velocity due to the body angular velocity may also be used, i. e. $\boldsymbol{v}_{0,k} = \boldsymbol{R}_{pb,k}\boldsymbol{\omega} \times \boldsymbol{r}_k$, where $\boldsymbol{R}_{pb,k}$ is the rotation from the body to the k-th propeller frame. The Jacobian of this formulation can be found as (4.29) in Section 4.4.3.

This formulation allows us to determine all three components of the freestream velocity independently. It also can be used to obtain the instantaneous wind velocity components when the propellers are not mounted to the multicopter frame in a coplanar configuration.

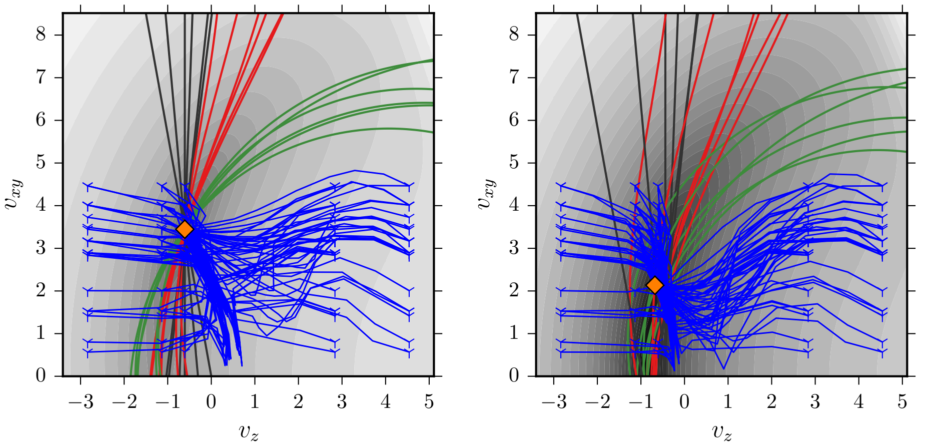
Solving the system of equations. In perfect conditions, the solution to (4.16) will be at the intersection of all nonlinear functions, where $\boldsymbol{F} = \boldsymbol{0}$. This corresponds to a multidimensional root-finding problem. However, when the measured aerodynamic power does not match momentum theory (i.e. under model mismatch), the functions will not necessarily intersect. The solution is therefore the point that is closest to all functions. In this case a nonlinear least squares problem has to be solved, with the objective function

$$f = \tfrac{1}{2}\boldsymbol{F}^T\boldsymbol{F}, \tag{4.19}$$

for example using a Levenberg-Marquardt solver [102, 23].

When an exact solution exists, it will be at $f = 0$, i. e. the intersection of $\boldsymbol{F} = \boldsymbol{0}$. Otherwise, if there is a model mismatch or noise in P_a, a least squares solution is obtained. Figure 4.15 shows convergence of the solver for different initial guesses and noise on P_a.

Limiting the search space. The space of (4.19) can contain local optima. From the underlying physics, the same measured power can be obtained by various wind and induced velocities. The optimized variables are velocities. Therefore, physical considerations may be used to determine the set of feasible solutions. A flying robot must expend power to generate thrust, which implies $T > 0$ and $P_a > 0$, for which (2.10) and (2.12) are used, respectively. The induced velocity is $v_i < v_h$ in the normal working state, and $v_i > v_h$ in



(a) No noise in P_a, contours at real v_i (b) Noise in P_a, contours at converged v_i

Figure 4.15: Zero contours of (4.19), with color indicating the function value. Green lines are contours of $F_1^{(k)}$, magenta lines are contours of $F_2^{(k)}$, and black lines are contours of $F_3^{(k)}$ at $v_i = \text{const}$. The blue lines show convergence of a Levenberg-Marquardt solver for different initial guesses. The converged solution is depicted as a red cross. The velocity components v_x and v_y lumped into v_{xy}. $N = 6$ measurements were used, velocity $v_\infty = 3.5\,\text{m/s}$, angle of attack $\alpha = 10°$, $v_h \in [4.1 \ldots 6.7]\,\text{m/s}$, and measurement angles up to 10°. Without noise on the power measurement (Figure 4.15(a)), the solution converges to the exact wind velocity $\boldsymbol{v}_w = [-3.45, 0\ -0.61]^T\,\text{m/s}$. With noise in the power measurements, the least-squares solution moves depending on measurement conditioning. Copyright © 2016 IEEE [140].

the VRS. VRS is excluded from the search space because momentum theory is invalid in that state. Therefore, the induced velocity is limited to $0 < v_i < v_h$. Likewise, limit $\boldsymbol{v}_w$ can be limited in case its maxima are known. In order to limit the search space using the Levenberg-Marquardt method, a quadratic barrier function F_4 is added to the optimization problem formulation [102]. which increases the size of the problem, as the function becomes $\boldsymbol{F} \in \mathbb{R}^{4N}$, and the Jacobian becomes $\boldsymbol{J}|_N \in \mathbb{R}^{4N \times N+3}$.

Normalization. In order to improve stability of the numerical solution, The goal function is normalized to its initial value f_0, i. e. we minimize $f' = f_0^{-1} f$. Furthermore, the functions $F_{1\ldots3}^{(k)}$ are normalized to $v_h^{(k)}$, such that $F_1' = F_1/v_h^4$, $F_2' = F_2/v_h^3$, $F_3' = F_3/v_h^3$, $F_4' = F_4/v_h^2$. In this way, the function values are dimensionless and have the same order of magnitude.

Sensitivity analysis. Measurement noise will shift the estimated wind velocity in a nonlinear manner, see Figure 4.15. A sensitivity analysis helps estimating this effect. Figure 4.16 depicts the converged solutions for increasing noise amplitude in the measured power. Since the quality of the solution will depend on the distribution of measurement poses, these are uniformly distributed under different maximum angles, from 5° to 20°. Higher relative angles between measurement poses increase robustness of the solution. However, estimation of the horizontal wind velocity components is very sensitive to power measurements.

The horizontal velocity components may therefore be estimated using the induced drag model, i. e. from the external force. As shown in Figure 4.17, this allows a robust estimation of the vertical wind velocity component and the propeller induced velocity even for a high error in power measurements. A minimum angular distance between measurements should also be considered when choosing suitable measurements for the NLS problem. Having an offset velocity $\boldsymbol{v}_0$ in (4.18) additionally reduces sensitivity to noise in the power

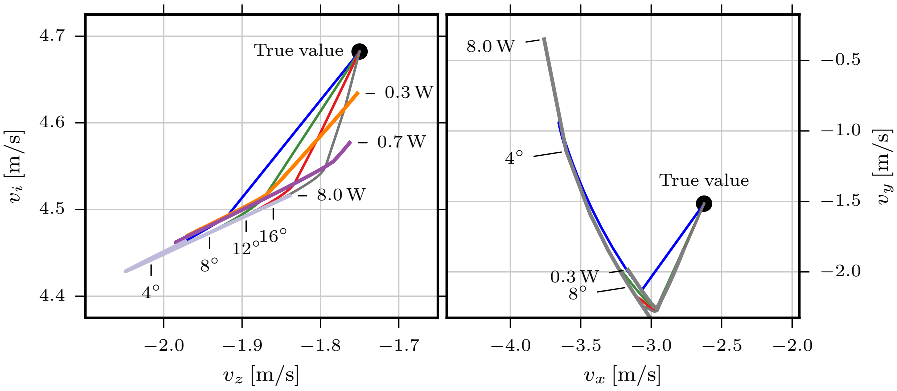


Figure 4.16: Sensitivity of the NLS solution to error in aerodynamic power $\tilde{P}_a$ up to 8 W, using $N = 12$ measurements, and maximum measurement angles σ_α up to 20°. For $\sigma_\alpha = 0°$, the solution diverges out of the depicted range and is not shown. Wind speed is chosen to be $v_\infty = 3.5\,\mathrm{m/s}$. Larger measurement angles lead to a more robust solution, as the estimated wind velocity is closer to the real value even for high errors in the aerodynamic power. The vertical wind component v_z and propeller induced velocity v_i are estimated with good accuracy for a wide range of $\tilde{P}_a$. However, the horizontal wind components v_x, v_y diverge from their real values even for low $\tilde{P}_a$. Copyright © 2016 IEEE [140].

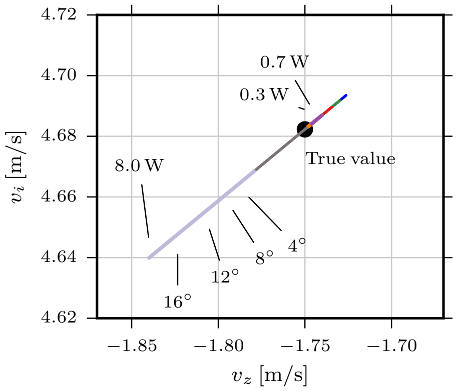


Figure 4.17: Sensitivity analzsis of the NLS solution for the same case as in Figure 4.16, however for perfect knowledge of horizontal velocity components (v_x, v_y). These may be obtained from the induced drag model, i. e. the external force. When the horizontal wind velocity is known, the vertical component maz be determined robustly from aerodynamic power measurements. Copyright © 2016 IEEE [140].

measurements.

Combined wind estimator. In order to overcome limitations of the two presented methods, see Section 4.3 and Section 4.4, a combined wind estimator is finally proposed, see Figure 4.18. The horizontal velocity components (v_x, v_y) are obtained from the external wrench or aerodynamic power, using models identified in Section 4.3. The estimated aerodynamic power and known (v_x, v_y) are then used to calculate v_z using the nonlinear least squares formulation, by minimizing (4.19).

4.4.1 Comparison to data-driven estimation

Next, the optimization-based combined wind estimator is compared to machine learning based models. Different sources of horizontal velocity in the optimization are also compared. Figure 4.19 shows components of the airspeed for the complete wind tunnel dataset. Ground truth is shown in black; estimation using the external force and rotor speed $\boldsymbol{v}_r(\boldsymbol{f}_e/\sum\boldsymbol{\varpi})$ is shown in red; estimation using aerodynamic power of the coaxial pairs $\boldsymbol{v}_r(\bar{\boldsymbol{P}}_a)$ is shown in

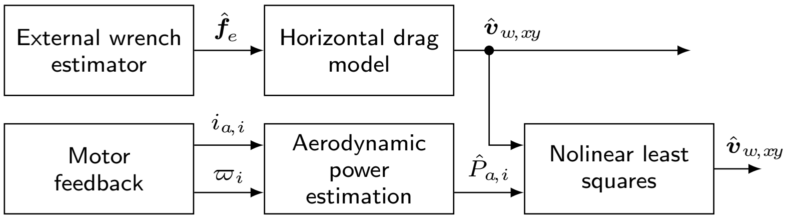


Figure 4.18: The combined estimation scheme calculates the horizontal wind velocity from the external wrench. This is then used as an input to the wind velocity estimator using the motor power, which leads to improved accuracy.

light blue. Finally, for the $z-$component the velocity obtained by the combined estimator is also shown in light green as v_z^{opt}. In this case the $\boldsymbol{v}_r(\bar{\boldsymbol{P}}_a)$ model was used to obtain the horizontal airspeed components. The result therefore uses only motor power to estimate airspeed. The result mirrors conclusions from Section 4.3. The mean of all models follows the ground truth, however when using motor power, the result is noisier. This is due to the noisy motor current measurements in our experiments. Second, large errors occur where the aerodynamic power cannot be estimated correctly due to motor saturation, such as at the beginning of the dataset.

For the optimization based combined estimator, a window of 3 measurements was used, utilizing all 3 motor pairs, for a total of 9 measurements per optimization. As these are close in time and orientation, the results become inaccurate and noisy with increasing horizontal airspeed components. The result is otherwise comparable to the machine learning based method, however the latter outperforms the optimization slightly. This shows that the optimization method can be used to obtain airspeed from real in-flight measurements.

Figure 4.20 compares how utilizing different sources of horizontal airspeed for the combined estimator affects prediction accuracy. A histogram is shown only for the $z-$component of the relative airspeed. As a benchmark, we show the airspeed obtained by using the external force as $v_z^{f_e}$. It has the smallest variance in comparison. The machine learning model $\boldsymbol{v}_r(\bar{\boldsymbol{P}}_a)$ is shown as $v_z^{P_a}$, and has the second smallest variance. Using the ground truth horizontal velocity in the combined wind estimator is shown as v_z^{opt,v_r}. Using the

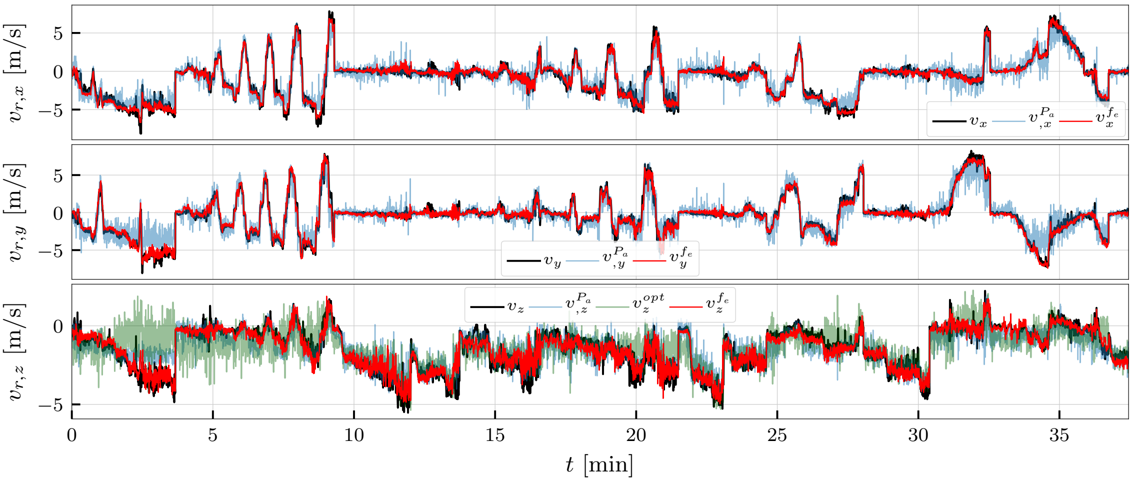


Figure 4.19: Estimation of relative airspeed for the complete dataset from Figure 4.4, using machine learning models $\boldsymbol{v}_r(\boldsymbol{f}_e \sum \varpi)$ (shown as v^{f_e}), $\boldsymbol{v}_r(\boldsymbol{P}_a)$ (shown as v^{P_a}), and the optimization based combined wind estimator described in this section (shown as v^{opt}). For the latter, the horizontal velocity was obtained from motor power.

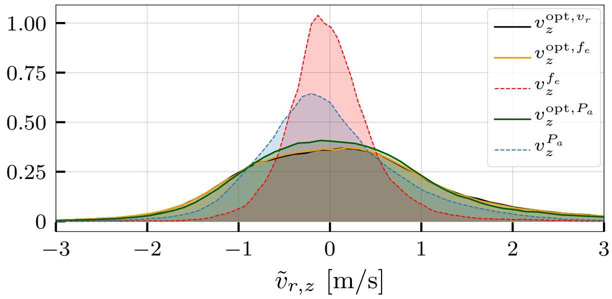


Figure 4.20: Normalized error histograms of the vertical airspeed component v_z for the dataset from Figure 4.4, For the combined wind estimator, we compare different sources of horizontal airspeed components: The model $\boldsymbol{v}_r(\boldsymbol{f}_e \sum \varpi)$ is used in v_z^{opt,f_e}. The model $\boldsymbol{v}_r(\boldsymbol{P}_a)$ is used in v_z^{opt,P_a}. Lastly, we use the exact airspeed in v_z^{opt,v_r}.

external force model for this purpose is depicted as v_z^{opt,f_e}. Finally, the histogram for the result shown in Figure 4.19 is depicted as v_z^{opt,P_a}. Interestingly, the combined estimator is not sensitive to the source of horizontal airspeed. However, the variance of the estimated velocity is higher than when using machine learning based models.

In conclusion, it was shown that airspeed may be obtained using motor power measurements only. The underlying physics of the problem was described in the optimization based approach. Accuracy of the estimation will clearly be limited by the accuracy of the estimated aerodynamic power. Therefore, having a good measurement of the motor current and a good estimate of the aerodynamic power are crucial to the applicability of this approach.

4.4.2 Choosing optimal measurements

Due to the sensitivity of the optimization problem, a natural question is how to choose measurements to maximize observability of the wind velocity, i.e. reduce sensitivity to noise. This problem is closely related to generating exciting trajectories for optimal parameter estimation in robotics, see [10, 132, 111]. The results of the following analysis may then be applied to path planning, trajectory generation, or as a control signal to achieve active sensing.

According to matrix function literature [144, 59, 46], the sensitivity to noise of an optimization problem such as (4.19) depends on the condition number of its Jacobian. By minimizing the condition number of the Jacobian $\boldsymbol{J} = \partial \boldsymbol{f}(\boldsymbol{x})/\partial \boldsymbol{x}$, the observability of the problem is maximized. The absolute condition number $\hat{\kappa}(\cdot)$ is defined as

$$\hat{\kappa} = \|\boldsymbol{J}(\boldsymbol{x})\|, \tag{4.20}$$

whereas the *relative condition number* $\kappa(\cdot)$ is defined as

$$\kappa = \frac{\|\boldsymbol{J}(\boldsymbol{x})\|}{\|\boldsymbol{f}(\boldsymbol{x})\| \, \|\boldsymbol{x}\|}, \tag{4.21}$$

where the Frobenius norm of a matrix

$$\|\boldsymbol{A}\|_F = \sqrt{\mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{A}\}} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} \tag{4.22}$$

is commonly used. Next, the Jacobian of the previously defined least squares problem is of interest. Recall the Levenberg-Marquardt update step

$$\delta \boldsymbol{x} = \left(\boldsymbol{J}^T \boldsymbol{J} + \mu \boldsymbol{I}\right)^{-1} \boldsymbol{J}^T \boldsymbol{f}(\boldsymbol{x}), \tag{4.23}$$

where μ is the regularization parameter, $\delta \boldsymbol{x}$ is the update step of the optimized variable, and $\boldsymbol{f}(\boldsymbol{x})$ is the cost function value evaluated at $\boldsymbol{x}$. Hence, the condition number of the inverted matrix $\boldsymbol{J}^T \boldsymbol{J}$ is a measure of the problem's sensitivity. Note also that $\kappa(\boldsymbol{J}^T \boldsymbol{J}) = \kappa^2(\boldsymbol{J})$.

Next, an optimization problem is defined for choosing propeller orientations that result in the best conditioning of the least squares problem. First, define a *sequential optimization problem* where a set of $K-1$ measurements is available, and the next measurement K is chosen to maximize the observability of the least squares problem. This approach is useful for online motion planning or control, where only local information is available. Let $\boldsymbol{\varphi}$ be a suitable parameterization of the measurement orientation. The orientation $\boldsymbol{\varphi}_K^*$ associated with the *next best measurement* is then obtained by solving the optimization problem

$$\boldsymbol{\varphi}_K^* = \underset{\boldsymbol{\varphi}}{\arg\min}\ \kappa(\boldsymbol{J}). \tag{4.24}$$

The problem (4.19) is normalized for easily interpretable physical values. The optimization may therefore be simplified to minimizing the absolute condition number $\hat{\kappa}$, obtaining

$$\boldsymbol{\varphi}_K^* = \underset{\boldsymbol{\varphi}}{\arg\min}\ \left\|\boldsymbol{J}\right\|_F^2, \tag{4.25}$$

where the norm was squared to simplify computation of the derivatives. Note that this approach is similar to observability analysis, with the Jacobian $\boldsymbol{J}$ corresponding to the observation matrix.

This local problem may also be solved online by means of a gradient descent algorithm. Define $\boldsymbol{\varphi}$ to be a suitable parameterization of the measurement orientation. In continuous time, solving

$$\dot{\boldsymbol{\varphi}} = -\gamma \frac{\partial \left\|\boldsymbol{J}\right\|_F^2}{\partial \boldsymbol{\varphi}}, \tag{4.26}$$

where γ is the descent factor, will locally converge to the optimal measurement angles. Note that barrier functions should be included as well in order to ensure physically feasible solutions only. Using the squared Frobenius norm leads to

$$\frac{\partial \left\|\boldsymbol{J}\right\|_F^2}{\partial \varphi_i} = 2 \sum_{i=1}^{m} \sum_{j=1}^{n} J_{ij} \frac{\partial J_{ij}}{\partial \varphi_i} \tag{4.27}$$

In other cases it might be desirable to plan a path or trajectory that will contain optimal measurement poses. Such a *simultaneous optimization* problem can be defined by having a set of roll and pitch angles $(\boldsymbol{\phi}, \boldsymbol{\theta})$. The optimal angles are then a solution to the optimization problem over all $\boldsymbol{\varphi}$ such that

$$\boldsymbol{\varphi}_{1\ldots K}^* = \underset{\varphi_{1\ldots K}}{\arg\min}\ \kappa(\boldsymbol{J}). \tag{4.28}$$

Though it was shown how trajectories for optimal wind estimation can be obtained in principle, this active sensing approach to wind estimation is out of scope of this thesis and left for future work.

4.4.3 Jacobian of the optimization problem

Take $j = (1,2,3)$ and the substitutions $\boldsymbol{z} = \boldsymbol{v} + \boldsymbol{R}_k^T \boldsymbol{v}_{0,k}$, and $\boldsymbol{y} = \boldsymbol{v} - \boldsymbol{R}_k^T \left(\boldsymbol{v}_{0,k} - v_{i,k}\boldsymbol{e}_3\right)$. Elements of the Jacobian (4.17) can now be written as

$$\begin{aligned}
J_{1j,k} &= 2v_{i,k}^2\, z_{1,j} - 2R_{3j,k}\, v_{i,k}^3, \\
J_{14,k} &= 4v_{i,k}^3 - 6v_{z,k}v_{i,k}^2 + 2v_{i,k}\|\boldsymbol{v}_k\|^2, \\
J_{2j,k} &= -\tfrac{v_{i,k}}{U_k}\left(R_{3j,k}U_k^2 + \left(v_{z,k} - v_{i,k}\right) y_j\right), \\
J_{24,k} &= \tfrac{1}{U_k}\left(U_k^2\left(2v_{i,k} - v_{z,k}\right) + v_{i,k}\left(v_{z,k} - v_{i,k}\right)^2\right), \\
J_{3j,k} &= -R_{3j,k}, \\
J_{34,k} &= 1.
\end{aligned} \tag{4.29}$$

In order to obtain the gradient (4.26), the partial derivatives of the Jacobian (4.17) are required. Define the partial derivative of A w.r.t φ as $A^\varphi := \frac{\partial A}{\partial \varphi}$. For $j = (1,2,3)$, the partial derivatives of the Jacobian w.r.t a rotation parameter φ are

$$\begin{aligned}
J_{1j}^\varphi &= 2v_i^2 z_j - v_i^3 R_{3j}^\varphi, \\
J_{14}^\varphi &= -6v_i^2 v_z^\varphi, \\
J_{2j}^\varphi &= -v_i\Big\{U R_{3j}^\varphi + U^\varphi R_{3j} \\
&\qquad + \tfrac{1}{U}\left[v_z^\varphi y_j + \tfrac{U-1}{U} y_j^\varphi (v_z - v_i)\right]\Big\}, \\
J_{24}^\varphi &= U^\varphi(2v_i - v_z) - U v_z^\varphi \\
&\qquad + \tfrac{v_z - v_i}{U}\Big[2v_i v_z^\varphi - \tfrac{U^\varphi}{U}(v_z - v_i)\Big], \\
J_{3j}^\varphi &= -R_{3j}^\varphi, \\
J_{34}^\varphi &= 0.
\end{aligned} \tag{4.30}$$

Note that the term associated with $\|\boldsymbol{v}_k\|^\varphi = 0$ was dropped because the rotation does not affect the wind velocity norm.

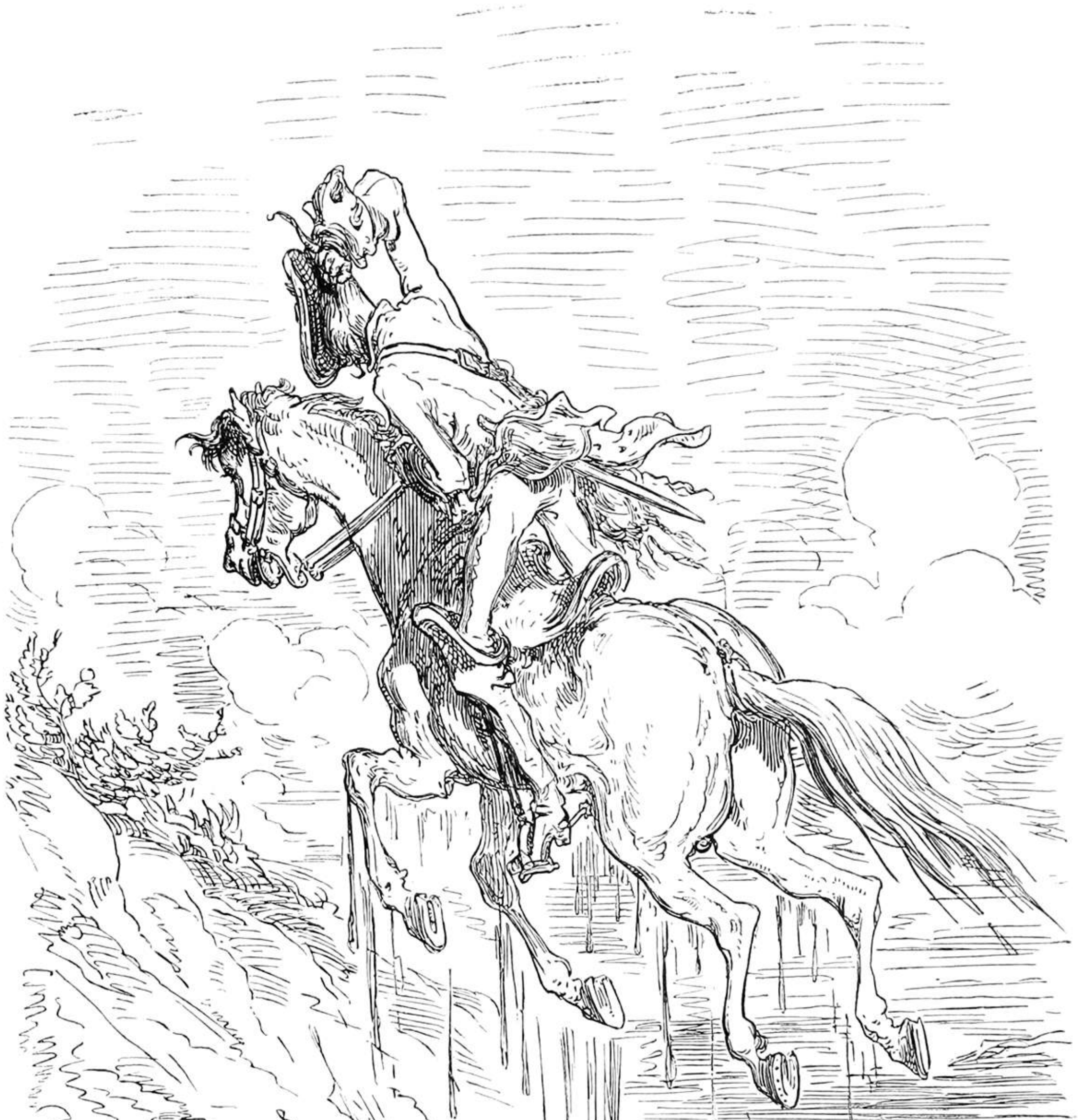


Illustration: O. E. Jahyer: Wunderbare Reisen von Münchhausen

CHAPTER 5

Force discrimination

Now that the dynamics and aerodynamic models have been identified, the problem of discriminating between the terms of the external wrench acting on the robot is investigated. In this chapter, several novel methods for input discrimination are presented in the context of the awareness pipeline presented in [142]. The awareness pipeline is a generalization of Fault Detection, Identification and Isolation [54, 56]. *Detection* provides a binary signal (true/false) whether a signal is present. Collision detection is extended to flying robots, to also be effective under wind influence. A novel method for *contact detection under wind influence* is also developed. *Isolation* deals with determination of the kind and location of the signal. An example is obtaining the contact or collision position on the robot's convex hull upon detecting either. Lastly, *identification* deals with monitoring the time-variant behavior of the signal. In this context, this is the reconstruction of the signal of interest.

This chapter is organized as follows. In Section 5.1 the problem of force discrimination is formally defined. Collision detection under wind influence is investigated in Section 5.2. Next, the contact location is obtained in an isolation step in Section 5.3. In Section 5.4 an aerodynamic torque model is used to detect slow contact forces under wind influence, lying in the same frequency region. This is then used in Section 5.5 to develope the modified model checking scheme for force discrimination. In Section 5.6, it is shown how the the contact wrench may be obtained from an aerodynamics model and a known contact position. This is incorporated in a particle filter framework in Section 5.7 to directly estimate the contact position under wind influence. The aerodynamic model that regresses motor power to airspeed is used in Section 5.8 to perform force discrimination using an independent measurement of the wind speed. All developed methods are fused in a Kalman filter framework in Section 5.9. Finally, a summary of the results is given in Section 5.10

5.1 Problem statement

The goal of force discrimination is outlined in Figure 5.1: given an external wrench $\boldsymbol{\tau}_e$ that is a sum of the fault wrench $\boldsymbol{\tau}_f$ (e.g. caused by collisions), the physical interaction wrench $\boldsymbol{\tau}_i$ (e.g. caused by a person pushing the vehicle), and the disturbance wrench $\boldsymbol{\tau}_d$ (caused by wind), obtain a reconstruction of the constituent terms. In other words, given

$$\boldsymbol{\tau}_e = \boldsymbol{\tau}_f + \boldsymbol{\tau}_i + \boldsymbol{\tau}_d, \tag{5.1}$$

force discrimination deals with detecting the presence of a signal, extracting context dependent information (isolation), and obtaining the time-varying reconstruction of the signal (isolation). In the following, it is assumed that the external wrench $\boldsymbol{\tau}_e$ is perfectly

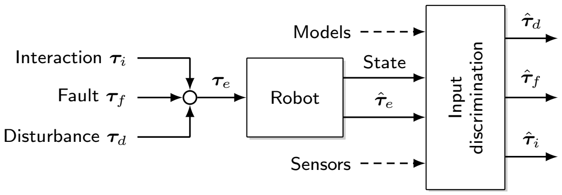
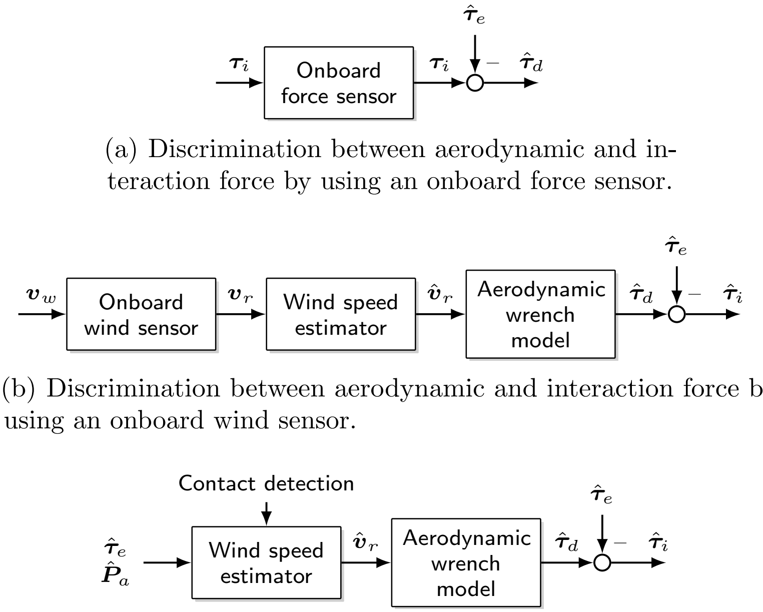

Interaction $\boldsymbol{\tau}_i$
Fault $\boldsymbol{\tau}_f$
Disturbance $\boldsymbol{\tau}_d$
$\boldsymbol{\tau}_e$
Robot
Models
State
$\hat{\boldsymbol{\tau}}_e$
Sensors
Input discrimination
$\hat{\boldsymbol{\tau}}_d$
$\hat{\boldsymbol{\tau}}_f$
$\hat{\boldsymbol{\tau}}_i$

Figure 5.1: Force discrimination reconstructs the time-varying additive inputs summed in the external wrench $\boldsymbol{\tau}_e$, given an estimate thereof $\hat{\boldsymbol{\tau}}_e$, the robot's state, and additional models and/or sensor inputs.



(a) Discrimination between aerodynamic and interaction force by using an onboard force sensor.

(b) Discrimination between aerodynamic and interaction force by using an onboard wind sensor.

(c) Discrimination between aerodynamic and interaction force by estimating the wind speed.

Figure 5.2: An overview of methods to discriminate slow aerodynamic and interaction wrenches. In this chapter, model-based methods in the (c) family are developed. They do not require dedicated wind or force sensors.

known. When using the external wrench estimator this means that $\hat{\boldsymbol{\tau}}_e \approx \boldsymbol{\tau}_e$. Figure 5.2 depicts some possible approaches to solving this problem. For the scheme in Figure 5.2(a), an onboard force sensor is needed, and for Figure 5.2(b) an onboard wind sensor and an aerodynamics model are needed. In this chapter, model-based methods like the scheme in Figure 5.2(c) are developed, as they do not require such additional onboard sensors.

5.2 Collision detection under wind influence

A collision will cause a force on the flying robot with a large amplitude and short duration. Briod et al. [25] have used a constant threshold on the estimated external force as a *collision detection signal*:

$$CD_A = \begin{cases} 1 & \text{if } \|\hat{\boldsymbol{f}}_e\| \geq f_k \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

In this way, both slow contacts and impacts will be detected as collisions, but the detection sensitivity will then highly depend on modeling errors and measurement noise. A more robust method of detecting *contacts* is to require the external wrench to be larger than a threshold f_k for a certain amount of time T_k. The *contact detection signal* CD_B may be obtained as

$$CD_B = \begin{cases} 1 & \text{if } \int \operatorname{sign}(\|\hat{\boldsymbol{f}}_e\| - f_k)\mathrm{d}\tau > T_k \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

where the integral has a zero lower limit. However, these signals will also trigger in strong winds, i.e. they cannot distinguish between wind and collision forces.

Collision detection for robot manipulators has been discussed in more detail in [56, 30], and in [136] and [138] for aerial robots in particular. The main idea is that discrimination between aerodynamic and *collision* forces may be achieved by considering the respective signals' frequency characteristics. The frequency ranges of the constituent terms are used to design appropriate filters. Collisions may be detected by applying a highpass filter $H(f, \omega_f)$ on the external force, with break frequency ω_f, to obtain the *collision detection signal*

$$CD = \begin{cases} 1 & \text{if } \exists i: \ H(|\hat{f}_{e,i}|, \omega_f) > f_{c,i} \\ 0 & \text{otherwise,} \end{cases} \tag{5.4}$$

where $f_{c,i}$ is the collision detection threshold. The threshold is be state-dependent, as increasing airspeed will also increase noise in the external wrench. Note that quasistatic contact forces cannot be distinguished from aerodynamic forces from the frequency content alone. A method is needed to distinguish constituent terms of the external wrench in the same frequency range. Two methods for isolating the collision wrench from the aerodynamic wrench are presented next – the *filter bank* and the *model checking* method.

Filter bank. The first method is based on spectrum analysis of the external wrench during flight, structured as in Figure 5.4(a). The frequency ranges of its constituent terms are isolated and appropriate filters are designed. The design approach is illustrated in Figure 5.3, which shows spectrograms of the external force components acting on a quadrotor in simulation, with darker shades indicating higher amplitude. The flight is divided in four phases, each lasting 20 s:

- Phase 1: No drag forces, no collisions. The quadrotor hovers, hence no external wrench is acting on the robot.
- Phase 2: Drag forces, no collisions. The aerodynamic model is active, the quadrotor hovers, and the wind speed changes. This results in displacement from the hovering position due to the used PD controller.
- Phase 3: No drag forces, collisions. The quadrotor flies into a surface located at $x = 5\,\mathrm{m}$, resulting in an high force and a static contact force. The position remains constant during contact.
- Phase 4: Drag forces, collisions. Same scenario as in Phase 3, but with the aerodynamic model enabled.

The spectrograms indicate that collision events can be clearly isolated using the high frequency components of the external force, while the aerodynamic forces principally contain low frequency components. Therefore, after analyzing the resulting spectrograms of this example, shown in Figure 5.3, a lowpass filter with a cutoff frequency of 0.2 Hz and a highpass filter at 5 Hz are applied to isolate the signals. Note that quasistatic contact forces cannot be distinguished from aerodynamic forces in this scheme.

Model checking. The second isolation method is based on wind model checking, and assumes a known wind velocity propagation model. Its structure is shown in Figure 5.4(b).

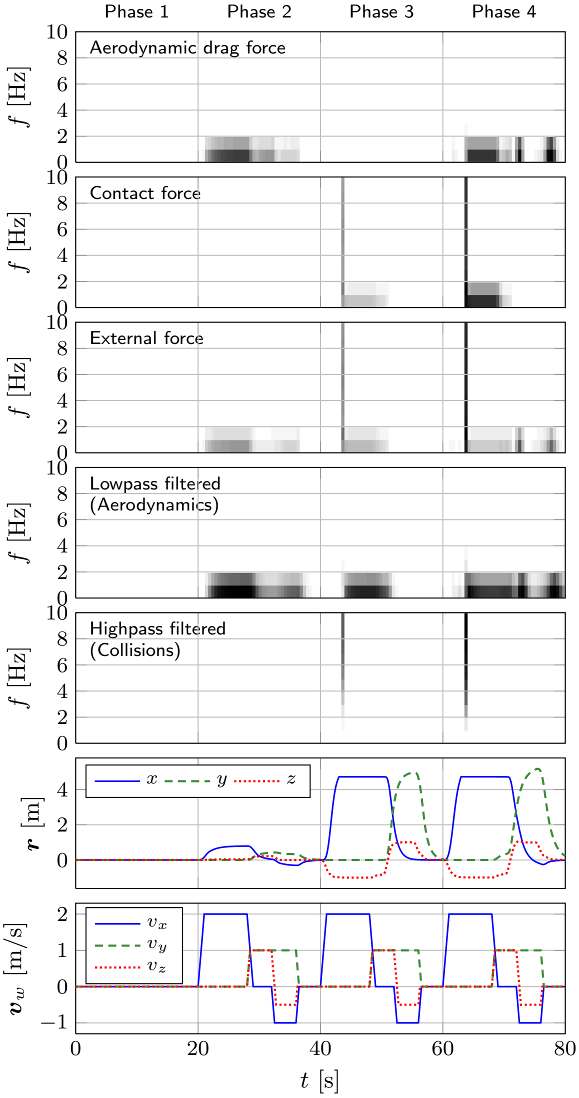


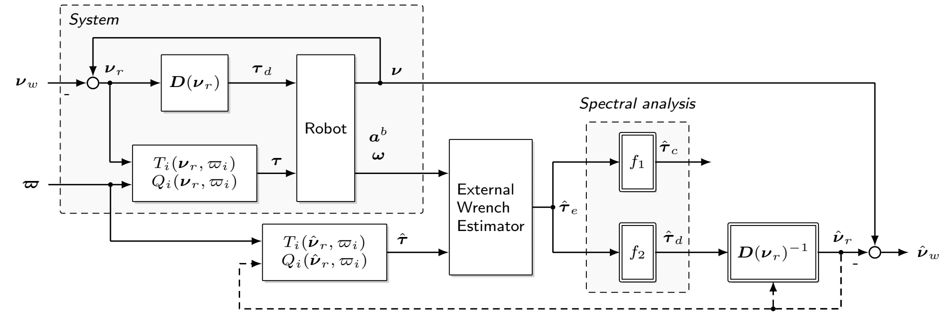
Figure 5.3: Spectrograms of the wrenches acting on a quadrotor during simulated flight and the according robot position as well as wind speed. See the text for an explanation of the phases. The spectrograms show that aerodynamic forces can be distinguished from collisions by exploiting the respective frequency characteristics in the estimated external wrench. Copyright © 2015 IEEE [136].

This is motivated by wind models from aerospace literature, which are based on the power spectral density, such as the Dryden wind gust model [87]. This allows us to model the wind velocity as a dynamic system $H(s)$. In this method, $\hat{\boldsymbol{\tau}}_e = \boldsymbol{\tau}_d$ is first assumed to obtain the wind velocity input. After updating the model, the expected aerodynamic wrench $\hat{\boldsymbol{\tau}}_d$ is calculated using the estimated wind velocity $\hat{\boldsymbol{\nu}}_w$ and compare it to the external wrench. The residual wrench $\boldsymbol{\tau}_r$ will then contain the contact wrench and model errors.

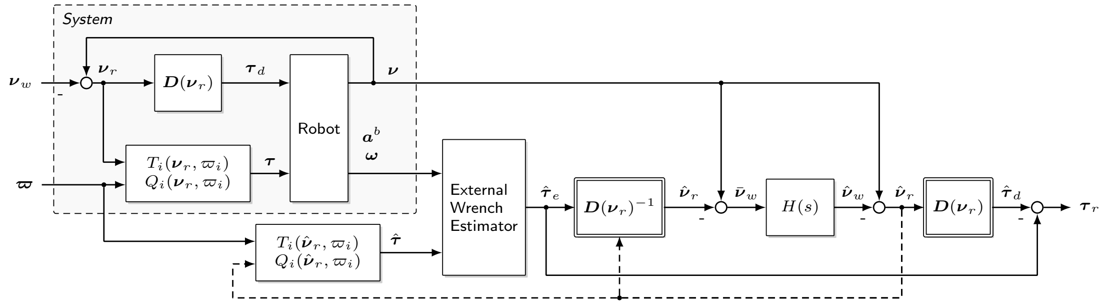
The contact torque $\hat{\boldsymbol{m}}_c$ is contained in $\boldsymbol{\tau}_r$, and provides additional information for detecting contacts in the presence of wind. Notice that the contact force and contact wrench may be filtered separately, thus $\hat{\boldsymbol{m}}_c$ may be obtained using both methods.

Results

The presented concepts are further illustrated through quadrotor simulation results with parameters from Table 5.1. Contacts are modeled with a Hunt-Crossley model [33], which


System
$\boldsymbol{\nu}_w$
$\boldsymbol{\nu}_r$
$\boldsymbol{D}(\boldsymbol{\nu}_r)$
$\boldsymbol{\tau}_d$
$\boldsymbol{\nu}$
Robot
$\boldsymbol{a}^b$
$\boldsymbol{\omega}$
$T_i(\boldsymbol{\nu}_r, \varpi_i)$
$Q_i(\boldsymbol{\nu}_r, \varpi_i)$
$\boldsymbol{\tau}$
ϖ
$T_i(\hat{\boldsymbol{\nu}}_r, \varpi_i)$
$Q_i(\hat{\boldsymbol{\nu}}_r, \varpi_i)$
$\hat{\boldsymbol{\tau}}$
External Wrench Estimator
$\hat{\boldsymbol{\tau}}_e$
Spectral analysis
f_1
$\hat{\boldsymbol{\tau}}_c$
f_2
$\hat{\boldsymbol{\tau}}_d$
$\boldsymbol{D}(\boldsymbol{\nu}_r)^{-1}$
$\hat{\boldsymbol{\nu}}_r$
$\hat{\boldsymbol{\nu}}_w$

(a) Discrimination between aerodynamic and collision wrenches by the *filter bank* method. The filters f_1 and f_2 are obtained from spectral analysis of the lumped external wrench $\hat{\boldsymbol{\tau}}_e$. The wind velocity is then estimated from the drag component $\hat{\boldsymbol{\tau}}_d$.


System
$\boldsymbol{\nu}_w$
$\boldsymbol{\nu}_r$
$\boldsymbol{D}(\boldsymbol{\nu}_r)$
$\boldsymbol{\tau}_d$
$\boldsymbol{\nu}$
Robot
$\boldsymbol{a}^b$
$\boldsymbol{\omega}$
$T_i(\boldsymbol{\nu}_r, \varpi_i)$
$Q_i(\boldsymbol{\nu}_r, \varpi_i)$
$\boldsymbol{\tau}$
ϖ
$T_i(\hat{\boldsymbol{\nu}}_r, \varpi_i)$
$Q_i(\hat{\boldsymbol{\nu}}_r, \varpi_i)$
$\hat{\boldsymbol{\tau}}$
External Wrench Estimator
$\hat{\boldsymbol{\tau}}_e$
$\boldsymbol{D}(\boldsymbol{\nu}_r)^{-1}$
$\hat{\boldsymbol{\nu}}_r$
$\bar{\boldsymbol{\nu}}_w$
$H(s)$
$\hat{\boldsymbol{\nu}}_w$
$\hat{\boldsymbol{\nu}}_r$
$\boldsymbol{D}(\boldsymbol{\nu}_r)$
$\hat{\boldsymbol{\tau}}_d$
$\boldsymbol{\tau}_r$

(b) In the *model checking* discrimination method, we first assume $\hat{\boldsymbol{\tau}}_e = \boldsymbol{\tau}_d$ to obtain the wind velocity. Based on physical reasoning we then filter the *wind velocity* using a propagation model $H(s)$. The estimate is used to obtain an expected aerodynamic wrench. The residual $\boldsymbol{\tau}_r$ then contains contact forces and model errors.

Figure 5.4: Structures of the *filter bank* (a) and *model checking* (b) methods for discriminating between the aerodynamic and contact wrench from the lumped wrench $\hat{\boldsymbol{\tau}}_e$. Collision detection is performed on the contact wrench $\boldsymbol{\tau}_c$ or the residual $\boldsymbol{\tau}_r$. Copyright © 2015 IEEE [136].

Table 5.1: Simulation parameters used to obtain the wind estimation and collision detection results. The robot is an quadrotor with arm length $L = 0.17$ m. Copyright © 2015 IEEE [136].

Parameter	Value
$\mathcal{M}$	1.75 kg
$\boldsymbol{\mathcal{I}}$	$\mathrm{diag}\{1,1,1\} \cdot 10^{-2}$ kg m^2
D	0.254 m
ρ	1.205 kg m^{-3}
k	$1{\cdot}10^4$ N/m
λ	$2.3{\cdot}10^5$ Ns/m
n	1.5

Parameter	Value
C_T	$1.330 \cdot 10^{-1} + 3.195 \cdot 10^{-6}\,\varpi$
C_Q	$2.221 \cdot 10^{-3}$
$\boldsymbol{D}_L$	$\mathrm{diag}\{0.6, 0.6, 1.2, 0.2, 0.2, 0.12\}$ N s^{-1}
$\boldsymbol{D}_Q$	$\mathrm{diag}\{0.12, 0.12, 0.12, 0, 0, 0\}$ N s^{-2}
A_{ref}	0.392 m^2
$\boldsymbol{A}_d$	$\mathrm{diag}\{0.06, 0.06, 0\}$ s^{-1}

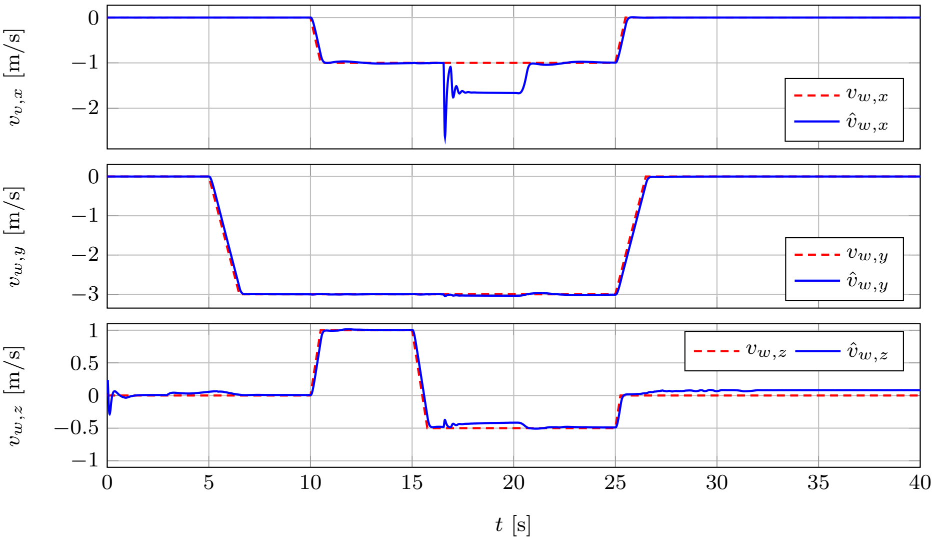


Figure 5.5: Wind velocity estimation in a collision and contact scenario, showing erroneously estimated wind velocity during contact. The quadrotor flies toward a wall starting at $t = 10$ s, and collides at $t = 15.5$ s. Contact is maintained until $t = 21$ s. The static contact force results in erroneous estimation of the wind velocity. Copyright © 2015 IEEE [136].

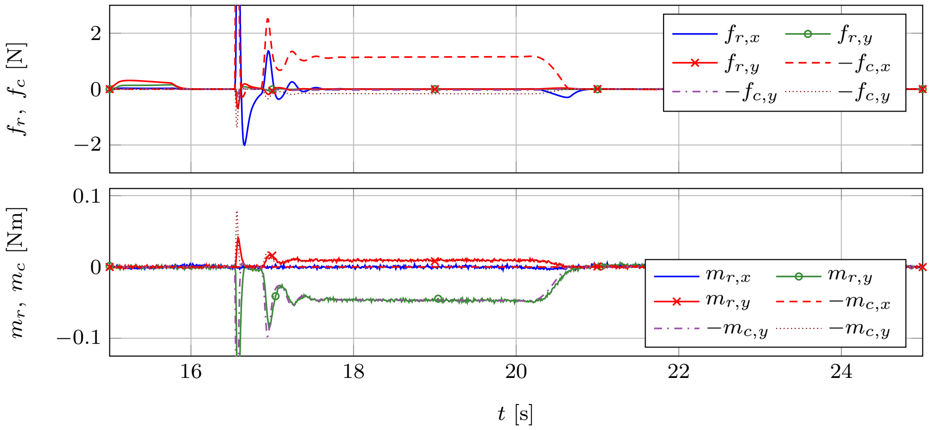


Figure 5.6: The residual wrench for the flight shown in Figure 5.5, with a first-order wind velocity propagation model. The collision may clearly be determined from the estimated external forces. However, we cannot estimate the static contact forces, as the wind model captures low frequency components, as can be seen from the misinterpreted $f_{c,x}$ component. Interestingly, the residual $\boldsymbol{m}_r$ reconstructs the contact torque. Therefore, if the contact is applying a torque to the robot, the static contact phase is visible from the external torque. Copyright © 2015 IEEE [136].

models the contact force F based on penetration depth x as

$$F(t) = \begin{cases} kx^n(t) + \lambda x^n(t)\dot{x}^n(t), & x \geq 0 \\ 0, & x < 0 \end{cases}. \tag{5.5}$$

The exponent n is a real number that takes into account the geometry of contact surfaces. A PD position controller is used for flight, which naturally results in poor position tracking under wind influence.

Figure 5.5 depicts the wind velocity estimated using inversion of the aerodynamic model

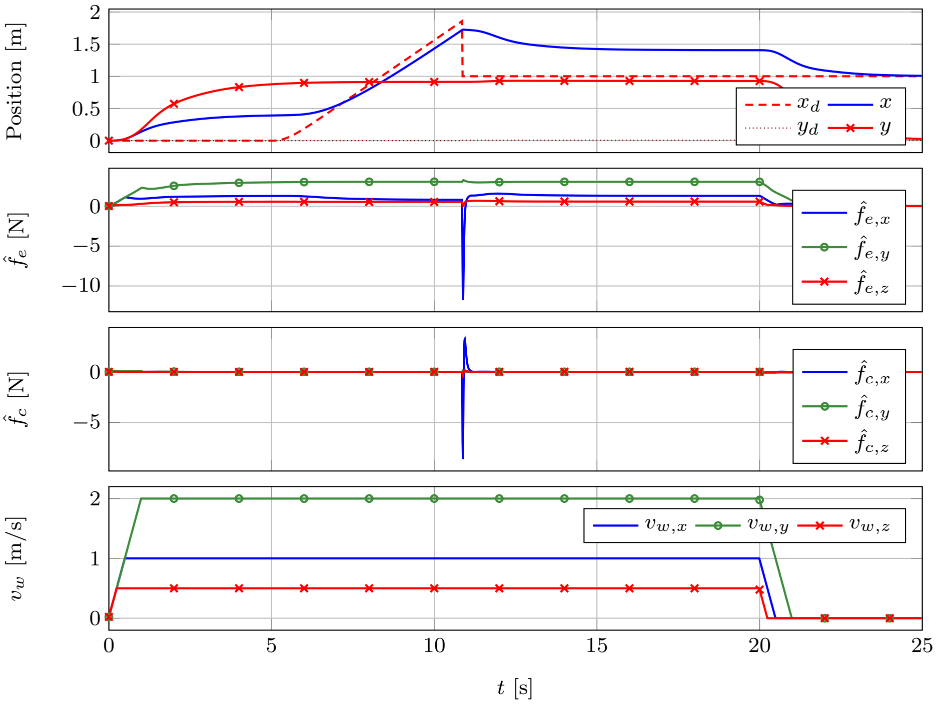


Figure 5.7: A collision reflex when the wind force creates an offset in the external force. The quadrotor collides with a surface at $t = 11$ s with a velocity of 0.2 m/s. This produces a collision force in the same order of magnitude as the wind force. The reference position $\boldsymbol{r}_d$ is set in the direction of the collision force in order to fly to a safe distance from the obstacle. By isolating the collision force $\hat{\boldsymbol{f}}_c$ to obtain the reflex direction we are able to react in the correct direction, as without wind influence. Copyright © 2015 IEEE [136].

in the *model checking* scheme. The wind model is a lowpass-filter with a time constant of 250 ms. The thrust model is known and used in the estimation, therefore it is possible estimate the vertical wind component accurately. In the simulation, a collision occurs at $t = 15.5$ s, and contact with the surface is maintained until $t = 21$ s. Due to the contact force being constant, the x component of the wind velocity is erroneous. This illustrates that contacts should be avoided if accurate wind velocity estimation is required.

Figure 5.6 shows components of the wind model residual $\boldsymbol{\tau}_r$ in the same scenario. The collision is clearly visible as a peak in the residual force. Interestingly, the contact torque can be seen in the residual. This indicates that contact could further be discriminated from wind by using a filtered residual torque signal. Note that this is not possible in general, but *only* in contact scenarios that generate torque, i.e. where the contact force acts with a lever.

Lastly, to illustrate the *filter bank* method, the quadrotor collided with a surface at a velocity of 0.2 m/s at 2.3 m/s wind. This results in a collision force that has the same order of magnitude as the aerodynamic force. The aerodynamic and collision force filters were designed according to Figure 5.3. Figure 5.7 shows the position, external force, isolated contact force and wind velocity in this scenario. The use of a PD position controller results in a position error due to wind. Upon collision, the desired position was set in direction of the estimated collision force $\hat{\boldsymbol{f}}_c$, i.e. away from the collided surface. Using the external wrench $\hat{\boldsymbol{f}}_e$ would result in an erroneous offset reflex direction due to the aerodynamic wrench. Our approach significantly robustifies collision reflexes under wind influence.

In conclusion, the simulations show the effectiveness of the presented methods in discriminating between aerodynamic and collision components of the external wrench.

5.3 Contact location

The contact location can be obtained from the estimated contact wrench during the contact phase [135, 138]. This may be used after discriminating the interaction force and torque by any presented method to obtain the contact position. By using the torque model $\boldsymbol{m}_c = \boldsymbol{r}_c \times \boldsymbol{f}_c$, the contact location is obtained by intersecting the ray

$$\begin{aligned} \boldsymbol{r}_c &= \boldsymbol{o} + k\boldsymbol{d}, \\ \boldsymbol{o} &= \frac{\boldsymbol{f}_c \times \boldsymbol{m}_c}{\boldsymbol{f}_c^T \boldsymbol{f}_c}, \\ \boldsymbol{d} &= \boldsymbol{f}_c, \end{aligned} \tag{5.6}$$

with the vehicle's convex hull to obtain the unknown parameter k. An intersection with the hull is required in order to obtain a unique solution. Note that this problem has two solutions when the ray passes through the hull. Under the assumption that the contact force points into the robot, $k < 0$ can be chosen. Now, a collision plane may be estimated in the inertial frame $\boldsymbol{n}_p^T(\boldsymbol{r}_p - \boldsymbol{o}) = 0$ with the associated normal $\boldsymbol{n}_p = \hat{\boldsymbol{f}}_e / \|\hat{\boldsymbol{f}}_e\|$ and position $\boldsymbol{r}_p = \boldsymbol{r} + \boldsymbol{R}^T \boldsymbol{r}_c$. This information could e.g. be stored in a contact map to prevent further collisions with static surfaces. The plane information may also be included in a more involved mapping solution, e.g. an octomap, and further used for (re-)planning. This tactile mapping idea is further explored in Section 6.1.2.

To show the effectiveness of this method, a quadrotor is collided with a plane for different collision speeds in simulation. The contact with the plane occurs at a single point on the robot's convex hull. The robot geometry is approximated using only 8 rectangular surfaces, which underlines the robustness of the method. The resulting plane estimation errors are depicted in Figure 5.9. The estimated surface normal is very accurate for the entire velocity range and all collision configurations. This is due to the normal being estimated from the force direction, which has a high signal to noise ratio on collision. The position determination accuracy depends highly on the used geometric model. Even for the quite approximate geometric model, the accuracy is better than 1 cm for *straight* collisions. The maximum standard deviation of *all* collision cases is 2 mm and it is consistent across collision velocities. This shows strong robustness and repeatability. The error of the *45° horizontal* case is caused by the geometric approximation. The approximated robot's bottom surface is already 5 cm below the actual contact point, which causes a systematic error in the position estimation.

5.4 Contact detection under wind influence

Contact detection is based on the aerodynamic torque model as a function of the aerodynamic force $\hat{\boldsymbol{m}}_d(\boldsymbol{f}_d)$, see Section 4.3.5. When only the aerodynamic force is acting on the robot, the external torque $\boldsymbol{m}_e$ will match the aerodynamic torque model, i.e. $\hat{\boldsymbol{m}}_d(\boldsymbol{f}_e) = \boldsymbol{m}_e$, up to modeling errors. If there is another torque-generating force acting on the robot, the external torque will not match the model. Based on this insight the *aerodynamic torque residual* $\tilde{\boldsymbol{m}}_d$ is defined as

$$\tilde{\boldsymbol{m}}_d = \hat{\boldsymbol{m}}_d(\boldsymbol{f}_e) - \boldsymbol{m}_e. \tag{5.7}$$

The assumption of the nominal state is that only the aerodynamic wrench is acting on the robot. The identified aerodynamic torque model is evaluated at the current external force

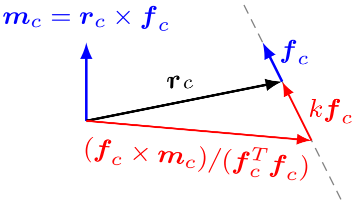

$\boldsymbol{m}_c = \boldsymbol{r}_c \times \boldsymbol{f}_c$
$\boldsymbol{f}_c$
$\boldsymbol{r}_c$
$k\boldsymbol{f}_c$
$(\boldsymbol{f}_c \times \boldsymbol{m}_c)/(\boldsymbol{f}_c^T \boldsymbol{f}_c)$

Figure 5.8: The collision position $\boldsymbol{r}_c$ can be obtained from the external force $\boldsymbol{f}_c$ and torque $\boldsymbol{m}_c$ through the line of action of $\boldsymbol{f}_c$. All forces lying on the dashed ray produce the same torque. The free parameter k is then found by intersecting the ray with the robot's convex hull. Copyright © 2017 IEEE [138].

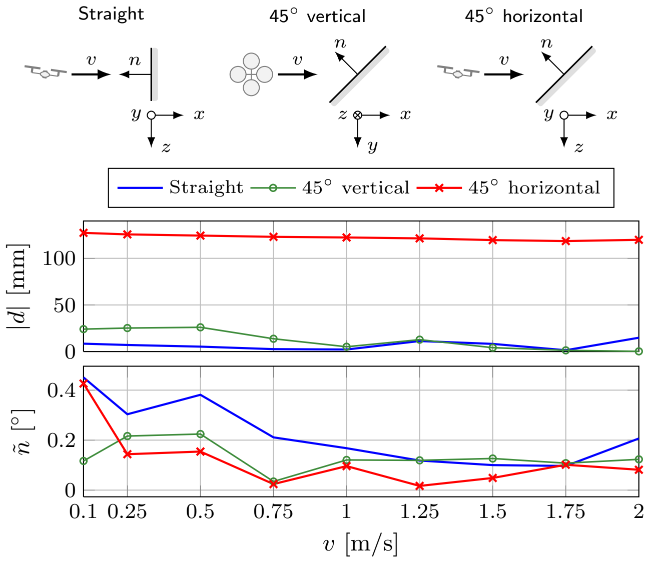


Figure 5.9: Simulation of collision location determination for a quadrotor in different collision configurations. The upper plot shows the absolute distance of the estimated plane position to the real plane $d = \boldsymbol{n}^T(\boldsymbol{r}_p - \boldsymbol{o})$. The lower plot shows the angular error of the estimated normal $\tilde{n} = \text{acos}\, \boldsymbol{n}^T \boldsymbol{n}_p$. The mean of multiple collisions is shown for each velocity. The standard deviation in both cases is too small to be visible (2 mm and $10^{-3\circ}$, respectively). Copyright © 2014 IEEE [141, 138].

$\boldsymbol{f}_e$. The residual $\tilde{\boldsymbol{m}}_d$ will be nonzero if there is an additional wrench acting on the robot that does not correspond to the aerodynamics model. The *contact detection signal* CD_1 may now be defined as

$$CD_1 = \begin{cases} 1 & \text{if } \|\tilde{\boldsymbol{m}}_d\| > \delta_d \\ 0 & \text{otherwise,} \end{cases} \tag{5.8}$$

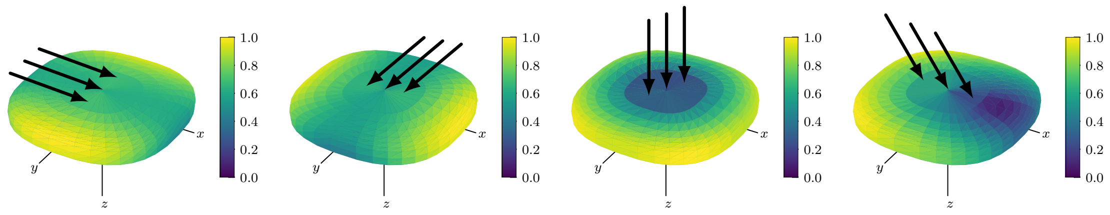
where δ_d is the threshold on the residual norm.

Failure cases. The aerodynamic torque model may be as simple as a linear combination $\hat{\boldsymbol{m}}_d(\boldsymbol{f}_d) := \boldsymbol{D}\boldsymbol{f}_d$, i.e. the aerodynamic force acting at a center of pressure. Assume

$$\begin{aligned} \boldsymbol{f}_e &= \boldsymbol{f}_d + \boldsymbol{f}_i, \\ \boldsymbol{m}_e &= \boldsymbol{D}\boldsymbol{f}_d + \boldsymbol{m}_i, \end{aligned} \tag{5.9}$$

the aerodynamic model is perfectly known, and $\boldsymbol{f}_i \neq \boldsymbol{0}$, $\boldsymbol{m}_i \neq \boldsymbol{0}$, and $\tilde{\boldsymbol{m}}_d = \boldsymbol{0}$. Then, by applying (5.9) to (5.7),

$$\boldsymbol{D}\boldsymbol{f}_d + \boldsymbol{D}\boldsymbol{f}_i - \boldsymbol{D}\boldsymbol{f}_d - \boldsymbol{m}_i = \boldsymbol{0}.$$



(a) $\boldsymbol{f}_i = [1,\ 0,\ 0]^T$ N (b) $\boldsymbol{f}_i = [0,\ 1,\ 0]^T$ N (c) $\boldsymbol{f}_i = [0,\ 0,\ 1]^T$ N (d) $\boldsymbol{f}_i = [1,\ 0,\ 1]^T$ N

Figure 5.10: Torque residual for constant forces at positions around a superellipsoidal convex hull. A contact force will only be detected by the residual if it generates a torque. The color represents $\|\tilde{\boldsymbol{m}}_d\|$ and is normalized to the highest residual being equal to 1. A linear aerodynamic torque model $\boldsymbol{m}_d(\boldsymbol{f}_d)$ was used, along with a quadratic aerodynamic wrench model with a relative airspeed of $\boldsymbol{v}_r = [1.0,\ 5.0,\ 0.2]^T$ m/s.

This means that the contact detection scheme fails if

$$\boldsymbol{D}\boldsymbol{f}_i = \boldsymbol{m}_i.$$

In other words, the scheme does not work if the exerted wrench exactly matches the aerodynamics model, i.e. is indistinguishable from aerodynamic effects. In the nonlinear case, the equivalent failure is

$$\boldsymbol{0} = \hat{\boldsymbol{m}}_d(\boldsymbol{f}_d + \boldsymbol{f}_i) - \boldsymbol{m}_d(\boldsymbol{f}_i) - \boldsymbol{m}_i.$$

Limitations. The proposed detection scheme relies on the torque produced by the interaction. Therefore, as discussed above, not all combinations of force and torque will generate a residual. The detection will be local, depending on the convex hull of the robot, and the aerodynamic model. Figure 5.10 illustrates the residual norm when the contact occurs on the convex hull of the robot. Clearly, some forces generate a stronger signal than others. In the cases Figure 5.10(a) and Figure 5.10(b), contact positions that generate a torque about the $z-$axis will generate the strongest residual. Conversely, a purely vertical force as in Figure 5.10(c) will produce no torque, making it difficult to distinguish from wind when acting on the top or bottom of the hull. Similarly, in the case Figure 5.10(d), when the force is acting at particular contact positions, it will not be distinguishable from wind, as described in the failure cases.

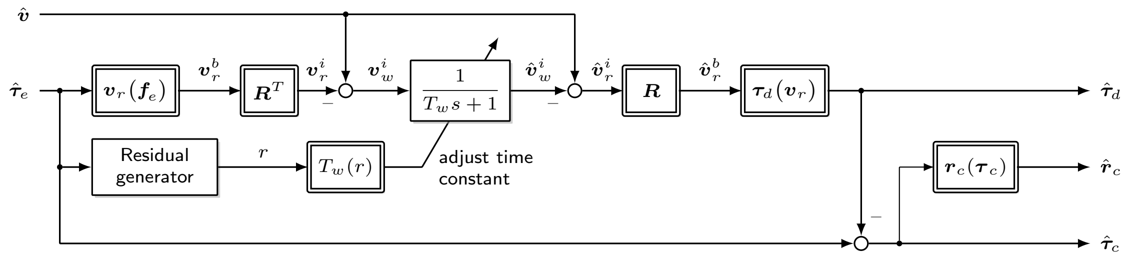

$\hat{\boldsymbol{v}}$
$\hat{\boldsymbol{\tau}}_e$
$\boldsymbol{v}_r(\boldsymbol{f}_e)$
$\boldsymbol{v}_r^b$
$\boldsymbol{R}^T$
$\boldsymbol{v}_r^i$
$-$
$\boldsymbol{v}_w^i$
$\frac{1}{T_w s + 1}$
$\hat{\boldsymbol{v}}_w^i$
$-$
$\hat{\boldsymbol{v}}_r^i$
$\boldsymbol{R}$
$\hat{\boldsymbol{v}}_r^b$
$\boldsymbol{\tau}_d(\boldsymbol{v}_r)$
$\hat{\boldsymbol{\tau}}_d$
Residual generator
r
$T_w(r)$
adjust time constant
$\boldsymbol{r}_c(\boldsymbol{\tau}_c)$
$\hat{\boldsymbol{r}}_c$
$-$
$\hat{\boldsymbol{\tau}}_c$

Figure 5.11: Modified model checking scheme for simultaneous wind and interaction estimation. The wind speed is estimated in the inertial frame by filtering the raw prediction made by an aerodynamics model learned in Section 4.3. Once contact is detected, the time constant of the wind estimator is decreased and everything else is treated as the interaction force.

5.5 Modified model-checking force discrimination scheme

Based on the contact detection signal, the force discrimination scheme depicted in Figure 5.11 is proposed next. This scheme makes the assumption that wind speed does not change significantly during physical interaction. It estimates the wind speed in the inertial frame with a filter (e.g. Kalman filter). For simplicity of argumentation, in Figure 5.11 this is a simple first-order lowpass filter. It is a modification of the model-checking scheme presented in [136] with a time-varying wind estimation time constant. Once the contact detection signal CD_1 becomes true, the time constant T_w of the wind estimator (or process noise in the case of a Kalman filter) is slowed down. This essentially pauses wind speed estimation by slowing down wind estimation. By reprojecting the estimated wind speed $\hat{\boldsymbol{v}}_w$ back into the body frame to get the relative airspeed $\hat{\boldsymbol{v}}_r$, the estimated aerodynamic wrench $\hat{\boldsymbol{\tau}}_d := \boldsymbol{\tau}_d(\hat{\boldsymbol{v}}_r)$ is obtained by passing the airspeed through the aerodynamic model. Note that this is similar to the model checking scheme in [136]. The interaction wrench is then obtained from the forward aerodynamic model as $\hat{\boldsymbol{\tau}}_i = \boldsymbol{\tau}_e - \hat{\boldsymbol{\tau}}_d$. However, it does not make any assumptions about the interaction force itself.

Simulation results. In order to evaluate the proposed discrimination scheme, the hexacopter identified in Section 2.2 was simulated, with an aerodynamic wrench model identified from data presented in Section 4.3. The robot is controlled using the compensated impedance controller (3.53), with a constant position setpoint (hover). The controller is compensating the estimated aerodynamic wrench, while being compliant w.r.t. the estimated interaction wrench. To illustrate determination of the contact position, the convex hull is an oblate superellipsoid convex hull with horizontal major semiaxes of 0.3 m, and the vertical minor semiaxis of 0.05 m. If (5.6) has no solution, i.e. does not intersect the convex hull, the previous estimated values were kept. Then a contact force was virtually injected at a position on the convex hull, while the robot motion is unconstrained. To illustrate how the scheme behaves in time-varying conditions, the simulated conditions are as follows. Until $t = 15$ s, the vehicle is yawing at a 9 °/s, and the wind speed is time-varying. After $t = 15$ s, the wind speed and yaw reference remain constant. The simulation has 4 distinct contact phases to illustrate the behavior and failure cases of the discrimination scheme:

1. Time-varying wind speed, constant contact force and position in the body frame for $3 \text{ s} < t < 6 \text{ s}$.
2. Time-varying wind speed, constant contact force in the inertial frame, time-varying contact position for $10 \text{ s} < t < 13 \text{ s}$.
3. Constant wind speed, pure contact force (no torque) for $15 \text{ s} < t < 18 \text{ s}$.
4. Wind speed step change at $t = 24$ s during contact during $21 \text{ s} < t < 26 \text{ s}$.

Simulation results are shown in Figure 5.12. Throughout the simulation, when no interaction is detected, the vehicle converges to the setpoint, as the complete external wrench is compensated as a disturbance. The wind velocity also converges to the correct value. During phase (1), wind estimation is essentially paused, and the error of the time-varying aerodynamic force is interpreted as the interaction force. The vehicle is compliant to the estimated interaction force. The raw contact position is estimated correctly. During phase (2), the time-varying interaction force is correctly estimated because the wind speed is estimated in the inertial frame, and transformed into the body frame to obtain the aerodynamic wrench. The compensated impedance controller is making the vehicle compliant to the estimated interaction wrench. This makes it possible to physically move

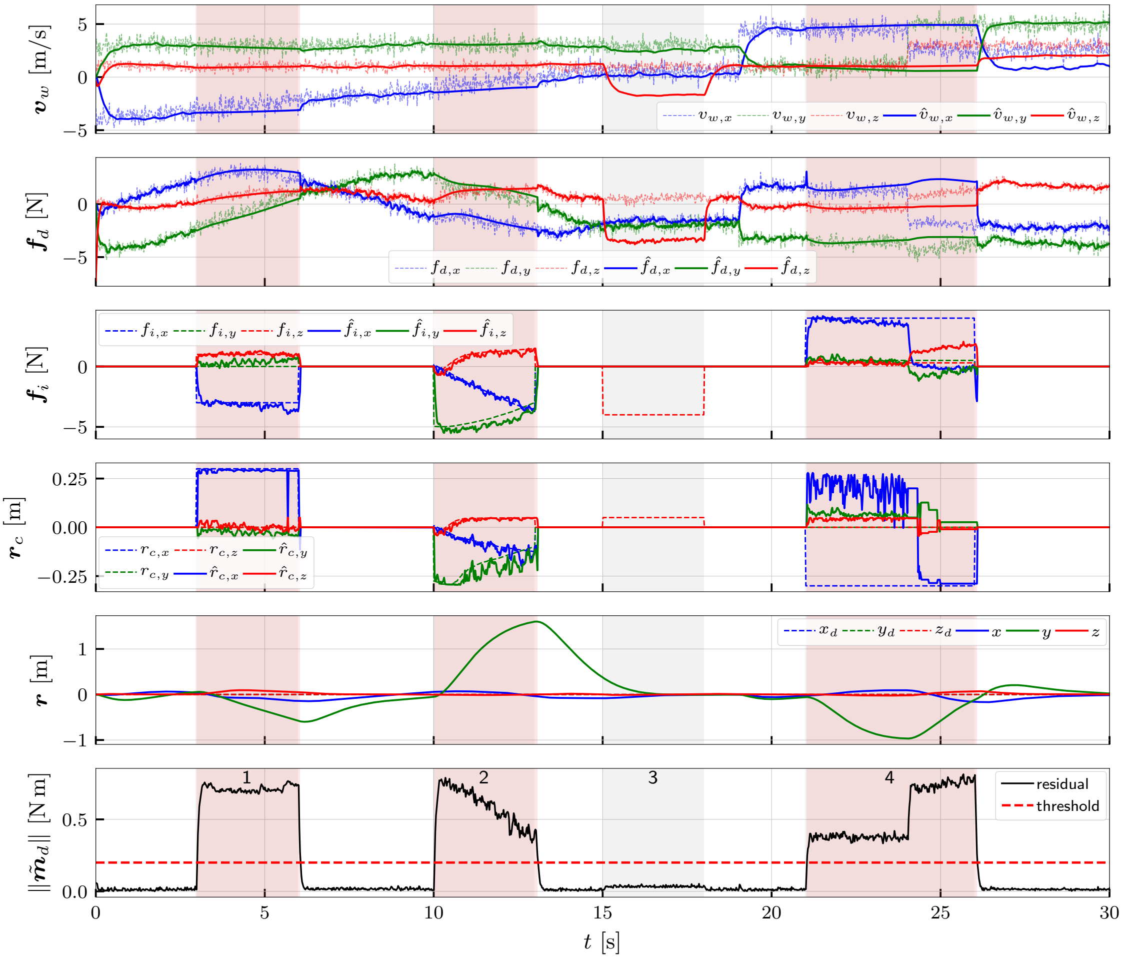
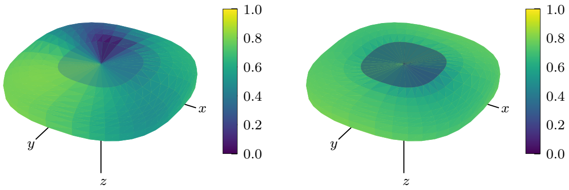


Figure 5.12: Simulation results for discrimination between aerodynamic and contact forces under wind influence, using exactly the residual based estimation scheme from Figure 5.11. The aerial robot is yawing in the time-varying wind field, while running the compensated impedance controller. From top to bottom: wind speed $\boldsymbol{v}_w$; robot position $\boldsymbol{r}$; aerodynamic force $\boldsymbol{f}_d$; contact force $\boldsymbol{f}_i$; contact position $\boldsymbol{r}_c$; torque residual $\tilde{\boldsymbol{m}}_e$. Periods where contact was detected are shaded in red.

the vehicle in the inertial frame while it is yawing under time-varying wind influence, see the robot position $\boldsymbol{r}$. The time-varying contact position is also correctly estimated in this case. There is additional noise due to the small error in the interaction force. In phase (3), no contact is detected because the purely vertical force does not exert a torque, as can also be seen in Figure 5.10. This is wrongly interpreted as a change in wind. Lastly, in phase (4), the wind speed exhibits a step change. The change from the initial aerodynamic force gets falsely interpreted as interaction, hence the scheme fails. The ambiguity in the contact position determination can also be seen here – the contact position is estimated on the wrong side of the convex hull (wrong sign of the $x-$axis) after the wind change. Notice also that the estimated interaction wrench does not intersect with the convex hull most of the time, since $\hat{\boldsymbol{r}}_c$ is held constant.

In summary, the force discrimination scheme presented in Figure 5.11 works well for scenarios where

- the wind speed is constant or slowly varying, and



(a) Linear model (b) Perceptron model

Figure 5.13: Observability of contact forces on a superellipsoidal convex hull, with different aerodynamic torque models. The scale is normalized to the smallest condition number being 1.0, and higher condition numbers going down to zero. Darker colors therefore indicate a higher condition number of the Jacobian at the contact position for this particular model. For the perceptron model the initial Jacobian at $\hat{\boldsymbol{f}}_d = \mathbf{0}$ is shown.

- the interaction force exhibits a torque sufficiently different from the aerodynamic torque.

Conversely, it will fail if

- the interaction force does not generate a torque residual, and/or
- the wind speed changes significantly during the interaction.

Note that this discrimination scheme does not assume a model of the interaction force. The contact position determination in the simulation result is obtained under the assumption of a point contact using (5.6).

5.6 Interaction force at known contact position

In the following, it is shown that given a known contact position, the interaction force can be computed by using the aerodynamic torque model, under the assumption that the interaction can be reduced to a point contact on the robot's convex hull. This assumption applies to applications such as slung load transportation or contact inspection. The location of the slung load may e.g. be known by mechanical design. Alternatively, the contact position may be initialized on collision with the inspected surface, when the signal-to-noise ratio allows for frequency-based discrimination.

When the contact position $\boldsymbol{r}_c$ is known, the external torque $\boldsymbol{m}_e$ may be written as $\boldsymbol{m}_e = \boldsymbol{m}_d + \boldsymbol{m}_i$ as

$$\boldsymbol{m}_e = \boldsymbol{m}_d\big(\boldsymbol{f}_e - \boldsymbol{f}_i\big) + \mathcal{S}(\boldsymbol{r}_c)\,\boldsymbol{f}_i, \tag{5.10}$$

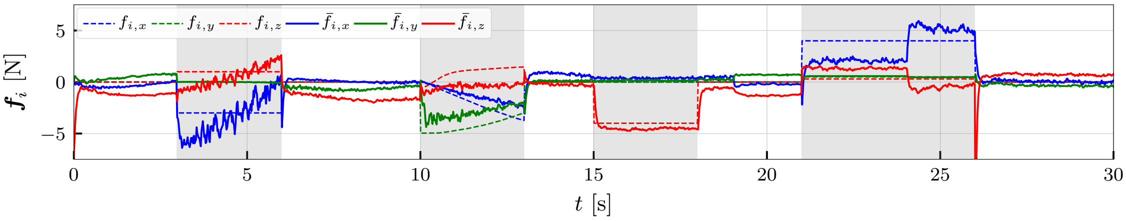
where $\boldsymbol{m}_d(\bullet)$ is the nonlinear aerodynamic torque model. In the nonlinear case, the nonlinear equation

$$\boldsymbol{F}_i = \boldsymbol{m}_d\big(\boldsymbol{f}_e - \boldsymbol{f}_i\big) + \mathcal{S}(\boldsymbol{r}_c)\,\boldsymbol{f}_i - \boldsymbol{m}_e = \mathbf{0}, \tag{5.11}$$

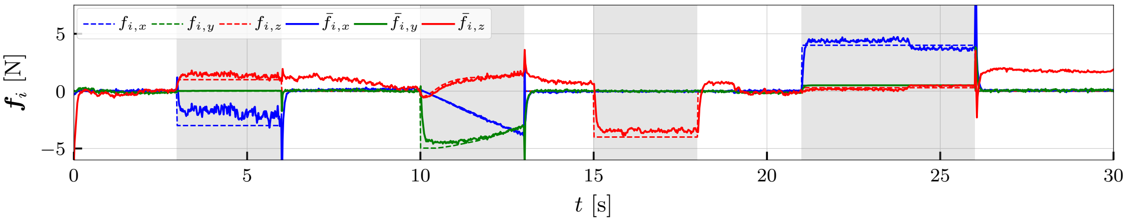
must be solved, which may be done using e.g. Levenberg-Marquardt. The Jacobian of (5.11) is

$$\boldsymbol{J}_i = \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{f}_i} = \mathcal{S}(\boldsymbol{r}_c) - \frac{\partial \boldsymbol{m}_d(\boldsymbol{f}_d)}{\partial \boldsymbol{f}_d} = \mathcal{S}(\boldsymbol{r}_c) - \boldsymbol{J}_d, \tag{5.12}$$

where $\boldsymbol{J}_d = \frac{\partial \boldsymbol{m}_d(\boldsymbol{f}_d)}{\partial \boldsymbol{f}_d}$ is the Jacobian of the aerodynamic torque model. In the special case



(a) Linear aerodynamic torque model trained with a regularization factor $\alpha_1 = 10^{-5}$, using the direct solution (5.13).



(b) Perceptron aerodynamic torque model (ground truth), using nonlinear optimization of (5.11), initialized with $\hat{\boldsymbol{f}}_i = \boldsymbol{f}_e$.

Figure 5.14: Force estimation at the exactly known contact position for the simulation in Figure 5.12, using different aerodynamic torque models. A perceptron aerodynamic torque model was used for simulation. The error in the estimated interaction force for the linear model is mainly caused by the aerodynamic torque modeling error, as the true aerodynamic wrench is modeled as a nonlinear perceptron model. Even with perfect model knowledge (b), the force is not reconstructed perfectly. The force spikes are caused by transients in the external wrench estimate.

of a linear model $\boldsymbol{m}_d(\boldsymbol{f}_d) = \boldsymbol{D}\boldsymbol{f}_d$, the solution may then be directly found as

$$\hat{\boldsymbol{f}}_i = \Big(\mathcal{S}(\boldsymbol{r}_c) - \boldsymbol{D}\Big)^{-1}\Big(\boldsymbol{m}_e - \boldsymbol{D}\boldsymbol{f}_e\Big) = \boldsymbol{J}_i^{-1}\tilde{\boldsymbol{m}}_d. \tag{5.13}$$

The observability of the interaction force will thus depend on the contact position and the aerodynamic model, as $\boldsymbol{J}_i$ must be nonsingular. Note also that (5.11) allows the solution for nonlinear torque models.

Figure 5.13 illustrates the interaction force observability on an example superellipsoidal hull and a linear aerodynamic torque identified in Section 4.3.5. Here, observability is defined as the condition number of $\boldsymbol{J}_i$. The color is normalized to the smallest condition number being brightest, and largest condition number (lowest observability) being darkest. Areas with a lower observability will be more susceptible to modeling errors and noise in the measurements. The solution (5.13) can also guide the contact point selection for best observability, Note also that the contact point does not have to be on the convex hull, but can also be the end of a tool attached to the flying robot. The model $\hat{\boldsymbol{m}}_d(\boldsymbol{f}_d)$ contains implicitly a (possibly varying) center of pressure, in other words the effective lever $\boldsymbol{r}_d$ of aerodynamic force $\boldsymbol{f}_d$. For our linear model, this is assumed to be constant, thus $\boldsymbol{D}$ is constant. This explains the strongly asymmetric behavior in Figure 5.13(a) – forces acting near to the constant center of pressure will be less observable due to the shorter lever to $\boldsymbol{r}_d$. Notice that the behavior tends towards more symmetry for the richer nonlinear model in Figure 5.13(b).

Figure 5.14 depicts the interaction force computed at the exact (simulated) contact position using a linear model and direct computation (5.13). The aerodynamic wrench in the simulation was obtained using a nonlinear model, and therefore using a linear model results in an estimation error. Furthermore, a delay in estimating the external wrench causes some transient errors, for example at $t = 0$ s, and $t = 26$ s. Note that the pure force interaction in contact phase 3 is correctly identified because the exact contact position is used. However, modeling errors are reflected in a state-dependent error of the estimated interaction force. The result shows that even with perfect knowledge of the contact position, under modeling errors, the method presented here is not sufficient to accurately obtain the interaction force.

5.7 Particle filter-based force discrimination

As shown in the previous subsection, even exactly knowing the contact force leads to inaccurate estimation of the interaction force when modeling errors are present. In this section a framework that fuses interaction force and wind speed models into a unified force discrimination scheme is proposed. It is based on directly estimating the contact position on the robot's convex hull by using a particle filter. By estimating the contact position that best matches the observed external wrench, the modeling error is essentially transferred from the interaction wrench to the estimated contact position. The latter is by definition bounded through the convex hull, which results in a more robust overall scheme. The scheme is based on a Sequential Importance Resampling (SIR) particle filter [34]. Each particle $\boldsymbol{x}_i = [\boldsymbol{r}_{c,i}^T \; \boldsymbol{v}_{w,i}]^T$ contains a contact position $\boldsymbol{r}_{c,i}$ on the convex hull CH, as well as a wind speed $\boldsymbol{v}_{w,i}$. Then (5.12) is used to obtain the interaction wrench $\boldsymbol{\tau}_{i,i}$ at the contact point, and an aerodynamic wrench model $\boldsymbol{\tau}_d\big(\boldsymbol{v}_r\big)$ at the relative airspeed computed from the particle's wind speed to obtain the particle aerodynamic wrench $\hat{\boldsymbol{\tau}}_{d,i}$. In summary,

$$\hat{\boldsymbol{\tau}}_{i,i} = \begin{bmatrix} \boldsymbol{J}_i^{-1}\tilde{\boldsymbol{m}}_d \\ \boldsymbol{r}_{c,i} \times \big(\boldsymbol{J}_i^{-1}\tilde{\boldsymbol{m}}_d\big) \end{bmatrix}, \tag{5.14}$$

$$\hat{\boldsymbol{\tau}}_{d,i} = \boldsymbol{\tau}_d\big(\boldsymbol{R}^T(\boldsymbol{v} - \boldsymbol{v}_{w,i})\big). \tag{5.15}$$

Based on $\hat{\boldsymbol{\tau}}_{e,i} = \hat{\boldsymbol{\tau}}_{d,i} + \hat{\boldsymbol{\tau}}_{i,i}$, the external wrench error $\tilde{\boldsymbol{\tau}}_{e,i}$ of particle i is defined as

$$\tilde{\boldsymbol{\tau}}_{e,i} = \boldsymbol{\tau}_e - \hat{\boldsymbol{\tau}}_{i,i} - \hat{\boldsymbol{\tau}}_d, \tag{5.16}$$

which is the error between the estimated external wrench and the wrench predicted by particle i. A Gaussian distribution of the particles around zero of the external wrench error is adopted. The probability of particle i is then

$$p\left(\hat{\boldsymbol{\tau}}_{e,i} | \boldsymbol{r}_{c,i}\right) = \exp\left(-\frac{1}{\sigma^2}\tilde{\boldsymbol{\tau}}_{e,i}^T\tilde{\boldsymbol{\tau}}_{e,i}\right), \tag{5.17}$$

where σ is a shape parameter. The algorithm is described in detail in Algorithms 1 and 2. The filter runs when contact is detected using the torque residual. The contact positions $\boldsymbol{r}_{c,i}$ are initialized uniformly on the convex hull, and assigned to the currently estimated wind speed. The process model for both, the contact position and wind speed, is constant with per-axis Gaussian noise. At each iteration, the contact position is projected onto the convex hull.

Convergence. Figure 5.15 depicts four iterations of the particle filter. The aerodynamic force is obtained from a nonlinear perceptron model, while the particle filter uses a linear model, which results in a modeling error. Already after the first iteration, the result is near the actual contact position and force. After resampling, particles are distributed around the actual contact position, as expected. Here, a portion of the particles is randomized

Algorithm 1: Contact particle filter for simultaneous estimation of contact and aerodynamic wrenches.

```
Input      : $\boldsymbol{\tau}_e$: External wrench
             $\boldsymbol{v}$: Translational velocity
             $\boldsymbol{R}$: Orientation
Output     : $\hat{\boldsymbol{\tau}}_i$: Estimated interaction wrench
             $\hat{\boldsymbol{\tau}}_d$: Estimated aerodynamic wrench
             $\hat{\boldsymbol{r}}_c$: Estimated contact position
             $\hat{\boldsymbol{v}}_w$: Estimated wind velocity
Models     : $CH$: Convex hull
             $\boldsymbol{\tau}_d(\boldsymbol{v}_r)$: Aerodynamic wrench model
             $\boldsymbol{m}_d(\boldsymbol{f}_d)$: Aerodynamic torque model
Variables  : $\boldsymbol{x} \in \mathbb{R}^{6 \times N_p}$: Particles
             $\boldsymbol{w} \in \mathbb{R}^{N_p}$: Particle weights
Parameters : $N_p$: Number of particles
             $N_r$: Resampling threshold
             $\boldsymbol{Q}_r$: Contact position noise
             $\boldsymbol{Q}_w$: Wind speed noise
             $\rho_{\text{rand}}$: Particle randomization ratio
ForceDiscrimination:
Initialize ()
repeat
    $k \leftarrow k + 1$
    ProcessModel ()
    for $i \leftarrow 1$ to $N_p$ do
        $\boldsymbol{v}_{r,i|k} \leftarrow \boldsymbol{R}^T \big(\boldsymbol{v} - \boldsymbol{v}_{w,i|k}\big)$
        $\boldsymbol{\tau}_{d,i|k} \leftarrow \boldsymbol{\tau}_d(\boldsymbol{v}_{r,i|k})$
        $\boldsymbol{\tau}_{i,i|k} \leftarrow$ InteractionWrench $\big(\boldsymbol{r}_{c,i|k}, \boldsymbol{\tau}_e\big)$
        $\tilde{\boldsymbol{\tau}}_{i|k} \leftarrow \boldsymbol{\tau}_e - \boldsymbol{\tau}_{i,i|k} - \boldsymbol{\tau}_{d,i|k}$
        $p\big(\boldsymbol{x}_{i|k}|\boldsymbol{x}_{i|k-1}, \boldsymbol{\tau}_e\big) = -\exp\big(-\frac{1}{2\sigma^2}\tilde{\boldsymbol{\tau}}_{i|k}^T \tilde{\boldsymbol{\tau}}_{i|k}\big)$
        $w_{i|k} \leftarrow w_{i|k-1}\, p\big(\boldsymbol{x}_{i|k}|\boldsymbol{x}_{i|k-1}, \boldsymbol{\tau}_e\big)$
    end
    $\bar{\boldsymbol{\tau}}_{i|k} = \sum_{i=1}^{N_p} w_{i|k} \boldsymbol{\tau}_{i,i|k}$
    $\bar{\boldsymbol{\tau}}_{d|k} = \sum_{i=1}^{N_p} w_{i|k} \boldsymbol{\tau}_{d,i|k}$
    $\bar{\boldsymbol{r}}_{c|k} = \sum_{i=1}^{N_p} w_{i|k} \boldsymbol{r}_{c,i|k}$
    $\hat{\boldsymbol{v}}_{w|k} = \sum_{i=1}^{N_p} w_{i|k} \boldsymbol{v}_{w,i|k}$
    Resample ()
until $\|\boldsymbol{m}_d\big(\boldsymbol{f}_e\big) - \boldsymbol{m}_e\| \leq \delta$;
```

after each iteration as an "exploration" step. Iterating further, as expected, particles are still concentrated around the contact position, and variance of the contact force direction and magnitude is low. This shows that the particle filter can converge to the actual contact position and force even with modeling errors. Conversely, Figure 5.16 shows a degenerate case where the filter can fall into a local minimum. As discussed in Section 5.3, the same wrench can be generated by a force on opposite sides of the convex hull, see Figure 5.16. Notice the large variance of the contact force direction and magnitude.

Discrimination scheme. The particle filter is one component in the discrimination scheme depicted in Figure 5.17 In this scheme, the time constant of the wind estimator is not adapted. Instead, the interaction wrench $\hat{\boldsymbol{\tau}}_i$ is estimated directly. The estimated aerodynamic wrench $\hat{\boldsymbol{\tau}}_d = \boldsymbol{\tau}_e - \hat{\boldsymbol{\tau}}_i$ is then used directly to obtain an estimated relative

Algorithm 2: Auxiliary functions of the particle filter.

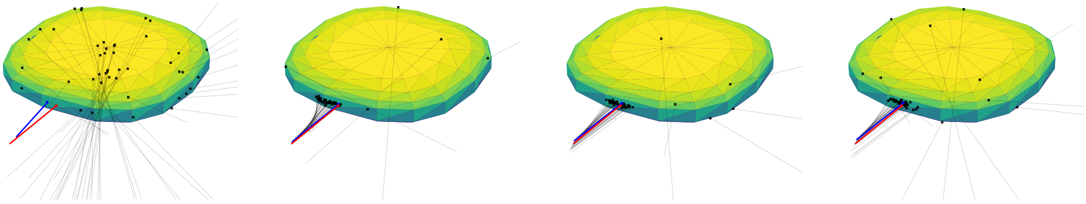
```
Initialize:
$k \leftarrow 0$
for $i \leftarrow 1$ to $N_p$ do
    $\boldsymbol{r}_{c,i|0} \leftarrow$ UniformSampleConvexHull $(CH)$
    $\boldsymbol{v}_{w,i|0} \leftarrow \boldsymbol{v}_{w|0}$
    $\boldsymbol{x}_{i|0} \leftarrow \left[\boldsymbol{r}_{c,i|0}^T \; \boldsymbol{v}_{w,i|0}^T\right]^T$
    $w_i = \frac{1}{N_p}$
end

ProcessModel:
for $i \leftarrow 1$ to $N_p$ do
    if UniformRandomNumber $() < 1 - \rho_{\mathrm{rand}}$ then
        Contact position noise model
        $\bar{\boldsymbol{r}}_{c,i|k-1} \leftarrow \boldsymbol{r}_{c,i|k-1} + \boldsymbol{Q}_r \cdot$ GaussianRandom $()$
    else
        Randomize contact position
        $\bar{\boldsymbol{r}}_{c,i|k-1} \leftarrow$ RandomSampleConvexHull $(CH)$
    end
    $\boldsymbol{r}_{c,i|k} \leftarrow$ ProjectToConvexHull $(\bar{\boldsymbol{r}}_{c,i|k-1}, CH)$
    $\boldsymbol{v}_{w,i|k} \leftarrow \boldsymbol{v}_{w|k-1} + \boldsymbol{Q}_w \cdot$ GaussianRandom $()$
end

Resample:
$N_{\mathrm{eff}} = 1/\left(\sum_i w_i^2\right)$ Number of effective samples
if $N_{\mathrm{eff}} < N_r$ then
    $\gamma =$ cumsum $(\boldsymbol{w})$
    for $i \leftarrow 1$ to $N_p$ do
        /* Uniformly draw new sample index $m$ from the cumulative distribution in $\gamma$ */
        $\rho =$ UniformRandomNumber $()$
        $m =$ first $j$ for $\gamma_j < \rho,\ j \in 1 \ldots N_p$
        $\boldsymbol{x}_{i|k} = \boldsymbol{x}_{m|k-1}$
        $w_i = \frac{1}{N_p}$
    end
end
```

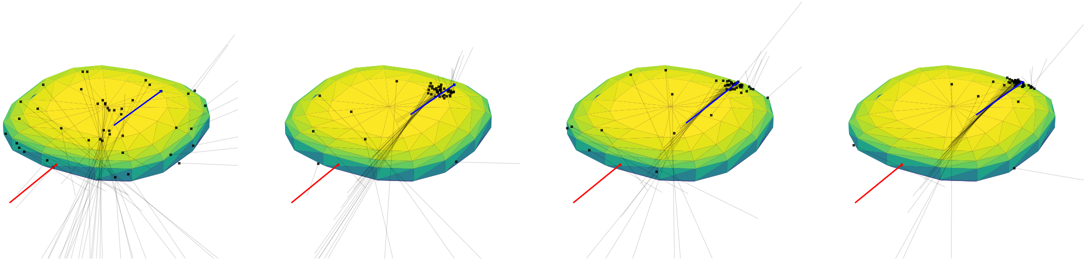
airspeed. The raw wind speed is filtered in the same manner as in the previous discrimination scheme. Note, however, that estimation of the wind speed does not need to be slowed down.

Results. To test particle filter based discrimination, the same simulation scenario as for the residual based discrimination was used, with $N_p = 45$ particles, contact position noise $\boldsymbol{Q}_r = \mathrm{diag}\{0.025, 0.025, 0.005\}$ m, and wind speed noise $\boldsymbol{Q}_w = 0.001\boldsymbol{I}_{3\times3}$ m/s. The results are shown in Figure 5.18. The position, aerodynamic force, and residual plots are omitted as they are comparable. The notable difference to Figure 5.12 is that the contact position fluctuates more. This behavior can be tuned through $\boldsymbol{Q}_r$. In phase (4), the contact position estimation is stable even after the wind speed changes abruptly. However, the filter converges to the wrong side of the convex hull during the contact (see $\hat{r}_{c,x}$), but switches back to the correct side after a short period. This is due to the ambiguity of the contact determination problem discussed above. In this simulation, wind speed estimation is not slowed down during contact. Instead, the particle filter outputs the estimated interaction wrench $\hat{\boldsymbol{\tau}}_i$, and the resulting $\hat{\boldsymbol{\tau}}_d$ is used to estimate wind speed. Overall, the results for this simulation scenario are not significantly better than the modified model-checking



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

Figure 5.15: Particle filter convergence with a linear model, while the aerodynamic wrench is obtained with a nonlinear model. Black points represent candidate contact positions on the convex hull (particles). The estimated interaction force at that position is represented as a line. The red line depicts the ground truth interaction force, while the blue line represents the output of the particle filter (contact position and force).



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

Figure 5.16: Example when particles converge to the wrong side of the convex hull, estimating the wrong contact position. This is due to the ambiguity of the contact determination problem: the same torque can be achieved by the force on both sides of the convex hull. The force direction and magnitude are still estimated correctly. Note the high variability of the interaction force around the sample mean.

estimation scheme, because it is based on the same principle. However, this framework may be used to more accurately determine the contact position, given an interaction wrench.

Extensions. Note that this algorithm can be extended to also include the interaction force $\boldsymbol{f}_i$ in the particle state. However, that further increases the dimensionality of the problem. Lastly, wind speed may be omitted from the particle state to simplify the filter

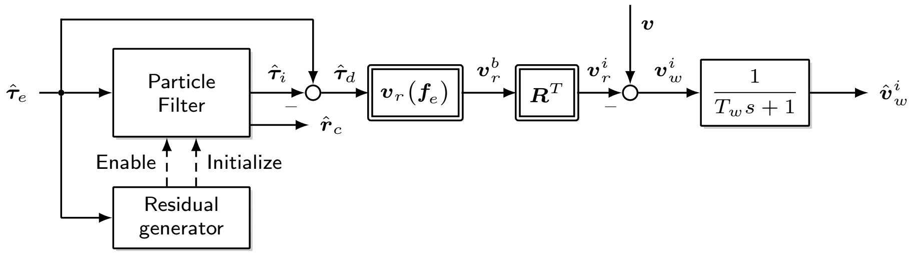


Figure 5.17: Input discrimination and wind estimation scheme based on the particle filter. The time constant of the wind speed filter is not adapted. Instead, the particle filter outputs the estimated interaction wrench $\hat{\boldsymbol{\tau}}_i$ only when contact is detected by the residual generator. Then, the drag wrench $\hat{\boldsymbol{\tau}}_d$ is subsequently used to obtain the relative airspeed.

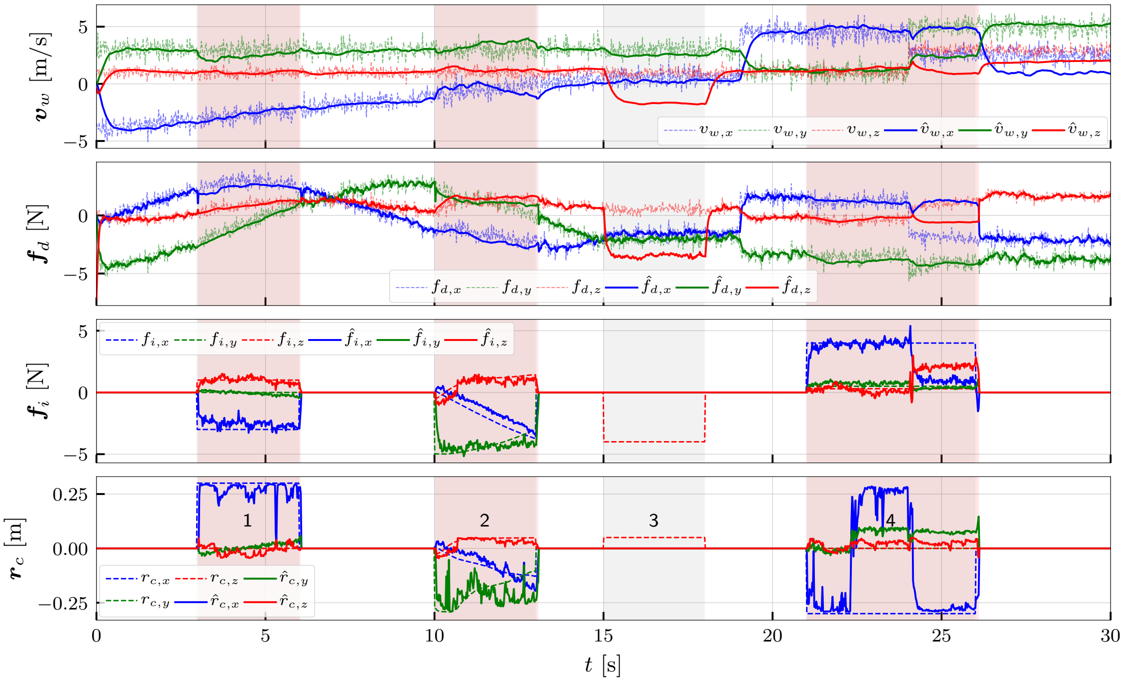


Figure 5.18: Discrimination between aerodynamic and contact forces under wind influence, using the particle filter based approach. The aerial robot is yawing in the time-varying wind field, while running the compensated impedance controller. From top to bottom: wind speed $\boldsymbol{v}_w$; robot position $\boldsymbol{r}$; aerodynamic force $\boldsymbol{f}_d$; contact force $\boldsymbol{f}_i$; contact position $\boldsymbol{r}_c$; torque residual $\tilde{\boldsymbol{m}}_e$. Periods where contact was detected are shaded in red.

to only determine the contact position.

5.8 Power-based discrimination

The previously presented force discrimination schemes rely on the aerodynamic torque residual for contact detection. In Section 4.3 and Section 4.4 we have shown that the airspeed may be obtained also by using the aerodynamic power instead of the external wrench. In this section, a force discrimination scheme that exploits this fact is proposed, see Figure 5.19.

First, an estimate of the aerodynamic power $\hat{\boldsymbol{P}}_a$ is obtained from the motor current $\boldsymbol{i}_a$, motor speed $\boldsymbol{\varpi}$, and motor acceleration $\dot{\boldsymbol{\varpi}}$ as described in Section 2.1.5. Second, a raw body-frame airspeed measurements is obtained by employing a model $\boldsymbol{v}_r(\boldsymbol{P}_a)$, identified in Section 4.3. This is converted into an inertial-frame wind velocity and filtered using an appropriate time constant T_w. The aerodynamic wrench $\hat{\boldsymbol{\tau}}_d$ is then obtained from the aerodynamic model $\boldsymbol{\tau}_d(\boldsymbol{v}_r)$, identified in Section 4.3. The estimated contact wrench and position are computed as in the modified model-checking scheme. Note that if the airspeed was estimating directly in the body frame, the accuracy would depend on the trajectory, i.e. the filter time constant would have to change with the movement of the robot, for example during the yawing motion in phases 1 and 2 of the simulation. Estimating the wind speed in the inertial frame is therefore crucial because of the that is the relevant quantity which is slowly time-varying in the inertial frame.

The scheme is tested for the same simulation scenario as previous schemes, and results are shown in Figure 5.20. In the simulation, the aerodynamic power of coaxial rotor pairs was used, and Gaussian noise with a standard deviation of 8 W was added to the power

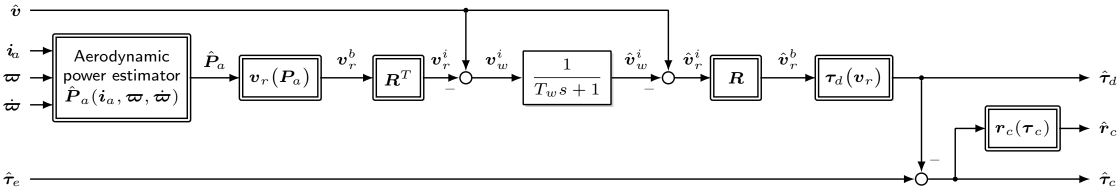

$\hat{\boldsymbol{v}}$
$\boldsymbol{i}_a$
$\boldsymbol{\varpi}$
$\dot{\boldsymbol{\varpi}}$
Aerodynamic power estimator $\hat{\boldsymbol{P}}_a(\boldsymbol{i}_a, \boldsymbol{\varpi}, \dot{\boldsymbol{\varpi}})$
$\hat{\boldsymbol{P}}_a$
$\boldsymbol{v}_r(\boldsymbol{P}_a)$
$\boldsymbol{v}_r^b$
$\boldsymbol{R}^T$
$\boldsymbol{v}_r^i$
$\boldsymbol{v}_w^i$
$\frac{1}{T_w s+1}$
$\hat{\boldsymbol{v}}_w^i$
$\hat{\boldsymbol{v}}_r^i$
$\boldsymbol{R}$
$\hat{\boldsymbol{v}}_r^b$
$\boldsymbol{\tau}_d(\boldsymbol{v}_r)$
$\hat{\boldsymbol{\tau}}_d$
$\boldsymbol{r}_c(\boldsymbol{\tau}_c)$
$\hat{\boldsymbol{r}}_c$
$\hat{\boldsymbol{\tau}}_e$
$\hat{\boldsymbol{\tau}}_c$
−

Figure 5.19: Aerodynamic power based force discrimination scheme. The motor current i_a, speed ϖ and angular acceleration $\dot{\varpi}$ are used to obtain the aerodynamic power $\hat{\boldsymbol{P}}_a$. The airspeed is then obtained by a model $\boldsymbol{v}_r(\boldsymbol{P}_a)$, as learned in Section 4.3.

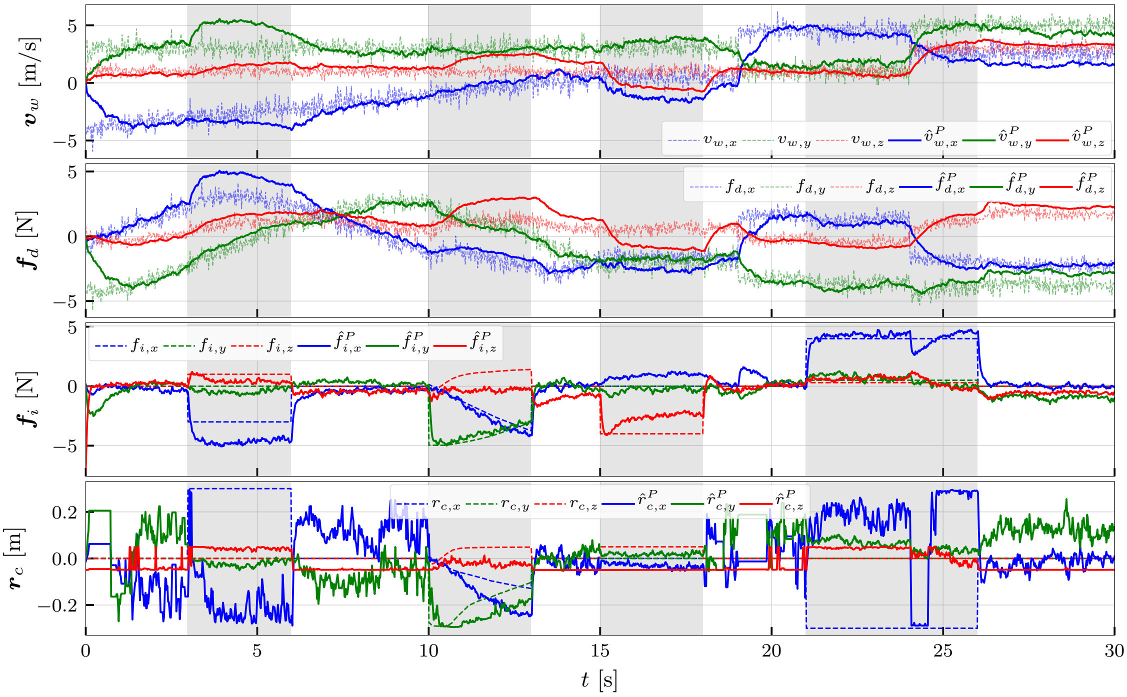


Figure 5.20: Simulation results for power based force discrimination, for the same case as in Figure 5.12. Periods of contact are shaded in gray. Explicit contact detection is not needed in this scheme.

measurements. The wind estimation time constant was $T_w = 0.2$ s. The resulting wind speed and therefore aerodynamic wrench exhibit bias errors in some cases. Note that, contrary to the above discrimination schemes, there is no explicit slowing down of the wind estimation. There is also no explicit contact detection, as the scheme runs continuously. This causes some systematic errors. For example, during the first contact phase the force is falsely interpreted as $x-$axis airspeed. This is probably because the underlying model fitted from wind tunnel data has partly also fitted the external force. The same would happen in the above schemes if wind speed estimation were not slowed down. The power based scheme performs better in the other contact phases, despite the same systematic errors. Notably, where the scheme successfully detects the interaction force is coming from above in the third phase, which is a failure case of residual based methods. Finally, in the third contact phase, the interaction force is correctly identified despite the change in wind speed during the contact. This is another failure case of residual based methods.

It can be conclude that the motor power may be used to provide an independent measurement of the aerodynamic wrench, which can then successfully be applied to discriminate between interaction and aerodynamic wrenches. The accuracy of the method

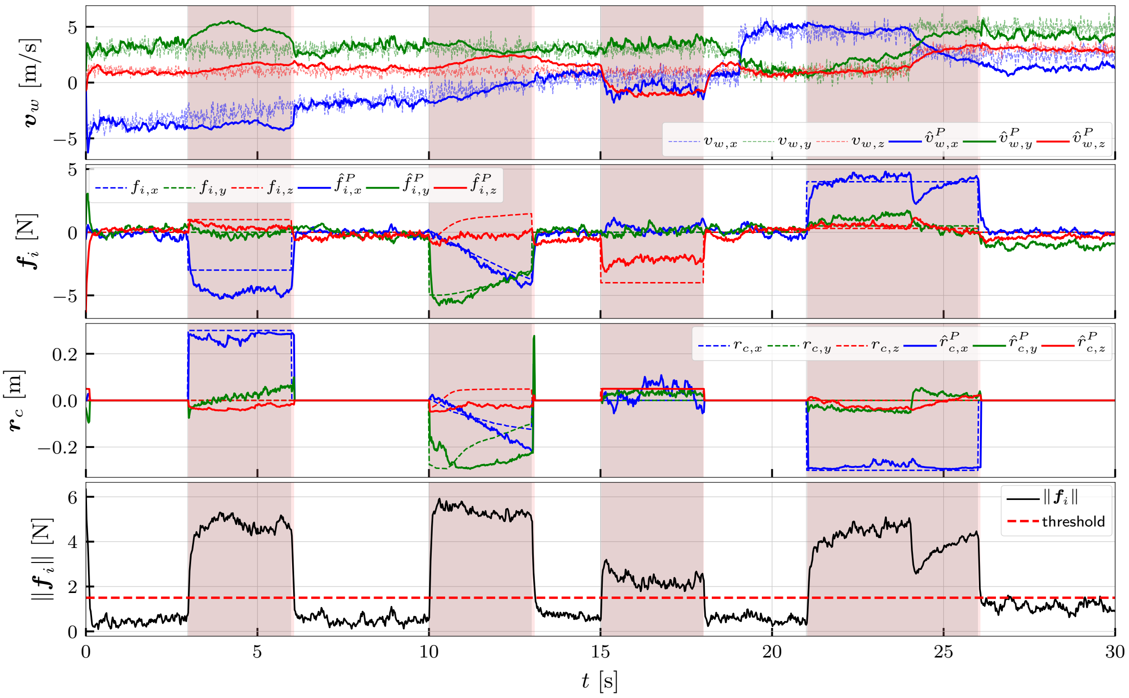


Figure 5.21: Simulation results for Kalman filter-based force discrimination, which combines airspeed obtained from external forces, and airspeed obtained from power. Red shaded areas indicate periods during which the filter has detected contact.

will depend on measurement noise of the motor current and speed; the fit from the motor power to propeller aerodynamic power; the aerodynamic model mapping aerodynamic power to airspeed ; and the aerodynamic model mapping airspeed to the aerodynamic wrench. Therefore, the difficulty in applying this scheme lies in the effort to obtain these models accurately, as modeling errors propagate through the estimation chain.

5.9 Kalman filter-based force discrimination

The modified model checking method fails when a pure force is applied to the vehicle and when wind speed changes during contact. On the other hand, the power based discrimination can run continuously and can detect pure forces, but is prone to modeling errors (offset). Neither method performs robustly for estimating the contact position directly from the estimated interaction wrench, especially if the wind speed changes during contact. The particle filter performs favorably for this task, as it is designed to estimate the contact position. Our aim is to combine the strengths of the previously proposed discrimination schemes into a unifying framework. A Kalman filter is an established tool for such sensor fusion. In the filter the wind speed is estimated directly, i.e. the state is $\boldsymbol{x} = \boldsymbol{v}_w$. The process model is a constant with Gaussian noise $\boldsymbol{Q} = Q_w \boldsymbol{I}_{3\times3}$. The measurements $\boldsymbol{z}$ are the instantaneous model-based estimates of the wind speed

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{v} - \boldsymbol{R}\boldsymbol{v}_r(\bar{\boldsymbol{f}}_d) \\ \boldsymbol{v} - \boldsymbol{R}\boldsymbol{v}_r(\bar{\boldsymbol{P}}_a) \end{bmatrix}, \tag{5.18}$$

where $\bar{\boldsymbol{f}}_d$ is to be defined below. The measurement matrix $\boldsymbol{H}$ and measurement covariance $\boldsymbol{R}_z$ are simply

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_{3\times3} \\ \boldsymbol{I}_{3\times3} \end{bmatrix}, \quad \boldsymbol{R}_z = \begin{bmatrix} R_f \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & R_p \boldsymbol{I}_{3\times3} \end{bmatrix}. \tag{5.19}$$

The implementation details of the Kalman filter are omitted, as they can be found in any textbook on the subject. Because the estimated interaction force from the power based method can estimate a pure interaction force, the additional contact detection signal

$$CD_2 = \begin{cases} 1 & \text{if } \|\hat{\boldsymbol{f}}_i\| > \delta_f, \\ 0 & \text{otherwise}, \end{cases} \tag{5.20}$$

is added, where δ_f will depend on the aerodynamic modeling error and state (e.g. velocity). The new contact detection signal for the Kalman filter is $CD_K = CD_1 \vee CD_2$, i.e. if either of the contact detection signals are true. The particle filter is initialized when contact is detected. During contact, *only* the contact position estimate of the particle filter is used. It is obtained by applying the Kalman-filter estimated aerodynamic wrench to evaluate the sample probability density function. This leads to a more robust contact position estimate than based purely on the torque residual, as it fuses information from multiple sources.

A discrete-time Kalman filter was simulated with a time step of $T = 0.02$ s, and used perceptron models for both the aerodynamic power and external force based airspeed estimation. Similar to the modified model-checking method, filter parameters are changed when a contact detection signal CD_K is true. When not in contact, the parameters

$$\begin{aligned} Q_w &= 3.6 \cdot 10^{-7}, & R_f &= 1.6 \cdot 10^{-7}, \\ R_p &= 3.6 \cdot 10^{-5}, & \bar{\boldsymbol{f}}_d &= \boldsymbol{f}_e \end{aligned}$$

are used, and when $CD_1 = 1$, and the parameters

$$\begin{aligned} Q_w &= 9.0 \cdot 10^{-10}, & R_f &= 1.6 \cdot 10^{-7}, \\ R_p &= 1.0 \cdot 10^{-6}, & \bar{\boldsymbol{f}}_d &= \boldsymbol{f}_d\big(\boldsymbol{R}^T(\boldsymbol{v} - \hat{\boldsymbol{x}})\big) \end{aligned}$$

are used. In essence, this mimics the behavior of the modified model checking methods where wind estimation is slowed down when contact is detected. In contact, the measurement covariance of the power based estimate is decreased to take it into account more strongly. The particle filter is run with $N_p = 20$ particles, and contact position noise $\boldsymbol{Q}_r = \mathrm{diag}\{1.25, 1.25, 0.25\} \cdot 10^{-2}$ m. The particles are initialized uniformly across the convex hull once contact is detected. Note that these parameters can be tweaked to fine-tune the overall behavior. The optimization-based estimation of the vertical airspeed component was omitted, as it did not provide substantially better performance than the data-driven model. However, this information can also be easily included in the filter if needed. Figure 5.21 shows the resulting estimated wind speed, interaction force, and contact position. Note that the result is largely similar to the power based estimation. However, in the periods without interaction, the offset of the interaction force is lower. A negative side effect is that the pure force interaction in phase 3 is underestimated in this scheme, due to the filter also trusting the force based measurement. In phase 4, the filter can maintain the same discrimination performance even if the wind speed rapidly changes. Finally, because the contact position is estimated by the particle filter, it exhibits more stable behavior than computing it from the estimated interaction wrench directly. Note that it is still prone to switching the side of convex hull during contact. This effect can be prevented by stronger filtering, however that is out of scope of this paper. In summary,

Table 5.2: Overview of pipeline stages for simultaneous collision handling and wind estimation, in the context of a Fault Detection, Identification and Isolation pipeline [55].

Sec.	Pipeline stage	Algorithm	Input	Required models
5.2	Detection	$CD = H(\lvert\hat{f}_{e,i}\rvert, \omega_f) > f_{c,i}$	$\boldsymbol{f}_e$	Fault isolation frequency ω_f, threshold $f_{c,i}$
5.3	Isolation	$\boldsymbol{r}_c = \boldsymbol{o} + k\boldsymbol{d}$	$\boldsymbol{\tau}_e$	Convex hull
5.4	Detection	$CD_1 = \lVert\tilde{\boldsymbol{m}}_d\rVert > \delta$	$\boldsymbol{\tau}_e$	Aerodynamic torque model $\boldsymbol{m}_d(\boldsymbol{f}_d)$
5.5	Isolation, Identification	$\hat{\boldsymbol{\tau}}_i$, $\hat{\boldsymbol{\tau}}_d, \hat{\boldsymbol{v}}_w, \hat{\boldsymbol{r}}_c \leftarrow$ Figure 5.11	$\boldsymbol{\tau}_e$	Aerodynamic torque model $\boldsymbol{m}_d(\boldsymbol{f}_d)$, wind speed propagation model
5.6	Identification	$\boldsymbol{f}_i = \boldsymbol{J}_i^{-1}\tilde{\boldsymbol{m}}_d$	$\boldsymbol{\tau}_e, \boldsymbol{r}_c$	Aerodynamic torque model $\boldsymbol{m}_d(\boldsymbol{f}_d)$, interaction position $\boldsymbol{r}_c$
5.7	Isolation, Identification	$\hat{\boldsymbol{\tau}}_i$, $\hat{\boldsymbol{\tau}}_d, \hat{\boldsymbol{v}}_w, \hat{\boldsymbol{r}}_c \leftarrow$ Algorithms 1 and 2, Figure 5.17	$\boldsymbol{\tau}_e$	Aerodynamic torque model $\boldsymbol{m}_d(\boldsymbol{f}_d)$, convex hull, contact position model
5.7	Isolation, Identification	$\hat{\boldsymbol{\tau}}_i$, $\hat{\boldsymbol{\tau}}_d, \hat{\boldsymbol{v}}_w, \hat{\boldsymbol{r}}_c \leftarrow$ Algorithms 1 and 2 with $\hat{\boldsymbol{v}}_w$ in particle state	$\boldsymbol{\tau}_e$	Aerodynamic torque model $\boldsymbol{m}_d(\boldsymbol{f}_d)$, aerodynamic wrench model $\boldsymbol{\tau}_d(\boldsymbol{v}_r)$, convex hull, contact position model, wind speed propagation model
5.8	Isolation, Identification	$\hat{\boldsymbol{\tau}}_d := \boldsymbol{\tau}_d\Big(\boldsymbol{v}_r(\hat{\boldsymbol{P}}_a)\Big)$, $\hat{\boldsymbol{\tau}}_i, \hat{\boldsymbol{v}}_w, \hat{\boldsymbol{r}}_c \leftarrow$ Figure 5.19	$\boldsymbol{\tau}_e, \bar{\boldsymbol{P}}_a$	Airspeed model $\boldsymbol{v}_r(\boldsymbol{P}_a)$, convex hull, aerodynamic power model $\hat{\boldsymbol{P}}_a(\boldsymbol{P}_m)$, aerodynamic wrench model $\boldsymbol{\tau}_d(\boldsymbol{v}_r)$
5.9	Identification	Kalman filter with measurement vector (5.18) and matrices (5.19)	$\boldsymbol{\tau}_e, \bar{\boldsymbol{P}}_a$	Aerodynamic torque model $\boldsymbol{m}_d(\boldsymbol{f}_d)$, Airspeed model $\boldsymbol{v}_r(\hat{\boldsymbol{P}}_a)$, aerodynamic power model $\hat{\boldsymbol{P}}_a(\boldsymbol{P}_m)$, aerodynamic wrench model $\boldsymbol{\tau}_d(\boldsymbol{v}_r)$
5.9	Detection	$CD_2 = \lVert\hat{\boldsymbol{f}}_i\rVert > \delta_f$	$\hat{\boldsymbol{f}}_i$	Threshold δ_f

the Kalman filter is a framework that can easily combine the two developed methods to provide quite robust discrimination between aerodynamic and contact wrenches.

5.10 Summary

Table 5.2 provides an overview of the novel force discrimination methods developed in this paper, in the context of a Fault Detection, Identification and Isolation pipeline, see Haddadin et al. [55]. Three different *detection* signals may be used. The collision detection signal CD detects collisions based on frequency. The contact detection signal CD_1 detects contacts based on the external torque. Finally, the contact detection signal CD_2 detects contacts based on the estimated interaction force. *Isolation* in this context means obtaining the contact position on the robot's convex hull. This may be achieved by raycasting as discussed in Section 5.3, or by a particle filter as discussed in Section 5.7. Finally, the *identification* stage reconstructs the constituent wrenches $\boldsymbol{\tau}_i$ and $\boldsymbol{\tau}_d$. Two distinct approaches are used for this purpose. In the modified model-checking scheme, Section 5.5, wind estimation is slowed down when contact is detected. In the power-based scheme, Section 5.8, the wind speed is obtained from the motor power, which is a measurement independent of the external wrench. Finally, it was shown that these two schemes may be successfully combined in a Kalman filter framework.

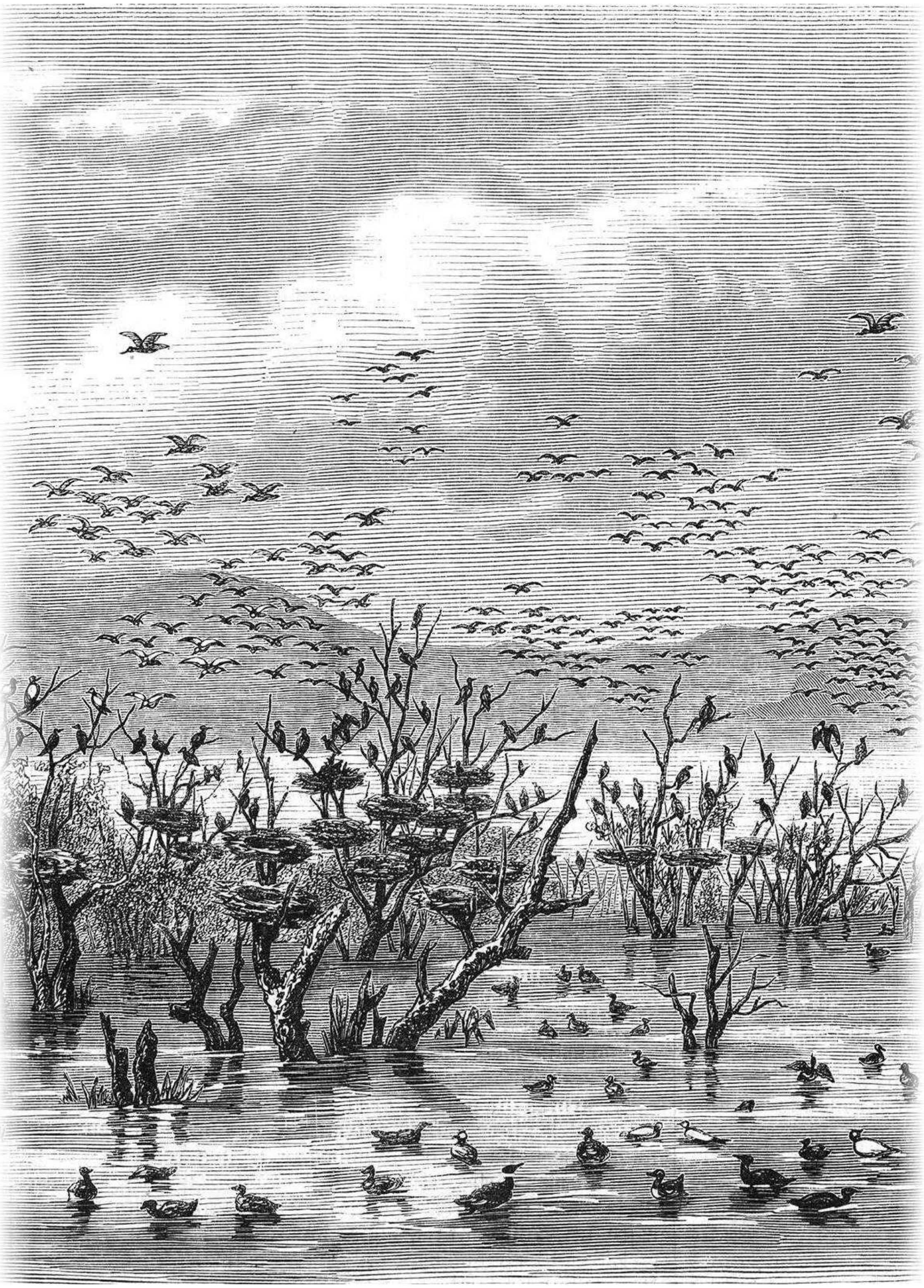


Illustration (excerpt): M. Lucy: Le lac Fetzara, en Afrique; Book: L'Illustration (Issue 755), Paris: Dubochet et Cie, 1857

CHAPTER 6

Applications and outlook

This chapter presents applications of the collision detection schemes from Chapter 5, and a high level research outlook for the methods developed in this theseis. Collision detection applications are presented in Section 6.1, namely collision reflexes and collision location determination in Section 6.1.1, and tactile mapping as part of an autonomy stack of a flying robot in Section 6.1.2.

In Section 6.2, the goals and research question posed in this thesis are extrapolated towards the goal of interaction, disturbance and fault-aware flying robot swarms. It is argued that robust operation of interacting flying robots requires systematic handling of interactions and external inputs from individual robot to swarm level. For this, a scalable methodology for interaction, disturbance and fault handling is introduced, resulting in an awareness pipeline scheme that can be applied to robot swarms. Another algorithmic key element for unification is the extension of well established methods from operational space and multipriority robot control to this system class, potentially leading to novel controls and skills of flying robot swarms.

6.1 Collision detection applications

During flight in unknown environments, collisions with the environment might occur due to unfavorable sensing conditions. In these cases, it is necessary to have a low-level collision detection and reflex strategy to minimize harm to the robot and its environment. Next, we explore applications of the collision detection methods developed in Chapter 5.

6.1.1 Collision reaction and location

Several basic collision reflex reaction strategies are investigated next. A *reflex* can be described as an "involuntary" action that does not involve active planning. The x-axis trajectory of a quadrotor colliding with a stiff surface (polycarbonate) is depicted in Figure 6.1. A Hunt-Crossley surface contact model was used, with parameters taken from [33]. The collision occurs at a velocity of 1 m/s, and the surface is located at $x = 1$ m. The quadrotor arm is 27 cm in length and acts as an offset. Depending on the desired robot behavior, the following reactions are investigated: *no reaction*, *trajectory stop*, *equilibrium bounce* and *collision force amplification.*

No reaction. The collision is not considered, hence the quadrotor becomes unstable and may crash.

Trajectory Stop. Upon collision, the trajectory is stopped at the current reference position. This halts the robot approximately at the obstacle position.

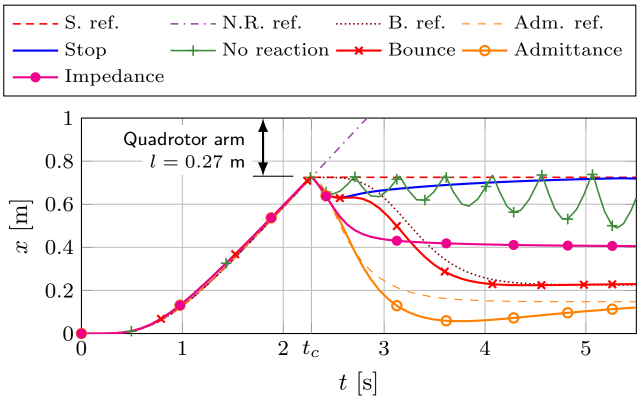


Figure 6.1: Simulated robot trajectories for different collision reaction strategies. Copyright © 2014 IEEE [141, 138].

Equilibrium Bounce. Upon detecting a collision, the equilibrium position is set to $\boldsymbol{r}_d = \boldsymbol{r}_k + \delta_c \hat{\boldsymbol{f}}_e^0$ where $\boldsymbol{r}_k$ is the robot position at the time of collision, and $\delta_c > 0$ is the bounce distance. This will drive the path backwards along the collision normal. The robot will then stabilize at a remote and safe distance from the obstacle.

Collision force amplification. Immediately upon collision, the controller is instantaneously switched to impedance or admittance mode. Only damping is performed, i.e. $\boldsymbol{K}_v = \boldsymbol{0}$. The input to the control system becomes the amplified collision force $\boldsymbol{\tau}_v = \boldsymbol{K}_r \hat{\boldsymbol{\tau}}_e$. For admittance control we then have

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}}_v + \boldsymbol{D}_v \dot{\boldsymbol{x}}_v = \boldsymbol{K}_r \hat{\boldsymbol{\tau}}_e, \tag{6.1}$$

and for impedance control

$$\boldsymbol{J}^T \boldsymbol{\tau} = (\boldsymbol{M}\boldsymbol{M}_v^{-1} - \boldsymbol{I})\boldsymbol{K}_r \hat{\boldsymbol{\tau}}_e - \boldsymbol{M}\boldsymbol{M}_v^{-1}\boldsymbol{D}_v \tilde{\boldsymbol{\nu}} + \boldsymbol{N}. \tag{6.2}$$

This essentialy produces an energy dissipation reflex. The reaction is in both cases significantly faster than trajectory-based approaches. Note that the amplified wrench $\boldsymbol{\tau}_v$ acting on the robot should be saturated to prevent instability.

Results from Figure 6.1 show that without reaction, the robot might become unstable. The system mass in the simulation was $\mathcal{M} = 0.55$ kg, and $\mathcal{M}_v = 0.25$ kg. Damping during admittance control is $d_{v,x} = 3$ Ns/m, and during impedance control $d_{v,x} = 0.25$ Ns/m. A stop reaction halts the robot at the collision position. A bounce reaction with $\delta_c = 0.5$ m that sets the equilibrium position in the opposite direction of the collision normal is effective in providing a smooth and deterministic response. The interaction controllers ($K_{r,x} = 1$) show very fast reaction.

To compare the reactions experimentally, a polystyrene block was placed on the robot's flight path and used it as an obstacle. Figure 6.2 depicts the position response of different collision reaction schemes along the inertial x-direction. Video stills of the experiment can be seen in Figure 6.3. The obstacle position did not change between the experiments. The highpass filtered external force was used as the collision detection signal.

From top to bottom: position, estimated external force, estimated external pitch torque, collision detection signal are shown. The force estimate is filtered with $K_I^f = 3$, and the torque with $K_I^m = 12$. The collision detection threshold was set to $f_c = 0.6$. The parameters $\mathcal{M}_v = 0.25$ kg, $d_v = 0.5$ Ns/m, and $K_{r,x} = 1$ were used. Collisions are detected at $\approx t_c$. The red shaded area depicts the contact phase. It can be concluded that even

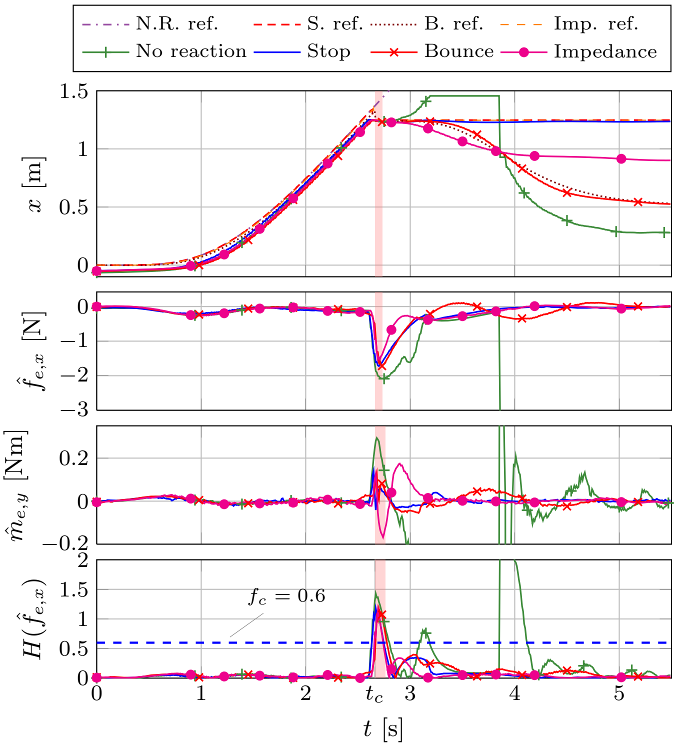


Figure 6.2: Collision reaction experiment with a polystyrene block. Copyright © 2014 IEEE [141, 138].

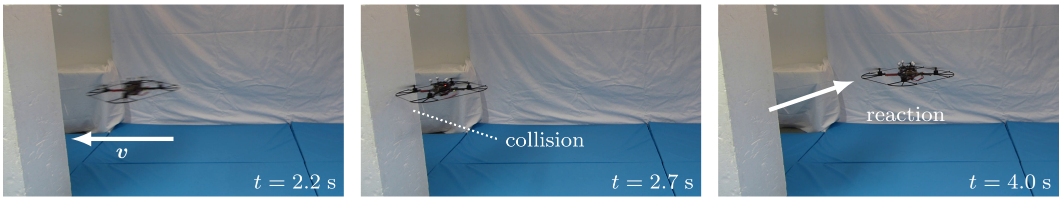


Figure 6.3: Video stills of the equilibrium bounce collision reaction experiment depicted in Figure 6.2. From left to right: shortly before impact ($t = 2$s), at impact ($t = 2.7$s), and stable state after the impact ($t = 4$s). Copyright © 2017 IEEE [138].

for a highly filtered (delayed) external force, the presented schemes are still effective. The quadrotor crashed when the collision was not accounted for, i.e. without a reaction scheme. The trajectory-based reaction schemes (stop, bounce) successfully stabilized the quadrotor after a collision, while switching to impedance damping mode provided the fastest and smoothest response to the collision. The estimated obstacle locations listed in Table 6.1 show that the obstacle plane estimation is precise and consistent across experiments.

Takeoff, landing, and multiple collisions

Figure 6.4 shows a complete flight with multiple *bounce* collision reactions. External force estimation in flight was performed with $K_I^f = 50$. Individual events can clearly be seen from the external forces. The first three reactions were due to the quadrotor hitting a static obstacle. Normal flight is resumed after stabilization. After $t = 25$ s, the quadrotor hovers and is hit by a human. Takeoff and landing events can be clearly recognized from the magnitude of the external force $\hat{f}_{e,z}$, see that takeoff is detected at $t = t_t$, and landing at $t = t_l$. Subsequent reactions are caused by a human hitting the robot during hovering.

Table 6.1: Estimated obstacle location and associated surface normal in the collision reaction experiment. The position is expressed in meters, in the inertial frame. Copyright © 2014 IEEE [135].

Reaction	x	y	z	n_x	n_y	n_z
No reaction	1.129	-0.339	-0.841	-0.964	0.051	-0.262
Stop	1.140	-0.335	-0.864	-0.983	-0.006	-0.180
Bounce	1.139	-0.242	-0.851	-0.982	-0.072	-0.175
Impedance	1.145	-0.304	-0.834	-0.989	0.034	-0.145

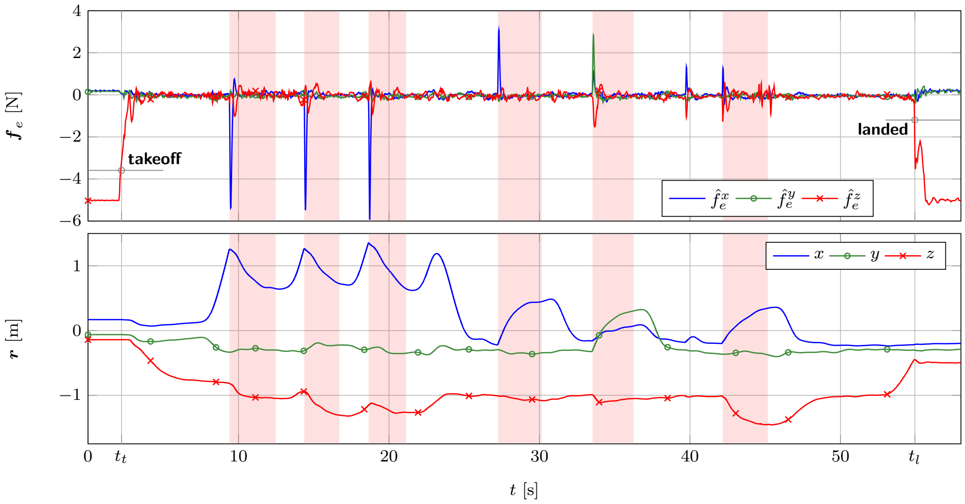
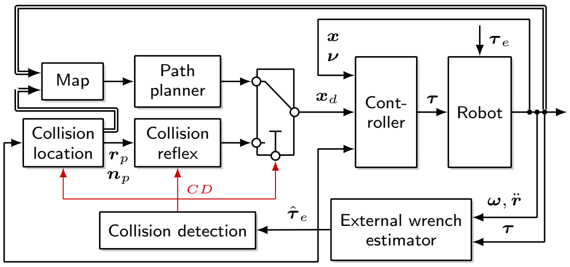


Figure 6.4: Flight with multiple collisions. *Equilibrium bounce* reactions are indicated by shaded areas. Copyright © 2014 IEEE [135].

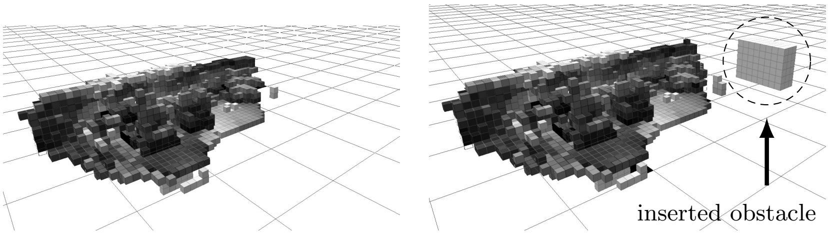
Clearly, the method's sensitivity is evident, as it reacts also to small peak forces. Note that the threshold has to be tuned according to the noise level in the estimated external wrench, These experiments were performed in a laboratory environment without wind. At higher airspeeds, or in wind, the noise level of the external wrench signal will be higher, therefore the thresholds could be adapted based on the estimated airspeed. The threshold set for collision detection is clearly a trade-off between false positives due to noise on the one hand, and sensitivity to small impacts on the other.

6.1.2 Tactile mapping

Following a collision reflex, information about the identified collision location may be built into an obstacle map. Such tactile mapping complements mapping by exteroceptive sensation. For carrying out the experiment, the hybrid external force estimator and the equilibrium bounce collision reaction strategy were implemented on an Asctec Pelican quadrotor, see Figure 6.5. The quadrotor [123] navigates using IMU–stereo fusion only and builds an onboard *octomap*, which is used for subsequent path planning. The reference trajectory was generated such that the onboard cameras purposefully look 90° rotated away from the flight direction, so that the polystyrene-block obstacle can not be seen. Upon collision, a $60 \times 60 \times 30$ cm obstacle is added to the octomap at the position and orientation of the collision, estimated via (5.6). An A* path planner is subsequently applied to go around the new obstacle mapped by using the collision reflex.



(a) Implementation of the collision reflex in an autonomous control structure. The collision reflex is independent of the path planner, and aids environment mapping parallel to exteroceptive sensors.



(b) Octomap before collision

(c) Octomap after collision — the inserted collision object is visible

Figure 6.5: A tactile mapping application. The collision reaction is a low-level reflex. Upon detecting a collision, an object is added at the estimated contact position and orientation to the onboard octomap and replan to avoid further collisions. This assumes collisions with a static environment. Copyright © 2017 IEEE [138].

6.2 Interaction, disturbance, and fault aware flying robot swarms

With the maturing of onboard visual navigation and perception for flying robots, research has partly focused on developing physical interaction and manipulation capabilites for these machines. In order to deploy such capabilities at a large scale, a high level of robustness against faults and disturbances is required. With the majority of the literature focusing on basic capabilities such as interaction control and (cooperative) aerial manipulation, handling general classes of faults and disturbances has not been covered in great detail and is still in its infancy.

In this section, we provide a synopsis of recent developments in physically interacting flying robots, put them in a larger taxonomical context, and develop a generalized concept of interacting flying robots at scale, such as in a swarm. A main contribution is the the introduction of the awareness pipeline that unifies the representation of interaction, disturbance, and fault awareness for a single flying robot as well as flying robot swarms. A synopsis of already developed individual elements of the pipeline from the literature is provided, and future research directions that would validate the proposed approach are speculated.

Interaction, disturbance, and fault aware flying robots. Figure 6.6 depicts a flying robot system (single robot, robot group or robot swarm) that is simultaneously subject to (possibly multiple) physical interactions (e.g. from a human, or manipulation), wind disturbances, and external faults such as collisions. An *interaction* is a desired dynamic behavior of the robot w.r.t. some input quantity. A *disturbance* is an uncontrolled input to the system, while a *fault* is an unpermitted deviation of at least one characteristic property or parameter of the system from the nominal operating condition [127]. A

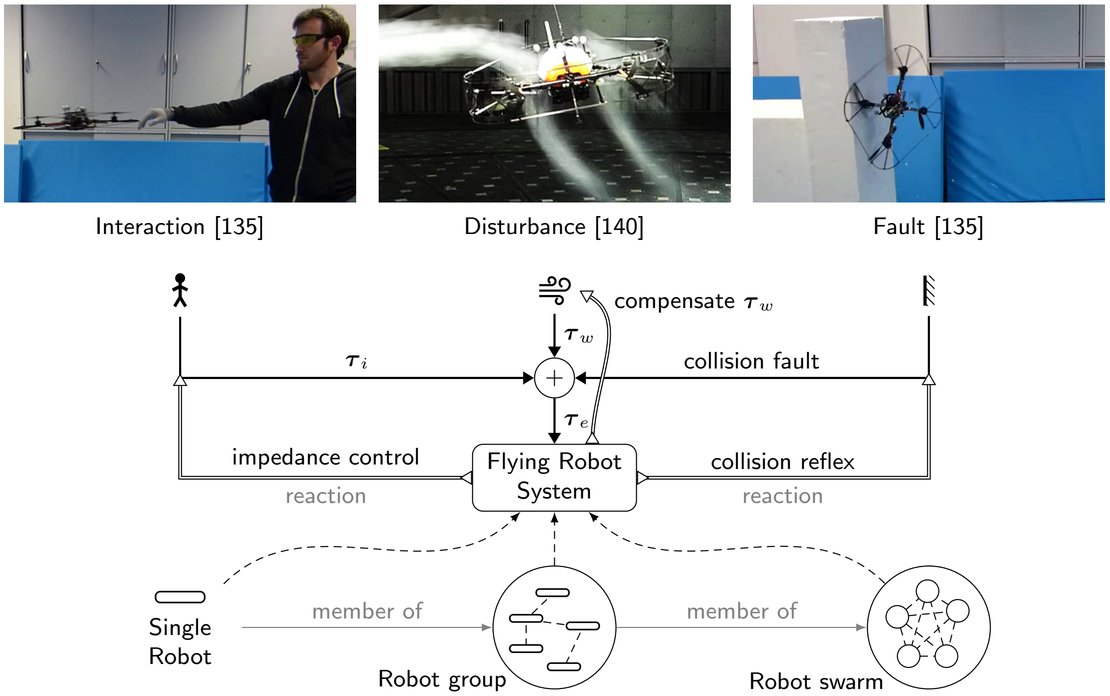

Interaction [135]
Disturbance [140]
Fault [135]

Figure 6.6: An interaction, disturbance, and fault aware flying robot system understands the source and context of its simultaneous inputs and reacts accordingly.

hypothetical *interaction, disturbance and fault aware flying robot system* will be able to sense these inputs and safely react consistent to a task-specific context. Furthermore, by exploiting contextual information from disturbance models it is argued that fundamentally new interaction paradigms may be achieved that allow e.g. physical interaction under wind influence, or following the wind speed gradient. Figure 6.7 depicts a speculative application of such a system. The task is to cooperatively slide an object along a compliant surface, under wind influence, while generating an accessible 3D map of the surface. A robot in the swarm serves as haptic input device for a human user. The system is composed of three robot groups. The blue group preforms cooperative slung load transport [19]. The red group grasps [45] for controlling the horizontal motion of the object. The green group is maintaining visual line of sight. To achieve complex interaction task, flying robots swarms must be *aware* of all involved interactions, in particular physical ones. In Section 6.2.2 the considered types of awareness are refined further.

Swarm hierarchy. Figure 6.8 depicts a hierarchy of an interaction, disturbance, and fault aware flying robot swarm, with some example tasks and disturbances. The flying robot swarm can be subdivided into multiple hiearchical subsystems: *propulsion*, *flying robot*, *robot group*, and *robot swarm*. Each subsystem subsumes multiple units of the lower level subsystem, and is subject to specific disturbances and faults (e.g. wind, latency, packet loss). Faults may occur at each hierarchy level and propagate upwards. Each subsystem is assigned prioritized tasks (e.g. trajectory tracking, formation, obstacle avoidance). Obviously, higher level tasks should not destabilize presumably safety critical lower level tasks – for example, formation control should not destabilize vehicle trajectory tracking or local collision avoidance. Lower hierarchy level tasks therefore have higher priority. Disturbances may be compensated and potentially exploited, and faults must be handled in a hierarchical manner due to propagation. In this section, methods to handle interaction, disturbances, and faults at multiple hierarchy levels are explored.

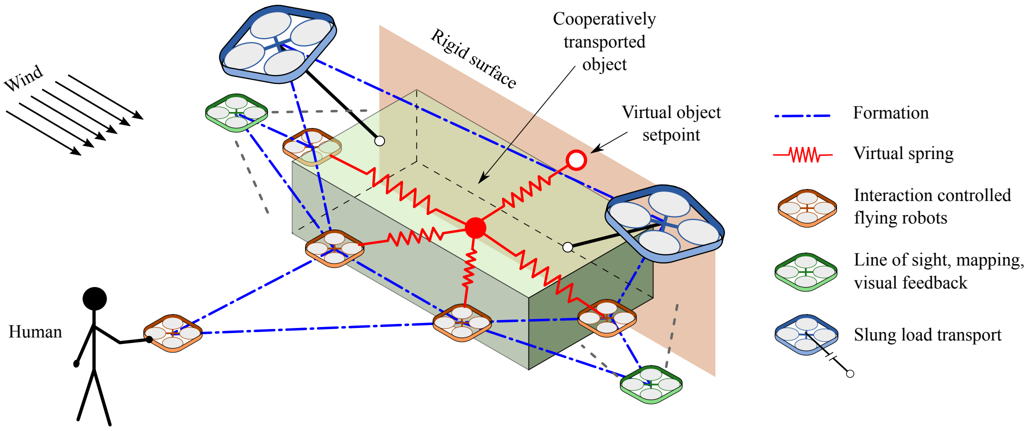

Wind
Rigid surface
Cooperatively transported object
Virtual object setpoint
Human
Formation
Virtual spring
Interaction controlled flying robots
Line of sight, mapping, visual feedback
Slung load transport

Figure 6.7: Exemplary hypothetical future scenario of cooperative flying robot swarms (top view). So far, no unified framework for dealing with such problems exists.

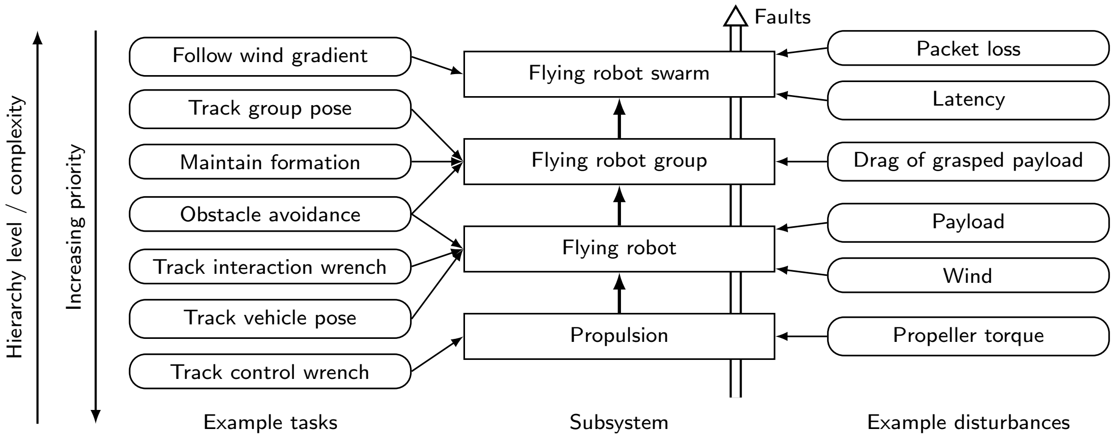


Figure 6.8: A hierarchy of flying robot subsystems, and example tasks and disturbances.

Physically interacting flying robots. Table 1.4 lists some examples of state of the art developments in physical interaction control of flying robots. Note that so far aerodynamic effects due to wind are largely ignored in the literature, or they are simply compensated as a lumped disturbance. As of today, this limits the applicability of physical interaction methods in harsh environments. The aim of this section is to develop a framework that also provides a roadmap and guidelines for future developments to enhance existing methods with disturbance and fault awareness.

6.2.1 Swarm system dynamics

System dynamics with disturbances and faults. In this paper we consider *interaction*, *disturbance* and *fault* aware flying robots, which should be reflected in their dynamics formulation. Start with the ideal system dynamics of an individual flying robot, which may be described by the Lagrangian dynamics

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau}, \tag{6.3}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the generalized inertia matrix, $\boldsymbol{q}$ are generalized coordinates, $\boldsymbol{g}(\boldsymbol{q})$ is the generalized gravity vector, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the matrix of Coriolis and centripetal terms, and $\boldsymbol{\tau}$ is the control input. For an individual flying robot without an articulated structure, $\boldsymbol{q}$ is the position and orientation of the robot, and $\boldsymbol{\tau}$ is the control wrench.

The actual system dynamics is subject to disturbances, faults, and parameter and structural changes [32, 127]. Dropping arguments, we may write

$$(\boldsymbol{M} + \Delta\boldsymbol{M})\ddot{\boldsymbol{q}} + \Delta\boldsymbol{g} + \boldsymbol{g} + (\Delta\boldsymbol{C} + \boldsymbol{C})\dot{\boldsymbol{q}} = \boldsymbol{\tau} + \Delta\boldsymbol{\tau} + \boldsymbol{\tau}_d + \boldsymbol{\tau}_{f,e}, \tag{6.4}$$

where $\boldsymbol{\tau}_d$ is the generalized disturbance (e.g. aerodynamics), $\boldsymbol{\tau}_{f,e}$ are the additive *external faults*. The prefix Δ denotes a change in the respective parameter or wrench, for example $\Delta\boldsymbol{\tau}$ is the change of propeller thrust due to airspeed. External faults violate the free-flight assumption, and are caused by external factors. Examples are collisions, contacts, undesired constrained motion, etc. The proper reaction is to change the control objective, and/or escalate to higher level planning. We also consider multiplicative *model faults* $\boldsymbol{\tau}_{f,m}$ due to parameter uncertainty or a change in the system dynamics, such as loss of motor actuation, change of aerodynamic coefficients, a shifting center of mass, or structural failure. The reaction is e.g. a structural change or adaptation of the controller. This distinction is made because the two fault types differ in effect and required recovery, see Section 6.2.3. By defining $\boldsymbol{\tau}_{f,m}$ from (6.4) as

$$\boldsymbol{\tau}_{f,m}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}, t) = \Delta\boldsymbol{\tau} - \Delta\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} - \Delta\boldsymbol{g}(\boldsymbol{q}) - \Delta\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}, \tag{6.5}$$

the dynamics of a single robot with disturbances and faults may be written as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} + \boldsymbol{\tau}_d + \boldsymbol{\tau}_f, \tag{6.6}$$

where faults are lumped into a single fault wrench $\boldsymbol{\tau}_f = \boldsymbol{\tau}_{f,m} + \boldsymbol{\tau}_{f,e}$.

System dynamics of a flying robot swarm. In the swarm, it is assumed that $\boldsymbol{g}(\boldsymbol{q})$ and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ are compensated by the local controllers at individual robot level, and a set of controllable coordinates $\boldsymbol{q}_c$ is exposed to the swarm level. Quadrotors, for example, are underactuated and their local controller ensures tracking of a Cartesian position and orientation about the body $z-$axis. For flying robots with full controllability, see Rajappa et al. [116] and Ryll et al. [120], $\boldsymbol{q}_c$ will contain the full 6D pose. For a swarm of robots, let the swarm generalized coordinates $\boldsymbol{q}_s$ contain the controllable coordinates $\boldsymbol{q}_c$ of each robot. The dynamics equations of the swarm may then be simplified to

$$\boldsymbol{M}_s\ddot{\boldsymbol{q}}_s = \boldsymbol{\tau}_s, \tag{6.7}$$

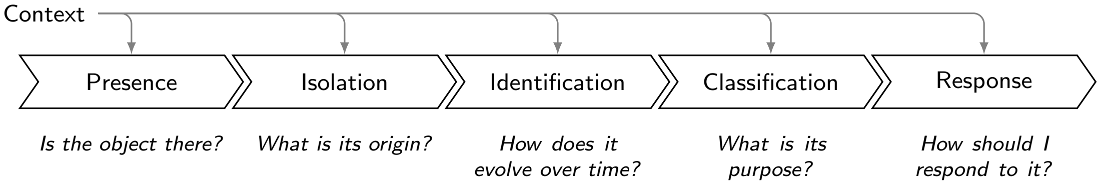
where $\boldsymbol{M}_s$ is the generalized swarm inertia matrix, and $\boldsymbol{\tau}_s$ is the swarm control input in the generalized coordinates. The dynamics with *swarm-level* disturbances and faults become

$$\boldsymbol{M}_s\ddot{\boldsymbol{q}}_s = \boldsymbol{\tau}_s + \boldsymbol{\tau}_{d,s} + \boldsymbol{\tau}_{f,m,s} + \boldsymbol{\tau}_{f,e,s}, \tag{6.8}$$

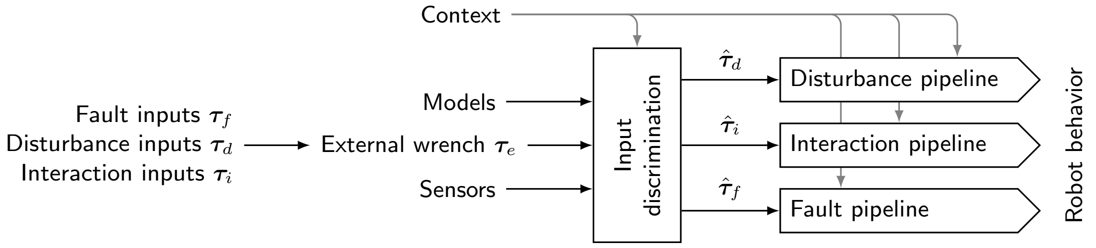
where $\boldsymbol{\tau}_{d,s}$, $\boldsymbol{\tau}_{f,m,s}$, and $\boldsymbol{\tau}_{f,e,s}$ are swarm-level disturbances, modeling, and external faults, respectively. Note that these equations have been presented in order to define the nomenclature and apply model-based concepts to the problem. They present the basic equations for the awareness pipeline. In the following sections, specifics of disturbance, interaction control, and fault awareness are discussed.

6.2.2 Awareness pipeline and interaction control

To deal with complex tasks, flying robot swarms must be *aware* of physical interactions. This awareness may be conceptualized through the novel concept *awareness pipeline*, depicted in Figure 6.9(a). It is derived from a deductive sequential question-driven process to generate knowledge. This is a generalization of a fault handling pipeline introduced in Haddadin et al. [55] to interactions and disturbances.



(a) The awareness pipeline for structural handling of faults, disturbances, and interaction.


Context
Fault inputs $\boldsymbol{\tau}_f$
Disturbance inputs $\boldsymbol{\tau}_d$
Interaction inputs $\boldsymbol{\tau}_i$
Models
External wrench $\boldsymbol{\tau}_e$
Sensors
Input discrimination
$\hat{\boldsymbol{\tau}}_d$
$\hat{\boldsymbol{\tau}}_i$
$\hat{\boldsymbol{\tau}}_f$
Disturbance pipeline
Interaction pipeline
Fault pipeline
Robot behavior

(b) Interaction, disturbance and fault awareness pipelines run in parallel.

Figure 6.9: The awareness pipeline and its parallel instantiation.

The choice of pipeline defines the scope of the awareness. First, the robot must know about the *presence* of an object (anything from a real physical object, a data-driven signal input to the arising of a monitoring signal). Second, an *isolation* step determines its origin (e.g. location). The *identification* step determines the time evolution of the object, enabling the *classification* of its purpose. Once this is known, a proper *response* may be taken by the robot.

The external wrench $\boldsymbol{\tau}_e$ acting on the robot is a sum of all interaction, disturbance, and fault inputs. Most control schemes operate under the assumption that only the interaction wrench acts on the robot [17, 92, 135]. Consider the interaction in Figure 6.6, where wind is added to the interaction wrench. In order to simultaneously compensate for the aerodynamic disturbance, perform impedance control towards the human, and react to collisions, the flying robot must be able to discriminate between the separate forces acting on it. *Input discrimination* is therefore a basic enabling component for running awareness pipelines in parallel, see Figure 6.9(b). It may use sensors (e.g. IMU, force sensing), models (e.g. dynamics, geometry), estimates and context (e.g. environment maps, task definitions, proximity of humans). This is discussed in detail in Section 6.2.3. Discriminated wrenches are used as inputs to the respective pipelines. The overall system behavior is a composition of the pipeline responses. Awareness pipelines run at each swarm hierarchy level. Note that contextual input can be provided by other levels in the hierarchy. In the following subsections, each pipeline is discussed in more detail.

6.2.3 Disturbance awareness pipeline

Disturbance presence, isolation and classification are usually obtained from context information, during the controller design stage. Table 1.2 lists the most common disturbances in flying robots, and sensors that may be used to discriminate between them. Each disturbance type (the *what*) requires a separate awareness pipeline, because their interpretation and responses will differ. Identification (*i.e.* time evolution of the signal) may be used to adapt the response online. In the following, two disturbance response strategies are discussed: *compensation* and *exploitation*.

Disturbance compensation. In most situations, it is desirable that a single interacting flying robot under the influence of disturbances maintains the simplifed dynamics (6.7). The

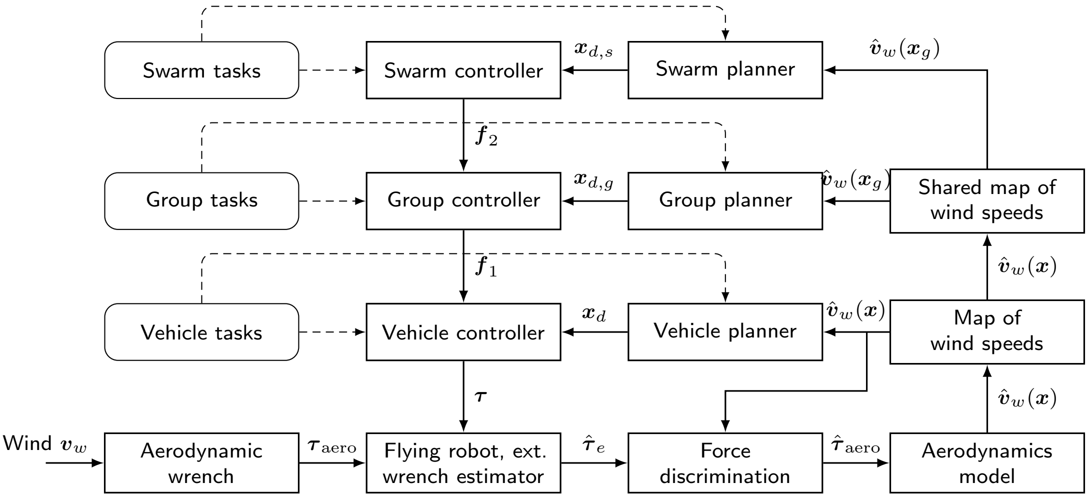


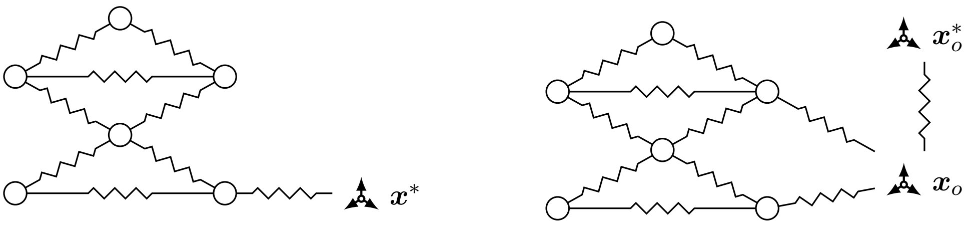
Figure 6.10: A framework for exploiting an aerodynamics model to map the wind speed and use the resulting wind map for planning on multiple levels of the hierarchy.

control law may be decomopsed as $\boldsymbol{\tau} = \boldsymbol{\tau}_{\text{task}} + \boldsymbol{\tau}_{\text{comp}}$, where $\boldsymbol{\tau}_{\text{task}}$ is a nominal controller that executes the desired task(s), e.g. impedance control, while $\boldsymbol{\tau}_{\text{comp}}$ compensates the disturbances. A common approach is to lump all disturbances and faults into one unknown input wrench, which is interpreted as a disturbance and compensated. The lumped disturbance estimation is obtained by an external wrench estimator [30, 118, 135], which is based on IMU measurements, a dynamics model, and the control input. Then, methods such as disturbance observers [80, 141, 118], or integral control may be applied for compensation. A compensated impedance controller that reacts to the interaction force and compensates wind forces has been proposed in Section 3.2.2.

Exploitation of disturbances. In some applications it may be beneficial to exploit a known model of the disturbance. Undoubtedly, there is large potential in exploiting wind information, which can be obtained from the aerodynamic disturbance using model-based methods [149, 136, 140]. Initial advancements have been made toward motion planning using known wind fields [148, 150]. In Ware and Roy [148], motion planning in an urban wind field is performed that minimizes the required motor power. The most promising direction seems to be building a shared map, as depicted in Figure 6.10. Instead of interpreting wind as a disturbance, disturbance context may be shared among robots. For example, in Yao et al. [150] the authors learned a map of disturbance wrenches to improve planning in the proximity of structures. Figure 6.10 depicts a generalized scheme that exploits a known aerodynamics model. Here, the wind $\boldsymbol{v}_w$ is causing an aerodynamics wrench $\boldsymbol{\tau}_{\text{aero}}$ on a flying robot. The estimated external wrench $\hat{\boldsymbol{\tau}}_e$ is then used for discriminating between wrenches acting on the robot, see Chapter 5 and Section 6.2.3. The result is an estimated aerodynamics wrench $\hat{\boldsymbol{\tau}}_{\text{aero}}$, which is inverted in a learned aerodynamics model to obtain an estimate $\hat{\boldsymbol{v}}_w$. This is stored in a local map of wind speeds, and shared among multiple robots. The wind map may then be used for planning e.g. energy-optimal trajectories $\boldsymbol{x}_d$, or to follow the wind gradient. Lastly, the controller at each level commands a lower-priority task $\boldsymbol{f}_i$ to the respective lower level controller, according to (6.10).

Interaction aware control and sensing

Handling of aerodynamic effects during interaction. The topic has not gained a lot of attention in literature, as discussed in Section 1.2. Model-based *contact wrench isolation* with minimal additional sensing, or at least obtaining the *contact position* on the



(a) Swarm formation flight of virtual springs in a leader-follower configuration.

(b) Cooperative grasping with flying robots through intermediate frames.

Figure 6.11: Operational space behaviors on swarm level imposed via virtual springs.

robot hull in a continuous fashion is an open problem in literature. This topic has been investigated in detail in Chapter 5, in the context of a single robot.

Multi-priority task definition. After the interaction input has been discriminated, the particular response of the interaction pipeline in Figure 6.9(b) is defined by one or more prioritized *tasks*. The response of the pipeline is then a generalized control force obtained from a multi-priority task controller. Our belief is that multi-priority task control for swarms will become an important research topic. Tasks may be defined w.r.t an *operational space* $\mathcal{O}$ [73]. The operational space task $\boldsymbol{x}$ can be mapped to the generalized coordinates $\boldsymbol{q}$ via the Jacobian $\boldsymbol{J}(\boldsymbol{q}) := \frac{\partial x}{\partial \boldsymbol{q}}$. For example, the operational space of a quadrotor may be its position and yaw, while its generalized coordinates are its full 6-DoF pose. A controller in the operational space will command a generalized wrench $\boldsymbol{f}(\boldsymbol{x})$, which may then be translated to the generalized coordinates by the map

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\,\boldsymbol{f}(\boldsymbol{x}). \tag{6.9}$$

Due to redundancy, multiple tasks may be imposed on a robot or a swarm simultaneously. In particular, redundancy resolution methods from multi-priority control of redundant robot manipulators and humanoids may be extended to flying robot swarms. The basic idea is to project lower-priority tasks into the nullspace of higher-priority tasks so they do not interfere with each other [4, 9, 13]. Doing this at the swarm level dynamics (6.7) would result in a hierearchical multi-priority wrench swarm controller. Note that this is still an open research problem. Alternatively, consensus algorithms can be used, see Olfati-Saber et al. [105] and references therein. The control input for task with priority k would then be given by

$$\boldsymbol{\tau}_k = \boldsymbol{\tau}_{k-1} + (\boldsymbol{J}_k \boldsymbol{P}_{k-1})^{-T}(\boldsymbol{f}_k - \boldsymbol{J}_k \boldsymbol{\tau}_{k-1}), \tag{6.10}$$

where $\boldsymbol{f}_k$ is the desired operational space wrench of task i, and $\boldsymbol{\tau}_0 = \boldsymbol{0}$. The matrix $\boldsymbol{P}_k$ is the projection operator onto the nullspace of matrix $\boldsymbol{J}_k$ (i.e. $\boldsymbol{P}_k = \boldsymbol{I} - \boldsymbol{J}_k^+ \boldsymbol{J}_k$) [125]. This state-of-the art approach decouples tasks and ensures that the higher priority tasks are not perturbed by lower priority tasks, at the cost of reduced perfomance of the latter. Of course, it is important to have good models for all levels of the swarm hierarchy. Well-researched simultaneous tasks for swarms include maintaining a formation, obstacle avoidance, and tracking desired positions [104].

Operational space interaction behaviors. In the context of interaction, it is also possible to include desired physical interaction with the environment and/or objects to be manipulated, such as admittance [11, 120, 135, 138], impedance [92, 117, 118, 135, 138], or force control [6, 17, 110]. Typically, impedance control for the tracking case is defined through the desired error dynamics

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{D}\dot{\boldsymbol{x}} + \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}^*) = \boldsymbol{f}_k, \tag{6.11}$$

where $\boldsymbol{x}^*$ is the desired equilibrium state, and $\boldsymbol{D}$ and $\boldsymbol{K}$ are appropriate damping and stiffness gain matrices, respectively. Operational space controllers $\boldsymbol{f}(\boldsymbol{x})$ may be defined on the robot, group [13, 156] or swarm level, see Figure 6.11. Only the task Jacobian must be changed. Note that using this framework, also perception-based tasks on all hierarchy levels may be included.

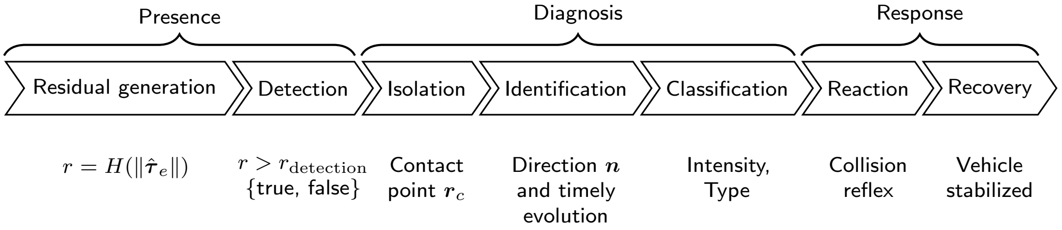
Physical interaction with swarms. There have been significant advancements in cooperative aerial manipulation [13, 91, 156]. However, physical interaction with a flying robot swarm is still and unresearched area. *External wrench estimation* for robot swarms would enable novel physical interaction paradigms, such as moving the swarm through physical interaction with its agents. Furthermore, input discrimination may be applied at higher levels of the hierarchy by sharing context between the robots. Each robot may be understood as a particle of a rigid, elastic or articulated body. Then, physical interaction properties on this body may be defined on swarm level analogous to the single robot or manipulator case, respectively.

Fault handling pipeline

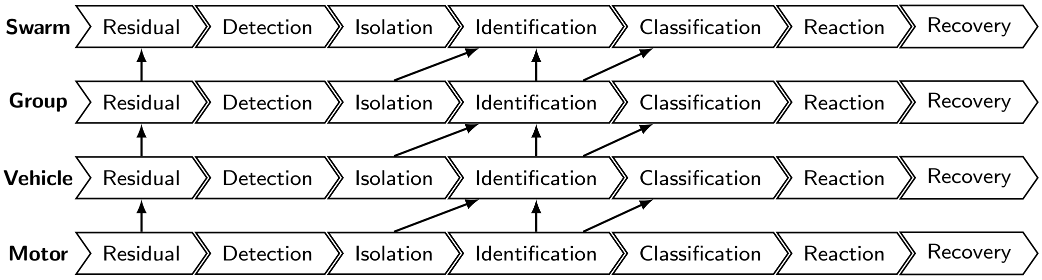
Figure 6.12 depicts a fault handling pipeline [55, 32, 127] on a collision detection example as in Section 6.1. The pipeline starts with a *residual generator*, which is a fault indicator based on a deviation between measurements and a system model (practically also a data-driven model). The residual is used for fault *detection*, which outputs a binary signal (true/false), i.e. whether the fault is present, as well as the time of detection. Fault *isolation* deals with determination of the kind and location of the fault. The fault is then *identified* to determine the size and time-variant behavior of the fault. This can further be *classified*, *e.g.* by severity, intensity, intent, etc. The appropriate *reaction* depends on the result of the fault *diagnosis*, which consists of isolation, identification and classification. The last step in the pipeline is fault *recovery.*

Hierarchical fault handling. Faults may occur at multiple levels of the robot or robot swarm, as depicted in Figure 6.12(b). In this hierarchical scheme, multiple fault handling pipelines run in parallel. Residual generators may use signals from lower levels in the hierarchy as well. The diagnosis is similarly connected, where identification may use lower level isolation and identification signals, and classification may use lower level identification signals. The reaction depends on the fault type, as explained below. Note that there is a inherent and possibly time-varying latency between the levels in the fault handling pipeline.

Model faults cause an internal reconfiguration of the system. This may be *adaptation* or *structural* (morphological) reconfiguration. Adaptation is, for example, changing parameters of an adaptive controller, or changing the spring stiffness in a formation of robots. For example, the loss of a motor in a quadrotor requires a different control strategy than in the nominal case because the dynamics changes considerably [94]. For a group of robots, losing one robot may require reconfiguring the morphology of the formation [41]. Similarly, when the individual robot cannot maintain (6.7) anymore (e.g. due to actuator failure), the robot may safely land, while removing itself from the swarm, in turn necessitating the swarm to react with according awareness (swarm-level reflexes). Picking up a payload with a single robot can also be handled as a model fault, detected e.g. by a change detection algorithm [53]. The proper reaction would be parameter adaptation [91], e.g. by resetting the covariance of estimated parameters. The recovery is complete once the parameter


Presence
Diagnosis
Response
Residual generation
Detection
Isolation
Identification
Classification
Reaction
Recovery
$r = H(\|\hat{\boldsymbol{\tau}}_e\|)$
$r > r_{\text{detection}}$ {true, false}
Contact point $\boldsymbol{r}_c$
Direction $\boldsymbol{n}$ and timely evolution
Intensity, Type
Collision reflex
Vehicle stabilized

(a) Single level of a fault handling pipeline [55] on a collision detection example [136].



(b) Hierarchical fault handling. Higher level diagnosis stages may use signals from other levels of the hierarchy. A low level fault may also escalate the recovery to a higher level.

Figure 6.12: Hierarchical model-based fault handling pipeline.

covariance converges below a threshold. In this case, no escalation to a higher level in the hierarchy is necessary, as the local fault-aware controller is ensuring that the robot behaves as (6.7). However, if multiple robots grasp a payload simultaneously, this may trigger a *group-level* model fault, leading to distributed estimation of the payload parameters.

External faults trigger a change in the task objectives. This might be a *reflex* reaction, or a *switch* to a different interaction task. In the hierarchical fault handling pipeline, re-planning at the group or swarm level may also be triggered. *Unwanted collisions with the environment* are an example of external faults [136], and may be handled according to the pipeline from Figure 6.12 as follows. On motor level, a residual may be used to detect collisions with the propeller. An example reaction is slowing down or stopping the motor to prevent further damage. Going up to the vehicle level, the motor residual may be used for detection, together with the highpass filtered external force as depicted in Figure 6.12(b). After the collision direction and location are isolated, the classification may use information from the motor level to determine whether the collision was on the hull or within the propeller. Contextual information such as vehicle velocity at the collision time may be used. An example reaction is stopping the vehicle away from the collision pose, as to “bounce off”. Inclusion in the environment map and replanning is also possible in this case, see Section 6.1. The vehicle is recovered once the reflex is complete. Going further up in the hierarchy, in the group and swarm levels, this may trigger a change in the formation parameters, like changing the spring stiffness between the robots, or changing the desired formation.

Disturbance and fault estimation for robot groups. *Disturbance estimation* exist for individual robots, however this area is unexplored for groups of flying robots. Furthermore, flying robot swarm literature is mostly concerned with motion planning. *General fault disturbance handling* can be seen as an enabling factor in robust operation of swarms. Therefore, fault detection at the group and swarm level seem to be important underrepresented research topics. *Parameter and aerodynamic disturbance estimation* during cooperative load carrying seem to be topics that have not yet been handled in the

Table 6.2: Common faults for single flying robots and robot swarms.

Type	Description	Reaction
	Model faults $\boldsymbol{\tau}_{f,m}$	
Motor loss	Loss of motor or propeller, reducing vehicle controllability.	Change controller [94] and land (underactuated vehicle), adapt control allocation matrix (redundant propellers) [121]
Parameter change	Parameter uncertainty or change of parameters due to e.g. picking up payload	Reset covariance of a parameter estimation algorithm [53]
Contact with envirionment	Contact is achieved, changing sytem dynamics [6, 110, 120]	Switch to contact controller
Loss of robot	A robot has failed critically.	Remove the robot from the swarm and reconfigure formation.
	External faults $\boldsymbol{\tau}_{f,e}$	
Hull collision	Unwanted collision of robot hull with environment.	Stabilize vehicle at safe distance from collision pose, replan trajectory [138].
Motor collision	Objects in propellers.	Stop or slow down the motor, escalate to hull collision reaction.
Contact	One or more robots come into contact with environment	Engage force closure

literature. However, this may change as aerial manipulation systems mature and start executing missions under wind influence.

6.2.4 Conclusions

Significant advancements in flying robots physically interacting with the environment were made in literature. Still, in comparison to well established disciplines like industrial robotics the entire field is still rather young. Based on the state of the art and by taking significant speculative steps, a synopsis and high-level control framework for concretizing a fully elaborated vision of interaction aware flying robot systems were assembled. This was possible since on the one hand the major technological challenges when aiming for general robust interaction capabilities, high-performance disturbance compensation and exploitation, and fault handling in flying single robots and swarms were isolated in this section. On the other hand, a unifying methodology was proposed, inspired from deductive awareness concepts in order to define and develop awareness in flying systems. The resulting awareness pipelines aim to generalize methods from fault detection and isolation literature to interactions and disturbances. Lastly, speculation about future research that is needed to achieve truly interactive, disturbance and fault aware flying robot swarms in the wild was undertaken.

Illustration: J.-J. Grandville: Café; Book: Vie privée et publique des animaux; Paris: Hetzel, 1867

CHAPTER 7

Conclusion

This thesis aimed to make aerial robots aware of the disturbance, interaction, and faults acting on them. This yielded the following goals:

G1) Make flying robots *aware* of contacts and collisions, by providing a low-level detection and reaction framework.

G2) Provide an estimate of the wind velocity that is independent of the external wrench.

G3) Bridge the gap between trajectory tracking and interaction control, to allow *physical interaction under wind influence*.

These goals in turn posed following research questions:

Q1) Is discrimination between the external wrench components for control purposes possible in real time?

Q1) Given the individual wrench components, what are effective control schemes for interaction and trajectory tracking control under wind influence?

Q3) How can unexpected faults, such as collisions with the environment, be detected and handled efficiently and effectively?

Q4) Is it possible to obtain a measurement of the wind speed that is independent of the external wrench?

Table 7.1 provides an overview how the goals and questions map to chapters in this thesis.

The following contributions were made. In Chapter 2, a systematic three-stage parameter identification procedure for aerial robots was developed, to accurately identify required models [139]. Then, external wrench estimation techniques were extended to be suitable for aerial robots without the need of velocity measurements [135, 138]. Results show that using only the IMU makes it possible to implement the wrench estimator on embedded platforms knowing only the control input, IMU measurements, and the system model. This technique allows the reasoning about external wrenches without the need for additional sensors such as force, contact or wind sensors, forming the basis of all other methods developed in the thesis.

In Chapter 3 tracking and physical interaction control for flying robots were developed, with explicit estimation of the external wrench. Therefore, effective control schemes for interaction and trajectory tracking control under wind influence are explored. It is shown that directly compensating the estimated disturbance is an effective method for improving

Table 7.1: Matrix of thesis goals and questions to chapters.

	Ch 3	**Ch 4**	**Ch 5**	**Ch 6**	**Achieved?**
G1	·	·	●	●	Y
G2	·	●	·	·	Y
G3	●	·	●	·	Y/N
Q1	·	·	●	·	Y
Q2	●	·	·	·	Y
Q3	·	·	●	●	Y
Q4	·	●	·	·	Y

tracking performance of flying robots under wind influence. The impedance controller is extended to compensate aerodynamic disturbances, i.e. to facilitate physical interaction under wind influence (goal 3). This controller was used to obtain simulation results in Chapter 5. Since only simulation results are provided, the effectiveness of the proposed controller in real-world conditions still remains an open question.

Wind estimation is investigated in Chapter 4, using different data-driven aerodynamic models [140, 139]. To train and evaluate the models, measurements obtained by flying a custom-built hexacopter in a 3D wind tunnel are used. Solutions to Goal 2 and Question 4 have been explored in this chapter. A major contribution ot this thesis is related to estimating wind speed using measured motor power. Measurements obtained from wind tunnel flights could then be used to estimate the aerodynamic power of each coaxial rotor pair during flight, and then the wind speed. Two novel methods were developed for this purpose. The first employs a data-driven approach to build nonlinear regression models from aerodynamic power to airspeed. The second is a first-principles model driven approach that builds an optimization problem based on the propeller aerodynamics model and online motor power measurements. This contribution allows the propellers to be used as wind speed sensors and to provide a measurement that is independent of the IMU.

The topic of discriminating between aerodynamic, interaction, and fault wrenches is introduced in Chapter 5, touching on Goals 1 and 3, and Question 1. The methods developed in this thesis allow the external wrench to be decomposed into the constituent terms continuously and in real-time. This is achieved by modeling the relevant aerodynamics models and reasoning about the wind velocity, as evaluated in Chapter 4. Force discrimination outputs are applied in the compensated impedance controller developed in Chapter 3 to allow compensation of the wind disturbance while being compliant to interaction forces. This allows interaction control under wind influence without the need for additional sensors, further facilitating Goal 3. In Chapter 5, four novel methods to discriminate between the inputs, using previously identified models were developed and evaluated in simulation. All of them rely on estimating the wind speed in an inertial frame and applying an aerodynamics model to obtain the aerodynamic wrench. This is subtracted from the estimated external wrench to obtain the interaction wrench. The first, *modified model-checking* scheme, is based on slowing down wind estimation when contact is detected. Contact detection is based on the residual between the external torque and the expected aerodynamic torque as a function of the external force. Limits of this contact detection methods are also shown. The second method is a *particle filter* that directly estimates the contact position, under the assumption of a point contact on the robot's convex hull. The third method uses the estimated *aerodynamic power* to obtain the airspeed. Therefore, airspeed estimation in this scheme is independent of the vehicle's IMU. Lastly, the modified model-checking and power-based method are combined in a

Kalman filter and use the particle filter to obtain the contact position. This results in a robust overall scheme for obtaining the interaction wrench and contact position when a flying robot is operating under wind influence, without the need for additional sensors.

The problem of collision detection and safe reaction for flying robots without the need of additional sensors has been solved in Chapter 5. Collisions are detected based on the frequency content of the external force. Effective collision reflex schemes were presented in Chapter 6, relating to Question 3. The geometry of the problem can be used to obtain the collision location on the robot's convex hull. Next, slow contact forces can be distinguished from wind through a residual of the aerodynamic torque model. Discrimination can then be performed by simultaneously estimating the wind speed, and slowing down wind estimation when contact is detected. Due to the failure cases of this method, an alternative method that uses aerodynamic power based wind estimation is developed. This relies on other aerodynamic models to distinguish between the constituent terms. In summary, this contribution presents a starting point of a quite new research field for aerial robots, as it opens interesting new questions.

It can be concluded that all posed goals have been successfully achieved, with some force discrimination methods requiring experimental validation.

Future work

This thesis opens new research opportunities, mainly in two areas. The first is estimating wind by using propellers as sensors in flight, providing a measurement independent of the external force, and therefore supporting Goal 2. This method re-uses existing hardware (motors and speed controllers), and requires only modeling and computational effort. In this thesis, the first successful estimation techniques to obtain wind speed from aerodynamic power estimates were developed. This largely unexplored field has opened new research questions. In the case of physical model based wind estimation, while the power optimization problem works in principle, its performance and robustness can be improved dramatically, providing new research opportunities. For example, state of the art trajectory optimization techniques for maximizing observability of a problem may be applied to this problem in future work. Furthermore, extending and improving the problem formulation to also include other measurements and make it more robust is also left for future work. Note that this field is still young, as motor current sensing, which is a prerequisite for this method, is still not commonly deployed in flying robot research platforms.

The second novel area is force discrimination. In this thesis significant steps have been made to formulate the problem and provide the first effective and efficient model-based techniques towards solving Goal 3. For collision detection, extensive experiments and high-level architectures were presented, showing the robustness of this method and how it can be applied in the context of fully autonomous flying robots. Discrimination between slow contact and aerodynamic forces was only explored in simulation. Nevertheless, this was based on real-world data, leveraging aerodynamics models identified from flights in a 3D wind tunnel. As such, the developed methods are promising, however they have yet to be experimentally tested in real-world conditions, which is left for future work and might require architectural changes.

Finally, the thesis explored interaction, disturbance, and fault aware flying robot swarms in Chapter 6, touching on Goal 1. Extensions of the developed methods to robot swarms were discussed and novel research questions beyond state of the art were postulated, based on insights gained in this thesis.

Bibliography

[1] F. A. A. (FAA), *Registration and marking requirements for small unmanned aircraft*, online, 2015. [Online]. Available: `https://www.federalregister.gov/documents/2015/12/16/2015-31750/registration-and-marking-requirements-for-small-unmanned-aircraft`.

[2] D. Abeywardena, Z. Wang, G. Dissanayake, S. L. Waslander, and S. Kodagoda, 'Model-aided State Estimation for Quadrotor Micro Air Vehicles amidst Wind Disturbances', in *IROS 2014*, Chicago, IL, USA, Sep. 2014, pp. 4813–4818.

[3] M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, 'Inversion based direct position control and trajectory following for micro aerial vehicles', in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 2933–2939.

[4] J. A. Acosta, C. R. de Cos, and A. Ollero, 'A robust decentralised strategy for multi-task control of unmanned aerial systems. application on underactuated aerial manipulator', in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2016, pp. 1075–1084.

[5] A. Albers, S. Trautmann, T. Howard, T. A. Nguyen, M. Frietsch, and C. Sauter, 'Semi-autonomous flying robot for physical interaction with environment', in *Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on*, 2010, pp. 441–446. DOI: `10.1109/RAMECH.2010.5513152`.

[6] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, 'Aerial robotic contact-based inspection: Planning and control', *Autonomous Robots*, vol. 40, no. 4, pp. 631–655, Apr. 2016, ISSN: 1573-7527.

[7] K. Alexis, G. Nikolakopoulos, and A. Tzes, 'Experimental model predictive attitude tracking control of a quadrotor helicopter subject to wind-gusts', in *MED '10*, 2010, pp. 1461–1466. DOI: `10.1109/MED.2010.5547844`.

[8] ——, 'On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances', *Asian Journal of Control*, vol. 16, no. 1, pp. 209–224, 2014.

[9] G. Antonelli, F. Arrichiello, and S. Chiaverini, 'The null-space-based behavioral control for autonomous robotic systems', *Intelligent Service Robotics*, vol. 1, no. 1, pp. 27–39, Jan. 2008, ISSN: 1861-2784.

[10] B. Armstrong, 'On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics', *The International Journal of Robotics Research*, vol. 8, no. 6, pp. 28–48, 1989. DOI: `10.1177/027836498900800603`. eprint: `https://doi.org/10.1177/027836498900800603`. [Online]. Available: `https://doi.org/10.1177/027836498900800603`.

[11] F. Augugliaro and R. D'Andrea, 'Admittance Control for Physical Human-Quadrocopter Interaction', in *Control Conference (ECC), 2013 European*, Zürich, Switzerland, Jul. 2013, pp. 1805–1810.

[12] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, 'The flight assembled architecture installation: Cooperative construction with flying machines', *IEEE Control Systems*, vol. 34, no. 4, pp. 46–64, 2014.

[13] K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, 'Behavioral control of unmanned aerial vehicle manipulator systems', *Autonomous Robots*, vol. 41, no. 5, pp. 1203–1220, Jun. 2017, ISSN: 1573-7527.

[14] M. Bangura, H. Lim, H. J. Kim, and R. Mahony, 'Aerodynamic Power Control for Multirotor Aerial Vehicles', in *ICRA 2014*, Hong Kong, China, Jun. 2014, pp. 529–536.

[15] M. Bangura and R. Mahony, 'Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor', in *Australasian Conference on Robotics and Automation, Proceedings of*, Wellington, New Zealand, Dec. 2012.

[16] ——, 'Thrust control for multirotor aerial vehicles', *IEEE Transactions on Robotics*, vol. 33, accepted, Mar. 2017.

[17] S. Bellens, J. De Schutter, and H. Bruyninckx, 'A hybrid pose / wrench control framework for quadrotor helicopters', in *ICRA 2012*, 2012, pp. 2269–2274. DOI: `10.1109/ICRA.2012.6224682`.

[18] V. M. H. Bennetts, T. P. Kucner, E. Schaffernicht, P. P. Neumann, H. Fan, and A. J. Lilienthal, 'Probabilistic air flow modelling using turbulent and laminar characteristics for ground and aerial robots', *IEEE Robotics and Automation Letters*, 2017.

[19] M. Bernard and K. Kondak, 'Generic slung load transportation system using small size helicopters', in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, IEEE, 2009, pp. 3258–3264.

[20] L. Besnard, Y. Shtessel, and B. Landrum, 'Control of a quadrotor vehicle using sliding mode disturbance observer', in *ACC '07*, 2007, pp. 5230–5235. DOI: `10.1109/ACC.2007.4282421`.

[21] S. Bouabdallah and R. Siegwart, 'Backstepping and sliding-mode techniques applied to an indoor micro quadrotor', in *ICRA '05*, 2005, pp. 2247–2252. DOI: `10.1109/ROBOT.2005.1570447`.

[22] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] D. Brescianini and R. D'Andrea, 'Design, modeling and control of an omni-directional aerial vehicle', in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 3261–3266.

[25] A. Briod, P. Kornatowski, A. Klaptocz, A. Garnier, M. Pagnamenta, J.-C. Zufferey, and D. Floreano, 'Contact-based navigation for an autonomous flying robot', in *IROS 2013*, 2013, pp. 3987–3992. DOI: `10.1109/IROS.2013.6696926`.

[26] P. Bruschi, M. Piotto, F. Dell'Agnello, J. Ware, and N. Roy, 'Wind speed and direction detection by means of solid-state anemometers embedded on small quadcopters', *Procedia Engineering*, vol. 168, pp. 802–805, 2016.

[27] R. Chartrand and W. Yin, 'Iteratively reweighted algorithms for compressive sensing', in *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on*, IEEE, 2008, pp. 3869–3872.

[28] I. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, 'A prototype of an autonomous controller for a quadrotor UAV', in *ECC '07*, 2007.

[29] E. Davis and P. E. I. Pounds, 'Direct sensing of thrust and velocity for a quadrotor rotor array', *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1360–1366, Jul. 2017, ISSN: 2377-3766.

[30] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, 'Collision detection and safe reaction with the dlr-iii lightweight manipulator arm', in *IROS 2006*, 2006, pp. 1623–1630. DOI: `10.1109/IROS.2006.282053`.

[31] G. Dicker, F. Chui, and I. Sharf, 'Quadrotor collision characterization and recovery control', in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, 2017, pp. 5830–5836.

[32] S. Ding, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.

[33] N. Diolaiti, C. Melchiorri, and S. Stramigioli, 'Contact impedance estimation for robotic systems', *Robotics, IEEE Transactions on*, vol. 21, no. 5, pp. 925–935, 2005, ISSN: 1552-3098. DOI: `10.1109/TRO.2005.852261`.

[34] A. Doucet and A. M. Johansen, 'A tutorial on particle filtering and smoothing: Fifteen years later', in *Oxford Handbook of Nonlinear Filtering*, 2011.

[35] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, 'Adaptive control of quadrotor UAVs: A design trade study with flight evaluations', *Control Systems Technology, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2012, ISSN: 1063-6536. DOI: `10.1109/TCST.2012.2200104`.

[36] J. Escareño, S. Salazar, H. Romero, and R. Lozano, 'Trajectory control of a quadrotor subject to 2D wind disturbances', *Journal of Intelligent & Robotic Systems*, pp. 1–13, 10.1007/s10846-012-9734-1, ISSN: 0921-0296.

[37] ——, 'Trajectory Control of a Quadrotor Subject to 2D Wind Disturbances', *J. Intell. Robotics Syst.*, vol. 70, no. 1-4, pp. 51–63, Apr. 2013, ISSN: 0921-0296. DOI: `10.1007/s10846-012-9734-1`. [Online]. Available: `http://dx.doi.org/10.1007/s10846-012-9734-1`.

[38] M. Faessler, A. Franchi, and D. Scaramuzza, 'Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories', *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018. DOI: `10.1109/LRA.2017.2776353`. [Online]. Available: `https://doi.org/10.1109/LRA.2017.2776353`.

[39] Z. Fang and W. Gao, 'Adaptive integral backstepping control of a micro-quadrotor', in *ICICIP '11*, vol. 2, 2011, pp. 910–915. DOI: `10.1109/ICICIP.2011.6008382`.

[40] F. Forte, R. Naldi, A. Macchelli, and L. Marconi, 'Impedance control of an aerial manipulator', in *American Control Conference (ACC), 2012*, 2012, pp. 3839–3844.

[41] A. Franchi, C. Secchi, M. Ryll, H. Bülthoff, and P. R. Giordano, 'Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs', *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, 2012.

[42] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, 'Modeling and control of a flying robot for contact inspection', in *IROS 2012*, 2012, pp. 3532–3537. DOI: `10.1109/IROS.2012.6385917`.

[43] M. Fumagalli, R. Naldi, A. Macchelli, F. Forte, A. Q. L. Keemink, S. Stramigioli, R. Carloni, and L. Marconi, 'Developing an aerial manipulator prototype: Physical interaction with the environment', *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 41–50, Sep. 2014, ISSN: 1070-9932.

[44] M. Fumagalli and R. Carloni, 'A modified impedance control for physical interaction of UAVs', in *IROS 2013*, 2013, pp. 1979–1984. DOI: `10.1109/IROS.2013.6696619`.

[45] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo, 'The flying hand: A formation of uavs for cooperative aerial tele-manipulation', in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 4335–4341.

[46] G. Golub and C. Van Loan, *Matrix Computations*, ser. Matrix Computations. Johns Hopkins University Press, 2012, ISBN: 9781421408590. [Online]. Available: `https://books.google.de/books?id=5U-l8U3P-VUC`.

[47] X. Gong, Y. Tian, Y. Bai, and C. Zhao, 'Trajectory tacking control of a quad-rotor based on active disturbance rejection control', in *ICAL '12*, 2012, pp. 254–259. DOI: `10.1109/ICAL.2012.6308207`.

[48] F. A. Goodarzi, D. Lee, and T. Lee, 'Geometric adaptive tracking control of a quadrotor unmanned aerial vehicle on se (3) for agile maneuvers', *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 9, p. 091 007, 2015.

[49] F. Goodarzi, D. Lee, and T. Lee, 'Geometric nonlinear pid control of a quadrotor uav on se (3)', in *Control Conference (ECC), 2013 European*, IEEE, 2013, pp. 3845–3850.

[50] F. Guerrero Castellanos, N. Marchand, S. Lesecq, and J. Delamare, 'Bounded attitude stabilization: Real-time application on four-rotor helicopter', Anglais, in *Preprints of the 17th IFAC World Congress*, Département Automatique Département Automatique, Seoul, Corée, République De: IFAC, 2008, pp. -.

[51] J. F. Guerrero-Castellanos, A. Hably, N. Marchand, and S. Lesecq, 'Bounded attitude stabilization: Application on four-rotor helicopter', in *ICRA '07*, 2007, pp. 730–735. DOI: `10.1109/ROBOT.2007.363073`.

[52] J. Guerrero, J. Escareno, and Y. Bestaoui, 'Quad-rotor MAV Trajectory Planning in Wind Fields', in *ICRA*, Karlsruhe, Germany, May 2013, pp. 778–783.

[53] F. Gustafsson and F. Gustafsson, *Adaptive filtering and change detection*. Wiley New York, 2000, vol. 1.

[54] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger, 'Collision detection and reaction: A contribution to safe physical Human-Robot Interaction', in *IROS 2008*, Sep. 2008, pp. 3356–3363. DOI: `10.1109/IROS.2008.4650764`.

[55] S. Haddadin, A. D. Luca, and A. Albu-Schäffer, 'Robot collisions: A survey on detection, isolation, and identification', *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–21, 2017, ISSN: 1552-3098. DOI: `10.1109/TRO.2017.2723903`.

[56] S. Haddadin, *Towards Safe Robots - Approaching Asimov's 1st Law*, ser. Springer Tracts in Advanced Robotics. Springer Verlag, 2014, vol. 90, pp. 1–343.

[57] C. Hancer, K. Oner, E. Sirimoglu, E. Cetinsoy, and M. Unel, 'Robust position control of a tilt-wing quadrotor', in *CDC '10*, 2010, pp. 4908–4913. DOI: `10.1109/CDC.2010.5717283`.

[58] H. Hangan, 'The Wind Engineering Energy and Environment (WindEEE) Dome at Western University, Canada', *Wind Engineers*, vol. 39, p. 350, 2014.

[59] N. J. Higham, *Functions of Matrices: Theory and Computation.* Philadelphia, PA, USA: Society of Industrial and Applied Mathematics (SIAM), 2008. DOI: `10.1137/1.9780898717778`.

[60] G. M. Hoffmann, S. L. Wasl, and C. J. Tomlin, 'Quadrotor helicopter trajectory tracking control', in *AIAA GNC*, 2008.

[61] N. Hogan, 'Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications', *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, Mar. 1985.

[62] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, 'An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications', in *ICRA 2013*, 2013, pp. 1736–1741. DOI: `10.1109/ICRA.2013.6630805`.

[63] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, 'Introduction to feedback control of underactuated vtol vehicles: A review of basic control design ideas and principles', *IEEE Control Systems*, vol. 33, no. 1, pp. 61–75, 2013.

[64] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, 'Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering', in *ICRA 2009*, Kobe, Japan, May 2009, pp. 3277–3282.

[65] S. Jeong, S. Jung, and M. Tomizuka, 'Attitude control of a quad-rotor system using an acceleration-based disturbance observer: An empirical approach', in *AIM '12*, 2012, pp. 916–921. DOI: `10.1109/AIM.2012.6265965`.

[66] S.-H. Jeong and S. Jung, 'Experimental studies of a disturbance observer for attitude control of a quad-rotor system', in *ICCAS '12*, 2012, pp. 579–583.

[67] Joon-Young Park, Byung-Hak Cho, and Jae-Kyung Lee, 'Trajectory-tracking control of underwater inspection robot for nuclear reactor internals using time delay control', *Nuclear Engineering and Design*, vol. 239, pp. 2543–2550, 2009.

[68] J. Jung, B. Glasner, C. R. Russell, and G. Willink, 'Wind tunnel and hover performance test results for multicopter uas vehicles', in *Proceedings of the AHS 72nd Annual Forum*, West Palm Beach, FL, USA, 2016.

[69] S. Jung, 'A position-based force control approach to a quad-rotor system', in *Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference on*, 2012, pp. 373–377. DOI: `10.1109/URAI.2012.6463019`.

[70] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart, 'Fast nonlinear model predictive control for multicopter attitude tracking on so (3)', in *Control Applications (CCA), 2015 IEEE Conference on*, IEEE, 2015, pp. 1160–1166.

[71] M. Kamel, M. Burri, and R. Siegwart, 'Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles', *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463–3469, 2017.

[72] H. K. Khalil, *Nonlinear Systems*, Third. Prentice-Hall, 2002.

[73] O. Khatib, 'A unified approach for motion and force control of robot manipulators: The operational space formulation', *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, Feb. 1987, ISSN: 0882-4967.

[74] K. Kondak, M. Bernard, N. Meyer, and G. Hommel, 'Autonomously flying VTOL-robots: Modeling and control', in *ICRA '07*, 2007, pp. 736–741. DOI: `10.1109/ROBOT.2007.363074`.

[75] K. Lee, J. Back, and I. Choy, 'Disturbance observer based trajectory tracking controller for quadrotor', in *ICCAS '12*, 2012, pp. 157–161.

[76] T. Lee, M. Leok, and N. H. McClamroch, 'Geometric tracking control of a quadrotor UAV on SE(3)', in *CDC '10*, Atlanta, GA, USA, Dec. 2010.

[77] T. Lee, M. Leoky, and N. McClamroch, 'Geometric tracking control of a quadrotor uav on se(3)', in *Decision and Control (CDC), 2010 49th IEEE Conference on*, Dec. 2010, pp. 5420–5425. DOI: `10.1109/CDC.2010.5717652`.

[78] J. G. Leishman, *Principles of Helicopter Aerodynamics*, 2nd ed., ser. Cambridge Aerospace Series. Cambridge University Press, 2006.

[79] F. Leonard, A. Martini, and G. Abba, 'Robust nonlinear controls of model-scale helicopters under lateral and vertical wind gusts', *Control Systems Technology, IEEE Transactions on*, vol. 20, no. 1, pp. 154–163, Jan. 2012, ISSN: 1063-6536. DOI: `10.1109/TCST.2010.2102023`.

[80] S. Li, J. Yang, W.-H. Chen, and X. Chen, *Disturbance Observer-Based Control: Methods and Applications*, 1st. Boca Raton, FL, USA: CRC Press, Inc., 2014, ISBN: 1466515791, 9781466515796.

[81] T. Madani and A. Benallegue, 'Backstepping control for a quadrotor helicopter', in *IROS '06*, 2006, pp. 3255–3260. DOI: `10.1109/IROS.2006.282433`.

[82] ——, 'Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles', in *ACC '07*, 2007, pp. 5887–5892. DOI: `10.1109/ACC.2007.4282548`.

[83] L. Manuelli and R. Tedrake, 'Localizing external contact using proprioceptive sensors: The contact particle filter', in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 5062–5069. DOI: `10.1109/IROS.2016.7759743`.

[84] M. Marino, A. Fisher, R. Clothier, S. Watkins, S. Prudden, and C. S. Leung, 'An Evaluation of Multi-Rotor Unmanned Aircraft as Flying Wind Sensors', in *International Micro Aerial Vehicle Conference (IMAV)*, Aachen, Germany, Sep. 2015.

[85] P. Martin and E. Salaun, 'The true role of accelerometer feedback in quadrotor control', in *ICRA 2010*, May 2010, pp. 1623–1629. DOI: `10.1109/ROBOT.2010.5509980`.

[86] C. D. McKinnon and A. P. Schoellig, 'Unscented External Force and Torque Estimation for Quadrotors', *IROS2016*, pp. 5651–5657, Oct. 2016.

[87] D. T. McRuer, D. Graham, and I. Ashkenas, *Aircraft Dynamics and Automatic Control.* Princeton, NJ, USA: Princeton University Press, 1972, ISBN: 0691080836.

[88] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, 'Design, modeling, estimation and control for aerial grasping and manipulation', in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 2668–2673.

[89] ——, 'Design, modeling, estimation and control for aerial grasping and manipulation', in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 2668–2673.

[90] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, 'Design, modeling, estimation and control for aerial grasping and manipulation', in *IROS '11*, 2011, pp. 2668–2673. DOI: `10.1109/IROS.2011.6094871`.

[91] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, 'Cooperative grasping and transport using multiple quadrotors', in *Distributed autonomous robotic systems*, Springer, 2013, pp. 545–558.

[92] A. Y. Mersha, S. Stramigioli, and R. Carloni, 'Variable impedance control for aerial interaction', in *IROS2014*, IEEE, 2014, pp. 3435–3440.

[93] N. Michael, J. Fink, and V. Kumar, 'Cooperative manipulation and transportation with aerial robots', *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[94] M. W. Mueller and R. D'Andrea, 'Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers', in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 45–52.

[95] G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, 'Experiments on coordinated motion of aerial robotic manipulators', in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1224–1229.

[96] R. Naldi, A. Torre, and L. Marconi, 'Robust blind navigation for a miniature ducted-fan aerial robot', in *American Control Conference (ACC), 2013*, 2013, pp. 988–993.

[97] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, 'Autonomous gas-sensitive microdrone: Wind vector estimation and gas distribution mapping', *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 50–61, 2012.

[98] P. P. Neumann and M. Bartholmai, 'Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit', *Sensors and Actuators A: Physical*, vol. 235, pp. 300–310, 2015.

[99] M. Neunert, C. De Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, 'Fast nonlinear model predictive control for unified trajectory optimization and tracking', in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 1398–1404.

[100] H.-N. Nguyen and D. Lee, 'Hybrid Force/Motion Control and Internal Dynamics of Quadrotors for Tool Operation', in *IROS 2013*, Tokyo, Japan, Nov. 2013, pp. 3458–3464.

[101] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, 'A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam', in *ICRA 2014*, Hong Kong, China, May 2014, pp. 431–437.

[102] J. Nocedal and S. Wright, *Numerical optimization.* Springer Science & Business Media, 2006.

[103] Ola-Erik Fjellstad, 'Control of unmanned underwater vehicles in six degrees of freedom - a quaternion feedback approach', PhD thesis, University of Trondheim, 1994.

[104] R. Olfati-Saber, 'Flocking for multi-agent dynamic systems: Algorithms and theory', *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006, ISSN: 0018-9286.

[105] R. Olfati-Saber, J. A. Fax, and R. M. Murray, 'Consensus and cooperation in networked multi-agent systems', *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007, ISSN: 0018-9219. DOI: `10.1109/JPROC.2006.887293`.

[106] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel, 'Hardware and software architecture for nonlinear control of multirotor helicopters', *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 6, pp. 1724–1736, Dec. 2013, ISSN: 1083-4435. DOI: `10.1109/TMECH.2013.2274558`.

[107] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel, 'Nonlinear Control of VTOL UAVs Incorporating Flapping Dynamics', in *IROS 2013*, Tokyo, Japan, Nov. 2013, pp. 2419–2425.

[108] C. Ott, R. Mukherjee, and Y. Nakamura, 'A hybrid system framework for unified impedance and admittance control', *Journal of Intelligent & Robotic Systems*, Jul. 2014.

[109] R. Oung and R. D'Andrea, 'The distributed flight array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle', *The International Journal of Robotics Research*, vol. 33, no. 3, pp. 375–400, 2014.

[110] C. Papachristos, K. Alexis, and A. Tzes, 'Efficient force exertion for aerial robotic manipulation: Exploiting the thrust-vectoring authority of a tri-tiltrotor uav', in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 4500–4505.

[111] K.-J. Park, 'Fourier-based optimal excitation trajectories for the dynamic identification of robots', *Robotica*, vol. 24, no. 5, pp. 625–633, 2006.

[112] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, 'Scikit-learn: Machine learning in python', *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[113] L. Planckaert and P. Coton, 'Quadrotor UAV aerodynamic model identification using indoor flight experiment and feasibility of UAV as wind gust sensor', in *International Micro Aerial Vehicle Conference (IMAV)*, Aachen, Germany, Sep. 2015.

[114] S. Prudden, A. Fischer, A. Mohamed, and S. Watkins, 'A flying anemometer quadrotor: Part 1', in *Proceedings of the International Micro Aerial Vehicle Conference (IMAV)*, Beijing, PR of China, Oct. 2016, pp. 15–21.

[115] G. V. Raffo, M. G. Ortega, and F. R. Rubio, 'An integral predictive/nonlinear control structure for a quadrotor helicopter', *Automatica*, vol. 46, no. 1, pp. 29–39, 2010, ISSN: 0005-1098. DOI: `10.1016/j.automatica.2009.10.018`.

[116] S. Rajappa, M. Ryll, H. H. Bülthoff, and A. Franchi, 'Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers', in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4006–4013.

[117] S. Rajappa, H. Bülthoff, and P. Stegagno, 'Design and implementation of a novel architecture for physical human-uav interaction', *The International Journal of Robotics Research*, p. 0 278 364 917 708 038, 2017.

[118] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, 'Impedance Control of VToL UAVs with a Momentum-based External Generalized Forces Estimator', in *ICRA 2014*, Hong Kong, China, Jun. 2014, pp. 2093–2099.

[119] ——, 'Passivity-based control of VToL UAVs with a momentum-based estimator of external wrench and unmodeled dynamics', *Robotics and Autonomous Systems*, vol. 72, pp. 139–151, 2015.

[120] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, Antonelli, C. G, and A. F Franchi, '6d physical interaction with a fully actuated aerial robot', in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017.

[121] M. Ryll, D. Bicego, and A. Franchi, 'Modeling and Control of FAST-Hex: a Fully-Actuated by Synchronized-Tilting Hexarotor', in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2016*, Daejeon, South Korea, 2016.

[122] F. Schiano, J. Alonso-Mora, K. Rudin, P. Beardsley, R. Siegwart, and B. Siciliano, 'Towards Estimation and Correction of Wind Effects on a Quadrotor UAV', in *International Micro Aerial Vehicle Conference (IMAV)*, Delft, Netherlands, Aug. 2014.

[123] K. Schmid, P. Lutz, T. Tomić, E. Mair, and H. Hirschmüller, 'Autonomous vision-based micro air vehicle for indoor and outdoor navigation', *Journal of Field Robotics*, vol. 31, no. 4, pp. 537–570, 2014, ISSN: 1556-4967. DOI: `10.1002/rob.21506`. [Online]. Available: `http://dx.doi.org/10.1002/rob.21506`.

[124] K. Schmid, T. Tomić, F. Ruess, H. Hirschmüller, and M. Suppa, 'Stereo vision based indoor/outdoor navigation for flying robots', in *IROS 2013*, 2013, pp. 3955–3962. DOI: `10.1109/IROS.2013.6696922`.

[125] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[126] L. Sikkel, G. de Croon, C. De Wagter, and Q. Chu, 'A novel online model-based wind estimation approach for quadrotor micro air vehicles using low cost mems imus', in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, IEEE, 2016, pp. 2141–2146.

[127] S. Simani, C. Fantuzzi, and R. J. Patton, *Model-based fault diagnosis in dynamic systems using identification techniques*. Springer Science & Business Media, 2013.

[128] *Skydio r1*, Mar. 2018. [Online]. Available: `https://www.skydio.com/product/`.

[129] K. Sreenath, T. Lee, and V. Kumar, 'Geometric control and differential flatness of a quadrotor uav with a cable-suspended load', in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, IEEE, 2013, pp. 2269–2274.

[130] Sung-Uk Lee and Pyung Hun Chang, 'The development of anti-windup scheme for time delay control with switching action using integral sliding surface', *Journal of Dynamic Systems, Measurement, and Control*, vol. 125, 2003.

[131] J. Svacha, K. Mohta, and V. Kumar, 'Improving quadrotor trajectory tracking by compensating for aerodynamic effects', in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2017, pp. 860–866.

[132] J. Swevers, C. Ganseman, D. B. Tukel, J. De Schutter, and H. Van Brussel, ‘Optimal robot excitation and identification’, *IEEE transactions on robotics and automation*, vol. 13, no. 5, pp. 730–740, 1997.

[133] N. Sydney, B. Smyth, and D. Paley, ‘Dynamic control of autonomous quadrotor flight in an estimated wind field’, in *CDC2013*, Dec. 2013, pp. 3609–3616. DOI: `10.1109/CDC.2013.6760438`.

[134] N. J. Sydney, ‘Rotorcraft Flight Dynamics and Control in Wind for Autonomous Sampling of Spatiotemporal Processes’, PhD thesis, University of Maryland, 2015.

[135] T. Tomić and S. Haddadin, ‘A Unified Framework for External Wrench Estimation, Interaction Control and Collision Reflexes for Flying Robots’, in *IROS 2014*, Chicago, IL, USA, Sep. 2014.

[136] ——, ‘Simultaneous estimation of aerodynamic and contact forces in flying robots: Applications to metric wind estimation and collision detection’, in *ICRA 2015*, Seattle, WA, USA, May 2015, pp. 5290–5296. DOI: `10.1109/ICRA.2015.7139937`.

[137] T. Tomić, M. Maier, and S. Haddadin, ‘Learning quadrotor maneuvers from optimal control and generalizing in real-time’, in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1747–1754. DOI: `10.1109/ICRA.2014.6907087`.

[138] T. Tomić, C. Ott, and S. Haddadin, ‘External wrench estimation, collision detection, and reflex reaction for flying robots’, *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1467–1482, Dec. 2017, ISSN: 1552-3098. DOI: `10.1109/TRO.2017.2750703`.

[139] T. Tomić, K. Schmid, P. Lutz, A. Mathers, and S. Haddadin, ‘Simultaneous Contact and Aerodynamic Force Estimation (s-CAFE) for Aerial Robots’, *International Journal of Robotics Reseach (IJRR)*, 2018 (submitted).

[140] ——, ‘The Flying Anemometer: Unified Estimation of Wind Velocity from Aerodynamic Power and Wrenches’, in *IROS 2016*, IEEE, Daejeon, Korea, Oct. 2016, pp. 1637–1644.

[141] T. Tomić, ‘Evaluation of acceleration-based disturbance observation for multicopter control’, in *Control Conference (ECC), 2014 European*, Jun. 2014, pp. 2937–2944. DOI: `10.1109/ECC.2014.6862237`.

[142] T. Tomić and S. Haddadin, ‘Towards Interaction, Disturbance and Fault Aware Flying Robot Swarms’, in *Proceedings of the 18th International Symposium on Robotics Research (ISRR)*, Puerto Varas, Chile, Dec. 2017.

[143] T. Tomić, K. Schmid, P. Lutz, A. Dömel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, ‘Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue’, *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 46–56, 2012.

[144] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. Society of Industrial and Applied Mathematics (SIAM), 1997.

[145] Vadim Utkin, Jürgen Guldner, and Jingxin Shi, *Sliding Mode Control in Electro-Mechanical Systems*. CRC Press, 2009.

[146] H. Voos, ‘Nonlinear control of a quadrotor micro-uav using feedback-linearization’, in *ICM ’09*, 2009, pp. 1–6. DOI: `10.1109/ICMECH.2009.4957154`.

[147] ——, ‘Nonlinear control of a quadrotor micro-uav using feedback-linearization’, in *ICM ’09*, 2009, pp. 1–6. DOI: `10.1109/ICMECH.2009.4957154`.

[148] J. Ware and N. Roy, ‘An analysis of wind field estimation and exploitation for quadrotor flight in the urban canopy layer’, in *ICRA 2016*, IEEE, 2016, pp. 1507–1514.

[149] S. L. Waslander and C. Wang, ‘Wind Disturbance Estimation and Rejection for Quadrotor Position Control’, in *AIAA Infotech@Aerospace Conference*, Seattle, WA, USA, Apr. 2009.

[150] J. W. Yao, V. R. Desaraju, and N. Michael, ‘Experience-based models of surface proximal aerial robot flight performance in wind’, in *International Symposium on Experimental Robotics*, Springer, 2016, pp. 563–573.

[151] D. W. Yeo, N. Sydney, D. Paley, and D. Sofge, ‘Onboard flow sensing for downwash detection and avoidance with a small quadrotor helicopter’, in *Proc. AIAA GNC Conference*, 2015.

[152] D. W. Yeo, N. Sydney, and D. A. Paley, ‘Onboard Flow Sensing for Rotary-Wing UAV Pitch Control in Wind’, in *Proc. AIAA GNC Conference*, 2016, p. 1386.

[153] Y. Al-Younes, M. Jarrah, and S. Sukkarieh, ‘Adaptive integral backstepping controller for an autonomous rotorcraft’, in *ISMA ’09*, 2009, pp. 1–7. DOI: `10.1109/ISMA.2009.5164795`.

[154] B. Yüksel, C. Secchi, H. H. Buelthoff, and A. Franchi, ‘A nonlinear force observer for quadrotors and application to physical interactive tasks’, in *Advanced Intelligent Mechatronics, IEEE/ASME International Conference on 2014*, Besançon, France, Jul. 2014.

[155] B. Yüksel, C. Secchi, H. H. Bülthoff, and A. Franchi, ‘Reshaping the physical properties of a quadrotor through ida-pbc and its application to aerial physical interaction’, in *ICRA 2014*, Hong Kong, China, Jun. 2014, pp. 6258–6265.

[156] M. Zhao, K. Kawasaki, X. Chen, Y. Kakiuchi, K. Okada, and M. Inaba, ‘Transformable multirotor with two-dimensional multilinks: Modeling, control, and whole-body aerial manipulation’, in *2016 International Symposium on Experimental Robotics*. Cham: Springer International Publishing, 2017, pp. 515–524, ISBN: 978-3-319-50115-4.

List of Figures

1.1 The techniques developed in the framework of this thesis have been applied and experimentally verified on the depicted research and commercial multirotor flying robots. 1
1.2 An interaction, disturbance, and fault aware flying robot system understands the source and context of its simultaneous inputs and reacts accordingly. . . 3
1.3 Different multirotor flying robot systems were used to develop and experimentally verify the techniques developed in this thesis. 12
1.4 Applications of the external wrench estimate developed in this thesis. . . . 13
1.5 Overview of the objectives of this thesis. 14

2.1 High-level overview of subsystems of a mutirotor flying robot. 17
2.2 Free-body diagram of a quadrotor and a hexacopter 19
2.3 Thrust T is generated by increasing the wind velocity $\boldsymbol{v}_w$ by the propeller induced velocity v_i, which goes through the propeller normal. The propeller slipstream finally merges into the wind flow to produce $\boldsymbol{w}$. 20
2.4 Relative induced velocity v_i/v_h in forward flight, depending on angle of attack and relative airspeed v_∞/v_h, depicted on the radial axis. 21
2.5 Parameter identification procedure in three steps to minimize coupling effects in the high-dimensional parameter space. 24
2.6 Setup of the force-torque sensor experiment for parameter identification of the DLR Ardea hexacopter. 25
2.8 Estimation of the external yaw torque, using the measured motor speed only ($m_{e,z}^{\varpi}$), using the motor speed and rotor acceleration ($m_{e,z}^{\varpi,\dot{\varpi}}$) and using motor current ($m_{e,z}^{i_a}$). 27
2.9 Rigid body torque prediction using parameters obtained by the IRLS method, minimizing the ℓ_1-norm. Overall the model shows a good match to the ground truth. 28
2.10 Structure of the hybrid external wrench estimator near hovering conditions. It uses the robot acceleration $\ddot{\boldsymbol{r}}$ and angular velocity $\boldsymbol{\omega}$, both measured directly by the onboard IMU. 29

3.1 Block diagram of the attitude controller and disturbance observer. 33
3.2 Block diagram of the position controller with disturbance observer. 36
3.3 Transformations involved in creating an attitude reference from the virtual control force $\boldsymbol{f}$. 36
3.4 Simulation comparison of position controller errors for setpoint control and disturbance. 38

3.5 Estimation and compensation of time-varying disturbances in simulation, using the ISM+DO method. 38
3.6 Experimental setups for hovering tests with external disturbance investigated in this thesis. 39
3.7 Hovering performance using different position and attitude controllers. . . . 39
3.8 Position and velocity tracking during aggressive waypoint trajectory tracking. 40
3.9 Simulation and experimental results for a Hummingbird quadrotor executing Dynamic Movement Primitive (DMP) trajectories 41
3.10 Position, pitch and disturbance estimate in the suspended weight experiment with PD+DO, ISM+DO and AIBC controllers. 42
3.11 Structure of an impedance controller (3.31) if it were possible to apply forces in every direction. This would be the case for a UAV with fully actuated degrees of freedom, where $\operatorname{rank}(\boldsymbol{B}) = 6$. 43
3.12 Control structure of an impedance controller for an underactuated UAV, like a typical quadrotor with $\operatorname{rank}(\boldsymbol{B}) < 6$. 44
3.13 Simulated response of the position impedance controller to an external force in the inertial $x-$direction, for varying virtual inertia and damping parameters. 44
3.14 Blocks considered in the passivity proof of the cascaded impedance controlled flying robot, considering the external wrench estimator and attitude error dynamics. 46
3.15 Structure of the admittance controller (3.54). The controller generates a position and yaw reference, which is then tracked by a position controller. The external wrench causes a velocity. 50
3.16 Admittance controller response. The parameters determine the steady-state velocity of the virtual system. The system is in damping mode — the virtual stiffness is $\boldsymbol{K}_v = \boldsymbol{0}$, and the real system mass is $m = 0.55$ kg. 50
3.17 Trajectory and estimated external force in the impedance controller experiment. The quadrotor can be freely moved along the inertial x and y axes 51

4.1 Convergence of the Newton-Raphson (4.2) (solid) and iterative linearization (4.1) (dashed) methods for obtaining the wind velocity from a force by model inversion. 56
4.2 Experimental setup inside the wind tunnel and schematic layout of the wind tunnel test. 57
4.3 Flow visualization in the wind tunnel at various horizontal wind speeds. . . 58
4.4 The training and validation set for aerodynamics models, containing over 37 minutes of flight data under horizontal wind speed up to 8 m/s, and vertical wind speeds up to 5 m/s. 59
4.5 Flight data distribution. 59
4.6 Fit of aerodynamic power for individual propeller pairs. 60
4.7 Power ratio at different airspeeds, relative to aerodynamic power in hover. . 62
4.8 Normed error histograms of predicted airspeed as a function of different input sets, for different models. 64
4.9 Distribution of external force samples used for training and validation of the mapping *force (input) to relative airspeed (output)*. 66
4.10 Change of model shape with increasing regularization. 67
4.11 Training and validation mean square errors for 10-fold cross-validation of different models fitting $\boldsymbol{v}_r(\boldsymbol{f}_e)$ 67

4.12 Training and validation mean square errors for 10-fold cross-validation of a Perceptron model with $N_h = 8$ 68
4.13 Training and validation mean square errors for 10-fold cross-validation of a quadratic model 68
4.14 Error histograms for models predicting the aerodynamic torque as a function of the aerodynamic force $\boldsymbol{m}_d(\boldsymbol{f}_d)$, for the validation dataset. 69
4.15 Zero contours of (4.19), with color indicating the function value. 72
4.16 Sensitivity of the NLS solution to error in aerodynamic power. 73
4.17 Sensitivity analzsis of the NLS solution for the same case as in Figure 4.16, however for perfect knowledge of horizontal velocity components (v_x, v_y). . 73
4.18 The combined wind estimation scheme. 74
4.19 Estimation of relative airspeed for the complete dataset from Figure 4.4 . . 74
4.20 Normalized error histograms of the vertical airspeed component v_z for the dataset from Figure 4.4 75

5.1 Force discrimination reconstructs the time-varying additive inputs summed in the external wrench $\boldsymbol{\tau}_e$, given an estimate thereof $\hat{\boldsymbol{\tau}}_e$, the robot's state, and additional models and/or sensor inputs. 80
5.2 An overview of methods to discriminate slow aerodynamic and interaction wrenches. 80
5.3 Spectrograms of the wrenches acting on a quadrotor during simulated flight and the according robot position as well as wind speed. 82
5.4 Structures of the *filter bank* (a) and *model checking* (b) methods for discriminating between the aerodynamic and contact wrench from the lumped wrench $\hat{\boldsymbol{\tau}}_e$. 83
5.5 Wind velocity estimation in a collision and contact scenario, showing erroneously estimated wind velocity during contact. 84
5.6 The residual wrench for the flight shown in Figure 5.5, with a first-order wind velocity propagation model. 84
5.7 A collision reflex when the wind force creates an offset in the external force. 85
5.8 Obtaining the collision position $\boldsymbol{r}_c$ from the external force $\boldsymbol{f}_c$ and torque $\boldsymbol{m}_c$ 87
5.9 Simulation of collision location determination for a quadrotor in different collision configurations. 87
5.10 Torque residual for constant forces at positions around a superellipsoidal convex hull. 88
5.11 Modified model checking scheme for simultaneous wind and interaction estimation. 88
5.12 Simulation results for discrimination between aerodynamic and contact forces under wind influence, using exactly the residual based estimation scheme from Figure 5.11. 90
5.13 Observability of contact forces on a superellipsoidal convex hull, with different aerodynamic torque models. 91
5.14 Force estimation at the exactly known contact position for the simulation in Figure 5.12, using different aerodynamic torque models. 92
5.15 Particle filter convergence with a linear model, while the aerodynamic wrench is obtained with a nonlinear model. 96
5.16 Example when particles converge to the wrong side of the convex hull, estimating the wrong contact position. 96
5.17 Input discrimination and wind estimation scheme based on the particle filter. 96

5.18 Discrimination between aerodynamic and contact forces under wind influence, using the particle filter based approach. 97
5.19 Aerodynamic power based force discrimination scheme. 98
5.20 Simulation results for power based force discrimination, for the same case as in Figure 5.12. 98
5.21 Simulation results for Kalman filter-based force discrimination. 99

6.1 Simulated robot trajectories for different collision reaction strategies. 104
6.2 Collision reaction experiment with a polystyrene block. 105
6.3 Video stills of the equilibrium bounce collision reaction experiment depicted in Figure 6.2. 105
6.4 Flight with multiple collisions and collision reactions 106
6.5 A tactile mapping application with the collision reaction as a low-level reflex. 107
6.6 An interaction, disturbance, and fault aware flying robot system understands the source and context of its simultaneous inputs and reacts accordingly. . . 108
6.7 Exemplary hypothetical future scenario of cooperative flying robot swarms (top view). So far, no unified framework for dealing with such problems exists. 109
6.8 A hierarchy of flying robot subsystems, and example tasks and disturbances. 109
6.9 The awareness pipeline and its parallel instantiation. 111
6.10 A framework for exploiting an aerodynamics model to map the wind speed and use the resulting wind map for planning on multiple levels of the hierarchy. 112
6.11 Operational space behaviors on swarm level imposed via virtual springs. . . 113
6.12 Hierarchical model-based fault handling pipeline. 115

List of Tables

1.1 An overview of work related to trajectory tracking and disturbance compensation. 4
1.2 A list of the most common disturbances acting on flying robots. 5
1.3 An overview of work related to physical interaction control. 6
1.4 Application examples of physical interaction between flying robots and environment. 6
1.5 Overview of work related to collision detection and handling. 8
1.6 Overview of related work in the field of wind estimation and force discrimination. 8
1.7 Work related to discriminating between aerodynamic and interaction forces. 11
1.8 Most typical sensors and models that enable the discrimination of contact forces acting on flying robots. 11

2.1 System parameters identified in the first identification step, using data from a force-torque sensor. 25
2.2 Results of the rigid body parameter identification step, using data from an identification flight, and the identified propulsion model. 27
2.3 Motivation for the hybrid estimation scheme — the translational velocity required by the momentum observer would have to be estimated. 29

3.1 System and control parameters used in the impedance control experiments with the AscTec Hummingbird. 52

4.1 Summary of tested regression models. 63
4.2 Summary of tested inputs for predicting relative airspeed. 63

5.1 Simulation parameters used to obtain the wind estimation and collision detection results. The robot is an quadrotor with arm length $L = 0.17$ m. . . 83
5.2 Overview of pipeline stages for simultaneous collision handling and wind estimation, in the context of a Fault Detection, Identification and Isolation pipeline [55]. 101

6.1 Estimated obstacle location and associated surface normal in the collision reaction experiment. 106
6.2 Common faults for single flying robots and robot swarms. 116

7.1 Matrix of thesis goals and questions to chapters. 120

Publications

Journal publications

[1] K. Schmid, P. Lutz, T. Tomić, E. Mair, and H. Hirschmüller, 'Autonomous vision-based micro air vehicle for indoor and outdoor navigation', *Journal of Field Robotics*, vol. 31, no. 4, pp. 537–570, 2014, ISSN: 1556-4967. DOI: `10.1002/rob.21506`. [Online]. Available: `http://dx.doi.org/10.1002/rob.21506`.

[2] T. Tomić, C. Ott, and S. Haddadin, 'External wrench estimation, collision detection, and reflex reaction for flying robots', *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1467–1482, Dec. 2017, ISSN: 1552-3098. DOI: `10.1109/TRO.2017.2750703`.

[3] T. Tomić, K. Schmid, P. Lutz, A. Mathers, and S. Haddadin, 'Simultaneous Contact and Aerodynamic Force Estimation (s-CAFE) for Aerial Robots', *International Journal of Robotics Reseach (IJRR)*, 2018 (submitted).

[4] T. Tomić, K. Schmid, P. Lutz, A. Dömel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, 'Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue', *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 46–56, 2012.

Conference publications

[1] M. Mueller, F. Steidle, M. J. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomic, and W. Stürzl, 'Robust Visual-Inertial State Estimation with Multiple Odometries and Efficient Mapping on an MAV with Ultra-Wide FOV Stereo Vision', in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3701–3708.

[2] K. Schmid, T. Tomić, F. Ruess, H. Hirschmüller, and M. Suppa, 'Stereo vision based indoor/outdoor navigation for flying robots', in *IROS 2013*, 2013, pp. 3955–3962. DOI: `10.1109/IROS.2013.6696922`.

[3] T. Tomić and S. Haddadin, 'A Unified Framework for External Wrench Estimation, Interaction Control and Collision Reflexes for Flying Robots', in *IROS 2014*, Chicago, IL, USA, Sep. 2014.

[4] ——, 'Simultaneous estimation of aerodynamic and contact forces in flying robots: Applications to metric wind estimation and collision detection', in *ICRA 2015*, Seattle, WA, USA, May 2015, pp. 5290–5296. DOI: `10.1109/ICRA.2015.7139937`.

[5] T. Tomić, M. Maier, and S. Haddadin, 'Learning quadrotor maneuvers from optimal control and generalizing in real-time', in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1747–1754. DOI: `10.1109/ICRA.2014.6907087`.

[6] T. Tomić, K. Schmid, P. Lutz, A. Mathers, and S. Haddadin, ‘The Flying Anemometer: Unified Estimation of Wind Velocity from Aerodynamic Power and Wrenches’, in *IROS 2016*, IEEE, Daejeon, Korea, Oct. 2016, pp. 1637–1644.

[7] T. Tomić, ‘Evaluation of acceleration-based disturbance observation for multicopter control’, in *Control Conference (ECC), 2014 European*, Jun. 2014, pp. 2937–2944. DOI: `10.1109/ECC.2014.6862237`.

[8] T. Tomić and S. Haddadin, ‘Towards Interaction, Disturbance and Fault Aware Flying Robot Swarms’, in *Proceedings of the 18th International Symposium on Robotics Research (ISRR)*, Puerto Varas, Chile, Dec. 2017.

Patents

[1] T. Tomić, S. Haddadin, and P. Lutz, ‘Windmessung mittels eines Multikopters’, Patentanmeldung DE 10 2016 119 152.3 (im Prüfungsverfahren).

Curriculum Vitae

Personal information

Name	Teodor Tomić born on Aug 4, 1986 in Požega, Croatia

Education

–2000	OŠ Vilima Korajca, Kaptol, Croatia
2000–2004	Gimnazija Požega, Požega, Croatia

Higher education

2004–2008	Bachelor of Mechanical Engineering, Mechatronics and Robotics, Faculty of Mechanical Engineering and Naval Architecture (FAMENA), University of Zagreb, Zagreb, Croatia
2008–2010	Master of Mechanical Engineering, Mechatronics and Robotics, Faculty of Mechanical Engineering and Naval Architecture (FAMENA), University of Zagreb, Zagreb, Croatia
04/2014 –	Pursuing Dr.-Ing at Institute of Automatic Control, Leibniz Universität Hannover (LUH), Hanover, Germany

Work experience

10/2008–10/2010	Guidance, Navigation and Control Systems Engineer, Hipersfera d.o.o, Zagreb, Croatia
03/2011–03/2017	Research Scientist, German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Weßling, Germany
03/2017–	Software Engineer, Skydio, Redwood City, CA, USA

Acknowledgments

The last 7 years have been a period of personal and professional growth. I am thankful for being given the rare opportunity to work in intelectually inspiring and demanding environments that have pushed me to new heights. Most of the work relating to this thesis was done during my 6 years at the German Aerospace Center (DLR) Institute of Robotics and Mechatronics, whereas for the last year thereof I have been at Skydio. First of all, I would like to thank Selma for the love, support and engaging conversations during our complicated and complex time together, while remaining on the left half-plane. Finally there is one fewer imaginary pole. I would also like to thank my close family for the support throughout the years.

The most credit for finishing this thesis goes to Prof. Haddadin. Sami has not only been a true mentor, but also a good friend. By always demanding the best, without compromises, he taught me how to approach problems with good scientific questions and to never quit pushing towards set goals. I will remember the sleepless nights before deadlines, and the long train rides filled with interesting discussions.

I am grateful to Alin Albu-Schäffer, Michael Suppa and Korbinian Schmid for giving me the opportunity to join and work with the amazing people at the DLR Institute of Robotics and Mechatronics. I have learned a lot working with Korbinian, during our (sometimes literally bloody) long hours working on autonomous quadcopters, doing experiments for papers and preparing demos. Heiko Hirschmüller and Armin Wedler have held our back to allow us to work on our research. I have also learned a great deal from Christian Ott, who was a very supportive department head. Due to my long tenure at DLR, there are figuratively countless people to thank for engaging discussions. Entropy nevertheless dictates that the list of people is finite. I would therefore like to thank Michael Kaßecker for the endless discussions about all thing mechanical and electrical, and other social ineptitude (like computer science, life, the universe, and everything). The institution of Florian Schmidt, always ready to help, listen, talk, and party, will always remain an inspiration. I am also thankful to have worked with Elmar Mair, Felix Rueß, Moritz Maier, Christoph Brand, Martin Schuster, Iris Grixa, Samantha Stoneman, Marcus Müller, for reasons that would cumulatively be too long to list. I have also learned a great deal from discussions with Daniel Seth, Oli Eiberger, Robert Haslinger, Alexander Werner, Johannes Englsberger, Oliver Porges, and almost everyone at DLR. It has been a great period in my career and life.

Relating to wind tunnel experiments and development of the DLR Ardea hexacopter, I would like to thank Philipp Lutz for developing the electronics, embedded software and infrastructure; Ingo v. Bargen for experimentation and mechanical designing; Harald Wagner for manufacturing, design ideas and laminating when it was not really needed; Tilo Wüsthoff for the excellent and engaging designs; Markus Grebenstein for general discussion regarding science and mechatronics. Lastly, Andrew Mathers has supported the experiments at WindEEE in London, Canada. Ingo Kossyk and Saumitro Dasgupta have been a great help on the machine learning side of things.

Lastly, I really appreciate the opportunity to work with people that make up Skydio, and their support and understanding for the writing of this thesis, most notably Hayk Martyros, Adam Bry, Abraham Bachrach, Rowland O'Flaherty and Saumitro Dasgupta. I am thankful to have had the opportunity to implement parts of this thesis into a useful consumer product.