Control and Estimation Strategies for Autonomous MAV Landing on a Moving Platform in Turbulent Wind Conditions

by

Aleix París i Bordas

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2020

© Massachusetts Institute of Technology 2020. All rights reserved.

Author .................................................................

Department of Aeronautics and Astronautics

May 19, 2020

Certified by .............................................................

Jonathan P. How

Richard C. Maclaurin Professor of Aeronautics and Astronautics

Thesis Supervisor

Accepted by .............................................................

Sertac Karaman

Associate Professor of Aeronautics and Astronautics

Chair, Graduate Program Committee
Control and Estimation Strategies for Autonomous MAV Landing on a Moving Platform in Turbulent Wind Conditions

by

Aleix París i Bordas

Submitted to the Department of Aeronautics and Astronautics on May 19, 2020, in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics

Abstract

Micro aerial vehicles (MAVs) are increasing in popularity both for recreational and commercial purposes. In particular, autonomous MAVs are promising in the field of drone delivery due to their scalability, deployability, and low cost compared to traditional ground delivery. Hybrid systems, which combine MAVs taking off and landing from/to trucks and the ground vehicles themselves to increase package delivery efficiency, present even more advantages. Nevertheless, these systems require reliable control strategies that account for the challenging environment present in the vicinity of moving ground vehicles, as well as techniques to estimate these conditions.

This thesis presents a novel planning and control strategy that allows a fast, autonomous landing of a micro aerial vehicle (MAV) on a moving ground vehicle, which will be required for truck-drone delivery systems. The turbulent wind conditions near the landing platform are measured online using small, inexpensive whisker-like sensors. The measurements from these sensors are then used by an unscented Kalman filter to estimate the wind speed acting on the MAV, which is then compensated for by a boundary layer sliding controller. The experiments performed validate the robustness of the approach, which allows fast, dynamic landings of a MAV in moving ground vehicles in challenging environments without the need for hovering above the landing platform first.

Thesis Supervisor: Jonathan P. How
Title: Richard C. Maclaurin Professor of Aeronautics and Astronautics
Acknowledgments

First, I would like to thank my advisor Jonathan How for his help and guidance. His insights have been of great value for the papers I have published during my Master’s, and for this thesis. Thank you as well for giving me the opportunity to study my M.S. at MIT, and thanks to Ford Motor Company for providing me funding to do so.

I am very grateful to my colleagues at the Aerospace Controls Laboratory. Jesus, Samir, Dong Ki, Jeremy, Andrea, Lena, Parker, Michael, Stewart, Rose, Sebastian, Brett, Kaveh... Thank you all for your feedback, fruitful discussions, and for being a source of inspiration. Bryt, thanks for your help handling my frequent purchase orders.

I would also like to thank my family away from home. Toni, Bruno, Antonella, Íñigo, Juanjo, Marc, and the MIT Cheerleading team: thanks for making sure I spent time outside the lab too.

Of course, I also want to acknowledge my partner Ruiya, for being very understanding and supportive during stressful times. The motivation and energy you always give me are invaluable.

Last, and definitely not least, my deepest gratitude goes to my family. Jordi, Gerard, this would not have been possible without you. You are a source of strength and support, and I know I can always count on you. Thanks, a lot.
## Contents

1 Introduction 13  
1.1 Motivation .................................................. 13  
1.2 Literature Review ............................................ 14  
1.3 Thesis Aims and Outline ..................................... 16  

2 Dynamic Landing with Wind Measured a Priori 19  
2.1 Introduction .................................................. 19  
2.1.1 Contributions .............................................. 20  
2.2 System Overview ............................................. 21  
2.2.1 Finite State Machine ...................................... 22  
2.2.2 Trajectory Planning: Model Predictive Control ......... 22  
2.2.3 Ancillary Controller: Boundary Layer Sliding Controller ... 24  
2.2.4 Landing Platform Estimation: Extended Kalman Filter ... 26  
2.2.5 Visual Detection: AprilTag Visual Fiducial System ...... 27  
2.3 Experimental Results ......................................... 28  
2.3.1 Hardware ................................................... 28  
2.3.2 Static Platform Experiments ............................... 29  
2.3.3 Moving Platform Experiments ............................. 32  

3 Wind Estimation 35  
3.1 Introduction .................................................. 35  
3.2 Sensor Design ................................................ 37  
3.2.1 Initial Sensor Tests ........................................ 37
# List of Figures

2-1 Dynamic landing maneuver on a moving platform .......................... 20  
2-2 Dynamic landing trajectory ...................................................... 20  
2-3 Diagram of the system’s architecture ......................................... 21  
2-4 Finite state machine ............................................................... 22  
2-5 Quadrotor vehicle used for the dynamic landing experiments .......... 28  
2-6 Leaf blower array and static platform ....................................... 30  
2-7 Mean wind speed and standard deviation from the leaf blowers ....... 30  
2-8 Static platform, standard BLSC ................................................ 31  
2-9 Static platform, disturbance-aware BLSC ................................... 31  
2-10 Landing platform with vertical tag bundle ................................ 33  
2-11 Ground vehicle leaf blowers assembly .................................... 33  
2-12 Landing platform on dolly towed by UGV ................................. 34  
2-13 Moving platform, disturbance-aware BLSC ............................... 34  

3-1 MAV with four bio-inspired sensors ......................................... 36  
3-2 Bend Labs sensor ................................................................. 37  
3-3 Bend Labs sensors on hexarotor .............................................. 38  
3-4 Airflow sensor ........................................................................ 40  
3-5 Model-based approach diagram ............................................... 42  
3-6 Learning-based approach diagram ........................................... 46  
3-7 Sensor deflection angle vs wind speed ...................................... 49  
3-8 Model- and learning-based relative velocity comparison ............... 51  
3-9 Estimated wind speed 2D plot ................................................. 53
3-10 Simultaneous estimation of drag and interaction force – setup . . . . 55
3-11 Simultaneous estimation of drag and interaction force – plots . . . . 55

4-1 Gazebo environment . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
4-2 Moving platform, standard BLSC . . . . . . . . . . . . . . . . . . . . . 62
4-3 Moving platform, disturbance-aware BLSC with online wind estimation 63
List of Tables

3.1 RMS between LSTM and UKF .................................. 54

4.1 Wind parameters ................................................. 61
Chapter 1

Introduction

1.1 Motivation

Autonomous unmanned aerial vehicles (UAVs) are becoming more popular in industry because of their flexibility and fast deployment. UAVs have demonstrated their utility in applications such as aerial photography for topology and agriculture [1–4], search and rescue operations [5, 6], and mapping [7–9], both alone and in drone formations [10, 11]. The large recent growth in online shopping has also attracted interest in reducing package shipment time and costs, and UAVs provide an efficient alternative to delivery trucks [12, 13]. However, their payload and flight time is limited, so several researchers have investigated using a truck-drone delivery system [14, 15]. The UAVs in current truck-drone delivery methods can only take off and land when the truck is stopped visiting a customer node, which has substantial synchronization costs. Thus, autonomous landing on a moving truck offers some further efficiencies [16].

Additionally, the deployment of micro aerial vehicles (MAVs) in uncertain and constantly changing atmospheric conditions [17–19] requires the ability to estimate and adapt to disturbances such as the aerodynamic drag force applied by wind gusts. Simultaneously, as many new interaction-based missions [20–22] arise, so increases the need to better differentiate between forces caused by aerodynamic disturbances and other sources of interaction [23–26]. Differentiating between aerodynamic drag force and interaction force can be extremely important for safety reasons. For example, the
controller of a robot should react differently depending on whether a large disturbance is caused by a wind gust, or by a human trying to interact with the machine [27].

Distinguishing between drag and interaction disturbances can be challenging, as they both apply forces to the center of mass (CoM) of the multirotor that cannot be easily differentiated by examining the inertial information commonly available from the robot’s onboard IMU or odometry estimator. Successful approaches for this task include a model-based method that measures the change in thrust-to-power ratio of the propellers caused by wind [28] and an approach which monitors the frequency component of the total disturbance (estimated via inertial information) to distinguish between the two possible sources of force [29].

This thesis presents a strategy for simultaneously estimating the interaction force and the aerodynamic drag disturbances using novel bio-inspired, whisker-like sensors that measure the airflow around a multirotor, as shown in Fig. 3-1. The estimated wind speed is then used by an ancillary controller that is able to reject the disturbances acting on the MAV and allows a fast landing maneuver on a moving platform.

1.2 Literature Review

Autonomous UAV landing has been investigated by several researchers. Ref. [30] landed a drone on a static kayak in a reservoir with mild wind and water ripple conditions, but the image processing was done off-board and the landing time was close to 1 min. Ref. [31] developed a system capable of landing a UAV on a moving platform with PID control, an EKF estimator, and an AprilTag visual fiducial bundle. However, the computation was also done off-board, and there was no relative wind present: the platform’s speed was just 0.18m/s and the tests were done indoors. Ref. [32] demonstrated a quadrotor landing on a moving platform using only onboard sensing and computation, but the environment was not turbulent due to the tests being indoors and the platform was moving relatively slowly at 1.2m/s. Furthermore, besides two cameras, a distance sensor was required to estimate the UAV-platform relative pose. Ref. [33] demonstrated an autonomous landing of a quadrotor on a car
moving at 14m/s by detecting an AprilTag on its roof. A proportional navigation-based guidance law was used for the approach phase, and a PID controller for the landing phase, with no disturbance rejection considerations. The landing maneuver consisted of acquiring the tag while hovering, and the descent was initiated once the quadrotor stabilized over it. Additionally, the quadrotor used was large and had a broad sensor suite, including a downward-facing camera, a three-axis gimballed camera to track the target, and an inertial navigation system. Furthermore, besides the ground vehicle’s GPS coordinates, the quadrotor also used the ground vehicle’s IMU data to improve its estimation of the landing platform’s state. Ref. [34] also demonstrated an outdoors landing, but the landing platform’s speed was only 0.5m/s and the landing maneuver took 12–20 s to complete. Moreover, the ground vehicle’s wheel encoders data was used to estimate its state. Ref. [35] used model predictive control (MPC) to land a quadrotor on a moving ship. Both the UAV and the ship collaborated to reach their goal. The waves were modeled as sinusoidal, but the wind disturbances considered were not turbulent – this approach only compensated the effects caused by a steady-state wind. Furthermore, the vehicle needed to hover above the platform for at least 5 s, and the ship was simulated. A simulated boat landing was also carried out in [36], where the platform’s state was estimated fusing GPS and visual measurements. This work incorporated a velocity feed-forward term to the controller to catch the platform, but its only consideration of external wind was an offset of the hovering position to ensure the target was inside the field of view of the downwards-facing camera, and the landing maneuver lasted 24 s. Ref. [37] developed an adaptive controller to track a ground vehicle with only relative position data from ArUco tags, and tested it in outdoor experiments at 5.6 m/s. But this approach only considered disturbances due to ground effect, and the landing maneuver needed 20 s for tracking and 10 s for descent.

In summary, most current approaches for MAV landing on moving platforms involve hovering above the platform for a period of time to visually acquire it, which is then followed by a relatively slow descent. In contrast, the approach presented in this thesis investigates a direct trajectory to the landing platform, which has the pos-
sible advantages of providing faster landings and enabling several UAVs to approach
the platform at the same time from different directions (improving the utilization of
this limited resource). Additionally, most of current approaches do not make special
considerations to reject the turbulent wind present near the target vehicle, and thus
safety in challenging conditions cannot be guaranteed.

In regard to distinguishing between interaction and aerodynamic disturbance,
most of the current approaches focus on the estimation of one or the other disturbance
only.

Aerodynamic disturbances: Accurate wind or airflow sensing is at the heart of
the techniques employed for aerodynamic disturbance estimation. A common strategy
is based on directly measuring the airflow surrounding the robot via sensors, such
as pressure sensors [38], ultrasonic sensors [39], or whisker-like sensors [40]. Other
strategies estimate the airflow via its inertial effects on the robot, for example using
model-based approaches [41, 42], learning-based approaches [43, 44], or hybrid (model-
based and learning-based) solutions [45].

Generic wrench-like disturbances: Multiple related works focus instead on es-
timating wrench disturbances, without explicitly differentiating for the effects of the
drag force due to wind: [22–24, 46] propose a model-based approach which utilizes an
unscented Kalman filter (UKF) for wrench estimation, while [47] proposes a factor
graph-based estimation scheme.

1.3 Thesis Aims and Outline

This thesis presents a system capable of landing a MAV on a moving platform, even in
the presence of turbulent wind. First, a boundary layer sliding controller (BLSC) that
takes into account these conditions is presented, as well as a planner with changing
objectives that allows a fast maneuver, and a vision-based extended Kalman filter
(EKF) to estimate the moving platform’s state. The UAV’s state estimator used a
motion capture system, but the important information in the demonstrated maneuver
(the relative position of the UAV and the moving platform) is estimated onboard by
the quadrotor using a vision system. Then, an unscented Kalman filter (UKF) is
presented, which uses bio-inspired sensors to estimate online the wind speed acting
on the vehicle. Both a model- and a learning-based strategy are introduced and
compared.

The previous section identified a research gap in the fields of control and estimation
for MAVs in windy conditions. Therefore, this thesis has three main goals:

- To develop a planning and control strategy that allows an UAV to fly in chal-
  lenging conditions and land rapidly on a moving platform

- To design a reliable wind estimation technique

- To validate, both in simulation and in hardware experiments, the robustness of
  the approach

The remainder of this thesis is organized as follows:

Chapter 2 introduces the control strategy used in this work, a boundary layer
sliding controller, as well as a planning strategy that allows a MAV to land on a
moving platform in turbulent wind conditions from which the mean and standard
deviation have been measured beforehand.

Chapter 3 presents model- and deep learning-based approaches to estimate the
wind acting on a MAV using bio-inspired, whisker-like sensors.

The strategies presented in Chapter 2 and Chapter 3 are used simultaneously in
Chapter 4 to demonstrate a MAV landing on a moving platform in turbulent wind
conditions measured in real time. The wind speed estimation obtained with a UKF
is sent to the ancillary boundary layer sliding controller to reject the disturbances
acting on the vehicle.

Finally, Chapter 5 concludes this thesis by summarizing it, presenting its contri-
butions and limitations, and proposing future work.
Chapter 2

Dynamic Landing with Wind
Measured a Priori

This chapter is based on [26]. This paper was accepted to the International Conference on Robotics and Automation (ICRA) 2020, taking place virtually from May 31 to August 31.

2.1 Introduction

Autonomous landing on a moving platform presents unique challenges for multirotor vehicles, including the need to accurately localize the platform, fast trajectory planning, and precise/robust control. Previous works studied this problem but most lack explicit consideration of the wind disturbance, which typically leads to slow descents onto the platform. This chapter presents a fully autonomous vision-based system that addresses these limitations by tightly coupling the localization, planning, and control, thereby enabling fast and accurate landing on a moving platform. The platform’s position, orientation, and velocity are estimated by an extended Kalman filter using simulated GPS measurements when the quadrotor-platform distance is large, and by a visual fiducial system when the platform is nearby. The landing trajectory is computed online using receding horizon control and is followed by a boundary layer sliding controller that provides tracking performance guarantees in the presence of un-
known, but bounded, disturbances. To improve the performance, the characteristics of the turbulent conditions are accounted for in the controller. The landing trajectory is fast, direct, and does not require hovering over the platform, as is typical of most state-of-the-art approaches (see Fig. 2-2). The experiments presented validate the robustness of the approach.

Figure 2-2. Dynamic landing trajectory (green) compared with traditional landing maneuvers on moving platforms (red). An AprilTag bundle composed of 2 tags is placed in the ground vehicle to allow accurate detection by the MAV with its onboard camera, as explained in Section 2.2.5. The proposed approach is fast and efficient, and will be crucial for truck-drone delivery systems.

2.1.1 Contributions

This chapter demonstrates a vision-based system capable of dynamic landing (i.e., the multirotor does not need to hover above the vehicle before descending) which also
accounts for the turbulent conditions that would be present near a rapidly-moving ground vehicle. The resulting framework allows thus a maneuver that will be crucial for efficient truck-drone delivery systems. The contributions of this chapter are:

- Demonstration of a boundary layer sliding controller to incorporate and compensate for turbulence based on a model of the conditions near the landing platform.
- An algorithm for computing fast, vision-based dynamic landing maneuvers, is demonstrated in hardware experiments that include challenging turbulent wind conditions.

### 2.2 System Overview

This chapter addresses the current limitations of landing on a moving platform by:

1. Using optimization-based trajectory generation to enable dynamic landing; and

2. Using robust control to explicitly compensate for turbulent wind conditions

The system that achieves this is comprised of several components, described in this section and shown in Fig. 2-3.
2.2.1 Finite State Machine

The quadrotor’s behavior is determined by a finite state machine (FSM) shown in Fig. 2-4 and comprised of four states:

1. **Stand By**: This is the initial state of the quadrotor, which consists of taking off and hovering at a predefined altitude above the starting point. The FSM then transitions to **Search** mode.

2. **Search**: The quadrotor uses simulated GPS coordinates of the unmanned ground vehicle (UGV) —as explained in Section 2.2.4— to predict a rendezvous location and flies there. When the front-facing camera detects the landing platform as described in Section 2.2.5, the state automatically switches to **Landing**.

3. **Landing**: In this mode, the quadrotor approaches the target following a direct trajectory towards it. When the distance and relative velocity UAV-UGV are below threshold values, the quadrotor switches to the **End** mode. If the last detection happened more than 0.8s ago, the mode returns to **Search**.

4. **End**: motors are stopped, maneuver is finished.

2.2.2 Trajectory Planning: Model Predictive Control

The planner solves a convex optimization problem with changing objectives depending on the state of the FSM. In **Search** mode, the UAV-UGV rendezvous point is predicted by assuming a constant linear velocity and yaw rate for the UGV, and a constant velocity for the UAV. This position is then offset by a small distance backwards in
the direction of the UGV to ensure target detection by the front-facing camera, and is used by the planner as the final position of the trajectory, while the final velocity is the UGV’s. The time taken to reach the UGV is minimized, to reduce delivery turnarounds. In *Landing* mode, the planner initially minimizes the jerk to obtain a trajectory which ensures adequate tag acquisition. As the UAV approaches the target, the effect of disturbances increases so the planner minimizes the time spent in the turbulent area. The final position of the planned trajectory is the vertical tag’s position (offset a few cm backwards) and predicted ahead (by an amount that depends on the computation time) assuming constant linear/angular velocities. The final velocity of the trajectory is set to match the UGV’s.

The minimum jerk approach produces smooth trajectories and has a long heritage for planning quadrotor paths [32, 48, 49]. The trajectory is re-planned using an MPC approach every time a new estimate of the platform’s position is obtained. CVXGEN [50] generated the code to solve the following convex optimization problem

\[
\min_{x,u} J = \sum_{t=0}^{N} \ell (\ddot{u}, d)[t]
\]

subject to
\[
x[t+1] = Ax[t] + Bu[t]
\]
\[
|a[t]|_\infty \leq a_{\text{max}}
\]
\[
|\ddot{u}[t]|_\infty \leq j_{\text{max}}
\]
\[
x[0] = x_0, \ x[N] = x_f
\]

for \( t = 0 \ldots N \)

where \( N \) is the timestep when the quadrotor has to reach the final state \( x_f \), \( \ell \) is a quadratic cost function, \( \ddot{u} \) is the open-loop control input (the jerk of the trajectory), \( d \) is the UAV-tag distance, \( x \) is the position, velocity, and acceleration of the UAV, \( A \) and \( B \) are, respectively, the state and input matrices for a triple integrator, and \( a \) is the subvector of \( x \) representing the acceleration. Note that it is possible to plan using this linear model for a nonlinear system because the nonlinear dynamics of the quadrotor are canceled by the ancillary controller, as explained next.
2.2.3 Ancillary Controller: Boundary Layer Sliding Controller

While MPC has been used extensively in industry [51], the control of systems with nonlinear dynamics requires expensive optimization. Sliding control [52] has proven to be effective in quadrotors [53, 54]. This control strategy guarantees bounds on the tracking error and has been combined with MPC [55, 56]. In this thesis’ approach, a disturbance-aware boundary layer sliding controller (BLSC) is derived, which is a nonlinear ancillary controller that models the disturbances found near the landing platform and leverages sliding control techniques.

In quadrotors, the attitude dynamics are much faster than the position dynamics, and thus control of both can be decoupled [57]: the output of a controller is the setpoint for the other. The position and velocity controller is derived in this section to account for the turbulent wind present near the landing platform, and the attitude control is performed by a quaternion-based controller [58].

The following derives the BLSC. Define the manifold $S(t)$ by $s = \dot{x} + \lambda \ddot{x} = 0$, where $\ddot{x} = x - x_d$ and $\lambda > 0$. The objective of sliding control is to maintain $s = 0$ at all times. If the control action’s frequency is high enough, zero tracking error is guaranteed [52]. This high control action is impractical in many applications because of actuator limits and the excitation of unmodelled dynamics. An approach taken in [53, 56] is BLSC, where the control discontinuity is smoothed in a thin boundary layer of thickness $\Phi$:

$$B := \{ s : |s| \leq \Phi \}$$

(2.2)

Consider a system whose dynamics can be expressed as

$$\ddot{x} = f(x) + b(x) u + d$$

(2.3)

where $d$ is the disturbance. Then, the BLSC strategy is

$$u = \hat{b}^{-1} \left[ \ddot{x}_d - \lambda \dot{x} - \hat{f}(x) - K \text{sat} \left( \frac{s}{\Phi} \right) \right]$$

(2.4)

where $\text{sat}(\cdot)$ is the saturation function, $\hat{f}$ is the estimated acceleration caused by drag,
and $K$ is determined by the uncertainty in the dynamics and the disturbance of the system. As noted before, in (2.1) it is possible to plan using linear MPC because of the cancellation of $f$ in (2.3) and (2.4).

The leaf blowers generate turbulence as shown in Fig. 2-1 and Fig. 2-6. The turbulent wind parameters are the mean $v_w$ and standard deviation $\sigma$ of the speed. Define $V = v + v_w$ where $v$ is the quadrotor’s speed. Then, $V$ is the total wind speed relative to the UAV and $\hat{f}$ is $\hat{f} = \hat{c} \|V\|V$, where $\hat{c}$ is the estimated drag coefficient of the quadrotor.

The quadrotor plans a trajectory to approach the UGV in the direction it is facing to match its speed, and thus is never outside the wind field generated by the leaf blowers during the landing maneuver. Therefore, it is reasonable to assume that this wind field is constant in the directions perpendicular to where the leaf blowers point to. The true acceleration caused by drag is

$$f = c \|V \pm 2\sigma u_w\| (V \pm 2\sigma u_w)$$

where $u_w$ is a unit vector in the direction of the wind.

The variation of $b$ is very small for a UAV with constant weight. Therefore, $\beta = (b_{max}/b_{min})$ is approximately 1, where $b_{max}$ and $b_{min}$ are the maximum and minimum control gains respectively (or throttle gains in the context of quadrotors). Thus, $K$ is simplified as [52] $K = \bar{F} + \eta$, where $\eta > 0$ is a constant in the sliding condition

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s|.$$  

The larger the $\eta$, the faster the system will reach the sliding surface. Nevertheless, $K$ should only be as large as the disturbance magnitude requires to avoid a high-frequency control signal. $\bar{F}$ is

$$F = |f - \hat{f}| \leq \bar{F}$$

$$\bar{F} = |(\hat{c} + \tilde{c}) \|V \pm 2\sigma u_w\| (V \pm 2\sigma u_w) - \hat{c} \|V\|V|$$

25
where \( \tilde{c} > 0 \) is a bound on the absolute value of the drag coefficient error \( |c - \tilde{c}| \). By taking the sign that makes this coefficient larger, \( K \) is defined. In this application, the quadrotor moves towards the generated wind and therefore this occurs when the 2\( \sigma \) is increasing the magnitude of \( v_w \).

### 2.2.4 Landing Platform Estimation: Extended Kalman Filter

To estimate the state of the moving platform, an extended Kalman filter (EKF) is used. This filtering algorithm minimizes the mean of the squared error and has demonstrated its efficacy in robot localization [59–61]. The state vector of the platform is \( x_p = [p_x, p_y, v_p, \theta, \dot{\theta}]^\top \), where \( p_x, p_y \) are the 2D coordinates, \( v_p \) is the magnitude of the velocity, \( \theta \) is the orientation angle with respect to the \( x \)-axis, and \( \dot{\theta} \) is the angular velocity. Since real-world roads are mostly horizontal and in particular the experiments in this thesis were carried on completely flat surfaces, the velocity in the \( z \)-direction is not estimated. The moving platform is modeled as a unicycle with dynamics

\[
\dot{x}_p(t) = f_p(x_p(t)) + w(t),
\]

where \( w(t) \) is the process noise, assumed to be a white Gaussian noise. A constant linear and angular velocity are considered, and the UGV dynamics are

\[
\begin{align*}
\dot{p}_x &= v_p \cos(\theta), & \dot{v}_p &= 0, \\
\dot{p}_y &= v_p \sin(\theta), & \dot{\theta} &= 0.
\end{align*}
\]

The measurement vector is \( z_p = [p_x, p_y, \theta]^\top \), and when a measurement is received, the EKF approach is used to perform an update of the estimated state. The measurements are obtained in two ways. First, when the quadrotor is far from the platform (that is, in the Search state defined in Section 2.2.1), these measurements are obtained by adding a white Gaussian noise to the ground truth pose of the vehicle (obtained from the motion capture data) to simulate inaccurate GPS measurements that a ground vehicle could provide to the UAV for the rendezvous. Note that receiving \( \theta \)
is not necessary to estimate the orientation of a moving platform because its motion could be used to infer that quantity. Nevertheless, $\theta$ measurements are used in this work, which enables testing for static platform experiments. The update frequency is 2 Hz, which is realistic for UAV applications [62]. These first set of measurements are simply used to help the UAV locate the ground vehicle, but that information could be estimated without requiring a link between the two vehicles.

Second, and most importantly, when the quadrotor is near the ground vehicle and detects the onboard tag, both the simulated GPS and the visual detection measurements are fused to estimate $x_p$. The vision-based detection is explained in the next subsection, and it provides a far more accurate estimate of the position and orientation of the tag/platform. The 2D positions $p_x$ and $p_y$ and the heading angle $\theta$ are then used to update the platform’s state, and the estimated velocity $v_p$ is incorporated into the vector $x_f$ in (2.1) to ensure the quadrotor matches the moving platform’s speed at the landing point.

### 2.2.5 Visual Detection: AprilTag Visual Fiducial System

When the UAV is relatively close to the platform, visual estimation provides more accurate UAV-UGV poses than GPS. The AprilTag visual fiducial system [63] obtained them, and a ROS wrapper [64] based on AprilTag 2 [65] to interface with the core detection algorithm. A tag bundle is a set of several coplanar tags used simultaneously by the visual fiducial system: the algorithm extracts the information of all of them to estimate a single “bundle pose”. Thus, they are useful when accurate detection is required, which is the case in this thesis. Additionally, by using tags of different sizes, detection at various distances is ensured. The tag bundle used consisted of a 14×14 cm tag on top of a 5×5 cm tag. Despite the relatively small bundle size, it can be detected at a maximum distance of approximately 3.5 m, and a minimum distance of about 5 cm. The bundle can be seen in Fig. 2-10.
2.3 Experimental Results

In this section, the experimental results are presented and analyzed. A video of the flights is available at [66].

2.3.1 Hardware

Static and moving platform landing tests were done (see Fig. 2-1 and Fig. 2-6), both of which had turbulent wind at the landing site from the leaf blowers. The quadrotor used for the hardware experiments (Fig. 2-5) weighs 0.564kg including the 1500mAh 3S battery. It is $36 \times 29$cm (approximately half the platform size) and its thrust-to-weight ratio is 1.75. The onboard computer is a Qualcomm Snapdragon Flight APQ8074, whose front-facing camera provides $640 \times 480$ black-and-white images at a rate of 30fps to the AprilTag detection module. Hover tests were carried out to determine $\hat{b}$ in (2.4). To measure the drag coefficient $\hat{c}$ and the bound on its error $\bar{c}$, tests with the quadrotor flying in front of a strong wind were performed, and the accurate pitch angle was obtained by the motion capture system. By balancing forces, the drag could be determined, yielding $\hat{c}$. Additionally, the IMU utils package
was used to compute the Snapdragon’s IMU accelerometer and gyroscope noise density and bias random walk. The \texttt{kalibr} visual-inertial calibration toolbox then used this IMU intrinsic information to find the camera-IMU transform [68].

\subsection*{2.3.2 Static Platform Experiments}

\subsubsection*{Experiment Setting}

The first set of hardware experiments presents a static platform in front of an array of 5 leaf blowers, as shown in Fig. 2-6. The leaf blowers are set at two different heights and point to the negative $x$-direction. The platform is composed of a $60 \times 60$cm horizontal plate and a $60 \times 30$cm vertical plate, which has attached the tag bundle described in Section 2.2.5. Kraft paper covers the horizontal plate to avoid skidding of the MAV upon landing. As it can be seen in the videos, the kraft paper shakes violently due to the large magnitude of the turbulent wind.

The parameters $v_w$ and $\sigma$ of the turbulent wind were measured at distances $l$ from the vertical platform every 0.5m until $l = 3.5$m, which is approximately the tag detection range. For every $l$, a measurement of the speed was taken every second for 60s using a high-precision hot-wire anemometer. Figure 2-7 shows the data obtained. Interestingly, at $l = 0.5$m, the mean speed decreases while the standard deviation increases to its maximum value. This might be caused by the vortices that are generated when the flow traverses the vertical platform.

\subsubsection*{Results}

The first set of experiments tested a standard BLSC that does not take into account the turbulent wind generated by the leaf blowers, that is, the factors $\hat{f}$ and $K$ only consider the drag generated by the quadrotor’s speed relative to the ground. Figure 2-8 shows the tracking and estimation performance obtained. The quadrotor starts at $p_q = (-3.5, 2.0, 1.3)\text{m}$, and the platform is located at $p_p = (1.1, -1.1, 0.8)\text{m}$. As expected, the tracking performance is poor, and the landing takes a long time: 27.2s since the first tag detection, which occurs at $t_d = 3.3$s (indicated with a vertical dashed
Figure 2-6. Leaf blower array in front of the static platform showing the wind measurement procedure. To the left, a hot-wire anemometer on top of a tripod takes measurements to characterize the wind field. To the right, the 5 leaf blowers generate a wind of 6-8 m/s at the landing platform. The kraft paper covering the black horizontal plate flutters rapidly due to the turbulent wind.

Figure 2-7. Mean wind speed (in m/s) from the leaf blowers vs. distance to the vertical platform (with $1 - \sigma$ error bars).
Figure 2-8. Position and velocity tracking and estimation performance of a static platform experiment using a BLSC that does not model the turbulence. Wind blowing towards \(-x\) resulting in poor tracking performance in the \(x\)-direction. Vertical dashed lines indicate time of the first tag detection, \(t_d\).

Figure 2-9. Position and velocity tracking and estimation performance of a static platform experiment using the disturbance-aware BLSC. The tracking error remains small during the flight. (line). Also note that the platform’s estimation improves after \(t_d\). In the video, the covariance ellipses for the tag’s 2D position can be seen, and decrease abruptly in size at \(t_d\).

Next, the disturbance-aware BLSC was used with the same initial conditions as in Fig. 2-8 to compare its improvement. The results are shown in Fig. 2-9. It can be seen that the tracking is much better, and it only worsens slightly when the quadrotor is inside the wind field, after the time of the first tag detection \(t = 2.5s\). The landing time is just 3.2s (measured since the first tag detection) even in challenging conditions, which is 8.5 times faster compared to the standard BLSC.
2.3.3 Moving Platform Experiments

Experiment Setting

To fully test the approach presented in this chapter, landing experiments on a moving
platform were also performed. The *Clearpath Jackal* was used as the ground vehicle
that tows a dolly were the landing platform is mounted (see Fig. 2-1, Fig. 2-10, and
Fig. 2-12). On top of the UGV, two cordless leaf blowers were placed on a mount so
that the expelled air pointed to the top of the vertical pane of the landing platform,
as shown in Fig. 2-11. This provided turbulent air at the landing pad even though
the MAV and UGV were not moving very quickly, therefore recreating the conditions
found on a vehicle moving at a high speed outdoors. The frame is made of 80/20
T-slot aluminum bars, which are easy to cut and attach together. Therefore, they
are useful for rapid prototyping. The mount is covered by 5 Delrin plates that can
be attached and detached using velcro. The laptop that controls the ground vehicle
is housed inside the mount, and the two cordless leaf blowers are placed on top. The
distance from the leaf blowers to the platform is such that this turbulent wind follows
the same plot as Fig. 2-7.

Results

A standard BLSC was also tested first for this experiment setting. There was consid-
erable tracking error and the quadrotor was not able to land on the platform before
the vehicle arrived at the final point (see [66] for details).

The results obtained using the disturbance-aware BLSC are shown in Fig. 2-
13. The quadrotor starts at \( \mathbf{p}_q = (-1.5, 2.7, 0.9) \)m, and the UGV starts at \( \mathbf{p}_p = (-3.9, 0.1, 0.4) \)m. When the maneuver begins, the UGV is commanded to move at
1m/s and rotate to its right at a rate of 4\(^\circ\)/s. The tag is detected at \( t_d = 4.7 \)s, and
the landing time is 6.8s. Note that the quadrotor is at approximately 4m from the
moving platform at \( t_d \), a value larger than for the static experiment (1.5m).
Figure 2-10. Landing platform with vertical tag bundle. The MAV can detect the upper tag at a maximum distance of approximately 3.5 m, and the lower tag at a minimum distance of 5 cm. Yellow pieces of thread were added to visualize the wind, as seen in the video [66].

(a) Metal mount on top of the ground vehicle. The strong mount, made of 80/20 T-slot aluminum bars, houses the ground vehicle’s laptop and supports the leaf blowers.

(b) Two cordless leaf blowers on top of the ground vehicle. Delrin plates cover the frame for aerodynamic and aesthetic purposes.

Figure 2-11. Assembly of the ground vehicle with two cordless leaf blowers. This system allows the recreation of turbulent conditions even at small speeds of the ground vehicle.
Figure 2-12. Landing platform on a dolly towed by the UGV, showing two cordless leaf blowers mounted on top of the *Clearpath Jackal* to recreate the conditions of a vehicle moving outdoors at high speed.

Figure 2-13. Position and velocity tracking and estimation performance of a moving platform experiment using the disturbance-aware BLSC. The quadrotor quickly matches the UGV’s velocity and is able to track successfully the planned trajectory.
Chapter 3

Wind Estimation

This chapter is based on [69], a paper submitted to the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020.

3.1 Introduction

Disturbance estimation for MAVs is crucial for robustness and safety. This chapter uses novel, bio-inspired airflow sensors to measure the airflow acting on a MAV, and this information is fused in a UKF to simultaneously estimate the three-dimensional wind vector, the drag force, and other interaction forces (e.g. due to collisions, interaction with a human) acting on the robot. To this end, a fully model-based and a deep learning-based strategy are presented and compared. The model-based approach considers the MAV and airflow sensor dynamics and its interaction with the wind, while the deep learning-based strategy uses a long short-term memory (LSTM) to obtain an estimate of the relative airflow, which is then fused in the proposed filter. Hardware experiments validate the methods, showing that relative airflow of up to 4 m/s can be accurately estimated, and that drag and interaction forces can be differentiated.

The approach presented in this chapter takes inspiration from the way insects sense airflow [70], which is by measuring the deflections caused by the aerodynamic drag force acting on the appendix of some receptors. By fusing the information of
four heterogeneous airflow-sensors distributed across the surface of the robot, it is possible to obtain a three-dimensional estimate of the relative velocity of the MAV with respect to the surrounding airflow. This information is then fused in a UKF-based force estimator that uses an aerodynamic model together with the robot’s pose and velocity to predict the wind, the expected drag force, and other interaction forces.

To account for the complex aerodynamic interactions between sensors and propellers [71, 72], this model-based approach (based on first-order physical principles) is extended with a data-driven strategy. This strategy employs a recurrent neural network (RNN) based on an LSTM network to provide an estimate of the relative airflow of the robot, which is then fused in the proposed model-based estimation scheme. Experiments show that the approach achieves an accurate estimate of the relative airflow with respect to the robot with velocities up to 4 m/s, and enables interaction forces and aerodynamic drag forces to be distinguished. Based on the experimental results, the model-based and learning-based approaches are compared, highlighting their advantages and disadvantages.

The contributions of this chapter are:

- Model- and deep learning-based strategies to simultaneously estimate wind, drag force, and other interaction forces using novel bio-inspired sensors similar to the ones discussed in [73, 74]; and
Experimental validation of the approaches, showing that relative airflow of up to 4 m/s is accurately estimated, and interaction force and aerodynamic forces can be distinguished.

3.2 Sensor Design

3.2.1 Initial Sensor Tests

In order to measure the relative wind in 3D affecting a MAV, lightweight and multidirectional sensors need to be used. The two-axis Bend Labs Digital Flex Sensor [75] shown in Fig. 3-2 was used for initial tests and as a proof of concept. It contains two compliant capacitors inside, which are sensitive to strain. When the sensor is deflected, the difference in capacitance from the interior and exterior capacitors is linearly proportional to the deflection angle. Since the total amount of bending is integrated along the length of the sensor, the measured angle $\Delta \theta$ only depends on the relative angle of the base and the tip of the sensor, a property called path independence [75].

The outputs are, therefore, the roll and pitch deflection angles of the tip of the sensor. It was interfaced via I2C with an Nvidia Jetson TX2 mounted onboard the MAV. Figure 3-3 shows two sensors attached to the hexarotor flown for the hardware
experiments of this chapter. The sensor on the left, mounted vertically, is covered with wire along its length, to make it stiffer, and has a fin appendix made of foam attached to increase its drag coefficient thus amplifying its sensitivity to relative wind. The sensor on the right, mounted horizontally, does not have any modifications. As it can be observed, gravity deflects it downwards.

While this initial testing showed that the relative airflow effects on the deflection angles of the sensors could be observed, their low stiffness, large size, and weight made them impractical to be used for this application. The sensors were largely affected by inertia, and oscillated every time the MAV changed its acceleration.

### 3.2.2 Sensor Design and Considerations

The previous subsection indicated the need to find a better airspeed sensor. To this end, this thesis adopts sensors based in [73, 74], which are lighter, more economical,
and stiffer compared with the Bend Labs sensors described in Section 3.2.1. The sensors, shown in Fig. 3-4, consist of a base and an easily-exchangeable tip. The base is composed of a magnetic field sensor connected to a conditioning circuit that interfaces with the robot via I2C and a 3D-printed case that encloses the sensor. The tip consists of a planar spring mounted in a 3D-printed enclosure that fits with the base, with a permanent magnet attached to its bottom and a carbon-fiber rod glued on the spring’s top. Eight foam fins are attached on the other end of this rod. When the sensor is subjected to airflow, the drag force from the air on the fins causes a rotation about the center of the planar spring which results in a displacement of the magnet. This displacement is then measured by the magnetic sensor. The fins are placed with even angular distribution in order to achieve homogeneous drag for different airflow directions. Foam and carbon fiber were chosen as the material of the fin structure due to their low density, which is crucial to minimize the inertia of the sensor. Therefore, this sensor has a low response to inertia but a high response to drag, as desired. See [73] for more information about the sensor characteristics and manufacturing procedure.

Due to the complex aerodynamic interactions between the relative airflow and the blade rotor wakes, the sensor placement needs to be chosen carefully [71, 72]. To determine the best locations, short pieces of string were attached both directly on the vehicle and on metal rods extending away from it horizontally and vertically. Then, the hexarotor was flown indoors and it was observed that the pieces of string on top of the vehicle and on the propeller guards were mostly unaffected by the blade rotor wakes. Therefore, these are the two locations chosen to mount the sensors, as seen in Fig. 3-1. They are distributed so that the relative airflow coming from any direction excites at least one sensor (that is, for at least one sensor, the relative airflow is not aligned with its length).

3.2.3 Sensor Measurements

The sensors detect the magnetic field $\mathbf{b} = (b_x, b_y, b_z)$, but the model outlined in Section 3.3.2 requires the deflection angles of the $i$th sensor $\theta_x,i$ and $\theta_y,i$, which correspond...
Figure 3-4. Illustration of an airflow sensor and its reference frame $S$, with the main components labeled.

to the rotation of the carbon fiber rod about the $x$ and $y$ axes in reference frame $S_i$. At the spring’s equilibrium, the rod is straight and $b = (0, 0, b_z)$, where $b_z > 0$ if the magnet’s north pole is facing the carbon-fiber rod. The angles are then

$$
\theta_i = \begin{bmatrix}
\theta_{x,i} \\
\theta_{y,i}
\end{bmatrix} = \begin{bmatrix}
-\arctan(b_y/b_z) \\
\arctan(b_x/b_z)
\end{bmatrix}
$$

Note that if the magnet was assembled with the south pole facing upward instead, $-b$ must be used in Eq. (3.1).

### 3.3 Model-based Approach

This section presents the model-based approach used to simultaneously estimate airflow, interaction force, and aerodynamic drag force on a MAV. The estimation scheme is based on the UKF [76] approach presented in previous works [22, 24], augmented
with the ability to estimate a three-dimensional wind vector via the relative airflow measurements provided by the whiskers.

A diagram of the most important signals and system-level blocks related to this approach is included in Fig. 3-5.

**Reference frame definition** The reference frames used in this chapter are an inertial reference frame W, a body-fixed reference frame B attached to the CoM of the robot, and the i-th sensor reference frame \( S_i \), with \( i = 1, \ldots, N \), as shown in Fig. 3-4.

### 3.3.1 MAV Dynamic Model

The MAV considered has a mass \( m \) and inertia tensor \( J \), and the dynamic equations of the robot can be written as

\[
\begin{align*}
\dot{w}\mathbf{p} &= w\mathbf{v} \\
\dot{R}^B_W &= R^B_W[B\omega \times] \\
m_w\dot{v} &= R^B_W f_{cmd} + w f_{drag} + m_w g + w f_{touch} \\
J_B\dot{\omega} &= -B\omega \times J_B\omega + B\tau_{cmd}
\end{align*}
\]

where \( \mathbf{p} \) and \( \mathbf{v} \) represent the position and velocity of the MAV, respectively, \( R^B_W \) is the rotation matrix representing the attitude of the robot (i.e., such that a vector \( w\mathbf{p} = R^B_W B\mathbf{p} \)), and \( [\times] \) denotes the skew-symmetric matrix. The vector \( B f_{cmd} = B e_3 f_{cmd} \) is the thrust force produced by the propellers along the \( z \)-axis of the body frame, \( w g = -w e_3 g \) is the gravitational acceleration, and \( w f_{touch} \) is the interaction force expressed in the inertial frame. For simplicity, it is assumed that interaction and aerodynamic disturbances do not cause any torque on the MAV, due to its symmetric shape and the fact that interactions (in this thesis' hardware setup) can only safely happen in proximity of the center of mass of the robot. Vector \( B\tau_{cmd} \) represents the torque generated by the propellers and \( B\omega \) the angular velocity of the MAV, both expressed in the body reference frame. Here \( f_{drag} \) is the aerodynamic drag force on
Figure 3.5. Diagram of the most important signals used by each step of the proposed model-based approach for simultaneous estimation of wind, drag force, and interaction force.

The robot, expressed as an isotropic drag [77]

\[ \mathbf{w} \mathbf{f}_{\text{drag}} = (\mu_1 v_\infty + \mu_2 v_\infty^2) \mathbf{w} \mathbf{e}_{v_\infty} = \mathbf{f}_{\text{drag}} \mathbf{w} \mathbf{e}_{v_\infty} \]

\[ \mathbf{w} \mathbf{e}_{v_\infty} = \frac{\mathbf{w} \mathbf{v}_\infty}{v_\infty}, \text{ where } v_\infty = \|\mathbf{w} \mathbf{v}_\infty\|, \] (3.3)

\( \mathbf{w} \mathbf{v}_\infty \) is the velocity vector of the relative airflow acting on the CoM of the MAV (expressed in the inertial frame)

\[ \mathbf{w} \mathbf{v}_\infty = \mathbf{w} \mathbf{v}_{\text{wind}} - \mathbf{w} \mathbf{v}, \] (3.4)

and \( \mathbf{w} \mathbf{v}_{\text{wind}} \) is the velocity vector of the wind expressed in the inertial frame.

### 3.3.2 Airflow Sensor Model

The \( i \)-th airflow sensor is assumed to be rigidly attached to the body reference frame \( B \), with \( i = 1, \ldots, N \). The reference frame of each sensor is translated with respect to \( B \) by a vector \( B_r S_i \) and rotated according to the rotation matrix \( R^B_{S_i} \). To derive a model of the whiskers subject to aerodynamic drag, several assumptions are made.
Each whisker is massless; its tilt angle is not significantly influenced by the accelerations from the base $B$ (due to the high stiffness of its spring and the low mass of the fins), but is subject to the aerodynamic drag force $f_{\text{drag},i}$.

Another assumption made is that each sensor can be modeled as a stick hinged at the base via a linear torsional spring. Each sensor outputs the displacement angle $\theta_{x,i}$ and $\theta_{y,i}$, which correspond to the rotation of the stick around the $x$ and $y$ axis of the $S_i$ reference frame. The aerodynamic drag force acting on the aerodynamic surface of each sensor $S_i f_{\text{drag},i}$ can be expressed as a function of the (small) angular displacement

$$S_{x,y} S_i f_{\text{drag},i} \approx \begin{bmatrix} 0 & \frac{k_i}{l_i} \\ -\frac{k_i}{l_i} & 0 \end{bmatrix} \begin{bmatrix} \theta_{x,i} \\ \theta_{y,i} \end{bmatrix} = K_i \theta_i$$

where $k_i$ represents the stiffness of the torsional spring, $l_i$ the length of the sensor, and

$$S_{x,y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

captures the assumption that the aerodynamic drag acting on the $z$-axis of the sensor is small (given the fin shapes) and has a negligible effect on the sensor deflection.

Regarding the aerodynamic force acting on a whisker, non-isotropic drag is assumed, proportional to the squared relative velocity w.r.t. the relative airflow

$$S_i f_{\text{drag},i} = \frac{\rho}{2} c_{D,i} A_i \| S_i v_{\infty,i} \| S_i v_{\infty,i}$$

where $\rho$ is the density of the air, $c_D$ is the aerodynamic drag coefficient, $A_i = \text{diag}([a_{xy,i}, a_{xy,i}, a_z]^{\top})$ is the aerodynamic section of each dimension, and $c_{D,i}$ the corresponding drag coefficient. Due to the small horizontal surface of the fin of the sensor, it is assumed that $a_z = 0$. The vector $S_i v_{\infty,i}$ is the velocity of the relative airflow experienced by the $i$-th whisker, and expressed in the $i$-th whisker reference frame, and can be obtained as

$$S_i v_{\infty,i} = R_B^{S_i \top} (B v_{\infty} - B \omega \times B r_{S_i})$$
where $Bv_\infty$ is the relative airflow in the CoM of the robot expressed in the body frame, given by:

$$Bv_\infty = RW^T_w v_\infty = RW^T_w(v_{\text{wind}} - w_v) \quad (3.9)$$

### 3.3.3 Model-based Estimation Scheme

#### Process Model, State and Output

The MAV dynamic model described in Eq. (3.2) is discretized, augmenting the state vector with the unknown wind $Wv_{\text{wind},k}$ and unknown interaction force $Wf_{\text{touch},k}$ that are to be estimated. It is assumed that these two state variables evolve as:

$$Wf_{\text{touch},k+1} = Wf_{\text{touch},k} + \epsilon_{f,k}$$
$$Wv_{\text{wind},k+1} = Wv_{\text{wind},k} + \epsilon_{v,k} \quad (3.10)$$

where $\epsilon_{f,k}$ and $\epsilon_{v,k}$ represent the white Gaussian process noise, with covariances used as tuning parameters.

The full, discrete time state of the system used for estimation is

$$x_k^T = \{Wp_k^T, q_{W,k}^B, Wv_k^T, B\omega_k^T, Wf_{\text{touch},k}^T, Wv_{\text{wind},k}^T\} \quad (3.11)$$

where $q_{W,k}^B$ is the more computationally efficient quaternion-based attitude representation of the robot, obtained from the rotation matrix $R_{W,k}^B$.

The filter output is then

$$y_k^T = \{Wf_{\text{touch},k}^T, Wv_{\text{wind},k}^T, Bv_\infty,k^T, Wf_{\text{drag},k}^T\} \quad (3.12)$$

where $Wf_{\text{drag},k}$ is obtained from Eq. (3.3) and Eq. (3.4), and $Bv_\infty,k$ is obtained from Eq. (3.9).
Measurements and Measurement Model

Two sets of measurements are available asynchronously:

**Odometry** The filter fuses odometry measurements (position $\hat{p}_k$, attitude $\hat{\mathbf{q}}_{B,k}$, linear velocity $\hat{\mathbf{v}}_k$ and angular velocity $\hat{\mathbf{\omega}}_k$) provided by a cascaded state estimator

$$z_{\text{odometry},k}^\top = \{W\hat{p}_k^\top, \hat{\mathbf{q}}_{W,k}^B, W\hat{\mathbf{v}}_k^\top, B\hat{\mathbf{\omega}}_k^B\}$$  \hspace{1cm} (3.13)

the odometry measurement model is linear, as shown in [22].

**Airflow sensors** The $N$ sensors are sampled synchronously, providing the measurement vector

$$z_{\text{airflowsensor},k}^\top = \{\hat{\theta}_1^k, \ldots, \hat{\theta}_N^k\} = \hat{\theta}_k^T$$  \hspace{1cm} (3.14)

The associated measurement model for the $i$-th sensor can be obtained by combining Eq. (3.5) and Eq. (3.7)

$$\theta_{i,k} = \frac{D}{2}\mathbf{c}_D\mathbf{K}_i^{-1}S_{x,y}A_i S_{\infty,i,k} S_{\infty,i,k}$$  \hspace{1cm} (3.15)

where $S_{\infty,i,k}$ is obtained using information about the attitude of the robot $\mathbf{q}_{W,k}^B$, its velocity $W\mathbf{v}_k$, and angular velocity $B\mathbf{\omega}_k$, and the estimated windspeed $W\mathbf{v}_{\text{wind}}$ as described in Eq. (3.8) and Eq. (3.9). The synchronous measurement update is obtained by repeating Eq. (3.15) for every sensor $i = 1, \ldots, N$.

**Prediction and Update Step**

**Prediction** The prediction step (producing the a priori state estimate) [76] is performed using the unscented quaternion estimator (USQUE) [78] prediction technique for the attitude quaternion. The process model is propagated using the commanded thrust force $f_{\text{cmd}}$ and torque $B\mathbf{\tau}_{\text{cmd}}$ output of the position and attitude controller on the MAV.
Update The odometry measurement update step is performed using the linear Kalman filter update step [76], while the airflow-sensor measurement update is performed via the Unscented Transformation [76] due to the non-linearities in the associated measurement model.

3.4 Deep Learning-based Approach

This section presents a deep-learning based strategy, which makes use of a RNN based on the LSTM architecture to create an estimate of the relative airflow $\mathbf{v}_\infty$ using the airflow sensors and other measurements available on board of the robot. The complexity in modeling the effects of the aerodynamic interference caused by the airflow between the propellers, the body of the robot and the surrounding air, as observed in the literature [71, 72] and in this chapter’s experimental results, motivates the use of a learning-based strategy to map sensors’ measurement to relative airflow.
3.4.1 Output and Inputs

The output of the network is the relative airflow $Bv_\infty$ of the MAV. The inputs to the network are the airflow sensor measurements $\theta$, the angular velocity of the robot $B\omega$, the raw acceleration measurement from the IMU and the normalized throttle commanded to the six propellers (which ranges between 0 and 1). The sign of the throttle is changed for the propellers spinning counterclockwise, in order to provide information to the network about the spinning direction of each propeller. The reason for the choice of the input is dictated by the derivation from the model-based approach: from Eq. (3.8) and Eq. (3.7) it can be observed that the relative airflow depends on the angle of the sensors and on the angular velocity of the robot. The acceleration from the IMU is included to provide information about hard to model effects, such as the orientation of the body frame w.r.t. gravity (which causes small changes in the angle measured by the sensors), as well as the effects of accelerations of the robot. Information about the throttle and spinning direction of the propellers is instead added to try to capture the complex aerodynamic interactions caused by their induced velocity. Every output and input of the network is expressed in the body reference frame in order to make the network invariant to the orientation of the robot, thus potentially reducing the amount of training data needed.

3.4.2 Network Architecture

An LSTM architecture is employed, which is able to capture time-dependent effects [79, 80], such as, in the case of this thesis, the dynamics of the airflow surrounding the robot and the dynamics of the sensor. A 2-layer LSTM was chosen, with the size of the hidden layer set to 16 (with the input size, 20, and the output size, 3). A single fully connected layer is added to the output of the network, mapping the hidden layer into the the desired output size.
3.4.3 Interface with the Model-based Approach

The UKF treats the LSTM output as a new sensor which provides relative airflow measurements $B\hat{v}_\infty$, replacing the airflow sensor’s measurement model provided in Section 3.3. The output of the LSTM is fused via the measurement model in Eq. (3.9), using the unscented transform. A block-diagram representing the interface between the learning-based approach and the model-based approach is represented in Fig. 3-6.

3.5 Experimental Evaluation

3.5.1 System Identification

Drag Force

Estimating the drag force acting on the vehicle is required to differentiate from force due to relative airflow and force due to other interactions with the environment. To this purpose, the vehicle was commanded to follow a circular trajectory at speeds of 1 to 5 m/s, keeping its altitude constant (see Section 3.5.2 for more information about the trajectory generator). In this scenario, the thrust produced by the MAV’s propellers $\hat{f}_{\text{thrust}}$ is

$$\hat{f}_{\text{thrust}} = \frac{m}{\cos \phi \cos \theta} g$$

(3.16)

where $m$ is the vehicle’s mass, $g$ is the gravity acceleration, and $\phi$ and $\theta$ are respectively the roll and pitch angles of the MAV. The drag force is then

$$\hat{f}_{\text{drag}} = (B\hat{\mathbf{f}}_{\text{thrust}} - m B\hat{\nu}) \cdot B\mathbf{e}_v$$

(3.17)

where $B\hat{\mathbf{f}}_{\text{thrust}} = [0, 0, \hat{f}_{\text{thrust}}]$, and $B\mathbf{e}_v$ is the unit vector in the direction of the vehicle’s velocity in body frame. By fitting a second-degree polynomial to the collected data, $\mu_1 = 0.20$ and $\mu_2 = 0.07$ are obtained (see Eq. (3.3)).
Sensor Parameters Identification

The parameters required to fuse the output $\theta_i$ of $i$-th airflow sensor are its position $B \mathbf{r}_S$ and rotation $R^S_B$ with respect to the body frame $B$ of the MAV, and a lumped parameter coefficient $c_i$ mapping the relative airflow $S_i \mathbf{v}_{\infty,i}$ to the measured deflection angle $\theta_i$. The coefficient $c_i = \frac{\rho}{2} C_D a_{xy} \frac{\mathbf{a}_{xy} \mathbf{a}_{xy}}{k \ell}$ can be obtained by re-arranging Eq. (3.15) and by solving

$$c_i = \frac{\|\theta_i\|}{\|S_i \mathbf{v}_{\infty,i}\|\|\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} S_i \mathbf{v}_{\infty,i}\|}$$

and the velocity $S_i \mathbf{v}_{\infty,i}$ is obtained from indoor flight experiments (assuming no wind, so that $W \mathbf{v}_{\infty} = -W \mathbf{v}$), or by wind tunnel experiments. Wind tunnel experiments have also been used to validate this chapter’s model choice (quadratic relationship between wind speed and sensor deflection), as show in Fig. 3-7. Furthermore, these experiments also confirmed the assumption on the structure of $A_i$, i.e., the variation of the sensor’s deflection with respect to the direction of the wind speed is small and therefore it can be considered that $a_x = a_y = a_{xy}$.
**LSTM Training**

The LSTM is trained using two different datasets collected in indoor flight. In the first flight the hexarotor follows a circular trajectory at a set of constant velocities ranging from 1 to 5 m/s, spaced of 1 m/s each. In the second data-set the robot is commanded via a joystick, making aggressive maneuvers, while reaching velocities up to 5.5 m/s. Since the robot flies indoor (and thus wind can be considered to be zero) the relative airflow of the MAV $Bv_\infty$ corresponds to its estimated velocity $-Bv$, which is used to train the network. The network is implemented and trained using PyTorch [81]. The data is pre-processed by re-sampling it at 50 Hz, since the inputs of the network used for training have different rates (e.g. 200 Hz for the acceleration data from the IMU and 50 Hz from the airflow sensors). The network is trained for 400 epochs using sequences of 5 samples of length, with a learning rate of $10^{-4}$, using the Adam optimizer [82] and the mean squared error (MSE) loss. Unlike the model-based approach, the LSTM does not require any knowledge of the position and orientation of the sensors, nor the identification of the lumped parameter for each sensor. Once the network has been trained, however, it is not possible to reconfigure the position or the type of sensors used.

### 3.5.2 Implementation Details

**System Architecture**

The custom-built hexarotor used weighs 1.31 kg. The pose of the robot is provided by a motion capture system, while odometry information is obtained by an estimator running on-board, which fuses the pose information with the inertial data from an IMU. The algorithms run on the onboard Nvidia Jetson TX2 and are interfaced with the rest of the system via ROS. Aerospace Controls Laboratory’s snap stack [83] controls the MAV.
Sensor Driver

The sensors are connected via I2C to the TX2. A ROS node (sensor driver) reads the magnetic field data at 50 Hz and publishes the deflection angles as in Eq. (3.1). Slight manufacturing imperfections are handled via an initial calibration of offset angles. The sensor driver rejects any measured outliers by comparing each component of $b$ with a low-pass filtered version. If the difference is large, the measurement is discarded, but the low-pass filter is updated nevertheless. Therefore, if the sensor deflects very rapidly and the measurement is incorrectly regarded an outlier, the low-pass filtered $b$ quickly approaches the true value and consequent false positives do not occur.

![Figure 3-8](image.png)

Figure 3-8. Comparison of the relative velocity estimated by the model based (UKF) and the learning-based (LSTM) approaches. The assumption is that the ground truth (GT) is given by the velocity of the robot.

Trajectory Generator

A trajectory generator ROS node commands the vehicle to follow a circular path at different constant speeds or a line trajectory between two points with a maximum desired velocity. This node also handles the finite state machine transitions: take off,
flight to the initial position of the trajectory, execution of the trajectory, and landing where the vehicle took off. This trajectory generator is used to identify the drag coefficient of the MAV (see Section 3.5.1), to collect data for training, and to execute the experiments described below.

3.5.3 Relative Airflow Estimation

For this experiment, the vehicle was commanded with a joystick along the flight space at different speeds, to show the ability of the approach to estimate the relative airflow. Since the space is indoors (no wind), it is assumed that the relative airflow is opposite to the velocity of the MAV. Thus, the velocity of the MAV (obtained from a motion capture system) is compared to the opposite relative airflow estimated via the model-based strategy and the deep-learning based strategy.

Figure 3-8 shows the results of the experiment. Each subplot presents the velocity of the vehicle in body frame. The ground truth (GT) in red is the MAV’s speed obtained via the motion capture system, the green dotted line represents the relative airflow velocity in body frame $-B\bm{v}_\infty$ as estimate via the deep-learning based strategy (LSTM), and the blue dashed line represents $-B\bm{v}_\infty$ as estimated by the the fully model-based strategy (UKF). The root mean squared errors of the UKF and LSTM’s estimation for this experiment are shown in Table 3.1. The results demonstrate that both approaches are effective, but show that the LSTM is more accurate.

3.5.4 Wind Gust Estimation

To demonstrate the ability to estimate wind gusts, the vehicle was flown in a straight line commanded by the trajectory generator outlined in Section 3.5.2 along the diagonal of the flight space while a leaf blower was pointing approximately to the middle of this trajectory. Figure 3-9 shows in red the estimated wind speed of the UKF drawn at the 2D position where this value was produced, and in green the leaf blower pose obtained with the motion capture system. As expected, the wind speed is increased in the area affected by the leaf blower.
Figure 3-9. In this plot the vehicle is flown in a straight line at high speed, from left to right, while a leaf blower (shown in black) aims at the middle of its trajectory. The red arrows indicate the intensity of the estimated wind speed.
<table>
<thead>
<tr>
<th>Method</th>
<th>RMS error $x$</th>
<th>RMS error $y$</th>
<th>RMS error $z$</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKF</td>
<td>0.44</td>
<td>0.34</td>
<td>0.53</td>
<td>m/s</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.38</td>
<td>0.31</td>
<td>0.28</td>
<td>m/s</td>
</tr>
</tbody>
</table>

### 3.5.5 Simultaneous Estimation of Drag and Interaction Forces

The approach presented can differentiate between drag and interaction forces, which is shown in the following experiments. There are four main parts to the experiment: hovering with no external force, hovering in a wind field generated by two leaf blowers, simultaneously pulling the vehicle with a string attached to it while the vehicle is still immersed in the wind field, and turning off the leaf blowers so that there is only interaction force. A video of the experiment is available at [84]. Figure 3-10, a frame from the video, shows the experimental setup: two leaf blowers point to the MAV while it is hovering, and a yellow string attached to the MAV is used to pull it and thus cause an interaction force. Figure 3-11 shows the forces acting on the MAV in world frame estimated by the UKF: $\mathbf{w}f_{\text{drag}}$ and $\mathbf{w}f_{\text{touch}}$. As expected, the drag force is close to zero when no wind is present even when the MAV is pulled, and similarly the interaction force is approximately zero when the vehicle is not pulled even when the leaf blowers are acting on it. Therefore, drag and interaction forces are differentiated correctly. Note that the leaf blowers turn on quickly and thus the drag force resembles a step, while the interaction force was caused by manually pulling the MAV with a string following approximately a ramp from 0 to 4 N as measured with a dynamometer. The UKF estimates $\mathbf{w}f_{\text{touch}}$ to about 6N, potentially due to inaccuracies of the external force ground truth measurement procedure and miscalibration of the commanded throttle to thrust mapping. As for the wind speed generated by the leaf blowers, it has an average value of 3.6 m/s at the distance where the vehicle was flying. According to the derived model, a drag force of approximately 1.2 N as shown in Fig. 3-11 should correspond to a wind speed of 3 m/s. The difference is due to the fact that the leaf blowers are not perfectly aimed to the MAV, and the wind field that they generate is narrow.
Figure 3-10. Experimental setup for simultaneous estimation of drag and interaction force. The leaf blowers eject wind to the MAV while it is being pulled by a yellow string. A qualitative visualization of the estimated drag and interaction forces can be seen to the bottom right. The blue and red arrows stemming from the MAV represent the estimated drag force and interaction force vectors.

Figure 3-11. Simultaneous estimation of drag and interaction force. Vertical bars separate the four phases of the experiment.
Chapter 4

Dynamic Landing with Wind Measured in Real Time

4.1 Introduction

Chapter 2 presented a control strategy and a planner that allows a fast landing of a MAV in turbulent wind conditions measured beforehand, and Chapter 3 proposed a method to estimate online the wind acting on the vehicle. Thus, the aim of this chapter is to demonstrate the full approach of simultaneously estimating the wind speed acting on the MAV and forwarding it to the disturbance-aware BLSC to allow a dynamic landing in challenging conditions.

4.2 Simulation Environment

This chapter shows the BLSC using the UKF that estimates the wind acting on the MAV. Contrary to the previous chapters, hardware experiments could not be performed due to the lack of access to the flight space during the COVID-19 pandemic.

The simulation environment consists of several components, detailed in this section.
4.2.1 MAV Simulation

The dynamics of a quadrotor are simulated using ACL’s snap sim [85]. Besides a physics engine, this ROS package implements software-in-the-loop (SIL) simulations of the Aerospace Controls Laboratory’s snap stack [83]. That is, the IMU and motion capture measurements are simulated realistically and interfaced with the rest of the components to allow the same inner-loop quaternion-based attitude controller and outer-loop disturbance-aware BLSC to be run without any changes in the code. The frequency of IMU measurements is 500 Hz, the same as in the Qualcomm Snapdragon Flight used for hardware experiments, and the simulated motion capture measurements frequency matches as well the real frequency of 100 Hz.

The forces and moments due to the following effects are simulated:

- MAV’s motors. A generic implementation allows any number of motors pointing to any direction. Additionally, the throttle to thrust and torque mapping can be specified and are set to match the experimentally obtained values.

- Drag force. The drag force acting on the MAV is simulated by assuming the model in Eq. (3.3).

- Wind speed. The wind acting on the MAV is simulated and subtracted from the MAV’s speed to calculate the total drag force. Section 4.2.3 details the implementation of the wind simulation.

- Gravity

4.2.2 Airspeed Sensor Simulation

In order to estimate wind as detailed in Chapter 3, the airflow sensors presented in Section 3.3.2 need to be simulated. For all the experiments presented in this chapter, the exact same 4 sensors in Fig. 3-1 have been reproduced. Each sensor is affected by the relative velocity as shown in Eq. (3.15), and has a specific drag coefficient $c_D$, length $l$, stiffness $k$, and aerodynamic section of each dimension $A_i$. For each sensor $i$
and timestep \( k \), the sensor’s \( \theta_{i,k} \) (a 2-dimensional vector that represents the roll and pitch angles) is simulated and sent to the UKF.

### 4.2.3 Wind Simulation

The Dryden turbulence model in [86] is used to replicate the turbulent conditions that would be found in hardware experiments were a MAV lands on a moving platform at a high speed. This turbulence model can be viewed as a linear time-invariant (LTI) single-input and single-output (SISO) system per component \( x, y, \) and \( z \). The input to the system is randomized, and the state is added to a certain nominal value. Therefore, this implementation of a Dryden turbulence model is similar to a random walk with a certain mean value (nominal wind) and standard deviation (wind gust).

To control the intensity of the turbulence, a constant \( c_t \geq 1 \) has been added, which increases the simulation speed. The constant \( c_t \) is nominally 1, but it can be increased to increment the changes in the generated wind.

Two wind generators run simultaneously:

- **Uniform wind generator**: represents uniform wind conditions, such as light gusts and a constant wind.
- **Ground vehicle wind generator**: simulates the turbulent wind present on the back of a ground vehicle, stronger the closer to this vehicle.

Each generator is characterized by the nominal wind speed \( w_n \) and standard deviation per component \( w_\sigma \). The total wind \( w^T \) experienced by the MAV is then

\[
w^T = w^U + e^{-d \tau} R^G_{WB} w^G
\]  

(4.1)

where \( w^U \) is the uniform wind, \( d \) is the distance between the MAV and the ground vehicle, \( \tau \) is a constant that characterizes the exponential decay of the turbulent wind generated by the ground vehicle, \( R^G_{WB} \) is the rotation matrix of the ground vehicle with respect to the inertial frame, and \( w^G \) is the ground vehicle wind.
4.2.4 UGV Simulation and Visualization

Gazebo was used to simulate the ground vehicle, the AprilTag bundle, and also for visualization of the landing maneuver. Figure 4-1 shows the tag bundle attached to the UGV, the simulated quadrotor performing the dynamic landing, and the images with resolution $640 \times 480$ from the simulated camera.

4.3 Results

This section presents the results obtained by applying the model-based wind estimation technique and the BLSC described previously, in the simulation environment, to allow a dynamic MAV landing on a moving platform. The platform is traveling towards the -x direction at a constant speed of 1 m/s. The wind parameters used are shown in Table 4.1.

Similarly to Chapter 2, a standard BLSC was tested first, and Fig. 4-2 shows the results. The first row represents the position for each component: MAV desired
Table 4.1. Wind parameters used for the simulations in this chapter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform generator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nominal wind speed</td>
<td>$w_n^U$</td>
<td>$[5, 3, 0]^\top$</td>
<td>m/s</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>$w^\sigma$</td>
<td>$[1, 1, 0.2]^\top$</td>
<td>m/s</td>
</tr>
<tr>
<td>Simulation speed const</td>
<td>$c_t^U$</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Ground vehicle generator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nominal wind speed</td>
<td>$w_n^G$</td>
<td>$[-2, 0, 0]^\top$</td>
<td>m/s</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>$w^\sigma$</td>
<td>$[3, 1.5, 0]^\top$</td>
<td>m/s</td>
</tr>
<tr>
<td>Simulation speed const</td>
<td>$c_t^G$</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>Inverse of the exponential decay constant</td>
<td>$\tau$</td>
<td>2</td>
<td>m</td>
</tr>
</tbody>
</table>

Position, MAV ground truth, UGV estimated position, and the UGV ground truth in blue, red, yellow, and purple respectively. The second row shows the difference in the position of the MAV and the UGV. Similarly as the first row, the third row shows the velocity for all three components. It can be seen that there is a constant tracking error due to the wind not being accounted for of approximately 2 m, and therefore landing cannot occur. This is expected, since there is a higher mean and standard deviation of wind – which is not accounted for – in the x-direction, as shown in the last row, and the UGV moves in this direction as well. The time of the first tag detection, $t_d$, is indicated with dashed vertical lines. Note that the wind becomes more turbulent after the MAV detects the tag and approaches the UGV. Also note that before the first tag detection, the estimation of the UGV’s y-position is not accurate (the error is approximately +60 cm), and the MAV moves in the positive y direction even though the landing platform is in the negative y direction with respect to the vehicle. This is due to consecutive noisy GPS measurements offset in the positive y, but the estimation error decreases abruptly when the tag is first detected, as is expected.

Then, a disturbance-aware BLSC that uses the wind estimated by the model-based UKF presented in Chapter 3 was used. The results are shown in Fig. 4-3. The plotted lines are the same as in the previous case, but the last row, which represents the wind speed for each component, includes the estimated wind speed in yellow. It can be seen that the tracking error remains small during all the maneuver, the MAV can reach the UGV, and landing occurs 5.1 s after the first tag detection (even with a stronger wind compared with the first case, of 5-7 m/s in the x-axis).
Figure 4-2. Moving platform experiment with a standard BLSC. The first and third rows show the position and velocity tracking and estimation performance of the simulated moving platform experiment. The second row shows the difference of the MAV and the UGV position, in black. It can be seen that the MAV is not able to reach the UGV, keeping a large tracking error in the x-direction. The last row shows the ground truth wind velocity, which becomes more turbulent near the UGV. Vertical dashed lines indicate time of the first tag detection, $t_d$. 
Figure 4-3. Moving platform experiment with the disturbance-aware BLSC and online wind estimation. The first and third rows show the position and velocity tracking and estimation performance of the simulated moving platform experiment. The second row shows the difference of the MAV and the UGV position, in black, which converges to 0 for all three components. It can be seen that the vehicle is able to reach the UGV and to track the desired trajectory. The last row shows the estimated and ground truth wind velocity, which becomes more turbulent near the UGV. Vertical dashed lines indicate time of the first tag detection, $t_d$. 
Chapter 5

Conclusion

5.1 Thesis Summary

This thesis has motivated the problem of MAV control in challenging environments and estimation of wind disturbances, and presented related research in the fields. This exposed a research gap, which this thesis aimed at solving.

Chapter 2 developed a boundary layer sliding controller to allow a quadrotor to fly in challenging conditions. An MPC planner designed a fast trajectory to land on a moving platform immersed in turbulent wind, which was estimated by an EKF that used the AprilTag visual fiducial system. The hardware experiments, both for a static and a moving platform, demonstrated the effectiveness of the approach. Nevertheless, the mean and standard deviation of the turbulent wind conditions needed to be measured beforehand.

To avoid offline measurements, Chapter 3 presented a model- and a learning-based approach to estimate the relative airflow, the drag force and the interaction force acting on a hexarotor using bio-inspired sensors. The results obtained in flight experiments showed that this approach allowed to accurately identify the relative airflow experienced by a multirotor in flight, and that wind gusts acting on the MAV could be detected. Experimental results showed that the proposed deep-learning based strategy is more accurate than the model-based strategy, and does not require a significant amount of training data. The deep-learning based strategy, however,
does not allow the flexibility to re-position or easily change sensors without having to re-train the network. Additionally, it was shown that the approach can correctly distinguish between drag and interaction forces.

Finally, Chapter 4 demonstrated the strategies presented in the previous chapters working jointly in simulation. The small tracking error and landing time showed that this thesis’ approach is promising in fields where accurate and quick landing of a MAVs is needed, such as in drone delivery.

5.2 Limitations and Future Work

Since the estimation part of this work focused on the relative pose of the MAV-UGV – by using the MAV’s onboard camera – and on the wind, the state estimation of the MAV relied on motion capture measurements provided by external cameras. Future work includes landing on the back of a pickup truck driving outdoors, to test this work’s approach in a more realistic setting. This will, therefore, require visual-inertial odometry (VIO) for onboard-only state estimation. Additionally, incorporating adaptation to allow for more varied flight conditions could be another research direction.

In regard to the learning-based approach for wind estimation, possible extensions to this work include further testing and comparison of different learning algorithms, data collection, and training strategies.
Bibliography


