External Wrench Estimation in Multi-rotor UAVs

Francesco Peracchia[1], Francesco Starna[1] and Francesco Vincelli[1]

[1]*Department of Computer, Control and Management Engineering, Sapienza Univerisity of Rome, Italy*

Abstract

A multirotor flying robot is moving in a simulated environment under wind influence and external force, in this paper is presented an implementation of an online algorithm for wrenches estimation, therefore this information is embedded in the quadrotor control and tested in a wide variety of situations, simulating a load charged by the drone. In order to determine contact events and localize the contact point during fly, we supplied our model to a further aerodynamic wrench estimation. Lack of access to real and coherent data for recreating an accurate aerodynamic behavior forced us to perform contact detection using an alternative approach. As discussed in detail in the relative paragraph, the discrimination between aerodynamics force and external contact force could not be achieved.

1. Introduction

Multi-rotor flying robots are increasing their level of autonomy, flying under a variety of conditions forced us to consider real-world situations. This class of robots is subject to local disturbances due to wind effects, collisions, or loss of propellers. Autonomous quadcopters are commonly developed for non-physical interaction scenarios, however many environments presents physical agents that could play an active role in design more robust and suitable models. For instance, tasks such as aerial shots and monitoring, surveillance, freight and goods delivery are nowadays the main goal for many robotics enterprises. Under real-world conditions such systems are usually subject to local disturbances due to wind, and potentially also faults (e.g. collisions, loss of propellers). The external wrench (forces and torques) $\boldsymbol{\tau}_e$ acting on flying robots can be interpreted as a sum of the aerodynamics wrench $\boldsymbol{\tau}_d$, the physical interaction wrench $\boldsymbol{\tau}_i$, and a fault wrench $\boldsymbol{\tau}_f$, such that: $\boldsymbol{\tau}_e = \boldsymbol{\tau}_d + \boldsymbol{\tau}_i + \boldsymbol{\tau}_f$. Therefore, robust operation under wind influence typically would require additional sensors to discriminate between these constituent wrenches. In this work we developed a trajectory tracking controller to make a quadrotor performing a trajectory tracking task on two path classes: a hovering path and an elliptical path, under the action of external disturbances due to contact and aerodynamic forces.

Elective in Robotics - Unmanned Aerial Vehicles, 2021, Sapienza, Rome

peracchia.1906895@studenti.uniroma1.it (F. Peracchia); starna.1613660@studenti.uniroma1.it (F. Starna); vincelli.1727959@studenti.uniroma1.it (F. Vincelli)

https://github.com/Starnino/ (F. Starna)

1906895 (F. Peracchia); 1613660 (F. Starna); 1727959 (F. Vincelli)

2. Related Work

2.1. Model-based wind speed estimation

In order to obtain the freestream velocity, aerodynamics model and onboard measurements in flight could be used. Different paradigms are generally used for this estimation process, these solutions are divided in two main categories Data-driven and Model-based approaches [1], a short introduction of both is proposed below.

Physical modeling: In general these approaches exploit the effects produced by blade flapping and propeller induced drag, which produce a horizontal force that can be measured by the onboard accelerometer. The blade flapping effect can be used to estimate the relative airspeed (Martin and Salaun)[2]. An alternative approach could be use a linear drag model combined with a model for the propeller as proposed by (Waslander and Wang)[3]. Blade flapping model is identified by flying in a wind tunnel and it is used for wind speed estimation (Sikkel et al.)[4]. These flapping effects and the induced drag could be combined in a nonlinear control scheme proposed by (Omari et al.)[5], finally a position controller proposed by (Huang et al.)[6] could improve tracking performance during aggressive maneuvers by handling the dependencies between thrust and free-stream velocity. (Tomic and Haddadin)[7] have shown how to incorporate this effect into external wrench estimation.

Data driven : in this estimation technique a regressor between a measured variable and the free-stream velocity is found. An important assumption is that: Aerodynamic and control forces are in equilibrium under static conditions. Hence it is possible to infer that the aerodynamic force is estimated indirectly from the position controller output. A heavy dependency on the controller and system parameters and the ill-suited effect for wind estimation during aggressive flight, encourage to use this technique only in restricted situations such as wind tunnel experiments. (Neumann and Bartholmai) [8] and (Ware and Roy) [9] use to relates the quadrotor pitch angle, as measured variable, to the wind speed.

2.2. Wind tunnel measurements

On a static quadrotor, (Schiano et al.)[10] and (Planckaert and Coton)[11] estimate the forces and torques that effects the robot, multicopter vehicles were studied by (Jung et al.)[12], they measure drag and thrust by comprehensive wind tunnel tests. Other studies relate the motor power in steady-state wind conditions to wind velocity, or using an horizontal wind tunnel during the fly, as proposed by(Ware and Roy)[9]. In order to estimate the aerodynamic model, (Sikkel et al.)[13] a quadcopter flew in tunnel, the model was based on blade flapping, and a learned aerodynamic model was used to estimate the wind velocity.

3. Preliminaries

3.1. Modeling

Quadrotors are a special subset of Vertical Take-Off and Landing (VTOLs) Unmanned Aerial Vehicles (UAV), which can be modeled as rigid bodies moving in free 3D space with respect to an inertial frame RF_i typically chosen North-East-Down (NED), with i pointing to the north, j pointing to the east, and k pointing down to the center of the earth. A frame RF_b is attached to the body B to its center of mass (CoM), having the same orientation of RF_i. According to the Newton-Euler formalism [14] [15] the equation of motion can be written as:

$$\mathcal{M}\ddot{\mathbf{r}} = \mathcal{M}g\mathbf{e}_3 + \mathbf{R}_b^i\mathbf{f} + \mathbf{R}_b^i\mathbf{f}_e \tag{1}$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} = -(\mathbf{I}\boldsymbol{\omega}) \times \boldsymbol{\omega} + \mathbf{m} + \mathbf{m}_e \tag{2}$$

$$\dot{\mathbf{R}_b^i} = \mathbf{R}_b^i(\boldsymbol{\omega})_\times \tag{3}$$

where (1) is the equation of the translational dynamics of the body expressed in the inertial frame RF_i, $\sum \mathbf{f}_i = \mathcal{M}\ddot{\mathbf{r}}$, (2) is the equation of the rotational dynamics of the body expressed in its body frame RF_b, $\sum \mathbf{m}_b = \mathbf{I}\dot{\boldsymbol{\omega}} + (\mathbf{I}\boldsymbol{\omega}) \times \boldsymbol{\omega}$, $\mathbf{r}$ is the position in the inertial frame RF_i, $\mathcal{M}$ the vehicle's mass, g is the acceleration of gravity, $\mathbf{e}_3$ is the z-axis unit vector, $\mathbf{f} = \begin{bmatrix} fx & fy & fz \end{bmatrix}^T$ is the control force in the body frame, $\mathbf{I}$ is the inertia matrix, $\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is the vehicle's angular velocity, $\mathbf{m} = \begin{bmatrix} m_\phi & m_\theta & m_\psi \end{bmatrix}^T$ is the control torque in the body frame, $\mathbf{f}_e$ and $\mathbf{m}_e$ are the external force and torque acting on the vehicle and expressed in the body frame. The external wrench is denoted as $\boldsymbol{\tau}_e = \begin{bmatrix} \mathbf{f}_e & \mathbf{m}_e \end{bmatrix}^T$ while the control wrench is $\boldsymbol{\tau} = \begin{bmatrix} \mathbf{f} & \mathbf{m} \end{bmatrix}^T$. $\mathbf{R}_b^i$, in the equation of the time derivative of a rotation matrix (3), is a roll-pitch-yaw rotation expressing the orientation of the body B with respect to the inertial frame RF_i

$$\mathbf{R}_b^i = \mathbf{R}_{XYZ} = \begin{pmatrix} cos(\psi)cos(\theta) & cos(\psi)sin(\phi)sin(\theta) - cos(\phi)sin(\psi) & sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta) \\ cos(\theta)sin(\psi) & cos(\phi)cos(\psi) + sin(\phi)sin(\psi)sin(\theta) & cos(\phi)sin(\psi)sin(\theta) - cos(\psi)sin(\phi) \\ -sin(\theta) & cos(\theta)sin(\phi) & cos(\phi)cos(\theta) \end{pmatrix} \tag{4}$$

An image description of the vehicle's dynamics parameters can be seen in Fig. 1. For further purposes, we rewrite the equations of motion in Lagrangian form as

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g} = \mathbf{J}^T\boldsymbol{\tau} + \boldsymbol{\tau}_e \tag{5}$$

where $\mathbf{g} = \begin{bmatrix} -\mathcal{M}g\mathbf{e}_3 & 0 \end{bmatrix}^T$, $\boldsymbol{\nu} = \begin{bmatrix} \dot{\mathbf{r}} & \boldsymbol{\omega} \end{bmatrix}^T$ is the twist, $\mathbf{J} = blockdiag\left\{\mathbf{R}_b^i, \mathbf{I}_{3\times3}\right\}$ and

$$\mathbf{M} = \begin{bmatrix} \mathcal{M}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I} \end{bmatrix}, \mathbf{C}(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -(\mathbf{I}\boldsymbol{\omega})\times \end{bmatrix} \tag{6}$$

The term $\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}$ represents the aerodynamics drag forces.

3.2. Control

The mathematical model of the system [16] is an affine system of the type $\dot{\boldsymbol{\xi}} = f(\boldsymbol{\xi}) + g(\boldsymbol{\xi})$, where the state is the 1×12 vector $\boldsymbol{\xi} = (\mathbf{r}, \dot{\mathbf{r}}, \boldsymbol{\phi}, \boldsymbol{\omega})$ and the input of the system is the 1×4

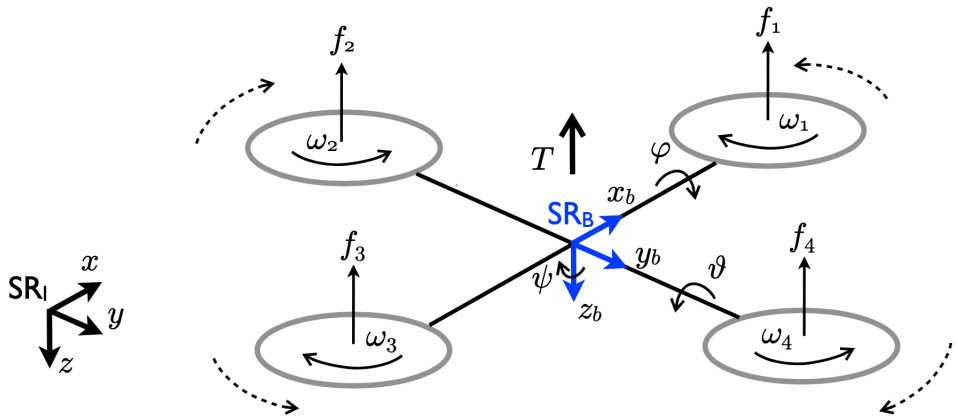

Figure 1: visualization of the quadrotor dynamic model. SR_I and SR_B are respectively the inertial and body frame.

vector $\mathbf{u} = (T, \mathbf{m})$, where the where the thrust $T = ||f_i||$ is the total force generated by the propellers in the inertial frame. The cascaded control structure is composed of a position tracking controller and an attitude controller. The position controller tracks a desired position $\mathbf{r}_d = [x_d\ y_d\ z_d]^T$, velocity $\dot{\mathbf{r}}_d$, and acceleration $\ddot{\mathbf{r}}_d$. The inertial control force is controlled by

$$\mathbf{f}_i = \mathbf{R}_{di}\mathbf{f} = \mathcal{M}(\ddot{\mathbf{r}}_d - K_{fd}\,\mathbf{e}_{\dot{r}} - K_{fp}\,\mathbf{e}_r) - \mathcal{M}g\mathbf{e}_3 - \mathbf{f}_e \tag{7}$$

with $K_{fp}, K_{fd} \in \mathcal{R}^3$ diagonal positive definite proportional and derivative gains and $\mathbf{e}_r$, $\mathbf{e}_{\dot{r}}$ the position and velocity errors. The resulting control force in the body frame is $\mathbf{f} = [0\ 0\ -T]^T$. The required attitude $\mathbf{R}_{di}$ is obtained from the desired angles ϕ_d, θ_d, ψ_d. While ϕ_d and θ_d are obtained by geometric inspection from the inertial control forces f_{ix} and f_{iy}, ψ_d remains left to be controlled. The attitude is then controlled by

$$\mathbf{m} = \mathbf{I}(K_{mp}\,\mathbf{e}_\mathbf{R} - K_{md}\,\boldsymbol{\omega}) + (\mathbf{I}\boldsymbol{\omega}) \times \boldsymbol{\omega} - \mathbf{m}_e \tag{8}$$

with $K_{mp}, K_{md} \in \mathcal{R}^3$ diagonal positive definite proportional and derivative gains and $\mathbf{e}_\mathbf{R} = \mathbf{R}_{di}^T\mathbf{R}_{bi}$ the attitude error. The estimated external wrench $\boldsymbol{\tau}_e = [\mathbf{f}_e\ \mathbf{m}_e]^T$ is included in the controller in order to compensate for external disturbances. The quality of the control depends in fact on the external wrench estimation accuracy. A control scheme can be visualized in Fig. 2.

4. External Wrench Estimation

Knowledge of the system model can be used to estimate the external wrench and include it in the control architecture for better smooth external forces and torques which may be produced by continuous or impulsive stimulus, such as loads hung on the quadrotor, wind, or a human hand interaction. The following estimation methods [17] are both derived according to the Lagrangian form of the equations of motion.

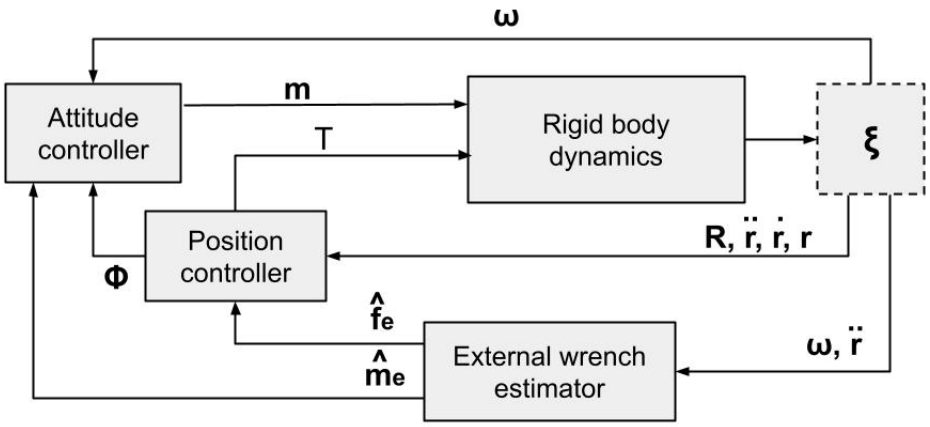

Figure 2: controller scheme of the UAV. The dashed state block stands for the assumption of a perfect state estimator, since it was not part of the aim of the project. The state is read from the Coppelia simulator.

4.1. Momentum-based Estimation

The generalized momentum according to (5) satisfies the following equation

$$\dot{\mathbf{p}} = \mathbf{M}\dot{\boldsymbol{\nu}} = \mathbf{J}^T\boldsymbol{\tau} + \boldsymbol{\tau}_e - \mathbf{N} \tag{9}$$

where $\mathbf{N} = \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}$. According to the research on collision detection and identification [18], we can write the residual vector

$$\boldsymbol{\rho} = K_I \left[\mathbf{p} - \int \left(\mathbf{J}\boldsymbol{\tau} + \boldsymbol{\tau}_e - \mathbf{N} + \boldsymbol{\rho} \right) dt \right] \tag{10}$$

with positive diagonal $K_I \in R^6$. By differentiating (10) one obtains

$$\dot{\boldsymbol{\rho}} = K_I\boldsymbol{\tau}_e - K_I\boldsymbol{\rho} \tag{11}$$

which is a linear exponentially stable system driven by the external wrench $\boldsymbol{\tau}_e$. Therefore the estimated external wrench will be $\hat{\boldsymbol{\tau}}_e = \boldsymbol{\rho}$. This method requires the measure of the twist $\boldsymbol{\nu}$.

4.2. Acceleration-based Estimation

The external wrench can be obtained directly from acceleration information. In fact, rearranging (5) we calculate the external wrench as

$$\hat{\boldsymbol{\tau}}_e = \mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g} - \boldsymbol{\tau} \tag{12}$$

This method is suitable for flying robots because the acceleration can be measured with an inertial measurement unit (IMU) sensor.

4.3. Hybrid Estimation

Estimating the external wrench with one of two methods, leads to poor quality of the result. In fact, in the momentum-based estimation, a drift-free translational velocity $\dot{\mathbf{r}}$ requires more sensors and a fusion algorithm, and this limits the applicability of the system. Moreover, in the acceleration-based estimation, the angular acceleration $\dot{\boldsymbol{\omega}}$ can be obtained only with numerical differentiation, reducing the quality of the torque estimation. Therefore an optimal choice for measuring the external wrench is obtained combining the two methods. This leads to the following general formulation

$$\hat{\boldsymbol{\tau}}_e = \begin{bmatrix} \hat{\mathbf{f}} \\ \hat{\mathbf{m}}_e \end{bmatrix} = \begin{cases} \int K_f \left(\mathcal{M}\mathbf{a} - \mathbf{f} - \hat{\mathbf{f}}_e \right) dt \\ K_m \left(\mathbf{I}\boldsymbol{\omega} + \int \left((\mathbf{I}\boldsymbol{\omega}) \times \boldsymbol{\omega} - \mathbf{m} - \hat{\mathbf{m}}_e \right) dt \right) \end{cases} \tag{13}$$

where we used the acceleration-based formulation for estimating the external force and the momentum-based estimation for the external torque, and K_f and K_m are integral filter gains $\in R^3$, $a = \mathbf{R}_b^{iT}(\dot{\mathbf{r}} - g\mathbf{e}_3)$ is the acceleration measured by an accelerometer in the center of mass and expressed in the body frame. The accelerometer also measures the acceleration of gravity. For the control input we assume hover conditions for better estimating the external wrench.

4.4. Discrimination

Associated to the primary task of the project, the external wrench estimation, was the discrimination between the wrenches, which, assuming no faults during the experiments, can be divided into aerodynamic and interaction wrenches, such that $\boldsymbol{\tau}_e = \boldsymbol{\tau}_d + \boldsymbol{\tau}_i$.

4.4.1. Data-driven approach

Following [1] we started with defining the aerodynamic force as a function of the airspeed $\mathbf{v}_r$, which is the speed of an aircraft relative to the air. Once the aerodynamic force is found, one can also determine the aerodynamic torque $\mathbf{m}_d = \mathbf{D}\mathbf{f}_d$ as a function of it. With these quantities is then easy to discriminate between aerodynamic and interaction torque as $\boldsymbol{\tau}_i = \boldsymbol{\tau}_e - \boldsymbol{\tau}_d$. According to the blade flapping model [19] the aerodynamic force can be modeled as

$$\mathbf{f}_d = \mathbf{A}_d \mathbf{v}_r c_T \sum_i \varpi_i \tag{14}$$

where $\mathbf{A}_d = diag(cd_x, cd_y, 0)$ with cd_x and cd_y the induced drag coefficients, c_T is the thrust factor, and ϖ_i is the angular velocity of the ith propeller. Given the relationship of the thrust magnitude $T = c_T \sum_i \varpi_i$, the aerodynamic force can be simply written as $\mathbf{f}_d = \mathbf{A}_d \mathbf{v}_r \, T$. While the thrust T is known, and the induced drag matrix $\mathbf{A}_d$ can be derived experimentally, the problem was finding the true airspeed $\mathbf{v}_r$. According to [1] one should find a model of the airspeed $\mathbf{v}_r(\mathbf{u})$, with some inputs $\mathbf{u}$ and train, over the wind tunnel dataset [13], a perceptron model

$$\mathbf{y}(\mathbf{u}) = \mathbf{W}_2((\mathbf{W}_1\mathbf{u} + \mathbf{b}_1) + \mathbf{b}_2) \tag{15}$$

where $\mathbf{W}_1 \in R^{N_h \times N_u}$, $\mathbf{W}_2 \in R^{N_y N_h}$ are the neuron weight matrices, $\mathbf{b}_1 \in R^{N_h}$ and $\mathbf{b}_2 \in R^{N_y}$ are the bias vectors, N_y, N_h, N_u are dimensions of non-zero parameters, and $\sigma(\cdot)$ is the activation function.

However, based on our knowledge, the simulator does not allow neither to perform the wind tunnel experiment nor setting and applying a certain wind velocity (and so a force) into the environment. To overcome this issue, we proceeded with empirical test using directly a linear model $\mathbf{v}_r = \alpha \mathbf{u}$ with $\alpha \in [0,1]$ a constant scale value. The inputs $\mathbf{u}$ we have tested are the quantities $\mathbf{f}_e$ and $\frac{\mathbf{f}_e}{\sum_i \varpi_i}$, because we know from the blade flapping model that there is a close relationship with the external force. Unfortunately without fitting a model with the perceptron, the results on the estimate of the aerodynamic force were poor and not satisfying at all the requirements. Therefore we decided to stop our investigation on the discrimination due to these limits related to the simulation environment and the lack of available datasets.

4.4.2. Physical modeling approach

(Tomic et al.)[1] presented a novel method to obtain wind velocity from aerodynamic power measurements, based on momentum theory. Starting from the aerodynamics equations of propellers, one can provide a system of equations independent from external forces. The forces exerted by a propeller depend on its *freestream* velocity (relative wind velocity). The freestream velocity of the k-th propeller as expressed in the propeller frame is

$$v_{\infty,k} = \mathbf{R}_{bp,k}^T (\mathbf{R}_{bi}^T \mathbf{v}_r + \boldsymbol{\omega} \times \mathbf{r}_k) \tag{16}$$

where $\mathbf{R}_{bp,k}$ is the rotation matrix from the body to the propeller frame, $\mathbf{v}_e$ is the airspeed, $\mathbf{r}_k$ is the location of the propeller relative to the center of gravity.
According to momentum theory (Leishman) [20], the thrust acting in positive z direction can be written as $T = 2\rho A v_i U$ where A is the propeller disk area, ρ is the air density, $U = ||v_i \mathbf{e}_3 + v_\infty||$ is the velocity of the propeller slipstream, and v_i is the induced velocity that can be obtained as

$$v_i = \frac{v_h^2}{\sqrt{v_{xy}^2 + (v_i - v_z)^2}} \tag{17}$$

with the horizontal and vertical components of the freestream velocity $\mathbf{v}_{xy} = \mathbf{v}_\infty - \mathbf{v}_z$ and $\mathbf{v}_z = \mathbf{e}_3^T \mathbf{v}_\infty$, and the hover induced velocity $v_h = T/\sqrt{2\rho A}$. Therefore the ideal aerodynamic power of a propeller is

$$P_a = 2\rho A v_i U (v_i - v_z) \tag{18}$$

and the aerodynamic power in forward flight is related to the hover power as

$$\frac{P_a}{P_h} = \frac{v_i - v_z}{v_h} \tag{19}$$

with $P_h = 2\rho A v_h^3$. Rearranging the above equations (17)(18)(19) as a system of nonlinear equations $\mathbf{F}(v_i, v_z, v_{xy}, v_h, P_a) = \mathbf{0}$ we get:

$$F_1 = v_i^4 - 2v_i^3 v_z + v_i^2 (v_z^2 + v_{xy}^2) - v_h^4 = 0 \tag{20}$$

$$F_2 = v_i U(v_i - v_z) - P_a/(2\rho A) = 0 \tag{21}$$
$$F_3 = v_h^2(v_i - v_z) - P_a/(2\rho A) = 0. \tag{22}$$

We consider $P_a/2\rho A$ and v_h to be known inputs and the aim is to determine the velocity component $\mathbf{x} = (v_x, v_y, v_z, v_i)^T$. This system is undetermined, as we have two knowns and three unknowns. By doing a transformation in a common frame we can estimate all three wind velocity $\mathbf{v}_w = (v_x, v_y, v_z)^T$ solving a nonlinear least square problem.

We further investigated on these mathematical relationships, in order to obtain the wind velocity, but again we have been stopped by the issues already discussed in the previous section. First, the lack of knowledge about the quadrotor physical model, prevented us to find a plausible value of the propeller disk area A, rotation matrix $\mathbf{R}_{bp,k}$, and translation vector of the propeller relative to the center of gravity $\mathbf{r}_k$. Moreover, we dealt again with the airspeed $\mathbf{v}_r$ problem.

5. Contact Detection

Detecting contact force under wind influence could not be a trivial task since different aerodynamics forces would affect the quadrotor generating different aerodynamics torque. Given the aerodynamics force model $\tilde{\mathbf{m}}_d$, the external torque will not match the model in case of another torque-generating force is acting on the robot. A residual between aerodynamics torque and external torque could be used for contact detection, this residual is hence defined :

$$\tilde{\mathbf{m}}_d = \mathbf{m}_d(\mathbf{f}_e) - \mathbf{m}_e \tag{23}$$

if is acting only the aerodynamic force and no other external forces are acting on the quadrotor, the external torque match the aerodynamic torque model.

$$\mathbf{m}_d(\mathbf{f}_e) = \mathbf{m}_e \tag{24}$$

Hence the residual is close to zero, depending on the estimation accuracy.

$$\tilde{\mathbf{m}}_d = \mathbf{m}_d(\mathbf{f}_e) - \mathbf{m}_e = \mathbf{0} \tag{25}$$

A threshold CD must be used, both external torque $\mathbf{m}_e$ and $\mathbf{m}_d$ are estimated, hence the residual must be compared with the threshold, CD change the sensibility of the entire contact detection algorithm, and must be tuned accordingly with the intensity of aerodynamic and contact forces, there is a contact if:

$$||\tilde{\mathbf{m}}_d|| > CD_{threshold} \tag{26}$$

As a consequence of the missing discrimination between external and aerodynamics forces, the aerodynamic wrench $\mathbf{m}_d(\mathbf{f}_e)$ cannot be estimated, in order to mimic a contact detection in hovering condition and without a payload, the residuals $\mathbf{r}$ are computed by using the *External Wrench Estimator* and the current control input $\mathbf{u}$. This approach is resulting effective only in the case that the previous assumptions are completely fulfilled, tuning the interval sampling and the threshold leads the model to correctly identify contact detection.

6. Experiments

6.1. Simulation Environment

We used Matlab framework to develop the logic of the project, i.e., the dynamic model of the system, the tracking controller and the external wrench estimator. To test and validate the experimental results we defined a set of structured simulations in CoppeliaSim environment, in which we analyzed the behavior of the system and the ability of the designed controller to correctly achieve desired task trajectories and estimation of external disturbances. After a brief introduction to the development tools used for the implementation, it is provided a detailed description of the environmental settings and of the experimental validation results.

6.1.1. Matlab

Matlab is a computational framework that combines a desktop environment optimized for iterative analysis and design processes with a programming language that expresses mathematical operations with matrices and arrays directly. It is one of the most used tool and framework for robotics application, in particular for educational and research purposes.

6.1.2. CoppeliaSim

CoppeliaSim, formerly known as V-REP, is a robot simulator used in industry, education and research. It provides with a kinematics engine for forward and inverse kinematics calculations, and several physics simulation libraries. Models and scenes for different kinds of experimental simulations can be built by assembling various objects into a hierarchical structure.

6.2. Implementation

The project is articulated into two main modules, a Matlab client application, implementing the core logic of the system and a Coppelia simulation *scene*, modeling the experimental environment and the simulation settings.

The Matlab application is made up of three main classes: Quadrotor, QuadrotorSim and Aerodynamics.

- The class *Quadcopter* implements the dynamic model and properties of the system and provides the control laws for trajectory tracking of the desired task trajectory.
- The class *Aerodynamics* contains the wrench estimation algorithm and the contact detection procedure.
- Finally, the class *QuadcopterSim* acts as an interface between Matlab logic and Coppelia simulation, managing the data communication between the two main modules and the project.

6.3. Problem Settings

To evaluate the performances of the designed controller and of the external wrench estimator in correctly detecting and compensating the wind and contact actions to achieve trajectory tracking task, we set up different experimental scenarios; two task trajectories are considered: a hovering and an elicoidal trajectory. For each of the task trajectory we tested the ability of the designed controller to balance the external wrenches due to wind action or contact forces.For the hovering task trajectory, we tested the external wrench estimation and contact detection, in the following cases:

- in presence of a impulsive contact action but in absence of wind action;
- in presence of a continuous contact action but in absence of wind action;
- in presence of a continuous contact action and of wind action.

The experimental pipeline can be summarized as follows:

1. The simulation environment is initialized by launching the CoppeliaSim scene.
2. The Matlab core script *simulation* is then launched. This script deals with establishing the connection with the simulation environment and to start the selected experimental case among the ones defined above.
3. The QuadcopterSim script provides with an interface between the robot simulation modeling in the virtual environment and its system dynamics defined in Matlab framework. After the connection is done, it retrieves from the simulation the parameters for the initialization of the dynamic model of the system.
4. The initial state for all the experimental cases is set with position at the origin of the world reference frame and zero orientation along all the three axes of rotation.
5. All the experiments last for a time interval of 20 seconds sampled by a time step of 0.01 seconds.
6. When both the simulation and the core logic are set, the main control loop is started, running the following steps:
    a) first, the next desired position and orientation in the considered task trajectory is computed;
    b) then, the control command to achieve the desired pose is obtained, using a PD control scheme;
    c) finally the desired motion is executed by sending the control command to the simulation environment, along with the estimation of the detection of contact forces and the estimation of the total external wrenches.

6.4. Results

In this section we provide details about the modeling and environmental settings for each experimental scenario, as described in Problem Settings paragraph.

Experiment 1 - Hovering with wind action only (aerodynamic external wrench)

- ◇ An aerodynamic wrench due to wind is simulated by generating a linear force of $5N$ and a nearly zero angular momentum is applied for 5 seconds from the initial time instant t_0.
- ◇ Due to the effect of the wind action, the quadrotor looses the desired position along the vertical path and also changes its orientation around the pitch angle.
- ◇ Soon after the application of the external wrench, the residual value reaches and overcomes the threshold value of 0.2 thus meaning that a contact force has been detected.
- ◇ The external wrench estimator provides thus with the reference of the disturbance to be compensated in order to recover the desired configuration.
- ◇ At time $t_f = 20$ seconds, the quadrotor has successfully completed the trajectory tracking task, with a negligible position error with respect to the target configuration.

Experiment 2 - Hovering with continuous external contact force

- ◇ At time $t = 10$ seconds a randomly generated contact wrench acts on the right side of the load carried by the quadrotor.
- ◇ After an initial force of about $4N$, the contact force intensities oscillate around $0.5N$ value, to then reach again a $4N$ intensities at time $t = 20$ seconds, before vanishing.
- ◇ From the residual graph we can see how the contact force is immediately detected few instants after it is applied to the load, as shown by the residual value overcoming the positive threshold value 0.2.
- ◇ From the control graph we can see the effect of the external disturbances, in particular a change in the orientation around the yaw angle. After a side maneuver to recover the initial acceleration, the quadrotor reach the steady state in around 10 seconds, thus achieving the desired target configuration for the hovering task.

Experiment 3 - Hovering with both wind and contact force actions

- ◇ In this experiment a double source of external disturbances is considered: (i) a aerodynamic wrench due to wind, acting on the quadrotor and (ii) a contact force with point of application on the left side of the quadrotor. Both external disturbances are randomly generated in direction and intensity.
- ◇ As in the previous experiment, the robot can detect the contact force and recover the desired position for the hovering task. However, this time a longer side maneuver is required: the robot moves left and right as in an oscillating behavior before reaching the steady state.
- ◇ From the final plots we can also notice the correct estimation of the total external wrench resulting in a peak of $6N$ of the external force value from time $t = 10$ to time $t = 20$ seconds.

Experiment 4 - Spiral trajectory with wind action only

In this scenario the robot has to track an elliptical trajectory to reach 4 meters height.

- ◇ At time $t = 10$ seconds, a randomly generated aerodynamic perturbation, with intensity below $1N$, begins to affect the load of the quadrotor, disturbing the tracking of the elicoidal trajectory.

- ◇ The quadrotor looses the reference orientation around the roll and pitch angle and also the position of the reference point of the robot moves away from the reference position along the desired path.
- ◇ Also in this more complex situation, the designed controller is able to correct the position and orientation errors compensating for the external disturbances and allowing the quadrotor to recover the desired state along the trajectory thanks to the correct estimation of the external wrench due to the wind action.

Experiment 5 (extra) - Hovering with wind action and not uniformly distributed load mass

We defined an extra experimental case in which the quadrotor performs the hovering task by carrying a huge load with very asymmetric and not uniformly distributed mass, under the action of external aerodynamics wrenches.

- ◇ At the initial time instant $t = 0$, a randomly generated aerodynamics wrench with linear component $f_e = 15N$ and angular momentum $m_e = 0.5N$ acts on the back side of the quadrotor.
- ◇ Under the wind action, the drone is pushed away from the reference trajectory and also performs a positive rotation around the pitch angle.
- ◇ As soon as the external wrench hits the quadrotor, the detector module observes it by registering a value for the x component of the residual wrench below the given negative treshold of -0.2.
- ◇ The wind action lasts for 6 seconds and as many are required for the complete recovering of the desired position and orientation along the reference trajectory.
  We can notice in the final time instants of the simulation that, after the wind action is ended, the wrench estimator module still registers an angular momentum of $1.5N$ around the pitch angle. We concluded it may represents the torque due to the weight of the asymmetric and not uniformly distributed load, that is perceived as an external wrench and in fact compensated by the controller, as shown by the negative torque produced by the quadrotor around the pitch angle.

7. Conclusions

In this project, we provided the implementation and evaluation of a simultaneous contact and aerodynamic wrench estimation system for aerial robots without the need for additional on-board sensors. We achieved this by modeling the relevant aerodynamics models and reasoning about wind velocity. In particular, the contact detection procedure is based on the computation of the residual between the external torque and the expected aerodynamic torque as a function of the external force.

7.1. Weaknesses

Unfortunately, as described in the Discrimination paragraph the provided system for external wrench estimation is not complete since the module aimed at discriminating between aerody-

namics and contact actions is missing. This is in particular due to the limits set by the simulation environment that prevented us to correctly perform the required empiric evaluation of the airspeed needed to fit a suitable model for the aerodynamic forces according to the *blade flapping model*.

7.2. Improvements

Possible improvements could be achieved by including the discrimination of the external wrenches, collecting the required data for the estimation of the aerodynamics model by performing empiric studies in a controlled real-world environment or by switching to different simulation tools. Moreover, a particle filter can be used, as proposed in [1], to directly estimate the contact force position on the robot convex hull. This would result in a more 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.

Acknowledgments

Thanks to the Professor Marina Vendittelli, for delivering the course on Elective in Robotics - Modeling and Control of Mulit-rotor UAVs, for being always available for supporting us during the whole development process of the project.

References

[1] T. Tomić, P. Lutz, K. Schmid, A. Mathers, S. Haddadin, Simultaneous contact and aerodynamic force estimation (s-cafe) for aerial robots, 2018. arXiv:1810.12908.

[2] M. Philippe, S. Erwan, The true role of accelerometer feedback in quadrotor control, 2010. doi:10.1109/ROBOT.2010.5509980.

[3] S. Waslander, C. Wang, Wind disturbance estimation and rejection for quadrotor position control., 2019. doi:10.2514/6.2009-1983.

[4] D. W. C. Sikkel L, de Croon G, C. Q, A novel online model-based wind estimation approach for quadrotor micro air vehicles using low cost mems imus., 2016. doi:10.1109/IROS.2016.7759336.

[5] D. G. Omari S, Hua MD, H. T, Nonlinear control of vtol uavs incorporating flapping dynamics., 2013. doi:10.1109/IROS.2013.6696696.

[6] W. S. Huang H, Hoffmann GM, T. CJ, Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering., 2009. doi:10.1109/ROBOT.2009.5152561.

[7] T. T, Haddadin, Simultaneous estimation of aerodynamic and contact forces in flying robots: Applications to metric wind estimation and collision detection., 2015. doi:10.1109/ICRA.2015.7139937.

[8] N. PP, B. M, Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit., 2015. arXiv:Sensors and Actuators A: Physical 235: 300–310.

[9] J. Ware, N. Roy, An analysis of wind field estimation and exploitation for quadrotor flight in the urban canopy layer., 2016. doi:10.1109/ICRA.2016.7487287. arXiv:2016 IEEE International Conference on Robotics and Automation (ICRA).

[10] S. F. et.al., Towards estimation and correction of wind effects on a quadrotor uav., 2014. doi:10.3929/ethz-a-010286793. arXiv:International Micro Aerial Vehicle Conference (IMAV).

[11] P. L, C. P, Quadrotor uav aerodynamic model identification using indoor flight experiment and feasibility of uav as wind gust sensor., 2015. arXiv:International Micro Aerial Vehicle Conference (IMAV).

[12] R. C. Jung J, Glasner B, W. G, Quadrotor uav aerodynamic model identification using indoor flight experiment and feasibility of uav as wind gust sensor., 2016. arXiv:Proceedings of the AHS 72nd Annual Forum. West Palm Beach, FL, USA.

[13] L. Sikkel, G. Croon, C. De Wagter, Q. Chu, A novel online model-based wind estimation approach for quadrotor micro air vehicles using low cost mems imus, 2016, pp. 2141–2146. doi:10.1109/IROS.2016.7759336.

[14] R. Mahony, V. Kumar, P. Corke, Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor, IEEE Robotics Automation Magazine 19 (2012) 20–32. doi:10.1109/MRA.2012.2206474.

[15] T. Luukkonen, Modelling and control of quadcopter, Independent research project in applied mathematics, Espoo 22 (2011) 22.

[16] M.-D. Hua, T. Hamel, P. Morin, C. Samson, Introduction to feedback control of underactuated vtolvehicles: A review of basic control design ideas and principles, IEEE Control Systems Magazine 33 (2013) 61–75. doi:10.1109/MCS.2012.2225931.

[17] T. Tomić, S. Haddadin, A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 4197–4204. doi:10.1109/IROS.2014.6943154.

[18] A. De Luca, A. Albu-Schaffer, S. Haddadin, G. Hirzinger, Collision detection and safe reaction with the dlr-iii lightweight manipulator arm, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 1623–1630. doi:10.1109/IROS.2006.282053.

[19] S. Omari, M.-D. Hua, G. Ducard, T. Hamel, Nonlinear control of vtol uavs incorporating flapping dynamics, 2013. doi:10.1109/IROS.2013.6696696.

[20] G. J. Leishman, Principles of helicopter aerodynamics with CD extra, Cambridge university press, 2006.