Estimation, Navigation and Control of Multi-Rotor Drones in an Urban Wind Field

Vahram Stepanyan [1]
University of California Santa Cruz, Santa Cruz, CA 95064

Kalmanje Krishnakumar [2]
NASA Ames Research Center, Moffett Field, CA 94035

The paper presents an on-board estimation, navigation and control architecture for multi-rotor drones flying in urban environment. It consists of adaptive algorithms to estimate vehicle's aerodynamic drag coefficients with respect to still air and the urban wind components along the flight trajectory, with guaranteed fast and reliable convergence to the true values; navigation algorithms to generate feasible trajectories between given way-points that take into account the estimated wind; and of control algorithms to track the generated trajectories as long as the vehicle retains sufficient number of functioning rotors capable of compensating for the estimated wind. All components of this on-board system are computationally effective and are intended for a real time implementation. The algorithms were tested in simulations.

I. Introduction

Drones are becoming increasingly popular for research, commercial and military applications due to their affordability resulting from their small size, low cost and simple hardware structure. One of the critical aspects of these uses is the reliability of the drones while maintaining their

[1] Senior Research Scientist, Universities Space Research Association, NASA Ames Research Center/Mail Stop 269/1, AIAA Senior Member, email: vahram.stepanyan@nasa.gov
[2] Autonomous Systems and Robotics Branch Chief, Intelligent Systems Division, NASA Ames Research Center/Mail Stop 269/1, AIAA Associate Fellow, email: kalmanje.krishnakumar@nasa.gov

simplicity in design. However, this type of design with the light weight structure and limited power makes them vulnerable to wind disturbances, hence difficult for accurate navigation and control in outdoors, especially in urban environment where the wind field is more complex and has more uncertainties.

One way of enabling reliable UAV operations in urban environment is to design a controller capable of compensating for atmospheric disturbances, see for example [15], [1], [17], [12], [22], [25], [7] and references therein. In [15], the wind effects are estimated by a nonlinear disturbance observer and used to design a path following controller. In [1], a controller is presented to achieve trajectory tracking for kinematic models of unmanned aerial vehicles. In [17], a linear observer with integral action is used to stabilize a qudrotor at hover flight taking into account only the wind effects in roll and pitch angles. In [12], L1 adaptive control augmentation of the baseline outer-loop controller is used for position tracking in the presence of wind disturbances. In [22], path-following guidance method is presented in the presence of quasi-constant but unknown wind disturbances. A quaternion-based adaptive attitude control for a quadrotor in the presence of external disturbances is considered in [25].

Handling wind disturbances with control design is easy to implement, but may generate problems when the wind in the direction of predefined path/trajectory is strong enough to cause actuator saturations or rapid discharge of batteries. In this case, the mission cannot be accomplished. However, it may be possible to redefine the trajectories incorporating the available or estimated wind information and make the mission possible. That is the trajectory or path generation phase of the UAS operations in urban environment should include the wind wield. However, the majority of the existing trajectory generation algorithms do not take into account the atmospheric drag and wind effects (see for example [19] and references therein). These algorithms can be related to two main groups. First group incorporates only spacial variables in the design. That is, first a geometric path is planned in space from a class of primitives such as lines, polynomials or splines, then the path is parameterized in time in order to enforce the vehicle's dynamic constraints (see for example [4, 6, 11]). A literature survey on path planning can be found in [13]. The second group simultaneously incorporates spatial and temporal variables and designs trajectories based

on solutions of some optimal control problems. Examples of such methods include minimum snap trajectory generation [18], the minimization of a weighted sum of derivatives [21], a learning-based model predictive control method for linear dynamics [3], a time-optimal control methods based on Pontryagin's minimum principle [10] or numerical optimization [16]. To simplify the optimization routines, the differential flatness of quadrotor dynamics can be exploited (see for example [19] and references therein), which results in decoupled trajectory generations for the translational axes and yaw angle, the latter is usually fixed at zero in the literature. This method can be used to generate trajectories by minimizing the snap (forth order derivative of the position) [18], the time of flight [9], [2] or the jerk (third order derivative of the position) [19].

Trajectory generation problems in the wind field have received less attention. In [8], a time optimal trajectory generation method in known constant in time and linear in space wind field is presented for the kinematic model of quadrotors. In [28] and [24], minimum time algorithm and trochoids are respectively used for path planning in known steady uniform wind fields for fixed wing UAVs. These approaches are not applicable in urban environment since the wind field may not be uniform or known. For this reason wind estimation techniques have to be employed to accommodate for the trajectory generation.

One way to estimate the wind components is using air data measurements from available onboard sensors (see for example [5], [14] and references therein). While this approach may be suitable for fixed wing UAVs, no reliable air data sensors have been reported for the multi-rotor UAVs in the literature to our best knowledge. Therefore there is a need for wind estimation methods using only inertial data.

This paper presents a unified approach to autonomous flights of multi-rotor vehicles in urban environment without prior knowledge of the wind field, which is variable in time and in space. It is assumed that the inertial position and velocity of the vehicle's center of mass, orientation angles around the center of mass and the angular rates are available for feedback through on-board sensor package. The approach includes a set of real-time algorithms to accurately estimate the atmospheric drag forces and moments and the wind linear and angular velocities and accelerations, generate minimum time trajectories trough the given set of way points, which take into account

the estimated drag and wind, and a control design that produces proper input signals to individual motors to reliably track the generated trajectories in the presence of the atmospheric disturbances. The benefits of the approach have been demonstrated through the simulation for an octocopter flying in the cityscape with simulated wind field.

II. Drone's Dynamic Model

A. Equations of Motion

The dynamics of the multi-rotor vehicle's center of mass in the East-North-Up Earth (inertial) frame (F_E) are given by

$$\begin{aligned} \dot{\boldsymbol{r}}(t) &= \boldsymbol{v}(t) \\ m\dot{\boldsymbol{v}}(t) &= R_{B/E}(t)\boldsymbol{e}_3^B f_T(t) + \boldsymbol{f}_D(t) + m\boldsymbol{g}\,, \end{aligned} \tag{1}$$

where $\boldsymbol{r}(t) = [x(t)\ y(t)\ z(t)]^\top$ is the position of the center of mass in F_E, $\boldsymbol{v}(t) = [v_x(t)\ v_y(t)\ v_z(t)]^\top$ is the inertial velocity, m is the mass, $f_T(t)$ is the total thrust generated by the rotors, $R_{B/E}(t)$ is the rotation matrix from the body frame F_B (Forward-Left-Up) to F_E, $\boldsymbol{e}_3^B = [0\ 0\ 1]^\top$ is the third unit vector of F_B, $\boldsymbol{f}_D(t)$ is the aerodynamic drag force and $\boldsymbol{g} = [0\ 0\ -g]^\top$ is the gravity acceleration.

The vehicle's rotational dynamics about the center of mass are given in the frame F_B as

$$\begin{aligned} \dot{R}_{B/E}(t) &= R_{B/E}(t)\boldsymbol{\omega}^\times(t) \\ J\dot{\boldsymbol{\omega}}(t) &= -\boldsymbol{\omega}(t) \times J\boldsymbol{\omega}(t) + J_m\omega_m(t)\bar{\boldsymbol{\omega}}(t) + \boldsymbol{\tau}(t) + \boldsymbol{\tau}_D(t)\,, \end{aligned} \tag{2}$$

where $\boldsymbol{\omega}(t) = [p(t)\ q(t)\ r(t)]^\top$ is the angular rate of F_B with respect to the inertial frame F_E expressed in F_B, $J = diag(J_1,\ J_2,\ J_3)$ is the vehicle's inertia matrix (the body frame is aligned with the principal axes of inertia), J_m is the rotor inertia about the axis of rotation (assuming identical for all of them), $\bar{\boldsymbol{\omega}}(t) = [-q(t)\ p(t)\ 0]^\top$, $\omega_m(t) = \sum_{i=1}^n (-1)^i \Omega_i(t)$, $\Omega_i(t)$ is the i-th rotor angular rate about its axis of rotation, $\boldsymbol{\tau}(t)$ is the torque generated by the rotors, $\boldsymbol{\tau}_D(t)$ is the aerodynamic rotational drag torque.

It is assumed that all motors generate thrust in the positive z-direction in F_B frame ($\boldsymbol{e}_3^B$), and $f_T(t) = \sum_{i=1}^n f_i(t)$, where $f_i(t)$ is the thrust generated by the i-th rotor at time t.

B. Aerodynamic Drag

In this paper, we adopt commonly used quadratic model $D = -\frac{1}{2}\rho v_a^2 S C_D$ for the translational drag (or drag force), where ρ is the air density, v_a is the speed of the body relative to air, S is the cross sectional area, and C_D is the drag coefficient. We can reasonably assume that the air density is constant at the altitude corresponding to the specific urban environment, however S and C_D depend on the body configuration and orientation with respect to air speed. In other words, S and C_D are constants in the body frame. Therefore, the drag force can be modeled in the body frame as $\boldsymbol{f}_D^B = [-v_{a_x}^B|v_{a_x}^B|c_{D_x} \quad -v_{a_y}^B|v_{a_y}^B|c_{D_y} \quad -v_{a_z}^B|v_{a_z}^B|c_{D_z}]^\top$, where superscript B indicates body frame quantities, $v_{a_x}^B,\ v_{a_y}^B,\ v_{a_z}^B$ are the components of the body relative to the air velocity in the body frame, and the coefficients $c_{D_i} = \frac{1}{2}\rho S_i C_{D_i}$ are constant for each axis $i = x, y, z$. The drag force $\boldsymbol{f}_D^B$ needs to be translated to the inertial frame in order to apply to the translational dynamics (1). That is $\boldsymbol{f}_D = R_{B/E}\boldsymbol{f}_D^B$, which in the vector-matrix form can be written as

$$\boldsymbol{f}_D = R_{B/E}\begin{bmatrix} -v_{a_x}^B|v_{a_x}^B|c_{D_x} \\ -v_{a_y}^B|v_{a_y}^B|c_{D_y} \\ -v_{a_z}^B|v_{a_z}^B|c_{D_z} \end{bmatrix} = -R_{B/E}\Phi(\boldsymbol{v}_a^B|\boldsymbol{v}_a^B|)\boldsymbol{c}_D = -R_{B/E}\Phi(\boldsymbol{c}_D)\boldsymbol{v}_a^B|\boldsymbol{v}_a^B|\,, \tag{3}$$

where we have introduced a notation $\Phi(\boldsymbol{b}) = \mathrm{diag}(b_x,\ b_y,\ b_z)$ for a vector $\boldsymbol{b}$.

We adopt a similar model for the rotational drag (or drag torque) using the body angular velocity $\boldsymbol{\omega}_a$ with respect to the air. In the body frame the rotational drag is expressed as $\boldsymbol{\tau}_D^B = [-\omega_{a_x}^B|\omega_{a_x}^B|c_{\tau_x} \quad -\omega_{a_y}^B|\omega_{a_y}^B|c_{\tau_y} \quad -\omega_{a_z}^B|\omega_{a_z}^B|c_{\tau_z}]^\top$, where $\omega_{a_x}^B,\ \omega_{a_y}^B,\ \omega_{a_z}^B$ are the components of the relative to the air angular velocity and $c_{\tau_x},\ c_{\tau_y},\ c_{\tau_z}$ are rotational drag coefficients, which are constant in the body frame. In this case, there is no need to express the rotational drag in the inertial frame. In what follows, we will use the following expressions for the translational and rotational drags

$$\boldsymbol{f}_D = -R_{B/E}\Phi(\boldsymbol{v}_a^B|\boldsymbol{v}_a^B|)\boldsymbol{c}_D = -R_{B/E}\Phi(\boldsymbol{c}_D)\boldsymbol{v}_a^B|\boldsymbol{v}_a^B| \tag{4}$$

$$\boldsymbol{\tau}_D = -\Phi(\boldsymbol{\omega}_a^B|\boldsymbol{\omega}_a^B|)\boldsymbol{c}_\tau = -\Phi(\boldsymbol{c}_\tau)\boldsymbol{\omega}_a^B|\boldsymbol{\omega}_a^B| \tag{5}$$

where $\boldsymbol{c}_D$ and $\boldsymbol{c}_\tau$ are constant vectors of translational and rotational drag coefficients respectively.

III. Adaptive Estimation

A. Drag Estimation

First, we estimate the translational drag coefficient $\boldsymbol{c}_D$, when the airspeed and the inertial velocity are equal, that is when the drone flies in the still air (or indoor). To this end, we use the vehicle's inertial velocity and orientation angle measurements, available from the sensor package. In addition we assume that the total thrust generated by the rotors is available from the rotor models and spin rate measurements. Representing the translational dynamics (1) in the form

$$m\dot{\boldsymbol{v}}(t) = f_T(t)R_{B/E}(t)\boldsymbol{e}_3^B + m\boldsymbol{g} - R_{B/E}(t)\Phi(\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|)\boldsymbol{c}_D \tag{6}$$

and following the steps from [26], we design a prediction model

$$m\dot{\hat{\boldsymbol{v}}}(t) = f_T(t)R_{B/E}(t)\boldsymbol{e}_3^B + m\boldsymbol{g} - R_{B/E}(t)\Phi(\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|)\hat{\boldsymbol{c}}_D(t) + \lambda_l\tilde{\boldsymbol{v}}(t)\,, \tag{7}$$

where $\hat{\boldsymbol{v}}(t)$ is the velocity prediction, $\hat{\boldsymbol{c}}_D(t)$ is the translational drag coefficient's estimate, λ_l is the error feedback gain and $\tilde{\boldsymbol{v}}(t) = \boldsymbol{v}(t) - \hat{\boldsymbol{v}}(t)$ is the prediction error. The adaptive law for $\hat{\boldsymbol{c}}_D(t)$ is derived from the Lyapunov stability analysis for the prediction error dynamics

$$m\dot{\tilde{\boldsymbol{v}}}(t) = -\lambda_l\tilde{\boldsymbol{v}}(t) - R_{B/E}(t)\Phi(\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|)\tilde{\boldsymbol{c}}_D(t)\,, \tag{8}$$

where $\tilde{\boldsymbol{c}}_D(t) = \boldsymbol{c}_D - \hat{\boldsymbol{c}}_D(t)$ is the estimation error. The Lyapunov function is chosen as

$$L(t) = \frac{m}{2}\tilde{\boldsymbol{v}}^\top(t)\tilde{\boldsymbol{v}}(t) + \frac{1}{2\gamma_l}\tilde{\boldsymbol{c}}_D^\top(t)\tilde{\boldsymbol{c}}_D(t)\,, \tag{9}$$

where $\gamma_l > 0$ is the adaptation rate. It is straightforward to see that

$$\dot{L}(t) = -\lambda_l\tilde{\boldsymbol{v}}^\top(t)\tilde{\boldsymbol{v}}(t) + \tilde{\boldsymbol{c}}_D^\top(t)\left[-\Phi(\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|)R_{E/B}(t)\tilde{\boldsymbol{v}}(t) + \frac{1}{\gamma_l}\dot{\tilde{\boldsymbol{c}}}_D(t)\right]\,, \tag{10}$$

Therefore, defining the adaptive law as

$$\dot{\hat{\boldsymbol{c}}}_D(t) = -\gamma_l\Phi(\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|)R_{E/B}(t)\tilde{\boldsymbol{v}}(t) \tag{11}$$

renders $\dot{L}(t)$ negative semidefinite, implying that $\tilde{\boldsymbol{v}}(t)$ and $\tilde{\boldsymbol{c}}_D(t)$ are globally bounded. In addition, application of Barbalat's lemma ([23], p.19) insures that $\tilde{\boldsymbol{v}}(t) \to 0$ as $t \to \infty$ when $\boldsymbol{v}(t)$ is bounded. Since (8) is an LTI system with the input $R_{B/E}(t)\Phi(\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|)\boldsymbol{c}_D(t)$, it follows that $R_{B/E}(t)\Phi(\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|)\tilde{\boldsymbol{c}}_D(t) \to 0$ as $t \to \infty$. Therefore, we conclude that $\tilde{\boldsymbol{c}}_D(t) \to 0$ as $t \to \infty$ if

$\Phi(\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|)$ is nonsingular, in other words, if the airspeed components $v_x^B(t),\ v_y^B(t),\ v_z^B(t)$ are nonzero. In fact, in this case the convergence is exponential (see for example [20]).

Next, we estimate the rotational drag using the prediction of the angular rate dynamics

$$J\dot{\hat{\boldsymbol{\omega}}}(t) = -\boldsymbol{\omega}(t) \times J\boldsymbol{\omega}(t) + J_m\omega_m(t)\bar{\boldsymbol{\omega}}(t) + \boldsymbol{\tau}(t) - \Phi(\boldsymbol{\omega}_a^B(t)|\boldsymbol{\omega}_a^B(t)|)\hat{\boldsymbol{c}}_\tau(t) + \lambda_r\tilde{\boldsymbol{\omega}}(t)\,, \tag{12}$$

where $\hat{\boldsymbol{\omega}}(t)$ is the prediction of the vehicle's angular rate, $\hat{\boldsymbol{c}}_\tau(t)$ is the estimation of the rotational drag coefficient, $\lambda_r > 0$ is the error feedback gain and $\tilde{\boldsymbol{\omega}}(t)$ is the prediction error. The adaptive law for the estimate $\hat{\boldsymbol{c}}_\tau(t)$ is given by

$$\dot{\hat{\boldsymbol{c}}}_\tau(t) = -\gamma_r\Phi(\boldsymbol{\omega}_a^B(t)|\boldsymbol{\omega}_a^B(t)|)\tilde{\boldsymbol{\omega}}(t)\,, \tag{13}$$

which results in the error system

$$\begin{aligned} J\dot{\tilde{\boldsymbol{\omega}}}(t) &= -\lambda_r\tilde{\boldsymbol{\omega}}(t) - \Phi(\boldsymbol{\omega}_a^B(t)|\boldsymbol{\omega}_a^B(t)|)\tilde{\boldsymbol{c}}_\tau(t) \\ \dot{\tilde{\boldsymbol{c}}}_\tau(t) &= \gamma_r\Phi(\boldsymbol{\omega}_a^B(t)|\boldsymbol{\omega}_a^B(t)|)\tilde{\boldsymbol{\omega}}(t)\,, \end{aligned} \tag{14}$$

As in the previous case, it can be shown that the error system (14) is globally stable, and $\tilde{\boldsymbol{\omega}}(t) \to 0$ as $t \to \infty$ when $\boldsymbol{\omega}(t)$ is bounded. In addition, if all of the components of $\boldsymbol{\omega}(t)$ are nonzero, $\tilde{\boldsymbol{c}}_\tau(t) \to 0$ exponentially as $t \to \infty$. It should be noted that in the case of rotational drag partial convergence is possible. That is, if not all components are nonzero, the rotational drag coefficients corresponding to nonzero components exponentially converge to the true values because the equations in (14) are decoupled.

B. Wind Estimation

Once we have estimated both translational and rotational drag coefficients, we can proceed with the estimation of the wind velocities and accelerations. To this end we write the dynamic equations as

$$\begin{aligned} \dot{\boldsymbol{v}}(t) &= \frac{f_T(t)}{m}R_{B/E}(t)\boldsymbol{e}_3^B + \boldsymbol{g} - \frac{1}{m}R_{B/E}(t)\Phi(\boldsymbol{c}_D)\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)| \\ \dot{\boldsymbol{\omega}}(t) &= -J^{-1}\boldsymbol{\omega}(t) \times J\boldsymbol{\omega}(t) + J_m\omega_m(t)J^{-1}\bar{\boldsymbol{\omega}}(t) + J^{-1}\boldsymbol{\tau}(t) - J^{-1}\Phi(\boldsymbol{c}_\tau)\boldsymbol{\omega}_a^B(t)|\boldsymbol{\omega}_a^B(t)|\,, \end{aligned} \tag{15}$$

where the airspeed $\boldsymbol{v}_a^B(t) = \boldsymbol{v}^B(t) - \boldsymbol{w}^B(t)$ includes also the wind velocity $\boldsymbol{w}^B(t)$ expressed in the body frame, and the relative angular rate $\boldsymbol{\omega}_a^B(t) = \boldsymbol{\omega}(t) - \boldsymbol{\omega}_c^B(t)$ includes the air mass circulation rate (or vorticity) $\boldsymbol{\omega}_c^B(t)$ expressed in the body frame. We treat the terms $-m^{-1}R_{B/E}(t)\Phi(\boldsymbol{c}_D)\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)|$ and $-J^{-1}\Phi(\boldsymbol{c}_\tau)\boldsymbol{\omega}_a^B(t)|\boldsymbol{\omega}_a^B(t)|$ as external disturbance signals $\boldsymbol{s}_v(t)$ and $\boldsymbol{s}_\omega(t)$ respectively, so the equations (15) take the form

$$\begin{aligned}
\dot{\boldsymbol{v}}(t) &= \frac{1}{m} f_T(t) R_{B/E}(t) \boldsymbol{e}_3^B + \boldsymbol{g} + \boldsymbol{s}_v(t) \\
\dot{\boldsymbol{\omega}}(t) &= -J^{-1}\boldsymbol{\omega}(t) \times J\boldsymbol{\omega}(t) + J_m \omega_m(t) J^{-1}\bar{\boldsymbol{\omega}}(t) + J^{-1}\boldsymbol{\tau}(t) + \boldsymbol{s}_\omega(t)\,. \qquad (16)
\end{aligned}$$

Similar to the drag estimation case, we introduce the prediction model and adaptive law for the translational dynamics as

$$\begin{aligned}
\dot{\hat{\boldsymbol{v}}}(t) &= \frac{1}{m} f_T(t) R_{B/E}(t) \boldsymbol{e}_3^B + \boldsymbol{g} + \hat{\boldsymbol{s}}_v(t) + \lambda_v \tilde{\boldsymbol{v}}(t) \\
\dot{\hat{\boldsymbol{s}}}_v(t) &= \gamma_v \tilde{\boldsymbol{v}}(t)\,, \qquad (17)
\end{aligned}$$

where $\lambda_v > 0$ and $\gamma_v > 0$ are design parameters, $\tilde{\boldsymbol{v}}(t) = \boldsymbol{v}(t) - \hat{\boldsymbol{v}}(t)$ is the inertial velocity prediction error, $\hat{\boldsymbol{s}}_v(t)$ is the disturbance estimate, and for the rotational dynamics as

$$\begin{aligned}
\dot{\hat{\boldsymbol{\omega}}}(t) &= -J^{-1}\boldsymbol{\omega}(t) \times J\boldsymbol{\omega}(t) + J_m \omega_m(t) J^{-1}\bar{\boldsymbol{\omega}}(t) + J^{-1}\boldsymbol{\tau}(t) + \hat{\boldsymbol{s}}_\omega(t) + \lambda_\omega \tilde{\boldsymbol{\omega}}(t) \\
\dot{\hat{\boldsymbol{s}}}_\omega(t) &= \gamma_\omega \tilde{\boldsymbol{\omega}}(t)\,. \qquad (18)
\end{aligned}$$

where $\lambda_\omega > 0$ and $\gamma_\omega > 0$ are design parameters, $\tilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) - \hat{\boldsymbol{\omega}}(t)$ is the angular rate prediction error, $\hat{\boldsymbol{s}}_\omega(t)$ is the disturbance estimate. Denoting the estimation errors as $\tilde{\boldsymbol{s}}_v(t) = \boldsymbol{s}_v(t) - \hat{\boldsymbol{s}}_v(t)$ and $\tilde{\boldsymbol{s}}_\omega(t) = \boldsymbol{s}_\omega(t) - \hat{\boldsymbol{s}}_\omega(t)$, we derive the error system

$$\begin{aligned}
\dot{\tilde{\boldsymbol{v}}}(t) &= -\lambda_v \tilde{\boldsymbol{v}}(t) + \tilde{\boldsymbol{s}}_v(t) \\
\dot{\tilde{\boldsymbol{s}}}_v(t) &= -\gamma_v \tilde{\boldsymbol{v}}(t) + \dot{\boldsymbol{s}}_v(t)\,, \qquad (19)
\end{aligned}$$

for the translational dynamics and

$$\begin{aligned}
\dot{\tilde{\boldsymbol{\omega}}}(t) &= -\lambda_\omega \tilde{\boldsymbol{\omega}}(t) + \tilde{\boldsymbol{s}}_\omega(t) \\
\dot{\tilde{\boldsymbol{s}}}_\omega(t) &= -\gamma_\omega \tilde{\boldsymbol{\omega}}(t) + \dot{\boldsymbol{s}}_\omega(t)\,. \qquad (20)
\end{aligned}$$

for the rotational dynamics.

Obviously, the error system (19) and (20) are stable LTI systems with inputs $\dot{\boldsymbol{s}}_v(t)$ and $\dot{\boldsymbol{s}}_\omega(t)$ respectively, therefore have bounded solutions if the signals $\dot{\boldsymbol{s}}_v(t)$ and $\dot{\boldsymbol{s}}_\omega(t)$ are essentially bounded, that is bounded everywhere except for on the sets of measure zero. Assuming that the flight control system provides a bounded and continuous inertial velocity and angular rate, the proposed method can produce valid estimates of any wind field, even if the wind components are abruptly changing at countably many time instances.

To derive the upper bounds on the components of the estimation errors, we notice that the equations (19) and (20) are decoupled. Therefore we introduce a generic system

$$\begin{aligned} \dot{x}_1(t) &= -\lambda x_1(t) + x_2(t) \\ \dot{x}_2(t) &= -\gamma x_1(t) + f(t)\,, \end{aligned} \tag{21}$$

where $x_1(t)$ represents any component of the linear or angular velocity prediction errors $\tilde{\boldsymbol{v}}(t)$ or $\tilde{\boldsymbol{\omega}}(t)$, $x_2(t)$ and $f(t)$ represent the corresponding components of $\tilde{\boldsymbol{s}}_v(t)$ or $\tilde{\boldsymbol{s}}_\omega(t)$ and $\dot{\boldsymbol{s}}_v(t)$ or $\dot{\boldsymbol{s}}_\omega(t)$ respectively.

First of all, we notice that if $f(t) \equiv 0$, then $x_1(t)$ and $x_2(t)$ exponentially converge to zero from all initial conditions, which means that translational and rotational drag estimates exponentially converge to true values on any interval where the corresponding linear or angular drag components are constant.

Next, ignoring the exponentially decaying effects of the initial errors, the rate of decay which is given by the design parameters k_v and k_ω, we represent the solution of (21) in the operator form as

$$x_1(s) = \frac{1}{s^2 + \lambda s + \gamma} f(s), \quad x_2(s) = -\frac{s + \lambda}{s^2 + \lambda s + \gamma} f(s) \tag{22}$$

Since

$$\left\| \frac{1}{s^2 + \lambda s + \gamma} \right\|_{H_\infty} = \frac{1}{\gamma}, \quad \left\| \frac{s + \lambda}{s^2 + \lambda s + \gamma} \right\|_{H_\infty} = \frac{\lambda}{\gamma}$$

we conclude that

$$|x_1(t)| \leq \frac{1}{\gamma} \operatorname{ess\,sup} |f_{[0,t]}|, \quad |x_1(t)| \leq \frac{\lambda}{\gamma} \operatorname{ess\,sup} |f_{[0,t]}|\,,$$

where $\operatorname{ess\,sup} |f_{[0,t]}|$ denotes the essential supremum of $|f(t)|$ on the $[0, t]$ interval. It follows that the

drag estimation errors $\tilde{\boldsymbol{s}}_v(t)$ and $\tilde{\boldsymbol{s}}_\omega(t)$ can be decreased as desired by the proper choice of design parameters γ and λ.

We will use the translational and rotational drag estimates $\hat{\boldsymbol{s}}_v(t)$ and $\hat{\boldsymbol{s}}_\omega(t)$ for the trajectory generation and control design purposes. Their respective rate of change is generated according to the corresponding adaptive laws.

The wind linear and angular velocities are computed from the equations

$$\begin{aligned} \dot{\hat{\boldsymbol{s}}}_v(t) &= -m^{-1} R_{B/E}(t)\Phi(\boldsymbol{c}_D)\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)| \\ \dot{\hat{\boldsymbol{s}}}_\omega(t) &= -J^{-1}\Phi(\boldsymbol{c}_\tau)\boldsymbol{\omega}_a^B(t)|\boldsymbol{\omega}_a^B(t)|\,. \end{aligned} \tag{23}$$

Since $\hat{\boldsymbol{s}}_v(t)$ is in the inertial frame, first we translate it to the body frame $\hat{\boldsymbol{s}}_v^B(t) = R_{E/B}(t)\hat{\boldsymbol{s}}_v(t)$, then notice that the components of $\boldsymbol{s}_v^B(t)$ have signs opposite to that of the corresponding components of $\boldsymbol{v}_a^B(t)$ in the body frame. Therefore, we can write the equations

$$\begin{aligned} \boldsymbol{c}_{Dx}|v_{ax}^B(t)|^2 &= m|\hat{s}_{vx}^B(t)| \\ \boldsymbol{c}_{Dy}|v_{ay}^B(t)|^2 &= m|\hat{s}_{vy}^B(t)| \\ \boldsymbol{c}_{Dz}|v_{az}^B(t)|^2 &= m|\hat{s}_{vz}^B(t)|\,, \end{aligned} \tag{24}$$

solving which for the wind components in the body frame we obtain

$$\begin{aligned} \hat{w}_x^B(t) &= v_x^B(t) - \mathrm{sign}(\hat{s}_{vx}^B(t))\sqrt{\frac{m}{c_{Dx}}|\hat{s}_{vx}^B(t)|} \\ \hat{w}_y^B(t) &= v_y^B(t) - \mathrm{sign}(\hat{s}_{vy}^B(t))\sqrt{\frac{m}{c_{Dy}}|\hat{s}_{vy}^B(t)|} \\ \hat{w}_z^B(t) &= v_z^B(t) - \mathrm{sign}(\hat{s}_{vz}^B(t))\sqrt{\frac{m}{c_{Dz}}|\hat{s}_{vz}^B(t)|}\,, \end{aligned} \tag{25}$$

The wind angular velocity components are found similarly

$$\begin{aligned} \hat{\omega}_{cx}^B(t) &= p(t) - \mathrm{sign}(\hat{s}_{\omega x}^B(t))\sqrt{\frac{J_1}{c_{\tau x}}|\hat{s}_{\omega x}^B(t)|} \\ \hat{\omega}_{cy}^B(t) &= q(t) - \mathrm{sign}(\hat{s}_{\omega y}^B(t))\sqrt{\frac{J_2}{c_{\tau y}}|\hat{s}_{\omega y}^B(t)|} \\ \hat{\omega}_{cz}^B(t) &= r(t) - \mathrm{sign}(\hat{s}_{\omega z}^B(t))\sqrt{\frac{J_3}{c_{\tau z}}|\hat{s}_{\omega z}^B(t)|}\,. \end{aligned} \tag{26}$$

Next we compute the wind linear and angular accelerations using $\dot{\hat{\boldsymbol{s}}}_v(t)$ and $\dot{\hat{\boldsymbol{s}}}_\omega(t)$ from the prediction

models (17) and (18). Differentiating equations (23) with respect to time we obtain

$$\begin{aligned}\dot{\hat{\boldsymbol{s}}}_v(t) &= -m^{-1}R_{B/E}(t)\boldsymbol{\omega}^{\times}(t)\Phi(\boldsymbol{c}_D)\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)| - 2m^{-1}R_{B/E}(t)\Phi(\boldsymbol{c}_D)\dot{\boldsymbol{v}}_a^B(t)|\boldsymbol{v}_a^B(t)| \qquad (27)\\ \hat{\boldsymbol{s}}_\omega(t) &= -2J^{-1}\Phi(\boldsymbol{c}_\tau)\dot{\boldsymbol{\omega}}_a^B(t)|\boldsymbol{\omega}_a^B(t)|\,.\end{aligned}$$

Taking into account $\dot{\boldsymbol{v}}_a^B(t) = \dot{\boldsymbol{v}}^B(t) - \dot{\boldsymbol{w}}^B(t)$ and $\dot{\boldsymbol{\omega}}_a^B(t) = \dot{\boldsymbol{\omega}}^B(t) - \dot{\boldsymbol{\omega}}_c^B(t)$ we can write

$$\begin{aligned}\dot{\hat{\boldsymbol{w}}}^B(t) &= \frac{\dot{\hat{\boldsymbol{s}}}_v(t) + m^{-1}R_{B/E}(t)\boldsymbol{\omega}^{\times}(t)\Phi(\boldsymbol{c}_D)\boldsymbol{v}_a^B(t)|\boldsymbol{v}_a^B(t)| + 2m^{-1}R_{B/E}(t)\Phi(\boldsymbol{c}_D)\dot{\boldsymbol{v}}^B(t)|\boldsymbol{v}_a^B(t)|}{2m^{-1}R_{B/E}(t)\Phi(\boldsymbol{c}_D)|\boldsymbol{v}_a^B(t)|}\\ \dot{\hat{\boldsymbol{\omega}}}_c^B(t) &= \frac{\hat{\boldsymbol{s}}_\omega(t) + 2J^{-1}\Phi(\boldsymbol{c}_\tau)\dot{\boldsymbol{\omega}}^B(t)|\boldsymbol{\omega}_a^B(t)|}{2J^{-1}\Phi(\boldsymbol{c}_\tau)|\boldsymbol{\omega}_a^B(t)|}\,, \qquad (28)\end{aligned}$$

where the division is understood component-wise. We notice that the wind accelerations estimates involve the vehicle's inertial linear and angular accelerations expressed in body frame.

IV. Trajectory Generation

In this section we present a trajectory generation algorithm that takes into account the atmospheric effects in multi-copters dynamics using the estimates $\hat{\boldsymbol{w}}(t)$, $\hat{\boldsymbol{\omega}}_c(t)$, $\hat{\boldsymbol{s}}_v(t)$ and $\hat{\boldsymbol{s}}_\omega(t)$ and their derivatives from the previous section. For this purpose, we consider the simplified equation of motion

$$\dot{\boldsymbol{v}}(t) = \bar{f}(t)R_{B/E}(t)\boldsymbol{e}_3^B + \boldsymbol{g} + \hat{\boldsymbol{s}}_v(t)\,, \qquad (29)$$

where the rotation matrix $R_{B/E}(t)$ evolves according to equation

$$\dot{R}_{B/E}(t) = R_{B/E}(t)\boldsymbol{\omega}^{\times}(t)\,, \qquad (30)$$

and the mass-normalized total thrust $\bar{f}(t) = \frac{f_T(t)}{m}$ and the angular rate $\boldsymbol{\omega}(t)$ are viewed as control inputs. The justification of this simplification is that the controller designed for the angular rate dynamics

$$\dot{\boldsymbol{\omega}}(t) = -J^{-1}\boldsymbol{\omega}(t) \times J\boldsymbol{\omega}(t) + J_m\omega_m(t)J^{-1}\bar{\boldsymbol{\omega}}(t) + J^{-1}\boldsymbol{\tau}(t) + \hat{\boldsymbol{s}}_\omega(t) \qquad (31)$$

can provide fast and accurate tracking of the angular rate commands in the presence of rotational drag with or without wind. The design of this controller is presented in the next section.

To generate trajectories we modify the jerk minimization approach of [19] so that the estimate $\hat{\boldsymbol{s}}_v(t)$ of the aerodynamic drag can be directly taken into account. For the given trajectory $\boldsymbol{r}(t) =$

$[x(t)\ y(t)\ z(t)]^\top$, the required mass-normalized thrust vector $\boldsymbol{T}(t) = \bar{f}(t)R_{B/E}(t)\boldsymbol{e}_3^B$ can be expressed as

$$\boldsymbol{T}(t) = \ddot{\boldsymbol{r}}(t) - \boldsymbol{g} - \hat{\boldsymbol{s}}_v(t)\,, \tag{32}$$

which implies that

$$\|\boldsymbol{T}(t)\| = \|\ddot{\boldsymbol{r}}(t) - \boldsymbol{g} - \hat{\boldsymbol{s}}_v(t)\| = \bar{f}(t)\,, \tag{33}$$

That is the mass-normalized total thrust magnitude necessary to traverse the given trajectory is defined by (33). The orientation of the thrust vector is defined by the roll and pitch angles, the rate of change of which is related to the jerk (third derivative of the position) of the given trajectory through the equations (29) and (30). Differentiating (29) and (33) we obtain

$$\dddot{\boldsymbol{r}}(t) = \dot{\bar{f}}(t)R_{B/E}(t)\boldsymbol{e}_3^B + \bar{f}(t)R_{B/E}(t)\boldsymbol{\omega}^\times(t)\boldsymbol{e}_3^B + \dot{\hat{\boldsymbol{s}}}_v(t) \tag{34}$$

$$\dot{\bar{f}}(t) = (\boldsymbol{e}_3^B)^\top R_{E/B}(t)\left(\dddot{\boldsymbol{r}}(t) - \dot{\hat{\boldsymbol{s}}}_v(t)\right)\,, \tag{35}$$

solving which for angular rates results in

$$\begin{bmatrix} \omega_y(t) \\ -\omega_x(t) \\ 0 \end{bmatrix} = \frac{1}{\bar{f}(t)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} R_{E/B}(t)\left(\dddot{\boldsymbol{r}}(t) - \dot{\hat{\boldsymbol{s}}}_v(t)\right)\,, \tag{36}$$

which implies that

$$\sqrt{\omega_x^2(t) + \omega_y^2(t)} \leq \frac{1}{\bar{f}(t)}\left\|\dddot{\boldsymbol{r}}(t) - \dot{\hat{\boldsymbol{s}}}_v(t)\right\|\,. \tag{37}$$

Equation (36) defines the angular rates ω_x and ω_y necessary to traverse the given trajectory. Obviously, a given trajectory can be traversed using the control inputs $\bar{f}$, ω_x and ω_y. That is, only these three control inputs are needed to generate a 3D trajectory. Then, ω_z can be used by the user to control the on-board sensor direction (the rotation of the vehicle around the trust vector) for surveillance, mapping, etc. In this paper, we assume $\omega_z = 0$.

The total thrust generated by the motors satisfies the physical constraint

$$0 \leq f_{\min} \leq \bar{f}(t) \leq f_{\max}\,, \tag{38}$$

and the angular rate input is bounded due to sensor limitations as

$$-\omega_{\max} \leq \boldsymbol{\omega}_a^B(t) \leq \omega_{\max}\,, \tag{39}$$

which directly takes into account the estimate of the wind vorticity $\hat{\boldsymbol{\omega}}_c^B(t)$.

Following the steps of [19], we generate a single axis motion primitives using a third order system

$$\dddot{s}_j(t) = u_j(t) \tag{40}$$

with performance index

$$J = \int_0^{t_f} u_j^2(\tau) d\tau\,, \tag{41}$$

initial conditions $s_j(0),\ \dot{s}_j(0),\ \ddot{s}_j(0)$ and final conditions $s_j(t_f),\ \dot{s}_j(t_f),\ \ddot{s}_j(t_f)$ for each $j = x, y, z$, where t_f is the time to traverse. The resulting closed form solution s_j is a 5th order polynomial in time, the coefficients of which depend on final time t_f and initial and final conditions (we refer the interested reader to [19] for details). Therefore, to fully define $\boldsymbol{s}(t) = [s_x(t)\ s_y(t)\ s_z(t)]^\top$ one needs to select the initial state $\boldsymbol{s}(0),\ \dot{\boldsymbol{s}}(0),\ \ddot{\boldsymbol{s}}(0)$, which we assume to be coincident with the vehicle's current state $\boldsymbol{r}(t), \boldsymbol{v}(t), \boldsymbol{a}(t)$ (assuming all measurements are available), the final time t_f and the final state (fully or partially defined) $\boldsymbol{s}(t_f),\ \dot{\boldsymbol{s}}(t_f),\ \ddot{\boldsymbol{s}}(t_f)$.

In this paper, we propose the following algorithm for trajectory generation. Let a series of waypoints $\boldsymbol{p}_j,\ j = 1, \ldots, N_p$ be provided by some type of planner (not included in the paper), and let $\boldsymbol{p}_0 = \boldsymbol{r}(t)$, where $\boldsymbol{r}(t)$ is the vehicle's position at current time t. For each portion of trajectory between two way points, we compute minimum possible time-to-go by dividing the distance between way points by the maximum possible velocity $V_{\max}$. If $V_{\max}$ is not available from the vehicle's specifications, we solve the equation

$$f_{\max}\boldsymbol{e}_3^B + \max R_{E/B}(t)\boldsymbol{g} = \Phi(\boldsymbol{c}_D)(\boldsymbol{v}^B(t) - \hat{\boldsymbol{w}}^B(t))|\boldsymbol{v}^B(t) - \hat{\boldsymbol{w}}^B(t)| \tag{42}$$

for $\boldsymbol{v}^B(t)$ and set $V_{\max} = \|\boldsymbol{v}^B(t)\|$, using the available wind velocity estimate $\hat{\boldsymbol{w}}^B(t)$. The resulting time-to-go t_{go} is used as a first iteration for final time $t_{f_1} = t_{go}$.

First, we generate a trajectory $\boldsymbol{s}(\eta)$ on the interval $t \leq \eta \leq t_{f_1}$ according to the optimal control problem (40) and (41).

Next, we compute the mass-normalized thrust vector $\boldsymbol{T}(\eta)$ required to traverse the trajectory $\boldsymbol{s}(\eta)$ component-wise using the equations

$$T_x(\eta) = \ddot{s}_x(\eta) - \hat{s}^*_{vx}(t) \tag{43}$$

$$T_y(\eta) = \ddot{s}_y(\eta) - \hat{s}^*_{vy}(t) \tag{44}$$

$$T_z(\eta) = \ddot{s}_z(\eta) - \hat{s}^*_{vz}(t) + g\,. \tag{45}$$

where $\boldsymbol{s}^*_v(t)$ is the worst case prediction of the maximum drag force, since the estimates of the drag force $\hat{\boldsymbol{s}}(t)$ and the wind velocity $\hat{\boldsymbol{w}}(t)$ are available only at current time t. This prediction can be computed as $\boldsymbol{s}^*_v(t) = \hat{\boldsymbol{s}}_v(t) + \eta\dot{\hat{\boldsymbol{s}}}_v(t)$, assuming a constant $\dot{\hat{\boldsymbol{s}}}_v(t)$ on the interval $t \leq \eta \leq t_{f_1}$. The total mass-normalized thrust is computed as $\bar{f}(\eta) = \sqrt{T_x^2(\eta) + T_y^2(\eta) + T_z^2(\eta)}$ and the minimum and maximum values of it $\bar{f}_{\min}$ and $\bar{f}_{\max}$ are numerically computed on the interval $t \leq \eta \leq t_{f_1}$.

Then, we check the thrust feasibility condition (38) for $\bar{f}_{\max}$ and $\bar{f}_{\min}$. If the conditions are satisfied, we compute $\chi(t, t_{f_1}) = \max_{t\leq\eta\leq t_{f_1}} \sqrt{[\ddot{s}_x(\eta) - \dot{\hat{s}}_{vx}(t)]^2 + [\ddot{s}_y(\eta) - \dot{\hat{s}}_{vy}(t)]^2 + [\ddot{s}_z(\eta) - \dot{\hat{s}}_{vz}(t)]^2}$ assuming that $\dot{\hat{\boldsymbol{s}}}_v(t)$ is constant on the interval $t \leq \eta \leq t_{f_1}$, and check the rate feasibility (39). If it is satisfied the trajectory $\boldsymbol{s}(\eta),\ t \leq \eta \leq t_{f_1}$ is marked as feasible trajectory between waypoints $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$, and the algorithm is advanced to the next way point taking the $\boldsymbol{s}(t_{f_1}),\ \dot{\boldsymbol{s}}(t_{f_1}),\ \ddot{\boldsymbol{s}}(t_{f_1})$ as the initial state of the next portion of the trajectory.

If any of the checks fail, we set $t_f = t_f + i\Delta t$ for some time step Δt and $i = 1, \ldots, N$ and return to the first step. The iteration is continued until preset reasonable number N is reached, after which the algorithm exits with no feasible trajectory generated for the corresponding waypoint. In this case, the algorithm waits for the planner to provide new set of waypoints. Meantime, the drone can execute the generated portion of the trajectory (if there exists one) or hover (in not) or land (if not enough battery power remains).

When the algorithm exits with a feasible trajectory, it is a sub-minimum time trajectory. That is, for each portion of the trajectory, the time to traverse is within Δt margin of a true minimum time trajectory, which also minimizes the aggressiveness of the traverse (see [19] for the explanation of the aggressiveness).

A remark on the final state $\boldsymbol{s}(t_{f_i}),\ \dot{\boldsymbol{s}}(t_{f_i}),\ \ddot{\boldsymbol{s}}(t_{f_i})$ selection in the optimal control problem for-

mulation (40) and (41) is in order. The first state (position) $s_j(t_{f_i})$ $i = 1, \dots N_p$ in each direction $j = x, y, z$ is set to corresponding waypoint $\boldsymbol{p}_i = [x_i \; y_i \; z_i]^\top$ provided by the planner. The third (acceleration) state $\dot{\boldsymbol{s}}(t_{f_i})$ is set to zero for all way points. The second (velocity) state $\dot{\boldsymbol{s}}(t_{f_i})$ is selected as follows. We set $\dot{\boldsymbol{s}}(t_{f_{N_p}}) = \mathbf{0}$ and $\ddot{\boldsymbol{s}}(t_{f_{N_p}}) = \mathbf{0}$ at final waypoint $\boldsymbol{p}_{N_p}$ (arriving to-rest). For all other waypoints we set $\dot{s}_x(t_{f_i}) = 0$ if $|x_{i+1} - x_i| \leq \varepsilon_x$ and leave $\dot{s}_x(t_{f_i}) = 0$ free (unspecified) otherwise. Here, $\varepsilon > 0$ is a design parameter that the designer can choose according to the scale of the distance to be traveled in x direction. For remaining two directions the way point velocity is similarly set. This setup enables the vehicle to travel along the straight (or approximately straight) trajectories with maximal speed without slowing down at way points and excessive cross-track maneuvers.

V. Trajectory Tracking Control Design

In this section we design a controller for the multi-rotor to track the 3D trajectory $\boldsymbol{r}_{com}(t) = [x_{com}(t) \; y_{com}(t) \; z_{com}(t)]^\top$ and the heading angle ($\psi_{com}(t)$) commands generated in the previous section.

A. Center of Gravity (CG) Control

The motion of drone's CG is controlled by the thrust vector $\bar{f}(t)R_{B/E}(t)\boldsymbol{e}_3$ or by the magnitude of total thrust $\bar{f}(t)$ and the orientation angles $\phi(t)$ and $\theta(t)$ according the force equation (29), which are designed from the perspective of tracking the trajectory command $\boldsymbol{r}_{com}(t)$ or velocity command $\boldsymbol{v}_{com}(t)$ depending ot the users preferences. When the objective is to track $\boldsymbol{r}_{com}(t)$, the corresponding $\boldsymbol{v}_{com}(t)$ is generated using backstepping approach as

$$\boldsymbol{v}_{com}(t) = c_1[\boldsymbol{r}_{com}(t) - \boldsymbol{r}(t)] + \dot{\boldsymbol{r}}_{com}(t)\,, \tag{46}$$

where $c_1 > 0$ is a design parameter, otherwise

$$\boldsymbol{v}_{com}(t) = \dot{\boldsymbol{r}}_{com}(t)\,. \tag{47}$$

The control law is defined according to equation

$$\bar{f}(t)R_{B/E}(t)\boldsymbol{e}_3 = -\hat{\boldsymbol{s}}_v(t) - \boldsymbol{g} + c_2\left[\boldsymbol{v}_{com}(t) - \boldsymbol{v}(t)\right] + \dot{\boldsymbol{v}}_{com}(t)\,, \tag{48}$$

where $c_2 > 0$ is a design parameter, and $\dot{\boldsymbol{v}}_{com}(t)$ is computed respectively from (46) or (47). Substituting the control law in (29) results in the exponentially stable error dynamics

$$\dot{\boldsymbol{e}}_v(t) = -c_2 \boldsymbol{e}_v(t)\,, \tag{49}$$

for the tracking error $\boldsymbol{e}_v(t) = \boldsymbol{v}(t) - \boldsymbol{v}_{com}(t)$.

The required total thrust and orientation angle commands are obtained from (48) assuming that $-\pi/2 < \phi, \theta < \pi/2$, that is there are no flip-over maneuvers. This assumption ensures that the functions $\cos\phi$ and $\cos\theta$ are nonzero, and the $\sin\phi$ and $\sin\theta$ are one-to-one invertible. It follows from the equation (48) written component-wise

$$\begin{aligned}
\bar{f}(t)\left[\cos\phi(t)\sin\theta(t)\cos\psi(t) + \sin\phi(t)\sin\psi(t)\right] &= -\hat{s}_{vx}(t) - c_1\dot{e}_x(t) - c_2 e_x(t) + \ddot{x}_{ref}(t) \triangleq \kappa_x(t) \\
\bar{f}(t)\left[\cos\phi(t)\sin\theta(t)\sin\psi(t) - \sin\phi(t)\cos\psi(t)\right] &= -\hat{s}_{vy}(t) - c_1\dot{e}_y(t) - c_2 e_y(t) + \ddot{y}_{ref}(t) \triangleq \kappa_y(t) \\
\bar{f}(t)\cos\phi(t)\cos\theta(t) &= g - \hat{s}_{vz}(t) - c_1\dot{e}_z(t) - c_2 e_z(t) + \ddot{z}_{ref}(t) \triangleq \kappa_z(t)
\end{aligned} \tag{50}$$

that the total thrust is readily obtained from the third equation as

$$f_T(t) = m\frac{\kappa_z(t)}{\cos\phi(t)\cos\theta(t)}\,. \tag{51}$$

which basically controls the drone's altitude or vertical speed. Next, multiplying the first equation by $\cos\psi(t)$, the second one by $\sin\psi(t)$, then adding and subtracting them we obtain

$$\begin{aligned}
\phi_{com}(t) &= \sin^{-1}\left(\frac{\kappa_x(t)\sin\psi(t) - \kappa_y(t)\cos\psi(t)}{f_T(t)}\right) \\
\theta_{com}(t) &= \sin^{-1}\left(\frac{\kappa_x(t)\cos\psi(t) + \kappa_y(t)\sin\psi(t)}{f_T(t)\cos\phi(t)}\right)\,.
\end{aligned} \tag{52}$$

B. Attitude Control

Now, we derive the control torque for the rotational dynamics such that the Euler angle $\boldsymbol{E}(t)$ tracks the reference signal $\boldsymbol{E}_{ref}(t)$ generated through the dynamics

$$\dot{\boldsymbol{E}}_{ref}(t) = -c_E\left[\boldsymbol{E}_{ref}(t) - \boldsymbol{E}_{com}(t)\right]\,, \tag{53}$$

where $c_\phi > 0$ is a design constant and $\boldsymbol{E}_{com}(t) = [\phi_{com}(t) \;\; \theta_{com}(t) \;\; \psi_{com}(t)]^\top$ is the combination of roll and pitch angles commands obtained from the perspective of the position tracking and yow

angle command provided by the trajectory generation algorithm. Using time scale separation and dynamic inversion techniques, we first derive an expression for the desired angular rates

$$\boldsymbol{\omega}_{com}(t) = H^{-1}(t)[-c_\omega \boldsymbol{e}_E(t) + \dot{\boldsymbol{E}}_{ref}(t)]\,, \tag{54}$$

where $\boldsymbol{e}_E(t) = \boldsymbol{E}(t) - \boldsymbol{E}_{ref}(t)$ is the attitude angles tracking error, $c_E > 0$ is the control gain, and $H^{-1}(t)$ is the inverse of the matrix $H(t)$ given by

$$H^{-1}(t) = \begin{bmatrix} 1 & 0 & -\sin\theta(t) \\ 0 & \cos\phi(t) & \sin\phi(t)\cos\theta(t) \\ 0 & -\sin\phi(t) & \cos\phi(t)\cos\theta(t) \end{bmatrix}.$$

then we derive the required control torque using equation (31)

$$\boldsymbol{\tau}(t) = \boldsymbol{\omega}(t) \times J\boldsymbol{\omega}(t) - J_{r3}\Omega(t)\bar{\boldsymbol{\omega}}(t) - \hat{\boldsymbol{s}}_\omega(t) + J[-c_\omega \boldsymbol{e}_\omega(t) + \dot{\boldsymbol{\omega}}_{ref}(t)]\,, \tag{55}$$

where $c_\omega > 0$ is the control gain, $\boldsymbol{e}_\omega(t) = \boldsymbol{\omega}(t) - \boldsymbol{\omega}_{ref}(t)$ is the angular rate tracking error, which satisfies the exponentially stable dynamics

$$\dot{\boldsymbol{e}}_\omega(t) = -c_\omega \boldsymbol{e}_\omega(t)\,, \tag{56}$$

and the signal $\boldsymbol{\omega}_{ref}(t)$ is generated through the reference dynamics

$$\dot{\boldsymbol{\omega}}_{ref}(t) = -c_\omega\left[\boldsymbol{\omega}_{ref}(t) - \boldsymbol{\omega}_{com}(t)\right]. \tag{57}$$

The individual motor inputs are obtained by solving the control allocation equation

$$\begin{bmatrix} f_T \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ b_{31} & b_{32} & \dots & b_{3n} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \tag{58}$$

where the coefficients $b_{ij},\ i = 1, 2,\ j = 1, \dots, n$ are easily derived from the geometry of the drone, and $b_{3j} = (-1)^j d,\ j = 1, \dots, n$, where d is the ratio between the drag and the thrust coefficients of the propeller blade, for $f_i,\ i = 1, \dots, n$ following the steps from [27].

VI. Simulation Results

The performance of the presented algorithms are demonstrated in simulations using the dynamic model of DJI S1000 octocopter. Waypoints are generated by means of A^* path planning algorithm

for a digital cityscape assuming that information about dynamic obstacles (city transportation) is provided by on-board sensors (for example LIDAR). Initially, we generate a trajectory through way points using the presented algorithm which takes less than a millisecond using Matlab on a conventional laptop computer. We then we re-plan the trajectory every 5 sec, which corresponds to time interval required by the sensor information processing and way points generation. The octocopter camera direction is commanded to periodically sweep the field of view from $-60deg$ to $60deg$. Figure 1 displays the generated 3D trajectory with corresponding waypoints.

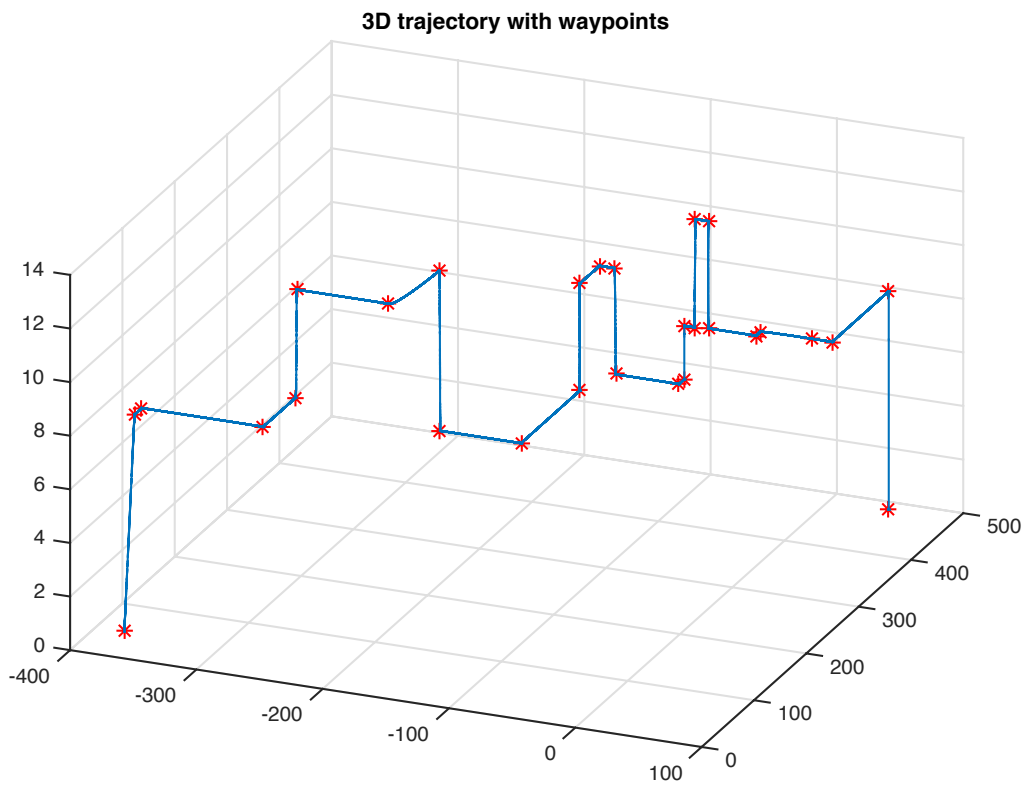
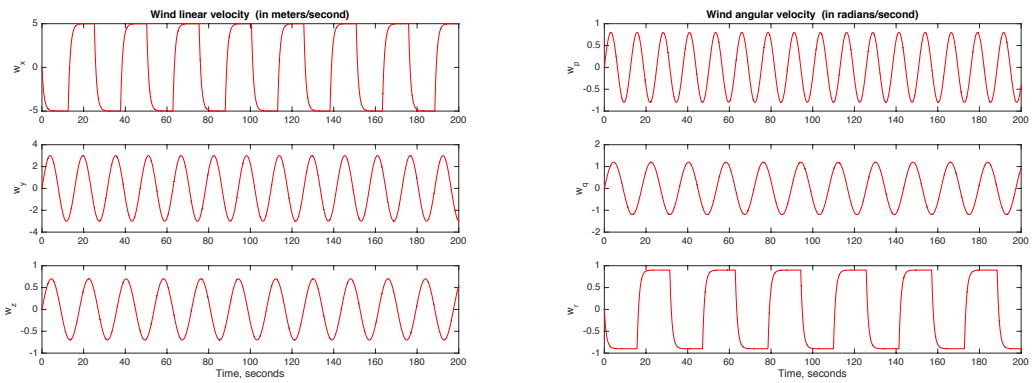


Fig. 1 Generated 3D trajectory and corresponding waypoints.

We introduce wind field with variable linear and angular velocities in all directions. The wind maximum velocity reaches $6m/s$, and the maximum vorticity reaches $1.7rad/sec$. The wind profile along the trajectory is displayed in Figure 2.

The wind estimates are computed from the on-line estimation of resulting linear and rotational drag force and torque according to the presented algorithm. Figures 3 and 4 display the performance



(a) Wind linear velocitis time histories

(b) Wind angular velocitis time histories

Fig. 2 Wind field profile along the trajectory.

of the adaptive estimation algorithms for linear and rotational drag respectively. It can be seen that a good convergence is achieved in all drag components.

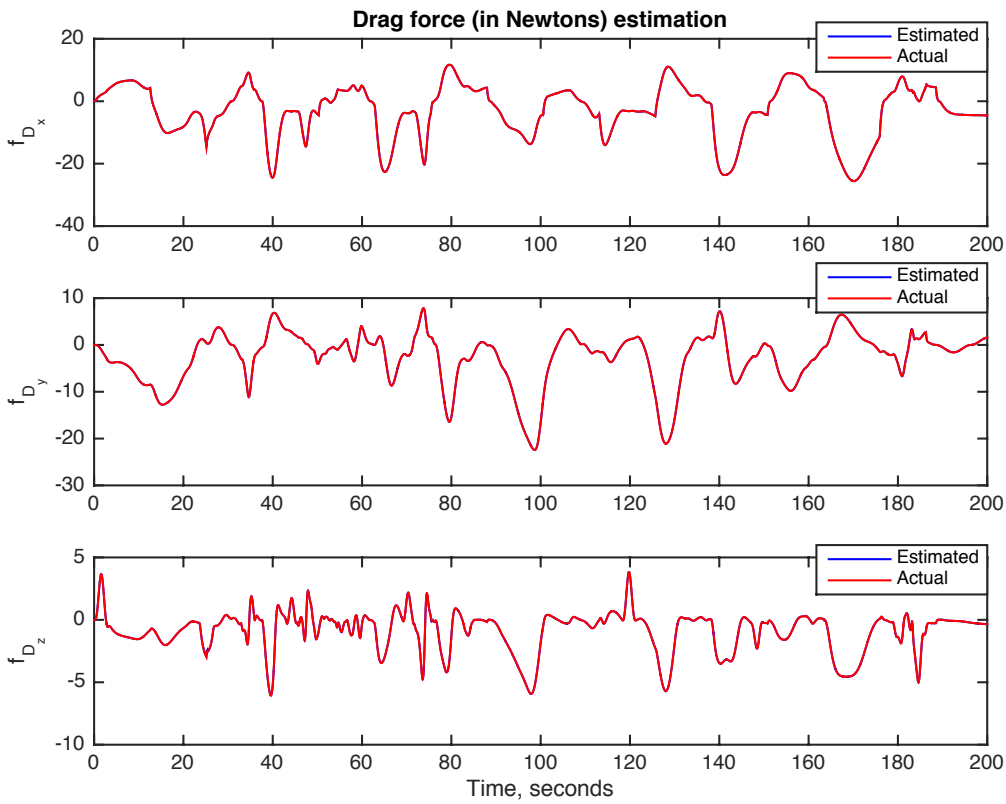


Fig. 3 Drag force estimation.

Figures 5 and 6 display the estimated and actual wind linear and angular velocity components

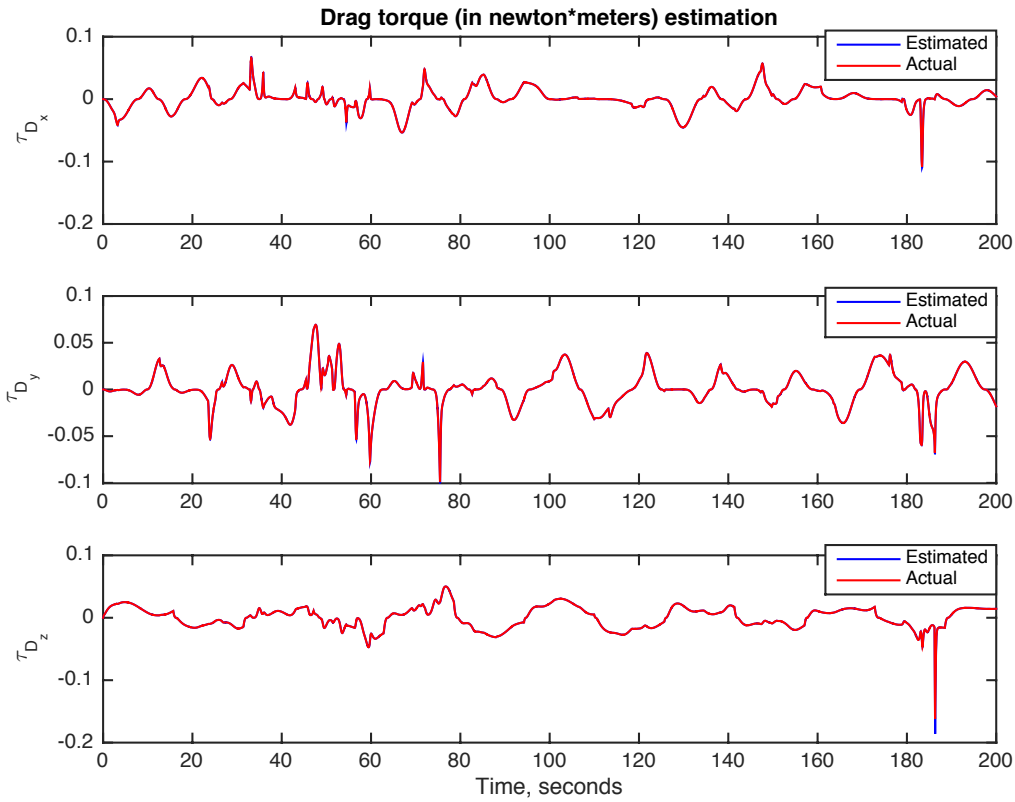


Fig. 4 Drag torque estimation.

along the trajectory of flight. In the computation of these components the *sign* function is used, which in some cases (frequent zero crossing because of numerical errors in computations) results in errors (spikes in w_z are most visible). Further analysis are required to resolve these numeric issues.

The estimated wind linear and angular velocities current at time are used in trajectory re-planning. The controller uses the estimated drag force and torque to generate required total thrust and three-axis torques, which is distributed to the individual motors trough the control allocation technique from [27].

Figure 7 displays the trajectory commands and the controller's tracking performance in three-axis directions. It can be observed that the close tracking is achieved despite severe wind condition, which is presented in Fig. 2. Figure 8 presents the generated velocity command tracking performance. It can be observed that the close tracking is achieved although the objective is to track the position and heading angle commands. Figure 9 displays the octocopter roll and pitch angle

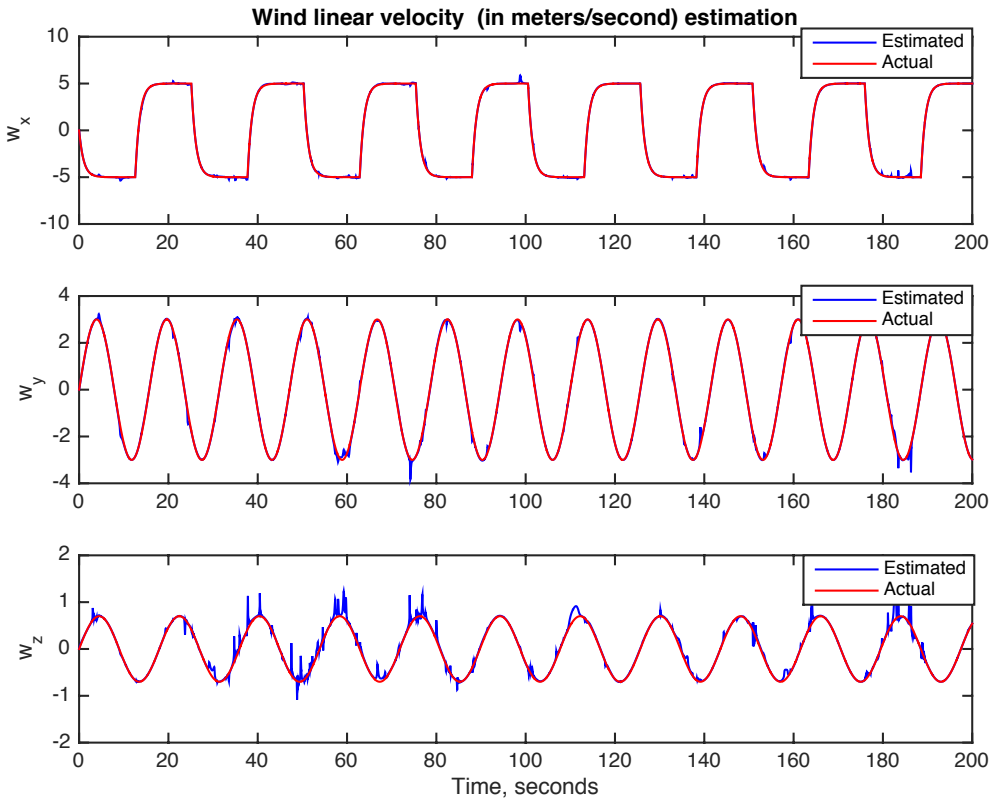


Fig. 5 Wind linear velocity estimation.

time histories and yaw angle tracking performance. It can be observed that during the tracking of the generated trajectories the roll and pitch angles do not exceed angle limits of $45deg$ set in the trajectory generation algorithm.

VII. Conclusion

We have presented a unified estimation, navigation and control approach for multi-rotor drones flying in urban environment. The enclosed adaptive algorithms provide capabilities for the fast and reliable estimation of the drone's aerodynamic drag coefficients in zero wind conditions, and the wind components along the flight trajectory. The navigation algorithms generate sub-minimum time and minimum jerk trajectories between given way-points taking into account the estimated wind. These algorithms are very fast and rely on the analytic solutions of the single axis optimal control problem, the feasibility of which is checked with respect to drone's dynamics, which include the

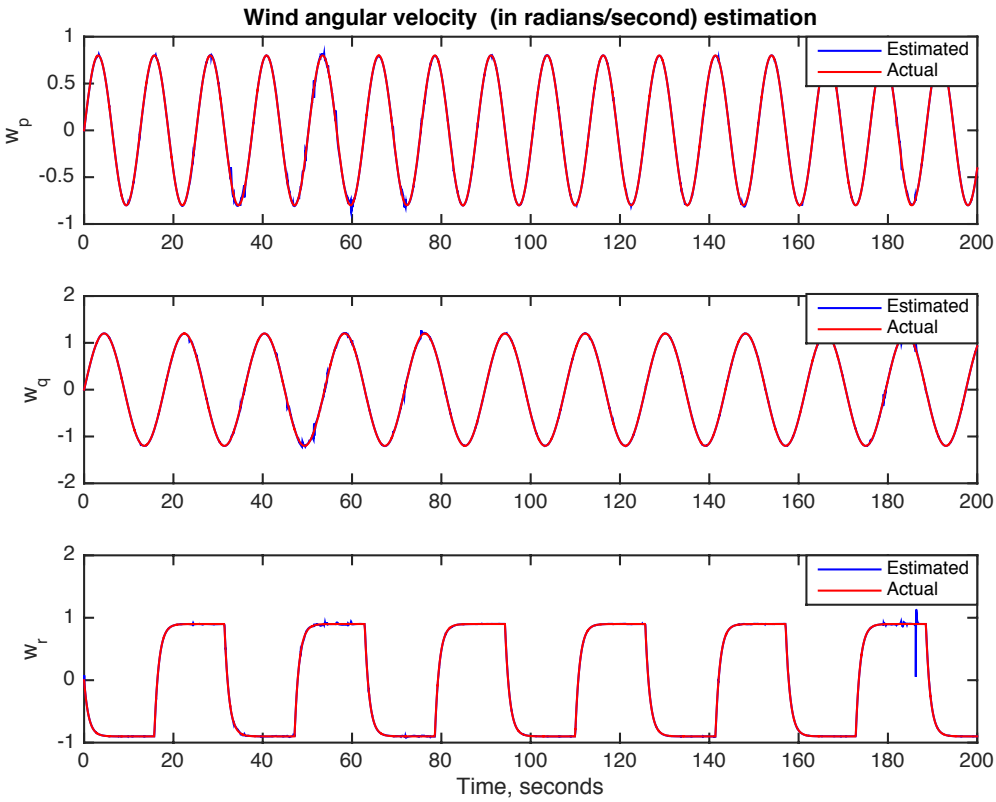


Fig. 6 Wind angular velocity estimation.

real time estimate of aerodynamic drag. The control algorithms are designed to track the generated trajectories as long as the vehicle retains controllability. All algorithms are computationally effective and can be easily implemented in real time using on-board computing power. The benefits of the algorithms were tested in simulations.

Acknowledgment

This work was supported by the NASA Ames SAFE50 Center Innovation Fund (CIF) project and the UAS Traffic Management (UTM) Sub-project, under the NASA's Safe Autonomous Systems Operations (SASO) Project. This research is also partly funded by a NASA-DHS inter-agency agreement. The authors gratefully acknowledge all members of the SAFE50 team for engaging in many hours of discussions on various topics discussed in this paper.

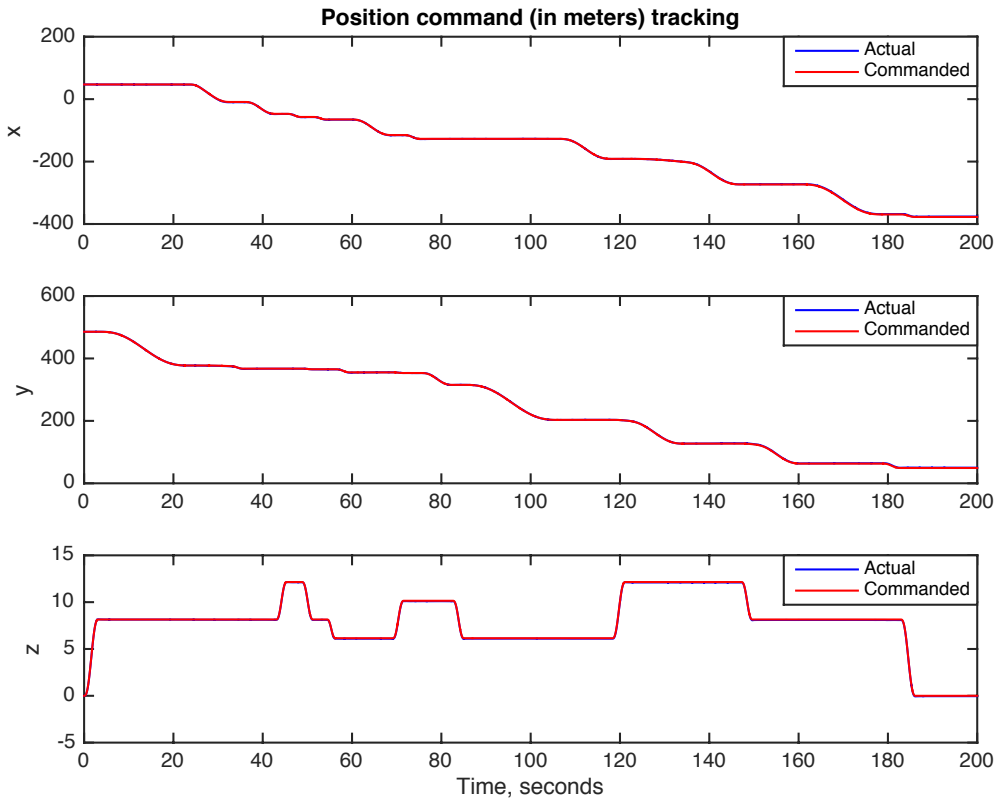


Fig. 7 Trajectory command and tracking performance.

References

[1] A. Ailon1 and I. Zohar. Control of Unmanned Aerial Vehicle With Restricted Input in the Presence of Additive Wind Perturbations. *In Proc. of the 11th International Conference on Control, Automation and Systems, Gyeonggi-do, Korea*, pages 873–878, 2011.

[2] A. Boeuf, J. Cortes, R. Alami, and T. Simeon. Planning Agile Motions for Quadrotors in Constrained Environments. *In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL*, 2014.

[3] P. Bouffard, A. Aswani, and C. J. Tomlin. Learning-Based Model Predictive Control on a Quadrotor: Onboard Implementation and Experimental Results. *In Proc. of the IEEE International Conference on Robotics and Automation*, 2012.

[4] Y. Bouktir, M. Haddad, and T. Chettibi. Trajectory Planning for a Quadrotor Helicopter. *In Proc. of the 16th Mediterenean Conference on Control and Automation, Ajaccio, France*, pages 1258–1263, 2008.

[5] A. Chakrabarty and J. Langelaan. UAV Flight Path Planning in Time Varying Complex Wind-fields.

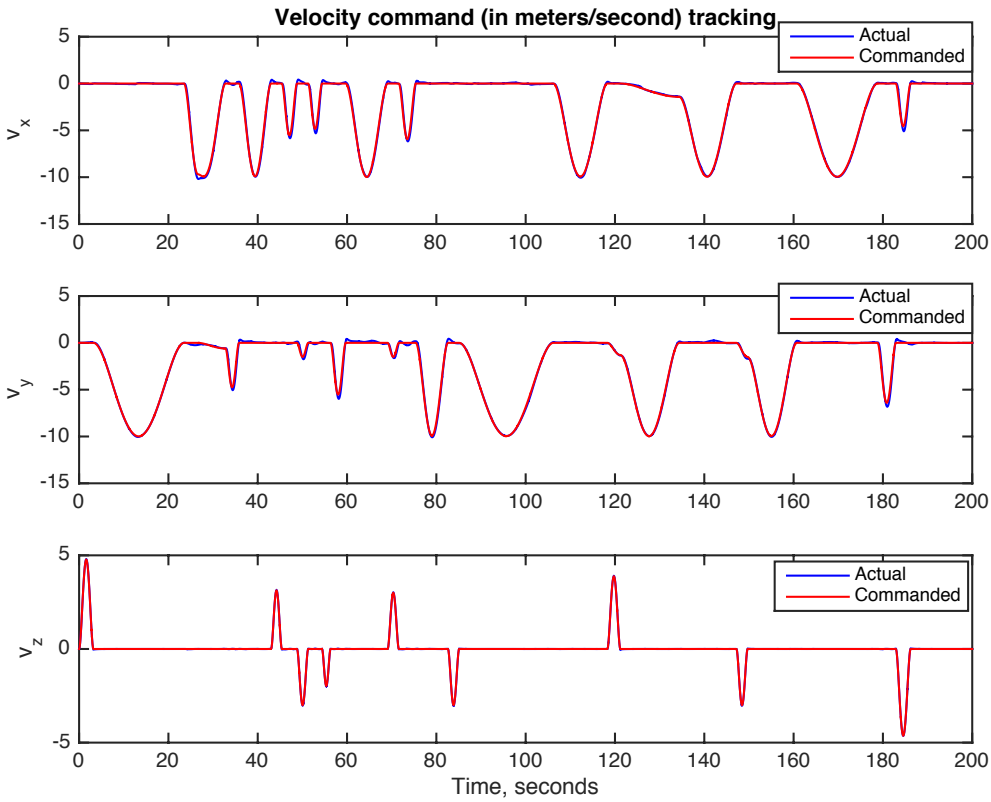


Fig. 8 Velocity command and tracking performance.

In Proc. of the American Control Conference, Washington, DC, pages 2568–2574, 2013.

[6] I. D. Cowling, O. A. Yakimenko, and J. F. Whidborne. A Prototype of an Autonomous Controller for a Quadrotor UAV. *In Proc. of the European Control Conference*, pages 4001–4008, 2007.

[7] G. P. Falconi, C. D. Heise, and F. Holzapfel. Fault-Tolerant Position Tracking of a Hexacopter Using an Extended State Observer. *In Proceedings of the 6th International Conference on Automation, Robotics and Applications (ICARA), DOI: 10.1109/ICARA.2015.7081207*, pages 550 – 556, 2015.

[8] J.A. Guerrero, J.A. Escareno, and Y. Bestaoui. Quad-rotor MAV Trajectory Planning in Wind Fields. *In Proc. of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany*, pages 770–775, 2013.

[9] M. Hehn and R. D'Andrea. Quadrocopter Trajectory Generation and Control. *In Proc. of the IFAC World Congress*, 18(1):1485–1491, 2011.

[10] M. Hehn, R. Ritz, and R. D'Andrea. Performance Benchmarking of Quadrotor Systems Using Time-optimal control. *Lournal of Aitomation and Robotics*, 33(1/2):69–88, 2012.

[11] G. M. Hoffmann, S. L.Waslander, and C. J. Tomlin. Quadrotor Helicopter Trajectory Tracking Control.

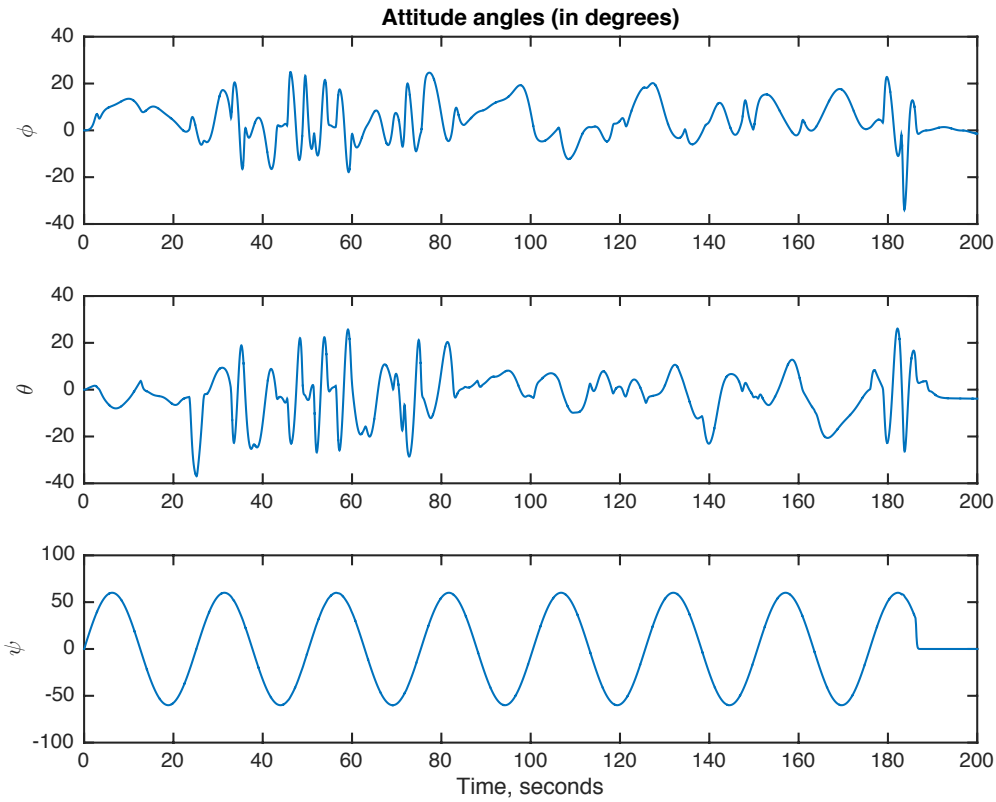


Fig. 9 The octocopter attitude angles in degrees.

In Proc. of the AIAA Guidance, Navigation, and Control Conference, 2008.

[12] M. Q. Huynh, W. Zhao, and L. Xie. L1 Adaptive Control for Quadcopter: Design and Implementation. *In Proc. of the 13th International Conference on Control, Automation, Robotics and Vision, Marina Bay Sands, Singapore*, pages 1496–1501, 2014.

[13] l. Yang, J. Xiao, and X. Yong. A Literature Review of UAV 3D Path Planning. *In Proc. of the 11th World Congress on Intelligent Control and Automation, Shenyang, Chaina*, pages 2376–2381, 2014.

[14] T. Larrabee, H. Chao, M. Rhudy, Y. Gu, and M. R. Napolitano. Wind Field Estimation in UAV Formation Flight. *In Proc. of the American Control Conference, Portland, Oregon*, pages 5408–5413, 2014.

[15] C. Liu, O. McAreet, and W. Chen. Path Following for Small UAV s in the Presence of Wind Disturbance. *In Proc. of the UKACC International Conference on Control, Cardiff, UK*, pages 613–618, 2012.

[16] W. Van Loock, G. Pipeleers, and J. Swevers. Time-Optimal Quadrotor Flight. *In Proc. of the European Control Conference*, pages 1788–1792, 2013.

[17] A. Mart?nez-Vasquez, A. Rodriguez-Mata, and I. Gonzalez-Hernandez. Linear Observer for Estimat-

ing Wind Gust in UAV's. *In Proc. of the 12th International Conference on Electrical Engineering, Computing Science and Automatic Control, Mexico City, Mexico*, pages 873–878, 2015.

[18] D. Mellinger and V. Kumar. Minimum Snap Trajectory Generation and Control for Quadrotors. *In Proc. of the IEEE International Conference on Robotics and Automation, Shanghai, China*, pages 2520–2525, 2011.

[19] M. W. Mueller, M. Hehn, and R. D'Andrea. A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation. *IEEE Transactions on Robotics*, 31(6):820–825, December 2015.

[20] K.S. Narendra and A.M. Annaswamy. *Stable Adaptive Control*. Prentice Hall, 1989.

[21] C. Richter, A. Bry, and N. Roy. Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments. *In Proc. of the IEEE International Symposium on Robotics Research*, 2013.

[22] I. Sarras and H. Siguerdidjane. On the Guidance of a UAV Under Unknown Wind Disturbances. *In Proc. of the IEEE Multi-conference on Systems and Control, Antibes, France*, pages 820–825, 2014.

[23] S. S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence and Robustness*. Prentice Hall, 1989.

[24] S. Schopferer and T. Pfeifer. Performance-Aware Flight Path Planning for Unmanned Aircraft in Uniform Wind Fields. *In Proc. of the International Conference on Unmanned Aircraft Systems, Denver, Colorado*, pages 1138–1147, 2015.

[25] Q. Shen, D. Wang, S. Zhu, and E. K. Poh. Fault-Tolerant Attitude Tracking Control for a Quadrotor Aircraft. *In Proceedings of the 53rd IEEE Conference on Decision and Control, DOI: 10.1109/CDC.2014.7040349*, pages 6129 – 6134, 2014.

[26] V. Stepanyan and K. Krishnakumar. Adaptive Control with Reference Model Modification. *AIAA Journal of Guidance, Control, and Dynamics*, 35(4):1370–1374, 2012.

[27] V. Stepanyan, K. Krishnakumar, and A. Bencomo. Identification and reconfigurable Control of Impaired Multi-Rotor Drones. *In Proc. of the AIAA Guidance, Navigation, and Control Conference, San Diego, CA*, 2016.

[28] L. Techy and C. A. Woolsey. Minimum-Time Path Planning for Unmanned Aerial Vehicles in Steady Uniform Winds. *AIAA Journal of Guidance, Control, and Dynamics*, 32(6):1736–1746, 2009.