Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles

Ewoud J. J. Smeur,* Qiping Chu,† and Guido C. H. E. de Croon‡
Delft University of Technology, 2629 HS Delft, The Netherlands

DOI: 10.2514/1.G001490

Incremental nonlinear dynamic inversion is a sensor-based control approach that promises to provide high-performance nonlinear control without requiring a detailed model of the controlled vehicle. In the context of attitude control of micro air vehicles, incremental nonlinear dynamic inversion only uses a control effectiveness model and uses estimates of the angular accelerations to replace the rest of the model. This paper provides solutions for two major challenges of incremental nonlinear dynamic inversion control: how to deal with measurement and actuator delays, and how to deal with a changing control effectiveness. The main contributions of this article are 1) a proposed method to correctly take into account the delays occurring when deriving angular accelerations from angular rate measurements; 2) the introduction of adaptive incremental nonlinear dynamic inversion, which can estimate the control effectiveness online, eliminating the need for manual parameter estimation or tuning; and 3) the incorporation of the momentum of the propellers in the controller. This controller is suitable for vehicles that experience a different control effectiveness across their flight envelope. Furthermore, this approach requires only very coarse knowledge of model parameters in advance. Real-world experiments show the high performance, disturbance rejection, and adaptiveness properties.

Nomenclature

b	=	width of the vehicle, m
$\boldsymbol{I}$	=	identity matrix
$\boldsymbol{I}_r$	=	moment of inertia matrix of the rotor, $\text{kg} \cdot \text{m}^2$
$\boldsymbol{I}_v$	=	moment of inertia matrix of the vehicle, $\text{kg} \cdot \text{m}^2$
i	=	rotor index
k_1	=	force constant of the rotors, $\text{kg} \cdot \text{m/rad}$
k_2	=	moment constant of the rotors, $\text{kg} \cdot \text{m}^2/\text{rad}$
l	=	length of the vehicle, m
$\boldsymbol{M}_a$	=	aerodynamic moment vector acting on the vehicle, $\text{N} \cdot \text{m}$
$\boldsymbol{M}_c$	=	control moment vector acting on the vehicle, $\text{N} \cdot \text{m}$
$\boldsymbol{M}_r$	=	moment vector acting on the propeller, $\text{N} \cdot \text{m}$
T_s	=	sample time of the controller, s
$\boldsymbol{u}$	=	actuator input vector, rad/s
$\boldsymbol{v}$	=	vehicle velocity vector, m/s
$\boldsymbol{\mu}$	=	adaptation rate diagonal matrix
$\boldsymbol{\Omega}$	=	vehicle angular rate vector, rad/s
$\dot{\boldsymbol{\Omega}}$	=	angular acceleration vector, rad/s^2
$\boldsymbol{\omega}$	=	angular rate vector of the four rotors around the body z axis, rad/s
$\boldsymbol{\omega}_i$	=	angular rate vector of rotor i around each of the body axes, rad/s

I. Introduction

MICRO air vehicles (MAVs) have increased in popularity as low-cost lightweight processors and inertial measurement units have become available through the smartphone revolution. The inertial sensors allow stabilization of unstable platforms by feedback algorithms. Typically, the stabilization algorithm used for MAVs is simple proportional integral derivative (PID) control [1,2]. Problems with PID control occur when the vehicle is highly nonlinear or when the vehicle is subject to large disturbances like wind gusts.

Alternatively, we could opt for a model-based attitude controller. A model-based controller that can deal with nonlinear systems is nonlinear dynamic inversion (NDI), which involves modeling all of the MAV's forces and dynamics. Theoretically, this method can remove all nonlinearities from the system and create a linearizing control law. However, NDI is very sensitive to model inaccuracies [3]. Obtaining an accurate model is often expensive or impossible with the constraints of the sensors that are carried onboard a small MAV.

The incremental form of nonlinear dynamic inversion (INDI) is less model-dependent and more robust. It has been described in the literature since the late 1990s [4,5], sometimes referred to as simplified [6] or enhanced [7] NDI. Compared to NDI, instead of modeling the angular acceleration based on the state and inverting the actuator model to get the control input, the angular acceleration is measured, and an increment of the control input is calculated based on a desired increment in angular acceleration. This way, any unmodeled dynamics, including wind gust disturbances, are measured and compensated. Because INDI makes use of a sensor measurement to replace a large part of the model, it is considered a sensor-based approach.

INDI faces two major challenges. First, the measurement of angular acceleration is often noisy and requires filtering. This filtering introduces a delay in the measurement, which should be compensated for. Second, the method relies on inverting and therefore modeling the controls. To achieve a more flexible controller, the control effectiveness should be determined adaptively.

Delay in the angular acceleration measurement has been a prime topic in INDI research. A proposed method to deal with these measurement delays is predictive filtering [8]. However, the prediction of angular acceleration requires additional modeling. Moreover, disturbances cannot be predicted. Initially, a setup with multiple accelerometers was proposed by Bacon and Ostroff [5] to measure the angular acceleration. This setup has some drawbacks because it is complex and the accelerometers are sensitive to structural vibrations. Later, they discussed the derivation of angular acceleration from gyroscope measurements by using a second-order filter [9]. To compensate for the delay introduced by the filter, Bacon and Ostroff [5] use a lag filter on the applied input to the system. We show in this paper that perfect synchronization of input and measured output can be achieved by applying the filter used for the gyroscope differentiation on the incremented input as well.

Received 23 May 2015; revision received 25 September 2015; accepted for publication 2 October 2015; published online 28 December 2015.

*Ph.D. Candidate, Control and Simulation Department.
†Associate Professor, Control and Simulation Department. Member AIAA.
‡Assistant Professor, Control and Simulation Department.

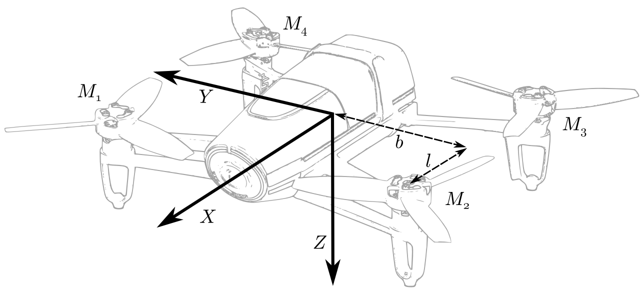

Fig. 1 Bebop quadcopter used in the experiments with axis definitions.

Other research focused on compensating delays in the inputs by using a Lyapunov-based controller design [10]. In this paper, we show that delayed inputs (actuator dynamics) are naturally handled by the INDI controller.

The control effectiveness is the sole model still required by INDI. The parameters can be obtained by careful modeling of the actuators and the moment of inertia or by analyzing the input output data from flight logs. However, even if such a tedious process is followed, the control effectiveness can change during flight. For instance, this can occur due to changes in flight conditions [11] or actuator damage [12]. To cope with this, we propose a method to adaptively determine the control effectiveness matrices.

In this paper, we present three main contributions: 1) a mathematically sound way of dealing with the delays originating from filtering of the gyroscope measurements, 2) the introduction of an adaptive INDI scheme, which can estimate the control effectiveness online, and 3) incorporation of propeller momentum in the controller design. These contributions are implemented and demonstrated on a Parrot Bebop quadrotor running the Paparazzi open-source autopilot software. This is a commercially available quadrotor, and the code is publicly available on Github.[§]

The presented theory and results generalize to other vehicles in a straightforward manner. We have applied this control approach successfully to a variety of quadrotors. Some of these MAVs were able to measure the rotational rate of the rotors (actuator feedback), but some did not have this ability. The INDI controller is believed to scale well to different types of MAVs like helicopter, multirotor, fixed wing, or hybrid.

The outline of this paper is as follows. First, a model of the MAV will be discussed in Sec. II. Second, Sec. III will deal with INDI and the analysis for this controller for a quadrotor. Section IV is about the adaptive extension of INDI. Finally, in Sec. V, the experimental setup is explained, followed by the results of the experiments in Sec. VI.

II. Micro Air Vehicle Model

The Bebop quadrotor is shown in Fig. 1 along with axis definitions. The actuators drive the four rotors, whose angular velocity in the body frame is given by $\boldsymbol{\omega}_i = [\omega_{i_x}, \omega_{i_y}, \omega_{i_z}]$, where i denotes the rotor number. The center of gravity is located in the origin of the axis system, and the distance to each of the rotors along the X axis is given by l and along the Y axis by b.

If the angular velocity vector of the vehicle is denoted by $\boldsymbol{\Omega} = [p, q, r]^T$ and its derivative by $\dot{\boldsymbol{\Omega}}$, the rotational dynamics are given by Euler's equation of motion [13], more specifically the one that describes rotation. If we consider the body axis system as our coordinate system, we get Eq. (1) for the angular velocity of the vehicle:

$$\boldsymbol{I}_v\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \boldsymbol{I}_v\boldsymbol{\Omega} = \boldsymbol{M} \tag{1}$$

where $\boldsymbol{M}$ is the moment vector acting on the vehicle. If we consider the rotating propellers, still in the body coordinate system, we obtain

$$\boldsymbol{I}_r\dot{\boldsymbol{\omega}}_i + \boldsymbol{\Omega} \times \boldsymbol{I}_r\boldsymbol{\omega}_i = \boldsymbol{M}_{r_i} \tag{2}$$

where $\boldsymbol{\omega}_i$ is the angular rate vector of the ith propeller in the vehicle body axes, and $\boldsymbol{\Omega}$ is the angular rotation of the coordinate system, equal to the vehicle body rates. The rotors are assumed to be flat in the z axis, such that the inertia matrix $\boldsymbol{I}_r$ has elements that are zero: $I_{r_{xz}} = I_{r_{yz}} = 0$. Because the coordinate system is fixed to the vehicle, $I_{r_{xx}}$, $I_{r_{xy}}$, and $I_{r_{yy}}$ are not constant in time. However, as is shown later on, the terms containing these moments of inertia will disappear. Expanding Eq. (2) into its three components gives

$$\begin{aligned}
&I_{r_{xx}}\dot{\omega}_{i_x} - I_{r_{yy}}\Omega_z\omega_{i_y} - I_{r_{xy}}\Omega_z\omega_{i_x} + I_{r_{zz}}\Omega_y\omega_{i_z} = M_{r_{i_x}}\\
&I_{r_{yy}}\dot{\omega}_{i_y} + I_{r_{xx}}\Omega_z\omega_{i_x} + I_{r_{xy}}\Omega_z\omega_{i_y} - I_{r_{zz}}\Omega_x\omega_{i_z} = M_{r_{i_y}}\\
&I_{r_{zz}}\dot{\omega}_{i_z} - I_{r_{xx}}\Omega_y\omega_{i_x} - I_{r_{xy}}\Omega_y\omega_{i_y} + I_{r_{yy}}\Omega_x\omega_{i_y} + I_{r_{xy}}\Omega_x\omega_{i_x} = M_{r_{i_z}}
\end{aligned} \tag{3}$$

The propellers are lightweight and have a small moment of inertia compared to the vehicle. Relevant precession terms are therefore those that contain the relatively large ω_{i_z}. Because the rotors spin around the z axis, it is safe to assume that $\omega_{i_x} \ll \omega_{i_z}$ and $\omega_{i_y} \ll \omega_{i_z}$ and that $\dot{\omega}_{i_x}$ and $\dot{\omega}_{i_y}$ are negligible. Then, the moments exerted on the rotors due to their rotational dynamics are given by Eq. (4). Note the presence of the term $I_{r_{zz}}\dot{\omega}_{i_z}$, which is the moment necessary to change the angular velocity of a rotor. In Sec. VI, it will be shown that this term is important

$$\boldsymbol{M}_{r_i} = \begin{bmatrix} M_{r_{i_x}} \\ M_{r_{i_y}} \\ M_{r_{i_z}} \end{bmatrix} = \begin{bmatrix} I_{r_{zz}}\Omega_y\omega_{i_z} \\ -I_{r_{zz}}\Omega_x\omega_{i_z} \\ I_{r_{zz}}\dot{\omega}_{i_z} \end{bmatrix} \tag{4}$$

This equation holds for each of the four rotors, and so the moment acting on a rotor is given a subscript i to indicate the rotor number. The total moment due to the rotational effects of the rotors is shown in Eq. (5). Because motors 1 and 3 spin in the opposite direction of rotors 2 and 4, a factor $(-1)^i$ is introduced. Because we are left with only the z component for the angular velocity of each rotor, we will omit this subscript and continue with the vector $\boldsymbol{\omega} = [\omega_{1_z}, \ldots, \omega_{4_z}]^T = [\omega_1, \ldots, \omega_4]^T$:

$$\begin{aligned}
\boldsymbol{M}_r &= \sum_{i=1}^{4} \boldsymbol{M}_{r_i} = \sum_{i=1}^{4} (-1)^{i+1} \begin{bmatrix} I_{r_{zz}}\Omega_y\omega_i \\ -I_{r_{zz}}\Omega_x\omega_i \\ I_{r_{zz}}\dot{\omega}_i \end{bmatrix}\\
&= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} \end{bmatrix} \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \\ \dot{\omega}_4 \end{bmatrix}\\
&+ \begin{bmatrix} I_{r_{zz}}\Omega_y & -I_{r_{zz}}\Omega_y & I_{r_{zz}}\Omega_y & -I_{r_{zz}}\Omega_y \\ -I_{r_{zz}}\Omega_x & I_{r_{zz}}\Omega_x & -I_{r_{zz}}\Omega_x & I_{r_{zz}}\Omega_x \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}
\end{aligned} \tag{5}$$

Now consider the Euler equation [Eq. (1)] for the entire vehicle. The moments from the rotor dynamics are subtracted from the other moments, yielding

$$\boldsymbol{I}_v\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \boldsymbol{I}_v\boldsymbol{\Omega} = \boldsymbol{M}_c(\boldsymbol{\omega}) + \boldsymbol{M}_a(\boldsymbol{\Omega}, \boldsymbol{v}) - \boldsymbol{M}_r(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \boldsymbol{\Omega}) \tag{6}$$

Here, $\boldsymbol{I}_v$ is the moment of inertia matrix of the vehicle, $\boldsymbol{M}_r(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \boldsymbol{\Omega})$ is the gyroscopic effect of the rotors, $\boldsymbol{M}_c(\boldsymbol{\omega})$ is the control moment vector generated by the rotors, and $\boldsymbol{M}_a(\boldsymbol{\Omega}, \boldsymbol{v})$ is the moment vector generated by aerodynamic effects, which depends on the angular rates and the MAV velocity vector $\boldsymbol{v}$. The control moment $\boldsymbol{M}_c(\boldsymbol{\omega})$ is elaborated in Eq. (7), where k_1 is the force constant of the rotors, k_2 is the moment constant of the rotors, and b and l are defined in Fig. 1

[§]Data available online at https://github.com/EwoudSmeur/paparazzi/tree/bebop_indi_experiment [retrieved 23 November 2015].

$$\boldsymbol{M}_c = \begin{bmatrix} bk_1(-\omega_1^2+\omega_2^2+\omega_3^2-\omega_4^2) \\ lk_1(\omega_1^2+\omega_2^2-\omega_3^2-\omega_4^2) \\ k_2(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2) \end{bmatrix} = \begin{bmatrix} -bk_1 & bk_1 & bk_1 & -bk_1 \\ lk_1 & lk_1 & -lk_1 & -lk_1 \\ k_2 & -k_2 & k_2 & -k_2 \end{bmatrix} \boldsymbol{\omega}^2 \tag{7}$$

If we now take Eq. (6), insert Eqs. (4) and (7), and solve for the angular acceleration $\dot{\boldsymbol{\Omega}}$, we arrive at the following:

$$\begin{aligned} \dot{\boldsymbol{\Omega}} &= \boldsymbol{I}_v^{-1}(\boldsymbol{M}_a(\Omega, \boldsymbol{v}) - \Omega \times \boldsymbol{I}_v \Omega) + \boldsymbol{I}_v^{-1}(\boldsymbol{M}_c - \boldsymbol{M}_r) \\ &= \boldsymbol{F}(\Omega, \boldsymbol{v}) + \frac{1}{2}\boldsymbol{G}_1\boldsymbol{\omega}^2 - T_s\boldsymbol{G}_2\dot{\boldsymbol{\omega}} - \boldsymbol{C}(\Omega)\boldsymbol{G}_3\boldsymbol{\omega} \end{aligned} \tag{8}$$

where $\boldsymbol{F}(\boldsymbol{\Omega}, \boldsymbol{v}) = \boldsymbol{I}_v^{-1}(\boldsymbol{M}_a(\boldsymbol{\Omega}, \boldsymbol{v}) - \boldsymbol{\Omega} \times \boldsymbol{I}_v \boldsymbol{\Omega})$ are the forces independent of the actuators, and $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, $\boldsymbol{G}_3$, and $\boldsymbol{C}(\boldsymbol{\Omega})$ are given by Eqs. (9–12), respectively. Note that the sample time T_s of the quadrotor is introduced to ease future calculations:

$$\boldsymbol{G}_1 = 2\boldsymbol{I}_v^{-1} \begin{bmatrix} -bk_1 & bk_1 & bk_1 & -bk_1 \\ lk_1 & lk_1 & -lk_1 & -lk_1 \\ k_2 & -k_2 & k_2 & -k_2 \end{bmatrix} \tag{9}$$

$$\boldsymbol{G}_2 = \boldsymbol{I}_v^{-1} T_s^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} \end{bmatrix} \tag{10}$$

$$\boldsymbol{G}_3 = \boldsymbol{I}_v^{-1} \begin{bmatrix} I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} \\ -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

$$\boldsymbol{C}(\boldsymbol{\Omega}) = \begin{bmatrix} \boldsymbol{\Omega}_y & 0 & 0 \\ 0 & \boldsymbol{\Omega}_x & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{12}$$

Note that traditionally in the literature, the system solved by INDI has the form of $\dot{x} = f(x) + g(x, u)$ where x is the state of the system and u the input to the system. However, as becomes clear from Eq. (8), the quadrotor is actually a system of the form $\dot{x} = f(x) + g(x, u, \dot{u})$. In Sec. III, a solution to this type of problem will be shown.

III. Incremental Nonlinear Dynamic Inversion

Consider Eq. (8) from the previous section. This equation has some extra terms compared to previous work [8] because the gyroscopic and angular momentum effects of the rotors are included. We can apply a Taylor expansion to Eq. (8) and if we neglect higher-order terms, this results in Eq. (13):

$$\begin{aligned} \dot{\boldsymbol{\Omega}} &= \boldsymbol{F}(\boldsymbol{\Omega}_0, \boldsymbol{v}_0) + \frac{1}{2}\boldsymbol{G}_1\boldsymbol{\omega}_0^2 + T_s\boldsymbol{G}_2\dot{\boldsymbol{\omega}}_0 - \boldsymbol{C}(\boldsymbol{\Omega}_0)\boldsymbol{G}_3\boldsymbol{\omega}_0 \\ &+ \frac{\partial}{\partial \boldsymbol{\Omega}}(\boldsymbol{F}(\Omega, \boldsymbol{v}_0) + \boldsymbol{C}(\Omega)\boldsymbol{G}_3\boldsymbol{\omega}_0)|_{\Omega=\Omega_0}(\boldsymbol{\Omega} - \boldsymbol{\Omega}_0) \\ &+ \frac{\partial}{\partial \boldsymbol{v}}(\boldsymbol{F}(\boldsymbol{\Omega}_0, \boldsymbol{v}))|_{v=v_0}(\boldsymbol{v} - \boldsymbol{v}_0) \\ &+ \frac{\partial}{\partial \boldsymbol{\omega}}\left(\frac{1}{2}\boldsymbol{G}_1\boldsymbol{\omega}^2 - \boldsymbol{C}(\boldsymbol{\Omega}_0)\boldsymbol{G}_3\boldsymbol{\omega}\right)\Big|_{\omega=\omega_0}(\boldsymbol{\omega} - \boldsymbol{\omega}_0) \\ &+ \frac{\partial}{\partial \dot{\boldsymbol{\omega}}}(T_s\boldsymbol{G}_2\dot{\boldsymbol{\omega}})\Big|_{\dot{\omega}=\dot{\omega}_0}(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0) \end{aligned} \tag{13}$$

This equation predicts the angular acceleration after an infinitesimal time step ahead in time based on a change in angular rates of the vehicle and a change in rotational rate of the rotors. Now observe that the first terms give the angular acceleration based on the current rates and inputs: $\boldsymbol{F}(\boldsymbol{\Omega}_0, \boldsymbol{v}_0) + \frac{1}{2}\boldsymbol{G}_1\boldsymbol{\omega}_0^2 + T_s\boldsymbol{G}_2\dot{\boldsymbol{\omega}}_0 - \boldsymbol{C}(\boldsymbol{\Omega}_0)\boldsymbol{G}_3\boldsymbol{\omega}_0 = \dot{\boldsymbol{\Omega}}_0$. This angular acceleration can be obtained by deriving it from the angular rates, which are measured with the gyroscope. In other words, these terms are replaced by a sensor measurement, which is why INDI is also referred to as sensor-based control.

The second and third term, partial to $\boldsymbol{\Omega}$ and $\boldsymbol{v}$, are assumed to be much smaller than the fourth and fifth term, partial to $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$. This is commonly referred to as the principle of time scale separation [14]. This assumption only holds when the actuators are sufficiently fast and have more effect compared to the change in aerodynamic and precession moments due to changes in angular rates and body speeds. These assumptions and calculation of the partial derivatives give Eq. (14):

$$\begin{aligned} \dot{\boldsymbol{\Omega}} &= \dot{\boldsymbol{\Omega}}_0 + \boldsymbol{G}_1\,\mathrm{diag}(\boldsymbol{\omega}_0)(\boldsymbol{\omega} - \boldsymbol{\omega}_0) + T_s\boldsymbol{G}_2(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0) \\ &- \boldsymbol{C}(\boldsymbol{\Omega}_0)\boldsymbol{G}_3(\boldsymbol{\omega} - \boldsymbol{\omega}_0) \end{aligned} \tag{14}$$

Previously, it is stated that the angular acceleration is measured by deriving it from the angular rates. In most cases, the gyroscope measurements from a MAV are noisy due to vibrations of the vehicle due to the propellers and motors. Because differentiation of a noisy signal amplifies the noise, some filtering is required. The use of a second-order filter is adopted from the literature [9], of which a transfer function in the Laplace domain is given by Eq. (15). Satisfactory results were obtained with $\omega_n = 50$ rad/s and $\zeta = 0.55$. Other low-pass filters are also possible, for instance the Butterworth filter

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{15}$$

The result is that, instead of the current angular acceleration, a filtered and therefore delayed angular acceleration $\dot{\boldsymbol{\Omega}}_f$ is measured. Because all the terms with the zero subscript in the Taylor expansion should be at the same point in time, they are all replaced with the subscript f, yielding Eq. (16). This indicates that these signals are also filtered and are therefore synchronous with the angular acceleration:

$$\begin{aligned} \dot{\boldsymbol{\Omega}} &= \dot{\boldsymbol{\Omega}}_f + \boldsymbol{G}_1\,\mathrm{diag}(\boldsymbol{\omega}_f)(\boldsymbol{\omega} - \boldsymbol{\omega}_f) + T_s\boldsymbol{G}_2(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_f) \\ &- \boldsymbol{C}(\boldsymbol{\Omega}_f)\boldsymbol{G}_3(\boldsymbol{\omega} - \boldsymbol{\omega}_f) \end{aligned} \tag{16}$$

This equation is not yet ready to be inverted because it contains the derivative of the angular rate of the propellers. Because we are dealing with discrete signals, consider the discrete approximation of the derivative in the z domain: $\dot{\boldsymbol{\omega}} = (\boldsymbol{\omega} - \boldsymbol{\omega}z^{-1})T_s^{-1}$, where T_s is the sample time. This is shown in Eq. (17):

$$\begin{aligned} \dot{\boldsymbol{\Omega}} &= \dot{\boldsymbol{\Omega}}_f + \boldsymbol{G}_1\,\mathrm{diag}(\boldsymbol{\omega}_f)(\boldsymbol{\omega} - \boldsymbol{\omega}_f) + \boldsymbol{G}_2(\boldsymbol{\omega} - \boldsymbol{\omega}z^{-1} - \boldsymbol{\omega}_f + \boldsymbol{\omega}_f z^{-1}) \\ &- \boldsymbol{C}(\boldsymbol{\Omega}_f)\boldsymbol{G}_3(\boldsymbol{\omega} - \boldsymbol{\omega}_f) \end{aligned} \tag{17}$$

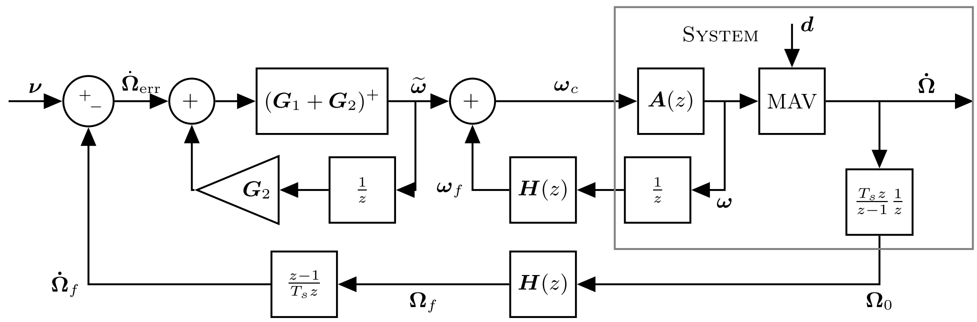

Fig. 2 INDI control scheme. $A(z)$ denotes the actuator dynamics, and $H(z)$ is the second-order filter.

Collecting all terms with $(\boldsymbol{\omega}-\boldsymbol{\omega}_f)$ yields Eq. (18):

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}}_f + (\boldsymbol{G}_1\,\mathrm{diag}(\boldsymbol{\omega}_f) + \boldsymbol{G}_2 - \boldsymbol{C}(\boldsymbol{\Omega}_f)\boldsymbol{G}_3)(\boldsymbol{\omega}-\boldsymbol{\omega}_f) - \boldsymbol{G}_2 z^{-1}(\boldsymbol{\omega}-\boldsymbol{\omega}_f) \tag{18}$$

Inversion of this equation for $\boldsymbol{\omega}$ yields Eq. (19), where $^+$ denotes the Moore–Penrose pseudoinverse:

$$\boldsymbol{\omega}_c = \boldsymbol{\omega}_f + (\boldsymbol{G}_1\,\mathrm{diag}(\boldsymbol{\omega}_f) + \boldsymbol{G}_2 - \boldsymbol{C}(\boldsymbol{\Omega}_f)\boldsymbol{G}_3)^+ (\boldsymbol{\nu} - \dot{\boldsymbol{\Omega}}_f + \boldsymbol{G}_2 z^{-1}(\boldsymbol{\omega}_c - \boldsymbol{\omega}_f)) \tag{19}$$

Note that the predicted angular acceleration $\dot{\boldsymbol{\Omega}}$ is now instead a virtual control, denoted by $\boldsymbol{\nu}$. The virtual control is the desired angular acceleration, and with Eq. (19), the required inputs $\boldsymbol{\omega}_c$ can be calculated. The subscript c is added to $\boldsymbol{\omega}$ to indicate that this is the command sent to the motors. This input is given with respect to a previous input $\boldsymbol{\omega}_f$. If we define the increment in the motor commands as $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}_c - \boldsymbol{\omega}_f$, it is clearly an incremental control law.

A. Parameter Estimation

Equation (19) shows the general quadrotor INDI control law. The parameters of this equation are the three matrices $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, and $\boldsymbol{G}_3$, which need to be identified for the specific quadrotor. This can be done through measurement of each of the components that make up these matrices, including the moments of inertia of the vehicle and the propellers as well as the thrust and drag coefficients of the rotors. Identifying the parameters in this way requires a significant amount of effort.

A more effective method is to use test flight data to determine the model coefficients. Of course, to do this, the MAV needs to be flying. This can be achieved by initially tuning the parameters. Alternatively, a different controller can be used at first to gather the test flight data, such as PID control. Once a test flight has been logged, Eq. (18) is used for parameter estimation and is written as Eq. (20). From this equation, a least-squares solution is found for the matrices $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, and $\boldsymbol{G}_3$

$$\Delta\dot{\boldsymbol{\Omega}}_f = [\boldsymbol{G}_1 \quad \boldsymbol{G}_2 \quad \boldsymbol{C}(\boldsymbol{\Omega}_f)\boldsymbol{G}_3]\begin{bmatrix} \mathrm{diag}(\boldsymbol{\omega}_f)\Delta\boldsymbol{\omega}_f \\ (\Delta\boldsymbol{\omega}_f - z^{-1}\Delta\boldsymbol{\omega}_f) \\ -\Delta\boldsymbol{\omega}_f \end{bmatrix} \tag{20}$$

Here, Δ denotes the finite difference between two subsequent samples. From the data, we can also investigate the importance of some of the terms by comparing the least-squares error with and without the terms. It turns out that, on a typical dataset, leaving out the matrix $\boldsymbol{G}_3$ only results in an estimation squared error increase of $\sim$0.2%. Furthermore, modeling the rotor as linear with the rotational speed of the rotor instead of quadratic gives an estimation squared error increase of $\sim$0.9%. Therefore, we can simplify the INDI control law of Eq. (19) into Eq. (21):

$$\boldsymbol{\omega}_c = \boldsymbol{\omega}_f + (\boldsymbol{G}_1 + \boldsymbol{G}_2)^+(\boldsymbol{\nu} - \dot{\boldsymbol{\Omega}}_f + \boldsymbol{G}_2 z^{-1}(\boldsymbol{\omega}_c - \boldsymbol{\omega}_f)) \tag{21}$$

B. Implementation

With the simplifications described in Sec. III.A, the final INDI control scheme is shown in Fig. 2. The input to the system is the virtual control $\boldsymbol{\nu}$, and the output is the angular acceleration of the system, $\dot{\boldsymbol{\Omega}}$. The angular velocity measurement from the gyroscope is fed back through the differentiating second-order filter and subtracted from the virtual control to give the angular acceleration error $\dot{\boldsymbol{\Omega}}_{\mathrm{err}}$.

Because the matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are not square, we take the pseudoinverse to solve the problem of control allocation, denoted by $^+$. The contents of the block "MAV" are shown in Fig. 3 because it allows the closed-loop analysis in Sec. III.C. In this diagram, $\boldsymbol{d}$ is a disturbance term that bundles disturbances and unmodeled dynamics.

Note that Eq. (21) provides a desired angular velocity of the rotors. However, the actuators do not have an instantaneous response. Instead, it is assumed they have first-order dynamics $\boldsymbol{A}(z)$. The reference sent to the motors is denoted by $\boldsymbol{\omega}_c$ and $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}_c - \boldsymbol{\omega}_f$. In Fig. 2, it is assumed that actuator feedback is available. However, if this is not the case, the actuator state $\boldsymbol{\omega}_0$ has to be estimated with a model of the actuator dynamics as is shown in Fig. 4. Here, $\boldsymbol{A}'(z)$ is a model of the actuator dynamics.

C. Closed-Loop Analysis

Consider the control diagram shown in Fig. 2. We can verify that this is a stable controller by doing a closed-loop analysis. First, the transfer function of each of the two small loops is calculated, shown by Eqs. (22) and (23). Here, $\boldsymbol{TF}_{x\to y}$ denotes the transfer function from point x to y in the control diagram:

$$\begin{aligned} \tilde{\boldsymbol{\omega}} &= (\boldsymbol{G}_1+\boldsymbol{G}_2)^+\dot{\boldsymbol{\Omega}}_{\mathrm{err}} + (\boldsymbol{G}_1+\boldsymbol{G}_2)^+\boldsymbol{G}_2 z^{-1}\tilde{\boldsymbol{\omega}} \\ (\boldsymbol{G}_1+\boldsymbol{G}_2)\tilde{\boldsymbol{\omega}} &= \dot{\boldsymbol{\Omega}}_{\mathrm{err}} + \boldsymbol{G}_2 z^{-1}\tilde{\boldsymbol{\omega}} \\ (\boldsymbol{G}_1+\boldsymbol{G}_2-\boldsymbol{G}_2 z^{-1})\tilde{\boldsymbol{\omega}} &= \dot{\boldsymbol{\Omega}}_{\mathrm{err}} \\ \boldsymbol{TF}_{\dot{\Omega}_{\mathrm{err}}\to\tilde{\omega}}(z) &= (\boldsymbol{G}_1+\boldsymbol{G}_2-\boldsymbol{G}_2 z^{-1})^+ \end{aligned} \tag{22}$$

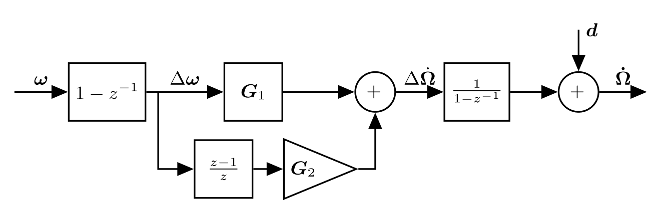

Fig. 3 Contents of the block named "MAV" in Fig. 2.

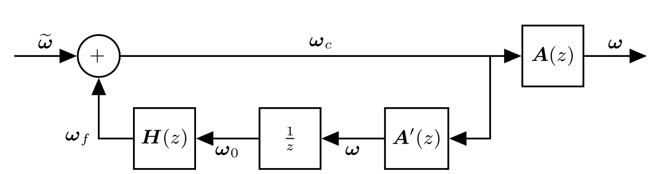

Fig. 4 Block diagram for estimation of actuator state if actuator feedback is not available.

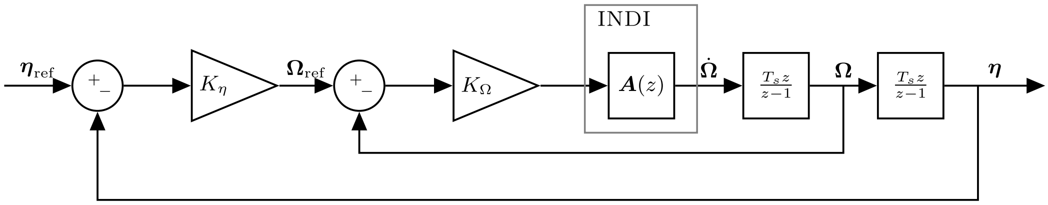

Fig. 5 Design of the attitude controller based on the closed-loop response of the INDI controller.

We define $\boldsymbol{H}(z) = \boldsymbol{I}H(z)$ and assume that all actuators have the same dynamics, and so $\boldsymbol{A}(z) = \boldsymbol{I}A(z)$. This means that each matrix in $\boldsymbol{TF}_{\tilde{\omega}\to\omega}(z)$ is a diagonal matrix, and therefore $\boldsymbol{TF}_{\tilde{\omega}\to\omega}(z)$ is a diagonal matrix function:

$$\begin{aligned}\boldsymbol{TF}_{\tilde{\omega}\to\omega}(z) &= (\boldsymbol{I} - \boldsymbol{A}(z)\boldsymbol{H}(z)z^{-1})^{-1}\boldsymbol{A}(z)\\ &= (\boldsymbol{I} - \boldsymbol{I}A(z)\boldsymbol{I}H(z)z^{-1})^{-1}\boldsymbol{I}A(z)\\ &= (\boldsymbol{I}(1 - A(z)H(z)z^{-1}))^{-1}\boldsymbol{I}A(z)\\ &= \boldsymbol{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\end{aligned} \tag{23}$$

Then, the last part of the open loop is from $\boldsymbol{\omega}$ to $\dot{\boldsymbol{\Omega}}$, as shown by Fig. 3. Using this figure, the transfer function is calculated in Eq. (24). Note that, for this analysis, disturbances are not taken into account:

$$\boldsymbol{TF}_{\omega\to\dot{\Omega}}(z) = \boldsymbol{G}_1 + \frac{z-1}{z}\boldsymbol{G}_2 = \boldsymbol{G}_1 + \boldsymbol{G}_2 - \boldsymbol{G}_2 z^{-1} \tag{24}$$

Using these intermediate results, the open-loop transfer function of the entire system is shown in Eq. (25):

$$\begin{aligned}\boldsymbol{TF}_{\dot{\Omega}_{\mathrm{err}}\to\dot{\Omega}}(z) &= \boldsymbol{TF}_{\omega\to\dot{\Omega}}(z)\boldsymbol{TF}_{\tilde{\omega}\to\omega}(z)\boldsymbol{TF}_{\dot{\Omega}_{\mathrm{err}}\to\tilde{\omega}}(z)\\ &= (\boldsymbol{G}_1 + \boldsymbol{G}_2 - \boldsymbol{G}_2 z^{-1})\boldsymbol{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\\ &\quad (\boldsymbol{G}_1 + \boldsymbol{G}_2 - \boldsymbol{G}_2 z^{-1})^{+}\\ &= \boldsymbol{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\end{aligned} \tag{25}$$

Using Eq. (25) and Fig. 2, we can calculate the closed-loop transfer function of the entire system in Eq. (26):

$$\begin{aligned}\boldsymbol{TF}_{\nu\to\dot{\Omega}}(z) &= (\boldsymbol{I} + \boldsymbol{TF}_{\dot{\Omega}_{\mathrm{err}}\to\dot{\Omega}}(z)\boldsymbol{I}H(z)z^{-1})^{-1}\boldsymbol{TF}_{\dot{\Omega}_{\mathrm{err}}\to\dot{\Omega}}(z)\\ &= (\boldsymbol{I} + \boldsymbol{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\boldsymbol{I}H(z)z^{-1})^{-1}\\ &\quad \boldsymbol{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\\ &= \boldsymbol{I}\frac{(1 - A(z)H(z)z^{-1})^{-1}A(z)}{1 + (1 - A(z)H(z)z^{-1})^{-1}A(z)H(z)z^{-1}}\\ &= \boldsymbol{I}\frac{A(z)}{1 - A(z)H(z)z^{-1} + A(z)H(z)z^{-1}}\\ &= \boldsymbol{I}A(z)\end{aligned} \tag{26}$$

From this equation, it appears that the closed-loop transfer function from the virtual input to the angular acceleration is, in fact, the actuator dynamics $A(z)$. In most cases, the actuator dynamics can be represented by first- or second-order dynamics. Note that this shows the importance of applying the $H(z)$ filter on the input as well. By doing this, a lot of terms cancel, and all that remains is the actuator dynamics.

Now, consider the transfer function from disturbances $\boldsymbol{d}$ (see Fig. 2) to the angular acceleration. The derivation is given in Eq. (27) in which use is made of Eq. (25):

$$\begin{aligned}\boldsymbol{TF}_{d\to\dot{\Omega}}(z) &= (\boldsymbol{I} - \boldsymbol{TF}_{\dot{\Omega}_{\mathrm{err}}\to\dot{\Omega}}(z)(-1)\boldsymbol{H}(z)z^{-1})^{-1}\boldsymbol{I}\\ &= (\boldsymbol{I} + \boldsymbol{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\boldsymbol{I}H(z)z^{-1})^{-1}\boldsymbol{I}\\ &= \boldsymbol{I}\frac{1}{1 + (1 - A(z)H(z)z^{-1})^{-1}A(z)H(z)z^{-1}}\\ &= \boldsymbol{I}\frac{1 - A(z)H(z)z^{-1}}{1 - A(z)H(z)z^{-1} + A(z)H(z)z^{-1}}\\ &= \boldsymbol{I}(1 - A(z)H(z)z^{-1})\end{aligned} \tag{27}$$

With Eq. (27), we show that disturbances in the angular acceleration are rejected as long as the actuator dynamics and the designed filter are stable. The term $A(z)H(z)z^{-1}$ will go to 1 over time, with a response determined by the actuator dynamics, filter dynamics, and a unit delay. This means that if the angular acceleration is measured faster, the drone can respond to disturbances faster. Moreover, if the actuators can react faster, disturbances can be neutralized faster.

D. Attitude Control

The angular acceleration of the MAV is accurately controlled by the system shown in Fig. 2. To control the attitude of the MAV, a stabilizing angular acceleration reference needs to be passed to the INDI controller. This outer-loop controller can be as simple as a proportional derivative (PD) controller (a gain on the rate error and a gain on the angle error), as shown in Fig. 5. Here, $\boldsymbol{\eta}$ represents the attitude of the quadcopter. The benefit of the INDI inner-loop controller is that the outer PD controller commands a reference, independent of the effectiveness of the actuators (including the inertia of the quadrotor).

This means that the design of this controller depends only on the speed of the actuator dynamics $\boldsymbol{A}(z)$. In case the actuator dynamics are known (through analysis of logged test flights, for instance), values of K_η and K_Ω can be determined that give a stable response.

This outer-loop controller does not involve inversion of the attitude kinematics, as has been done in other work [3]. However, the attitude angles for a quadrotor are generally small, in which case the inversion of the attitude kinematics can be replaced with simple angle feedback.

E. Altitude Control

The INDI controller derived in the beginning of this section controls the angular acceleration around the axes x, y, and z, which correspond to roll, pitch, and yaw. However, there is a fourth degree of freedom that is controlled with the rotors, which is the acceleration along the z axis.

Control of this fourth axis is handled by a separate controller. This controller scales the average input to the motors to a value commanded by the pilot, after the input has been incremented by the INDI controller.

IV. Adaptive Incremental Nonlinear Dynamic Inversion

The INDI approach only relies on modeling of the actuators. The control effectiveness depends on the moment of inertia of the vehicle as well as the type of motors and propellers. A change in any of these will require re-estimation of the control effectiveness. Moreover, the control effectiveness can even change during flight, due to a change in flight velocity, battery voltage, or actuator failure.

To counteract these problems and obtain a controller that requires no manual parameter estimation, the controller was extended with onboard adaptive parameter estimation using a least mean squares (LMS) [15] adaptive filter. This filter is often used in adaptive signal filtering and adaptive neural networks.

The LMS implementation is shown in Eq. (28), where $\boldsymbol{\mu}_1$ is a diagonal matrix whose elements are the adaptation constant for each input, and $\boldsymbol{\mu}_2$ is a diagonal matrix to adjust the adaptation constants per axis. This is necessary because not all axes have the same signal-to-noise ratio.

The LMS formula calculates the difference between the expected acceleration based on the inputs and the measured acceleration. Then, it increments the control effectiveness based on the error. The control effectiveness includes both $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, as is shown in Eq. (29):

$$\boldsymbol{G}(k) = \boldsymbol{G}(k-1) - \boldsymbol{\mu}_2\left(\boldsymbol{G}(k-1)\begin{bmatrix}\Delta\boldsymbol{\omega}_f \\ \Delta\dot{\boldsymbol{\omega}}_f\end{bmatrix} - \Delta\dot{\Omega}_f\right)\begin{bmatrix}\Delta\boldsymbol{\omega}_f \\ \Delta\dot{\boldsymbol{\omega}}_f\end{bmatrix}^T \boldsymbol{\mu}_1 \tag{28}$$

$$\boldsymbol{G} = [\boldsymbol{G}_1 \quad \boldsymbol{G}_2] \tag{29}$$

Clearly, when there is no change in input, the control effectiveness is not changed. The reverse is also true; more excitation of the system will result in a faster adaptation. This is a benefit of the LMS algorithm over, for instance, recursive least squares with a finite horizon because recursive least squares will “forget” everything outside the horizon. Note that the filtering for the online parameter estimation can be different from the filtering for the actual control. Equation (28) makes use of $\Delta\dot{\boldsymbol{\Omega}}_f$, which is the finite difference of $\dot{\boldsymbol{\Omega}}_f$ in the control Eq. (21). Because differentiating amplifies high frequencies, a filter that provides more attenuation of these high frequencies is necessary. We still use the second-order filter described by Eq. (15), but with $\omega_n = 25$ rad/s and $\zeta = 0.55$.

When an approximate control effectiveness is given before takeoff, the adaptive system will estimate the actual values online and thereby tune itself. The only knowledge provided to the controller is an initial guess of the control effectiveness. It is generally not possible to take off without any estimate of the control effectiveness because the UAV might crash before the adaptive system has converged.

The choice of the adaptation constants $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ determines the stability and the rate of adaptation. By making these constants larger, a faster convergence is achieved. By making them too large, the adaptation will no longer be stable. The theoretical limit has been discussed in the literature [15], and it depends on the autocorrelation matrix of the input to the filter. In practice, the filter stability deteriorates before the theoretical limit, and so to find a good adaptation constant, some tuning is required.

V. Experimental Setup

To validate the performance of the INDI controller developed in Sec. III and the adaptive parameter estimation from Sec. IV, several experiments were conducted. These experiments were performed using the Bebop quadcopter from Parrot shown in Fig. 1. The Bebop weighs 396.2 g and can be equipped with bumpers, which are 12 g per bumper. For these experiments, the bumpers were not equipped unless explicitly stated. The quadcopter was running the Paparazzi open-source autopilot software, which contains all the code for wireless communication, reading sensor measurements, etc. The accelerometer, gyroscope, and control loops were running at 512 Hz.

Four experiments test the key properties of the controller: 1) performance, 2) disturbance rejection, and 3) adaptation.

During these experiments, the reference attitude and average thrust level were controlled by a pilot and sent to the drone over Wi-Fi. All other computations were done on the drone itself, including the online adaptation.

A. Performance

To put the responsiveness of the system to the test and to make sure that the angular acceleration reference is tracked by the INDI controller, a doublet input was applied on the attitude roll angle. The amplitude of the doublet is 30 deg, and the period is half a second (0.25 s positive and 0.25 s negative). This test is only done for the roll and not for the pitch because there is no fundamental difference between these axes. The yaw axis is covered separately in Sec. V.D. Note that this experiment is performed without the adaptation.

The performance is compared to a manually tuned PID controller. The INDI controller is not expected to be faster or slower than a traditional PID controller because the result of Eq. (26) shows that the response of the INDI inner loop is simply the actuator dynamics. Considering that the outer loop is a PD controller, the rise time and overshoot should be similar.

Finally, this test will also be performed with an INDI controller that does not contain the filter delay compensation, more specifically by using $\boldsymbol{\omega}_0$ in the controller increment instead of $\boldsymbol{\omega}_f$. It is expected that this will not fly well, because in Sec. III.C, we showed that with this compensation all terms cancel, and the closed-loop transfer function reduces to $\boldsymbol{I}A(z)$.

By inspection of Fig. 2, we can get a feel for what will happen if we omit this filter compensation. When there is an angular acceleration error, a control increment $\tilde{\boldsymbol{\omega}}$ will be the result, which is added to $\boldsymbol{\omega}_0$ to produce $\boldsymbol{\omega}_c$. $\boldsymbol{\omega}_c$ goes through the actuator dynamics to produce the new $\boldsymbol{\omega}$. The next time step, the result of this new $\boldsymbol{\omega}$, does not yet appear in $\dot{\Omega}_f$, because it is filtered and therefore delayed. Therefore, $\tilde{\boldsymbol{\omega}}$ will be the same. However, $\boldsymbol{\omega}_0$ did update, and so $\boldsymbol{\omega}_c$ will be incremented even more, while we are still waiting to see the result of the first increment in $\dot{\boldsymbol{\Omega}}_f$.

B. Disturbance Rejection

The disturbance rejection property is validated by adding a disturbance to the system. One possibility would be to apply aerodynamic disturbances by flying in the wake of a big fan. The disturbances occurring would be realistic but not very repeatable. Moreover, the magnitude of the disturbance would be unknown.

Instead, it is possible to apply a disturbance in the form of a step function to the system. This is done by adding a weight of 42.5 g to a container located in an off-centered position on the quadrotor while it is flying, as shown in Fig. 6. The container is located on the front of the drone and has a distance of about 11 cm to the center of gravity, and so any weight added will shift the center of gravity forward. This will cause a misalignment of the thrust vector with respect to the center of gravity and therefore a pitch moment. This moment will be persistent and therefore have the form of a step disturbance. This is indicated with $\boldsymbol{d}$ in Fig. 2. Although this moment is created with a center of gravity shift, the situation is the same as in the case of a persistent gust or an unmodeled aerodynamic moment.

A normal PID controller would respond to such a disturbance very slowly because it takes time for the integrator to accumulate. But the introduction of the INDI inner loop leads to a cascaded control structure, which is much more resistant to disturbances than a single-

Fig. 6 Container attached to the nose of the quadrotor with one weight inside.

Fig. 7 Bebop quadrotor with bumpers.

loop design [16]. Because of this, the reference pitch angle is expected to be tracked shortly after the disturbance.

C. Adaptation

The Bebop quadcopter has the possibility to fly with bumpers, as is shown in Fig. 7. Though these bumpers only weigh 12 g apiece, they are located far from the center of gravity and therefore increase the moment of inertia. Furthermore, they can influence the airflow around the propellers. These system changes affect the $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ matrices. Therefore, the adaptive algorithm from Sec. IV should deal with adding or removing the bumpers.

First, two flights are performed to show the effect of adding or removing the bumpers when the adaptive algorithm is not active. For the first flight, the bumpers are added, whereas the $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ matrices correspond to the quadrotor without bumpers. For the second flight, the bumpers are removed, and the $\boldsymbol{G}$ matrices from the quadrotor with bumpers are used. In both flights, doublets are performed like in Sec. V.A. The performance is expected to degrade compared to the previous results for both cases because the $\boldsymbol{G}$ matrices do not correspond to what they should be.

Second, the ability of the quadrotor to adapt its $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ matrices is tested. In this experiment, the drone starts with bumpers equipped, but with system matrices that represent the configuration without bumpers. The pilot flies the drone in a confined area while performing some pitch, roll, and yaw maneuvers to excite the system. While flying, the correct matrices should be estimated. Then, the Bebop is landed, and the bumpers are removed. After takeoff, the matrices should converge to their original state.

Finally, doublets are performed with and without the bumpers equipped while the adaptation algorithm is active. We expect the same performance as in Sec. V.A.

D. Yaw Control

The purpose of this experiment is to show the improvement in yaw performance due to the incorporation of the rotor spin-up torque in the controller design. This is done by applying a doublet input on the yaw set point. The amplitude of the doublet is 5 deg, and the period is 1 s (0.5 s positive and 0.5 s negative). As a comparison, the same experiment is performed with a traditional PID controller. This PID controller is manually tuned to give a fast rise time with minimal overshoot.

Additionally, the same test is performed with a zero $\boldsymbol{G}_2$ matrix. Here, we expect an oscillation because the persistent effect of a change in rotor angular velocity on the yaw axis is small. We take the pseudoinverse in Eq. (21), and so the resulting gain will be very large. Because there is the angular momentum effect of the propellers, the initial angular acceleration will be larger than expected, and the controller will start to oscillate.

VI. Results

This section deals with the results of the experiments described in Sec. V. The angular acceleration shown in the plots in this section is not the onboard estimate of the angular acceleration because it is delayed through filtering. Instead, it is computed after the experiment from the finite difference of the gyroscope data. The signal is filtered with a fourth-order Butterworth filter with a cutoff frequency of 15 Hz. It is filtered twice (forward and reverse), resulting in a zero-phase (noncausal) filter. For the actual control, the onboard filtered (and delayed) angular acceleration was used.

A. Performance

Figure 8 shows the angular acceleration around the x axis denoted by $\dot{p}$ and the reference angular acceleration denoted by $\dot{p}_{\mathrm{ref}}$. Additionally, the reference is filtered with the actuator dynamics, resulting in $\dot{p}_{\mathrm{ref}_A}$. This signal is the angular acceleration that is expected based on the calculations in Sec. III.C, specifically Eq. (26). It might seem that the controller does not track the reference well because it lags behind the reference, but this was expected based on the model of the actuator dynamics. The angular acceleration is actually very close to the expected angular acceleration $\dot{p}_{\mathrm{ref}_A}$. Finally, we also show the angular acceleration as calculated onboard the quadrotor using the second-order filter. The filtered angular acceleration onboard the quadrotor is significantly delayed with respect to the actual angular acceleration, which is why we will run into problems if we do not take this delay into account in the INDI controller.

The outer-loop controller, which generates the angular acceleration reference to track, was designed such that the resultant accelerations give a desired response of the roll angle, shown in Fig. 9. From this figure, it can be seen that the quadcopter reaches its reference roll angle within 0.2 s with a very small overshoot.

The roll angle response of the PID controller is shown in Fig. 10. As expected, the PID controller performs very similar to the INDI controller in terms of rise time and overshoot. The integral gain included in the PID controller, which needs to eliminate steady-state offsets, degrades the dynamic performance of the closed-loop system. This shows that the INDI controller marginally improves the

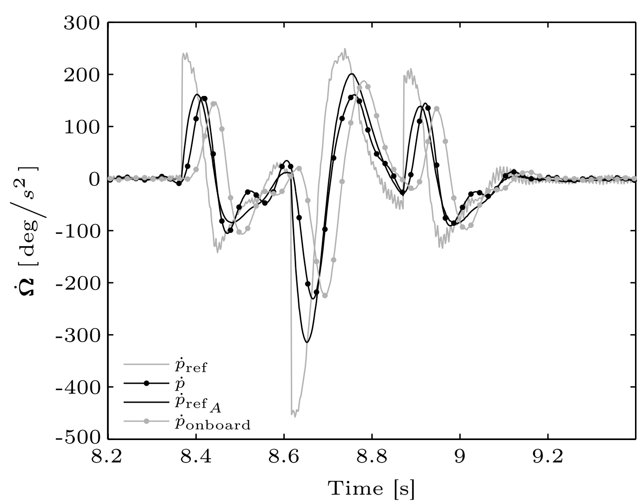

Fig. 8 Angular acceleration in the roll axis during doublet input.

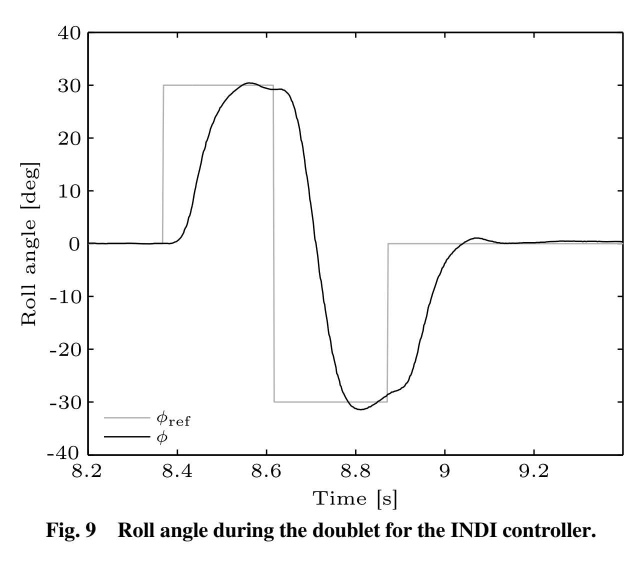

Fig. 9 Roll angle during the doublet for the INDI controller.

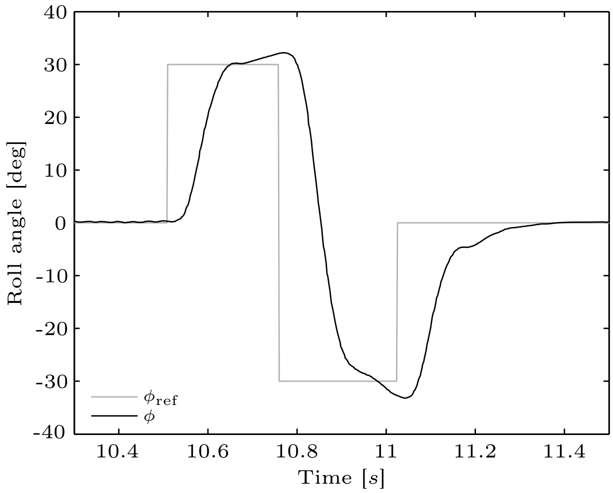

Fig. 10 Roll angle during the doublet for the PID controller.

performance of a traditional PID controller in terms of responsiveness for the roll.

As discussed previously, the onboard filtered measurement of the angular acceleration is significantly delayed. If we remove the filter delay compensation from the INDI controller, the quadrotor was severely oscillating, as can be seen in Fig. 11. The doublet was not performed because this did not seem safe. The oscillation might be reduced by lowering K_η and K_Ω, but this will make the response slower as well. From this figure, we can conclude that the filter delay compensation is an important part of the INDI controller and is crucial in obtaining good performance with an INDI controller.

B. Disturbance Rejection

The weight, shown in Fig. 6, was placed in the container attached to the nose of the quadrotor by hand. The weight was placed in the container gently, but it probably arrived in the container with some small velocity. The disturbance in the angular acceleration is therefore a combination of a step and a delta pulse.

Figure 12 shows the angular acceleration that is the result of the disturbance. From the figure, it is clear that the disturbance happened just after 13 s. As the angular acceleration increases in the negative direction, the reference angular acceleration starts to go the opposite way, because now an angular rate and a pitch angle error start to arise. About 0.1 s after losing track of the reference, the angular acceleration again coincides with the expected angular acceleration, having overcome the disturbance in the angular acceleration.

This results in a pitch angle with no steady-state error, as can be seen from Fig. 13. After 0.3 s, the pitch angle is back at zero. To show that the weight in the container really is a step disturbance, which can be compared to a constant aerodynamic moment, consider Fig. 14. It shows the difference of the rotational rate of the front and rear motors divided by 4: $(\omega_1 + \omega_2 - \omega_3 - \omega_4)/4$. This indicates the average magnitude in rounds per minute that each motor contributes to the pitch control; see Eq. (7). Clearly, there is a difference before and after the disturbance, which can be quantified as an average change of 578 rounds per minute over the interval [12.6 13.0] versus [13.4 13.8].

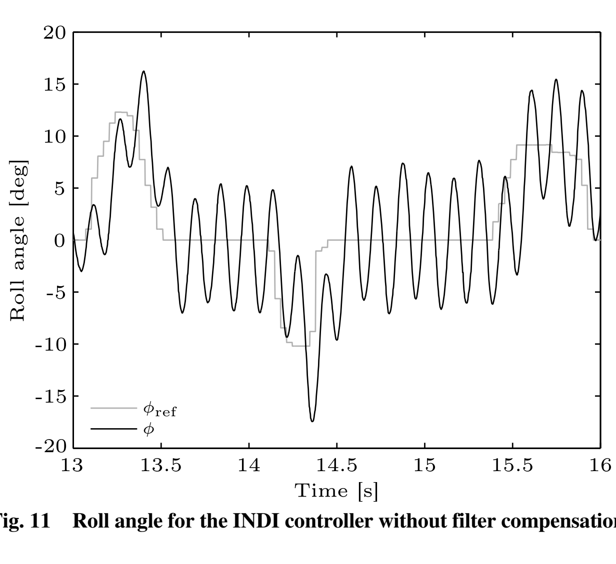

Fig. 11 Roll angle for the INDI controller without filter compensation.

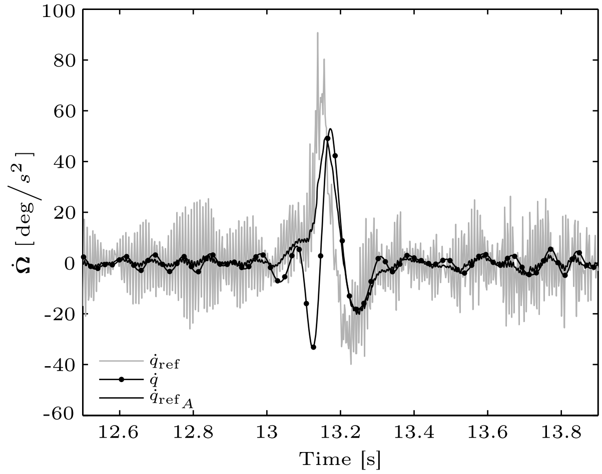

Fig. 12 Angular acceleration during the disturbance.

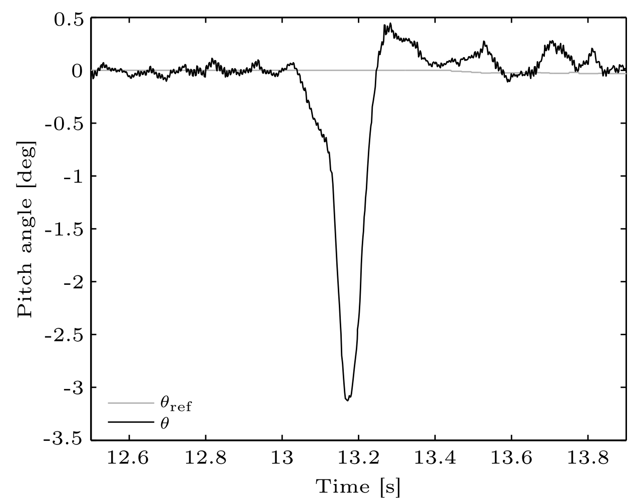

Fig. 13 Pitch angle during the disturbance.

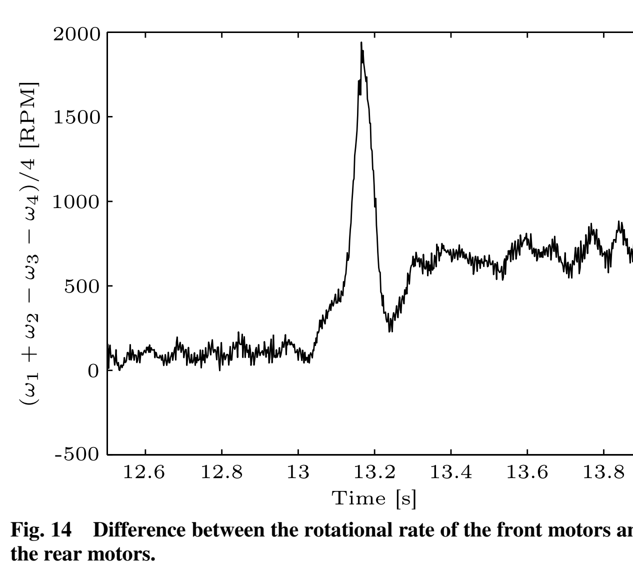

Fig. 14 Difference between the rotational rate of the front motors and the rear motors.

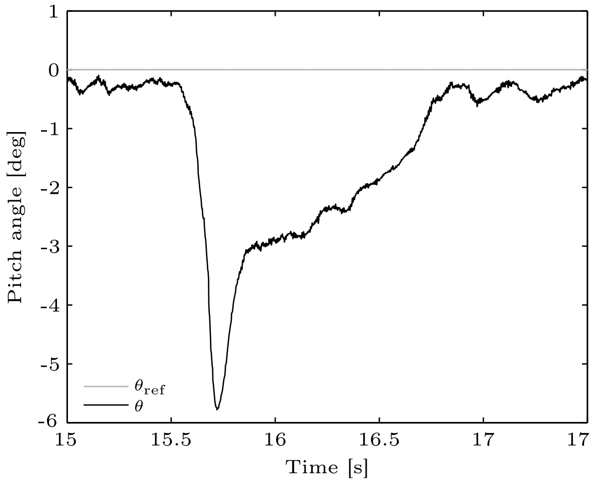

Fig. 15 Pitch angle during the disturbance for the PID controller.

This demonstrates that the disturbance was really a step and that the INDI controller can rapidly cope with such a disturbance.

Figure 15 shows the same experiment performed with a PID controller. Of course, the weight was not dropped in exactly the same manner and with the same velocity, and so the initial disturbance was probably different. However, the persisting disturbance is the same because the weight has exactly the same mass. It takes about 1.5 s before the pitch angle is back at zero again, which is approximately five times longer than for the INDI controller. One might say that the integral gain of the PID controller should be larger, but this will deteriorate the performance in the previous experiment.

C. Adaptation

Figures 16 and 17 show the response to a roll doublet without adaptation if there is a mismatch in the control effectiveness. Even though the bumpers are lightweight, their effect is significant because they are located far from the center of gravity. In Fig. 16, we see what happens if the actuators are less effective than in the model because the inertia is higher. Additional increments of the input are needed to reach a desired angular acceleration. The oscillation occurs because this takes more time. The oscillation can be reduced by reducing the K_η and K_ω gains, at the cost of having a slower response.

In Fig. 17, we see the opposite; the control effectiveness is higher than what was modeled. This results in a fast oscillation, which cannot be removed by reducing the attitude gains. This is because the cause of the oscillation is different; now, too much input is applied to reach a certain angular acceleration. This will happen regardless of what angular acceleration is requested by the attitude controller.

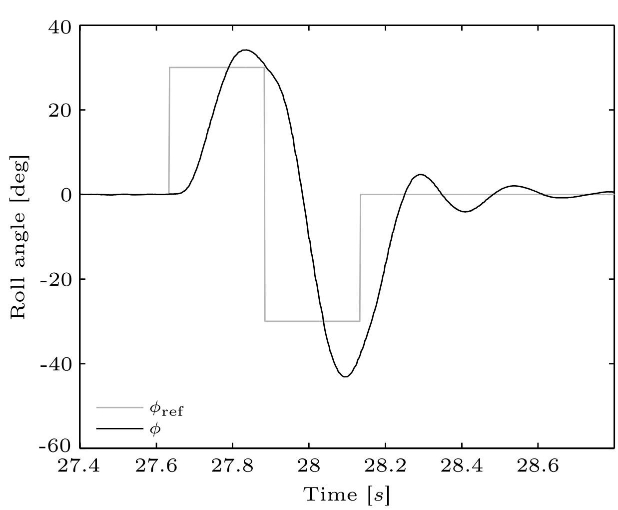

Fig. 16 Flight without adaptation, with bumpers equipped, while the control effectiveness has been determined without bumpers.

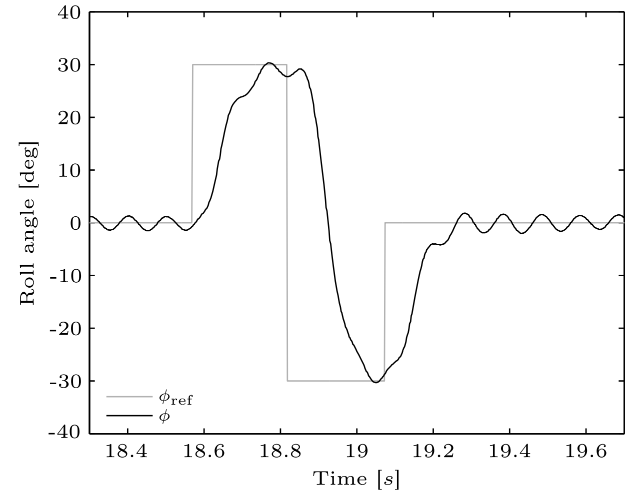

Fig. 17 Flight without adaptation, without bumpers equipped, while the control effectiveness has been determined with bumpers.

We can conclude that the performance degrades when the modeled control effectiveness does not closely correspond to the actual control effectiveness. When the adaptation algorithm is enabled, Figs. 18–20 show how each row of the $\boldsymbol{G}_1$ matrix evolves over time as a result of the second experiment described in Sec. V.C. The same is shown in Fig. 21 for the third row of the $\boldsymbol{G}_2$ matrix. Each line represents one of

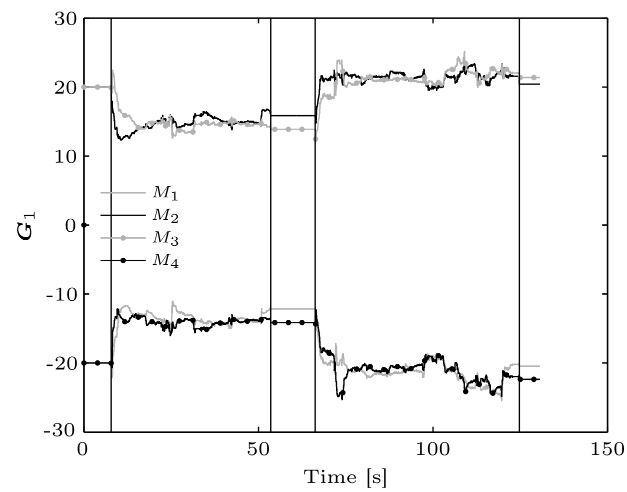

Fig. 18 First row of the G_1 matrix corresponding to the roll.

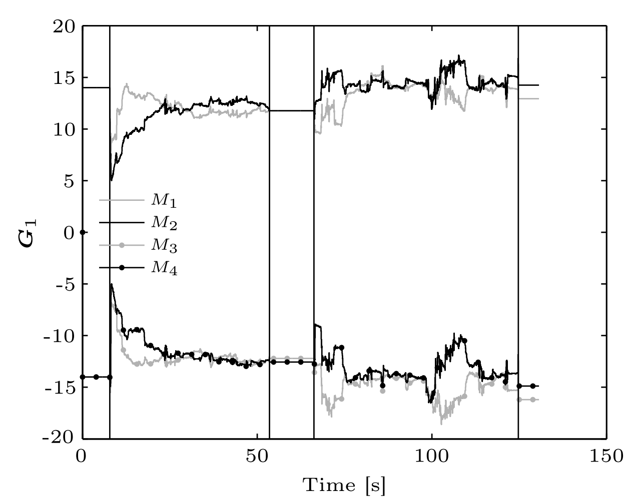

Fig. 19 Second row of the G_1 matrix corresponding to the pitch.

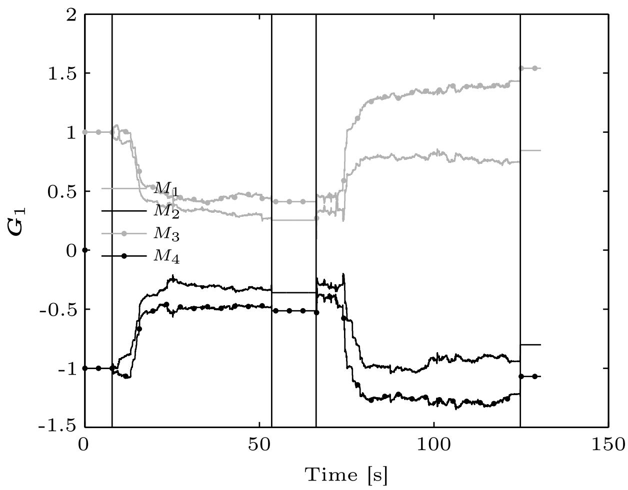

Fig. 20 Third row of the G_1 matrix corresponding to the yaw.

the elements of that row, indicating the effectiveness of that motor on the specified axis.

Note that the drone is flying in the interval of [8 54 s] and again in [66 125 s]; in between these times, the drone is landed and the bumpers are removed. This is indicated by vertical lines in the figures. A large change in effectiveness due to the addition and removal of the bumpers can be seen in the third row of the $\boldsymbol{G}_1$ matrix, shown in Fig. 20, which corresponds to the yaw.

Also in Fig. 18, a change in effectiveness can be seen between the flights with and without bumpers. Once converged, the effectiveness values are stable with little noise. Upon takeoff and landing, the effectiveness seems to diverge for a short period of time. This is not a failure of the adaptation algorithm but merely the result of the interaction with the floor.

The controller is engaged once the pilot gives a thrust command that exceeds idle thrust. At that point, the quadrotor does not produce enough lift to take off, and so it is still standing on the floor. When the INDI controller tries to attain certain angular accelerations, the quadrotor does not rotate, and the adaptation algorithm will adapt to this. When landing, these interactions with the floor can also occur.

Notice the large difference in effectiveness between the actuators in the second part of the flight in Fig. 20. This illustrates the added value of adaptive INDI because often the actuators are assumed to perform equal to each other, whereas in this case, they do not. These differences between the actuators are also observed with the estimation method described in Sec. III.A for multiple flights. The differences may be caused by small imperfections that are not clearly visible on some of the rotors.

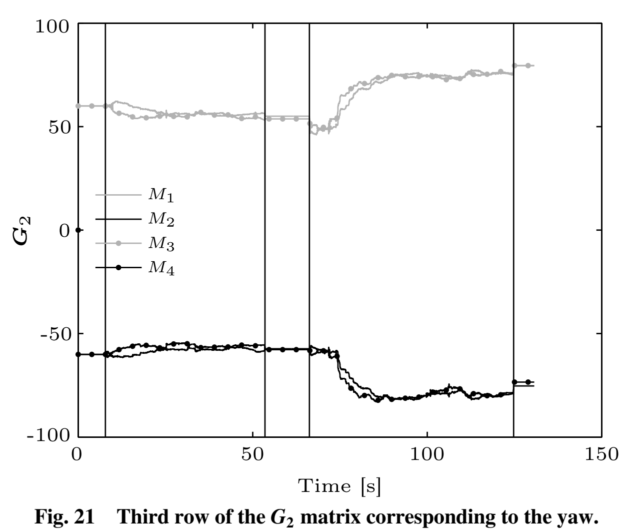

Fig. 21 Third row of the G_2 matrix corresponding to the yaw.

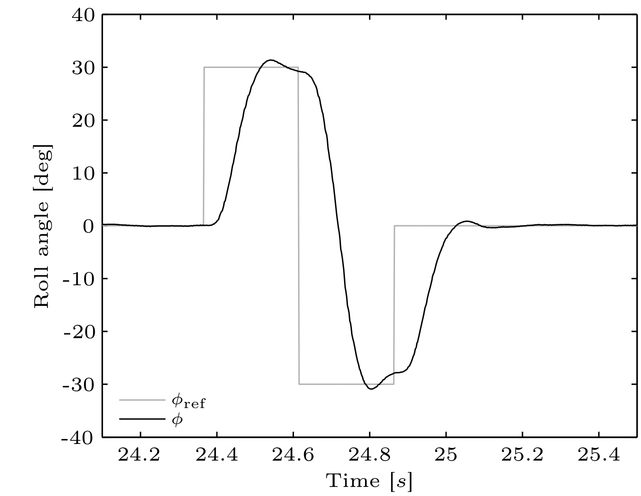

Fig. 22 Flight with adaptation, with bumpers equipped, while the control effectiveness has been determined without bumpers.

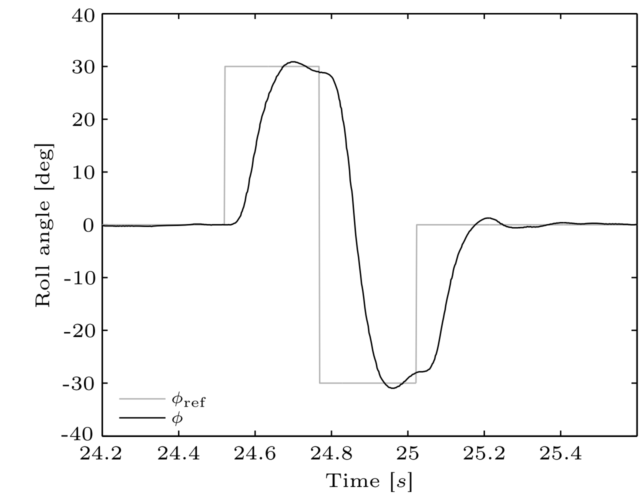

Fig. 23 Flight with adaptation, without bumpers equipped, while the control effectiveness has been determined with bumpers.

Finally, we can observe how the online parameter estimation affects the response to a roll doublet in Figs. 22 and 23. Regardless of whether the bumpers are equipped or not, or with what control effectiveness model the quadrotor starts flying, the same performance is achieved as in Sec. V.A. This shows the robustness of the adaptive algorithm against control effectiveness changes.

D. Yaw Control

Finally, consider Fig. 24. It shows for each time step the change in angular acceleration in the yaw axis, $\Delta\dot{r}$, during the large control inputs discussed previously. A careful reader up until this point may wonder: "Is the rotor spin-up torque really significant? Can we not omit the $\boldsymbol{G}_2$ matrix?" The figure shows the predicted change in angular acceleration based on the change in motor speeds according to Eq. (21), which is a close match. Additionally, the figure also shows the predicted change in angular acceleration if we neglect $\boldsymbol{G}_2$, denoted by $\Delta\dot{r}_{\mathrm{simple}}$. Clearly, the motor spin-up torque is very significant.

Moreover, if we try to fly with a zero $\boldsymbol{G}_2$ matrix, the resulting oscillation is so strong that a takeoff is not possible. To fly without this matrix, we cannot use the estimated values for the control effectiveness in the yaw axis. Instead, we can take a higher effectiveness for the model parameters than in reality to avoid overshooting the reference angular acceleration due to the rotor spin-up torque that is now not taken into account. Figure 25 shows that it is possible to fly with a zero $\boldsymbol{G}_2$ matrix, at the cost of a severe performance penalty.

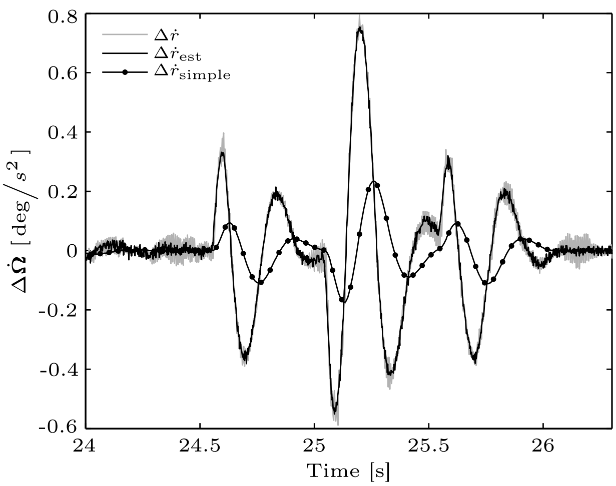

Fig. 24 Change in angular acceleration in the yaw axis along with the predicted change.

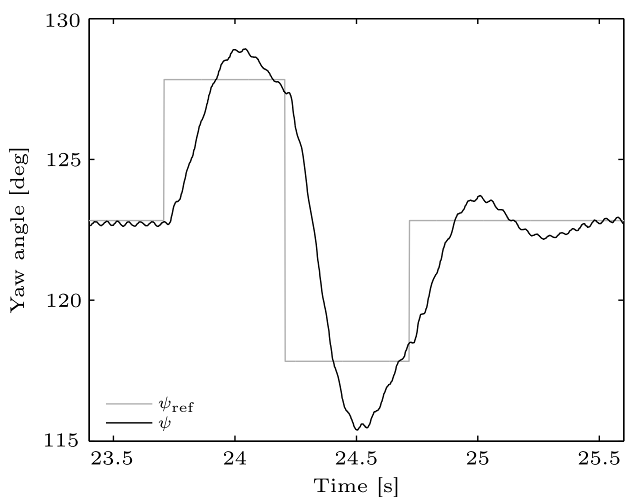

Fig. 25 Yaw angle during the doublet for the INDI controller without G_2 matrix.

If we do take the rotor angular momentum into account, Fig. 26 shows the resultant doublet response of the yaw angle. Compare this with Fig. 27, which shows the doublet response for the PID controller. The INDI controller clearly has a faster rise time and less overshoot.

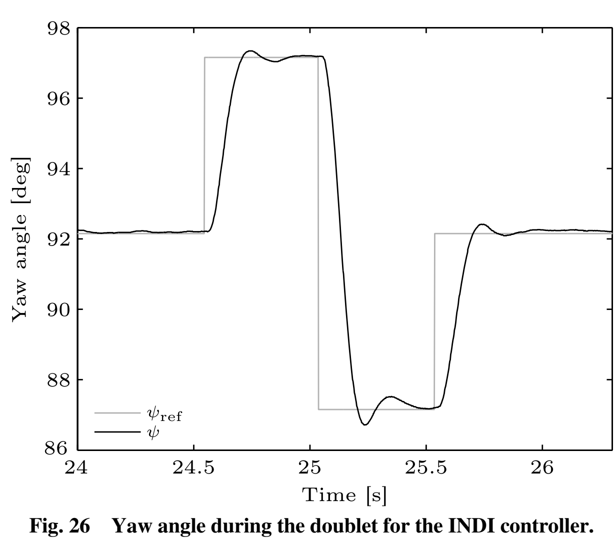

Fig. 26 Yaw angle during the doublet for the INDI controller.

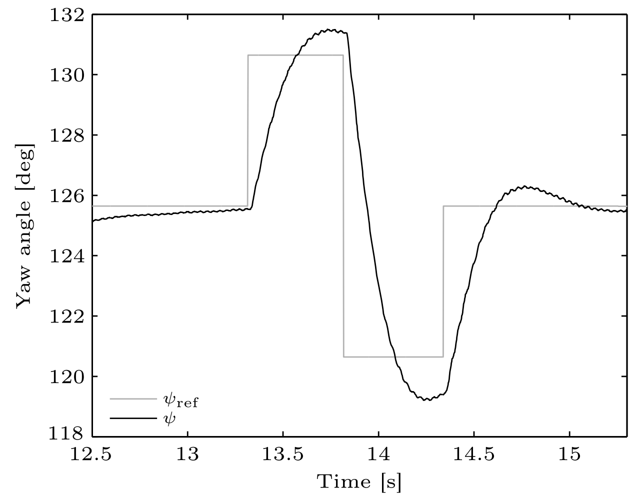

Fig. 27 Yaw angle during the doublet for the PID controller.

VII. Conclusions

Adaptive incremental nonlinear dynamic inversion is a very promising technique for control of micro air vehicles (MAVs). Because of incorporation of the spin-up torque, fast yaw control is possible, which is typically very slow on a quadrotor. The disturbance rejection capabilities are vital when flying in windy conditions or with MAVs that have complex aerodynamics. Because unmodeled aerodynamic moments are measured with the angular acceleration, no complex aerodynamic modeling is needed. Even the control effectiveness matrices are shown to be adapted online, resulting in a controller that can handle changes in the MAV configuration and needs little effort to set up on a new platform. Only when a high-performance outer loop is required is some knowledge of the actuator dynamics needed. These properties result in a very flexible and powerful controller.

Acknowledgments

This work was financed by the Delphi Consortium. The authors would like to thank Bart Remes and the MAVLab for their support.

References

[1] Mahony, R., Kumar, V., and Corke, P., "Multirotor Aerial Vehicles: Modeling, Estimation and Control of Quadrotor," *IEEE Robotics and Automation Magazine*, Vol. 38, No. 12, 2012, pp. 20–32. doi:10.1109/MRA.2012.2206474

[2] Fresk, E., and Nikolakopoulos, G., "Full Quaternion Based Attitude Control for a Quadrotor," *Proceedings of the European Control Conference*, IEEE Publ., Piscataway, NJ, 2013, pp. 3864–3869.

[3] da Costa, R., Chu, Q., and Mulder, J., "Reentry Flight Controller Design Using Nonlinear Dynamic Inversion," *Journal of Spacecraft and Rockets*, Vol. 40, No. 1, 2003, pp. 64–71. doi:10.2514/2.3916

[4] Smith, P. R., "A Simplified Approach to Nonlinear Dynamic Inversion Based Flight Control," *23rd Atmospheric Flight Mechanics Conference and Exhibit*, AIAA Paper 1998-4461, 1998. doi:10.2514/6.1998-4461

[5] Bacon, B. J., and Ostroff, A. J., "Reconfigurable Flight Control Using Nonlinear Dynamic Special Accelerometer Implementation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2000-4565, Aug. 2000. doi:10.2514/6.2000-4565

[6] Cox, T. H., and Cotting, M. C., "A Generic Inner-Loop Control Law Structure for Six-Degree-of-Freedom Conceptual Aircraft Design," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2005-0031, Jan. 2005. doi:10.2514/6.2005-31

[7] Ostroff, A. J., and Bacon, B. J., "Enhanced NDI Strategies for Reconfigurable Flight Control," *Proceedings of the American Control Conference*, IEEE Publ., Piscataway, NJ, 2002, pp. 3631–3636. doi:10.1109/ACC.2002.1024492

[8] Sieberling, S., Chu, Q. P., and Mulder, J. A., "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, 2010, pp. 1732–1742. doi:10.2514/1.49978

[9] Bacon, B. J., Ostroff, A. J., and Joshi, S. M., "Reconfigurable NDI Controller Using Inertial Sensor Failure Detection and Isolation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 4, 2001, pp. 1373–1383. doi:10.1109/7.976972

[10] Koschorke, J., Falkena, W., van Kampen, E.-J., and Chu, Q. P., "Time Delayed Incremental Nonlinear Control," *AIAA Guidance, Navigation, and Control (GNC) Conference*, AIAA Paper 2013-4929, 2013. doi:10.2514/6.2013-4929

[11] Theys, B., Dimitriadis, G., Andrianne, T., Hendrick, P., and de Schutter, J., "Wind Tunnel Testing of a VTOL MAV Propeller in Tilted Operating Mode," *Proceedings of the International Conference on Unmanned Aircraft Systems*, IEEE Publ., Piscataway, NJ, 2014, pp. 1064–1072. doi:10.1109/ICUAS.2014.6842358

[12] Chowdhary, G., Johnson, E. N., Chandramohan, R., Kimbrell, M. S., and Calise, A., "Guidance and Control of Airplanes Under Actuator Failures and Severe Structural Damage," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1093–1104. doi:10.2514/1.58028

[13] Bedford, A., and Fowler, W., *Engineering Mechanics Dynamics*, Prentice–Hall, Upper Saddle River, NJ, 2008, pp. 507–515.

[14] Simplicio, P., Pavel, M., van Kampen, E., and Chu, Q., "An Acceleration Measurements-Based Approach for Helicopter Nonlinear Flight Control Using Incremental Nonlinear Dynamic Inversion," *Control Engineering Practice*, Vol. 21, No. 8, 2013, pp. 1065–1077. doi:10.1016/j.conengprac.2013.03.009

[15] Haykin, S., and Widrow, B., *Least-Mean-Square Adaptive Filters*, Wiley, Hoboken, NJ, 2003, pp. 1–12.

[16] Love, J., *Process Automation Handbook*, Springer–Verlag, London, 2007, pp. 173–178. doi:10.1007/978-1-84628-282-9