AIAA SciTech Forum
4-8 January 2016, San Diego, California, USA
AIAA Guidance, Navigation, and Control Conference

10.2514/6.2016-1390

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

Ewoud J.J. Smeur* Qiping Chu†

Guido C.H.E. de Croon‡

Delft University of Technology, Delft, Zuid-Holland, 2629HS, Netherlands

Incremental Nonlinear Dynamic Inversion (INDI) 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, INDI only uses an actuator 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 INDI control: how to deal with measurement and actuator delays and how to deal with a changing actuator 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 INDI, which can estimate the actuator effectiveness online, further reducing the dependence on any modeling effort and (3) the incorporation of the momentum of the propellers in the controller. This controller is suitable for vehicles that experience a different actuator effectiveness across their flight envelope. Furthermore, this approach requires only very course knowledge of model parameters in advance. Real-world experiments show the high performance, disturbance rejection and adaptiveness properties. This approach is applicable to any Micro Aerial Vehicle equipped with a gyroscope, rotors and/or control surfaces.

Nomenclature

$\boldsymbol{\Omega}$	Vehicle angular rate vector
$\dot{\boldsymbol{\Omega}}$	Angular acceleration vector
$\boldsymbol{\omega}$	Rotor angular rate vector
$\boldsymbol{I}_v$	Moment of inertia matrix of the vehicle
$\boldsymbol{I}_r$	Moment of inertia matrix of the rotor
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{M}_r$	Moment vector acting on the propeller
$\boldsymbol{M}_c$	Control moment vector acting on the vehicle
$\boldsymbol{M}_a$	Aerodynamic moment vector acting on the vehicle
b	Width of the vehicle
l	Length of the vehicle
k_1	Force constant of the rotors
k_2	Moment constant of the rotors
T_s	Sample time of the controller
$\boldsymbol{u}$	Actuator input vector
$\boldsymbol{\mu}$	Adaptation rate diagonal matrix

*PhD Candidate, Delft University of Technology - Control and Simulation.
†Associate Professor, Delft University of Technology - Control and Simulation.
‡Assistant Professor, Delft University of Technology - Control and Simulation.

I. Introduction

Micro Aerial Vehicles (MAVs) have increased a lot in popularity as low-cost lightweight CPUs and inertial measurement units (IMUs) 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 PID control.[1,2] Problems with PID control occur when the vehicle is very nonlinear or when the vehicle is subject to large disturbances like wind gusts.

The rotational dynamics of (multirotor) MAVs are quite well understood, so instead 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 the inversion of a model of the MAV's 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 NDI, *Incremental* NDI or INDI, is less model dependent and more robust. It has been described in the literature since the late nineties,[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. Since 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. Firstly, the measurement of angular acceleration is often very noisy and requires filtering. This filtering introduces a delay in the measurement, which should be compensated for. Secondly, the method relies on inverting and therefore modeling the actuators. To achieve a more flexible and model-free controller, the actuator 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 can not be predicted. Initially a setup with multiple accelerometers was proposed by Ostroff and Bacon[5] to measure the angular acceleration. This setup has some drawbacks, because it is complex and the accelerometers are very sensitive to structural vibrations, therefore requiring a lot of filtering. 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, Ostroff and Bacon 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.

Other research focused on delays in the inputs and solving this 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 actuator effectiveness is the sole model still required by INDI. It 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 actuator effectiveness can change during flight. For instance, this can occur due to changes in flight conditions[11] or actuator damage.[12] In order to cope with this, we propose a method to adaptively determine the actuator 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 actuator 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 a commercially available quadrotor and the code is publicly available on Github at https://github.com/EwoudSmeur/paparazzi in the branch *bebop_indi_experiment*. Therefore, the experiments are easy to verify.

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 RPM 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, fixedwing or hybrid.

The outline of this paper is as follows. First, a model of the MAV will be discussed in Section II. Second, Section 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 Section V the experimental setup is explained, followed by the results of the experiments in Section VI.

II. MAV Model

The Bebop quadrotor is shown in figure 1 along with axis definitions, forces and moments. The actuators drive the four rotors, which have an angular velocity of $\boldsymbol{\omega} = [\omega_1, ..., \omega_4]^T$.

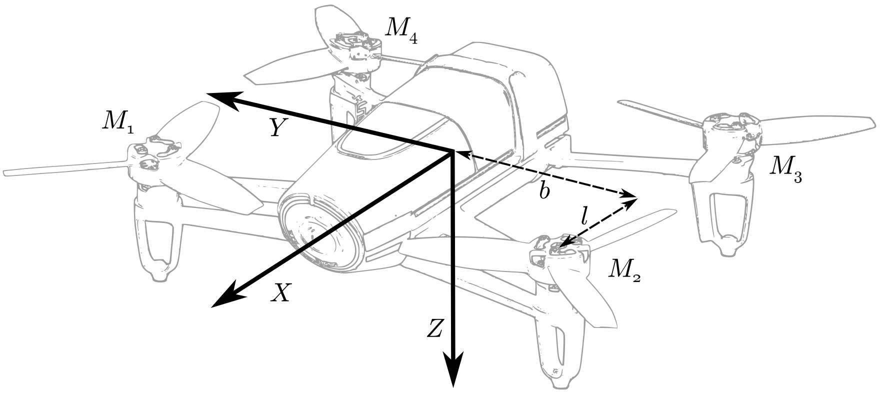

Figure 1: The Bebop Quadcopter used in the experiments with axis definitions. The distance from the center of gravity to each of the rotors along the X axis is given by l and along the Y axis by b.

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 those that describe rotation, shown in equation 1.

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

This equation holds for the rotating propellers, in which case $\boldsymbol{\omega}$ is the angular rate vector of the propeller. The origin of the axis system is located at the center of mass of the propeller but the axes are fixed to the vehicle.

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

Assume that $\boldsymbol{I}_r$ is a diagonal matrix. Because the coordinate system is fixed to the vehicle, I_{r_x} and I_{r_y} are not constant. However, as is shown later on, the terms containing these moments of inertia will disappear.

$$\begin{aligned} I_{r_x}\dot{\omega}_x - I_{r_y}\Omega_z\omega_y + I_{r_z}\Omega_y\omega_z &= M_{r_x} \\ I_{r_y}\dot{\omega}_y + I_{r_x}\Omega_z\omega_x - I_{r_z}\Omega_x\omega_z &= M_{r_y} \\ I_{r_z}\dot{\omega}_z - I_{r_x}\Omega_y\omega_x + I_{r_y}\Omega_x\omega_y &= M_{r_z} \end{aligned} \tag{3}$$

The propellers are light-weight and have a small moment of inertia compared to the vehicle. Relevant precession terms are therefore those that contain the relatively large ω_z. Since the rotors spin very fast around the z axis, it is safe to assume $\omega_x \ll \omega_z$ and $\omega_y \ll \omega_z$ and assume that $\dot{\omega}_x$ and $\dot{\omega}_y$ are negligible. Then the moments exerted on the rotors due to the their rotational dynamics are given by equation 4. Note presence of the term $I_{r_z}\dot{\omega}_z$, which is the moment necessary to change the angular velocity of a rotor. In Section VI it will be shown that this term is very important.

$$\boldsymbol{M}_{r_i} = \begin{bmatrix} M_{r_x} \\ M_{r_y} \\ M_{r_z} \end{bmatrix} = \begin{bmatrix} I_{r_z}\Omega_y\omega_z \\ -I_{r_z}\Omega_x\omega_z \\ I_{r_z}\dot{\omega}_z \end{bmatrix} \tag{4}$$

This equation holds for each of the four rotors, so the moment acting on a rotor is given a subscript i to indicate the rotor number $\boldsymbol{M}_{r_i}$. In equation 5 the total moment due to the rotational effects of the rotors is shown. Since motors 1 and 3 spin in the opposite direction of rotors 2 and 4, a factor $(-1)^i$ is introduced.

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

Now consider the Euler equation 1 for the entire vehicle. Since action equals minus reaction, the moments from the rotor dynamics are subtracted from the other moments.

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

Where $\boldsymbol{I}_v$ 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 body velocity. The control moment $\boldsymbol{M}_c(\boldsymbol{\omega})$ is elaborated in equation 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 figure 1.

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

If we now take equation 6, insert equations 4 and 7 and solve for the angular acceleration $\dot{\boldsymbol{\Omega}}$, we arrive at equation 8.

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

Where $\boldsymbol{F}(\boldsymbol{\Omega},\boldsymbol{v}) = \boldsymbol{I}_v^{-1}(\boldsymbol{M}_a(\boldsymbol{\Omega},\boldsymbol{v}) - \boldsymbol{\Omega}\times\boldsymbol{I}\boldsymbol{\Omega})$ 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 equations 9, 10, 11 and 12 respectively. Note that the sample time T_s of the quadrotor is introduced to ease future calculations.

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

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

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

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

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 equation 8, the quadrotor is actually a system of the form $\dot{x} = f(x) + g(x,u,\dot{u})$. In Section III a solution to this type of problem will be shown.

III. Incremental Nonlinear Dynamic Inversion

Consider equation 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. Now a Taylor expansion can be applied to equation 8 resulting in equation 13.

$$\begin{aligned}\dot{\boldsymbol{\Omega}} = \;& \boldsymbol{F}(\boldsymbol{\Omega}_0, \boldsymbol{v}_0) + \tfrac{1}{2}\boldsymbol{G}_1\boldsymbol{\omega}_0^2 + T_s\boldsymbol{G}_2\dot{\boldsymbol{\omega}}_{\mathbf{0}} - \boldsymbol{C}(\boldsymbol{\Omega}_0)\boldsymbol{G}_3\boldsymbol{\omega}_0 \\ & + \tfrac{\partial}{\partial\boldsymbol{\Omega}}(\boldsymbol{F}(\boldsymbol{\Omega}, \boldsymbol{v}_0) + \boldsymbol{C}(\boldsymbol{\Omega})\boldsymbol{G}_3\boldsymbol{\omega}_0)|_{\boldsymbol{\Omega}=\boldsymbol{\Omega}_0}(\boldsymbol{\Omega} - \boldsymbol{\Omega}_0) \\ & + \tfrac{\partial}{\partial\boldsymbol{v}}(\boldsymbol{F}(\boldsymbol{\Omega}_0, \boldsymbol{v}))|_{\boldsymbol{v}=\boldsymbol{v}_0}(\boldsymbol{v} - \boldsymbol{v}_0) \\ & + \tfrac{\partial}{\partial\boldsymbol{\omega}}(\tfrac{1}{2}\boldsymbol{G}_1\boldsymbol{\omega}^2 - \boldsymbol{C}(\boldsymbol{\Omega}_0)\boldsymbol{G}_3\boldsymbol{\omega})|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0}(\boldsymbol{\omega} - \boldsymbol{\omega}_0) \\ & + \tfrac{\partial}{\partial\dot{\boldsymbol{\omega}}}(T_s\boldsymbol{G}_2\dot{\boldsymbol{\omega}})|_{\dot{\boldsymbol{\omega}}=\dot{\boldsymbol{\omega}}_0}(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0)\end{aligned} \tag{13}$$

This equation predicts the angular acceleration after an infinitesimal timestep 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: $\boldsymbol{F}(\boldsymbol{\Omega}_0, \boldsymbol{v}_0) + \frac{1}{2}\boldsymbol{G}_1\boldsymbol{\omega}_0^2 + T_s\boldsymbol{G}_2\dot{\boldsymbol{\omega}}_{\mathbf{0}} - \boldsymbol{C}(\boldsymbol{\Omega}_0)\boldsymbol{G}_3\boldsymbol{\omega}_0 = \dot{\boldsymbol{\Omega}}_0$. 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 a lot 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 gives equation 14.

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}}_0 + \boldsymbol{G}_1\text{diag}(\boldsymbol{\omega}_0)(\boldsymbol{\omega} - \boldsymbol{\omega}_0) + T_s\boldsymbol{G}_2(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0) - \boldsymbol{C}(\boldsymbol{\Omega}_0)\boldsymbol{G}_3(\boldsymbol{\omega} - \boldsymbol{\omega}_0) \tag{14}$$

Above it is stated that the angular acceleration is measured by deriving it from the angular rates. In most cases, the gyroscope measurements from an MAV are very noisy due to vibrations of the vehicle due to the propellers and motors. Since 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 equation 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.

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

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

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}}_f + \boldsymbol{G}_1\text{diag}(\boldsymbol{\omega}_f)(\boldsymbol{\omega} - \boldsymbol{\omega}_f) + T_s\boldsymbol{G}_2(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_f) - \boldsymbol{C}(\boldsymbol{\Omega}_f)\boldsymbol{G}_3(\boldsymbol{\omega} - \boldsymbol{\omega}_f) \tag{16}$$

This equation is not yet ready to be inverted, because it contains the derivative of the angular rate of the propellers. Since 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 equation 17.

$$\dot{\boldsymbol{\Omega}} = \dot{\boldsymbol{\Omega}}_f + \boldsymbol{G}_1\text{diag}(\boldsymbol{\omega}_f)(\boldsymbol{\omega} - \boldsymbol{\omega}_f) + \boldsymbol{G}_2(\boldsymbol{\omega} - \boldsymbol{\omega}z^{-1} - \boldsymbol{\omega}_f + \boldsymbol{\omega}_f z^{-1}) - \boldsymbol{C}(\boldsymbol{\Omega}_f)\boldsymbol{G}_3(\boldsymbol{\omega} - \boldsymbol{\omega}_f) \tag{17}$$

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

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

Inversion of this equation for $\boldsymbol{\omega}$ yields equation 19, where $^+$ denotes the Moore-Penrose pseudoinverse. 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 equation 19 the required inputs $\boldsymbol{\omega}$ can be calculated. This input is given with respect to a previous input $\boldsymbol{\omega}_f$. If we define $\Delta\boldsymbol{\omega} = \boldsymbol{\omega} - \boldsymbol{\omega}_f$, it clearly an *incremental* control law.

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

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 $\boldsymbol{G}_3$ and 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, thrust and drag measurements of the propellers at different rotational speeds. Clearly this requires a lot 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 first. 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, equation 18 is used for parameter estimation and is written as equation 20. From this equation a least squares solution is found for the matrices $\boldsymbol{G}_1$, $\boldsymbol{G}_2$ and $\boldsymbol{G}_3$.

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

From the data we can also investigate the importance of some of the terms by comparing the least squares error with and without the term. It turns out that on a typical dataset, leaving out the matrix $\boldsymbol{G}_3$ only results in an estimation squared error increase of ~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 ~0.9%. Therefore, we can simplify the INDI control law of equation 19 to equation 21.

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

B. Implementation

With the simplifications described in subsection A, the final INDI control scheme is shown in figure 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 an error of the angular acceleration $\dot{\boldsymbol{\Omega}}_{\text{err}}$.

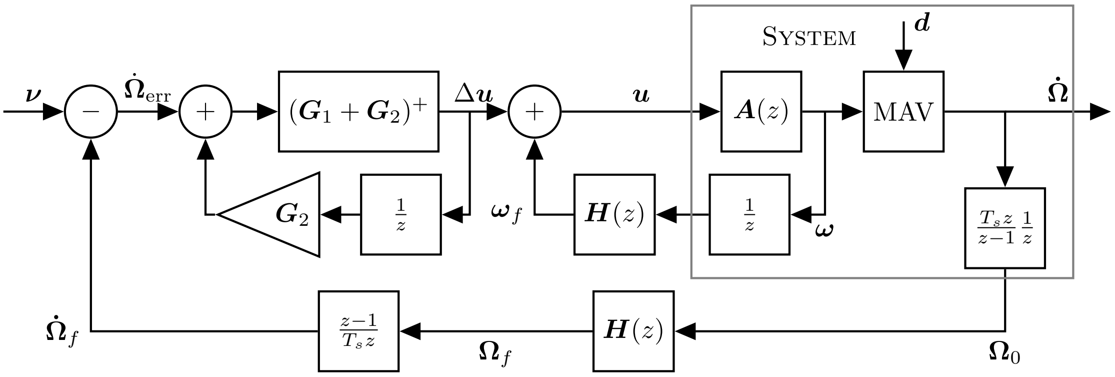

Figure 2: INDI control scheme. $\boldsymbol{A}(z)$ denotes the actuator dynamics and $\boldsymbol{H}(z)$ is the second order filter. In the diagram we include a model of the quadcopter inside the box, because it allows the closed loop analysis in Section C. Since the matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are not square, we take the pseudo inverse to solve the problem of control allocation, denoted by $^+$.

Note that equation 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 $\boldsymbol{u}$ and $\Delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{\omega}_f$. In figure 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 figure 4. Here $\boldsymbol{A}'(z)$ is a model of the actuator dynamics.

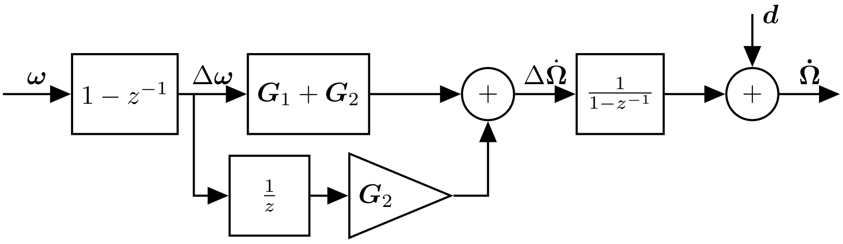
$\boldsymbol{\omega}$
$1 - z^{-1}$
$\Delta\boldsymbol{\omega}$
$\boldsymbol{G}_1 + \boldsymbol{G}_2$
$+$
$\Delta\dot{\boldsymbol{\Omega}}$
$\frac{1}{1-z^{-1}}$
$\boldsymbol{d}$
$+$
$\dot{\boldsymbol{\Omega}}$
$\frac{1}{z}$
$\boldsymbol{G}_2$

Figure 3: The contents of the block named 'MAV' in figure 2. $\boldsymbol{d}$ is a disturbance term that bundles disturbances and unmodeled dynamics.

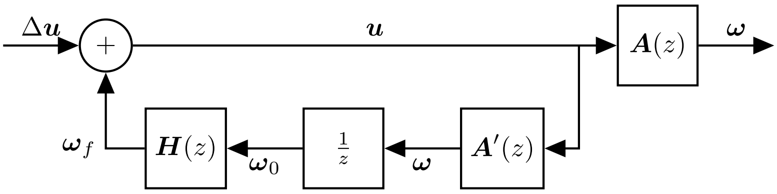

Figure 4: If actuator feedback is not available, the actuator state should be estimated with a model of the actuator dynamics.

C. Closed Loop Analysis

Consider the control diagram shown in figure 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 equation 22 and 23. Here $\mathbf{TF}_{x\rightarrow y}$ denotes the transfer function from point x to y in the control diagram.

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

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

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

Then, the last part of the open loop is represented by equation 24. Note that for this analysis disturbances are not taken into account.

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

Using these intermediate results, the open loop transfer function of the entire system is shown in equation 25.

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

Using this result, the closed loop transfer function of the entire system is shown in equation 26.

$$\begin{aligned}
\mathbf{TF}_{\boldsymbol{\nu}\rightarrow\dot{\boldsymbol{\Omega}}}(z) &= (\boldsymbol{I} + \mathbf{TF}_{\dot{\boldsymbol{\Omega}}_{\text{err}}\rightarrow\dot{\boldsymbol{\Omega}}}(z)\boldsymbol{I}H(z)z^{-1})^{-1}\mathbf{TF}_{\dot{\boldsymbol{\Omega}}_{\text{err}}\rightarrow\dot{\boldsymbol{\Omega}}}(z) \\
&= (\boldsymbol{I} + \boldsymbol{I}(1 - A(z)H(z)z^{-1})^{-1}A(z)\boldsymbol{I}H(z)z^{-1})^{-1}\boldsymbol{I}(1 - A(z)H(z)z^{-1})^{-1}A(z) \\
&= \boldsymbol{I}\tfrac{(1-A(z)H(z)z^{-1})^{-1}A(z)}{1+(1-A(z)H(z)z^{-1})^{-1}A(z)H(z)z^{-1}} \\
&= \boldsymbol{I}\tfrac{A(z)}{1-A(z)H(z)z^{-1}+A(z)H(z)z^{-1}} \\
&= \boldsymbol{I}A(z)
\end{aligned} \tag{26}$$

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 figure 2) to the angular acceleration. The derivation is given in equation 27 in which use is made of equation 25.

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

With equation 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 the faster the angular acceleration is measured, the faster the drone can respond and the faster the actuators can react, the faster the disturbance is neutralized.

D. Attitude Control

The angular acceleration of the MAV is accurately controlled by the system shown in figure 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 PD controller (a gain on the rate error and a gain on the angle error), shown in figure 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), a value 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.

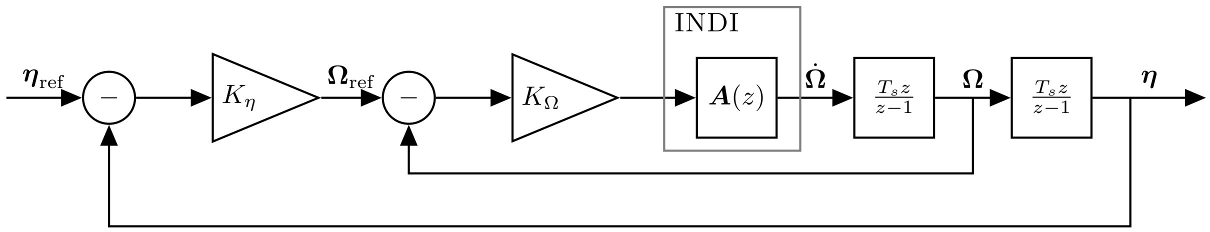

Figure 5: The design of the attitude controller based on the closed loop response of the INDI controller.

E. Altitude Control

The INDI controller derived in the beginning of this section controls the angular acceleration around the three axes roll, pitch and yaw x, y and z. 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 INDI

The INDI approach does not rely on a complex model of the system. Instead, only the actuator effeciveness needs to be estimated. The actuator effectiveness depends on the moment of inertia of the vehicle, the type of motors and propellers. A change in any of these will require re-estimation of the actuator effectiveness. Moreover, the actuator 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 equation 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 necesaary as 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 actuator effectiveness based on the error. The actuator effectiveness includes both $\boldsymbol{G}_1$ as well as $\boldsymbol{G_2}$, as is shown in equation 29. Clearly, when there is no change in input, the actuator 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.

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

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{G}_2 \end{bmatrix} \tag{29}$$

Note that the filter f can be different for the online parameter estimation than for the actual control. Equation 28 makes use of $\Delta\dot{\boldsymbol{\Omega}}_f$, which is a derivative higher than $\dot{\boldsymbol{\Omega}}_f$ in the control equation 21. Since differentiating amplifies high frequencies, a 'stronger' filter is necessary. Such a filter also gives more delay, but this is not a problem as the parameter estimation is not time-critical.

When an approximate actuator 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 thus an initial guess of the actuator 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, so in order to find a good adaptation constant some tuning is required.

V. Experimental Setup

To validate the performance of the INDI controller developed in Section III and the adaptive parameter estimation from Section IV, several experiments were conducted. These experiments were performed using the Bebop quadcopter from Parrot shown in Figure 1. The Bebop weighs 396.2 grams and can be equipped with bumpers, which are 12 grams per bumper. For these experiments, the bumpers where 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 measurments etc. The accelerometer, gyroscope and control loops were running at 512 Hz.

Three experiments test three key properties of the controller:

- Performance

- Disturbance rejection
- Adaptation

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

A. Performance

In order to put the responsiveness of the system to the test and 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 degrees and the period is half a second (0.25 seconds positive and 0.25 seconds negative).

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 equation 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.

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 occuring 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 grams to a container located in an off-centered position on the quadrotor while it is flying, as shown in Figure 6. The container is located on the front of the drone and has a distance of about 11 cm to the center of gravity, 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 Figure 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.

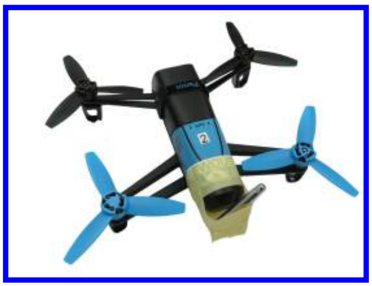

Figure 6: The container attached to the nose of the Bebop quadrotor with one weight inside.

Figure 7: The Bebop quadrotor with bumpers.

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 resistent to disturbances than a single loop design.[16] Because of this, the reference is expected to be tracked right after the disturbance.

C. Adaptation

The Bebop quadcopter has the possibility to fly with bumpers, as is shown in Figure 7. Though these bumpers are lightweight, 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 Section IV should deal with adding or removing the bumpers.

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 again. After take off, the matrices should converge to their original state.

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 setpoint. The amplitude of the doublet is 5 degrees and the period is one second (0.5 seconds positive and 0.5 seconds 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.

VI. Results

This section deals with the results of the experiments described in Section 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 (non-causal) filter. For the actual control, the onboard filtered (and delayed) angular acceleration was used.

A. Performance

The results of one of the performed doublets are visible below. Figure 9 shows the angular acceleration around the x axis denoted by $\dot{p}$ and the reference angular acceleration denoted by $\dot{p}_{\text{ref}}$. Additionally, the reference is filtered with the actuator dynamics, resulting in $\dot{p}_{\text{ref}_A}$. This signal is the angular acceleration that is expected based on the calculations in Section C. 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 $\dot{p}_{\text{ref}_A}$.

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 Figure 8. From this figure it can be seen that the quadcopter reaches its reference roll angle within 0.2 seconds with a very small overshoot.

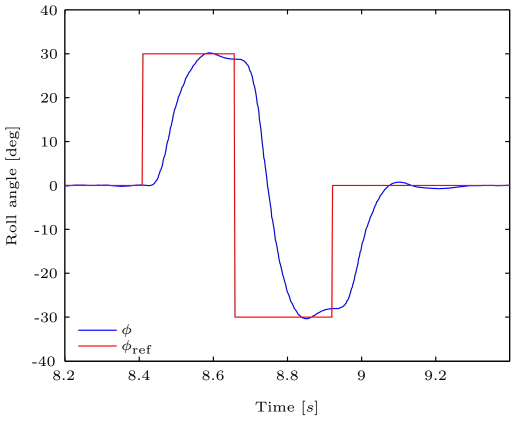

Figure 8: The measured (blue) and reference (red) roll angle during the doublet for the INDI controller.

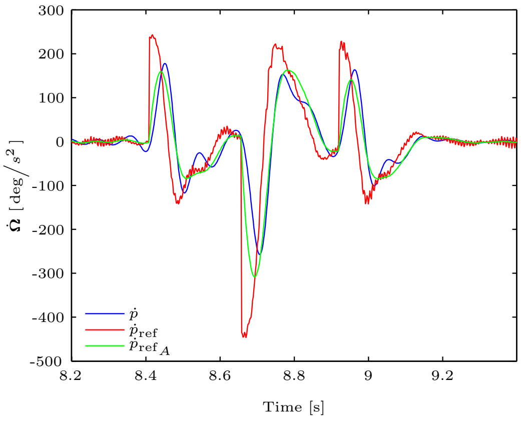

Figure 9: Angular acceleration in the roll axis during doublet input. The blue, red and green lines are the measured, reference and expected angular acceleration respectively.

As expected, the PID controller performs very similar to the INDI controller in terms of rise time and overshoot. This shows that the INDI controller is on par with a traditional PID controller in terms of responsiveness for the pitch and roll. In subsection D it will be shown that the INDI controller performs better in the yaw axis.

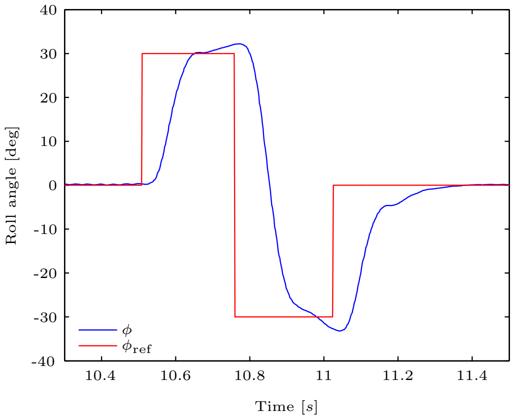

Figure 10: The measured (blue) and reference (red) roll angle during the doublet for the PID controller.

B. Disturbance Rejection

The way the weight of 42.5 grams was placed in a container attached to the nose of the quadrotor is shown in Figure 11. 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.

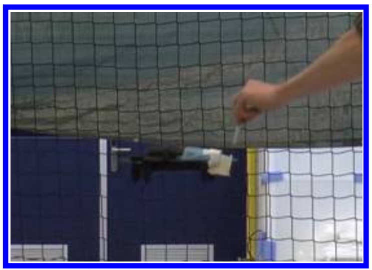

Figure 11: A weight is placed in the container.

Figure 13 shows the angular acceleration that is the result of the disturbance. From the figure it is clear that the disturbance happened just after 13 seconds. 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 seconds 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 Figure 12. After 0.3 seconds 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 Figure 15. It shows the difference of the rotational rate of the front and rear motors divided by four: $(\omega_1 + \omega_2 - \omega_3 - \omega_4)/4$. This indicates the average magnitude in RPM that each motor contributes to the pitch control (see equation 7). Clearly, there is a difference before and after the disturbance which can be quantified as an average change of 578 RPM over the interval [12.6 13.0] versus [13.4 13.8]. This demonstrates that the disturbance was really a step and that the INDI controller can cope with such a disturbance rapidly.

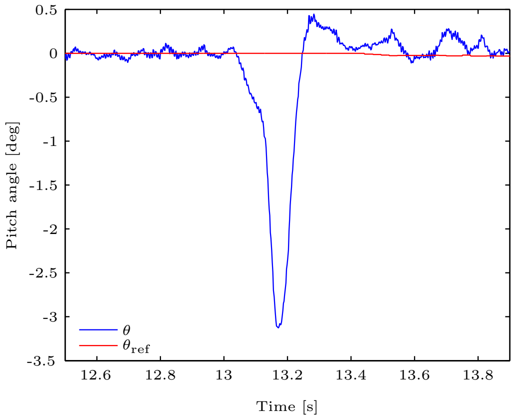

Figure 12: The pitch angle during the disturbance. Note the the absence of a steady state error within 0.3 seconds of the start of the disturbance.

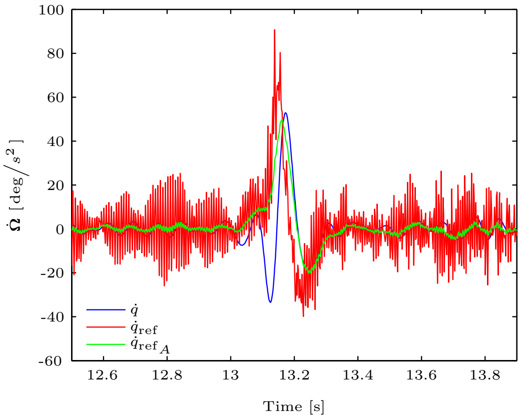

Figure 13: The angular acceleration during the disturbance. The blue and red lines are the measured and reference angular acceleration respectively.

Figure 14 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, so the initial disturbance can be different. However, the persisting disturbance is the same, because the weight has exactly the same mass. Iit takes about 1.5 seconds before the pitch angle is back at zero again, which is approximately 5 times longer than for the INDI controller. One might say that the I gain should be larger, but this will deteriorate the performance in the previous experiment.

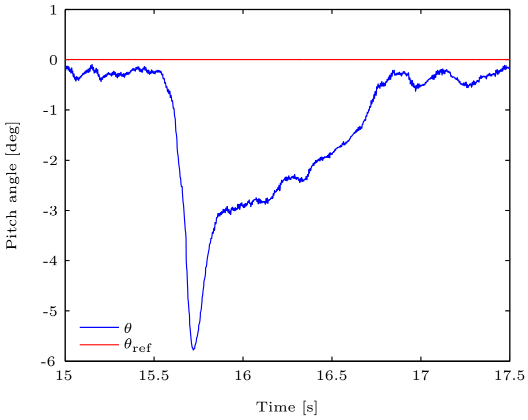

Figure 14: The pitch angle during the disturbance for the PID controller. A steady state error persists almost 1.5 seconds after the start of the disturbance.

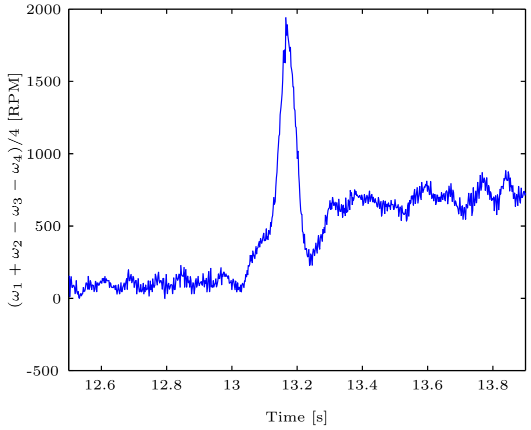

Figure 15: The difference between the rotational rate of the front motors and the rear motors, indicating the moment generated in the pitch axis. The INDI controller can rapidly compensate offsets due to the incremental control structure.

C. Adaptation

Figure 16 through 18 show how each row of the $\boldsymbol{G}_1$ matrix evolves over time as a result of the experiment described in subsection C of Section V. The same is shown in Figure 19 for the third row of the $\boldsymbol{G}_2$ matrix. Each line represents one of 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] seconds and again in [66 125] seconds, 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 figure 18, which corresponds to the yaw.

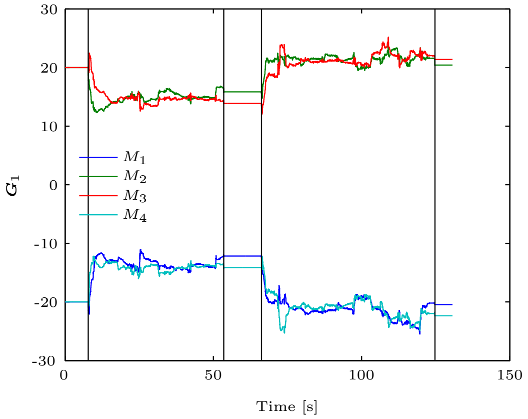

Figure 16: The first row of the $\boldsymbol{G}_1$ matrix corresponding to the roll.

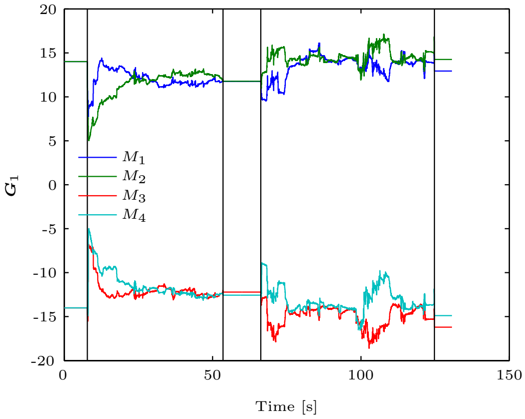

Figure 17: The second row of the $\boldsymbol{G}_1$ matrix corresponding to the pitch.

Also in Figure 16 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 take-off 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. When the INDI controller will try to attain certain angular accelerations, the quadrotor will not rotate and the adaptation algorithm will adapt to this. When landing, these interactions with the floor can also occur.

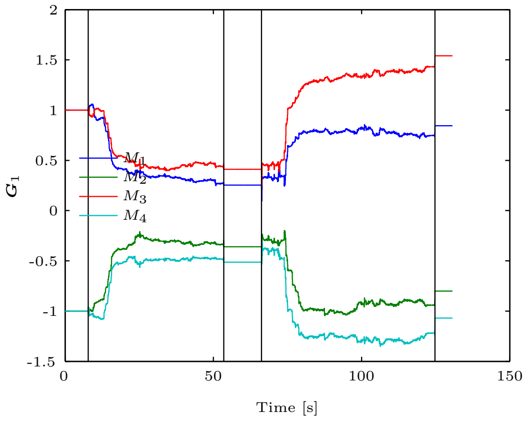

Figure 18: The third row of the $\boldsymbol{G}_1$ matrix corresponding to the yaw.

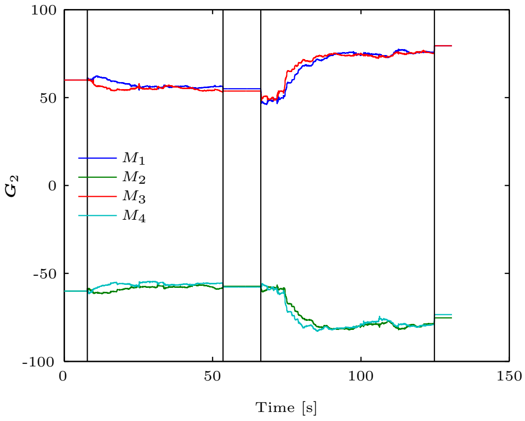

Figure 19: The third row of the $\boldsymbol{G}_2$ matrix corresponding to the yaw.

Notice the large difference in effectiveness between the actuators in the second part of the flight in Figure 18. This illustrates the added value of adaptive INDI, as often the actuators are assumed to perform equal to each other, while at least in this case they do not. These differences between the actuators are also observed with the estimation method described in subsection A of Section III for multiple flights.

D. Yaw control

Finally, consider Figure 20. It shows for each timestep the change in angular acceleration in the yaw axis ($\Delta r = r - r_0$) during the large control inputs discussed above. 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 equation 21, which is a close match. In green, the figure shows the predicted change in angular acceleration if we neglect $\boldsymbol{G}_2$. Clearly the motor spin-up torque is very significant.

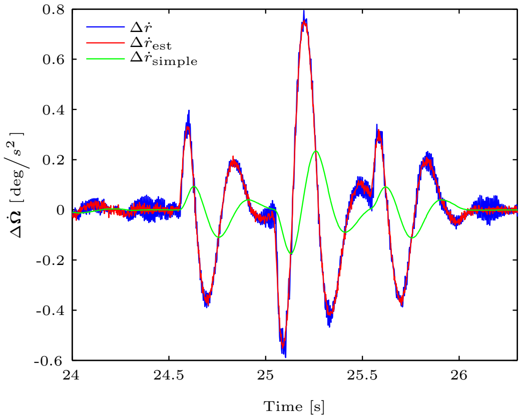

Figure 20: The change in angular acceleration in the yaw axis along with the predicted change in angular acceleration based on the change in inputs and the $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ matrices. Also the prediction based on only the $\boldsymbol{G}_1$ matrix is shown, denoted by $\Delta\dot{r}_{\text{simple}}$

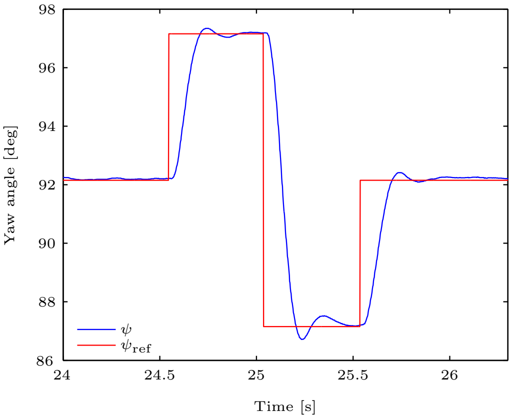

Figure 21: The measured and reference yaw angle during the doublet for the INDI controller.

Figure 21 shows the resultant doublet response of the yaw angle. Compare this with Figure 22, which shows the doublet response for the PID controller. The INDI controller clearly has a faster rise time and less overshoot.

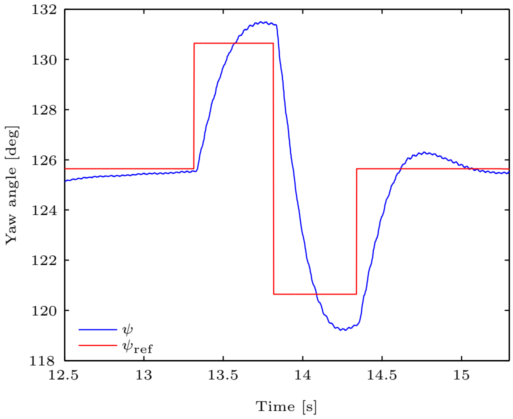

Figure 22: The measured and reference yaw angle during the doublet for the PID controller.

VII. Conclusion

Concluding, adaptive INDI is a very promising technique for MAV control. Due to 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 actuator effectiveness matrices are shown to be adapted online, resulting in a model free controller. Only if a high performance outer loop is required, some knowledge of the actuator dynamics is needed. These properties result in a very flexible and powerful controller, that is still easy to understand.

Acknowledgments

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

References

[1] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor Aerial Vehicles: Modeling, Estimation and Control of Quadrotor. IEEE Robotics & Automation Magazine, 2012.

[2] Emil Fresk and George Nikolakopoulos. Full Quaternion Based Attitude Control for a Quadrotor. European Control Conference (ECC), 2013.

[3] R.R. da Costa, Q.P. Chu, and J.A. Mulder. Reentry Flight Controller Design Using Nonlinear Dynamic Inversion. Journal of Spacecraft and Rockets, Volume 40, Issue 1, pp. 64-71, 2003.

[4] P. R. Smith. A Simplified Approach To Nonlinear Dynamic Inversion Based Flight Control. AIAA Atmospheric Flight Mechanics Conference and Exhibit, AIAA, 1998.

[5] Barton J. Bacon and Aaron J. Ostroff. Reconfigurable Flight Control Using Nonlinear Dynamic Special Accelerometer Implementation. AIAA Paper 2000-4565, AIAA Guidance, Navigation and Control Conference, Denver, CO, 2000.

[6] Timothy H. Cox and M. Christopher Cotting. A Generic Inner-Loop Control Law Structure for Six-Degree-of-Freedom Conceptual Aircraft Design. AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2005.

[7] Aaron J. Ostroff and Barton J. Bacon. Enhanced NDI Strategies For Reconfigurable Flight Control. Proceedings of the American Control Conference, 2002.

[8] S. Sieberling, Q. P. Chu, and J. A. Mulder. Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction. Journal of Guidance Control and Dynamics, Volume 33, Issue 6, pp. 1732-1742, 2010.

[9] Barton J. Bacon, Aaron J. Ostroff, and Suresh M. Joshi. Reconfigurable NDI Controller Using Inertial Sensor Failure Detection & Isolation. IEEE Transactions On Aerospace And Electronic Systems Volume. 37, Issue 4, 2001.

[10] Johann Koschorke, Wouter Falkena, Erik-Jan van Kampen, and Qi Ping Chu. Time Delayed Incremental Nonlinear Control. AIAA Guidance, Navigation, and Control Conference, 2013.

[11] T. Andrianne P. Hendrick B. Theys, G. Dimitriadis and J. De Schutter. Wind Tunnel Testing of a VTOL MAV Propeller in Tilted Operating Mode. International Conference on Unmanned Aircraft Systems (ICUAS), 2014.

[12] Girish Chowdhary, Eric N. Johnson, Rajeev Chandramohan, M. Scott Kimbrell, and Anthony Calise. Guidance and Control of Airplanes Under Actuator Failures and Severe Structural Damage. Journal of Guidance Control and Dynamics, Volume 36, Issue 4, pp. 1093-1104, 2013.

[13] Anthony Bedford and Wallace Fowler. Engineering Mechanics Dynamics. Prentice Hall, Pearson Education South Asia Pte Ltd, ISBN 981-06-7940-8, 2008.

[14] P. Simplicio, M.D. Pavel, E. van Kampen, and Q.P. Chu. An acceleration measurements-based approach for helicopter nonlinear flight control using Incremental Nonlinear Dynamic Inversion. Control Engineering Practice, Volume 21, Issue 8, pp. 1065-1077, 2013.

[15] Simon Haykin and Bernard Widrow. Least-Mean-Square Adaptive Filters. Wiley, 2003.

[16] K. Åström and T. Hägglund. PID Controllers: Theory, Design and Tuning. ISA, 1998.

This article has been cited by:

1. Robert (Wes) W. Morgan. Aerospace Vehicle Steering with Uncontrolled Axial Acceleration . [Citation] [PDF] [PDF Plus]
2. Y. Bouzid, H. Siguerdidjane, Y. Bestaoui. 2016. Hierarchical Autopilot Design based on Immersion & Invariance and Nonlinear Internal Model Tracking Controllers for Autonomous system. *IFAC-PapersOnLine* **49**:5, 103-108. [Crossref]