Proceedings of 2012 IEEE
International Conference on Mechatronics and Automation
August 5 - 8, Chengdu, China

Modeling and Adaptive Control of a Quadrotor

Matthias Schreier
Institute of Automatic Control and Mechatronics
Control Theory and Robotics Lab
TU Darmstadt
Landgraf-Georg-Str. 4, 64283 Darmstadt, Germany
schreier@rtr.tu-darmstadt.de

***Abstract*—In this paper, we propose two variants of adaptive state space controllers for attitude stabilization and self-tuning of a four-rotor aerial robot, a quadrotor. First of all, the use of a Model Identification Adaptive Controller (MIAC) is proposed in terms of combining a recursive least-squares estimator with exponential forgetting with an integral discrete-time state space controller. Furthermore, a continuous-time Model Reference Adaptive Control (MRAC) scheme based on Lyapunov theory is applied to the simplified dynamics of a quadrotor, which guarantees global asymptotic stability for at least linear overall systems. The effectiveness of the suggested adaptive methods is demonstrated in simulations with a quaternion-based nonlinear dynamic model of a quadrotor derived in this work. The results are compared to a designed nonadaptive integral state space controller.**

***Index Terms*—UAV, VTOL, quadrotor, adaptive control, identification, stabilization**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are a class of mobile robots which recently received considerable interest. This especially holds for quadrotors that belong to the so-called Vertical Take-Off and Landing (VTOL) systems. A quadrotor is an UAV with four rotors in cross configuration whose motion behavior is influenced solely by changing its individual propellers' speeds. This is mainly done by use of lightweight, highly efficient brushless direct current motors with advantageous dynamic properties. In contrast to helicopters, quadrotors have better maneuverability, are easier and cheaper to produce due to simpler mechanics and are therefore more and more favored [1], [2]. However, they require more complex control algorithms in order to successfully stabilize the nonlinear unstable multivariable system. In the past, a considerable amount of work has been done to fulfill this requirement. The following list, which is not intended to be exhaustive, provides a short overview over some of the contributions.

- P(I)D control [1], [3], [18]
- PD^2 control [3], [4]
- Sliding mode control [1], [5]
- Adaptive sliding mode control [2]
- State-dependent Riccati control [1], [6]
- Backstepping control [1], [5], [7]
- Feedback linearization [2], [8], [9]
- H_∞ control [10]

The present paper focuses on the adaptation of the stabilization control loops for coping with either varying or even unknown system inertia tensor. This is especially useful for the increasing amount of modular quadrotor platforms which will often change sensor configuration or load.

The remainder of this paper is structured as follows. In Section II, the mathematical model of a quadrotor UAV is presented in terms of Euler angles as well as in a more sophisticated quaternion representation which is used for simulation studies. Section III explains the quadrotor control concept without adaptive extensions and the design of a discrete-time integral state space controller for roll and pitch movements. This serves as the basis for the comparison with the two adaptive variants presented in Section IV which are considered the core of this paper. First, the use of a Model Identification Adaptive Controller (MIAC) is proposed in terms of combining a recursive least-squares estimator with exponential forgetting with an integral discrete-time state space controller. Furthermore, a continuous-time Model Reference Adaptive Control (MRAC) scheme based on Lyapunov theory is applied to the simplified dynamics of a quadrotor which guarantees global asymptotic stability for at least linear overall systems. Simulation results underline the effectiveness of the proposed control schemes.

II. Modeling of a quadrotor UAV

The dynamic behavior of a quadrotor has been described in a variety of publications to varying degrees of complexity, see for example [11]–[14]. The following derivation is roughly based upon [8], but extends its mathematical description by the full consideration of nonlinear coupling between the axes as well as by a quaternion-based representation. Aerodynamic side effects and elastic deformations play a minor role at slow speeds, sufficient stiffness and realistic flight maneuvers and are therefore omitted. The basis of the model is Fig. 1 which shows a freely moving quadrotor in three-dimensional space. The origin of the body-fixed frame $(CS)_B$ (basis vectors $\boldsymbol{e}_{1B}, \boldsymbol{e}_{2B}, \boldsymbol{e}_{3B}$) is located at the center of gravity whose position in earth-fixed inertial frame $(CS)_I$ (basis vectors $\boldsymbol{e}_{1I}, \boldsymbol{e}_{2I}, \boldsymbol{e}_{3I}$) is given by the vector ${}_{(I)}\boldsymbol{r} = (x, y, z)^T$. The orientation of the quadrotor is first described by three Euler angles (roll angle ϕ, pitch angle θ, yaw angle ψ) combined in the vector $\boldsymbol{\Omega} = (\phi, \theta, \psi)^T$. A rotation from $(CS)_I$ to $(CS)_B$ is realized by three consecutive elementary rotations. Earth-fixed

978-1-4673-1278-3/12/$31.00 ©2012 IEEE

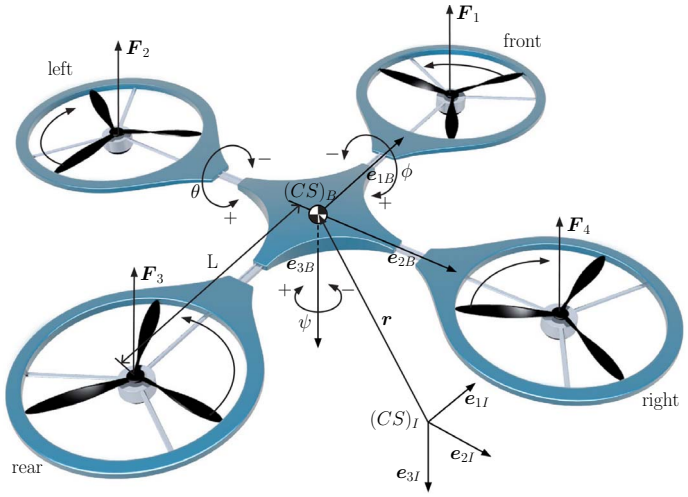

Fig. 1. Configuration of a quadrotor

basis vectors are transfered to body-fixed ones by first rotating around the z-axis with rotation matrix $\boldsymbol{R}_z(\psi)$, afterwards around the new y-axis with rotation matrix $\boldsymbol{R}_y(\theta)$ and at last around the newest resulting x-axis with rotation matrix $\boldsymbol{R}_x(\phi)$. The entire orthonormal rotation matrix ${}^I\boldsymbol{R}_B \in SO(3)$ which transforms direction vectors from the body-fixed to the inertial frame results from right multiplication of elementary rotations. With the use of the abbreviations c_x for $\cos(x)$ and s_x for $\sin(x)$ it reads

$$\begin{aligned} {}^I\boldsymbol{R}_B &= \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\phi) \\ &= \begin{pmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{pmatrix} \\ &= \begin{pmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix}. \end{aligned} \tag{1}$$

Every rotor ($i = 1, \ldots, 4$) generates a thrust force $\boldsymbol{F}_i$ proportional to the square of the rotor's angular speed ω_i [8]. The proportionality factor is called thrust factor b and is dependent on the air density, the geometry of the rotor blade as well as its pitch angles. In body-fixed coordinates we consequently get ${}_{(B)}\boldsymbol{F}_i = -b\omega_i^2 \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T$. By transforming the thrust forces to the inertial frame and making use of the principle of linear momentum, the following equations can be introduced for a quadrotor of mass m under gravity g:

$${}_{(I)}\ddot{\boldsymbol{r}} = \begin{pmatrix} 0 & 0 & g \end{pmatrix}^T + {}^I\boldsymbol{R}_B \sum_{i=1}^{4} \frac{{}_{(B)}\boldsymbol{F}_i}{m} \tag{2}$$

Consequently, the translational movement of the quadrotor is fully described. With the principle of angular momentum a second vector differential equation can be formulated that also describes the rotatory movement of the body:

$${}_{(B)}\boldsymbol{J}\,{}_{(B)}\dot{\boldsymbol{\omega}}_Q + \left({}_{(B)}\boldsymbol{\omega}_Q \times {}_{(B)}\boldsymbol{J}\,{}_{(B)}\boldsymbol{\omega}_Q\right) = {}_{(B)}\boldsymbol{M} - {}_{(B)}\boldsymbol{M}_G \tag{3}$$

Here, $\boldsymbol{J}$ is the inertia tensor of the symmetric rigid body around its center of mass, $\boldsymbol{\omega}_Q$ its angular velocity vector with ${}_{(B)}\boldsymbol{\omega}_Q = (\omega_x, \omega_y, \omega_z)^T$, $\boldsymbol{M}$ the vector of the extern torques applied to the body, and $\boldsymbol{M}_G$ the vector of gyroscopic torques. All values are expressed in the body-fixed frame which facilitates the calculations tremendously. The inertia tensor only contains diagonal entries due to the symmetric structure, so ${}_{(B)}\boldsymbol{J} = \mathrm{diag}(J_x, J_y, J_z)$. The vector of extern torques $\boldsymbol{M}$ is composed of the thrust differences and drag moments of the individual rotors and, under consideration of the rotation directions, results in

$${}_{(B)}\boldsymbol{M} = \begin{pmatrix} Lb(\omega_2^2 - \omega_4^2) \\ Lb(\omega_1^2 - \omega_3^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{pmatrix}. \tag{4}$$

The so called drag factor d describes an air resistance, the parameter L is equal to the length of the lever between center of mass and the four motors. The gyroscopic torques are a result of rotational movements of the quadrotor in combination with rotating rotors of inertia J_r and are calculated as

$${}_{(B)}\boldsymbol{M}_G = J_r \left({}_{(B)}\boldsymbol{\omega}_Q \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right) (\omega_1 - \omega_2 + \omega_3 - \omega_4). \tag{5}$$

Furthermore, the relation between the angular velocity vector $\boldsymbol{\omega}_Q$ and the vector of Euler angles $\boldsymbol{\Omega} = (\phi, \theta, \psi)^T$ is required[1]. The nonlinear relationship can be found by a coefficient comparison with a skew symmetric matrix $\boldsymbol{S}$ that is determined by the rotation matrix ${}^I\boldsymbol{R}_B$ from (1) and its time derivative: $\boldsymbol{S}({}_{(B)}\boldsymbol{\omega}_Q) = ({}^I\boldsymbol{R}_B)^T\,{}^I\dot{\boldsymbol{R}}_B$. For the defined Euler angle rotation sequence, the kinematic euler equations result in

$$\dot{\boldsymbol{\Omega}} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} {}_{(B)}\boldsymbol{\omega}_Q. \tag{6}$$

The four angular velocities of the rotors are the real input variables of the quadrotor, but in order to simplify the control design, the artificial input vector $\boldsymbol{u} = (u_1, u_2, u_3, u_4)^T$ is defined as follows:

$$\boldsymbol{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} b & b & b & b \\ 0 & b & 0 & -b \\ b & 0 & -b & 0 \\ d & -d & d & -d \end{pmatrix} \begin{pmatrix} {\omega_1}^2 \\ {\omega_2}^2 \\ {\omega_3}^2 \\ {\omega_4}^2 \end{pmatrix} \tag{7}$$

The variable u_1 is equal to the sum of all rotor thrust forces and consequently matches the resulting lift-forces in hover flight. The variables u_2 and u_3 correspond to forces which result from a speed difference of two opposite motors leading to roll and pitch movements while the input variable u_4 can be interpreted as a yaw moment. Introducing the abbreviation $g(\boldsymbol{u}) = \omega_1 - \omega_2 + \omega_3 - \omega_4$, (2), (3), (6) and (7) can be used to formulate a nonlinear state space

[1]It should be mentioned that in many publications on quadrotors this dependency is simplified by implicitly assuming small angle approximation which results in an equality between the Euler rates $\dot{\boldsymbol{\Omega}} = (\dot{\phi}, \dot{\theta}, \dot{\psi})^T$ and the angular velocity vector ${}_B\boldsymbol{\omega}_Q$ (see for example [8], [11]).

system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ with the 12-dimensional state vector $\boldsymbol{x} = (x_1, ..., x_{12})^T = (\dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \omega_x, \omega_y, \omega_z, x, y, z)^T$ and the artificial input vector $\boldsymbol{u} = (u_1, u_2, u_3, u_4)^T$:

$$\dot{\boldsymbol{x}} = \begin{pmatrix} -(\cos x_4 \sin x_5 \cos x_6 + \sin x_4 \sin x_6)\frac{u_1}{m} \\ -(\cos x_4 \sin x_5 \sin x_6 - \sin x_4 \cos x_6)\frac{u_1}{m} \\ g - (\cos x_4 \cos x_5)\frac{u_1}{m} \\ x_7 + x_8 \sin x_4 \tan x_5 + x_9 \cos x_4 \tan x_5 \\ x_8 \cos x_4 - x_9 \sin x_4 \\ \frac{\sin x_4}{\cos x_5} x_8 + \frac{\cos x_4}{\cos x_5} x_9 \\ x_8 x_9 \frac{J_y - J_z}{J_x} - \frac{J_r}{J_x} x_8 g(\boldsymbol{u}) + \frac{L}{J_x} u_2 \\ x_7 x_9 \frac{J_z - J_x}{J_y} + \frac{J_r}{J_y} x_7 g(\boldsymbol{u}) + \frac{L}{J_y} u_3 \\ x_7 x_8 \frac{J_x - J_y}{J_z} + \frac{1}{J_z} u_4 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{8}$$

However, this Euler angles-based representation suffers from the so called gimbal lock problematic which in this case results in a singular configuration for pitch angles of $\theta = \pm 90°$ (division by zero).

This problem can be overcome by using a quaternion-based description of the system [20]. The normalized quaternion $\boldsymbol{q} = (q_0, q_1, q_2, q_3)^T$ describes a one-axis rotation by the rotation angle σ around the so-called euler axis with unit vector $\boldsymbol{e} = (e_1, e_2, e_3)^T$ and is defined as $\boldsymbol{q} = \begin{pmatrix} \cos\frac{\sigma}{2} \\ \boldsymbol{e} \sin\frac{\sigma}{2} \end{pmatrix}$ [20]. To adapt the derivation, the relation between the quaternion $\boldsymbol{q}$ and the rotation matrix ${}^I\boldsymbol{R}_B$ from (1) is required. It can be formulated according to [14] as

$${}^I\boldsymbol{R}_B = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}. \tag{9}$$

Moreover, the quaternion differential equation

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \bullet \left(\begin{pmatrix} 0 \\ {}_{(B)}\boldsymbol{\omega}_Q \end{pmatrix} \right) \tag{10}$$

has to be solved instead of the kinematic euler equations (6). The operator " $\bullet$ " describes the common quaternion multiplication which can also be formulated as a matrix-vector multiplication [14]:

$$\dot{\boldsymbol{q}} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{11}$$

Using (2), (3), (9) and (11), it is possible to replace (8) with an equivalent state space system on quaternion basis with the now 13-dimensional state vector $\boldsymbol{x} = (x_1, ..., x_{13})^T = (\dot{x}, \dot{y}, \dot{z}, q_0, q_1, q_2, q_3, \omega_x, \omega_y, \omega_z, x, y, z)^T$ as follows:

$$\dot{\boldsymbol{x}} = \begin{pmatrix} -(2x_5x_7 + 2x_4x_6)\frac{u_1}{m} \\ -(2x_6x_7 - 2x_4x_5)\frac{u_1}{m} \\ g - (x_4^2 - x_5^2 - x_6^2 + x_7^2)\frac{u_1}{m} \\ \frac{1}{2}(-x_5x_8 - x_6x_9 - x_7x_{10}) \\ \frac{1}{2}(x_4x_8 - x_7x_9 + x_6x_{10}) \\ \frac{1}{2}(x_7x_8 + x_4x_9 - x_5x_{10}) \\ \frac{1}{2}(-x_6x_8 + x_5x_9 + x_4x_{10}) \\ x_9x_{10}\frac{J_y - J_z}{J_x} - \frac{J_r}{J_x}x_9 g(\boldsymbol{u}) + \frac{L}{J_x}u_2 \\ x_8x_{10}\frac{J_z - J_x}{J_y} + \frac{J_r}{J_y}x_8 g(\boldsymbol{u}) + \frac{L}{J_y}u_3 \\ x_8x_9\frac{J_x - J_y}{J_z} + \frac{1}{J_z}u_4 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{12}$$

This singularity-free model serves as the basis for all simulations in this paper. It should be mentioned at this point that in order to facilitate the specification of the initial orientation as well as the later interpretation of the quaternion-based representation, it is useful to have two conversion formulas at hand which help converting quaternions to Euler angles and vice versa [14]:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \arctan_2(2(q_2q_3 + q_0q_1), (q_0^2 - q_1^2 - q_2^2 + q_3^2)) \\ -\arcsin(2(q_1q_3 - q_0q_2)) \\ \arctan_2(2(q_1q_2 + q_0q_3), (q_0^2 + q_1^2 - q_2^2 - q_3^2)) \end{pmatrix} \tag{13}$$

$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} - \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \end{pmatrix} \tag{14}$$

The hover state $\boldsymbol{\Omega} = \boldsymbol{0}$, for example, equals an initial quaternion of $\boldsymbol{q} = (\pm 1, 0, 0, 0)^T$. The parameter values of the quadrotor used for simulations are summarized in table I. They are based on either measurements on a real system that is currently under development (m, b, d, L) or extracted from appropriate CAD-models (J_x, J_y, J_z, J_r).

TABLE I
QUADROTOR PARAMETERS

parameter	abbreviation	value	unit
mass	m	0.58	kg
lever length	L	0.25	m
roll inertia	J_x	0.01	kg m^2
pitch inertia	J_y	0.01	kg m^2
yaw inertia	J_z	0.02	kg m^2
rotor inertia	J_r	3.8e-05	kg m^2
drag factor	d	2.82e-07	kg m^2
thrust factor	b	1.55e-05	kg m

III. DISCRETE-TIME INTEGRAL STATE SPACE STABILIZATION CONTROL

This section presents the control concept as well as the design of a nonadaptive integral state space controller for attitude stabilization which will serve as a basis for the adaptive schemes in Section IV. The goal of the control structure, which is sketched in Fig. 2, is to stabilize the quadrotor and to allow

free movement in space by a selection of appropriate reference signals. Controlled variables are the height z, the roll angle ϕ,

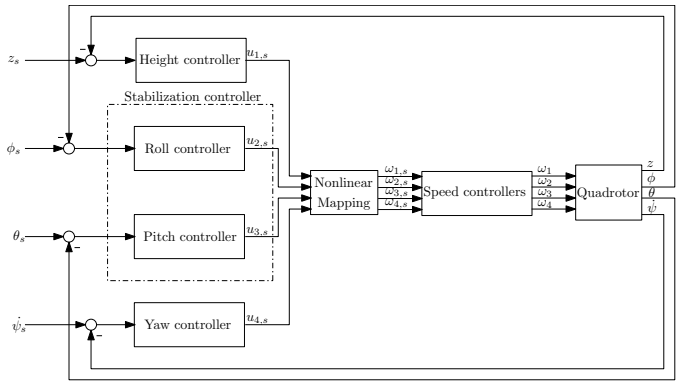

Fig. 2. Control concept

the pitch angle θ and the yaw rate $\dot{\psi}$. With this selection, any position in space can be reached. However, it is not possible to control an arbitrary combination of position and orientation due to the underactuated system. This can be explained to the effect that a nonzero roll or pitch angle always forces a translation of the quadrotor. The relationship is used here directly to initiate position changes by selecting appropriate reference angles. These can be provided either by a hand-operated remote control or a superimposed flight controller. The output variables $\boldsymbol{u_s} = (u_{1,s}, u_{2,s}, u_{3,s}, u_{4,s})^T$ of the four controllers are mapped to reference variables of four minor motor speed control loops which are considered sufficiently fast to neglect their dynamics. Consequently, in the rest of this paper we assume that $\boldsymbol{u} = \boldsymbol{u_s}$ holds. The mapping is realized by inverting (7), which is possible for $b \neq d$. In order to realize the concept only a subset of the mentioned quadrotor state equations are needed. By neglecting gyroscopic torques and linearizing the rotatory system around the hover state, we optain the simplified differential equations for the stabilization control loops and their corresponding Laplace transfer functions

$$\ddot{\phi} = \frac{L}{J_x} u_2 \circ\!\!-\!\!\bullet F_\phi(s) = \frac{\phi(s)}{U_2(s)} = \frac{L}{J_x s^2}, \quad (15a)$$

$$\ddot{\theta} = \frac{L}{J_y} u_3 \circ\!\!-\!\!\bullet F_\theta(s) = \frac{\theta(s)}{U_3(s)} = \frac{L}{J_y s^2}, \quad (15b)$$

which result in double-integrating systems. Due to the similar roll and pitch dynamics, only the roll control is further explained[2]. Equation (15a) can also be formulated as an equivalent continuous-time state space system with the reduced state vector $\boldsymbol{x} = (\phi, \dot{\phi})^T$:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}}_{\boldsymbol{A}} \boldsymbol{x}(t) + \underbrace{\begin{pmatrix} 0 \\ \frac{L}{J_x} \end{pmatrix}}_{\boldsymbol{b}} u_2(t) \\ \phi(t) &= \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{\boldsymbol{c}^T} \boldsymbol{x}(t) \end{aligned} \quad (16)$$

By applying the conversion formulas $\boldsymbol{A}_d = \boldsymbol{\Phi}(T)$ and $\boldsymbol{b}_d = \int_0^T \boldsymbol{\Phi}(T-\tau)\boldsymbol{b} d\tau$ with the state-transition matrix $\boldsymbol{\Phi}(t) = \mathcal{L}^{-1}\left\{(\boldsymbol{I}s - \boldsymbol{A})^{-1}\right\}$ (see for example [15]) and sample time T to (16), the equivalent discrete-time state space system can be formulated as follows:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \underbrace{\begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}}_{\boldsymbol{A_d}} \boldsymbol{x}(k) + \underbrace{\begin{pmatrix} \frac{T^2 L}{2 J_x} \\ \frac{TL}{J_x} \end{pmatrix}}_{\boldsymbol{b}_d} u_2(k) \\ \phi(k) &= \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{\boldsymbol{c}^T} \boldsymbol{x}(k) \end{aligned} \quad (17)$$

As a basis, we use two decoupled state space controllers for roll and pitch angles. Although the system itself has double-integrating behavior, we extend the controllers by a discrete-time integrator of the form $F_I(z) = \frac{k_I}{z-1}$ with gain k_I as shown in Fig. 3. The reason for this modification is the fact

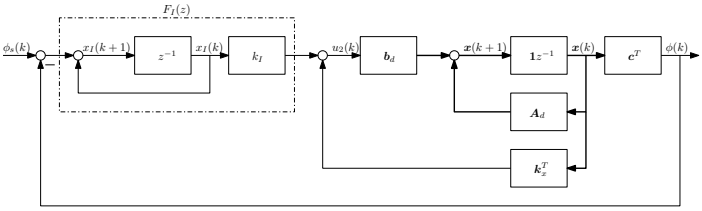

Fig. 3. Integral state-space roll stabilization control loop

that only then it is possible to compensate for (constant) disturbances that act upon the input side of the controlled system. Indeed these disturbances occur in reality due to blade flapping effects [13]. In order to design the controller parameters ${\boldsymbol{k_x}}^T$ and k_I, system (17) is enhanced by the integrator, and a pole placement operation is realized by using Ackermann's formula [16], so that all poles of the closed-loop system lie at the real locations $z_i = 0.9$ ($T = 0.01\,\mathrm{s}$). This choice results in a good compromise between actuator stress and dynamics. Fig. 4 shows the stabilization of the quadrotor with initial values $\phi(0) = \theta(0) = 70°$ (all other initial values are zero) as well as an input disturbance regulation at $t = 1.2\,\mathrm{s}$. One can see that the fairly basic linear controllers are able to successfully stabilize the system without big overshoots even under these extreme initial orientation values with strong coupling between the axes. Furthermore, a strong simultaneous positive constant load disturbance momentum of $0.25\,\mathrm{Nm}$ acts in roll and pitch direction from time $t = 1.2\,\mathrm{s}$ on. This corresponds to disturbance values of $u_{2\mathrm{d}} = u_{3\mathrm{d}} = 1\,\mathrm{N}$ for the artificial input variables and thus 50% of the saturation limits. Despite this large disturbance, the

[2]Height and yaw control are not in the focus of this paper, but it should be mentioned for completeness that the height control is composed of a discrete-time PDT_1-controller with transfer function $G_z(z) = \frac{-70(z-0.99)}{z-0.7}$ combined with a PT_1-prefilter $V_z(z) = \frac{0.02z}{z-0.98}$ and weight force compensation while the yaw rate controller is a simple gain $k_p = 0.1$ combined with a prefilter of the same type. Set points of these two controllers are zero. Furthermore, realistic limits have been imposed on the artificial input variables, so that $u_1 \in [1\,\mathrm{N}, 15\,\mathrm{N}]$, $u_2 \in [-2\,\mathrm{N}, 2\,\mathrm{N}]$, $u_3 \in [-2\,\mathrm{N}, 2\,\mathrm{N}]$, $u_4 \in [-0.05\,\mathrm{Nm}, 0.05\,\mathrm{Nm}]$ as well as appropriate anti-windup extensions.

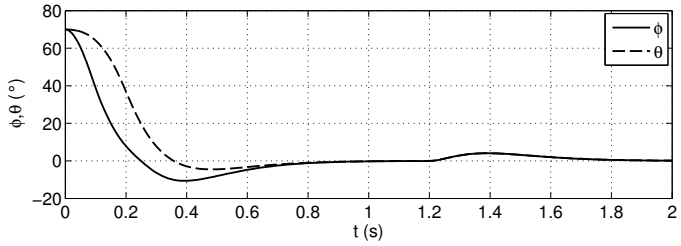

Fig. 4. Stabilization and disturbance rejection of the integral state control

angle control deviation only increases to a maximum of $4°$ and is compensated completely by the additional integral term. The simulation underlines the effectiveness and quality of the state controller. However, the fixed control parameters cannot cope with varying system inertias. Fig. 5 shows the step response of the roll controller mentioned above, which is designed for the nominal inertia $J_x = 0.01\,\mathrm{kg\,m^2}$ when the system actually has twice and four times that inertia respectively. While an increase to the double value is unproblematic due

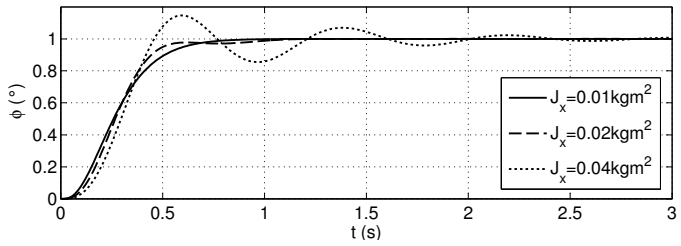

Fig. 5. Step response of nonadaptive control loop for different inertias

to the robustness of the controller, the oscillation tendency is much more pronounced if the fourfold inertia is applied. This would disturb video recordings from a quadrotor platform tremendously, for example, so that an adaptation seems vital in this case. Furthermore, an adaptation makes sense if the inertia tensor is completely unknown which will render a standard control design impossible. A self-tuning controller avoids the complex measurement or calculation of the inertias and will facilitate first-time operation. Therefore, two adaptive schemes are proposed to solve the in hand deficits of a standard quadrotor control design.

IV. Adaptive control of a quadrotor

A. Model Identification Adaptive Control (MIAC)

In the first case, adaptation is realized for the integral state controller described in Section III. Roll and pitch inertias are estimated by two parallel running Recursive Least-Squares (RLS) estimators with exponential forgetting in closed-loop combined with an online controller synthesis. The proposed parallel operation is valid for small angles – resulting in a decoupling of the axes – and a diagonally dominant inertia tensor[3]. Input variables of the estimators are u_2 and u_3 as well as the actual roll and pitch angles as can be seen in Fig. 6. In

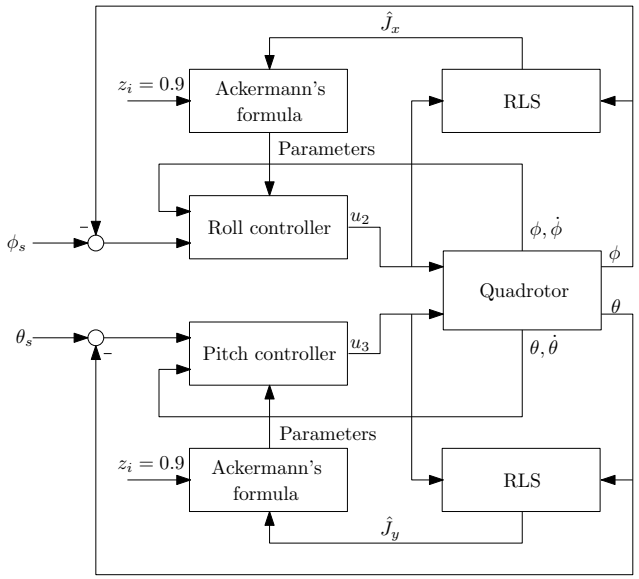

Fig. 6. Model identification adaptive stabilization control loops

order to calculate u_2 and u_3, the thrust factors of the propellers have to be known. This condition is considered to be fulfilled. The online controller synthesis is done using Ackermann's formula [16]. As a design criterion the poles of the closed-loop system should stay at the optimal location $z_i = 0.9$ ($T = 0.01\,\mathrm{s}$) independent of the inertias. This should result in a non-oscillating transient step response. For estimating the parameter vector $\hat{\boldsymbol{\Theta}} = (\hat{a}_1, ..., \hat{a}_m, \hat{b}_1, ..., \hat{b}_m)^T$ of a Z-transfer function with dead time d of the form

$$F(z) = \frac{Y(z)}{U(z)} = \frac{b_1 z^{-1} + ... + b_m z^{-m}}{1 + a_1 z^{-1} + ... + a_m z^{-m}} z^{-d}, \qquad (18)$$

the recursive least squares estimation is given by

$$\begin{aligned} \hat{\boldsymbol{\Theta}}(k+1) &= \hat{\boldsymbol{\Theta}}(k) + \boldsymbol{\gamma}(k)\left(y(k+1) - \boldsymbol{\psi}^T(k+1)\hat{\boldsymbol{\Theta}}(k)\right), \\ \boldsymbol{\gamma}(k) &= \frac{\boldsymbol{P}(k)\boldsymbol{\psi}(k+1)}{\boldsymbol{\psi}^T(k+1)\boldsymbol{P}(k)\boldsymbol{\psi}(k+1) + \lambda}, \\ \boldsymbol{P}(k+1) &= \frac{\left(\boldsymbol{I} - \boldsymbol{\gamma}(k)\boldsymbol{\psi}^T(k+1)\right)\boldsymbol{P}(k)}{\lambda}, \end{aligned} \qquad (19)$$

with the correction vector $\boldsymbol{\gamma}(k)$, the regression vector $\boldsymbol{\psi}(k) = (-y(k-1), ..., -y(k-m), u(k-d-1), ..., u(k-d-m))^T$, the forgetting factor λ and the covariance matrix $\boldsymbol{P}$ [17]. Adapting to our case, we first convert (15a) to discrete-time by applying the zero-order hold conversion

$$F_{\mathrm{zoh}}(z) = \frac{z-1}{z} Z\left[\mathcal{L}^{-1}\left[\frac{1}{s}F(s)\right]_{t=kT}\right] \qquad (20)$$

in which Z represents the Z-transform, $\mathcal{L}^{-1}$ conforms to the inverse Laplace-transform of the continuous-time process, k is equal to a nonnegative integer and T refers to the

[3]This is only a minor constraint in practice as one will always aim to place additional loads symmetrically to the structure.

sampling time [16]. The corresponding discrete-time system can consequently be written in form of the Z-transfer function

$$F_\phi(z) = \frac{\frac{T^2 L}{2J_x}(z^{-1} + z^{-2})}{1 - 2z^{-1} + z^{-2}}. \tag{21}$$

A coefficient comparison with (18) reveals that the denominator coefficients a_i are independent of the inertia and that J_x can be calculated from $\hat{b}_1$ as well as $\hat{b}_2$. In our case the inertia is therefore calculated from the more accurately estimated average as $\hat{J}_x = T^2 L/(\hat{b}_1 + \hat{b}_2)$. Fig. 7 shows the simulation results for the proposed MIAC in comparison to the nonadaptive control loops. During the first 5 seconds, the

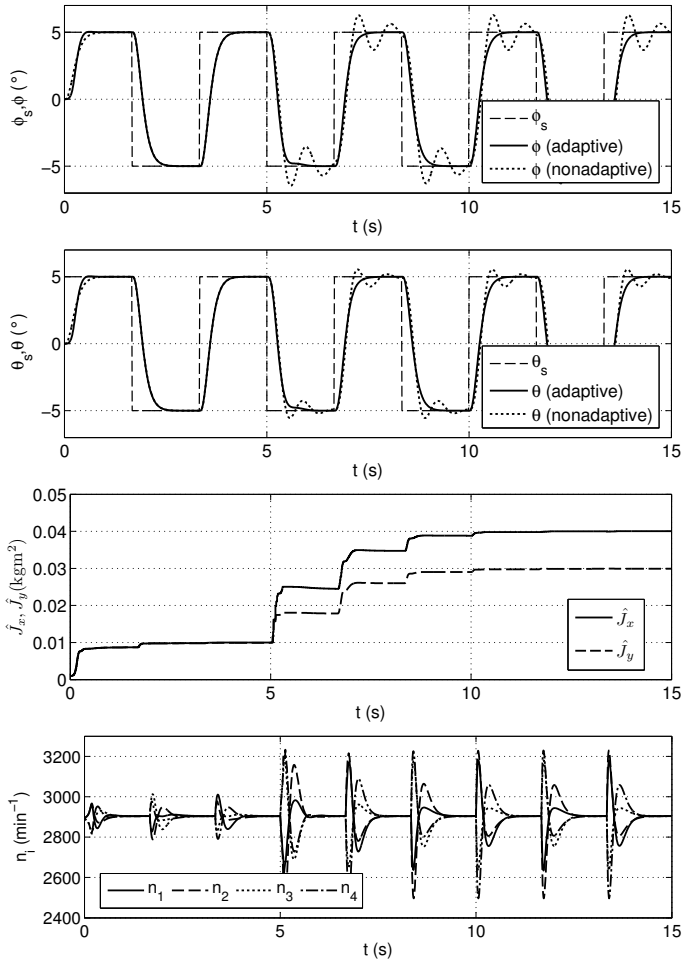

Fig. 7. Self-tuning and behavior of MIAC during inertia changes

inertias have their nominal values $J_x = J_y = 0.01\,\mathrm{kg\,m^2}$ for which the nonadaptive controllers were optimized. At $t = 5\,\mathrm{s}$ they are instantly increased to $J_x = 0.04\,\mathrm{kg\,m^2}$ and $J_y = 0.03\,\mathrm{kg\,m^2}$ which can be considered a worst case scenario of a sudden load change during flight. Plotted are the reference values ϕ_s and θ_s, the controlled variables ϕ and θ, the inertia adaptation as well as the drive speeds n_i. For the excitation signals, square waves of amplitude 5° are chosen as reference variables for the roll and pitch control loop. This can be considered sensible because it will result in a self-adaptation of the stabilization controllers under minor oscillatory translational movements of the quadrotor. The initial controller parameters are $\boldsymbol{k}^T(0) = \mathbf{0}, k_I(0) = 0$, i.e. considered completely unknown. Initial numerator parameters are fixed at $\hat{b}_1(0) = \hat{b}_2(0) = 1.25 \cdot 10^{-2}$ which corresponds to only 10% of the actual inertia and simulates a strong misestimation of the initial parameters. The initial values of the denominator are set to the right values $\hat{a}_1(0) = -2$ and $\hat{a}_2(0) = 1$ which are known because of the system structure. The forgetting factor is $\lambda = 0.99$, initial covariance matrices are diagonal matrices with $\boldsymbol{P}(0) = \mathrm{diag}(100, 100)$. The estimators rapidly converge to the correct nominal inertia values and the control quality is equal to that of the optimal adjusted fixed state controller. Under load changes, the adaptation displays its potential as it shows an almost instantaneous optimal step response without overshoot in comparison to the nonadaptive case. The drives' speed values clearly reveal the swaying of the quadrotor as well as the increased required actuating values to satisfy the dynamic specifications under larger inertias. If the adaptation is only used for self-tuning, the estimator can of course be deactivated after convergence and the controller parameters remain fixed.

B. Model Reference Adaptive Control (MRAC)

As a second variant, a Model Reference Adaptive Control scheme as sketched as in Fig 8 is proposed. There are two

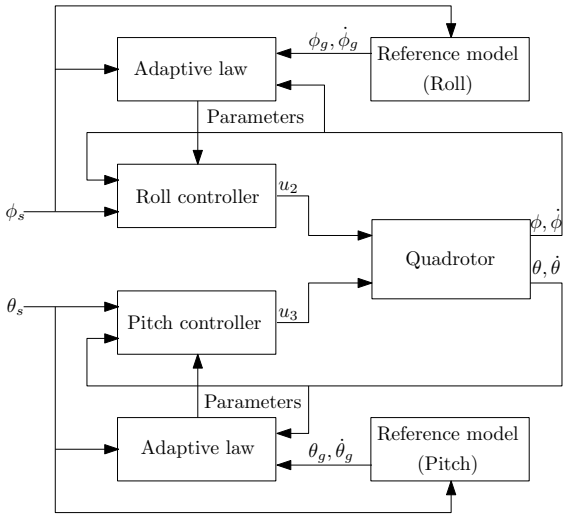

Fig. 8. Model reference adaptive stabilization control loops

equivalent reference models, one for roll and one for pitch movements, which represent the desired closed-loop dynamics of the quadrotor[4]. A Lyapunov-based adaptive law[5], that guarantees the asymptotic stability in case of linear systems, estimates the parameters of a state space controller of the form

$$u_2 = -\hat{\boldsymbol{k}}^T \boldsymbol{x} + \hat{v}\phi_s, \tag{22}$$

[4]Only the roll control loop is described further, the pitch loop behaves accordingly.

[5]The derivation can be found in the appendix and is roughly based on [19].

with feedback gain $\hat{\boldsymbol{k}}^T$, prefilter $\hat{v}$ and state vector $\boldsymbol{x} = (\phi, \dot{\phi})^T$ with the help of the reference variable ϕ_s and state errors $\boldsymbol{\epsilon} = \boldsymbol{x} - \boldsymbol{x}_g$ according to

$$\hat{\boldsymbol{k}}^T = \int_0^t \boldsymbol{b}_g^T \boldsymbol{P} \boldsymbol{\epsilon} \boldsymbol{x}^T d\tau + \hat{\boldsymbol{k}}^T(0), \tag{23a}$$

$$\hat{v} = -\int_0^t \boldsymbol{b}_g^T \boldsymbol{P} \boldsymbol{\epsilon} \phi_s d\tau + \hat{v}(0). \tag{23b}$$

The matrix $\boldsymbol{P}$ in (23) is symmetric, positive definite and has to fulfill the matrix Lyapunov equation

$$\boldsymbol{P}\boldsymbol{A}_g + \boldsymbol{A}_g^T \boldsymbol{P} = -\boldsymbol{Q} \tag{24}$$

for symmetric, positive definite matrices $\boldsymbol{Q}$ which act as a design parameter that influences the transient of the adaptation itself[6] [19]. The stable linear roll reference model is given in the same canonical form as (16) in order to fulfill the compatibility criteria of perfect model-following automatically:

$$\dot{\boldsymbol{x}}_g = \boldsymbol{A}_g \boldsymbol{x} + \boldsymbol{b}_g \phi_s = \begin{pmatrix} 0 & 1 \\ -100 & -20 \end{pmatrix} \boldsymbol{x}_g + \begin{pmatrix} 0 \\ 100 \end{pmatrix} \phi_s \tag{25}$$

All eigenvalues are consequently fixed to -10 which equals eigenvalues of 0.9 of a discrete-time system for $T = 0.1\,\mathrm{s}$. Thus the goal of the adaptation is to enforce the same closed-loop dynamics as the nonadaptive control of Section III in order to facilitate the comparison. Fig. 9 shows the self-tuning as well as the adaptation of the controller to changing system inertia in comparison to the nonadaptive version. During

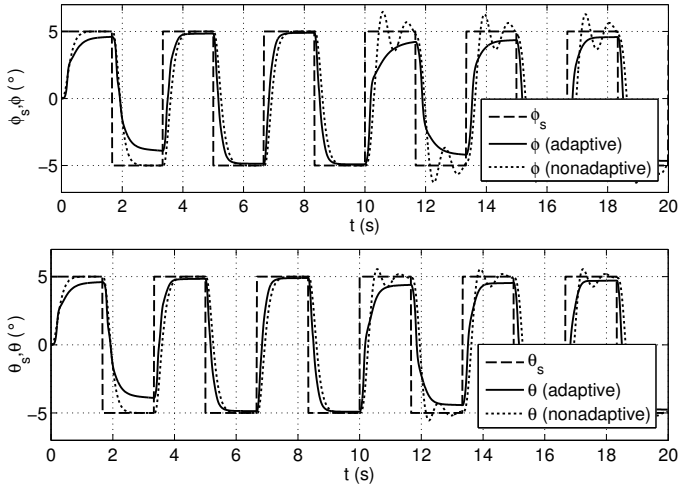

Fig. 9. Self-tuning and behavior of MRAC during inertia changes

the first 10 seconds the inertias have their nominal values $J_x = J_y = 0.01\,\mathrm{kg\,m^2}$ for which the nonadaptive controllers were optimized. At $t = 10\,\mathrm{s}$ they are instantly increased to $J_x = 0.04\,\mathrm{kg\,m^2}$ and $J_y = 0.03\,\mathrm{kg\,m^2}$ as in the simulation of Section IV-A. Plotted are once again the reference variables ϕ_s and θ_s and the controlled variables ϕ and θ. The initial controller parameters are $\boldsymbol{k}^T(0) = \boldsymbol{0}$, i.e. considered completely unknown, and $\boldsymbol{Q} = \mathrm{diag}(150, 150)$. The dynamics of the MRAC equals the dynamics of the nonadaptive controller and the self-tuning can be considered fulfilled after a short adaptation transient. Furthermore, the adaptation to the inertia parameter step works smoothly and the benefit in opposite to the nonadaptive controller becomes obvious. Disturbing oscillations are completely suppressed.

[6]This equation can be computed off-line.

V. Conclusion

Two adaptive control designs (MIAC and MRAC) have been applied to a quadrotor unmanned aerial vehicle in order to cope with unknown and varying system inertia. Numerical simulations based on a singularity-free nonlinear model in quaternion representation reveal their superiority over the fixed-parameter integral state controller designed in this work. Oscillations in the step response of the attitude stabilization control loops due to varying inertia are removed effectively by both schemes which is considered especially useful for the increasingly used modular quadrotor platforms with changing loads.

Appendix

The objective of the applied MRAC design method according to [19] is to choose an input vector $\boldsymbol{u} \in \mathbb{R}^p$ in a way that all signals in closed-loop remain bounded and that the states of the n-dimensional unknown system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$ with system matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and input matrix $\boldsymbol{B} \in \mathbb{R}^{n \times p}$ follow the states of a parallel reference model of the form $\dot{\boldsymbol{x}}_g = \boldsymbol{A}_g \boldsymbol{x}_g + \boldsymbol{B}_g \boldsymbol{w}$ with $\boldsymbol{A}_g \in \mathbb{R}^{n \times n}$ being a stable reference system matrix, $\boldsymbol{B}_g \in \mathbb{R}^{n \times p}$ the desired input matrix and $\boldsymbol{w} \in \mathbb{R}^p$ a bounded reference input vector. If $\boldsymbol{A}$ and $\boldsymbol{B}$ were known, a controller of the form $\boldsymbol{u} = -\boldsymbol{K}_o \boldsymbol{x} + \boldsymbol{V}_o \boldsymbol{w}$ with optimal feedback matrix $\boldsymbol{K}_o \in \mathbb{R}^{p \times n}$ and optimal prefilter matrix $\boldsymbol{V}_o \in \mathbb{R}^{p \times p}$ could be applied, which would result in the closed-loop behavior $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_o)\boldsymbol{x} + \boldsymbol{B}\boldsymbol{V}_o \boldsymbol{w}$. If one chooses $\boldsymbol{K}_o$ and $\boldsymbol{V}_o$ in a way that $\boldsymbol{A}_g = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_o$ and $\boldsymbol{B}_g = \boldsymbol{B}\boldsymbol{V}_o$ holds, then the dynamics of the controlled system would equal that of the reference system. Due to $\boldsymbol{A}$ and $\boldsymbol{B}$ being unknown, the (time-dependent) controller parameters $\hat{\boldsymbol{K}}$ and $\hat{\boldsymbol{V}}$ have to be estimated by an appropriate adaptive law. With the state error $\boldsymbol{\epsilon} = \boldsymbol{x} - \boldsymbol{x}_g$ as well as the parameter errors $\Delta\boldsymbol{K} = \hat{\boldsymbol{K}} - \boldsymbol{K}_o$ and $\Delta\boldsymbol{V} = \hat{\boldsymbol{V}} - \boldsymbol{V}_o$, the error differential equation

$$\dot{\boldsymbol{\epsilon}} = \boldsymbol{A}_g \boldsymbol{\epsilon} + \boldsymbol{B}_g \boldsymbol{V}_o^{-1}(-\Delta\boldsymbol{K}\boldsymbol{x} + \Delta\boldsymbol{V}\boldsymbol{w}) \tag{26}$$

can be derived which itself can be considered a nonautonomous system. If one finds a Lyapunov function V, then the stability of the equilibrium $\boldsymbol{\epsilon} = \Delta\boldsymbol{K} = \Delta\boldsymbol{V} = \boldsymbol{0}$ is guaranteed. According to [19], the positive definite function

$$V(\boldsymbol{\epsilon}, \Delta\boldsymbol{K}, \Delta\boldsymbol{V}) = \boldsymbol{\epsilon}^T \boldsymbol{P} \boldsymbol{\epsilon} + \mathrm{tr}(\Delta\boldsymbol{K}^T \boldsymbol{\Gamma} \Delta\boldsymbol{K} + \Delta\boldsymbol{V}^T \boldsymbol{\Gamma} \Delta\boldsymbol{V}) \tag{27}$$

in which $\boldsymbol{P}\boldsymbol{A}_g + \boldsymbol{A}_g^T \boldsymbol{P} = -\boldsymbol{Q}$ holds for arbitrary positive definite matrices $\boldsymbol{Q}$ can be considered a Lyapunov candidate. Introducing the new quantity $\boldsymbol{\Gamma}^{-1} = \boldsymbol{V}_o \mathrm{sgn}(l)$ with $l > 0$ if

$\boldsymbol{V}_o$ positive definite and $l < 0$ if $\boldsymbol{V}_o$ negative definite, the time derivative of (27) along the trajectories of (26) results in

$$\begin{aligned}\dot{V} &= \dot{\boldsymbol{\epsilon}}^T\boldsymbol{P}\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^T\boldsymbol{P}\dot{\boldsymbol{\epsilon}} + \mathrm{tr}(\Delta\dot{\boldsymbol{K}}^T\boldsymbol{\Gamma}\Delta\boldsymbol{K} + \Delta\boldsymbol{K}^T\boldsymbol{\Gamma}\Delta\dot{\boldsymbol{K}} \\ &\quad + \Delta\dot{\boldsymbol{V}}^T\boldsymbol{\Gamma}\Delta\boldsymbol{V} + \Delta\boldsymbol{V}^T\boldsymbol{\Gamma}\Delta\dot{\boldsymbol{V}}) \\ &= -\boldsymbol{\epsilon}^T\boldsymbol{Q}\boldsymbol{\epsilon} + 2\mathrm{tr}(\Delta\boldsymbol{K}^T\boldsymbol{\Gamma}(\Delta\dot{\boldsymbol{K}} - \boldsymbol{B}_g^T\boldsymbol{P}\boldsymbol{\epsilon}\boldsymbol{x}^T\mathrm{sgn}(l)) \\ &\quad + \Delta\boldsymbol{V}^T\boldsymbol{\Gamma}(\Delta\dot{\boldsymbol{V}} + \boldsymbol{B}_g^T\boldsymbol{P}\boldsymbol{\epsilon}\boldsymbol{w}^T\mathrm{sgn}(l)))\end{aligned} \tag{28}$$

after a longer calculation. Choosing the adaptive law according to [19] as

$$\Delta\dot{\boldsymbol{K}} = \dot{\hat{\boldsymbol{K}}} - \dot{\boldsymbol{K}}_o = \dot{\hat{\boldsymbol{K}}} = \boldsymbol{B}_g^T\boldsymbol{P}\boldsymbol{\epsilon}\boldsymbol{x}^T\mathrm{sgn}(l), \tag{29a}$$

$$\Delta\dot{\boldsymbol{V}} = \dot{\hat{\boldsymbol{V}}} - \dot{\boldsymbol{V}}_o = \dot{\hat{\boldsymbol{V}}} = -\boldsymbol{B}_g^T\boldsymbol{P}\boldsymbol{\epsilon}\boldsymbol{w}^T\mathrm{sgn}(l), \tag{29b}$$

then the trace in (28) vanishes and we get $\dot{V} = -\boldsymbol{\epsilon}^T\boldsymbol{Q}\boldsymbol{\epsilon}$, which renders V a valid Lyapunov function because of its negative semidefinite derivative. Consequently, the equilibrium state of the error equation is stable in the sense of Lyapunov, but not necessarily asymptotically stable. However, with the help of the Lemma of Barbalat [21], it can be easily verified that $\lim_{t\to\infty}\dot{V} = \lim_{t\to\infty}(-\boldsymbol{\epsilon}^T(t)\boldsymbol{Q}\boldsymbol{\epsilon}(t)) = 0$ holds. Consequently the state errors decline to zero independent of the initial parameters and the adaptive control system is said to be globally asymptotically stable.

REFERENCES

[1] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, 2007.

[2] D. Lee, H. J. Kim, and S. Sastry, "Feedback Linearization vs. Adaptive Sliding Mode Control for a Quadrotor Helicopter," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 419–428, jun 2009.

[3] A. Tayebi and S. McGilvray, "Attitude stabilization of a four-rotor aerial robot," in *Proc. of the IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, dec 2004, pp. 1216–1221.

[4] A. Tayebi and S. McGilvray, "Attitude Stabilization of a VTOL Quadrotor Aircraft," in *IEEE Transactions on Control Systems Technology*, vol. 14, may 2006, pp. 562–571.

[5] S. Bouabdallah and R. Siegwart, "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor," in *Proc. of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, apr 2005, pp. 2247–2252.

[6] H. Voos, "Nonlinear State-Dependent Riccati Equation Control of a Quadrotor UAV," in *Proc. of the IEEE International Conference on Control Applications*, Munich, Germany, oct 2006.

[7] T. Madani and A. Benallegue, "Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles," in *Proc. of the American Control Conference*, New York, USA, jul 2007, pp. 5887–5892.

[8] H. Voos, "Entwurf eines Flugreglers für ein vierrotoriges unbemanntes Fluggerät (Control Systems Design for a Quadrotor UAV)," *Automatisierungstechnik*, vol. 57, no. 9, pp. 423–431, 2009.

[9] A. Mokhtari, A. Benallegue, and B. Daachi, "Robust Feedback Linearization and GH_∞ Controller for a Quadrotor Unmanned Aerial Vehicle," *Journal of Electrical Engineering*, vol. 57, no. 1, pp. 20–27, 2006.

[10] G. Raffo, M. Ortega, and F. Rubio, "An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, jan 2010.

[11] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and Control of an Indoor Micro Quadrotor," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2004.

[12] A. Sanca, P. Alsina, and J. Cerqueira, "Dynamic Modelling of a Quadrotor Aerial Vehicle with Nonlinear Inputs," in *Proc. of the IEEE Latin American Robotic Symposium*, oct 2008, pp. 143 –148.

[13] G. M. Hoffmann, H. Huang, S. L. Waslander *et al.*, "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment," in *Proc. of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, South Carolina, 2007.

[14] J. Wendel, *Integrierte Navigationssysteme. Sensordatenfusion, GPS und Inertiale Navigation*. Munich: Oldenbourg Wissenschaftsverlag, 2007.

[15] J. Lunze, *Regelungstechnik 2. Mehrgrößenregelung, Digitale Regelungstechnik*, 4th ed. Berlin/Heidelberg/New York: Springer-Verlag, 2006.

[16] J. Ackermann, *Abtastregelung*, 3rd ed. Berlin/Heidelberg/New York: Springer-Verlag, 1988.

[17] R. Isermann, *Digitale Regelsysteme. Stochastische Regelungen, Mehrgrößenregelungen, Adaptive Regelungen, Anwendungen*, 2nd ed. Berlin/Heidelberg/New York: Springer-Verlag, 1987, vol. 2.

[18] J. Li and Y. Li, "Dynamic Analysis and PID Control for a Quadrotor," in *Proc. of the IEEE International Conference on Mechatronics and Automation*, Beijing, China, aug 2011.

[19] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. New Jersey: Prentice-Hall, 1996.

[20] S. Winkler, "Zur Sensordatenfusion für integrierte Navigationssysteme unbemannter Kleinstflugzeuge," Ph.D. dissertation, Technische Universität Carolo-Wilhelmina zu Braunschweig, 2007.

[21] J. E. Slotine and W. Li, "Applied Nonlinear Control". New Jersey: Prentice-Hall, 1991.