

## ADAPTIVE FAULT-TOLERANT POSITION CONTROL OF A HEXACOPTER SUBJECT TO AN UNKNOWN MOTOR FAILURE

GUILLERMO P. FALCONÍ<sup>a,\*</sup>, JORG ANGELOV<sup>a</sup>, FLORIAN HOLZAPFEL<sup>a</sup>

<sup>a</sup>Institute of Flight System Dynamics (FSD)  
Technical University of Munich (TUM), Boltzmannstrasse 15, 85748 Garching, Germany  
e-mail: {guillermo.falconi, jorg.angelov, florian.holzapfel}@tum.de

This paper presents a fault tolerant position tracking controller for a hexarotor system. The proposed controller has a cascaded structure composed of a position and an attitude control loop. The nominal controller is augmented by an adaptive control allocation which compensates for faults and failures within the propulsion system without reconfiguration of the controller. Simultaneously, it is able to implement a degraded control strategy which prioritizes specific control directions in the case of extreme degradation. The main contribution is a controller that is a step closer to application scenarios by including outdoor GPS-based flight tests, onboard computation and the handling of unknown degradation and failure of any rotor.

**Keywords:** adaptive control, fault tolerant control, fault recovery, MAV flight dynamics and control.

### 1. Introduction

The variety and complexity of missions for which multirotor systems are being envisioned is constantly increasing. According to this application-driven development, the requirements on performance and safety are becoming more demanding. In this context, fault tolerant control (FTC) has gained the interest of various researchers. In order to achieve fault tolerance, actuator redundancy is desired and therefore hexacopters and octocopters have been studied.

In this paper, the problem of controlled flight of a hexacopter in the event of unknown faults or failures within the actuation system is addressed. Although there are several approaches to fault tolerant position control of multirotors which are tested in simulation (Merheb *et al.*, 2015; Cen *et al.*, 2015), there are rather few experimental results which come close to the different application scenarios. Most of them are partial results showing only failure detection and identification (Freddi *et al.*, 2014), or only controller reconfiguration assuming that the fault has been already identified (Achtelik *et al.*, 2012; Du *et al.*, 2015; Schneider *et al.*, 2012; Santos *et al.*, 2015), or using external sensors for having an accurate position measurement (Amoozgar *et al.*, 2012; Dydek

*et al.*, 2010; Mueller and D’Andrea, 2014; Saied *et al.*, 2015; Vey and Lunze, 2016), or making computations off-board (Dydek *et al.*, 2013). Furthermore, in the special case of the hexarotor, the controllers are applied to different levels of actuator faults: only degradation (Heise *et al.*, 2014; Mühlegg *et al.*, 2015), the total failure of one rotor excluding a group of rotors (Vey and Lunze, 2016; Yang *et al.*, 2016), the total failure of any rotor (Falconí *et al.*, 2016) and the total failure of more than one rotor (Mueller and D’Andrea, 2014). The complexity increases with each of these cases because after a rotor failure the set of attainable forces and moments is noticeably reduced and saturations become an issue.

The main contribution of the paper is the fault tolerant position controller that is a step closer to application scenarios. This includes handling unknown degradation and failure of any rotor, outdoor GPS-based flights and onboard computation of the algorithms. In our previous work (Falconí *et al.*, 2016), we presented a fault tolerant attitude controller based on adaptive control allocation which consists of the estimation of the actuator control effectiveness and an optimal control allocation. The main advantages of this strategy are the following: (a) a unified controller handles the nominal, the degradation and the failure cases; (b) it exploits the full attainable moment set avoiding saturations whenever possible; (c)

\*Corresponding author

finally, it is possible to implement a degraded control strategy in order to prioritize specific virtual controls in the case of extreme degradation. This paper extends our previous results (Falconi et al., 2016) with mainly three contributions: (a) extension of the attitude controller to a position controller maintaining the fault tolerant properties; (b) improvement of the control effectiveness estimates by introducing the null space shaping approach; (c) outdoor GPS-based flight tests including failure scenarios. To the best of our knowledge, this paper presents the first outdoor flight tests of a position tracking controller of a hexarotor subject to an unknown failure of any of the six propellers. This is a main contribution towards the usability of multirotor systems for real-life applications.

The remainder of the paper is organized as follows. Section 2 describes the dynamics of the hexarotor. Section 3 presents the baseline controller, and includes the extension to a position tracking controller in Section 3.1. The adaptive control allocation is introduced in Section 4 and the null space shaping approach in Section 4.2. The flight tests are found in Section 5 and the results are summarized in Section 6.

**Notation.**  $\mathbb{R}$  denotes the set of real numbers,  $\mathbb{R}^n$  denotes the set of  $n \times 1$  real column vectors,  $\mathbb{R}^{n \times m}$  denotes the set of  $n \times m$  real matrices,  $(\cdot)^T$  denotes transpose,  $(\cdot)^{-1}$  denotes the inverse and  $(\cdot)^+$  denotes the pseudoinverse. In addition, we write  $\mathbf{I}_n \in \mathbb{R}^{n \times n}$  for the identity matrix,  $\mathbf{0}_n \in \mathbb{R}^{n \times n}$  for the zero matrix, and  $\text{tr}(\cdot)$  for the trace operator.

## 2. Plant description

For describing the dynamics of the hexacopter, we use an inertial frame  $I$  with the origin at the Earth's surface. The axes at the origin point towards the north, the east and downwards. The body-fixed frame  $B$  is defined as in Fig. 1, such that the origin is at the center of gravity. The translational dynamics can be written in the inertial frame  $I$  using Newton's second law and by neglecting drag forces and disturbances as follows

$$m\dot{\mathbf{v}} = \mathbf{t} + m\mathbf{g}, \quad (1)$$

where  $\mathbf{g} \in \mathbb{R}^3$  is the constant gravitational acceleration vector,  $m$  is the mass of the hexacopter,  $\mathbf{t} \in \mathbb{R}^3$  is the total propulsion force and  $\mathbf{v} \in \mathbb{R}^3$  is the velocity of the hexacopter with respect to the  $I$ -frame. Furthermore, the position kinematics are

$$\dot{\mathbf{x}} = \mathbf{v}, \quad (2)$$

where  $\mathbf{x} \in \mathbb{R}^3$  is the position in the  $I$ -frame.

The rotational dynamics are given in the body-fixed frame  $B$  by the Euler equation

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}_p, \quad (3)$$

where  $\boldsymbol{\omega} \in \mathbb{R}^3$  is the angular rate of the body-fixed frame relative to the inertial frame,  $\mathbf{J} \in \mathbb{R}^{3 \times 3}$  is the moment of inertia of the hexacopter and  $\mathbf{M}_p \in \mathbb{R}^3$  is the propulsion moment.

In order to describe the attitude of the hexarotor, we choose the thrust vector  $\mathbf{t} = -T\mathbf{z}_B$  as in the work of Falconi and Holzapfel (2013). Here,  $\mathbf{z}_B \in \mathbb{R}^3$  is the unit vector pointing along the body-fixed  $z$ -axis and  $T \in \mathbb{R}$  is the total thrust. The equation of motion is then given by (Falconi and Holzapfel, 2013)

$$\begin{aligned} \dot{\mathbf{t}} &= \mathbf{R}_{IB}\mathbf{T}_u(T) \begin{pmatrix} \boldsymbol{\omega}_x \\ \boldsymbol{\omega}_y \\ T \end{pmatrix}, \\ &= -\dot{T}\mathbf{z}_B + \mathbf{R}_{IB}\mathbf{T}_\omega(T)\boldsymbol{\omega}, \end{aligned} \quad (4)$$

where  $\mathbf{R}_{IB} \in SO^3$  is the rotation matrix which maps a vector from the  $B$ -frame to the  $I$ -frame. The matrices  $\mathbf{T}_\omega \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{T}_u \in \mathbb{R}^{3 \times 3}$  are given by

$$\begin{aligned} \mathbf{T}_\omega(T) &= \begin{bmatrix} 0 & -T & 0 \\ T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ \mathbf{T}_u(T) &= \begin{bmatrix} 0 & -T & 0 \\ T & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \end{aligned}$$

For a detailed derivation of the equations of motion, the reader is referred to the work of Falconi and Holzapfel (2013).

Note that  $T$  and  $\mathbf{M}_p$  can be seen as virtual controls of the system dynamics (1)–(4). The roll and pitch moments depend on the geometrical arrangement of the rotors. The yaw moment depends on the rotation direction of the rotors. Considering the configuration as in Fig. 1, it is possible to write their relationship to the angular velocities of the rotors as (Falconi et al., 2015)

$$\boldsymbol{\nu} := \begin{pmatrix} T \\ \mathbf{M}_p \end{pmatrix} = \begin{bmatrix} \mathbf{B}_{T\omega} \\ \mathbf{B}_{M\omega} \end{bmatrix} \mathbf{u} = \mathbf{B}_{\nu\omega} \mathbf{u}, \quad (5)$$

where  $\mathbf{u} = (\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2, \omega_5^2, \omega_6^2)^T$  is the vector of the squared rotor speeds and  $\mathbf{B}_{\nu\omega} \in \mathbb{R}^{4 \times 6}$  is defined as (Falconi et al., 2015)

$$\begin{bmatrix} k_T & k_T & k_T \\ -\frac{1}{2}lk_T & -lk_T & -\frac{1}{2}lk_T \\ \frac{\sqrt{3}}{2}lk_T & 0 & -\frac{\sqrt{3}}{2}lk_T \\ -k_M & k_M & -k_M \\ k_T & k_T & k_T \\ \frac{1}{2}lk_T & lk_T & \frac{1}{2}lk_T \\ -\frac{\sqrt{3}}{2}lk_T & 0 & \frac{\sqrt{3}}{2}lk_T \\ k_M & -k_M & k_M \end{bmatrix}.$$

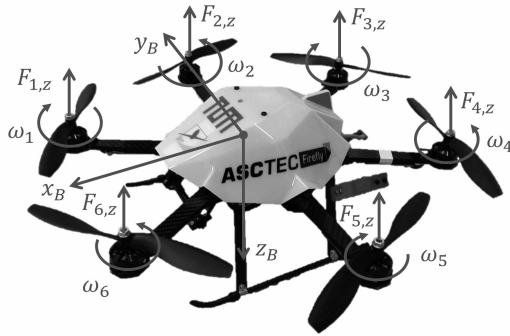


Fig. 1. Hexacopter.

The partitions of the matrix are  $\mathbf{B}_{T\omega} \in \mathbb{R}^{1 \times 6}$  and  $\mathbf{B}_{M\omega} \in \mathbb{R}^{3 \times 6}$ , respectively. The constants  $k_T, k_M > 0 \in \mathbb{R}$  are specific rotor parameters and  $l > 0 \in \mathbb{R}$  is the arm length. For a detailed derivation of the actuator model (5), refer to (Falconí *et al.*, 2015). As the matrix  $\mathbf{B}_{\nu\omega}$  has a full row rank, for a given desired  $\nu_d \in \mathbb{R}^4$  it is possible to find a function which maps it into the input space  $\mathbf{u} = \mathbf{u}(\nu_d)$  so that  $\nu = \mathbf{B}_{\nu\omega} \mathbf{u}(\nu_d) = \nu_d$ . Hence, we can design a control law for  $\nu$  and then allocate the commands to the different propellers.

### 3. Baseline controller

The presented controller is a position tracking controller where the pilot commands the position  $\mathbf{x}_c \in \mathbb{R}^3$  and the yaw rate  $\omega_{z,c} \in \mathbb{R}$ . It is composed of a baseline controller and an adaptive control allocation (ACA); see Fig. 2.

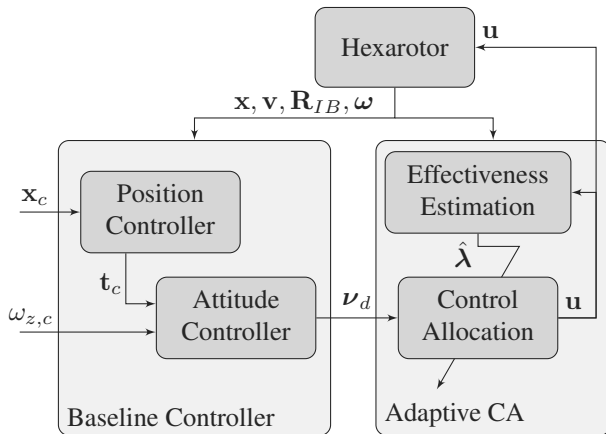


Fig. 2. Controller structure.

The baseline controller stabilizes the tracking error dynamics for the nominal case, i.e., the fault free case. The ACA is responsible for compensating modeling errors, disturbances and especially faults within the propulsion system. The baseline controller is similar

to that found in the works of Falconí and Holzapfel (2013; 2014) but with a cascaded structure composed of a position and an attitude control loop. The main assumption of the cascade is that the attitude loop is much faster than the position loop allowing a separated analysis of the subsystems. The attitude state, given by the thrust vector  $\mathbf{t}$ , can then be seen as the input of the position subsystem and the computed command  $\mathbf{t}_c \in \mathbb{R}^3$  is the tracking signal of the attitude subsystem as depicted in Fig. 2. The extension of the attitude controller from the work of Falconí *et al.* (2016) to a position controller which maintains the fault tolerant properties is a main contribution of this paper. In the following sections the two controllers are derived.

**3.1. Position control loop.** In a first step, the desired position  $\mathbf{x}_d \in \mathbb{R}^3$  is generated from the commanded position  $\mathbf{x}_c$  using a second order reference model

$$\begin{pmatrix} \dot{\mathbf{x}}_d \\ \dot{\mathbf{v}}_d \end{pmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ -\mathbf{K}_p & -\mathbf{K}_d \end{bmatrix} \begin{pmatrix} \mathbf{x}_d \\ \mathbf{v}_d \end{pmatrix} + \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{K}_p \end{bmatrix} \mathbf{x}_c. \quad (6)$$

Here,  $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$  is the identity matrix,  $\mathbf{0}_3 \in \mathbb{R}^{3 \times 3}$  is the zero matrix and  $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{3 \times 3}$  are positive definite matrices such that the system is exponentially stable. Note that  $\mathbf{x}_d \in \mathbb{C}^2$  for a continuous command  $\mathbf{x}_c$  and that (6) has a zero steady-state error  $\mathbf{x}_c - \mathbf{x}_d$ .

In order to track the states of the reference model (6), the zero equilibrium of the dynamics of the position tracking error

$$\mathbf{e}_p := \begin{pmatrix} \mathbf{e}_x \\ \mathbf{e}_v \end{pmatrix} = \begin{pmatrix} \mathbf{x}_d - \mathbf{x} \\ \mathbf{v}_d - \mathbf{v} \end{pmatrix}$$

must be stabilized. The dynamics of  $\mathbf{e}_v$  can be written using (1) as

$$m\dot{\mathbf{e}}_v = m\dot{\mathbf{v}}_d - m\mathbf{g} - \mathbf{t}.$$

In this first step, the thrust vector  $\mathbf{t}$  is considered as an input and it is given by the linear control law

$$\mathbf{t}_c := m(\dot{\mathbf{v}}_d - \mathbf{g} + \mathbf{K}_x \mathbf{e}_x + \mathbf{K}_v \mathbf{e}_v + \mathbf{K}_i \mathbf{e}_{x,i}) \quad (7)$$

where  $\mathbf{K}_x, \mathbf{K}_v, \mathbf{K}_i \in \mathbb{R}^{3 \times 3}$  are positive definite matrices and the integrated position error state is given by

$$\mathbf{e}_{x,i} := \int_{\tau=0}^t \mathbf{e}_x(\tau) d\tau.$$

Assuming that  $\mathbf{t} = \mathbf{t}_c$ , the position error dynamics including the integrator states are given by

$$\begin{pmatrix} \dot{\mathbf{e}}_{x,i} \\ \dot{\mathbf{e}}_x \\ \dot{\mathbf{e}}_v \end{pmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ -\mathbf{K}_i & -\mathbf{K}_x & -\mathbf{K}_v \end{bmatrix} \begin{pmatrix} \mathbf{e}_{x,i} \\ \mathbf{e}_x \\ \mathbf{e}_v \end{pmatrix}. \quad (8)$$

As it is a stable linear system, this guarantees position and velocity tracking for  $\mathbf{t} = \mathbf{t}_c$ .

**3.2. Attitude loop.** The attitude loop is connected to the position loop through the thrust vector command  $\mathbf{t}_c$ . Analogously to the position reference model,  $\mathbf{t}_c$  is used to generate the desired thrust vector  $\mathbf{t}_d \in \mathbb{R}^3$ . The difference is that here a nonlinear reference model is used in order to take the system dynamics in to account. First, the command is separated into total thrust  $T_c$  and body-fixed  $z$ -axis  $\mathbf{z}_c$  as

$$T_c = \|\mathbf{t}_c\|, \quad \mathbf{z}_c = -\frac{\mathbf{t}_c}{T_c}.$$

For the thrust, an exponentially stable first-order filter is used as a reference model

$$\dot{T}_d = -K(T_d - T_c), \quad (9)$$

where  $T_d \in \mathbb{R}$  is the desired thrust and  $K > 0 \in \mathbb{R}$  is the eigenfrequency of the filter. The attitude reference model is of the second order and corresponds to a reduced attitude controller which tracks the commanded vector  $\mathbf{z}_c$ . It is given by

$$\begin{pmatrix} \dot{\mathbf{R}}_{IR} \\ \dot{\boldsymbol{\omega}}_r \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{IR}\boldsymbol{\omega}_r^\times \\ -\mathbf{K}_d\boldsymbol{\omega}_r + \mathbf{K}_p\mathbf{R}_{RI}(\mathbf{z}_c \times \mathbf{z}_d) \end{pmatrix}, \quad (10)$$

where  $\mathbf{R}_{IR} \in SO^3$  is the rotation matrix which maps a vector from the reference model  $R$ -frame to the  $I$ -frame,  $\mathbf{z}_d = \mathbf{R}_{IR}(0, 0, 1)^T \in \mathbb{R}^3$  is the corresponding unit  $z$ -vector given in the  $I$ -frame and  $\boldsymbol{\omega}_r \in \mathbb{R}^3$  is the angular rate of the  $R$ -frame relative to the  $I$ -frame. The skew-symmetric matrix  $\boldsymbol{\omega}_r^\times \in \mathbb{R}^{3 \times 3}$  is defined such that for any vector  $\mathbf{a} \in \mathbb{R}^3$  we have that  $\boldsymbol{\omega}_r \times \mathbf{a} = \boldsymbol{\omega}_r^\times \mathbf{a}$ . The positive definite matrices  $\mathbf{K}_d, \mathbf{K}_p \in \mathbb{R}^{3 \times 3}$  guarantee that the equilibrium  $(\mathbf{z}_d, \boldsymbol{\omega}_r) = (\mathbf{z}_c, \mathbf{0})$  is almost global asymptotically stable (Chaturvedi et al., 2011). Finally, the desired thrust vector  $\mathbf{t}_d \in \mathbb{R}^3$  can be computed from

$$\mathbf{t}_d = -T_d\mathbf{z}_d. \quad (11)$$

We use the attitude baseline controller as presented in (Falconi and Holzapfel, 2013). For the sake of completeness, the main result is summarized in the following. In order to track the desired attitude  $\mathbf{t}_d \in \mathbb{R}^3$  and desired yaw rate  $\boldsymbol{\omega}_{z,d} \in \mathbb{R}$ , the zero equilibrium of the dynamics of the attitude error  $\mathbf{e}_t := \mathbf{t}_d - \mathbf{t}$  and the angular velocity error  $\mathbf{e}_\omega := \boldsymbol{\omega}_d - \boldsymbol{\omega}$  must be stabilized. The controller is based on block backstepping (Khalil, 2002). Hence, in the first step we consider the vector  $(\boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \dot{T})^T$  as the input of the attitude dynamics (4) and design the control law  $\mathbf{u}_2 \in \mathbb{R}^3$ . In the next backstepping step, we consider the angular velocity error  $\mathbf{e}_\omega := \boldsymbol{\omega}_d - \boldsymbol{\omega}$  with  $\boldsymbol{\omega}_d = (\mathbf{u}_{2x}, \mathbf{u}_{2y}, \boldsymbol{\omega}_{z,d})^T$ . Finally, the control law for the moments  $\mathbf{M}_d$  is designed.

Taking the error definitions into account and using (3) and (4), the dynamics of the whole error  $\mathbf{e} =$

$[\mathbf{e}_t^T, \mathbf{e}_\omega^T]^T \in \mathbb{R}^6$  are

$$\begin{aligned} \dot{\mathbf{e}} &:= \begin{pmatrix} \dot{\mathbf{e}}_t \\ \dot{\mathbf{e}}_\omega \end{pmatrix} \\ &= \begin{pmatrix} -\mathbf{R}_{IB}\mathbf{T}_\omega\boldsymbol{\omega} + \mathbf{z}_B\dot{T} + \dot{\mathbf{t}}_d \\ \mathbf{J}^{-1}(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} - \mathbf{M}_p) + \dot{\boldsymbol{\omega}}_d \end{pmatrix}, \\ &= \begin{pmatrix} -\mathbf{R}_{IB}\mathbf{T}_u\mathbf{u}_2 + \mathbf{R}_{IB}\mathbf{T}_\omega\mathbf{e}_\omega + \dot{\mathbf{t}}_d \\ \mathbf{J}^{-1}(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} - \mathbf{M}_p) + \dot{\boldsymbol{\omega}}_d \end{pmatrix}. \end{aligned}$$

The control laws from the work of Falconi and Holzapfel (2013) are given in terms of the virtual control of the first backstepping step  $\mathbf{u}_2$  and the desired moment  $\mathbf{M}_d$  and thrust  $T_d$  as follows:

$$\mathbf{u}_2 = \mathbf{T}_u^{-1}\mathbf{R}_{BI}(\dot{\mathbf{t}}_d + \mathbf{K}_t\mathbf{e}_t),$$

$$\mathbf{M}_d = \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{J}(\mathbf{T}_\omega^T\mathbf{R}_{BI}\mathbf{e}_t + \dot{\boldsymbol{\omega}}_d + \mathbf{K}_\omega\mathbf{e}_\omega),$$

$$T_d = \int_{\tau=t_0}^t \mathbf{u}_{2,z} d\tau, \quad (12)$$

Hence, the error dynamics become

$$\dot{\mathbf{e}} = \underbrace{\begin{bmatrix} -\mathbf{K}_t & \mathbf{R}_{IB}\mathbf{T}_\omega \\ -\mathbf{T}_\omega^T\mathbf{R}_{IB}^T & -\mathbf{K}_\omega \end{bmatrix}}_{:=\bar{\mathbf{A}}} \mathbf{e}$$

for  $\boldsymbol{\nu} = \boldsymbol{\nu}_d$ . Here, the matrices  $\mathbf{K}_t, \mathbf{K}_\omega \in \mathbb{R}^{3 \times 3}$  are positive definite and therefore  $\bar{\mathbf{A}}$  is negative definite. Exponential stability of the equilibrium  $\mathbf{e} = \mathbf{0}$  can be shown using, e.g., the Lyapunov function  $V = \frac{1}{2}\mathbf{e}^T\mathbf{e}$ . The only obstacle that prevent us from achieving global exponential stability is the system's inherent singularity at  $T = 0$  (see Falconi and Holzapfel, 2013). The results presented in this section hold as long as  $\boldsymbol{\nu} = \boldsymbol{\nu}_d$ . In the nominal case, this can be achieved with the mapping

$$\mathbf{u} = \mathbf{B}_{\nu\omega}^+\boldsymbol{\nu}_d, \quad (13)$$

where  $\mathbf{B}_{\nu\omega}^+$  is the Moore–Penrose pseudoinverse of  $\mathbf{B}_{\nu\omega}$  from (5). In the next sections we consider faults and failures within the propulsion system.

#### 4. Adaptive control allocation

Given the rigid-body control law  $\boldsymbol{\nu}_d$  from (12), the task of the adaptive control allocation (ACA) is to allocate it to the six actuators. In order to take faults and failures into account, we propose to estimate the control effectiveness of the rotors and use this information to solve the control allocation problem using an online optimization. Hence, the ACA is composed of two main elements as can be seen in Fig. 2: the online optimization-based control allocation and the adaptive effectiveness estimation. The following results are based on the findings of Falconi et al. (2016) and are applied here to the position tracking controller

in order to exhibit fault tolerant properties. Furthermore, the control effectiveness estimates are improved by introducing the null space shaping approach, which is a main contribution of the paper.

To model unknown actuator faults and failures, we extend (5) using a typical representation of adaptive control as done by Falconí *et al.* (2016):

$$\boldsymbol{\nu} = \mathbf{B}_{\nu\omega} \boldsymbol{\Lambda} \mathbf{u}. \quad (14)$$

Here,

$$\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6)$$

denotes the control effectiveness matrix, which scales every actuator command  $u_i$  by  $\lambda_i \geq 0$ . The nominal case is  $\lambda_i = 1$  and a complete failure corresponds to  $\lambda_i = 0$ .

Furthermore, let  $\mathcal{U} \subset \mathbb{R}^6$  be the set of possible inputs. In the case of multirotor systems, the possible rotation speed of one rotor is limited from above and below by  $\omega_{i,\max}$  and  $\omega_{i,\min}$ , respectively. Hence, the set of possible inputs can be written as

$$\mathcal{U} := \left\{ \mathbf{u} \in \mathbb{R}^6 \mid \omega_{i,\min}^2 \leq u_i \leq \omega_{i,\max}^2, \forall i = 1, \dots, 6 \right\}.$$

The attainable virtual control set (AVCS) is then the output image  $\mathcal{V}$  of the linear mapping (14) on  $\mathcal{U}$

$$\mathcal{V} := \{ \boldsymbol{\nu} \in \mathbb{R}^4 \mid \boldsymbol{\nu} = \mathbf{B}_{\nu\omega} \boldsymbol{\Lambda} \mathbf{u}, \mathbf{u} \in \mathcal{U} \}.$$

It is important to note that the size of the set  $\mathcal{V}$  heavily depends on the current control effectiveness  $\boldsymbol{\Lambda}$ . Figures 3 and 4 compare the AVCS of the hexarotor in Fig. 1 in the nominal case with the AVCS in the case of a failure of the rear right rotor (number 3).

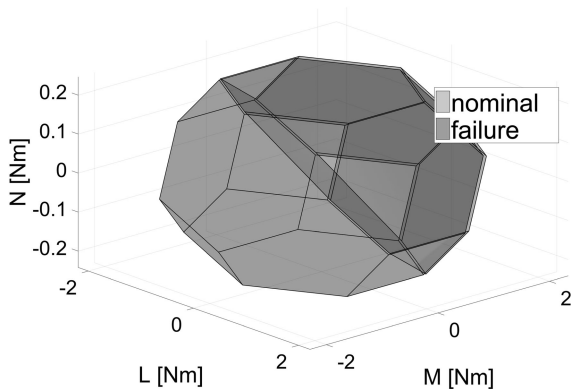


Fig. 3. Comparison of the attainable virtual control set for  $T = mg$  between nominal conditions and failure of the rear-right propeller (the plot generated with MPT 3.0 (Herceg *et al.*, 2013)).

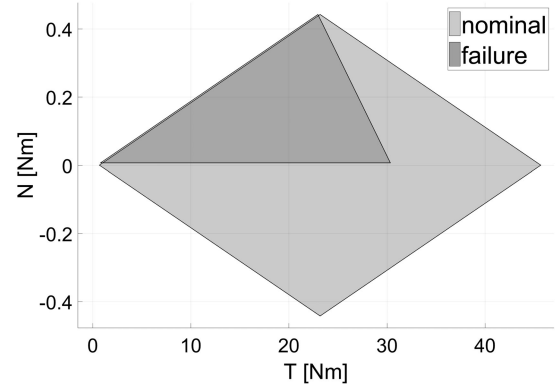


Fig. 4. Comparison of the attainable virtual control set for  $L = M = 0$  between nominal conditions and failure of the rear-right propeller (the plot generated with MPT 3.0 (Herceg *et al.*, 2013)).

**4.1. Optimal control allocation.** The task of the control allocation is to find an input command  $\mathbf{u}_{\text{cmd}}^*$  such that

$$\boldsymbol{\nu}_d = \mathbf{B}_{\nu\omega} \hat{\boldsymbol{\Lambda}} \mathbf{u}_{\text{cmd}}^* \quad \text{subject to} \quad \mathbf{u}_{\text{cmd}}^* \in \mathcal{U}.$$

Note that here we use the estimate  $\hat{\boldsymbol{\Lambda}}$  instead of the unknown  $\boldsymbol{\Lambda}$ .

Since the desired input will be calculated using an optimization process, the available search space for the actuator commands highly depends on the particular model of the actuation system. Therefore, the performance of the optimization also depends on the system at hand. In order to remove the model dependency and improve the numerical conditioning, the following linear transformation for the input is introduced:

$$u_i = (u_{i,\max} - u_{i,\min}) \bar{u}_i + u_{i,\min}, \quad (15)$$

with  $\bar{u}_i \in [0, 1]$  being the normalized actuator command. Inserting (15) in (14) yields

$$\underbrace{\boldsymbol{\nu} - \boldsymbol{\nu}_{\text{offset}}}_{\tilde{\boldsymbol{\nu}}} = \underbrace{\mathbf{B}_{\nu\omega} \mathbf{S}}_{\tilde{\mathbf{B}}_{\nu\omega}} \boldsymbol{\Lambda} \bar{\mathbf{u}}, \quad (16)$$

with  $\boldsymbol{\nu}_{\text{offset}} = \mathbf{B}_{\nu\omega} \boldsymbol{\Lambda} (u_{1,\min}, \dots, u_{6,\min})^T$  and  $\mathbf{S} = \text{diag}((u_{1,\max} - u_{1,\min}), \dots, (u_{6,\max} - u_{6,\min}))$ . Using the dimensionless input  $\bar{\mathbf{u}} \in \mathbb{R}^6$ , the available input space becomes a six dimensional unit cube and the parameters of the optimization can be tuned irrespective of the actuator limits.

A fundamental question of the optimization based control allocation is which cost function  $J$  is appropriate to the problem at hand. Following Falconí *et al.* (2016),

the proposed cost function in this paper is

$$J = \frac{1}{2} (\bar{\nu}_d - \bar{\mathbf{B}}_{\nu\omega} \hat{\mathbf{\Lambda}} \bar{\mathbf{u}}_{\text{cmd}})^T \mathbf{W} (\bar{\nu}_d - \bar{\mathbf{B}}_{\nu\omega} \hat{\mathbf{\Lambda}} \bar{\mathbf{u}}_{\text{cmd}}) + \sum_{i=1}^6 \left( \frac{\mu/2}{(1 - \bar{u}_{i,\text{cmd}})^2} + \frac{\mu/2}{\bar{u}_{i,\text{cmd}}^2} \right), \quad (17)$$

where the positive definite  $\mathbf{W} \in \mathbb{R}^{4 \times 4}$  and  $\mu > 0 \in \mathbb{R}$  are weighting parameters and  $\bar{\nu}_d = \nu_d - \nu_{\text{offset}}$  is the transformed desired virtual control vector.

The primary objective of the control allocation is to obtain a feasible nondimensional actuator command that satisfies  $\bar{\nu}_d = \bar{\mathbf{B}}_{\nu\omega} \hat{\mathbf{\Lambda}} \bar{\mathbf{u}}_{\text{cmd}}$ . Therefore, the first term in (17) penalizes commands that do not lead to the desired virtual control. The last two terms of (17) are barrier functions, which take input saturations into account. Hence, the constrained optimization problem is reformulated as an unconstrained one using (17).

If the virtual control vector  $\nu_d \in \mathcal{V}$  is attainable, there exists at least one actuator command  $\mathbf{u}_{\text{cmd}}$  that makes the first expression tend to zero. For overactuated systems such as hexarotors there are infinitely many commands that lead to the same virtual control. In the case of severe actuator degradation, the attainable virtual control set  $\mathcal{V}$  is noticeably reduced, possibly leading to commands  $\nu_d \notin \mathcal{V}$  that are not attainable. In this situation the weighting matrix  $\mathbf{W} \in \mathbb{R}^{4 \times 4}$  allows for prioritizing certain virtual controls. In the multirotor case the usual strategy is prioritizing the desired thrust, pitch and roll moments over the desired yaw moment (Mueller and D'Andrea, 2014), since they are primarily responsible for position control and hence controlled flight of the hexacopter system.

By choosing  $|\mu|$  much smaller than the smallest eigenvalue of  $\mathbf{W}$ , the last two terms of (17) become negligible as long as the required actuator command is not near the limits. Therefore, the resulting actuator command  $\mathbf{u}_{\text{cmd}}^*$  that minimizes the cost function will approximately satisfy  $\nu_d \approx \bar{\mathbf{B}}_{\nu\omega} \hat{\mathbf{\Lambda}} \mathbf{u}_{\text{cmd}}^*$ .

If the required virtual control  $\nu_d$  is near the borders of  $\mathcal{V}$ , the feasible actuator commands will tend to their limits, making the second term in  $J$  increase. In the case of redundant actuators, the optimization process allows to aim for non-saturating actuator commands in the null space of  $\bar{\mathbf{B}}_{\nu\omega} \hat{\mathbf{\Lambda}}$ .

Since (17) describes a convex function, gradient based optimization is used to minimize  $J$ . The gradient of the cost function (17) is given by

$$\nabla J = -\hat{\mathbf{\Lambda}}^T \bar{\mathbf{B}}_{\nu\omega}^T \mathbf{W}^T (\bar{\nu}_d - \bar{\mathbf{B}}_{\nu\omega} \hat{\mathbf{\Lambda}} \bar{\mathbf{u}}_{\text{cmd}})$$

$$+ \begin{pmatrix} \frac{\mu}{(1 - \bar{u}_{1,\text{cmd}})^3} - \frac{\mu}{\bar{u}_{1,\text{cmd}}^3} \\ \vdots \\ \frac{\mu}{(1 - \bar{u}_{6,\text{cmd}})^3} - \frac{\mu}{\bar{u}_{6,\text{cmd}}^3} \end{pmatrix}. \quad (18)$$

Furthermore, a backtracking line search algorithm according to Armijo-Goldstein (Potra, F.A. and Shi, 1995) is used to improve the results. The schematic structure of the gradient based optimization algorithm with  $\epsilon > 0$ ,  $0 < \alpha < 1$ , and  $0 < c < 1$  can be seen in Fig. 5. The maximum iteration numbers for the outer loop and for the inner loop are set to 5 and 10, respectively, leading to a worst-case computational effort of approximately 5000 multiplications and 4500 additions for the whole optimization.

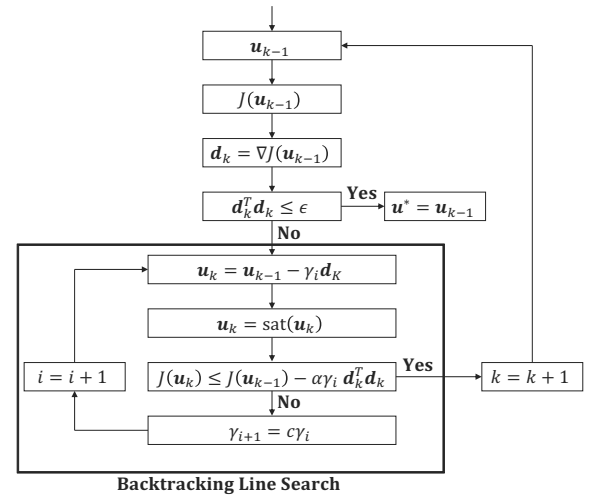


Fig. 5. Gradient based optimization.

In this section, an optimization-based control allocation approach has been introduced, which is not only capable of covering the whole attainable set of virtual controls, but also aims to sustain stability in the case of highly degraded configurations. In the next section, the estimation of the actuator effectiveness is presented.

**4.2. Control effectiveness estimation.** In order to estimate the effectiveness of each rotor, we propose a concurrent learning approach (Chowdhary, 2010; Chowdhary and Johnson, 2010). First, the actuator model which includes the faults (14) is rewritten as

$$\nu = \mathbf{B}_{\nu\omega} \mathbf{U} \lambda.$$

Here,  $\mathbf{U} = \text{diag}(\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2, \omega_5^2, \omega_6^2)$  is a diagonal matrix consisting of the squared rotor speeds and  $\lambda \in$

$\mathbb{R}^6$  is a vector, whose entries correspond to the diagonal entries of the control effectiveness matrix  $\mathbf{A}$  in (14).

Subsequently, let the output  $\mathbf{y} \in \mathbb{R}^4$  for the concurrent learning approach be as follows:

$$\mathbf{y} = \begin{pmatrix} \dot{\boldsymbol{\omega}} + \mathbf{J}^{-1}(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}) \\ f_z \end{pmatrix} = \underbrace{\begin{bmatrix} \mathbf{J}^{-1}\mathbf{B}_{M\boldsymbol{\omega}} \\ -\frac{1}{m}\mathbf{B}^{T\boldsymbol{\omega}} \end{bmatrix}}_{:=\mathbf{B}_{y\boldsymbol{\omega}}} \mathbf{U}\boldsymbol{\lambda},$$

with  $\boldsymbol{\omega}$  and  $\dot{\boldsymbol{\omega}}$  being the measurement of the angular rate and its numerical derivative respectively,  $m$  being the mass of the multirotor and  $f_z$  denoting the measurement of the specific force along the body fixed vertical axis. The matrices  $\mathbf{B}^{T\boldsymbol{\omega}}$  and  $\mathbf{B}_{M\boldsymbol{\omega}}$  are defined in (5). Furthermore,  $\mathbf{U}$  has to be either measured or estimated with an actuator model. The actuator model used in the presented approach is given by the first-order filter  $\dot{\mathbf{U}} = T_{\text{act}}^{-1}(\mathbf{U}_{\text{cmd}} - \mathbf{U})$  with the actuator time constant  $T_{\text{act}} > 0 \in \mathbb{R}$ .

Since the numerical derivative of the angular rate may lead to undesired noise in the output channel, a low-pass filter is incorporated. Let  $\mathbf{y}_f$  and  $\mathbf{U}_f$  be the low-pass filtered version of  $\mathbf{y}$  and  $\mathbf{U}$  respectively, such that

$$\mathbf{y}_f = \mathbf{B}_{y\boldsymbol{\omega}}\mathbf{U}_f\boldsymbol{\lambda}. \quad (19)$$

They can be computed by the following update laws:

$$\dot{\mathbf{y}}_f = \frac{1}{T_f}(\mathbf{y} - \mathbf{y}_f), \quad \dot{\mathbf{U}}_f = \frac{1}{T_f}(\mathbf{U} - \mathbf{U}_f),$$

with  $T_f > 0 \in \mathbb{R}$  denoting the time constant of the low-pass filter.

Equation (19) expresses the low-pass filtered measurement in terms of the rotor effectiveness vector  $\boldsymbol{\lambda}$  and the low-pass filtered squared rotor speed matrix  $\mathbf{U}_f$ . Hence, an estimate of  $\mathbf{y}_f$  can be written as

$$\hat{\mathbf{y}}_f = \mathbf{B}_{y\boldsymbol{\omega}}\mathbf{U}_f\hat{\boldsymbol{\lambda}}.$$

Furthermore, let

$$\tilde{\mathbf{y}}_f = \hat{\mathbf{y}}_f - \mathbf{y}_f = \mathbf{B}_{y\boldsymbol{\omega}}\mathbf{U}_f\tilde{\boldsymbol{\lambda}}$$

be the error between the estimated and the true outputs, where  $\tilde{\boldsymbol{\lambda}} = \hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}$  denotes the estimation error of the rotor effectiveness vector. Then, we can define the concurrent learning cost function  $J_{\text{CL}}(\hat{\boldsymbol{\lambda}}) : \mathbb{R}^6 \rightarrow \mathbb{R}$  as follows:

$$J_{\text{CL}} = \sum_{i=1}^p \frac{1}{2} \tilde{\mathbf{y}}_{f,i}^T \tilde{\mathbf{y}}_{f,i},$$

where  $\tilde{\mathbf{y}}_{f,i}$  denotes the error at some instant of time  $t_i$  and  $p$  is the number of stored data points. In order to make the cost function decrease in time, the following gradient

based update law for the rotor effectiveness vector is used:

$$\begin{aligned} \dot{\hat{\boldsymbol{\lambda}}} &= -\gamma \underbrace{\sum_{i=1}^p \mathbf{U}_{f,i}^T \mathbf{B}_{y\boldsymbol{\omega}}^T (\mathbf{B}_{y\boldsymbol{\omega}} \mathbf{U}_{f,i} \hat{\boldsymbol{\lambda}} - \mathbf{y}_{f,i})}_{:=\boldsymbol{\xi}} + \mathbf{Q}_{\boldsymbol{\xi}} \mathbf{w} \\ &= -\gamma \underbrace{\sum_{i=1}^p \mathbf{U}_{f,i}^T \mathbf{B}_{y\boldsymbol{\omega}}^T \mathbf{B}_{y\boldsymbol{\omega}} \mathbf{U}_{f,i}}_{:=\mathbf{H}} \tilde{\boldsymbol{\lambda}} + \mathbf{Q}_{\boldsymbol{\xi}} \mathbf{w}. \end{aligned} \quad (20)$$

Here,  $\gamma > 0 \in \mathbb{R}$  is the adaptive tuning parameter. The second term in the equation corresponds to the null space shaping term. The gradient of the cost function with respect to the estimated control effectiveness vector is given by  $\boldsymbol{\xi} = \partial J_{\text{CL}} / \partial \hat{\boldsymbol{\lambda}}$ . The vector  $\mathbf{w} \in \mathbb{R}^6$  is used to shape the update law in the null space of the gradient  $\boldsymbol{\xi}$  using the projection matrix

$$\mathbf{Q}_{\boldsymbol{\xi}} := \left( \mathbf{I} - \frac{\boldsymbol{\xi} \boldsymbol{\xi}^T}{\boldsymbol{\xi}^T \boldsymbol{\xi}} \right). \quad (21)$$

By choosing appropriate instants of time for recording the data points, the matrix  $\mathbf{H} \in \mathbb{R}^{6 \times 6}$  can be made positive definite (Chowdhary, 2010; Chowdhary and Johnson, 2010). While the first representation of (20) is used for implementation, the second one is used within the stability assessment in the following. To this end, we adapt the Lyapunov function candidate as follows:

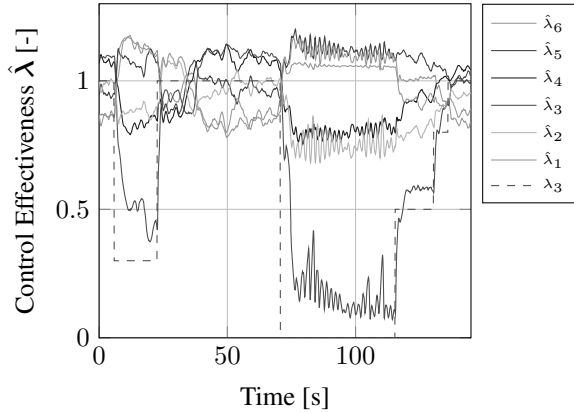
$$V = \frac{1}{2} \tilde{\boldsymbol{\lambda}}^T \mathbf{H} \tilde{\boldsymbol{\lambda}},$$

with  $\mathbf{H}$  denoting the positive definite history stack matrix. The derivative of the Lyapunov function yields

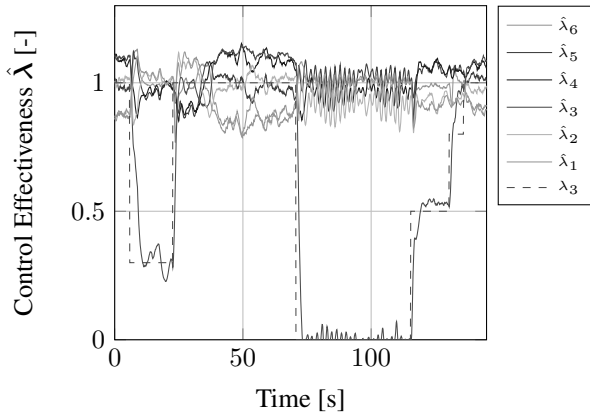
$$\begin{aligned} \dot{V} &= \tilde{\boldsymbol{\lambda}}^T \mathbf{H} \dot{\tilde{\boldsymbol{\lambda}}}, \\ &= \tilde{\boldsymbol{\lambda}}^T \mathbf{H} \left( -\gamma \mathbf{H} \tilde{\boldsymbol{\lambda}} + \mathbf{Q}_{\boldsymbol{\xi}} \mathbf{w} \right), \\ &= -\tilde{\boldsymbol{\lambda}}^T \gamma \mathbf{H}^2 \tilde{\boldsymbol{\lambda}} + \boldsymbol{\xi}^T \mathbf{Q}_{\boldsymbol{\xi}} \mathbf{w}, \\ &= -\tilde{\boldsymbol{\lambda}}^T \gamma \mathbf{H}^2 \tilde{\boldsymbol{\lambda}}. \end{aligned} \quad (22)$$

The matrix  $\gamma \mathbf{H}^2$  is symmetric and positive definite since  $\gamma > 0$  and  $\mathbf{H}$  is symmetric and positive definite, too. Hence, the origin of the estimation error dynamics is exponentially stable. As can be seen from (22),  $\mathbf{w}$  does not have any influence on the stability and can thus be used to modify the update law by shaping the update direction of the control effectiveness vector without violating the negative definiteness of  $\dot{V}$ .

The null space shaping approach is a main contribution in the paper and its main idea is to use  $\mathbf{w}$  to alter the update direction of  $\hat{\boldsymbol{\lambda}}$  during the minimization of the cost function in order to improve the convergence. Assuming that most of the time the rotors are working



(a)



(b)

Fig. 6. Flight tests: control effectiveness estimation without (a) and with (b) null space shaping.

properly, the shaping direction  $\mathbf{w}$  is chosen to point towards one as

$$\mathbf{w} = \text{diag}(\sigma_1 \dots \sigma_6) \left( \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} - \begin{bmatrix} \hat{\lambda}_1 \\ \vdots \\ \vdots \\ \hat{\lambda}_6 \end{bmatrix} \right), \quad (23)$$

where  $\sigma_i \in \{1, 0\}$  is the fault detection function

$$\sigma_i = \begin{cases} 0 & \text{if } \hat{\lambda}_i < \lambda_{TH}, \\ 1 & \text{otherwise.} \end{cases}$$

As long as the estimated effectiveness  $\hat{\lambda}_i$  of the  $i$ -th rotor is above the threshold  $\lambda_{TH}$ , the update direction of  $\hat{\lambda}_i$  is deflected towards the value 1 by projecting  $\mathbf{w}$  from (23) onto the hyperplane perpendicular to the current gradient of the cost function  $J_{CL}$ . However, if the effectiveness estimation of the  $i$ -th rotor drops below the threshold value, the corresponding entry in  $\mathbf{w}$  is removed by  $\sigma_i$ . In the next section, the flight test results are shown including a comparison of the estimation performance with and without the null space shaping approach.

Table 1. Flight Test: controller parameters.

Sym.	Value	Sym.	Value
$\mathbf{K}_x$	diag (0.5, 0.5, 0.5)	$T_{act}$	0.08
$\mathbf{K}_v$	diag (1.5, 1.5, 1.5)	$\epsilon$	10
$\mathbf{K}_t$	diag (4, 4, 4)	$c$	0.3
$\mathbf{K}_\omega$	diag (20, 20, 5)	$\gamma_0$	0.1
$\mathbf{K}_i$	diag (0.05, 0.05, 0.1)	$\alpha$	0.5
$\mathbf{W}$	diag (100, 100, 20, 1)	$\mu$	$10^{-4}$
$k_{max}$	5	$i_{max}$	10
$\mathbf{\Gamma}$	$\mathbf{I}_6 \cdot 10^{-4}$	$p$	10
$\lambda_{TH}$	0.6	$T_f$	0.05

### 5. Flight test results

The main contribution of this paper is the presentation of a position tracking controller that deals with unknown actuator failures and addresses application-driven requirements. As already mentioned in the introduction, this is in contrast to partial results found in the literature which address only failure detection and identification, or only controller reconfiguration, or need accurate external position sensors, or use off-board computations. The effectiveness of the presented system is validated in outdoor flight experiments subject to an unknown failure of any of the propellers. To the best of our knowledge this type of result has not yet been published.

The testbed is the AscTec Firefly (Achtelik et al., 2012), which can be seen in Fig. 1. For flight tests the controller runs with a sampling rate of 3 [ms] on a Gumstix Overo FireSTORM Computer-on-Module (CoM), which communicates with the AscTec Autopilot. The controller is designed in Simulink® and ported to C using the Simulink® Coder™. The AscTec Autopilot delivers the sensor data including a GPS position estimate. This sensor data is used to compute the state of the hexacopter within a navigation filter which is out of focus of this paper. For the flight test, the parameters in Table 1 have been used. The goal of the flight test is to show the performance of the presented fault-tolerant controller during an outdoor flight test. Therefore, the degradation and failure of the rear-right propeller are emulated by scaling down the resulting  $u_{3,cmd}$ . It is important to note that these induced faults are unknown to the controller. Figure 6 shows the control effectiveness estimation with and without null space shaping (NSS) for the performed outdoor flight test. The dashed line represents the current effectiveness. The first fault corresponds to a loss of effectiveness of about 70% at approximately  $t = 5$  [s] for about 17 [s]. At  $t = 70$  [s] for about 45 [s] the motor is completely turned off simulating a real failure, i.e.,  $\lambda_3 = 0$ , which results in an idle command. Afterwards, the effectiveness is stepwise increased up to  $\lambda_3 = 1$ . As can be seen, the estimates with NSS are more accurate as without NSS. Nevertheless, note that both types of estimates in Fig. 6 are coherent with the observations



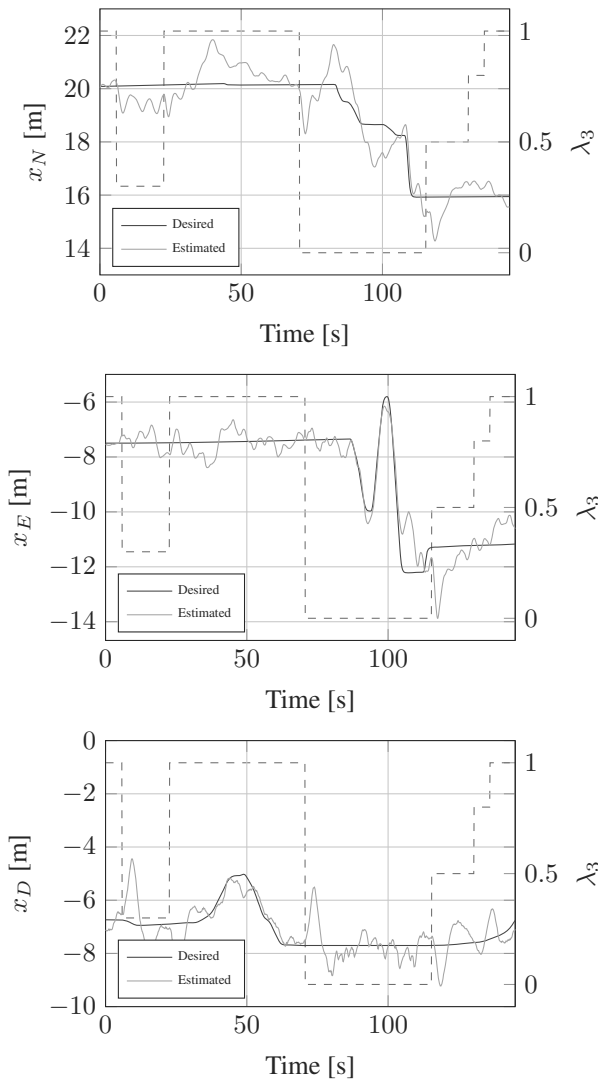


Fig. 7. Flight tests: position tracking.

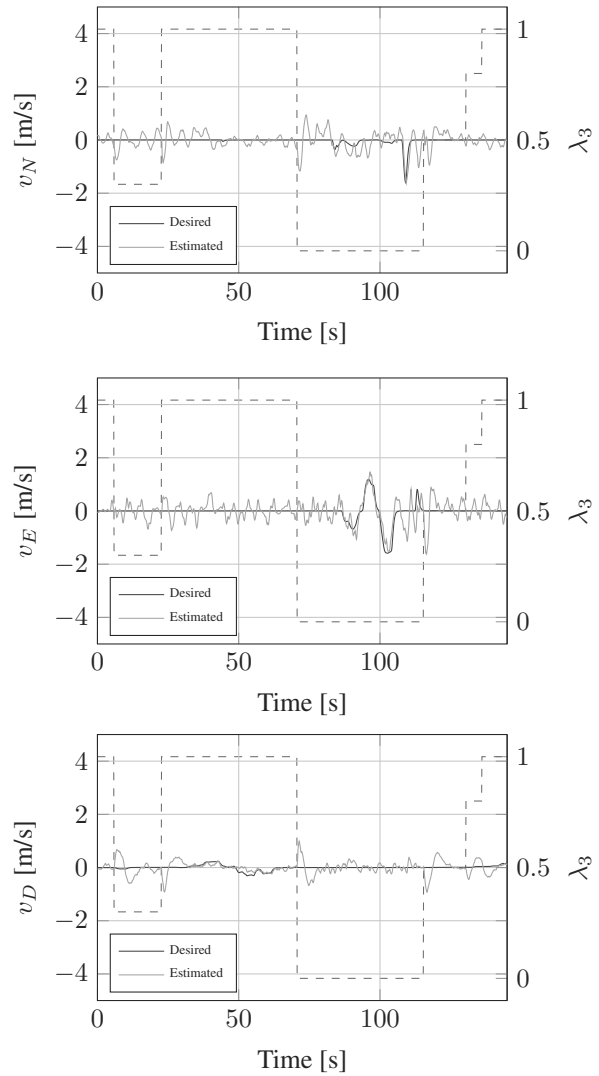


Fig. 8. Flight tests: velocity tracking.

made during the flight test and allow for a stable flight (see also Falconi *et al.*, 2016). However, if the hexarotor performs a maneuver that introduces new dynamics, the transient performance would be worse in the case without the NSS until the estimates are corrected. In the following, the tracking performance of the controller will be shown using the estimates  $\hat{\lambda}$  computed with NSS.

In Figs. 7 and 8 the position and velocity tracking performance can be seen. In order to easily identify the degradation and the total failure cases, the control effectiveness  $\lambda_3$  is overlaid into the following plots. During the first fault, the clearest effect is a height loss of about 2 [m]. The effect in the  $x$  and  $y$  axis is not profound. During the total failure at about 70 [s], there is again a loss in height of about 2 [m] and also a drift of about 2 [m] to the south. The reason is that the rear-right rotor was pointing to the south when the failure occurred, i.e., a heading angle of  $\Psi \approx 30$  [°]; see Fig. 9. From Figs. 7 and

8 it can be seen that the faults are compensated after the rotor effectiveness is correctly estimated and the system performance is recovered. During the total failure case different position commands were successfully tracked.

In Fig. 9 the attitude tracking performance is depicted. Here the Euler angles are used only for visualization purposes as the controller uses the thrust vector  $\mathbf{t}$  as explained in Section 3. The roll and pitch angles show small transient deviations in the positive direction right after the introduction of a fault. This is because of the position of the faulty rotor: see Fig. 1. The oscillatory characteristics of the attitude commands result from the rapid yaw motion during the total failure case. This is explained from the attainable virtual control set as shown in Fig. 4. Because of the fault only positive yaw moments can be produced while hovering and maintaining zero roll and pitch moments. Therefore, the hexacopter rotates at a yaw rate of 200 [deg/s] in order to successfully

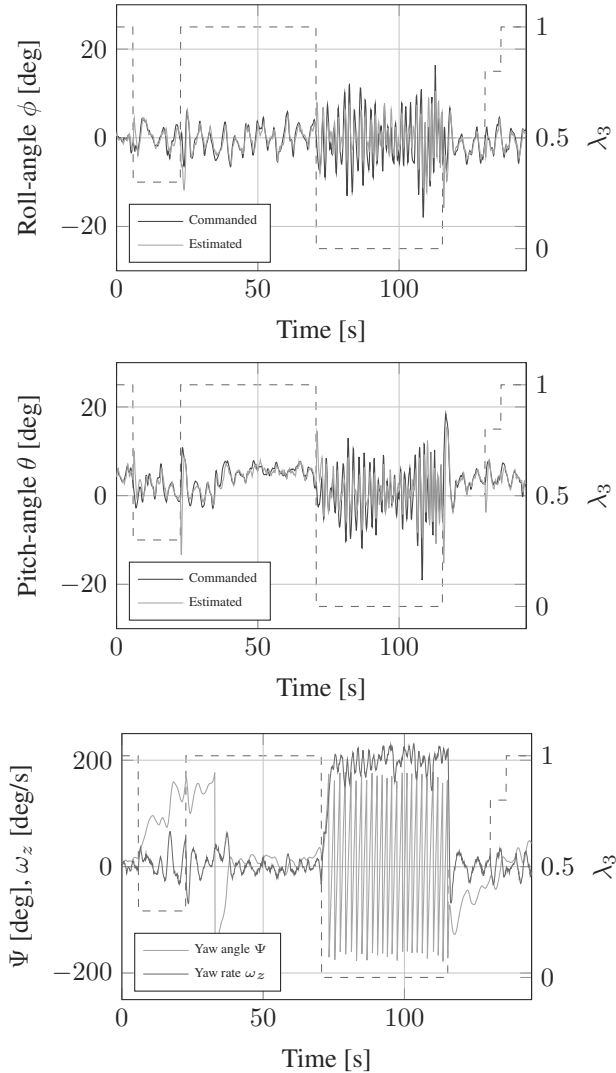


Fig. 9. Flight tests: attitude tracking.

track the roll and pitch commands.

The prioritization of the roll and pitch moments over the yaw moment can be better seen in Fig. 10. Here, the comparison of the desired virtual controls, the virtual controls achieved by the presented control allocation approach for  $\Lambda = \hat{\Lambda}$  and the virtual controls achieved by the pseudoinverse control allocation for  $\Lambda = \hat{\Lambda}$  is shown. The pseudoinverse solution using the estimate  $\hat{\Lambda}$  has been computed offline as a comparison as follows  $\mathbf{u}_{\text{cmd}} = \text{sat}(\hat{\Lambda}^{-1} \mathbf{B}_{\nu\omega}^+ \nu_d)$ . Here,  $\text{sat}(\cdot)$  is a saturation function that saturates the individual  $u_{i,\text{cmd}}$  such that  $\mathbf{u}_{\text{cmd}} \in \mathcal{U}$ . After saturating the respective actuators, the resulting virtual control vector does not necessarily match the desired one, leading to an unexpected output (Ducard and Hua, 2011). It is interesting to note that during the nominal and the degradation case, both control allocations perform similarly. The difference is only noticeable in the

total failure case. Here, the desired virtual controls  $\nu_d$  lie outside the AVCS. While the pseudoinverse solution leads to a deviation of all four virtual controls, the optimal control allocation is able to prioritize thrust, roll and pitch moments over yaw moments. This is achieved by selecting the weighting matrix  $\mathbf{W}$  of the optimization as in Table 1. The corresponding actuator commands can be seen in Fig. 11. It can be seen that the commands generated by the pseudoinverse control allocation are saturated most of the time for four out of six rotors. This explains the large deviations in Fig. 10. The adaptive control allocation rather slows down the front-left rotor and by ignoring the yaw moment it is able to stabilize the system almost as a quadrotor.

## 6. Conclusion

This paper presented an adaptive fault-tolerant controller for position tracking of a hexacopter system. The presented algorithms were validated in outdoor experiments using GPS as a position measurement and computing all the algorithms onboard. The flight tests include an unknown degradation test and an unknown total failure case for the usual hexacopter configuration leading to an extreme reduction in the attainable virtual control set (AVCS). The considered controller was able to maintain position tracking by allowing deviations in the yaw tracking. In this specific test, the yaw rate reached 200 [deg/s]. The flight test intends to imitate the application scenarios of multirotors.

The considered controller has a baseline controller and the adaptive augmentation. The baseline controller is a cascade composed of position and attitude control loops. The adaptive augmentation is composed of the control effectiveness estimation and the gradient-based optimal control allocation. The estimation of the control effectiveness is based on concurrent learning and null space shaping. The optimal control allocation allows the prioritization of selected virtual controls. In this application this feature is key in order to track the position of the hexacopter in the case of a total failure. The contributions of this paper are the extension from an attitude to a position controller and the introduction of the null space shaping approach.

In order to allow a full stabilization in the hexacopter case, the future work will take into account that the motors are able to rotate in both directions as in the work of Achtelik et al. (2012). Furthermore, an analysis of which maneuvers improve the observability of the control effectiveness is of interest. Finally, the new information of the control effectiveness estimates can also be used in combination with trajectory generators or command shapers for improving the overall system performance as proposed by Mühlegg et al. (2014).

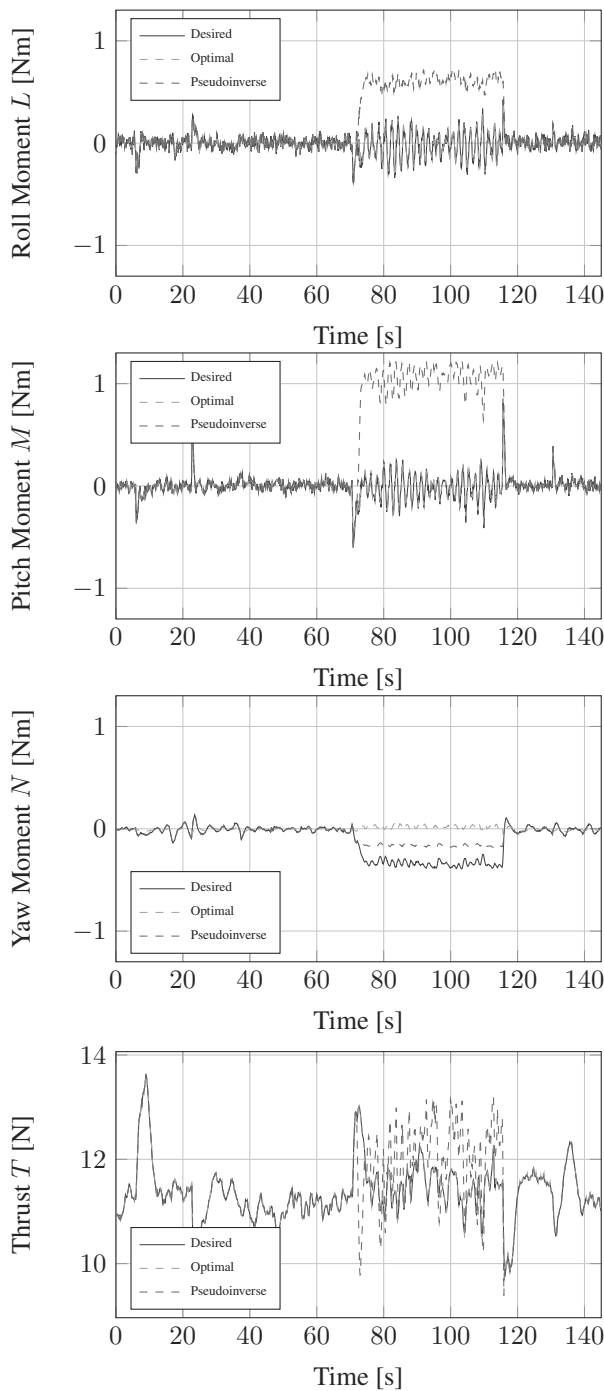


Fig. 10. Flight tests: virtual controls.

## References

- Achtelik, M., Doth, K.-M., Gurdan, D. and Stumpf, J. (2012). Design of a multi rotor MAV with regard to efficiency, dynamics and redundancy, *Proceedings of Guidance, Navigation, and Control Conference, Minneapolis, MN, USA*.
- Amoozgar, M.H., Chamseddine, A. and Zhang, Y. (2012). Fault-tolerant fuzzy gain scheduled PID for a quadrotor

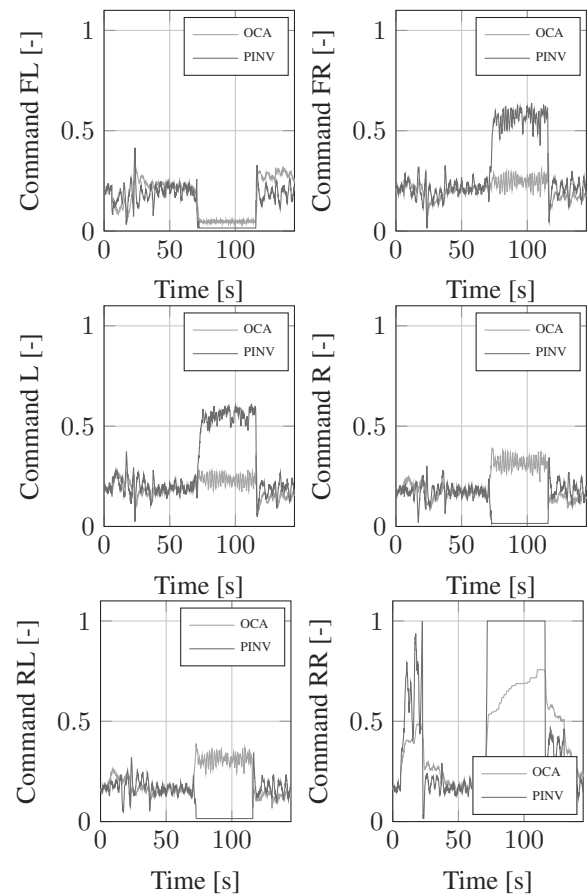


Fig. 11. Flight tests: normalized propeller commands.

helicopter testbed in the presence of actuator faults, *IFAC Proceedings Volumes* **45**(3): 282–287.

- Cen, Z., Noura, H. and Younes, Y.A. (2015). Systematic fault tolerant control based on adaptive Thau observer estimation for quadrotor UAVs, *International Journal of Applied Mathematics and Computer Science* **25**(1): 159–174, DOI: 10.1515/amcs-2015-0012.
- Chaturvedi, N., Sanyal, A. and McClamroch, N. (2011). Rigid-body attitude control, *IEEE Control Systems* **31**(3): 30–51.
- Chowdhary, G. (2010). *Concurrent Learning for Convergence in Adaptive Control without Persistency of Excitation*, PhD thesis, Georgia Institute of Technology, Atlanta, GA.
- Chowdhary, G. and Johnson, E. (2010). Concurrent learning for convergence in adaptive control without persistency of excitation, *Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA*, pp. 3674–3679.
- Du, G.-X., Quan, Q. and Cai, K.-Y. (2015). Controllability analysis and degraded control for a class of hexacopters subject to rotor failures, *Journal of Intelligent & Robotic Systems* **78**(1): 143–157.
- Ducard, G. and Hua, M.D. (2011). Discussion and practical aspects on control allocation for a multi-rotor

- helicopter, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII(1/C22)*: 95–100.
- Dydek, Z., Annaswamy, A. and Lavretsky, E. (2010). Adaptive control of quadrotor UAVs in the presence of actuator uncertainties, *Proceedings of AIAA Infotech@Aerospace 2010, Atlanta, GA, USA*.
- Dydek, Z.T., Annaswamy, A.M. and Lavretsky, E. (2013). Adaptive control of quadrotor UAVs: A design trade study with flight evaluations, *IEEE Transactions on Control Systems Technology* **21**(4): 1400–1406.
- Falconí, G.P., Angelov, J. and Holzapfel, F. (2016). Hexacopter outdoor flight test results using adaptive control allocation subject to an unknown complete loss of one propeller, *Proceedings of the 3rd Conference on Control and Fault-Tolerant Systems (SysTol), Barcelona, Spain*, pp. 373–380.
- Falconí, G.P., Heise, C. and Holzapfel, F. (2015). Fault-tolerant position tracking of a hexacopter using an extended state observer, *Proceedings of the 6th International Conference on Automation, Robotics and Applications (ICARA), Queenstown, New Zealand*, pp. 550–556.
- Falconí, G.P. and Holzapfel, F. (2013). Position tracking of a multicopter using a geometric backstepping control law, *CEAS EuroGNC, Delft, The Netherlands*.
- Falconí, G.P. and Holzapfel, F. (2014). Position tracking of a hexacopter using a geometric backstepping control law—Experimental results, *Proceedings of the IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Yogyakarta, Indonesia*, pp. 20–25.
- Freddi, A., Longhi, S., Monteriù, A. and Prist, M. (2014). Actuator fault detection and isolation system for an hexacopter, *Proceedings of the IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA), Senigallia, Italy*, pp. 1–6.
- Heise, C., Falconí, G. P. and Holzapfel, F. (2014). Hexacopter outdoor flight test results of an extended state observer based controller, *Proceedings of the IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Yogyakarta, Indonesia*, pp. 26–33.
- Herceg, M., Kvasnica, M., Jones, C. and Morari, M. (2013). Multi-parametric toolbox 3.0, *Proceedings of the European Control Conference, Zürich, Switzerland*, pp. 502–510.
- Khalil, H.K. (2002). *Nonlinear Systems*, 3rd Edn., Prentice Hall, Upper Saddle River, NJ.
- Merheb, A.-R., Noura, H. and Bateman, F. (2015). Design of passive fault-tolerant controllers of a quadrotor based on sliding mode theory, *International Journal of Applied Mathematics and Computer Science* **25**(3): 561–576, DOI:10.1515/amcs-2015-0042.
- Mueller, M.W. and D'Andrea, R. (2014). Stability and control of a quadcopter despite the complete loss of one, two, or three propellers, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, pp. 45–52.
- Mühlegg, M., Niermeyer, P., Falconí, G.P. and Holzapfel, F. (2015). L1 fault tolerant adaptive control of a hexacopter with control degradation, *Proceedings of the IEEE Conference on Control Applications (CCA), Sydney, Australia*, pp. 750–755.
- Mühlegg, M., Niermeyer, P. and Holzapfel, F. (2014). Reference command shaping for approximate dynamic inversion based model reference adaptive control, *Proceedings of the IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Yogyakarta, Indonesia*, pp. 179–184.
- Potra, F. A. and Shi, Y. (1995). Efficient line search algorithm for unconstrained optimization, *Journal of Optimization Theory and Applications* **85**(3): 677–704.
- Saied, M., Lussier, B., Fantoni, I., Francis, C., Shraim, H. and Sanahuja, G. (2015). Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA*, pp. 5266–5271.
- Santos, M.F., Honório, L.M., Costa, E.B., Oliveira, E.J. and Visconti, J. P. P. G. (2015). Active fault-tolerant control applied to a hexacopter under propulsion system failures, *2015 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania*, pp. 447–453.
- Schneider, T., Ducard, G., Rudin, K. and Strupler, P. (2012). Fault-tolerant control allocation for multirotor helicopters using parametric programming, *International Micro Air Vehicle Conference (IMAV), Braunschweig, Germany*.
- Vey, D. and Lunze, J. (2016). Experimental evaluation of an active fault-tolerant control scheme for multirotor UAVs, *3rd Conference on Control and Fault-Tolerant Systems (SysTol), Barcelona, Spain*, pp. 125–132.
- Yang, Y., Iwakura, D., Namiki, A., Nonami, K. and Wang, W. (2016). Autonomous flight of hexacopter under propulsion system failure, *Journal of Robotics and Mechatronics* **28**(6): 899–910.



**Guillermo P. Falconí** received his MSc degree from Technische Universität München (TUM), Germany, in 2011. Since 2012 he has been a research associate and is currently pursuing the PhD degree at the Institute of Flight System Dynamics (FSD), TUM. His main research interests involve adaptive control, fault tolerant control and UAV control systems. He has published several papers in international conference proceedings and received the 2016 IEEE MSC Best Student Paper Award.



**Jorg Angelov** received his BSc degree in aerospace engineering from Technische Universität München (TUM), Germany, in 2015 and passed his MSc degree in aerospace engineering with distinction at TUM in 2017. His studies are focused on nonlinear adaptive flight control, fault tolerant control, navigation, and data fusion. He did his MSc thesis at the UAV Research Facility (UAVRF) of the Georgia Institute of Technology.



**Florian Holzapfel** received his PhD degree from the Institute of Flight Mechanics and Flight Control, Technische Universität München (TUM), Germany, in 2004. Since 2007 he has been a full professor and the director of the Institute of Flight System Dynamics (FSD), TUM. His research interests are in making novel control methods accessible for real-life application and demonstrating their viability and benefits in real flight.

Received: 19 March 2017

Revised: 13 November 2017

Accepted: 13 January 2018