Robotics and Autonomous Systems 79 (2016) 87–98

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

# Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator

A. Chovancová *, T. Fico, P. Hubinský, F. Duchoň

*Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19, Bratislava, Slovakia*

## HIGHLIGHTS

- Several attitude controllers using quaternion were designed to control quadrotor.
- Several position controllers were designed to control quadrotor.
- The disturbance observer and the state-space estimator were designed.
- The comparison of performance of combinations of controllers was carried out.
- The external disturbance was applied in the verification process.

## ARTICLE INFO

*Article history:*
Received 1 June 2015
Received in revised form
20 December 2015
Accepted 15 January 2016
Available online 4 February 2016

*Keywords:*
LQR
Backstepping controller
Quadrotor
Position estimator
Disturbance observer
Quaternion

## ABSTRACT

The aim of this article is to design and verify various control techniques for a quadrotor using a quaternion representation of the attitude. All attitude controllers use a quaternion error to compute control signals that are calculated from an actual quaternion and a desired quaternion obtained from a position controller. Attitude and position control laws are computed using a PD, LQR and backstepping control technique.

All combinations of controllers will be verified by simulation. We add noise, apply an actuator restriction and use a different sampling period for position and attitude feedback signals to get the simulation closer to real conditions.

Moreover, external disturbances were implemented into the simulation; hence a disturbance observer along with a position estimator will be designed to improve the performance of the presented controllers.

The performance of all combinations of controllers was evaluated using various quality indicators, such as the integral of absolute errors and total thrust, settling times and also maximum overshoots when external disturbance was applied. Some of the controllers exhibit very similar behaviour, so we chose the three best controllers for each scenario used in the simulation.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

In the last decades interest in Unmanned Aerial Vehicles (UAVs), also called drones, has increased. This group of vehicles consists of various flying platforms such as airships, fixed-wing or Vertical Take-Off and Landing (VTOL) vehicles. In the article we will focus on a quadrotor, which belongs to VTOL UAVs.

The main advantage of VTOL UAVs over fixed-wing UAVs is the ability to hover, allowing them to operate in a small and cluttered environment [1,2].

Comparing a multirotor to a helicopter other advantages can be identified such as greater trust-weight ratio and better manoeuvrability. The various numbers of rotors provide the possibility to use smaller blades instead of one large blade to produce a particular thrust. This leads to less structural and dynamical problems and in the case of an accident, the resulting injury is usually less heavy when compared to a helicopter [1,3,4].

Multirotors are normally controlled by changing the angular speed of rotors, so there is no need for a swashplate which simplifies not only the mechanics but also the maintenance of the system [3–5].

Multirotors can also continue in flight after an actuator failure occurrence when equipped with a failsafe controller. Although the failsafe controller is more straightforward to design for multirotors with 6 or 8 rotors, some controllers were also designed to manage an actuator failure of a quadrotor. Examples of failsafe controllers for a quadrotor can be found in the following works: [6–8].

* Corresponding author.
*E-mail addresses:* chovancova.anezka@gmail.com (A. Chovancová), tomas.fico@stuba.sk (T. Fico), peter.hubinsky@stuba.sk (P. Hubinský), frantisek.duchon@stuba.sk (F. Duchoň).

http://dx.doi.org/10.1016/j.robot.2016.01.011
0921-8890/© 2016 Elsevier B.V. All rights reserved.

Multirotors have become very popular and their usage has spread over all fields of life. Some demanded tasks can be complicated and require the control algorithms to be faster, more efficient and reliable also under windy, uncertain and changing conditions. A disturbance observer (DO) is usually designed to compensate such uncertainties.

Various controllers based on classic or modern control theory were designed. The non-linear or linear model of a quadrotor is used depending on the chosen method.

The linear model of a quadrotor is achieved by linearization of the non-linear model around a hover operating point. Controllers using the linearized model generally perform well around a hover point. When the vehicle goes away from the linearized point, the performance may worsen. Furthermore, the input saturation can cause control failure when large rapid manoeuvres are required [4,5,9,10].

The use of a quaternion instead of Euler angles to describe the dynamics and to design controllers for a quadrotor is becoming very popular nowadays. A feedback signal in the form of a quaternion can be used to design linear and also non-linear attitude and position controllers.

The main advantages of linear controllers are simplicity and ease of implementation to a real platform. The disadvantage of this approach is the use of the linearized model of the quadrotor during the process of designing a controller. Commonly used linear controllers are a Linear Quadratic regulator (LQR) and a Proportional Derivative (PD) controller.

In [10] a cascade attitude controller was proposed. Both an inner and an outer control loop were formed by proportional controllers. The angular velocity and the quaternion were used as feedback signals. From this combination an attitude P2-controller arose. However, the final control law of the P2-controller corresponds to the standard PD controller designed in [3] and [11].

A scheduling LQR controller in [12] was tuned for two different situations; when a quadrotor is far from the reference and when the quadrotor is already tracking.

Another group of control units consists of a wide variety of non-linear controllers. The serious disadvantage of these controllers is the complexity that prevents the wide adoption of nonlinear controllers in real applications.

Among non-linear methods, the backstepping control technique based on the Lyapunov function is widely adopted due to its systematic design and a physically intuitive approach. The proposed control law is based on the compensation of non-linear forces or torques depending on whether an attitude or position controller is being designed. Applying Lyapunov stability analysis proves that the closed-loop system is asymptotically stable. This approach was used to stabilize the quadrotor in [4,5,9,13–15].

A backstepping-based inverse optimal attitude controller (BIOAC) was derived in [4] taking into account the input saturation. In [9] command filters are used to avoid a difficult analytic computation of required command derivatives in each step. The double-integral observer was developed in [13] to design a control law based on the Lyapunov function to track a reference trajectory. A decoupling attitude parametrization was presented in [14]. It allows the design of an independent and straightforward position heading tracking control using the backstepping control technique.

Some works try to overcome uncertainties (e.g. sensor noise, parametric uncertainties, and external disturbance) by designing adaptive controllers.

In another recent work [16], a flight controller based on a Neural Network model has been presented for stabilization and trajectory control.

The problem of disturbance rejection of the attitude subsystem of a quadrotor was addressed in [2]. An acceleration based disturbance observer was applied to a quaternion-based integral sliding mode attitude controller. This combination shows a significant improvement of the performance in position control as well as the compensation of large external forces.

Authors in [17] made a review of control techniques used to control a quadrotor pointing out their advantages and disadvantages.

Despite the interest in quadrotor control techniques, no one, as far as we know, has compared various types of linear and non-linear attitude and position controllers with the use of a disturbance observer. Most studies have only focused on the comparison of attitude controllers or of position controllers using the same attitude controller. Only a few works implemented a disturbance observer. Moreover, the position and linear velocity estimator is designed due to the different sampling periods of the feedback signals and the control loop, and the noise corruption of the feedback signals. This paper also calls into question an assumption that nonlinear controllers exhibit better performance when the external disturbance is present or the implementation of the disturbance observer to linear controllers is sufficient.

The article is structured as follows: firstly a model of the quadrotor dynamics will be derived using a quaternion representation of the attitude. Secondly, various attitude and position controllers will be designed using a quaternion-based feedback. Since the sampling of position is 10 times slower than the control loop period, a state space estimator is developed to predict and filter the actual position. Creating a disturbance observer to identify an attitude and position disturbance is important when a flying platform can operate in an environment, where constant wind and wind gusts can occur. Finally, a performance comparison of the designed controllers is followed by the conclusion.

## 2. Model of the quadrotor

This section focuses on the derivation of the mathematical model of the quadrotor using a quaternion representation. Before a short introduction to the quaternion algebra, advantages of this approach will be outlined.

Even though Euler angles are widely used to represent attitude, this representation suffers from several drawbacks. The most serious problem associated with the use of Euler angles is a "gimbal lock". The "gimbal lock" occurs when two of the rotational axes align and lock together. It implies the loss of one degree of freedom in a three-dimensional space. Another, but not less serious one is a high computational expense caused by the evaluation of quite a large number of trigonometric functions, such as sine and cosine [10].

An alternative representation of the attitude is the use of the Direction Cosine Matrix (DCM) or the quaternion. The DCM solves the singularity problem, but it is still very computationally expensive. Quaternions are free of singularities and are computationally more efficient in comparison with Euler angles or the DCM [3,10,18].

### 2.1. Quaternion algebra

This section is dedicated to build a mathematical background of the quaternion algebra. This knowledge will be exploited when deriving the mathematical model of the quadrotor and designing attitude and position controllers.

A quaternion defines a single rotation $\alpha$ around an axis $\mathbf{r}$. The notation of the rotation is given as a hyper complex number of rank 4 composed of the scalar $q_0 \in \Re$ and the vector $\mathbf{q_{13}} \in \Re^3$. The quaternion formulation is given by the following equation, where $\alpha$ and $\mathbf{r}$ stand for the angle and axis of the rotation, respectively [10–12,18].

$$\mathbf{q} = \left[\cos\frac{\alpha}{2} \quad \mathbf{r}^T \sin\frac{\alpha}{2}\right]^T = \left[q_0 \quad \mathbf{q_{13}}\right]^T$$
$$= \left[q_0 \quad q_1 \quad q_2 \quad q_3\right]^T. \qquad (1)$$

The same orientation can be described by two quaternions $\mathbf{q}$ and $-\mathbf{q}$, which differ in the direction of the rotation. If the rotation angle $\alpha$ meets the condition $|\alpha| \leq \pi$, then the scalar $q_0$ will always be non-negative and the quaternion representation of a given attitude will be unique.

If $\mathbf{q}$ represents the rotation from orientation $A$ to $B$, then the reverse rotation from orientation $B$ to $A$ is given by the conjugate of the quaternion $\mathbf{q}^*$.

$$\mathbf{q}^* = \left[q_0 \quad -\mathbf{q_{13}}\right]. \qquad (2)$$

The norm of the quaternion is calculated the same way as the norm of any complex number (i.e. $N(\mathbf{q}) = q_0 + \mathbf{q}_{13}^T\mathbf{q_{13}}$). If the norm of the quaternion is equal to one, this quaternion is called a unit quaternion.

Given a vector $\mathbf{p}$, the pure imaginary quaternion can be formed as $\mathbf{q_p} = \left[0 \quad \mathbf{p}^T\right]$.

The inverse of the quaternion is given by Eq. (3). If the quaternion satisfies the condition for a unit quaternion, then the inverse is given by the conjugate of the quaternion.

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{N(\mathbf{q})}. \qquad (3)$$

Given two rotations $\mathbf{q_1}$ and $\mathbf{q_2}$, the combined rotation is expressed by their multiplication $\mathbf{q_1} \circ \mathbf{q_2}$. The rotations are non-commutative, hence the quaternion multiplication is non-commutative either, i.e. $\mathbf{q_1} \circ \mathbf{q_2} \neq \mathbf{q_2} \circ \mathbf{q_1}$.

$$\mathbf{q_1} \circ \mathbf{q_2} = \begin{bmatrix} q_{0\_1}q_{0\_2} - \mathbf{q}_{13\_1}^T\mathbf{q}_{13\_2} \\ q_{0\_1}\mathbf{q}_{13\_2} + q_{0\_2}\mathbf{q}_{13\_1} + \mathbf{q}_{13\_1} \times \mathbf{q}_{13\_2} \end{bmatrix}. \qquad (4)$$

Given a unit quaternion $\mathbf{q}$, the multiplication of the quaternion $\mathbf{q}$ and its inverse is commutative, i.e. $\mathbf{q} \circ \mathbf{q}^{-1} = \mathbf{q}^{-1} \circ \mathbf{q}$ and is equal to $\left[1 \quad 0 \quad 0 \quad 0\right]^T$.

The three-dimensional rotation of any vector is given as a quaternion multiplication on the left by the unit quaternion $\mathbf{q}$ and on the right by its conjugate $\mathbf{q}^*$. This mathematical operation can be rewritten as a multiplication of the matrix $\mathbf{R_q}$ and the abovementioned vector. Because we are rotating a vector, only the vector part from the rotation matrix is extracted, as can be seen in Eq. (5), where $\mathbf{q}_{13\times}$ stands for a skew-symmetric matrix. The rotation matrix $\mathbf{R_q}$ is orthogonal; therefore the expression $\mathbf{R}_\mathbf{q}^{-1} = \mathbf{R}_\mathbf{q}^T$ is true.

$$\begin{aligned}\mathbf{R_q} &= (q_0\mathbf{I} + \mathbf{q}_{13\times})^2 + \mathbf{q_{13}}\mathbf{q}_{13}^T \\ &= (q_0^2 - \mathbf{q}_{13}^T\mathbf{q_{13}})\mathbf{I} + 2q_0\mathbf{q}_{13\times} + 2\mathbf{q_{13}}\mathbf{q}_{13}^T \\ &= \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_3q_0) & 2(q_1q_3+q_2q_0) \\ 2(q_1q_2+q_3q_0) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_1q_0) \\ 2(q_1q_3-q_2q_0) & 2(q_2q_3+q_1q_0) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}.\end{aligned} \qquad (5)$$

The derivative of a quaternion is given by the quaternion multiplication of the quaternion $\mathbf{q}$ and the angular velocity of the system $\boldsymbol{\eta}$, which in this case is the quadrotor.

$$\dot{\mathbf{q}} = \begin{bmatrix}\dot{q}_0 \\ \dot{\mathbf{q}}_{\mathbf{13}}\end{bmatrix} = \frac{1}{2}\mathbf{q} \circ \begin{bmatrix}0 \\ \boldsymbol{\eta}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}0 & -\boldsymbol{\eta}^T \\ \boldsymbol{\eta} & -\boldsymbol{\eta}_\times\end{bmatrix}\mathbf{q}. \qquad (6)$$

When designing a controller it is very important to define a quaternion error. The quaternion error is given as the quaternion multiplication of the conjugate of the actual quaternion $\mathbf{q}$ and the desired quaternion $\mathbf{q_d}$.

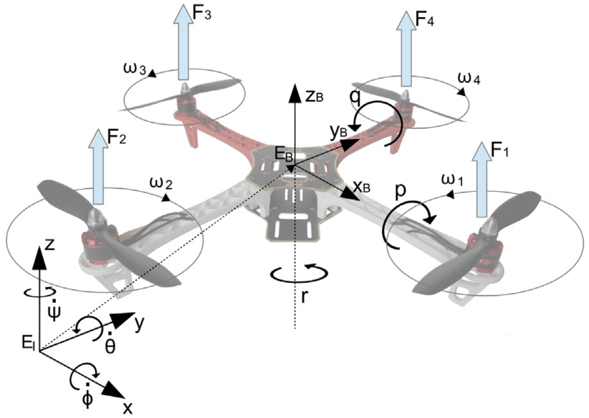

**Fig. 1.** Inertial and body-fixed frame of the quadrotor.

$$\mathbf{q_{err}} = \mathbf{q}^* \circ \mathbf{q_d} = \begin{bmatrix} q_0 q_{0_d} + \mathbf{q}_{13}^T\mathbf{q_{13d}} \\ q_0\mathbf{q_{13d}} - q_{0_d}\mathbf{q_{13}} - \mathbf{q_{13}} \times \mathbf{q_{13d}} \end{bmatrix}. \qquad (7)$$

The quaternion error is the rotation that should be performed to reach the desired orientation.

### 2.2. Mathematical model of quadrotor

The quadrotor is an under-actuated system with 4 control inputs and 6 degrees of freedom.

The approach used more than any other to create a mathematical model of a wide variety of systems is to use the Newton–Euler formulation. It is based on the balance of forces and torques. An alternative energy-based approach is using Euler–Lagrange equations.

In this article the Newton–Euler equations will be derived, while assuming that the quadrotor is a rigid body and the centre of gravity coincides with the body-fixed frame origin.

While developing the mathematical model of the quadrotor, three coordinate frames are considered: the non-moving inertial frame $\mathbf{E_I}$, the body-fixed frame $\mathbf{E_B}$, and the desired frame $\mathbf{E_D}$, which will be considered later, when an attitude controller will be designed.

Each propeller rotates at the angular speed $\omega_i$ producing the corresponding force $\mathbf{F}_i$ [N] directed upwards and the counteracting torque directed opposite to the direction of the particular rotation. Propellers with the angular speed $\omega_2$ and $\omega_4$ [rad/s] spin clockwise and the other two spin counter-clockwise. The variation of the position and orientation is reached by varying the angular speed of a specific rotor. Angular velocities corresponding to the inertial frame $\mathbf{E_I}$ and the body-fixed frame $\mathbf{E_B}$ ($\boldsymbol{\eta}$ [rad/s]) are presented in Fig. 1. Note that east-north-up (ENU) coordinates are used to define all frames.

Assuming a symmetric mass distribution of the quadrotor, the inertia matrix of the quadrotor $\mathbf{I_q}$ [kg m$^2$] is a diagonal matrix. The thrust generated by rotor $i$ is proportional to the square of the angular speed of the rotor and the thrust constant $k_T$ [kg m]. The thrust constant is formed of the air density $\rho$ [kg m$^{-3}$], the radius of the propeller $r$ [m], and the thrust coefficient $c_T$, that depends on blade rotor characteristics, such as number of blades, chord length of the blade, and cube of the rotor blade radius [1,19,20].

$$\mathbf{F}_i = c_T\rho\pi r^4\omega_i^2 = k_T\omega_i^2. \qquad (8)$$

The torque developed around the rotor axis is given as the product of the square of the angular speed and the drag coefficient $k_D$ [kg m$^2$]. The drag coefficient depends on the same factors as the thrust constant and even on the angular acceleration of rotor $i$ multiplied by the moment of inertia of the rotor $J_r$ [1,11,19].

$$\mathbf{D_i} = c_D \rho \pi r^5 \omega_i^2 + J_r \dot{\omega}_i \approx k_D \omega_i^2. \tag{9}$$

The torque created around a particular axis by particular rotors with respect to the body-fixed frame is defined in Eq. (10), where $\Phi_i$ [rad] denotes the multiple of the angle between the arms of the quadrotor [1,20].

$$\boldsymbol{\tau} = \begin{bmatrix} l \sum_{i=1}^{4} \sin \Phi_i F_i \\ -l \sum_{i=1}^{4} \cos \Phi_i F_i \\ \sum_{i=1}^{4} (-1)^i D_i \end{bmatrix} = \begin{bmatrix} l k_T \left(\omega_2^2 - \omega_4^2\right) \\ l k_T \left(\omega_1^2 - \omega_3^2\right) \\ k_D \left(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right) \end{bmatrix}. \tag{10}$$

The thrust $T$ [N] of the quadrotor is always aligned with the $z$ axis in the body-fixed frame and it is given as a sum of all thrusts $\mathbf{F_i}$ generated by the rotors.

$$T = \sum_{i=1}^{4} \mathbf{F_i} = k_T \left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right). \tag{11}$$

The equation describing the rotation of the quadrotor with respect to the body-fixed frame is given by Eq. (12). It involves the torque developed by the four rotors $\boldsymbol{\tau}$ [N m], the centripetal torque $\boldsymbol{\tau_c}$, the gyroscopic torque $\boldsymbol{\tau_g}$ and the disturbance torque $\boldsymbol{\tau_{dist}}$ that contains, among other things, the aerodynamic drag [1,5,11,21].

$$\mathbf{I_q}\dot{\boldsymbol{\eta}} = \boldsymbol{\tau} - \boldsymbol{\eta} \times \mathbf{I_q}\boldsymbol{\eta} - J_r \sum_{i=1}^{4} \boldsymbol{\eta} \times \begin{bmatrix} 0 \\ 0 \\ (-1)^{i+1}\omega_i \end{bmatrix} - \boldsymbol{\tau_{dist}}. \tag{12}$$

The nonlinear model of translation with respect to the inertial frame consists of the force given by the acceleration of the quadrotor with mass $m$ [kg], total thrust $T$ rotated using the rotation matrix $\mathbf{R_q}$, the gravity force and the disturbance force [1,11].

$$m\ddot{\boldsymbol{\xi}} = \mathbf{R_q} \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \mathbf{F_{dist}}. \tag{13}$$

Some methods for designing a controller, such as LQR and PID controller tuning using a pole placement method require a linear model of the system. For this purpose a linearized model is derived from (6), (12) and (13) assuming that the quadrotor is in hover, that means that the thrust of the quadrotor is equal to the product of the earth gravity and the mass of the quadrotor. Change in altitude is caused by the variation of the thrust around the hover point given as $\Delta T = T - mg$.

$$\begin{aligned} \mathbf{I_q}\dot{\boldsymbol{\eta}} &= \boldsymbol{\tau} \\ \dot{\mathbf{q}} &= \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\eta} \end{bmatrix} \\ m\ddot{\boldsymbol{\xi}} &= \begin{bmatrix} mg\,(2q_0q_2) \\ -mg\,(2q_0q_1) \\ \Delta T \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = \begin{bmatrix} mg\,(u_x) \\ -mg\,(u_y) \\ \Delta T \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}. \end{aligned} \tag{14}$$

Parameter values used to describe the dynamics of the quadrotor and later used in the simulation are revealed in Table 1.

## 3. Introduction to proposed control techniques

The translational motion of the quadrotor depends on the attitude of the quadrotor as can be seen from the equations of the motion of the quadrotor (12) and (13). This sets that the quadrotor is a cascade system; therefore, a cascade structure of a controller will be applied to control the position and attitude of the quadrotor.

**Table 1**
Quadrotor parameters used in simulation.

| Symbol | Value | Unit |
|---|---|---|
| $l$ | 0.22 | (m) |
| $m$ | 1.5 | (kg) |
| $g$ | 9.81 | (m/s$^2$) |
| $k_T$ | 8.54858e−06 | (kg m) |
| $k_D$ | 1.3678e−07 | (kg m$^2$) |
| $I_{xx}$ | 0.0108 | (kg m$^2$) |
| $I_{yy}$ | 0.0108 | (kg m$^2$) |
| $I_{zz}$ | 0.0213 | (kg m$^2$) |
| $J_r$ | 1.676e−04 | (kg m$^2$) |

The outer trajectory tracking controller is comprised of the altitude and the position controller as depicted in Fig. 2.

The output of the altitude controller is the desired total thrust $T_d$. The position controller generates the desired position quaternion value $\mathbf{q_{pd}}$ for the attitude controller. Note that the position quaternion will always have the 4th element equal to zero.

$$\mathbf{q_{pd}} = [q_{0pd} \quad q_{1pd} \quad q_{2pd} \quad 0]. \tag{15}$$

The 4th element of the quaternion refers to a rotation around the $z$ axis, which is given by the desired yaw angle and computed using expression (1), where the 2nd and 3rd elements of the quaternion will be zero, as we can see in Eq. (16).

$$\mathbf{q_{zd}} = [q_{0zd} \quad 0 \quad 0 \quad q_{13zd}]. \tag{16}$$

The desired quaternion for the attitude controller is then calculated as a combination of both rotations using the quaternion multiplication (4), which results in Eq. (17).

$$q_d = q_{pd} \circ q_{zd}. \tag{17}$$

The quadrotor is controlled by adjusting angular speeds of the rotors which are spun by BLDC motors. In the process of designing a controller, the dynamics of the BLDC motor will be neglected; it is relatively fast.

Desired rotor angular speeds can be expressed as a product of the inverse matrix gained from Eqs. (10) and (11), and the vector that is composed of the desired torques from the attitude controller and the desired total thrust from the altitude controller.

Note that the desired total thrust is limited to 75% of the maximum possible total thrust, because of the conservation of the attitude control while ascending.

$$\begin{aligned} \omega_1^2 &= \frac{1}{2lk_T}\tau_{yd} - \frac{1}{4k_D}\tau_{zd} + \frac{1}{4k_T}T_d \\ \omega_2^2 &= \frac{1}{2lk_T}\tau_{xd} + \frac{1}{4k_D}\tau_{zd} + \frac{1}{4k_T}T_d \\ \omega_3^2 &= -\frac{1}{2lk_T}\tau_{yd} - \frac{1}{4k_D}\tau_{zd} + \frac{1}{4k_T}T_d \\ \omega_4^2 &= -\frac{1}{2lk_T}\tau_{xd} + \frac{1}{4k_D}\tau_{zd} + \frac{1}{4k_T}T_d. \end{aligned} \tag{18}$$

In this article three different approaches are applied to design a position, altitude and attitude controller, namely the PD controller tuned by pole placement method, an LQR and the backstepping controller.

The first two methods use the linear model of the system so the simplified model of the quadrotor will be used. The backstepping controller will be derived from the non-linear model of the quadrotor. Disturbance forces and torques will be identified using a disturbance observer and implemented in the proposed control law.

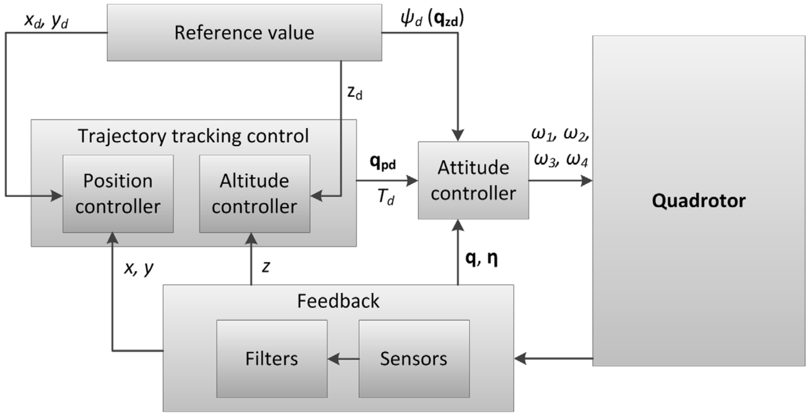

**Fig. 2.** Block diagram of the proposed control structure.

Once the linearized model of the quadrotor in hover is derived given by Eq. (14), it can be written in the form of a state space system. The general formula of the state space representation is given by the following equation.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu}\\ \mathbf{y} &= \mathbf{Cx}.\end{aligned} \tag{19}$$

The state space model of the simplified quadrotor attitude and position is given by Eqs. (20) and (21), respectively. The attitude state vector is $\mathbf{x_A} = \begin{bmatrix}\mathbf{q_{13}} & \boldsymbol{\eta}\end{bmatrix}^T$ and the position state vector is $\mathbf{x_P} = \begin{bmatrix}\boldsymbol{\xi} & \dot{\boldsymbol{\xi}}\end{bmatrix}^T$.

$$\dot{\mathbf{x}}_{\mathbf{A}} = \begin{bmatrix}\mathbf{0}_{3x3} & 0.5\mathbf{I}_{3x3}\\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3}\end{bmatrix}\mathbf{x_A} + \begin{bmatrix}\mathbf{0}_{3x3}\\ \mathbf{I}_{\mathbf{q}}^{-1}\end{bmatrix}\begin{bmatrix}\tau_{xd}\\ \tau_{yd}\\ \tau_{zd}\end{bmatrix} \tag{20}$$

$$\dot{\mathbf{x}}_{\mathbf{P}} = \begin{bmatrix}\mathbf{0}_{3x3} & \mathbf{I}_{3x3}\\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3}\end{bmatrix}\mathbf{x_P} + \begin{bmatrix} & \mathbf{0}_{3x3} & \\ g & 0 & 0\\ 0 & -g & 0\\ 0 & 0 & 1/m\end{bmatrix}\begin{bmatrix}u_{xd}\\ u_{yd}\\ \Delta T_d\end{bmatrix}. \tag{21}$$

### 3.1. PD controller

The PID control method is frequently used in real systems, because of its simplicity. It can be used for a wide variety of systems and succeeds to stabilize both a linear and a non-linear system.

The dynamics of the quadrotor contain a double integrator; hence the integral component of the PID will not be used.

The general form of the PD controller is given by the following equation, where $e(t)$ denotes the error of the controlled signal.

$$\begin{aligned}e(t) &= x_d(t) - x(t)\\ u(t) &= K_p e(t) + K_D\frac{de(t)}{dt}.\end{aligned} \tag{22}$$

A controller for each axis will be designed and tuned separately.

Given the characteristic polynomial of the closed-loop system, the parameters of the PD controller are chosen depending on the location of the poles of the system in s-plane. The system consists of the quadrotor in connection with the PD controller. If $\lambda_1$ and $\lambda_2$ are the desired poles, then the parameters of the PD controller are given by Eq. (23), where $b$ depends on the particular subsystem, i.e. the motion along $x$, $y$ and $z$ axis or the rotation around these axes.

$$\begin{aligned}K_D &= \frac{\lambda_1\lambda_2}{b}\\ K_P &= \frac{-\lambda_1 - \lambda_2}{b}.\end{aligned} \tag{23}$$

### 3.2. LQR

The aim of the optimal control is to operate a system at minimum cost. Because we are solving an infinite-time LQR problem, a quadratic objective function and a state feedback control are expressed by expression (24), where **R** and **Q** are positive semi-definite matrices.

$$\begin{aligned}J_\infty &= \frac{1}{2}\int_0^\infty \left(\mathbf{x}(t)^T\mathbf{Qx}(t) + \mathbf{u}(t)^T\mathbf{Ru}(t)\right)dt\\ \mathbf{u}(t) &= -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{Px}(t) = -\mathbf{Kx}(t).\end{aligned} \tag{24}$$

Supposing the pair $(\mathbf{A}, \mathbf{B})$ is stabilizable and the pair $(\sqrt{\mathbf{Q}}, \mathbf{A})$ is detectable then the solution of the differential Riccati equation converges for $t \to \infty$ to a stationary solution. The matrix **P** is found by solving the algebraic Riccati equation (25).

$$0 = \mathbf{PA} - \mathbf{PBR}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} + \mathbf{A}^T\mathbf{P}. \tag{25}$$

The existence of the stationary solution can be verified by calculating the eigenvalues of the Hamiltonian matrix **Z**.

$$\mathbf{Z} = \begin{bmatrix}\mathbf{A} & -\mathbf{BR}^{-1}\mathbf{B}^T\\ -\mathbf{Q} & -\mathbf{A}^T\end{bmatrix}. \tag{26}$$

Half of the eigenvalues must be stable and may not be pure imaginary numbers and the rest of them must be the mirror value of the first half and unstable.

### 3.3. Backstepping controller

The backstepping control technique is very suitable for cascade systems. The process of designing a recursive controller involves choosing some of the states of the system to be virtual control signals. This control technique is intuitive, but the disadvantage is the complex calculation related to obtaining a derivative of virtual control signals [14,22–24].

The state-space model of the system utilized to design the backstepping controller is given by expression (27).

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= f(\mathbf{x}) + bu.\end{aligned} \tag{27}$$

The parameter $b$ depends on the particular subsystem as it was in the section of designing the PD controller.

Defining errors $e_1$ and $e_2$ as (28), the virtual control $x_{2d}$ is calculated from the chosen Lyapunov function $V_1$ (29) so that the derivative of the given Lyapunov function $\dot{V}_1$ is equal to (30).

$$\begin{aligned}e_1 &= x_{1d} - x_1\\ e_2 &= x_{2d} - x_2\end{aligned} \tag{28}$$

**Table 2**
Components of attitude PD controller.

| Output | $\lambda_1$ | $\lambda_2$ | $K_P$ | $K_D$ |
|---|---|---|---|---|
| $\tau_{xd}$ | −21 | −21 | 9.5256 | 0.9072 |
| $\tau_{yd}$ | −21 | −21 | 9.5256 | 0.9072 |
| $\tau_{zd}$ | −15 | −17 | 10.863 | 1.3632 |

$$V_1 = \frac{1}{2}e_1^2 \tag{29}$$

$$\dot{V}_1 = -c_1 e_1^2 - e_1 e_2. \tag{30}$$

Because the error $e_2$ does not appear in the Lyapunov function $V_1$, a new Lyapunov function $V_2$ is established.

$$V_2 = V_1 + \frac{1}{2}e_2^2. \tag{31}$$

The control signal $u$ is chosen so that the derivative of the Lyapunov function $\dot{V}_2$ acquires the following form.

$$\dot{V}_1 = -c_1 e_1^2 - c_2 e_2^2. \tag{32}$$

Since the Lyapunov function $V_2$ is positive and its derivative is negative as long as parameters $c_1$ and $c_2$ are positive numbers, the system is globally asymptotically stable according to the Lyapunov theory [14,22].

### 3.4. Attitude control

The goal of the attitude controller is to align the body-fixed frame $\mathbf{E_B}$ with the desired frame $\mathbf{E_D}$. This should be achieved following the fastest way that means the closest rotation should be performed.

As already mentioned the same orientation can be represented by two quaternions $\mathbf{q}$ and $-\mathbf{q}$. This fact must be considered while designing an attitude controller. Otherwise the controller can make the quadrotor rotate up to full rotation, which results in higher power consumption and longer time to reach the desired orientation.

To be sure that the quaternion error represents the closest rotation, the scalar part of the quaternion error $q_{0d}$ must be examined. In case of $q_{0d}$ being negative, the rotation angle is more than $\pi$ radians, so $-\mathbf{q_{err}}$ should be used to ensure the closest rotation.

### 3.5. PD attitude controller

Table 2 lists chosen desired poles and corresponding gains for proportional and derivative components of the attitude PD controller.

### 3.6. Attitude LQR

After checking stabilizability of $(\mathbf{A_A}, \mathbf{B_A})$ and detectability of $(\sqrt{\mathbf{Q_A}}, \mathbf{A_A})$, the solution of the Riccati equation and also the optimal regulator gain matrix $\mathbf{K_A}$ was calculated.

$$\mathbf{K_A} = \begin{bmatrix} 4.472 & 0 & 0 & 0.542 & 0 & 0 \\ 0 & 4.472 & 0 & 0 & 0.542 & 0 \\ 0 & 0 & 3.162 & 0 & 0 & 0.877 \end{bmatrix}. \tag{33}$$

Because the simplified model of the attitude of the quadrotor does not contain coupling between its states, both $\mathbf{Q_A}$ and $\mathbf{R_A}$ are diagonal matrices.

### 3.7. Attitude backstepping controller

The Lyapunov function $V_A$ used to design a backstepping attitude controller is given by Eq. (34), where the quaternion error is computed as (7) and $\mathbf{e_2}$ denotes angular velocity error $\boldsymbol{\eta}_\mathbf{d} - \boldsymbol{\eta}$.

$$V_A = |q_{e0}| + \frac{1}{2}\mathbf{e}_2^T\mathbf{e_2}. \tag{34}$$

The derivative of the Lyapunov function $V_A$ is expressed in Eq. (35).

$$\dot{V}_A = -\frac{1}{2}\operatorname{sign}(q_{e0})\, \mathbf{q}_{\mathbf{e13}}^T\boldsymbol{\eta}_\mathbf{d} + \mathbf{e_2}(\dot{\boldsymbol{\eta}}_\mathbf{d} - \dot{\boldsymbol{\eta}}). \tag{35}$$

Eq. (36) shows the desired derivative of the Lyapunov function $\dot{V}_A$, which is negative as long as $c_{1A}$ is a positive constant and matrix $\mathbf{c_{2A}}$ is a positive definite matrix.

$$\dot{V}_A = -\frac{1}{2}c_{1A}\mathbf{q}_{\mathbf{e13}}^T\mathbf{q_{e13}} - \mathbf{e}_\mathbf{2}^T\mathbf{c_{2A}}\mathbf{e_2}. \tag{36}$$

Assuming the virtual control $\boldsymbol{\eta}_\mathbf{d}$ is expressed by (37) then the control law $\boldsymbol{\tau}$ is expressed by formula (38).

$$\boldsymbol{\eta}_\mathbf{d} = \operatorname{sign}(q_{e0})\, c_{1A}\mathbf{q_{e13}} \tag{37}$$

$$\boldsymbol{\tau} = \left(\mathbf{c_{2A}e_2} + \dot{\boldsymbol{\eta}}_\mathbf{d}\right)\mathbf{I_q} + \boldsymbol{\tau}_\mathbf{g} + \boldsymbol{\tau}_\mathbf{c} + \boldsymbol{\tau}_\mathbf{dist}. \tag{38}$$

The value of parameter $c_{1A}$ is 35 and $\mathbf{c_{2A}}$ is a diagonal matrix with vector $\begin{bmatrix}35 & 35 & 3\end{bmatrix}$ on the main diagonal.

## 4. Trajectory tracking controller

The trajectory tracking controller consists of two parts, namely the position and the altitude controller. The altitude controller generates the desired total thrust $T$ in the case of the backstepping controller and the desired change of the thrust around the hover condition $\Delta T$ in the case of the LQR and PD controllers.

The output of the position controller is the same for all of the used methods and its mathematical formulation is given as follows

$$\begin{bmatrix} u_{xd} \\ u_{yd} \end{bmatrix} = \begin{bmatrix} 2q_{0d}q_{2d} \\ 2q_{0d}q_{1d} \end{bmatrix}. \tag{39}$$

The conversion from (39) to $\mathbf{q_{pd}}$ must be done. As already mentioned, the desired position quaternion has the 4th element equal to zero. Therefore, the desired rotation axis $\mathbf{r_d}$ has the following form $\mathbf{r_d} = \begin{bmatrix} r_{xd} & r_{yd} & 0 \end{bmatrix}$ and using Eq. (1) the next expression can be made.

$$\begin{aligned} u_{xd}^2 + u_{yd}^2 &= (2q_{0d}q_{2d})^2 + (2q_{0d}q_{1d})^2 \\ &= \left(2\cos\frac{\alpha}{2}\sin\frac{\alpha}{2}\right)^2 \left(r_x^2 + r_y^2\right). \end{aligned} \tag{40}$$

Assuming that $\mathbf{r_d}$ is a unit vector, the desired rotation $\alpha_d$ and the desired rotation axis $\mathbf{r_d}$ can be derived.

$$\begin{aligned} \alpha_d &= \arcsin\left(\left\|u_{xyd}\right\|\right) \\ r_{xd} &= \frac{r_x}{\left\|u_{xyd}\right\|} \\ r_{xd} &= \frac{r_y}{\left\|u_{xyd}\right\|}. \end{aligned} \tag{41}$$

Given the desired rotation angle and axis, the desired position quaternion can be computed using Eq. (1).

Tuning parameters of all position controllers were as follows: a maximum overshoot of less than 1% of the desired change and minimal possible settling time.

**Table 3**
Components of position PD controller.

| Attitude controller | Output | $\lambda_1$ | $\lambda_2$ | $K_P$ | $K_D$ |
|---|---|---|---|---|---|
| PD+LQR | $u_{px}$ | −25 | −1 | 2.5484 | 2.6504 |
| | $u_{py}$ | −25 | −1 | −2.5484 | −2.6504 |
| | $T_d$ | −9 | −1.45 | 13.05 | 10.45 |
| B | $u_{px}$ | −28.5 | −1 | 2.9052 | 3.0071 |
| | $u_{py}$ | −28.5 | −1 | −2.9052 | −3.0071 |
| | $T_d$ | −10 | −1.44 | 14.4 | 11.44 |

**Table 4**
Components of position backstepping controller.

| Attitude controller | $\mathbf{v_{1P}}$ | $\mathbf{v_{2P}}$ |
|---|---|---|
| PD | $\begin{bmatrix}1 & 1 & 3\end{bmatrix}$ | $\begin{bmatrix}1.75 & 1.75 & 1.5\end{bmatrix}$ |
| LQR | $\begin{bmatrix}1 & 1 & 3\end{bmatrix}$ | $\begin{bmatrix}1.85 & 1.85 & 1.5\end{bmatrix}$ |
| B | $\begin{bmatrix}1 & 1 & 3\end{bmatrix}$ | $\begin{bmatrix}2 & 2 & 1.5\end{bmatrix}$ |

### 4.1. PD trajectory tracking controller

The desired poles were chosen with respect to the connected attitude controller. Chosen poles and corresponding gains for the proportional and derivative component of the trajectory tracking PD controller are listed in Table 3.

### 4.2. Trajectory tracking LQR

After checking stabilizability of $(\mathbf{A_P}, \mathbf{B_P})$ and detectability of $(\sqrt{\mathbf{Q_P}}, \mathbf{A_P})$, the solution of the Riccati equation and also the optimal regulator gain matrix $\mathbf{K_P}$ was calculated.

$$\mathbf{K_P} = \begin{bmatrix} 2.45 & 0 & 0 & 2.739 & 0 & 0 \\ 0 & -2.45 & 0 & 0 & -2.739 & 0 \\ 0 & 0 & 3.873 & 0 & 0 & 3.57 \end{bmatrix}. \tag{42}$$

Because the simplified model of the position of the quadrotor does not contain coupling between its states, both $\mathbf{Q_P}$ and $\mathbf{R_P}$ are diagonal matrices.

### 4.3. Trajectory tracking backstepping controller

The Lyapunov function $V_\mathrm{P}$ used to design the trajectory tracking backstepping controller is given by Eq. (43), where $\mathbf{e_1} = \boldsymbol{\xi}_\mathbf{d} - \boldsymbol{\xi}$ is the position error and $\mathbf{e_2} = \mathbf{v_d} - \dot{\boldsymbol{\xi}}$ is the velocity error.

$$V_\mathrm{P} = \frac{1}{2}\mathbf{e}_1^T\mathbf{e}_1 + \frac{1}{2}\mathbf{e}_2^T\mathbf{e}_2. \tag{43}$$

Given the desired velocity as (44) and the desired derivative of the Lyapunov function $\dot{V}_\mathrm{P}$ as (45) that is negative as long as $\mathbf{c_{1P}}$ and $\mathbf{c_{2P}}$ are positive definite matrices, then the control law $\mathbf{u_P}$ is derived from Eq. (46), where $\ddot{\boldsymbol{\xi}}$ is substituted for (13).

$$\mathbf{v_d} = \mathbf{c_{1P}}\mathbf{e_1} + \dot{\boldsymbol{\xi}}_\mathbf{d} \tag{44}$$

$$\dot{V}_\mathrm{P} = -\mathbf{e}_1^T\mathbf{c_{1P}}\mathbf{e_1} - \mathbf{e}_2^T\mathbf{c_{2P}}\mathbf{e_2} \tag{45}$$

$$\ddot{\boldsymbol{\xi}} = \mathbf{e_1}\left(\mathbf{I} - \mathbf{c}_\mathbf{1P}^2\right) + \mathbf{e_2}\left(\mathbf{c_{1P}} + \mathbf{c_{2P}}\right) + \ddot{\boldsymbol{\xi}}_\mathbf{d}. \tag{46}$$

Parameter $\mathbf{c_{1P}}$ is a diagonal matrix with the vector $\mathbf{v_{1P}}$ on the main diagonal and similarly parameter $\mathbf{c_{2P}}$ has the vector $\mathbf{v_{2P}}$ on the main diagonal. The backstepping trajectory controller was tuned differently for the PD, LQR and the backstepping attitude controller. Chosen parameters are shown in Table 4.

## 5. State space estimator and disturbance observer

The proposed feedback control techniques require feedback signals and disturbance identification. To obtain these signals, various sensors should be implemented to the quadrotor. The most used sensors are an IMU for orientation; GPS, a vision system and an optical flow sensor for position; and an infrared/ultrasonic sensor for altitude.

Signals from these sensors must be processed because all suffer from noise disturbance and depending on the particular sensor from other drawbacks, for example the GPS suffers from the possibility of temporary unavailability, the gyroscope from the flowing bias and the infrared/ultrasonic sensor from environment effects [13].

A state-space estimator is used to compute signals that cannot be measured, or to join various noisy signals in order to remove the noise. The presence of noise can lead to various undesirable effects, e.g. a deterioration of the final position with time.

Under windy conditions, the DO or the robust controller can be designed to improve the performance of the position tracking.

In this work a state space estimator for the position and the linear velocity along with the attitude and position disturbance observers are designed in order to reduce the noise and improve the performance when external disturbances are present.

### 5.1. State space estimator

The state space estimator computes the position and the velocity of the quadrotor that are used in the position control. The measured position contains noise, which is even amplified after a derivation when acquiring the linear velocity.

The most often used sensor for the position is a GPS module. A common GPS module works at a 10 Hz update rate, i.e. new values are obtained every 100 ms.

As it is depicted in Fig. 3 the goal of the estimator is to acquire both the position and the linear velocity of the quadrotor at a 100 Hz rate. The measured variables are the actual position from the pose sensor and the orientation in the form of the quaternion from the IMU. The desired total thrust from the altitude controller is used to estimate a linear acceleration using the mathematical model of the quadrotor.

The estimated linear velocity is computed by merging the velocity from the estimated linear acceleration and the velocity obtained by the derivation of the position from the pose sensor. Because the IMU operates at 100 Hz, the main value is taken from the IMU and the correction is carried out by the filtered derivation of the position.

The calculation of the estimated position is conducted depending on the state of the measured signal. When it is in the transient phase the position is predicted by the double integrated estimated linear acceleration from the IMU and corrected by the measured position every 100 ms. On the contrary, when the signal is steady, the filter is used to smooth the measured signal.

The performance of the observer is shown in Fig. 4, where the real, measured and the estimated signal of the position and the velocity in the $z$ axis are compared.

### 5.2. Disturbance observer

The aim of the disturbance observer is to detect and determine the amplitude of the disturbance calculated as a linear and an angular acceleration.

Fig. 5 shows the block diagram of both observers. They follow the same pattern. The estimated acceleration is calculated using the mathematical model of the quadrotor, then filtered and compared to the acceleration obtained from the measurement.

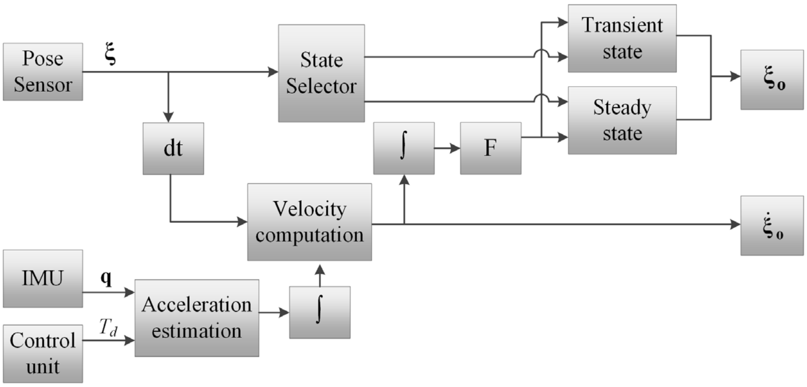

**Fig. 3.** Block diagram of the proposed position and linear velocity state space estimator.

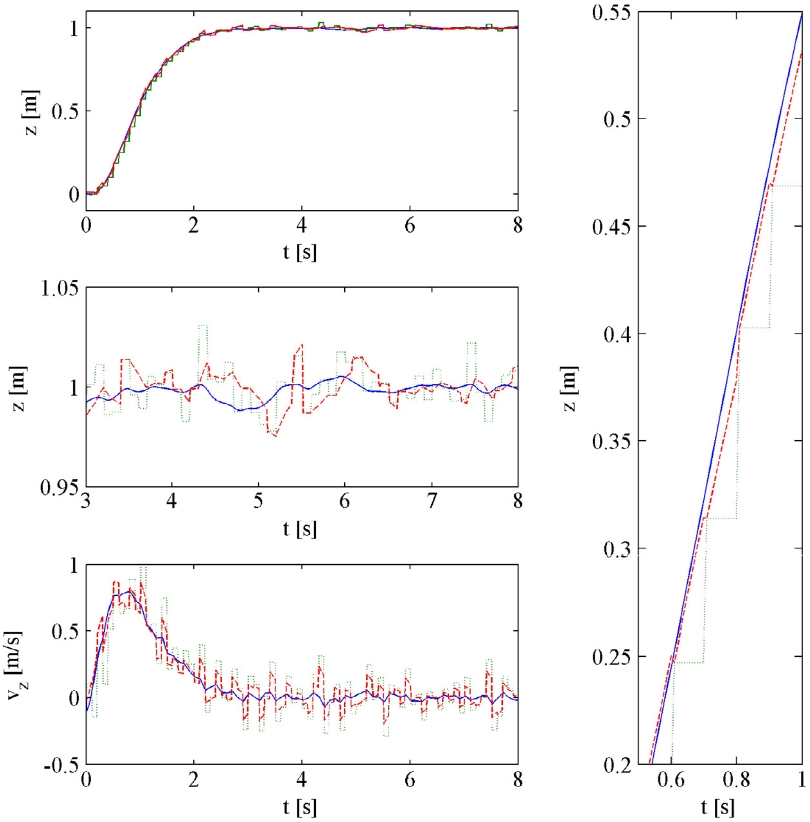

**Fig. 4.** An example of the performance of the state space estimator; the blue line represents the real signal, the green line represents the measured signal and the red line represents the estimated signal. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

When using signals from the control unit, the delay of the motors is implemented so that the measured and estimated values coincide in time.

In the case of the position disturbance observer, the estimated acceleration is computed from the quaternion and the desired total thrust. The linear acceleration, obtained from the accelerometer, also contains a gravity acceleration that is removed using the orientation of the quadrotor.

The attitude disturbance observer computes the estimated angular acceleration from the desired rotor angular speeds. The measured angular acceleration is obtained via derivation of the angular velocity from the IMU.

Fig. 6 highlights the identification of an external disturbance that is applied at time $t = 20$ s and disappears at time $t = 40$ s. The position disturbance corresponds to the force equal to the vector [2 2 −5]. This force is identified in the form of a linear acceleration and implemented to the control algorithm. The external disturbance shown in Fig. 6 will be used later to verify the performance of the designed controllers under the presence of external disturbances.

## 6. Results

This section deals with the verification of the controllers designed in previous sections through a numerical simulation.

The non-linear model from Eqs. (12) and (13) was used to carry out all simulations. We adjusted several simulation conditions in order to set the simulation closer to reality. The sampling period of the simulation was set to 10 ms. All signals used to control the quadrotor were sampled at this rate, except for the position, which is sampled at a frequency of 10 Hz. Furthermore, some restrictions related to the actuators were applied based on measurements done in [25]. The maximum angular speed of the actuators was set to 650 rad/s and the minimum to 150 rad/s. Also, the delay of the

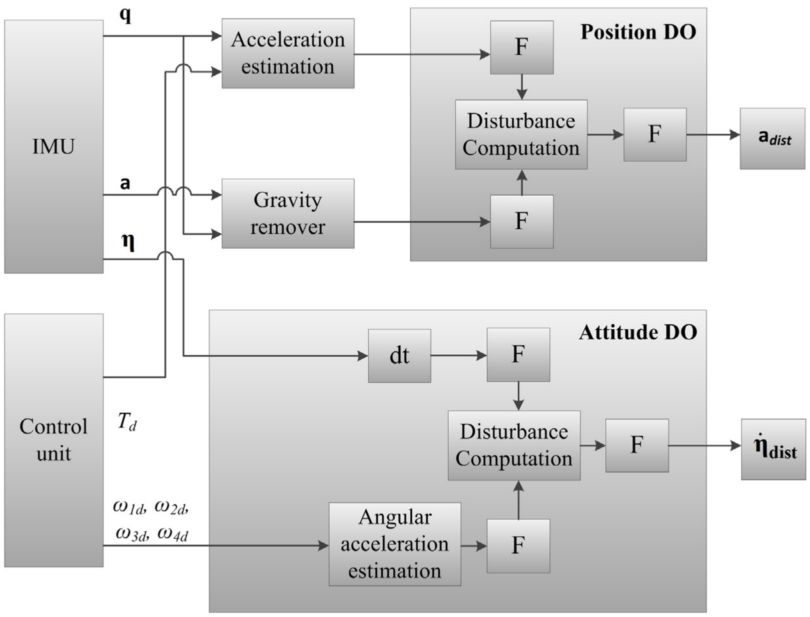

**Fig. 5.** Block diagram of the proposed attitude and position disturbance observers.

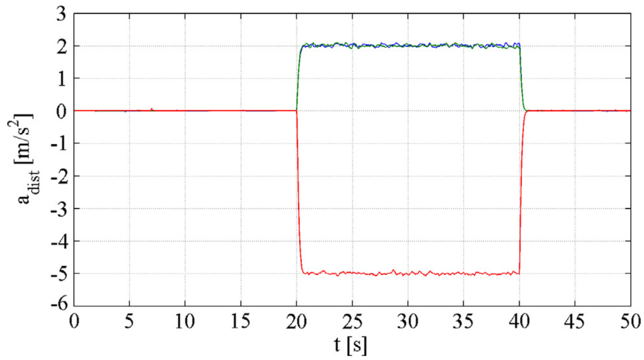

**Fig. 6.** Identification of the disturbance that starts to act on the system at time $t = 20$ s and ends at time $t = 40$ s. The blue, green and red lines represent the amplitude of force equal to the vector [2 2 −5] converted to the linear acceleration in the $x$, $y$ and $z$ axis, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

actuators was implemented due to the use of the electronic speed controller (ESC).

Moreover, sensor noise was implemented to the measured feedback signals. Gaussian noise with a variance of 1e-6 was added to the angular velocity. The quaternion is computed from the angular velocity, therefore there is no need to add another noise particularly to the quaternion.

The position was deteriorated with Gaussian noise with a variance of 1e-4.

The evaluation of the designed controllers was done under two conditions: (a) disturbance free condition and (b) constant wind condition. In both scenarios the quadrotor has to follow the same trajectory; firstly the quadrotor takes off to an altitude of 1 m, then at time $t = 15$ s flies from point [0 0 1] to point [5 5 6] with respect to the inertial frame $\mathbf{E_I}$ and also rotates 90° around the $z$ axis. At time $t = 20$ s it rotates back and at time $t = 35$ s comes back to point [0 0 1] and lands at time $t = 45$ s.

The quadrotor is an under-actuated system, i.e. all desired movements are interconnected and generated by actuators with performance restrictions. We consider the position to have a higher priority than the rotation around the $z$ axis; therefore the desired yaw angle was filtered by a low pass filter.

**Table 5**
Quality indicators of controllers performance: Integral of absolute error and thrust.

| Att.+Pos. | $I_{AEx}$ (m) | $I_{AEy}$ (m) | $I_{AEz}$ (m) | $I_{AT}$ (N) |
|---|---|---|---|---|
| PD+PD | **12.98** | 12.96 | 12.31 | 495.610 |
| LQR+PD | **12.76** | **12.74** | 12.13 | 495.578 |
| B+PD | 13.31 | 13.26 | 12.66 | 495.280 |
| PD+LQR | 13.69 | 13.54 | 13.33 | **495.084** |
| LQR+LQR | 13.35 | 13.26 | 13.15 | 495.098 |
| B+LQR | 13.88 | 13.66 | 13.71 | **494.727** |
| PD+B | **12.98** | 13.03 | **11.70** | 495.915 |
| LQR+B | **12.75** | **12.81** | **11.57** | 495.989 |
| B+B | 13.30 | 13.38 | 12.07 | 495.466 |

**Table 6**
Quality indicators of controllers performance: Settling times.

| Att.+Pos. | $T_{Sr\_xy}$ (s) | $T_{Sf\_xy}$ (s) | $T_{Sr\_z}$ (s) | $T_{Sf\_z}$ (s) |
|---|---|---|---|---|
| PD+PD | 3.2 | **2.65** | **2.00** | 3.7 |
| LQR+PD | **2.7** | **2.90** | **2.00** | 3.7 |
| B+PD | 3.0 | 3.20 | **2.10** | 3.7 |
| PD+LQR | 3.5 | 3.50 | 3.35 | **2.6** |
| LQR+LQR | 3.1 | 3.35 | 3.30 | **2.7** |
| B+LQR | 3.1 | 3.30 | 3.30 | **2.7** |
| PD+B | 3.0 | **2.65** | **2.10** | 3.0 |
| LQR+B | **2.7** | **2.90** | 2.40 | 3.0 |
| B+B | **2.9** | 3.10 | 2.30 | 3.0 |

### 6.1. Trajectory tracking without the external disturbance

Various quality indicators are chosen to discuss the well-working and efficiency of the proposed algorithms, such as the integral of absolute error $I_{AE}$, the integral of absolute total thrust $I_{AT}$ and settling times. From the dynamics it is evident, that settling time related to the $x$ and $y$ axis should be the same for distancing as well as for approaching, but there is also a change in the altitude, which can result in different settling times. For the mentioned reason the settling time for distancing and ascending $T_{Sr}$ and also for approaching and descending $T_{Sf}$ will be determined.

All combinations of the attitude and the position controllers were simulated and the results are shown in Tables 5 and 6 in the form of quality indicators. The two best values of each quality indicator are highlighted in bold.

We can see that all proposed controllers are energetically approximately equally efficient. When comparing integrals of

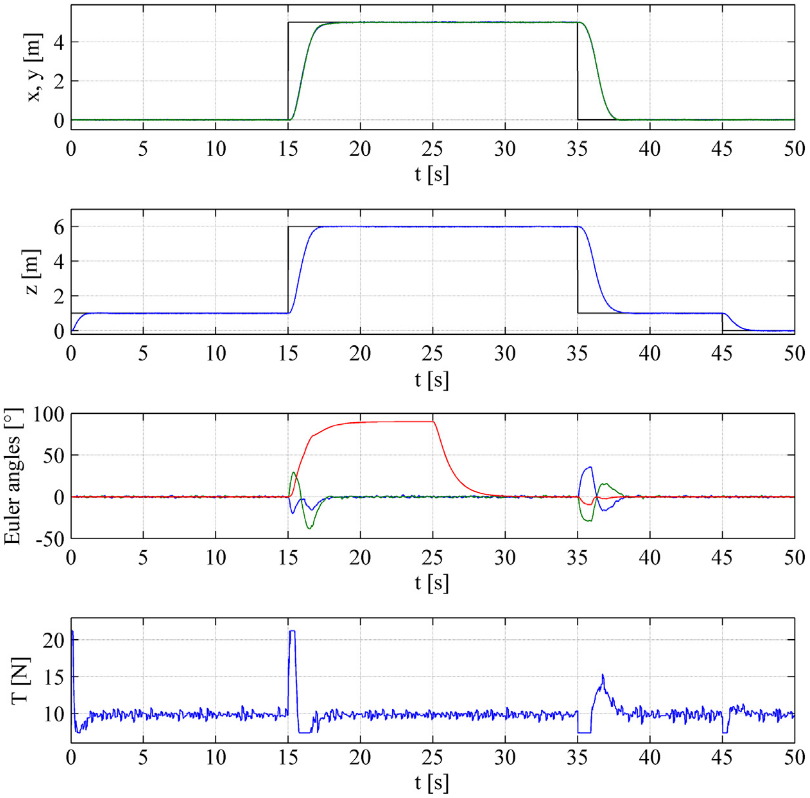

**Fig. 7.** The performance of the controller comprised of the PD attitude and the backstepping position controller. At time $t = 15$ s, the quadrotor is ordered to move 5 m in all axes and to rotate 90° around the $z$ axis; at time $t = 20$ s the quadrotor rotates back around the $z$ axis and at time $t = 35$ s is ordered to come back to the initial position. The third subplot depicts the attitude of the quadrotor in terms of Euler angles, i.e. roll (blue line), pitch (green line) and yaw (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

absolute error, the controllers with the PD and the backstepping position controller show better performance than controllers with the LQR position controller. Settling times are compared in the form of proportions. The worst performance corresponds to controllers with the LQR position controller, where settling time is 1.4–1.6 times longer than the shortest settling time, excluding the settling time corresponding to the descending of the quadrotor, which was the best achieved value. Controllers with the PD and the backstepping position controllers exhibit comparable behaviour. After considering all quality indicators, we chose the following controllers to be the best designed and tuned controllers of all the abovementioned controllers without the occurrence of the external disturbance: the LQR attitude controller in connection with the PD and the backstepping position controller, and the PD attitude controller with the backstepping position controller.

Fig. 7 highlights the performance of one of the best controllers, namely the PD attitude and the backstepping position controller.

### 6.2. Trajectory tracking with the occurrence of external disturbance

The quadrotor often flies in an environment where external disturbances can occur. Therefore a verification of the designed controllers is done under the presence of an external disturbance force. The desired trajectory remains the same as in the previous section, but a force given by the vector [2 2−5] is applied at time $t = 20$ s. The disturbance force then disappears at time $t = 40$ s.

The quality indicators from the previous section are expanded by adding the settling time after the occurrence of the disturbance $T_{SD}$, the settling time after the disappearance of the disturbance $T_{ED}$ and maximum overshoots in both mentioned cases $O_{SD}/O_{ED}$. The settling time for distancing and ascending $T_{Sr}$ is the same as in the previous case; hence we will determine only the settling time for approaching and descending $T_{Sf}$ that will change because of the presence of the external disturbance.

**Table 7**
Quality indicators of controllers performance: Integral of absolute error and thrust.

| Att.+Pos. | $I_{AEx}$ (m) | $I_{AEy}$ (m) | $I_{AEz}$ (m) | $I_{AT}$ (N) |
|---|---|---|---|---|
| PD+PD | 13.77 | 13.74 | 10.97 | 600.885 |
| LQR+PD | 13.63 | **13.57** | 11.02 | 600.740 |
| B+PD | 13.98 | 13.98 | 11.06 | 600.736 |
| PD+LQR | 14.33 | 14.16 | 12.63 | 600.444 |
| LQR+LQR | 14.07 | 13.95 | 12.67 | 600.310 |
| B+LQR | 14.55 | 14.35 | 12.94 | **599.981** |
| PD+B | **13.51** | **13.56** | **10.89** | 601.262 |
| LQR+B | **13.33** | **13.36** | 10.94 | **601.206** |
| B+B | 13.79 | 13.78 | **10.85** | 600.880 |

All combinations of the attitude and the position controllers were simulated and the results in the form of quality indicators are listed in Tables 7–9. The two best values are indicated in bold.

As in the previous analysis, the proposed controllers are energetically very similar and combinations with the LQR position controllers reach the highest value of the integral of the absolute error.

From the results given by Table 8, we can see a dependency between the attitude controller used and the value of the maximum overshoot in the $xy$ plane. The best performance is achieved by using the backstepping attitude controller, a slightly worse one with the LQR attitude controller and the worst with the PD attitude controller.

Naturally, the maximum overshoot in the $z$ axis depends only on the position controller. The smallest overshoot of 0.12 m was achieved by the PD position controller. The maximum overshoot created by controllers with the backstepping position controller was around 1.5 times higher than that, but the highest value was obtained by the LQR position controller.

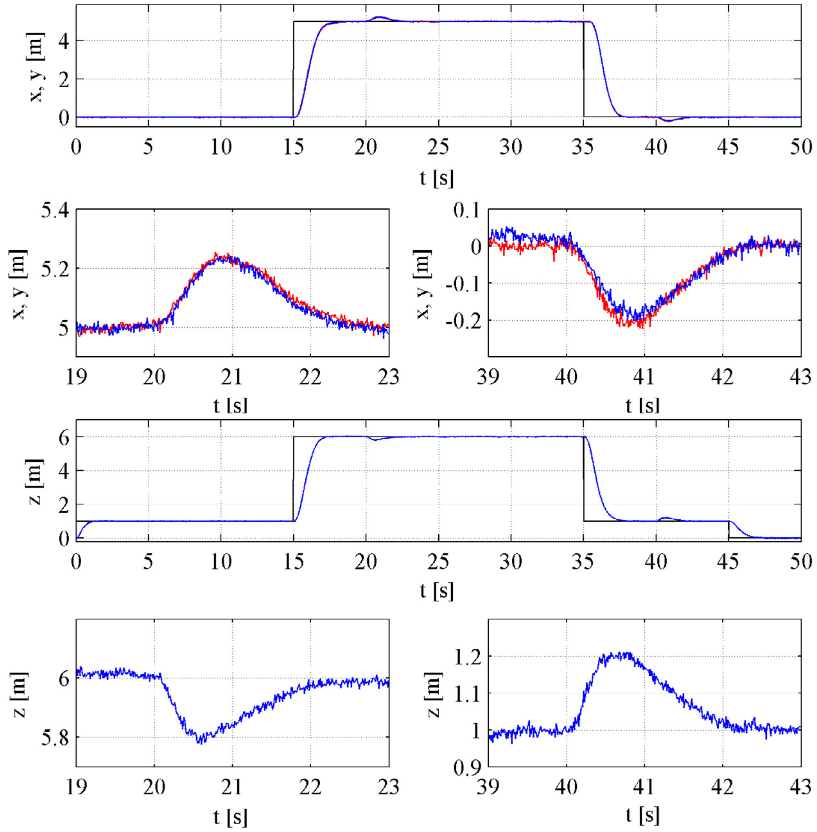

**Fig. 8.** The performance of the controller comprised of the PD attitude and the backstepping position controller. The desired trajectory is the same as in the previous section and the disturbance force given by the vector [2 2−5] is applied at time $t = 20$ s and it disappears at time $t = 40$ s. In the first two subplots the red and blue lines represent the $x$ and $y$ position of the quadrotor, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 8**
Quality indicators of controllers performance: Max. Overshoots.

| Att.+Pos. | $O_{SD\_xy}$ (m) | $O_{ED\_xy}$ (m) | $O_{SD\_z}$ (m) | $O_{ED\_z}$ (m) |
|---|---|---|---|---|
| PD+PD | 0.25 | 0.20 | **0.13** | **0.13** |
| LQR+PD | 0.20 | 0.18 | **0.13** | **0.13** |
| B+PD | **0.14** | **0.13** | **0.13** | **0.12** |
| PD+LQR | 0.24 | 0.20 | 0.24 | 0.24 |
| LQR+LQR | 0.20 | 0.19 | 0.25 | 0.24 |
| B+LQR | **0.15** | **0.15** | 0.25 | 0.24 |
| PD+B | 0.23 | 0.20 | 0.20 | 0.20 |
| LQR+B | 0.18 | 0.18 | 0.20 | 0.20 |
| B+B | **0.14** | **0.13** | 0.20 | 0.20 |

Almost all controllers manage to descend from 6 m of altitude to 1 m during the presence of the external disturbance without overshooting except the controller comprised of the backstepping attitude controller and the PD position controller. The maximum value of this overshoot was 0.4 m.

The settling time of the deviation of the position caused by the external disturbance is slightly longer when the LQR position controller is used. Settling times of the rest of the controllers are comparable to each other.

Taking into account all quality indicators, we consider the best combinations to be the PD attitude controller in connection with the PD and the backstepping position controller, and the combination of the backstepping controller for both the attitude and the position of the quadrotor.

The performance of the PD attitude and the backstepping position controller is depicted in Fig. 8.

## 7. Conclusion and future work

In this article different control methods were used to design attitude and also position controllers, namely the PD, the LQR and the backstepping control techniques. All controllers use the quaternion representation of the attitude. The output of the position controllers is the desired quaternion and the desired thrust of the quadrotor. The attitude controllers use the quaternion error to compute desired torques.

**Table 9**
Quality indicators of controllers performance: Settling times.

| Att.+Pos. | $T_{SD\_xy}$ (s) | $T_{ED\_xy}$ (s) | $T_{SD\_z}$ (s) | $T_{ED\_z}$ (s) | $T_{Sf\_xy}$ (s) | $T_{Sf\_z}$ (s) |
|---|---|---|---|---|---|---|
| PD+PD | 2.1 | 1.90 | **1.50** | **1.25** | **2.70** | 3.50 |
| LQR+PD | 2.1 | 1.95 | **1.50** | **1.30** | 2.95 | 3.50 |
| B+PD | **1.9** | **1.80** | **1.50** | **1.30** | 3.40 | 3.00[a] |
| PD+LQR | 2.3 | 1.90 | 2.20 | 1.90 | 3.60 | **2.25** |
| LQR+LQR | 2.3 | 2.00 | 2.20 | 2.00 | 3.40 | 2.70 |
| B+LQR | **1.9** | 1.90 | 3.00 | 2.00 | 3.10 | **2.00** |
| PD+B | 2.0 | **1.85** | **1.70** | 1.70 | **2.65** | 2.50 |
| LQR+B | **1.9** | 1.90 | **1.70** | 1.70 | 2.90 | 2.60 |
| B+B | **1.8** | **1.80** | 2.25 | 1.70 | 3.10 | 2.80 |

[a] There was an overshoot of 0.4 m when moving from 6 m of altitude to 1 m.

Moreover an estimator of the linear velocity and the position along with a disturbance observer was designed to improve trajectory tracking under the presence of an external disturbance.

Two scenarios were proposed to verify all combinations of attitude and position controllers. The first one was following the desired trajectory, where the movement along all axes and also the rotation around the $z$ axis were ordered. The occurrence of an external disturbance was added in the second scenario.

In order to set the simulation closer to reality some modifications were made, namely the consideration of actuator restrictions, the different sampling for feedback signals and the addition of noise to all feedback signals.

The quality indicators were chosen in order to be able to compare the performance of all controllers. Some of the controllers exhibit very similar behaviour, so we choose the three best controllers for each scenario. The only controller overlapping both scenarios was the PD attitude controller in connection with the backstepping position controller.

In the future we are planning to implement at least one of the chosen controllers to the real platform and verify the control algorithm in a real environment.

## Acknowledgements

This work was supported by projects VEGA 1/0178/13, KEGA 003STU-4/2014 and Req-00347-0001.

## References

[1] A. Chovancová, T. Fico, L'. Chovanec, P. Hubinský, Mathematical modelling and parameter identification of quadrotor (a survey), in: Proc. MMaMS, Starý Smokovec, SK, 2014, pp. 172–181.

[2] T. Tomić, Evaluation of acceleration-based disturbance observation for multicopter control, in: Proc. ECC, Strasbourg, FR, 2014, pp. 2937–2944.

[3] A. Alaimo, V. Artale, C.R.L. Milazzo, A. Ricciardello, PID controller applied to hexacopter flight, J. Intell. Rob. Syst. 72 (1–4) (2013) 261–270.

[4] A. Honglei, L. Jie, W. Jian, W. Jianwen, M. Hongxu, Backstepping-based inverse optimal attitude control of quadrotor, Int. J. Adv. Robot. Syst. 10 (2013) 1–9.

[5] X. Huo, M. Huo, H.R. Karimi, Attitude stabilization control of a quadrotor UAV by using backstepping approach, Math. Probl. Eng. 2014 (2014) 1–9.

[6] A.S. Cândido, R.K.H. Galvão, T. Yoneyama, Actuator fault diagnosis and control of a quadrotor, in: Proc. INDIN, Porto Alegre RS, BR, 2014, pp. 310–315.

[7] V. Lippiello, F. Ruggiero, D. Serra, Emergency landing for a quadrotor in case of a propeller failure: A PID based approach, in: Proc. SSRR, Lake Toya, Hokkaido, JP, 2014, pp. 1–7.

[8] V. Lippiello, F. Ruggiero, D. Serra, Emergency landing for a quadrotor in case of a propeller failure: A backstepping approach, in: Proc. IROS, Chicago, USA, 2014, pp. 4782–4788.

[9] S. Zhao, W. Dong, J.A. Farrell, Quaternion-based trajectory tracking control of VTOL-UAVs using command filtered backstepping, in: Proc. ACC, Washington, DC, USA, 2013, pp. 1018–1023.

[10] E. Fresk, G. Nikolakopoulos, Full quaternion based attitude control for a quadrotor, in: Proc. ECC, Zürich, CH, 2013, pp. 3864–3869.

[11] A. Alaimo, V. Artale, C. Milazzo, A. Ricciardello, L. Trefiletti, Mathematical modeling and control of a hexacopter, in: Proc. ICUAS, Atlanta, USA, 2013, pp. 1043–1050.

[12] E. Reyes-Valeria, S. Camacho-Lara, J. Guichard, LQR control for a quadrotor using unit quaternions: Modeling and simulation, in: Proc. CONIELECOMP, Cholula, MX, 2013, pp. 172–178.

[13] X. Wang, B. Shirinzadeh, M.H. Ang, Nonlinear double-integral observer and application to quadrotor aircraft, IEEE Trans. Ind. Electron. 62 (2) (2015) 1189–1200.

[14] P. Monte, B. Lohmann, Trajectory tracking control for a quadrotor helicopter based on backstepping using a decoupling quaternion parametrization, in: Proc. MED, Create, GR, 2013, pp. 507–512.

[15] S. Bouhired, M. Bouchoucha, M. Tadjine, Quaternion-based global attitude tracking controller for a quadrotor UAV, in: Proc ICSC, Algiers, Algeria, 2013, pp. 933–938.

[16] V. Artale, M. Collotta, C. Milazzo, G. Pua, A. Ricciardello, An adaptive trajectory control for UAV using a real-time architecture, in: Proc. ICUAS, Orlando, USA, 2014, pp. 32–39.

[17] A. Zulu, S. John, A review of control algorithms for autonomous quadrotors, Open J. Appl. Sci. 4 (2014) 547–556.

[18] J.S. Yuan, Closed-loop manipulator control using quaternion feedback, IEEE J. Robot. Autom. 4 (4) (1988) 434–440.

[19] F. Rinaldi, S. Chiesa, F. Quagliotti, Linear quadratic control for quadrotors UAVs dynamics and formation flight, J. Intell. Rob. Syst. 70 (1–4) (2013) 203–220.

[20] R. Mahony, V. Kumar, P. Corke, Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor, IEEE Robot. Autom. Mag. 19 (3) (2012) 20–32.

[21] L. Derafa, T. Madani, A. Benallegue, Dynamic modelling and experimental identification of four rotors helicopter parameters, in: Proc. IEEE International Conference on Industrial Technology, Mumbai, IN, 2006, pp. 1834–1839.

[22] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, H. Li, Control and navigation framework for quadrotor helicopter, J. Intell. Robot. Syst. 70 (1–4) (2013) 1–12.

[23] R. Kristiansen, P.J. Nicklasson, J.T. Gravdahl, Satellite attitude control by quaternion-based backstepping, IEEE Trans. Control Syst. Technol. 17 (1) (2009) 227–232.

[24] Y. Tan, J. Chang, H. Tan, Advanced nonlinear control strategy for motion control systems, in: Proc. Power Electronics and Motion Control Conference, Beijing, CN, 2000, pp. 116–121.

[25] A. Chovancová, T. Fico, Design of a test bed for parameter identicification of actuator used in quadrotors, in: Proc. ELITECH'14, Bratislava, SK, 2014, pp. 1–6.

**Anežka Chovancová** was born in Dolný Kubín, Slovakia in 1987. She received the M.Sc. degree in cybernetics from the Slovak University of Technology, Bratislava, in 2012. In this same year, she started her doctoral studies in the field of Automation and Control in the Slovak University of Technology in Bratislava. Her research interests include modelling and control of UAV.

**Tomáš Fico** born in 1988 received the Engineer degree in Robotics from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava in 2012. In 2012 he started the Ph.D. study from the same University. His research and activities are in the areas of microcontrollers, RADAR technology, robotics and automation.

**Peter Hubinský** born in 1962 has received M.Sc. and Ph.D. Degrees at Faculty of Electrical Engineering from the Slovak University of Technology in 1985 and 1992, respectively. After shorter work in praxis in the area of industrial robotics he has returned back and worked in role of assistant, since 1999 Assoc. Professor and 2011 in role of Professor in area of robotics, mechatronics and dynamic system control. He cooperated for long period with foreign partners mainly in Germany. His current research interests are resonance free control of mechatronic systems, mobile robot visual control and other related areas.

**František Duchoň** was born in Krnov, Czech Republic, in 1981. He received the M.Sc. degree in Automation and Ph.D. in Automation and Control from the Slovak University of Technology, Bratislava, in 2005 and 2010 respectively. He received degree in Automation from Slovak University of Technology, Bratislava, in 2011. From 2006 to 2008, he was Ph.D. student. From 2008 to 2011, he was an Intern Researcher with the Department of Robotics and Artificial Intelligence, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava. Since 2011, he is an Associate Professor. His research interests include mobile robotics, robotic manipulators and image processing.