International Review of Aerospace Engineering (I.RE.AS.E), Vol. 11, N. 5
ISSN 1973-7459 October 2018

# Dynamic Modeling and Flight Control Design for Multicopter

Drilon Bunjaku[1], Gorjan Nadzinski[2], Mile Stankovski[2], Jovan Stefanovski[3]

**Abstract** – *This paper presents a step by step dynamic modeling of the multicopter – quadcopter based on Newton Euler formalism, including the dynamics of the motor and the propellers. The linearization of the nonlinear mathematical model of the quadrotor is derived systematically when* $\psi \neq 0$. *Consequently, the simulation model of the flight controller based on the cascade control has been designed, which ensures a stabilization of the quad rotor and a robust-like trajectory tracking performance. The proposed cascade control strategy including the outer inverse dynamics provides controlling of the yaw orientation angle. The comparison of a use of a linear cascade PID-PD and the combination of PID-MPC is performed on 3D referent trajectories tracking with and without the presence of periodic external torque disturbances. The results are obtained from the MATLAB simulation model.*

***Keywords****: Quadrotor dynamics, Control system design, Flight controller, Trajectory tracking*

## Nomenclature

| | |
|---|---|
| AV | Aerial vehicle |
| CCD | Cascade control designs |
| CG | Center of gravity |
| DOF | Degrees of freedom |
| ESC | Electronics speed control |
| MPC | Model predictive control |
| NWU | North west up |
| UAV | Unmanned Aerial Vehicle |
| $\phi, \theta, \psi$ | Euler angles (roll, pitch and yaw) |
| $p, q, r$ | Angular rate in body coordinate |
| $m$ | Total mass of the body |
| $g$ | Gravitational acceleration |
| $d$ | Length of a single arm from CG |
| $J_{xx}, J_{yy}, J_{zz}$ | Moment of inertia in (x, y and z-axis) |
| $\Omega_n$ | nth propeller rotational speed |
| $V_b, V_n$ and $\omega_b, \omega_n$ | linear and angular velocity vectors in the body (b), respectively in NWU coordinate systems |
| $C_T, C_Q$ | Thrust and torque constant of the motor with propeller |
| $C_t, C_q$ | Propeller aerodynamic constants |
| $B$ | The circular area covered by rotating propeller |
| $R$ | Radius of the propeller |
| $\rho$ | Air density |
| $\sigma$ | Joystick command in metaphoric explanation |
| $R_b^n$ | Rotational matrix from body (b) to earth (n) coordinate system, and vice versa |
| $S^{-1}$ | Lumped transformation matrix |
| $F_n$ | Total force acting on the quadcopter body represented in earth coordinate |
| $M_b$ | Moment vector acting on the quadcopter body represented in body coordinate |
| $T_i$ | Thrust of the i-th motor |
| $Q_i$ | Torque of the i-th motor |
| $N_p, N_c$ | The prediction and the control horizon |

## I. Introduction

It is a known fact from history, science, and art, that Leonardo da Vinci is the first one who has drawn many designs for flying machines and mechanisms and has written about them, including ornithopters, fixed-wing gliders, rotorcraft, and parachutes. Since December 17, 1903, when Wright brothers invented and built the world's first successful airplane and made the first controlled, powered, and sustained heavier-than-air human flight, in the following one hundred years, many types of aircraft emerged.

In general, AV systems have been used for a long time as the main convenient transportation method in the shortest time over long distances. The greatest advancement of AV systems has been made during First and Second World Wars, in military bases. Nowadays, the advancement in technology, embedded systems and especially the development of artificial intelligence, triggered military and research interests on extending their research from conventional moving vehicles, to unmanned conventional and unconventional moving vehicles, such as UGV (Unmanned Ground Vehicles), UAV (Unmanned Aerial Vehicles), UUV (Unmanned Underwater Vehicles), see ([5-7], [10], [11], [13]).

This paper focuses only on the dynamic modeling and control system design for the Multicopter Aerial Vehicle which could easily be extended to UAV systems. An unmanned aerial vehicle (UAV or drone) is an aircraft that is equipped with necessary data processing units, sensors, automatic control, and communications systems, and it is capable of performing autonomous flight missions without the interference of a human pilot [2]. While UGV (which depending on their configurations are characterized as non-holonomic systems) are capable of carrying heavier payload systems, when it comes to take stairs and to avoid obstacles in the dynamic environment,

 https://doi.org/10.15866/irease.v11i5.15512

those tasks are understandably easier for UAV systems.

Depending on various applications and missions, AV systems according to [2] are classified into four major categories: Fixed wings, Flapping wings, Rotorcraft, and Hybrid systems.

The Rotorcraft category is actually the second leading category based on its application. One of its most lasting members is the helicopter, due to the heavy payload and the possibility of vertical takeoff and landing, which is the most advantageous aspect of the multicopters. In the last decade, multicopters have become very attractive due to the advancement of technology and the possibility for reduction of their mechanical size [1],[3-7]. Consequently, this had a direct impact on the reduction of the price for commercial applications. For instance, the quad-copter is one of the most widely used rotorcrafts in commercial applications, due to its capability of moving in all directions. The main disadvantage of multicopters is the inefficiency in power consumption. The energy endurance during long-distance flights and the power efficiency of the fixed wings, could not be beaten neither from the category of multicopters nor from other categories as well. Multicopters find application in many sectors, such as monitoring hazardous environments, traffic, weather, monitoring pipelines, firefighter, police investigation, first aid emergency and rescue, etc. Besides major applications for video recording and photography, drones have recently been also used in building bridges, inspecting power lines, etc. Since aerial multicopter systems do not require a pilot, it is possible to make them smaller, lighter, safer and cheaper. Recently, there have also been some multicopter applications for daily personal transportation, but such systems are not reliable enough and they are still subjected to ongoing researches.
The design of control strategies for quadcopters has been an attractive field for many researchers and many control algorithms have been proposed. In [15], authors have developed a cascade control design for quadcopter based on PID regulators and they have provided the tracking of different referent trajectories. In [16], authors have identified a dynamic model based on experimental measurements and they have been able to control the plant by using PID control. The comparison of two control designs PID and LQ, related to the same task of stabilizing the quadcopter from different initial orientations, is given in [1]. Two full linear control (single loop – noncascade control) designs for quadcopter based on LQ and $H_\infty$ are presented in [17]. The comparison of these control laws is made by tracking a referent trajectory. In general, it has been discovered that the full linear control structure is not very reliable to the real application of the quadcopter, due to its high sensitivity in presence of parameter uncertainties, (see [18], and references therein). In [19], a cascade control structure is designed such that MPC provides tracking of the referent trajectory, while a nonlinear $H_\infty$ controller ensures the stabilization of the rotational motions. The comparison of PID and MPC control structure for one DoF is elaborated for the tracking of a referent trajectory in [20].
This paper is another contribution in the design of the control structure for a quadcopter with a cascade PID - Linear MPC on tracking a referent trajectory. The motivation for the paper comes from the fact that a lot of the recent researches on the design of linear cascade controllers for a quadcopter focus on controlling the dynamic model of the multicopter for the case where the yaw angle is $\psi = 0$ [14]-[22]; it simplifies the linearized mathematical model of the quadrotor, while on the other hand the case where $\psi \neq 0$ is generally avoided. Thus, in this paper, a systematic dynamic modeling is considered, including the outer loop inverse dynamics, which makes the yaw angle controllable. Considering that MPC is popular for its capability of dealing with constraints on the output and on the input signals, authors have designed a linear cascade PID-MPC control structures, which provide control on all measured states.
The organization of the paper is as follows: in Section II, the general dynamic modeling of the x-frame quadcopter is presented. This section begins with the kinematic model, followed by the subsection of 6-DOF nonlinear dynamic modeling based on the first principle, and the linearized dynamic model is elaborated at the end of this section. In Section III, two CCDs are presented. The comparison of the two CCDs under the same trajectory tracking task is presented in Section IV. The concluding remarks are given in Section V.

**Remarks on the notation.** Matrices are denoted by uppercase letters, and vectors and scalars are denoted by lower-case letters.

## II. Dynamic Modeling of Quadrotor x-frame

Modeling is essential for engineering, as it is a process of writing mathematical equations based on the laws of physic to describe a physical system. It often includes considerable iterative refinement of the model in order to better fit the measurement data. The accuracy of the designed mathematical model is all about the process of modeling a physical system. This is stated very clearly in the following quote:

*"Make sure to write a correct physics, because the rest is all mathematics" by R. Kalman.*

Actually, because of the high cost of building and flight testing a real aircraft, the importance of the mathematical models goes far beyond design. The mathematical model is used, in conjunction with computer simulation, to evaluate the performance of the prototype aircraft and hence to improve the design [9].

### *II.1. Kinematics*

The relation between body coordinate system and NWU (North West Up) based on transnational and rotational motion is given by the following equations:

$$V_n = \mathrm{R}_{n/b} \cdot V_b, \tag{1}$$

$$\omega_n = \mathrm{S}^{-1} \cdot \omega_b. \tag{2}$$

The rotational matrix and the lumped transformation matrix are defined as follows:

$$\mathrm{R}_{n/b} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \tag{3}$$

$$\mathrm{S}^{-1} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}, \tag{4}$$

where $s_{(\cdot)}, c_{(\cdot)}, t_{(\cdot)}$ are short notations for trigonometric functions such as *sin*, *cos*, *tan*. The rotational matrix $R_{n/b}$ has a property of orthogonality, thus ${R_{n/b}}^{-1} = {R_{n/b}}^{T}$. Such a property is very useful for application in UAV systems, since it reduces the processing power, i.e. the numerical calculations for each iteration.

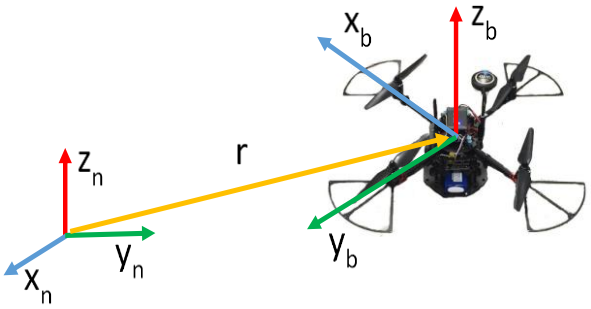



Fig.1. Relative coordinate systems

### *II.2. 6-DOF Dynamic model*

In general, there are two methods used for modeling: the Data-driven modeling ([12] and references therein) and the First modeling principle. In this paper, authors focus on the second method, by using Newton-Euler formalism to describe the translation and the rotational motion of Rigid body dynamics. For the derivation of the quadrotor model, the selection of first principle modeling over data-driven modeling has been made not because the latter method is less reliable or has some general drawbacks, but because the quadcopter is a very nonlinear aerial vehicle. For that reason, applying the system identification will increase the risk of crashing the flying vehicle while collecting data. This also directly increases the modeling costs for such multirotor vehicles.

In this analysis, authors consider that the UAV system is a point mass, and the analysis is made in a body coordinate frame. The following two dynamic equations consider the mass of the quadcopter, $m$, and its inertia matrix, $\mathbf{J}$:

$$m\dot{\mathbf{V}}_n = \mathbf{F}_n, \tag{5}$$

$$\mathbf{J}\dot{\omega}_b + \omega_b \times (\mathbf{J}\omega_b) = \mathbf{M_b}, \tag{6}$$

The moment of inertia matrix is defined as a diagonal matrix based on the assumption/fact that quadcopter body is symmetrical in x, y and z-axis, hence all crossed terms $(J_{ij} = 0 \,|\, i \neq j)$ are zero.

$$\mathbf{J} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \tag{7}$$

**Rotor Component -** According to [2], [8], [9], each motor creates its thrust $T_i$ and torque $Q_i$, i=1,2,..4. By considering aerodynamics, the thrust and the torque of the motor including the propeller are formulated as:

$$T_i = C_t \rho B R^2 \Omega_i^2, \tag{8}$$

$$Q_i = C_q \rho B R^3 \Omega_i^2, \tag{9}$$

Due to the fact that the propellers of a quadcopter have a fixed pitch angle, it can considered that all parameters from equations (11) and (12) are constants, except for $\Omega$:

$$T_i = C_T \Omega_i^2, \tag{10}$$

$$Q_i = C_Q \Omega_i^2, \tag{11}$$

where $C_T$ and $C_Q$ are constants that can be denoted as thrust and torque coefficient respectively. These two coefficients can be obtained experimentally.

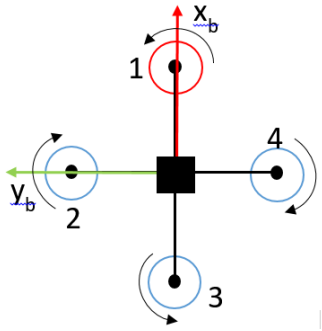



Fig. 2. Quadcopter 'Plus' frame

The total thrust generated by the four rotors that act on the quadcopter is:

$$f_{thz} = \sum_{n=1}^{i} T_i = C_T \cdot \left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right). \tag{12}$$

Based on Fig. 2, the moments generated by propellers for the 'plus'-frame configuration, are defined as:

$$\begin{aligned} m_x &= dC_T\left(\Omega_2^2 - \Omega_4^2\right) \\ m_y &= dC_T\left(-\Omega_1^2 + \Omega_3^2\right) \\ m_z &= C_Q\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \end{aligned} \tag{13}$$

The force vector $F_r$ is obtained as a sum of all rotor thrusts which result in a total lift of the quadcopter body in the positive direction of the z-axis, such as:

$$\mathbf{F_r} = \begin{bmatrix} 0 \\ 0 \\ f_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ C_T & C_T & C_T & C_T \end{bmatrix} \cdot \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}. \quad (14)$$

The moment vector $M_r$ is obtained as the difference between thrusts of rotors (roll and pitch angle), and summation of total torque of rotors (yaw angle):

$$\mathbf{M_r} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} 0 & dC_T & 0 & -dC_T \\ -dC_T & 0 & dC_T & 0 \\ -C_Q & C_Q & -C_Q & C_Q \end{bmatrix} \cdot \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}. \quad (15)$$

The gravitational and rotor component is compounded as follows:

$$\begin{bmatrix} \mathbf{F_n} \\ \mathbf{M_b} \end{bmatrix} = \mathbf{\Gamma}_g + \mathbf{\Gamma}_{rot} = \begin{bmatrix} \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^T \\ \mathbf{0}_{3x1} \end{bmatrix} + \begin{bmatrix} \mathbf{R_{n/b}} \cdot \mathbf{F_r} \\ \mathbf{M_r} \end{bmatrix}. \quad (16)$$

After recalling equations of rigid body dynamics (5), (6), and substituting (16), after simplification, the result is:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} \begin{bmatrix} c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi \\ c_\phi c_\theta \end{bmatrix} f_z$$
$$\begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} qr(J_{zz} - J_{yy}) \\ pr(J_{zz} - J_{xx}) \\ pq(J_{zz} - J_{yy}) \end{bmatrix} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}. \quad (17)$$

Equation (17) represents the 6-DOF nonlinear dynamic model of a quadcopter with 'plus'-frame configuration, where $[f_z, \mathrm{m_x}, \mathrm{m_y}, \mathrm{m_z}]^T$ are inputs to the systems, and for simplicity, they can be denoted by $\mathbf{u} = [u_1, u_2, u_3, u_4]^T$.

### *II.3. Linearized dynamic model*

Since the dynamic model of the quadcopter is nonlinear, controlling it with a linear controller makes the system unreliable or controllable within a very slow speed, which is not the intention of this modeling process. However, due to the symmetric structure of quadcopter, it is possible to linearize the dynamic model of quadrotor at hovering mode.

Linearizing at:

$$\phi = 0, \theta = 0, \psi = \psi_0, \quad (18)$$

as a result:

$$\dot{\phi} = 0, \dot{\theta} = 0, \dot{\psi} = 0. \quad (19)$$

For small angles, values can be approximated:

$$sin\phi \approx \phi, sin\theta \approx \theta, cos\phi \approx 1, cos\theta \approx 1. \quad (20)$$

Rearranging equation (2), equation (21) is obtained:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & -s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (21)$$

Now, if equations (18), (19), (20) are substituted in (21), the result is:

$$p = \dot{\phi} - \theta\dot{\psi}, \quad q = \dot{\theta} - \phi\dot{\psi}, \quad r = \dot{\psi} - \phi\dot{\theta}. \quad (22)$$

The following approximation can also be done, namely $\theta\dot{\psi} \approx \phi\dot{\psi} \approx \phi\dot{\theta} \approx 0$, so that:

$$p = \dot{\phi}, \quad q = \dot{\theta}, \quad r = \dot{\psi}. \quad (23)$$

Considering equation (19), the result is:

$$p = 0, q = 0, r = 0. \quad (24)$$

In hovering mode, it is understandable that linear velocities $[u, v]^T$ in x and y coordinate with respect to the body frame have to be zero, hence:

$$u = \dot{u} = 0, \quad v = \dot{v} = 0. \quad (25)$$

Finally, the linearization of the 6-DOF dynamic model of the drone could be achieved by using equations (18), (19), (20), (24) and (25) in the nonlinear dynamic model, and taking $sin\psi \approx 0$ and $f_z \approx mg$. Thus, the linearized model is as follows:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = g \begin{bmatrix} \theta \\ -\phi \\ -1 \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} f_z$$
$$\begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}. \quad (26)$$

The overall linearized dynamic model (26) of the quadcopter could be represented as a single state space system:

$$\left.\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} \\ \mathbf{y} &= c(\mathbf{x}, \mathbf{u}) = \mathbf{C} \cdot \mathbf{x} \end{aligned}\right\}, \quad (27)$$

where **A**, **B**, **C** are constant matrixes and $\mathbf{x}$, $\mathbf{u}$ are state and control vectors explicitly defined as:

$$\mathbf{x} = [z, \dot{z}, \phi, \theta, \psi, p, q, r]^T,$$
$$\mathbf{u} = [\mathrm{u}_1, u_2, u_3, \mathrm{u}_4]^T = [\mathrm{f_{thz}}, m_x, m_y, m_z]^T,$$

$$\mathbf{A}=\begin{bmatrix}\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} & \mathbf{0}_{2x6}\\ \mathbf{0}_{3x5} & \mathbf{I}_{2x6}\\ \mathbf{0}_{3x5} & \mathbf{0}_{3x6}\end{bmatrix},\ \mathbf{B}=\begin{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} & \mathbf{0}_{2x3}\\ \mathbf{0}_{3x1} & \mathbf{0}_{3x3}\\ \mathbf{0}_{3x1} & \mathbf{\Lambda}\end{bmatrix},$$

$$\mathbf{\Lambda}=\begin{bmatrix}\frac{1}{J_{xx}} & 0 & 0\\ 0 & \frac{1}{J_{xx}} & 0\\ 0 & 0 & \frac{1}{J_{xx}}\end{bmatrix},\ \mathbf{C}=\mathbf{I}_{8x8}.$$

The state space system (27) is used in the simulation model for the design of linear controllers.

## III. Cascade Control Design

Designing a quadcopter control strategy is an important process in the development and building of UAV systems. Along with dynamic modeling and sensor accuracy, control design also has a big impact on the performance of a quadcopter in achieving the desired point in space, tracking a reference trajectory, or completing the desired mission.

The aim of the control system design, regardless of the internal and external uncertainties and disturbances, is stabilizing the system state around the desired reference state. Using feedback and comparing the measurable or estimated states of the dynamic model process with desired or reference ones is necessary. A quadcopter system contains 6-DOF $(x, y, z, \phi, \theta, \psi)$ and only four control inputs $(u_1, u_2, u_3, u_4)$. Therefore, it is an under-actuated and an unstable dynamically system. Since it is impossible to control 6 DOF simultaneously, it is not possible to control it with only one loop. The quadcopter is not capable of moving within the *(x-y)* plane and remaining in hovering state, i.e. *x* and *y* movements are performed by rotating the body frame with pitch angle $\theta$ and roll angle $\phi$ respectively. In other words, transnational motions are performed by rotational motions and not vice versa in a quadcopter. For the aforementioned reasons, an additional control loop, called outer control loop, needs to be designed. It will be tasked to govern the inner control loop to make the *x* and *y* translation motion.

The control system design is separated in two subsections: in the inner loop control design, and outer loop control design.

### *III.1. Cascade PID-PD control design*

#### *A. Cascade PID-PD control design*

##### *1) Outer loop control design*

The outer loop control is separated into two individual loops: altitude control and position control. The first one ensures achieving the desired altitude in the z-axis, while the latter ensures that UAV achieves the desired point in the (x-y) plane.

***a) Altitude control***

Considering the body coordinate frame in NWU (North(x), West(y), Up (z)), the z-axis is pointing upward. The gravity force is toward the negative z-axis, while the total lift force of the four motors is in the positive direction.

The altitude controller has to ensure that the UAV will achieve and maintain the desired height. The desired force of the quadcopter to achieve the referent point in the z-axis is denoted by the following equation:

$$F_d = m\cdot g + m\left(K_{1p}\cdot e_z + K_{1i}\cdot\int e_z + K_{1d}\cdot\dot{e}_z\right), \quad (28)$$

where: $K_{1d}$, $K_{1i}$ and $K_{1p}$ are gains of the PID controller, $e_z = (z_{ref} - z_{mes})$, is the displacement error along the z-axis. The $F_d$ represents the first control input $u_1$ to the dynamic model of the quadcopter.

For the large referent height application, the saturation of motors can be easily reached and the altitude control may become uncontrollable. For that reason, it is necessary to use the saturation PID controller:

$$e_z = sat((z_{ref} - z_{mes}), e_{z\min/\max}), \quad (29)$$

$$F_d = sat\left(mg + m\left(K_{1d}\dot{e}_z + K_{1p}e_z + K_{1i}\int e_z\right), F_{d\min/\max}\right), \quad (30)$$

where $e_{z\min/\max}$, $F_{d\min/\max}$, represent the constraints used for the saturation of the PID controller.

***b) Position control***

The design of position control relies on two traditional PID controllers. The position of a point in the x-y plane is denoted by $\mathbf{p} = [x, y]^T$. The distance error between the referent point and the current measured point is denoted by:

$$\mathbf{e}_p = p_{ref} - p_{mes} \quad (31)$$

The output of the position controller is given by the following equation:

$$\begin{bmatrix}a_x\\ a_y\end{bmatrix} = \left(\begin{bmatrix}K_{2px} & 0\\ 0 & K_{2py}\end{bmatrix}\mathbf{e}_p + \begin{bmatrix}K_{2ix} & 0\\ 0 & K_{2iy}\end{bmatrix}\int\mathbf{e}_p + \begin{bmatrix}K_{2dx} & 0\\ 0 & K_{2dy}\end{bmatrix}\dot{\mathbf{e}}_p\right)$$

.

The kinematic relation between the desired linear acceleration in x-y plane and the desired angular displacements is given as follows:

$$\mathbf{\Theta} = \frac{1}{g}\cdot\begin{bmatrix}sin(\psi_{ref}) & cos(\psi_{ref})\\ -cos(\psi_{ref}) & sin(\psi_{ref})\end{bmatrix}^{-1}\cdot\begin{bmatrix}a_x\\ a_y\end{bmatrix}, \quad (32)$$

where $\Theta$, represents an output vector of desired roll and pitch angle, i.e. $\Theta = [\phi_d, \theta_d]^T$.

For the large distance application of the quadcopter, the proposed position control will not work because it may reach the saturation of the motor. For instance, if the desired location is 1 km away, then distance error from equation (31) will be too large. Hence a saturation PID has to be used instead. The saturated PID has to be of the following form:

$$e_p = sat((p_{ref} - p_{mes}), \mathrm{e}_{\mathrm{p\,min/max}}),$$

$$\Theta = sat\left(\frac{1}{g}\cdot A^{-1}\left(\mathbf{K}_{2P}\mathbf{e}_p + \mathbf{K}_{2D}\dot{e}_p + \int \mathbf{K}_{2I}\mathbf{e}_p\right), \Theta_{\min/\max}\right),$$

where $\mathrm{e}_{\mathrm{p\,min/max}}$, $\Theta_{\min/\max}$ are the constraints used for the saturation of PID controller.

*2) Inner loop control design*

In other words, the inner loop control, is known in literature as attitude control. It assures hovering of the quadcopter body at $\phi, \theta \approx 0$. The attitude control is designed such that its aim is to ensure the control of the UAV orientations around the hovering posture for given desired orientation angles $\Theta_d = [\phi_d, \theta_d, \psi_d]^T$ and measured orientation angles $\Theta_m = [\phi_m, \theta_m, \psi_m]^T$ from sensors:

$$e_\Theta = \Theta_d - \Theta_{mes} \tag{33}$$

The attitude control loop is design based on three uncoupled traditional PID controllers. The outputs of this control loop are the desired moments in each direction x, y, z:

$$\tau_d = K_p \cdot e_\Theta + K_d \cdot \dot{e}_\Theta, \tag{34}$$

where: $\tau_d = [\tau_x, \tau_y, \tau_z]^T$ is the vector of desired moments, $K_p, K_d \in \mathbb{R}^{3x3}$ are gain matrices of three PDs regulators.

The PD regulators in equation (34) becomes saturated as follows:

$$e_\Theta = sat\left(\Theta_d - \Theta_{mes}, e_{\Theta\min/\max}\right), \tag{35}$$

$$\tau_d = sat\left(Kp \cdot e_\Theta + Kd \cdot \dot{e}_\Theta, \tau_{d\min/\max}\right). \tag{36}$$

*B. Cascade PID-MPC control design*

In comparison with the last subsection, altitude and position control remains the same, only the attitude control is redesigned using Model Predictive Control with constraints.

The following general form of a discrete plant should be considered:

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) \\ y(k) &= C_m x_m(k), \quad x(0) = x_0 \end{aligned}, \tag{37}$$

where: $u \in \mathbb{R}^p$ is the input control vector, $y \in \mathbb{R}^q$ is the output vector, $x \in \mathbb{R}^n$ is the state vector of dimension n, while $A_m, B_m, C_m$ are some constant matrices representing the plant. Here, the number of outputs is set to be smaller or equal to the number of inputs, alternatively, uncontrollable independent states will appear. The difference between the state and control variables can be denoted by:

$$\begin{aligned} \delta x_m(k+1) &= x_m(k+1) - x_m(k) \\ \delta u(k) &= u(k) - u(k-1) \end{aligned}. \tag{38}$$

The augmented state space model of the plant is obtained by combining (37), (38), thus:

$$\overbrace{\begin{bmatrix}\delta \mathbf{x}_m(k+1)\\ \mathbf{y}(k+1)\end{bmatrix}}^{\dot{\bar{\mathbf{x}}}(k+1)} = \overbrace{\begin{bmatrix}\mathbf{A}_m & \mathbf{0}_{nxq}\\ \mathbf{C}_m\mathbf{A}_m & I_{qxq}\end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix}\delta \mathbf{x}_m(k)\\ \mathbf{y}(k)\end{bmatrix}}^{\bar{\mathbf{x}}(k)} + \overbrace{\begin{bmatrix}\mathbf{B}_m\\ \mathbf{C}_m\mathbf{B}_m\end{bmatrix}}^{\mathbf{B}} \delta\mathbf{u}(k),$$

$$\mathbf{y}(k) = \overbrace{\begin{bmatrix}\mathbf{0}_{qxn} & I_{qxq}\end{bmatrix}}^{\mathbf{C}} \begin{bmatrix}\delta \mathbf{x}_m(k)\\ \mathbf{y}(k)\end{bmatrix} \tag{39}$$

Introducing the new state vector $\bar{\mathbf{x}}(\mathrm{k}) = [\delta \mathbf{x}_m(k) \quad \mathbf{y}(k)]^T$, where $\bar{\mathbf{x}}(\mathrm{k}) \in \mathbb{R}^{n_+}$ such that $n_+ = n + q$, and three real matrices (A, B, C), the augmented state space (39) is denoted as:

$$\begin{aligned} \dot{\bar{\mathbf{x}}}(k+1) &= \mathbf{A}\,\bar{\mathbf{x}}(k) + \mathbf{B}\,\delta\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\,\bar{\mathbf{x}}(\mathrm{k}) \end{aligned}. \tag{39}$$

For simplicity on the notation, it will continue to be written as $\mathbf{x}(k)$. The future state vector could be calculated from (40) for the discrete time k, by substituting the initial condition of the control vector $\delta u(k) = u_0$:

$$\begin{aligned} &x(k+1\,|\,k) = Ax(k) + B\delta u(k) \\ &x(k+2\,|\,k) = A^2x(k) + AB\delta u(k) + B\delta u(k+1) \\ &\vdots \\ &x(k+N_p\,|\,k) = A^{N_p}x(k) + A^{N_{p-1}}B\delta u(k) + ... \\ &+A^{N_p-N_c}B\delta u(k+N_c-1) \end{aligned}. \tag{40}$$

When equation (41) is substituted in (40), the predicted output is obtained:

$$\begin{aligned} &\mathrm{y}(k+1\,|\,k) = CAx(k) + CB\delta u(k) \\ &\mathrm{y}(k+2\,|\,k) = CA^2x(k) + CAB\delta u(k) + CB\delta u(k+1) \\ &\vdots \\ &\mathrm{y}(k+N_p\,|\,k) = CA^{N_p}x(k) + CA^{N_{p-1}}B\delta u(k) + ... \\ &+CA^{N_p-N_c}B\delta u(k+N_c-1) \end{aligned}, \tag{41}$$

Hence for the output vector $\mathbf{Y}$ and increment control vector , equation (43) we have:

$$\begin{aligned} \mathbf{Y} &= [y(k+1\,|\,k)^T, \quad y(k+2\,|\,k)^T, ..., y(k+N_p\,|\,k)^T]^T \\ \Delta\mathbf{U} &= [\delta u(k)^T, \quad \delta u(k+1)^T, ..., \delta u(k+N_c-1)^T]^T \end{aligned}. \tag{42}$$

The general form in matrix notation of (42) and (43) could be as follows:

$$\mathbf{Y} = \mathbf{H}\mathbf{x}(k) + \mathbf{\Theta}\Delta\mathbf{U}, \tag{43}$$

where:

$$\mathbf{H}=\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix};\ \mathbf{\Theta}=\begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & & & 0 \\ CA^{N_p-1}B & CA^{N_p-1}B & \cdots & CA^{N_p-N_c}B \end{bmatrix}.$$

For a given referent signal at sample time k, $\mathbf{r}(k)$, the objective of MPC is to bring the predicted output as close as possible to the referent signal using the best possible control vector $\Delta\mathbf{U}$. Assuming that the r(k) will remain constant within the optimization horizon:

$$\mathbf{R}_s=\bar{\mathbf{R}}_s r(k)=\overbrace{\begin{bmatrix}1\ 1\ \ldots\ 1\end{bmatrix}^T}^{N_p} r(k)\ .$$

The cost function J related to the minimization of control objectives is defined as follows:

$$\mathbf{J}=(\mathbf{R}_s-\mathbf{Y})^T(\mathbf{R}_s-\mathbf{Y})+\Delta\mathbf{U}^T\bar{\mathbf{G}}\Delta\mathbf{U}\,, \qquad (44)$$

where the diagonal matrix $\bar{G}$ is of the form $\bar{G}=\lambda I_{N_c x N_c}$ such that $\lambda$ is a positive tuning parameter used for the desired closed loop performance. In order to find an optimal control $\Delta U$ that minimizes J, one could substitute (45) in (44), then the necessary and sufficient condition for J to be minimal is that the partial derivative of J with respect to $\Delta U$ will equal zero. Consequently, rearranging the expression and finding the incremental optimal control vector $\Delta U$, goes as follows:

$$\Delta\mathbf{U}=(\mathbf{\Theta}^T\mathbf{\Theta}+\bar{\mathbf{G}})^{-1}(\mathbf{\Theta}^T\bar{\mathbf{R}}_s\mathbf{r}(k)-\mathbf{\Theta}^T\mathbf{H}\mathbf{x}(k))\,, \qquad (45)$$

where referent set-point vector signal $\mathrm{r}(k)=[r_1(\mathrm{k}),r_2(\mathrm{k}),r_3(\mathrm{k})]^T$, for this case the referent input coming from the position control is $[\phi_d(k),\theta_d(k),\psi_d(k)]^T$. By applying receding horizon control, the first p elements are taken from $\Delta U$ to form the incremental optimal control:

$$\delta\mathbf{u}=\overbrace{\begin{bmatrix}I_p & o_p\ldots o_p\end{bmatrix}}^{N_c}(\mathbf{\Theta}^T\mathbf{\Theta}+\bar{\mathbf{G}})^{-1}(\mathbf{\Theta}^T\bar{\mathbf{R}}_s\mathbf{r}(k)-\mathbf{\Theta}^T\mathbf{H}\mathbf{x}(k)) \qquad (46)$$

The control algorithm works as follows: based on the initial conditions, the future state and output vectors are calculated accordingly with the prediction horizon $N_p$. Then matrices $H, \Theta,$ are calculated, which it is useful in order to obtain the incremental optimal control vector $\Delta U$ for control horizon $N_c$. Thus, the first $p$ elements are used from $\Delta U,$ and are fed into the plant, and the procedure repeats over and over. The idea is to ensure that the predicted output vector $\mathbf{Y}$ is as close as possible to the referent input vector.

## IV. Simulation Results

### A. Trajectory tracking

In order to evaluate the presented cascade control design structures, a Simulink MATLAB based simulation environment is created (Fig. 3).

In the general simulation scheme, a nonlinear dynamic model of the quadrotor is taken to evaluate the performance of the designed control structures under the task of tracking a referent trajectory. The outputs from the control blocks are fed to the command conversion which in the real flight controller application is obtained as an approximation to the inverse of the propeller and motor dynamics. It is obtained by choosing the configuration through the selection of a frame of the UAV system and defining its parameters, like mass, arm length, propeller dimensions, motor type, ESC type etc. The ESC+Motor dynamics block is formulated based on the following equation:

$$\Omega(s)=\frac{1}{1+0.05s}(8.427\sigma+0.602)\cdot 10^3\,, \qquad (47)$$

where the relation between $\sigma$ and $\Omega$ is obtained experimentally by measuring ESC response and motor dynamics response, the latter one is identified and approximated to behave as a first-order system with the time constant $\tau_m=0.05$. The parameter $\sigma$ in metaphoric explanation is the joystick command, and its value varies between 0 and 1. The outer inverse dynamic block consists the inversion of equation (32), but instead of $\psi_{ref}$, $\psi_{mes}$ is used. Additionally, the double integrator is necessary in order to obtain $x_{mes},\ y_{mes}$.

Parameters used in the simulation are presented in the following table. The first three parameters are directly measurable, while the rest are experimentally identified.

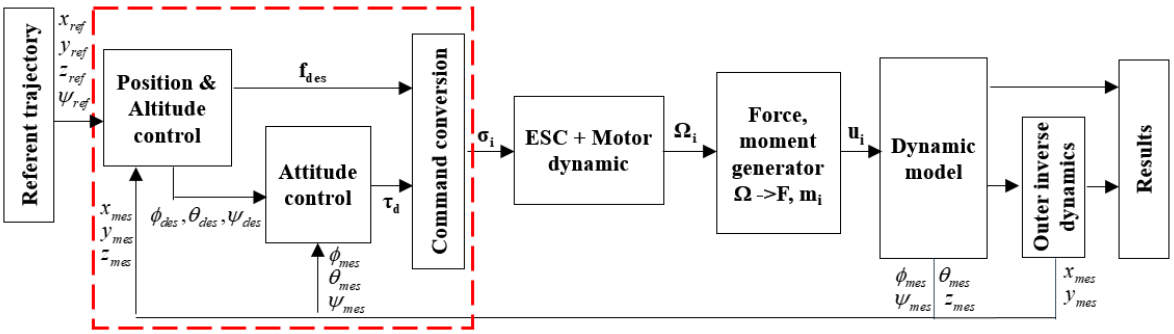



Fig. 3. The General simulation scheme of the quadcopter,
(the red rectangle block represents the flight controller, i.e. pixhawk from the real application excluding sensors)

TABLE I
PARAMETERS USED IN THE SIMULATION

| Parameter | Value | Unit |
|---|---|---|
| $m$ | 1.2 | $kg$ |
| $g$ | 9.81 | $m/s^2$ |
| $d$ | 0.185 | $m$ |
| $I_{xx}$ | $8.623 \cdot 10^{-3}$ | $kg \cdot m^2$ |
| $I_{yy}$ | $8.631 \cdot 10^{-3}$ | $kg \cdot m^2$ |
| $I_{zz}$ | $1.615 \cdot 10^{-3}$ | $kg \cdot m^2$ |
| $C_T$ | $9.25 \cdot 10^{-6}$ | $Ns^2/rad^2$ |
| $C_Q$ | $1.217 \cdot 10^{-3}$ | $Ns^2/rad^2$ |

The referent trajectory is taken such that the first part is almost identical to the takeoff in the z-direction, the second part consists of flying on the hovering region (z=const), and the third part is the landing part.

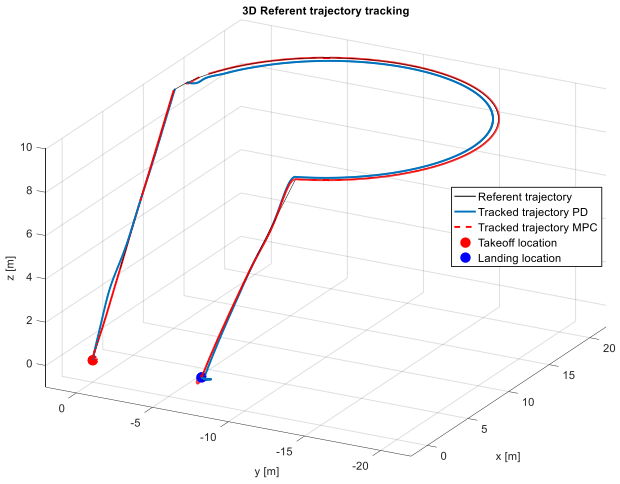
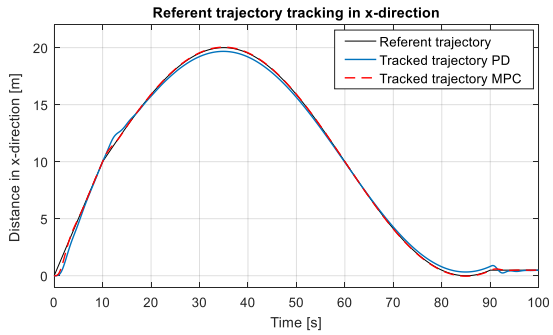



Fig. 4. Tracking a 3D referent trajectory with PD & MPC.



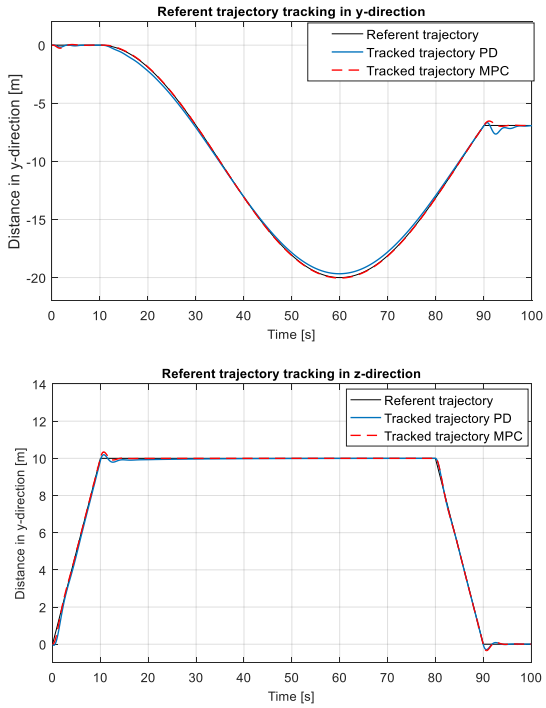




Fig. 5. Referent trajectory in x, y, z directions with PD & MPC.

The MPC control has been designed based on prediction horizon $N_P = 10$, and control horizon $N_C = 2$.

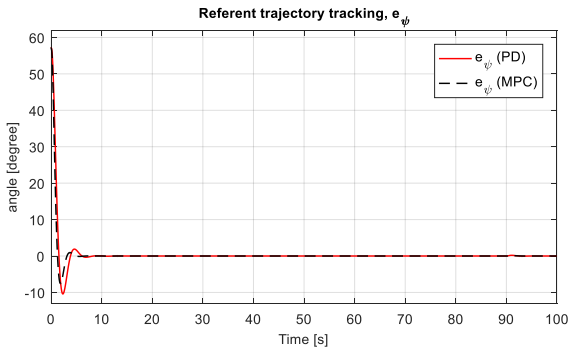



Fig. 6. Yaw angle error convergence $e_\psi$ with PD & MPC.

It is observable that the results of the tracking are very close to the referent trajectories. Linear PID-MPC control structure provides better results than the PID-PD control structure.

From Fig. 4 and Fig. 5 it is notable that trajectory tracking with PID-PD control structure contains a considerable steady-state error, and along the trajectory, there are some overshoots, which are not present with PID-MPC. According to Fig. 6, the control of the yaw angle orientation in both cases is almost the same from the aspect of the settling time, but orientation tracking with MPC is slightly better. As an overall conclusion from the simulation, it can be said that the cascade control system design has achieved its goal to satisfy the required trajectory tracking as much as possible, while maintaining the smoothness during its flight.

### B. Helical trajectory tracking

The comparison of two cascade control strategies has consequently been conducted under the helical trajectory tracking, in presence of periodic external torque disturbances applied in x and y-axis. The periodic disturbance applied to the torque moment in x has the magnitude $37 \cdot 10^{-3}$ Nm, period 10s, duty cycle 5% and the phase shifted 8s. On the other hand, the periodic disturbance applied to the torque moment in y has the magnitude $28 \cdot 10^{-3}$ Nm, period 6s, duty cycle 10% and the phase shifted 5s.

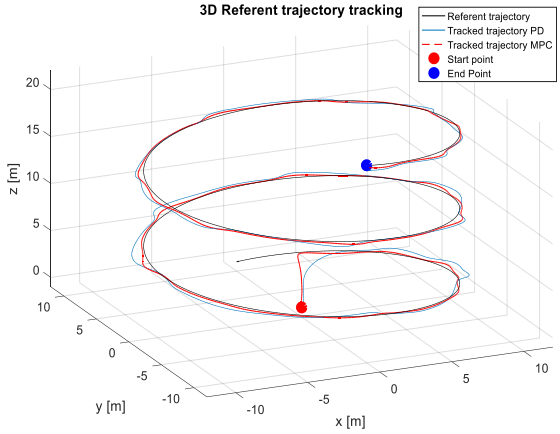



Fig. 7. Tracking a 3D referent trajectory with PD & MPC in presence of external periodic torque disturbances.

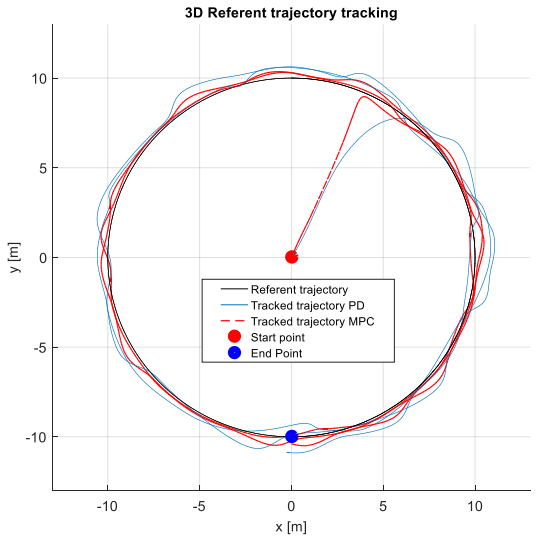



Fig. 8. Tracking a 3D referent trajectory with PD & MPC in presence of external periodic torque disturbances (Top view x-y plane)

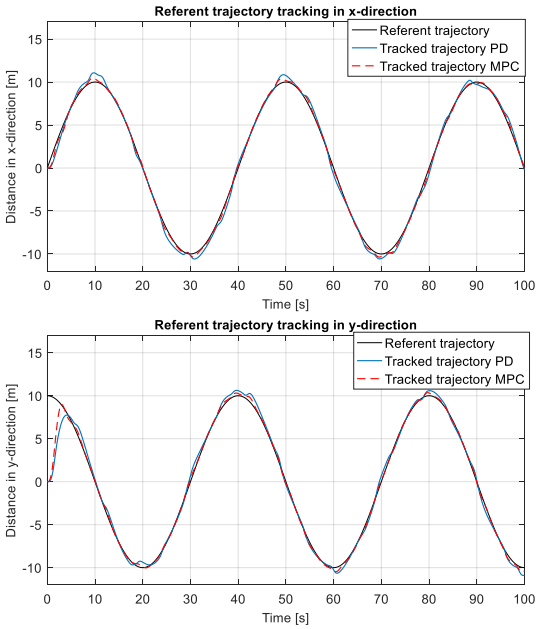



Fig. 9. Referent trajectory tracking in x, y directions with PD & MPC in presence of external periodic torque disturbances

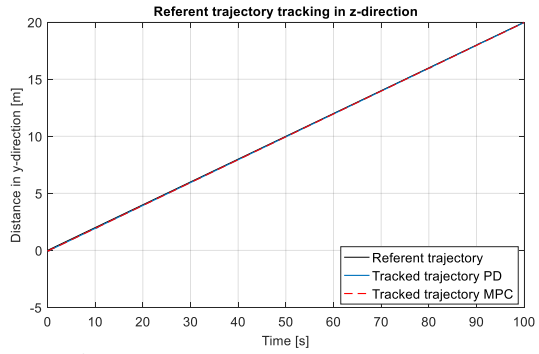



Fig. 10. Referent trajectory tracking in z directions with PD & MPC

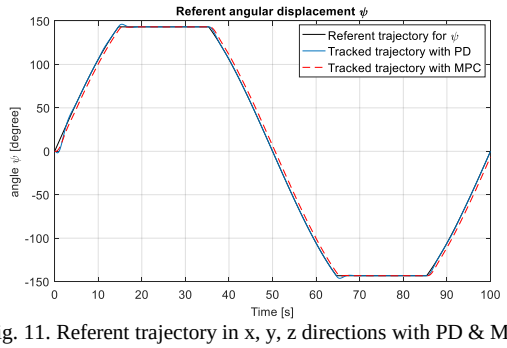



Fig. 11. Referent trajectory in x, y, z directions with PD & MPC.

From the simulation results elaborated in Fig. 7 and Fig. 8 on tracking 3D helix referent trajectory, is obvious the comparison between tracked trajectories with PD regulator and with MPC on the attitude control. Even though both controllers do not loose tracking, the latter one is less sensitive from external disturbances and it provides better tracking. Consequently, the trajectory tracking in three directions x, y, z, is explicitly elaborated in Fig 9-10. According to Fig, 11, the yaw referent

orientation angle $\psi$, is given in a desired orientation trajectory form, in comparison with Fig. 6 which is given in a referent desired angle.

## V. Conclusion

Considering the fact that quadrotor dynamics is very nonlinear and hard to control, the approach on linearizing the model around the hovering point and obtaining a linear dynamic model is presented. In addition, the flight controller is designed, and together with a dynamic model of quadrotor the generalized UAV system scheme is developed and simulated in MATLAB Simulink. The control system design relies on three sub-control designs (altitude control, position control and attitude control), which in total consists of 6 PID/PD controllers or 3 PID plus one linear MIMO MPC. It is proven that the designed flight controller is able to stabilize the quadrotor and perform various trajectory tracking (cases when $\psi \neq 0$) with a minimum error when the nonlinear model of the quadrotor is used. In case of tracking referent trajectories with periodic external torque disturbances, simulation results show that both strategies of cascade control do not loose tracking, but the one with MPC is less sensitive and provides better tracking.

Future works will be focused on implementing the designed control system on a real quadrotor with plus configuration, and observe the real performance with sensors on tracking desired trajectories.

## Acknowledgements

The first author would like to thank Prof. Dr. Ben M. Chen and his UAV Team for the valuable discussions and significant practical contributions during his two months academic visit (July-Sep. 2017) at ECE Department, National University of Singapore. Also, would like to thank the National University of Singapore and the University of Mitrovica “Isa Boletini” for their financial support for this academic visit.

## References

[1] S. Bouabdallah, A. Noth, and R. Siegwart, Pid vs LQ control techniques applied to an indoor micro quadrotor, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2451–2456 (2004).

[2] G. Cai, B. M. Chen, and T. H. Lee, *Unmanned rotorcraft systems* (Springer, 2011).

[3] R. Goel, S. M. Shah, N. K. Gupta, and N. Ananthkrishnan, Modeling, simulation and flight testing of an autonomous quadrotor, *Proceedings of ICEAE*, pp. 1–7 (2009).

[4] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, Dynamic modeling and configuration stabilization for an x4-flyer, *IFAC Proceedings Volumes*, vol. 35 (1) (2002), 217–222.

[5] Y. Bi, M. Lan, J. Li, K. Zhang, H. Qin, Sh. Lai, and B. M. Chen. Robust autonomous flight and mission management for MAVs in GPS-denied environments, *IEEE Asian Control Conference (ASCC)*, pp. 67-72 (2017).

[6] S. K. Phang, S. Lai, F. Wang, M. Lan, and B. M. Chen, Uav calligraphy, *IEEE International Conference Control & Automation (ICCA)*, pp. 422–428 (2014).

[7] S. K. Phang, K. Li, K. H. Yu, B. M. Chen, and T. H. Lee, Systematic design and implementation of a micro unmanned quadrotor system, *Unmanned Systems*, vol. 2 (2) (2014), 121–141.

[8] Q. Quan, *Introduction to multicopter design and control* (Springer, 2017).

[9] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems* (John Wiley & Sons, 2015).

[10] X. Zhang, B. Xian, B. Zhao, Y. Zhang, Autonomous Flight Control of a Nano Quadrotor Helicopter in a GPS-Denied Environment Using On-Board Vision. *IEEE Transactions on Industrial Electronics*. vol. 62 (2015), 6392–6403.

[11] D. Bunjaku, J. Stefanovski, and M. Stankovski, Dynamic modeling and asymptotic point stabilization control of two differential wheeled mobile robot, *Journal of Electrical Engineering and Information Technologies - JEEIT*, vol. 1(1-2) (2017), 25–35,

[12] D. Bunjaku, and M. Stankovski, The system identification in industrial control: Case study on the differential wheeled mobile robot, *IEEE International Conference Control & Automation (ICCA)*, pp. 94-99 (2017).

[13] K. Wang, Y. Ke, and B. M. Chen, Development of autonomous hybrid UAV U-Lion with VTOL and cruise flying capabilities, *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. *1053-1060* (2016).

[14] T. N. Dief, M. M. Kamra, and S Yoshida, Modeling, System Identification, and PID-A Controller for Tethered Unmanned Quad-Rotor Helicopter, *International Review of Aerospace Engineering*, 10(4) (2017), 215-223.

[15] S. Deskovski, V. Sazdovski, and Z. Gacovski, Guidance Laws and Navigation Systems for Quadrotor UAV: Theoretical and Practical Findings, In *Complex Systems*, (Springer, Cham, 2016, pp. 395-407).

[16] J. C. V. Junior, J. C. De Paula, G. V. Leandro, and M. C. Bonfim, Stability control of a quad-rotor using a PID controller, *Journal of Applied Instrumentation and Control* 1(1) (2013), 15-20.

[17] O. Araar, and N. Aouf, Full linear control of a quadrotor UAV, LQ vs H∞, *UKACC International Conference on Control*, pp. 133-138 (2014).

[18] A. Sorensen, *Autonomous control of a miniature quadrotor following fast trajectories, Master thesis, Aalborg University*, 2010.

[19] G. V. Raffo, M. G. Ortega, and F. R. Rubio, MPC with Nonlinear $\mathscr{H}\infty$ Control for Path Tracking of a Quad-Rotor Helicopter, *IFAC Proceedings Volumes* 41(2) (2008), 8564-8569.

[20] G. Ganga and M. M. Dharmana, MPC controller for trajectory tracking control of quadcopter, *Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1-6 (2017).

[21] T. N. Deif, A. H. Kassem, G. M. El Baioumi, Modeling, robustness, and attitude stabilization of indoor quad rotor using fuzzy logic control, *International Review of Aerospace Engineering*, **7** (2014), 197–201.

[22] T. N. Deif, A. H. Kassem, G. M. El Baioumi, Modeling and Attitude Stabilization of Indoor Quad rotor, *International Review of Aerospace Engineering*, **7** (2014), 43-47.

## Authors’ information

[1]Department of Informatics Engineering, FMCE at University of Mitrovica, Republic of Kosovo (email drilon.bunjaku@uni-pr.edu)

[2]Department of Automation & System Engineering, FEIT at Ss. Cyril & Methodius University, Skopje, Republic of Macedonia (e-mail: gorjan & milestk@feit.ukim.edu.mk)

[3]Control & Informatics Div., JP ”Strezevo”, Bitola, Republic of Macedonia (e-mail: jovanstef@t.mk).

**Drilon Bunjaku** received his B.Sc. degree in Telecommunication and M.Sc. degree in Mechatronics, both from the University of Prishtina, Faculty of Electrical and Computer Engineering, in 2009 and 2012 respectively. Currently is pursuing his Ph.D. degree in Automation and System Engineering at the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University in Skopje. Since 2013, he holds the position of Teaching and Research Assistant at the Faculty of Mechanical and Computer Engineering, University of Mitrovica. His research interests are in modeling systems and control, unmanned aerial systems, robust control theory and applications. He is a student member of IEEE since 2013.

**Gorjan Nadzinski** received the Ph.D. degree in 2018 from the Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Macedonia, where he is currently working as an assistant professor.
His research interests include industrial control, networked control systems, robust control, noise robustness, secure communications, and decentralized state estimation.

**Mile Stankovski** received the Ph.D. degree from Ss Cyril and Methodius University in Skopje, Macedonia, in 1997. He was with the University of Woolverhampton, U.K. He is currently a Professor of automation, system engineering and robotics and the Head of the Institute of Automation and System Engineering with the Faculty of Electrical Engineering and Information Technology, UKIM, Macedonia. He has almost 20 years of industrial experience and has been employed in a number of applicative and scientific projects. He has extensive experience in industrial process control, networked control, robotics, and also in water and waste water system control.

**Jovan D. Stefanovski** received the B.E.E. degree from the University SS Cyril & Methodius, Skopje, Macedonia, in 1981, the M.S.E.E. degree from the University of Belgrade, Serbia, in 1984, and the Ph.D. degree from the University SS Cyril & Methodius, in 1996, all in electrical engineering. He works on theory and engineering applications of control systems. He has published 52 journal papers, all contributing to control systems. Since 2014, he has been an Associate Professor at the Inst. Autom. & Syst. Eng., University SS Cyril & Methodius.