Nonlinear Quadrotor Control Based on Simulink Support Package for Parrot Minidrones ⋆ ⋆⋆

Alexey E. Golubev[1][0000−0002−1511−4583] and
Timur V. Glazkov[2][0000−0002−2409−2910]

[1] Bauman Moscow State Technical University, Moscow, Russia
v-algolu@hotmail.com
[2] Ishlinsky Institute for Problems in Mechanics RAS, Moscow, Russia
t.glazkov@bk.ru

Abstract. This paper deals with nonlinear control design for Parrot Mambo or Parrot Rolling Spider quadcopters using the Simulink Support Package for Parrot Minidrones. A full rigid body model of the flying vehicle that doesn't assume smallness of the Euler angles is considered. For synthesis of the control the nonlinear dynamics inversion and integrator backstepping approaches are used. Block diagrams illustrate how the control laws are applied to Parrot Minidrone flight control and can be used in nonlinear control education. Exercises to design nonlinear Parrot Minidrone control algorithms as Simulink Subsystem blocks are suggested.

Keywords: Nonlinear control · Control education · Quadrotor UAVs · Integrator backstepping.

1 Introduction

During the last two decades control of quadrotors became extremely popular among control theorists and practitioners. There are many reasons for such popularity. In spite of the fact that a quadrotor is inherently an underactuated mechanical system, it demonstrates nice controllability properties. This turned quadrotors into a test application for many control theories. Moreover, in contrast to wide-spread in control theory academical or educational mechanical examples, see e.g. [4], quadcopters can be considered as real industrial systems which are employed in many civil and military tasks.

Such theoretical and practical appeal resulted in a bulk of papers, see e.g. [2, 3, 5–7, 9–12, 15–17]. To solve position reference trajectory tracking control problems different approaches can be found in the literature, e.g. the PID and LQR control (see [7, 10]), integrator backstepping based designs ([3, 11]) or neural

⋆ This work is supported by the Russian Foundation of Basic Research (projects 19-07-00817 and 20-07-00279)

networks (see e.g. [2]). Unknown model parameters, in particular, uknown quadcopter mass and moments of inertia, were accounted for in [1] and [13]. Still, it remains a challenge for a quadcopter control system to account for the influence of external uncontrolled disturbances, e.g. wind, and to satisfy state and control constraints during quadrotor motion, see e.g. [14–16].

The main feature of the current paper is that the suggested nonlinear control algorithms and the corresponding block diagrams are presented for the purpose of implementation on the Parrot Minidrones (Parrot Mambo or Parrot Rolling Spider) using the Simulink Support Package for Parrot Minidrones (SSPPM) [18]. This package is included in the Matlab environment and is being actively developed. SSPPM allows to create Parrot Minidrones flight control algorithms using Simulink blocks and deploy control algorithms directly on the drone via a Bluetooth wireless network.

Parrot Minidrones together with the SSPPM can be considered as a nice and affordable control laboratory equipment due to a very low price of the minidrones and availability of the Matlab/Simulink environment at technical universities. The Parrot Mambo and Parrot Rolling Spider Minidrones are equipped with an ultrasonic sensor, accelerometer, gyroscope, pressure sensor and a downward facing camera, from which one can restore acceleration, angular velocity, altitude and displacement in the horizontal plane.

The appeal of using Parrot Minidrones together with the SSPPM for control education purposes is also underpinned by the fact that the SSPPM is a thorough modelling environment ready for implementation on the hardware. It contains all necessary default control system components, such as a quadrotor nonlinear mathematical model with the identified physical parameters, the state estimator subsystem that recovers state of the model from the measured data, a tuned PID controller to realize some basic angular and position reference motions. Moreover, researchers can replace any component of the system with their own one and test how their control or state estimation algorithm performs on a true-to-life quadrotor model or real flying device.

The paper is organized as follows. The quadcopter equations of motion are revised in Section 2. The synthesis of nonlinear control for tracking reference altitude and angular position trajectories is considered in Section 3. Section 4 presents design of nonlinear control for tracking reference position trajectories. Nonlinear adaptive control in case when the quadcopter mass and its moments of inertia are treated as unknown constants is discussed in Section 5. Section 6 gives numerical simulation and experimental results. Finally, the paper concludes with some remarks in Section 7.

2 Mathematical Model of Quadcopter Motion

Consider a quadcopter rigid body model, with translational and rotational dynamics described by the following systems, respectively, (see e.g. [5, 10]):

$$m\ddot{\xi} = F \begin{pmatrix} -\cos\gamma\cos\psi\sin\theta + \sin\gamma\sin\psi \\ -\cos\gamma\sin\psi\sin\theta - \cos\psi\sin\gamma \end{pmatrix}, \tag{1}$$

$$m\ddot{z} = -mg + F\cos\theta\cos\gamma \tag{2}$$

and

$$\begin{aligned}\dot{\eta} &= C\omega,\\ I\dot{\omega} &= M - \omega \times I\omega,\end{aligned} \tag{3}$$

where $\xi = (x, y)^{\mathrm{T}}$ and z are coordinates of the vehicle center of mass in the inertial frame; γ, θ, ψ are roll, pitch and yaw angles, respectively, $\eta = (\gamma, \theta, \psi)^{\mathrm{T}}$; m stands for the quadcopter mass, g is the acceleration due to gravity; F represents the thrust produced by the quadcopter rotors; $M = (M_x, M_y, M_z)^{\mathrm{T}}$ is the vector of torques; $\omega = (\omega_x, \omega_y, \omega_z)^{\mathrm{T}}$ is the vector of angular velocities in the body-fixed frame, $I = \operatorname{diag}(I_x, I_y, I_z)$ is the diagonal inertia matrix,

$$C = \begin{pmatrix} 1 & -\sin\gamma\operatorname{tg}\theta & -\cos\gamma\operatorname{tg}\theta \\ 0 & -\cos\gamma & \sin\gamma \\ 0 & \sin\gamma\sec\theta & \cos\gamma\sec\theta \end{pmatrix}.$$

Let us note that for a quadcopter the thrust F and the vector of torques M are functions of the four rotor angular velocities Ω_i and can be modeled by

$$\begin{pmatrix} F \\ M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ b/k & -b/k & b/k & -b/k \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix},$$

where $f_i = k\Omega_i^2$ is the thrust force of the i-th rotor; k and b are the rotors lift and drag aerodynamic coefficients, respectively; l is the distance between the quadrotor center of mass and the rotors.

3 Attitude and Altitude Control

In this section we consider the synthesis of nonlinear control for tracking reference altitude and angular position trajectories. Let the angular $\eta = \eta_0(t)$ and the altitude $z = z_0(t)$ reference signals be given as twice continuously differentiable functions of time. Suppose that absolute values of the roll γ and the pitch θ at any time does not reach the value of $\pi/2$ to avoid singularity of the C and G functional matrices defined in this paper.

Introduce the tracking error variables $e_z = z - z_0(t)$, $e_\eta = \eta - \eta_0(t)$ and rewrite the equations (2) and (3) in the variables e_z and e_η, respectively, as

$$\ddot{e}_z = -g + \frac{F}{m}\cos\theta\cos\gamma - \ddot{z}_0(t) \tag{4}$$

and

$$\ddot{e}_\eta = \dot{C}\omega + CI^{-1}M - CI^{-1}\omega \times I\omega - \ddot{\eta}_0(t). \tag{5}$$

The control problem is to find F and M such that

$$\lim_{t\to+\infty} e_z(t) = 0, \quad \lim_{t\to+\infty} e_\eta(t) = 0.$$

Choose the stabilizing control laws as below

$$F = \frac{m}{\cos\theta\cos\gamma}\left(g + \ddot{z}_0(t) - k_1\dot{e}_z - k_2 e_z\right), \tag{6}$$

$$M = \omega \times I\omega + IC^{-1}\left(\ddot{\eta}_0(t) - \dot{C}\omega - C_1\dot{e}_\eta - C_2 e_\eta\right), \tag{7}$$

where $k_1 > 0$, $k_2 > 0$ are positive gain coefficients and $C_1 > 0$, $C_2 > 0$ are positive definite gain matrices. Then, the equations (4) and (5) with the controls (6) and (7), respectively, are written as

$$\ddot{e}_z + k_1\dot{e}_z + k_2 e_z = 0, \quad \ddot{e}_\eta + C_1\dot{e}_\eta + C_2 e_\eta = 0, \tag{8}$$

with the equilibrium point $e_z = 0$, $e_\eta = 0$ being globally asymptotically stable.

One can take the gain coefficients $k_1 > 0$, $k_2 > 0$ and matrices $C_1 > 0$, $C_2 > 0$ to guarantee that $|e_z| \le \Delta_z$ if $t \ge t_z$ and $\|e_\eta\| \le \Delta_\eta$ if $t \ge t_\eta$. Here $\Delta_z = 0.05e_z(0)$, $\Delta_\eta = 0.05e_\eta(0)$; t_z and t_η are the desired transient times, respectively; $\|\cdot\|$ stands for the Euclidian norm. As the desired characteristic polynomials of equations (8) one takes

$$Q_z(\lambda) = \lambda^2 + 2\omega_z\lambda + \omega_z^2, \quad Q_\eta(\lambda) = \lambda^2 + 2\omega_\eta\lambda + \omega_\eta^2,$$

respectively, where $\omega_z = 4.8/t_z$, $\omega_\eta = 4.8/t_\eta$. This results in $k_1 = 2\omega_z$, $k_2 = \omega_z^2$ and $C_1 = 2\omega_\eta E$, $C_2 = \omega_\eta^2 E$. Here, E is the identity matrix of size 3×3.

Figure 1 illustrates how the control laws (6) and (7) are applied to Parrot Minidrone flight control with the help of Simulink Support Package for Parrot Minidrones. For educational purposes we propose the following exercise.

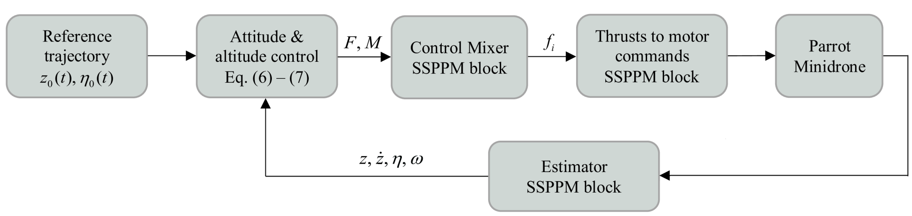

Fig. 1. Angular stabilization and altitude control block diagram.

Exercise 1. Design a nonlinear Parrot Minidrone control algorithm as a Simulink Subsystem block to track the following reference trajectories:

(a) $\eta_0(t) = [0, 0, 0]^T, \quad z_0(t) = 1;$

(b) $$\eta_0(t) = \begin{cases} [0, 0, 0]^T, \ t \le 5, \\ [0, 0, 0.2\pi t - \pi]^T, \ t \in (5, 15), \\ [0, 0, 2\pi]^T, \ t \ge 15; \end{cases}$$

$$z_0(t) = \begin{cases} \sum\limits_{i=0}^{5} a_i t^i, \ t \in [0, 5], \\ 1, \ t \ge 5. \end{cases}$$

In both cases the following initial conditions are suggested: $z(0) = \dot{z}(0) = 0$, $\eta(0) = \omega(0) = 0$.

Hint. In case (b) find the coefficients a_i to fulfill the initial and terminal conditions on $z(t)$.

4 Horizontal and Vertical Position Control

In the current section we deal with the design of nonlinear control for tracking reference position trajectories. Let the altitude $z = z_0(t)$ and the x, y position $\xi = \xi_0(t) = [x_0(t),\, y_0(t)]^T$ reference signals be four times continuously differentiable.

For the convenience sake, introduce the new control variables

$$(\tilde{M}_x, \tilde{M}_y, \tilde{M}_z)^T = \tilde{M} = \dot{C}\omega + CI^{-1}\left(M - \omega \times I\omega\right) \tag{9}$$

and rewrite the system (3) as

$$\ddot{\eta} = \tilde{M}.$$

Let a reference yaw trajectory $\psi_0(t)$ be given (see [5] for a discussion why to choose a reference yaw trajectory at this stage instead of a pitch or roll reference behavior). Define the tracking error variable $e_\psi = \psi - \psi_0(t)$. Then, the yaw tracking control $\tilde{M}_z$ is written as

$$\tilde{M}_z = \ddot{\psi}_0(t) - k_3\dot{e}_\psi - k_4 e_\psi, \tag{10}$$

where $k_3 > 0$, $k_4 > 0$ are some positive gain coefficients. Hence, the zero equilibrium of the closed-loop yaw error dynamics given by

$$\ddot{e}_\psi + k_3\dot{e}_\psi + k_4 e_\psi = 0$$

is globally asymptotically stable.

Next, the system (1) with control (6) takes the form

$$\begin{aligned}\ddot{\xi} = {} & (g + \ddot{z}_0(t) - k_1\dot{e}_z - k_2 e_z) \\ & \times \begin{pmatrix} -\cos\psi \operatorname{tg}\theta + \sin\psi \operatorname{tg}\gamma \sec\theta \\ -\sin\psi \operatorname{tg}\theta - \cos\psi \operatorname{tg}\gamma \sec\theta \end{pmatrix}.\end{aligned} \tag{11}$$

To find the x, y tracking control law rewrite the system (11) with control (10) as

$$\begin{aligned}\xi^{(IV)} = {} & \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} - (g + \ddot{z}_0 - k_1\dot{e}_z - k_2 e_z)\sec\theta \\ & \times \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} 0 & \sec\theta \\ \sec^2\gamma & \operatorname{tg}\gamma \operatorname{tg}\theta \end{pmatrix} \begin{pmatrix} \tilde{M}_x \\ \tilde{M}_y \end{pmatrix},\end{aligned} \tag{12}$$

where f_i are nonlinear scalar functions of the state.

Introduce the tracking error variable $e_\xi = \xi - \xi_0(t)$ and let G denote the matrix of coefficients of controls $\tilde{M}_x$ and $\tilde{M}_y$ in (12). Then, the x, y trajectory tracking control is written using nonlinear dynamics inversion as

$$\begin{pmatrix} \tilde{M}_x \\ \tilde{M}_y \end{pmatrix} = G^{-1} \Bigg[- \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} + \xi_0^{(4)}(t) \\ - C_1 e_\xi^{(3)} - C_2 \ddot{e}_\xi - C_3 \dot{e}_\xi - C_4 e_\xi \Bigg]. \tag{13}$$

Here the gain matrices $C_i > 0$ are such that the zero equilibrium of the resulting closed-loop system given by

$$e_\xi^{(4)} + C_1 e_\xi^{(3)} + C_2 \ddot{e}_\xi + C_3 \dot{e}_\xi + C_4 e_\xi = 0 \tag{14}$$

is globally asymptotically stable.

Additionally, to fulfill the condition $\|e_\xi\| \leq \Delta_\xi$ if $t \geq t_\xi$, where $\Delta_\xi = 0.05 e_\xi(0)$ and t_ξ is the required transient time, one takes

$$Q_\xi(\lambda) = \lambda^4 + 4\omega_\xi \lambda^3 + 6\omega_\xi^2 \lambda^2 + 4\omega_\xi^3 \lambda + \omega_\xi^4, \ \omega_\xi = 7.8/t_\xi$$

as the desired characteristic polynomial of each equation of the system (14). This choice yields $C_1 = 4\omega_\xi E$, $C_2 = 6\omega_\xi^2 E$, $C_3 = 4\omega_\xi^3 E$, $C_4 = \omega_\xi^4 E$, where E is the identity matrix of size 2×2.

Figure 2 describes how the control laws (6), (10) and (13) are applied to Parrot Minidrone flight control using Simulink Support Package for Parrot Minidrones. Finally, the following exercise is suggested.

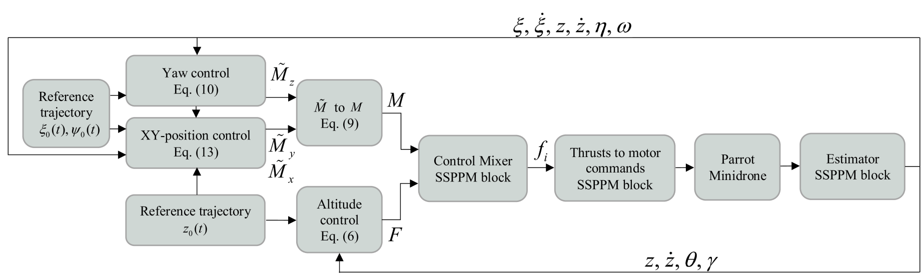

Fig. 2. Position control block diagram.

Exercise 2. Design a nonlinear Parrot Minidrone control algorithm as a Simulink Subsystem block to track the following reference trajectories:

(a) $\xi_0(t) = [0, 0]^T, \quad z_0(t) = 1, \quad \psi_0(t) = 0.$

(b) $\xi_0(t) = [\cos t, \sin t]^T, \quad z_0(t) = 1, \quad \psi_0(t) = t + \pi/2.$

The initial conditions are $\xi(0) = \dot{\xi}(0) = 0$, $z(0) = \dot{z}(0) = 0$, $\eta(0) = \omega(0) = 0$.

5 Adaptive Control

This section considers the synthesis of nonlinear adaptive control for tracking reference altitude and angular position trajectories. The quadcopter mass m and components I_x, I_y, I_z of the diagonal inertia matrix I are treated as unknown constants.

Let $J = [I_x, I_y, I_z]^T$ and $D(\nu)$, $\nu = [\nu_1, \nu_2, \nu_3]^T$ be the diagonal matrix with the elements $d_{ii} = \nu_i$, $i = 1, 2, 3$. Then, the following equalities hold

$$\omega \times I\omega \equiv SI\omega, \quad I\omega \equiv D(\omega)J,$$
$$S = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

and the equations (3) can be written as

$$\begin{aligned} \dot{\eta} &= C\omega, \\ I\dot{\omega} &= M - SD(\omega)J. \end{aligned} \tag{15}$$

The reference angular position trajectory tracking problem is to find the control M such that

$$\lim_{t \to +\infty} e_\eta(t) = 0,$$

where $e_\eta = \eta - \eta_0(t)$ is the tracking error, $\eta_0(t)$ is the reference angular position trajectory.

To find the stabilizing control M let us use the adaptive integrator backstepping technique, see [8]. To that end, consider first the function

$$V_1(e_\eta) = \frac{1}{2} e_\eta^T e_\eta > 0$$

and introduce the error $e_\omega = \omega - \chi_1$, where χ_1 is the desired reference behavior of the ω variable to be defined later. The time derivative of $V_1(e_\eta)$ along the trajectories of the system (15) is given by

$$\dot{V}_1(e_\eta) = e_\eta^T \dot{e}_\eta = e_\eta^T \left[Ce_\omega + C\chi_1 - \dot{\eta}_0(t) \right]. \tag{16}$$

The choice $\chi_1 = C^{-1}\dot{\eta}_0(t) - C^{-1}K_1 e_\eta$, where $K_1 > 0$ is some positive definite matrix, transforms (16) into the form

$$\dot{V}_1(e_\eta) = e_\eta^T C e_\omega - e_\eta^T K_1 e_\eta.$$

Further, introduce the estimation error $\tilde{J} = J - \hat{J}$, where $\hat{J}$ is an estimate for the unknown parameter vector J. Hence, since J is a constant vector, the following equality holds $\dot{\tilde{J}} = -\dot{\hat{J}}$. To find the tracking control M consider the function

$$V_2(e_\eta, e_\omega, \tilde{J}) = V_1(e_\eta) + \frac{1}{2} e_\omega^T I e_\omega + \frac{1}{2} \tilde{J}^T \Gamma_a^{-1} \tilde{J} > 0,$$

where $\Gamma_a > 0$ is a positive definite matrix. The time derivative of $V_2(e_\eta, e_\omega, \tilde{J})$ along the trajectories of the system (15) is written as

$$\begin{aligned}
&\dot{V}_2(e_\eta, e_\omega, \tilde{J}) = e_\eta^T \dot{e}_\eta + e_\omega^T I \dot{e}_\omega + \tilde{J}^T \Gamma_a^{-1} \dot{\tilde{J}} \\
&= e_\eta^T C e_\omega - e_\eta^T K_1 e_\eta + e_\omega^T (M - SD(\omega)J - I\dot{\chi}_1) - \tilde{J}^T \Gamma_a^{-1} \dot{\hat{J}} \\
&= -e_\eta^T K_1 e_\eta + e_\omega^T \left(C^T e_\eta + M - SD(\omega)\hat{J} - D(\dot{\chi}_1)\hat{J}\right) \\
&- \tilde{J}^T \left[SD(\omega) + D(\dot{\chi}_1)\right]^T e_\omega - \tilde{J}^T \Gamma_a^{-1} \dot{\hat{J}},
\end{aligned}$$

where $\dot{\chi}_1 = \dot{C}^{-1}\dot{\eta}_0(t) + C^{-1}\ddot{\eta}_0(t) - \dot{C}^{-1}K_1 e_\eta - C^{-1}K_1\dot{e}_\eta$. To eliminate the unknown parameter estimation error $\tilde{J}$ one takes

$$\dot{\hat{J}} = -\Gamma_a \left[SD(\omega) + D(\dot{\chi}_1)\right]^T e_\omega. \tag{17}$$

Finally, the choice

$$M = SD(\omega)\hat{J} + D(\dot{\chi}_1)\hat{J} - C^T e_\eta - K_2 e_\omega, \tag{18}$$

where $K_2 > 0$ is a positive definite matrix, results in

$$\dot{V}_2(e_\eta, e_\omega) = -e_\eta^T K_1 e_\eta - e_\omega^T K_2 e_\omega < 0.$$

Therefore, for the system (15) in closed-loop form with the control (18) and parameter update law (17) by the LaSalle-Yoshizawa theorem holds the following: $e_\eta(t) \to 0$ as $t \to +\infty$.

Next, consider the reference altitude trajectory tracking adaptive control problem.

Let $\hat{m}$ be an estimate of the unknown quadcopter mass m. Define also the error variable

$$\tilde{m} = m - \hat{m}. \tag{19}$$

The control problem is to find F in (4) which guarantees that

$$\lim_{t \to +\infty} e_z(t) = 0.$$

To find the stabilizing control F using the backstepping approach consider first the function

$$V_1(e_z) = \frac{1}{2} e_z^2 > 0$$

and introduce the error variable $\zeta = \dot{e}_z - \chi_2$, where χ_2 is the desired reference behavior of $\dot{e}_z$ to be given later. The time derivative of $V_1(e_z)$ is as follows

$$\dot{V}_1(e_z) = e_z \dot{e}_z = e_z (\zeta + \chi_2).$$

The choice $\chi_2 = -c_1 e_z$, where $c_1 > 0$ is some positive constant, gives

$$\dot{V}_1(e_z) = e_z \zeta - c_1 e_z^2.$$

Notice that the unknown coefficient $1/m$ of the thrust F in the system (4) can be written as

$$\frac{1}{m} = \frac{1}{\hat{m}} - \frac{\tilde{m}}{m\hat{m}}. \tag{20}$$

Additionally, since m is a constant, from (19) follows that $\dot{\tilde{m}} = -\dot{\hat{m}}$. Finally, to find the tracking control F consider the function

$$V_2\left(e_z,\ \zeta,\ \tilde{m}\right) = V_1(e_z) + \frac{1}{2}\zeta^2 + \frac{1}{2k_a m}\tilde{m}^2 > 0.$$

Its time derivative along the trajectories of the system (4) in view of (20) is written as follows

$$\begin{aligned}
\dot{V}_2\left(e_z, \zeta, \tilde{m}\right) &= e_z\dot{e}_z + \zeta\dot{\zeta} + \frac{1}{k_a m}\tilde{m}\dot{\tilde{m}} \\
&= e_z\zeta - c_1 e_z^2 + \zeta\Big(-g + \frac{F}{\hat{m}}\cos\theta\cos\gamma - \frac{F\tilde{m}}{m\hat{m}}\cos\theta\cos\gamma \\
&- \ddot{z}_0 + c_1\dot{e}_z\Big) - \frac{1}{k_a m}\tilde{m}\dot{\hat{m}} \\
&= -c_1 e_z^2 + \zeta\left(e_z - g + \frac{F}{\hat{m}}\cos\theta\cos\gamma - \ddot{z}_0 + c_1\dot{e}_z\right) \\
&- \left(\zeta\frac{F}{\hat{m}}\cos\theta\cos\gamma + \frac{1}{k_a}\dot{\hat{m}}\right)\frac{\tilde{m}}{m}.
\end{aligned}$$

To eliminate the unknown values of m and $\tilde{m}$ one chooses

$$\dot{\hat{m}} = -k_a\zeta\frac{F}{\hat{m}}\cos\theta\cos\gamma. \tag{21}$$

Finally, one takes

$$F = \frac{\hat{m}}{\cos\theta\cos\gamma}\left(g + \ddot{z}_0 - c_1\dot{e}_z - e_z - c_2\zeta\right), \tag{22}$$

where $c_2 > 0$ is a positive constant, to obtain

$$\dot{V}_2\left(e_z, \zeta\right) = -c_1 e_z^2 - c_2\zeta^2 < 0.$$

Hence, for the system (4) in closed-loop form with the control (22) and parameter update law (21) by the LaSalle-Yoshizawa theorem one has $e_z\left(t\right) \to 0$ as $t \to +\infty$.

Figure 3 describes how the control laws (22), (21) and (18), (17) are applied to Parrot Minidrone flight control using Simulink Support Package for Parrot Minidrones. Additionally, the following exercise is suggested.

Exercise 3. Design an adaptive nonlinear Parrot Minidrone control algorithm as a Simulink Subsystem block to track the reference trajectories given in Exercise 1.

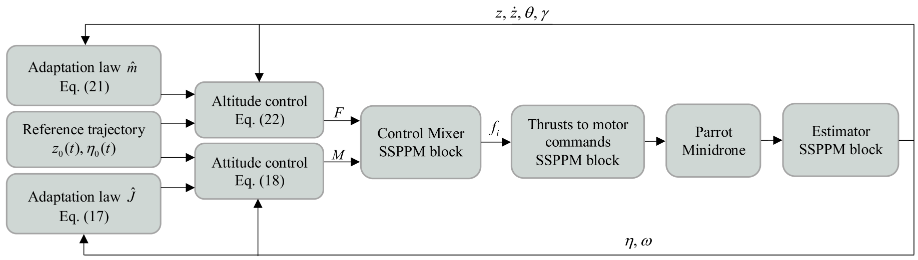

Fig. 3. Adaptive angular stabilization and altitude control block diagram.

6 Simulation and Experimental Results

Figures 4 − 10 show numerical simulation and experimental results for the Exercise 1. For numerical simulations we used the values of the quadrotor model parameters which were taken from the SSPPM files. Figures 4, 5 give attitude behaviour of the quadrotor under the control laws (6) and (7). The altitude performance for the considered hovering task is shown in figure 6. The quadrotor body-frame and rotor angular velocities are given in figures 7 − 10. Let us note that the simulation and experimental results show rather good performance of the designed station-keeping control laws (6) and (7) on the Parrot Mambo Minidrone.

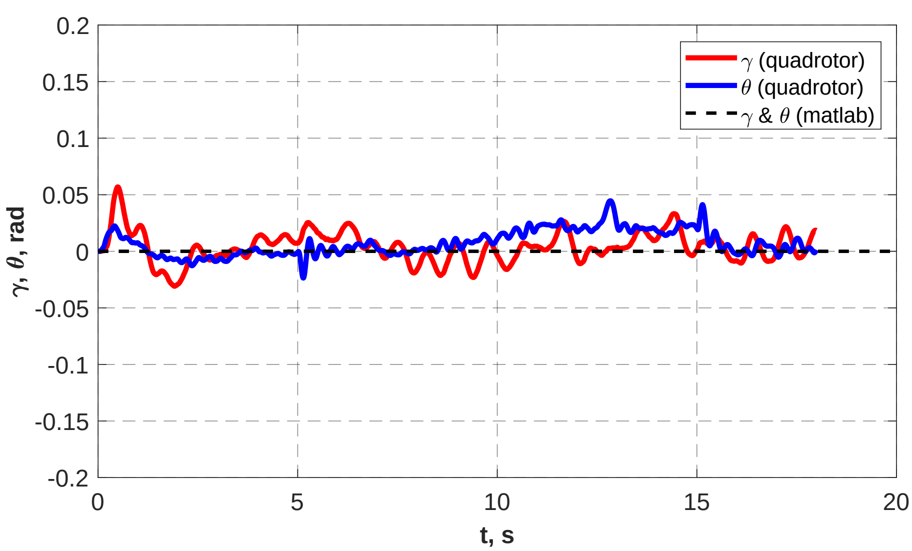

Fig. 4. Quadrotor pitch and roll angles (rad) versus time (s) (solid lines) and their simulated values (dashed line).

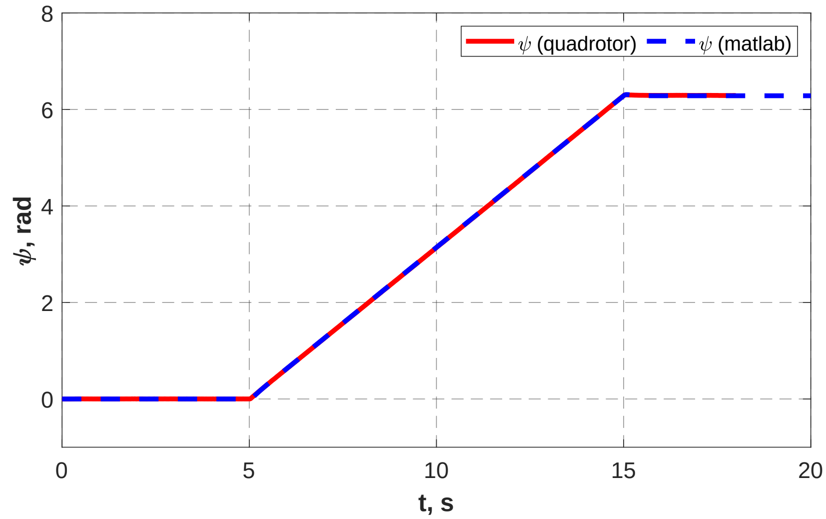

Fig. 5. Quadrotor yaw angle (rad) versus time (s) (solid line) and its simulated values (dashed line).

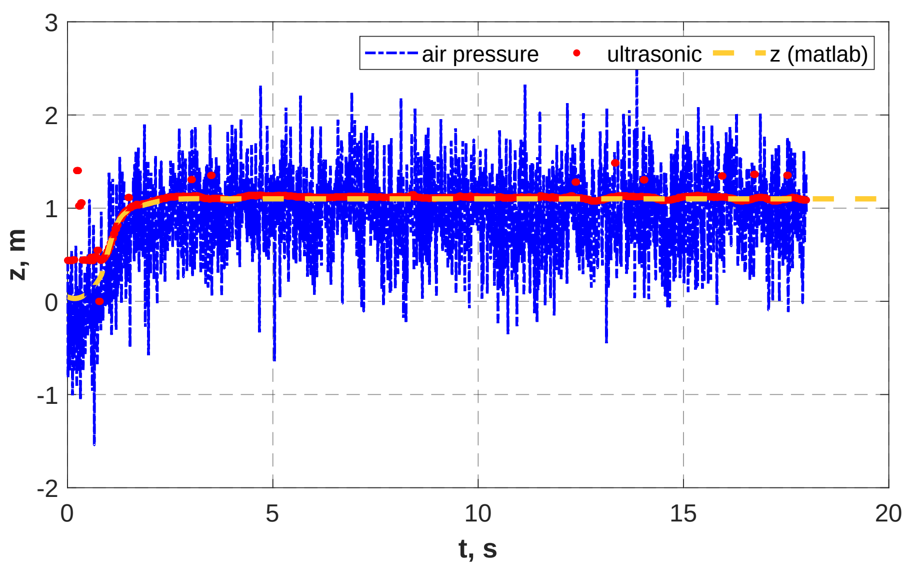

Fig. 6. Quadrotor altitude (m) versus time (s) (blue and red lines) and its simulated values (yellow line).

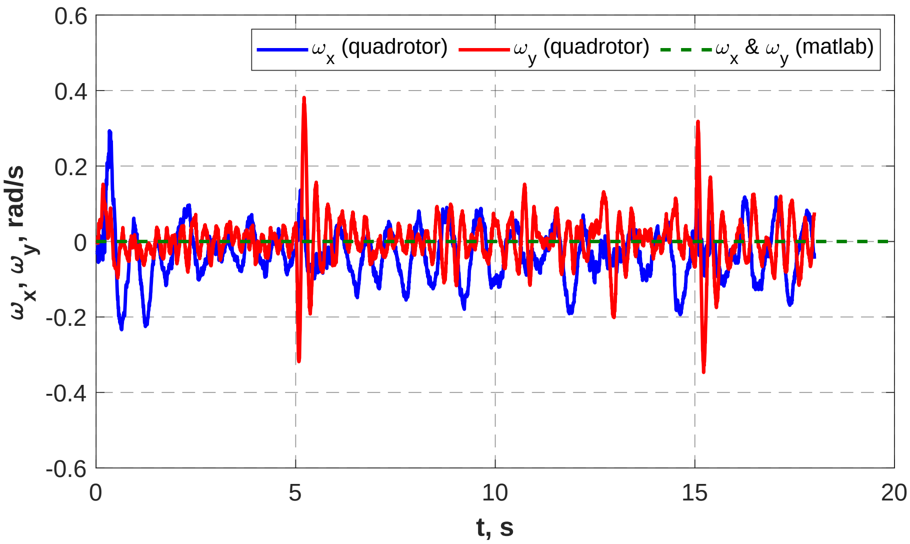

Fig. 7. Quadrotor angular velocities w_x, w_y (rad/s) versus time (s) (solid lines) and their simulated values (dashed line).

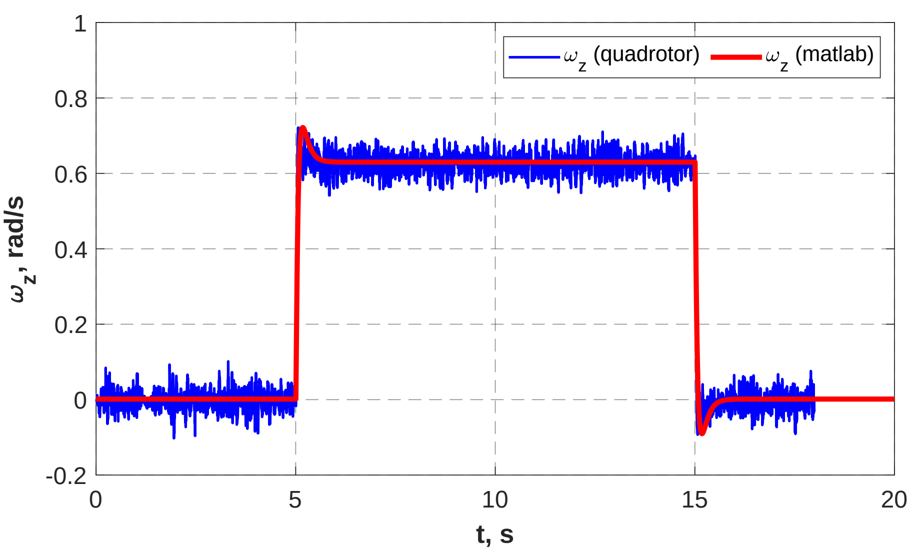

Fig. 8. Quadrotor angular velocity w_z (rad/s) versus time (s) (blue line) and its simulated values (red line).

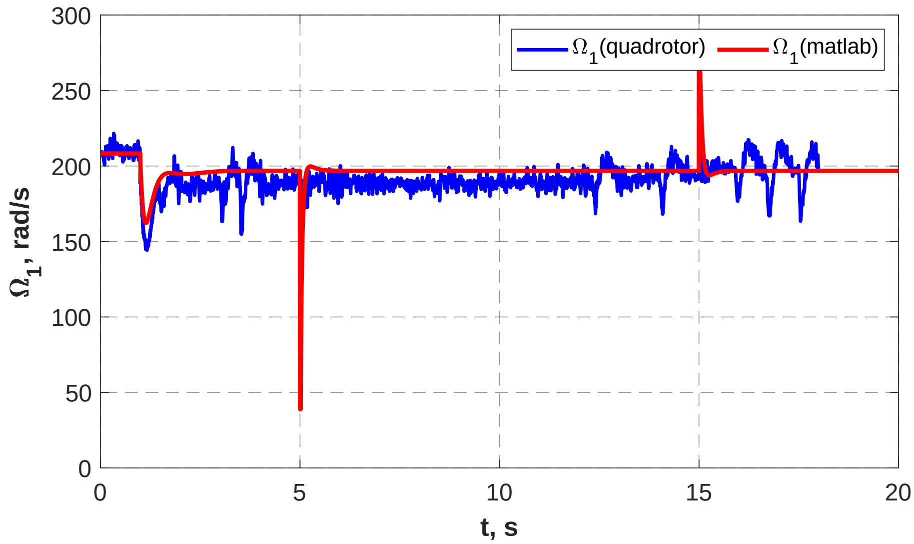

Fig. 9. Quadcopter rotor angular velocity Ω_1 (rad/s) versus time (s) (blue line) and its simulated values (red line).

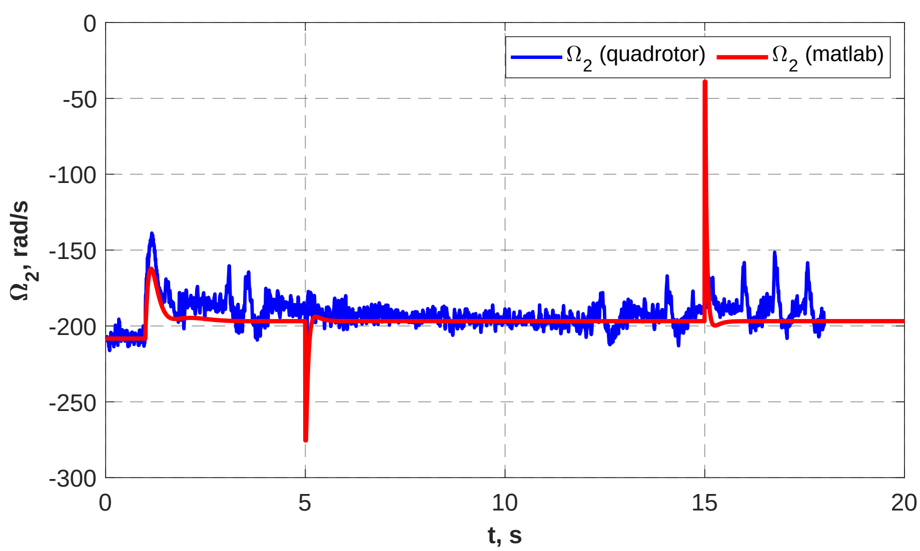

Fig. 10. Quadcopter rotor angular velocity Ω_2 (rad/s) versus time (s) (blue line) and its simulated values (red line).

7 Conclusion

This paper extended the functionality of the Simulink Support Package for Parrot Minidrones (SSPPM) by proposing nonlinear quadrotor control algorithms. We suggested using Parrot Minidrones together with the SSPPM as a nice and affordable control laboratory equipment for nonlinear control education. Block diagrams illustrated how the control laws could be applied to Parrot Minidrone flight control. Exercises to design nonlinear Parrot Minidrone control algorithms as Simulink Subsystem blocks were suggested. The experimental results show rather good performance of the designed station-keeping control laws on the Parrot Mambo Minidrone.

References

1. Chen, F., Zhang, K., Wang, Z., Tao, G., Jiang, B.: Trajectory tracking of a quadrotor with unknown parameters and its fault-tolerant control via sliding mode fault observer. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering **229**(4), 279–292 (2015)
2. Dierks, T., Jagannathan, S.: Output feedback control of a quadrotor UAV using neural networks. IEEE transactions on neural networks **21**(1), 50–66 (2010)
3. Falconi, G.P., Marvakov, V.A., Holzapfel, F.: Fault tolerant control for a hexarotor system using Incremental Backstepping. In: IEEE Conference on Control Applications, pp. 237–242 (2016)
4. Fantoni, I., Lozano, R.: Non-linear control for underactuated mechanical systems. Springer-Verlag (2002)
5. Glazkov, T.V., Golubev, A.E., Gorbunov, A.V., Krishchenko, A.P.: Control of quadcopter motion in the horizontal plane. AIP Conference Proceedings **2116**, 380003 (2019)
6. Golubev, A.E., Thway, N., Gorbunov, A.V., Krishchenko, A.P., Utkina N.V.: Construction of quadrocopter programmed motion in a flat labyrinth. AIP Conference Proceedings **2116**, 380004 (2019)
7. Khatoon, S., Gupta, D., Das, L. K.: PID & LQR control for a quadrotor: Modeling and simulation. In: 2014 International Conference on Advances in Computing Proceedings, Communications and Informatics, pp. 796–802 (2014)
8. Krstić, M., Kanellakopoulos, I., Kokotović, P.V.: Nonlinear and adaptive control design. John Wiley and Sons, New York (1995)
9. Loianno, G., Brunner, C., McGrath, G., Kumar, V.: Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU. IEEE Robotics and Automation Letters **2**(2), 404–411 (2017)
10. Luukkonen, T.: Modelling and control of quadcopter. Independent research project in applied mathematics, http://sal.aalto.fi/publications/pdf-files/eluu11_public.pdf. Last accessed 14 Sept 2020
11. Madani, T., Benallegue, A.: Backstepping control for a quadrotor helicopter. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems Proceedings, pp. 3255–3260 (2006)
12. Nguyen, N.T., Prodan, I., Stoican, F., Lefevre, L.: Reliable nonlinear control for quadcopter trajectory tracking through differential flatness. IFAC-PapersOnLine **50**(1), 6971–6976 (2017)

13. Rashid, M.I., Akhtar, S.: Adaptive control of a quadrotor with unknown model parameters. In: 9th International Bhurban Conference on Applied Sciences and Technology Proceedings, pp. 8–14 (2012)
14. Sanchez-Cuevas, P., Heredia, G., Ollero, A.: Characterization of the aerodynamic ground effect and its influence in multirotor control. International Journal of Aerospace Engineering. **2017**, (2017)
15. Shi, D., Wu, Z., Chou, W.: Generalized extended state observer based high precision attitude control of quadrotor vehicles subject to wind disturbance. IEEE Access **6**, 32349–32359 (2018)
16. Spedicato, S., Notarstefano, G.: Minimum-time trajectory generation for quadrotors in constrained environments. IEEE Transactions on Control Systems Technology **26**(4), 1335–1344 (2018)
17. Wang, R., Liu, J.: Adaptive formation control of quadrotor unmanned aerial vehicles with bounded control thrust. Chinese Journal of Aeronautics **30**(2), 807–817 (2017)
18. Simulink Support Package for Parrot Minidrones. https://www.mathworks.com/help/supportpkg/parrot. Last accessed 14 Sept 2020