Project Report

On the

Optimizing PID controller gains to model the performance of a quadcopter

Written by:

Mahmood Rezaee Qotb Abadi	**40217213**
Kimia Firoozi	**40190505**

Master of Applied Science

Department of Electrical and Engineering

Concordia University

Professor:

Dr. Khashayar Khorasani

20 June 2022

Table of Contents

Abstract. This document presents an optimization-based method to tune the gains of the PID controller to control the altitude and attitude of a quadcopter using MATLAB and Simulink. The quadcopter model is derived using Newton's equations of motion and implemented in Simulink. This work is done based on the article [1]. The optimization method used in the article is "random search" and in this project "reduced gradient method" is implemented in MATLAB additionally and the results are compared.

1 Introduction

1.1 Quadcopter

Flying objects have always intrigued man's interest, spurring all sorts of research and advancement. A quadcopter, sometimes known as a quadrotor, is a drone or unmanned aerial vehicle with four rotors, each with its motor and propeller. A quadcopter can be controlled manually or autonomously. This project examines the dynamics of a four-rotor unmanned aerial vehicle.

1.2 Controller

Controlling such multidimensional systems necessitates the use of appropriate controllers. The usage of PID controllers has been studied in this area of research. PID controllers, on the other hand, are the most popular controller Despite their widespread use in quadcopters, selecting an optimal PID setting remains a challenge.

1.3 Optimization

Most investigations on the subject have been carried out with the aid of computer tools such as MATLAB. By minimizing the error integral, this article proposes using MATLAB to identify the best PID controller for a quadcopter. One for the altitude control and three for the attitude control, i.e., roll, pitch, and yaw.

The critical difference between them is that one looks for a local minimum while the other oversees finding a global minimum of a limited function. The used algorithms in MATLAB are "Random Search" [1], and "Reduced Gradient Algorithm".

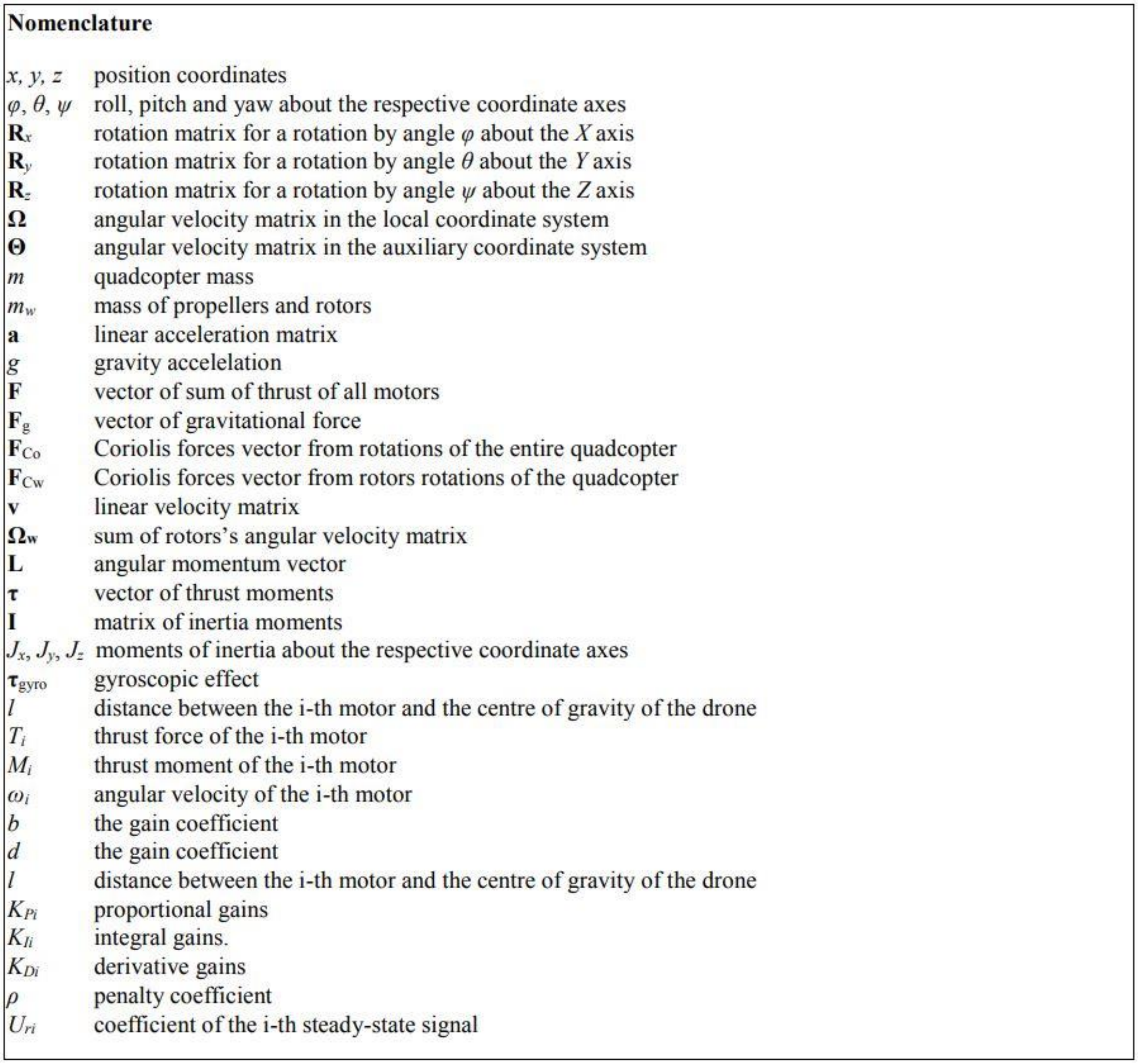
2 Literature Review

Finding an optimal control technique for quadrotors was an important aspect of this thesis. The methodologies were researched based on a simplified model, from theoretical development to final trials, and were applied to the attitude control of the helicopter using a Proportional Integral Derivative (PID). The main algorithms that we compared during this project were Random Search[1] and Reduced Gradient Algorithm with the first algorithm being considered in the main article, but the second method is applied in this case study retrieved from our optimization course, and this project will go over the details of the second algorithm that were used in our case study.

3 Quadrotor Model

3.1 Notations

The table below indicates the notations used throughout this report. All notations are from the main article [1].



Nomenclature

x, y, z	position coordinates
φ, θ, ψ	roll, pitch and yaw about the respective coordinate axes
$\mathbf{R}_x$	rotation matrix for a rotation by angle φ about the X axis
$\mathbf{R}_y$	rotation matrix for a rotation by angle θ about the Y axis
$\mathbf{R}_z$	rotation matrix for a rotation by angle ψ about the Z axis
$\mathbf{\Omega}$	angular velocity matrix in the local coordinate system
$\mathbf{\Theta}$	angular velocity matrix in the auxiliary coordinate system
m	quadcopter mass
m_w	mass of propellers and rotors
$\mathbf{a}$	linear acceleration matrix
g	gravity accelelation
$\mathbf{F}$	vector of sum of thrust of all motors
$\mathbf{F}_g$	vector of gravitational force
$\mathbf{F}_{Co}$	Coriolis forces vector from rotations of the entire quadcopter
$\mathbf{F}_{Cw}$	Coriolis forces vector from rotors rotations of the quadcopter
$\mathbf{v}$	linear velocity matrix
$\mathbf{\Omega}_w$	sum of rotors's angular velocity matrix
$\mathbf{L}$	angular momentum vector
$\boldsymbol{\tau}$	vector of thrust moments
$\mathbf{I}$	matrix of inertia moments
J_x, J_y, J_z	moments of inertia about the respective coordinate axes
$\boldsymbol{\tau}_{gyro}$	gyroscopic effect
l	distance between the i-th motor and the centre of gravity of the drone
T_i	thrust force of the i-th motor
M_i	thrust moment of the i-th motor
ω_i	angular velocity of the i-th motor
b	the gain coefficient
d	the gain coefficient
l	distance between the i-th motor and the centre of gravity of the drone
K_{Pi}	proportional gains
K_{Ii}	integral gains.
K_{Di}	derivative gains
ρ	penalty coefficient
U_{ri}	coefficient of the i-th steady-state signal

Table. 1 Notation [1]

3.2 Nonlinear Quadrotor Model

Quadcopters typically have two clockwise (CW) and two counterclockwise (CCW) rotors (CCW). The rotors and propellers are equidistant for optimal performance and the simplest control methods. A quadrotor system is as follows [1][2][3][4][5]:

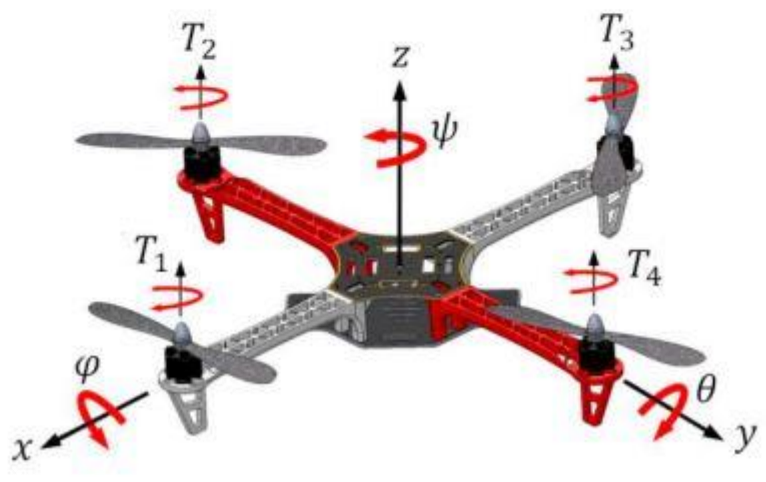


Fig. 1 A quadrotor system [1]

The altitude and attitude of the drone are reported in the form of vector q as follows. The first three parameters describe the drone's position in space, while the remaining three parameters describe its rotations:

$$\mathrm{q} = [x \quad y \quad z \quad \varphi \quad \theta \quad \psi] \tag{1}$$

The theoretical calculation of the angular velocity in the local coordinate system is given as follows:

$$\Omega = \begin{bmatrix} \omega_x \\ \omega_x \\ \omega_x \end{bmatrix} = \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + R_x \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_x R_y \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = P_1^2 \dot{\Theta} \tag{2}$$

Where:

$$P_1^2 = \begin{bmatrix} 1 & 0 & cos\varphi \\ 0 & cos\varphi & sin\varphi cos\theta \\ 0 & cos\varphi & cos\varphi sin\theta \end{bmatrix} \tag{3}$$

In order to find $\dot{\Theta}$, it is necessary to calculate the inverse of P_1^2:

$$P_2^1 = (P_1^2)^{-1} = \begin{bmatrix} 1 & sin\varphi tg\theta & cos\varphi tg\theta \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi sec\theta & cos\varphi sec\theta \end{bmatrix} \tag{4}$$

Therefore,

$$\dot{\Theta} = P_1^2 \Omega \tag{5}$$

The general form of the equation form for defining the linear acceleration using Newton's second law of motion is given:

$$ma = F - F_g - F_{Co} - F_{Cw} \tag{6}$$

Where:

$$F_g = P_2^1 m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} -mgsin\theta \\ -mgsin\varphi cos\theta \\ mgcos\varphi cos\theta \end{bmatrix} \qquad (7)$$

In order to find $\dot{\Theta}$, it is necessary to calculate the inverse of P_1^2: Therefore,

$$\dot{\Theta} = P_1^2 \Omega \qquad (8)$$

The general form of the equation form for defining the linear acceleration using Newton's second law of motion is given:

$$ma = F - F_g - F_{Co} - F_{Cw} \qquad (9)$$

The force of gravity can be described as a projection of force from one local coordinate system to another. The overall mass of the quadcopter is multiplied by the gravity acceleration, and the rotation matrix:

$$F_g = P_2^1 m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} -mgsin\theta \\ -mgsin\varphi cos\theta \\ mgcos\varphi cos\theta \end{bmatrix} \qquad (10)$$

One can describe the first of Coriolis forces using the rotation of the whole quadcopter as follows:

$$F_{Co} = 2m\,\Omega \times \mathrm{v} = 2m \begin{bmatrix} v_z\omega_y - v_y\omega_z \\ v_x\omega_z - v_z\omega_x \\ v_y\omega_x - v_x\omega_y \end{bmatrix} \qquad (11)$$

And the second Coriolis forces from the rotation of the rotors:

$$F_{Cw} = 2m_w\Omega_w \times \mathrm{v} = 2m_w \begin{bmatrix} -v_y(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ v_x(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ 0 \end{bmatrix} \qquad (12)$$

Where:

$$\Omega_w = \begin{bmatrix} 0 \\ 0 \\ \omega_1 + \omega_2 + \omega_3 + \omega_4 \end{bmatrix} \qquad (13)$$

The thrust F, which is the sum of all motor thrust, is created vertically upwards along the quadcopter's vertical axis z, and it is defined as follows:

$$F = \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ T_1 + T_2 + T_3 + T_4 \end{bmatrix} \qquad (14)$$

To calculate the angular acceleration mathematically, the Euler equation of a rigid body motion defined in the following equation can be used.

$$\tau = \frac{dL}{dt} + \Omega \times \mathrm{L} \qquad (15)$$

Where for rigid body one can write:

$$\mathrm{L} = \mathrm{I}\Omega \tag{16}$$

Where "I" is the matrix of inertia moments:

$$FI = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \tag{17}$$

Assuming the constancy of inertia moment, one may obtain the following equation:

$$\tau = \mathrm{I}\dot{\Omega} + \Omega \times \mathrm{I}\Omega \tag{18}$$

Due to the gyroscopic effect caused by the motors and rotors, the last equation must be modified as follows:

$$\tau = \mathrm{I}\dot{\Omega} + \Omega \times (\mathrm{I}\Omega) + \tau_{gyro} \tag{19}$$

Where:

$$\tau_{gyro} = \Omega \times L_p \tag{20}$$

The angular acceleration is the needed quantity in Equation (19), thus after modifications, it can be represented as:

$$\dot{\Omega} = I^{-1} + (\tau - \Omega \times (\mathrm{I}\Omega) + \tau_{gyro} \tag{21}$$

The angular momentum vector is created by multiplying the rotor's moment of inertia (I_w) by the rotor's rotational velocity in the rotation axis (Ω_w):

$$L_p = I_w\Omega_w = \begin{bmatrix} 0 \\ 0 \\ I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4) \end{bmatrix} \tag{22}$$

Now one can write the Equation (20) as given:

$$\tau_{gyro} = \begin{bmatrix} \omega_x \\ \omega_x \\ \omega_x \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4) \end{bmatrix} = \begin{bmatrix} \omega_y I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ -\omega_x I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ 0 \end{bmatrix} \tag{23}$$

Where:

$$\tau = \mathrm{I}\dot{\Omega} + \Omega \times (\mathrm{I}\Omega) + \tau_{gyro} \tag{24}$$

In order to find $\dot{\Theta}$, it is necessary to calculate the inverse of P_1^2:

$$\dot{\Omega} = I^{-1} + (\tau - \Omega \times (\mathrm{I}\Omega) + \tau_{gyro} \tag{25}$$

The moments are given for the configuration coordinates responsible for the rotation by φ, θ, ψ as follows:

$$M_\varphi = -T_1 l - T_3 l \tag{26}$$

$$M_\theta = T_2 l - T_4 l \tag{27}$$

$$M_\psi = \sum_{i=1}^{4} M_i \tag{28}$$

As it is clear from the above equations, in order to control the quadcopter effectively, one needs the thrust force generated along the z-axis (F_z) and the rotational moments above. All these quantities for the i-th motor can be calculated using the angular velocity of that rotor and the gain coefficient b:

$$T_i = b\omega_i^2 \tag{29}$$

$$M_i = d\omega_i^2 \tag{30}$$

After substituting Equations (29) and (30) in Equations (26) – (28), one can obtain:

$$\tau = \begin{bmatrix} M_\varphi \\ M_\theta \\ M_\psi \end{bmatrix} = \begin{bmatrix} Ib(-\omega_1^2 + -\omega_3^2) \\ Ib(\omega_2^2 + -\omega_4^2) \\ d(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \tag{31}$$

To write a complete model of the quadcopter, one can divide the equations into two separated parts. The first part refers to linear acceleration in the local coordinate system:

$$\begin{aligned} \mathrm{a} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{b}{m}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} - \begin{bmatrix} -gsin\theta \\ gsin\varphi cos\theta \\ gcos\varphi cos\theta \end{bmatrix} - 2\begin{bmatrix} v_z\omega_y - v_y\omega_z \\ v_x\omega_z - v_z\omega_x \\ v_y\omega_x - v_x\omega_y \end{bmatrix} \\ - 2\frac{m_w}{m}\begin{bmatrix} -v_y(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ v_x(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ 0 \end{bmatrix} \end{aligned} \tag{32}$$

The second part contains Euler rates, which will be utilized to determine changes in the orientation of the quadcopter local system in relation to the auxiliary system:

$$\dot{\Theta} = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_x + \omega_y sin\varphi tg\theta + \omega_z cos\varphi tg\theta \\ \omega_y cos\varphi + \omega_z sin\varphi \\ \omega_y sin\varphi sec\theta + \omega_z cos\varphi sec\theta \end{bmatrix} \tag{33}$$

And the angular accelerations:

$$\dot{\Omega} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \frac{lb}{J_x}(-\omega_1^2 + \omega_3^2) \\ \frac{lb}{J_y}(-\omega_2^2 + \omega_4^2) \\ \frac{d}{J_x}(\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} - \begin{bmatrix} \frac{1}{J_x}(\omega_y\, \omega_z(J_z + J_y) \\ \frac{1}{J_y}(\omega_x\, \omega_z(J_x + J_z) \\ \frac{1}{J_z}(\omega_x\, \omega_y(J_y + J_x) \end{bmatrix} - \begin{bmatrix} \frac{1}{J_x}(-\omega_y I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4)) \\ \frac{1}{J_y}(-\omega_x I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4)) \\ 0 \end{bmatrix} \tag{34}$$

The figure below shows the connection between these two parts:

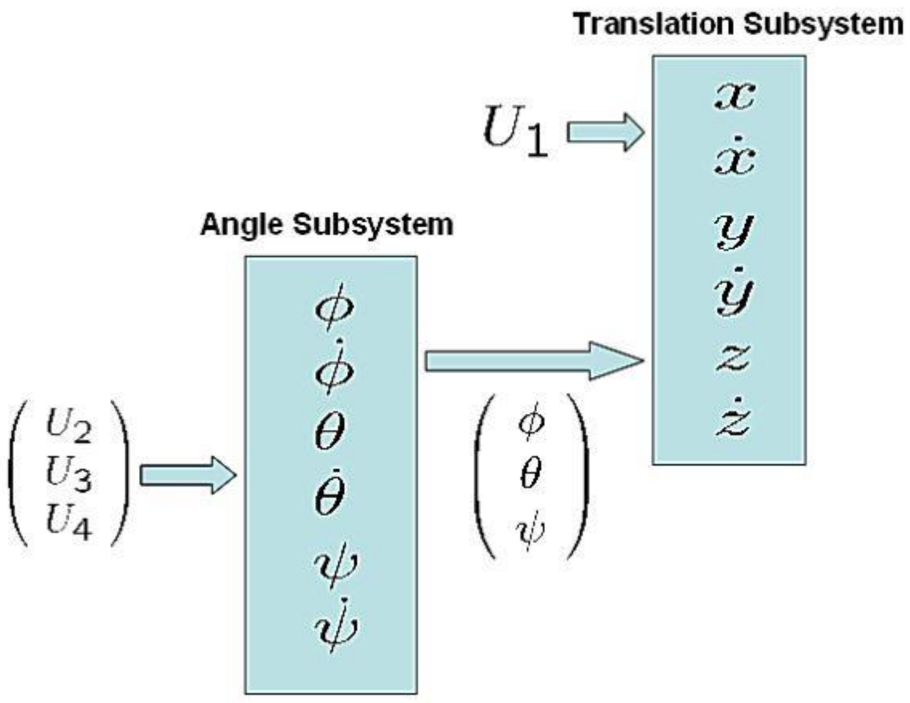


Fig. 2 Connection between the angular subsystem and the translation (linear motion) subsystem [2]

4 Control System

There are several controllers such as Linear Quadratic Regulators (LQR) and fuzzy logic controllers that can be used to control the quadcopter. In this project, one of the simplest but widely employed and effective controllers, the Proportional-Integral_Derivative controller (PID), is utilized to control the quadcopter's altitude and attitude (pitch, roll, yaw angle).

The equations (33)-(34) present that altering the angular velocities of the rotors is necessary for the physical control of a quadcopter. The thrust force is assumed to be provided by all four motors in equation (29). The four controllers are used to control the altitude z (by manipulating the force F_z), roll (by controlling the moment M), pitch (by managing the moment M), and yaw (by controlling the moment M) in practice. The equations for the specific PID controllers are as follows: (35)– (38):

$$U_1 = K_{P1}e_1 + K_{L1}\int e_1(\tau)d\tau - K_{D1}\dot{z} \tag{35}$$

$$U_2 = K_{P2}e_2 + K_{L2}\int e_2(\tau)d\tau - K_{D2}\dot{z} \tag{36}$$

$$U_3 = K_{P3}e_3 + K_{L3}\int e_3(\tau)d\tau - K_{D3}\dot{z} \tag{37}$$

$$U_4 = K_{P4}e_4 + K_{L4}\int e_4(\tau)d\tau - K_{D4}\dot{z} \tag{38}$$

The control errors are the altitude error e_1, the roll attitude error e_2, the pitch attitude error e_3, and the yaw attitude error e_4:

$$e_1 = z_z - z \tag{39}$$

$$e_2 = \varphi_z - \varphi \tag{40}$$

$$e_3 = \theta_z - \theta \tag{41}$$

$$e_4 = \psi_z - \psi \tag{42}$$

Where $z_z, \varphi_z, \theta_z, and\ \psi_z$ are setpoints. It is necessary to determine the angular velocity of each motor by solving the matrix equation (43) since the force F_z and the moments M_φ, M_θ and M_ψ cannot be controlled directly:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -bl & 0 & bl & 0 \\ 0 & bl & 0 & -bl \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{43}$$

$$\omega_1 = \sqrt{\frac{U_1}{4b} - \frac{U_2}{2bl} + \frac{U_4}{4d}} \tag{44}$$

$$\omega_2 = \sqrt{\frac{U_1}{4b} + \frac{U_3}{2bl} - \frac{U_4}{4d}} \tag{45}$$

$$\omega_3 = \sqrt{\frac{U_1}{4b} + \frac{U_2}{2bl} + \frac{U_4}{4d}} \tag{46}$$

$$\omega_4 = \sqrt{\frac{U_1}{4b} - \frac{U_3}{2bl} - \frac{U_4}{4d}} \tag{47}$$

5 Optimization

The most important step in designing the PID controller for a quadcopter is tuning each gain (K_{Pi}, K_{Ii}, K_{Di}) in a way that the controlled system satisfies the requirements. To tune these gains the minimization algorithm is used which seeks to discover the best PID control settings with the lowest control quality indicator. Here, the error integral for the controlled coordinates $z, \varphi, \theta\ and\ \psi$ is the quality indicator.

To decrease the input signal level, the quality indicator is calculated by multiplying the input signal by the penalty coefficient ρ. The quality indicator is written as follows:

$$QI = \int_{t_0}^{t_k} |e_i| + |\rho(U_i - U_{ri})| dt \tag{48}$$

As written in the article [1], $\rho = 1$ can be used to complete the minimization function. Also, since certain gains might become negative, unconstrained techniques are unsuitable. Therefore, the following constraints (calculated practically) must be accurately defined for the optimization problem to obtain satisfactory gains.

$$5 \le K_{P1} \le 17\,, \qquad 1 \le K_{Pi} \le 10\,, \qquad for\ i = 2, 3, 4 \tag{49}$$

$$0.1 \le K_{Ii} \le 0.5\,, \qquad for\ i = 2, 3, 4 \tag{50}$$

$$0.1 \le K_{Di} \le 2\ \,, \qquad for\ i = 2, 3, 4 \tag{51}$$

Therefore, there are four minimization problem that needs to be solved for the altitude (z) and attitude $(\varphi, \theta\ and\ \psi)$ as follows:

$$\min QI = \int_{t_0}^{t_k} |e_i| + |\rho(U_i - U_{ri})| dt\,, \qquad for\ i = 2, 3, 4 \tag{52}$$

$$subject\ to\colon\ 5 \le K_{P1} \le 17\,, \qquad 1 \le K_{Pi} \le 10\,, \qquad for\ i = 2, 3, 4 \tag{53}$$

$$0.1 \le K_{Ii} \le 0.5\,, \qquad for\ i = 2, 3, 4 \tag{54}$$

$$0.1 \le K_{Di} \le 2\ \,, \qquad for\ i = 2, 3, 4 \tag{55}$$

These problems are all nonlinear optimization problems with boxing constraints. To solve these problems, the article [1] used the random search method. In this project, the reduced gradient method is used for optimization.

The reduced gradient method is an optimization method for equality restricted situations as our problem here. The reduced gradient method is based on a basic variable elimination approach (Abadie, 1970). The generalized reduced gradient (GRG) technique is a nonlinear inequality constraint-aware variant of the reduced gradient method which here we don't need to use this generalized method since the constraint here is linear. In this approach, a search direction is chosen such that the present active constraints stay precisely active for every little move. Therefore, this method is an extension of the gradient method for constrained problems as follows:

$$\min f(x) \tag{56}$$

$$subject\ to: \ \ Ax = b, \qquad x \geq 0 \tag{57}$$

In this method, it is assumed that $f(x)$ is continuously differentiable which in our problem it is true since the QI is an integral function that inside the integral, we have a continuous function.

We may create equality restricted NLP model by converting inequality constraints to equalities by adding slack variables. We may also use the potential constraint technique and consider all the subproblem's constraints as equalities.

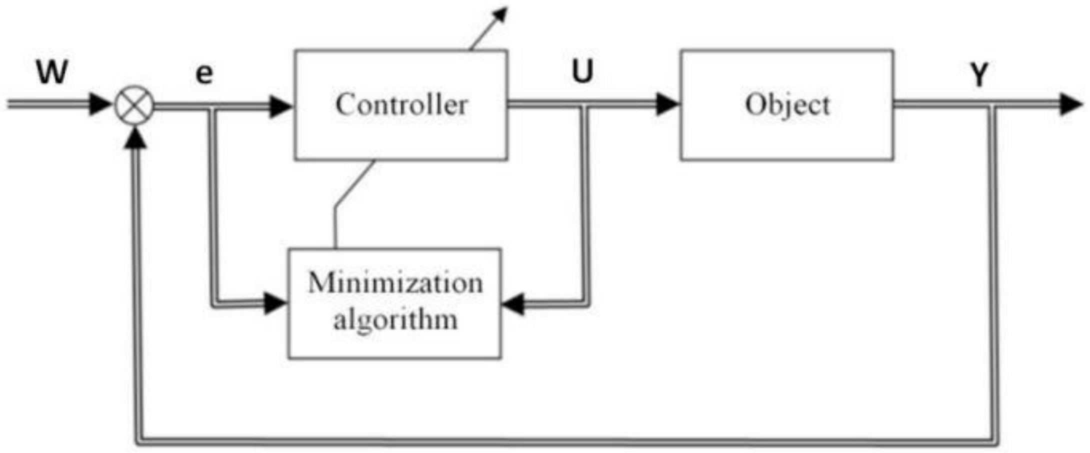


Fig. 3 Block diagram of the PID controller with the minimization algorithm [1]

6 Simulations and Results

6.1 Simulations

In order to simulate the system, first, the quadcopter equation derived in section 3.2 was implemented in Simulink. The two subsystems (angular subsystem and translation subsystem) were implemented, and the quadcopter is controlled using $U_1, U_2, U_3, and\ U_4$ as control inputs of the system. (You can find the block diagrams of the system implemented in Simulink in detail in section 9 (appendices).)

The initial conditions and parameters of the quadcopter are from the values given in the article [1] as follows:

$x(0) = 0.1,\ x'(0) = 0,\ y(0) = 0.1,\ y'(0) = 0,\ z(0)=5,\ z'(0) = 0,\ \varphi(0) = 0,\ \theta(0) = 0,\ \psi(0) = 0,\ \omega_x(0)=0, \omega_y(0)=0,\ \omega_z(0)=0,\ i_{e1}(0)=0,\ i_{e2}(0)=0,\ i_{e3}(0)=0,\ i_{e4}(0)=0,$

Table 2. Parameters of the quadcopter control model

Parameter	Value	Unit
m	0.65	kg
m_w	0.01	kg
l	0.23	m
g	9.81	kg m/s^2
I_w	0.000065	kg m^2
J_x	0.0075	kg m^2
J_y	0.0075	kg m^2
J_z	0.0013	kg m^2

Four PID controllers were designed and implemented just to start and stabilize the system. The initial gains were set just to stabilize the system in a way that quadcopter goes up to a certain point along the z-axis and move along the x-axis and a small movement along the y-axis. Also, the pitch, roll, and yaw angle reach a certain value.

The values of the quadcopter's translation (x, y, and z) with these settings are shown in the figure below. The vertical axis is the value of each of these states (meter), and the horizontal axis is time (s).

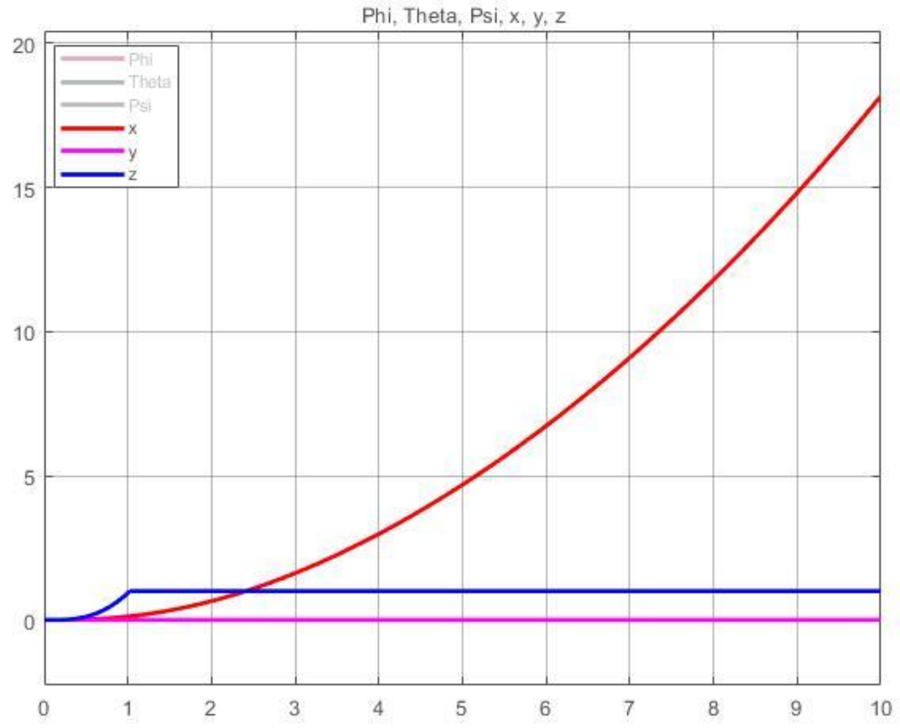


Fig. 4 Quadcopter's translation (x, y, and z) with initial PID settings.

The values of the quadcopter's angular states (φ, θ and ψ) with these settings are shown in the figure below. The vertical axis is the value of each of these states (rad), and the horizontal axis is time (s).

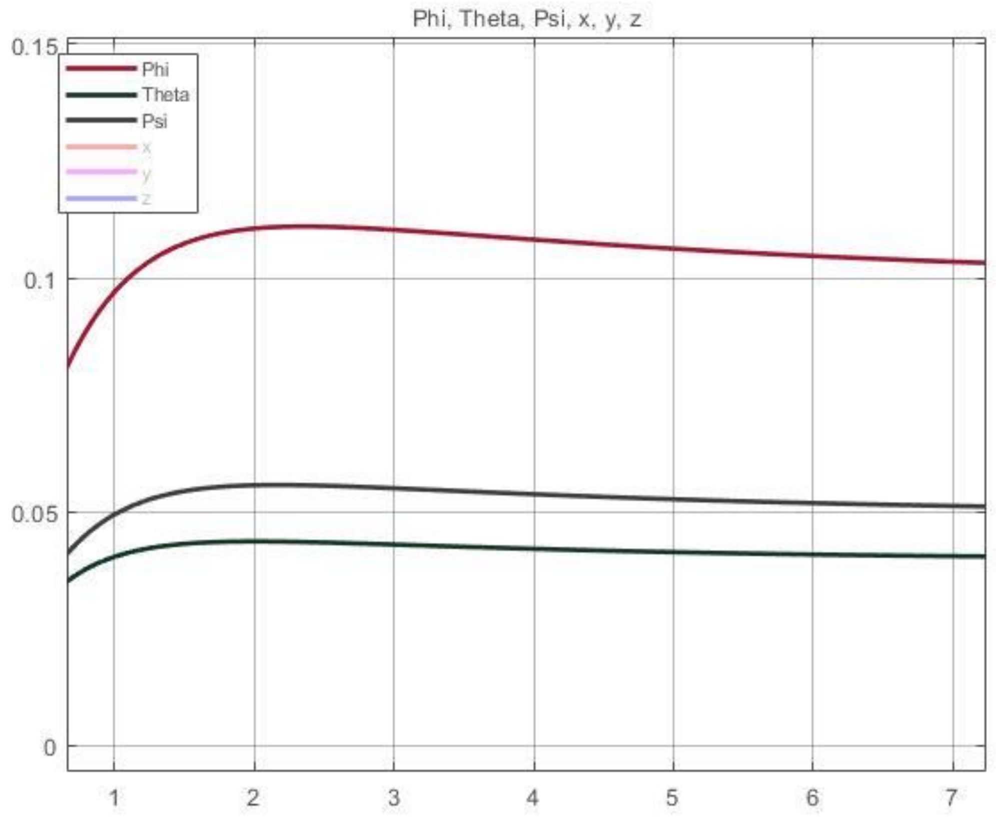


Fig. 5 Quadcopter's angular values (φ, θ, and ψ) with initial PID settings.

As it is clear from Fig.4 and Fig.5, the desired behavior for the quadcopter is reached. Although the PID gains are not the optimal gains we can find. To find the optimal PID gains that minimize the quality indicator mentioned in Equation (48), solving the optimization problem given in (52) is necessary. In the next section, the optimization method to do so is explained.

6.2 Results of implementation of the reduced gradient method

To implement the reduced gradient method for optimization, MATLAB coding is used. (Not the optimization toolbox) (The code is provided in the appendices section.)

After running the quadcopter system with the initial conditions mentioned in the article [1], and finding the quality indicator function, we used the reduced gradient method to find the optimal PID gains. Then if we simulate the system with these new optimal gains, the results are as given in fig.6 and fig.7. As it is clear in the figures, not only do we achieve the desired behavior but also the desired values for z, φ, θ, and ψ given in the article [1] as 6m, 0.1rad, 0.04rad, and 0.05rad is reached in a shorter time.

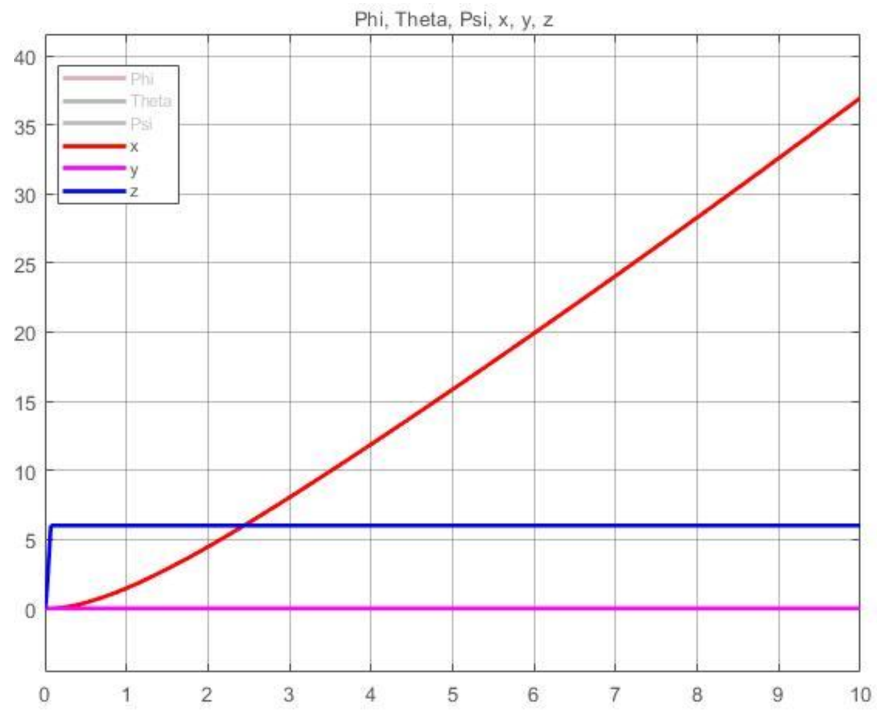


Fig. 6 Quadcopter's translation (x, y, and z) with optimal PID settings.

The values of the quadcopter's angular states (φ, θ *and* ψ) with these settings are shown in the figure below. The vertical axis is the value of each of these states (rad), and the horizontal axis is time (s).

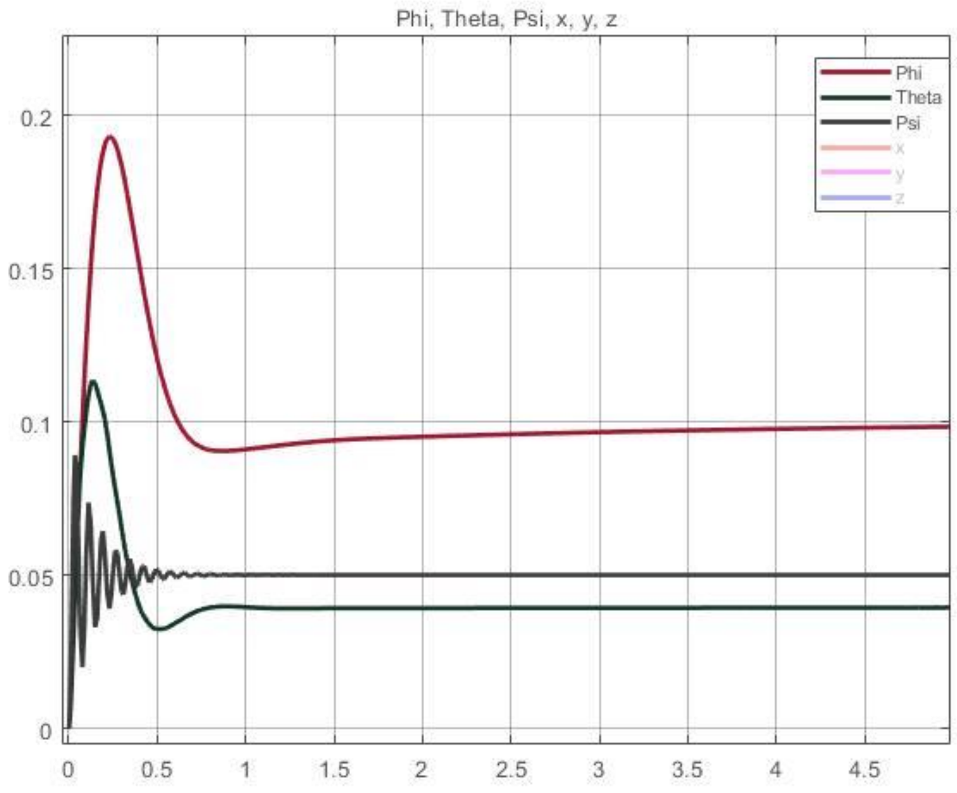
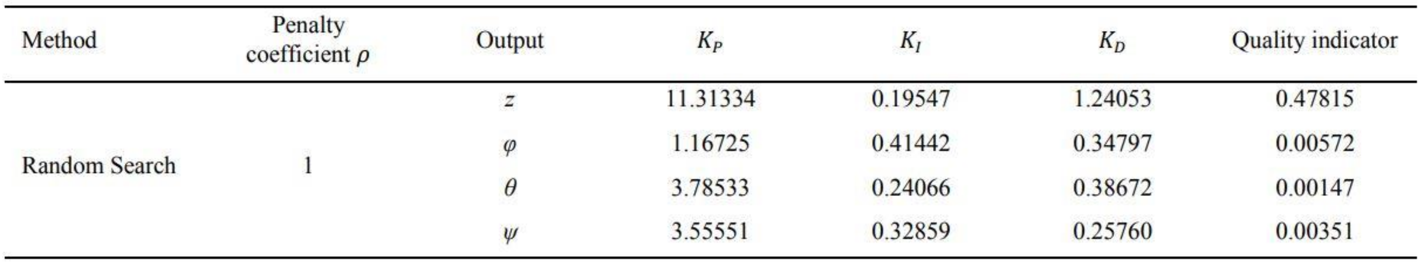


Fig. 7 Quadcopter's angular values (φ, θ, and ψ) with optimal PID settings.

7 Comparative Study

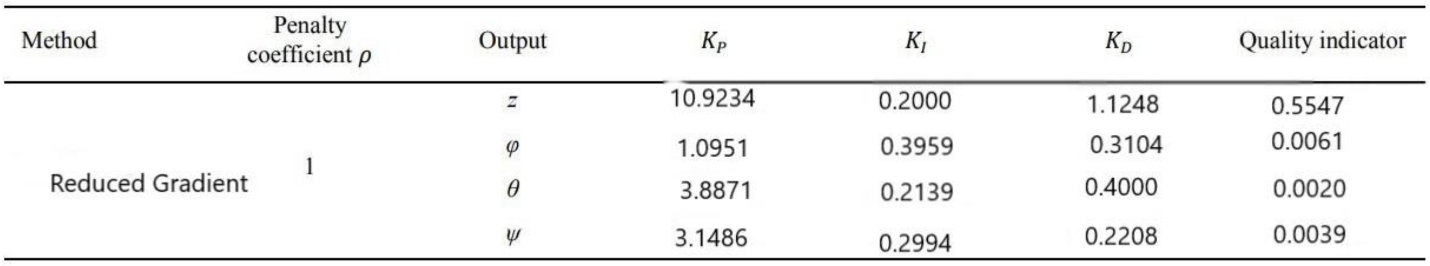
The following table is the results from the article [1] using the Random Search constrained method. As it is given, for each z, φ, θ and ψ, there are optimal PID gains and the resulting quality indicator.

Table 1. PID gains obtained using the RandomSearch constrained method



Method	Penalty coefficient ρ	Output	K_P	K_I	K_D	Quality indicator
Random Search	1	z	11.31334	0.19547	1.24053	0.47815
		φ	1.16725	0.41442	0.34797	0.00572
		θ	3.78533	0.24066	0.38672	0.00147
		ψ	3.55551	0.32859	0.25760	0.00351

The results from the reduced gradient method after 4 iterations are as follows:



Method	Penalty coefficient ρ	Output	K_P	K_I	K_D	Quality indicator
Reduced Gradient	1	z	10.9234	0.2000	1.1248	0.5547
		φ	1.0951	0.3959	0.3104	0.0061
		θ	3.8871	0.2139	0.4000	0.0020
		ψ	3.1486	0.2994	0.2208	0.0039

As it is clear, the Random search method performs better and is more accurate in finding the optimal gains, since the Quality indicator for the Random search method is lower than the reduced gradient method.

8 Conclusion

This report presented new methods rather than the random search method in the article [1] for optimizing the tuning process of the PID controller. This tuned PID controller is used for a nonlinear quadcopter to control the altitude and attitude (pitch, roll, yaw angles). The optimization methods are utilized to minimize the quality indicator. It was necessary for this minimization problem to have constraints on PID gains to be solvable and has a minimum. The modeling of the quadcopter and implementation of the optimization methods is done using MATLAB and Simulink which are one the best modeling programs and have several toolboxes to ensure fast optimization of complex criteria. The optimization results were considered without the input signal or the penalty coefficient. The method used in the article [1] for optimization was random search and the method in this project is reduced gradient. Comparing results from these two methods shows that the random search method performs accurately in finding the optimal gains.

9 Appendices

This section presents the block diagrams of the quadcopter in Simulink in detail. The following figure is the block diagram of the whole system with four PID controllers. The figures after are each subsystem in detail. These subsystems are from the equations in section 3.

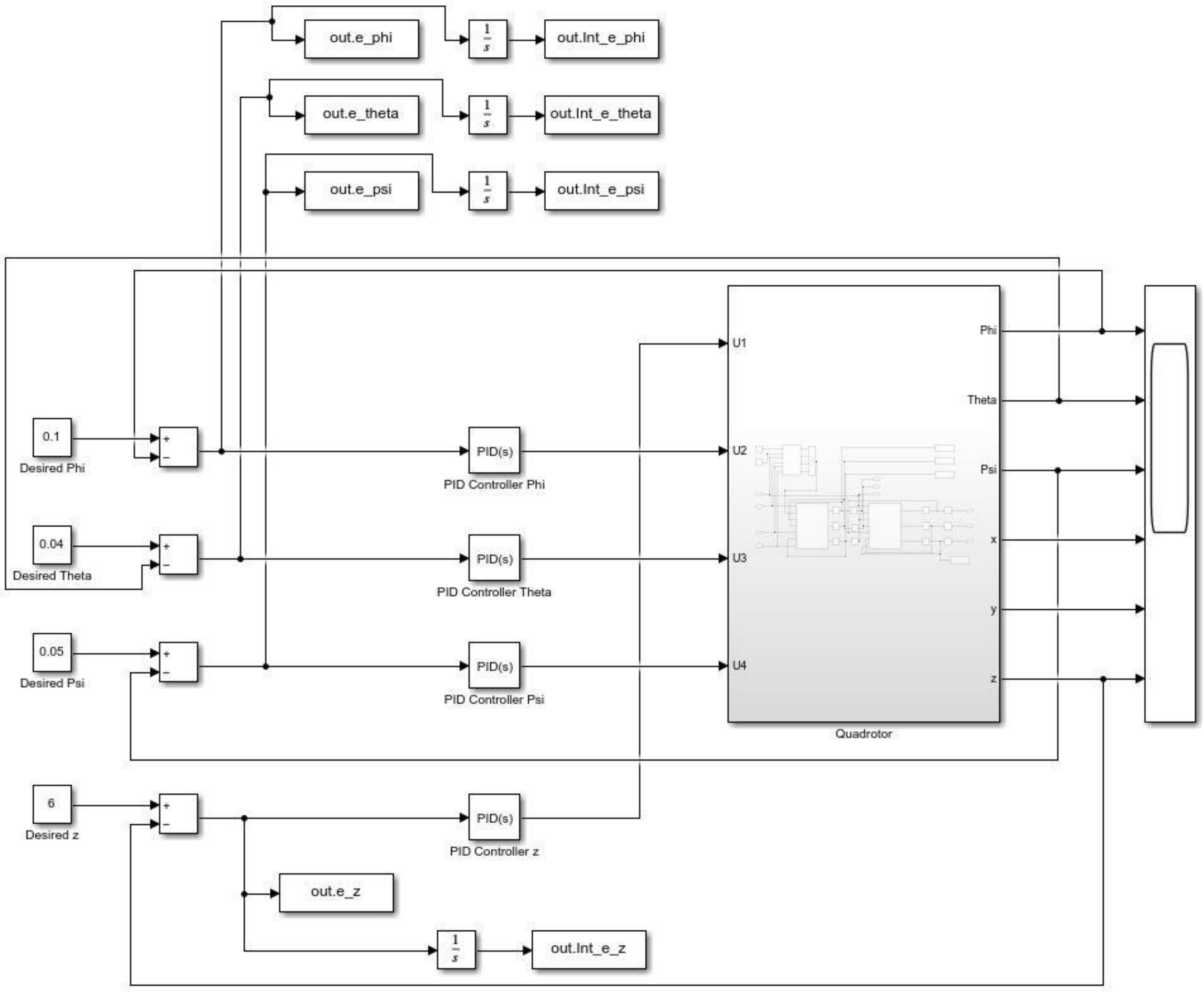


Fig. 8 Block diagram of the complete system (quadcopter) with four PID controller

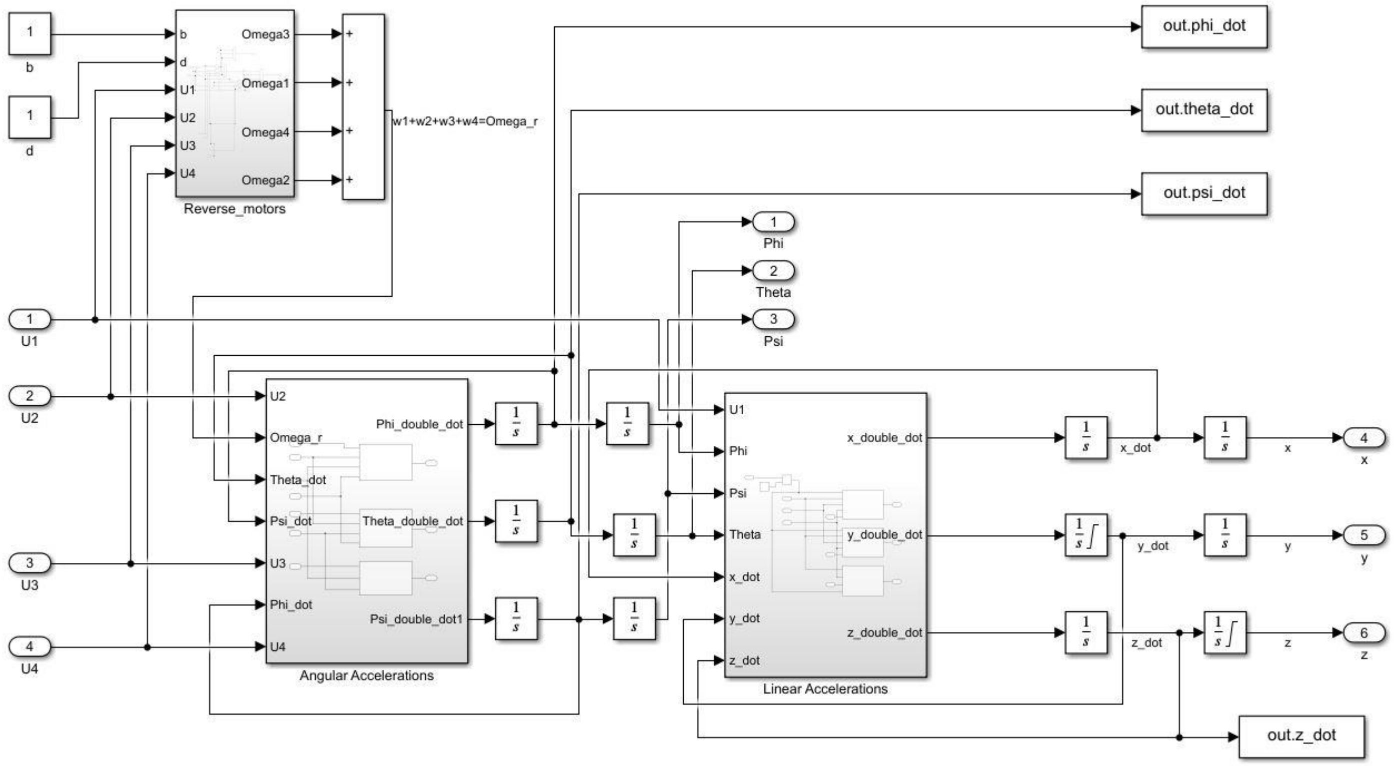


Fig. 9 Block diagram of the nonlinear quadcopter system

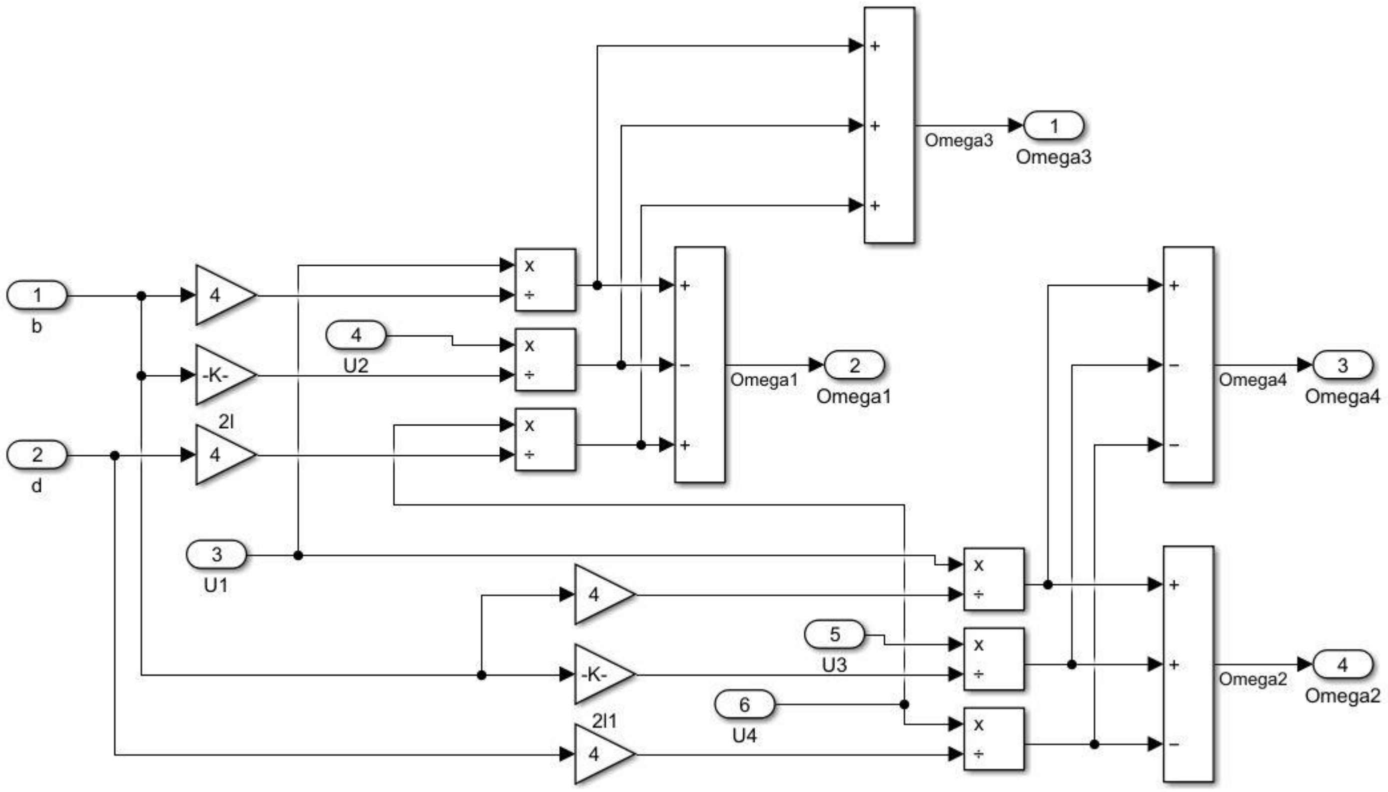


Fig. 10 Block diagram of the reverse motor subsystem. Equations ()

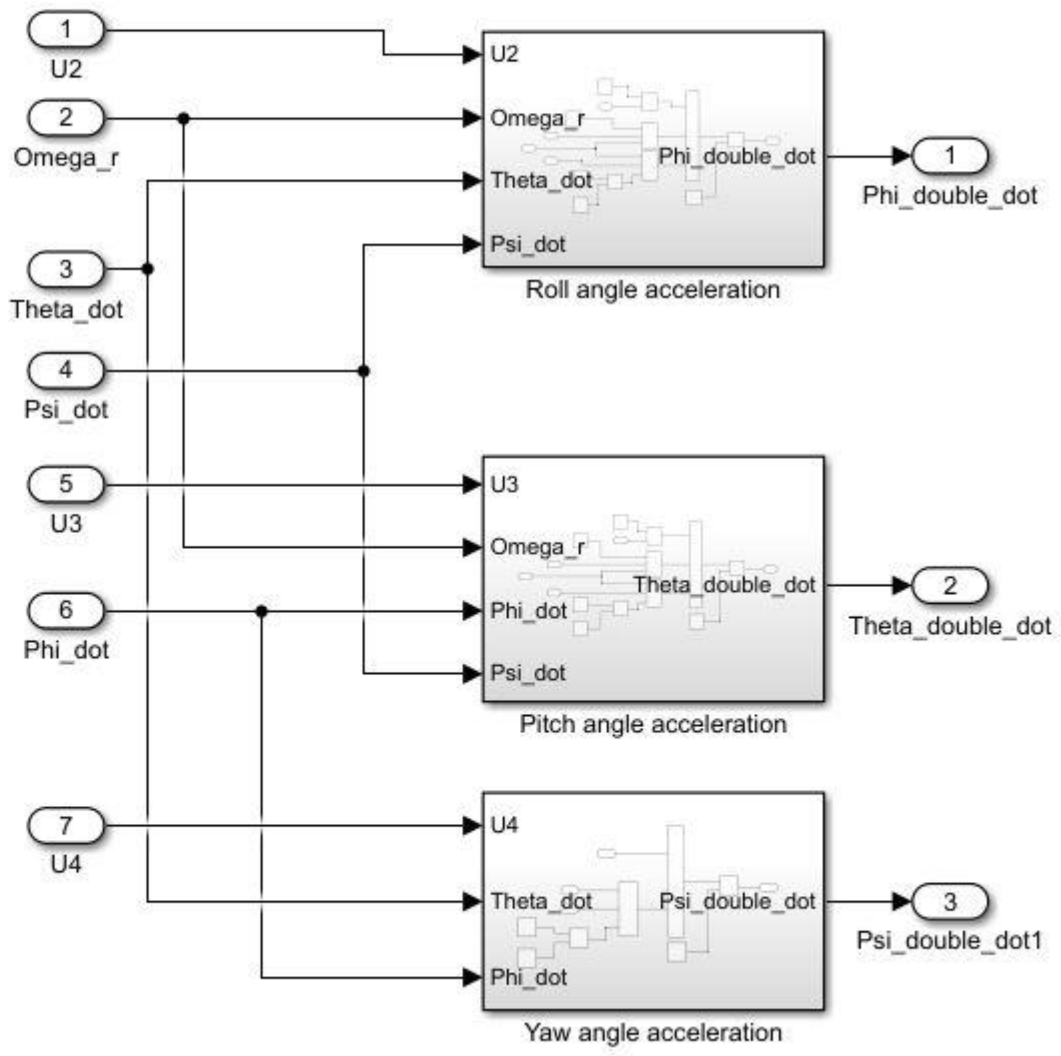


Fig. 11 Block diagram of the angular acceleration subsystem

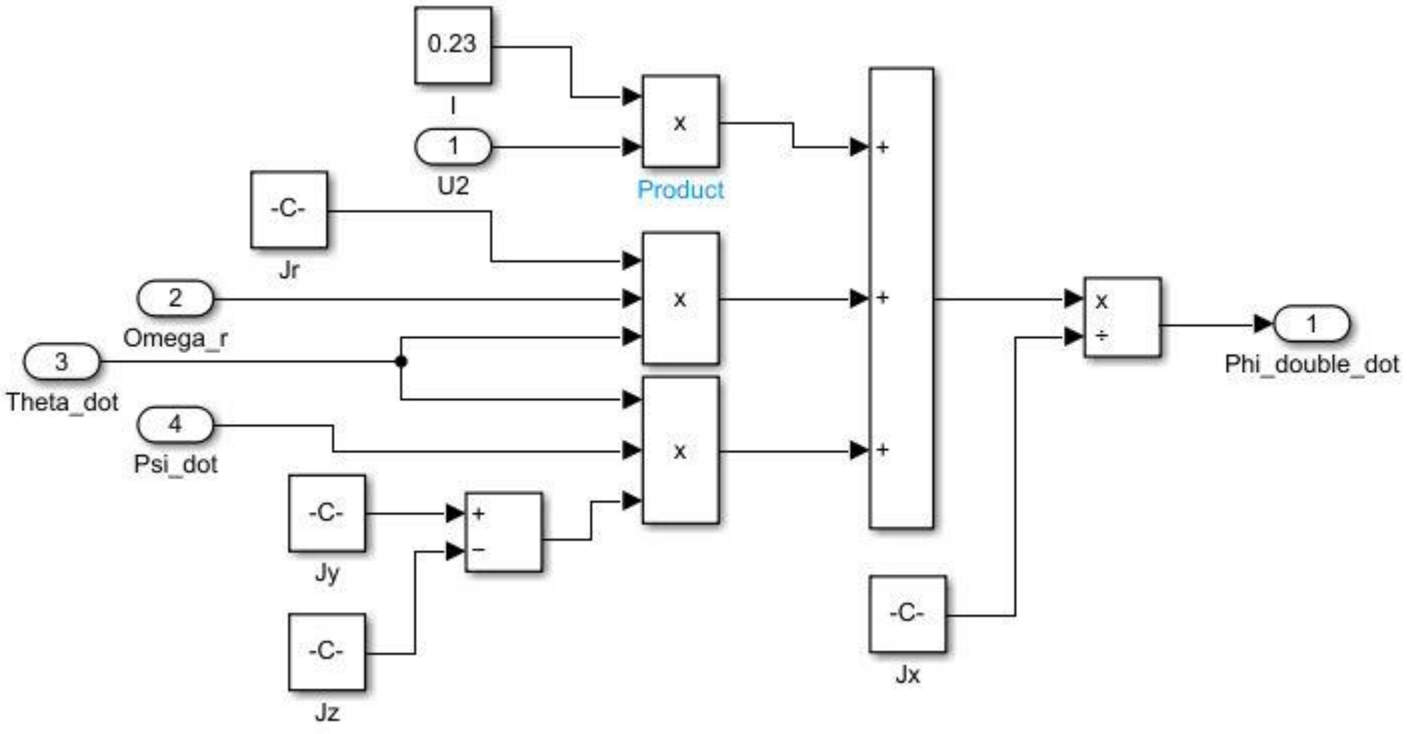


Fig.12 Block diagram of the roll angle acceleration subsystem

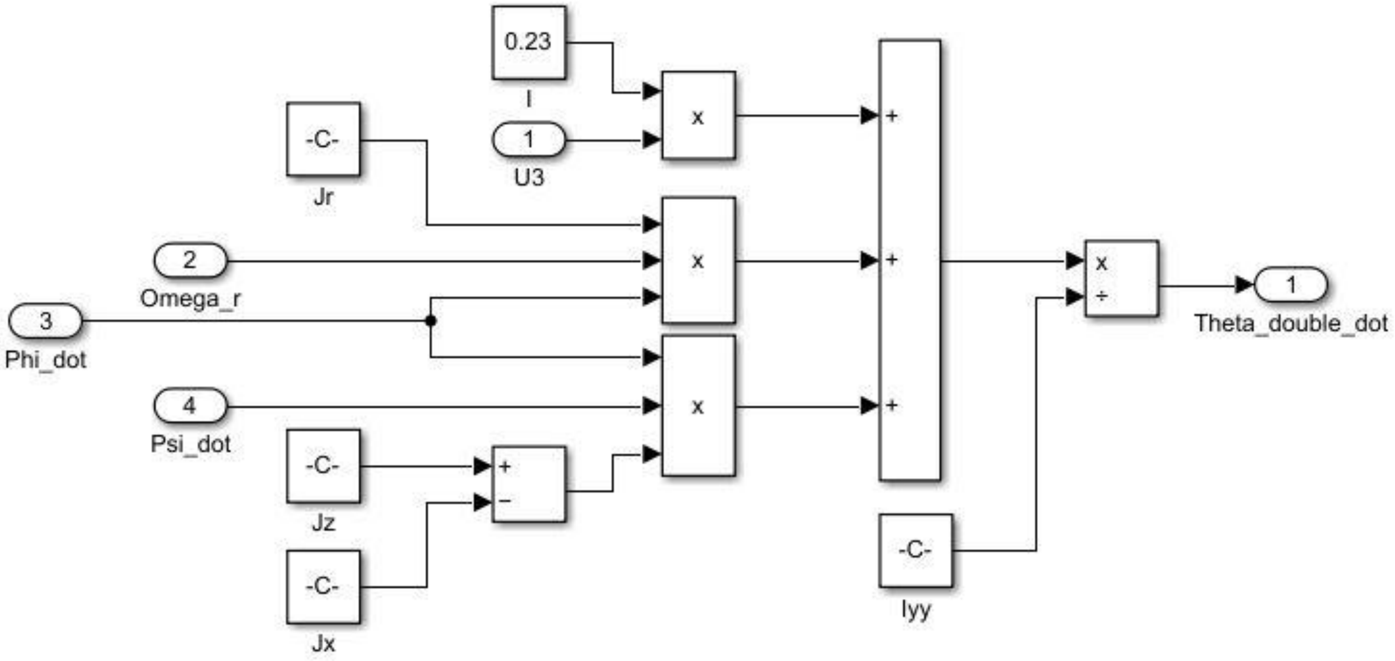


Fig. 13 Block diagram of the pitch angle acceleration subsystem

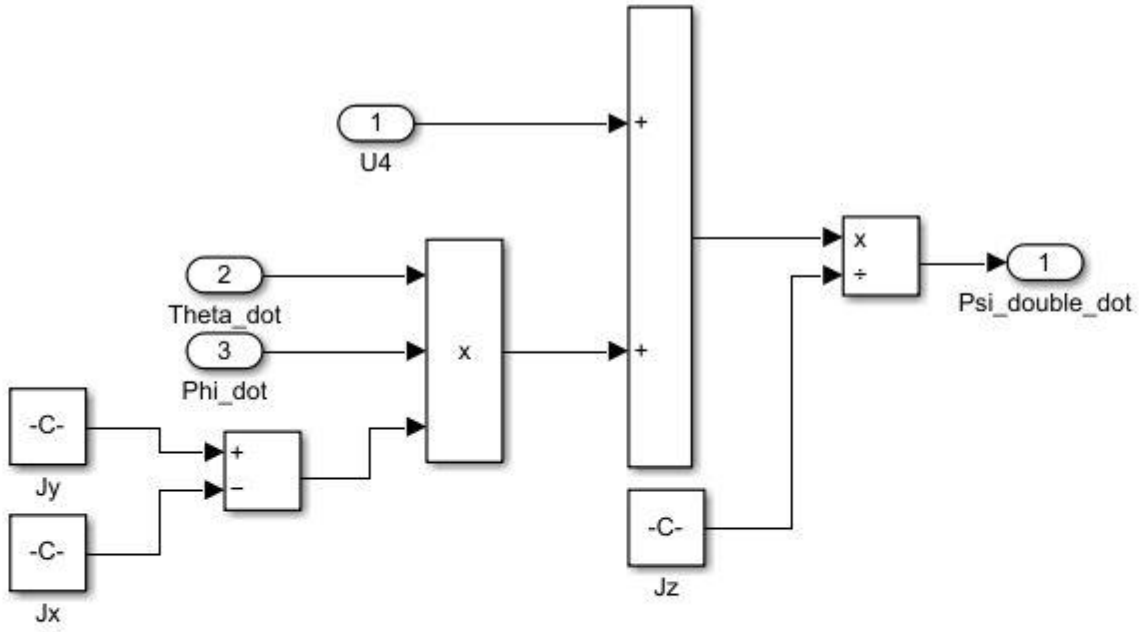


Fig. 14 Block diagram of the yaw angle acceleration subsystem

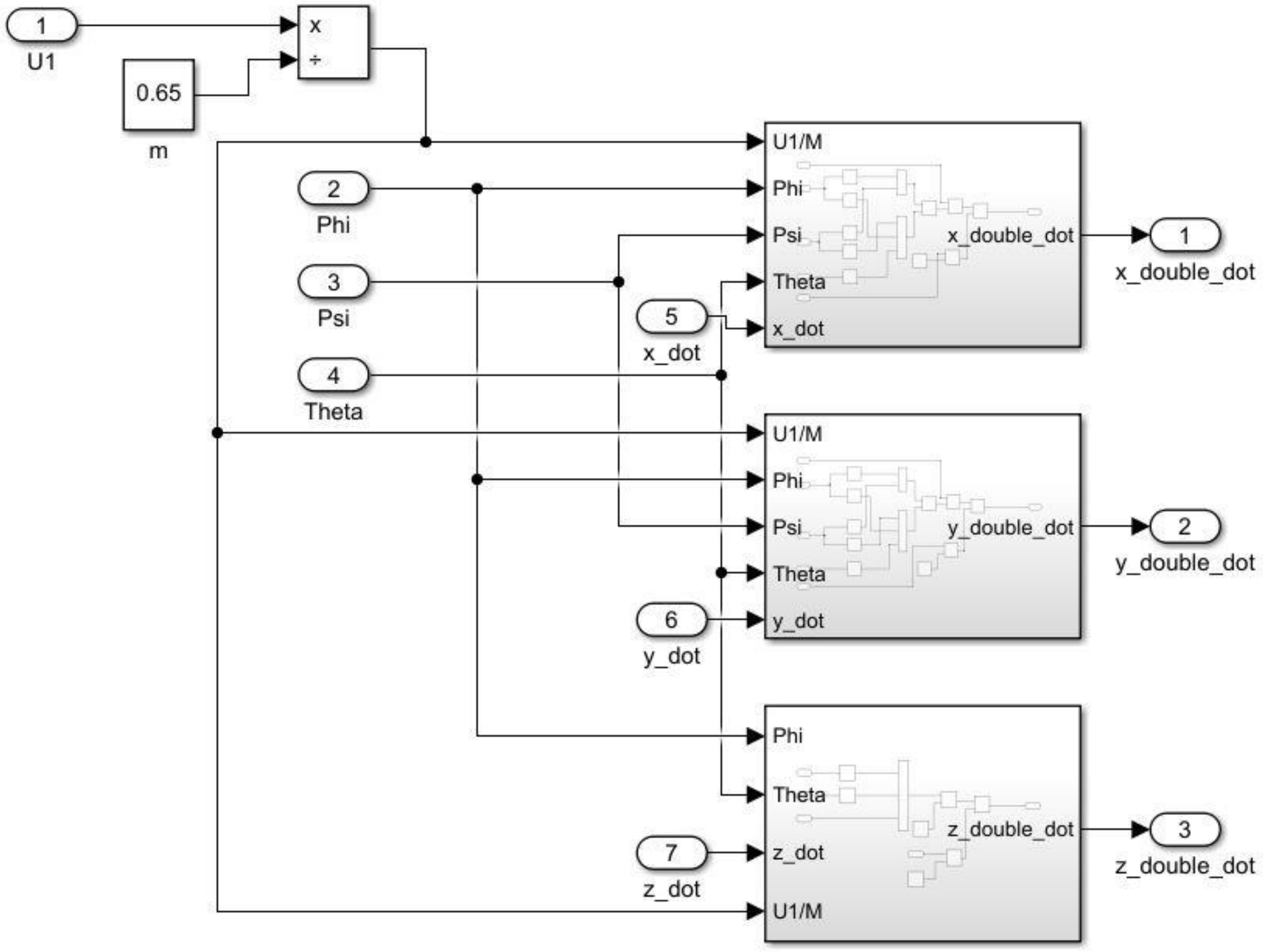


Fig. 15 Block diagram of the linear acceleration subsystem

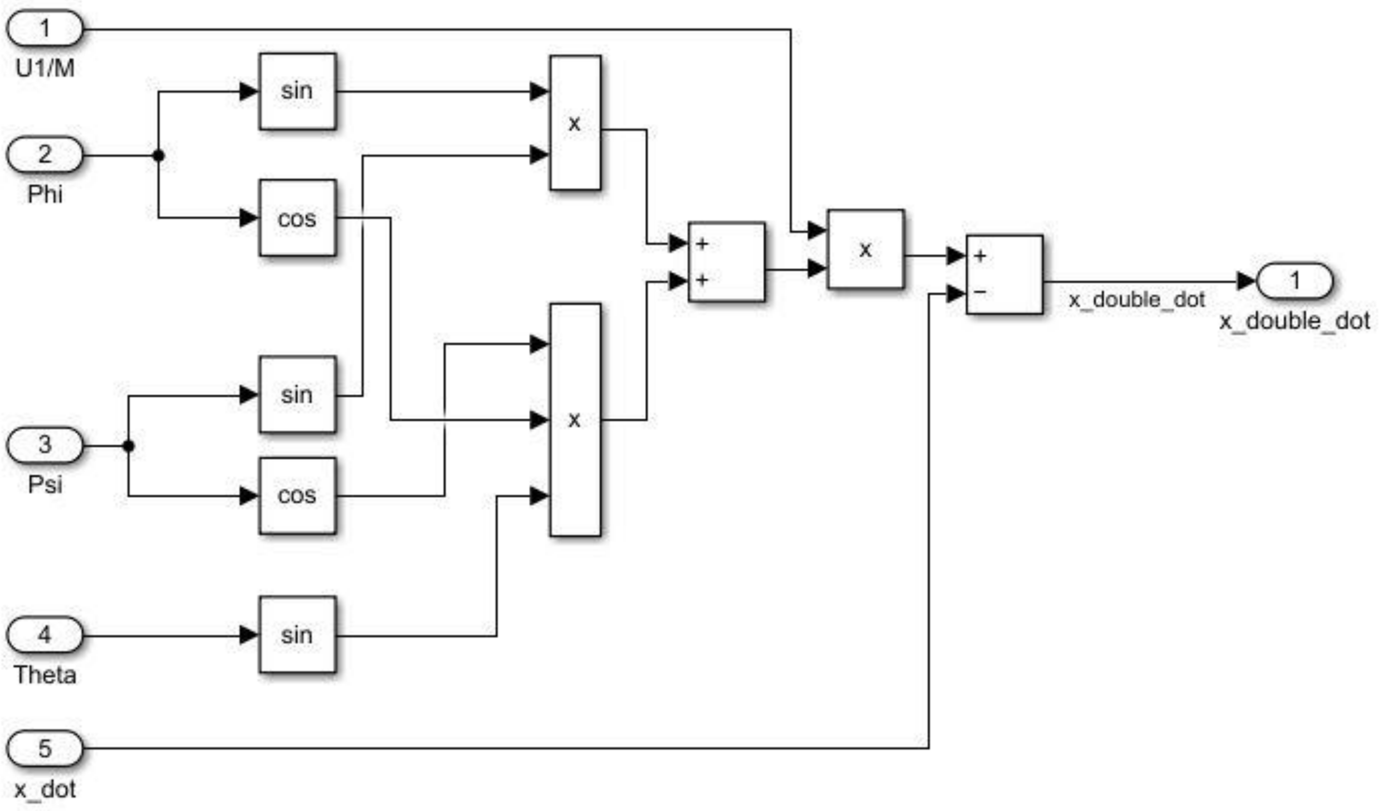


Fig. 16 Block diagram of the acceleration along x-axis subsystem

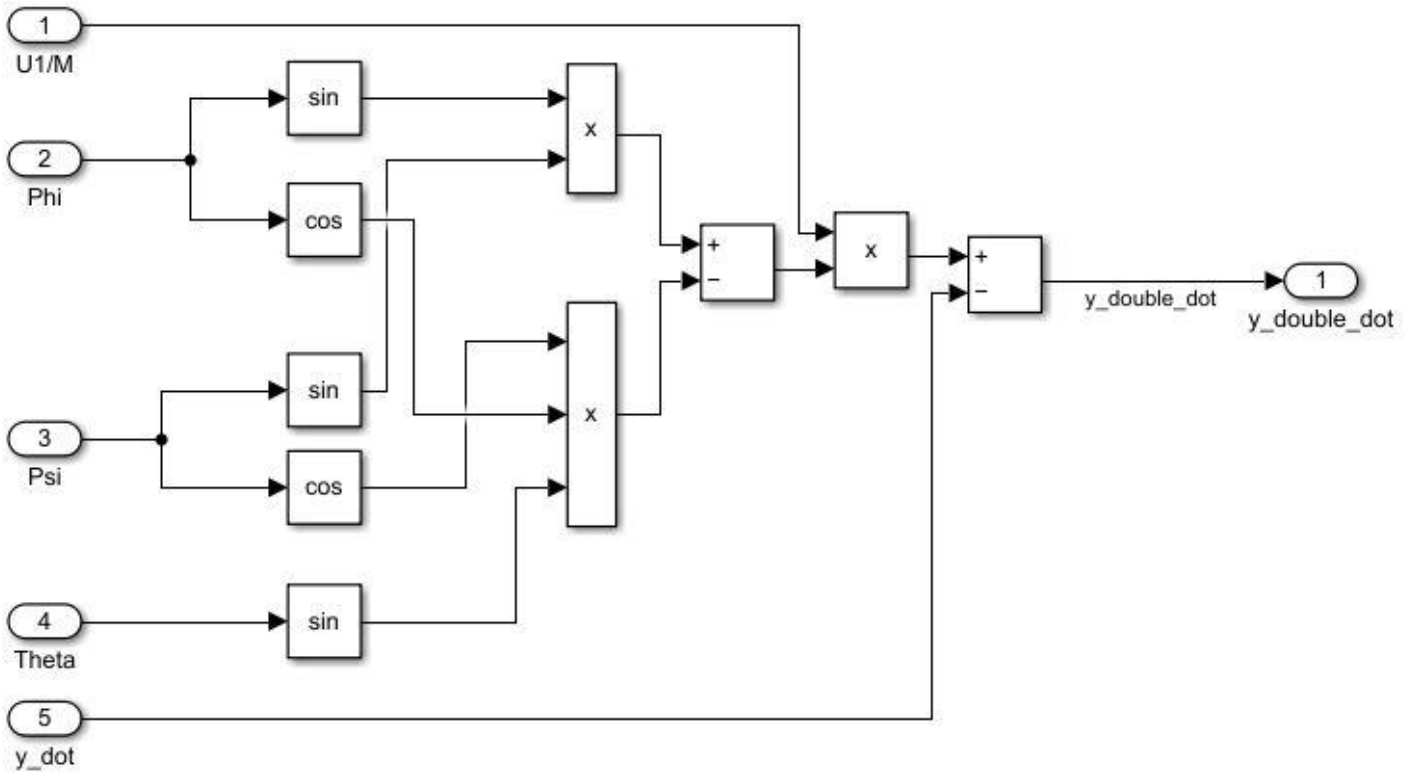


Fig. 17 Block diagram of the acceleration along y-axis subsystem

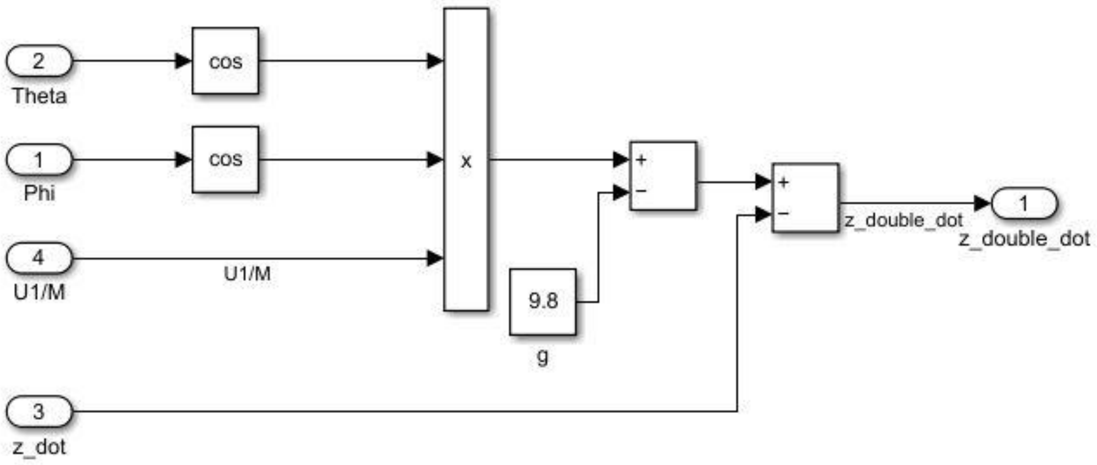
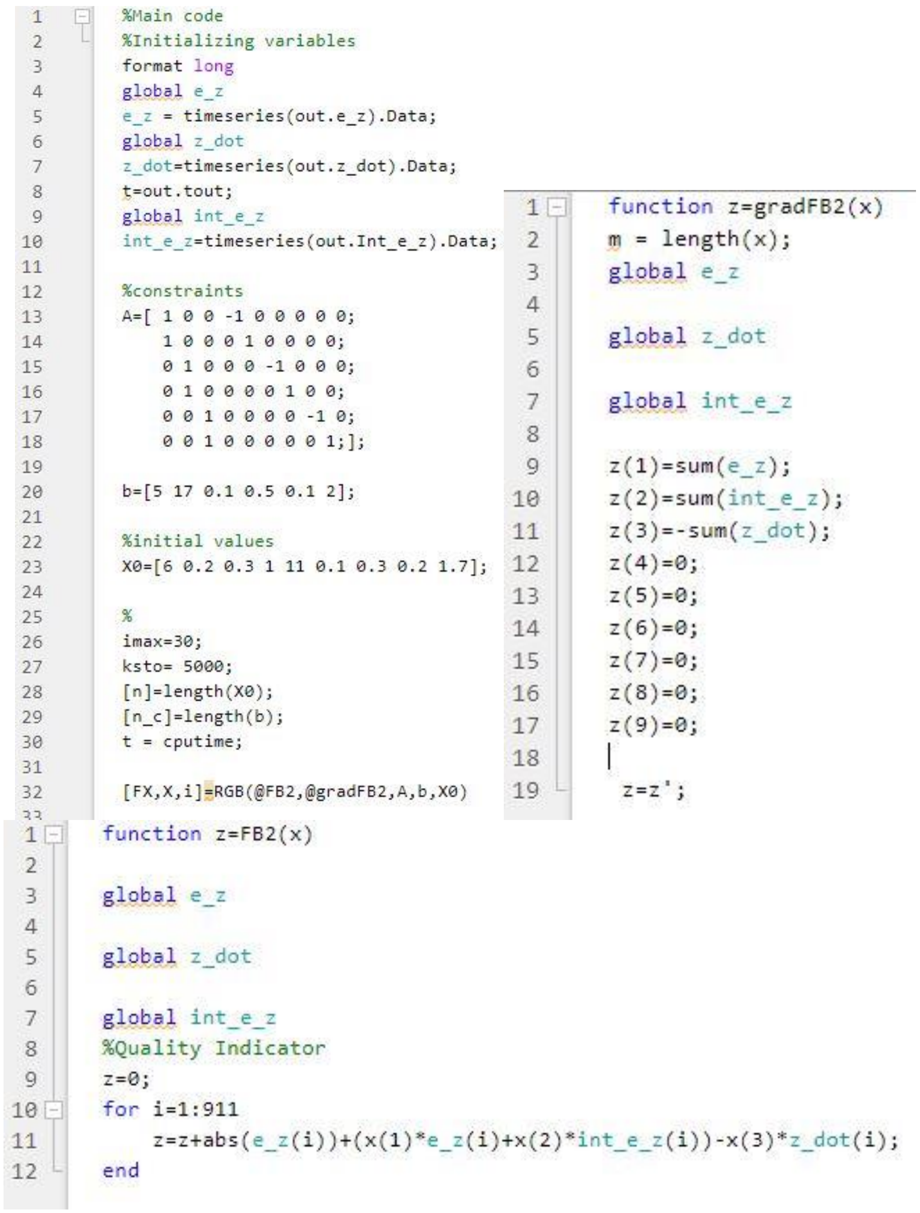


Fig. 18 Block diagram of the acceleration along the z-axis subsystem



```
%Main code
%Initializing variables
format long
global e_z
e_z = timeseries(out.e_z).Data;
global z_dot
z_dot=timeseries(out.z_dot).Data;
t=out.tout;
global int_e_z
int_e_z=timeseries(out.Int_e_z).Data;

%constraints
A=[ 1 0 0 -1 0 0 0 0 0;
    1 0 0 0 1 0 0 0 0;
    0 1 0 0 0 -1 0 0 0;
    0 1 0 0 0 0 1 0 0;
    0 0 1 0 0 0 0 -1 0;
    0 0 1 0 0 0 0 0 1;];

b=[5 17 0.1 0.5 0.1 2];

%initial values
X0=[6 0.2 0.3 1 11 0.1 0.3 0.2 1.7];

%
imax=30;
ksto= 5000;
[n]=length(X0);
[n_c]=length(b);
t = cputime;

[FX,X,i]=RGB(@FB2,@gradFB2,A,b,X0)
```

```
function z=gradFB2(x)
m = length(x);
global e_z

global z_dot

global int_e_z

z(1)=sum(e_z);
z(2)=sum(int_e_z);
z(3)=-sum(z_dot);
z(4)=0;
z(5)=0;
z(6)=0;
z(7)=0;
z(8)=0;
z(9)=0;

 z=z';
```

```
function z=FB2(x)

global e_z

global z_dot

global int_e_z
%Quality Indicator
z=0;
for i=1:911
    z=z+abs(e_z(i))+(x(1)*e_z(i)+x(2)*int_e_z(i))-x(3)*z_dot(i);
end
```

Fig. 19 MATLAB code for the reduced gradient optimization method

10 References

[1] L. Cedro och K. Wieczorkowski, ”Optimizing PID controller gains to model the performance of a quadcopter,” *Transportation Research Procedia*, vol. 40, pp. 156-169, 2019.

[2] S. Bouabdallah och R. Siegwart, ”Full Control of a Quadrotor,” i *Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007.

[3] S. Bouabdallah, ”Design and control of quadrotors with application to autonomous flying,” 2007.

[4] S. Bouabdallah, P. Murrieri och R. Siegwart, ”Design and Control of an Indoor Micro Quadrotor,” i *Conference on Robotics*, New Orleans, LA, USA, 2004.

[5] S. Bouabdallah och R. Siegwart, ”Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor,” i *International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[6] M. Usman, ”Quadcopter Modelling and Control With MATLAB/Simulink Implementation,” MATLAB/Simulink Implementation Technology Lappeenranta, 2020.

[7] D. Bertsekas, Nonlinear Programming Third Edition, 2016.

[8] D. G. L. a. Y. Ye, Linear and Nonlinear Programming Third edition, 2008.