Expert PID Algorithm for Autonomous Navigation of an Octocopter UAV

Pablo A. Tellez-Belkotosky[1] [0000-0002-1583-605X], Manuel A. Gutierrez Martinez[1] [0000-0003-1916-9634], Alejandro Jimenez-Flores[1] [0000-0002-7656-4203], Edmundo Javier Ollervides-Vazquez[1,2] [0000-0003-0022-3485], Francisco Jurado[2] [0000-0001-5568-0435], and Octavio Garcia-Salazar[1] [0000-0002-0445-1541]

[1] Aerospace Engineering Research and Innovation Center, Faculty of Mechanical and Electrical Enginnering, Autonomous University of Nuevo Leon, Apodaca NL, Mexico.
{pablo.tellezb,manuel.gutierrezmrt,alejandro.jimenezf, edmundo.ollervidesvz,octavio.garciasl}@uanl.edu.mx

[2] Tecnologico Nacional de Mexico/Instituto Tecnologico de La Laguna, Torreon, Coahuila, Mexico.
{ejollervidesv,fjuradoz}@lalaguna.tecnm.mx

Abstract. In this research work, the design of a PID flight controller with the rules of an expert system was applied to the navigation of an octocopter UAV. The Newton-Euler mathematical formulation is used to calculate the dynamic model of the octocopter UAV. The proposed set of decision rules for the Expert PID flight controller design allows the moment controllers to increase their response capability versus error variations. The simulation results of the navigation algorithm implemented in the octocopter UAV demonstrate that the translation Expert PID controller in interconnection with the attitude Expert PID controller can track a helical trajectory, fulfilling the objective of trajectory tracking as a validation for the proposed navigation algorithm.

Keywords: Expert PID · UAV · Octocopter.

1 Introduction

In recent years, applications such as agriculture, construction, civil operations, surveillance, and rescue missions have seen increasing demand, presenting new challenges. UAVs have proven to be highly useful in these areas; however, adapting to these evolving needs has led to the emergence of new challenges and requirements for UAV applications. New UAV configurations, such as the eight-rotor design, have been introduced, demonstrating greater stability and robustness against faults compared to the classic four-rotor configuration. For this reason, new controller techniques for octocopter UAVs have been proposed, seeking better response times when addressing faults.

There are some ways to set the configuration of an octocopter UAV. One combines a "+" way where the rotors turn counterclockwise; in the "x" way, the rotors are clockwise. With this configuration, the octocopter UAV has four rotors turning in one way and the other four rotors turning in the other way to stabilize the octocopter UAV when

https://doi.org/10.61728/AE20255039

all rotors are in optimal condition [1]. The authors of this work employed a PD control technique to manage attitude and altitude, utilizing a linearized dynamic model. For rotor control, they implemented a robust multicopter motor control approach that accounts for variations in battery voltage. The results were validated through simulations using the parameters of a quadrotor UAV. Additionally, a linear-quadratic regulator (LQR) was specifically designed for 3D trajectory tracking in the octocopter UAV, and the simulation was carried out by studying the altitude, attitude, and navigation controller for tracking position. The results were compared with the previous works mentioned above using the PD controller, concluding that the PD controller presents a fast response [2]. The octocopter is a UAV that presents the capability of playing with the general shape of the aircraft; the classic shape is symmetrical, but other authors, as in [3], changed the UAV to create a "V" shape where four motors are in the left side and other four motors are in the right side. The authors of this octocopter UAV named it a V8 configuration that used a Linear Quadratic Gaussian with an integral (LQG-I) control to stabilize the UAV attitude; this control was compared with a PID and an LQR controller to see the efficiency of energy usage. The authors in [4] presented an octocopter UAV with an asymmetric configuration, where two different arm distances were selected for the rotor in each arm; this design intends to improve the solution using multiple sensors or cameras. The authors used a PID controller with a Grey Wolf Optimizer (GWO) for parameters. They concluded that this octocopter UAV has room to improve due to the experiment not being better than a symmetric octocopter UAV for specific tasks. In [5], an attitude estimation algorithm was designed and implemented on an IMU hardware board. Two attitude controllers were developed and analyzed. The first controller employed a PID approach, which is commonly used for maintaining stable flight conditions in multicopter UAVs. The second control method employed utilized an L1 adaptive control framework. Simulation results suggest that this approach delivers excellent performance, particularly in handling modeling errors and rejecting disturbances, when compared to the PID controller. However, the L1 adaptive controller, being more complex, was more challenging to implement and showed limited noise attenuation. In contrast, the PID controller was successfully deployed on real-time hardware, where experimental flight tests delivered positive outcomes.

A star-shaped octocopter UAV configuration is considered in [6], where an active fault tolerant controller was implemented on such UAV using dynamic control allocation performing a 3-D path. Control allocation intends to use the available UAV actuators to achieve the controls. In [7], the application of fault-tolerant control strategies for an octocopter UAV was discussed. Specifically, it focuses on passive and active fault-tolerant control methods using Second Order Sliding Mode Control (SOSMC). Passive fault tolerance involves designing the system to inherently tolerate faults, while active fault tolerance involves using control strategies to mitigate the effects of faults. A nonlinear optimal control approach for the model of an octocopter UAV was proposed in [8]. They designed a control system that optimally manages the octocopter UAV dynamics in a nonlinear environment. Nonlinear optimal control typically involves sophisticated mathematical techniques to find the best control inputs that optimize a specific performance criterion.

In this work, an Expert PID flight navigation controller that designates the PID strength according to the octocopter UAV error value, each axis and angle is presented. Five main rules are designed by analyzing the actual and previous error values. Simulation results are presented, illustrating the behavior of each axis and angle of the octocopter UAV.

To facilitate a better understanding of this research, the paper has been organized as follows: The equations of motion for the octocopter UAV are presented in Section 2. The proposed Expert PID controller is introduced in Section 3. The control system architecture is described in Section 4, and the simulation results are discussed in Section 5. Finally, a summary of the conclusions and key findings is provided in Section 6.

2 Equations of motion for an octocopter UAV

The mathematical representation of the octocopter UAV dynamics utilizes two reference frames. The first is the body frame, which is attached to the structure of the octocopter UAV and aligned with its center of mass, represented as $B=\{x_b, y_b, z_b\}$. The second is the inertial frame, fixed to the ground, represented as $I=\{x_i, y_i, z_i\}$, as illustrated in Fig. 1. The equations of motion are derived using the Newton-Euler formulation, which serves as a compact mathematical framework for describing the dynamics of the octocopter UAV [10], [11]. Definitions of parameters are shown as follows:

- ξ : Vector coordinate position relative to the inertial frame, $[x\ y\ z]^\top$.
- V : Translational velocity vector expressed in the inertial reference frame, $\left[v_x\ v_y\ v_z\right]^\top$.
- R : Orthogonal rotation matrix in $SO(3)$.
- T_T : Total force vector that contains the total thrust in the direction of gravity $[0\ 0\ f_T]^\top$.
- f_T : Summatory of actuated forces produce by each rotor f_k of the octocopter UAV.
- ω_k : Rotation speed of the k-th rotor in rad/s.
- m : Total mass of the octocopter UAV.
- g : Gravity force in the inertial frame.
- e_3 : Aligned in the direction of the center of Earth.
- η : Vector that describes the rotation coordinates for the octocopter in the inertial frame, $[\phi\ \theta\ \psi]^\top$.
- E : Euler matrix to convert angular velocity in the body frame to the inertial frame.
- Ω : Angular velocity vector defined in the body-fixed reference frame, $[p\ q\ r]^\top$.
- J : Matrix 3×3 that contain the inertial moments in diagonal form J_{xx}, J_{yy}, and J_{zz}.
- τ_a : Moment vector due the actuators, $\left[\tau_\phi\ \tau_\theta\ \tau_\psi\right]^\top$.
- τ_ϕ : Roll moment affected by the rotors ω_3, ω_4, ω_7, and ω_8.
- τ_θ : Pitch moment affected by the rotors ω_1, ω_2, ω_5, and ω_6.
- τ_ψ : Yaw moment affected by the drag of all rotors and their direction of rotation.
- d : Distance of each arm from the center of mass to the rotation axis of the rotor.
- c_T : Constant value representing the propeller thrust factor.
- c_Q : Constant value representing the drag coefficient of the propeller.

$$\dot{\xi} = V \tag{1}$$

$$m\dot{V} = R(-T_T) + mge_3 \tag{2}$$

$$\dot{\eta} = E\Omega \tag{3}$$

$$J\dot{\Omega} = -\Omega \times J\Omega + \tau_a \tag{4}$$

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{5}$$

$$E = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \tag{6}$$

The complete dynamic model of the UAV is represented as follows, where c_a, s_a, t_a with $a = \phi, \theta, \psi$ denote the cosine, sine, and tangent functions of the Euler angles, respectively.

$$\begin{aligned} \dot{x} &= v_x & \dot{\phi} &= \frac{c_\theta p + s_\phi s_\theta q + c_\phi s_\theta r}{c_\theta} \\ \dot{y} &= v_y & \dot{\theta} &= c_\phi q - s_\phi r \\ \dot{z} &= v_z & \dot{\psi} &= \frac{s_\phi q + c_\phi r}{c_\theta} \\ \dot{v_x} &= -\frac{f_T(s_\phi s_\psi + c_\phi s_\theta c_\psi)}{m} & \dot{p} &= \frac{(J_{yy} - J_{zz})qr + \tau_\phi}{J_{xx}} \\ \dot{v_y} &= -\frac{f_T(c_\phi s_\theta s_\psi - s_\phi c_\psi)}{m} & \dot{q} &= \frac{(J_{zz} - J_{xx})pr + \tau_\theta}{J_{yy}} \\ \dot{v_z} &= -\frac{f_T c_\phi c_\theta + mg}{m} & \dot{r} &= \frac{(J_{xx} - J_{yy})pq + \tau_\psi}{J_{zz}} \end{aligned} \tag{7}$$

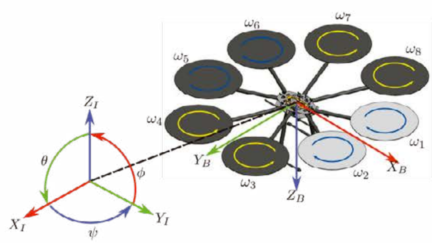

Fig. 1: Coordinate systems and rotational directions of each rotor in the octocopter UAV.

The total thrust force vector of the octocopter UAV propulsion (f_T) is calculated by summing all the individual thrust forces produced by the eight rotors ($f_1, \cdots, f_8$) of the octocopter UAV, as shown below.

$$f_T = \sum_{k=1}^{8} f_k = c_T \sum_{k=1}^{8} \omega_k^2, \quad \text{with} \quad f_k = c_T \omega_k^2 \tag{8}$$

The total thrust force vector T_T is described in the body frame of the octocopter UAV as:

$$T_T = \begin{bmatrix} 0 \\ 0 \\ f_T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{k=1}^{8} f_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c_T \sum_{k=1}^{8} \omega_k^2 \end{bmatrix} \tag{9}$$

The variables $\omega_1, \cdots, \omega_8$ represent the rotor speeds, and variables $f_1 = c_T\omega_1^2$, $f_2 = c_T\omega_2^2, \cdots, f_8 = c_T\omega_8^2$ represent the lift forces of each one of the eight rotors.

The moments acting in the octocopter UAV are strictly related to the rotor speeds; depending on which rotors perform more speed than the other rotors, the octocopter UAV performs the moments that are indicated in the definitions at the begining of the section 2 in which rotors move each moment. The roll moment is influenced by rotors ω_7 and ω_8 operating at high speed, while rotors ω_3 and ω_4 perform with low speed, operate at low speed; this action produces a positive moment. To generate a negative moment, the speed of rotors ω_3 and ω_4 must be increased, and the speed of rotors ω_7 and ω_8 must be reduced. The positive pitch moment is actuated with the high speed of rotors ω_1 and ω_2 and the lower speed of rotors ω_5 and ω_6. The case of the positive yaw moment uses the direction of rotation speed; in this case, rotors ω_1, ω_2, ω_5, and ω_6 rotate in the clockwise direction, these rotors rotate with high speed in comparison of the other rotors that rotate in counterclockwise and with lower speed.

$$\begin{bmatrix} f_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T & c_T & c_T & c_T & c_T \\ 0 & 0 & -c_TP & -c_TP & 0 & 0 & c_TP & c_TP \\ c_TP & c_TP & 0 & 0 & -c_TP & -c_TP & 0 & 0 \\ c_Q & c_Q & -c_Q & -c_Q & c_Q & c_Q & -c_Q & -c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \\ \omega_5^2 \\ \omega_6^2 \\ \omega_7^2 \\ \omega_8^2 \end{bmatrix} \tag{10}$$

3 Expert PID controller design

The Expert PID control is to design PID parameters with plant characteristics and the control expert experience, with no need for modeling information. The experience is mainly based on the system error response. At time t, the desired value is considered as $r_d(t)$, the measurement value as $y_o(t)$, and then the tracking error is $e_c(t) = r_d(t) - y_o(t)$ at time t, and $e_c(t-1)$ and $e_c(t-2)$, represent the error at time $t-1$ and $t-2$, respectively [9]. The parameters definitions for the Expert PID controller are shown as follows:

- u_c : Input control that change depending of the specific rule with $c = \phi, \theta, \psi, x, y, z$ referring to whose control is applied.
- M_{1c}, M_{2c} : Setting the limit values of the errors in rules.
- C_{nc} : Constant value to increase the PID controller to a maximum value where n is a designed number.
- k_{pc}, k_{ic}, k_{dc} : Gains of the PID controller for the corresponding c parameter.
- $e_c(t)$: Proportional error at time t and $t-1$, respectively.
- $\Delta e_c(t), \Delta e_c(t-1)$: Differential error at time t and $t-1$, respectively.
- ts : Time sample.
- e_{ic} : Integrative error.
- $k_{1c}, \ldots, k_{5c}$: Gains setting the strength of the PID controller.

3.1 Rules for Expert PID controller

The Expert PID rules are designed for each controller of the octocopter UAV. These rules are given as follows:

Rule 1 When $|e_c(t)| > M_{1c}$, the strongest PID controller is used to minimize error quickly.

$$u_c(t) = C_{nc}(k_{pc}e_c(t) + k_{dc}\frac{\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t)) \quad (11)$$

Rule 2 When $e_c(t)\Delta e_c(t) > 0$ or $\Delta e_c(t) = 0$, two conditions are considered as follow: A stronger PID controller is used if $|e_c(t)| \geq M_{2c}$.

$$u_c(t) = u_c(t-1) + k_{1c}(k_{pc}e_c(t) + k_{dc}\frac{\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t)) \quad (12)$$

A weak PID controller is used if $|e_c(t)| < M_{2c}$.

$$u_c(t) = u_c(t-1) + k_{2c}(k_{pc}e_c(t) + k_{dc}\frac{\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t)) \quad (13)$$

Rule 3 When $e_c(t)\Delta e_c(t) < 0$, $\Delta e_c(t)\Delta e_c(t-1) > 0$, or $e_c(t) = 0$, which indicates the absolute of error value tends to smaller or constant value, the control input can hold.

$$u_c(t) = u_c(t) \quad (14)$$

Rule 4 When $e_c(t)\Delta e_C(t) < 0$, and $\Delta e_c(t)\Delta e_c(t-1) < 0$, which indicates the error value is in extremism state. Two conditions are considered as follows:
A strong PID controller is used if the absolute error value is big, $|e_c(t)| \geq M_{2c}$.

$$u_c(t) = u_c(t-1) + k_{3c}(k_{pc}e_c(t) + k_{dc}\frac{\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t)) \quad (15)$$

A weaker PID controller is used if the absolute error value is small, $|e_C(t)| < M_{2c}$.

$$u_c(t) = u_c(t-1) + k_{4c}(k_{pc}e_c(t) + k_{dc}\frac{\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t)) \quad (16)$$

Rule 5 When $|e(t)| \leq \varepsilon_c$ indicates that the error magnitude approaches a very small value, a significantly weaker PID control strategy is used for reducing the static error.

$$u_c(t) = k_{5c}(k_{pc}e_c(t) + k_{dc}\frac{\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t)) \quad (17)$$

Where $\Delta e_c(t) = e_c(t) - e_c(t-1)$, $\Delta e_c(t-1) = e_c(t-1) - e_c(t-2)$, and $e_{ic}(t) = e_{ic}(t) + e_c(t)ts$. Also, $k_{1c} > k_{3c} > k_{2c} > k_{4c} > k_{5c}$ are the gains that set the strength of the PID controller depending on the rule.

4 Flight controller design

4.1 Mixer rotor controller

The control vector $U = [u_1, \ldots, u_8]^\top = [\omega_1^2, \ldots, \omega_8^2]^\top$ is utilized to generate the velocity, which depends on the total thrust force and the aerodynamic moments acting on the octocopter UAV. Also, the parameters definitions of the flight controller are shown as follows:

- U : Input controller vector containing each rotor quadratic speed of the octocopter UAV.
- C_A : aerodynamic thrust and drag coefficients matrix of the octocopter UAV.
- Γ : vector containing the total thrust and momentos $[f_T,\ \tau_\phi,\ \tau_\theta,\ \tau_\psi]^\top$.
- u_k^{Sat} : input controller of the quadratic kth rotor speed saturated.
- $u_{k_{max}}$: maximum quadratic kth rotor speed saturated.
- $u_{k_{min}}$: minimum quadratic kth rotor speed saturated.
- F_{Tc} : Total thrust control associated with the altitude PID controller.
- ϕ_d^{Sat}, θ_d^{Sat} : roll and pitch angles constrained in the range between $-\pi/2\ rad$ and $\pi/2\ rad$.
- u_z : altitude PID controller.
- u_x, u_y : navigation PID controller.
- u_a : moment PID controllers, $[u_\phi,\ u_\theta,\ u_\psi]^\top$.

The simplified form of equation (10) is expressed as:

$$\Gamma = C_A U \tag{18}$$

Due to the fact that the C_A matrix is not square, a pseudo-inverse equation is used to obtain the inverse matrix, as shown below.

$$C_A^{-1} = C_A^\top (C_A C_A^\top)^{-1} \tag{19}$$

This allows for obtaining the vector of the desired control variables corresponding to each rotor. The developed matrix expression is provided as follows:

$$U = C_A^{-1}\Gamma \tag{20}$$

Obtaining the vector of the desired controller variables corresponds to each rotor. Below is such a developed matrix expression:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{bmatrix} = \begin{bmatrix} \frac{1}{8c_T P} & 0 & \frac{1}{4c_T P} & \frac{1}{8c_Q} \\ \frac{1}{8c_T P} & 0 & \frac{1}{4c_T P} & \frac{1}{8c_Q} \\ \frac{1}{8c_T P} & \frac{-1}{4c_T P} & 0 & \frac{-1}{8c_Q} \\ \frac{1}{8c_T P} & \frac{-1}{4c_T P} & 0 & \frac{-1}{8c_Q} \\ \frac{1}{8c_T P} & 0 & \frac{-1}{4c_T P} & \frac{1}{8c_Q} \\ \frac{1}{8c_T P} & 0 & \frac{-1}{4c_T P} & \frac{1}{8c_Q} \\ \frac{1}{8c_T P} & \frac{1}{4c_T P} & 0 & \frac{-1}{8c_Q} \\ \frac{1}{8c_T P} & \frac{1}{4c_T P} & 0 & \frac{-1}{8c_Q} \end{bmatrix} \begin{bmatrix} f_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{21}$$

Thus, the required control strategies are determined for the individual rotors in the octocopter UAV. A saturation function will be introduced to establish the operational limits of the rotors, including their minimum and maximum rotation speeds. The saturation functions consider the practical limits of the motor angular velocity. This specific saturation function is defined as:

$$u_k^{\text{Sat}} = \text{Sat}(u_k) = \begin{cases} u_{k_{max}} \text{ if } u_k \geq u_{k_{max}} \\ u_k \text{ if } u_{k_{min}} < u_k < u_{k_{max}} \\ u_{k_{min}} \text{ if } u_k \leq u_{k_{min}} \end{cases} \tag{22}$$

4.2 Altitude controller

The total thrust aligned with Z_B, is not necessarily directed along the inertial frame e_3. The thrust control is connected to the altitude controller and is calculated based on the roll and pitch angles. The dynamics of the desired thrust force f_T is proposed as F_{Tc} by the actuators, with the aim of ensuring that the stationary or ascending flight regime is maintained, leading to the following formulation:

$$F_{Tc} = \frac{1}{\cos(\theta_d^{\text{Sat}})\cos(\phi_d^{\text{Sat}})}(mg - mu_z) \tag{23}$$

where ϕ_d^{Sat} and θ_d^{Sat} are constrained to avoid the singularity presented at $\pm 90(deg)$, and the thrust control F_{Tc} is reached by an altitude Expert PID controller, such that this controller is defined as

$$u_z = k_{pz}e_z(t) + k_{dz}\frac{\Delta e_z(t)}{ts} + k_{iz}e_{iz}(t) \tag{24}$$

where $e_z = z_d - z$ z is defined as the altitude error, z_d is identified as the desired altitude, and z is referred to as the current altitude. To enhance trajectory tracking, a saturation function is applied to the desired thrust force. The saturated thrust controller is formulated as follows:

$$F_{Tc}^{\text{sat}} = \text{Sat}(F_{Tc}) = \begin{cases} F_{T_{max}} \text{ if } F_{Tc} \geq F_{T_{max}} \\ F_{Tc} \text{ if } F_{T_{min}} < F_{Tc} < F_{T_{max}} \\ F_{T_{min}} \text{ if } z \leq F_{T_{min}} \end{cases} \tag{25}$$

4.3 Attitude controller

Defining a controller vector $u_a = [u_\phi,\ u_\theta,\ u_\psi]^\top$ is the main focus of the control problem. To manage the octocopter UAV attitude, a specialized PID control strategy is implemented. Based on the definition established in vector (10), the external moments are reformulated and expressed according to the proposed control law, as detailed here:

$$\begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} = \begin{bmatrix} k_{p\phi}e_\phi(t) + k_{d\phi}\frac{\Delta e_\phi(t)}{ts} + k_{i\phi}e_{i\phi}(t) \\ k_{p\theta}e_\theta(t) + k_{d\theta}\frac{\Delta e_\theta(t)}{ts} + k_{i\theta}e_{i\theta}(t) \\ k_{p\psi}e_\psi(t) + k_{d\psi}\frac{\Delta e_\psi(t)}{ts} + k_{i\psi}e_{i\psi}(t) \end{bmatrix} \tag{26}$$

Then u_ϕ, u_θ, and u_ψ represent the Expert PID controller law for managing the octocopter UAV attitude. The trajectory tracking of the attitude control system is enhanced by introducing saturation to the desired moments. The expression for the saturated moment controllers is presented below.

$$u_a^{\mathrm{Sat}} = \mathrm{Sat}(u_a) = \begin{cases} u_{a_{max}} \text{ if } u_a \geq u_{a_{max}} \\ u_a \text{ if } u_{a_{min}} < u_a < u_{a_{max}} \\ u_{a_{min}} \text{ if } \eta \leq u_{a_{min}} \end{cases} \tag{27}$$

where $u_a^{\mathrm{Sat}} = [u_\phi^{\mathrm{Sat}},\ u_\theta^{\mathrm{Sat}},\ u_\psi^{\mathrm{Sat}}]^\top$. With attitude and altitude controllers, the next step is to define the navigation controller.

4.4 Navigation controller

To achieve effective flight control, the attitude control system is incorporated within the octocopter UAV, ensuring that the desired setpoints of the attitude vector η_d are calculated based on the translational position controller functioning over the $X-Y$ plane, while utilizing the reference rotation matrix $R(\psi) \in SO(2)$. An Expert PID control strategy is proposed to compute the desired forces for the translational position controller in the $X-Y$ plane.

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} k_{px}e_x(t) + k_{dx}\frac{\Delta e_x(t)}{ts} + k_{ix}e_{ix}(t) \\ k_{py}e_y(t) + k_{dy}\frac{\Delta e_y(t)}{ts} + k_{iy}e_{iy}(t) \end{bmatrix} \tag{28}$$

Position errors are defined as $e_x = x_d - x$ and $e_y = y_d - y$ corresponding to e_1 and e_2, respectively, with x_d and y_d representing the target positions along these directions. The proposed Expert PID-type navigation controller is connected to the external control layer through the rotation matrix $R(\psi) \in SO(2)$, aiming to generate the desired reference angles θ_d and ϕ_d for the attitude controller. This setup establishes the relationship between the attitude and translational controllers, as detailed below.

$$\begin{bmatrix} \phi_d^{\mathrm{Sat}} \\ \theta_d^{\mathrm{Sat}} \end{bmatrix} = \begin{bmatrix} sin(\psi) & -cos(\psi) \\ cos(\psi) & sin(\psi) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{29}$$

Equation (29) outlines the proposed navigation controller for the octocopter UAV. The trajectories are tracked within the plane defined by the X and Y axes, and the complete control scheme and system dynamics are shown in Fig. 2.

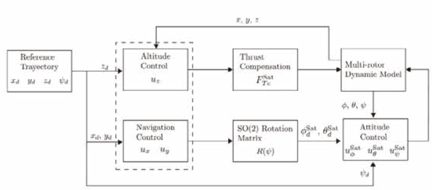

Fig. 2: Diagram of controller and dynamics interactions.

5 Simulation of Flight Control Systems

The rules of the Expert PID are used as follows to define which strength PID controller is being used according to the condition of the rules.

Algorithm 1 Expert PID algorithm

Rule 1
if $|e_c(t)| > M_{1c}$ **then**
  $u_c(t) = C_{1c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
else
  if $|e_c(t)| > 0.7M_{1c}$ **then**
    $u_c(t) = C_{2c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
  end if
else
  if $|e_c(t)| > 0.5M_{1c}$ **then**
    $u_c(t) = C_{3c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
  end if
else
  if $|e_c(t)| > 0.2M_{1c}$ **then**
    $u_c(t) = C_{4c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
  end if
end if
Rule 2
if $e_c(t)\Delta e_c(t) > 0$ **or** $\Delta e_c(t) = 0$ **then**
  if $|e_c(t)| \geq M_{2c}$ **then**
    $u_c(t) = u_c(t-1) + k_{1c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
  else
    $u_c(t) = u_c(t-1) + k_{2c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
  end if
end if
Rule 3
if $e_c(t)\Delta e_c(t) < 0$ **and** $\Delta e_c(t)\Delta e_c(t-1) > 0$ **or** $e_c(t) = 0$ **then**
  $u_c(t) = u_c(t)$
end if
Rule 4
if $e_c(t)\Delta e_c(t) < 0$ **and** $\Delta e_c(t)\Delta e_c(t-1) < 0$ **then**
  if $|e_c(t)| \geq M_{2c}$ **then**
    $u_c(t) = u_c(t-1) + k_{3c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
  else
    $u_c(t) = u_c(t-1) + k_{4c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
  end if
end if
Rule 5
if $|e_c(t)| \leq \varepsilon_c$ **then**
  $u_c(t) = k_{5c}(k_{pc}e_c(t) + \frac{k_{dc}\Delta e_c(t)}{ts} + k_{ic}e_{ic}(t))$
end if

5.1 Parameters of the octocopter UAV controller

Numerical simulations utilize parameters obtained from the octocopter UAV model developed in SolidWorks. Table 1 lists the numerical values associated with the model parameters. To enable trajectory tracking, the gains of the saturated Expert PID controller in the attitude control layer are tuned to match the desired sinusoidal inputs for each input of the octocopter UAV. The adjusted gains for the saturated Expert PID controller in the inner attitude layer are presented in Table 2

To tune the parameters of the navigation controller, the altitude controller is implemented as the initial step in the tuning process. The position control in the X-Y plane is then adjusted, applying setpoint references to manage the translational movement along

Table 1: Newton-Euler aircraft model parameters.

Parameter	Value	Parameter	Value	Parameter	Value
c_T	8.33×10^{-5} $(N-s^2)$	J_{xx}	83×10^{-2} $(kg-m^2)$	m	8 (kg)
c_Q	2.42×10^{-6} $(N-m-s^2)$	J_{yy}	83×10^{-2} $(kg-m^2)$	g	9.81 (m/s^2)
d	0.721 (m)	J_{zz}	165×10^{-2} $[kg-m^2]$		

Table 2: Attitude controller gains for Expert PID.

Gain	Value	Gain	Value	Gain	Value	Gain	Value	Gain	Value
$k_{p\phi}$	80	$k_{d\psi}$	6	ε_θ	$\frac{\pi}{600}$ (rad)	$C_{4\phi}$	60	$k_{3\theta}$	30
$k_{d\phi}$	4	$k_{i\psi}$	6	$C_{1\phi}$	120	$C_{4\theta}$	60	$k_{4\phi}$	8
$k_{i\phi}$	20	$M_{1\phi}$	$\frac{\pi}{18}$ (rad)	$C_{1\theta}$	120	$k_{1\phi}$	50	$k_{4\theta}$	8
$k_{p\theta}$	80	$M_{2\phi}$	$\frac{\pi}{180}$ (rad)	$C_{2\phi}$	100	$k_{1\theta}$	50	$k_{5\phi}$	7
$k_{d\theta}$	4	$M_{1\theta}$	$\frac{\pi}{18}$ (rad)	$C_{2\theta}$	100	$k_{2\phi}$	10	$k_{5\theta}$	7
$k_{i\theta}$	20	$M_{2\theta}$	$\frac{\pi}{180}$ (rad)	$C_{3\phi}$	80	$k_{2\theta}$	10		
$k_{p\psi}$	7	ε_ϕ	$\frac{\pi}{600}$ (rad)	$C_{3\theta}$	80	$k_{3\phi}$	30		

the X and Y axes. Table 3. presents the adjusted gains for the Expert PID navigation controller. Circular trajectory tracking is achieved by defining the position references xd and yd through the following equations.

Table 3: Navigation controller gains for Expert PID.

Gain	Value	Gain	Value	Gain	Value	Gain	Value	Gain	Value
k_{px}	10	M_{1x}	1 (m)	C_{1x}	12	C_{2z}	100	k_{2y}	1
k_{dx}	0.2	M_{2x}	0.4 (m)	C_{2x}	10	C_{3z}	80	k_{3y}	3
k_{ix}	100	M_{1y}	1 (m)	C_{3x}	8	C_{4z}	60	k_{4y}	0.8
k_{py}	10	M_{2y}	0.4 (m)	C_{4x}	6	k_{1x}	5	k_{5y}	0.7
k_{dy}	0.2	M_{1z}	10 (m)	C_{1y}	12	k_{2x}	1	k_{1z}	50
k_{iy}	100	M_{2z}	1 (m)	C_{2y}	10	k_{3x}	3	k_{2z}	10
k_{pz}	60	ε_x	0.2 (m)	C_{3y}	8	k_{4x}	0.8	k_{3z}	30
k_{dz}	3	ε_y	0.2 (m)	C_{4y}	6	k_{5x}	0.7	k_{4z}	8
k_{iz}	10	ε_z	0.3 (m)	C_{1z}	120	k_{1y}	5	k_{5z}	7

$$\begin{aligned} x_d &= A_x \sin(2\pi f t) \\ y_d &= A_y \cos(2\pi f t) \end{aligned} \tag{30}$$

where A_x and A_y represent the radius of the cylinder, f is the frequency, and t is the time. For the cylindrical trajectory, A_x and A_y were set to 3 meters. The saturation values are provided in the table below.

Table 4: Saturator values.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$u_{k_{min}}$	100 (RPM)	$\phi_{d_{min}}$	$-\frac{\pi}{10}$ (rad)	$u_{\phi_{min}}$	-20 $(N \cdot m)$	$u_{\psi_{min}}$	-13 $(N \cdot m)$
$u_{k_{max}}$	3700 (RPM)	$\phi_{d_{max}}$	$\frac{\pi}{10}$ (rad)	$u_{\phi_{max}}$	20 $(N \cdot m)$	$u_{\psi_{max}}$	13 $(N \cdot m)$
$F_{T_{min}}$	0 (N)	$\theta_{d_{min}}$	$-\frac{\pi}{10}$ (rad)	$u_{\theta_{min}}$	-20 $(N \cdot m)$		
$F_{T_{max}}$	156 (N)	$\theta_{d_{max}}$	$\frac{\pi}{10}$ (rad)	$u_{\theta_{max}}$	20 $(N \cdot m)$		

5.2 Navigation and Trajectory Tracking

MATLAB/Simulink was used to generate the simulation results. At first, the signals corresponding to the navigation trajectory are presented. The octocopter UAV ascends while following a circular path, creating a helical pattern. Figure 3 illustrates the desired and actual trajectory tracking of the octocopter UAV, forming a geometrically helical trajectory. Additionally, the total thrust compensation is depicted. Finally, the control moments of the octocopter UAV are shown.

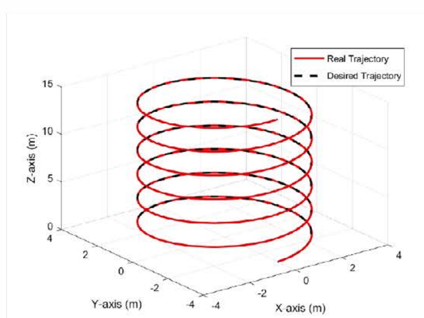

Fig. 3: Trajectory tracking of the octocopter UAV.

Figure 4 presents the response to a sinusoidal wave due to the circling behavior of the helical trajectory of the octocopter UAV; a negative peak of -0.49 (deg) is presented at 0.2 (s) where the amplitude of the cosine is ± 0.2 (deg). Figure 5 shows a sinusoidal behavior. The action of the saturators ϕ_d^{Sat} and θ_d^{Sat} are not presented because the helical trajectory is defined to make a slow motion, a negative peak of -11 (deg) and a positive peak of 8 (deg) are presented, while the amplitude of angles is ± 0.27 (deg).

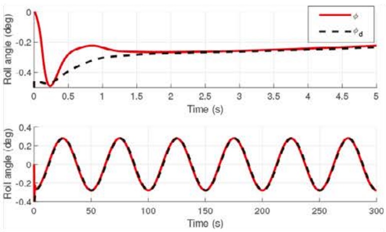

Fig. 4: Response of ϕ angle.

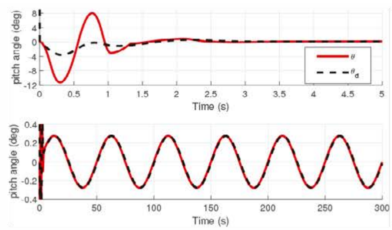

Fig. 5: Response of θ angle.

A regulation in yaw angle is shown in Figure 6. At a time of 0.6 (s) a negative peak of -0.7 (deg) is presented. The reason for $\psi_d = 0$ resides in the idea of always looking in the north direction. Meanwhile, this controller is set as a classic PID due to the response behavior as a regulation that never goes to zero, so the rules of the Expert PID are not an option for this case.

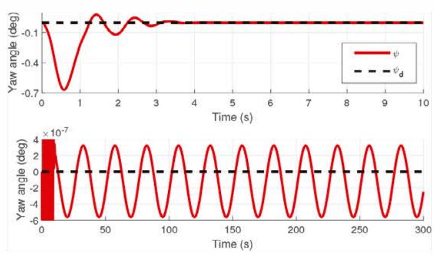

Fig. 6: Response of ψ angle.

Figure 7 shows the altitude response where the response is constantly ascending, which means the total of the octocopter UAV is stabilized.
Figures 8 and 9 show the navigation performing without problems. Due to the response in the Euler angles, there is a minimal deviation unnoticed in the trajectory tracking that performs an outside cylinder from the desired helical trajectory.
In Figure 10, the total thrust compensation is shown. The maximum total thrust that can be performed in the octocopter UAV is 156 N , which is the double force required to maintain the equilibrium of the octocopter UAV. The total thrust goes to 102 N at the beginning of the response; after the stabilization, the total thrust is 78.5 N, which is the necessary force to maintain the flight of the octocopter UAV during navigation.

6 Conclusions

In this research work, we have designed a navigation algorithm for an octocopter UAV using a PID flight controller, which implements expert decision rules. It has been named

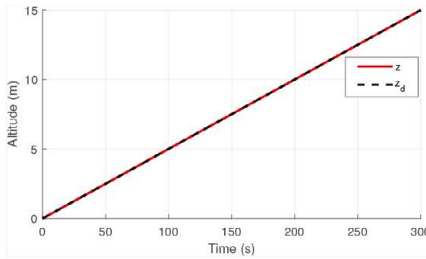

Fig. 7: Response of Z axis position.

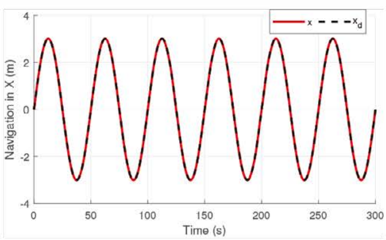

Fig. 8: Response of X axis position.

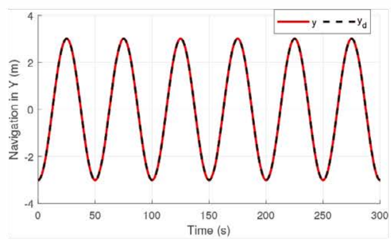

Fig. 9: Response of Y axis position.

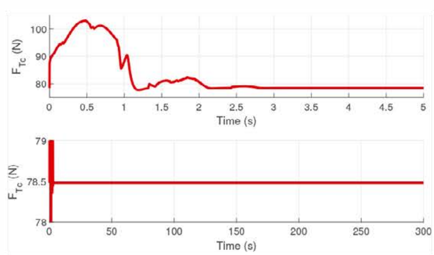

Fig. 10: Total thrust response.

"Expert PID Flight Controller." The navigation algorithm used by the Expert PID flight controller is subjected to simulation tests to fulfill the tracking of a time-varying trajectory, which has a helical geometric shape. The set of expert rules that are part of the "Expert PID Flight Controller" has the ability to reassign the values of the PID controller parameters for a small region of the error of pitch and roll angles, as well as the translation positions in the X, Y, Z axes, the controller keeps the gains in a low profile to implement a weak, very weak and extremely weak controller, on the other hand for a large region of the error of the pitch and roll orientation angles, as well as the translation positions in the X, Y, Z axes, the controller keeps the gains in a high profile to implement a strong, very strong and extremely strong controller. The added capabilities of the navigation algorithm to be able to change the gains of the "Expert PID Flight Controller" allow it to have an improved response to small and large variations of roll and pitch attitude error and translation error in X, Y, Z, versus variations in time of the desired trajectory that the octocopter UAV must follow.

Acknowledgments. The authors would like to thank the Autonomous University of Nuevo Leon (CIIIA-FIME-UANL) and the Tecnologico Nacional de Mexico/Instituto Tecnologico de La Laguna.

Disclosure of Interests. The authors declare no conflict of interest.

References

1. Osmić, N., Kurić, M., Petrović, I.: Detailed octocopter modeling and PD control. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, pp. 002182-002189 (2026). https://doi.org/10.1109/SMC.2016.7844562
2. Velagić, J., Osmić, N., Klovo, V., Lačević, H.: Design of LQR Controller for 3D Trajectory Tracking of Octocopter Unmanned Aerial Vehicle. In: 8th International Conference on Control, Decision and Information Technologies (CoDIT), pp. 63-68, Istanbul, Turkey (2022). https://doi.org/10.1109/CoDIT55151.2022.9803884
3. Sadeghi, P.S., Shahri, A.M., Ardestani, M.A., Rezazadeh, S.: LQG-I control for attitude stabilization of V8 octocopter flying robot. 2016 Artificial Intelligence and Robotics (IRANOPEN), Qazvin, Iran. pp. 151-157 (2016). https://doi.org/10.1109/RIOS.2016.7529506
4. Çabuk, N., Yildirim, Ş.: Design, Modelling and Control of an Eight-Rotors UAV with Asymmetric Configuration for Use in Remote Sensing Systems. Journal of Aviation, vol. 5(2) , pp. 72-81, (2021). https://doi.org/10.30518/jav.943804
5. Bergman, K., Ekström, J.: Modeling, estimation and attitude control of an octocopter using PID and L1 adaptive control techniques. Master's Thesis, Linköping University, Sweden (2014).
6. Merheb, A., Nourra, H., Bateman, F.: Active fault tolerant control of octocopter uav using dynamic control allocation. In International Conference on Intelligent Unmanned Systems. (2014)
7. Merheb, A.R., Noura, H.: Passive and active fault tolerant control of octocopter UAV using second order sliding mode control. In 2015 IEEE Conference on Control Applications (CCA), pp. 1907-1912. IEEE, Sydney, NSW, Australia (2015). https://doi.org/10.1109/CCA.2015.7320888
8. Rigatos, G., Siano, P., Abbaszadeh, M., Monteriu, A.: A nonlinear optimal control approach for the autonomous octocopter. Advanced Control for Applications: Engineering and Industrial Systems, vol. 2(3), Wiley (2020). https://doi.org/10.1002/adc2.50
9. Liu, J.: Intelligent control design and Matlab simulation. 1st edn. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-5263-7
10. Tellez-Belkotosky, P.A., Cabriales-Ramirez, L.E., Gutierrez-Martinez, M.A., Ollervides-Vazquez, E.J.: Intelligent PIV Fuzzy Navigation and Attitude Controller for an Octorotor Mini-UAV. Machines. vol. 11(2):226, (2023). https://doi.org/10.3390/machines11020266
11. Ollervides-Vazquez, E.J., Tellez-Belkotosky, P.A., Santibañez, V., Rojo-Rodriguez, E.G., Reyes-Osorio, L.A., Garcia-Salazar, O.: Modeling and Simulation of an Octorotor UAV with Manipulator Arm. Drones.7(3):168, (2023). https://doi.org/10.3390/drones7030168