İSTANBUL UNIVERSITY CERRAHPAŞA

RESEARCH ARTICLE

Electrica 2020; 20(2): 199-206

# Design and Implementation of a Multi-Stage PID Controller for Non-inertial Referenced UAV

**Fady Alami[1], Abdulrahman Hussian[2], Naim Ajlouni[3]**

[1]Department of Electrical and Electronics Engineering, İstanbul University-Cerrahpaşa, İstanbul, Turkey
[2]Academic Coordination, Al-Maaly University Tripoli Libya, İstanbul, Turkey
[3]Deparment of Computer Engineering/Science Istanbul Aydin University, İstanbul, Turkey

**Cite this article as:** Alami F, Hussian A, Ajlouni N. Design and Implementation of a Multi-Stage PID Controller for Non-inertial Referenced UAV. *Electrica*, 2020; 20(2): 199-206.

**ABSTRACT**

This work addresses the tracking issue for a non-inertial frame referenced quadrotor unmanned aerial vehicle (UAV) controlled by a cascaded proportional–integral–derivative (PID) controller. Some of the current applications of quadrotors, such as those used in sea search and rescue operations, are launched from a moving vessel. The landing of such quadrotors must consider the non-inertial position of the vessel to be landed on. Nearly every study in this area has represented the dynamics of a quadrotor UAV based on a fixed inertial frame. The most widely used inertial frames are the geodetic coordinate system that depends on the Earth's surface and the Earth-centered fixed coordinate system. This work aims to analyze the orientation, rotation, velocity, and position of a quadrotor that is based on a moving object. The quadrotor kinematics will consider the rotation and orientation for both a non-inertial frame of reference (vessel) and a fixed inertial frame of reference (base point). The system dynamics will depend on the initial take-off point as an inertial reference to give the correct dynamical effects on the quadrotor body frame. Most accidents occur during bad weather conditions, in which case, cascaded PID controllers should be used to control a quadcopter to face weather disturbances.

**Keywords:** Multi-Stage PID controller, non-inertial reference, unmanned aerial vehicles modeling

## Introduction

Various tasks have been performed using drones. One of the most useful tasks is the search and rescue (SAR) operations task. Many types of SAR operations are performed, and one of which is done over water. SAR operations over water are much more difficult than those done over land because harsh weather conditions exist over water. Drones are mounted on vessels, and in emergency situations, they can do SAR operations on their way to their destinations. Therefore, a drone should make the SAR operations referenced to the vessel as a reference base point for its movement. We need to relate the drone as a moving body frame to the vessel as a moving object, that is, the non-inertial frame of reference (NIFR). Most of the relevant studies have concentrated on referencing the quadcopter to a fixed frame or inertial frame of reference (IFR). Some of the known inertial frames of reference are the Earth-centered coordinate system, geodetic coordinate system (longitude, latitude, and altitude) and local north-east-up or down coordinate system [1]. These coordinate systems can be used as a fixed frame to determine our translational and rotational kinematics, such as the position, velocity, acceleration, and orientation of the moving frame (drone body frame). The translational kinematics of the system will be changed according to the NIFR, but the dynamical kinematics will be the same, and the initial take-off point will be considered as the reference for the dynamical calculations. Therefore, in the IFR, returning to the take-off point is easy, but returning to a moving reference is difficult. Let us consider this scenario. A drone should take off from a moving object vessel and return back to the same take-off point, but the problem is that the take-off point is a moving point attached to our moving frame (i.e., the vessel). In this case, we need to relate the drone to a moving object, such that its reference will be a NIFR. This idea of relating the drone to a NIFR can help us do many applications that can be very useful. One of the most important applications is the SAR operation. Previous studies and projects have been implemented to perform SAR operations done over water. One of these studies presented Pars Project, which was done by RTS Ideas [2]. Pars Project can be used on coasts by lifeguards to do rescue missions. It does the lifeguards' job of rescuing drowning people at the moment of the accident until the lifeguards reach the victims. Pars Project has not considered weather conditions. Moreover, Pars is prepared to save the lives of people who are swimming beside coasts; hence, it depends on the take-off IFR.

**Corresponding Author:**
Fady Alami

**E-mail:**
eng.fadi.alami@gmail.com

**Received:** 12.01.2020

**Accepted:** 02.03.2020

**DOI:** 10.26650/electrica.2020.20004

Abbasi and Mahjoob [3] wanted to implement intelligent control techniques to control a quadrotor and achieve stability. They determined the quadrotor model as a nonlinear model system. They used fuzzy control to adjust the proportional–integral–derivative (PID) controller gains, but did not consider a dynamical model system.

Marcu [4] used an intelligent control technique (fuzzy control) to identify the position of the UAV that is related to a destination point and based on two angles. He compared this technique with genetic algorithm and neural networks. The advantage of the technique was the way it took decisions in real time without the need to present any domain-specific data to the system before running the simulation. This method cannot be applied to multi-copters. It can only be applied to fixed-wing UAVs because in a multi-copter, all the three angles we have must be considered.

This work presents a UAV model and constructs a nonlinear cascaded controller to control the UAV's position and attitude. The mechanical and electrical structures of the UAV will be explained. Some terms and definitions (i.e., kinematics, dynamics, Newton–Euler equations, and Euler–Lagrange equations of motion) will be explained in this paper. The conversion from the IFR to the NIFR is also presented. The simulation and its results are explained in the end.

**UAV Modeling**

Some important base terms and symbols are defined below. Table 1 presents the description.

$$\boldsymbol{\xi} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \boldsymbol{\eta} = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}, \boldsymbol{V}_b = \begin{bmatrix} u_b \\ v_b \\ w_b \end{bmatrix}, \boldsymbol{V}_i = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} \text{ or } \dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \ddot{\boldsymbol{\xi}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \dot{\boldsymbol{\upsilon}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}, \ddot{\boldsymbol{\eta}} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}.$$

**Kinematics Modeling**

Kinematics is the study of the motion of a point, object, or multi-object "geometry of motion" like the motion of a ball, robotic arms, aircraft, and other objects without referring to the causes of motion. It is the study of position (X, Y, and Z) and rotational angles (θ, φ, and ψ) to a specific initial frame (i.e., Earth-centered coordinate system). To describe drone kinematics, one must understand the transformation between inertial and body frames. The transformation between frames can be represented by a set of rotations; thus, we can use the rotation matrices to transform any point from the inertial frame to the body frame. The rotation matrix can be easily explained by the projections of the vector of each axis. Let us consider a rotation about the Z axis by an angle of ψ (Figure 1).

From the projections of the three vectors, we can combine the rotation matrices as shown in Equation [5].:

$$\boldsymbol{R} = \begin{bmatrix} x_1 . x_0 & y_1 . x_0 & z_1 . x_0 \\ x_1 . y_0 & y_1 . y_0 & z_1 . y_0 \\ x_1 . z_0 & y_1 . z_0 & z_1 . z_0 \end{bmatrix} \quad (1)$$

Any transformation from a base frame to a body frame or vice versa can be represented by three successive rotations. We will use a roll–pitch–yaw representation (Euler angles). We will have a rotation about the current Z axis, then about the current the Y axis, and then about the current X axis. The resultant rotational transformation matrix for the roll–pitch–yaw representation (ZYX Euler angles) is presented as follows:

$$\boldsymbol{R}_{Z,Y,X} = \boldsymbol{R}_{Z,\psi} \boldsymbol{R}_{Y,\theta} \boldsymbol{R}_{X,\phi} \quad (2)$$

Thus, the rotation matrix can be presented as

$$\boldsymbol{R}_{Z,Y,X} = \begin{bmatrix} C_\theta C_\psi & C_\psi S_\theta S_\phi - C_\phi S_\psi & S_\phi S_\psi + C_\phi C_\psi S_\theta \\ C_\theta C_\psi & C_\phi C_\psi + S_\theta S_\phi S_\psi & C_\phi S_\theta S_\psi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \quad (3)$$

**Translational Kinematics**

To obtain the linear velocity vector in the inertial frame, the body frame velocities must be converted using a rotation matrix from the body frame to the inertial frame. The velocities of the drone in the body frame coordinate can be taken from the sensors (i.e., gyroscope sensor) attached to the drone. The linear velocity is $\acute{\varepsilon} = [\dot{x}\ \dot{y}\ \dot{z}]^T$. The body frame velocity vector is $Vb = [u_b\ v_b\ w_b]^T$. The $v_b$ vector values are from the gyroscope sensor. We multiply the rotation matrix $R$ by $v_b$ to obtain the values of the $\acute{\varepsilon}$ vector as follows:

**Table 1.** UAV modeling terms and symbols

| Description | Symbol | Unit |
|---|---|---|
| Position vector of the drone on the inertial frame | (ξ) | Meters (m) |
| Angular position vector (rotation) | (η) | Radians (rad) |
| Linear velocity vector in the body frame | (Vb) | Meter/second (m/s) |
| Linear velocity vector in the inertial frame | ($\acute{\varepsilon}$) | Meter/second (m/s) |
| Angular velocities vector of the body frame | (ω) | Radians per second (rad/s) |
| Angular velocity vector of the inertial frame | ($\dot{\eta}$) | Radians per second (rad/s) |
| Linear acceleration vector | ($\ddot{\xi}$) | m/s$^2$ |
| Angular acceleration vector | ($\ddot{\eta}$) | rad/s$^2$ |

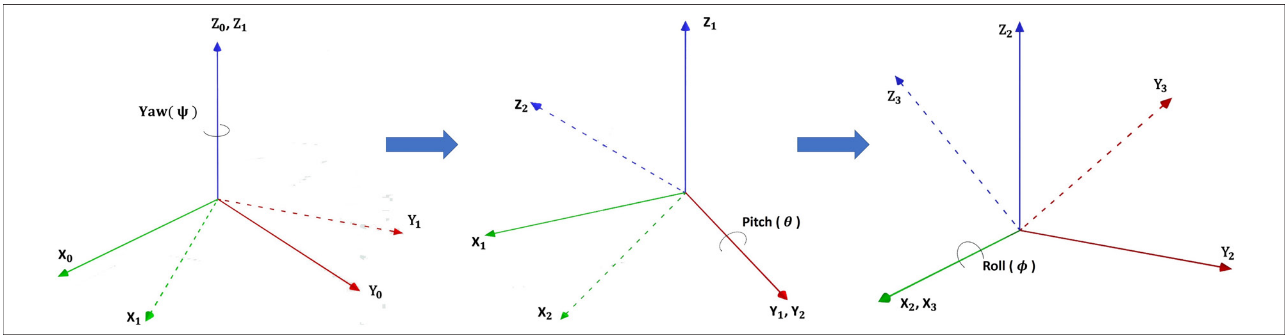


**Figure 1.** Transformation from the inertial frame to the body frame

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & C_\psi S_\theta S_\phi - C_\phi S_\psi & S_\phi S_\psi + C_\phi C_\psi S_\theta \\ C_\theta C_\psi & C_\phi C_\psi + S_\theta S_\phi S_\psi & C_\phi S_\theta S_\psi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}\begin{bmatrix}u_b\\ v_b\\ w_b\end{bmatrix} \tag{4}$$

**Rotational Kinematics**

In linear velocity transformations, the angles do not change with time; however, in the angular velocity, they do. If we consider the general case of the angular velocity about an arbitrary possibly moving axis, the rotation matrix $R$ is time-varying. The time derivative $\dot{R}(t)$ can be derived from the rotation matrix as,

$$\dot{\boldsymbol{R}} = \boldsymbol{S}\big(w(t)\big)\boldsymbol{R}(t) \tag{5}$$

where matrix $S(\omega(t))$ is skew-symmetric for a unique vector $\omega(t)$. This vector $\omega(t)$ is the angular velocity of the rotating frame with respect to the fixed frame at time t. $S(\omega(t)$ can also be represented as $\theta S(i)$, where $\theta$ is the angle of rotation. Equation 5 can be rewritten as

$$\boldsymbol{S}\big(w(t)\big)\boldsymbol{R}(t) = \dot{\theta}\boldsymbol{S}(i) \tag{6}$$

where $\theta S(i)$ is the derivative of a rotating angle $\theta$ over time. Equation 6 can be written as

$$\boldsymbol{S}\big(w(t)\big)\boldsymbol{R}(t) = \frac{d\theta}{dt}\cdot\frac{d\boldsymbol{R}}{d\theta} = \frac{d\boldsymbol{R}}{dt} \tag{7}$$

Hence, Equation 7 can be written as $S(w(t))R(t) = \dot{R}$ as a proof to Equation 5. We can implement a suitable relevant rotating angle formulation from the above base equations:

$$\dot{\boldsymbol{R}}_n^0 = \boldsymbol{S}(\omega_{0,n}^0)\boldsymbol{R}_n^0 \tag{8}$$

The notation $\omega^0{}_0$,n relates the base frame with the current and next frames. Where the rotation matrix between many frames is $R^0{}_n$,

$$\boldsymbol{\omega}_{0,3}^0 = \boldsymbol{\omega}_{0,1}^0 + \boldsymbol{R}_1^0\boldsymbol{\omega}_{1,2}^1 + \boldsymbol{R}_2^0\boldsymbol{\omega}_{2,3}^2 \tag{9}$$

For three axes, the angular velocity vector will be expressed in three relative transformations related to three angular velocities; thus,

$$\boldsymbol{\omega}_{0,3}^0 = \begin{bmatrix}1\\0\\0\end{bmatrix}\dot{\phi} + \begin{bmatrix}0\\ cos\,(\phi)\\ -sin\,(\phi)\end{bmatrix}\dot{\theta} + \begin{bmatrix}-sin\,(\theta)\\ cos\,(\theta)sin\,(\phi)\\ cos\,(\phi)cos\,(\theta)\end{bmatrix}\dot{\psi} \tag{10}$$

The angular velocity vector of the body frame can be obtained by multiplying the angular velocity transformation matrix by η,

$$\begin{bmatrix}p\\q\\r\end{bmatrix} = \begin{bmatrix}1 & 0 & -sin\,(\theta)\\ 0 & cos\,(\phi) & cos\,(\theta)sin\,(\phi)\\ 0 & -sin\,(\phi) & cos\,(\phi)cos\,(\theta)\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \tag{11}$$

The inverse of the angular velocity matrix will be used as follows to obtain the angular velocities represented in the inertial frame from the body frame:

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & sin\,(\phi)tan\,(\theta) & cos\,(\phi)tan(\theta)\\ 0 & cos\,(\phi) & -sin\,(\phi)\\ 0 & \dfrac{sin\,(\phi)}{cos\,(\theta)} & \dfrac{cos\,(\phi)}{cos\,(\theta)}\end{bmatrix}\begin{bmatrix}p\\q\\r\end{bmatrix} \tag{12}$$

**Translational Dynamics**

Many forces affect the linear and rotational motion of a drone. We will consider the thrust force, gravitational force, and aerodynamic drag force.

$$\boldsymbol{F} = \boldsymbol{F}_g + \boldsymbol{F}_t + \boldsymbol{F}_d \tag{13}$$

where $F_g$ is the gravitational force; $F_t$ denotes the thrust forces; and $F_d$ represent the drag forces. The thrust force is the force generated by the motors over the drone. Each rotor has an angular velocity $\omega_i$ with the direction along the $Z_B$ axis that generates a force, $f_i$, with only one component, $f_i$, along the same direction [6, 7].

$$f_t = \sum_{i=1}^{4} f_i = k\sum_{i=1}^{4}\omega_i^2 \tag{14}$$

where $k$ is the lift constant. The thrust force is acting on the Z axis of the body frame, which is considered as a rigid body. The direction of the thrust force vectors from the motors is exactly parallel to the Z axis of the body frame.

$$\boldsymbol{F}_t = \begin{bmatrix} S_\phi S_\psi + C_\phi C_\psi S_\theta \\ C_\phi S_\theta S_\psi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix} f_t \tag{15}$$

$$\boldsymbol{F}_d = \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = \frac{1}{2}\rho_{air} I \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} \begin{bmatrix} V_x^2 & 0 & 0 \\ 0 & V_y^2 & 0 \\ 0 & 0 & V_z^2 \end{bmatrix} \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} \tag{16}$$

where  is the fluid density (Air)[kg/m³]; $A_i$ is the cross-sectional area, which is the area of the drone exposed to the wind [m²]; V denotes the speed of the object relative to the fluid or wind speed [m/s]; and $C_f$ is the drag coefficient in the inertial frame. The total force will be [8],

$$\boldsymbol{F} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} S_\phi S_\psi + C_\phi C_\psi S_\theta \\ C_\phi S_\theta S_\psi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix} f_t + \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} \tag{17}$$

The result of the investigation on forces can be summarized in one basic equation involving the linear acceleration expressed in the body frame:

$$\begin{bmatrix} \ddot{x}^b \\ \ddot{y}^b \\ \ddot{z}^b \end{bmatrix} = \begin{bmatrix} \dfrac{(S_\phi S_\psi + C_\phi C_\psi S_\theta) f_x + D_x}{m} \\ \dfrac{(C_\phi S_\theta S_\psi - C_\psi S_\phi) f_y + D_y}{m} \\ \dfrac{(C_\theta C_\phi) f_z + D_z}{m} - g \end{bmatrix} \tag{18}$$

We multiply the linear acceleration vector of the body frame by the rotational matrix to obtain the acceleration vector referenced to the inertial frame:

$$\begin{bmatrix} \ddot{x}^i \\ \ddot{y}^i \\ \ddot{z}^i \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} \ddot{x}^b \\ \ddot{y}^b \\ \ddot{z}^b \end{bmatrix} \tag{19}$$

**Rotational Dynamics**

Two methods can be used to analyze the rotational dynamics, that is, the Newton–Euler and Euler–Lagrange methods. In the Newton–Euler method, the forces affecting the drone are summed to obtain the angular acceleration from analyzing these forces [9, 10].

$$\boldsymbol{\tau} = \boldsymbol{I}\dot{\boldsymbol{v}} + \boldsymbol{v} \times (\boldsymbol{I}\boldsymbol{v}) + \boldsymbol{\Gamma} \tag{20}$$

$\tau$ denotes the torques generated by the rotors; $Iv$ is the angular acceleration of the inertia in the body frame; $v \times (Iv)$ represents the centripetal forces; and $\Gamma$ denotes gyroscopic forces.

The torques generated by the motors can be expressed as

$$\begin{bmatrix} \Sigma T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & lc_T & 0 & -lc_T \\ -lc_T & 0 & lc_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{21}$$

where $c_Q$ is the torque coefficient for the motor-prop system; $c_r$ is the lumped parameter of the thrust coefficient proportion to the motor-prop system; and $l$ is the length of the arm from the quadcopter center to the motor-prop.

The centripetal force will be

$$\boldsymbol{v} \times (\boldsymbol{I}\boldsymbol{v}) = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} p \\ I_{yy} q \\ I_{zz} r \end{bmatrix} \tag{22}$$

The gyroscopic force term is presented as

$$\boldsymbol{\Gamma} = I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_\Gamma \tag{23}$$

where $I_{xx}$, $I_{yy}$ and $I_{zz}$ denote the moment of inertia relative to the center of mass; $I_r$ is the moment of inertia of the rotor; and $\omega_\Gamma$ can be obtained as follows: $\omega_\Gamma = \omega_1 - \omega_2 - \omega_3 - \omega_4$. The resultant angular acceleration equation is

$$\dot{\boldsymbol{\upsilon}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz}) qr / I_{xx} \\ (I_{zz} - I_{xx}) pr / I_{yy} \\ (I_{xx} - I_{yy}) pq / I_{zz} \end{bmatrix} - I_r \begin{bmatrix} q/I_{xx} \\ -p/I_{yy} \\ 0 \end{bmatrix} \omega_\Gamma + \begin{bmatrix} \tau_\phi / I_{xx} \\ \tau_\theta / I_{yy} \\ \tau_\psi / I_{zz} \end{bmatrix} \tag{24}$$

The angular acceleration in the inertial frame is then attracted from the body frame accelerations with the transformation matrix $W_\eta^{-1}$ and its time derivative:

$$\ddot{\boldsymbol{\eta}} = \begin{bmatrix} 0 & \dot{\phi} C_\phi T_\theta + \dot{\theta} S_\phi / C_\theta^2 & -\dot{\phi} S_\phi C_\theta + \dot{\theta} C_\phi / C_\theta^2 \\ 0 & -\dot{\phi} S_\phi & -\dot{\phi} C_\phi \\ 0 & \dot{\phi} C_\phi / C_\theta + \dot{\phi} S_\phi T_\theta / C_\theta & -\dot{\phi} S_\phi / C_\theta + \dot{\theta} C_\phi T_\theta / C_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \boldsymbol{W}_\eta^{-1} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \tag{25}$$

We now have the model variables that can be used to construct the UAV system. The basic variables in the model are the linear acceleration and the angular acceleration vectors $\ddot{\boldsymbol{\xi}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \ddot{\boldsymbol{\eta}} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}$.

**Transformation from an IFR to the NIFR**

To transform from an IFR to a NIFR, we must make a relation between the body frame position as a drone and the moving object as a car, ship, or even a plane. Normally, we need to obtain the state equation of this system to make the mathematical model of any system. We also need to obtain the velocity to derive the position equation because the position state is the body frame velocity.

$$\begin{bmatrix} \dot{x}_b^i \\ \dot{y}_b^i \\ \dot{z}_b^i \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} u_b \\ v_b \\ w_b \end{bmatrix} \tag{26}$$

A relation between the velocity vector of the body frame referenced to the inertial frame and the velocity of the reference moving object also referenced to the inertial frame should be established. This will provide the velocity of the body frame referenced to the moving object as a non-inertial frame:

$$\dot{\boldsymbol{\xi}}_b^m = \dot{\boldsymbol{\xi}}_b^i - \dot{\boldsymbol{\xi}}_m^i = \begin{bmatrix} \dot{x}_b^i \\ \dot{y}_b^i \\ \dot{z}_b^i \end{bmatrix} - \begin{bmatrix} \dot{x}_m^i \\ \dot{y}_m^i \\ \dot{z}_m^i \end{bmatrix} = \begin{bmatrix} \dot{x}_b^m \\ \dot{y}_b^m \\ \dot{z}_b^m \end{bmatrix} \tag{27}$$

$$Velocity\ error = \dot{\boldsymbol{\xi}}_{desired}^m - \dot{\boldsymbol{\xi}}_b^m \tag{28}$$

where $\dot{\xi}_b^i$ is the position state vector of the body frame referenced to the inertial frame, and $\dot{\xi}_m^i$ is the position state vector of the moving object related to the inertial frame, $\dot{\xi}_b^m$, and is the resultant position state vector of the body frame related to the moving object.

The output state vector of the system model should be changed to make a suitable transformation from the IFR to a NIFR. The normal output state vector inertially referenced can be expressed as

$$\boldsymbol{X}^T = [x\ \ y\ \ z\ \ u\ \ v\ \ w\ \ \phi\ \ \theta\ \ \psi\ \ p\ \ q\ \ r]^T \tag{29}$$

The state vector of the inertial referenced frame contains information on the position, velocity, angular position, and rotational velocity of the body frame. In the case of the non-inertial reference frame, the information on the position and velocity of the moving body should be included in the state vector,

$$\boldsymbol{X}^T = [x_b\ y_b\ z_b\ u_b\ v_b\ w_b\ x_m\ y_m\ z_m\ u_m\ v_m\ w_m\ \phi\ \theta\ \psi\ p\ q\ r]^T \tag{30}$$

To show our new model result, we first need to obtain the system model in MATLAB Simulink [11] with the inertial frame referenced. We then need to transform it from the IFR to a NIFR. The error of position in the body frame will be a result of relating the position feedback of the body frame with the position feedback of the moving frame:

$$x_{err}^b = x_d^m + x_m^i - x_b^i \tag{31}$$

where  is the desired position referenced to the moving object (non-inertial frame referencing); $x_m^i$ is the feedback position of the moving object; and $x_b^i$ is the feedback position of the body frame referenced to the inertial frame.

The modification result will make the desired position to be the actual desired position referring to the moving object non-inertial frame and any movement of the non-inertial frame. The position and the velocity state vector of the non-inertial frame will be derived from its acceleration. The acceleration information on the

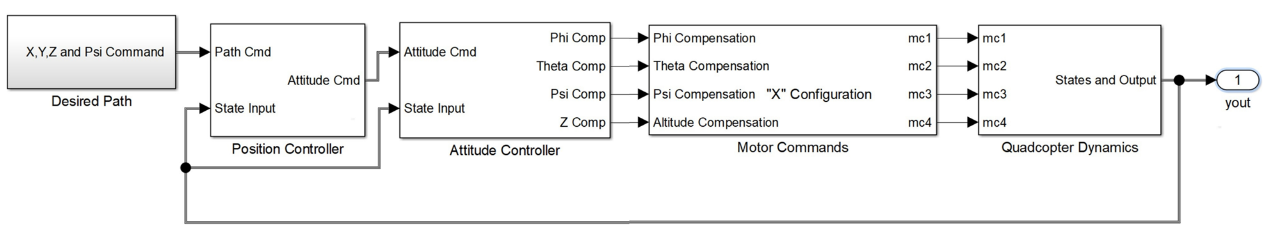


**Figure 2.** General structure of the control system

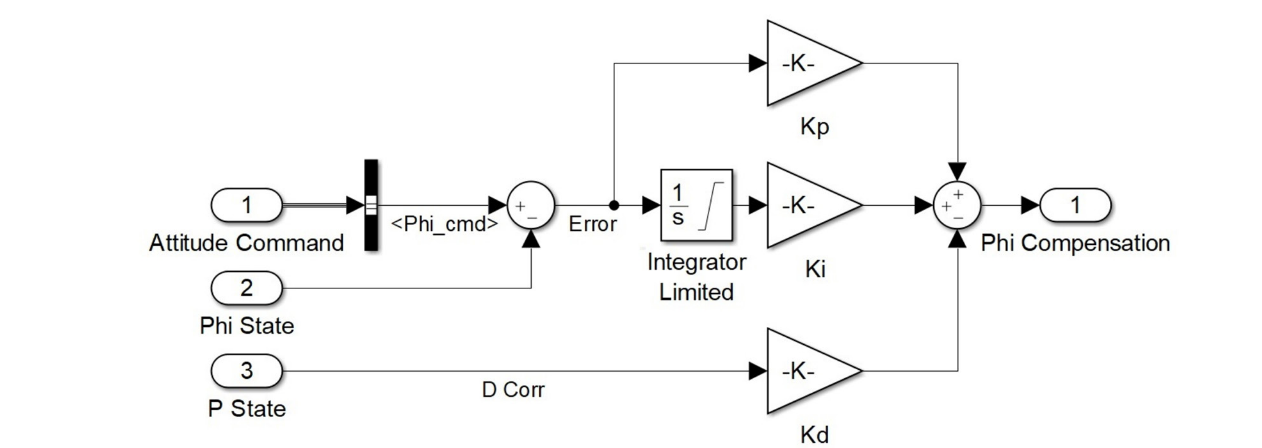


**Figure 3.** PID controller structure used for controlling the position and angles of vectors

moving object can be directly obtained from the moving object system or by the external sensors mounted over it.

**Cascade PID controller**

Our system is a nonlinear multi-input, multi-output system. To control this system, we used a cascade PID controller comprising two stages. The first stage is for controlling the drone velocity depending on the position error vector information. This stage uses two PID controllers: one for the velocity in the X axis of the body frame and another for the Y axis of the body frame. The second stage is for controlling the pitch of the angles of vectors (roll–pitch–yaw) and the altitude of the drone. Figure 2 shows the structure of the controller system.

In the second stage, we have three PID controllers for controlling the angles of the drone. The input of these controllers will be the desired angle and the state feedback of the angle. The desired angle will depend on the desired velocity that can be driven from the position control error. As an example, we have the controller of the Phi angle shown in Figure 3.

The PID controller equation is

$$\varphi_{Comp}(t) = K_p(\varphi_{cmd}(t) - \varphi_{state}(t)) + K_i \int (\varphi_{cmd}(t) - \varphi_{state}(t))dt + K_d * P_{state}(t) \quad (32)$$

where $P_{state}$ $(t)$ is the angular velocity feedback. In this case, the angular velocity feedback $P_{state}$ is an input to the derivative part of the controller. We do not need to derive again the change of error. The angular velocity state (P) can directly be multiplied with gain $K_d$ (Figure 3). The same procedure is done for $\varphi$, $\psi$, and z.

Figure 4 shows the result of the PID controller, where the black dashed line represents the desired Phi angle, and the blue dotted line denotes the actual Phi angle.

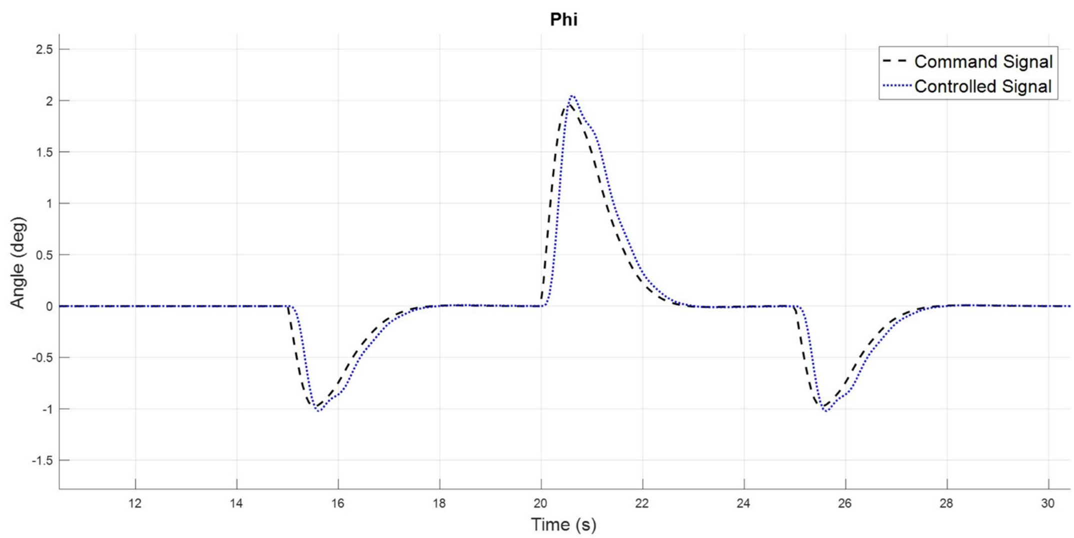


**Figure 4.** Controller result for the Phi angle

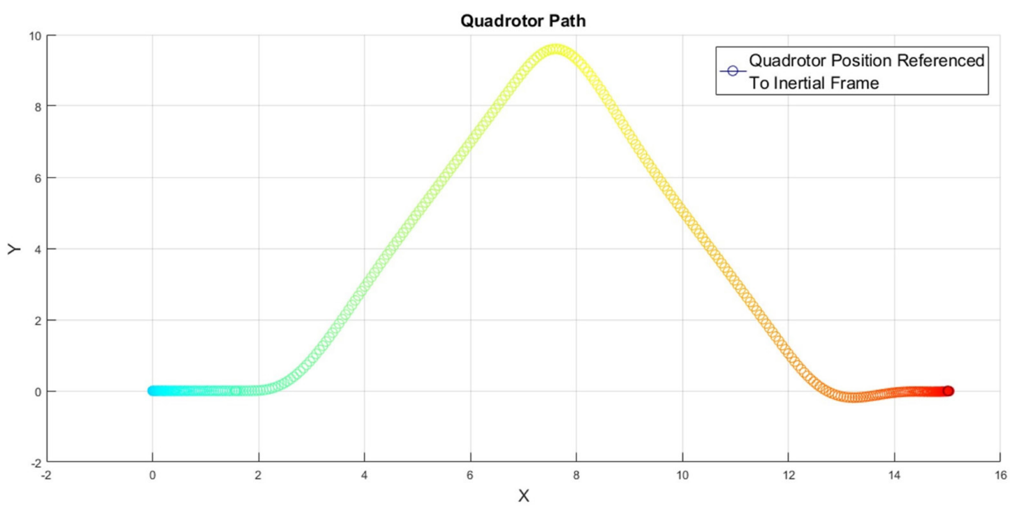


**Figure 5.** Result of relating the body frame with the moving frame referenced to the inertial frame

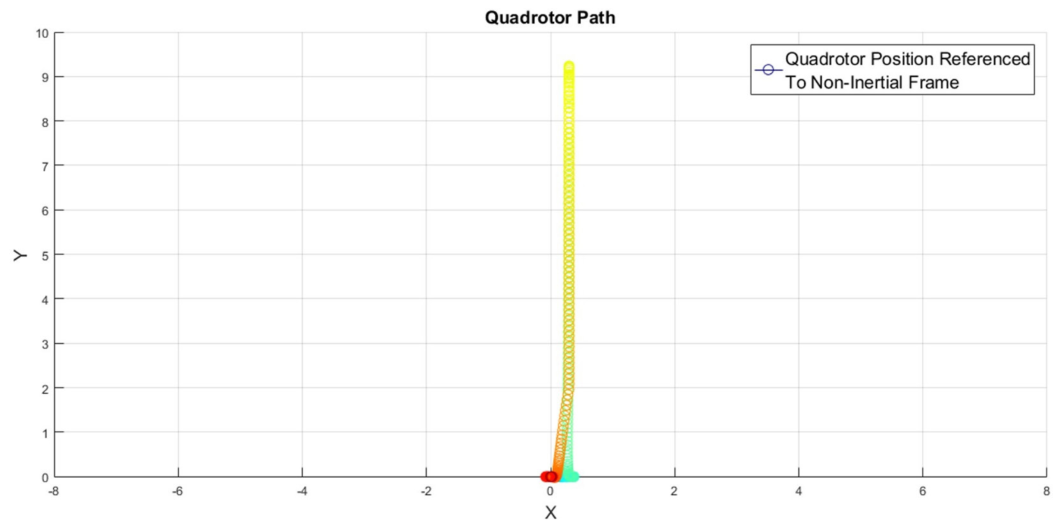


**Figure 6.** Result of relating the body frame with the moving frame referenced to the non-inertial frame

## Simulation

A numerical example was implemented to compare the inertial and non-inertial frame referenced results. A command signal to the body frame was given to move 10 m in the positive Y axis direction referenced to the moving object with no movement in the X axis. The moving object will also move 15 m in the positive X axis with no movement in the Y axis referenced to the inertial frame as the initial position. Figure 5 depicts the result and clearly shows that the drone was moving in both X and Y axes. This is the real movement referred to the inertial frame. The cause of movement in the Y axis is the command given to the drone as a desired position relative to the moving object, while that in the X axis is that the drone is trying to relate its movement to the moving object.

The results can be clarified by changing the plot from the inertial frame of the reference side of view to the non-inertial frame reference side of view. Let us consider the case of tracking the body frame from the moving object. Figure 6 shows the result for this. Thus, the same drone movement can be presented even relatively to the inertial frame reference (Figure 5) or relatively to the non-inertial frame reference (Figure 6).

## Conclusion

SAR operations are considered as the most important sensitive operations. We can obtain perfect results if we use an intelligent algorithm to optimize these operations. Using UAVs in these operations can prevent life loss or injuries.

For SAR operations, more than one algorithm and device must be used to obtain useful information that can help us achieve our aim. We found that using a non-inertial frame can work as a reference to our UAV and help the SAR operation succeed. Most of the previous studies depended on an inertial frame as the reference to the UAV. We showed herein how we can use a non-inertial frame as a reference to the UAV. We also compared the inertial frame and the non-inertial frame as a reference and how it can be easily used a non-inertial frame in special cases, such as in SAR operations.

These operations generally occur in harsh weather conditions; hence, we need advanced control techniques to be applied to our system. We prefer using a cascaded PID controller because it gives us better results compared to a one-stage control system.

**Peer-review:** Externally peer-reviewed.

**Conflict of Interest:** Authors have no conflicts of interest to declare.

**Financial Disclosure:** The authors declared that this study has received no financial support.

## References

1. G. Cai, B. M. Chen, and T. H. Lee, "Unmanned Rotorcraft Systems," Advances in Industrial Control, Springer, Singapore, 2011. [Crossref]
2. L. Kelion, "Iran develops sea rescue drone prototype in Tehran," BBC News, 13-Dec-2013. [Online]. Available: https://www.bbc.com/news/technology-24929924. [Accessed: 23-Jul-2019].
3. E. Abbasi, M. J. Mahjoob "Controlling of Quadrotor UAV Using a Fuzzy System for Tuning the PID Gains in Hovering Mode," University of Tehran, 2013
4. E. Marcu, "Fuzzy Logic Approach in Real-time UAV Control," Control Engineering and Applied Informatics, vol. 13, no. 1, pp. 12-17, Polytechnics University of Bucharest, 2011
5. M. W. Spong, S. Hutchinson, and M. Vidyasagar, "Robot modeling and control", Wiley, Hoboken, NJ, 2006.
6. S. Musa, "Techniques for Quadcopter Modelling & Design: A review", Journal of Unmanned System Technology, vol. 5, no. 3, pp. 66-75, 2017.
7. M. Cano, Javier, "Quadrotor UAV for wind profile characterization", M.S. thesis, Universidad Carlos III de Madrid, Spain, 2013.
8. I. C. Dikmen, A. Arisoy, and H. Temeltas, "Attitude control of a quadrotor," 2009 4th International Conference on Recent Advances in Space Technologies, Istanbul, 2009. [Crossref]
9. V. Artale, C. Milazzo, and A. Ricciardello, "Mathematical modeling of hexacopter," Applied Mathematical Sciences, vol. 7, pp. 4805–4811, 2013. [Crossref]
10. T. Luukkonen, "Modelling and control of quadcopter," Independent research project in applied mathematics, Aalto University, 2011.
11. D. Hartman, K. Landis, M. Mehrer, S. Moreno, J. Kim, "Quadcopter Dynamic Modeling and Simulation Using MATLAB and Simulink", Drexel University, Simulink Student Challenge 2014, 2014.

Fady Alami was born in Gaza in 1987. He has obtained his B.Sc. in Electric and Communication Engineering from Islamic University in 2009. He has obtained M.Sc. degree in Electrical Engineering from Islamic University in 2013. He started Ph.D. education in Electrical and Electronics Engineering at Istanbul University in 2014, currently he is in thesis stage. He worked as a Research and Teaching Assistant in the Department of Electrical and Electronics Engineering University College of Applied Sciences – Gaza in 2017. His current research interests include Automation, Programmable Logic Controllers (PLC), Automated Guided Vehicles (AGV), Unmanned Aerial Vehicles (UAV), CNC machines.

Prof. Dr. Abdulrahman HUSSIAN is specializes in automation Engineering at Istanbul University. His research interests in voice controlling robots. He is currently works as Academic Coordinator at Al-Maaly University and He is also a Chairman of the Organizing Committee of Arabic conference networks. He severed for 6 years as a head of Automation Control Department at Aleppo university

Naim Ajlouni Obtained his BSc in Electric and Electronic Engineering from Salford University in 1983, His MSc degree in robotics from Salford University in 1992, PhD in Intelligent Control from Salford University in 1995. Currently, he is a Professor of Computer Engineering at the faculty of Engineering, Istanbul Aydin University, Turkey. His research interest includes intelligent systems, Optimization, signal processing, AI, Machine Learning, and Data Science.