Quadcopter Modelling, Control Design and PIL Verification Based on DSP F28377s

Yassine El houm*, Ahmed Abbou, Ali Mousmi
Mohamed V University
Mohammadia Schools of Engineers
Street Ibn Sina B, P 765 Agdal Rabat, Morocco
Email: elhoum.yassine@gmail.com, abbou@emi.ac.ma, amousmi@gmail.com

***Abstract*—This paper focuses on modeling and controlling a nonlinear quadcopter plant system using two different approaches. The first represents a pure mathematical model calculated based on Euler Lagrange formalism using only Matlab/Simulink environment, the second one shows a physical model (CAD model) imported from Solidworks to Simscape multibody which include a 3d simulation environment. The controller part of both models are constituted from two cascaded loop, an inner-loop for attitude control and an outer-loop for position and altitude control. Finally, in order to make sure that the result obtained in the simulation environment matched the real implementation, the paper offers a processor-in-the loop simulation using as a hardware target LAUNCHXL-F28377s.**

Keywords: Quadcopter, PD control, nonlinear control, position control, attitude control, PIL simulation

I. Introduction

Research in the field of drones is essentially multidisciplinary. Indeed it involves very varied fields such as aerodynamics, signal and image processing, control design mechanics, composite materials, real-time computing, etc. Recently the interest in drones seems to be increasing in multiple applications such as agriculture, products and goods transport or even for military applications and intervention in hostile environments [1] [2] [3]. We can imagine a drone that explores a contaminated building or tunnel and to make a first observation before any human intervention.

The aim of this paper is to design a controller to track a desired trajectory of a 6DOF quadcopter plant model [4] and to perform a stable fly experience even in presence of some sustained external disturbances. An inner loop PD controller was chosen in first place to control the three rotational angles ϕ, θ, ψ (attitude control), and an outer loop PD position controller takeover of the x and y translational variable. Altitude is separately controlled out of the rest by a typical PID controller [5]. Since the plant model represent many non-linear behaviors and some variables are coupled due to natural factors like gyroscopic and inertial effects, in order to enhance the overall system performance the paper treat a nonlinear controller based on backstepping technique [6] [7] to replace the PD attitude controller. The simulation of the system is carried out in Matlab/Simulink environment with the use of a CAD model imported from Solidworks for the physical modeling part. The final task focuses on a Processor-in-the-Loop simulation (PIL) which verify whether the deployed C/C++ code in the hardware target matched the simulation design, for this purpose a LAUNCHXL-F28377s DSP platform was chosen to accomplish this task.

II. Mathematical Modeling and Control Design

This section presents the working principle of a 6 DOF quadcopter with a full detailed description of its dynamics, the mathematical model system is obtained using Euler-Lagrange formalism with the assumption that:

- The components of the system are rigid bodies.
- The body mass is concentrated at the center of gravity.
- The body axes are the principal axes for the quadcopter.
- The rotational speed of the rotors relative to the ground is not taken in consideration.
- The lift and the drag are proportional to the square of the speed rotation of the motors.

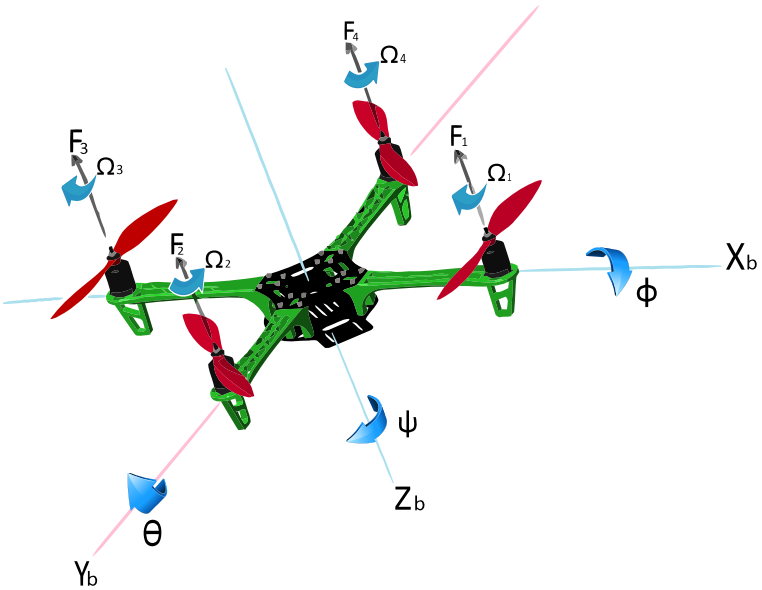

Fig. 1. Quadcopter body frame with principal axis and forces

978-1-5386-2847-8/17/$31.00 ©2017 IEEE

A. Mathematical Modeling

In order to define the quadcopter model we need to identify the system with appropriate coordinate frames. The inertial coordinate frame or earth reference frame is considered as a fixed coordinate at the ground, while the body coordinate frame is attached to the center of the quadcopter and its axis are parallel with the earth reference frame. The transition between the two coordinate frames is assured by introducing three elementary rotation angles ϕ, θ, ψ which are defined in the following way fig.2:

Rotation of $\phi(t)$ around the x-axis (angle: $-\pi/2 \leq \phi \leq \pi/2$)
Rotation of $\theta(t)$ around the y-axis (angle: $-\pi/2 \leq \theta \leq \pi/2$)
Rotation of $\psi(t)$ around the z-axis (angle: $-\pi \leq \psi \leq \pi$)
The rotation matrices are given by :

$$R(x,\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{pmatrix}$$

$$R(y,\theta) = \begin{pmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{pmatrix}$$

$$R(z,\psi) = \begin{pmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

By multiplying the three matrices, we obtain an orthogonal matrix representing the transformation between two frames in equation 1:

$$R = \begin{pmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi) - s(\phi)c(\psi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{pmatrix} \quad (1)$$

1) Euler Lagrange formalism:

Developing the mathematical model using Euler-Lagrange formalism [8] [9] using kinetic and potential energies. The general Lagrange equation is given by the equation 2:

$$\Gamma_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \quad ; \quad \text{With} \quad L = T - V \quad (2)$$

Γ_i: Generalized forces given by non-conservative forces.
q_i: Generalized coordinates.
T: Total kinetic energy.
V: Total potential energy.

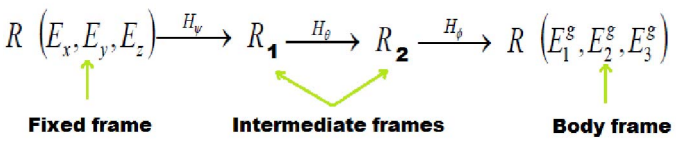

Fig. 2. The transition from the fixed frame to the body frame

2) Expression of Kinetic and Potential Energy :

Let $[\vec{X}\vec{Y}\vec{Z}]$ an orthonormal base constituting a fixed reference frame. If the rigid body rotates three times according to aeronautics angles, then by using equation 1 we have:

$$r_{XYZ} = R(\phi,\theta,\psi)\begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (3)$$

$$\begin{aligned} r_X &= cos(\psi)cos(\theta)x + [cos(\psi)sin(\theta)sin(\phi) - sin(\psi)cos(\phi)]y \\ &\quad + [cos(\psi)sin(\theta)cos(\phi) + sin(\psi)sin(\phi)]z \\ r_Y &= sin(\psi)cos(\theta)x + [sin(\psi)sin(\theta)sin(\phi) + cos(\psi)cos(\phi)]y \\ &\quad + [sin(\psi)sin(\theta)cos(\phi) - sin(\phi)cos(\psi)]z \\ r_Z &= -sin(\theta)x + cos(\theta)sin(\phi)y + cos(\theta)cos(\phi)z \end{aligned}$$

By deriving and simplifying those equations, we obtain the corresponding speed of r_X, r_Y and r_Z which can be represented as :

$$\begin{pmatrix} \nu_X(x,y,z) \\ \nu_Y(x,y,z) \\ \nu_Z(x,y,z) \end{pmatrix} = \begin{pmatrix} \nu_{x_x} & \nu_{x_y} & \nu_{x_z} \\ \nu_{y_x} & \nu_{y_y} & \nu_{y_z} \\ \nu_{z_x} & \nu_{z_y} & \nu_{z_z} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

The squared norm of the speed is calculated by: $\nu^2_{x,y,z} = \nu^2_X + \nu^2_Y + \nu^2_Z$

$$\nu^2_{i=X,Y,Z} = \begin{pmatrix} \nu_{i_x} & \nu_{i_y} & \nu_{i_z} \end{pmatrix} \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix} \begin{pmatrix} \nu_{i_x} \\ \nu_{i_y} \\ \nu_{i_z} \end{pmatrix}$$

By developing the expression and introducing inertial elements, the kinetic energy can be expressed in the following way:

$$\begin{aligned} T &= \frac{1}{2}I_x\left(\dot{\phi} - \dot{\psi}sin(\theta)\right)^2 + \frac{1}{2}I_y\left(\dot{\theta}cos(\phi) + \dot{\psi}sin(\phi)cos(\theta)\right)^2 \\ &\quad + \frac{1}{2}I_z\left(\dot{\theta}sin(\phi) - \dot{\psi}cos(\phi)cos(\theta)\right)^2 \\ I_x &= \frac{1}{2}\int\left(y^2+z^2\right)dm; I_y = \frac{1}{2}\int\left(x^2+z^2\right)dm; \\ I_z &= \frac{1}{2}\int\left(x^2+y^2\right)dm \end{aligned}$$

I_x, I_yandI_z represents the moments of inertia of the system. The potential energy is calculated by the following equation:

$$\begin{aligned} V &= g\int(-sin(\theta)x + sin(\phi)cos(\theta)y + cos(\phi)cos(\theta)z)\,dm \\ V &= \int xdm\,(-gsin(\theta)) + \int ydm\,(gsin(\phi)cos(\theta)) \\ &\quad + \int zdm\,(gcos(\phi)cos(\theta)) \end{aligned}$$

3) Developing Stat Model with Euler-Lagrange:

In order to determine the equations of each angle, we have to exploit the kinetic and potential energy calculated recently,

using euler-lagrange formalism the equation of roll, pitch and yaw are given by:

$$\begin{aligned}\Gamma_\phi &= \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) - \frac{\partial L}{\partial \phi} = \tau_x \\ \Gamma_\theta &= \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau_y \qquad ; \quad \text{With} \quad L = T - V \\ \Gamma_\psi &= \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) - \frac{\partial L}{\partial \psi} = \tau_z\end{aligned}$$

Considering the small angles hypothesis, equations after derivations and simplifications becomes:
$\Gamma_\phi = I_x\dot{\omega_x} + (I_z - I_y)\,\omega_y\omega_z$
$\Gamma_\theta = I_y\dot{\omega_y} + (I_x - I_z)\,\omega_x\omega_z$
$\Gamma_\psi = I_z\dot{\omega_z} + (I_y - I_x)\,\omega_x\omega_y$
ω_i: Represents the angular velocity in the body axes

Expression of non-conservative forces τ_x, τ_y and τ_z:
The lift of the motors created in the direction of the axes x and y, by torques τ_x, τ_y are given by:
$\tau_x = bl\left(\Omega_4^2 - \Omega_2^2\right)$, $\tau_y = bl\left(\Omega_3^2 - \Omega_1^2\right)$
Where b, l and Ω_i Are respectively a constant uniting the thrust and the speed of rotation of the motor, the half wingspan of the quadcopter and the rotor i speed rotation.

The drag of the propellers creates a vertical torque:
$\tau_z = d\left(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2\right)$ where d is a constant connecting the drag and the speed of rotation of a motor.

Gyroscopic effect: as a result of a rotation of the propeller around the x or y axis, a non-intuitive effect appears and must be taken in consideration:

$$\begin{aligned}\tau_x &= I_{rotor}\omega_y\left(+\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4\right) \\ \tau_y &= I_{rotor}\omega_x\left(-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4\right)\end{aligned} \tag{4}$$

Where I_{rotor} represents the rotor inertia ,
and $\Omega_r = (\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3)$
The resulted couples applied in each direction are shown in 5:

$$\begin{aligned}\tau_x &= bl\left(\Omega_4^2 - \Omega_2^2\right) + I_{rotor}\omega_y\left(+\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4\right) \\ \tau_y &= bl\left(\Omega_3^2 - \Omega_1^2\right) + I_{rotor}\omega_x\left(-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4\right) \\ \tau_z &= d\left(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2\right)\end{aligned} \tag{5}$$

The equation which describes the angular movement result from 4 and 5:

$$\begin{aligned}\ddot{\phi} &= \frac{I_{rotor}\dot{\theta}\left(+\Omega_3 + \Omega_1 - \Omega_2 - \Omega_4\right)}{I_x} + \frac{(I_y - I_z)}{I_x}\dot{\theta}\dot{\psi} + \frac{bl\left(\Omega_4^2 - \Omega_2^2\right)}{I_x} \\ \ddot{\theta} &= \frac{I_{rotor}\dot{\phi}\left(-\Omega_3 - \Omega_1 + \Omega_2 + \Omega_4\right)}{I_y} + \frac{(I_z - I_x)}{I_y}\dot{\phi}\dot{\psi} + \frac{bl\left(\Omega_3^2 - \Omega_1^2\right)}{I_y} \\ \ddot{\psi} &= \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\phi} + \frac{d\left(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2\right)}{I_z}\end{aligned} \tag{6}$$

Based on the equation 6, the overall system can be described as follows 7:

$$f(X,U) = \begin{pmatrix} \dot{\phi} \\ a_1\dot{\theta}\dot{\psi} + a_2\dot{\theta}\Omega_r + b_1U_2 \\ \dot{\theta} \\ a_3\dot{\phi}\dot{\psi} - a_4\dot{\phi}\Omega_r + b_2U_3 \\ \dot{\psi} \\ a_5\dot{\theta}\dot{\phi} + b_3U_4 \\ \dot{z} \\ g - cos\phi cos\theta\frac{1}{m}U_1 \\ \dot{x} \\ (cos\phi sin\theta cos\psi + sin\phi sin\psi)\frac{1}{m}U_1 \\ \dot{y} \\ (cos\phi sin\theta sin\psi - sin\phi cos\psi)\frac{1}{m}U_1 \end{pmatrix} \tag{7}$$

$$\begin{cases} U_1 = b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ U_2 = b\left(\Omega_4^2 - \Omega_2^2\right) \\ U_3 = b\left(\Omega_1^2 - \Omega_3^2\right) \\ U_4 = d\left(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2\right) \end{cases}$$

$$\begin{aligned} a_1 &= (I_{yy} - I_{zz})/I_{xx} & b_1 &= l/I_{xx} \\ a_2 &= I_{rotor}/I_{xx} & & \\ a_3 &= (I_{zz} - I_{xx}/I_{yy} & b_2 &= l/I_{yy} \\ a_4 &= I_{rotor}/I_{yy} & & \\ a_5 &= (I_{xx} - I_{yy})/I_{zz} & b_3 &= l/I_{zz} \\ x_1 &= \phi & x_7 &= z \\ x_2 &= \dot{x_1} = \dot{\phi} & x_8 &= \dot{x_7} = \dot{z} \\ x_3 &= \theta & x_9 &= x \\ x_4 &= \dot{x_3} = \dot{\theta} & x_{10} &= \dot{x_9} = \dot{x} \\ x_5 &= \psi & x_{11} &= y \\ x_6 &= \dot{x_5} = \dot{\psi} & x_{12} &= \dot{x_{11}} = \dot{y} \end{aligned}$$

Stat and control vector from the model 7 are represented as follows:

$$X = [\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\ z\ \dot{z}\ x\ \dot{x}\ y\ \dot{y}\ ]^T$$
$$U = [\ U_1\ U_2\ U_3\ U_4\ ]^T$$

B. Control Design

In this section the paper presents the control techniques used to design a position and attitude controller for a quadcopter model and stabilized it at quasi-stationary conditions. Two control techniques are implemented, the PID controller is used in first place, which consists of two cascaded loops: an outer-loop for controlling position and altitude, which takes as reference input the desired values of x, y and z and produces U_1 as an altitude control input. Thus the position controller output generates the roll, pitch and yaw reference values for the inner control loop (attitude control) which takes over the stabilization of the drone by feeding U_2, U_3 and U_4 as an attitude control inputs for the quadcopter plant model. A backstepping controller is used secondly instead of the inner PID controller, the rest of the system is maintained unchanged.

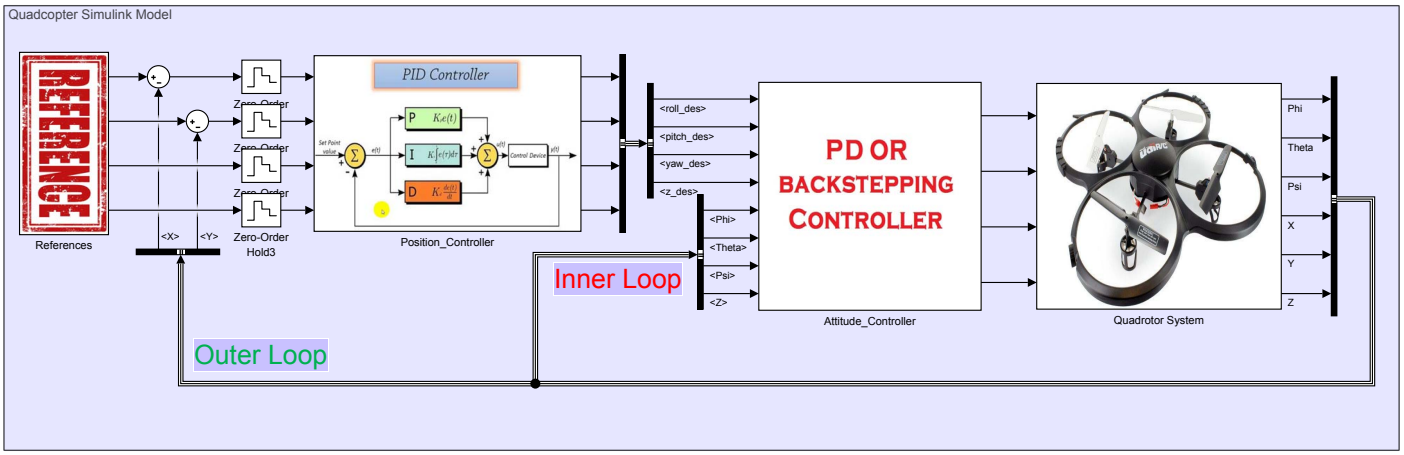

Fig. 3. Quadcopter plant model with the controller form Simulink environment

1) PID Control Design Approach:

Proportional, integral and derivative controller also known as PID controllers are commonly and widely used in multiple control design applications, this technique calculates an error value $e(t)$ as the difference between a desired set-point and a measured process variable and applies a correction based on proportional, integral, and derivative terms. This approach is simple to tune and to implement, but in the flip-side the PID controllers are naturally linear. They can achieve a decent performance in some nonlinear applications, however in many cases where the system is heavily non linear and variate rapidly, the use of this architecture without linearizing the system around an operating point, seems to be unsuitable and can result a poor performance in return.

As the figure 4 shows, the structure is composed from two cascaded controller loop. A position and altitude controller receive as inputs the desired value of x_r, y_r and z_r calculate the errors: $e_1 = x_r - x$, $e_2 = y_r - y$ and $e_3 = z_r - z$. The altitude controller result a control input U_1 while the position controller output the tree reference angles (ϕ_r, θ_r and ψ_r) needed for the inner loop (attitude control) which calculate the errors and adjust the stat of the plant model by feeding it with tree control inputs U_2, U_3 and U_4. The parameters of each controller are tuned by defining the desired performance profile using a specific tool "PID Tuner" built in Matlab/Simulink.

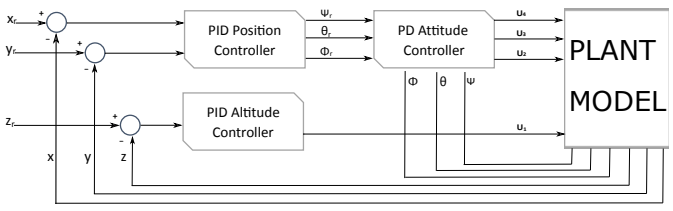

Fig. 4. Cascaded loop PID structure

2) Backstepping Control Design Approach:

The Lyapunov theory has been for a long period of time a very important tool in linear and nonlinear applications as well, however the use of this technique had been hampered by the difficulties in finding a Lyapunov function for a given system. The invention of constructive tools for nonlinear control design based on Lyapunov theory, such as Backstepping, was thus received with open arms by the control community. Here, a control law stabilizing the system is derived with a Lyapunov function to prove stability. The principle of this theory is to establish in a constructive way the control law of the nonlinear system by considering some stat variables as virtual commands and designing intermediate control laws.

Considering the stat representation in equation 7, the command inputs U_2, U_3 and U_4 can be calculated using this approach as follow:

- Roll, pitch and yaw commands calculation:

Let consider the error: $\varepsilon_1 = x_{1d} - x_1$
the derivative of the error is: $\dot{\varepsilon_1} = \dot{x_{1d}} - \dot{x_1} = \dot{x_{1d}} - x_2$
the new virtual command is considered as $\alpha_1 = (x_2)_d$
using the Lyapunov theorem by considering the Lyapunov function ε_1 positive definite and its time derivative negative:
$V_1(\varepsilon_1) = \frac{1}{2}\varepsilon_1^2 \quad \Rightarrow \quad \dot{V}_1(\varepsilon_1) = \varepsilon_1\dot{\varepsilon_1}$
$\dot{V}_1(\varepsilon_1) = \varepsilon_1[\dot{x_{1d}} - x_2] < 0$

In order to satisfy the negativity of the equation, we have to consider $[\dot{x_{1d}} - x_2] = -c_1\varepsilon_1$ so the equation becomes:
$\dot{V}_1(\varepsilon_1) = -c_1\dot{\varepsilon_1}^2 \quad \Rightarrow \quad \alpha_1 = \dot{x_{1d}} + c_1\dot{\varepsilon_1}$

Let consider the error: $\varepsilon_2 = x_2 - \alpha_1 \quad \Rightarrow \quad \dot{\varepsilon_2} = \dot{x_2} - \dot{\alpha_1}$
$\Rightarrow \quad \dot{\varepsilon_2} = a_1x_4x_6 + a_2x_4\Omega_r + b_1U_2 - \dot{\alpha_1}$

The Lyapunov function of $(\varepsilon_1, \varepsilon_2)$ is given by the equation:
$V_2(\varepsilon_1, \varepsilon_2) = V_1(\varepsilon_1) + \frac{1}{2}\varepsilon_2^2$

$$\dot{V}_2(\varepsilon_1,\varepsilon_2)=\dot{V}_1(\varepsilon_1)+\varepsilon_2\dot{\varepsilon_2}<0$$
$$\dot{V}_1(\varepsilon_1)=\varepsilon_1[\dot{x_{1d}}-x_2]=\varepsilon_1[\alpha_1-c_1\varepsilon_1-x_2]$$
$$\Rightarrow \quad \dot{V}_1(\varepsilon_1)=-c_1\varepsilon_1^2-\varepsilon_1\varepsilon_2$$
$$\dot{V}_2(\varepsilon_1,\varepsilon_2)=-c_1\varepsilon_1^2-\varepsilon_1\varepsilon_2+\varepsilon_2[a_1x_4x_6+a_2x_4\Omega_r+b_1U_2-\dot{\alpha_1}]$$
$$\dot{V}_2(\varepsilon_1,\varepsilon_2)=-c_1\varepsilon_1^2+\varepsilon_2[-\varepsilon_1+a_1x_4x_6+a_2x_4\Omega_r+b_1U_2-\dot{\alpha_1}]$$

To satisfy the negativity of the equation, we have to consider: $[-\varepsilon_1+a_1x_4x_6+a_2x_4\Omega_r+b_1U_2-\dot{\alpha_1}]=-c_2\varepsilon_2$
$$\Rightarrow \quad U_2=\frac{1}{b_1}[\varepsilon_1-a_1x_4x_6-a_2x_4\Omega_r-c_2\varepsilon_2+\dot{\alpha_1}]$$

We know that: $\alpha_1=\dot{x_{1d}}+c_1\varepsilon_1 \Rightarrow \quad \dot{\alpha_1}=\ddot{x_{1d}}+c_1\dot{\varepsilon_1}$
$$\Rightarrow \quad \dot{\alpha_1}=\ddot{x_{1d}}+c_1[\dot{x_{1d}}-x_2]=\ddot{x_{1d}}+c_1[\alpha_1-c_1\varepsilon_1-x_2]$$
$$\Rightarrow \quad \dot{\alpha_1}=\ddot{x_{1d}}-c_1[c_1\varepsilon_1+\varepsilon_2]$$

The control input U_2 for Roll command is obtained with the assumption of $\ddot{x_{1d}}=0$ and $\dot{V}_2(\varepsilon_1,\varepsilon_2)<0$:
$$\Rightarrow \quad U_2=\frac{1}{b_1}[\varepsilon_1-a_1x_4x_6-a_2x_4\Omega_r-c_2\varepsilon_2-c_1(c_1\varepsilon_1+\varepsilon_2)]$$

The same structure was used to calculate Pitch and Yaw commands with the consideration of:
$$\varepsilon_3=x_{3d}-x_3$$
$$\varepsilon_4=x_4-\dot{x_{3d}}-c_3\varepsilon_3$$
$$\varepsilon_5=x_{5d}-x_5$$
$$\varepsilon_6=x_6-\dot{x_{5d}}-c_5\varepsilon_5$$

$$\Rightarrow \quad U_3=\frac{1}{b_2}[\varepsilon_3-a_3x_2x_6-a_4x_2\Omega_r-c_4\varepsilon_4-c_3(c_3\varepsilon_3+\varepsilon_4)]$$
$$\Rightarrow \quad U_4=\frac{1}{b_3}[\varepsilon_5-a_5x_2x_4-c_6\varepsilon_6-c_5(c_5\varepsilon_5+\varepsilon_6)]$$

The outer loop command is maintained as the previous section with a PID controller of position and altitude.

III. Physical Modeling and PIL Simulation

A. Physical Modeling

Physical modeling or simscape multibody simulation environment for 3d mechanical systems is considered one of the strongest tools which simplify the build up of dynamic systems and integrates with others physical modeling libraries in Matlab/Simscape environment, making dynamic system modeling much easier without the need of deriving equations which differs from the standard Simulink modeling approach and is particularly suited to simulate systems that consist of real physical components [10]. Unlike Simulink input/output connection based design, the component models in simscape have physical connections which enable modeling physical systems by assembling a physical schematic. Connection between components enable a bidirectional flow of energy which means that simscape can determine the system level of equations for a given model.

Quadcopter model: Modeling the parts and assembly was carried out under SolidWork 3D design system. This platform guarantees a favorable environment for volume 3D modeling and offers several advantages by decreasing design time and integrating by default the different characteristics of the material, like inertia, mass, rigidity, etc.

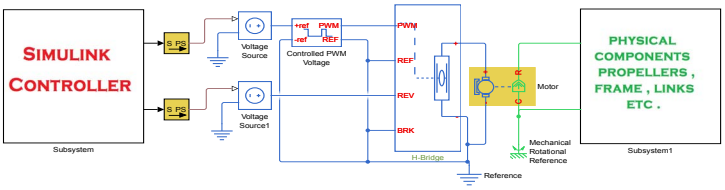

Fig. 5. Linkage between the physical model and the controller

The 3D CAD model was imported to Simscape Multibody(formerly SimMechanics) and linked to the controller part modeled in Simulink Fig.5.

B. Processor-In-the-Loop (PIL) Simulation

The design of any controller requires the mathematical development of the model in question followed by adjustments of its parameters which depend on the structure and nature of the plant model. However, the real implementation of this controller takes into account several variables that can affect the performance of the system, meaning that the deployed code will not necessarily behave as the simulated one. The PIL technique makes it possible to verify if the implemented c/c++ code on a processor target correspond well to the result obtained during the simulation.
PIL requires drivers to communicate the computer platform with the aimed hardware. The resulting object code generated in the PC links with other test-management functionality and is then downloaded, typically to an off-the-shelf evaluation board with the target processor. The simulation tool, running on the PC machine, then communicates with the downloaded software, typically via a serial communication link.

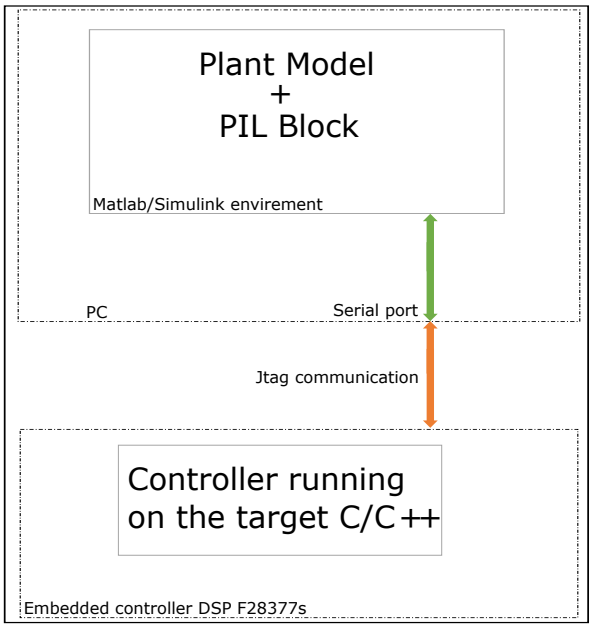

Fig. 6. Processor-In-the-Loop (PIL) schematic

IV. Experiments Results

This section discusses some simulation results of the 6DOF quadcopter model and the process of the verification under the real C2000 processor. All the shown figures were taken from Matlab/Simulink envirement.

TABLE I
QUADCOPTER PARAMETERS

Parameters	Value
I_x	$7.2 \times 10^{-3}\ kg.m^2$
I_y	$7.2 \times 10^{-3}\ kg.m^2$
I_z	$1.1 \times 10^{-2}\ kg.m^2$
J_r	$6.3 \times 10^{-5}\ kg.m^2$
l	$0.2\ m$
g	$9.81\ m/s^2$
m	$0.55\ kg$
b	$2.93 \times 10^{-5}\ kg.m$
d	$7.2 \times 10^{-7}\ kg.m^2$

A. PID Control Approach Results

The parameters of the controller are shown in the table:

Loop	Controller	Parameters	Value
Inner Loop	Attitude controller	Kp_ϕ	280.55
		Kd_ϕ	6.15
		Kp_θ	53.39
		Kd_θ	1.43
		Kp_ψ	25.52
		Kd_ψ	0.95
Outer Loop	Position controller	Kp_x	10
		Kd_x	3
		Kp_y	10
		Kd_y	3
	Altitude controller	Kp_z	2.83×10^5
		Kd_z	6.15×10^3

TABLE II
PARAMETERS OF THE CONTROLLER

The desired trajectory of the quadcopter is as follow:

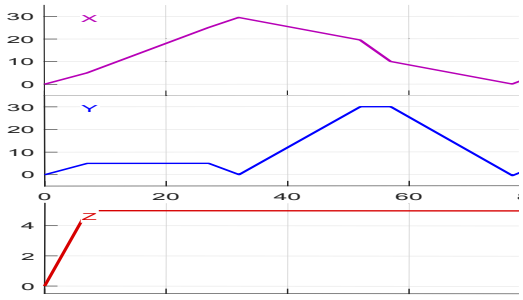

Fig. 7. Desired trajectory

Results of the simulation using Table II parameters:

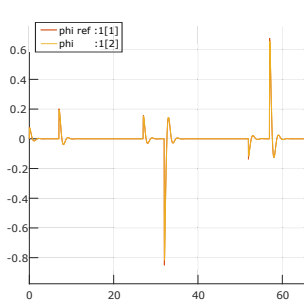

Fig. 8. Roll command

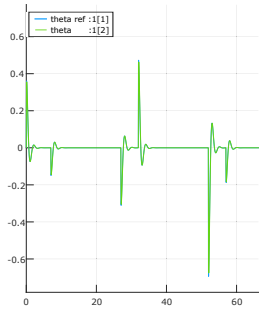

Fig. 9. Pitch command

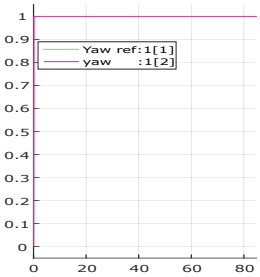

Fig. 10. Forced yaw to 1

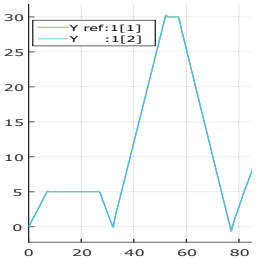

Fig. 11. X command

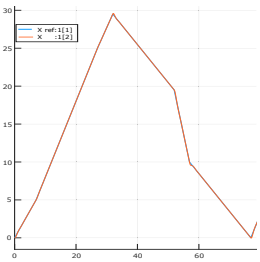

Fig. 12. Y command

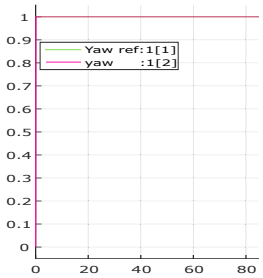

Fig. 13. Z command

B. Backsteping Controller Approach Results

The parameters of the Backsteping attitude controller are in the table:

TABLE III
BACKSTEPING CONTROLLER PARAMETERS

Parameters	Value
c_1	27.79
c_2	285.74
c_3	4.70
c_4	45.02
c_5	4.55
c_6	21.67

The results obtained from the backsteping control technique are quite similar to the results from the PID approach technique which gives a very decent performance according to the recent shown result. The only noticeable difference comes from the overshooting rate and stabilization

time from some hard profiles where many variations occur in a short period of time , in fact with the nonlinear controller we were able to decrease the overshooting rate even further more without pushing the system to the instability zone.
The figure 14 shows a plotting of XY simulation for the a given trajectory design for each PID and Backstepping controllers:

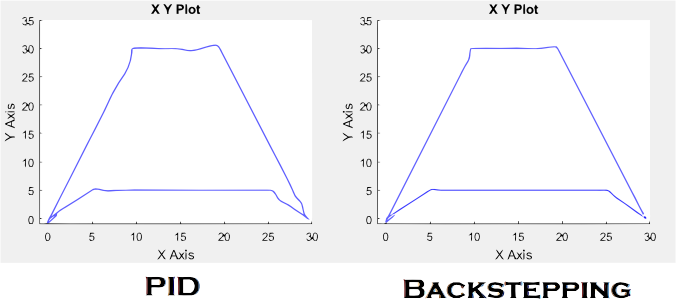

Fig. 14. XY trajectory plotting of PID and Backstepping techniques

C. PIL Execution Time and Profiling

The PIL verification process was carried out with Simulink embedded coder for TI C2000 processors, the running code in the target processor F28377s matched perfectly the results of the simulation. The execution profiling time shown in figure 15 gives some details about the execution time profile.

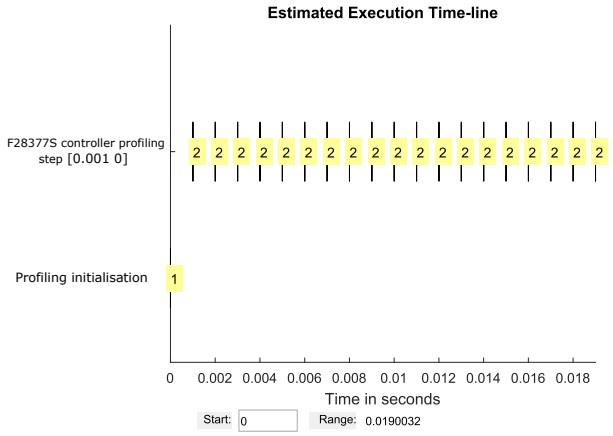

Fig. 15. Executing time in hardware target

V. Conclusion

This paper shows a methodological approach of designing a six degree of freedom quadcopter model based on Euler-Lagrange theory. The controller part was chosen in first place to be a simple PID linear controller for each inner and outer loop, the second one was a nonlinear backstepping controller. The results of both controllers were quite similar, this is due to the simplifications and the assumptions taken in consideration when designing the mathematical model. In fact, to simulate the reality more additional effects and uncertainties should be taken into account leading to a more complicated system, thus the need to a robust controller technique based on nonlinear controller. The proposed verification process using a DSP target hardware was a helpful step in order to make sure that the generated code work properly in the hardware target, since the future work intended to build a real quadcopter system with a standalone controller and to perform a stable fly in such disturbed and uncertain environment.

Acknowledgment

The author would like to express his sincere appreciation and acknowledgment as regards the financial support of the national center for scientific and technical research in morocco (CNRST).

References

[1] P. N. Patel, M. A. Patel, R. M. Faldu, and Y. R. Dave, "Quadcopter for agricultural surveillance," *Advance in Electronic and Electric Engineering*, vol. 3, no. 4, pp. 427–432, 2013.
[2] J. A. Paredes, J. Acevedo, H. Mogrovejo, J. Villalta, and R. Furukawa, "Quadcopter design for medicine transportation in the peruvian amazon rainforest," in *Electronics, Electrical Engineering and Computing (INTERCON), 2016 IEEE XXIII International Congress on*. IEEE, 2016, pp. 1–6.
[3] R. Williams, B. Konev, and F. Coenen, "Multi-agent environment exploration with ar. drones," in *Conference Towards Autonomous Robotic Systems*. Springer, 2014, pp. 60–71.
[4] M. Khan, "Quadcopter flight dynamics," *International Journal of Science and Technology Research*, pp. 130–135, 2014.
[5] P. E. Pounds and A. M. Dollar, "Stability of helicopters in compliant contact under pd-pid control," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1472–1486, 2014.
[6] J. Wang, T. Raffler, and F. Holzapfel, "Nonlinear position control approaches for quadcopters using a novel state representation," in *AIAA Guidance, Navigation and Control Conference*, 2012.
[7] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 3255–3260.
[8] R. Ortega, J. A. L. Perez, P. J. Nicklasson, and H. Sira-Ramirez, *Passivity-based control of Euler-Lagrange systems: mechanical, electrical and electromechanical applications*. Springer Science & Business Media, 2013.
[9] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, 2011.
[10] R. F. Gordon, P. Kumar, and R. Ruff, "Simulating quadrotor dynamics using imported cad data," *AIAA Modeling and Simulation Technologies (MST) Conference Boston*, 2013.