Quadcopter Control System
Modelling and Implementation

Daniel Gheorghiță, Ionuţ Vîntu, Letiţia Mirea, Cătălin Brăescu*
Automatic Control and Applied Informatics
Faculty of Automatic Control and Computer Engineering
Technical University of Iaşi
Iaşi, România
*e-mail: daniel0392@gmail.com

Abstract **— The paper presents two different types of approach for mathematical modeling of quadcopter kinematics and dynamics. The first one is based on the equations of classical mechanics and the other one is derived from Denavit-Hartenberg formalism and Lagrangian mechanics. The obtained models were used to design the control of the quadcopter motion on one rotation axis. The paper also offers some details of the physical and software implementation of the system on which tests have been made.**

Keywords—quadcopter; robotics; mathematical modelling; embedded system; control engineering

I. INTRODUCTION

The paper presents an educational method for students with undergraduate knowledge of mechanical, computer and electrical engineering on how to approach a modern control problem, i.e. the control of a quadcopter system. Its interdisciplinarity recommends the problem suitable for a team assignment for a group of students with different engineering skills and interests. Due to the complexity of the addressed problem, the paper is focused on obtaining dynamical models for the quadcopter system, while simulation, control and a graphic user interface are only briefly discussed.

The study of the quadcopter control problem stalled until recently because the control of four independent motor-based propulsion system was almost impossible without advanced electronic devices. Only in the last few decades these devices (microprocessors, microcontrollers, FPGAs, electronic stability controllers etc.) became more advanced, flexible, faster and cheaper. Low prices and high performance of modern microcontrollers made quadcopters available for a large scale of applications: military, commercial products and for hobby purposes.

The control of a quadcopter is an interesting problem because of its complexity. Six degrees of freedom (three rotational axes and three translational axes) and only four inputs (angular velocity of propellers) give the system the property of being underactuated. Nowadays, the quadcopter is not the only type of Unmanned Aerial Vehicle (UAV) with parallel rotors: tricopters, hexacopters and octocoptersare also available. In spite of their different structures, the control problem for all of them is rather similar. All have the same number of axes to control and are underactuated even though some have more than six inputs. This is because all those inputs can directly control only the three rotational axes and not the translational axes. Also, the dynamics on which this type of UAV functions offers flexibility in movement and robustness towards propulsion malfunctions. As an example, control algorithms can be implemented so that an UAV can hold its stability even if half of the propellers that control one axis of rotation lose their functionality. On the other hand, being an aerial vehicle, frictions of the chassis are negligible and the control algorithm has to take care of the damping [1].

Considering all these reasons, many research centers focus their efforts on the development of control strategies for stability and path planning. Some of these centers are Technical University of Munich [2], Technical University of Zurich [3], Ecole Polytechnique Federale de Lausanne [4].

In the following sections, a mathematical model will be introduced based on the dynamic equations of the quadcopter. Furthermore, a kinematic model based of the Denavit-Hartenberg formalism will be presented followed by a dynamic model yielded by applying Lagrangian mechanics. A MATLAB simulation and its results using a simple PID controller will be shown. Also, in short terms, the physical and software systems will be presented.

II. MATHEMATICAL MODELLING

A. *Quadcopter Dynamics*

The quadcopter's dynamics is modeled with respect to two reference systems: the inertial one which is related to the Earth and the quadcopter itself (body frame), reference systems presented in Figure 1. The second one is attached to the quadcopter chassis and can be separated into translational – parallel with the fixed inertial coordinate system – and rotational – same origin as the translational one, but portraying the chassis orientation.

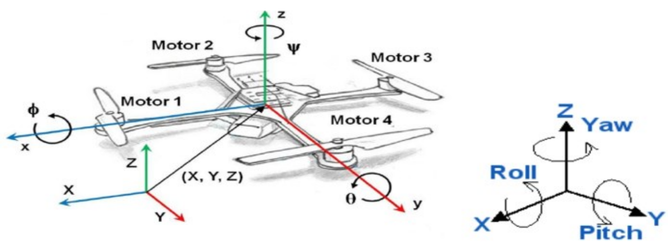

Fig. 1. Coordinate systems and Roll, Pitch, Yaw angles [5]

The orientation of the quadcopter is given by the Roll (Φ), Pitch (θ) and Yaw (ψ) angles defined as rotations with respect to the X axis, Y axis and respectively, Z axis. The rotational matrix that defines the relation between the quadcopter system and Earth coordinate system has the following form:

$$R=\begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi - S_\psi C_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi - C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \quad (1)$$

where C_x and S_x are notations of cos(x) and respectively, sin(x).

The propulsion is the force with which the quadcopter propellers push the air downwards for gaining upward thrust. It is defined as:

$$T = k \cdot \omega^2 \quad (2)$$

where$T$ is the thrust generated by a single motor (propeller), $k$is the thrust constant and $\omega$is the angular velocity of the motor shaft.

By summating of all propellers' thrust, the expression of the total thrust in the body frame is as follows:

$$T_B = \sum_{i=1}^{4} T_i = k \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} \omega_i^2 \end{bmatrix} \quad (3)$$

Another force affecting the quadcopter is linear friction, which acts on each axis, being proportional with the corresponding linear velocity:

$$F_D = \begin{bmatrix} -k_d \cdot \dot{x} \\ -k_d \cdot \dot{y} \\ -k_d \cdot \dot{z} \end{bmatrix} \quad (4)$$

The rotational torque inflicted on the quadcopter is proportional with the squared angular velocity of the motors:

$$\tau_\psi = (-1)^{i+1} \cdot b \cdot \omega_i^2 \quad (5)$$

where motor i has a positive influence if the rotation is clockwise and negative if the rotation is counterclockwise. Having this in mind the formula for the torque on the Z axis is:

$$\tau_\psi = b \cdot (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \quad (6)$$

The torque is the cross product between the applied force and position vector of the application point. The component of torque on the Y axis has the following expression, considering motors 1 and 3 are positioned on the X axis:

$$\tau_\theta = \sum r \times T = L \cdot (-k \cdot \omega_1^2 + k \cdot \omega_3^2) = L \cdot k \cdot (-\omega_1^2 + \omega_3^2) \quad (7)$$

and motors 2 and 4 on the Y axis:

$$\tau_\theta = L \cdot k \cdot (-\omega_2^2 + \omega_4^2) \quad (8)$$

Finally, the total torque in the body frame is:

$$\tau_B = \begin{bmatrix} L \cdot k \cdot (-\omega_1^2 + \omega_3^2) \\ L \cdot k \cdot (-\omega_2^2 + \omega_4^2) \\ b \cdot (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{bmatrix} \quad (9)$$

The modeling has not taken into consideration nonlinearities such as wind, propeller flexibility or nonlinear frictions.

In the Earth coordinate system, quadcopter acceleration is a superposition of propulsion, gravity force and linear frictions:

$$m\ddot{x} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \cdot T_B + F_D \quad (10)$$

where x is the position vector, $\ddot{x}$ is the linear acceleration, m is the quadcopter mass, g is the gravitational acceleration, F_D is the linear friction, T_B is the propulsion and R is the rotation matrix defined earlier.

The state equation of the angular velocity has the expression:

$$I \cdot \dot{\omega} + \omega \times (I \cdot \omega) = \tau \quad (11)$$

Where ω is the angular velocity, τ is the total rotational torque, I is the inertia matrix of the system (assuming the structure is symmetrical and the arms are alongside X and Y axis, the inertia matrix is diagonal:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (12)$$

and $I_{xx} = I_{yy}$ [5]. From equations (11) and (12), the angular velocity is determined:

$$\dot{\omega} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = I^{-1} \cdot (\tau - \omega \times (I \cdot \omega)) \quad (13)$$

The dependency between angular velocity in the body frame and the derivatives of Roll, Pitch, Yaw angles is:

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & S_\theta \\ 0 & C_\phi & C_\theta \cdot S_\phi \\ 0 & -S_\phi & C_\theta \cdot C_\phi \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (14)$$

In this way all the state equations needed for the dynamic modeling of the quadcopter have been defined.

B. Kinematic model using Denavit-Hartenberg

The same coordinate systems indicated in section, sub-section A will be used.

For determining the correspondence between independent rotations on the three axes and the orientation of quadcopter with respect to the Earth coordinate system, the rotation matrix has to be calculated. The total propulsion generated by all propellers always has the direction of negative Z axis of the body frame. Using the rotation matrix, the total thrust can be decomposed with respect to the XYZ axis of a coordinate system parallel with the Earth system. By doing so, linear accelerations can be computed and also linear movement. Knowing this, we can assume that the control of a quadcopter can be resumed to controlling angles and rotations.

The kinematic model of a robot is a set of equations that define the dependence between a base point and a final point considering the states of joints (angular and linear) and lengths of segments. In other words, kinematics is the study of kinetic chain movement neglecting forces and masses. [7]

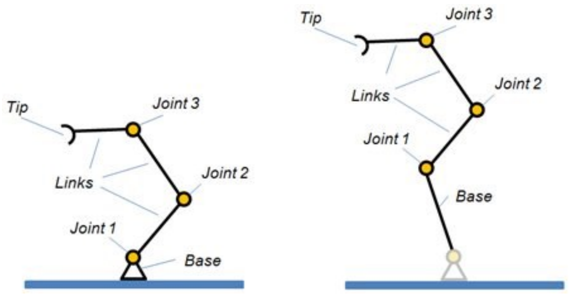

Fig. 2. Open kinematic chain [6]

In Figure 2 two examples of open kinematic chains can be seen. The difference between an open kinematic chain and a closed kinematic chain is that the first one has an end point that can be associated with an effector (tool, gripper).

For the development of the kinematic model the Denavit Hartenberg (D-H) formalism rules are applied [7]. Eventhough D-H parameters are applied to joints with a single Degree of Freedom (DOF), the rules can be applied to joints with multiple DOF. It can be done by considering a chain of single DOF joints and the distance between them equal to 0. The quadcopter has three axes of rotation, so 3 rotational DOF. Therefore, the body movement with respect to the Earth coordinate system (from the rotation point of view) is similar to a 3 DOF rotational joint.

The D-H parameters obtained using this method are expressed in the table:

Table 1. D-H parameters

di	ai	θi	αi
0	0	Pitch(q1)	-90˚
0	0	Roll(q2)	90˚
0	0	Yaw(q3)	0˚

From Table 1, considering [7], the rotation matrix, $R=[r_{ij}]_{i,j=\overline{1,3}}$ is obtained, having the elements:

$$
\begin{aligned}
r_{11} &= C(q_1)+C(q_2)+C(q_3)-S(q_1)+S(q_3);\\
r_{12} &= S(q_1)+C(q_2)+C(q_3)+C(q_1)+S(q_3);\\
r_{13} &= -S(q_2)+C(q_3)\\
r_{21} &= -C(q_1)+C(q_2)+S(q_3)-S(q_1)-C(q_3);\\
r_{22} &= -S(q_1)+C(q_2)+S(q_3)+C(q_1)+C(q_3);\\
r_{23} &= S(q_2)+S(q_3)\\
r_{31} &= C(q_1)+S(q_2);\\
r_{32} &= S(q_1)+S(q_2);\\
r_{33} &= C(q_2).
\end{aligned}
\tag{15}
$$

where C(x) and S(x) are notations of sin(x) and cos(x).

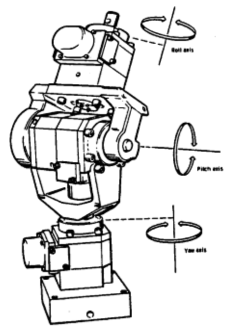

Fig. 3.Three consecutive rotation joints

Figure 3 shows an open kinematic chain with three consecutive rotation joints. The rotational kinematic model of the quadcopter will be associated with a similar structure.

C. Dynamic model using Euler-Lagrange method

The dynamic model is obtained by taking into consideration the rotational and linear movement with respect to the exterior forces (propulsion, rotational torques) and linear or rotational inertia. For the determination of the dynamic model, Euler-Lagrange equations [8] have been used. This method makes use of the expressions of energy in a system and therefore this approach is close to the study of systems' engineering. The method is based on the idea that any mechanical system in free motion (under the effect of gravitational force) will follow the trajectory that will ensure the minimum energy loss[9].

Lagrange introduced the formula that defines the dynamics of a system with respect to the kinetic (K) and potential (P) energy:

$$L(q)=K(q,q')-P(q) \tag{16}$$

Because a minimum is involved, the problem becomes an optimization problem. The solution is given by the Euler-Lagrange second keen equation:

free movement: $$\frac{d}{dt}\left(\frac{\partial L}{\partial q}\right)-\frac{\partial L}{\partial q}=0 \qquad (17)$$

with exterior torque: $$\tau\frac{d}{dt}\left(\frac{\partial L}{\partial q}\right)-\frac{\partial L}{\partial q}=\tau \qquad (18)$$

In the case of an robot with *n* DOF, the model will need n equations similar to (17) and (18).

The final form of the dynamic model is:

$$M(q)\times q^{(2)}+C(q,q')\times q'+G=\tau \qquad (19)$$

where M is the mass matrix of the system and depends on the q vector (generalized coordinates) and models the inertial torques, C is a matrix dependent of masses, positions and velocities in the system and models Coriolis and centrifugal torques and the G matrix is associated to the gravitational torques.

Starting from the scalar expression of kinematic energy:

$$K=\frac{1}{2}\sum_{i=1}^{n}m_i v_i^2 \qquad (20)$$

where $v_i=\omega_{0,i}\times x\times d_{0,i}$ (linear velocity of *i* point mass is the cross product between angular velocity and the distance vector from the axis of rotation). By equalizing with the matrix form, where q_i is the angle of rotation with respect to the *i* axis and q'_i is its derivative:

$$K=\frac{1}{2}\begin{bmatrix}q'_x & q'_y & q'_z\end{bmatrix}\cdot\begin{bmatrix}m_{11} & m_{12} & m_{13}\\ m_{21} & m_{22} & m_{23}\\ m_{31} & m_{32} & m_{33}\end{bmatrix}\cdot\begin{bmatrix}q'_x\\ q'_y\\ q'_z\end{bmatrix} \qquad (21)$$

will be obtained the components of M matrix, where m is the mass of a motor, and l is the length of an arm (distance between center of mass of the body and motor center of mass):

$$\begin{aligned}
&m_{11}=2ml^2;\quad m_{12}=0;\quad m_{13}=4ml^2\\
&m_{21}=0;\quad m_{22}=2ml^2;\quad m_{23}=2ml^2S(q_y)C(q_z)\\
&m_{31}=2ml^2S(q_x)S(q_z);\quad m_{32}=2ml^2S(q_y)S(q_z)\\
&m_{33}=4ml^2
\end{aligned} \qquad (22)$$

By equalizing (20) and (21) relations the components of the C matrix are obtained:

$$\begin{aligned}
&c_{11}=2ml^2q'_zS(q_z)C(q_x);\quad c_{12}=0\\
&c_{13}=2ml^2[q'_zS(q_x)C(q_z)+q'_xC(q_x)S(q_z)]\\
&c_{21}=0;\quad c_{22}=2ml^2q'_zS(q_z)C(q_y)\\
&c_{23}=2ml^2[q'_yS(q_z)C(q_y)+q'_zC(q_z)S(q_y)]\\
&c_{31}=2ml^2[q'_zS(q_x)C(q_z)+q'_xC(q_x)S(q_z)]\\
&c_{32}=2ml^2[q'_zS(q_y)C(q_z)+q'_yC(q_y)S(q_z)]\\
&c_{33}=2ml^2[q'_yS(q_y)C(q_z)+q'_xC(q_z)S(q_x)]
\end{aligned} \qquad (23)$$

Finally, from the angular point of view, G matrix has null components. This is due to considering the chassis perfectly simetrical with respect to the X,Y, Z axis of the body frame. Thus, the gravitational torques are null for angular movements in the body frame.

III. IMPLEMENTATION

A. *Physical system*

The quadcopter consists of a chassis with four arms perpendicular two by two consecutive, four BLDC motors Roxxy 2827-34 with 12x4.5 propellers, controlled by Modelcraft SP-BLC-22T ESCs. At maximum speed, one motor equipped with the mentioned propeller can generate a thrust of almost 70 Newtons. For control and data acquisition a STM32 F3Discovery development board is used.

The board is incorporated with an ARM microprocessor with 32 bit Cortex-M4 core, working at a frequency of 72MHz, DSP and FPU modules, IMU sensors (gyroscopic, accelerometer and compass) on all three axes of the body frame. An external HC-SR04 ultrasonic distance sensor is used for height approximation. For communication with the Graphical User Interface (GUI) packets are transmitted and received through the SPI interface to and from an NRF24L01 wireless module (2.4GHz).

Another wireless module is attached to an Arduino UNO R3 board. The Arduino board is used as a driver between the NRF24L01 module and USB interface connected to the MATLAB serial drivers. A MATLAB script processes the data and prints it on the GUI. It also packs the data that has to be sent to the F3Discovery board.

B. *Software system*

For an easy time period management (sampling time, synchronizations, periodical data transmission), the FreeRTOS real-time operating system has been installed. In the current development stage, three main tasks are implemented with well defined functions. Task1 is responsible with processing data received from the user and changing state variables accordingly (every 0.5 seconds). Task2 receives raw data from the wireless module, sensors every 10ms (sampling time enough to respect Shannon’s Theorem [10] and Valvano Postulate [11]). Task3, depending on the functioning mode, commands each motor.

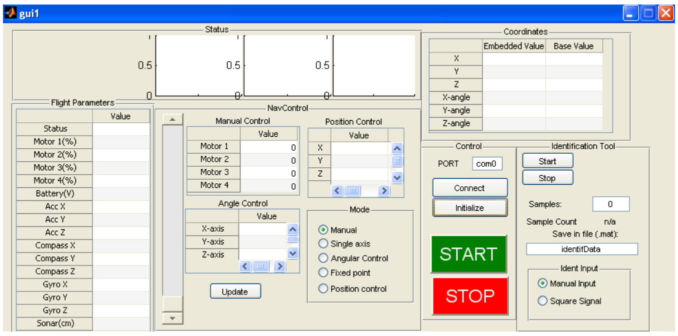

Fig. 4.MATLAB GUI

The implemented functioning modes are manual (the used decides a 0-100 value for each motor and presses "Update"), single axis control (the user specifies a reference for the desired value of X axis angle) and square signal with 1 second period useful for system dynamics identification. Task2 is also responsible for angle value computation using gyroscopic and accelerometer data and combining them through the complementary filter [12]. These parameters can be specified in the GUI which can be seen in Figure 4.

Before the real-time scheduler starts (for FreeRTOS), the peripheral devices are initialized (GPIO, ADC, timers, PWM module, SPI and USART interface, sensors) and required calibrations are performed (for computing of sensors offsets).

As an active safety feature, the GUI independently transmits every 5 data sets a message with the purpose of connection check. Task1 from F3Discovery checks if not too long time has passed since the last connect check message. If the last check message was received a long time before the current time, all motors are shutdown and their command signals are forced to 0. By doing so, the system is safe if the user loses control of the quadcopter (connection lost) or for any reason the sensors stop transmiting data.

IV. SIMULATION AND CONTROL

The simulation of the dynamics of a quadcopter is done by using a MATLAB s-function script that has as inputs the squared angular velocities. The Simulink system overview and the Simulink subsystem are presented in Figure 5 and Figure 7, respectively.

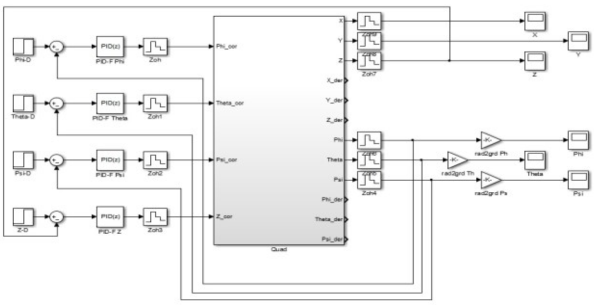

Fig. 5. Simulink system overview

The quadcopter dynamics areimplemented in the s-function block Quad. The roll, pitch, yaw corrections are processed in the Control Mixing block, presented in Figure 6, into motor commands.

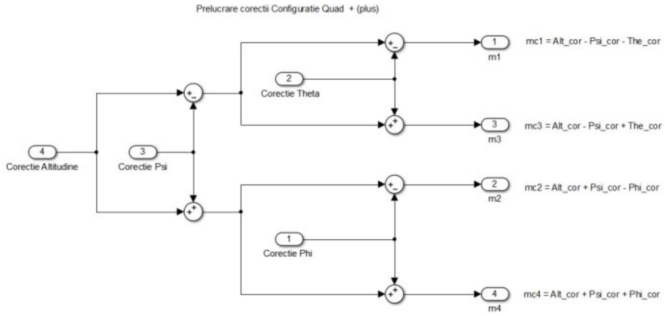

Fig. 6. Control mixing - Command generating bloc

The states considered whereas following: $x_1$is the position in space of the quadcoper, $x_2$the angular velocities of the quadcopter, x_3 are the angles roll, pitch, yaw and x_4 is the angular velocities vector:

$$\dot{x}_1 = x_2; \quad \dot{x}_2 = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} R \cdot T_B + \frac{1}{m} \cdot F_D$$

$$\dot{x}_3 = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \cdot x_4 \tag{24}$$

$$\dot{x}_4 = I^{-1} \cdot \left[\tau - x_4 \times (I \cdot x_4) \right]$$

The Throttle2AV blocks compute a command between 0 and 100 and transform it into angular velocity for the quadcopter model. The correlation between PWM and thrust is obtained by polynomial approximation (in a least-squares sense) of points resulting from experimental measuring. The relation between thrust and squared angular velocity is depicted in equation (2).

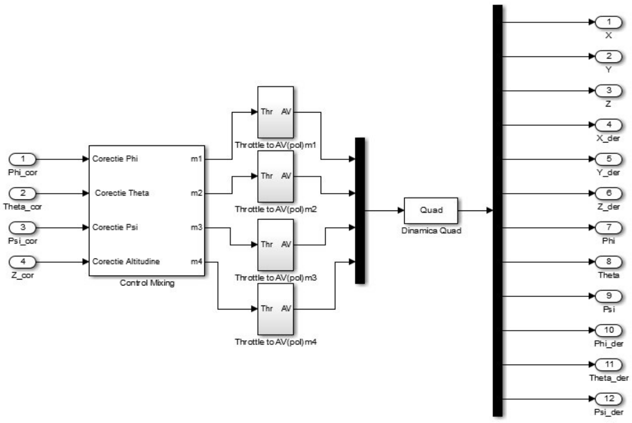

Fig. 7. Simulink Subsystem

The PID controllers ensure a certain value for the orientation and the altitude of the quadcopter. These generate corrections processed by the Quad block which in turn will induce the system's states.

The parameters of the PID controllers where generated by using the Tune function, imbedded in the Simulink PID Controller blocks. These values where afterwards used to simulate the performances of the quadcopter. The results are presented in Figure 8, for a reference of 10 degrees of the Roll angle.

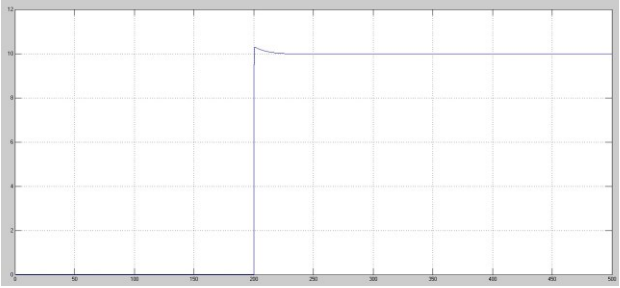

Fig. 8.System response for a 10 degrees reference for Roll angle (X axis)

To further test the performance of the model, the same reference was applied for a system with forces and torques perturbed. The result can be seen in Figure 9.

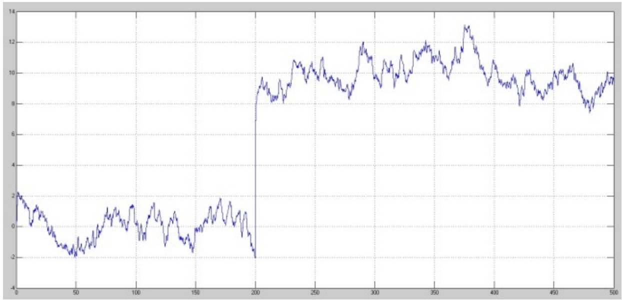

Fig. 9. System response for a 10 degrees reference for Roll angle (X axis) with perturbation

Considering the simulation performance achieved in close loop, control for a single axes was attempted by using the PID parameters obtained using the Tune function.

The results where satisfactory, but better result can be achieved by eliminating the slight oscillations and overshooting observed in the real system opposed to the model simulation.

V. CONCLUSION

In this paper the dynamic of a quadcopter UAV was discussed and two analytical models for the system were presented. The models were simulated in Mathworks Simulink environment. Those simulations were used to tune the parametersof a PID controller that would ensure a satisfactory response to step input reference. The control strategy for a single axis was tested in Simulink and then implemented on the system. The contribution of the authors consists of applying the dynamic equations from classical and Lagrangian mechanics to this particular type of UAV in order to obtain a physical model of the considered quadcopter. Also, the embedded software, communication data packaging and protocol, the MATLAB script and GUI were developed for this project using STM32F3Discovery firmware package and peripheral examples, Arduino libraries and MATLAB facilities for graphical interaction, data processing and methods for access to hardware devices.

The obtained results are satisfactory, but the dynamic performances can be improved. Future research will be focused ontotal control, on all three rotational axes of the quadcopter.

An important aspect of the development of the real system is the test area. The environment has to be symmetrical and in a closed space because the system is highly unstable and small perturbation can cause oscillations. The stability is clearly improved when the environment respects the mentioned criteria.

REFERENCES

[1] M. Rich, "Model development, system identification, and control of a quadrotor helicopter", Graduate Theses and Dissertations, Electrical and Computer Engineering, Paper 12770, Iowa State University, 2012

[2] J. Engel, J. Sturm, D. Cremers, "Camera-based navigation of a low-cost quadrocopter", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct. 2012, ISSN: 2153-0858, pp. 2815 – 2821.

[3] M. Hehn, R. D'Andrea, "A frequency domain iterative learning algorithm for high-performance, periodic quadrocopter maneuvers", Mechatronics, Vol. 24, Issue 8, Dec. 2014, ISSN 0957-4158, pp. 954 – 965.

[4] J. Das, W. Evans, M. Minnig, A. Bahr, G. Sukhatme, A. Martinoli, "Environmental Sensing Using Land-Based Spectrally-Selective Cameras and a Quadcopter", Experimental Robotics, The 13th Int. Symp. on Experim. Robotics, Vol. 88, Part IV, Springer International Publishing, DOI: 10.1007/978-3-319-00065-7_19, ISSN: 1610-7438, pp. 259 – 272.

[5] D. Hartman, K. Landis, M. Mehrer, S. Moreno, J. Kim, "Quadcopter Dynamic Modeling and Simulation", freeware project presented at"2014 MATLAB and Simulink Student Design Challenge"

[6] Virtual Robot Experimentation Platform USER MANUAL, Version 3.2.1, Coppelia Robotics, May 2015

[7] R.S. Hartenberg, J. Denavit, "Kinematic synthesis of linkages", McGraw-Hill series in mechanical engineering, New York: McGraw-Hill, 1965, ISBN-10: 0070269106, pp. 435.

[8] S.J.A. Malham, "An introduction to Lagrangian and Hamiltonian mechanics",Lecture notes, Maxwell Institute for Mathematical Sciences and School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, 2015

[9] J. Hanc, E.F. Taylor, "From conservation of energy to the principle of least action: A story line", American Journal of Physics, Vol. 72, No. 4, April 2004, DOI: 10.1119/1.1645282, pp. 514-521.

[10] A.J. Jerri,"The Shannon sampling theorem – Its various extenrions and applications: A tutorial review", Proceedings of the IEEE, Vol. 65, Issue 11, Nov. 1977, ISSN 0018-9219, pp. 1565 – 1596.

[11] J. Valvano, "Embedded Microcomputer Systems: Real Time Interfacing", Brooks-Cole, 2000, ISBN: 0534366422

[12] R. Mahony, T. Hamel, J.M. Pflimlin, "Complementary filter design on the special orthogonal group SO(3)", 44th IEEE Conference on Decision and Control, and the European Control Conference, December 12-15, 2005, DOI: 10.1109/CDC.2005.158236