Feedback Linearization approach for Standard and Fault Tolerant control: Application to a Quadrotor UAV Testbed

J. Ghandour, S. Aberkane, J.-C. Ponsart
University of Lorraine, CRAN, UMR 7039, Campus Sciences, B.P. 70239, 54506 Vandoeuvre-lès-Nancy Cedex, France
CNRS, CRAN, UMR 7039, France
Jean-Christophe.Ponsart@univ-lorraine.fr

Abstract. In this paper the control problem of a quadrotor vehicle experiencing a rotor failure is investigated. We develop a Feedback linearization approach to design a controller whose task is to make the vehicle performs trajectory following. Then we use the same approach to design a controller whose task is to make the vehicle enter a stable spin around its vertical axis, while retaining zero angular velocities around the other axis when a rotor failure is present. These conditions can be exploited to design a second control loop, which is used to perform trajectory following. The proposed double control loop architecture allows the vehicle to perform both trajectory and roll/pitch control. At last, to test the robustness of the feedback linearization technique, we applied wind to the quadrotor in mid flight.

1. Introduction
While the military and civil interest in Unmanned Arial Vehicles (UAVs) grows exponentially, the research centers race to publishing new and more efficient control techniques ([5], [16]). Among the various proposed control methods in the literature, one can cite for example LQ ([10]), H ([11], [12], [14]), backstepping ([7], [9], [15]) or sliding mode ([4], [13]) control techniques. In this paper, we propose to consider a nonlinear feedback linearization control ([2], [6]) in a fault tolerant control framework ([3]).

A UAV presents several basic advantages over a Manned Aerial Vehicle (MAV), including lower cost, prolonged continuous flight time, reduced radar signatures, smaller size, increased maneuverability, and the most important benefit is being less risky to crews.

Particular UAVs that have received much attention in the literature during the last ten years are quadrokers. This is mainly due to there suitability for a wide variety of UAV research applications and test-fly methods in control, fault diagnosis, fault tolerant control (FTC) as well as multi-agent based technologies in cooperative control, distributed control, formation flight, mobile wireless networks and communications.

During certain operations, a UAV cannot remains controlled from ground, for example in long continuous flights, in surveillance operations where multiple UAVs are in flight simultaneously, or behind the enemy line where signal transmission can put the stealth aspect of the UAV in jeopardy. This created the need for autonomous UAVs, which are programmed in a way to accomplish a given task without the need of human intervention at any point during the procedure. But with autonomous UAVs came another problem, these UAVs could perfectly handle themselves when fully operational, but a faulty sensor, or a damaged body, gets the system unstable, crashing the vehicle right away. Due
to the high cost of vehicles used by the military and civil organisms around the world, a FTC system was needed to be installed on these vehicles, assuring the damaged vehicle returns to its base without further damages.

This paper is divided into several sections. At first, the quadrotor description and modeling is presented. Then, the control law based on feedback linearization is detailed. The last section presents simulation results done on Matlab/Simulink®.

2. Quadrotor description

The quadrotor is a simple to manufacture aircraft with vertical landing/take-off capability, hovering ability, and is greatly maneuverable. The figure 1 shows the four actuators in a cross configuration, producing three rotations, Roll-Pitch-Yaw, and a vertical translation, Lift.

The first problem that faces us when creating a hovering machine is the problem of torque reaction based on Newton's third law of motion. This problem is solved in a quadrotor by turning two rotors clockwise, while turning the others counter-clockwise, thus canceling the global torque applied on the quadrotor.

A relation exists between the vector $U = [U_z, U_\phi, U_\theta, U_\psi]^T$ and $F = [F_1, F_2, F_3, F_4]^T$:

$$
\begin{bmatrix}
U_z \\
U_\phi \\
U_\theta \\
U_\psi
\end{bmatrix}
= 
\begin{bmatrix}
1 & 1 & 1 & 1 \\
0 & d & 0 & -d \\
-d & 0 & d & 0 \\
c & -c & c & -c
\end{bmatrix}
\begin{bmatrix}
F_1 \\
F_2 \\
F_3 \\
F_4
\end{bmatrix}
$$

(1)

Figure 1. The Quadrotor

where $d$ is the distance from the epicenter of the quadrotor to the actuator axes, $c > 0$ is the drag factor, $F_i$ is the lift generated by the $i^{th}$ actuator, $U_z$ is the total lift force generated by all four actuators and $U_\phi, U_\theta, U_\psi$ are the couples generated respectively in the Roll, Pitch and Yaw directions.

The mathematical model describing the position and attitude of a quadrotor is based on that of a rotating rigid body with six degrees of freedom. Close to hovering conditions ($\phi = 0$ and $\theta = 0$) one obtains ([14], [15]):

$$
\begin{align*}
    m \ddot{x} &= \left(C_y S_\psi C_\phi + S_\phi S_y\right) U_z - K_x \dot{x} \\
    m \ddot{y} &= \left(C_y S_\psi S_\phi - S_y C_\phi\right) U_z - K_y \dot{y} \\
    m \ddot{z} &= \left(C_y C_\phi\right) U_z - m g - K_z \dot{z} \\
    J_\rho \ddot{\psi} &= U_\psi - \dot{\psi} \left(J_\rho - J_\phi\right) - \dot{\theta} J_\tau \Omega_\tau - K_\rho \dot{\psi} \\
    J_\theta \ddot{\phi} &= U_\phi - \dot{\phi} \left(J_\phi - J_\theta\right) - \dot{\psi} J_\rho \Omega_\rho - K_\phi \dot{\phi} \\
    J_\phi \ddot{\psi} &= U_\psi - \dot{\psi} J_\rho \Omega_\rho - K_\phi \dot{\phi}
\end{align*}
$$

(2)
where \( m \) is the mass of the quadrotor, \( J_p, J_q, J_r \) are the moments of inertia of the quadrotor, \( J_t \) is the moment of inertia for the engines and \( \Omega_t = \Omega_1 + \Omega_2 + \Omega_3+ \Omega_4 \) is the overall speed of the propellers wherein \( \Omega_1 \) is the speed of the \( i^{th} \) propeller. \( S_* \) and \( C_* \) are the respective abbreviations of \( \sin(\cdot) \) and \( \cos(\cdot) \).

3. Controlling the vehicle
With 6 degrees of freedom, and 4 actuators, the quadrotor is an under-actuated system, making it impossible to control all of the states simultaneously. The controlled outputs are generally \( \{x, y, z, \psi, \theta, \phi\} \), thus with stable pitch and roll angles, a quadrotor is able to reach a desired position in space and a desired yaw angle. As stated in [1] several UAV flight control algorithms already exist, including the Gain—Scheduled PID, adaptive control, Model Reference Adaptive Control (MRAC), Sliding Mode Control, Backstepping Control (BSC), Control allocation, Model Predictive Control (MPC), and Feedback Linearization (FL).

3.1. Feedback Linearization

3.1.1. Procedure’s description
Feedback linearization is a common approach used in controlling nonlinear systems ([2], [6]). The approach involves coming up with a transformation of the nonlinear system into an equivalent linear system through a change of variables and a suitable control input. Feedback linearization may be applied to nonlinear systems as:

\[
\begin{align*}
\dot{x}(t) &= f(x) + g(x) u(t) \\
y(t) &= h(x)
\end{align*}
\] (3)

where \( x \in \mathbb{R}^n \) is the state vector, \( u \in \mathbb{R}^n \) is the vector of inputs, and \( y \in \mathbb{R}^n \) is the vector of outputs. The goal is to develop a control input \( u(t) = a(x) + b(x) v(t) \) that induces a linear input-output map between the new input \( v(t) \) and the output \( y(t) \). An outer-loop control strategy for the resulting linear system can then be applied.

3.1.2. Procedure’s application
In this section, we use the model (2) after neglecting the gyroscopic effects and the drag forces, while using the steps of the work in [2], to develop a feedback Linearization controller of a quadrotor ensuring that the position \( \{x(t), y(t), z(t), \psi(t)\} \) tracks the desired trajectory \( \{x_d(t), y_d(t), z_d(t), \psi_d(t)\} \) asymptotically.

Let \( \xi = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \dot{\psi}, \dot{\theta}, \dot{\phi}, \dot{\psi}]^T \) be the state vector, \( U = [T_x, T_y, T_z]^T \) the control input, and \( \eta(\xi) = [x, y, z, \psi]^T \) the output vector. Being under-actuated, we cannot use a static but a dynamic output feedback control. By successive differentiation, a nonsingular input appears when we reach the forth derivative of \( x, y, \) and \( z \), and the second derivative of \( \psi \):

\[
\begin{align*}
\dot{x}(t) &= \frac{U}{m}[\dot{\psi} \dot{S}_{p} \dot{C}_{p} - \dot{\psi} \dot{S}_{q} \dot{C}_{q} - \dot{\psi} \dot{S}_{r} \dot{C}_{r} - \dot{\psi} \dot{S}_{t} \dot{C}_{t} + \dot{\psi} \dot{S}_{p} \dot{C}_{p} + \dot{\psi} \dot{S}_{q} \dot{C}_{q} + \dot{\psi} \dot{S}_{r} \dot{C}_{r} + \dot{\psi} \dot{S}_{t} \dot{C}_{t} \\
&\quad+ \dot{\psi} \dot{S}_{p} \dot{S}_{q} + \dot{\psi} \dot{S}_{q} \dot{S}_{p} - \dot{\psi} \dot{S}_{p} \dot{S}_{p} + \dot{\psi} \dot{S}_{q} \dot{S}_{q} - \dot{\psi} \dot{S}_{r} \dot{S}_{r} + \dot{\psi} \dot{S}_{t} \dot{S}_{t} + \dot{\psi} \dot{S}_{p} \dot{C}_{p} + \dot{\psi} \dot{S}_{q} \dot{C}_{q} + \dot{\psi} \dot{S}_{r} \dot{C}_{r} + \dot{\psi} \dot{S}_{t} \dot{C}_{t}]
\end{align*}
\] (4)

\[
\begin{align*}
\dot{y}(t) &= \frac{U}{m}[\dot{\psi} \dot{S}_{p} \dot{C}_{p} + \dot{\psi} \dot{S}_{q} \dot{C}_{q} + \dot{\psi} \dot{S}_{r} \dot{C}_{r} - \dot{\psi} \dot{S}_{t} \dot{C}_{t} + \dot{\psi} \dot{S}_{p} \dot{C}_{p} + \dot{\psi} \dot{S}_{q} \dot{C}_{q} + \dot{\psi} \dot{S}_{r} \dot{C}_{r} + \dot{\psi} \dot{S}_{t} \dot{C}_{t} \\
&\quad- \dot{\psi} \dot{S}_{p} \dot{S}_{q} + \dot{\psi} \dot{S}_{q} \dot{S}_{p} + \dot{\psi} \dot{S}_{p} \dot{S}_{p} + \dot{\psi} \dot{S}_{q} \dot{S}_{q} + \dot{\psi} \dot{S}_{r} \dot{S}_{r} + \dot{\psi} \dot{S}_{t} \dot{S}_{t} + \dot{\psi} \dot{S}_{p} \dot{C}_{p} + \dot{\psi} \dot{S}_{q} \dot{C}_{q} + \dot{\psi} \dot{S}_{r} \dot{C}_{r} + \dot{\psi} \dot{S}_{t} \dot{C}_{t}]
\end{align*}
\] (5)
An effective fault recovery is achieved after two major successive procedures. A Fault Detection and Isolation (FDI), is a fast algorithm that indicates to the system the sensor, actuator, or body part that has been damaged. Following the FDI decision, the system switches to the adequate FTC algorithm.
The fault tolerant controller is developed following a double control loop architecture in which an inner and faster controller has the task to regulate the attitude angles and the altitude of the vehicle, while an outer and slower controller has the aim of modifying the desired values of the attitude angles in order to perform trajectory tracking. The control law of the whole system is computed by using the same steps of the work in [3].

Let $\chi = [\psi, \theta, \dot{\psi}, \dot{\theta}, x, y, z]^T$ be the state vector. The dynamic equations (2), after neglecting the gyroscopic effects due to the propeller's rotation, can be rewritten in a state-space form:

\[
\begin{align*}
\dot{x}_1 &= x_1 + x_3 S_c \frac{S_m}{C_m} + x_4 C_c \frac{S_m}{C_m}, \\
\dot{x}_2 &= x_3 C_c - x_4 S_c, \\
\dot{x}_3 &= \frac{1}{I_c}[x_3 S_c + x_4 C_c], \\
\dot{x}_4 &= \frac{1}{I_c}[K_1 x_3 - x_6 x_3 (J_1 - J_2)], \\
\dot{x}_5 &= \frac{1}{I_c}[K_2 x_3 - x_6 x_3 (J_1 - J_2) + U_e], \\
\dot{x}_6 &= \frac{1}{I_c}[K_3 x_3 - x_6 x_3 (J_1 - J_2) + U_e].
\end{align*}
\]

\[\chi = \frac{1}{m} [(C_c S_c + S_c S_5) U_1 - K_1 x_3]
\]

\[\chi = \frac{1}{m} [(C_c S_c + S_c S_5) U_1 - K_2 x_3]
\]

\[\chi = \frac{1}{m} [(C_c S_c + S_c S_5) U_1 - K_3 x_3 - m \ddot{x}]
\]

**In case of failure on M1:** the control input vector is chosen as $u = [U_1, U_2, U_3]^T$, where $U_e = \frac{d}{2} \left( U_1 + \frac{U_2}{c} \right)$.

**In case of failure on M2:** the control input vector is chosen as $u = [U_1, U_2, U_3]^T$, where $U_e = \frac{d}{2} \left( U_1 + \frac{U_2}{c} \right)$.

**In case of failure on M3:** the control input vector is chosen as $u = [U_1, U_2, U_3]^T$, where $U_e = \frac{d}{2} \left( U_1 + \frac{U_2}{c} \right)$.

**In case of failure on M4:** the control input vector is chosen as $u = [U_1, U_2, U_3]^T$, where $U_e = \frac{d}{2} \left( U_1 + \frac{U_2}{c} \right)$.

### 3.2.1. Inner Control Loop

From the state-space form it can be seen that the dynamics of the state variables $x_1, x_2, x_3, x_5, x_6, x_0, x_{12}$, which we will call $\chi$ for notation simplicity, can be written as $\dot{\chi} = f(\chi) + g(\chi) u$.

The dynamics of the states $x_1, x_2$ and $x_0$ can be similarly written as:

![Figure 2. The double control loop architecture](image-url)
\[
\begin{bmatrix}
\dot{X}_1 \\
\dot{X}_2 \\
\dot{X}_3
\end{bmatrix} = \begin{bmatrix}
X_4 + X_5 S_{\beta_i} \frac{S_{\alpha_i}}{C_{\alpha_i}} + X_6 C_{\beta_i} \frac{S_{\alpha_i}}{C_{\alpha_i}} \\
X_5 C_{\beta_i} - X_6 S_{\beta_i} \\
X_{12}
\end{bmatrix} = \hat{f}(\mathcal{Z}) 
\]

which, in this case, is independent of the input vector \( u \). This property becomes useful when calculating the second derivative of \( [X_1, X_2, X_3]^T \):

\[
\begin{bmatrix}
\dot{X}_1 \\
\dot{X}_2 \\
\dot{X}_3
\end{bmatrix} = \frac{d\hat{f}(\mathcal{Z})}{dt} = \frac{\partial \hat{f}(\mathcal{Z})}{\partial \mathcal{Z}} \hat{\mathcal{Z}} = \frac{\partial \hat{f}(\mathcal{Z})}{\partial \mathcal{Z}} f(\mathcal{Z}) + \frac{\partial \hat{f}(\mathcal{Z})}{\partial \mathcal{Z}} g(\mathcal{Z}) u 
\]

Denoting the Jacobian matrix with \( J(\mathcal{Z}) = \frac{\partial \hat{f}(\mathcal{Z})}{\partial \mathcal{Z}} \), the previous dynamics can be written as:

\[
\begin{bmatrix}
\dot{X}_1 \\
\dot{X}_2 \\
\dot{X}_3
\end{bmatrix} = J(\mathcal{Z}) f(\mathcal{Z}) + J(\mathcal{Z}) g(\mathcal{Z}) u 
\]

It can be proved that matrix \( J(\mathcal{Z}) g(\mathcal{Z}) \) is invertible if and only if \( X_i \neq \arctan \left( \frac{\pi J_i}{2 J_d} \right) \) and \( X_i \neq \arctan \left( \frac{\pi J_i}{2 J_d} \right) \). In most practical scenarios this condition is satisfied. In our case, with the parameters adopted for the simulation, the invertibility condition is guaranteed as long as the pitch and roll angles are limited to 80° which is a value never reached by the quadrotor during non-aerobatic flight. Let \( \chi_{id}, \chi_{id}, \chi_{id} \) be the desired values for \( \chi_1, \chi_2, \chi_3 \), and define the \( i \)-th error as \( e_i = \chi_i - \chi_{id} \). If the control inputs are chosen as:

\[
u^* = -(J(\mathcal{Z}) g(\mathcal{Z}))^{-1} J(\mathcal{Z}) f(\mathcal{Z}) + (J(\mathcal{Z}) g(\mathcal{Z}))^{-1} \begin{bmatrix}
\chi_{id} \\
\chi_{id} \\
\chi_{id}
\end{bmatrix} = \begin{bmatrix}
C_{\alpha_i} e_1 \\
C_{\alpha_i} e_2 \\
C_{\alpha_i} e_3
\end{bmatrix}
\]

where \( C_i \) are positive constants, then the error dynamics can be written as

\[
\begin{bmatrix}
\dot{e}_1 \\
\dot{e}_2 \\
\dot{e}_3
\end{bmatrix} + \begin{bmatrix}
C_{\alpha_i} e_1 \\
C_{\alpha_i} e_2 \\
C_{\alpha_i} e_3
\end{bmatrix} + \begin{bmatrix}
C_{\alpha_i}^2 \dot{e}_1 \\
C_{\alpha_i}^2 \dot{e}_2 \\
C_{\alpha_i}^2 \dot{e}_3
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\]

which yields stable second-order dynamics providing exponential decay of \( e_i \) for \( i = 1, 2, 9 \) (respectively the roll, pitch and altitude errors).

### 3.2.2. Outer Control Loop

An outer control loop’s task is to generate the desired values for the roll and pitch angles, in a way to reach a desired position in the horizontal plane. The constants \( C_i \) and \( C_0 \) represent the natural frequency of the subsystem’s dynamics, subsystems described by the state variables \( \chi_1, \chi_2, \chi_3 \) and \( \chi_1, \chi_2, \chi_3, \chi_4 \). To avoid compromising the stability of the system, we choose \( C_0 < C_i \) which makes the outer control loop act much slower than the inner control loop. Under the presumption that \( \chi_1 \) and \( \chi_2 \) are small angles, \( \chi_{id} \) and \( \chi_{id} \) are chosen close to zero as:
\[
\begin{bmatrix}
\dot{\chi}_{1d} \\
\dot{\chi}_{2d}
\end{bmatrix} = \frac{m}{U_1} \begin{bmatrix}
S_{\chi_1} & -C_{\chi_1} \\
C_{\chi_1} & S_{\chi_1}
\end{bmatrix} \begin{bmatrix}
\dot{\chi}_{10d} + K_1 \dot{\chi}_{10} - C_1 \dot{\chi}_{20} - C_3 \dot{\chi}_{21} \\
\dot{\chi}_{20d} + \frac{K_2}{m} \chi_{20} - C_1 \chi_{20} - C_3 \chi_{21}
\end{bmatrix}
\]

where \( C_i \) are positive constants, then error dynamics for the horizontal displacements in closed loop is:

\[
\begin{bmatrix}
\dot{e}_z + C_1 e_z + C_3 \dot{e}_z \\
\dot{e}_y + C_1 e_y + C_3 \dot{e}_y
\end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

which is asymptotically stable.

From a physical point of view, it means that when one of the rotor fails, the inner control law stabilizes roll, pitch and altitude, while the outer control law exploits the near hover condition to slowly change the pitch and roll angles in order to move the vehicle to a desired position in space.

### 4. Results and Discussions

Several simulations have been run using the Matlab and Simulink® softwares in order to validate the theoretical results. The model used to represent the quadrotor is preferred to be as realistic as possible, since eventually this model will be replaced by the quadrotor itself. On the other hand, the model used to calculate the control techniques tends to be as simplified as possible, with an acceptable error, in order to generate a simple and fast control algorithm.

In order to test the efficiency of the Feedback Linearization Control technique at handling and maneuvering the quadrotor, we needed a smooth trajectory with no sharp turns, and minimal acceleration value. A rectifiable polynomial trajectory that responds to the requirements was developed:

![Figure 3. Polynomial equation](image)

The polynomial equation shown in figure 3 starts ascending at \( t_0 \) with a speed and acceleration equal to 0, and after a \( T \) period of time, it gets to the desired \( F_{\text{max}} \) height with a speed and acceleration equal to 0 again:

\[
f(t) = \frac{6F_{\text{max}}}{T^5} (t-t_0)^5 - \frac{15F_{\text{max}}}{T^4} (t-t_0)^4 + \frac{10F_{\text{max}}}{T^3} (t-t_0)^3
\]

with \( f(t_0) = \dot{f}(t_0) = \ddot{f}(t_0) = \dddot{f}(t_0 + T) = \dddot{f}(t_0 + T) = 0 \) and \( f(T) = F_{\text{max}} \).

#### 4.1. Feedback Linearization Control

In order to verify the effectiveness and the efficiency of the proposed Feedback Linearization control technique, an application to quadrotor helicopter is conducted by simulation. The simulations were done with a Fixed-step size equal to 0.001s, and with the "ode5 (Dormand-Prince)" solver. The nominal parameters for a quadrotor helicopter are: \( m = 2 \text{ kg}, \ J_z = J_y = J/2 = 1.2416 \text{ Nm.s}^2/\text{rad}, \ d = 0.2 \text{ m}, \ c = 0.01 \text{ m}, \ J_r = 2.8e^3 \text{ Nm.s}^2/\text{rad}, \ K = K_y = K = 10^{-2} \text{ N.s/m} \) and \( K_x = K_{\phi} = K_{\psi} = 10^{-3} \text{ Nm.s/rad} \).

The proposed Feedback Linearization control law requires the knowledge of certain derivatives. In
order to avoid analytical derivation difficulties, we estimate them by using the \( \dot{v} = \frac{\Delta v}{\Delta t} \) differentiation approximation. Where \( \Delta v \) is the change in the value to be derived and \( \Delta t \) is the change in time since the previous simulation time step. The quadrotor is initially in hover flight and the initial conditions are \( U = [T, T_v, T_h, T_w] = [m g, 0, 0, 0] \).

Three modules consist the simulation system. The first being a nonlinear system module where we can find all the differential equations (2). The second is the controller module that includes the algorithm based on the control technique being tested. The third is an initialization module that sets the essential parameters to run the simulation.

By looking into the equations (14), we deduce that we have 3 fourth order integrators, and one second order integrator. In order to stabilize the outputs, a pole placement method will be employed. For example, the first equation of (14) where \( v_i \) is the input and \( O_i \) is the output, can be represented as:

\[
\begin{bmatrix}
\dot{x} \\
\dot{x}^{(2)} \\
\dot{x}^{(3)} \\
\dot{x}^{(4)}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
\dot{x} \\
\dot{x}^{(2)} \\
\dot{x}^{(3)}
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
0 \\
1
\end{bmatrix} v_i = A_i \begin{bmatrix}
\dot{x} \\
\dot{x}^{(2)} \\
\dot{x}^{(3)} \\
\dot{x}^{(4)}
\end{bmatrix} + B_i v_i
\]

(24)

Figure 4. Positions

Figure 5. Errors

Figure 6. Forces
4.2. Fault Tolerant Feedback Linearization Control results

To simulate a free fall recovery after few milliseconds of the $M_2$ failure, with the same nominal parameters for a quadrotor, the same Fixed-step size, and the same solver as in section 4.1, the initial conditions are set in a way that $x = 2m$, $y = 5m$, $z = 7m$, $\psi = 0rad$, $\dot{x} = -0.1m.s^{-1}$, $\dot{y} = 0.1m.s^{-1}$, $\dot{z} = 2m.s^{-1}$, $\dot{\psi} = \dot{\theta} = 0rad.s^{-1}$. At first, the desired values for the state variables are set to $x_d = 3m$, $y_d = 4m$, $z_d = 10m$, once the quadrotor has reached the desired lateral and longitudinal position, the altitude desired is then set to 10 (1-LP) where LP (landing procedure) is the step response of the following transfer function:

$$I(s) = \frac{1}{(s+1)^6}$$  (25)

where $s$ is the Laplace variable.

The time needed to reach the desired set points is caused by the fact that the outer controller tends to force the attitude angles to differ from zero as long as the desired position is not reached, while the inner control tends to force those angles to zero. The simulation has been therefore run for 200 s, which is a sufficient time to reach the landing point with stable Pitch and Roll angles.

As it can be seen in figure 7 the attitude angles $\psi$ and $\theta$ are stabilized, the oscillations are due to the presence of the outer controller. The $\psi$ angle does not reach a steady state value, but increases according to an approximate linear law, this behavior is actually not a surprise since the controllability of the $\psi$ variable has been sacrificed in the design phase of the fault tolerant controller. The linear

![Figure 7. Positions](image1.png)

![Figure 8. The Quadrotor's speed](image2.png)
positions reach their desired values, even if the regulation of \( z \) is much faster than that of \( x \) and \( y \) due to the fact that altitude regulation is operated by the inner and faster controller. The angular and linear velocities are all stabilized as it can be seen in figure 8, with the exception of \( \psi \), for the reasons already stated, and \( \dot{z} \) which becomes negative during the landing procedure.

\[ P = \begin{cases} 1 & \text{for } 5 < t < 25 \\ 0 & \text{everywhere else} \end{cases} \]

(26)

We can see that during the 20 to 25 seconds time interval, while wind is still blowing through the quadrotor (figure 10), \( x, y, z, \phi, \theta, \psi \) are stabilized (figure 11), the errors are set to zero (figure 12) and the forces generated by the actuators are acceptable (figure 13). When the first 40 seconds of the simulation pass, the wind forces are totally eliminated, and we can see that all errors are set to zero again, \( x, y, z \) and \( \psi \) are stabilized, while \( \phi \) and \( \theta \) tend towards zero. This indicates a good robustness of the FL.

4.3. Wind disturbance

To test the robustness of the Feedback Linearization technique, the same simulation in section 4.1 was repeated but this time a wind force was introduced in the form of a transfer function's response (25) to the following square pulse:

\[ P = \begin{cases} 1 & \text{for } 5 < t < 25 \\ 0 & \text{everywhere else} \end{cases} \]

Figure 9. Forces

Figure 10. Wind Forces

Figure 11. Positions
5. Conclusion and Perspective
It has been proven in this paper that a UAV can be automated using a Feedback Linearization approach, in a way that, without human intervention, it can reach any desired point in space, even in a faulty environment. The presence of tolerable errors in figure 5 could be explained by the differentiation approximation used to avoid analytical derivation difficulties, and by the fact that we used a simplified model to build the control technique. Last but not least, the Feedback Linearization technique proved to have excellent robustness in section 4.3.

References
[1] J. D’Azzo and C.H. Houpis, “Linear control system analysis and design, conventional and modern”, McGraw-Hill Series in Electrical and Computer Engineering, 1995.
[2] A.J. Fossard and D. Normand-Cyrot (Eds.), Nonlinear Systems, Vol. 3: Control, Springer, 1996.
[3] A. Freddi, A. Lanzon, S. Longhi, “A feedback linearization approach to fault tolerance in quadrotor vehicles”, in: Preprints of the 18th IFAC World Congress, Milano, Italy, 28 August–2 September 2011.
[4] L. Fridman A. Benallegue, A. Mokhtari. High-order sliding-mode observer for a quadrotor uav. International journal of robust and nonlinear control, 18 :427–440, 2007.
[5] G. M. Hoffmann, H. Huang, S. L. Wasi, and C. J. Tomlin. “Quadrotor helicopter flight dynamics and control: Theory and experiment”, in Proc. of the AIAA Navigation, and Control Conference, 2007.
[6] A. Isidori, “Nonlinear Control Systems”, 3rd Edition, Springer, 1995.
[7] H. Khebbache. Tolérance aux défauts via la méthode backstepping des systèmes non linéaires. Master’s thesis, Université Ferhat abbas de setif, Algérie, 2012
[8] Quanser official site : Control Challenges - Unmanned Systems - Qball-X4 http://www.quanser.com/Products/qball
[9] H. Ramirez Rodriguez et al., “Integral Sliding Mode Backstepping Control of Quadrotors for Robust Position Tracking”, ICUAS’13 International Conference on Unmanned Aircraft Systems, Atlanta, USA, May 28-31 2013.
[10] F.R. Rubio G.V. Raffo, M.G. Ortega. An integral predictive/nonlinear hinf control structure for a quadrotor helicopter. Automatica, 46 :29–39, 2009.
[11] I. Sadeghzadeh et al., “Linear Parameter Varying Control Synthesis: State Feedback versus H∞ Technique with Application to Quadrotor UAV”, ICUAS’14 International Conference on Unmanned Aircraft Systems, Orlando, USA, May 27-30 2014.
[12] R. Siegwart S. Bouabdallah, A. Noth. PID vs LQ control techniques applied to an indoor micro quadrotor. IEEE Intelligent robotics system, 2004.
[13] R. Siegwart S. Bouabdallah. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In International conference on robotics and automation, 2005.
[14] J.F. Whidborne A.K. Cooke I.D. Cowling, O.A. Yakimenko. A prototype of an autonomous controller for a quadrotor uav. European Journal of Control, 2007.
[15] X.B. Zhang, Y.M. Zhang, C.-Y. Su, Y. Feng, “Fault-tolerant control for quadrotor UAV via backstepping approach”, in: The 48th AIAA Aerospace Sciences Meeting, Orlando, Florida, USA, 47 January 2010
[16] Y.M. Zhang et al., “Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed”, Journal of the Franklin Institute. (2013), http://dx.doi.org/10.1016/j.jfranklin.2013.01.009