Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor

Samir Bouabdallah and Roland Siegwart

Autonomous Systems Lab
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
{*samir.bouabdallah, roland.siegwart*}@epfl.ch

Abstract— **The latest technological progress in sensors, actuators and energy storage devices enables the developments of miniature VTOL[1] systems. In this paper we present the results of two nonlinear control techniques applied to an autonomous micro helicopter called Quadrotor. A backstepping and a sliding-mode techniques. We performed various simulations in open and closed loop and implemented several experiments on the test-bench to validate the control laws. Finally, we discuss the results of each approach. These developments are part of the $OS4$[2] project in our lab.**

Index Terms— **Quadrotor Architecture, VTOL Control, Nonlinear control, Backstepping, Sliding-mode.**

Fig. 1. Quadrotor concept motion description, the arrow width is proportional to propeller rotational speed.

I. INTRODUCTION

Flying objects have always exerted a great fascination on man encouraging all kinds of research and development. The important recent technological progress in sensors, actuators, processors and power storage devices represents a real jump ahead. Enabling by the way the emergence of new applications like the indoor micro aerial robots. Compared with the other flying principles, VTOL systems have specific characteristics which allow the execution of applications that would be difficult or impossible otherwise, such as building surveillance and intervention in hostile environments. The $OS4$ project, initiated at the Autonomous Systems Laboratory (EPFL), focuses on micro VTOL vehicles evolving towards a full autonomy in indoor environments. The approach advocated for this project is to simultaneously work on design and control. The first step in this project after the prototype building is to develop a reliable control system for configuration stabilization and trajectory tracking. In this paper we present two nonlinear control techniques simulated and tested for $OS4$ configuration stabilization.

A. Quadrotor Configuration

The Quadrotor concept has been around for a long time. The Breguet-Richet Quadrotor helicopter *Gyroplane No.1* built in 1907 is reported to have lifted into flight. One can describe the vehicle as having four propellers in cross configuration. The two pairs of propellers (1,3) and (2,4) turn in opposite directions. By varying the rotor speed,
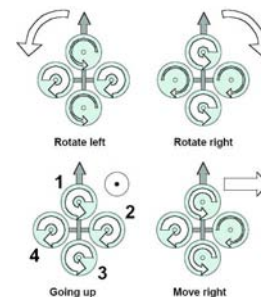
one can change the lift force and create motion. Thus, increasing or decreasing the four propeller's speeds together generates vertical motion. Changing the 2 and 4 propeller's speed conversely produces roll rotation coupled with lateral motion. Pitch rotation and the corresponding lateral motion result from 1 and 3 propeller's speed conversely modified as described in Fig. 1. Yaw rotation is more subtle, as it results from the difference in the counter-torque between each pair of propellers. In spite of the four actuators, the Quadrotor is still an underactuated and dynamically unstable system.

II. QUADROTOR DYNAMIC MODELLING

Micro VTOL are highly dynamic systems thus, an appropriate model ideally includes the gyroscopic effects resulting from both the rigid body rotation in space and the four propulsion groups rotation [1]. The latter effect includes the propellers, the gearbox and the motor rotation in case of a relatively high motor inertia like for the outer-rotor BLDC[3]. Let us consider earth fixed frame E and body fixed frame B, as seen in Fig. 2. Using Euler angles parametrization, the airframe orientation in space is given by a rotation R from B to E, where $R \in SO3$ is the rotation matrix. The dynamic model is derived using Newton-Euler formalism as shown in [2]. The dynamics of a rigid body under external forces applied to the center of mass and expressed in the body fixed frame are formulated in (1) as shown in [3].

---

[1]Vertical Take-Off and Landing
[2]Omnidirectional Stationary Flying Outstretched Robot
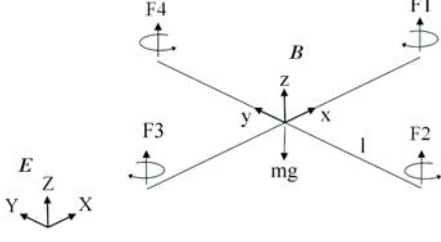
[3]BrushLess Direct Current

Fig. 2. Quadrotor configuration frame system with a body fixed frame B and the inertial frame E. The circular arrows indicate the direction of rotation of each propeller.

$$\begin{bmatrix} mI_{3x3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \quad (1)$$

Where $I \in \Re^{(3x3)}$ the inertia matrix, V the body linear speed vector and ω the body angular speed. In the frame system Fig. 2, the equations of motion for the helicopter can be written as in [4]:

$$\begin{cases} \dot{\zeta} = \nu \\ m\dot{\nu} = RF_b \\ \dot{R} = R\hat{\omega} \\ J\dot{\omega} = -\omega \times J\omega + \tau_a \end{cases} \quad (2)$$

The first-level approximate model (3) of the Quadrotor can be rewritten as:

$$\begin{cases} \dot{\zeta} = \nu \\ \dot{\nu} = -ge_3 + R_{e3}(\frac{b}{m}\sum \Omega_i^2) \\ \dot{R} = R\hat{\omega} \\ I\dot{\omega} = -\omega \times I\omega - \sum J_r(\omega \times e_3)\Omega_i + \tau_a \end{cases} \quad (3)$$

where :

Symbol	definition
ζ	position vector
R	rotation matrix
$\hat{\omega}$	skew symmetric matrix
ϕ	roll angle
θ	pitch angle
ψ	yaw angle
Ω	rotor speed
$I_{x,y,z}$	body inertia
J_r	rotor inertia
J_m	motor inertia
J_p	propeller inertia
τ_a	torque on airframe body
b	thrust factor
d	drag factor
l	lever

The torque applied on the vehicle's body along an axis is the difference between the torque generated by each propeller on the other axis.

$$\tau_a = \begin{pmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{pmatrix} \quad (4)$$

If we consider the motor inertia and a reversing gearbox with a negligible inertia, the rotor (propulsion group) inertia is then:

$$J_r = J_p - J_m r \quad (5)$$

The full Quadrotor dynamic model with the x,y,z motions as a consequence of a pitch or roll rotation is:

$$\begin{cases} \ddot{\phi} = \dot{\theta}\dot{\psi}(\frac{I_y - I_z}{I_x}) - \frac{J_r}{I_x}\dot{\theta}\Omega + \frac{l}{I_x}U_2 \\ \ddot{\theta} = \dot{\phi}\dot{\psi}(\frac{I_z - I_x}{I_y}) + \frac{J_r}{I_y}\dot{\phi}\Omega + \frac{l}{I_y}U_3 \\ \ddot{\psi} = \dot{\phi}\dot{\theta}(\frac{I_x - I_y}{I_z}) + \frac{1}{I_z}U_4 \\ \ddot{z} = -g + (\cos\phi\cos\theta)\frac{1}{m}U_1 \\ \ddot{x} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{1}{m}U_1 \\ \ddot{y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\frac{1}{m}U_1 \end{cases} \quad (6)$$

The first term in the orientation subsystem (ϕ, θ, ψ) is the gyroscopic effect resulting from the rigid body rotation in space and the second one is due to the propulsion group rotation. The system's inputs are posed U_1, U_2, U_3, U_4 and Ω a disturbance, obtaining:

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = b(\Omega_4^2 - \Omega_2^2) \\ U_3 = b(\Omega_3^2 - \Omega_1^2) \\ U_4 = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\ \Omega = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3 \end{cases} \quad (7)$$

A. Rotor Dynamics

The rotors are driven by DC-motors with the well known equations [5]:

$$\begin{cases} L\frac{di}{dt} = u - Ri - k_e\omega_m \\ J\frac{d\omega_m}{dt} = \tau_m - \tau_d \end{cases} \quad (8)$$

As we use a small motor with a very low inductance, the second order DC-motor dynamics may be approximated:

$$J\frac{d\omega_m}{dt} = -\frac{k_m^2}{R}\omega_m - \tau_d + \frac{k_m}{R}u \quad (9)$$

By introducing the propeller and the gearbox models, the equation (9) may be rewritten:

$$\begin{cases} \dot{\omega}_m = -\frac{1}{\tau}\omega_m - \frac{d}{\eta r^3 J_t}\omega_m^2 + \frac{1}{k_m\tau}u \\ with : \\ \frac{1}{\tau} = \frac{k_m^2}{RJ_t} \end{cases} \quad (10)$$

The equation (10) can be linearized (see [6]) around an operation point $\dot{w}_0$ to the form $\dot{w}_m = -Aw_m + Bu + C$ with:

$$A = \left(\frac{1}{\tau} + \frac{2dw_0}{\eta r^3 J_t}\right), \quad B = \left(\frac{1}{k_m\tau}\right), \quad C = \left(\frac{d\omega_0^2}{\eta r^3 J_t}\right) \quad (11)$$

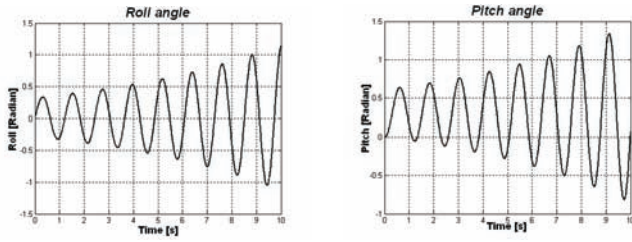| Symbol	Definition
u	motor input
τ	motor time-constant
R	motor internal resistance
k_e	back EMF constant
k_m	torque constant
ω_m	motor angular speed
τ_m	motor torque
τ_d	motor load
r	gear box reduction ratio
η	gear box efficiency
J_t	total inertia seen by the motor

Fig. 3. Simulation: Natural response of the roll and pitch angles to an initial angular speed. The oscillations are strongly and rapidly amplified.
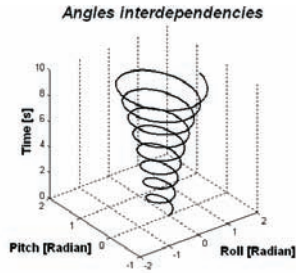

Fig. 4. Simulation: Roll and pitch angles interdependencies in natural response. This behavior was also observed on the real system.

III. OPEN LOOP BEHAVIOR

It is well known that the VTOL systems are dynamically unstable and thus very hard to control. In order to enhance our understanding of the system, we performed several simulations in open-loop mode. These simulations were particularly useful for the recognition of the contribution of each modelled effect to the dynamics of the system. Also, knowing the natural behavior of the system could be useful for establishing adapted control laws. The Fig. 3 shows the system natural response to an initial roll (or pitch) angular speed excitation. The system gains mechanical energy, starts to oscillate and tends to amplify rapidly these oscillations. As a perfect cross configuration was assumed for our Quadrotor, the roll and pitch angles behaves similarly, the phase shift being the only main difference. This is highlighted in Fig. 4 were the roll and pitch interdependencies are plotted through time. This simulated behavior was also observed on the real system. On the other hand, we used the same simulations to perform a mechanical system optimization in order to reduce the natural instability of the system. But this is out of the scope of this paper.

IV. OS4 TEST-BENCH

The design of a control system for a flying robot requires the usage of an adequate test-bench at least for the preliminary experiments. For our control experiments, we developed the test-bench in Fig. 5. From a PC and through a standard RS232 port, one can send orders to the test-bench (see Fig. 6). The RS232 to I2C module translates the serial signals to the I2C bus motor modules. These modules integer a PID regulator on a PIC16F876 microcontroller.
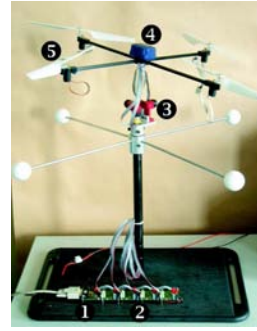

Fig. 5. $OS4$ test-bench for stabilization strategies testing, 3DOF are locked, the cross is made with carbon rods and the flying system weight is about 240g. 1)RS232 to I2C translator, 2)Motor modules, 3)3D captured universal joint, 4)Micro IMU, 5)Propulsion group.
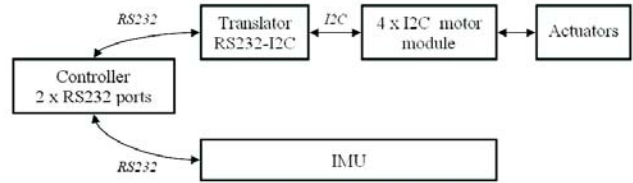

Fig. 6. $OS4$ test-bench block-diagram

The MT9-B IMU[4] estimates with a kalman filter the 3D orientation angles which reduces the drift. It gives also the calibrated data of acceleration and angular velocity. It weights about 33g and communicates at 115kbps. The $OS4$ test-bench has 4 propulsion groups, each one is composed of a 25g motor, a 6g gear box and a 6g propeller. To design the propulsion group, a test, evaluation and comparison method was developed.

V. BACKSTEPPING CONTROL OF "OS4" VTOL SYSTEM

The model (6) developed in section II can be rewritten in a state-space form $\dot{X} = f(X, U)$ by introducing $X = (x_1...x_{12})^T \in \Re^{12}$ as state vector of the system.

$$
\begin{array}{l|l}
x_1 = \phi & x_7 = z \\
x_2 = \dot{x}_1 = \dot{\phi} & x_8 = \dot{x}_7 = \dot{z} \\
x_3 = \theta & x_9 = x \\
x_4 = \dot{x}_3 = \dot{\theta} & x_{10} = \dot{x}_9 = \dot{x} \\
x_5 = \psi & x_{11} = y \\
x_6 = \dot{x}_5 = \dot{\psi} & x_{12} = \dot{x}_{11} = \dot{y}
\end{array} \tag{12}
$$

From (6) and (12) we obtain:

$$
f(X, U) = \begin{pmatrix}
x_2 \\
x_4 x_6 a_1 + x_4 a_2 \Omega + b_1 U_2 \\
x_4 \\
x_2 x_6 a_3 + x_2 a_4 \Omega + b_2 U_3 \\
x_6 \\
x_4 x_2 a_5 + b_3 U_4 \\
x_8 \\
-g + (\cos x_1 \cos x_3)\frac{1}{m} U_1 \\
x_{10} \\
u_x \frac{1}{m} U_1 \\
x_{12} \\
u_y \frac{1}{m} U_1
\end{pmatrix} \tag{13}
$$
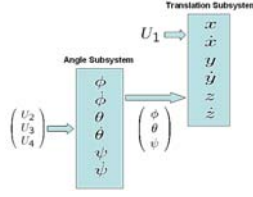
[4]Inertial Measurement Unit

Fig. 7. Connection of the two ideal subsystems of the overall dynamical system described by mapping (13).

With:

$$\begin{array}{ll} a_1 = (I_y - I_z)/I_x & b_1 = l/I_x \\ a_2 = -J_R/I_x & b_2 = l/I_y \\ a_3 = (I_z - I_x)/I_y & b_3 = l/I_z \\ a_4 = J_R/I_y & \\ a_5 = (I_x - I_y)/I_z & \end{array} \tag{14}$$

$$\begin{aligned} u_x &= (\cos x_1 \sin x_3 \cos x_5 + \sin x_1 \sin x_5) \\ u_y &= (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) \end{aligned} \tag{15}$$

It is worthwhile to note in the latter system that the angles and their time derivatives do not depend on translation components. On the other hand, the translations depend on the angles. We can ideally imagine the overall system described by (13) as constituted of two subsystems, the angular rotations and the linear translations, see Fig. 7.

The control scheme advocated for the overall system is then logically divided in a position controller and a rotation controller as schematized in Fig. 8.
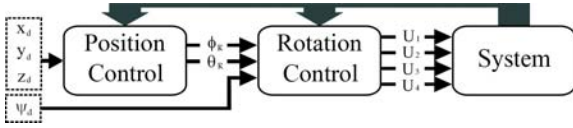

Fig. 8. Control Scheme: One set the desired (x_d, y_d, z_d, ψ_d), the position controller generates the required (ϕ_R, θ_R) to the rotation controller. The measured quantities are then fed-back to both the controllers.

A. Backstepping Control of the Rotations Subsystem

Using the backstepping approach, one can synthesize the control law forcing the system to follow the desired trajectory. Refer to [7] and [8] for more details. For the first step we consider the tracking-error:

$$z_1 = x_{1d} - x_1 \tag{16}$$

And we use the Lyapunov theorem by considering the Lyapunov function z_1 positive definite and it's time derivative negative semi-definite:

$$V(z_1) = \frac{1}{2} z_1^2 \tag{17}$$

$$\dot{V}(z_1) = z_1(\dot{x}_{1d} - x_2) \tag{18}$$

The stabilization of z_1 can be obtained by introducing a virtual control input x_2:

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1 \quad with : \alpha_1 > 0 \tag{19}$$

The equation (18) is then:

$$\dot{V}(z_1) = -\alpha_1 z_1^2 \tag{20}$$

let us proceed to a variable change by making:

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \tag{21}$$

For the second step we consider the augmented Lyapunov function:

$$V(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2) \tag{22}$$

And it's time derivative is then:

$$\begin{aligned} \dot{V}(z_1 z_2) &= z_2(a_1 x_4 x_6 + a_2 x_4 \Omega + b_1 U_2) \\ &\quad - z_2(\ddot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) \\ &\quad - z_1 z_2 - \alpha_1 z_1^2 \end{aligned} \tag{23}$$

The control input U_2 is then extracted ($\ddot{x}_{1,2,3d} = 0$), satisfying $\dot{V}(z_1 z_2) < 0$:

$$U_2 = \frac{1}{b_1}(z_1 - a_1 x_4 x_6 - a_2 x_4 \Omega - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2) \tag{24}$$

The term $\alpha_2 z_2$ with $\alpha_2 > 0$ is added to stabilize z_1. The same steps are followed to extract U_3 and U_4.

$$\begin{aligned} U_3 &= \frac{1}{b_2}(z_3 - a_3 x_2 x_6 - a_4 x_2 \Omega - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4) \\ U_4 &= \frac{1}{b_3}(z_5 - a_5 x_2 x_4 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6) \end{aligned} \tag{25}$$

with:

$$\begin{cases} z_3 = x_{3d} - x_3 \\ z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3 \\ z_5 = x_{5d} - x_5 \\ z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5 \end{cases} \tag{26}$$

B. Backstepping Control of the Linear Translations Subsystem

1) Altitude Control: The altitude control U_1 is obtained using the same approach described in V-A.

$$U_1 = \frac{m}{\cos x_1 \cos x_3}(z_7 + g - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8) \tag{27}$$

with:

$$\begin{cases} z_7 = x_{7d} - x_7 \\ z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7 \end{cases} \tag{28}$$

2) Linear x and y Motion Control: From the model (6) one can see that the motion through the axes x and y depends on U_1. In fact U_1 is the total thrust vector oriented to obtain the desired linear motion. If we consider u_x and u_y the orientations of U_1 responsible for the motion through x and y axis respectively, we can then extract from (15) the roll and pitch angles necessary to compute the controls u_x and u_y satisfying $\dot{V}(z_1 z_2) < 0$. The yaw control is then given as a desired angle (see Fig. 8).

$$\begin{aligned} u_x &= (m/U_1)(z_9 - \alpha_9(z_{10} + \alpha_9 z_9) - \alpha_{10} z_{10}) \\ u_y &= (m/U_1)(z_{11} - \alpha_{11}(z_{12} + \alpha_{11} z_{11}) - \alpha_{12} z_{12}) \end{aligned} \tag{29}$$
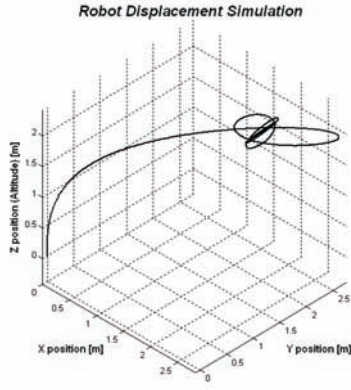
Fig. 9. Simulation: The system has to reach the position $(2, 2, 2)m$ from $(0, 0, 0)m$ and keep the same heading.
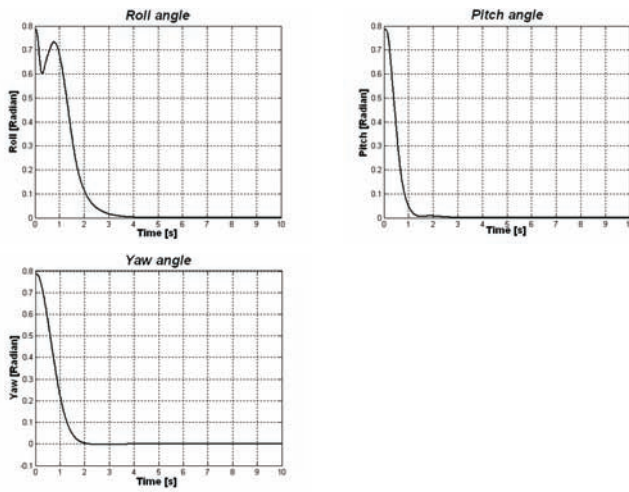

Fig. 10. Simulation: The backstepping controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero.

C. Backstepping Controller Simulation

We performed several simulations on Simulink using the dynamic model (6) with the 12 parameters $(\alpha_1, ..., \alpha_{12})$ controller. The task was to reach the position $x_d = y_d = z_d = 2$ m and $\psi_d = 0$ rad. The simulated performance was satisfactory as showed in Fig. 9. Before to test the controller on the real system which has only the 3D orientation sensor (IMU), we performed different simulations considering only the angular rotations subsystem and its corresponding controller. This controller has only 6 parameters $(\alpha_1, ..., \alpha_6)$, tuned simultaneously using the Nonlinear Control Design blockset (NCD) from the Optimization Toolbox under Matlab. The initial condition was $\pi/4$ rad for the three angles. The results were very satisfactory as shown in Fig. 10.

D. Backstepping Controller on the Real System

In order to validate the control law developed in the previous section, we implemented the controller in C under Linux on a machine running at 450Mhz simulating
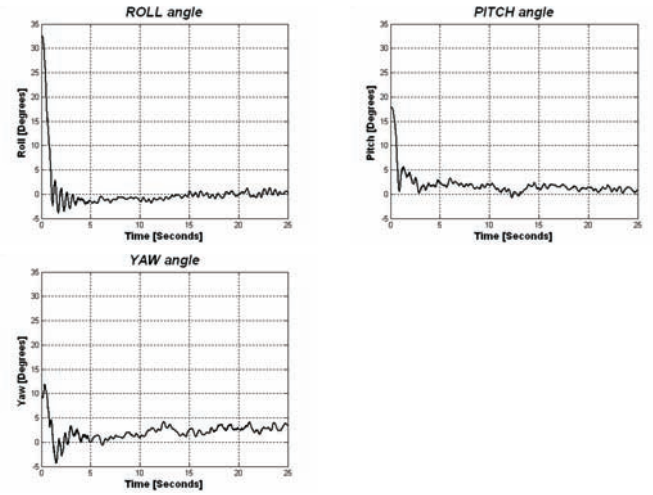

Fig. 11. Experiment: The backstepping controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero. It works well in spite of a hard initial conditions. Few yaw angle drift is observed due to the vibrations and EMI influence on the yaw sensor.

the future integration of a Single Board Computer[5]. We performed several experiments on the real system, were the task was to control the vehicle orientation as showed in Fig. 11. The altitude was then fixed by the test-bench. The initial condition was about 32 degrees for the roll angle and we obtained the stabilization in less than 5 seconds. It was difficult to give the same initial angular speed to the roll angles on the test-bench as in simulation. In spite of the test-bench limitations in term of delays and errors introduced by the tethering system, the experimental results obtained show that the proposed controller is able to stabilize the system even for relatively critical initial conditions.

VI. Sliding-mode Control of "OS4" VTOL System

A. Sliding Control of the Angular Rotations Subsystem

The mapping (13) is partially used to design the sliding-mode controller for the rotations subsystem of the $OS4$ helicopter. The first step in this design is similar to the one for the backstepping approach [9], except for the equation (19) were S_2 (Surface) is used instead of z_2 for more clearance.

$$s_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \tag{30}$$

For the second step we consider the augmented Lyapunov function:

$$V(z_1, s_2) = \frac{1}{2}(z_1^2 + s_2^2) \tag{31}$$

The chosen law for the attractive surface is the time derivative of (30) satisfying $(s\dot{s} < 0)$:

$$
\begin{aligned}
\dot{s}_2 &= -k_1 sign(s_2) - k_2 s_2 \\
&= \dot{x}_2 - \ddot{x}_{1d} - \alpha_1 \dot{z}_1 \\
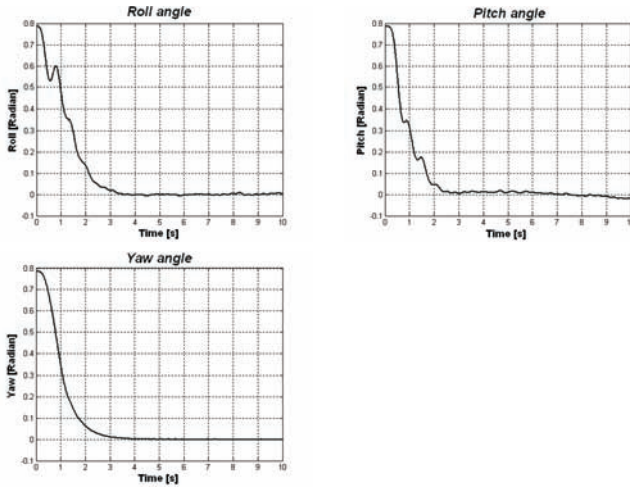&= a_1 x_4 x_6 + a_2 x_4 \Omega + b_1 U_2 - \ddot{x}_{1d} + \alpha_1 (z_2 + \alpha_1 z_1)
\end{aligned}
\tag{32}
$$

Fig. 12. Simulation: The sliding-mode controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero.
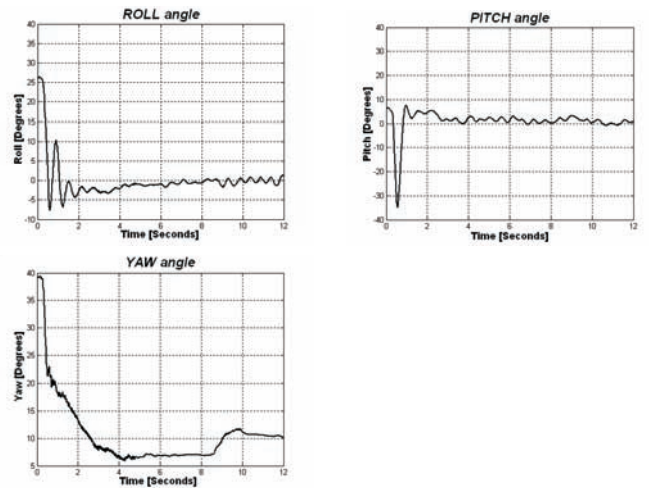

Fig. 13. Experiment: The Sliding-mode controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero. The controller stabilizes well the system for the roll and pitch angles while the shattering effect is present. The big negative overshot in the pitch angle is due to the huge initial condition for the yaw angle.

As for the backstepping approach, the control U_2 is extracted:

$$U_2 = \frac{1}{b_1}(-a_1 x_4 x_6 - a_2 x_4 \Omega - \alpha_1^2 z_1 - k_1 sign(s_2) - k_2 s_2) \tag{33}$$

The same steps are followed to extract U_3 and U_4.

$$U_3 = \frac{1}{b_2}(-a_3 x_2 x_6 - a_4 x_2 \Omega - \alpha_2^2 z_3 - k_3 sign(s_3) - k_4 s_3)$$
$$U_4 = \frac{1}{b_3}(-a_5 x_2 x_4 - \alpha_3^2 z_5 - k_5 sign(s_4) - k_6 s_4) \tag{34}$$

with:

$$\begin{cases} z_3 = x_{3d} - x_3 \\ s_3 = x_4 - \dot{x}_{3d} - \alpha_2 z_3 \\ z_5 = x_{5d} - x_5 \\ s_4 = x_6 - \dot{x}_{5d} - \alpha_3 z_5 \end{cases} \tag{35}$$

B. Sliding-mode Controller Simulation

For these simulations we considered only the angular rotations subsystem in order to be able to verify the development on the real system. The controller above contains 09 parameters $(\alpha_1, ..., \alpha_3, k_1, ..., k_6)$ tuned also using NCD. The initial conditions were $\pi/4$ rad for the three angles as shown in Fig. 12.

C. Sliding-mode Controller on the Real System

The experimental conditions were similar to the ones applied for the backstepping controller, see Fig. 13. The task was to control the vehicle orientation and the altitude was fixed by the test-bench. The initial condition was about 26 degrees for the roll angle and we obtained the stabilization in less than 8 seconds. The experimental results obtained show that the proposed controller is able to stabilize the roll and pitch angles but the shattering effect, even if reduced, disturbs the measurements and this is visible especially for the yaw angle.

VII. CONCLUSION

In this paper, we presented the application of two different control techniques "Backstepping" and "Sliding-mode" to a micro Quadrotor called $OS4$. As it can be seen from the experimental plots, the controller introduced using the sliding-mode approach provides average results. This is partly due to switching nature of the controller which introduces high frequency, low amplitude vibrations causing the sensor to drift. On the other hand, the backstepping controller proves the ability to control the orientation angles in the presence of a relatively high perturbations confirming by the way some previous studies on underactuated systems [4]. Our future work is to develop a fully autonomous vehicle with an enhanced full state backstepping controller. The positive results obtained through this development enhances our knowledge of this highly unstable system and encourages us to continue towards the fully autonomous indoor Quadrotor.

REFERENCES

[1] P. Müllhaupt, "Analysis and control of underactuated mechanical nonminimum-phase systems", Phd thesis, EPFL, 1999.
[2] A. Chriette, "Contribution à la commande et à la modélisation des hélicoptères", Phd Thesis, Université d'Evry, 2001.
[3] S. Sastry, "A mathematical introduction to robotic manipulation", Boca Raton, FL, 1994.
[4] R. Olfati-Saber, "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles", Phd thesis, MIT, 2001.
[5] E. Jucker, "Equations fondamentales des micromoteurs courant continu avec rotor sans fer", Bulletin technique Portescap, La Chaud-de-Fonds, 1974.
[6] P. Pounds, R.Mahony, "Design of a Four-Rotor Aerial Robot", Australasian Conference on Robotics and Automation, Auckland, 2002.
[7] R.Sepulchre, et al, "Constructive Nonlinear Control", Springer, 1997.
[8] I. Fantoni, R.Lozano, "Non-linear control for underactuated mechanical systems", Springer, 2002.
[9] J.J. Slotine, W. Li, "Applied nonlinear control", Prentice-Hall Inc, 1991.