Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

Towards a High–End Unmanned Tri–TiltRotor: Design, Modeling and Hover Control

Christos Papachristos, Kostas Alexis and Anthony Tzes

Abstract— **The design process, development considerations as well as hover modeling and control of a high–end capability unmanned Tri–TiltRotor aerial vehicle are the subjects of this article. Tilt–Rotor UAVs efficiently perform as fixed-wing aircraft and as rotorcrafts, being capable of both high–endurance high–speed flight as well as of vertical take-off and landing and precision manoeuvring. The additional integration of high–end computational power and sensors on such a type of aircraft provide a UAV platform with multiple operational usefulness capabilities. The system's hovering nonlinear dynamics and linearized model based on an identification of its actuation subsystem are presented, along with hovering attitude stabilization experimental results.**

I. INTRODUCTION

Recent Unmanned Aerial Vehicle (UAV) designs are characterized by their high-level automated operation and exhibit important qualities and capabilities such as with withstanding environmental perturbation [1, 2]. This has allowed for their utilization in a wide area of applications, such as buildings inspection [3], and border interdiction detection [4].

Current research goals include the development of specialized air vehicles, capable of coping with specific mission requirements by posing a number of flight advantages according to their projected use. Environmental perception subsystems are additionally required in order to achieve autonomous operation capabilities. Operation scenarios requiring: a) close-range specific area inspection and b) wide area, high-speed aerial surveillance and monitoring capabilities call for specifically designed UAV systems. Applications composed of a target acquisition phase in unknown, unstructured large areas and of an on-site intervention phase (to provide assistance or survey at close range) can benefit from a specialized system design. In the particularly frequent phenomenon of natural disasters occurring in wide areas, the life-saving task of authorities can be aided by UAV systems capable of: a) covering the surveillance of large areas at high speeds while retaining their power consumption at a minimum to achieve prolonged flight times, b) being directed to constrained area to carry out close-range tasks.

Guided by the aforementioned principles, a special UAV design illustrated in Figure 1 has been developed in the University of Patras (UPAT). Its crucial characteristics include the capability to execute flight mode conversion manoeuvres, from rotorcraft hovering mode to fixed-wing longitudinal flight mode and conversely by controlling the rotation of its main rotors relative to its airframe wing axis. The principal concept behind the design of the UPAT Tri–TiltRotor (UPAT–TTR) is the development of a UAV-experimental platform characterized by high quality equipment, robust assembly and a high-end interface allowing for seamless future system expansion capabilities (by incorporating advanced sensorial equipment).

In this article, the issues and considerations towards the implementation of the aforementioned unmanned system are addressed. The article is structured as follows: In Section II the design of the UPAT–TTR is presented. In Section III the system modeling process is presented. In Section IV experimental results of the system's attitude stabilization response are presented. The article is concluded in Section V.

Fig. 1. The UPAT Tri–TiltRotor experimental platform

II. UPAT–TTR SYSTEM DESIGN

The complete setup of the UPAT–TTR, depicted in Figure 2, consists of the hardware components as well as the software implementation. The main components of our UAV experimental platform are: a) the Actuators and their Control System, b) the Sensors and c) the Main Control Unit.

A. Actuators

The system is composed of 3 actuator types. The two main rotors are positioned on the wing axis and consist of two low-KV, high-power (max power of 800W each) DC brushless motors and two 3-bladed 13x8" propellers. The tail rotor

C. Papachristos and A. Tzes are with the Electrical and Computer Engineering Department, University of Patras, Eratosthenous 6, Rio 26500, Greece papachric@ece.upatras.gr

K. Alexis is with the Autonomous Systems Lab, Department of Mechanical and Process Engineering, ETH Zurich, Tannenstrasse 3, 8092, Switzerland alexisk@ethz.ch

This work was partially supported by the John S. Latsis Public Benefit Foundation, Greece, 2010-2011

978-1-4673-2529-5/12/$31.00 ©2012 IEEE

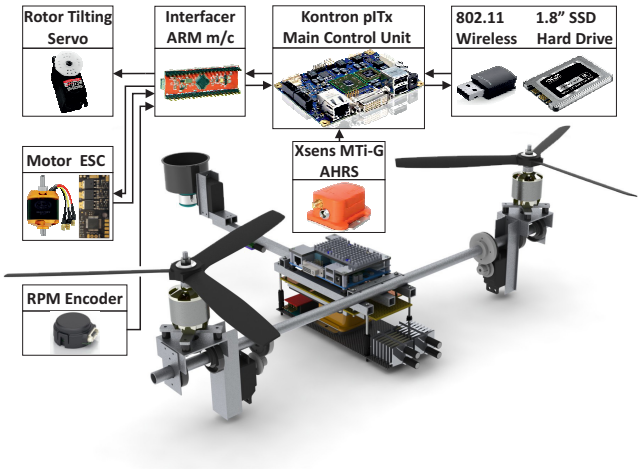

Fig. 2. UPAT–TTR hardware setup

consists of a Ducted Fan system, driven by a high-RPM, small-size brushless DC (BLDC) motor. The tail Ducted Fan system was selected so as to minimize the fuselage volume for better fixed-wing mode flight efficiency. Three high-torque servos are responsible for the positioning of the rotors relative to the UPAT–TTR's body. The tail rotor is mounted directly onto the tail servo, while the main rotors are rotated via a gear-train mechanism. This design principle serves four purposes: a) via the gear-reduction mechanism an even higher holding/driving-torque of the servomotors is achieved, b) the servos are constantly positioned behind the rotors -as they rotate along with them- serving for an uninhibited airflow c) a linearly controlled rotor positioning angle (solutions involving clevis/hinged designs do not allow for the precision control over a wide angle window while with this design a smooth and continuously rotating rotor over a near-360 degree angle has been achieved) and d) the moment of inertia of each rotor subsystem around the wing axis becomes more balanced resulting in reduced unbalanced vibrations.

All actuators are controlled via a specifically programmed ARM Cortex-M3-based microprocessor. This microprocessor serves mainly as a sensor and control message router, ensuring the fast interfacing between the Main Control Unit and the UPAT–TTR's actuators and sensors. This microprocessor forwards the control signals to the actuator's low-level controllers. The servos are controlled via PWM signals read by their casing-enclosed microcontroller while the motors are controlled by Electronic Speed Controllers (ESCs) utilizing the high-speed I2C communication protocol. The ESCs main characteristics include high-current operation, 11-bit command precision and onboard input current and operating temperature reporting.

B. Sensors

The most crucial sensor for UAV systems is the Attitude Heading and Reference System (AHRS). The sensor installed onto the UPAT–TTR is the commercially available Xsens MTi-G Inertial Measurement Unit (IMU), which is is capable of delivering a static accuracy of less than 0.5 degrees for the roll and pitch and less than 1 degree for the heading angle at 120 Hz by utilizing on-board Kalman Filtering-based data processing.

A sonar sensor, utilizing the I2C bus for communication is additionally installed to provide indoor altitude readings, while the MTi-G IMU's barometric sensor can be utilized for higher altitudes.

Additionally, the commercially available USDigital E4P small-sized incremental encoders have been installed onto each motor's shaft to enable rpm feedback for our rotors. These encoders output 100 pulses/revolution, a signal which is acquired by the ARM-microcontroller through interrrupt-based routines and translated into revolutions/min.

C. Main Control Unit

The UPAT–TTR is equipped with the Intel Atom Z530 1.6 GHz CPU-based commercially available Kontron pITx Single Board Computer. It is essentially a full-grade PC system designed in the pico-ATx form-factor which makes it ideal for installation onto small-sized UAVs. The unit is also equipped with 2 GB of RAM memory and a slim-sized lightweight 40GB 1.8" SSD Hard Drive. The unit offers multiple USB-based connectivity which allows for seamless interfacing with a number of COTS peripherals. Utilizing this feature, a WiFi 802.11n wireless adapter is installed providing connectivity utilized for telemetry and wireless communication with the Ground Control Station (GCS).

The Main Control Unit is responsible for performing data processing and control signal generation tasks. The UPAT–TTR's peripherals, namely the MTi-G IMU and the ARM-microcontroller are connected onto the Kontron SBC which in operation: a) gathers all available sensor signals regarding the system's state (attitude & heading from the IMU, and altitude, motor rpm and ESC debug information from the ARM-microcontroller), b) makes all necessary computations and other wireless communication-related functions and c) transmits control commands to the ARM-microcontroller which in turn is tasked to forward these commands via its low-level data buses (PWM and I2C) to the actuator controllers.

D. Design Process & Considerations

The material utilized for the construction of the body framework is aluminum, which in the shape of hollow rods and bars offers strength & flexibility while weighing not more than 0.25 kg. To achieve a tightly-packed resulting assembly, 3D CAD software was used in the design process, which significantly reduced the need for iterative design/assembly as the result of each modification was immediately visualized and its effects into rigid body-related qualities quickly calculated.

An important consideration for the design process is the necessity to reduce rotor-induced vibrations as much as possible. It was observed during testing that while high-frequency vibrations are efficiently rejected by the IMU's filtering, high-amplitude low-frequency vibrations lead to

noisy calibrated sensor data and drifting of the reported system attitude estimations. This issue has been addressed by a number of steps. Initially the most common source of rotor vibrations –the propellers– were manually balanced by iteratively slowly sanding off excess material off each blade and rebalancing it. Additionally, the motors were mounted on silicone-rubber stand-offs in order to isolate the vibration to an extent. Finally the IMU was tightly attached onto the body framework mass, which significantly minimized the impact of the motors' powerful vibrations.

E. Software Implementation

The UPAT–TTR's software architecture is based on the open-source Ubuntu Linux Operating System (OS) which is installed on the SSD Hard Drive. Using an OS comes with significant advantages, the most common of which are real-time execution control, multi-threading, a wide variety of predeveloped software tools and commercial device drivers to suit various applications. The latter significantly adds to the expandability potential of our experimental platform allowing for the seamless future integration of advanced peripherals, such as vision-based environment perception devices as well as the utilization of robotics-oriented software tools developed by the international robotics community.

The core design tool of the system is the Matlab Real-Time Workshop (RTWS) along with the Embedded Code Generation toolbox[5]. It is a software framework aimed at designing real-time control schemes while allowing for fast control design using the Simulink Graphical User Interface and also providing built-in support for most of the tools available in Matlab. The aforementioned are ideal features for the design of advanced control schemes requiring data processing, real-time datalogging and data analysis. An additional feature is the implementation of an External Mode, in which the control scheme may be designed and simulated onto a remote computer (GCS) and afterwards it is compiled into a Real-Time executable file which is wirelessly uploaded to the experimental platform.

The controller software scheme developed for the UPAT–TTR is depicted in Figure 3. It is composed of two separate processes, namely a) the *MATLAB* executable produced by the RTWS Coder and b) the *INTERFACER* executable. The latter creates a separate execution thread for each Input/Output peripheral connected to the system. This suits the need for the interfacing of peripherals running at different loop speeds (for instance the IMU provides Kalman state update data at 120 Hz while the ARM-microcontroller runs at 250 Hz). Data is transferred to and from the *MATLAB* process with minimum overhead using UNIX inter-process communication capabilities. In the *MATLAB* process the data acquired from the system's sensors are processed into signals suitable for the generation of actuator control actions. Such processing includes the minimum-order low-pass filtering of noisy measurement data and median filtering of out-of-place signal values. One more task of this process is the logging of data manipulated during its execution, as well as the real-time graphing of that data on demand. As already described, these processes are running in External Mode on the experimental platform during its operation, while the GCS computer is connected wirelessly via TCP protocol. This connection enables the seamless data transfer between the GCS and the experimental platform and it is used for wireless telemetry and for updating various parameters of our system's controller in operation time.

Finally, the control action signals computed by the main control process are output to the *INTERFACER* process and consequently forwarded to the ARM-microcontroller which handles the low-level communication with the Actuator Control Subsystem.

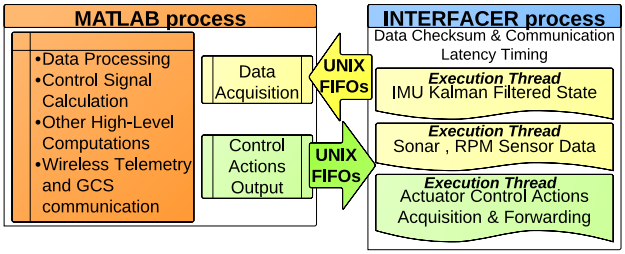

Fig. 3. UPAT–TTR controller scheme

III. UPAT–TTR SYSTEM MODELING

The UPAT–TTR's hovering mode operation principles are illustrated in Figure 4. The total thrust is produced by the 3 rotors and it is used to control the hovering altitude. The rolling motion is controlled by the differential thrusting of the main 2 rotors, the pitching motion by the tail rotor's thrust and the yawing motion by rotating the tail rotor perpendicular to the body axis and thus producing a lateral thrust component. The main rotors are not actively controlled and will be used in the future to control the vehicle in forward flight and to achieve conversion from rotorcraft to fixed-wing aircraft flight mode.

Let $\mathbf{B} = \{B_x,\ B_y,\ B_z\}$ be the coordinates Body-Fixed Frame (BFF) and $\mathbf{E} = \{N,\ E,\ D\}$ be the Earth North-East-Down (NED) Local Tangential Plane (LTP) as depicted in Figure 4.

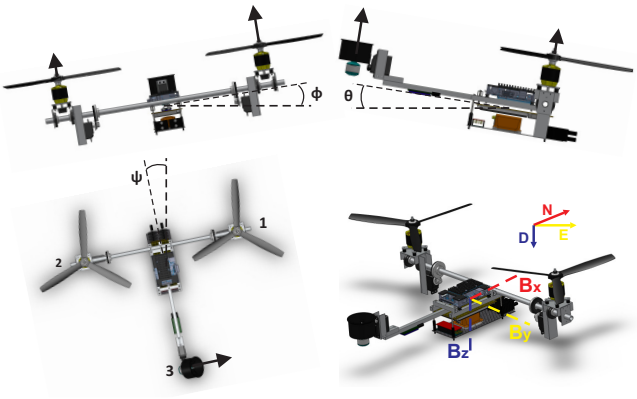

Fig. 4. Hovering Operation Principles and Coordinate Frames

Also let $\mathbf{U} = \{u,\ v,\ w\}$ be the BFF-based and aligned linear velocity vector, $\Omega = \{p,\ q,\ r\}$ the BFF-based angular rotation rate vector, $\mathbf{X} = \{x,\ y,\ z\}$ the LTP-based linear velocity vector and $\Theta = \{\phi,\ \theta,\ \psi\}$ the LTP-based angular rotation rate vector. The transformation of linear velocities and angular rates from the BFF to the LTP are provided by the skew-symmetric rotation matrix $\mathbf{R}^{B\to W}$ and the Tait-Bryan angles transformation matrix $\mathbf{J}^{B\to W}$ [6].

$$\dot{\mathbf{X}} = \mathbf{R}^{B\to I}\mathbf{U}\ , \tag{1}$$
$$\dot{\Theta} = \mathbf{J}^{B\to I}\Omega\ . \tag{2}$$

A. Hovering Mode Dynamics

In deducting the system's non-linear dynamics the Newton-Euler formulation was followed:

$$\mathbf{F}^B = m\dot{\mathbf{U}} + \Omega\times(m\mathbf{U})\ , \tag{3}$$
$$\mathbf{M}^B = \mathbf{I}\dot{\Omega} + \Omega\times(\mathbf{I}\Omega)\ , \tag{4}$$

where $\mathbf{F}^B$ the BFF-based total Force vector, $\mathbf{M}^B$ the BFF-based total Moment vector, m the total mass and $\mathbf{I}$ the moment of inertia matrix.

Let rotors 1, 2, 3 be the right, left and tail rotor in respect to the BFF positive x–axis and let γ_i be the angle between the BFF-aligned vertical axis and each rotor. Also let d_{i_j} be the geometric distance component from the Center of Mass (COM) to the ith rotor's tilt rotation axis in the j direction of the BFF, and h_i the distance of the ith rotor's tilt rotation axis from its propeller hub.

Gravitational Forces: Let $\mathbf{G}^B = (\mathbf{R}^{B\to I})^{-1}\mathbf{G}^W$, be the BFF-based gravity vector, where $\mathbf{G}^E = [0\ 0\ mg]^T$, $(g = 9.81 m/s^2)$ the LTP-based gravity vector.

Thrust Forces:

$$\mathbf{F}_1^B = \begin{bmatrix} F_1^{B_x} & F_1^{B_y} & F_1^{B_z}\end{bmatrix}^T = \begin{bmatrix} F_1\sin(\gamma_1)\\ 0\\ -F_1\cos(\gamma_1)\end{bmatrix}\ , \tag{5}$$

$$\mathbf{F}_2^B = \begin{bmatrix} F_2^{B_x} & F_2^{B_y} & F_2^{B_z}\end{bmatrix}^T = \begin{bmatrix} T_F\sin(\gamma_2)\\ 0\\ -F_2\cos(\gamma_2)\end{bmatrix}\ , \tag{6}$$

$$\mathbf{F}_3^B = \begin{bmatrix} F_3^{B_x} & F_3^{B_y} & F_3^{B_z}\end{bmatrix}^T = \begin{bmatrix} F_3\sin(\gamma_3)\\ 0\\ -F_3\cos(\gamma_3)\end{bmatrix}\ , \tag{7}$$

$$\mathbf{F}_i = k_{L,i}\omega_{p,i}^2\ , \tag{8}$$

where $\omega_{p,i}$ is the ith propeller's angular speed

Propeller Gyroscopic Moments: The rotating propellers are the subject of a gyroscopic effect when tilting:

$$\mathbf{M}_{G,i} = \mathbf{I}_{p,i}(\dot{\Gamma}_i\times\omega_{p,i})\ , \tag{9}$$

where $I_{p,i}$ the ith propeller's inertia and $\Gamma_i = [0\ \dot{\gamma}_i\ 0]^T$.

Propeller Drag Moments: The moment exerted on the hub of each rotor during rotation due to the propeller's drag:

$$\mathbf{M}_{D,i} = k_D\omega_{p,i}^2\ . \tag{10}$$

Moment Origins Position Vectors:

$$\mathbf{r}_1^B = \begin{bmatrix} r_1^{B_x} & r_1^{B_y} & r_1^{B_z}\end{bmatrix}^T = \begin{bmatrix} d_{1,2_x} + h_{1,2}\sin(\gamma_1)\\ d_{1,2_y}\\ -d_{1,2_z} - h_{1,2}\cos(\gamma_1)\end{bmatrix}\ , \tag{11}$$

$$\mathbf{r}_2^B = \begin{bmatrix} r_2^{B_x} & r_2^{B_y} & r_2^{B_z}\end{bmatrix}^T = \begin{bmatrix} d_{1,2_x} + h_{1,2}\sin(\gamma_2)\\ -d_{1,2_y}\\ -d_{1,2_z} - h_{1,2}\cos(\gamma_2)\end{bmatrix}\ , \tag{12}$$

$$\mathbf{r}_3^B = \begin{bmatrix} r_3^{B_x} & r_3^{B_y} & r_3^{B_z}\end{bmatrix}^T = \begin{bmatrix} -d_{3_x}\\ h_3\sin(\gamma_3)\\ -d_{3_z} - h_3\cos(\gamma_3))\end{bmatrix}\ . \tag{13}$$

B. System Linearization

The linearized system dynamics were derived based on the quasi-steady representation augmented by the actuators' dynamics. The state vector is: $\mathbf{X} = \begin{bmatrix}\xi\ |\ \eta\ |\omega\end{bmatrix}^T$, where

$$\xi = \begin{bmatrix} x\ u\ y\ v\ z\ w\end{bmatrix}^T\ , \tag{14}$$
$$\eta = \begin{bmatrix}\phi\ p\ \theta\ q\ \psi\ r\end{bmatrix}^T\ , \tag{15}$$
$$\omega = \begin{bmatrix}\omega_{p,1}\ \gamma_1\ \omega_{p,2}\ \gamma_2\ \omega_{p,3}\ \gamma_3\end{bmatrix}^T\ . \tag{16}$$

The linearized model is extracted around a generic hovering operation point of the $\mathbf{X}_\eta = \begin{bmatrix}\eta\ |\ \omega\end{bmatrix}^T$ attitude state subsystem:

$$\mathbf{X}_{\eta\mathbf{0}} = \begin{bmatrix} 0\ p_0\ 0\ q_0\ 0\ r_0\ \omega_{p,1_0}\ 0\ \omega_{p,2_0}\ 0\ \omega_{p,3_0}\ 0\end{bmatrix}^T\ . \tag{17}$$

Let

$$\dot{\omega}_{p,i} = A_{m,i}\ \omega_{p,i} + B_{m,i}\ U_{m,i}\ , \tag{18}$$
$$\dot{\gamma}_i = A_{s,i}\ \gamma_i + B_{s,i}\ U_{s,i}\ , \tag{19}$$

be the state-space representations of the rotor and servo subsystem dynamics respectively ($U_{m,i}$, $U_{s,i}$ are the motor and servo control inputs). Also let I_{xx}, I_{yy}, I_{zz} be the rigid body moments of inertia around the BFF xx', yy', zz' axis respectively. The resulting linearized system dynamics are the following:

$$\dot{\mathbf{X}}_\eta = \mathbf{A}_\eta\ \mathbf{X}_\eta + \mathbf{B}_\eta\ \mathbf{U}\ , \tag{20}$$
$$\mathbf{U} = \begin{bmatrix} U_{m,1}\ U_{s,1}\ U_{m,2}\ U_{s,2}\ U_{m,3}\ U_{s,3}\end{bmatrix}^T\ , \tag{21}$$
$$\mathbf{A}_\eta = \left[\begin{array}{c|c} \mathbf{A_1} & \mathbf{A_2}\\ \mathbf{0}_{[\mathbf{6}\times\mathbf{6}]} & \mathbf{A_3}\end{array}\right]\ , \tag{22}$$

$$\mathbf{A_1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & \frac{I_{yy}-I_{zz}}{I_{xx}}r_0 & 0 & \frac{I_{yy}-I_{zz}}{I_{xx}}q_0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & -\frac{I_{xx}-I_{zz}}{I_{yy}}r_0 & 0 & 0 & 0 & -\frac{I_{xx}-I_{zz}}{I_{yy}}p_0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & \frac{I_{xx}-I_{yy}}{I_{zz}}q_0 & 0 & \frac{I_{xx}-I_{zz}}{I_{yy}}p_0 & 0 & 0\end{bmatrix}\ ,$$

$$\mathbf{A_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ \frac{-c_{L,1_0}d_{1,2_y}}{I_{xx}} & 0 & \frac{c_{L,2_0}d_{1,2_y}}{I_{xx}} & 0 & 0 & \frac{F_{3_0}d_{3_z}}{I_{xx}}\\ 0 & 0 & 0 & 0 & 0 & 0\\ \frac{c_{L,1_0}d_{1,2_x}}{I_{yy}} & \frac{-F_{1_0}d_{1,2_z}}{I_{yy}} & \frac{c_{L,2_0}d_{1,2_x}}{I_{yy}} & \frac{-F_{2_0}d_{1,2_z}}{I_{yy}} & \frac{-c_{L,3_0}d_{3_x}}{I_{yy}} & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \frac{-F_{1_0}d_{1,2_y}}{I_{zz}} & 0 & \frac{F_{2_0}d_{1,2_y}}{I_{zz}} & 0 & \frac{-F_{3_0}d_{3_x}}{I_{zz}}\end{bmatrix}\ ,$$

$$\mathbf{A_3} = \mathbf{diag}(A_{m,1}\ ,\ A_{s,1}\ ,\ A_{m,2}\ ,\ A_{s,2}\ ,\ A_{m,3}\ ,\ A_{s,3})\ ,$$

where $c_{L,i_0} = 2k_{L,i}\omega_{p,i_0}$ and $F_{i_0} = k_{L,i}\omega_{p,i_0}^2$,

$$\mathbf{B}_\eta = \begin{bmatrix} \mathbf{0}_{[6\times 6]} \\ \mathbf{B_1} \end{bmatrix}, \quad (23)$$

$$\mathbf{B_1} = \mathbf{diag}(B_{m,1}\ ,\ B_{s,1}\ ,\ B_{m,2}\ ,\ B_{s,2}\ ,\ B_{m,3}\ ,\ B_{s,3})\ ,$$

where **diag**() the $[6\times 6]$ diagonal matrix of the elements in the parenthesis.

The values of the system's physical parameters were estimated using 3D CAD design software and are presented in Table I.

TABLE I

UPAT–TTR PARAMETERS

Property	Value	Units		Property	Value	Units
$d_{1,2_x}$	0.06294	m		I_{xx}	0.057272815	$kg\cdot m^2$
$d_{1,2_y}$	0.25	m		I_{yy}	0.032133169	$kg\cdot m^2$
$d_{1,2_z}$	0.006	m		I_{zz}	0.081910226	$kg\cdot m^2$
$h_{1,2}$	0.099	m		m	2.045	kg
d_{3_x}	0.33756	m		$k_{L,1}$	0	N/rpm^2
d_{3_y}	0.0	m		$k_{L,2}$	0	N/rpm^2
d_{3_z}	0.021	m		$k_{L,3}$	0	N/rpm^2
h_3	0.0355	m				

C. Actuators' Identification

The actuators' parameters were identified based on a frequency-domain approach. The system identification process consists of three distinct processes, namely: a) the data quality check and preparation, b) the grey-box identification process and finally c) the model validation both in frequency and time domain. The aforementioned steps are illustrated in Figure 5.

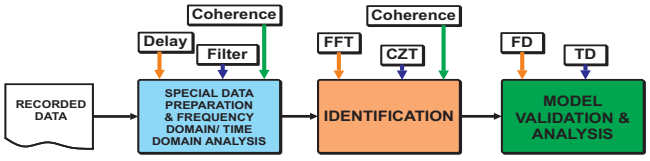

Fig. 5. Actuators system identification flowchart

Due to the frequency-domain approach of the identification process, the motors were excited using chirp signals covering the actuators' effective bandwidth. The motor angular velocity is measured via the rpm encoders, whereas the produced thrust was measured using a laboratory weighting scale and was utilized in experimentally defining the $k_{L,i}$ for each i propeller as marked in Table I. During the data preparation process, the data are checked mainly based on the coherence γ^2_{xy} between the excitation signal (ensuring good and fruitful excitation) and the random error. The experience of the identification community working in rotorcrafts [7] indicates that a good excitation level is achieved when the coherence is higher than 0.65 within the desired frequency area and is considered to be of very high quality when the coherence is higher than 0.85. In Figure 6, a plot that shows the unbiased (with mean values removed) experimental output and input, the coherence between the excitation signal and the response as well as the random error noise for the case of the right main rotor when excited by a chirp signal with frequencies between 3.5Hz and 5.5Hz is presented. As clearly indicated from the very high coherence, the recorded dataset contains fruitful information about the system, while also indicating that the system can be accurately approximated as a linear system (almost zero coherence in the frequency areas that were not excited by the input).

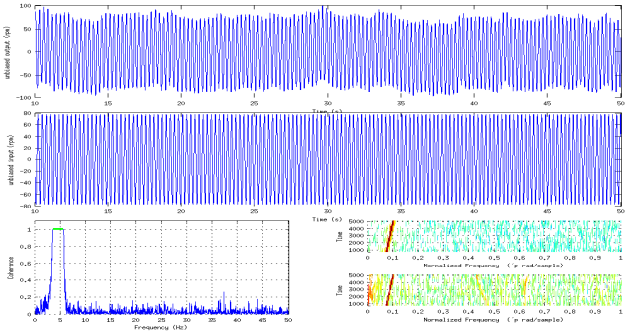

Fig. 6. Right main rotor identification data quality check.

Once the appropriate datasets are selected, the identification process is executed. The computation of the frequency response from the time-domain recorded data was performed using the Fast Fourier Transform. Once the frequency responses are computed the first-order transfer function parameters are estimated in a Maximum Likelihood optimization fashion.

Based on this process, the following three first-order transfer functions were identified:

$$\begin{aligned} G(s) &= \frac{K_{P_{M_i}}}{(1+T_{P_{M_i}}s)} e^{-T_{d_{M_i}}s},\ i\rightarrow[1,2,3]\ , \\ K_{P_{M_i}} &= [4.4375,\ 4.4007,\ 42.873]\ , \\ T_{P_{M_i}} &= [0.1867,\ 0.1925,\ 0.1619]\ , \\ T_{d_{M_i}} &= [0.03453,\ 0.03465,\ 0.0905]\ , \end{aligned}$$

where M_i corresponds to the ith Motor. These first-order transfer functions are directly equivalent to the first-order motor dynamics as modeled in (18).

The slight difference in the parameters of the ideally identical right and left motor is related to the propellers' differences as well as the motor and speed controller possible parameter variation. It must be noted that time-delays were introduced to capture the dynamics of the actual system. The presence of time delays was also examined during the data quality check based on an algorithm that tries to fit ARX models with different time-delays [8]. Considering this fact, the state-space representation is refined so that the input time delays are taken into account. Finally, a validation result is presented in Figure 7.

Both from a) the 79.56% fitting value, computed as $(\hat{y}-y)/(y-\bar{y})$ with $\hat{y}$ the estimated response, y the experimental output and $\bar{y}$ its mean value, and b) the coherence which is 0.99 it is clearly concluded that the identified model effectively captures the system dynamics, whereas the first order

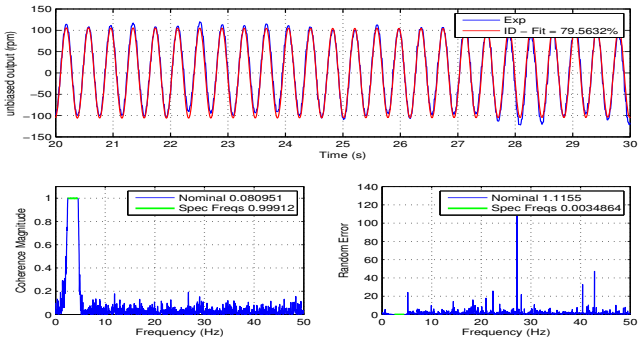

Fig. 7. Right motor identified model, time-domain fit, and coherence validation check.

linear approach turns out to be a sufficient way to model the ESC/BLDC/Propeller. Similar results were achieved for the left main rotor and the tail ducted fan rotor. Based on the response of the servo dynamics they were approximated as input delays, while the stability derivatives $A_{s,1}$, $A_{s,2}$, $A_{s,2}$ were neglected:

$$T_{d_{s_i}} = [0.01843,\ 0.01810,\ 0.01509]\ ,$$

where s_i corresponds to the ith servo.

IV. EXPERIMENTAL STUDIES

The designed UAV's ability to perform attitude stabilization around the hovering point of the augmented state vector:

$$\tilde{\mathbf{X}}_\eta = [\phi\ \dot{\phi}\ \ddot{\phi}\ \theta\ \dot{\theta}\ \ddot{\theta}\ \psi\ \dot{\psi}\ \omega_{p,1}\ \gamma_1\ \omega_{p,2}\ \gamma_2\ \omega_{p,3}\ \gamma_3]\ , \quad (24)$$

$$\tilde{\mathbf{X}}_{\eta\mathbf{0}} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \omega_{p,1_0}\ 0\ \omega_{p,2_0}\ 0\ \omega_{p,3_0}\ 0]\ , \quad (25)$$

was experimentally tested using a PD-dD (PD & double Derivative) control scheme illustrated in Figure 8 which was configured via an iterative ad-hoc tuning process aided by linear system analysis. The controller form is as follows:

$$\mathbf{Y_{roll}} = K_{p_{roll}}\phi + K_{d_{roll}}\dot{\phi} + K_{dd_{roll}}\ddot{\phi}\ , \quad (26)$$

$$\mathbf{Y_{pitch}} = K_{p_{pitch}}\theta + K_{d_{pitch}}\dot{\theta} + K_{dd_{pitch}}\ddot{\theta}\ , \quad (27)$$

$$\mathbf{Y_{yaw}} = K_{p_{yaw}}\psi + K_{d_{yaw}}\dot{\psi} + K_{dd_{yaw}}\ddot{\psi}\ , \quad (28)$$

$$\mathbf{U} = [Y_{roll}\ 0\ -Y_{roll}\ 0\ Y_{pitch}\ Y_{yaw}]^T\ . \quad (29)$$

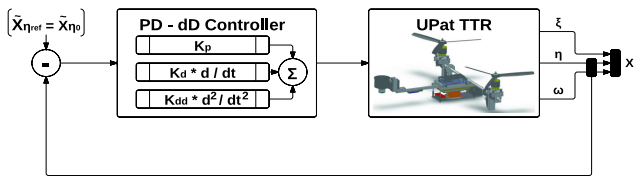

Fig. 8. UPAT–TTR PD–dD Controller Scheme

Based on the PD–dD controller gains summarized in Table II the closed-loop eigenvalues were placed in positions that led to well-damped stable dynamics, small effective time constants, small overshoot and a bandwidth which is sufficient for hovering operation. Table III summarizes the values of these properties for the closed-loop pitch dynamics. Similar values hold for the roll closed-loop dynamics.

TABLE II

UPAT–TTR PD–DD CONTROLLER GAINS

Controller	K_p	K_d	K_{dd}
Roll	350	60	8
Pitch	70	40	6
Yaw	60	120	2

TABLE III

PITCH CLOSED-LOOP ANALYSIS

Property	Value
Damping Ratios	[0.5673, 0.5673, 1]
Bandwidth	5.1579 rad/s
Time Constant	0.16
Overshoot	22%

The experimental results presented in Figure 9 demonstrate that the incorporated components and software framework is running efficiently and achieving the requested operation.

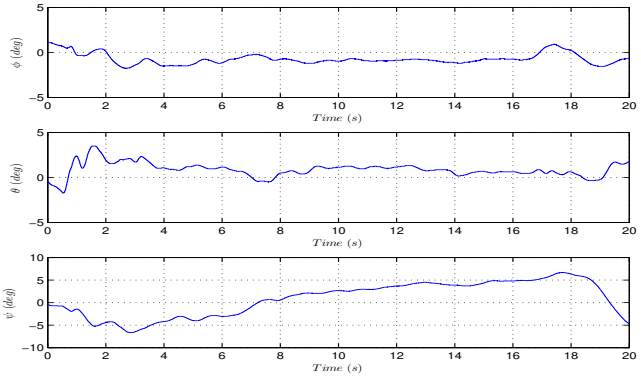

Fig. 9. Hovering Operation using a PD–dD Control Scheme

V. CONCLUSIONS

A high-end capability-enabled Tri–TiltRotor experimental UAV platform was developed, a system model was extracted and a hover stabilization controller was successfully implemented.

REFERENCES

[1] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances," *Control Engineering Practice*, vol. 19, no. 10, pp. 1195 – 1207, 2011.

[2] K. Alexis, C. Papachristos, G. Nikolakopoulos and A. Tzes, "Model predictive quadrotor indoor position control," in *Control Automation (MED), 2011 19th Mediterranean Conference on*, Corfu, Greece, june 2011, pp. 1247 –1252.

[3] N. Metni and T. Hamel, "A UAV for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, vol. 17, no. 1, pp. 3–10, November 2007.

[4] A. Girard, A. Howell, and J. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in *43rd IEEE Conference on Decision and Control*, vol. 1, Atlantis, Paradise Island, Bahamas, December 2004, pp. 620–625 Vol.1.

[5] *Real-Time Workshop^{TM} 7 Datasheet*, The Mathworks Inc.

[6] The MathWorks Inc., *Aerospace Blockset 3 User's Guide*, 2009.

[7] Mark B. Tischler, Robert K. Remple, *Aircraft and Rotorcraft System Identification: Engineering methods with Flight-Test examples*. American Institute of Aeronautics and Astronautics (AIAA).

[8] The MathWorks Inc., *MATLAB System Identification Toolbox*, 2010.