17th Mediterranean Conference on Control & Automation
Makedonia Palace, Thessaloniki, Greece
June 24 - 26, 2009

Design and Implementation of a Structured Flight Controller for a 6DoF Quadrotor Using Quaternions*

Emanuel Stingu, Frank Lewis
Automation & Robotics Research Institute
The University of Texas at Arlington
Arlington, Texas USA
pestingu@arri.uta.edu, lewis@uta.edu

Abstract—**This paper describes a practical implementation of simple control algorithms on a 6DoF quadrotor flying in an uncontrolled environment and being equipped with inexpensive sensors. A significant number of control algorithms that apply dynamic inversion or backstepping techniques on simplified state variable models of the vehicle dynamics are present in the literature, but they are only tested in simulations where real-life issues like sensor noise and precision, vibrations, measurement reference frames, and modeling errors are not included completely or even partially. Here it is shown that if the practical implementation is done correctly, even simple PD controllers can ensure the stability of the quadrotor platform in hover.**

Keywords - UAV; quadrotor; quaternions; control; model.

I. Nomenclature

$\boldsymbol{\omega}_b$	angular velocity resolved to body frame (rad/s)
$\mathbf{q}$	attitude quaternion
$\mathbf{V}_b$	velocity in body frame (m/s)
$\mathbf{R}_e$	position in the inertial frame (m)
$\mathbf{I}_{nb}$	moment of inertia tensor $(kg \cdot m^2)$
F_z	total force of the rotors on the z axis (N)
$M_x - M_z$	total rotor moments along each axis $(N \cdot m)$
$F_1 - F_4$	thrust force for each rotor (N)
Q	rotor torque $(N \cdot m)$
T	rotor thrust force (N)
D	rotor diameter (m)
d	offset of each rotor from the center of mass (m)
n_i	rotation frequency of the rotors (Hz)
Ω_i	rotation speed of the rotors (rad/s)

II. Introduction

Quadrotor helicopters have become popular for research in UAV control due to their relatively simple model and the low-cost involved in operating the experimental platforms. Many groups were successful in developing autonomous quadrotor vehicles. Until only a few years ago, good results were obtained exclusively by using tethers or motion guides, or by having precise external sensors to track the attitude and position [1–3]. Today there are a few projects that are able to do autonomous indoor or outdoor flight using only on-board sensors for attitude estimation and without needing any motion-constraining device. The project in [4] uses a commercially-available remote-controlled vehicle on which an IMU and a DSP board were installed. Only the attitude is controlled as there is no mechanism to measure the position of the quad-rotor. Quaternion representation is used for the attitude and backstepping techniques are applied to drive the error quaternion to zero. The OS4 project [5] uses integral backstepping for full control of attitude, altitude and position on a custom platform. The attitude is sensed using an IMU, the altitude using an ultrasound range sensor and the horizontal position using an external vision system. Finally, the vehicle developed by the STARMAC project [6] has the ability to fly outdoor. A comprehensive model was developed that includes the induced air velocity and the effects of blade flapping in translational flight. Unlike the OS4, a precise differential GPS unit can be optionally used as a replacement of the vision system to measure the position and the linear velocities.

This paper presents the quadrotor platform developed at the Automation & Robotics Research Institute to be used for research in multi-vehicle control. A simplified model is developed and its parameters are identified. It is used to design a proportional controller for stabilizing the attitude. Quaternion representation is employed.

III. Quadrotor Testbed

The testbed for the experiments consists in a quadrotor platform with on-board sensors and brushless motors, a ground computer and a remote control (Fig. 1 and 2). The mechanical platform is custom-built because the commercial versions available are either unreliable or too expensive. The outside perimeter is surrounded by carbon fiber rods to prevent damage to the rotors in case of collisions. The entire electronic system (e.g. circuit boards) and the software are developed and built by the authors in order to allow maximum flexibility when needed. The control algorithms are implemented in Simulink on the ground computer and run directly in the Matlab environment in normal or accelerated mode, allowing an instant transition between design and experiments (Fig. 3). A special S-function block was created to allow the Simulink model to receive sensor

* Supported by the ARO grant ARO W91NF-05-1-0314 and the Army National Automotive Center, and NSF grant ECCS-0801330.

data from the quadrotor and to send back the motor commands. The block communicates via USB with the base-station module connected to the computer. The sample rate is limited at 50 Hz by the speed of the wireless data link. All radio communication is done in short packets with minimum latency in such a way that during the sample period of 20 ms the sensors are sampled on the quadrotor, the data is sent over radio to the base-station and then through USB to the Simulink model, the results of the control algorithm are sent back to the base-station and from there through radio to the quadrotor to control the servomotors. The remote control uses the same two-way communication link to send commands to either the quadrotor or the computer (i.e. for reference inputs) and can also display important process variables to the pilot on the ground. Brushless DC motors with off the shelf speed controllers are used to improve the reliability and the repeatability of experiments. A remote-controlled safety switch was implemented in order to cut the power to the motors when the pilot is in the proximity of the unit and to prevent accidents due to faulty electronics, software of bad control algorithms.

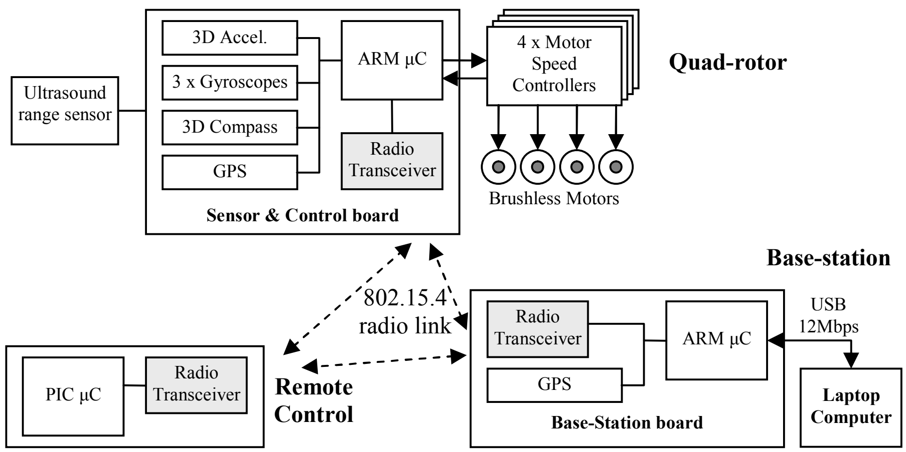

Fig. 1. The electronic system for the quadrotor helicopter

The variables that can be measured by the ARRI-built on-board sensors are the following:

- angular velocities in body coordinates
- linear accelerations in the body coordinates
- the intensity of the Earth magnetic field in body coordinates
- position and velocities in Earth coordinates (GPS),
- the altitude using the ultrasound range sensor,
- temperatures of the gyroscopes, accelerometers and magnetic field sensors for calibration,
- battery voltage and current,
- motor rotation speeds.

Instead of the joysticks on the remote control, a pilot can use a second quad-rotor board equipped with sensors to control the vehicle in a more intuitive manner. By tilting the board in different directions the pilot gives a vector velocity reference to the quadrotor. The compass is used on both the pilot board and the vehicle board such that independent of the yaw angle, the quadrotor follows the commands given by the pilot in the correct direction in the inertial reference frame. This greatly simplifies the work of the pilot and allows persons that have no previous training to control the vehicle in a natural way.

IV. QUADROTOR MODEL

Modeling quadrotors is not an easy task. A good model has to use theory usually applied for helicopters. Having four rotors in close proximity complicates the problem even further. There are interactions between the wakes produced by the rotors and the fuselage, and also between individual rotors. Because the propellers are made of plastic, they are quite flexible and present flapping at translational speeds. They can not be modeled precisely as propellers and require models similar to helicopter rotors. Except for hover, the expression for the rotor wash induced velocities can not be obtained in closed-form, creating difficulties when the model is used to design certain types of controllers.

The approach taken in this paper is to model only the most important elements of the quadrotor that define its behavior at hover and ignore the ones that have a significant effect only at high speeds.

The derivation of the nonlinear dynamics is performed in the North-East-Down (NED) inertial coordinates and in the x-y-z body-fixed coordinates (Fig. 2). Variables resolved to the inertial axes will be denoted by an e subscript and the ones resolved to the body axes will have the b subscript. The attitude is represented using quaternions. They parameterize the rotation from the inertial reference frame to the body frame using four values. The first is a scalar and the rest form a vector:

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\alpha/2) \\ \sin(\alpha/2)\cdot\mathbf{r} \end{bmatrix}, \tag{1}$$

where α is the angle of rotation and $\mathbf{r}$ is the axis around which the rotation is made, with the three components resolved to the inertial axes. Rotation quaternions have unitary norm and the vector $\mathbf{r}$ is a unit vector also. Addition of rotation quaternions does not generally result in a rotation quaternion and has no physical meaning. A special non-commutative multiplication operation is defined for the quaternions, denoted with the $\otimes$ operator. Multiplying two or more rotation quaternions produces another rotation quaternion that represents the total rotation obtained by performing each individual rotation for each quaternion in reverse order, starting with the last term of the product. Vectors can be rotated from one axis system to another if they are first transformed in quaternions with a scalar part equal to zero. For example, the same vector $\mathbf{r}$ can be resolved to the inertial frame as $\mathbf{r}_e$ and to the body frame as $\mathbf{r}_b$. The relationship between $\mathbf{r}_e$ and $\mathbf{r}_b$ is the following:

$$\begin{aligned} \begin{bmatrix} 0 \\ \mathbf{r}_b \end{bmatrix} &= \mathbf{q}^* \otimes \begin{bmatrix} 0 \\ \mathbf{r}_e \end{bmatrix} \otimes \mathbf{q} \\ \begin{bmatrix} 0 \\ \mathbf{r}_e \end{bmatrix} &= \mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{r}_b \end{bmatrix} \otimes \mathbf{q}^* \end{aligned} \tag{2}$$

where

$$\mathbf{q}^* = \mathbf{q}^{-1} = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} \tag{3}$$

is the conjugate of the rotation quaternion. More details about quaternions can be found in [7]. To simplify things, an abuse of notation will be used for rotations of vectors using quaternions. Instead of writing the augmented version of the vector as a quaternion with a zero scalar part, the normal vector will be used instead. It will be considered that the result is the rotated vector.

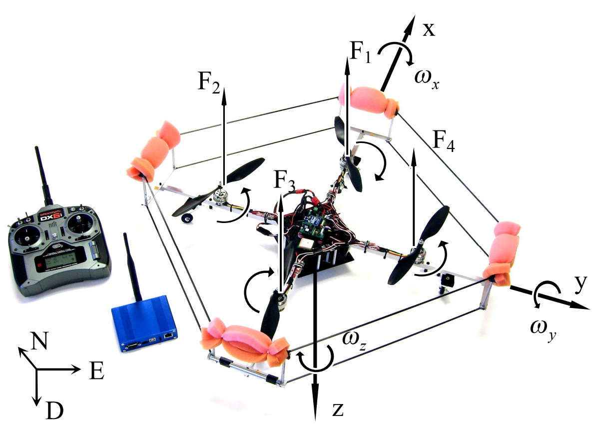

Fig. 2. Quadrotor platform. The exterior frame protects the propellers in case of collisions during experiments.

The kinematic and dynamic equations model the vehicle as a rigid body under the influence of the Earth gravity and the thrust forces produced by the rotors. Blade flapping, wake interaction and any other effects caused by the translational velocity are ignored.

$$\dot{\mathbf{V}}_b = \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} + \mathbf{q}^* \otimes \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \otimes \mathbf{q} - \boldsymbol{\omega}_b \times \mathbf{V}_b \tag{4}$$

$$\dot{\boldsymbol{\omega}}_b = \mathbf{I}_{nb}^{-1}\left(\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - \boldsymbol{\omega}_b \times \left(\mathbf{I}_{nb}\boldsymbol{\omega}_b\right)\right) \tag{5}$$

$$\dot{\mathbf{R}}_e = \mathbf{q} \otimes \mathbf{V}_b \otimes \mathbf{q}^* \tag{6}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \overline{\boldsymbol{\omega}}_b, \tag{7}$$

where

$$\overline{\boldsymbol{\omega}}_b = \begin{bmatrix} 0 \\ \boldsymbol{\omega}_b \end{bmatrix}. \tag{8}$$

The rotors are modeled as propellers in hover. The gyroscopic effect is ignored because the mass of all four propellers is only about 3% of the total mass of the vehicle. The thrust coefficient K_T and the torque coefficient K_Q were identified by measuring the forces and moments generated by the motors for given rotation speeds:

$$K_T = \frac{T}{\rho n^2 D^4} \tag{9}$$

$$K_Q = \frac{Q}{\rho n^2 D^5}. \tag{10}$$

The rotation velocities of the rotors are taken in their absolute value and are always positive. The direction of rotation is accounted for in the matrix in equation (13).

$$\Omega_i = 2\pi n_i \tag{11}$$

$$\tau\dot{\Omega}_i + \Omega_i = a_i f_{mot}\left(u_i, V_{batt}\right), \quad i = \overline{1,4} \tag{12}$$

Equation (12) models the behavior of the brushless DC motors with the propellers attached. From experiments it was observed that the transfer function of the motors together with the speed controllers is of first order. The time constant τ is 0.08 seconds, ensuring a high enough bandwidth for proper control. The input to the speed controllers is a servo-type PWM signal where the command u_i is sent as a variable-length rectangular pulse. The speed response of the motor is not linear and is approximated by the f_{mot} function that was identified experimentally. The battery voltage is measured online. To take into account the slightly different behavior of each motor, a gain a_i close to one was included also. This gain is estimated during flight for each motor and the suitable command is calculated in order to cancel its effect and to obtain an identical behavior for all motors.

The controllers used to stabilize the quadrotor generate a desired vertical force and desired moments about each axis in the body frame. The equation below maps the desired force and moments to the forces that have to be generated by each rotor. The controller will use the inverse of the matrix in equation (13) to obtain the rotor forces as a function of the commanded vertical force and moments.

$$\begin{bmatrix} F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -d & 0 & d \\ -d & 0 & d & 0 \\ c & -c & c & -c \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{13}$$

where

$$c = \frac{K_Q D}{K_T}. \tag{14}$$

The inputs into the quadrotor model are the desired forces $F_{di}, i = \overline{1,4}$. The actual platform receives a servo-type PWM signal u_i as an input. Equation (15) calculates the desired rotor speed and equation (16) the corresponding command u_i:

$$\Omega_{di} = \frac{2\pi}{D^2}\sqrt{\frac{-F_{di}}{K_T \rho}} \tag{15}$$

$$u_i = f_{mot}^{-1}\left(\frac{1}{a_i}\Omega_{di}, V_{batt}\right) \tag{16}$$

V. ATTITUDE ESTIMATION

The attitude of the quadrotor is estimated using the readings from the on-board inertial and magnetic sensors in body coordinates. The angular velocity $\boldsymbol{\omega}_b$ read using the gyroscopes is integrated to produce a high-bandwidth, low-noise estimate of the attitude quaternion. All sensors are temperature compensated and show no significant bias. Still, in the case of the gyroscopes, even small biases below the noise floor can cause significant drift if integrated. The accelerometer and the magnetometer readings are used together as inputs to the QUEST algorithm [8] to estimate the attitude quaternion also. In this case, because of vibrations and the body acceleration, the estimate is noisy. In the long term, it does not drift because it is based on absolute measurements. That's why it is used to do slow corrections to the estimate from the gyroscopes. This way, the final estimate is almost free of noise, has high bandwidth and presents no significant drift.

The error quaternion $\tilde{\mathbf{q}}$ is defined as the rotation needed to go from the estimated orientation quaternion $\hat{\mathbf{q}}$ to the quaternion obtained from the QUEST algorithm $\mathbf{q}_{QUEST}$:

$$\tilde{\mathbf{q}} = \mathbf{q}_{QUEST} \otimes \hat{\mathbf{q}}^*. \tag{17}$$

There are two values possible for $\tilde{\mathbf{q}}$. Only one will do the rotation with a minimum angle. It corresponds to the case when the first component of the quaternion, $\tilde{q}_0 = \cos(\alpha/2)$ is positive. If $\tilde{\mathbf{q}}$ is obtained with $\tilde{q}_0 < 0$ then $-\tilde{\mathbf{q}}$ will be used instead.

The error quaternion corrects the estimate by generating an extra angular velocity $\tilde{\boldsymbol{\omega}}$ in the body frame on top of the gyroscope measurements. In the next equation for the dynamics of the estimate of the attitude quaternion, $\tilde{\boldsymbol{\omega}}$ is augmented to a quaternion $\overline{\tilde{\omega}}$.

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}} \otimes \left(\overline{\boldsymbol{\omega}}_b + \overline{\tilde{\omega}}\right) \tag{18}$$

To find $\tilde{\boldsymbol{\omega}}$, $\tilde{\mathbf{q}}$ is written in the geometric form and its rotation axis is resolved to body coordinates:

$$\tilde{\mathbf{q}} = \begin{bmatrix} \cos(\alpha/2) \\ \sin(\alpha/2)\cdot \mathbf{r}_e \end{bmatrix} \text{ and } \tilde{\mathbf{q}}_b = \begin{bmatrix} \cos(\alpha/2) \\ \sin(\alpha/2)\cdot \mathbf{r}_b \end{bmatrix}, \tag{19}$$

with

$$\mathbf{r}_b = \mathbf{q}^* \otimes \mathbf{r}_e \otimes \mathbf{q}. \tag{20}$$

It is reasonable to make a small-angle assumption about the rotation angle α in the error quaternion. In this case equation (19) becomes

$$\tilde{\mathbf{q}} \approx \begin{bmatrix} 1 \\ \alpha/2 \cdot \mathbf{r}_e \end{bmatrix} \text{ and } \tilde{\mathbf{q}}_b \approx \begin{bmatrix} 1 \\ \alpha/2 \cdot \mathbf{r}_b \end{bmatrix}. \tag{21}$$

A proportional law is chosen to get $\tilde{\boldsymbol{\omega}}$:

$$\tilde{\boldsymbol{\omega}} = K_e \alpha \cdot \mathbf{r}_b, \tag{22}$$

where K_e is the time constant for the convergence of the estimate to the measured value. It is chosen to be small enough for noise rejection from measurements but large enough to allow for a good cancelation of the gyroscopes drift. Equation (18) becomes:

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}} \otimes \overline{\boldsymbol{\omega}}_b + \frac{1}{2}\hat{\mathbf{q}} \otimes \left(K_e \alpha \cdot \hat{\mathbf{q}}^* \otimes \mathbf{r}_e \otimes \hat{\mathbf{q}}\right) \tag{23}$$

and finally

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}} \otimes \overline{\boldsymbol{\omega}}_b + K_e \begin{bmatrix} 0 \\ \tilde{q}_1 \\ \tilde{q}_2 \\ \tilde{q}_3 \end{bmatrix} \otimes \hat{\mathbf{q}}. \tag{24}$$

Equation (24) is integrated using a method that maintains the unit norm of the rotation quaternion [9]. The integration over a sample period T is given by:

$$\hat{\mathbf{q}}(t+T) = \hat{\mathbf{q}}(t) \otimes e^{\int_t^{t+T} \mathbf{q}^*(\tau)\otimes\dot{\mathbf{q}}(\tau)d\tau}. \tag{25}$$

In order to be able to do the above integration in Simulink, a special S-function block was created. It overrides the built-in integration algorithms and performs the quaternion integration separately.

VI. ATTITUDE CONTROLLER

A. Motor Gain Estimation

The identical behavior of the motors is very important to keep the quadrotor stable in hover. The gains a_i from equation (12) are estimated on-line by comparing the measurements of Ω_i with their expected value $\hat{\Omega}_i$ obtained by integrating (12). The dynamics of the estimated gain is based on the estimation error:

$$\dot{a}_i = K_g\left(\Omega_{i\,meas} - \hat{\Omega}_i\right). \tag{26}$$

To avoid adaptation in unwanted operating points, the gain K_g is forced to zero at rotor speeds far from the nominal speed for hover.

B. Yaw and Tilt Errors

The quaternion formulation makes it easy to represent the orientation of a solid body relative to a reference axes system in a natural way by defining an axis of rotation $\mathbf{r}$ and an angle α for the amount of rotation. The attitude controller receives a desired orientation $\mathbf{q}_d$ and has to generate the right moment commands to rotate the vehicle from the current orientation $\mathbf{q}$ to $\mathbf{q}_d$. The error quaternion $\tilde{\mathbf{q}}$ is defined below:

$$\mathbf{q}_d = \tilde{\mathbf{q}} \otimes \mathbf{q} \tag{27}$$

$$\tilde{\mathbf{q}} = \mathbf{q}_d \otimes \mathbf{q}^* \tag{28}$$

The rotation axis of the error quaternion is resolved to the inertial (reference) coordinate system. In order to see what rotations are necessary in the body coordinates, the error quaternion will also be expressed in body coordinates as $\tilde{\mathbf{q}}_b$:

$$\tilde{\mathbf{q}} = \begin{bmatrix} \cos(\alpha/2) \\ \sin(\alpha/2)\cdot \mathbf{r}_e \end{bmatrix} \tag{29}$$

$$\mathbf{r}_b = \mathbf{q}^* \otimes \mathbf{r}_e \otimes \mathbf{q} \tag{30}$$

$$\tilde{\mathbf{q}}_b = \begin{bmatrix} \cos(\alpha/2) \\ \sin(\alpha/2)\cdot \mathbf{r}_b \end{bmatrix} \tag{31}$$

As opposed to the usual approach based on Euler angles where yaw is an angle in the horizontal plane and tilt is expressed using the pitch and roll angles, the attitude controller presented in this paper uses a different parameterization, relative to the body coordinate system. The yaw angle ψ defines a rotation around the z axis and the tilt is defined by two angles with a different meaning: a tilt direction γ_H in the $x-y$ plane and the tilt amount α_H. This parameterization is now linear and independent of the orientation of the vehicle relative to the inertial frame. The yaw and the tilt errors are defined by $\mathbf{q}_V$ and $\mathbf{q}_H$:

$$\tilde{\mathbf{q}}_b = \mathbf{q}_H \otimes \mathbf{q}_V \tag{32}$$

or in the geometric representation:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\alpha_H/2) \\ \sin(\alpha_H/2)\cdot r_x \\ \sin(\alpha_H/2)\cdot r_y \\ \sin(\alpha_H/2)\cdot 0 \end{bmatrix} \otimes \begin{bmatrix} \cos(\psi/2) \\ \sin(\psi/2)\cdot 0 \\ \sin(\psi/2)\cdot 0 \\ \sin(\psi/2)\cdot 1 \end{bmatrix}. \tag{33}$$

The amount of tilt is positive by convention:

$$\alpha_H = \arccos\left[1 - 2\left(q_1^2 + q_2^2\right)\right], \quad 0 \le \alpha_H < \pi \tag{34}$$

The function $\arctan_2(a,b)$ considers the sign of a and b, and places the angle in the correct quadrant.

$$\psi = 2\arctan_2(q_3, q_0), \quad -\pi \le \psi < \pi \tag{35}$$

The vehicle is tilted around the axis defined by r_x and r_y:

$$r_x = \frac{\cos(\psi/2)q_1 - \sin(\psi/2)q_2}{\sin(\alpha_H/2)}, \tag{36}$$

$$r_y = \frac{\sin(\psi/2)q_1 + \cos(\psi/2)q_2}{\sin(\alpha_H/2)}. \tag{37}$$

β_H is the direction of the tilt axis in the $x-y$ plane:

$$\beta_H = \arctan_2\left(r_y, r_x\right) \tag{38}$$

and γ_H is the direction of the tilt:

$$\gamma_H = \arctan_2\left(r_x, -r_y\right). \tag{39}$$

C. Attitude Stabilization

The attitude is stabilized by a yaw controller and by a tilt controller. There are no separate controllers for the x and for the y axes. The desired moments around each axis are

$$\begin{aligned} M_x &= M_{PH}\cos\beta_H + M_{Dx} \\ M_y &= M_{PH}\sin\beta_H + M_{Dy} \\ M_z &= M_{Pz} + M_{Dz} \end{aligned} \tag{40}$$

where

$$\begin{aligned} M_{PH} &= K_{PH}\alpha_H \\ M_{Pz} &= K_{Pz}\psi \end{aligned} \tag{41}$$

are the proportional controllers for tilt and yaw. The derivative action uses measurements directly from the gyroscopes:

$$\begin{aligned} M_{Dx} &= -K_{DH}\omega_x \\ M_{Dy} &= -K_{DH}\omega_y \\ M_{Dz} &= -K_{Dz}\omega_z \end{aligned} \tag{42}$$

The forces that go as commands to the motors are obtained by inverting equation (13) and the final PWM commands by using equations (15) and (16). Experimental data is shown in Fig. 4.

D. Altitude Control

An ultrasound range sensor is used to measure altitude. The force necessary to cancel the weight of the quadrotor is generated by a feed-forward path. The vertical component of the F_z force (along the D axis) is commanded to be equal to the weight plus some contribution from the PID controller for the altitude:

$$F_z = \frac{mg + F_{PIDz}}{K_{tilt}} \tag{43}$$

where

$$K_{tilt} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \left(q \otimes \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \otimes q^* \right) \tag{44}$$

E. *Pilot Control*

The first objective of the actual implementation of the control system for the quadrotor is to allow an untrained person to fly it using a remote control unit. The quadrotor is able to hover autonomously and to respond in an intuitive manner to the pilot commands. A second quadrotor electronic board was installed on the remote control. The pilot tilts the remote control in any direction and so generates a heading command. The amount of tilt is the velocity command. Using the same development as in the case of the quadrotor error quaternion $\tilde{\mathbf{q}}$, the remote control has a tilt amount α_{rem} and a tilt direction γ_{rem} relative to the NED coordinate system. Using these angles, $\mathbf{q}_d$ for the quadrotor is generated in the following way:

$$\mathbf{q}_d = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2) \cdot \mathbf{r}_{de} \end{bmatrix} \tag{45}$$

where

$$\mathbf{r}_{de} = \begin{bmatrix} \sin\gamma_{rem} \\ -\cos\gamma_{rem} \\ 0 \end{bmatrix} \quad \text{and} \quad \theta = \arcsin(K_s \alpha_{rem}). \tag{46}$$

The gain K_s determines the sensitivity of the pilot commands.

Independent of the yaw angle and independent of the orientation of the pilot, using this definition of the commanded quaternion ensures that the quadrotor is tilting in the same direction in which the pilot tilts the remote control.

VII. Conclusion

This paper presented the main components of a simple control system that was successful in stabilizing a custom-built quadrotor platform for hover. The most important issues were to estimate the attitude reliably and without significant noise using only the on-board sensors, to estimate the motor gains in order to ensure similar performance and to provide an intuitive control algorithm that can prove that the platform can be controlled satisfactory using simple PD controllers. The quaternion formulation allowed simple transformations from multiple coordinate systems and a natural representation of the controlled variables. The fact that the tilt was represented as a direction and the amount may prove useful for other types of controllers. An integral component could be added to compensate for the effects of the translational velocity. With this parameterization, the yaw motion would have little effect on the value of the integral term.

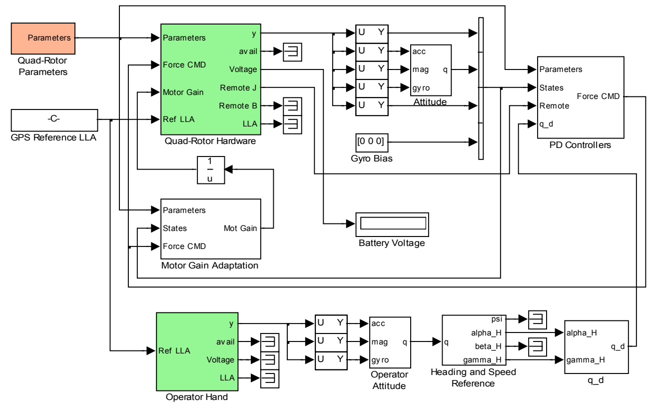

Fig. 3. Top-level Simulink implementation of the control algorithm. The control algorithm runs in real-time on a ground computer. Data is sent/received by radio to/from the quadrotor.

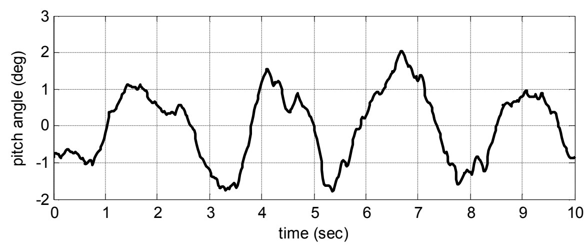

Fig. 4. Experimental data recorded for the pitch angle during hover. The PD controller is able to keep the attitude close to horizontal.

References

[1] P. Castillo, A. Dzul, R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," IEEE Transactions on Control Systems Technology, vol. 12, no. 4, pp. 510–516, July 2004.

[2] S. Park, D.H. Won, M.S. Kang, T.J. Kim, H.G. Lee, S.J. Kwon, "RIC (robust internal-loop compensator) based flight control of a quad-rotor type UAV", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3542–3547, Aug. 2005.

[3] M. Valenti, B. Bethke, G. Fiore, J.P. How and E. Feron, "Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery," AIAA Guidance, Navigation and Control Conference, Aug. 2006.

[4] N. Guenard, T. Hamel, V. Moreau, "Dynamic modeling and intuitive control strategy for an X4-flyer," International Conference on Control and Automation, vol. 1, pp. 141–146, June 2005.

[5] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.153–158, Oct. 2007.

[6] Gabriel M. Hoffmann, Haomiao Huang, Steven L. Wasl, Er Claire J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment", AIAA Guidance, Navigation, and Control Conference, 2007.

[7] J.C.K. Chou, "Quaternion kinematic and dynamic equations," IEEE Transactions on Robotics and Automation, 8(1), 53–64, 1992.

[8] M.D. Shuster, S.D. Oh, "Three-Axis Attitude Determination from Vector Observations," Journal of Guidance and Control, vol. 4, no. 1, pp. 70–77, January–February 1981.

[9] S. Gupta, "Linear quaternion equations with application to spacecraft attitude propagation", IEEE Aerospace Conference, vol. 1, pp. 69–76, March 1998.