

# Adaptive Quaternion Control of a Miniature Tailsitter UAV

Nathan B. Knoebel\*

*Scientific Systems Company, Inc.,  
500 West Cummings Park, Suite 3000,  
Woburn, MA 01801*

Timothy W. McLain†

*Brigham Young University,  
Provo, UT 84602*

**Abstract**—The miniature tailsitter is a unique aircraft with inherent advantages over typical unmanned aerial vehicles. With the capabilities of both hover and level flight, these small, portable systems can produce efficient maneuvers for enhanced surveillance and autonomy with little threat to surroundings and the system itself. Such vehicles create control challenges due to the two different flight regimes. These challenges are addressed with a computationally efficient adaptive quaternion control algorithm. A backstepping method for model cancellation and consistent tracking of reference model attitude dynamics is derived. This is used in conjunction with a regularized data-weighting recursive least-squares algorithm for the on-line identification of system parameters. Simulation and hardware results are presented as validation of the technique.

## I. INTRODUCTION

VTOL UAVs have inherent advantages due to their hover capabilities. Such vehicles can fly in confined areas and effectively takeoff and land in designated regions without a runway. These capabilities greatly enlarge the autonomy of the UAV, limiting the need for human interaction in recovery and deployment, and also allowing for perch-and-stare maneuvers, in-flight persistent target imaging, and navigation of obstacle filled terrain. All of these tasks are difficult, if not infeasible, for typical fix-wing UAVs to accomplish.

A largely unexplored UAV concept is the miniature tailsitter (Figure 1). Tailsitter UAVs are fix-wing VTOL aircraft, thus having all of the advantages of hover flight, as well as the benefits of efficient fix-wing flight, which is significant for miniature UAVs due to energy limitations. Moreover, the miniature tailsitter, because of size, is more portable and can navigate difficult terrain effectively and with little threat to immediate surroundings and the vehicle itself.

This UAV concept with its many advantages, poses control challenges. Conventionally the aeronautics community uses the 3 – 2 – 1 body-referenced Euler angles ( $\psi$ ,  $\theta$ , and  $\phi$ ) to relate the inertial and body frames of an aircraft. This method contains singularities at  $\theta = \pm\pi/2$ , where  $\theta = \pi/2$  is a common flight condition for the tailsitter. Another control challenge is the nonlinear dynamics of the tailsitter that are difficult to model. Typical autopilot attitude control techniques require that a good model of the aircraft's dynamics be available, all relevant states be measured or estimated, and the aircraft remains in the design flight conditions. Due to

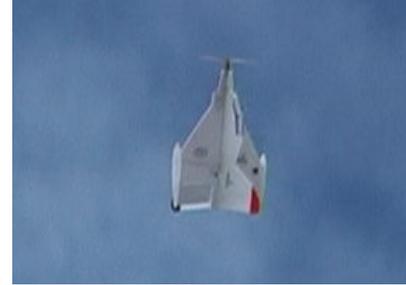


Fig. 1. The Brigham Young University tailsitter in hover flight.

the nature of miniature tailsitter UAVs, flight conditions can change drastically through typical maneuvers. Furthermore, aerodynamics in some flight regimes (particularly in stall conditions) are difficult to model and states such as aircraft angle of attack and sideslip angle are at present infeasible for miniature UAVs to measure or estimate. Also, small hover aircraft require lightweight hardware. This constraint extends to the autopilot processor. Currently, small autopilot hardware for miniature UAVs is available. However, computation on such small systems is limited.

Computationally efficient adaptive quaternion control is an obvious solution to these challenges. The quaternion attitude representation contains no singularities and attitude error can be represented conveniently in the aircraft body reference frame about principal  $x$ ,  $y$ , and  $z$  axes, which correspond to available actuator input torques. Also, through cancellation of system dynamics, adaptive methods can be used to produce stable consistent performance amidst significant system changes and despite modeling and sensing limitations.

The quaternion attitude control structure has been explored by many for various applications [3], [12], [22], [24]. For flight during aggressive maneuvers and in poorly modeled conditions, a variety of different adaptive control techniques have been proposed, including neural networks, least squares estimation, and Lyapunov based methods. The neural network approach is often employed to estimate nonlinear dynamics with off and on-line training for model inversion [7], [13], [17], [19], [21]. Recursive least squares techniques identify the airframe parameters on-line and use these parameters to adjust the controller. Such controllers have the ability to quickly converge on the airframe parameters [5], [18], [20], [21], and therefore, adapt rapidly to changes in flight

\*Research Engineer, email: nathan.knoebel@gmail.com

†Professor and Chair, Department of Mechanical Engineering, email: mclain@byu.edu

conditions. In Lyapunov based approaches, the parameter update law is selected to ensure stability of the tracking error. These methods generally cannot guarantee parameter convergence [6], [10], [11], [15], [16], [23]. The  $\mathcal{L}_1$  adaptive controller is a unique Lyapunov-based method that has also been used in flight control of UAVs [4]. The method utilizes filtering to enable fast adaptation and guarantee specific performance properties [8], [9].

The objective of this paper is to explore the application of a computationally efficient least-squares-based model reference adaptive quaternion backstepping control method for the control of a miniature tailsitter UAV. Section II contains a description of models used in the controller development, namely the aircraft attitude dynamic model, the controller reference model, and the open loop error model. The adaptive quaternion controller is presented in Section III and simulation and hardware results are shown in Section IV.

## II. DYNAMIC MODELS

### A. Aircraft Attitude Dynamics

In this section, attitude dynamics are modeled for controller derivation. The complex angular acceleration dynamics are simplified to facilitate on-line parameter estimation that is computationally easy, yet effective. Based upon knowledge about the system, these simplifications are chosen with design considerations in mind, to reduce computation and still capture the major contributions of angular acceleration given the information available to the autopilot.

For rigid-body aircraft, such as the tailsitter, the rotational acceleration dynamics are described as

$$\dot{\omega} = -J^{-1}\omega \times J\omega + J^{-1}M, \quad (1)$$

where  $\omega = (p, q, r)^T$  and  $M = (l, m, n)^T$  are the vectors of body frame angular velocities and moments and  $J$  is the constant positive-definite symmetric inertia matrix:

$$J = \begin{pmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{pmatrix}. \quad (2)$$

Assuming symmetry in the aircraft's  $xz$ -plane, equation (1) can be written explicitly as

$$\begin{aligned} \dot{p} &= \frac{J_{xz}(J_{xx} - J_{yy} + J_{zz})}{J_{xx}J_{zz} - J_{xz}^2}pq - \frac{J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2}{J_{xx}J_{zz} - J_{xz}^2}qr \\ &+ \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2}l(V, V_{pw}, \beta, p, \delta_a) \\ &+ \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2}n(V, V_{pw}, \beta, r, \delta_r), \end{aligned} \quad (3)$$

$$\dot{q} = \frac{J_{zz} - J_{xx}}{J_{yy}}pr - \frac{J_{xz}}{J_{yy}}(p^2 - r^2) + \frac{1}{J_{yy}}m(V, V_{pw}, \alpha, q, \delta_e), \quad (4)$$

$$\begin{aligned} \dot{r} &= \frac{J_{zz}(J_{zz} - J_{yy}) + J_{xz}^2}{J_{xx}J_{zz} - J_{xz}^2}pq - \frac{J_{xz}(J_{xx} - J_{yy} + J_{zz})}{J_{xx}J_{zz} - J_{xz}^2}qr \\ &+ \frac{J_{xz}}{J_{xx}J_{zz} - J_{xz}^2}l(V, V_{pw}, \beta, p, \delta_a) \\ &+ \frac{J_{zz}}{J_{xx}J_{zz} - J_{xz}^2}n(V, V_{pw}, \beta, r, \delta_r). \end{aligned} \quad (5)$$

It can be seen that the torques ( $l$ ,  $m$ , and  $n$ ) are a function of airspeed ( $V$ ), propeller wash airspeed ( $V_{pw}$ ), sideslip angle ( $\beta$ ), angle of attack ( $\alpha$ ), angular rates ( $p$ ,  $q$ , and  $r$ ), and actuator input deflections ( $\delta_a$ ,  $\delta_e$ , and  $\delta_r$ ).

For model simplification, angular acceleration about each body referenced axis is modeled with one bias acceleration term and one actuator-based input term:

$$\dot{p} = \theta_1 + \bar{V}^2\theta_2\delta_a, \quad \dot{q} = \theta_3 + \bar{V}^2\theta_4\delta_e, \quad \dot{r} = \theta_5 + \bar{V}^2\theta_6\delta_r. \quad (6)$$

The term  $\bar{V}$  is introduced to represent the dominant source of airflow over the control surfaces (aircraft velocity or propeller wash velocity). This simplified model reduces computation significantly for parameter estimation and control. These two types of contributions to angular acceleration are typically the most significant for the scale of aircraft described in this paper. From aerodynamic modeling [14], it can be seen that torques in level flight and propeller wash models are linear in one main actuator scaled by several constants (which denote actuator effectiveness) and either  $V^2$  or  $V_{pw}^2$ , as modeled above. All other contributions to angular acceleration are lumped into the bias terms. In view of design tradeoffs, only two terms were chosen to represent acceleration, under the assumption that the method will be able to track dynamics with the reduced model.

The angular acceleration model equation (6) can now be rewritten in vector form as

$$\dot{\omega} = C_1 + \bar{V}^2C_2U, \quad (7)$$

where  $C_1 = (\theta_1, \theta_3, \theta_5)^T$  and  $U = (\delta_a, \delta_e, \delta_r)^T$  are vectors of unknown bias parameters and actuator settings, and  $C_2$  is a diagonal matrix of unknown actuator effectiveness parameters:

$$C_2 = \begin{pmatrix} \theta_2 & 0 & 0 \\ 0 & \theta_4 & 0 \\ 0 & 0 & \theta_6 \end{pmatrix}.$$

From the quaternion time derivative, the attitude kinematics are

$$\dot{\eta} = \frac{1}{2}(\eta^\times + \eta_4I_3)\omega, \quad \dot{\eta}_4 = -\frac{1}{2}\eta^T\omega, \quad (8)$$

where the notation  $\xi^\times$  denotes the skew-symmetric matrix given by:

$$\xi^\times = \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix}.$$

The tailsitter rotational dynamics are therefore represented with equations (7) and (8).

### B. Reference Model Dynamics

Reference model dynamics are derived to smooth desired attitude trajectories. The reference model is chosen to have desired performance that is attainable by the actual system. Because a second-order model is used in the derivation of aircraft dynamics the reference model is also designed with second-order dynamics. These basic guidelines are pertinent to the Lyapunov-based adaptive controller.

The reference model quaternion ( $\bar{\eta}_m$ ) has the same kinematic relationship with the reference model angular velocities ( $\omega_m$ ) as given in equation (8), namely

$$\dot{\eta}_m = \frac{1}{2}(\eta_m^\times + \eta_{4m}I_3)\omega_m, \quad \dot{\eta}_{4m} = -\frac{1}{2}\eta_m^T\omega_m. \quad (9)$$

For second-order dynamics, the reference model angular accelerations are defined as

$$\dot{\omega}_m = -k_{2m}\omega_m + k_{1m}\eta_{em}, \quad (10)$$

where  $k_{1m}$  and  $k_{2m}$  are positive scalar gains which determine the damping and natural frequency of the reference model and  $\bar{\eta}_{em}$  is the error quaternion from the reference model to the commanded attitude.

### C. Error Dynamics

Open loop attitude error dynamics are derived to simplify the development of the backstepping controller. Quaternion attitude error ( $\bar{\eta}_e$ ) is defined as the rotation between the actual attitude and the controller reference model ( $\bar{\eta}_m = \bar{\eta}_e \otimes \bar{\eta}$ ). The tailsitter attitude error can then be written as

$$\eta_e = -\eta_{4m}\eta + \eta_4\eta_m - \eta^\times\eta_m, \quad \eta_{4e} = \eta_4\eta_{4m} + \eta^T\eta_m. \quad (11)$$

Given equations (8), (9), and (11) the attitude error dynamics can be defined as

$$\dot{\eta}_e = \frac{1}{2}(\eta_e^\times + \eta_{4e}I_3)(\tilde{R}\omega_m - \omega), \quad \dot{\eta}_{4e} = -\frac{1}{2}\eta_e^T(\tilde{R}\omega_m - \omega). \quad (12)$$

To rotate from the reference model reference frame to the actual body reference frame, the following rotation matrix ( $\tilde{R}$ ) is defined as

$$\tilde{R} = (\eta_{4e}^2 - \eta_e^T\eta_e)I_3 + 2\eta_e\eta_e^T + 2\eta_{4e}\eta_e^\times. \quad (13)$$

## III. CONTROLLER DERIVATION

### A. Backstepping Attitude Controller Derivation

Derivation of the adaptive quaternion controller is based upon Lyapunov theory. Because the system has a relative degree of two, the backstepping method is utilized to drive attitude error between the actual and reference model to zero by tracking desired angular rates with actuator inputs. As a result, the system is designed to asymptotically converge on desired angular rates and reference model attitude.

Let

$$\mathcal{V}' = \frac{1}{2}\eta_e^T\eta_e \quad (14)$$

be the initial candidate Lyapunov equation. Differentiating equation (14),

$$\dot{\mathcal{V}}' = \eta_e^T\frac{1}{2}(\eta_e^\times + \eta_{4e}I_3)(\tilde{R}\omega_m - \omega)$$

is obtained from equation (12). The derivation goal is for  $\dot{\mathcal{V}}' = -k_1\eta_e^T\eta_e$ , where  $k_1$  is a positive scalar gain. To achieve this, the desired angular velocity ( $\omega_d$ ) can be defined as

$$\omega_d = 2k_1(\eta_e^\times + \eta_{4e}I_3)^{-1}\eta_e + \tilde{R}\omega_m. \quad (15)$$

Thus,

$$\dot{\mathcal{V}}' = -k_1\eta_e^T\eta_e + \eta_e^T\frac{1}{2}(\eta_e^\times + \eta_{4e}I_3)\tilde{\omega}, \quad (16)$$

where  $\tilde{\omega}$  is the backstepping variable:

$$\tilde{\omega} = \omega_d - \omega. \quad (17)$$

Let

$$\mathcal{V} = \frac{1}{2}\eta_e^T\eta_e + \frac{1}{2}\tilde{\omega}^T\tilde{\omega} \quad (18)$$

be our final candidate Lyapunov equation. Equations (7) and (16) combined with equation (18) differentiated yields

$$\dot{\mathcal{V}} = -k_1\eta_e^T\eta_e + \eta_e^T\frac{1}{2}(\eta_e^\times + \eta_{4e}I_3)\tilde{\omega} + (\dot{\omega}_d - C_1 - \bar{V}^2C_2U)^T\tilde{\omega}. \quad (19)$$

As a result, the chosen control for model cancellation and asymptotic convergence is

$$U = \frac{1}{\bar{V}^2}C_2^{-1}\left(k_2\tilde{\omega} + \frac{1}{2}(-\eta_e^\times + \eta_{4e}I_3)\eta_e + \dot{\omega}_d\right) - \frac{1}{\bar{V}^2}C_2^{-1}C_1, \quad (20)$$

where  $k_2$  is a positive scalar gain. Thus,

$$\dot{\mathcal{V}} = -k_1\eta_e^T\eta_e - k_2\tilde{\omega}^T\tilde{\omega} \quad (21)$$

and with Lyapunov arguments it can be shown that  $\eta_e$  and  $\tilde{\omega} \rightarrow 0$  as  $t \rightarrow \infty$ .

### B. Stabilized Recursive Least Squares With Data Forgetting

The backstepping adaptive controller described in the previous section requires real-time accurate estimation of system parameters. System parameters can be chosen to minimize the error between estimated and observed or measured dynamics. Solving for the least-squares solution is a proven method, which lends itself well to a recursive algorithm. For adaptation, data forgetting, where recent data is weighted higher than old data, can be applied easily. Consequently, if system parameters change, weighting on new performance data allows the algorithm to learn the new parameters. Also, regularization, a stabilizing technique, can be employed to address instabilities introduced from noisy data combined with data forgetting.

Considering only the angular rate dynamics about the body reference frame  $x$ -axis where  $\dot{p} = (1 \quad \bar{V}^2\delta_a)(\theta_1 \quad \theta_2)^T = \Phi_1^T\Theta_1$ , the following convex function is chosen to be minimized:

$$J = \sum_{k=1}^N \frac{1}{2}(\Phi_1[k]^T\hat{\Theta}_1[N] - \dot{p}[k])^2\lambda^{N-k} + \sum_{k=1}^2 \frac{1}{2}\alpha_k(\hat{\theta}_k[N] - \hat{\theta}_k[N-1])^2. \quad (22)$$

The term  $\lambda$  weights the data according to the iteration difference  $N - k$ . A value between 0 and 1 is chosen for  $\lambda$ , where a value of one weights all of the past data the same; the closer the value gets to zero the less it weights old data. The terms  $\alpha_k$  penalize large changes in the estimated parameters from one time step to the next. This is a form of regularization that stabilizes the algorithm in the presence of noisy data when weighting is applied.

The gradient of  $J$  with respect to  $\hat{\Theta}_1$  is

$$\nabla J = \sum_{k=1}^N \Phi_1[k] (\Phi_1[k]^T \hat{\Theta}_1[N] - \dot{p}[k]) \lambda^{N-k} + \alpha (\hat{\Theta}_1[N] - \hat{\Theta}_1[N-1]), \quad (23)$$

where  $\alpha$  denotes the diagonal matrix

$$\alpha = \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{pmatrix}.$$

The least-squares solution is then found by setting  $\nabla J = 0$  and solving for  $\hat{\Theta}_1[N]$  as follows:

$$\hat{\Theta}_1[N] = \left( \sum_{k=1}^N \Phi_1[k] \Phi_1[k]^T \lambda^{N-k} + \alpha \right)^{-1} \sum_{k=1}^N \Phi_1[k] \dot{p}[k] \lambda^{N-k} + \alpha \hat{\Theta}_1[N-1]. \quad (24)$$

Because equation (22) is convex and the problem is unconstrained, equation (24) is a global minimum. Note that the inverse term is the covariance matrix

$$P[N] = \left( \sum_{k=1}^N \Phi_1[k] \Phi_1[k]^T \lambda^{N-k} + \alpha \right)^{-1}. \quad (25)$$

A simple recursive algorithm for the least-squares solution is

$$\hat{\Theta}_1[N] = \hat{\Theta}_1[N-1] + P[N] \Phi_1[N] (\dot{p} - \Phi_1[N]^T \hat{\Theta}_1[N-1]), \quad (26)$$

where

$$P^{-1}[N] = \lambda P^{-1}[N-1] + \Phi_1[N] \Phi_1[N]^T + \alpha(1 - \lambda). \quad (27)$$

This procedure can be easily reproduced for estimating angular acceleration parameters associated with the other two body-referenced axes. Note that this algorithm is computationally simple, resulting in little burden on the autopilot processor.

#### IV. RESULTS

For the testing of attitude control in both simulation and hardware, trajectories were generated from navigational controllers described in Ref. [14]; the actual quaternion attitude was estimated with the method also given in Ref. [14]. The simulation and hardware controllers flew the same desired flight path, which was chosen to represent all of the flight conditions a tailsitter might encounter. This path was comprised of a hover takeoff, followed by a hover waypoint, a transition to level flight, two level waypoints, a transition to hover, a hover waypoint, and finally a hover land. Figure 2 gives an example of the desired path just described.

##### A. Simulation Results

Simulation results of the adaptive attitude controller are presented in this section. The simulation environment, which is described in [14] incorporated the standard quaternion based translational and rotational kinematic and dynamic equations for nonlinear 6-DOF rigid-body aircraft simulation.

Quaternion attitude performance during the experiment is given in Figure 3. Note that the reference model is

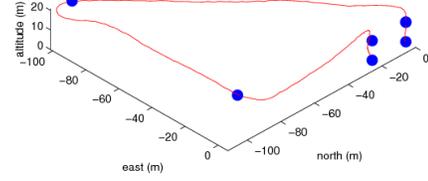


Fig. 2. The waypoint path selected for the attitude control experiment.

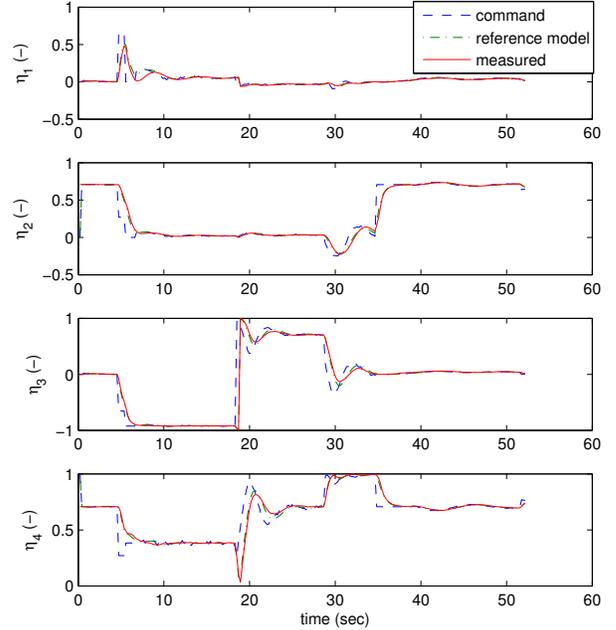


Fig. 3. Attitude results during simulation testing.

tracked consistently well throughout the entire flight, even in transition maneuvers, which occur at 5 and 35 seconds.

System identification performance can be seen in Figure 4, where  $\theta_1$ ,  $\theta_3$ , and  $\theta_5$  represent bias terms and  $\theta_2$ ,  $\theta_4$ , and  $\theta_6$  indicate the associated control surface effectiveness. It can be seen that despite drastic changes, the true parameters are identified accurately and with little delay. Consider for example, the aileron effectiveness term ( $\theta_2$ ) which changes from 1.8 to 0.2 instantly during a transition from level to hover flight at 35 seconds. The parameter is accurately identified within a tenth of a second. The level-flight roll moment due to poor aileron trim, can be seen in  $\theta_1$ , where it changes from roughly 0 to 5 in the transition to level flight. This is also identified and cancelled. In addition, note the pitch and yaw bias terms during level flight ( $\theta_3$  and  $\theta_5$ ). Large transients are accurately tracked, which occur during rotations about the associated axes due to aerodynamic damping and torque from sideslip and angle of attack.

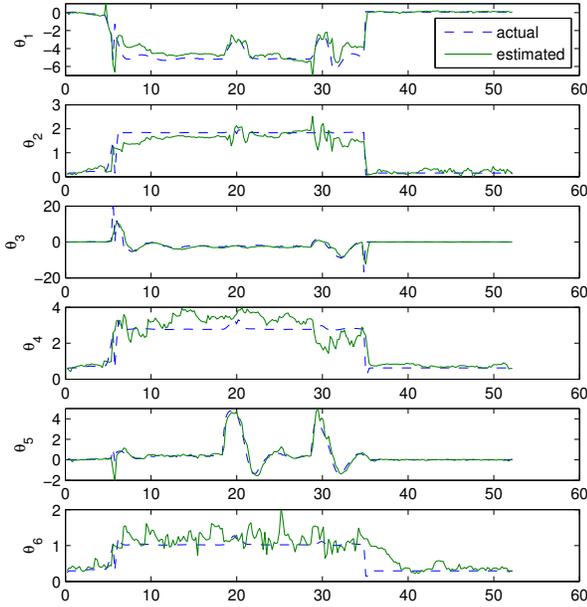


Fig. 4. Parameter estimation during the simulation test.

### B. Hardware Results

For hardware implementation a miniature tailsitter was produced. The airframe design chosen was the commercially available RC tailsitter model kit known as the Pogo [1] with a  $23\frac{1}{4}$  inch wing span, where corrugated plastic was substituted for Depron foam (see Figure 1). Lightweight construction, a powerful propulsion system, and large control surfaces in the propeller wash region allowed the miniature UAV to takeoff and land vertically, hover, and fly level, meeting the needs of a miniature tailsitter hardware testbed. The UAV was equipped with the Kestrel Autopilot [2] running a 29 MHz Rabbit microcontroller with 512K Flash and 512K RAM. The sensors on the autopilot include rate gyroscopes, accelerometers, magnetometers, an absolute pressure sensor for measuring altitude, a differential pressure sensor for measuring airspeed, and a GPS receiver.

Attitude results acquired during the hardware test flight are shown in Figure 5, where transitions occur at about 12 and 52 seconds. The controller demonstrated consistent attitude tracking throughout the maneuvers of tailsitter flight. This indicates accurate and fast adaptation as dynamics change significantly during transitions between hover and level flight. The yaw angular rate tracking throughout the flight is available in Figure 6. Note that it is through the tracking of desired angular rates that attitude is controlled with the backstepping method.

On-line parameter estimation during the flight is shown in Figure 7. Results appear to be similar to those presented from the simulation of the estimation algorithm. The first thing to note is that adaptation, like in simulation, occurs quite fast. Consider for example  $\hat{\theta}_2$ ,  $\hat{\theta}_4$ , and  $\hat{\theta}_6$ , it can be seen

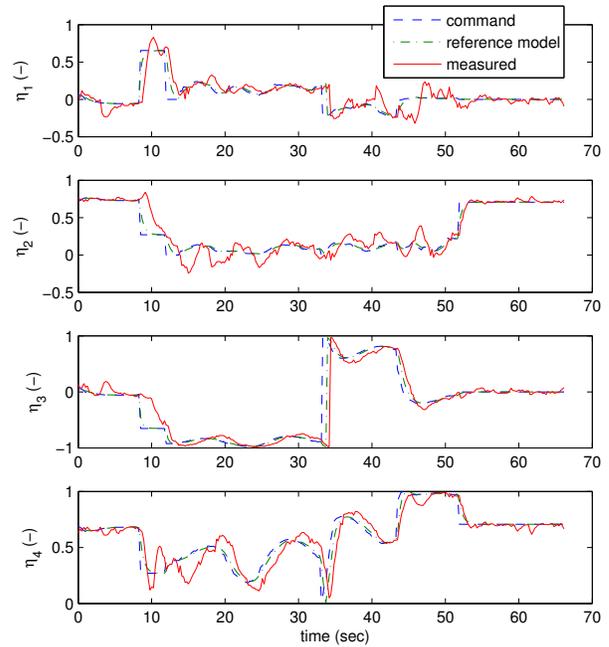


Fig. 5. Attitude results during the hardware flight test.

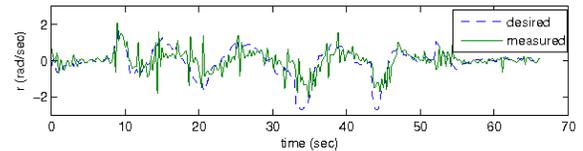


Fig. 6. Angular rate  $r$  during the hardware flight test.

that the estimated control surface effectiveness terms change instantly during transitions. Observe also that as expected, the actuator effectiveness terms are all relatively small in hover when compared to level flight. Level-flight bias angular acceleration due to poor aileron trim, which is identified quickly after the transition, can be seen as the nominal value of  $\hat{\theta}_1$  changes from roughly 0 to 15. This term reduces back to roughly zero after the transition to hover, indicating that aileron trim is poor in level flight only. Accelerations from sideslip and angular yaw rate damping are shown in the estimation of  $\theta_5$ . It can be seen that as expected angular acceleration opposing rotation is present in yaw rotations ( $\theta_5$  is shown to be the opposite sign of  $r$  in Figure 6).

Angular pitch acceleration tracking is presented in Figure 8. The plot of measured and estimated acceleration shows excellent modeling during both hover and level flight. Observe that the transition occurs at about 12 seconds. Pitch acceleration modeling error is  $3.7 \text{ rad/sec}^2$  on average throughout the flight.

### V. CONCLUSION

In this paper a novel backstepping method for consistent stable quaternion attitude control was derived enabling

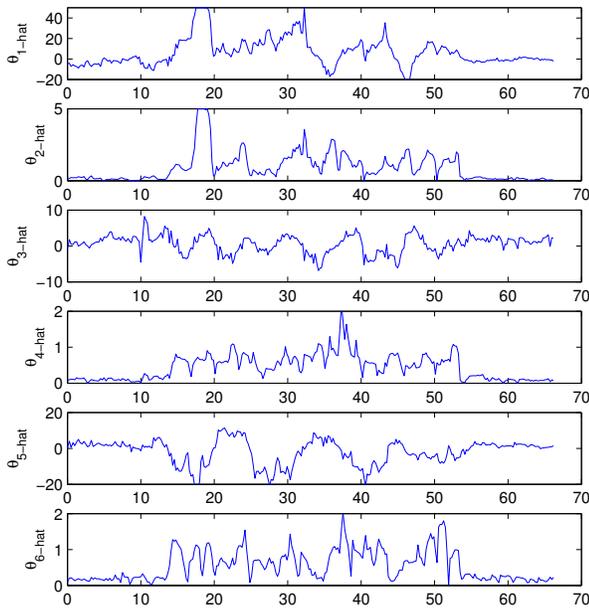


Fig. 7. Parameter estimation throughout the hardware test.

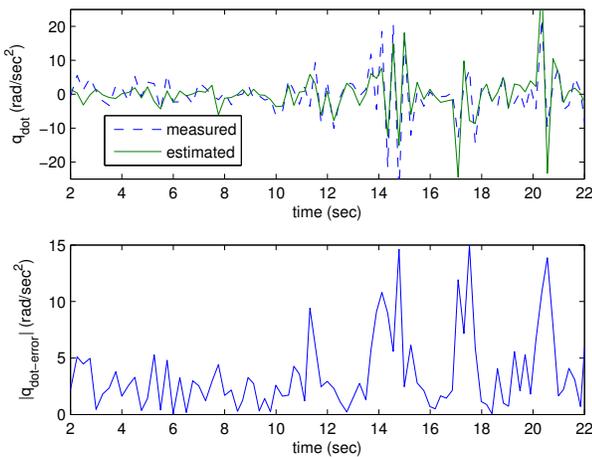


Fig. 8. Pitch angular acceleration and error throughout the hardware test.

miniature tailsitter flight. The controller with a regularized data-weighted recursive least-squares parameter estimation algorithm exhibited effective learning of rapidly changing dynamics for model cancellation and stable tracking of reference model attitude. This was demonstrated in both simulation and hardware results verifying the method.

#### ACKNOWLEDGMENTS

This research was funded by AFRL/MN award no. FA8651-05-1-0006.

#### REFERENCES

[1] <http://www.hobby-lobby.com/pogo.htm>.

[2] <http://procerusuav.com/>.

[3] R. Bach and R. Paielli. Linearization of attitude-control error dynamics. *IEEE Transactions on Automatic Control*, 38(10), Oct. 1993.

[4] R. Beard, N. Knoebel, C. Cao, N. Hovakimyan, and J. Matthews. An  $\mathcal{L}_1$  adaptive pitch controller for miniature air vehicles. In *AIAA Guidance, Navigation, and Control Conference*, pages 5788–5794, 21–24 August 2006.

[5] M. Bodson and J. Groszkiewicz. Multivariable adaptive algorithm for reconfigurable flight control. *IEEE Transactions on Control Systems Technology*, 5(2), Mar. 1997.

[6] J. D. Boskovic, L. Chen, and R. K. Mehra. Adaptive control design for nonaffine models arising in flight control. *AIAA Journal of Guidance, Control, and Dynamics*, 27(2):209–217, March–April 2004.

[7] R. L. Broderick. Statistical and adaptive approach for verification of a neural-based flight control system. In *Digital Avionics Systems Conference*, volume 2, pages 6.E.1–1 – 6.E.1–10, 24–28 Oct. 2004.

[8] C. Cao and N. Hovakimyan. Design and analysis of novel  $\mathcal{L}_1$  adaptive control architecture with guaranteed transient performance, part 1: Control signal and asymptotic stability. In *Proceedings of the 2004 American Control Conference*, 2006.

[9] C. Cao and N. Hovakimyan. Design and analysis of novel  $\mathcal{L}_1$  adaptive control architecture with guaranteed transient performance, part 2: Guaranteed transient response. In *Proceedings of the 2004 American Control Conference*, 2006.

[10] S. Chen, G. Tao, and S. Joshi. An adaptive actuator failure compensation controller for mimo systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 4778–4783, December 2002.

[11] J. Farrell, M. Polycarpou, and M. Sharma. Adaptive backstepping with magnitude, rate, and bandwidth constraints: Aircraft longitude control. In *Proceedings of the American Control Conference*, pages 3898–3904, June 2003.

[12] S. M. Joshi, A. G. Kelkar, and J. T. Wen. Robust attitude stabilization of spacecraft using nonlinear quaternion feedback. *IEEE Transactions on Automatic Control*, 40(10), 1995.

[13] B. Kim and A. Calise. Nonlinear flight control using neural networks. *IEEE Transactions on Control Systems Technology*, 20(1), Jan.-Feb. 1997.

[14] N. Knoebel. Adaptive quaternion control of a miniature tailsitter uav. Master’s thesis, Brigham Young University, 2007.

[15] N. Knoebel, S. Osborne, J. Matthews, and R. Beard. Computationally simple model reference adaptive control for miniature air vehicles. In *American Control Conference*, June 2006.

[16] J. Matthews, N. Knoebel, S. Osborne, and R. Beard. Adaptive backstepping control for miniature air vehicles. In *American Control Conference*, June 2006.

[17] P. Melin and O. Castillo. A new neuro-fuzzy-fractal approach for adaptive model-based control of non-linear dynamic systems: The case of controlling aircraft dynamics. In *1999 IEEE International Fuzzy Systems Conference Proceedings*, August 1999.

[18] B. Porter and C. Boddy. Design of adaptive digital controllers incorporating dynamic pole-assignment compensators for high-performance aircraft. In *Proceedings of the IEEE 1989 Aerospace and Electronics Conference*, pages 372–379, 22–26 May 1989.

[19] D. Shin and Y. Kim. Reconfigurable flight control system design using adaptive neural networks. *12*, (1), Jan. 2004.

[20] D. Shore and M. Bodson. Flight testing of a reconfigurable control system on an unmanned aircraft. In *Proceedings of the 2004 American Control Conference*, pages 3747–3752, June–July 2004.

[21] M. Steinberg and A. Page. High-fidelity simulation testing of intelligent and adaptive aircraft control laws. In *Proceedings of the American Control Conference*, pages 3264–3268, 8–10 May 2002.

[22] R. Stone. Control architecture for a tail-sitter unmanned air vehicle. In *5th Asian Control Conference*, pages 736–744, 2004.

[23] G. Tao, S. Chen, J. Fei, and S. Joshi. An adaptive actuator failure compensation scheme for controlling a morphing aircraft model. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 4926–4931, Dec. 2003.

[24] B. Wei, H. Weiss, and A. Arapostathis. Quaternion feedback regulator for spacecraft eigenaxis rotations. *AIAA Journal of Guidance and Control*, 12(3), 1989.