ROBUST NEURAL NETWORK CONTROL OF A QUADROTOR HELICOPTER

C. Nicol[1]*,C.J.B. Macnab*[1]*, A. Ramirez-Serrano*[2]

Schulich School of Engineering, University of Calgary
[1] Department of Electrical and Computer Engineering
[2] Department of Mechanical and Manufacturing Engineering
2500 University Dr. NW, Calgary, Alberta, Canada, T2N 1N2

ABSTRACT

This paper proposes a new adaptive neural network control to stabilize a quadrotor helicopter against modeling error and considerable wind disturbance. The new method is compared to both deadzone and e-modification adaptive techniques and through simulation demonstrates a clear improvement in terms of achieving a desired attitude and reducing weight drift.

Index Terms— Quadrotor Helicopter, Neural Network Control, Direct Adaptive Control, Cerebellar Model Articulation Controller (CMAC), Unmanned Aerial Vehicle

1. INTRODUCTION

Small electrically-actuated helicopters with four rotors, or simply *quadrotors*, have interested the robotics and control community in recent years. Many have proposed quadrotor helicopter attitude stabilization techniques including linear (PD) controls [1],[2] and nonlinear (backstepping and sliding-mode) controls [3], [4]. Some have even investigated advanced visual-feedback techniques [5],[6].

One major problem rarely addressed by researchers to date is that of a wind disturbance. A constant wind velocity [3] and very small wind gusts [4] are all that appear in the literature. Despite the lack of research, the effects of a significant wind disturbance should be considered a major concern because typical applications call for autonomous small lightweight vehicles. A buffeting (sinusoidal) wind disturbance could be considered as the worst case scenario for such a vehicle, throwing off angular and Cartesian estimations and making control very difficult.

The work presented in this paper represents a new method for stabilizing a quadrotor helicopter against unknown parameters in the system model as well as significant wind disturbances. The method is based on the use of a CMAC neural network in which neural-adaptive control techniques are used to train the approximation weights online. As sinusoidal disturbances tend to produce weight drift (which can eventually cause bursting and other undesirable effects), a secondary CMAC is incorporated which attempts to force the approximation weights towards smaller acceptable weights. Simulations establish the stability and performance of the new method in a situation where other robust adaptive control methods, e-modification and deadzone, both fail to produce a practical result.

978-1-4244-1643-1/08/$25 © 2008 IEEE

2. QUADROTOR MODEL

A quadrotor helicopter consists of four perpendicular arms, each with its own motor and rotor (Fig. 1). The four rotors provide upwards propulsion as well direction control. The system consist of two opposite rotor pairs (1,3) and (2,4). To balance torque, one pair rotates clockwise while the other rotates counter clockwise (Fig. 2 - note Ω_i, $i = 1...4$ are rotor speeds). A difference in speeds between the two pairs creates either positive or negative yaw acceleration. Increasing rotor 1 and decreasing rotor 2 speed produces positive pitch. Similarly, rotors 2 and 4 change roll.

Like any other aerial vehicle, the quadrotor helicopter has six degrees of freedom, Cartesian (X,Y,Z) and angular (ϕ,θ,ψ). As a result the model should consider both the angular and translational dynamics.

2.1. System Dynamics

Using the Newton-Euler method, the system angular dynamics can be expressed as [7]

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega} + J_r\Omega_r\mathbf{e_3}), \tag{1}$$

where:

- $\boldsymbol{\tau}$ a column vector of torques along the three axis,
- $\mathbf{J}$ is the rigid body inertia tensor,
- $\boldsymbol{\omega}$ is a column vector of angular velocities,
- J_r is the rotor inertia,
- $\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$ is total rotor speed,
- $\mathbf{e_3} = [0\ 0\ 1]^T$ describes rotor orientation.

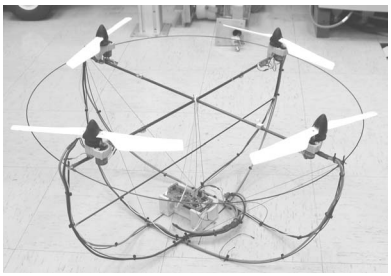

Fig. 1. Quadrotor

The translational dynamics in a fixed frame can be written as

$$m\dot{\mathbf{v}} = \mathbf{RT} - mg\mathbf{e_3} \tag{2}$$

where m is the mass of the quadrotor, $\mathbf{v}$ is a column vector of Cartesian velocities, $\mathbf{R}$ is a desired rotation matrix, $\mathbf{T}$ is a column vector of forces ($\mathbf{T} = [0\ 0\ b(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2)]^T$) and g is gravitational acceleration.

2.2. Equations of Motion

Rather than use rotor speeds as control inputs, it is simpler to define control signals:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{3}$$

The above matrix is invertible so commanded speeds follow directly from control signals. Using the dynamics developed in (1) and (2) along with input definitions (3), we arrive at the angular and Cartesian equations of motion

$$\ddot{\phi} = \dot{\theta}\dot{\psi}(I_Y - I_Z)/I_X - (J_r/I_X)\dot{\theta}\Omega_r + (lb/I_X)u_2 \tag{4}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}(I_Z - I_X)/I_Y + (J_r/I_Y)\dot{\phi}\Omega_r + (lb/I_Y)u_3 \tag{5}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}(I_X - I_Y)/I_Z + (d/I_Z)u_4 \tag{6}$$

$$\ddot{X} = -(\sin\theta\cos\phi)(b/m)u_1 \tag{7}$$

$$\ddot{Y} = (\sin\phi)(b/m)u_1 \tag{8}$$

$$\ddot{Z} = -g + (\cos\theta\cos\phi)(b/m)u_1 \tag{9}$$

where l is the length of an arm, b is a thrust factor and d is a drag factor. For details on deriving (4)-(6) see, for example, [4] or [8]. The Cartesian equations (7)-(9) use the rotation matrix $\mathcal{R}_{ZXY}$ rather than the more common $\mathcal{R}_{ZYX}$. This makes providing the desired roll and pitch (and their velocities) to achieve desired Cartesian coordinates possible without linearization. With this rotation matrix, the commanded yaw should always be zero and roll should be small to ensure Euler angles are close to roll, pitch and yaw provided by gyroscopes. Thus, the X axis faces into the flight direction, or ideally into the wind when hovering. The constants used for system development and simulations were obtained from [9] (Table 1).

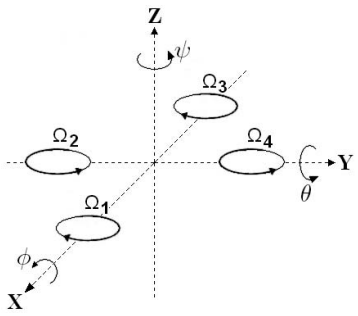

Fig. 2. Coordinate systems

3. NEURAL NETWORK IMPLEMENTATION

The choice to use a neural network for the main quadrotor control was based on the need to deal with large uncertainties in both parameter estimates and wind disturbances. The neural network chosen for the task was the Cerebellar Model Articulation Controller (CMAC) [10]. The CMAC consists of layers of offset look-up tables, divided into hypercube $cells$. Each input is a dimension in the CMAC structure. Each cell then has a basis function and weight associated with it. Similar to a radial basis function network, given inputs in vector $\mathbf{q} \in \mathcal{R}^n$ the output can be expressed as a weighted sum of basis functions:

$$\hat{f}(\mathbf{q}) = \sum_{i=1}^{m} \Gamma_i(\mathbf{q})w_i = \mathbf{\Gamma}(\mathbf{q})\mathbf{w} \tag{10}$$

where m is the number of layers, each Γ_i is the value of the basis function of the activated cell on the ith layer, and w_i is the associated weight. Thus $\mathbf{\Gamma}$ is a row vector and $\mathbf{w}$ is a column vector. The computer algorithm for calculating this output is more complex, using a hash-coding scheme to avoid allocating m arrays of dimension n [11].

4. SYSTEM DESIGN

The control strategy for the flying/hovering modes involved the development of two loops. The outer loop is responsible for providing a desired roll and pitch (and the velocities) to the inner control loop in an effort to achieve desired X and Y coordinates (or velocities). The (faster) inner loop is responsible for achieving the desired roll and pitch.

4.1. State Space Representation

In an effort to simplify stability proofs and demonstrate the system representation used in simulations, the system was reduced to the following state space form with state space variables given by

$$x_1 = Z \quad x_2 = \phi - \phi_d \quad x_3 = \theta - \theta_d \quad x_4 = \psi \tag{11}$$

Table 1. Constants

Parameter	Definition	Value
l	lever length	0.232 m
m	mass of quadrotor	0.52 Kg
d	drag coef.	7.5e-7 N m s^2
b	thrust coef.	3.13e-5 N s^2
I_X	X Inertia (I_{xx})	6.228e-3 Kg m^2
I_Y	Y Inertia (I_{yy})	6.225e-3 Kg m^2
I_Z	Z Inertia (I_{zz})	1.121e-2 Kg m^2
J_r	rotor inertia	6e-5 Kg m^2

Table 2. Parameters for Adaptive Techniques

Parameter	Value	Parameter	Value
β_1	0.4	$\beta_2, \beta_3, \beta_4$	0.2
$\epsilon_{\max}/G_1$	0.4	$\epsilon_{\max}/G_{2,3,4}$	0.01
κ_1	0.4	$\kappa_2, \kappa_3, \kappa_4$	0.2
ν	40	α	5
ρ	0.1	η	0.15
ζ_1	0.1	ζ_2	10
δ	0.0005		

and $x_{i+4} = \dot{x}_i$ for $i = 1 \ldots 4$. The system can be then be represented as

$$\dot{x}_i = x_{i+4} \qquad \text{for} \quad i = 1, 2, 3, 4$$
$$\begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} -d_1(t) - g + b_1 u_1 \\ -d_2(t) + x_7 x_8 a_1 + x_7 a_2 \Omega_r - \ddot{\phi}_d(t) + b_2 u_2 \\ -d_3(t) + x_6 x_8 a_3 + x_6 a_4 \Omega_r - \ddot{\theta}_d(t) + b_3 u_3 \\ -d_4(t) + x_6 x_7 a_5 + b_4 u_4 \end{bmatrix} \tag{12}$$

where

$$\begin{array}{ll} a_1 = (I_Y - I_Z)/I_X & b_1(t) = b\cos\theta\cos\phi/m \\ a_2 = J_r/I_X & b_2 = lb/I_X \\ a_3 = (I_Z - I_X)/I_Y & b_3 = lb/I_Y \\ a_4 = J_r/I_X & b_4 = d/I_Z \\ a_5 = (I_X - I_Y)/I_Z & d_i(t) = \text{wind disturbance} \end{array} \tag{13}$$

It is important to note that (12) is assuming the system is in hovering mode (i.e. $X_d = Y_d = Z_d = \psi_d = 0$).

4.2. Control strategy

For hovering, the outer loop controls designed to drive the system towards a desired Cartesian position are

$$\phi_d = \arcsin(K_1(Y_d - Y_0) - K_2\dot{Y}_0) \tag{14}$$
$$\theta_d = -\arcsin[(K_1(X_d - X_0) - K_2\dot{X}_0)/\cos\phi] \tag{15}$$

where K_1 and K_2 are simple control gains. The use of adaptive-neural techniques the for inner loop allows the controller to handle model uncertainty as well as possible disturbances. The four nonlinear equations depend on variables in vector $\mathbf{q}$ and it is assumed the CMACs uniformly approximate nonlinear terms:

$$\mathbf{f}(t, \mathbf{q}) = \hat{\mathbf{f}}(\mathbf{q}, \mathbf{w}) + \boldsymbol{\epsilon}(t, \mathbf{q}) = \boldsymbol{\Gamma}\mathbf{w} + \boldsymbol{\epsilon}(t, \mathbf{q}) \tag{16}$$

where $\boldsymbol{\Gamma}$ is now an $n \times m$ matrix of activated basis functions and

$$\boldsymbol{\epsilon}(t, \mathbf{q}) = \mathbf{d}(t) + \mathbf{d}_{\text{approx}}(\mathbf{q}) \tag{17}$$

represents all time-varying disturbances and the approximation error. Assuming the error is bounded in a local region of approximation ($\boldsymbol{\epsilon}(t, \mathbf{x}) < \epsilon_{\max} \forall \mathbf{x} \in \mathcal{R}^n$), a Lyapunov candidate based on sliding mode errors $z_i = x_i + x_{i+4}$ ($i = 1, 2, 3, 4$) is

$$V = \sum_{i=1}^{4} \frac{\hat{b}_i}{2b_i} z_i^2 + \frac{1}{2\beta_i} \tilde{\mathbf{w}}_i^T \tilde{\mathbf{w}}_i \tag{18}$$

In (18), the $\hat{b}_i$s are rough order-of-magnitude approximations of the constants chosen in (13), $\tilde{\mathbf{w}}_i$s are weight errors (i.e. the difference between the ideal and estimated weights $\tilde{\mathbf{w}}_i = \mathbf{w}_i - \hat{\mathbf{w}}_i$) and V is positive definite assuming $|\phi, \theta| < (\pi/2)$. Taking the time derivative of the Lyapunov function

$$\dot{V} = \sum_{i=1}^{4} z_i(\boldsymbol{\Gamma}_i(\mathbf{x})\mathbf{w}_i + \epsilon_i(t, \mathbf{x}) + \hat{b}_i u_i) - \frac{1}{\beta_i} \tilde{\mathbf{w}}_i^T \dot{\hat{\mathbf{w}}}_i \tag{19}$$

The control can then be chosen as

$$u_i = \hat{b}_i^{-1}(-\boldsymbol{\Gamma}_i(\mathbf{x})\hat{\mathbf{w}}_i - G_i z_i). \tag{20}$$

where the G_is are simple control gains. Two common techniques for designing a robust weight update law are deadzone and e-modification. The deadzone technique produces

$$\dot{\hat{\mathbf{w}}}_i = \begin{cases} \beta_i \boldsymbol{\Gamma}_i^T z_i & \text{if } |z_i| > \epsilon_{\max}/G_i, \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

The application of deadzone therefore ensures that the smallest level set $V_i(|z_i|, \|\tilde{\mathbf{w}}_\mathbf{i}\|)$ that encloses the initial condition will be a uniform ultimate bound on the trajectory. The e-modification technique results in

$$\dot{\hat{\mathbf{w}}}_i = \beta_i(\boldsymbol{\Gamma}_i^T z_i - \nu|z_i|\hat{\mathbf{w}}_\mathbf{i}) \tag{22}$$

where β_is are weight update gains and ν is a leakage gain. Using (22), it can be shown that $\dot{V}_i < 0$ if

$$|z_i| > \tfrac{e_{\max}}{G_i} + \tfrac{\nu\|\mathbf{w}_\mathbf{i}\|}{4G_i} \;\text{ or }\; \|\tilde{\mathbf{w}}_\mathbf{i}\| > \tfrac{\|\mathbf{w}_\mathbf{i}\|}{2} + \sqrt{\tfrac{e_{\max}^2}{4G_i\nu} + \tfrac{\|\mathbf{w}_\mathbf{i}\|}{4}}$$

The application of e-modification therefore ensures that the smallest level set $V_i(|z_i|, \|\tilde{\mathbf{w}}_\mathbf{i}\|)$ that encloses the area defined above (in ($|z_i|$,$\|\tilde{\mathbf{w}}_\mathbf{i}\|$) space) as well as the initial condition will be a uniform ultimate bound on the trajectory.

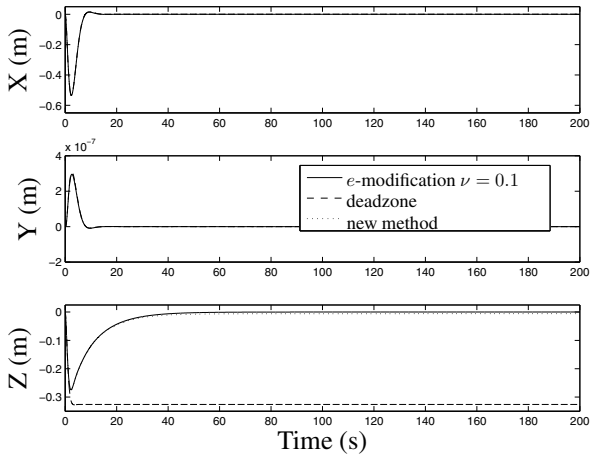

Fig. 3. All three techniques with no disturbance

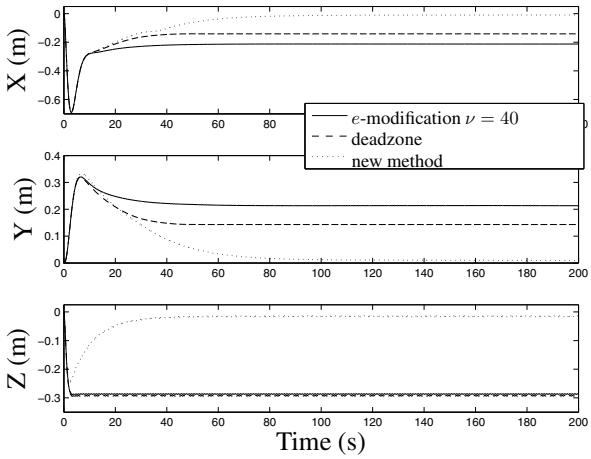

Fig. 4. All three techniques with sinusoidal disturbances

4.3. Inner Loop (Method of Alternative Weights)

The method of alternative weights was first proposed in [12] and investigated for use on a quadrotor helicopter in [13]. Since the training is done online (which really just means adjusting weights only to suite the particular trajectory), there exists any number of weights that could fulfill the task of approximating the nonlinear function.

Assuming that there is a set of alternate weights $\widehat{\mathbf{p}}$, then the error can be represented as

$$\sigma_i = (\mathbf{\Gamma}_i \widehat{\mathbf{w}}_i - \mathbf{\Gamma}_i \widehat{\mathbf{p}}_i). \tag{23}$$

With this in mind, the proposed alternate weights update rule that would bring the alternate output to within $\pm\delta$ of the approximation output is

$$\dot{\widehat{\mathbf{p}}}_i = \begin{cases} \kappa_i(\alpha \mathbf{\Gamma}_i^T \sigma_i + \rho(\bar{p}_i - \widehat{\mathbf{p}}_i) - \eta \widehat{\mathbf{p}}_i) & \text{if } |\sigma_i| > \delta \\ \kappa_i(\rho(\bar{p}_i - \widehat{\mathbf{p}}_i) - \eta \widehat{\mathbf{p}}_i) & \text{otherwise} \end{cases} \tag{24}$$

where κ_is, α, ρ and η are positive gains used for adjusting the alternative weight update law and $\bar{p}_i$=mean($\widehat{\mathbf{p}}_i$) The middle term $\rho(\bar{p}_i - \widehat{\mathbf{p}}_i)$ is used to not only reduce the size of the alternate weights, but guarantee that they are different from the approximation weights.

The key to creating a stable approximation weight update rule requires adaptation based on state error, while using knowledge of the alternate weights to reduce weight size. The proposed approximation weight update rule is

$$\dot{\widehat{\mathbf{w}}}_i = \begin{cases} \beta_i(\mathbf{\Gamma}_i^T z_i - \alpha \mathbf{\Gamma}_i^T \sigma_i + \zeta_1(\widehat{\mathbf{p}}_i - \widehat{\mathbf{w}}_i)) & \text{if } |\sigma_i| > \delta \\ \beta_i(\mathbf{\Gamma}_i^T z_i + \zeta_2(\widehat{\mathbf{p}}_i - \widehat{\mathbf{w}}_i)) & \text{otherwise.} \end{cases} \tag{25}$$

where β_is, ζ_1 and ζ_2 are positive gains used for adjusting the weight update law and $\zeta_1 \ll \zeta_2$ (these last terms are thus used to force the approximation weights towards the alternative weights once the two outputs are similar enough). Writing $V = \sum_{i=1}^{4} V_i$ allows us to analyze just

$$V_i = \frac{\hat{b}_i}{2b_i} z_i^2 + \frac{1}{2\beta_i} \tilde{\mathbf{w}}_i^T \tilde{\mathbf{w}}_i + \frac{1}{2\kappa_i} \tilde{\mathbf{p}}_i^T \tilde{\mathbf{p}}_i. \tag{26}$$

Taking the time derivative we arrive at

$$\dot{V}_i = z_i \epsilon_i - z_i^2 G_i + \tilde{\mathbf{w}}_i^T (\mathbf{\Gamma}_i^T z_i - \frac{\dot{\hat{\mathbf{w}}}_i}{\beta}) + \tilde{\mathbf{p}}_i^T (-\frac{\dot{\hat{\mathbf{p}}}_\mathbf{i}}{\kappa}). \tag{27}$$

When $\|\boldsymbol{\sigma}\| > \delta$, then

$$\dot{V}_i \leq \begin{bmatrix} |z_i| \\ \|\tilde{\mathbf{w}}_i\| \\ \|\tilde{\mathbf{p}}_i\| \end{bmatrix}^T \begin{bmatrix} -G_i & 0 & 0 \\ 0 & -\zeta_1 & \zeta_1/2 \\ 0 & \zeta_1/2 & -\eta \end{bmatrix} \begin{bmatrix} |z_i| \\ \|\tilde{\mathbf{w}}_i\| \\ \|\tilde{\mathbf{p}}_i\| \end{bmatrix} \tag{28}$$

$$+ \begin{bmatrix} \epsilon_{\max} & 0 & (2\rho+\eta)\|\mathbf{w}_i\| \end{bmatrix} \begin{bmatrix} |z_i| & \|\tilde{\mathbf{w}}_i\| & \|\tilde{\mathbf{p}}_i\| \end{bmatrix}^T.$$

As long as the first term has eigenvalues with negative real parts (parameters are chosen such that $\eta > \zeta_1/4$) then $\dot{V} < 0$ outside a compact set in ($\|\mathbf{z}\|$, $\|\tilde{\mathbf{w}}\|$, $\|\tilde{\mathbf{p}}\|$) space and all signals are uniformly ultimately bounded. When $\|\boldsymbol{\sigma}\| < \delta$, then $\|\tilde{\mathbf{p}}\| < p_{\max}$ and

$$\dot{V}_i \leq \begin{bmatrix} |z_i| \\ \|\tilde{\mathbf{w}}_i\| \\ \|\tilde{\mathbf{p}}_i\| \end{bmatrix}^T \begin{bmatrix} -G_i & 0 & 0 \\ 0 & -\zeta_2 & 0 \\ 0 & 0 & -\eta \end{bmatrix} \begin{bmatrix} |z_i| \\ \|\tilde{\mathbf{w}}_i\| \\ \|\tilde{\mathbf{p}}_i\| \end{bmatrix},$$

$$+ \begin{bmatrix} \epsilon_{\max} & p_{\max} & 0 \end{bmatrix} \begin{bmatrix} |z_i| \|\tilde{\mathbf{w}}_i\| \|\tilde{\mathbf{p}}_i\| \end{bmatrix}^T + p_{\max}(2\rho+\eta)\|\mathbf{w}_i\|$$

where again $\dot{V} < 0$ outside a compact set in ($\|\mathbf{z}\|$, $\|\tilde{\mathbf{w}}\|$, $\|\tilde{\mathbf{p}}\|$) space and all signals are uniformly ultimately bounded.

5. RESULTS

In the simulations the pitch position starts at 0.1 radians and all other states are zero.

Using a common set of parameters (Table 2), the first simulations compare the three adaptive techniques in the case of no disturbances. Here all three methods have very similar transient responses, except for deadzone which has 0.32m error in Z (Fig. 3).

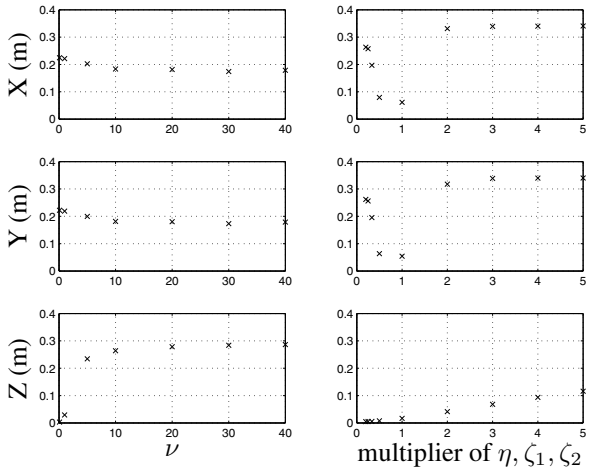

Fig. 5. Cartesian position error at $t = 200$ s when varying the robust parameters

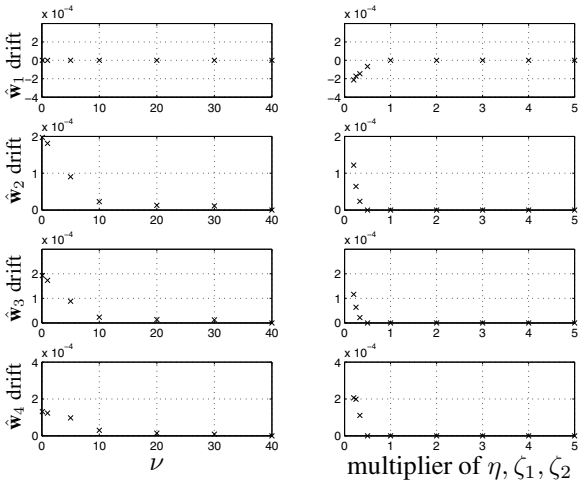

Fig. 6. Weight drift at $t = 200$ when varying the robust parameters

The second simulations included the addition of disturbances

$$d_1 = 0.01(10 + 5\sin(2\pi t) \tag{29}$$

$$d_2, d_3, d_4 = 0.1(10 + 5\sin(2\pi t) \tag{30}$$

For e-modification, ν needs to be at least 40 in order to avoid bursting in the first 200 seconds. No changes were required for either deadzone or the new method. The new method outperforms both e-modification and deadzone when the large oscillations are present, by a factor of more than 10 in all coordinates (Fig. 4).

Varying the robust parameter ν in e-modification and varying η, ζ_1, ζ_2 (by a constant multiplier) demonstrates that the new method has an optimum value for the parameters whereas no value of ν can achieve high performance (Fig. 5). The reason for this is apparent by looking at weight drift (measured as the average increase in the maximum weight magnitude per second in Fig. 6). In e-modification ν must be very large to halt weight drift entirely, but the new method can halt the weight drift easily, without sacrificing performance.

6. CONCLUSIONS

Although Lyapunov stability proofs gave uniform ultimate bounds for deadzone, e-modification and the new method, simulation results show that the new method is able to achieve far less error in terms of obtaining a desired attitude while also reducing weight drift when large oscillations are present. Further tests will be conducted with regards to other quadrotor flight modes and with such encouraging results, hopefully real prototype tests can be conducted in the near future.

7. REFERENCES

[1] S. McGilvray and A. Tayebi, "Attitude stabilization of a vtol quadrotor aircraft," in *IEEE Trans. Control Systems Technology*, 2006, vol. 14, pp. 562–571.

[2] E. Altug and B. Erginer, "Modeling and pd control of a quadrotor vtol vehicle," in *Proc. IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 2007, pp. 894–899.

[3] A. Benallegue and T. Madani, "Backstepping sliding mode control applied to a miniature quadrotor flying robot," in *Proc. IEEE Conf. on Industrial Electronics*, Paris, France, 2006, pp. 700–705.

[4] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, 2005, pp. 2247–2252.

[5] E. Altug, J.P. Ostrowski, and C.J. Taylor, "Quadrotor control using dual camera visual feedback," in *Proc. IEEE Conf. Robotics and Automation*, Taipei, Taiwan, 2003, vol. 3, pp. 4294–4299.

[6] S.G. Fowers, D. Lee, B.J. Tippetts, K.D. Lillywhite, A.W. Dennis, and J.K. Archibald, "Vision aided stabilization and the development of a quad-rotor micro uav," in *Proc. IEEE Int. Sym. Computational Intelligence in Robotics and Automation*, Jacksonville, Florida, 2007, pp. 143–148.

[7] P.C. Hughes, *Spacecraft Attitude Dynamics*, New York: Wiley, 1986.

[8] H. Voos, "Nonlinear state-dependent riccati equation control of a quadrotor uav," in *Proc. IEEE Int. Conf. Control Applications*, Munich, Germany, 2006, pp. 2547–2552.

[9] S. Bouabdallah and R. Siegwart, *Advances in Telerobotics*, chapter Towards Intelligent Miniature Flying Robots, pp. 429–440, 2006.

[10] J. Albus, "A new approach to manipulator control: the cerebellar model articulation controller (CMAC)," *J. Dyn. Sys. Meas. Contr.*, vol. 97, pp. 220–227, 1975.

[11] J. Albus, "Data storage in the cerebellar model articulation controller (CMAC)," *J. Dyn. Sys. Meas. Contr.*, vol. 97, pp. 228–233, 1975.

[12] C.J.B. Macnab, "Robust direct adaptive control with associative memories in the presence of persistent oscillations," *Neural Information Processing Letters*, vol. 10, no. 12, pp. 277–287, Dec. 2006.

[13] C. Coza and C.J.B. Macnab, "A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization," in *Proc. North American Fuzzy Information Processing Society Conference*, Montreal,Canada, 2006, pp. 454–458.

Intentional Blank Page