Mechatronics 21 (2011) 927–938

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Robust adaptive control of a quadrotor helicopter

C. Nicol [a], C.J.B. Macnab [a,*], A. Ramirez-Serrano [b]

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

ARTICLE INFO

Article history:
Received 30 November 2007
Accepted 18 February 2011
Available online 12 April 2011

Keywords:
Adaptive fuzzy control
Neural adaptive control
Quadrotor helicopter
Helicopter control
Lyapunov stability

ABSTRACT

This work presents a direct approximate-adaptive control, using CMAC nonlinear approximators, for an experimental prototype quadrotor helicopter. The method updates adaptive parameters, the CMAC weights, as to achieve both adaptation to unknown payloads and robustness to disturbances. Previously proposed weight-update methods, such as e-modification, provide robustness by simply limiting weight growth. In order to let the weights grow large enough to compensate unknown payloads, the proposed method relies on a set of alternate weights to guide the training. The alternate weights produce nearly the same output, but with values clustered closer to the average weight so that the output remains relatively smooth. This paper describes the design of a prototype helicopter suitable for testing the control method. In the experiment the new method stops weight drift during a shake test and adapts on-line to a significant added payload, whereas e-modification cannot do both.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Exhibiting open-loop unstable dynamics, the quadrotor poses challenges to present-day control engineers and several significant results appear in the literature (Table 1). Linear controls successfully stabilized the prototype X-4 Flyerhad in the presence of step-disturbances, while attached to a test platform [1]. Later the same research group tested a newer Mark II prototype without disturbances [2]. The STARMAC-II prototype achieved free-flight hovering using PID controls [3], where it was noted that wind disturbances cause the control to fail. Later, the same group accomplished path following outdoors [4]. Another prototype achieved autonomous flight in [5], where a linear control maintained stable hover providing robustness to small disturbances. Nonlinear controls can substantially expand the region of controllable flight angles compared to linear controls. A tethered Spectrolutions HMX-4 quadrotor used state inputs from a camera fed into a feedback-linearization control, without disturbances [6]. Nonlinear controls also achieved robustness to impulse disturbances, both in simulation [7,8] and using a test-stand experiment [9,10]. In [11,12], a nested-saturations controller stabilized a Draganfly III in the presence of impulse disturbances, and results were compared to linear feedback controls. A pre-trained neural-network stabilized a Draganfly II quadrotor in hover without disturbances [13]. Adaptive neural network controls successfully stabilized quadrotors in simulation [14,15]. Our proposed control strategy differs from those before as it aims to be both adaptive to model uncertainty (payloads) as well as robust to disturbances.

The nonlinear function approximator known as the Cerebellar Model Arithmetic Computer (CMAC) [17] provides fast learning or adaptation, similar to Radial-Basis-Function networks. The CMAC contains local basis functions with hypercube domains, or cells, and dealing with only the activated cells provides on-line computational efficiency. The CMAC is an associative memory equivalent to a neural network, although with normalized basis functions it also becomes a fuzzy approximator. Fast-adaptation and computational efficiency appear to make the CMAC ideal for use in direct adaptive control for the quadrotor. However, when sinusoidal disturbances cause inputs to oscillate between adjacent cells near the origin, the corresponding adjacent weights tend to drift to large positive and negative values causing control oscillations or chatter. An oscillating control signal may lead to sudden increases in state error, called *bursting*. Chattering can excite unmodeled dynamics causing instability [18]. Techniques traditionally used in direct adaptive control for robust weight updates restrict performance to stop the weight drift. These traditional techniques include parameter projection [19], deadzone [19], and *e*-modification [20]. The *e*-modification method appears attractive because it requires neither *a-priori* training on the model (like projection) nor knowledge of the bounds on disturbances (like deadzone) in order to guarantee uniformly ultimately bounded (UUB) signals. Because restricting weight magnitudes negatively affects performance in payload adaptive schemes, we seek an alternative method.

The proposed method for robust weight updates prevents adjacent weights from drifting far in opposite directions. In the new

* Corresponding author. Tel.: +1 403 277 9928; fax: +1 403 282 6855.
E-mail address: cmacnab@ucalgary.ca (C.J.B. Macnab).

0957-4158/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2011.02.007

Table 1
Summary of quadrotor literature review.

Disturbance type	Linear control	Nonlinear control			
		Backstepping	Sliding mode	Neural/fuzzy	
				Non-adaptive	Adaptive
None	[2][a]	[6,8]		[13][a]	[15]
Impulse	[5,3][a]	[9–12][a]	[7]		
Step	[1]				
Time varying	[4][a]				[16,14]

[a] Indicates prototype experimentation (semi-manual/autonomous flight or using test stand).

method, a set of alternate weights guides the adaptation process in order to prevent drift. The design of the alternate weight update ensures they stay close to a mean value while still providing approximately the same output as the weights used in the control. Thus, the alternate weights can grow large enough to compensate payloads, yet are prevented from drifting to large positive and negative values in adjacent cells. A leakage term in the control CMAC weight update keeps the weights close to the alternate values. A Lyapunov stability proof ensures the uniform ultimate boundedness of all signals. The control in this paper follows from a previously proposed adaptive-fuzzy adaptive control scheme tested in a Cartesian-space simulation [16]. This work substantially modifies the previous method, achieving robustness to disturbances, and tests the method experimentally with a quadrotor mounted on a test-bed moving in roll, pitch, and yaw.

2. Background

In the early 1900s, the Breguet Brothers built their first human carrying quadrotor helicopter called the Breguet–Richet Gyroplane No. 1. The Breguet Brothers found that such machine exhibited poor stability characteristics [21]. Although proving difficult to control, a quadrotor exhibits numerous advantages over other rotary wing UAVs such as helicopters. Control actuation consists of changing motor speeds rather than changing blade pitch (Fig. 1). Thus, quadrotors compare favorably to traditional helicopter design in the case of small, inexpensive electrically actuated UAVs where mechanical complexity is a disadvantage [22]. Recent advances in technology, including sensors and microcontrollers, now allow small UAV's to be built relatively easily and cheaply. Practical applications of UAV quadrotors will require a high level of controllability and flying capabilities. Successful autonomous operation of quadrotors in windy conditions while carrying an unknown payload remains an open problem.

2.1. Modeling of a quadrotor

The quadrotor moves with 6° of freedom (Fig. 2): movement in body-frame Cartesian coordinates (X,Y,Z) and rotation about the

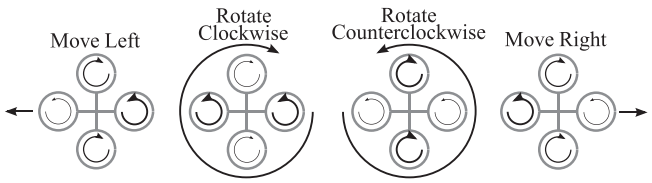

Fig. 1. Movement due to rotor speeds: thicker arrowed lines indicate increased rotor speed.

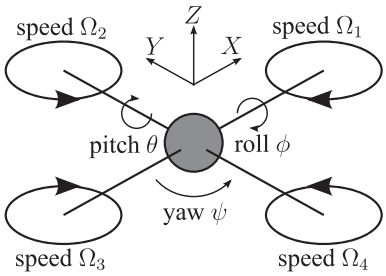

Fig. 2. Coordinates and assumption of rotor spin directions.

respective Cartesian axes (ϕ, θ, ψ are roll, pitch, and yaw respectively). We refer to the Cartesian Earth-frame coordinates as X_e, Y_e, Z_e

Different methods can produce dynamic equations of the helicopter model [22–26]. Ideally a model incorporates the nonlinear gyroscopic effects resulting from both the rigid body rotation in space and the four propulsion rotors rotation. Secondary effects include rotor acceleration, aerodynamic drag, the gearbox and the motor rotation. Our model relies on an assumption:

Assumption 1. The rotor (propeller) speeds are slowly varying making rotor acceleration a negligible effect in the dynamics.

Using Assumption 1 and ignoring other secondary effects a standard Newton–Euler formulation [27] gives the dynamics expressed in the body fixed frame

$$\mathbf{J}\dot{\vartheta} + \vartheta \times (\mathbf{J}\vartheta + h_{\mathbf{r}}\mathbf{e}_3) = \mathbf{T}, \tag{1}$$

where

- $\mathbf{J} = \mathrm{diag}(J_X, J_Y, J_Z)$ is the inertia matrix
- $\mathbf{T} \in \mathcal{R}^3$ is the applied torque from the propellers,
- $\vartheta = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$ are angular velocities,
- $h_{\mathbf{r}}$ is the angular momentum of the rotors and propellers,
- $\mathbf{e}_3 = [0 \quad 0 \quad 1]^T$ defines the axis of rotation of the rotors.

Again using Assumption 1 rotor angular momentum is approximately

$$h_{\mathbf{r}} = J_r\Omega_r = J_r(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4), \tag{2}$$

where J_r is rotor inertia and rotor speeds $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ have directions defined in Fig. 2.

In this work, no experiments identify numerical values of the parameters in (1). Rather, we use the mathematical form of (1) to design adaptive controls. As a rational for our approach, we point out that model-based linear controls exhibit stability in a limited region of attraction near the hovering position. Model-based nonlinear controls would not only require time-consuming system identification for each individual quadrotor, but may be inadequate for aircraft that pick up payloads.

A state-space model suitable for control design must include Z or Z_e, since vertical movement occurs as a function of rotor angular speeds $\mathbf{\Omega} = [\Omega_i, \Omega_2, \Omega_3, \Omega_4]^T$. We assume an outer control loop regulating Earth-frame X_e, Y_e commands desired angles ϕ_d, θ_d, so that state vector $\mathbf{x}$ contains

$$\begin{aligned} &x_1 = Z, \quad x_2 = \phi - \phi_d, \quad x_3 = \theta - \theta_d, \quad x_4 = \psi, \\ &x_5 = \dot{x}_1 \quad x_6 = \dot{x}_2 \quad x_7 = \dot{x}_3 \quad x_8 = \dot{x}_4 \end{aligned} \tag{3}$$

Regulating auxiliary state errors

$$\mathbf{z} = [\Lambda x_1 + x_5 \quad \Lambda x_2 + x_6 \quad \Lambda x_3 + x_7 \quad \Lambda x_4 + x_8]$$

where Λ is a positive constant, makes adaptive control design straightforward. Derivation of error dynamics appears in Appendix A and results in an equation of form

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{x}, \dot{\phi}_d, \dot{\theta}_d, \ddot{\phi}_d, \ddot{\theta}_d) + \mathbf{g}(\dot{\phi}, \dot{\theta}, \Omega_r) + \mathbf{d}(t) + \mathbf{B}(\boldsymbol{\phi}, \boldsymbol{\theta})\mathbf{u}, \tag{4}$$

where $\mathbf{d}(t)$ represents external disturbances, vector $\mathbf{f}$ is linear in its parameters, $\mathbf{g}$ describes gyroscopic forces, $\mathbf{B}$ contains constants on the diagonal, and $\mathbf{u}$ contains control signals described in Appendix A. A traditional adaptive control using a regressor matrix can estimate terms in $\mathbf{f}$. Note that secondary effects of drag and aerodynamic friction torque would appear in $\mathbf{f}$, and thus do not present a problem to an adaptive control. However, compensating only for $\mathbf{f}$ would result in terms in $\mathbf{g}$ being uncontrolled disturbances, since Ω_r from (2) depends on the angular velocities (control signals). Using a CMAC nonlinear approximator yields an advantage over traditional adaptive methods in this case, because a value from the previous time step can be provided as a CMAC input, $\Omega_r(t-\Delta)$, where Δ is the time period of the controller frequency. This provides the fuzzy approximator enough information to compensate for this term under most circumstances. The difference between Ω_Δ and $\Omega_r(t)$ still disturbs the system, but much less than Ω_r itself. Moreover, the effect is negligible by Assumption 1.

An Earth-frame Cartesian coordinate control simulation, using Z_e dynamics and an outer control loop for X_e, Y_e, appears in previous work [16]. However, the physical experiments in this paper constrict the quadrotor to angular motion and we leave free-flying motion to future work.

3. Approximate adaptive control

Achieving robustness to disturbances presents a challenge for nonlinear adaptive control techniques. Ideally, we wish the control system to be able to adapt to large model uncertainties, like acquired payloads, while still providing robustness to significant external disturbances like wind. In the results section, we test robustness by shaking the quadrotor in pitch, and in a separate test adding a payload on the *X*-axis arm.

3.1. Adaptive control of the roll, pitch and yaw

Albus's Cerebellar Model Arithmetic Computer (CMAC) was originally dubbed the Cerebellar Model Articulation Controller in [28]. With input vector $\mathbf{q} \in \mathcal{R}^n$ a CMAC can approximate a nonlinear function $h(\mathbf{q})$ in region $D \subset \mathcal{R}^n$ with uniform approximation error:

$$h(\mathbf{q}) = o(\mathbf{q}) + \epsilon(\mathbf{q})$$

where $o(\mathbf{q})$ is CMAC output and $\epsilon(\mathbf{q})$ a bounded approximation error ($|\epsilon| < \epsilon_{\max} \forall \mathbf{q} \in D$). CMAC output $o(\mathbf{q})$ follows from multiplying local basis functions in row vector $\boldsymbol{\Gamma}$ by adaptive parameters in vector $\mathbf{w}$ (*i.e.* $o = \boldsymbol{\Gamma}(\mathbf{q})\mathbf{w}$). Appendix B describes algorithmic computation of $\boldsymbol{\Gamma}$ and describes notation for a multiple output CMAC.

The following adaptive-fuzzy control strategy stabilizes the roll, pitch and yaw angles of the quadrotor helicopter in the presence of unknown parameters in the dynamics. The gyros and accelerometers estimate the roll and pitch angle. Fuzzy approximators estimate the nonlinear functions in the control law.

Consider adaptive-control Lyapunov function candidate

$$V = \frac{1}{2}\mathbf{z}^T\mathbf{B}^{-1}\mathbf{z} + \frac{1}{2\beta}\tilde{\mathbf{w}}^T\tilde{\mathbf{w}}, \tag{5}$$

where $\tilde{\mathbf{w}}$ denotes a vector of weight errors, and β is the adaptation gain. Differentiating (5) using (A.12) produces

$$\dot{V} = \mathbf{z}^T[\mathbf{B}^{-1}(\mathbf{f} + \mathbf{g} + \mathbf{d}(t)) + \mathbf{u}] - \frac{1}{\beta}\tilde{\mathbf{w}}^T\dot{\hat{\mathbf{w}}}, \tag{6}$$

From (A.12) we can linearly separate gyroscopic terms as

$$\mathbf{g}(t) = \mathbf{g}(\dot{\phi}(t), \dot{\theta}(t), \Omega_r(t-\Delta)) + \mathbf{g}_\Delta(t)$$

Thus, $\mathbf{g}_\Delta$ disturbs the system and the CMAC estimates the nonlinear terms

$$\mathbf{B}^{-1}[\mathbf{f}(\mathbf{q}) + \mathbf{g}(\dot{\phi}, \dot{\theta}, \Omega_r(t-\Delta)] = \boldsymbol{\Gamma}(\mathbf{q})\mathbf{w} + \boldsymbol{\epsilon}(\mathbf{q})$$

with approximation errors $\boldsymbol{\epsilon}$ and where the CMAC inputs are

$$\mathbf{q} = [\mathbf{x}^T \quad \dot{\phi}_d \quad \dot{\theta}_d \quad \ddot{\phi}_d \quad \ddot{\theta}_d \quad \Omega_r(t-\Delta)]^T$$

resulting in

$$\dot{V} = \mathbf{z}^T(\mathbf{o}(\mathbf{q}) + \boldsymbol{\epsilon}(\mathbf{q}) + \mathbf{g}_\Delta + \mathbf{d}(t) + \mathbf{u}) - \frac{1}{\beta}\tilde{\mathbf{w}}^T\dot{\hat{\mathbf{w}}}, \tag{7}$$

Assumption 1 implies $\mathbf{g}_\Delta \approx 0$ with small Δ, so ignoring this disturbance is consistent with approximations made in the dynamics already. Also assuming uniform approximation error and bounded disturbances, a positive constant $e_{\max}$ exists such that

$$\|\boldsymbol{\epsilon}(\mathbf{q}) + \mathbf{g}_\Delta(t) + \mathbf{d}(t)\| = \|\mathbf{e}(t, \mathbf{q})\| \leqslant e_{\max} \quad \forall \mathbf{q} \in D,\ \forall t \in \mathcal{R}^+$$

Since $\mathbf{o} = \boldsymbol{\Gamma}\mathbf{w}$ and $\tilde{\mathbf{w}} = \mathbf{w} - \hat{\mathbf{w}}$, rearranging (7) gives

$$\dot{V} = \mathbf{z}^T(\boldsymbol{\Gamma}(\mathbf{q})\hat{\mathbf{w}} + \mathbf{e} + \mathbf{u}) + \tilde{\mathbf{w}}^T\left(\boldsymbol{\Gamma}^T\mathbf{z} - \frac{1}{\beta}\dot{\hat{\mathbf{w}}}\right), \tag{8}$$

An appropriate control law and *e*-modification robust weight update law are

$$\mathbf{u} = -\boldsymbol{\Gamma}(\mathbf{q})\hat{\mathbf{w}} - \mathbf{K}\mathbf{z}, \tag{9}$$

$$\dot{\hat{\mathbf{w}}} = \beta(\boldsymbol{\Gamma}^T\mathbf{z} - \nu\|\mathbf{z}\|\hat{\mathbf{w}}) \tag{10}$$

with each β and ν positive constants and $\mathbf{K}$ a positive-definite symmetric matrix of gains. The *e*-modification term $(-\nu\|\mathbf{z}\|\hat{\mathbf{w}})$ makes the weight updates robust to each bounded approximation error and wind disturbances. Thus, the Lyapunov derivative (7) becomes:

$$\dot{V} = (-\mathbf{z}^T\mathbf{K}\mathbf{z} + \mathbf{z}^T\mathbf{e} + \nu\|\mathbf{z}\|\tilde{\mathbf{w}}^T\hat{\mathbf{w}}) \tag{11}$$

$$\dot{V} \leqslant \|\mathbf{z}\|(-\|\mathbf{z}\|K_{\min} + e_{\max} + \nu\|\tilde{\mathbf{w}}\|\|\mathbf{w}\| - \nu\|\tilde{\mathbf{w}}\|^2) \tag{12}$$

where $K_{\min}$ means the minimum eigenvalue of K. This ensures $\dot{V}(\mathbf{z}, \tilde{\mathbf{w}}) < 0$ outside of a compact set on the $(\|\mathbf{z}\|, \|\tilde{\mathbf{w}}\|)$ plane that includes the origin [20]. The smallest level set of V (Lyapunov surface) that encloses this compact set encloses an invariant set, *i.e.* the Lyapunov surface is a bound on trajectories that start within the Lyapunov surface. Furthermore, the signals are uniformly ultimately bounded (UUB).

3.2. The proposed method of alternate weights

The *e*-modification method prevents excessive weight drift, and the resulting bursting phenomenon, by limiting weight growth. However, the weights may need to grow large in order to compensate for large nonlinearities like added payloads. By limiting weight magnitudes, *e*-modification also limits performance in this kind of situation.

We proposed, instead, approximating the control CMAC output $\Gamma\hat{\mathbf{w}}$ with an alternate CMAC $\Gamma\hat{\mathbf{p}}$. The alternate weights $\hat{\mathbf{p}}$ should exhibit less variation in magnitude than the control weights $\hat{\mathbf{w}}$. The alternate weights can then serve to guide the training of the control weights. Reducing the variations in adjacent weights, *i.e.* spatial frequency content, will lead to a smoother applied control signal. The magnitude of the weights ceases to be a concern. We propose limiting variation by keeping the alternate weights close to their average value. Since the weights will be clustered closer to the mean, they avoid the extreme values of weights that cause control signal chatter in *e*-modification. In summary, the alternate weights must

1. produce approximately the same output, $\Gamma\hat{\mathbf{p}} \approx \Gamma\hat{\mathbf{w}}$
2. exhibit less variation, $\|\bar{\mathbf{p}} - \hat{\mathbf{p}}\| < \|\bar{\mathbf{w}} - \hat{\mathbf{w}}\|$
3. be bounded, $\|\hat{\mathbf{p}}\| \leqslant p_{\max}$

where $\bar{\mathbf{p}}$ and $\bar{\mathbf{w}}$ indicate mean values of alternate and control weight vectors respectively and $p_{\max}$ is a positive constant.

In order to train the alternate CMAC for requirement (1) above, one can use learning error

$$\boldsymbol{\tau} = \boldsymbol{\Gamma}\hat{\mathbf{w}} - \boldsymbol{\Gamma}\hat{\mathbf{p}}. \tag{13}$$

in a gradient descent update rule. To prevent adjacent alternate weights from drifting in opposite directions (positive and negative), we introduce a term $\rho(\bar{\mathbf{p}} - \hat{\mathbf{p}})$ in $\dot{\hat{\mathbf{p}}}$ that keeps them close to the mean, where ρ is a positive constant and the $\bar{\mathbf{p}}$ differs for each output in the four-output CMAC:

$$\bar{\mathbf{p}} = \left[\bar{\mathbf{p}}_1^T\ \bar{\mathbf{p}}_2^T\ \bar{\mathbf{p}}_3^T\ \bar{\mathbf{p}}_4^T\right]^T \quad \text{where} \quad \bar{\mathbf{p}}_i = \mathbf{I}\cdot\text{mean}\ (\hat{\mathbf{p}}_i)\ \text{ for } i = 1,2,3,4$$

An experiment can identify a correct learning deadzone of size δ that bound signals. A small leakage term $-\nu\hat{\mathbf{p}}$ becomes necessary in the weight update to guarantee boundedness of alternate weights in the case when an appropriate value of δ is not known. The proposed alternate CMAC update rule is

$$\dot{\hat{\mathbf{p}}} = \begin{cases} \kappa[\alpha\boldsymbol{\Gamma}^T\boldsymbol{\tau} + \rho(\bar{\mathbf{p}} - \hat{\mathbf{p}}) - \eta\hat{\mathbf{p}}] & \text{if } \|\boldsymbol{\tau}\| > \delta \\ \kappa[\rho(\bar{\mathbf{p}} - \hat{\mathbf{p}}) - \eta\hat{\mathbf{p}}] & \text{otherwise} \end{cases} \tag{14}$$

where α, κ, and η are positive constants. If one had good knowledge of the system *i.e.* from extensive pre-training, choice of deadzone size $\delta = \tau_{\max}$ would be appropriate (Appendix B).

We now propose control weight updates that use the alternate weights to guide the training

$$\dot{\hat{\mathbf{w}}} = \begin{cases} \beta(\boldsymbol{\Gamma}^T\mathbf{z} - \alpha\boldsymbol{\Gamma}^T\boldsymbol{\tau} + \zeta_1(\hat{\mathbf{p}} - \hat{\mathbf{w}})) & \text{if } \|\boldsymbol{\tau}\| > \delta \\ \beta(\boldsymbol{\Gamma}^T\mathbf{z} + \zeta_2(\hat{\mathbf{p}} - \hat{\mathbf{w}})) & \text{otherwise} \end{cases} \tag{15}$$

where ζ_1 and ζ_2 are positive constants. We choose ζ_2 much larger than ζ_1 in order to force the control weights close to the alternate weights when the learning error is small. The term $-\alpha\boldsymbol{\Gamma}^T\boldsymbol{\tau}$ helps guarantee bounded signals in the Lyapunov-derivation of Appendix C, where all signals are shown to be uniformly ultimately bounded.

4. Experimental prototype

To conduct control experiments, one can modify the electronics on commercially-available quadrotors like Dragonflyer from RC Toy and EADS Quattrocopter MAV12 or obtain vehicles designed for researchers like The X-4 Flyer. However, building our own quadrotor prototype provided low-cost control algorithm testing (Fig. 4). Moreover, the quadrotor design ensures modeling assumptions are met, especially the rigid frame, and the electronics are suitable for accomplishing the proposed control method.

The rigid Q4 Dragster frame from lipoly.de consists of a lightweight carbon fiber electronics cage with four aluminum arms. The arms are 23.2 cm long from center to tip, and the total mass

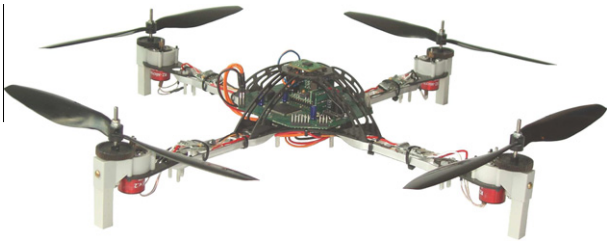

Fig. 4. Prototype quadrotor.

of the prototype (including battery and electronics) measures 753 g. Testing established the four rotors lift 960 g when running at 3500 RPM (366 rad/s), and speeds of close to 6000 RPM are possible.

4.0.1. Power train

Four custom-designed carbon fiber plates hold the motors: Razor Micro-Heli v2.0s, 3-pole brushless DC motors specifically designed for small helicopter applications. Plastic gearboxes provided 6.6:1 gear ratios. The four rigid plastic blade rotors (propellers) come from Maxx Products (EPP1045), and are 10 in. in diameter with a 4.5 in. pitch. A Castle Creations Phoenix-10 electronic speed controller provides power to each motor. Calibration of rotor speeds occurred via a simple tachometer built using an OP550A photo transistor and an IR LED on placed on opposite sides of each rotor. An oscilloscope measured the frequency compared to the PWM duty cycle, and a 4th-order polynomial models the relationship in the controller

$$\text{duty cycle}\,\% = \frac{-9.3\times10^{-8}\Omega^4 + 1.1\times10^{-4}\Omega^3 - 3.7\times10^{-2}\Omega^2 + 6.5\Omega + 4.7\times10^3}{9667}. \tag{16}$$

Note that this represents the only identification of quadrotor characteristics in this work used in the control; the adaptive control compensates for all other parameters.

The battery, a Parkzone PKZ1030, provides a nominal 11.1 V, 2.2Ah 3-cell lithium polymer (LiPo) battery, with a maximum continuous discharge capability of 27 A. The quadrotor uses about 10 A near equilibrium, giving at least 13 min of full-voltage flight. The battery weighs 159 g.

4.0.2. Electronics

The microcontroller, a Freescale MC9S12XDP512, offers 32 KB ram and 512 KB flash for program storage. It provides sixteen 10-bit analog to digital conversion channels, four 16-bit pulse width modulation (PWM) channels, as well as several SCI, SPI,

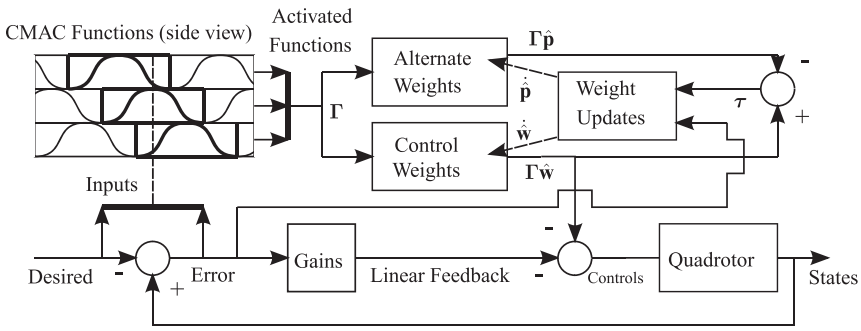

Fig. 3. Proposed method of alternate weights.

and I^2C serial interfaces for data transmission and communication with external devices. Moreover, memory and speed are sufficient to run a CMAC neural network with 10 layers at 130 Hz.

An Aerocomm AC4490-1000 transceiver provides data transmission between the quadrotor and a base station. The unit operates in the 902–928 MHz frequency band up to speeds of 115.2 kbps, and frequency hopping avoids interference. The transceiver takes up a small space, $1.65 \times 1.9 \times 0.2$ in., and is 3 or 5 V compatible. Adjustable output power (5–1000 mW) allows communication up to 32 km.

Each gyroscope, an Analog Devices ADXRS150, provides an analog signal proportional to the angular velocity (range = ±150 deg/s) as measured through an axis normal to the top surface of the sensor. The gyroscope exhibits nominal sensitivity of 12.5 mV/deg/s, with a 2.5 V bias at 25 °C. We set the bandwidth of the output filters to 5 Hz and then mounting gyros on rubber standoffs results in absolute maximum output errors of 2 deg/s. Numerical integration provides roll, pitch, and yaw measurements which are subject to drift. Measurements from a linear accelerometer, an Analog Devices ADXL203, give a measure of tilt. A digital compass, Honeywell's HMC6352, gives the heading. Using the accelerometer and compass value errors (with gains of $K = 0.5$) eliminates drift:

$$\theta_{\text{estimate}} = \int (\dot{\theta}_{\text{gyro}} + K(\theta_{\text{accelerometer}} - \theta_{\text{estimate}}))dt, \tag{17}$$

$$\phi_{\text{estimate}} = \int (\dot{\phi}_{\text{gyro}} + K(\phi_{\text{accelerometer}} - \phi_{\text{estimate}}))dt, \tag{18}$$

$$\psi_{\text{estimate}} = \int (\dot{\psi}_{\text{gyro}} + K(\psi_{\text{compass}} - \psi_{\text{estimate}}))dt. \tag{19}$$

4.0.3. Test stand

The test stand built for initial testing of the prototype consists of a thick wooden base, with a 3.5 foot long (1/2 in. thick) copper pipe providing the main support for the quadrotor. By using a longer pipe, ground effects were minimized. The pipe was reinforced with a 3/4 in. thick pipe and four stainless steel guy-wires (with turnbuckles to increase tension) after initial testing showed a single pipe was not rigid enough. Atop the support pole sits a spherical bearing, the EFOI-10 from IGUS, to which the prototype can be attached. The bearing allows the prototype to roll or pitch up to ±24°, while yaw is essentially unlimited.

5. Results and discussion

The results appear in two subsections: simulation and experiment. In both, addition of a significant payload tests the controllers. In the simulations, we compare neural-adaptive control using CMAC to both linear-quadratic-regular (LQR) and linear-in-parameters (LP) adaptive control, without looking at the issue of weight (parameter) drift in the presence of disturbances. The simulations merely establish the benefits of, and motivation for, using neural-adaptive control. The physical experiments with the prototype quadrotor use only neural-adaptive controls and compare our proposed robust weight-update method to *e*-modification. The experiments test both adaption to a payload and prevention of weight drift during a shake-test disturbance.

5.1. Simulation: comparison to other controls

Except for mass and distance, the simulation parameters in Table 2 are only estimates of our prototype; since we are relying on adaptive methods we will not require a precise model in the experimental section.

Table 2
Simulation parameters.

	Definition	Value
b	Thrust factor	3.13e−5 Ns^2
d	Drag factor	7.5e−7 $Nm\,s^2$
J_X	X Inertia	6.228e−3 $kg\,m^2$
J_Y	Y Inertia	6.225e−3 $kg\,m^2$
J_Z	Z Inertia	1.121e−2 $kg\,m^2$
J_r	Rotor inertia	6e−5 $kg\,m^2$
m	Mass	0.52 kg
L	Distance	0.232 m

The LQR control design, based on the linearized version of (4) with zero angles and gravity-balanced rotor speeds, has LQR weighting matrices $Q = 1e5\mathbf{I}$, $R = \mathbf{I}$. The LP adaptive control uses, similar to (9), law

$$\mathbf{u} = \mathbf{B}^{-1}\boldsymbol{\Gamma}_{\text{LP}}\hat{\mathbf{p}} - \mathbf{K}\mathbf{z}. \tag{20}$$

Note known constant parameters appear in $\mathbf{B}$ without loss of generality, since the adaptive laws can easily adjust to differing constants. The regressor matrix $\boldsymbol{\Gamma}_{\text{LP}}$ contains known functions and $\hat{\mathbf{p}}$ a vector of unknown parameters. The parameter updates follow from the same Lyapunov method used for designing weight updates. The result is

$$\boldsymbol{\Gamma}_{\text{LP}} = \begin{bmatrix} \beta_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \dot{\theta}\dot{\psi} & \dot{\theta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \dot{\phi}\dot{\psi} & \dot{\phi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dot{\phi}\dot{\theta} \end{bmatrix}, \quad \dot{\hat{\mathbf{p}}} = \beta\left(\boldsymbol{\Gamma}_{\text{LP}}^T\mathbf{z} - v\|\mathbf{z}\|\hat{\mathbf{p}}\right). \tag{21}$$

Note that we put gain parameter β_z inside $\boldsymbol{\Gamma}_{\text{LP}}^T$ for convenience, providing extra gain on the adaptive parameter in the Z coordinate in order to deal with a large payload. In the simulation the following controller parameters are used:

$$\beta = 1e5, \quad \beta_z = 10 \quad K_1 = 1.6e7 \quad K_2 = K_3 = K_4 = 2e3 \quad v = 0$$

Note that choosing $v = 0$ tests performance, and we leave testing robust weight updates to the experimental section. The CMAC neural-adaptive control uses the same $\mathbf{K}$, β_z v but $\beta = 1e6$, with 10 CMAC layers and CMAC cell sizes as follows:

$$3^{\circ} \text{ for } \phi, \theta, \psi, \quad 10\,\text{deg/s for } \dot{\phi}, \dot{\theta}, \dot{\psi}, \quad 10\,\text{cm for } Z, \quad 10\,\text{cm/s for } \dot{Z}$$

The results of adding and taking a away a payload of 100 gm at the end of the 'pitch' arm (creating a significant pitch torque) are shown in Fig. 7. Clearly, the LQR control is not sufficient to achieve

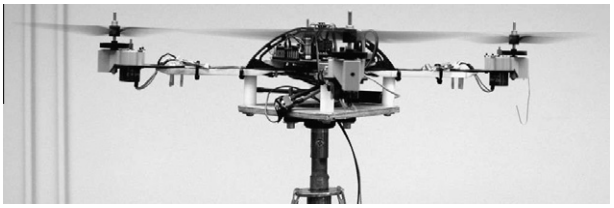

Fig. 5. No payload (disturbance-free) configuration.

Fig. 6. Configuration with payload (step-disturbance).

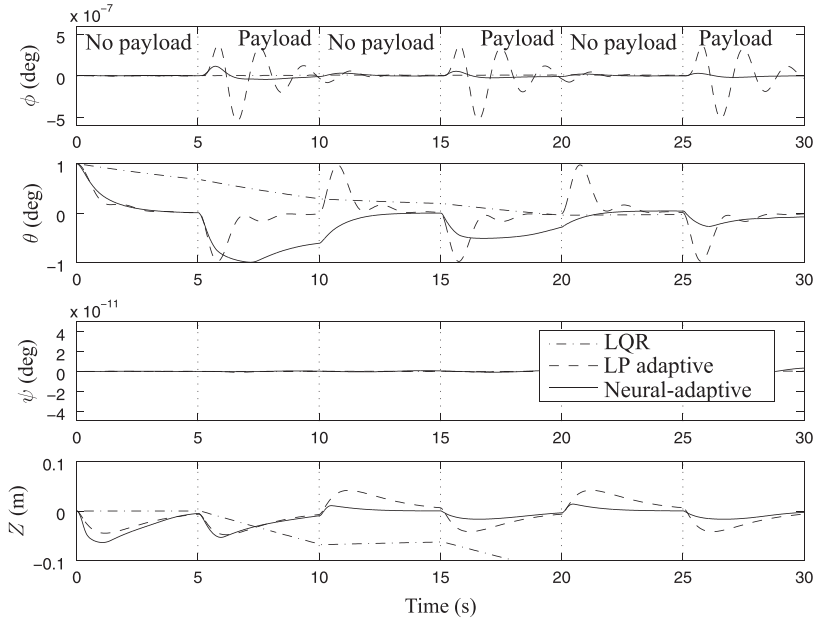

Fig. 7. Simulation results: adding and removing payload at 5 s intervals ($\theta(t=0)=1°$).

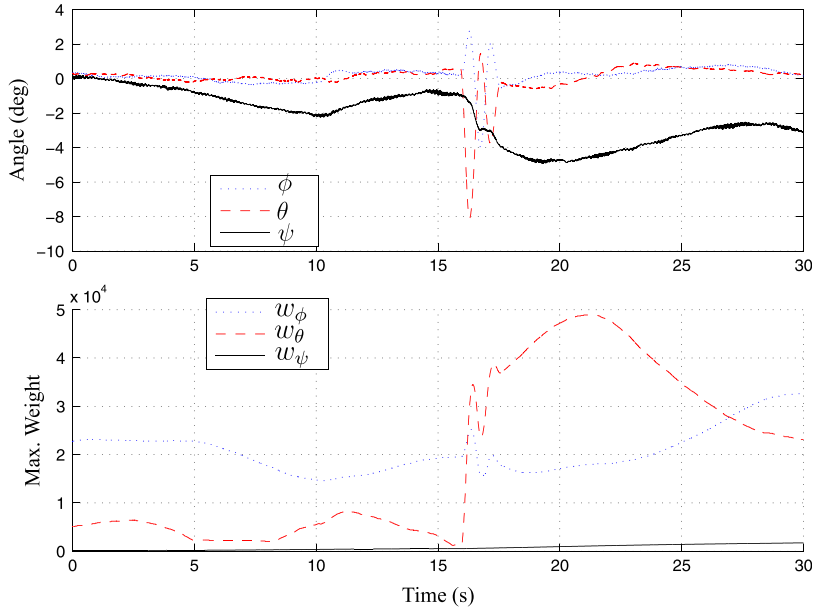

Fig. 8. No robust modification: acquiring payload.

adequate performance due to both nonlinearities and inability to adapt to payloads without an integral term. The LP adaptive control can adjust to payloads, but achieves the same performance every time a payload is added. The neural-adaptive control adjusts to payloads nearly as quickly as the LP adaptive control, and improves performance significantly the next time a payload is added since it has learned how to react. The neural-adaptive control does not overshoot as much as the LP adaptive control when the payload removed. Moreover, performance may keep improving every time the same payload is added, with convergence level depending on the approximation ability (size) of the CMAC.

5.2. Experiment

During the experiments, the quadrotor remains mounted to the bearing allowing motion in only roll, pitch, and yaw (Fig. 5). Since the quadrotor is constrained in Z, we provided a constant control $u_1 = 162{,}977$. The linear control gains for the other control signals are

$$\Lambda = 4 \quad K_2 = K_3 = 75{,}000 \quad K_4 = 50{,}000$$

The number of CMAC layers and cell sizes are the same as used in the simulation.

Testing adaptation ability, we add a payload after the quadrotor has found a stable equilibrium without payload. Specifically, a mass of 90 gm is hung from the X-axis arm (Fig. 6), equivalent to a step-disturbance of 0.021 Nm in pitch torque. The adaptive control should be able to regain the equilibrium. In another experiment, a shake test validate robustness to disturbances (without payload), where the X-axis arm is grasped and forced up and down. The shaking occurs at frequencies slightly higher than 1 Hz and amplitude θ in the 3–7° range.

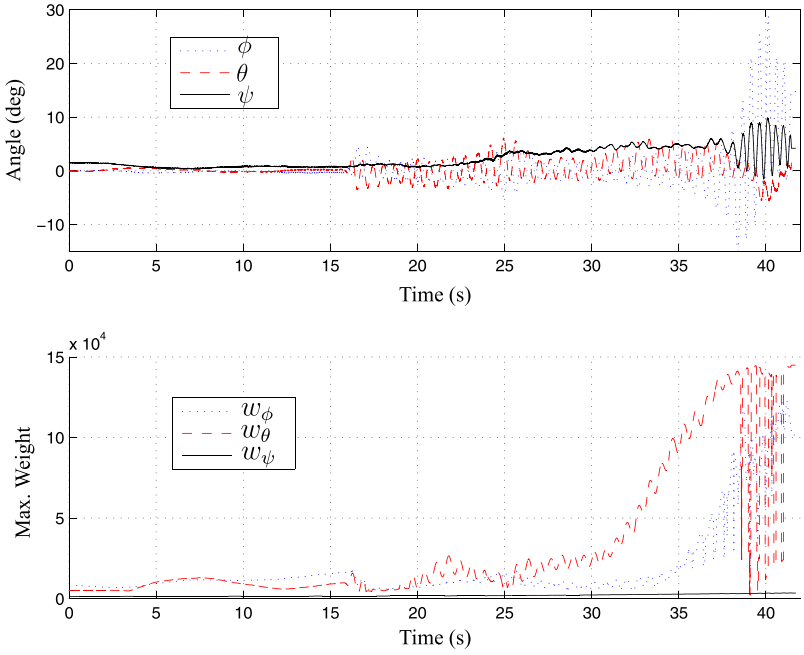

Fig. 9. No robust modification: shake test.

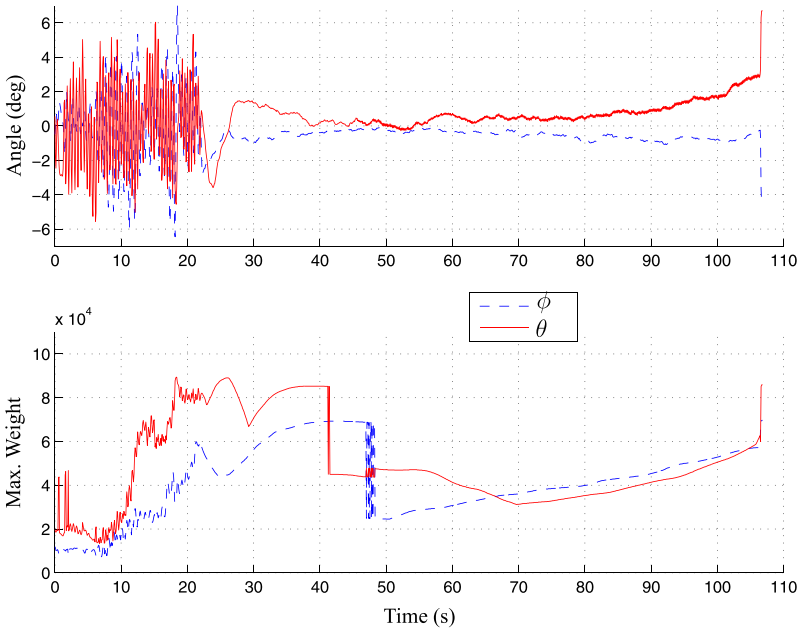

Fig. 10. *e*-modification with $\nu = 0.00005$: shake test with bursting.

The weights should find an equilibrium during and after the shake test, rather than drifting to large values. Note that all graphs of the weights' behavior show the activated weight with largest absolute value (with discontinuities resulting from a cell being deactivated and then an adjacent cell activated). Since weight drift is not frequency dependent, slower-frequency wind buffeting would also cause weights to grow slowly in outdoor operations.

5.2.1. No robust modification

Establishing the need for robust methods, the adaptation mechanism without robust modification, (10) with $\nu = 0$, first controls the quadrotor. When acquiring a payload at the 16 second mark, the pitch angle θ initially dips by 8° (Fig. 8, top). As the control system reduces θ error, roll ϕ and yaw ψ are also affected due to the coupled dynamics. Both θ and ϕ return to equilibrium after 15 s. Since control torque in yaw provides less actuation from changing rotor speeds than in pitch and roll, the significant friction introduced by the test-stand results in ψ taking far longer to return to zero. For this reason, the result of ψ dynamics does not appear in subsequent graphs.

When acquiring the payload, the weights remain well-behaved and no bursting occurs (Fig. 8, bottom). During the shake test, however, the weights grow too large after 31 s (Fig. 9, bottom) causing the errors in ϕ and ψ to increase dramatically after 36 s (Fig. 9, top). This test clearly establishes the need for a robust modification that can halt weight drift.

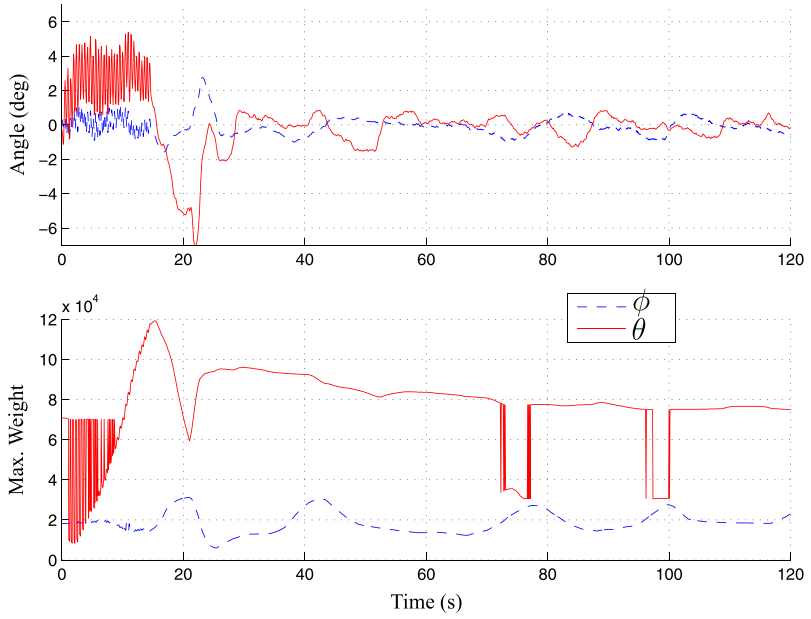

Fig. 11. *e*-modification with $v = 0.0001$: successful shake test.

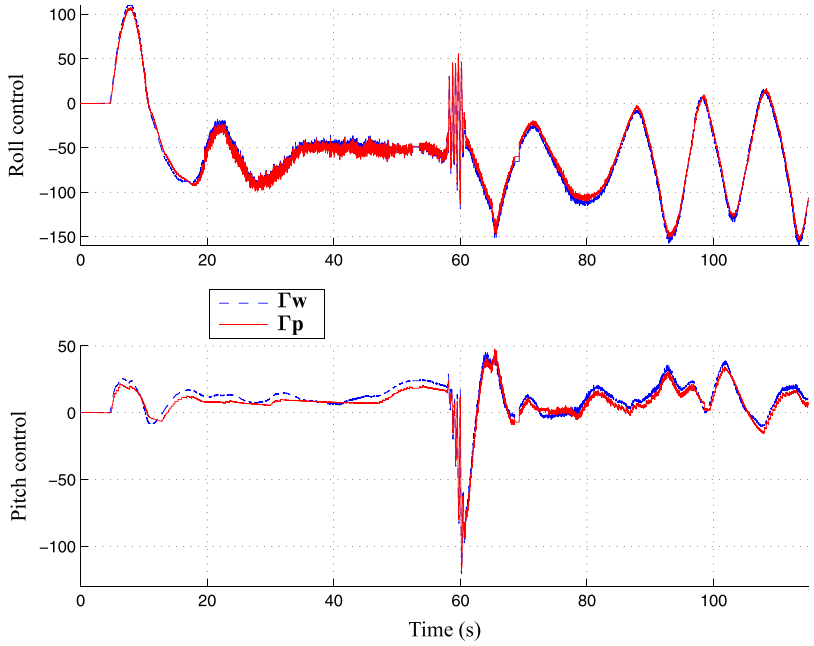

Fig. 12. Proposed method: training of alternate CMAC.

5.2.2. e-Modification

Through various experimentation using *e*-modification with $\beta = 1000$ we found a value of $v = 0.0001$ to be the most appropriate to prevent weight drift while still allowing adaptation to the 90 g mass. A slightly lower value of $v = 0.00005$ does not prevent the ϕ weights from drifting during 20 s of the shake test (Fig. 10, bottom). Moreover, after removing the forced oscillation the weights drift and eventually cause bursting after another 80 s (Fig. 10, top). The system then goes unstable at 106 s. The value of $v = 0.0001$, on the other hand, prevents ϕ weights from growing during and after the shake test (Fig. 11, bottom), thus preventing bursting (Fig. 11, top). Larger values of v decrease performance when acquiring the payload. Note that even successful *e*-modification results in large discontinuities in weight values when moving between cells, as can be seen in pitch weights in (Fig. 11, bottom).

5.2.3. Proposed method

Through various experimentation the proposed method with parameters chosen as

$$\begin{aligned} &\beta = 1000 \quad \kappa = 1000 \quad \alpha = 0.001 \quad \rho = 0.001 \\ &\eta = 0.000001 \quad \zeta_1 = 0.000001 \quad \zeta_2 = 0.00001 \end{aligned} \tag{22}$$

was found to be appropriate. A test without learning deadzone, acquiring a payload, shows that the alternate CMAC can approximate the control CMAC to a tolerance of 10 in the physical control signal (Fig. 12), and thus subsequent tests use learning deadzone $\delta = 10$. The weights remain well-behaved during and after the shake

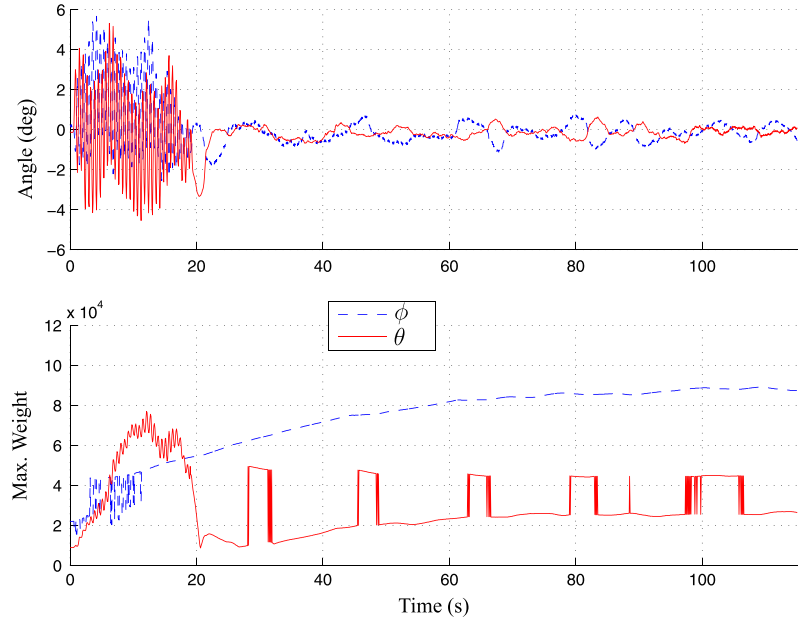

Fig. 13. Proposed method: shake test.

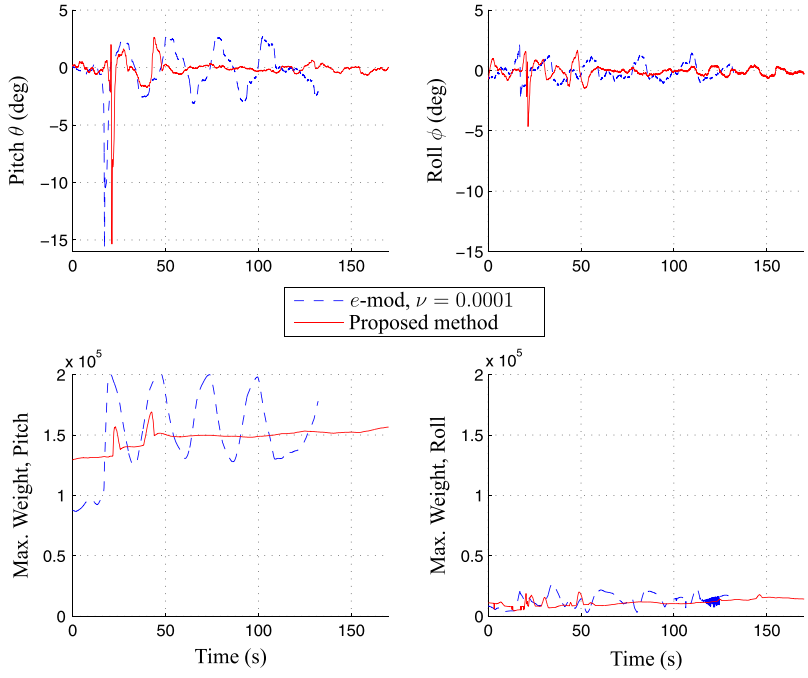

Fig. 14. Comparing *e*-modification with proposed method: acquiring payload.

test (Fig. 13, bottom), and afterward no bursting occurs (Fig. 13, top). Moreover, the discontinuity in weight values between adjacent cells is about one third that of *e*-modification.

When acquiring the payload at 16 s, the proposed method and *e*-modification both dip by 15° (Fig. 14, top left). Thus, both methods sacrifice some performance compared to the ideal performance found with no robust modification. However, the proposed method reaches equilibrium after 50 s whereas *e*-modification reaches a stable limit cycle consisting of an oscillation of about 3° amplitude in pitch (Fig. 14, top left) and 1 degree in roll (Fig. 14, top right). The *e*-modification limit cycle has weights growing and shrinking (Fig. 14, bottom graphs) identified as repetitive bursting. In contrast, the proposed method successfully adapts to the added payload using the same parameters that resulted in robustness in the shake test.

6. Conclusions

Construction of a unique prototype allowed us to test approximate adaptive methods for quadrotor control. Choosing the CMAC algorithm provides a computationally-efficient and accurate nonlinear approximator that can adapt quickly. Requiring the quadrotor to pick a significant unknown payload means previously proposed robust update methods, that limit the size of adaptive weights, are not appropriate. We propose a robust meth-

od that relies on an alternate set of weights to guide the training of the weights used in the controller CMAC. The alternate set of weights is trained online to approximate the same output, but with weights clustered closer to the mean. The resulting weights avoid the extreme variation in weight values that tend to lead to bursting or instability. A test-stand constrains quadrotor motion to roll, pitch, and yaw during the experiments which compare the proposed method to commonly-used *e*-modification. A shake test allows us to pick appropriate robust controller parameters for the two methods, which prevent weight drift and subsequent bursting. The proposed method then significantly outperforms *e*-modification when acquiring a payload, while preventing weight drift and bursting.

Appendix A. Quadrotor modeling

A.1. Control signals

Here we show how simple state-space controls proportional to forces/torques can be transformed into rotor speeds, which are in turn transformed into supply voltages. The first state is Z and the actuating force is proportional to the square of rotor speeds

$$F_Z = b\Omega_1^2 + b\Omega_2^2 + b\Omega_3^2 + b\Omega_4^2, \tag{A.1}$$

where b is a constant thrust coefficient of the propellers. The torques provided by propellers rotating as in Fig. 2 are

$$\mathbf{T} = \begin{bmatrix} T_\phi \\ T_\theta \\ T_\psi \end{bmatrix} = \begin{bmatrix} Lb\left(\Omega_4^2 - \Omega_2^2\right) \\ Lb\left(\Omega_3^2 - \Omega_1^2\right) \\ D\left(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2\right) \end{bmatrix}, \tag{A.2}$$

where L is length from rotor to quadrotor center of mass, and D is a constant drag factor. Solving (A.1) and (A.2) for angular speeds gives

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} 0.25 & 0 & -0.5 & 0.25 \\ 0.25 & -0.5 & 0 & -0.25 \\ 0.25 & 0 & 0.5 & 0.25 \\ 0.25 & 0.5 & 0 & -0.25 \end{bmatrix} \begin{bmatrix} F_Z/b \\ T_\phi/(Lb) \\ T_\theta/(Lb) \\ T_\psi/D \end{bmatrix}. \tag{A.3}$$

Thus, rather than design speeds directly we choose to design controls $u_1 = F_Z/b$, $u_2 = T_\phi/(Lb)$, $u_3 = T_\theta/(Lb)$, $u_4 = T_\psi/D$. Note, this will not require knowledge of b or D. Thus, once controls have been calculated the desired speeds follow from (A.3) and the PWM commands to the motor controllers from (16).

A.2. Dynamics in state space

The resulting dynamics suitable for control design are

$$\ddot{Z} = -gC_\phi C_\theta + \frac{b}{m}u_1, \tag{A.4}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\frac{J_Y - J_Z}{J_X} - \frac{J_r}{J_X}\dot{\theta}\Omega_r + \frac{Lb}{J_X}u_2, \tag{A.5}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\frac{J_Z - J_X}{J_Y} + \frac{J_r}{J_Y}\dot{\phi}\Omega_r + \frac{Lb}{J_Y}u_3, \tag{A.6}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\frac{J_X - J_Y}{J_Z} + \frac{D}{J_Z}u_4. \tag{A.7}$$

where notation $C_\phi = \cos(\phi)$ is used, and g is gravitational acceleration constant. Alternatively, one can describe (A.4) in the Earth-frame

$$\ddot{Z}_e = -g + C_\phi C_\theta \frac{b}{m}u_1.$$

A full discussion of measuring vertical displacement and the corresponding control is beyond the scope of the present paper. Using states defined in (3) the state-space model is

$$\dot{x}_i = x_{i+4} \quad \text{for } i = 1,2,3,4, \tag{A.8}$$

$$\dot{x}_5 = -g\cos(x_2 + \phi_d(t))\cos(x_3 + \theta_d(t)) + b_1u_1, \tag{A.9}$$

$$\dot{x}_6 = x_7x_8a_1 + x_7a_2\Omega_r - \ddot{\phi}_d(t) + b_2u_2, \tag{A.10}$$

$$\dot{x}_7 = x_6x_8a_3 + x_6a_4\Omega_r - \ddot{\theta}_d(t) + b_3u_3, \tag{A.11}$$

$$\dot{x}_8 = x_6x_7a_5 + b_4u_4, \tag{A.12}$$

with constant parameters

$$\begin{aligned} &a_1 = (J_Y - J_Z)/J_X, \quad a_2 = -J_r/J_X, \quad a_3 = (J_Z - J_X)/J_Y, \\ &a_4 = J_r/J_Y, \quad a_5 = (J_X - J_Y)/J_Z, \\ &b_1 = b/m, \quad b_2 = Lb/J_X, \quad b_3 = Lb/J_Y, \quad b_4 = D/J_Z. \end{aligned} \tag{A.13}$$

Appendix B. CMAC algorithm

In the CMAC n inputs $q_1 \cdots q_n$ index (*activate*) one n-dimensional hypercube (*cell*) in each look-up table (*layer*). The m layers each contain q^n cells in an array format, where q is referred to as the quantization number. The layers are offset resulting in a slightly different domain for each activated cell. In the original binary CMAC, the weights associated with each activated cell are added together to get the output. We use splines as activation functions to achieve a smooth output. The value of an activated spline for the ith input on the jth layer is calculated

$$s_{ij}(q_i) = h_j^2(q_i) - 2h_j^3(q_i) + h_j^4(q_i), \tag{B.1}$$

where $0 < h_j(q_i) < 1$ is the normalized position inside the activated cell on layer j. The value of activation function is zero when the cell is not activated. Numbering all the cells $k = 1 \cdots m\,q^n$ the kth activation function is thus

$$a_k(\mathbf{q}) = \begin{cases} \prod_{i=1}^{n} s_{ik}(q_i) & \text{if cell } k \text{ is activated,} \\ 0 & \text{otherwise.} \end{cases} \tag{B.2}$$

A single input CMAC with three layers ($n = 1, m = 3, q = 3$) is illustrated as part of the control diagram in Fig. 3, showing the activated cells in bold and illustrating all activation functions. We normalize the activation functions before multiplying them by the weights, thus for the jth activated cell the basis function is

$$\gamma_j = a_j \Big/ \left(\sum_{k=1}^{mq^n} a_k\right). \tag{B.3}$$

Normalizing makes the CMAC equivalent to fuzzy approximators, with the normalized activation functions interpreted as truth values. Rather than calculating all basis functions, only activated (non-zero) functions are calculated on-line. The scalar output is

$$o = [\gamma_{1,\text{activated}} \quad \gamma_{2,\text{activated}} \quad \cdots \quad \gamma_{m,\text{activated}}]\mathbf{w} = \boldsymbol{\gamma}(\mathbf{q})\mathbf{w}. \tag{B.4}$$

where $\mathbf{w} \in \mathcal{R}^m$ is a vector of activated weights. Rather than allocating memory for all weights, activated (and previously activated) weights are stored in a one-dimensional array using hash-coding [29]. The quadrotor dynamics requires four outputs and we use notation

$$\mathbf{o} = \boldsymbol{\Gamma}(\mathbf{q})\mathbf{w}. \tag{B.5}$$

where the same basis functions are used for each output $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma}, \boldsymbol{\gamma}, \boldsymbol{\gamma}, \boldsymbol{\gamma})$ and $\mathbf{w} = \begin{bmatrix} \mathbf{w}_1^T & \mathbf{w}_2^T & \mathbf{w}_3^T & \mathbf{w}_4^T \end{bmatrix}^T$.

Appendix C. Stability proof

This analysis establishes that control (9) with weight updates (14) and (15), applied to system (A.12) results in UUB signals. Consider direct adaptive control Lyapunov function

$$V = \frac{1}{2}\mathbf{z}^T\mathbf{B}^{-1}\mathbf{z} + \frac{1}{2\beta}\tilde{\mathbf{w}}^T\tilde{\mathbf{w}} + \frac{1}{2\kappa}\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}, \tag{C.1}$$

where κ is the adaptation gain for alternate weights. The ideal weights $\mathbf{w}$ are the same for both control and alternate weights, and thus weight errors are

$$\tilde{\mathbf{w}} = \mathbf{w} - \hat{\mathbf{w}}, \tag{C.2}$$

$$\tilde{\mathbf{p}} = \mathbf{w} - \hat{\mathbf{p}}. \tag{C.3}$$

Using the result without $\tilde{\mathbf{p}}$ from (5) and (11) we write

$$\dot{V} = -\mathbf{z}^T\mathbf{K}\mathbf{z} + \mathbf{z}^T\mathbf{e} + H(\mathbf{z}, \tilde{\mathbf{w}}, \dot{\tilde{\mathbf{w}}}, \tilde{\mathbf{p}}, \dot{\tilde{\mathbf{p}}}), \tag{C.4}$$

where

$$H(\mathbf{z}, \tilde{\mathbf{w}}, \dot{\tilde{\mathbf{w}}}, \tilde{\mathbf{p}}, \dot{\tilde{\mathbf{p}}}) = \tilde{\mathbf{w}}^T\left(\mathbf{\Gamma}^T\mathbf{z} - \frac{\dot{\tilde{\mathbf{w}}}}{\beta}\right) + \tilde{\mathbf{p}}^T\left(-\frac{\dot{\tilde{\mathbf{p}}}}{\kappa}\right). \tag{C.5}$$

C.1. Outside the deadzone

With weight updates (14) and (15), for the case of $\|\boldsymbol{\tau}\| > \delta$ (C.5) becomes

$$H = \alpha\tilde{\mathbf{w}}^T\mathbf{\Gamma}^T(\mathbf{\Gamma}\hat{\mathbf{w}} - \mathbf{\Gamma}\hat{\mathbf{p}}) - \zeta_1\tilde{\mathbf{w}}^T(\hat{\mathbf{p}} - \hat{\mathbf{w}}) - \alpha\tilde{\mathbf{p}}^T\mathbf{\Gamma}^T(\mathbf{\Gamma}\hat{\mathbf{w}} - \mathbf{\Gamma}\hat{\mathbf{p}}) - \rho\tilde{\mathbf{p}}^T(\bar{\mathbf{p}} - \hat{\mathbf{p}}) + \eta\tilde{\mathbf{p}}^T\hat{\mathbf{p}}. \tag{C.6}$$

Since $(\hat{\mathbf{w}} - \hat{\mathbf{p}}) = (\tilde{\mathbf{p}} - \tilde{\mathbf{w}})$, following from (C.2), (C.3),

$$H = -\alpha(\mathbf{\Gamma}\tilde{\mathbf{w}} - \mathbf{\Gamma}\tilde{\mathbf{p}})^T(\mathbf{\Gamma}\tilde{\mathbf{w}} - \mathbf{\Gamma}\tilde{\mathbf{p}}) - \zeta_1\tilde{\mathbf{w}}^T(\tilde{\mathbf{w}} - \tilde{\mathbf{p}}) - \rho\tilde{\mathbf{p}}^T(\bar{\mathbf{w}} - \bar{\tilde{\mathbf{p}}} - \mathbf{w} + \tilde{\mathbf{p}}) + \eta\tilde{\mathbf{p}}^T\mathbf{w} - \eta\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}. \tag{C.7}$$

Using

$$-\rho\tilde{\mathbf{p}}^T(\bar{\mathbf{w}} - \bar{\tilde{\mathbf{p}}} - \mathbf{w} + \tilde{\mathbf{p}}) \leqslant \rho(\|\tilde{\mathbf{p}}\|\|\mathbf{w}\| + \|\tilde{\mathbf{p}}\|^2 + \|\tilde{\mathbf{p}}\|\|\mathbf{w}\| - \|\tilde{\mathbf{p}}\|^2), \tag{C.8}$$

results in

$$H \leqslant -\zeta_1\|\tilde{\mathbf{w}}\|^2 + \zeta_1\|\tilde{\mathbf{w}}\|\|\tilde{\mathbf{p}}\| + (2\rho + \eta)\|\tilde{\mathbf{p}}\|\|\mathbf{w}\| - \eta\|\tilde{\mathbf{p}}\|^2. \tag{C.9}$$

and thus

$$\dot{V} \leqslant \begin{bmatrix} \|\mathbf{z}\| \\ \|\tilde{\mathbf{w}}\| \\ \|\tilde{\mathbf{p}}\| \end{bmatrix}^T \begin{bmatrix} -K_{\min} & 0 & 0 \\ 0 & -\zeta_1 & \zeta_1/2 \\ 0 & \zeta_1/2 & -\eta \end{bmatrix} \begin{bmatrix} \|\mathbf{z}\| \\ \|\tilde{\mathbf{w}}\| \\ \|\tilde{\mathbf{p}}\| \end{bmatrix} + [e_{\max} \quad 0 \quad (2\rho + \eta)\|\mathbf{w}\|] \begin{bmatrix} \|\mathbf{z}\| \\ \|\tilde{\mathbf{w}}\| \\ \|\tilde{\mathbf{p}}\| \end{bmatrix}. \tag{C.10}$$

The matrix in the first term contains eigenvalues with negative real parts (is negative definite) as long as the parameters are chosen such that $\eta > \zeta_1/4$. Thus, under this condition we have

$$\dot{V} \leqslant -G_{\text{out, min}}Z^2 + F_{\text{out}}Z, \tag{C.11}$$

where $Z = \|(\|\mathbf{z}\|, \|\tilde{\mathbf{w}}\|, \|\tilde{\mathbf{p}}\|)\|$, $G_{\text{out,min}}$ is the minimum eigenvalue of the negative-definite matrix and $F_{\text{out}} = \|(e_{\max}, (2\rho + \eta)\|\mathbf{w}\|)\|$. Thus, a uniform ultimate bound on signals is given by the boundaries of a set in Z where $\dot{V} \leqslant 0$. Specifically, a bound is

$$Z_{\max} = F_{\text{out}}/G_{\text{out, min}}. \tag{C.12}$$

Note that rather than using Z, smaller ultimate bounds on $\mathbf{z}$, $\tilde{\mathbf{w}}$, $\tilde{\mathbf{p}}$ can be found using (C.10) directly.

C.2. Inside the deadzone

When $\|\boldsymbol{\tau}\| < \delta$ Eq. (14) implies alternate weights are bounded, $\|\tilde{\mathbf{p}}\| < p_{\max}$ with $p_{\max}$ a positive constant and thus

$$g \leqslant -\zeta_2\|\tilde{\mathbf{w}}\|^2 + \zeta_2\|\tilde{\mathbf{w}}\|p_{\max} + (2\rho + \eta)p_{\max}\|\mathbf{w}\| - \eta\|\tilde{\mathbf{p}}\|^2. \tag{C.13}$$

The Lyapunov time-derivative becomes

$$\dot{V} \leqslant \begin{bmatrix} \|\mathbf{z}\| \\ \|\tilde{\mathbf{w}}\| \\ \|\tilde{\mathbf{p}}\| \end{bmatrix}^T \begin{bmatrix} -K_{\min} & 0 & 0 \\ 0 & -\zeta_2 & 0 \\ 0 & 0 & -\eta \end{bmatrix} \begin{bmatrix} \|\mathbf{z}\| \\ \|\tilde{\mathbf{w}}\| \\ \|\tilde{\mathbf{p}}\| \end{bmatrix} + [D_{\max} \quad p_{\max} \quad 0] \begin{bmatrix} \|\mathbf{z}\| \\ \|\tilde{\mathbf{w}}\| \\ \|\tilde{\mathbf{p}}\| \end{bmatrix} + (2\rho + \eta)p_{\max}\|\mathbf{w}\|, \tag{C.14}$$

or

$$\dot{V} \leqslant -G_{\text{in, min}}Z^2 + F_{\text{in}}Z + C, \tag{C.15}$$

implying a uniform ultimate bound

$$Z_{\max} = \left(F_{\text{in}} + \sqrt{F_{\text{in}}^2 + 4G_{\text{in, min}}C}\right)/(2G_{\text{in, min}}). \tag{C.16}$$

References

[1] Pounds P, Mahony R, Hynes P, Roberts J. Design of a four-rotor aerial robot. In: Australasian conference on robotics & automation, Auckland, New Zealand; 2002. p. 145–50.

[2] Pounds P, Mahony R, Corke P. Modelling and control of a quad-rotor robot. In: Proceedings of the Australasian conference on robotics & automation, Auckland, New Zealand; 2006.

[3] Hoffmann G, Huang H, Waslander S, Tomlin C. Quadrotor helicopter flight dynamics and control: theory and experiment. In: Proceedings of the AIAA guidance, navigation & control conference, Hilton Head, SC, USA, 2007 [aIAA Paper Number 2007-6461].

[4] Hoffmann G, Waslander S, Tomlin C. Quadrotor helicopter trajectory tracking control. In: Proceedings of the AIAA guidance, navigation & control conference, Honolulu, HI, USA; 2008 [aIAA Paper Number 2008-7410].

[5] Bouabdallah S, Siegwart R. Field and service robotics. Springer tracts in advanced robotics. Berlin/Heidelberg: Springer; 2006. p. 429–40 [chapter: Towards intelligent miniature flying robots].

[6] Altug E, Taylor C. Vision-based pose estimation and control of a model helicopter. In: Proceedings of the 2004 IEEE international conference on mechatronics, Istanbul, Turkey; 2004.

[7] Benallegue A, Mokhtari A, Fridman L. Feedback linearization and high order sliding mode observer for a quadrotor UAV. In: Proceedings of the 2006 international workshop on variable structure systems, Alghero, Italy; 2006. p. 365–72.

[8] Madani T, Benallegue A. Backstepping control for a quadrotor helicopter. In: Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots & systems, Beijing, China; 2006. p. 3255–60.

[9] Madani T, Benallegue A. Backstepping sliding mode control applied to a miniature quadrotor flying robot. In: Proceedings of the 2006 IEEE conference on industrial electronics, Paris, France; 2006. p. 700–5.

[10] Madani T, Benallegue A. Control of a quadrotor mini-helicopter via full state backstepping technique. In: Proceedings of the 2006 IEEE conference on decision & control, San Diego, CA, USA; 2006. p. 1515–20.

[11] Lara D, Sanchez A, Lozano R, Castillo P. Real-time embedded control system for VTOL aircrafts: application to stabilize a quad-rotor helicopter. In: Proceedings of the 2006 IEEE international conference on control applications, Munich, Germany; 2006.

[12] Castillo P, Albertos P, Garcia P, Lozano R. Simple real-time attitude stabilization of a quad-rotor aircraft with bounded signals. In: Proceedings of the 45th IEEE conference on decision & control, San Diego, CA, USA; 2006.

[13] Dunfied J, Tarbouchi M, Labonte G. Neural network based control of a four rotor helicopter. In: Proceedings of the 2004 IEEE international conference on industrial technology, Hammamet, Tunisia; 2004. p. 1543–8.

[14] Cesareo Raimundez J, Fernandez Villaverde A. Adaptive tracking control for a quad-rotor. In: Proceedings of ENOC-2008, Saint Petersburg, Russia; 2008.

[15] Dierks T, Jagannathan S. Neural network output feedback control of a quadrotor UAV. In: Proceedings of the 2008 IEEE conference on decision and control, Cancun, Mexico; 2008. p. 3633–9.

[16] Nicol C, Macnab C, Ramirez-Serrano A. Robust neural network control of a quadrotor helicopter. In: Proceedings of the 2008 IEEE Canadian conference on electrical and computer engineering, Niagara Falls, ON, Canada; 2008. p. 1233–8.

[17] Albus J. Brains, behaviour, and robotics. BYTE Publications Inc.; 1981.

[18] Yuan Y, Gu W, Yu J. Advances in neural networks – ISNN 2004. Berlin/Heidelberg: Springer; 2004. p. 117–22 [chapter: Online learning CMAC neural network control scheme for nonlinear systems]
[19] Egardt B. Stability of adaptive controllers. New York: Springer-Verlag; 1979.
[20] Narendra K, Annaswamy A. A new adaptive law for robust adaptation without persistent excitation. IEEE Trans Automat Control 1989;AC-32(2):134–45.
[21] Leishman JG. Principles of helicopter aerodynamics. 2nd ed. Cambridge University Press; 2006.
[22] Lozano PCR, Dzul A. Stabilization of a mini rotorcraft with four rotors. IEEE Control Syst Mag 2005:45–55.
[23] Bouabdallah S, Siegwart R. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In: IEEE international conference on robotics and automation, Barcelona, Spain; 2005. p. 2259–64.
[24] Morel Y, Leonessa A. Direct adaptive tracking control of quadrotor aerial vehicles. In: Florida conference on recent advances in robotics, Miami, Florida; 2006. p. 1–6.
[25] Altug JPOE, Mahony R. Control of a quadrotor helicopter using visual feedback. In: IEEE international conference on robotics and automation, Washington, DC; 2002. p. 72–7.
[26] Chen M, Huzmezan M. A simulation model and H_{∞} loop shaping control of a quad rotor unmanned air vehicle. In: Proc of IASTED international conference on modeling and simulation, Palm Springs, CA; 2003.
[27] Hughes PC. Spacecraft attitude dynamics. New York: Wiley; 1986.
[28] Albus. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). J Dyn Syst Meas Control 1975;97:220–7.
[29] Albus J. Data storage in the cerebellar model articulation controller (CMAC). J Dyn Syst Meas Control 1975;97:228–33.