J Intell Robot Syst (2009) 56:127–151
DOI 10.1007/s10846-009-9331-0

Backstepping Approach for Controlling a Quadrotor Using Lagrange Form Dynamics

Abhijit Das · Frank Lewis · Kamesh Subbarao

Received: 13 April 2008 / Accepted: 30 March 2009 / Published online: 21 April 2009

Abstract The dynamics of a quadrotor are a simplified form of helicopter dynamics that exhibit the same basic problems of underactuation, strong coupling, multi-input/multi-output design, and unknown nonlinearities. Control design for the quadrotor is more tractable yet reveals corresponding approaches for helicopter and UAV control design. In this paper, a backstepping approach is used for quadrotor controller design. In contrast to most other approaches, we apply backstepping on the Lagrangian form of the dynamics, not the state space form. This is complicated by the fact that the Lagrangian form for the position dynamics is bilinear in the controls. We confront this problem by using an inverse kinematics solution akin to that used in robotics. In addition, two neural nets are introduced to estimate the aerodynamic components, one for aerodynamic forces and one for aerodynamic moments. The result is a controller of intuitively appealing structure having an outer kinematics loop for position control and an inner dynamics loop for attitude control. The control approach described in this paper is robust since it explicitly deals with unmodeled state-dependent disturbances and forces without needing any prior knowledge of the same. A simulation study validates the results obtained in the paper.

A. Das (✉) · F. Lewis
Automation and Robotics Research Institute,
University of Texas at Arlington,
7300 Jack Newell Blvd. S., Fort Worth, TX 76118, USA
e-mail: adas@arri.uta.edu

F. Lewis
e-mail: lewis@uta.edu

K. Subbarao
Department of Mechanical and Aerospace Engineering,
University of Texas at Arlington,
Arlington, TX 76019, USA
e-mail: subbarao@uta.edu

Keywords Back stepping · Quadrotor · Neural network · Adaptive control · Kinematic inversion · Lyapunov candidate

Nomenclature

i	subscript or superscript index
m	mass of quadrotor
Σ	inertia matrix
x, y, z	position of quadrotor in inertial frame
φ	roll angle
θ	pitch angle
ψ	yaw angle
q	generalized state vector representation
ξ	position vector
η	attitude vector
ξ_d	desired position vector
η_d	desired attitude vector
g	gravity
f_i	force from the *i*'th motor
u	summation of four motor forces
ω_i	rotational speed of *i*'th motor
l	distance from motor to center of gravity
c	force-to-moment scaling factor
τ	torque vector
C	coriolis component
A_{xyz}	state dependent disturbances including aerodynamic nonlinearities
A_η	state dependent disturbances including aerodynamic nonlinearities
e_1, e_2	tracking errors
r_1, r_2	sliding mode errors
$\mu_i(\bullet)$	basis function for NN approximation
W_i	ideal NN weights
F_d	force input to the position subsystem
Λ_i	positive definite design matrix
F_i	positive definite design matrix
K_v	controller gain matrix
ε_v	NN tuning error vector
κ	design constant
$J(\eta)$	auxiliary inertia matrix
χ_1, χ_2	neural net inputs
r_1, r_2	sliding mode errors
$\mu_i(\bullet)$	basis function for NN approximation
K_i, K_r	diagonal gain matrices
$u_d\ \theta_d\ \varphi_d$	desired u, θ, φ respectively
f_{x_d,y_d,z_d}	elements of $\hat{F}_d\ (t)$
a, b	intermediate variables
q_d	desired trajectory
q_B	bounds for desired trajectory
W_B	bounds for NN weights
J_M	bound for auxiliary matrix

L	Lyapunov function candidate
t_f	final time
$x_{d_0}, y_{d_0}, z_{d_0}$	desired initial position
$x_{d_f}, y_{d_f}, z_{d_f}$	desired final position

1 Introduction

Nowadays helicopters are designed to operate with high agility and rapid maneuvering, and are capable of work in degraded environments including wind gusts etc. Helicopter control often requires holding at a particular trimmed state, generally hover, as well as making changes of velocity and acceleration in a desired way [1]. The control of unmanned rotorcraft is becoming more and more important due to their usefulness in rescue, surveillance, inspection, mapping etc. For these applications the ability of the rotorcraft to maneuver sharply and hover precisely is important.

Like fixed-wing aircraft control, rotorcraft control is involved in controlling attitude—roll, pitch, and yaw—and position—x, y, z—either separately or in a coupled way. But the main difference is that, due to the unique body structure of a rotorcraft, as well as the rotor dynamics, the rotorcraft attitude dynamics and position dynamics are strongly coupled. Therefore, it is very difficult to design a decoupled control law of good structure that stabilizes the faster and slower dynamics simultaneously. On the contrary, for a fixed-wing aircraft it is known how to design decoupled standard controllers [2] with intuitively comprehensible structure and performance. Flight controllers of good structure are needed for robustness, as well as to give some intuitive feel for the functioning of autopilots, Stability Augmentation System (SAS), and Control Augmentation System (CAS).

The dynamics of a quadrotor [3–7] are a simplified form of helicopter dynamics that exhibit the basic problems of rotorcraft including underactuation, strong coupling, multi-input/multi-output design, and unknown nonlinearities. In the quadrotor the movement is caused by the resultant forces and moments of four independent rotors. Control design for a quadrotor is quite similar to a helicopter but simplified, so that the quadrotor serves as a suitable, more tractable, case study for rotorcraft controls design. Therefore, numerous papers have dealt with control of the quadrotor, with many approaches available in that literature that use different control techniques. Popular methods include linearization and linear quadratic regulator (LQR) or H-infinity state-space design, model-predictive control [8] feedback linearization, and backstepping [9].

Specifically, [3, 10, 11] considered the Lagrangian model of a quadrotor and derived what was in effect a backstepping controller with three loops: heading/z motion, x motion, and y motion. Nested saturation control was used in each loop. Exact compensation was assumed in each loop. The control law was tested on a practical quadrotor model. References [4, 12] compared the performance of a proportional-integral-derivative (PID) controller with a LQ control law on a micro vertical take-off and landing (VTOL) test bench. These papers presented a good accurate dynamic model, and discussed the mechanical structure and sensors of the quadrotor. Reference [13] discussed the robust trajectory tracking of a Lagrangian quadrotor model using backstepping control and proved the stability for any given but bounded trajectory. A nonlinear H_∞ type robust control law was used in [7, 14], with [14] demonstrating that the robust hover controller can handle large parameter uncertainty with respect to their nominal values.

A popular method for handling unknown nonlinearities is to introduce neural networks, tuned online using adaptive control techniques. Reference [15] used an adaptive dynamic programming (ADP) based controller. Reference [16] used dynamic inversion with neural networks to account for unknown dynamics and dynamic inversion errors for a helicopter model. Reference [17] used an innovative method known as pseudo-control hedging (PCH) with a single neural network employed for both inner and outer loops to account for unknown dynamics and bounded control inputs. Reference [18] used backstepping with neural networks for an unmanned air vehicle to confront modeling errors and unmodeled dynamics. It was based on a nonlinear state space model of an aircraft. Reference [19] presented a Lagrangian dynamic model of X4-flyer with motor dynamics and gyroscopic effects included. References [20, 21] used a feedback linearization approach. Reference [22] presented different control architectures such as feedback linearization, backstepping based on visual feedback as the primary sensor. Reference [23] discussed the dynamic modeling of a unmanned aerial vehicle (UAV) Dragan flyer III and a H_∞ loop shaping flight controller as a position control law. Reference [24] presented a quasioptimal trajectory planner with a LQR path following controller. Reference [25] developed a simulation tool for an indoor flight-capable quadrotor based on MEMS sensors and rangefinders.

Most work in quadrotor control that uses backstepping design or other nonlinear control formulations like feedback linearization are based on state-variable formulations. This results in control laws that are complex and often require the evaluation of lie derivatives. Reference [26] does perform backstepping on the Euler–Lagrange dynamics, but backstepping is performed on a virtual input that is a bilinear product involving thrust input. This results in complications and extra states are added, which we avoid.

The contribution of this paper is to apply backstepping to the coupled Lagrangian form (not the state-variable form) of the dynamics; this yields a controller of appealing structure having an attitude control inner loop and a position control outer loop. It is not straightforward to apply backstepping to the Lagrangian dynamics since the position subsystem is bilinear in the virtual control (see [26]). This is confronted herein using an inverse kinematics solution such as is used in robotics [27]. This allows us to simply take the attitudes directly as the virtual inputs. To combat unknown nonlinearities we use neural networks [28, 29] to estimate unknown nonlinear terms and aerodynamic forces and moments. The resulting controller has a structure that is intuitively appealing, as in fixed-wing aircraft SAS and CAS. Moreover, rigorous stability proofs provided herein give increased confidence in this structure. Simulation results are shown to validate the control law discussed in this paper.

2 Quadrotor Dynamics

In this subsection we will describe the dynamical model of the quadrotor. See Fig. 1. The quadrotor has some basic advantages over the conventional helicopter [11] in terms of simplicity of dynamics and control design. Given that the front and the rear motors rotate counter-clockwise while the other two rotate clockwise, gyroscopic

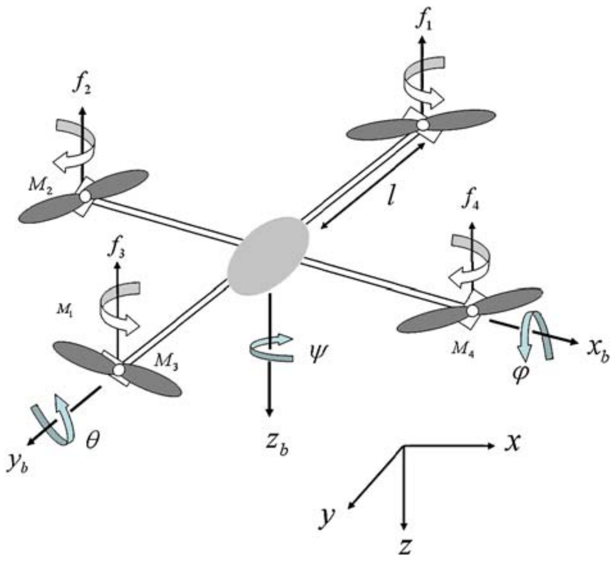

Fig. 1 Model of a typical quadrotor

effects and aerodynamic torques tend to cancel in trimmed flight. This four-rotor rotorcraft does not have a swashplate [3]. In fact it does not need any blade pitch control. The collective input (or throttle input) is the sum of the thrusts of the individual motors (see Fig. 1). This results in simplified dynamics. Pitch movement is obtained by increasing (resp. reducing) the speed of the rear motor while reducing (resp. increasing) the speed of the front motor. The roll movement is obtained similarly using the lateral motors. The yaw movement is obtained by increasing (resp. decreasing) the speed of the front and rear motors while decreasing (resp. increasing) the speed of the lateral motors [11]. The dynamics of the four rotors are relatively much faster than the main system and thus neglected.

The generalized coordinates of the quadrotor are $q = (x, y, z, \psi, \theta, \varphi)^T$ where (x, y, z) represents the relative position of the center of mass of the quadrotor with respect to an inertial frame and (ψ, θ, φ) are the three Euler angles representing the orientation of the rotorcraft, namely yaw, pitch, roll. Define the translational and rotational variables respectively as $\xi = (x, y, z)^T \in R^3$and $\eta = (\psi, \theta, \varphi)^{\mathrm{T}} \in R^3$.

The dynamic model of the rotorcraft can be written [11] as the position subsystem

$$m\ddot{\xi} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = F_d \tag{1}$$

and the rotation subsystem

$$J(\eta)\,\ddot{\eta} + \frac{d}{dt}\{J(\eta)\}\,\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial\eta}\left(\dot{\eta}^T J(\eta)\,\dot{\eta}\right) = \tau \tag{2}$$

or, by appropriate definition of variables,

$$J(\eta)\,\ddot{\eta} + C\left(\eta, \dot{\eta}\right)\dot{\eta} = \tau \tag{3}$$

The auxiliary (effective) inertia matrix is defined as,

$$J(\eta) = J = T_{\eta}^{T} \Sigma T_{\eta} \tag{4}$$

with $T_{\eta} = \begin{pmatrix} -\sin\theta & 0 & 1 \\ \cos\theta \sin\psi & \cos\psi & 0 \\ \cos\theta \cos\psi & -\sin\psi & 0 \end{pmatrix}$ a rotation transformation and Σ the (constant) quadrotor inertia matrix.

The virtual input (e.g. an input that we cannot directly control) into the position subsystem (1) is the force vector $F_d = u \begin{pmatrix} -\sin\theta \\ \cos\theta \sin\varphi \\ \cos\theta \cos\varphi \end{pmatrix}$. The force exerted by the ith rotor is denoted as f_i and the control inputs are defined as the total thrust $u = \sum_{i=1}^{4} f_i$, and the torques $\tau = \begin{pmatrix} \tau_{\psi} \\ \tau_{\theta} \\ \tau_{\varphi} \end{pmatrix}$. The control inputs $\left(u, \tau^{T}\right)^{T}$ are given in terms of the forces $\left(f_i = k_{\omega_i} \omega_i^2, i = 1 \ldots 4\right)$ of the four rotors of the quadrotor by the relation

$$\begin{pmatrix} u \\ \tau_{\varphi} \\ \tau_{\theta} \\ \tau_{\psi} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ l & 0 & -l & 0 \\ 0 & -l & 0 & l \\ c & -c & c & -c \end{pmatrix}}_{M} \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}}_{f} = Mf \tag{5}$$

where l is the distance from the motors to the center of gravity, and c is a constant known as force-to-moment scaling factor. Thus, if a required thrust and torque vector are given, one may solve for the rotor forces via matrix inversion using Eq. 5. Therefore, we show in the subsequent development how to design the control inputs $\left(u, \tau^{T}\right)^{T}$.

Vector $C\left(\eta, \dot{\eta}\right)\dot{\eta}$ is the coriolis/centripetal term. From Eq. 2, the matrix $C\left(\eta, \dot{\eta}\right)$ can be represented, assuming the inertia matrix Σ for a typical quadrotor is symmetric, as

$$\begin{aligned} C\left(\eta, \dot{\eta}\right) &= \frac{d}{dt} J - \frac{1}{2} \frac{\partial}{\partial \eta} \left(\dot{\eta}^{T} J(\eta) \right) \\ &= 2 T_{\eta}^{T} \Sigma \underset{\eta}{\dot{T}} - \frac{1}{2} \frac{\partial}{\partial \eta} \left(\dot{\eta}^{T} J(\eta) \right) \end{aligned} \tag{6}$$

Note that the representation of matrix $C\left(\eta, \dot{\eta}\right)$ is not unique [27]. However, using the specific representation (Eq. 6), it can be proved that $\left(\frac{d}{dt} J(\eta) - 2C\left(\eta, \dot{\eta}\right)\right)$ is skew-symmetric. This is needed later in Lyapunov proofs referred to the dynamics (Eq. 3). Note that for a quadrotor the inertia matrix Σ is constant but the effective inertia $J = J(\eta)$ is not a constant.

In summary, the coupled Lagrangian form of the dynamics is given as

$$m\ddot{\xi} = u \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\varphi \\ \cos\theta\cos\varphi \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \tag{7}$$

$$J\ddot{\eta} = -C\left(\eta, \dot{\eta}\right)\dot{\eta} + \tau \tag{8}$$

where total thrust $u \in R$ and torque $\tau \in R^3$ are the control inputs. This system is a coupled Lagrangian form underactuated system with six outputs and four inputs.

3 Backstepping Control Design

Backstepping control design is effective for a class of underactuated systems [9, 30]. Many robot and aircraft control problems are included in the class of underactuated systems [27]. Backstepping for the quadrotor has normally been applied to the state-variable form, which introduces unnecessary complications (e.g. lie derivatives). Note that the state of the quadrotor in Eqs. 7 and 8 is $\left(\xi, \dot{\xi}, \eta, \dot{\eta}\right)$, which has 12 components. In this paper we use backstepping control design directly on the Lagrangian quadrotor dynamics. This is complicated by the fact that Eq. 7 is bilinear in the controls due to the first term on the right-hand side (i.e. F_d). Taking this entire bilinear term as a virtual control leads to complications (see [26]).

The quadrotor dynamics have a simplified form due to its physical construction and the decoupled control effects inherent in the four motor design. However, for aggressive maneuverability, we would like to include aerodynamic forces and moments in the quadrotor equations. These are unknown forces containing nonlinear terms in the states that appear in any object that moves through the air. Note that the quadrotor has an equivalent area cross section in each of the x, y and z directions. These areas are difficult to determine, are state dependent, and can usually be found only through wind tunnel tests. The aerodynamic forces are of two types: lift force due to dynamic pressure and drag force due to viscosity effects. They can be separated into components in each of the x, y, z directions. Likewise, the aerodynamic torques have three components as torques around the three axes.

Therefore, rewrite the quadrotor dynamics Eqs. 7 and 8 to include unmodeled state-dependent disturbances that include aerodynamic forces and moments A_x, A_y, A_z, A_η. This yields the Lagrangian form

$$\begin{aligned} m\ddot{x} &= -u\sin\theta + A_x \\ m\ddot{y} &= u\cos\theta\sin\varphi + A_y \\ m\ddot{z} &= u\cos\theta\cos\varphi - mg + A_z \end{aligned} \tag{9}$$

$$J\begin{pmatrix} \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\varphi} \end{pmatrix} = \begin{pmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\varphi \end{pmatrix} - C\left(\eta, \dot{\eta}\right)\dot{\eta} + A_\eta \tag{10}$$

Now we will use backstepping to determine a control structure to make the position $\xi = (x, y, z)^T$ of the quadrotor follow a desired smooth (e.g. at least twice differentiable) trajectory $(x_d(t), y_d(t), z_d(t))$. The end result is shown in Fig. 2.

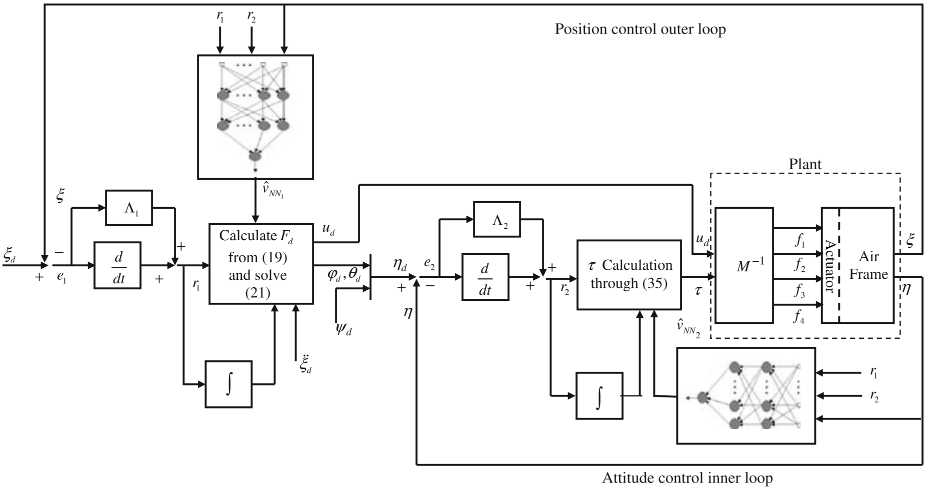

Fig. 2 Control configuration

To achieve this purpose, the first step of backstepping focuses on the position dynamics (Eq. 9). Define the position tracking error $e_1(t)$ of the subsystem given by Eq. 9 and the sliding mode error $r_1(t)$ as [28]

$$e_1 = (x_d - x, y_d - y, z_d - z)^T = \xi_d - \xi \tag{11}$$

$$r_1 = \dot{e}_1 + \Lambda_1 e_1 \tag{12}$$

where, Λ_1 is a diagonal positive definite design parameter matrix. Common usage is to select Λ_1 diagonal with positive entries. Then, Eq. 12 is a stable system so that e_1 is bounded as long as the controller guarantees that the filtered error r_1 is bounded. In fact it is easy to show [28] that one has

$$\|e_1\| \le \frac{\|r_1\|}{\sigma_{\min}(\Lambda_1)}, \left\|\dot{e}_1\right\| \le \|r_1\| \tag{13}$$

Note that $\dot{e}_1 + \Lambda_1 e_1 = 0$ defines a stable sliding mode surface. The function of the controller to be designed is to force the system onto this surface by making r_1 small. The parameter Λ_1 is selected for a desired sliding mode response

$$e_1(t) = e_1^{-\Lambda_1 t} e_1(0) \tag{14}$$

We now focus on designing a controller to keep $\|r_1\|$ small. Then the error dynamics becomes

$$m\dot{r}_1 = m\ddot{e}_1 + m\Lambda_1 \dot{e}_1 \tag{15}$$

$$\Rightarrow m\dot{r}_1 = m\begin{bmatrix} \ddot{x}_d \\ \ddot{y}_d \\ \ddot{z}_d \end{bmatrix} - m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + m\Lambda_1(r_1 - \Lambda_1 e_1) \tag{16}$$

$$m\dot{r}_1 = m\ddot{\xi}_d - \begin{pmatrix} -u\sin\theta \\ u\cos\theta\sin\varphi \\ u\cos\theta\cos\varphi \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} - \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} + m\Lambda_1\left(r_1 - \Lambda_1 e_1\right) \tag{17}$$

or,

$$m\dot{r}_1 = m\ddot{\xi}_d + m\Lambda_1 r_1 - m\Lambda_1^2 e_1 - F_d - m_g - A_{xyz} \tag{18}$$

where, $F_d = \begin{pmatrix} -u\sin\theta \\ u\cos\theta\sin\varphi \\ u\cos\theta\cos\varphi \end{pmatrix}$, $A_{xyz} = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix}$ and $m_g = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix}$.

The ideal force (virtual force) which stabilizes the tracking error dynamics (Eq. 18) is

$$\hat{F}_d = m\ddot{\xi}_d - m\Lambda_1^2 e_1 - m_g - \hat{A}_{xyz} + K_{r_1} r_1 + K_{i_1}\int_0^t r_1 dt' \tag{19}$$

where, A_{xyz} is an approximation (to be detailed later) of the aerodynamic function A_{xyz} and control gain matrices $K_{r_1} > 0$, $K_{i_1} > 0$ are diagonal.. Using Eq. 19 in Eq. 18 yields the closed-loop error dynamics

$$m\dot{r}_1 = -\left(K_{r_1} - m\Lambda_1\right) r_1 - \tilde{A}_{xyz} - K_{i_1}\int_0^t r_1 dt' \tag{20}$$

where, $\tilde{A}_{xyz} = A_{xyz} - \hat{A}_{xyz}$is the approximate error. We shall show in subsequent proofs the beneficial properties of this error dynamics system.

Now it is necessary to perform step 2 of the backstepping procedure, which selects the control inputs, u and torque τ, to generate the desired force (Eq. 19). The simplicity of the Lagrangian form must be preserved in an effective backstepping procedure. This means one must determine the desired values of u and of (ψ, θ, φ) for the second backstepping step that involves the attitude dynamics (Eq. 10). Note that yaw ψ does not appear in Eq. 9, so its desired value ψ_d can be selected as a reference input. Call the remaining desired values φ_d, θ_d and u_d. To generate the ideal force $\hat{F}_d$ in Eq. 19 would require the use of φ_d, θ_d and u_d given by the solution to

$$\begin{pmatrix} -u_d\sin\theta_d \\ u_d\cos\theta_d\sin\varphi_d \\ u_d\cos\theta_d\cos\varphi_d \end{pmatrix} = \hat{F}_d \tag{21}$$

Thus, given $\hat{F}_d$ as computed from Eq. 19, one must solve for φ_d, θ_d and u_d. This is straightforward using robot inverse kinematics notions. The procedure is given in Section 4.

Now a NN is used to approximate A_{xyz} which is unknown. It can be shown from [31] that A_{xyz} can be approximated using a NN as

$$\upsilon_{\mathrm{NN}_1} = -A_{xyz} \equiv W_1^T \mu_1\left(\chi_1\right) + \varepsilon_1 \tag{22}$$

where W_1 is an unknown ideal NN output weight matrix and $\chi_1 = \left(\xi, r_1, \eta, \dot{\eta}\right)^T$ is the input of the NN. The NN activation function vector $\mu_1(\cdot)$ must be selected as a basis set for the unknown function, which is required for suitable NN approximation. The NN activation function $\mu_1(\cdot)$ should be chosen to capture any known nonlinearity present in the aerodynamics. The NN estimation error ε_1 is bounded by $\|\varepsilon_1\| < \varepsilon_{N_1}$ on a compact set, with a known bound ε_{N_1}.

Note that one could use a two-layer neural net for approximation. This is nonlinear in the weight parameters (NLIP), and introduces extra complications in the proof, which are nonetheless tractable. This technology has been described in [28, 32]. In order to focus on the contributions of this paper without extra complications, we use the linear-in-the-parameter NN here. Therefore, an approximation to the unknown aerodynamic forces is taken as

$$\hat{\upsilon}_{\mathrm{NN}_1} = -\hat{A}_{xyz} = \hat{W}_1^T \mu_1(\chi_1) \tag{23}$$

with $\hat{W}_1$ the actual NN weights. In Section 5 we will discuss how to tune the NN weights $\hat{W}_1$ to provide stability and bounded sliding mode error dynamics (20).

The desired force input into the position dynamics is thus taken as

$$\hat{F}_d = m\ddot{\xi}_d - m\Lambda_1^2 e_1 - m_g + \hat{W}_1^T \mu_1(\chi_1) + K_{r_1} r_1 + K_{i_1}\int_0^t r_1 dt' \tag{24}$$

Define,

$$-\tilde{A}_{xyz} = W_1^T \mu_1(\chi_1) - \hat{W}_1^T \mu_1(\chi_1) + \varepsilon_1 \equiv \tilde{W}_1^T \mu_1(\chi_1) + \varepsilon_1 \tag{25}$$

Now Eq. 20 becomes,

$$m\underset{1}{\dot{r}} = -\left(K_{r_1} - m\Lambda_1\right) r_1 + \tilde{W}_1^T \mu_1(\chi_1) - K_{i_1}\int_0^t r_1 dt' + \varepsilon_1 \tag{26}$$

By solving Eq. 21 with $\hat{F}_d$ computed using Eq. 24, as detailed in Section 4, the desired pseudo-commands φ_d, θ_d, u_d for the subsystem given by Eq. 9 are determined. Note that $u_d$is directly applied to the plant as input. Now, the second step of backstepping is performed. We select a control input τ for Eq. 10 to generate φ_d, θ_d. Note that ψ_d does not appear in Eq. 9 and so can be selected as an independent desired reference input. Define $\eta_d = (\varphi_d, \theta_d, \psi_d)^T$

Defining the tracking error $e_2(t)$ and the sliding mode error $r_2(t)$ as,

$$e_2 = (\psi_d - \psi, \theta_d - \theta, \varphi_d - \varphi)^T \tag{27}$$

$$r_2 = \dot{e}_2 + {}_2e_2 \tag{28}$$

where, Λ_2 is a diagonal positive definite design parameter matrix with similar characteristic of Λ_1. Then designing a controller to keep $\|r_2\|$ small will guarantee that $\|e_2\|$ and $\left\|\dot{e}_2\right\|$ are small.

Then error dynamics given by

$$\begin{aligned} J\underset{2}{\dot{r}} &= J\,\ddot{e}_2 + J\Lambda_2\,\dot{e}_2 \\ &= J\ddot{\eta}_d - \left\{\tau - C\left(\eta, \dot{\eta}\right)\dot{\eta} + A_\eta\right\} + J\Lambda_2\left(r_2 - \Lambda_2 e_2\right) \end{aligned} \tag{29}$$

and from Eq. 28

$$\dot{\eta} = -r_2 + \underset{d}{\dot{\eta}} + \Lambda_2 e_2 \tag{30}$$

Substituting $\dot{\eta}$ from Eq. 30 in Eq. 29 yields

$$J\underset{2}{\dot{r}} = J\ddot{\eta}_d - \left\{\tau - C\left(\eta, \dot{\eta}\right)\left(-r_2 + \underset{d}{\dot{\eta}} + \Lambda_2 e_2\right) + A_\eta\right\} + J\Lambda_2\left(r_2 - \Lambda_2 e_2\right) \tag{31}$$

$$J\underset{2}{\dot{r}} = -\tau - C\left(\eta, \dot{\eta}\right) r_2 + \left\{J\ddot{\eta}_d - A_\eta + C\left(\eta, \dot{\eta}\right)\left(\underset{d}{\dot{\eta}} + \Lambda_2 e_2\right) - J\Lambda_2^2 e_2\right\} + J\Lambda_2 r_2 \tag{32}$$

Now, a second NN is introduced to approximate $\left\{J\ddot{\eta}_d - A_\eta + C\left(\eta, \dot{\eta}\right)\left(\underset{d}{\dot{\eta}} + \Lambda_2 e_2\right) - J\Lambda_2^2 e_2\right\}$. According to [31], this can be approximated as

$$\upsilon_{\mathrm{NN}_2} \triangleq \left\{J\ddot{\eta}_d - A_\eta + C\left(\eta, \dot{\eta}\right)\left(\underset{d}{\dot{\eta}} + \Lambda_2 e_2\right) - J\Lambda_2^2 e_2\right\} \equiv W_2^T \mu_2\left(\chi_2\right) + \varepsilon_2 \tag{33}$$

where W_2 is the unknown ideal NN weight matrix, $\mu_2(\cdot)$ is an another basis activation function which is to be chosen to estimate the known nonlinearities present in the system. e_2 is estimation error which is bounded by $\|\varepsilon_2\| \leq \varepsilon_{N_2}$ on a compact set with the input to the NN is selected as $\chi_2 = (\eta, r_1, r_2)^T$ and thus one can select

$$\hat{\upsilon}_{\mathrm{NN}_2} \equiv \hat{W}_2^T \mu_2\left(\chi_2\right) \tag{34}$$

with $\hat{W}_2$ is the actual NN weights.

Define the control input τas

$$\tau = \hat{\upsilon}_{\mathrm{NN}_2} + K_{r_2} r_2 + K_{i_2} \int_0^t r_2 dt' \tag{35}$$

with $K_{r_2} > 0$, $K_{i_2} > 0$ are diagonal gain matrices and also

$$\tilde{\upsilon}_{\mathrm{NN2}} = \upsilon_{\mathrm{NN2}} - \hat{\upsilon}_{\mathrm{NN2}} = W_2^T \mu_2\left(\chi_2\right) - \hat{W}_2^T \mu_2\left(\chi_2\right) + \varepsilon_2 = \tilde{W}_2^T \mu_2\left(\chi_2\right) + \varepsilon_2 \tag{36}$$

Then Eq. 32 reduces to the form,

$$J\dot{r}_2 = \tilde{\upsilon}_{NN_2} + J\Lambda_2 r_2 - C\left(\eta, \dot{\eta}\right) r_2 - K_{r_2} r_2 - K_{i_2} \int_0^t r_2 dt' \tag{37}$$

$$J\dot{r}_2 = \tilde{W}_2^T \mu_2\left(\chi_2\right) - \left\{K_{r_2} + C\left(\eta, \dot{\eta}\right) - J\Lambda_2\right\} r_2 - K_{i_2} \int_0^t r_2 dt' + \varepsilon_2 \tag{38}$$

with the estimation error

$$\tilde{\upsilon}_{\mathrm{NN}_2} = \upsilon_{\mathrm{NN}_2} - \hat{\upsilon}_{\mathrm{NN}_2} = W_2^T \mu_2\left(\chi_2\right) - \hat{W}_2^T \mu_2\left(\chi_2\right) + \varepsilon_2 = \tilde{W}_2^T \mu_2\left(\chi_2\right) + \varepsilon_2 \tag{39}$$

4 Solution and Feasibility of $\hat{F}_d$

Here, we show how to confront the bilinearity in the control which appears in the position dynamics given by Eq. 9. We derive the final control law based on the assumption that $\hat{F}_d$ is feasible, that is, it is possible to find a solution for u_d, φ_d and θ_d in Eq. 21. In fact, given any $\hat{F}_d$, the solution of Eq. 21 is easy using robot inverse kinematics [27] and is given here. Inverse Kinematics is one of the popular methods in robotics used for finding the joint variables given a desired Cartesian position [33].

For the problem discussed above, one must find angular positions $(u_d, \varphi_d, \theta_d)$ for a given $\hat{F}_d$ in

$$\begin{pmatrix} -u_d \sin\theta_d \\ u_d \cos\theta_d \sin\varphi_d \\ u_d \cos\theta_d \cos\varphi_d \end{pmatrix} = \hat{F}_d(t) \in R^3 \tag{40}$$

Define

$$a = u_d \sin\theta_d, b = u_d \cos\theta_d \tag{41}$$

Then

$$\begin{pmatrix} -a \\ b \sin\varphi_d \\ b \cos\varphi_d \end{pmatrix} = \hat{F}_d(t) \equiv \begin{pmatrix} f_{x_d}(t) \\ f_{y_d}(t) \\ f_{z_d}(t) \end{pmatrix} \tag{42}$$

so that

$$a = -f_{x_d}(t) \tag{43}$$

and

$$b^2 \sin^2\varphi_d + b^2 \cos^2\varphi_d = f^2_{y_d}(t) + f^2_{z_d}(t) \tag{44}$$

$$\Rightarrow b = \sqrt{f^2_{y_d}(t) + f^2_{z_d}(t)} \tag{45}$$

From Eq. 42,

$$(u_d \sin\theta_d)^2 + (u_d \cos\theta_d)^2 = a^2 + b^2 \tag{46}$$

$$\Rightarrow u_d^2 = a^2 + b^2 \tag{47}$$

$$u_d = \sqrt{a^2 + b^2} \tag{48}$$

and

$$\tan\theta_d = \frac{a}{b} \tag{49}$$

$$\theta_d = \tan^{-1}\left(\frac{a}{b}\right) \tag{50}$$

Finally from Eq. 42

$$\tan \varphi_d = \frac{f_{y_d}}{f_{z_d}} \tag{51}$$

$$\varphi_d = \tan^{-1}\left(\frac{f_{y_d}}{f_{z_d}}\right) \tag{52}$$

This procedure shows that Eq. 21 has a solution $(u_d, \varphi_d, \theta_d)$ for all $\hat{F}_d$. We assume that $f_{z_d} \neq 0$ i.e. φ and ψ are not 90°. Then $(u_d, \varphi_d, \theta_d)$ are given respectively by Eqs. 48, 50 and 52.

5 Stability Analysis and NN Tuning

In Section 3, the control law given by Eq. 35 is derived from desired attitudes φ_d, θ_d and ψ_d where φ_d, θ_d are obtained from the solution of Eq. 21 and given by Eqs. 50 and 52 respectively. The required lift u_d is given by Eq. 48, and desired yaw ψ_d is an external reference command. The control structure is shown in Fig. 2. There, one has an outer kinematics loop with inputs the desired reference positions (x_d, y_d, z_d), and an inner dynamics loop for attitude control to generate the required angles $(\varphi_d, \theta_d, \psi_d)$. Note that, the desired command inputs are $\xi_d = (x_d(t), y_d(t), z_d(t))$ as well as desired yaw $\psi_d(t)$, which are given as reference trajectory inputs.

Based on the Lyapunov theory the feasibility of the control law given by Eq. 35 is discussed in this section, and tuning laws are given for the NN weights to guarantee stability. The related background literature can be found in [28].

Assumptions:

(a) Desired trajectory $q_d = \left(\xi_d^T \;\; \psi_d\right)^T$ is bounded i.e. $\|q_d\| < q_B$for some known bound q_B.

(b) Ideal NN weights W_1, W_2 are bounded, i.e. $\left\| \begin{matrix} W_1 \\ W_2 \end{matrix} \right\| < W_B$ for some known bound W_B.

(c) The auxiliary matrix can be upper bounded as $J(\eta) \leq J_M$ with $J_M \in \Re$ a known positive constant.

Note that assumption (a) always holds. Assumption (b) holds on a compact set, and assumption (c) holds since J is an inertia matrix containing sines and cosines of $[\varphi\theta\psi]^T$ [27].

Now follows the main result given by Theorem 5.1 which proves that r_1 and r_2 are Uniformly Ultimately Bounded (UUB) if the NN are properly tuned. According to the definition given by Eq. 12 of r_1 and Eq. 28 of r_2, this guarantees that e_1 and e_1 are UUB since

$$\begin{aligned} \|e_1\| &\leq \|s + \Lambda_1\|^{-1} \|r_1\| \leq (\sigma_{\min}(\Lambda_1))^{-1} \|r_1\| \\ \|e_2\| &\leq \|s + \Lambda_2\|^{-1} \|r_2\| \leq (\sigma_{\min}(\Lambda_2))^{-1} \|r_2\| \end{aligned} \tag{53}$$

where $\sigma_{\min}(\Lambda_i)$ is the minimum eigenvalue of $\Lambda_i, i = 1, 2$.

Theorem 5.1 *Given the system as described in Eqs. 9 and 10 with a control architecture shown in Fig. 2 and control inputs given by Eqs. 48 and 48. Under the assumptions, suppose that*

$$\left[K_{r_1} - m\Lambda_1\right] > 0, \left[K_{r_2} - J_M\Lambda_2\right] > 0$$

and the NN weight tuning laws are given as

$$\left.\begin{aligned} \dot{\hat{W}}_1 &= F_1\mu_1 r_1^T - \kappa\left\|r\right\| F_1\hat{W}_1 \\ \dot{\hat{W}}_2 &= F_2\mu_2 r_2^T - \kappa\left\|r\right\| F_2\hat{W}_2 \end{aligned}\right\} \tag{54}$$

with proposed desired tuning parameter matrices $F_1 = F_1^T > 0$, $F_2 = F_2^T > 0$. Then the errors $r_1(t)$, $r_2(t)$ and the NN weight estimation errors $\tilde{W}_1(t)$, $\tilde{W}_2(t)$ are UUB with practical bounds given by Eqs. 63 and 65. Moreover the tracking errors $r_{1,}r_2$ can be made arbitrarily small with a suitable choice of design parameters K_{r_1}, K_{r_2}, Λ_1 and Λ_2.

Proof Define,$r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$, $\hat{W} = \begin{pmatrix} \hat{W}_1 \\ \hat{W}_2 \end{pmatrix}$, $\mu_r = \begin{pmatrix} \mu_1 r_1^T \\ \mu_2 r_2^T \end{pmatrix}$ and $F = F^T = \begin{bmatrix} \underline{F}_1 & \underline{0} \\ \underline{0} & \underline{F}_2 \end{bmatrix} > 0, \kappa > 0$

Consider the following Lyapunov function candidate:

$$\begin{aligned} L = {} & \tfrac{1}{2}r_1^T m r_1 + \tfrac{1}{2}r_2^T J r_2 + \tfrac{1}{2}\left[\textstyle\int_0^t r_1^T dt'\right] K_{i_1}\left[\textstyle\int_0^t r_1 dt'\right] \\ & + \tfrac{1}{2}\left[\textstyle\int_0^t r_2^T dt'\right] K_{i_2}\left[\textstyle\int_0^t r_2 dt'\right] + \tfrac{1}{2}tr\left\{\tilde{W}^T F^{-1}\tilde{W}\right\} \end{aligned} \tag{55}$$

Then,

$$\begin{aligned} \dot{L} = {} & -r^T\begin{bmatrix} K_{r_1} - m\Lambda_1 & 0 \\ 0 & K_{r_2} - J\Lambda_2 \end{bmatrix} r + r^T\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \\ & + \frac{1}{2}r_2^T\dot{J}r_2 - r_2^T C r_2 + tr\left\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} + \mu_r\right)\right\} \end{aligned} \tag{56}$$

Now let,

$$\dot{\tilde{W}} = -F\mu_r + \kappa\left\|r\right\| F\hat{W} \tag{57}$$

Continuing with Eq. 55 using Eq. 57,

$$\dot{L} \leq -r^T K_v r + r^T\varepsilon_v + \frac{1}{2}r^T\left(\dot{J} - 2C\right)r + \kappa\left\|r\right\| tr\left\{\tilde{W}^T\hat{W}\right\} \tag{58}$$

where

$$K_v = \begin{bmatrix} K_{r_1} - m\Lambda_1 & 0 \\ 0 & K_{r_2} - J_M\Lambda_2 \end{bmatrix}, \varepsilon_v = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \tag{59}$$

The hypothesis guarantees $K_v > 0$. Therefore

$$\dot{L} \leq -r^T K_v r + r^T \varepsilon_v + \frac{1}{2} r^T \left(\dot{J} - 2C\right) r + \kappa \left\| r \right\| tr\left\{ \tilde{W}^T \left(W - \tilde{W} \right) \right\} \tag{60}$$

where, $\left(\dot{J} - 2C\right)$ is skew symmetric [34] and thus it can be shown that $r^T \left(\dot{J} - 2C\right)$ $r = 0$.

Since, $tr\left\{ \tilde{W}^T \left(W - \tilde{W} \right) \right\} \leq \left\| \tilde{W} \right\|_F \left\| W \right\|_F - \left\| \tilde{W} \right\|_F^2$ with $\left\| \cdot \right\|_F$ the Frobenius norm, there results

$$\begin{aligned} \dot{L} &\leq -\left\| K_v \right\|_{\min} \left\| r \right\|^2 + \kappa \left\| r \right\| \left\{ \left\| \tilde{W} \right\|_F \left\| W \right\|_F - \left\| \tilde{W} \right\|_F^2 \right\} + \left\| \varepsilon_v \right\|_{\max} \left\| r \right\| \\ &= -\left\| r \right\| \left\{ \left\| K_v \right\|_{\min} \left\| r \right\| + \kappa \left\{ \left\| \tilde{W} \right\|_F^2 - \left\| \tilde{W} \right\|_F \left\| W \right\|_F \right\} - \left\| \varepsilon_v \right\|_{\max} \right\} \end{aligned} \tag{61}$$

where, $\left\| K_v \right\|_{\min}$ is the minimum possible norm of for different parameter values of K_v. Similarly, $\left\| \varepsilon_v \right\|_{\max}$ is the maximum possible vector norm of ε_v.

Now Eq. 61 can be rewritten as,

$$\dot{L} \leq -\left\| r \right\| \left\{ \kappa \left[\left\| \tilde{W} \right\|_F - \frac{\left\| W \right\|_F}{2} \right]^2 - \kappa \frac{\left\| W \right\|_F^2}{4} + \left\| K_v \right\|_{\min} \left\| r \right\| - \left\| \varepsilon_v \right\|_{\max} \right\} \tag{62}$$

If $\left\| W \right\|_F \leq W_B$ then, $\dot{L} \leq 0$ if and only if,

$$\left\| r \right\| > \frac{\frac{W_B^2}{4} + \left\| \varepsilon_v \right\|_{\max}}{\left\| K_v \right\|_{\min}} \equiv b_r \tag{63}$$

or from Eq. 61 we can write considering positive definiteness of K_v and

$$\kappa \left\| \tilde{W} \right\|_F \left\{ \left\| \tilde{W} \right\|_F - \left\| W \right\|_F \right\} - \left\| \varepsilon_v \right\|_{\max} > 0 \tag{64}$$

$$\left\| \tilde{W} \right\|_F > \frac{W_B}{2} + \sqrt{\left(\frac{W_B^2}{4} + \frac{\left\| \varepsilon_v \right\|_{\max}}{\kappa} \right)} \equiv b_w \tag{65}$$

Thus, $\dot{L}$ is negative outside a compact set. Selecting the gain K_v ensures that the compact set defined by $\left\| r \right\| \leq b_r$ is bounded, so that the approximation property holds throughout. This ensures UUB for $\left\| r \right\|$ and $\left\| \tilde{W} \right\|_F$ stated in Eqs. 63 and 65 respectively. □

6 Simulation Result

In practice the NN's can be initialized with zero weights. Then, during the initialization, the PID controllers defined by Eqs. 19 and 35 with NN's removed stabilize the closed loop system until the NN weights begin to learn using the tuning laws (Eq. 54). An assumption on the initial tracking error magnitude is technically needed. For details see [28] in page 198.

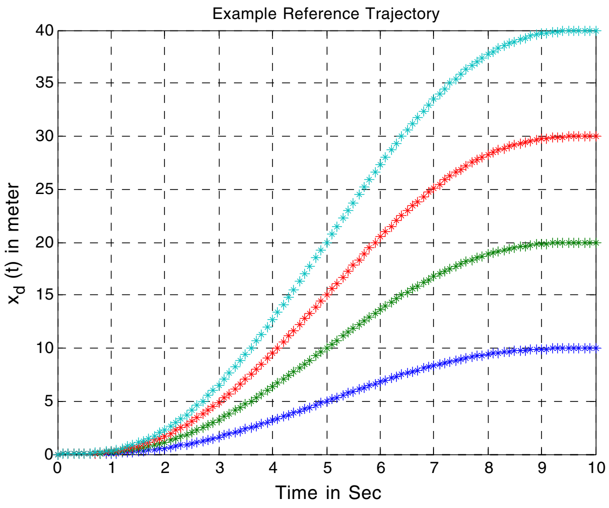

Fig. 3 Example trajectory simulation for different final positions

Rotorcraft model:

Simulation for a typical quadrotor is performed using the following parameters (SI units):

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \Sigma = \mathrm{diag}\left(\Sigma_x, \Sigma_y, \Sigma_z\right) = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}; g = 9.81.$$

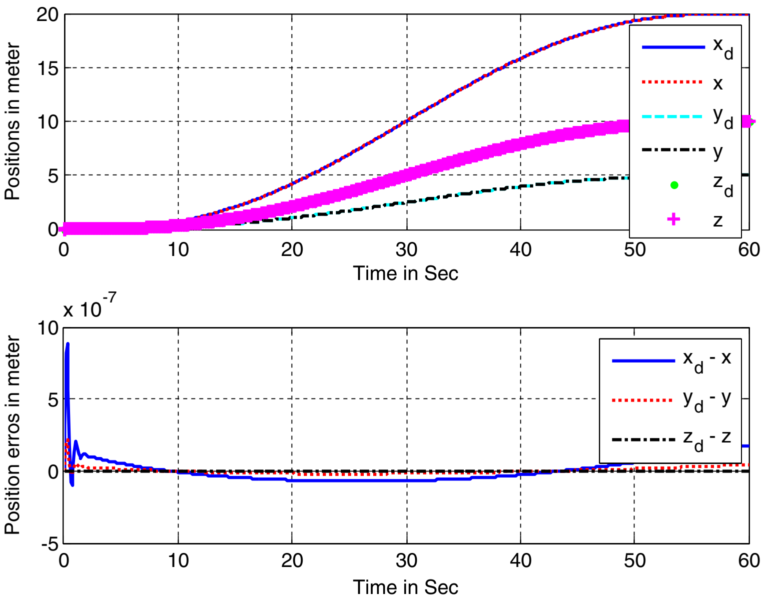

Fig. 4 Three position commands simultaneously

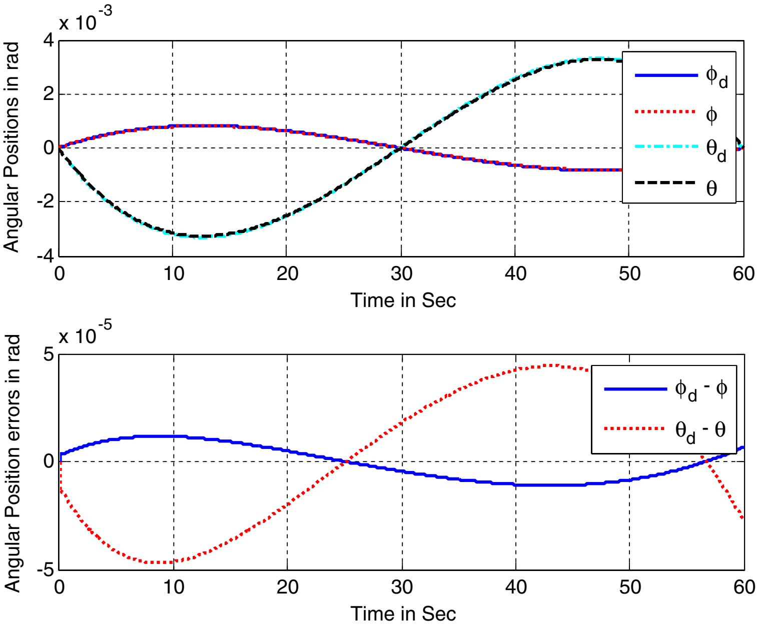

Fig. 5 Resultant angular positions and errors

The centripetal/coriolis components are defined by Eq. 6 with the auxiliary matrix

$$J(\eta) = \begin{pmatrix} \Sigma_x s_\theta^2 + \Sigma_y c_\theta^2 s_\psi^2 + \Sigma_z c_\theta^2 c_\psi^2 & \left(\Sigma_y - \Sigma_z\right) c_\theta s_\psi c_\psi & -\Sigma_x s_\theta \\ \left(\Sigma_y - \Sigma_z\right) c_\theta c_\psi s_\psi & \Sigma_y c_\psi^2 + \Sigma_z s_\psi^2 & 0 \\ -\Sigma_x s_\theta & 0 & \Sigma_x \end{pmatrix} \tag{66}$$

The aerodynamic components are taken as representative functions as follows

$$A_{xyz} = \begin{pmatrix} \dot{x}\,2\,\dot{\varphi} \\ \dot{y}\,2\,\dot{\theta} \\ \dot{z}\,2\,\dot{\psi} \end{pmatrix}, A_\eta = \begin{pmatrix} \dot{x}\,\dot{\varphi}\,2 \\ \dot{y}\,\dot{\theta}\,2 \\ \dot{z}\,\dot{\psi}\,2 \end{pmatrix} \tag{67}$$

The physical interpretation of A_{xyz}, A_η have been elaborated in Section 3. The nonlinear terms introduced here are merely a representation of unmodeled state-dependent disturbances used in this particular study to show the effectiveness of the controller in rejecting these unknown terms. These terms are used in the simulation but are not required to implement the control law. In actual UAV, the aerodynamic forces and moments will be more complicated, but they are nevertheless effectively estimated by the NNs.

Reference trajectory generation:

As outlined in [35, 36], a reference trajectory is derived that goes from a prescribed initial position $\left(x_{d_0}, y_{d_0}, z_{d_0}\right)$ to a prescribed final position $\left(x_{d_f}, y_{d_f}, z_{d_f}\right)$ such that

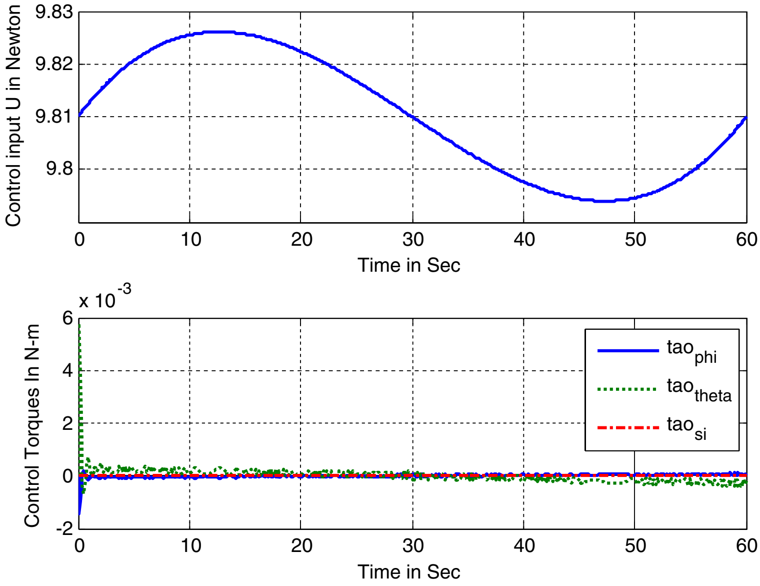

Fig. 6 Input commands for Case I

the jerk (rate of change of acceleration) is minimized over the time horizon. This trajectory ensures that the velocities and accelerations at the end point are zero while meeting the position tracking objective. The following summarizes this approach.

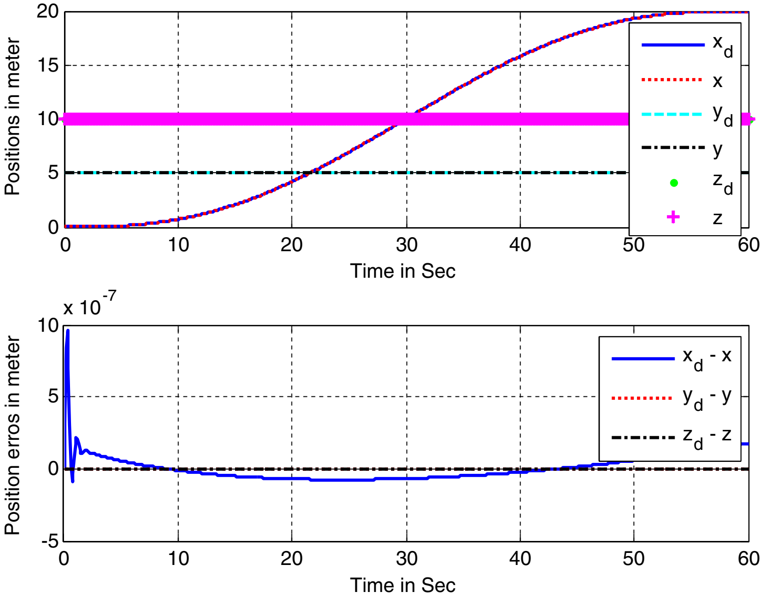

Fig. 7 Plots of position and position tracking errors for *x*command only

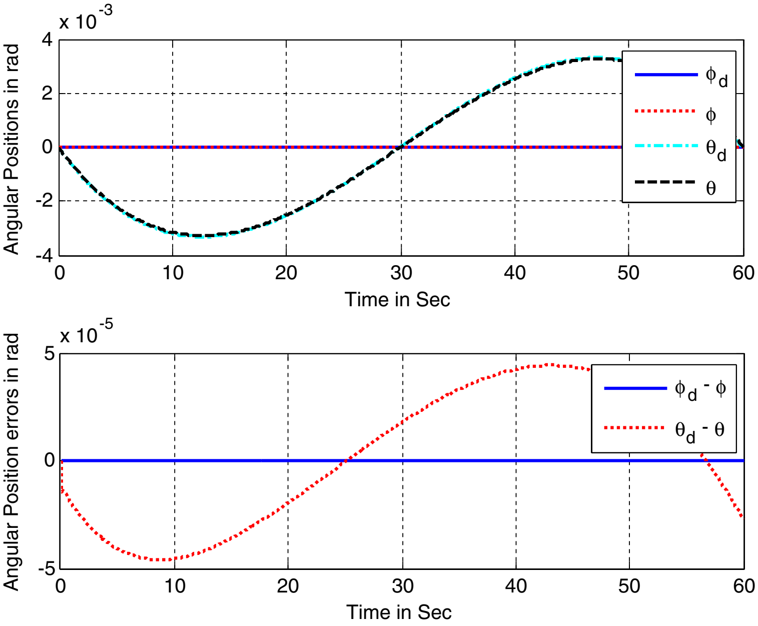

Fig. 8 Angular variations due to change in *x*

Select

$$\underset{d}{\dot{x}}(t) = a_{1_x} + 2a_{2_x}t + 3a_{3_x}t^2 + 4a_{4_x}t^3 + 5a_{5_x}t^4 \tag{68}$$

Differentiating again,

$$\ddot{x}_d(t) = 2a_{2_x} + 6a_{3_x}t + 12a_{4_x}t^2 + 20a_{5_x}t^3 \tag{69}$$

The initial and final velocities and accelerations are zero; therefore from [35, 36] one has

$$\begin{bmatrix} d_x \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & t_f & t_f^2 \\ 3 & 4t_f & 5t_f^2 \\ 6 & 12t_f & 20t_f^2 \end{bmatrix} \begin{bmatrix} a_{3_x} \\ a_{4_x} \\ a_{5_x} \end{bmatrix} \tag{70}$$

where, $d_x = \left(x_{d_f} - x_{d_0}\right) \Big/ t_f^3$. Now, solving for coefficients

$$\begin{bmatrix} a_{3_x} \\ a_{4_x} \\ a_{5_x} \end{bmatrix} = \begin{bmatrix} 1 & t_f & t_f^2 \\ 3 & 4t_f & 5t_f^2 \\ 6 & 12t_f & 20t_f^2 \end{bmatrix}^{-1} \begin{bmatrix} d_x \\ 0 \\ 0 \end{bmatrix} \tag{71}$$

Thus the desired trajectory for the *x* direction is given by

$$x_d(t) = x_{d_0} + a_{3_x}t^3 + a_{4_x}t^4 + a_{5_x}t^5 \tag{72}$$

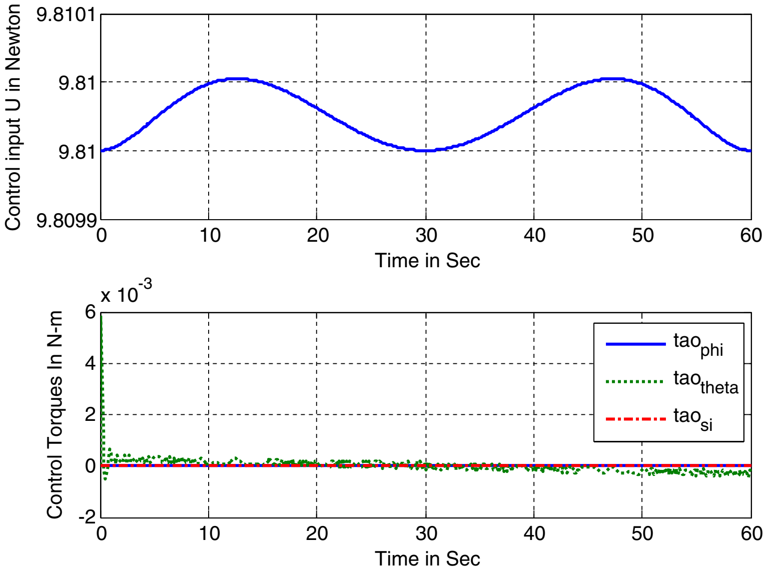

Fig. 9 Input commands for variation in x(Case II)

Similarly, the reference trajectories for the y and z directions are gives by Eqs. 73 and 74 respectively.

$$y_d\,(t) = y_{d_0} + a_{3_y}t^3 + a_{4_y}t^4 + a_{5_y}t^5 \tag{73}$$

$$z_d\,(t) = z_{d_0} + a_{3_z}t^3 + a_{4_z}t^4 + a_{5_z}t^5 \tag{74}$$

The beauty of this method lies in the fact that more demanding changes in position can be accommodated by varying the final time. That is, acceleration/torque ratio can be controlled smoothly as per requirement. For an example simulation of trajectory generation let us consider a initial condition $t = 0, x_{d_0} = 0$ and a desired final condition as $t = 10, x_{d_f} = 10$. Thus one has $d_x = 0.01$ and Eq. 71 yields the coefficients.

$$x_d\,(t) = 0.1t^3 - 0.015t^4 + 0.0006t^5 \tag{75}$$

The trajectories for the four final positions namely $x_{d_f} = 10, 20, 30, 40$ are shown in Fig. 3.

Neural network parameters:

The number of NN neurons used in the simulation is ten. This number was selected as follows. The simulation was done with ten neurons. Then, using eight neurons the performance deteriorated. On the other hand, a simulation using 12 neurons did not improve performance. Therefore, ten neurons are selected. NN weight tuning gains are given by $F_1 = diag(5[1;\ 1;\ 1])$; $F_2 = diag(5[1;\ 1;\ 1])$; $\kappa_1 = 0.01$; $\kappa_2 = 0.01$. The NN weights were initialized at zero, so that initially the tracking loops held the vehicle stable. Once the NNs begin to learn, their outputs are added to the control signals to better compensate the unknown nonlinear force and moment terms.

Two cases are considered in the simulation to show respectively the good tracking property of the control law and its decoupling effectiveness. In the all following cases we set the yaw angle to $\psi_d = 0$.

Case 1: To show effectiveness in tracking a desired (guided) trajectory

Figure 4 describes the controlled motion of the quadrotor from its initial position (0, 0, 0) to final position (20, 5, 10) for a given time (40 s). The desired trajectory were generated using Eqs. 72, 73, and 74. The actual trajectories $x(t)$, $y(t)$, $z(t)$ match exactly their desired values $x_d(t)$, $y_d(t)$, $z_d(t)$ respectively nearly exactly. The errors along the three axes are also shown in the same figure. It can be seen that the tracking is almost perfect as well as the tracking errors are significantly small. This is in spite of the unknown nonlinearities (Eq. 68) that are injected into the simulation dynamics, which are effectively compensated by the neural networks. Figure 5 describes the attitude of the quadrotor φ, θ along with their demands φ_d, θ_d and attitude errors in radian. Again the angles match their command values nearly perfectly. Figure 6 describes the control input requirement which is very much realizable. Note that as described before the control requirement for yaw angle is $\tau_\psi = 0$ and it is seen from Fig. 6.

Case 2: To show the motion decoupling in response to position commands

The quadrotor is commanded to move from (0,5,10) to (20,5,10), which only involves a translation in the x direction. Figures 7 and 8 illustrate the decoupling performance of the control law. Figure 7 shows that $x(t)$ follows the command $x_d(t)$ nearly perfectly. The other variables $y(t)$ and $z(t)$ are held at their initial values. Figure 8 shows that the change in x does not make any influence on φ. The corresponding control inputs are also shown in Fig. 9 and due to the full decoupling effect it is seen that τ_φ is almost zero.

Similar type of simulations performed for y and z directional motions separately yield similar plots that show excellent tracking with decoupling.

Case 3: Comparison study: adaptive backstepping proposed in this paper vs. the controller adopted from [11]

Reference [11] presented a three-loop design for a quadrotor controller. A first loop controlled altitude z and yaw. Assuming perfect control in that loop, a second loop stabilized lateral position y and roll to zero using a ‘nested saturation control’. Assuming again perfect compensation, a third and final loop stabilized forward position x and pitch to zero. Though not developed as such in [11], this is actually a very interesting and intuitive form of backstepping control. Here we compare the performance of the controller in [11] with our controller for an altitude change command trajectory.

Figures 10, 11 and 12 shows the comparison simulation results between the adaptive backstepping controller proposed in this paper and the controller adopted from [11]. Figures 10, 11 and 12 shows the simulation study for hovering control of a quadrotor. That is, the quadrotor is to be controlled along a guided desired trajectory z_d while x_d, y_d should remain silent. The response for the adaptive backstepping controller proposed in this paper is depicted with subscript n and the response of the controller described in [11] is depicted with subscript o. Firstly from Fig. 10 we can see that the controller proposed in this paper follows the trajectory z_d more

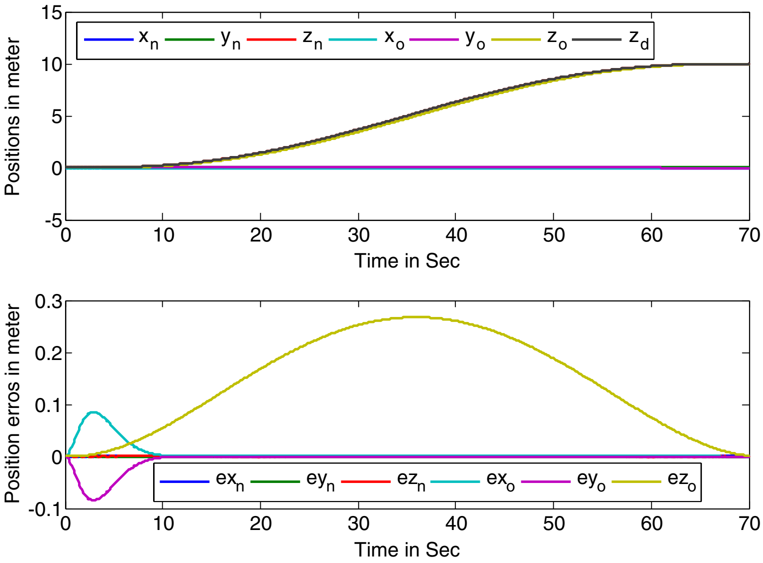

Fig. 10 Case III: Comparison result: position response

accurately than the backstepping controller in [11] (see error response $z_d - z_o$). From the error responses found in the simulation, it is seen that NN backstepping controller proposed in this paper also has less position error magnitude in x and y directions than the backstepping controller in [11]. This shows that the controller proposed in

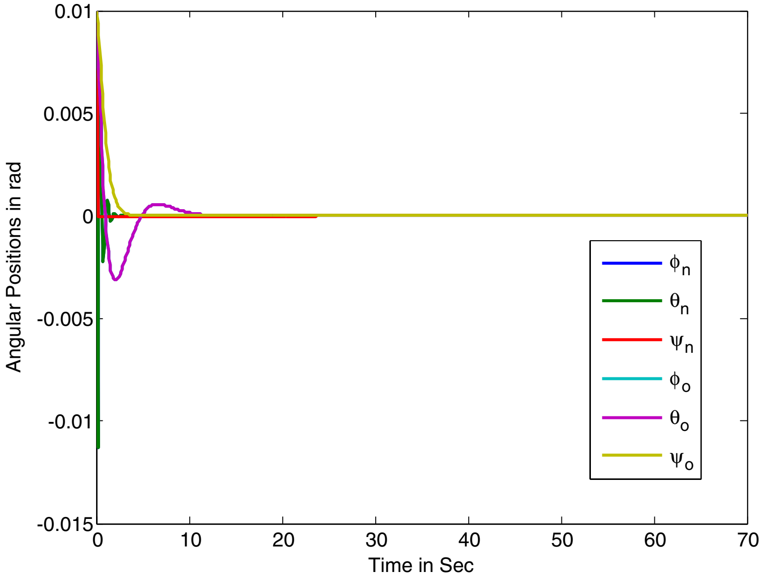

Fig. 11 Case III: Comparison result: attitude response

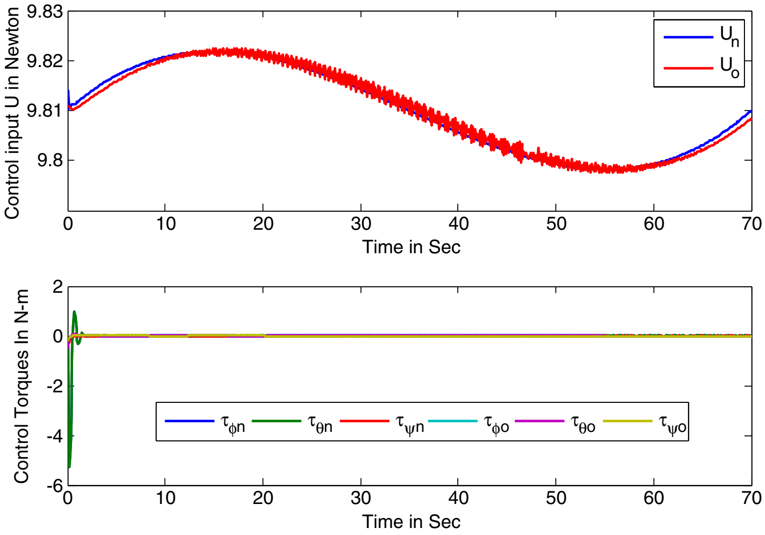

Fig. 12 Case III: Comparison result: input commands

this paper has more decoupling capacity than the controller proposed in [11]. Note that the controller in [11] did not consider nonlinear forces and moments.

From Fig. 11 it is seen that the attitude stabilization occurs more quickly for the controller proposed in this paper than controller proposed in [11]. The thrust input shown in Fig. 12 shows the chattering due to the nested saturation controller in [11], and reveals that our backstepping controller generates a smoother control.

It is also seen from Figs. 11 and 12, that initial variations in attitude and torque are higher for our controller than that in [11]. This is because of the transient phase of NN weights from their initial assigned values of zero to the learned final values for effective compensation of the unknown nonlinearities. From the simulation it is seen, however, that after a very short time these transients vanish.

Remark We conclude from the simulation results that NN backstepping controller gives satisfactory results for a wide range of nonlinearities. The resulting controller structure in Fig. 2 yields tracking with excellent decoupling between command channels. The NN learns on-line to effectively compensate for the unknown aerodynamics forces and moments as well as the nonlinear terms $C\left(\eta, \dot{\eta}\right)$.

7 Conclusion

A backstepping approach to design nonlinear controller for a quadrotor dynamics is discussed in this paper. Using this approach, an intuitively structured controller was derived that has an outer kinematics position control loop and an inner dynamic attitude control loop. The dynamics of a quadrotor are a simplified form of helicopter dynamics that exhibits the basic problems including underactuation, strong coupling, multi-input/multi-output, and unknown nonlinearities. The force dynamics of the quadrotor is completely independent of the rotational dynamics except through the aerodynamic state dependent disturbances including aerodynamic coefficients. These unmodeled state dependent dynamics are estimated by NN and thus yielding a robust control law. The simulation results shown at the end demonstrate the validity of the control law discussed in the paper.

Acknowledgements The research work presented in this paper is supported by NSF grant ECCS-0801330 and ARO grant W91NF-05-1-0314 and the Army National Automotive Center NAC.

References

1. Koo, T.J., Sastry, S.: Output tracking control design of a helicopter model based on approximate linearization. In: Proceedings of the 37th Conference on Decision and Control. Tampa, Florida (1998)
2. Stevens, B.L., Lewis, F.L.: Aircraft Control and Simulation. Wiley, New York (2003)
3. Castillo, P., Lozano, R., Dzul, A.: Modelling and Control of Mini Flying Machines. Springer, Berlin (2005)
4. Bouabdallah, S., Noth, A., Siegwart, R.: PID vs LQ control techniques applied to an weight augmentation high energy consumption indoor micro quadrotor. In: Proceedings of 2004 1EEElRS.J Internationel Conference on Intelligent Robots and Systems. Sendal, Japan (2004)
5. Madani, T., Benallegue, A.: Backstepping control for a quadrotor helicopter. In: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. Beijing, China (2006)
6. Mokhtari, A., Benallegue, A., Orlov, Y.: Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle. Int. J. Robot. Autom. **21**, 39–49 (2006). doi:10.2316/Journal.206.2006.1.206-2842
7. Mokhtari, A., Benallegue, A., Daachi, B.: Robust feedback linearization and GH∞ controller for a quadrotor unmanned aerial vehicle. J. Electr. Eng. **57**, 20–27 (2006)
8. Kim, H.J., Shim, D.H., Sastry, S.: Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In: Proceedings of the American Control Conference. Anchorage, AK (2002)
9. Kanellakopoulos, I., Kokotovic, P.V., Morse, A.S.: Systematic design of adaptive controllers for feedback linearizable systems. IEEE Trans. Automat. Contr. **36**, 1241–1253 (1991). doi:10.1109/9.100933
10. Castillo, P., Dzul, A., Lozano, R.: Real-time stabilization and tracking of a four-rotor mini rotorcraft. IEEE Trans. Contr. Syst. Technol. **12**, 510–516 (2004). doi:10.1109/TCST.2004.825052
11. Castillo, P., Lozano, R., Dzul, A.: Stabilization of a mini rotorcraft having four rotors. IEEE Contr. Syst. Mag. **25**, 45–55 (2005). doi:10.1109/MCS.2005.1550152
12. Bouabdallah, S., Noth, A., Siegwart, R.: PID vs LQ control techniques applied to an indoor micro quadrotor. In: International Conference on Intelligent Robots and Systems, pp. 2451–2456 (2004)
13. Mahony, R., Hamel, T.: Robust trajectory tracking for a scale model autonomous helicopter. Int. J. Robust Nonlinear Contr. **14**, 1035–1059 (2005). doi:10.1002/rnc.931
14. Yang, C.D., Liu, W.H.: Nonlinear Hoo Decoupling hover control of helicopter with parameter uncertainties. In: Proceedings of the American Control Conference. Denver, Colorado (2003)
15. Enns, R., Si, J.: Helicopter flight control design using a learning control Approach1. In: Proceedings of the 39th Conference on Decision and Control. Sydney, Australia (2000)
16. Calise, A.J., Kim, B.S., Leitner, J., Prasad, J.V.R.: Helicopter adaptive flight control using neural networks. In: Proceedings of the 33rd Conference on Decision and Control. Lake Buena Vista, FL (1994)
17. Johnson, E., Kannan, S.: Adaptive trajectory control for autonomous helicopters. AIAA J. Guid. Control Dyn. **28**, 524–538 (2005). doi:10.2514/1.6271
18. Farrell, J., Sharma, M., Polycarpou, M.: Backstepping-based flight control with adaptive function approximation. AIAA J. Guid. Contr. Dyn. **28**, 1089–1102 (2005)
19. Hamel, T., Mahony, R., Lozano, R., Ostrowski, J.: Dynamic modeling and configuration stabilization for an X4-flyer. In: IFAC 15th Triennial World Congress. Barcelona, Spain (2002)

20. Mistler, V., Benallegue, A., M'Sirdi, N.K.: Exact linearization and non-interacting control of a 4 rotors helicopter via dynamic feedback. In: 10th IEEE Int. Workshop on Robot–Human Interactive Communication. Paris (2001)
21. Bijnens, B., Chu, Q.P., Voorsluijs, G.M., Mulder, J.A.: Adaptive feedback linearization flight control for a helicopter UAV. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. San Francisco, California (2005)
22. Altug, E., Ostrowski, J.P., Mahony, R.: Control of a quadrotor helicopter using visual feedback. In: IEEE International Conference on Robotics and Automation. Washington, DC (2002)
23. Cheng, M., Huzmezan, M.: A combined MBPC/2 DOF hinf controller for quad rotor unmanned air vehicle. In: AIAA Atmospheric Flight Mechanics Conference and Exhibit. Austin, Texas, USA (2003)
24. Cowling, I.D., Yakimenko, O.A., Whidborne, J.F., Cooke, A.K.: A prototype of an autonomous controller for a quadrotor UAV. In: Proceedings of the European Control Conference. Kos, Greece (2007)
25. Guo, W., Horn, J.: Modeling and simulation for the development of a quad-rotor UAV capable of indoor flight. In: Modeling and Simulation Technologies Conference and Exhibit. Keystone, Colorado (2006)
26. Mahony, R., Hamel, T., Dzul, A.: Hover control via lyapunov control for an autonomous model helicopter. In: Proceedings of the 38th Conference on Decision & Control. Phoenix, Arizona (1999)
27. Lewis, F.L., Dawson, D.M., Abdallah, C.T.: Robot Manipulator Control: Theory and Practices. Marcel Dekker, New York (2004)
28. Lewis, F., Jagannathan, S., Yesildirek, A.: Neural Network Control of Robot Manipulators and Nonlinear Systems. Taylor and Francis, London (1999)
29. Kim, Y.H., Lewis, F.L.: High Level Feedback Control With Neural Networks. World Scientific. River Edge, NJ (1998)
30. Olfati-Saber, R.: Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles. In: Doctor of Philosophy, Dept. of Electrical Engineering and Computer Science, p. 307. MIT (2001)
31. Hornik, K., Stinchombe, M., White, H.: Multilayer feedforward networks are universal approximations. Neural Netw. **20**, 359–366 (1989). doi:10.1016/0893-6080(89)90020-8
32. Lewis, F.L., Yesildirek, A., Liu, K.: Multilayer neural-net robot controller with guaranteed tracking performance. IEEE Trans. Neural Netw. **7**, 388–399 (1996). doi:10.1109/72.485674
33. Fantoni, I., Zavala, A., Lozano, R.: Global stabilization of a PVTOL aircraft with bounded thrust. In: Proceedings of the 41st Conference on Decision and Control. Las Vegas, Nevada USA (2002)
34. Lewis, F.L., Abdallah, C.T., Dawson, D.M.: Control of Robot Manipulators. Macmillan, New York (1993)
35. Flash, T., Hogan, N.: The coordination of arm movements: an experimentally confirmed mathematical model. J. Neurosci. **5**, 1688–1703 (1985)
36. Hogan, N.: Adaptive control of mechanical impedance by coactivation of antagonist muscles. IEEE Trans. Automat. Contr. **29**, 681–690 (1984). doi:10.1109/TAC.1984.1103644