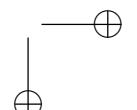
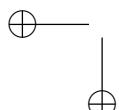
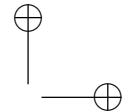
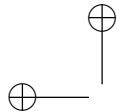


Metadata of the chapter that will be visualized online

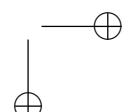
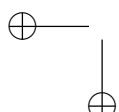
Book Title	Handbook of Unmanned Aerial Vehicles	
Chapter number	61	
Book Copyright - Year	2013	
Copyright Holder	Springer Science+Business Media Dordrecht	
Title	Adaptive Control of Unmanned Aerial Vehicles: Theory and Flight Tests	
Author	Degree	Dr.
	Given Name	Suresh K.
	Particle	
	Family Name	Kannan
	Suffix	
	Phone	
	Fax	
	Email	kanaero@gmail.com
Affiliation	Division	Controls Group, Systems Department
	Organization	United Technologies Research Center
	Postcode	06118
	City	East Hartford
	State	CT
	Country	USA
Author	Degree	Dr.
	Given Name	Girish V.
	Particle	
	Family Name	Chowdhary
	Suffix	
	Phone	
	Fax	
	Email	girishc@MIT.EDU
Affiliation	Division	Department of Aeronautics and Astronautics
	Organization	Massachusetts Institute of Technology
	Street	33-326, 77 Massachusetts Ave.
	Postcode	02139
	City	Cambridge
	State	MA
	Country	USA

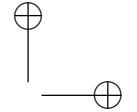




Author	Degree	Prof.
	Given Name	Eric N.
	Particle	
	Family Name	Johnson
	Suffix	
	Phone	
	Fax	
	Email	eric.johnson@ae.gatech.edu
Affiliation	Division	School of Aerospace Engineering
	Organization	Georgia Institute of Technology
	Street	270 Ferst Drive
	Postcode	30332
	City	Atlanta
	State	GA
	Country	USA

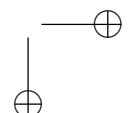
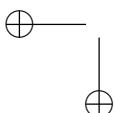
UNCORRECTED PROOF

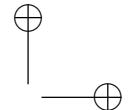
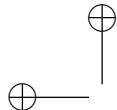




Abstract

Typically, unmanned aerial vehicles are underactuated systems, that is, they have fewer independent control inputs than degrees of freedom. In a helicopter, for example, the body axes roll, pitch, yaw, and altitude are fully actuated. However, lateral and longitudinal translational motion is only possible by tilting the thrust vector. This chapter develops a six degree-of-freedom flight control algorithm that can track both position and attitude trajectories. Approximate inverse models for vehicle attitude and position dynamics are used for feedback linearization leading to an inner-loop that tracks attitude and angular rate and an outer-loop that tracks position and velocity commands. A single adaptive element is used to compensate for inversion errors (uncertainty) in both loops. A key challenge in realizing an adaptive control design on real aircraft is dealing with actuator magnitude and rate saturation. Such saturation elements cannot be easily captured in inverse models and leads to incorrect learning in the adaptive element during periods of saturation. A mechanism to exactly remove such incorrect learning is provided. Additionally, nonlinear reference models are introduced to mitigate the risks of the closed-loop system entering regions of the flight envelope that result in loss-of-controllability. The resulting adaptive controller accepts trajectory commands comprising of desired position, velocity, attitude, and angular velocity and produces normalized actuator signals required for flight control. A modification to the baseline adaptive control system is also provided that enables long-term retention of the uncertainty approximation within the adaptive element. This architecture is validated through flight tests on several fixed wing and rotorcraft UAVs, including a 145-lb helicopter UAV (Yamaha R-Max or GTMax), a scale model fixed-wing aircraft (GTEdge), and a small ducted fan (GTSp).





1

2 Adaptive Control of Unmanned Aerial 3 Vehicles: Theory and Flight Tests

61

4 Suresh K. Kannan, Girish V. Chowdhary, and Eric N. Johnson

5 Contents

6 61.1 Introduction	2
7 61.1.1 Approach	4
8 61.2 Control of an Air Vehicle	7
9 61.2.1 Vehicle Dynamics	7
10 61.2.2 Control Design	10
11 61.2.3 Reference Model and Hedging	13
12 61.2.4 Tracking Error Dynamics	15
13 61.2.5 Boundedness	17
14 61.3 Concurrent Learning	18
15 61.4 Helicopter Specific Design	22
16 61.4.1 Approximate Model	22
17 61.4.2 Reference Model	24
18 61.4.3 Choice of Gain Linear Dynamics	25
19 61.4.4 Imposing Response Characteristics	27
20 61.5 Experimental Results	27
21 61.5.1 Parameter Selections	28
22 61.5.2 Flight Test	29
23 61.5.3 Application to a Ducted Fan	43

S.K. Kannan (✉)

Controls Group, Systems Department, United Technologies Research Center, 06118, East Hartford, CT, USA
e-mail: kanaero@gmail.com

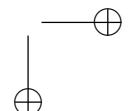
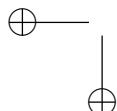
G.V. Chowdhary

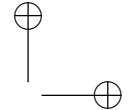
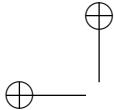
Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 33-326, 77 Massachusetts Ave., 02139, Cambridge, MA, USA
e-mail: girishc@MIT.EDU

AU1

E.N. Johnson

School of Aerospace Engineering, Georgia Institute of Technology, 270 Ferst Drive, 30332, Atlanta, GA, USA
e-mail: eric.johnson@ae.gatech.edu





24	61.5.4 Application to a Fixed-Wing Aircraft	44
25	61.5.5 Implementation of Concurrent-Learning Adaptive Controller on a	
26	VTOL UAV.....	45
27	61.6 Summary.....	54
28	A Adaptive Element	58
29	References	59

AU2

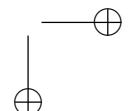
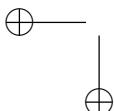
Abstract

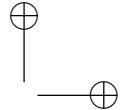
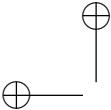
Typically, unmanned aerial vehicles are underactuated systems, that is, they have fewer independent control inputs than degrees of freedom. In a helicopter, for example, the body axes roll, pitch, yaw, and altitude are fully actuated. However, lateral and longitudinal translational motion is only possible by tilting the thrust vector. This chapter develops a six degree-of-freedom flight control algorithm that can track both position and attitude trajectories. Approximate inverse models for vehicle attitude and position dynamics are used for feedback linearization leading to an inner-loop that tracks attitude and angular rate and an outer-loop that tracks position and velocity commands. A single adaptive element is used to compensate for inversion errors (uncertainty) in both loops. A key challenge in realizing an adaptive control design on real aircraft is dealing with actuator magnitude and rate saturation. Such saturation elements cannot be easily captured in inverse models and leads to incorrect learning in the adaptive element during periods of saturation. A mechanism to exactly remove such incorrect learning is provided. Additionally, nonlinear reference models are introduced to mitigate the risks of the closed-loop system entering regions of the flight envelope that result in loss-of-controllability. The resulting adaptive controller accepts trajectory commands comprising of desired position, velocity, attitude, and angular velocity and produces normalized actuator signals required for flight control. A modification to the baseline adaptive control system is also provided that enables long-term retention of the uncertainty approximation within the adaptive element. This architecture is validated through flight tests on several fixed wing and rotorcraft UAVs, including a 145-lb helicopter UAV (Yamaha R-Max or GTMax), a scale model fixed-wing aircraft (GTEdge), and a small ducted fan (GTSPy).

AU3

61.1 Introduction

The US Department of Defense Integrated Unmanned Systems Roadmap (2011) defines four levels of autonomy for unmanned systems. Level 1 involves manual operator control. Level 2 assumes automatic control while using humans to delegate waypoints. Level 3 assumes the UAV is capable of performing high-level activities using sensed data when given some directions by a human. Level 4 assumes that the UAV is capable of taking a top-level goal, breaking it into tasks and executing them along with the contingency replanning necessary to accomplish a top-level goal. Level 4 approaches the abstract and high-level goals that are provided to human soldiers in today’s battlefield.



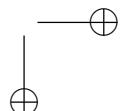
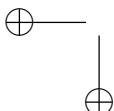


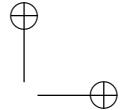
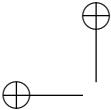
66 Level 2 behavior is available in almost all UAVs today. However, this
67 accomplishment is after at least two decades of development with multiple efforts
68 for each type of vehicle. The current US Department of Defense focus is on
69 research, development, and procurement of technologies that encompass Level 3
70 and Level 4 autonomous operations while assuming Level 2 is available. The key
71 to developing fully autonomous Level 4-type behaviors is the ability to plan and
72 also deal with contingencies. At the flight control system level this translates to
73 the closed-loop system being robust and adaptive to changes in the environment
74 and vehicle dynamics. To motivate further, future unmanned systems will involve
75 verification and validation using formal approaches where an amount of trust in
76 autonomy is generated. Current conventional FAA certification practices will be
77 superseded with certification methods that allow integration of UAV’s into the civil
78 airspace. One straightforward metric that may be used for *trust in flight control* is the
79 variance in trajectory tracking with uncertainty in the dynamics and environment.
80 Consequently, the vehicles may be required to perform at their limits in order
81 to maintain the required performance. Most current control systems still do not
82 leverage the full flight envelope of small helicopters, at least, unless significant and
83 expensive system identification and validation has been conducted.

84 Currently, fast algorithms to plan routes around obstacles are available (Frazzoli
85 et al. 2002; Karaman and Frazzoli 2011). To be truly useful, these routes would
86 include high-speed dashes, tight turns around buildings, avoiding dynamic obstacles
87 and other required aggressive maneuvers. Allowing control saturation and adapta-
88 tion allows higher-level planning algorithms to provide optimistic trajectories which
89 are then tracked by the flight controller to the best of the vehicles ability. The
90 following is a description of some key elements that affect flight control stability
91 and performance.

92 *Parametric uncertainty* arises from uncertainty in the mass, inertia properties,
93 and aerodynamic characteristics. This limits the safe operational envelope of the
94 vehicle to flight regimes where control designs are valid and parametric uncertainty
95 is small. The effects of parametric uncertainty and unmodeled dynamics can
96 be handled using a combination of system identification (Gavrilets et al. 2001;
97 Civita et al. 2002a; Mettler 2002) and robust control techniques (Civita et al.
98 2002b, 2003; Gavrilets et al. 2002). However, system identification is expensive and
99 when changes happen in real-time, for e.g., an unexpected payload is attached or
100 deployed, the available previously-identified models may not fit the current aircraft
101 configuration. In this chapter, parametric uncertainty arises due to approximate
102 inversion and appears as an unknown nonlinear function of the states and controls
103 (unstructured). An adaptive element (neural network) is then used as a nonlinear
104 function approximator to instantaneously cancel the inversion error.

105 *Unmodeled dynamics* arise when the vehicle model used for control design
106 neglects parts of the real system’s dynamics. Examples include the Bell-Hillier
107 stabilizer bar and the flapping dynamics of the helicopter rotor blades. For most
108 autonomous flight control work, the flapping dynamics may be safely neglected,
109 whereas the Bell-Hillier stabilizer bar found on small rotorcraft cannot be ig-
110 nored when high-bandwidth attitude control is desired. This chapter assumes state





111 feedback of rigid-body states keeping the control design simpler by leveraging the
112 control design’s robustness to unmodeled dynamics. Related adaptive designs that
113 use an output-feedback controller formulation to explicitly deal with unmodeled
114 dynamics are described in Calise et al. (2001) with experimental results in Corban
115 et al. (2003).

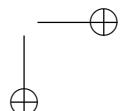
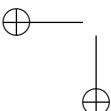
116 *Actuator magnitude and rate saturation* limit control authority and thus the
117 closed-loop stability and performance. Addressing input dynamics constitutes an
118 important class of control design methods for systems with bounded actuation and
119 includes Sontag’s universal formula approach using control Lyapunov functions
120 (Lin and Sontag 1991) and others (Bemporad et al. 1997; Pappas 1996). Avoiding
121 saturation, however, usually results in either conservative or highly complex control
122 laws leading to possibly very conservative domains of attraction and slow conver-
123 gence. See Bernstein and Michel (1995) and the references therein for a survey of
124 early work on constrained control.

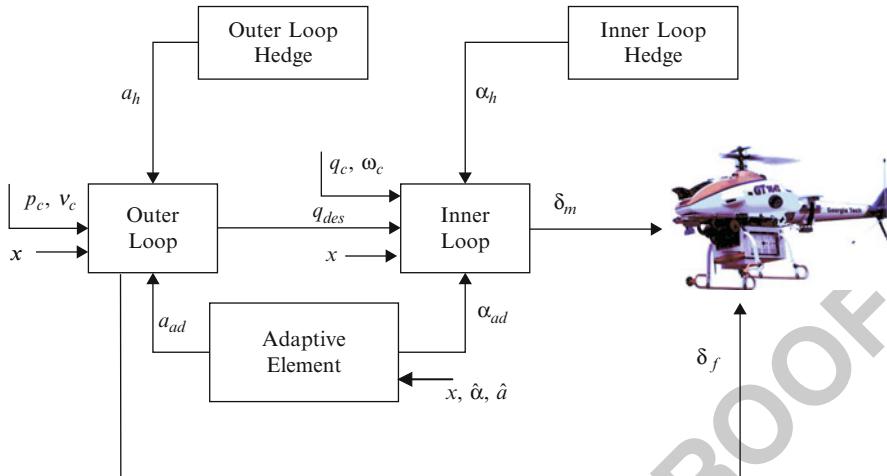
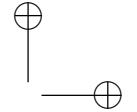
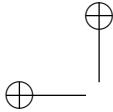
125 Other related works on rotorcraft control may be found in Kendoul et al.
126 (2006, 2007, 2010) and a recent survey on guidance, navigation and control methods
127 for rotorcraft is available in Kendoul (2012).

128 **61.1.1 Approach**

129 Helicopters have six degrees of freedom when considering just the rigid body modes
130 and 4 independent controls are available to control them. Traditionally, the control
131 variables lateral stick, δ_{lat} ; longitudinal stick, δ_{lon} ; and pedal, δ_{ped} , control moments
132 around the roll, pitch, and yaw axes, respectively. Finally, the collective input,
133 δ_{coll} , produces thrust along the main rotor shaft. The rotational dynamics are fully
134 actuated, whereas the translational dynamics are underactuated, but controllable.
135 The rotor thrust has to be oriented using the aircraft’s pitch and roll attitude to
136 produce translational accelerations.

137 An overall architecture of the approach is shown in Fig. 61.1 with details
138 in Fig. 61.6. The outer loop is responsible for tracking desired translational
139 accelerations. It generates δ_{coll} to vary rotor thrust along the main shaft and also
140 generates the desired roll and pitch angles to orient the thrust vector to generate
141 linear accelerations in these two underactuated degrees of freedom. Note here that
142 the desired pitch and roll angles are commands to the inner-loop controller. In
143 this respect the inner loop acts like a (virtual) actuator as far as the outer loop is
144 concerned. Similarly, the inner loop generates the actuator deflections necessary to
145 control the rotational dynamics. Of course, here the inner-loop’s output actuation
146 signal is subject to the real actuator dynamics of the physical aircraft. In both loops,
147 approximate models of the rotational (inner loop) and translational (outer loop)
148 dynamics are dynamically inverted to produce the actuator deflections (and desired
149 pitch and roll) necessary to achieve the desired angular and linear accelerations.
150 These desired accelerations are generated using reference models dictating the
151 desired ideal closed-loop response. The cascaded inner-outer loop architecture used
152 here is commonly employed in aerospace control applications due to the different



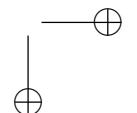
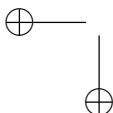
**Fig. 61.1** Overall architecture

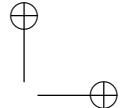
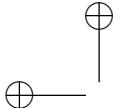
153 dynamical time-scales of the two loops. ▶ [Chapter 49](#) in this book discusses relevant
 154 details of cascaded control systems for UAVs.

155 Adaptation is introduced in all six degrees of freedom to account for inversion
 156 errors arising from the approximate models used for inversion purposes. There is
 157 no particular restriction on the inversion that results in *desired* actuator deflections
 158 to be bounded. Hence, at large desired accelerations, large actuator deflections may
 159 be commanded. Such saturation and dynamics will now appear in the adaptation
 160 training signal. This is also true in the case of the outer loop because the commanded
 161 pitch and roll attitudes are now subject to the closed-loop dynamics of the inner loop
 162 in addition to the actuator dynamics of the δ_{coll} actuator.

163 These nonlinearities appear in the stability analysis by way of their appearance
 164 in the error dynamics. The pseudocontrol hedging signal (PCH) is introduced in
 165 the outer-loop and inner-loop reference models in a manner that exactly removes
 166 elements of actuator saturation from the training signal for the adaptive element. The
 167 reference models themselves are nonlinear and prescribe the aggressiveness with
 168 which external commands are achieved. Thus, a comprehensive nonlinear, adaptive,
 169 trajectory tracking controller capable of adapting to uncertainties in all six degrees
 170 of freedom is developed. It must be noted that although the concrete example
 171 used throughout this chapter is one of a helicopter, the controller is not specific
 172 to a helicopter UAS. The development is generic; the only difference between a
 173 helicopter, a fixed-wing aircraft, or other esoteric aircraft is the manner in which
 174 the available controls are categorized and the approximate models used for dynamic
 175 inversion purposes.

176 An underlying assumption of this work is that the nonlinear modeling error
 177 uncertainty can be approximated by a continuous function over the flight domain
 178 of an aircraft. The goal is to capture an approximation of the uncertainty using





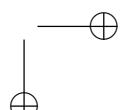
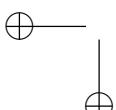
AU4

universal approximators such as neural networks. This universal approximation property guarantees that given a sufficient number of neurons, there exists an optimal set of (a priori unknown) weights that can approximate the uncertainty to a desired minimum approximation error. Once these weights are found, the learned dynamics can be used for online planning and health-monitoring purposes. The baseline adaptive laws developed in later sections of this chapter are designed to cancel instantaneous model error but do not necessarily guarantee convergence to the ideal weights during normal course of operation (Johnson and Kannan 2005; Kannan and Johnson 2010b; Kannan 2005). To alleviate this restriction, a modification, the *concurrent-learning adaptive control method*, is introduced that greatly improves the convergence of weights to their ideal values in real-world conditions (Chowdhary and Johnson 2011b). The method can in fact guarantee exponential convergence of the neural network weights to a neighborhood of their ideal values for linearly parameterized neural networks (Chowdhary 2010).

The adaptive controller described in this chapter has been extensively validated in flight on several aircrafts regularly since 2002. The range of aircraft types include the Yamaha RMAX (GTMax) helicopter ([Fig. 61.2](#)); an 11-in. ducted fan, the GTSpy ([Fig. 61.3](#)); a tailless fixed-wing aircraft, the D6; and a high-thrust-to-weight-ratio aircraft, the GTEdge ([Fig. 61.4](#)). The GTEdge is a tilt-body fixed-wing aircraft and capable of hovering on its propeller and flying like a regular fixed-wing aircraft. An interesting set of maneuvers performed by the GTEdge is the hover⇒forward flight⇒hover, all using the same adaptive control system. The methods discussed here have also been implemented on smaller aircraft such as



Fig. 61.2 The GTMax helicopter



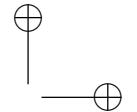
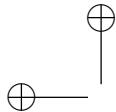


Fig. 61.3 The GT Spy 11-in. ducted fan

the GT Twinstar (Fig. 61.5), a foam-built twin-engine aircraft; the GT Logo, a small rotorcraft of about 1-m rotor diameter; and the GTQ (Chowdhary et al. 2011b), a miniature quadrotor. On the GT Twinstar, a variant of the algorithms presented here was used for flight with 25 % right wing missing (Chowdhary et al. 2010, 2011a, 2012).

207 **61.2 Control of an Air Vehicle**

208 **61.2.1 Vehicle Dynamics**

209 Consider an air vehicle modeled as a nonlinear system of the form

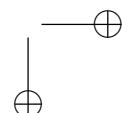
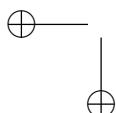
AU5

$$210 \quad \dot{p} = v, \quad (61.1)$$

$$211 \quad \dot{v} = a(p, v, q, \omega, \delta_f, \delta_m), \quad (61.2)$$

$$212 \quad \dot{q} = \dot{q}(q, \omega), \quad (61.3)$$

$$213 \quad \dot{\omega} = \alpha(p, v, q, \omega, \delta_f, \delta_m), \quad (61.4)$$



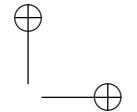
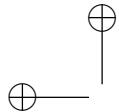


Fig. 61.4 The GTEdge aircraft with a high (greater than 1) thrust-to-weight ratio

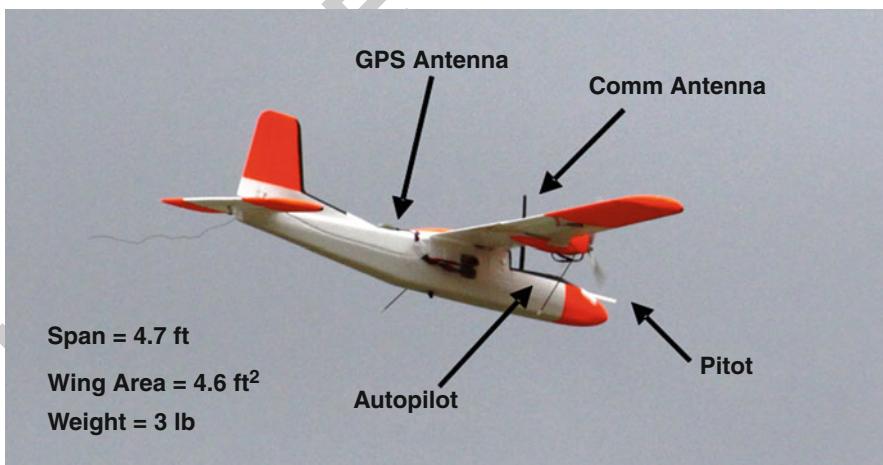
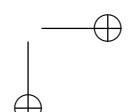
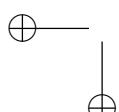
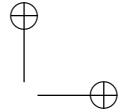
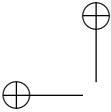


Fig. 61.5 The GT Twinstar foam-built twin-engine aircraft equipped for fault-tolerant control work (see, e.g., Chowdhary et al. 2011a)





214 where $p \in \mathcal{R}^3$ is the position vector, $v \in \mathcal{R}^3$ is the velocity of the vehicle, $q \in \mathcal{R}^4$
 215 is the attitude quaternion, and $\omega \in \mathcal{R}^3$ is the angular velocity. Equation (61.2)
 216 represents translational dynamics and Eq.(61.4) represents the attitude dynamics.
 217 Together, they represent rigid body dynamics and flat-earth kinematics as given
 218 in Etkin (1972) and Stevens and Lewis (2003b) and discussed in detail in the
 219 ►Chap. 49 in this book. Equation (61.3) represents the quaternion propagation
 220 equations (Stevens and Lewis 2003b). The use of quaternions, though not a minimal
 221 representation of attitude, avoids numerical and singularity problems that Euler-
 222 angle-based representations have. This enables the control system to be all attitude
 223 capable as required for aggressive maneuvering. The state vector x may now be
 224 defined as $x \triangleq [p^T \ v^T \ q^T \ \omega^T]^T$.

225 *Remark 61.1.* The objective is to design a control system that can track a given
 226 position, velocity, attitude, and angular rate trajectory. The consolidated trajectory
 227 command is given by $[p_c^T \ v_c^T \ q_c^T \ \omega_c^T]^T$.

228 The control vectors are denoted by δ_f and δ_m and represent actual physical
 229 actuators on the aircraft, where δ_f denotes the primary force-generating actuators
 230 and δ_m denotes the primary moment generating actuators. For a helicopter, the
 231 main force effector is the rotor thrust which is controlled by changing main rotor
 232 collective δ_{coll} . Hence, $\delta_f \in \mathcal{R} = \delta_{coll}$. There are three primary moment control
 233 surfaces, the lateral cyclic δ_{lat} , longitudinal cyclic δ_{lon} , and tail rotor pitch, also
 234 called the pedal input δ_{ped} . Hence, $\delta_m \in \mathcal{R}^3 = [\delta_{lat} \ \delta_{lon} \ \delta_{ped}]^T$. In this chapter, the
 235 primary moment producing controls are treated as the inner-loop control effector,
 236 whereas the $\delta_f = \delta_{coll}$ is treated as an outer-loop control effector. In general,
 237 both control inputs, δ_f and δ_m , may each produce both forces and moments. The
 238 helicopter is an underactuated system, and hence, the aircraft attitude, q , is treated
 239 like a *virtual actuator* used to tilt the main rotor thrust in order to produce desired
 240 translational accelerations in the longitudinal and lateral directions. Thus, it is not
 241 possible to track the commanded pitch and roll for a helicopter independently. It
 242 is only possible to track the heading component of the attitude q_c and body-yaw
 243 rate ω_3 independently. Direct control over the translational accelerations in the
 244 *body-z-axis* is possible using δ_{coll} .

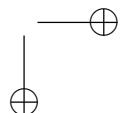
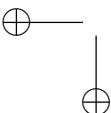
245 The consolidated control vector δ is defined as

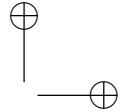
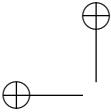
$$246 \quad \delta \triangleq [\delta_f^T \ \delta_m^T]^T;$$

247 the actuators themselves may have dynamics represented by

$$248 \quad \dot{\delta} = \begin{bmatrix} \dot{\delta}_m \\ \dot{\delta}_f \end{bmatrix} = \begin{bmatrix} g_m(x, \delta_m, \delta_{m_{des}}) \\ g_f(x, \delta_f, \delta_{f_{des}}) \end{bmatrix} = g(x, \delta, \delta_{des}), \quad (61.5)$$

249 where $g(\cdot)$ is generally unknown.





250 *Remark 61.2.* It is possible to extend the architecture in order to treat actuator
 251 dynamics as simply another system in cascade with the translational and attitude
 252 dynamics, and the control design would include an outer, inner, and actuator loop
 253 with the actuator loop being the lowest block in the cascade. However, unless the
 254 physical actuators need to be stabilized, their internal dynamics may be assumed
 255 to be asymptotically stable. In this chapter, rate and higher-order dynamics are
 256 ignored, but magnitude saturation will be handled explicitly. It can be shown that
 257 such an assumption is possible because the control design is robust to the unmodeled
 258 dynamics (Kannan 2005).

259 61.2.2 Control Design

260 The control architecture is based on a model reference adaptive control architecture
 261 (see Fig. 61.6). Noting that Eqs. (61.1) and (61.3) represent exactly known kinemat-
 262 ics, approximate models for translational acceleration, \hat{a} , and a model for angular
 263 acceleration, $\hat{\alpha}$, need to be established.

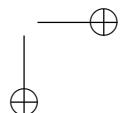
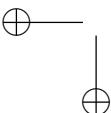
$$264 \quad \begin{bmatrix} a_{des} \\ \alpha_{des} \end{bmatrix} = \begin{bmatrix} \hat{a}(p, v, q_{des}, \omega, \delta_{fdes}, \hat{\delta}_m) \\ \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \delta_{m_{des}}) \end{bmatrix}.$$

265 Here, a_{des} and α_{des} are commonly referred to as the pseudocontrol and represent
 266 desired accelerations. Additionally, δ_{fdes} , $\delta_{m_{des}}$, q_{des} are the control inputs and
 267 attitude expected to achieve the desired pseudocontrol. This form assumes that
 268 translational dynamics are coupled strongly with attitude dynamics, as is the case
 269 for a helicopter. From the outer-loop’s point of view, q (attitude) is like a *virtual*
 270 *actuator* that generates translational accelerations and q_{des} is the desired attitude
 271 that the outer-loop inversion expects will contribute toward achieving the desired
 272 translational acceleration, a_{des} . The dynamics of q appear like actuator dynamics to
 273 the outer loop.

274 *Remark 61.3.* Although the models are approximate, their functional dependence
 275 on vehicle rigid body and actuator states are stated accurately for completeness. It is
 276 likely that a specific approximate model that is introduced might drop some of this
 277 dependency.

278 *Remark 61.4.* The attitude quaternion q_{des} will be used to augment the externally
 279 commanded attitude q_c to achieve the desired translational accelerations. Ideally,
 280 the trajectory generator would generate a commanded attitude q_c that is consistent
 281 with the translational acceleration profile needed to track $x_c(t)$ and $v_c(t)$. If not, the
 282 outer-loop inverse takes care of correcting it by an amount necessary to achieve the
 283 desired translational accelerations in the longitudinal and lateral directions.

284 The models have a functional dependence on current actuator position.
 285 Because actuator positions are often not measured on small unmanned aerial



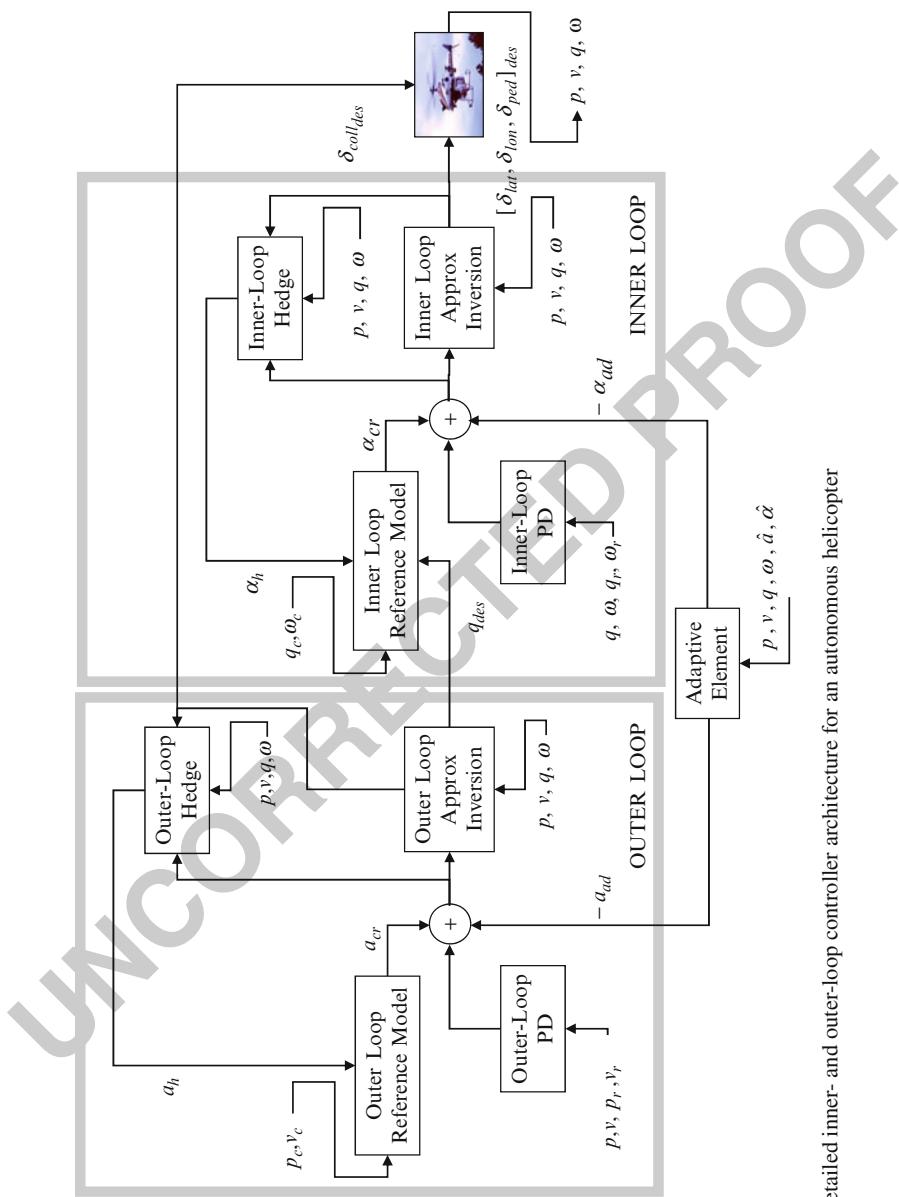
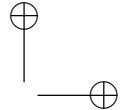
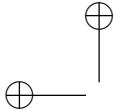


Fig. 61.6 Detailed inner- and outer-loop controller architecture for an autonomous helicopter



286 vehicles, estimates of the actuator positions $\hat{\delta}_m, \hat{\delta}_f$ can be used. When the actuator
 287 positions are directly measured, they may be regarded as known $\hat{\delta}_m = \delta_m$ and
 288 $\hat{\delta}_f = \delta_f$. In fact, in the outer-loop’s case, good estimates of the roll and pitch
 289 attitude virtual actuators are available using inertial sensors and a navigation
 290 filter. The approximate models may now be inverted to obtain the desired control
 291 and attitude

$$292 \quad \begin{bmatrix} \delta_{fdes} \\ q_{des} \end{bmatrix} = \begin{bmatrix} \hat{a}_{\delta_f}^{-1}(p, v, a_{des\delta_f}, \omega, \hat{\delta}_m) \\ \hat{a}_q^{-1}(p, v, a_{des_q}, \omega, \hat{\delta}_m) \end{bmatrix} \quad (61.6)$$

$$293 \quad \delta_{mdes} = \hat{\alpha}^{-1}(p, v, q, \omega, \hat{\delta}_f, \alpha_{des}),$$

294 with $a_{des\delta_f} + a_{des_q} = a_{des}$, $\hat{a}_{\delta_f}, \hat{a}_q$ formulated to be consistent with Eq. (61.6) and
 295 where actuator estimates are given by actuator models

$$296 \quad \dot{\delta} = \begin{bmatrix} \dot{\hat{\delta}}_f \\ \dot{\hat{\delta}}_m \end{bmatrix} = \begin{bmatrix} \hat{g}_f(x, \hat{\delta}_f, \delta_{fdes}) \\ \hat{g}_m(x, \hat{\delta}_m, \delta_{mdes}) \end{bmatrix} = \hat{g}(x, \hat{\delta}, \delta_{des}). \quad (61.7)$$

297 Introducing the inverse control law Eq. (61.6) into Eqs. (61.2) and (61.4) results
 298 in the following closed-loop translational and attitude dynamics:

$$299 \quad \dot{v} = a_{des} + \bar{\Delta}_a(x, \delta, \hat{\delta}) - a_h$$

$$300 \quad \dot{\omega} = \alpha_{des} + \bar{\Delta}_\alpha(x, \delta, \hat{\delta}) - \alpha_h, \quad (61.8)$$

302 where

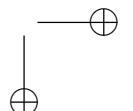
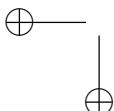
$$303 \quad \bar{\Delta}(x, \delta, \hat{\delta}) = \begin{bmatrix} \bar{\Delta}_a(x, \delta, \hat{\delta}) \\ \bar{\Delta}_\alpha(x, \delta, \hat{\delta}) \end{bmatrix} = \begin{bmatrix} a(x, \delta) - \hat{a}(x, \hat{\delta}) \\ \alpha(x, \delta) - \hat{\alpha}(x, \hat{\delta}) \end{bmatrix}, \quad (61.9)$$

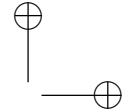
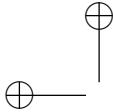
304 are static nonlinear functions (model error) that arise due to imperfect model
 305 inversion and errors in the actuator model \hat{g} . The main discrepancy between $g(\cdot)$
 306 and $\hat{g}(\cdot)$ is the lack of a magnitude-saturation function in \hat{g} . This is required in order
 307 to maintain invertibility. The signals, a_h and α_h , represent the pseudocontrol that
 308 cannot be achieved due to actuator input characteristics such as saturation. If the
 309 model inversion were perfect and no magnitude saturation were to occur, $\bar{\Delta}, a_h$ and
 310 α_h would vanish leaving only the pseudocontrols a_{des} and α_{des} .

311 Two tasks now remain, (1) stabilize the feedback linearized dynamics and (2)
 312 address the effects of model error. The desired accelerations may be designed as

$$313 \quad a_{des} = a_{cr} + a_{pd} - \bar{a}_{ad}$$

$$314 \quad \alpha_{des} = \alpha_{cr} + \alpha_{pd} - \bar{\alpha}_{ad}, \quad (61.10)$$





315 where a_{cr} and α_{cr} are outputs of reference models for the translational and attitude
 316 dynamics, respectively. a_{pd} and α_{pd} are outputs of proportional-derivative (PD)
 317 compensators; and finally, \bar{a}_{ad} and $\bar{\alpha}_{ad}$ are the outputs of an adaptive element
 318 designed to cancel model error $\bar{\Delta}$. The effects of input dynamics, represented by
 319 a_h, α_h , will first be addressed in the following section by designing the reference
 320 model dynamics such that they do not appear in the tracking error dynamics. The
 321 reference model, tracking error dynamics, and boundedness are discussed in the
 322 following sections with details of the adaptive element left to Appendix Sect. A.

323 61.2.3 Reference Model and Hedging

324 Any dynamics and nonlinearities associated with the actuators δ_m, δ_f have not yet
 325 been considered in the design. If they become saturated (position or rate), the
 326 reference models will continue to demand tracking as though full authority were
 327 still available. Furthermore, the inner loop appears like an actuator with dynamics
 328 to the outer loop. Practical operational limits on the maximum attitude of the aircraft
 329 may have also been imposed in the inner-loop reference model. This implies that the
 330 outer-loop desired attitude augmentation q_{des} may not actually be achievable or at
 331 the very least is subject to the inner-loop dynamics.

332 If the reference model is designed as

$$\begin{aligned} 333 \quad \dot{v}_r &= a_{cr}(p_r, v_r, p_c, v_c) \\ 334 \quad \dot{\omega}_r &= \alpha_{cr}(q_r, \omega_r, q_c \oplus q_{des}, \omega_c), \end{aligned} \quad (61.11)$$

AU6

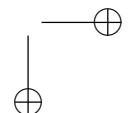
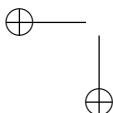
335 where p_r and v_r are the outer-loop reference model states and q_r, ω_r are the
 336 inner-loop reference model states. The external command signal is
 337 $x_c = [p_c^T \ v_c^T \ q_c^T \ \omega_c^T]^T$. The attitude rotation desired by the outer loop is now added
 338 to the commands for the inner-loop controller. Here, $q_c \oplus q_{des}$ denotes quaternion
 339 multiplication (Stevens and Lewis 2003a) and effectively concatenates the two
 340 rotations.

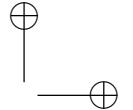
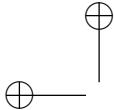
341 If tracking error dynamics is computed by subtracting Eq.(61.10) from
 342 Eq.(61.11), the un-achievable acceleration a_h, α_h will appear in the tracking error
 343 dynamics. When an adaptive element such as a neural network or integrator is
 344 introduced, these effects of input dynamics propagate into the training signal and
 345 eventually result in the adaptive element attempting to correct for them, leading to
 346 incorrect adaptation.

347 Tackling this issue involves redesigning the reference model by subtracting the
 348 deficit accelerations (pseudocontrol hedging):

$$349 \quad \dot{v}_r = a_{cr}(p_r, v_r, p_c, v_c) - a_h, \quad (61.12)$$

$$350 \quad \dot{\omega}_r = \alpha_{cr}(q_r, \omega_r, q_c \oplus q_{des}, \omega_c) - \alpha_h, l \quad (61.13)$$





351 where a_h and α_h are the differences between commanded pseudocontrol and an
 352 estimate of the achieved pseudocontrol. It is an estimate because actual actuator
 353 positions may not be known. Additionally, the aircraft state vectors p, v, q, ω are
 354 estimated using a Kalman filter (Christophersen et al. 2006; Chowdhary et al.
 355 2011b). However, for purposes of control design, they are assumed to be known,
 356 and thus the virtual actuators such as attitude may be assumed to be known
 357 in the PCH computation. This assumption may have to be revisited in the case
 358 where the control/observer pair is not assumed to be separable, perhaps in a tough
 359 localization problem where the control inputs directly affect the observability of the
 360 aircraft states.

361 The PCH signals are given by

$$\begin{aligned} 362 \quad a_h &= \hat{a}(p, v, q_{des}, \omega, \delta_{f_{des}}, \hat{\delta}_m) - \hat{a}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m) \\ 363 \quad &= a_{des} - \hat{a}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m), \end{aligned} \quad (61.14)$$

$$\begin{aligned} 364 \quad \alpha_h &= \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_{m_{des}}) - \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m) \\ 365 \quad &= \alpha_{des} - \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m). \end{aligned} \quad (61.15)$$

366 The hedge signals a_h, α_h , do not directly affect the reference model output a_{cr}, α_{cr} ,
 367 but do so only through subsequent changes in the reference model states.

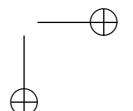
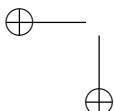
368 The command tracking error may now be defined as e_r

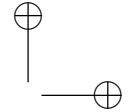
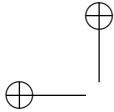
$$369 \quad e_r \triangleq \begin{bmatrix} p_c - p_r \\ v_c - v_r \\ \tilde{Q}(q_c, q_r) \\ \omega_c - \omega_r \end{bmatrix}, \quad (61.16)$$

370 with corresponding command tracking error dynamics given by

$$371 \quad \dot{e}_r = \begin{bmatrix} v_c - v_r \\ a_c - (a_{cr} - a_h) \\ \omega_c - \omega_r \\ \alpha_c - (\alpha_{cr} - \alpha_h) \end{bmatrix}. \quad (61.17)$$

372 The particular form of the reference model dynamics chosen for the translational
 373 dynamics, a_{cr} , and attitude dynamics, α_{cr} , has profound effects on the overall
 374 response and controllability of the system. This is fully expounded in Chap. 4 of
 375 Kannan (2005) and in Kannan and Johnson (2010a). Also see Kannan and Johnson
 376 (2003) for a discussion on the effects of reference model poles when various
 377 elements saturate.





378 As a summary, three reference models were considered:

- 379 • A *linear reference model* will attempt to elicit a linear response in the plant when
380 no such response is possible (peaking) as the plant is nonlinear, especially with
381 the magnitude saturation of actuators.
- 382 • The *nested saturation-based reference model* is an alternative to the linear
383 reference model containing saturations functions appearing in a nested form and
384 is based on the work by Teel (1996, 1997). This form allows one to restrict the
385 evolution of states in a prescribable manner.
- 386 • The *constrained linear reference model* is a special case of the nested saturation-
387 based reference model that is locally linear near the origin.

388 For the quadratic candidate Lyapunov functions chosen in Kannan (2005),
389 only the nested-saturation and constrained linear reference models have their
390 Lyapunov derivative bounds on the PCH signals a_h, α_h . In this chapter, the
391 constrained reference model is used with equations given later in
392 Sect. 61.4.2.

393 61.2.4 Tracking Error Dynamics

394 The tracking error vector is defined as, e as

$$395 e \triangleq \begin{bmatrix} p_r - p \\ v_r - v \\ \tilde{Q}(q_r, q) \\ \omega_r - \omega \end{bmatrix}, \quad (61.18)$$

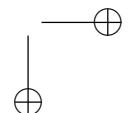
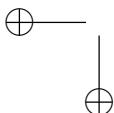
396 where $\tilde{Q} : \mathcal{R}^4 \times \mathcal{R}^4 \mapsto \mathcal{R}^3$ is a function (Johnson 2000) that, given two quaternions,
397 results in an error angle vector with three components. An expression for \tilde{Q} is
398 given by

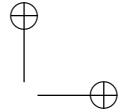
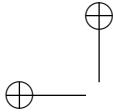
$$399 \tilde{Q}(p, q) = 2\operatorname{sgn}(q_1 p_1 + q_2 p_2 + q_3 p_3 + q_4 p_4) \times \\ 400 \begin{bmatrix} -q_1 p_2 + q_2 p_1 + q_3 p_4 - q_4 p_3 \\ -q_1 p_3 - q_2 p_4 + q_3 p_1 + q_4 p_2 \\ -q_1 p_4 + q_2 p_3 - q_3 p_2 + q_4 p_1 \end{bmatrix}. \quad (61.19)$$

401 The output of the PD compensators may be written as

$$402 \begin{bmatrix} a_{pd} \\ \alpha_{pd} \end{bmatrix} = \begin{bmatrix} R_p & R_d & 0 & 0 \\ 0 & 0 & K_p & K_d \end{bmatrix} e, \quad (61.20)$$

403 where $R_p, R_d \in \mathcal{R}^{3 \times 3}$, $K_p, K_d \in \mathcal{R}^{3 \times 3}$ are linear gain positive-definite matrices
404 whose choice is discussed below. The tracking error dynamics may be found by





405 directly differentiating Eq. (61.18):

$$406 \quad \dot{e} = \begin{bmatrix} v_r - v \\ \dot{v}_r - \dot{v} \\ \omega_r - \omega \\ \dot{\omega}_r - \dot{\omega} \end{bmatrix}.$$

407 Considering \dot{e}_2 ,

$$\begin{aligned} 408 \quad \dot{e}_2 &= \dot{v}_r - \dot{v} \\ 409 &= a_{cr} - a_h - a(x, \delta) \\ 410 &= a_{cr} - a_{des} + \hat{a}(x, \hat{\delta}) - a(x, \delta) \\ 411 &= a_{cr} - a_{pd} - a_{cr} + \bar{a}_{ad} + \hat{a}(x, \hat{\delta}) - a(x, \delta) \\ 412 &= -a_{pd} - (a(x, \delta) - \hat{a}(x, \hat{\delta}) - \bar{a}_{ad}) \\ 413 &= -a_{pd} - (\bar{\Delta}_a(x, \delta, \hat{\delta}) - \bar{a}_{ad}), \end{aligned}$$

414 \dot{e}_4 may be found similarly. Then, the overall tracking error dynamics may now be
415 expressed as

$$416 \quad \dot{e} = Ae + B \begin{bmatrix} \bar{v}_{ad} - \bar{\Delta}(x, \delta, \hat{\delta}) \end{bmatrix}, \quad (61.21)$$

417 where $\bar{\Delta}$ is given by Eq. (61.9),

$$418 \quad \bar{v}_{ad} = \begin{bmatrix} \bar{a}_{ad} \\ \bar{\alpha}_{ad} \end{bmatrix}, A = \begin{bmatrix} 0 & I & 0 & 0 \\ -R_p & -R_d & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -K_p & -K_d \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix}, \quad (61.22)$$

419 and so the linear gain matrices must be chosen such that A is Hurwitz's. Now, \bar{v}_{ad}
420 remains to be designed in order to cancel the effect of $\bar{\Delta}$.

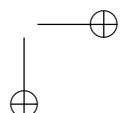
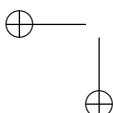
421 Note that commands, $\delta_{mdes}, \delta_{fdes}, q_{des}$, do not appear in the tracking error
422 dynamics. PCH allows adaptation to continue when the actual control signal has
423 been replaced by any arbitrary signal and thus allows switching between manual
424 and automatic flight during flight tests without any transients. Furthermore, if the
425 actuator is considered ideal and the actual position and the commanded position are
426 equal, addition of the PCH signals a_h, α_h has no effect on any system signal.

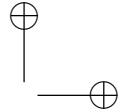
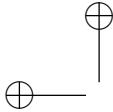
427 The adaptive signal \bar{v}_{ad} contains two terms

$$428 \quad \bar{v}_{ad} = v_{ad} + v_r = \begin{bmatrix} a_{ad} + a_r \\ \alpha_{ad} + \alpha_r \end{bmatrix},$$

AU7

429 where v_{ad} is the output of the single hidden layer (SHL) neural network (NN)
430 described in Appendix Sect. A. For an air vehicle with adaptation in all degrees





431 of freedom, $v_{ad} \in \mathcal{R}^6$, where the first three outputs, a_{ad} , approximate Δ_a and the
 432 last three outputs, α_{ad} , approximate Δ_α , is consistent with the definition of the error
 433 in Eq. (61.18). The term $v_r = [a_r^T, \alpha_r^T]^T \in \mathcal{R}^6$ is a robustifying signal that arises in
 434 the proofs of boundedness found in Kannan (2005).

435 61.2.5 Boundedness

436 Noting that the plant states are given by

$$437 \quad x(t) = x_r(t) - e(t), \quad (61.23)$$

438 boundedness of the reference model states $x_r(t)$ is sufficient to establish bound-
 439 edness of the plant states $x(t)$. However, the reference model dynamics now
 440 include the PCH signal which could be arbitrary and large. The problem of
 441 actuator saturation has effectively been moved from affecting the tracking er-
 442 ror dynamics to affecting the command tracking error dynamics of the refer-
 443 ence model. If boundedness of 61.17 can be established, then an assumption
 444 that the external command $x_c(t)$ is bounded is sufficient to establish bounded-
 445 ness of the overall system. The following assumptions are required to guarantee
 446 boundedness:

447 **Assumption 1** *The external command x_c is bounded:*

$$448 \quad \|x_c\| \leq \bar{x}_c.$$

449 **Assumption 2** *The NN approximation $\Delta(x, \hat{\delta}) = v_{ad}(x, \hat{\delta}) + \epsilon$ holds in a compact
 450 domain \mathcal{D} , which is large enough such that $\mathcal{D}_{x_c} \times \mathcal{D}_{e_r} \times \mathcal{D}_e \times \mathcal{D}_{\tilde{Z}}$ maps into \mathcal{D} . This
 451 assumption is required to leverage the universal approximation property of SHL NN
 452 (Hornik et al. 1989a).*

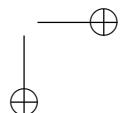
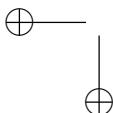
453 **Assumption 3** *The norm of the ideal weights (V^*, W^*) is bounded by a known
 454 positive value,*

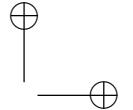
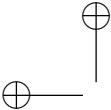
$$455 \quad 0 < \|Z^*\|_F \leq \bar{Z},$$

456 *where $\|\cdot\|_F$ denotes the Frobenius norm. This is justified due to the universal
 457 approximation property of SHL NN if the previous assumption holds (Hornik et al.
 458 1989a).*

459 **Assumption 4** *Note that Δ depends on v_{ad} through the pseudocontrol v , whereas
 460 \bar{v}_{ad} has to be designed to cancel Δ . Hence, the existence and uniqueness of a fixed-
 461 point solution for $v_{ad} = \Delta(x, v_{ad})$ is assumed. Sufficient conditions (Calise et al.
 462 2001) for this assumption are also available.*

463 **Assumption 5** *Noting that the null-controllable region of the plant \mathcal{C}_x is not
 464 necessarily a connected or closed set, assume that $\mathcal{D} \subseteq \mathcal{C}_x$ and that \mathcal{D} in addition
 465 to being compact is also convex.*





466 The adaptive element training signal, r ; adaptive element output, v_{ad} ; and
 467 robustifying term, v_r , are given by

$$468 \quad r = (e^T PB)^T$$

$$469 \quad \bar{v}_{ad} = v_{ad} + v_r$$

$$470 \quad v_{ad} = W^T \sigma(V^T \bar{x})$$

$$471 \quad v_r = -K_r (\|Z\|_F + \bar{Z}) r \frac{\|e\|}{\|r\|}. \\ 472$$

473 **Theorem 61.1.** Consider the system given by (61.1)–(61.4), with the inverse law
 474 (61.6), reference models (61.35) and (61.36) which is consistent with (61.12) and
 475 (61.13), where the gains are the same as those selected such that the system matrix
 476 in (61.21) is Hurwitz and Assumptions 1–5 are met. If $K_r > 0 \in \mathcal{R}^{k \times k}$ is chosen
 477 sufficiently large with lower limit stated in the proof, and adaptive element weights
 478 W, V satisfy the adaptation laws

$$479 \quad \dot{W} = -[(\sigma - \sigma' V^T \bar{x}) r^T + \kappa \|e\| W] \Gamma_W$$

$$480 \quad \dot{V} = -\Gamma_V [\bar{x}(r^T W^T \sigma') + \kappa \|e\| V], \quad (61.24)$$

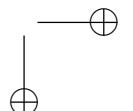
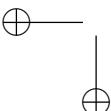
481 with $\Gamma_W, \Gamma_V > 0$, $\kappa > 0$ with lower limit stated in the proof, and the external
 482 command $x_c(t)$ is such that $e_r(t) \in \Omega(P_r, \rho)$, for some $\rho > 0$, then, the command
 483 tracking error, e_r ; the reference model tracking error, e ; and adaptive element
 484 weights (\tilde{W}, \tilde{V}) are uniformly ultimately bounded. Further, the plant states, x , are
 485 ultimately bounded.

Proof. See proof of Theorem 4 in Kannan (2005). \square

486 **Remark 61.5.** The update laws $\dot{W}(t), \dot{V}(t)$ closely resembles the backpropagation
 487 method of tuning neural network weights (Rumelhart and Williams 1986; Suykens
 488 et al. 1996; Haykin 1998; Kim and Lewis 1998). However, it is important to
 489 note that the training signal r is different from that of the backpropagation-based
 490 learning laws.

491 61.3 Concurrent Learning

492 The single hidden layer neural-network-based adaptive elements used in this chapter
 493 are known to have the universal approximation property (Haykin 1998; Hornik
 494 et al. 1989a), that is, given sufficient number of hidden layer neurons there exists
 495 a set of ideal weights W^*, V^* that brings the neural network output to within
 496 an ϵ neighborhood of the modeling error $\bar{\Delta}(x, \delta)$ (uncertainty). The adaptive
 497 laws in Eq. (61.24) are designed to minimize the instantaneous tracking error e .
 498 Although Theorem 61.1 guarantees boundedness of the tracking error e , it cannot



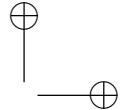
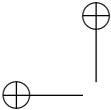
499 be guaranteed that the adaptive weights will approach the ideal weights over the
500 long term during a normal course of operation. It is useful to drive the weights
501 closer toward their ideal values, as the resulting NN representation forms a good
502 approximation of the uncertainty, which can result in improved performance and
503 can be used for planning and health-monitoring purposes.

504 One limitation of the adaptive laws in Eq. (61.24) (without the ϵ -modification
505 term) is that at any instant of time, they are constrained to search for the ideal
506 weights only in the direction of instantaneous tracking error reduction. In that
507 sense these adaptive laws are equivalent to a gradient-descent or a greedy update.
508 Therefore, the adaptive weights may not approach the ideal weights unless all
509 directions in which the weights can evolve to reduce the tracking error are
510 explored infinitely often during the course of operation. Intuitively, this explains
511 why *persistency of excitation* is required to guarantee weight convergence for most
512 adaptive laws (see, e.g., Boyd and Sastry 1986). The idea in concurrent learning is to
513 use specifically selected and online recorded data to ensure parameter convergence
514 without requiring persistent excitation. If data is recorded when the system states
515 are exciting and if invariant system properties, such as modeling error information,
516 can be inferred from the recorded data, then weight convergence can be guaranteed
517 without requiring persistent excitation (Chowdhary 2010). In an implementation of
518 a concurrent-learning adaptive controller, each measured data point is evaluated to
519 determine whether it should be added to a *history stack*. The maximum number of
520 recorded data points is limited, and when this number is reached, new data points
521 replace old points. Note that the history stack is not intended to be a buffer of last
522 p states. The approximation modeling error at a recorded data point, which is an
523 invariant system property, is inferred from the recorded data point by noting that
524 $\Delta(x_i, \delta_i) \approx \hat{x}_i - v(x_i, \delta_i)$ where \hat{x}_i is the smoothed estimate of \dot{x}_i (Chowdhary
525 and Johnson 2011b; Gelb 1974). Adaptation happens concurrently on recorded and
526 current data such that the instantaneous tracking error and the modeling error at all
527 recorded data points simultaneously reduces (Chowdhary and Johnson 2010, 2011b;
528 Chowdhary 2010).

529 It was shown in Chowdhary (2010) and Chowdhary and Johnson (2010) that for
530 linearly parameterized uncertainties, the requirement on persistency of excitation
531 can be relaxed if online recorded data is used concurrently with instantaneous data
532 for adaptation. If the uncertainty can be linearly parameterized, then

$$533 \quad \bar{\Delta}(x, \delta) = W^{*T} \phi(x, \delta) + \epsilon(x, \delta) \quad (61.25)$$

534 where $W^* \in \mathbb{R}^l$ denotes the ideal weights that guarantee for a given basis function
535 $\phi(x, \delta) \in \mathbb{R}^l$ $\sup_{\delta} \|\epsilon(x, \delta)\| \leq \bar{\epsilon}$ for some positive constant $\bar{\epsilon}$. In this case, the
536 adaptive element can also be linearly parameterized in the form $v_{ad} = W^T \phi(x, \delta)$.
537 In certain UAV applications, the basis functions for the modeling error are known
538 (see, e.g., the problem of wing-rock control (Singh et al. 1995)), in which case,
539 the existence of an unknown ideal weight vector W^* can be established such that
540 $\bar{\epsilon} = 0$. The representation in (61.25) can also be guaranteed for any continuous
541 modeling error approximated over a compact domain if elements of ϕ consist of set
542 of Gaussian radial basis functions and a scalar bias term b_w (see Park and Sandberg



543 1991; Haykin 1998). For either of these linearly parameterized representations of
 544 the uncertainty, the following theorem can be proven (Chowdhary 2010; Chowdhary
 545 and Johnson 2010, 2011a):

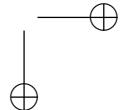
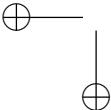
546 **Theorem 61.2.** Consider the system given by (61.1)–(61.4), with the inverse law
 547 (61.6), reference models (61.35) and (61.36) which is consistent with (61.12) and
 548 (61.13), where the gains are the same as those selected such that the system
 549 matrix in (61.21) is Hurwitz. Assume further that the uncertainty is linearly
 550 parameterizable using an appropriate set of bases over a compact domain D
 551 and that Assumptions 4 and 5 hold. For each recorded data point j , let $\epsilon_i(t) =$
 552 $W^T(t)\phi(x_i, \delta_i) - \hat{\Delta}(x_i, \delta_i)$, with $\hat{\Delta}(x_i, \delta_i) = \dot{\hat{x}}_i - v(x_i, \delta_i)$. Now consider the
 553 following update law for the weights of the RBF NN:

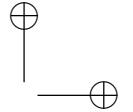
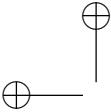
$$554 \quad \dot{W} = -\Gamma_W \sigma(z) e^T P B - \sum_{j=1}^p \Gamma_W \sigma(x_i, \delta_i) \epsilon_j^T, \quad (61.26)$$

555 and assume that $Z = [\phi(z_1), \dots, \phi(z_p)]$ and $\text{rank}(Z) = l$. Let B_α be the largest
 556 compact ball in D , and assume $\zeta(0) \in B_\alpha$, define $\delta = \max(\beta, \frac{2\|PB\|\bar{\epsilon}}{\lambda_{\min}(Q)} + \frac{p\bar{\epsilon}\sqrt{l}}{\lambda_{\min}(\Omega)})$,
 557 and assume that D is sufficiently large such that $m = \alpha - \delta$ is a positive scalar.
 558 If the states x_{rm} of the bounded-input bounded-output reference model of (61.11)
 559 remains bounded in the compact ball $B_m = \{x_{rm} : \|x_{rm}\| \leq m\}$ for all $t \geq 0$,
 560 then the tracking error e and the weight error $\tilde{W} = W - W^*$ are uniformly
 561 ultimately bounded. Furthermore, if the representation in (61.25) is exact over
 562 the entire operating domain, that is, $\bar{\epsilon} = 0$, then the tracking error and weight
 563 error converge exponentially fast to a compact ball around the origin for arbitrary
 564 initial conditions, with the rate of convergence directly proportional to the minimum
 565 singular value of the history stack matrix Z .

566 **Remark 61.6.** The size of the compact ball around the origin where the weight and
 567 tracking error converge is dependent on the representation error $\bar{\epsilon}$ and the estimation
 568 error $\check{\epsilon} = \max_i \|\dot{x}_i - \hat{x}_i\|$. The former can be reduced by choosing appropriate
 569 number of RBFs across the operating domain, and the latter can be reduced by an
 570 appropriate implementation of a fixed point smoother. Note that $\dot{\hat{x}}(t)$ is not needed
 571 at a current instant t . Therefore, an appropriate implementation of a fixed-point
 572 smoother alleviates several issues faced in estimating $\dot{\hat{x}}(t)$ by using recorded data
 573 before and after a data point is recorded to form very accurate estimates of $\dot{\hat{x}}_i$ (Gelb
 574 1974; Chowdhary and Johnson 2011b).

575 The history stack matrix $Z = [\phi(z_1), \dots, \phi(z_p)]$ is not a buffer of last p states.
 576 It can be updated online by including data points that are of significant interest
 577 over the course of operation. In the linearly parameterized case, convergence is
 578 guaranteed as soon as the history stack becomes full ranked. New data points could
 579 replace existing data points once the history stack reaches a predetermined size.
 580 It was shown in Chowdhary and Johnson (2011a) that the rate of convergence of
 581 the tracking error and weights is directly proportional to the minimum singular





582 value of Z . This provides a useful metric to determine which data points are most
 583 useful for improving convergence. Consequently, an algorithm for adding points that
 584 improve the minimum singular value of Z for the case of linearly parameterizable
 585 uncertainty was presented in Chowdhary and Johnson (2011a). The main limitation
 586 of the linearly parameterized RBF NN representation of the uncertainty is that
 587 the RBF centers need to be preallocated over an estimated compact domain of
 588 operation D . Therefore, if the system evolves outside of D , all benefits of using
 589 adaptive control are lost. This can be addressed by evolving the RBF basis to reflect
 590 the current domain of operation; a reproducing kernel Hilbert space approach for
 591 accomplishing this was presented in Kingravi et al. (2012).

592 On the other hand, the nonlinearly parameterized NN described in
 593 Appendix Sect. A is more flexible: it only requires the uncertainties to be bounded
 594 over a compact set, but does not require that the domain of operation be known.
 595 However, it is typically more difficult to analyze due to the nonlinear parameter-
 596izations. In Chowdhary and Johnson (2011b), a concurrent-learning adaptive law
 597 was proposed for SHL NN and was validated in flight on the GTMax rotorcraft (see
 598 Sect. 61.5.5). In particular, the following theorem can be proven (Chowdhary and
 599 Johnson 2011b; Chowdhary 2010):

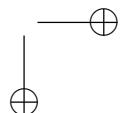
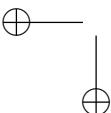
600 **Theorem 61.3.** Consider the system given by (61.1)–(61.4), with the inverse law
 601 (61.6), reference models (61.35) and (61.36) which is consistent with (61.12) and
 602 (61.13), where the gains are the same as those selected such that the system matrix
 603 in (61.21) is Hurwitz and Assumptions 1–5 are met. Let $i \in \aleph$ denote the index of an
 604 online recorded data point z_i , define $r_{b_i}(t) = v_{ad}(z_i) - \hat{\Delta}(z_i)$, where $\hat{\Delta}(z) = \dot{\hat{x}}_i - v_i$
 605 and $\dot{\hat{x}}_i$ is the smoothed estimate of \dot{x}_i , and consider the following adaptive law:

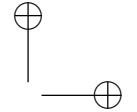
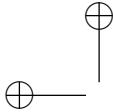
$$606 \quad \dot{W}(t) = -(\sigma(V^T(t)\bar{x}(t)) - \sigma'(V^T(t)\bar{x}(t))V^T(t)\bar{x}(t))r^T(t)\Gamma_w - k\|e(t)\|W(t) \\ 607 \quad - W_c(t) \sum_{i=1}^p (\sigma(V^T(t)\bar{x}_i) - \sigma'(V^T(t)\bar{x}_i)V^T(t)\bar{x}_i)r_{b_i}^T(t)\Gamma_w, \quad (61.27)$$

$$608 \quad \dot{V}(t) = -\Gamma_V \bar{x}(t)r^T(t)W^T(t)\sigma'(V^T(t)\bar{x}(t)) - k\|e(t)\|V(t) - \\ 609 \quad V_c(t) \sum_{i=1}^p \Gamma_V \bar{x}_i r_{b_i}^T(t)W^T(t)\sigma'(V^T(t)\bar{x}_i), \quad (61.28)$$

611 where W_c , V_c are orthogonal projection operators that restrict the update based on
 612 the recorded data in the null-space of update based on current data:

$$613 \quad W_c = I - \frac{(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})^T}{(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})^T(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})}, \\ 614 \quad V_c = I - \frac{\Gamma_V \bar{x} \bar{x}^T \Gamma_V}{\bar{x}^T \Gamma_V \Gamma_V \bar{x}}, \quad (61.29)$$





615 with $\Gamma_W, \Gamma_V > 0$, $\kappa > 0$ with lower limit stated in the proof, and the external
 616 command $x_c(t)$ is such that $e_r(t) \in \Omega(P_r, \rho)$, for some $\rho > 0$, then, the command
 617 tracking error, e_r ; the reference model tracking error, e ; and adaptive element
 618 weights (\tilde{W}, \tilde{V}) are uniformly ultimately bounded. Further, the plant states, x , are
 619 ultimately bounded.

620 For the nonlinearly parameterized neural network, the simplest way to record a
 621 data point $x(t)$ online is to ensure that for a given $\bar{\theta} \in \Re^+$,

$$622 \quad \frac{\|x(t) - x_k\|^2}{\|x(t)\|} \geq \bar{\theta}, \quad (61.30)$$

623 where x_k is the last recorded data point. The points can be stored in an online
 624 history stack which contains a maximum of \bar{p} points. Once the maximum number
 625 of recorded points are reached, points are added such that the newest point replaces
 626 the oldest one.

627 61.4 Helicopter Specific Design

628 Consider the application of the combined inner-outer-loop adaptive architecture
 629 to the trajectory control of a helicopter. The dynamics (Munzinger 1998; Mettler
 630 2002; Gavrilets et al. 2001) of the helicopter may be modeled in the same form
 631 as Eqs. (61.1)–(61.4). Most small helicopters include a Bell-Hiller stabilizer bar,
 632 which provides lagged rate feedback and is a source of unmodeled dynamics.
 633 The nonlinear model used for simulation in this work included the stabilizer bar
 634 dynamics. Additionally, blade flapping and other aspects such as gear and engine
 635 dynamics were also modeled.

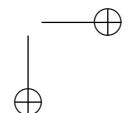
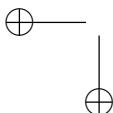
636 61.4.1 Approximate Model

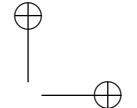
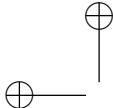
637 An approximate model for the attitude dynamics of the helicopter was generated by
 638 linearizing the nonlinear model around hover and neglecting coupling between the
 639 attitude and translational dynamics as well as the stabilizer bar:

$$640 \quad \alpha_{des} = \hat{A}_1 \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \hat{A}_2 \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \hat{B} \left(\underbrace{\begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix}}_{des} - \underbrace{\begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix}}_{trim} \right), \quad (61.31)$$

641 or

$$643 \quad \alpha_{des} = \hat{A}_1 \omega_B + \hat{A}_2 v_B + \hat{B} (\delta_{m_{des}} - \delta_{m_{trim}}),$$





644 where \hat{A}_1 and \hat{A}_2 represent the attitude and translational dynamics, respectively, and
 645 ω_B represents the angular velocity of the body with respect to the earth expressed
 646 in the body frame. The body velocity vector with respect to the earth expressed in
 647 the body frame is given by v_B , and $\delta_{m_{trim}}$ is the trim control vector that is consistent
 648 with the linear model. Choosing the control matrix \hat{B} such that it is invertible, the
 649 moment controls may be evaluated as

$$650 \quad \delta_{m_{des}} = \hat{B}^{-1}(\alpha_{des} - \hat{A}_1\omega_B - \hat{A}_2v_B) + \delta_{m_{trim}}.$$

651 The translational dynamics may be modeled as a point mass with a thrust vector
 652 that may be oriented in a given direction as illustrated in Fig. 61.7. More involved
 653 inverses (Lipp and Prasad 1993) may be used, but the simple relationships between
 654 thrust, attitude, and accelerations suffice when used with adaptation

$$655 \quad a_{des} = \begin{bmatrix} 0 \\ 0 \\ Z_{\delta_{coll}} \end{bmatrix} (\delta_{coll_{des}} - \delta_{coll_{trim}}) + L_B v g, \quad (61.32)$$

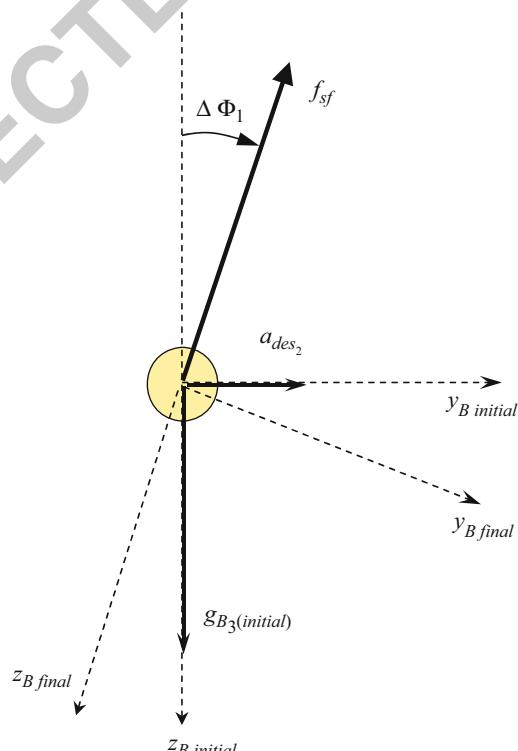
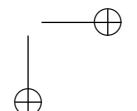
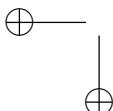
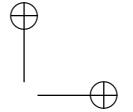
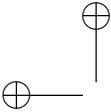


Fig. 61.7 Point mass model
for outerloop inversion





656 where $Z_{\delta_{coll}}$ is the control derivative for acceleration in the vertical axis. L_{bv} is the
 657 direction cosine matrix that transforms a vector from the vehicle (or local) frame to
 658 the body frame and g is an assumed gravity vector. The desired specific force along
 659 the body z axis may be evaluated as

$$660 \quad f_{sf} = (a_{des} - L_{bv}g)_3.$$

661 The required collective input may be evaluated as

$$662 \quad \delta_{coll_{des}} = \frac{f_{sf}}{Z_{\delta_{coll}}} + \delta_{coll_{trim}}.$$

663 The attitude augmentation required in order to orient the thrust vector to attain the
 664 desired translational accelerations are given by the following small angle corrections
 665 from the current reference body attitude and attitude command:

$$666 \quad \Delta\Phi_1 = \frac{a_{des_2}}{f_{sf}}, \quad \Delta\Phi_2 = -\frac{a_{des_1}}{f_{sf}}, \quad \Delta\Phi_3 = 0, \quad (61.33)$$

668 For this simplified helicopter model, heading change has no effect on accelerations
 669 in the x, y plane, and hence $\Delta\Phi_3 = 0$. These three correction angles may now be
 670 used to generate the attitude quaternion correction desired by the outer loop. Thus,

$$671 \quad q_{des} = q(\Delta\Phi_1, \Delta\Phi_2, \Delta\Phi_3), \quad (61.34)$$

672 where $q(\cdot)$ is a function (Stevens and Lewis 2003a) that expresses an Euler-angle-
 673 based rotation as a quaternion. The overall detailed controller architecture is shown
 674 in Fig. 61.6.

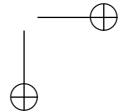
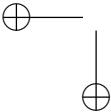
675 *Remark 61.7.* If the desired specific force f_{sf} is close to zero, which occurs when
 676 the desired acceleration in the body z axis is the same as the component of gravity
 677 vector along that axis, then, Eq.(61.33) is undefined. To overcome this problem,
 678 one can impose a restriction where 61.33 is only computed if $|f_{sf}| > \tilde{f}_{sf}$, where
 679 $\tilde{f}_{sf} > 0$ and is a lower limit. Essentially it means do not bother using attitude unless
 680 the desired specific force is greater than \tilde{f}_{sf} .

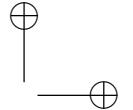
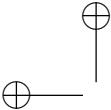
681 61.4.2 Reference Model

682 Using a linear a_{cr} and α_{cr} in Eqs.(61.12) and (61.13) results in the following
 683 reference model dynamics:

$$684 \quad \dot{v}_r = R_p(p_c - p_r) + R_d(v_c - v_r) - a_h$$

$$685 \quad \dot{\omega}_r = K_p(\tilde{Q}(q_c \oplus q_{des}, q_r)) + K_d(\omega_c - \omega_r) - \alpha_h,$$





686 where R_p, R_d, K_p, K_d are the same gains used for the PD compensator in
 687 Eq. (61.20). If limits on the angular rate or translational velocities are to be imposed,
 688 then they may be easily included in the reference model dynamics by choosing the
 689 following constrained linear reference for a_{cr} and α_{cr} :

$$690 \quad a_{cr} = R_d[v_c - v_r + \sigma(R_d^{-1}R_p(p_c - p_r), v_{lim})], \quad (61.35)$$

$$691 \quad \alpha_{cr} = K_d[\omega_c - \omega_r + \sigma(K_d^{-1}K_p\tilde{Q}(q_c \oplus q_{des}, q_r), \omega_{lim})]. \quad (61.36)$$

692 This reference model has prescribable aggressiveness, where $\sigma(\cdot)$ is a saturation
 693 function and v_{lim}, ω_{lim} are the translational and angular rate limits, respectively.

694 *Remark 61.8.* Note that there are no limits placed on the externally commanded
 695 position, velocity, angular rate, or attitude. For example, in the translational
 696 reference model, if a large position step is commanded, $p_c = [1,000, 0, 0]^T$ ft and
 697 $v_c = [0, 0, 0]^T$ ft/s, the speed at which this large step will be achieved is v_{lim} . On
 698 the other hand if $p_c = \int v_c dt$ and $v_c = [60, 0, 0]^T$ ft/s, the speed of the vehicle will
 699 be 60 ft/s. Similarly, ω_{lim} dictates how fast large attitude errors will be corrected.
 700 Additionally, aggressiveness with which translational accelerations will be pursued
 701 by tilting the body may be governed by limiting the magnitude of q_{des} to the scalar
 702 limit q_{lim} .

703 61.4.3 Choice of Gain Linear Dynamics

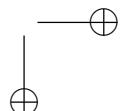
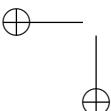
704 When the combined adaptive inner-outer-loop controller for position and attitude
 705 control is implemented, the poles for the combined error dynamics must be selected
 706 appropriately. The following analysis applies to the situation where inversion model
 707 error is compensated for accurately by the NN, and it is assumed that the system is
 708 exactly feedback linearized. The inner loop and outer loop each represent a second-
 709 order system, and the resulting position dynamics $p(s)/p_c(s)$ are fourth order in
 710 directions perpendicular to the rotor spin axis.

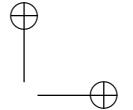
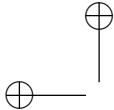
711 When the closed-loop longitudinal dynamics, near hover, are considered, and
 712 with an acknowledgment of an abuse of notation, it may be written as

$$713 \quad \ddot{x} = a_{des} = \ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x), \quad (61.37)$$

$$714 \quad \ddot{\theta} = \alpha_{des} = \ddot{\theta}_g + K_d(\dot{\theta}_g - \dot{\theta}) + K_p(\theta_g - \theta), \quad (61.38)$$

716 where R_p, R_d, K_p , and K_d are the PD compensator gains for the inner loop (pitch
 717 angle) and outer loop (fore-aft position). Now x is the position, θ the attitude, and θ_g
 718 the attitude command. Normally, $\theta_g = \theta_c + \theta_{des}$ where θ_c is the external command
 719 and θ_{des} the outer-loop-generated attitude command. Here, it is assumed that the
 720 external attitude command and its derivatives are zero; hence, $\theta_g = \theta_{des}$. In the
 721 following development, the transfer function $x(s)/x_c(s)$ is found and used to place





722 the poles of the combined inner-outer-loop system in terms of the PD compensator
723 gains.

724 When contributions of $\dot{\theta}_g(s)$ and $\ddot{\theta}_g(s)$, are ignored, the pitch dynamics Eq. 61.38
725 may be rewritten in the form of a transfer function as

$$726 \quad \theta(s) = \frac{\theta(s)}{\theta_g(s)} \theta_g(s) = \frac{K_p}{s^2 + K_d s + K_p} \theta_g(s). \quad (61.39)$$

727 If the outer-loop linearizing transformation used to arrive at Eq. 61.37 has the form
728 $\ddot{x} = f\theta$, where $f = -g$ and g is gravity, it may be written as

$$729 \quad s^2 x(s) = f\theta(s). \quad (61.40)$$

730 The outer-loop attitude command may be generated as

$$731 \quad \theta_{des} = \frac{\ddot{x}_{des}}{f} = \frac{a_{des}}{f}. \quad (61.41)$$

732 Note that $\theta_g = \theta_{des}$; if $\theta_c = 0$,

$$733 \quad \theta_g = \theta_{des} = \frac{1}{f} [\ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x)]. \quad (61.42)$$

735 When Eqs. (61.39) and (61.42) are used in Eq. 61.40,

$$736 \quad s^2 x(s) = \frac{K_p [s^2 x_c + R_d s (x_c - x) + R_p (x_c - x)]}{s^2 + K_d s + K_p}. \quad (61.43)$$

737 Rearranging the above equation results in the following transfer function:

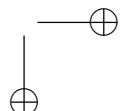
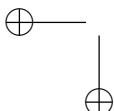
$$738 \quad \frac{x(s)}{x_c(s)} = \frac{K_p s^2 + K_p R_d s + K_p R_p}{s^4 + K_d s^3 + K_p s^2 + K_p R_d s + K_p R_p}. \quad (61.44)$$

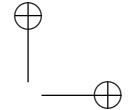
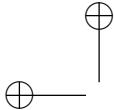
739 One way to choose the gains is by examining a fourth-order characteristic
740 polynomial written as the product of two second-order systems

$$741 \quad \Upsilon(s) = (s^2 + 2\zeta_o \omega_o + \omega_o^2)(s^2 + 2\zeta_i \omega_i + \omega_i^2) \\ 742 \quad = s^4 + (2\zeta_i \omega_i + 2\zeta_o \omega_o)s^3 \\ 743 \quad + (\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2)s^2 + (2\zeta_o \omega_o \omega_i^2 + 2\omega_o^2 \zeta_i \omega_i)s + \omega_o^2 \omega_i^2, \quad (61.45)$$

744 where the subscripts i, o represent the inner- and outer-loop values, respectively.

745 Comparing the coefficients of the poles of Eqs. (61.44) and (61.45) allows the
746 gains to be expressed as a function of the desired pole locations for each axis in turn





$$\begin{aligned}
 747 \quad R_p &= \frac{\omega_o^2 \omega_i^2}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \\
 748 \quad R_d &= 2 \frac{\omega_o \omega_i (\zeta_o \omega_i + \omega_o \zeta_i)}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \\
 749 \quad K_p &= \omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2 \\
 750 \quad K_d &= 2\zeta_i \omega_i + 2\zeta_o \omega_o. \tag{61.46}
 \end{aligned}$$

752 Additionally, the zeros of the transfer function given by Eq. 61.44 affect the transient
 753 response. Thus, $\omega_i, \zeta_i, \omega_o, \zeta_o$ must be selected such that performance is acceptable.

754 61.4.4 Imposing Response Characteristics

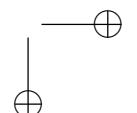
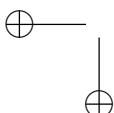
755 The methods presented in this chapter do not contain assumptions that limit its
 756 application to unmanned helicopters. Manned rotorcraft normally have to meet
 757 standards, such as those specified in the Aeronautical Design Standard-33 (2000)
 758 handling qualities specifications. Control system performance (Civita et al. 2002b;
 759 Rysdyk and Calise 2005) may be evaluated by imposing response requirements
 760 and computing metrics prescribed in the ADS-33. When there is no saturation, the
 761 hedging signals a_h, α_h are zero. When it is assumed that the adaptation has reached
 762 its ideal values of (V^*, W^*) , then

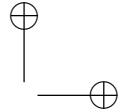
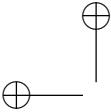
$$\begin{aligned}
 763 \quad \dot{v} &= a_{cr} + a_{pd} + \epsilon_a \\
 764 \quad \dot{\omega} &= \alpha_{cr} + \alpha_{pd} + \epsilon_\alpha,
 \end{aligned}$$

765 where ϵ_a and ϵ_α are bounded by $\bar{\epsilon}$. Additionally, the Lyapunov analysis provides
 766 guaranteed model following, which implies a_{pd} and α_{pd} are small. Thus, $\dot{v} \approx a_{cr}$
 767 and $\dot{\omega} \approx \alpha_{cr}$. Hence, as long as the preceding assumptions are valid over the
 768 bandwidth of interest, the desired response characteristics may be encoded into the
 769 reference model a_{cr} and α_{cr} .

770 61.5 Experimental Results

771 The proposed guidance and control architecture was applied to the Georgia Institute
 772 of Technology Yamaha R-Max helicopter (GTMax) shown in Fig. 61.2. The GTMax
 773 helicopter weighs about 157 lb and has a main rotor radius of 5.05 ft. Nominal
 774 rotor speed is 850 revolutions per minute. Its practical payload capability is about
 775 66 lb with a flight endurance of greater than 60 min. It is also equipped with
 776 a Bell-Hillier stabilizer bar. Its avionics package includes a Pentium 266 flight
 777 control computer, an inertial measurement unit (IMU), a global positioning system,
 778 a 3-axis magnetometer, and a sonar altimeter. The control laws presented in this



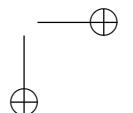
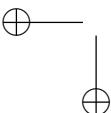


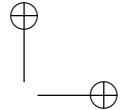
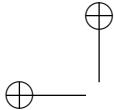
779 chapter were first implemented in simulation (Kannan et al. 2004) using a nonlinear
780 helicopter model that included flapping and stabilizer bar dynamics. Wind and gust
781 models were also included. Additionally, models of sensors with associated noise
782 characteristics were implemented. Many aspects of hardware such as the output
783 of sensor model data as serial packets was simulated. This introduced digitization
784 errors as would exist in real life and also allowed testing of many flight-specific
785 components such as sensor drivers. The navigation system (Christophersen et al.
786 2006) consists of a 17-state Kalman filter to estimate variables such as attitude and
787 terrain altitude. The navigation filter was executed at 100 Hz and corresponds to
788 the highest rate at which the IMU is able to provide data. Controller calculations
789 occurred at 50 Hz. The control laws were first implemented as C-code and tested in
790 simulation. Because almost all aspects specific to flight testing were included in the
791 simulation environment, a subset of the code from the simulation environment was
792 implemented on the main flight computer. During flight, Ethernet and serial-based
793 data links provided a link to the ground station computer that allowed monitor-
794 ing and uploading of waypoints. A simple kinematics-based trajectory generator
795 (with limits on accelerations) was used to generate smooth consistent trajectories
796 (p_c, v_c, q_c, ω_c) for the controller. Various moderately aggressive maneuvers were
797 performed during flight to test the performance of the trajectory tracking controller.
798 Controller testing began with simple hover followed by step responses and waypoint
799 navigation. Following initial flight tests, aggressiveness of the trajectory was
800 increased by relaxing acceleration limits in the trajectory generator and relaxing
801 ω_{lim} and v_{lim} in the reference models. Tracking error performance was increased
802 by increasing the desired bandwidth of the controllers. Selected results from these
803 flight tests are provided in the following sections.

804 61.5.1 Parameter Selections

805 The controller parameters for the inner loop involved choosing K_p, K_d based on a
806 natural frequency of 2.5, 2, 3 rad/s for the roll, pitch, and yaw channels, respectively,
807 and damping ratio of 1.0. For the outer loop, R_p, R_d were chosen based on a
808 natural frequency of 2, 2.5, 3 rad/s for the x, y, and z body axes, all with a damping
809 ratio of unity. The NN was chosen to have five hidden-layer neurons. The inputs
810 to the network included body axis velocities and rates as well as the estimated
811 pseudocontrols, that is, $x_{in} = [v_B^T, \omega_B^T, \hat{a}^T, \hat{\alpha}^T]$. The output layer learning rates
812 Γ_W were set to unity for all channels, and a learning rate of $\Gamma_V = 10$ was set
813 for all inputs. Limits on maximum translation rate and angular rate in the reference
814 model dynamics were set to $v_{lim} = 10$ ft/s and $\omega_{lim} = 2$ rad/s. Additionally, attitude
815 corrections from the outer loop, q_{des} were limited to 30°.

816 With regard to actuator magnitude limits, the helicopter has a radio-control
817 transmitter that the pilot may use to fly the vehicle manually. The full deflections
818 available on the transmitter sticks in each of the channels were mapped as
819 $\delta_{lat}, \delta_{lon}, \delta_{ped} \in [-1, 1]$, corresponding to the full range of lateral tilt and longi-
820 tudinal tilt of the swash plate and full range of tail rotor blade pitch. The collective





821 was mapped as $\delta_{coll} \in [-2.5, 1]$, corresponding to the full range of main rotor blade
 822 pitch available to the human pilot. The dynamic characteristics of the actuators were
 823 not investigated in detail. Instead, conservative rate limits were artificially imposed
 824 in software. Noting that $\delta = [\delta_{coll}, \delta_{lat}, \delta_{lon}, \delta_{ped}]^T$, the actuator model used for
 825 PCH purposes as well as artificially limiting the controller output has form

$$826 \quad \dot{\hat{\delta}} = \lim_{\lambda \rightarrow +\infty} \sigma \left(\lambda(\sigma(\delta_{des}, \delta_{min}, \delta_{max}) - \hat{\delta}), \dot{\delta}_{min}, \dot{\delta}_{max} \right), \quad (61.47)$$

827 where $\hat{\delta}$ is limited to lie in the interval $[\delta_{min}, \delta_{max}]$. The discrete implementation has
 828 the form

$$829 \quad \hat{\delta}[k+1] = \sigma \left(\hat{\delta}[k] + \sigma \left(\sigma(\delta_{des}, \delta_{min}, \delta_{max}) - \hat{\delta}[k], \Delta T \dot{\delta}_{min}, \Delta T \dot{\delta}_{max} \right), \right. \\ 830 \quad \left. \delta_{min}, \delta_{max} \right), \quad (61.48)$$

831 where ΔT is the sampling time. The magnitude limits were set to

$$832 \quad \delta_{min} = [-2.5, -1, -1, -1]^T \\ 833 \quad \delta_{max} = [1, 1, 1, 1]^T \quad (61.49)$$

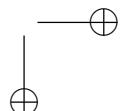
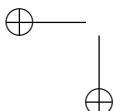
834 units, and the rate limits were set to

$$835 \quad \dot{\delta}_{min} = [-4, -2, -2, -2]^T \\ 836 \quad \dot{\delta}_{max} = [4, 2, 2, 2]^T \quad (61.50)$$

837 units per second.

838 61.5.2 Flight Test

839 Finally, the controller was flight tested on the GTMax helicopter shown in Fig. 61.2.
 840 A lateral-position step response is shown in Fig. 61.8. The vehicle heading was
 841 regulated due north during this maneuver. Lateral control deflections during the
 842 maneuver were recorded and are also shown. A step heading command response
 843 and pedal control history is shown in Fig. 61.9. It should be noted that during
 844 flight tests, states were sampled at varying rates in order to conserve memory
 845 and data link bandwidth. The trajectory commands p_c, v_c, q_c, ω_c were sampled
 846 at 1 Hz; actuator deflections $\delta_{coll}, \delta_{lon}, \delta_{lat}$, and δ_{ped} were sampled at 50 Hz; and
 847 vehicle position and speed was sampled at 50 Hz. Since the command vector
 848 is sampled at a low rate (1 Hz), a step command appears as a fast ramp in
 849 figures.



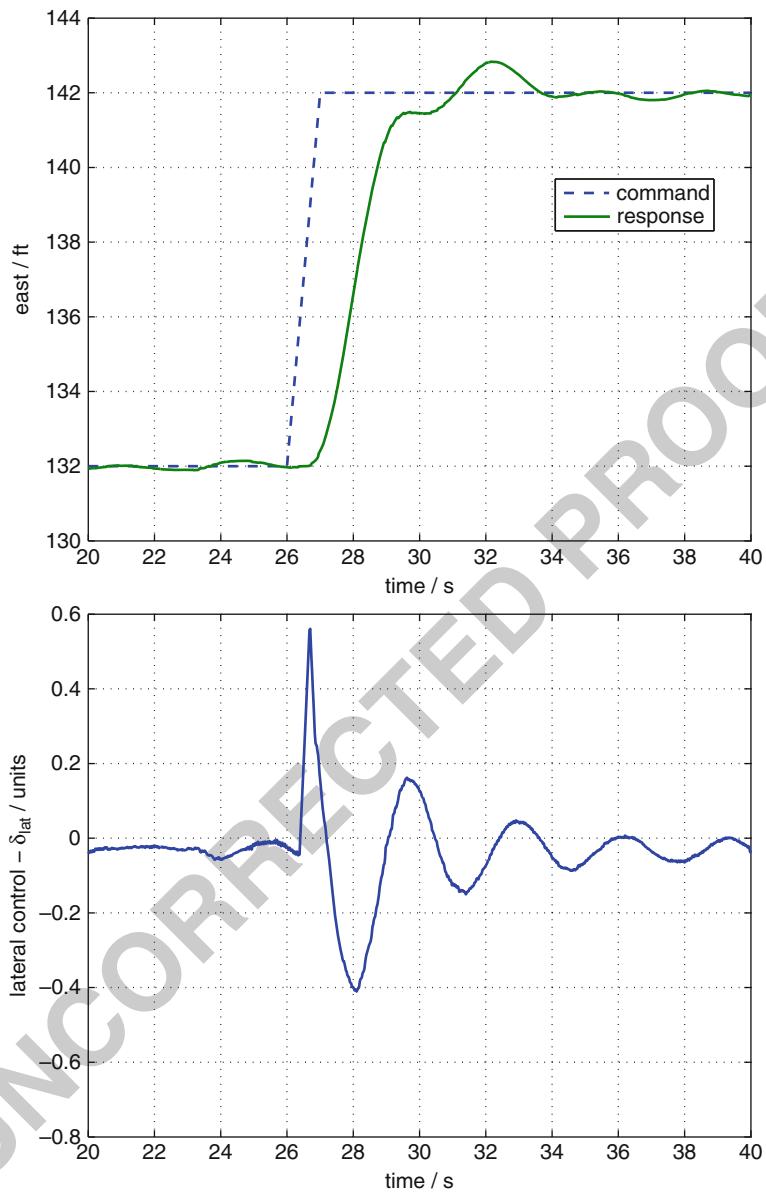
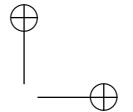
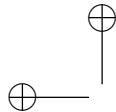
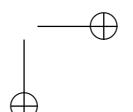
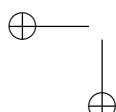


Fig. 61.8 Response to a 20-ft step in the lateral direction



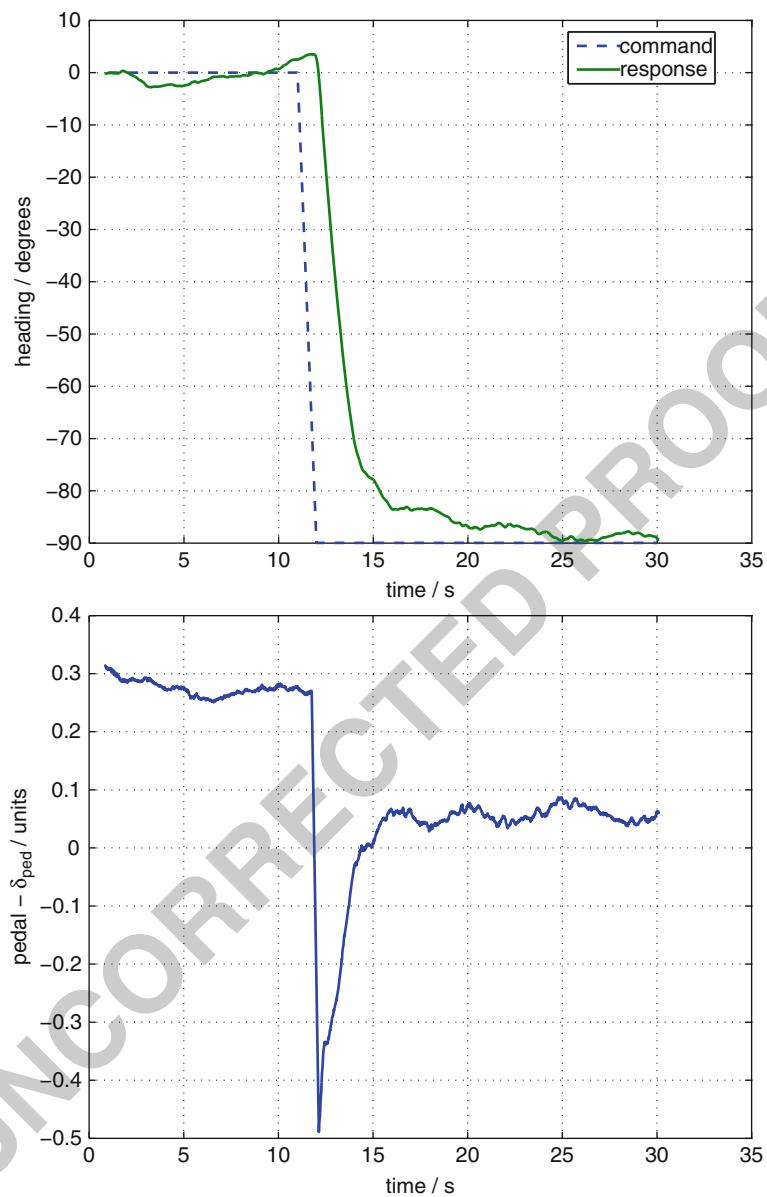
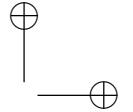
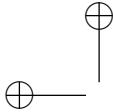


Fig. 61.9 Response to a 90° heading command



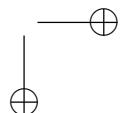
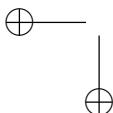
850 During takeoff and landing phases, a range sensor (sonar) is used to maintain
 851 and update the estimated local terrain altitude in the navigation system. The sonar
 852 is valid up to 8 ft above the terrain, sufficient for landing and takeoff purposes.
 853 [Figure 61.10](#) illustrates the altitude and collective profile during a landing. The
 854 vehicle starts at an initial hover at 300 ft, followed by a descent at 7 ft/s until the
 855 vehicle is 15 ft above the estimated terrain. The vehicle then descends at 0.5 ft/s
 856 until weight on skids is automatically detected at which point the collective is
 857 slowly ramped down. Automatic takeoff ([Fig. 61.11](#)) is similar where the collective
 858 is slowly ramped up until weight on skids is no longer detected. It should be noted
 859 that NN adaptation is active at all times except when weight on skids is active.
 860 Additionally, when weight is on skids, the collective ramp-up during takeoff and
 861 ramp-down during landing is open loop.

862 The approximate model used to compute the dynamic inverse (Eqs. [\(61.32\)](#) and
 863 [\(61.31\)](#)) is based on a linear model of the dynamics in hover. To evaluate controller
 864 performance at different points of the envelope, the vehicle was commanded to
 865 track a trajectory that accelerated up to a speed of 100 ft/s. To account for wind, an
 866 upwind and downwind legs were flown. In the upwind leg, the vehicle accelerated
 867 up to 80 ft/s, and during the downwind leg, the vehicle accelerated up to a speed
 868 of 97 ft/s as shown in [Fig. 61.12](#). Collective and longitudinal control deflections are
 869 also shown. In the upwind leg, the collective is saturated and the vehicle is unable
 870 to accelerate further. The longitudinal control deflections behave nominally as the
 871 vehicle accelerates and decelerates through a wide range of the envelope. The NN
 872 is able to adapt to rapidly changing flight conditions, from the baseline inverting
 873 design at hover through to the maximum speed of the aircraft. A conventional
 874 proportional-integral-derivative design would have required scheduling of gains
 875 throughout the speed range. More significantly, classical design would require
 876 accurate models at each point, unlike this design which does not. In addition to flight
 877 at high speeds, tracking performance was evaluated at moderate speeds, where a
 878 square pattern was flown at 30 ft/s for which position tracking is shown in [Fig. 61.13](#).
 879 External command position tracking errors are shown in [Fig. 61.14](#) with a peak total
 880 position error 3.3 ft and standard deviation of 0.8 ft.

881 Many maneuvers such as high-speed flight are quasi steady, in the sense that
 882 once in the maneuver, control deflection changes are only necessary for disturbance
 883 rejection. To evaluate performance where the controls have to vary significantly in
 884 order to track the commanded trajectory, the helicopter was commanded to perform
 885 a circular maneuver in the north-east plane with constant altitude and a constantly
 886 changing heading. The trajectory equations for this maneuver are given by

$$887 \quad p_c = \begin{bmatrix} \frac{V}{\omega} \cos(\omega t) \\ \frac{V}{\omega} \sin(\omega t) \\ -h \end{bmatrix}, \quad v_c = \begin{bmatrix} -V \sin(\omega t) \\ V \cos(\omega t) \\ 0 \end{bmatrix},$$

888 $\psi_c = \omega t f,$



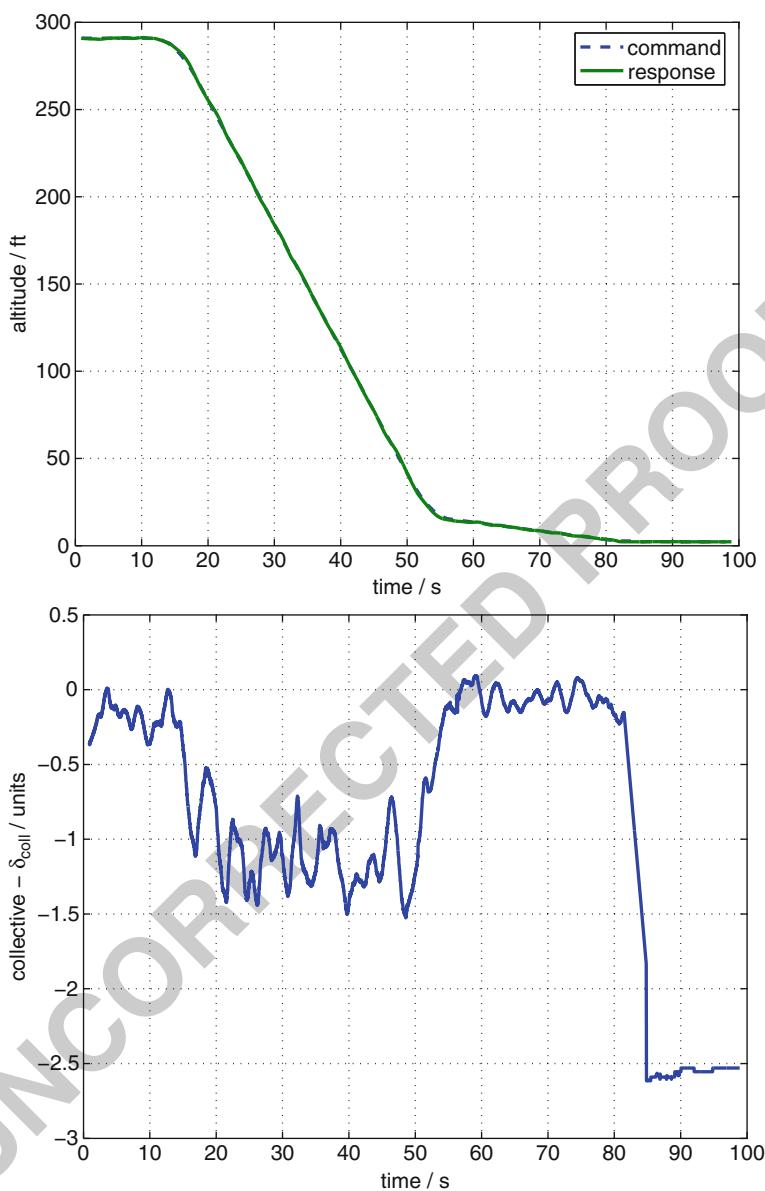


Fig. 61.10 Automatic landing maneuver

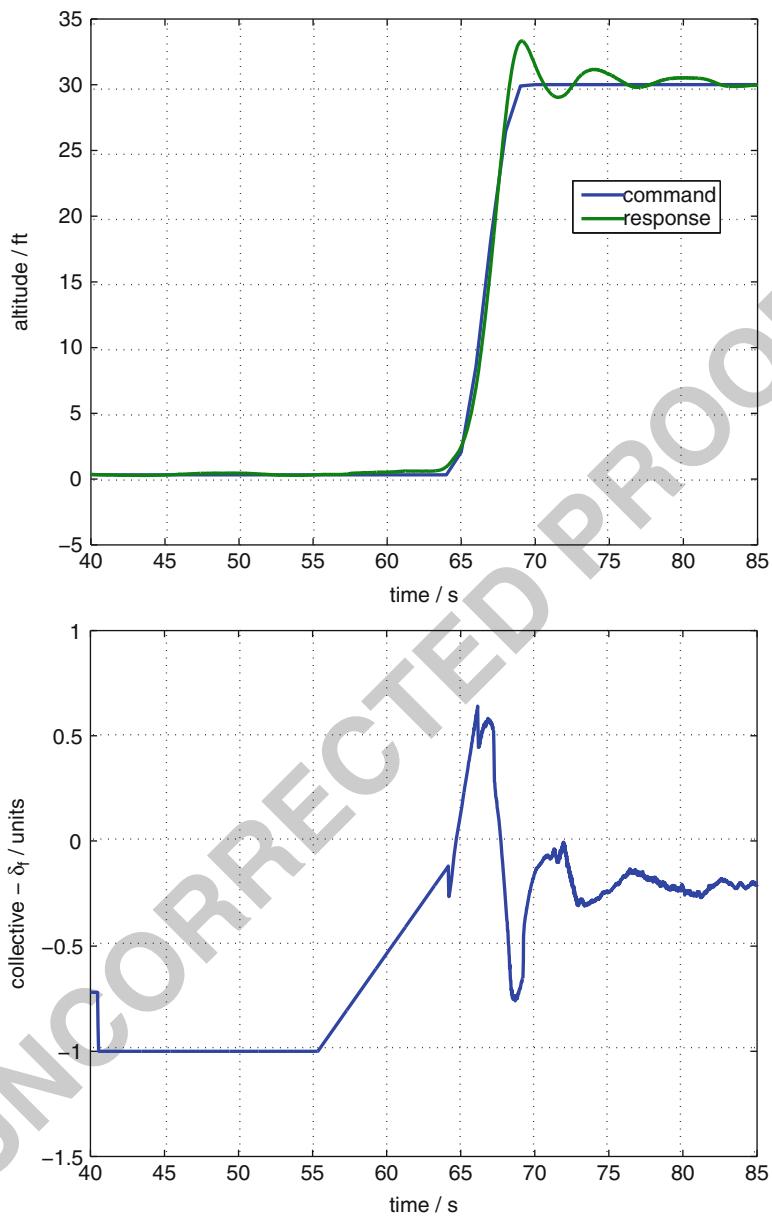


Fig. 61.11 Automatic takeoff maneuver

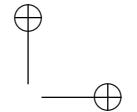
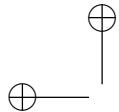
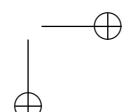
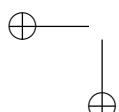
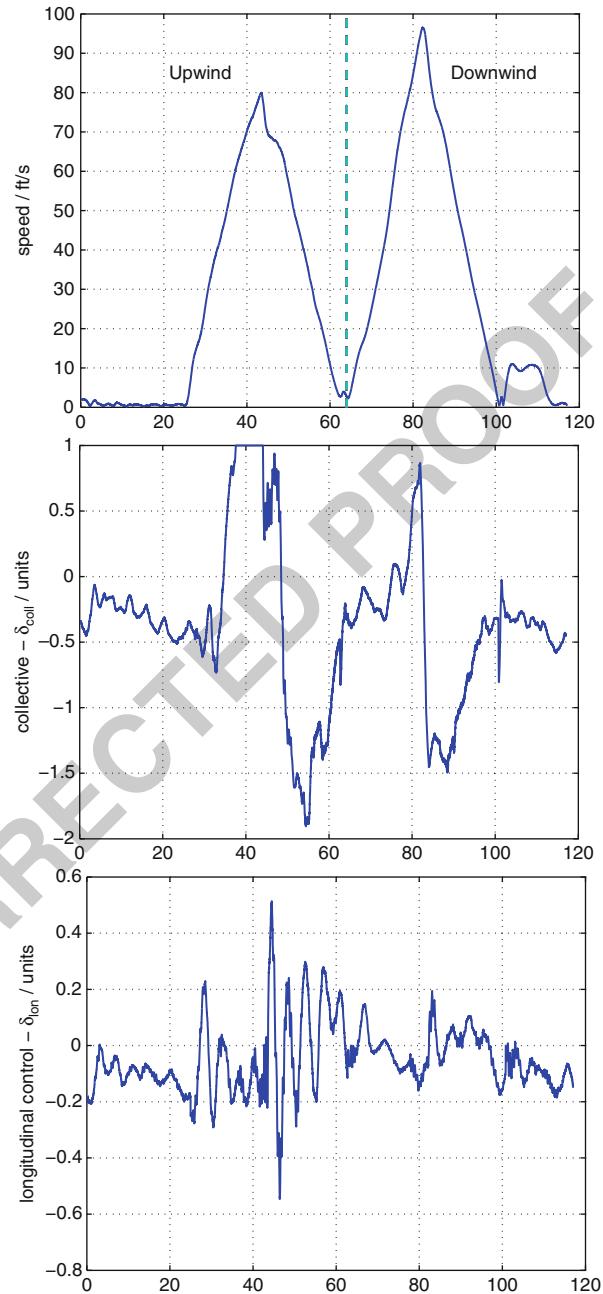


Fig. 61.12 High-speed forward flight up to 97 ft/s



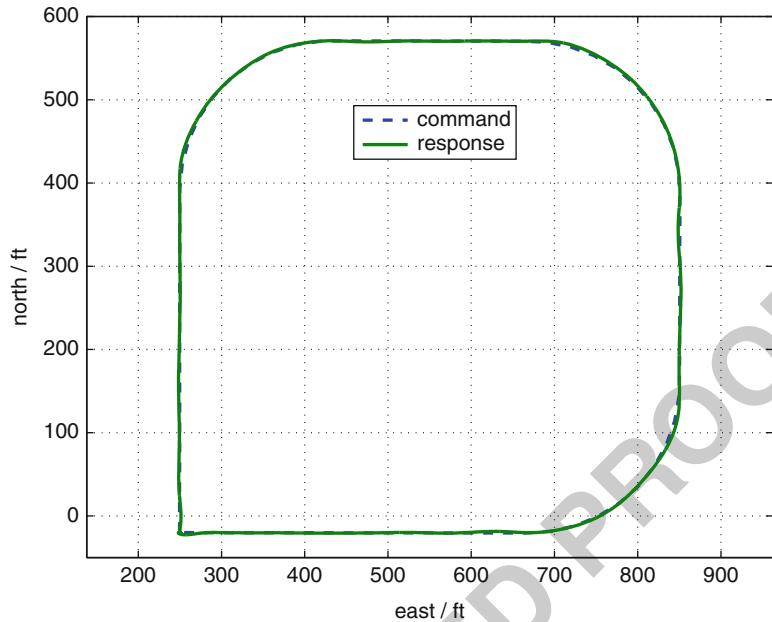


Fig. 61.13 Flying a square pattern at 30 ft/s

889 where t is current time and h is a constant altitude command. V is speed of
 890 the maneuver, ω is angular speed of the helicopter around the maneuver origin,
 891 and f is number of 360° changes in heading to be performed per circuit. If
 892 $\omega = \pi/2$ rad/s, the helicopter will complete the circular circuit once every 4 s. If
 893 $f = 1$, the helicopter will rotate anticlockwise 360° once per circuit. [Figure 61.15](#)
 894 shows the response to such a trajectory with parameters $\omega = 0.5$ rad/s, $f = 1$,
 895 $V = 10$ ft/s. After the initial transition into the circular maneuver, the tracking
 896 is seen to be within 5 ft. To visualize the maneuver easily, superimposed still
 897 images of the vehicle during the circular maneuver are shown. Both anticlockwise
 898 and clockwise heading changes during the maneuver were tested by changing the
 899 parameter from $f = 1$ (anticlockwise) to $f = -1$ (clockwise) at $t = 55$ s.
 900 [Figure 61.16](#) shows that heading tracking is good in both cases. The time history
 901 of the pedal input δ_{ped} and all other controls during the maneuver is also shown and
 902 illustrates how the vehicle has to exercise all of its controls during this maneuver.

903 Next, the ability of the controller to track a previous manually flown maneuver
 904 was tested. First, a human pilot flew a figure-eight, three-dimensional pattern with
 905 the vehicle. Vehicle state was recorded and was then played back as commands
 906 to the adaptive controller. A 3D plot of the pilot and controller flown trajectories
 907 are shown in [Fig. 61.17](#) along with projected ground track. Overall, the tracking in
 908 position was measured to be within 11.3 ft of the desired pilot-flown trajectory with
 909 a standard deviation of 4.7 ft.

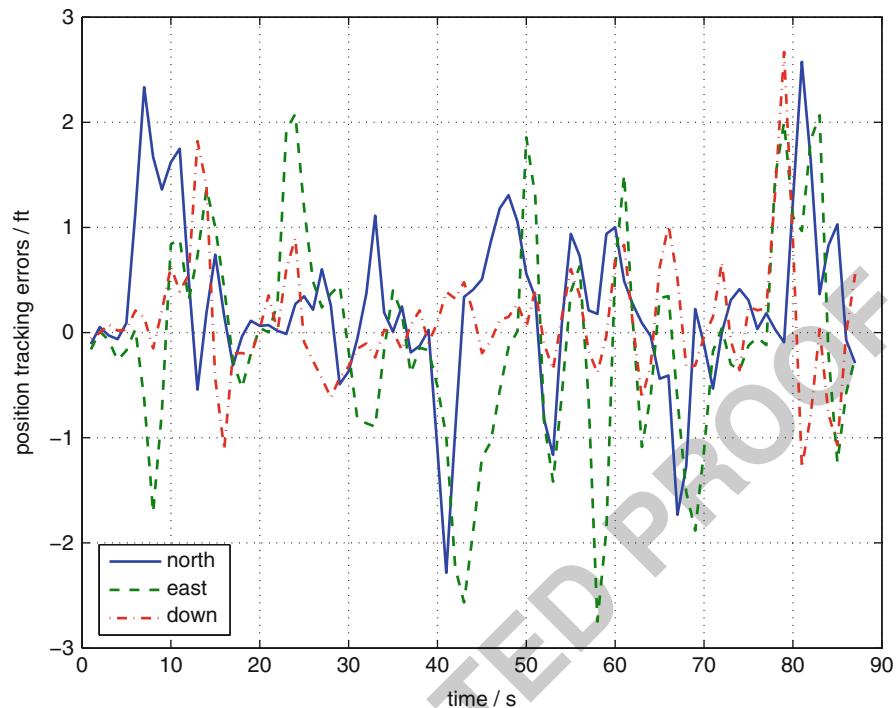


Fig. 61.14 Command tracking errors while flying a square pattern at 30 ft/s

Finally, a tactically useful maneuver was flown to test controller performance at high speeds and pitch attitudes. The objective of the maneuver is to make a 180-degree velocity change from a forward flight condition of 70 ft/s north to a 70 ft/s forward flight going south. The trajectory command and response in the north-altitude plane is shown in Fig. 61.18 along with the pitch angle. A time history of the altitude and the collective control deflection is shown in Fig. 61.19. During the maneuver, the helicopter is commanded to increase altitude by up to 50 ft in order to minimize saturation of the down collective. In the deceleration phase, the vehicle is able to track the command trajectory well; however, in accelerating to 70 ft/s going south, tracking performance suffers. In both the acceleration and deceleration phases, poor tracking corresponds with saturation of the collective control. The oscillations in altitude in Fig. 61.19 are expected and are due to control saturation which limits the vehicle’s descent rate. The large pitch attitudes experienced are what the outer-loop inversion evaluates as being required to perform such rapid decelerations and accelerations. This experiment is an example of maneuvering where the commanded trajectory is more aggressive than the capability of the vehicle and is reflected by the extended periods of saturation. It is possible to operate at the limits of the vehicle primarily due to PCH which protects the adaptation process.

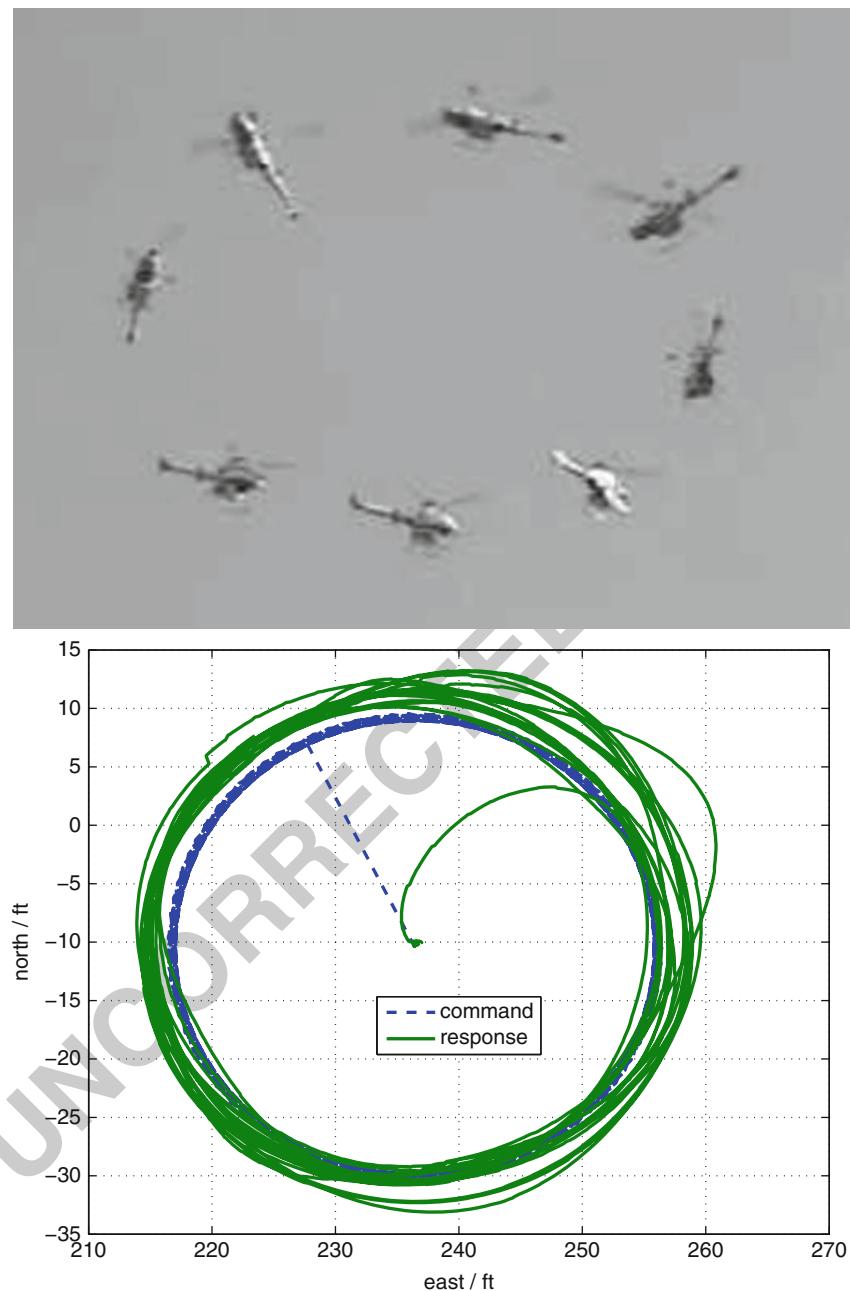


Fig. 61.15 Circular maneuver, with 360° heading changes during the circuit

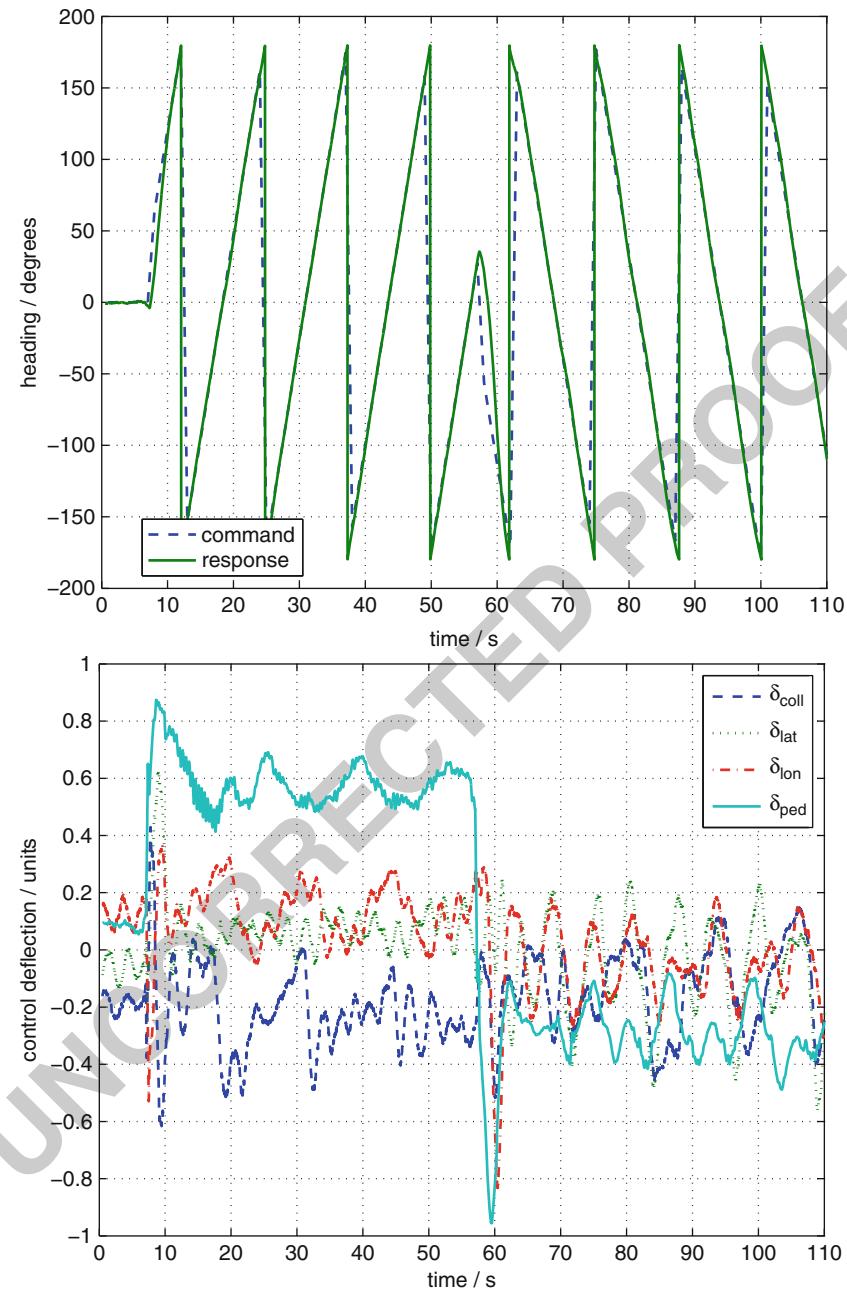
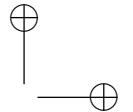
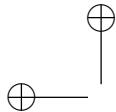
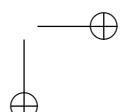
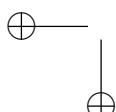


Fig. 61.16 Heading tracking during circular maneuver and control time history



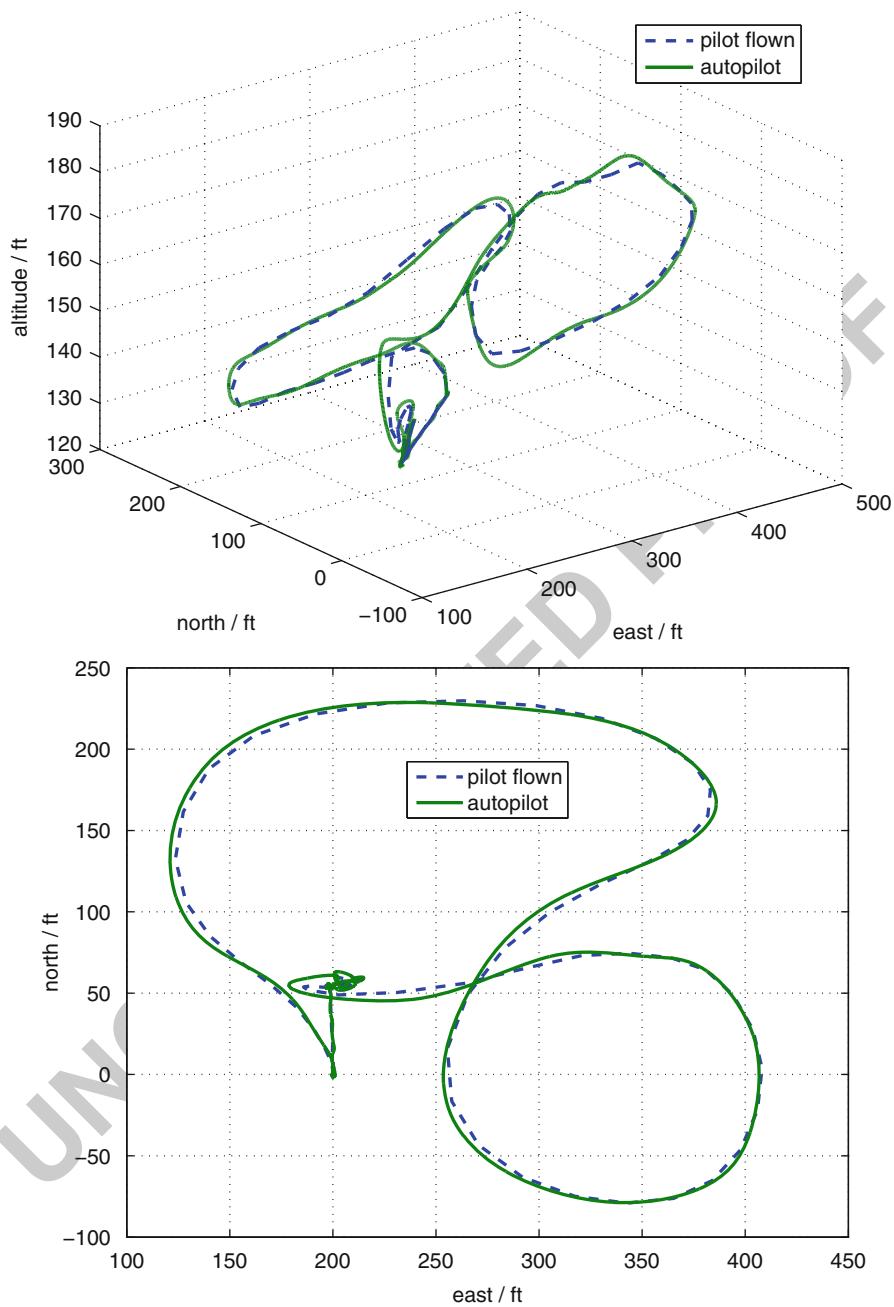
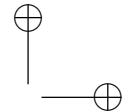
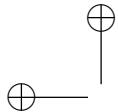
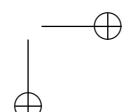
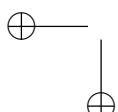


Fig. 61.17 A 3-D view and ground track view, of a trajectory initially flown manually by a pilot and then tracked by the controller



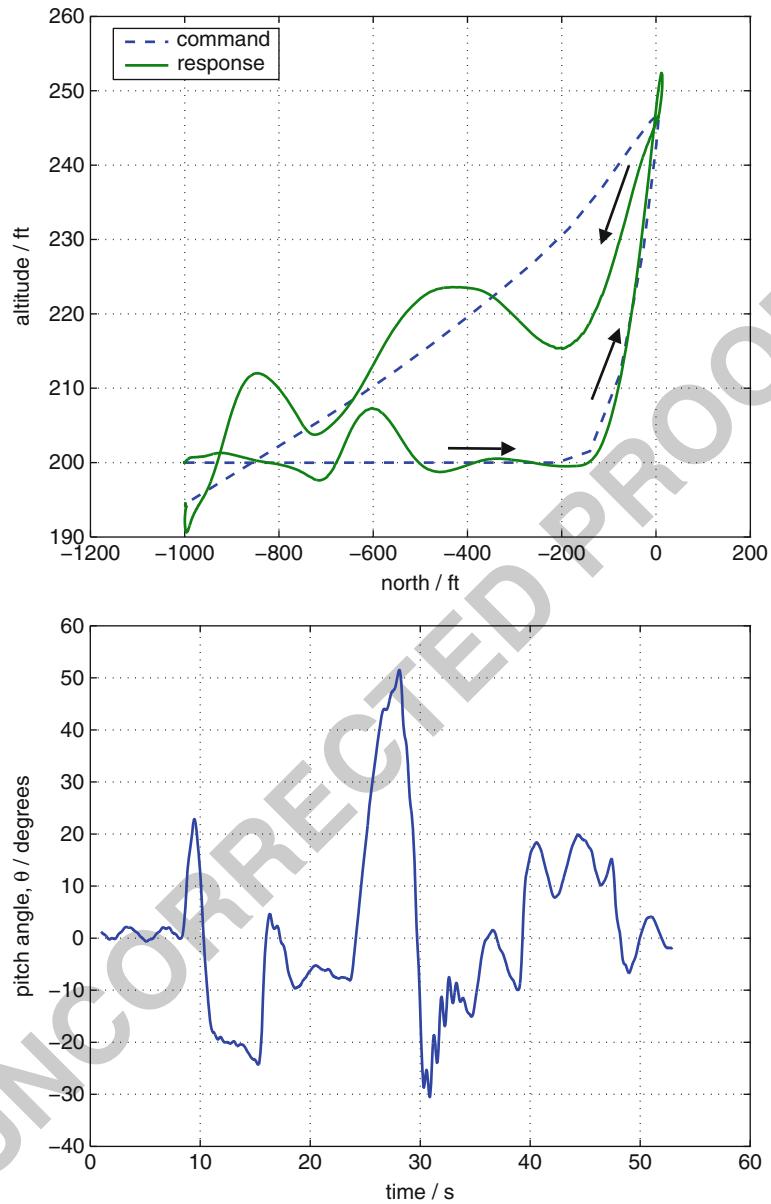
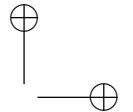
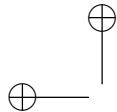
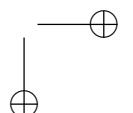
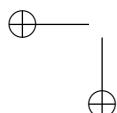


Fig. 61.18 North-altitude and pitch angle profile during a 180° velocity change maneuver.
Note: North axis and Altitude axis scales are not equal



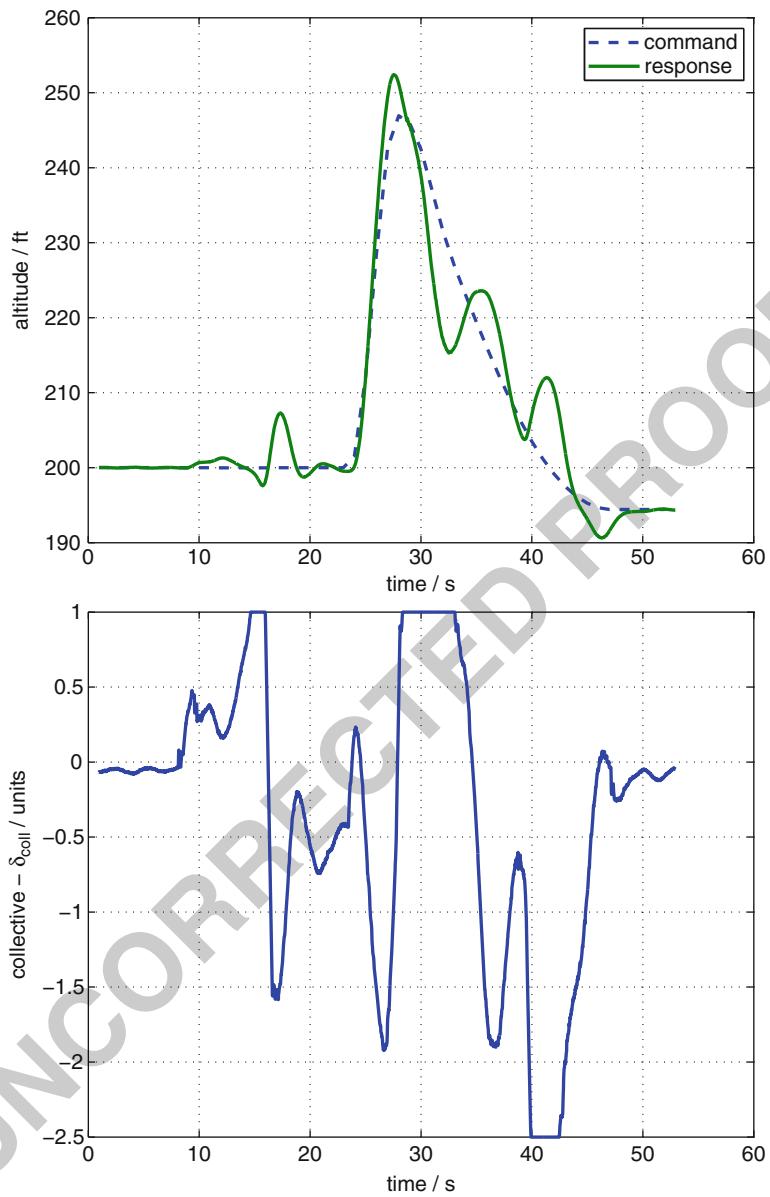
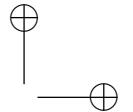
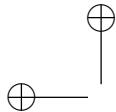
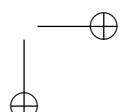
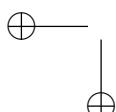


Fig. 61.19 Altitude and collective control history during a 180° velocity change maneuver



929 **61.5.3 Application to a Ducted Fan**

930 Following tests on the GTMax helicopter, the control method presented in this
 931 chapter was applied to other smaller aircraft. The algorithms were ported to a
 932 custom DSP/FPGA hardware device (the FCS20) along with a small sensor board
 933 that contained gyroscopes and accelerometers for inertial sensing and a GPS. The
 934 avionics package weighed less than 1 lb and fell within the payload capacity of
 935 the 11-in. ducted fan (GTSpy). The GTSpy has a maximum takeoff weight of
 936 5.5 lb and is driven by a two-bladed fixed-pitch propeller. The propeller is enclosed
 937 in an annular wing duct with an outer diameter of 11 in. Vanes located directly
 938 beneath the propeller move in order to provide yaw control about the propeller axis.
 939 Two sets of control surfaces located further below the propeller move in order to
 940 provide pitch and roll moments. Maneuvering is accomplished by tilting the thrust
 941 vector with the control surfaces relying primarily on inflow for dynamic pressure
 942 during hover. Following satisfactory tethered tests, the vehicle was untethered and
 943 allowed to fly simple missions. [Figure 61.20](#) shows a plan view of a small 50-ft
 944 box maneuver and the GTSpy’s tracking. The large deviation on the eastern side
 945 of the box is most likely due to a wind gust. Another maneuver performed was
 946 the mid-air deployment of the GTSpy. The GTSpy was mounted on the GTMax
 947 helicopter with its engine on and then deployed from a safe altitude. The GTSpy
 948 was able to recover from the initial deployment transient and maintain attitude and
 949 position within 5 s of launch. [Figure 61.21](#) shows the GTSpy and GTMax during

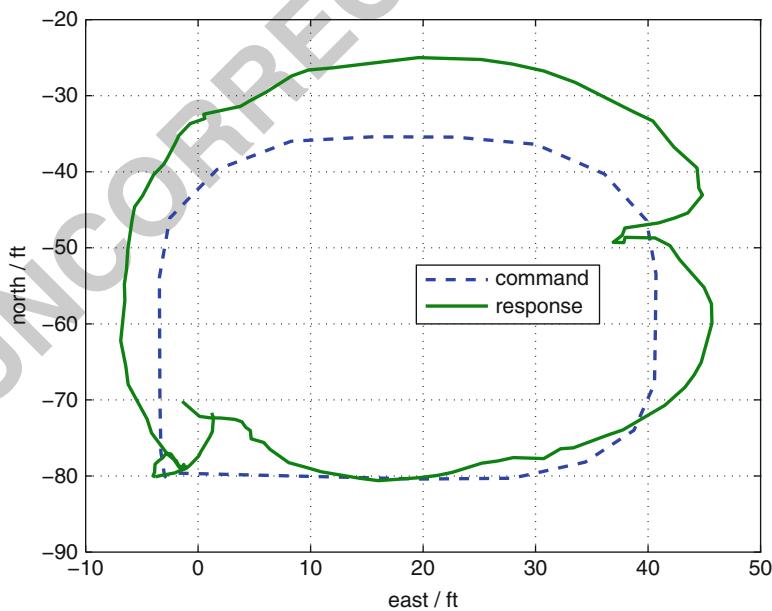


Fig. 61.20 The GTSpy performing a box maneuver

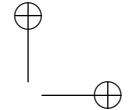
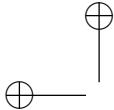


Fig. 61.21 Deployment of the GTSpy ducted fan from the GTMax helicopter



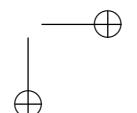
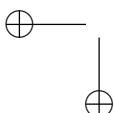
the deployment transient. Both the GTMax and GTSpy were under computer control during this maneuver and is the first known deployment of a rotorcraft from another rotorcraft.

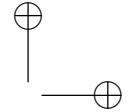
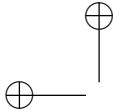
61.5.4 Application to a Fixed-Wing Aircraft

The control method presented in this chapter was further applied to a high thrust-to-weight ratio, fixed-wing aircraft with conventional aircraft controls and a fixed-pitch two-bladed propeller. The dynamic inverse used for control purposes approximated the aircraft in hover mode where the body axis was defined as

$$x_{heli} = L_2(-\pi/2)x_{airplane}$$

where L_2 is a rotation matrix around the airplane’s body y-axis. Hence, the ailerons control helicopter yaw, the rudder controls helicopter roll, and the elevators continue to control pitch. The external commands provided to the control algorithm contains a commanded pitch angle as a function of speed. Inner-loop gains were based on 2.5, 1.5, 2.5 rad/s for the (helicopter) roll, pitch and yaw axes, respectively. Outer-loop gains were based on 1.5, 1.0, 0.7 rad/s for the x, y, and z helicopter body axes, respectively. The output layer learning rates Γ_W were set to unity on all channels and a learning rate of Γ_V was set for all inputs. Reference model parameters were set to $v_{lim} = 10$ ft/s and $\omega_{lim} = 1.0$ rad/s. The control effectiveness B was scaled based on speed in order to reflect the reduced control authority of the control surfaces in hover. Flight tests were initiated with the airplane performing circular orbits and





970 gradually lowering airspeed until hover. The reverse transition to forward flight was
971 accomplished by a more aggressive command into forward flight.

972 The following figures illustrate the response of the aircraft during transitions
973 between hover and forward flight. [Figure 61.22](#) shows the vehicle in forward flight
974 at 80 ft/s performing a circular orbit. At $t = 26$ s a transition to hover is initiated by
975 supplying external trajectory commands that lower the vehicle’s speed. Transition
976 is completed at $t = 35$ s with a low residual speed of approximately 5 ft/s. At
977 $t = 55$ s a transition back to forward flight at 80 ft/s is initiated and completed
978 at $t = 65$ s. During hover, $t \in [35, 55]$, the control deflections are seen to be
979 significantly higher due to the lower effectiveness at lower speeds. The ailerons
980 are saturated for significant intervals in a particular direction in order to counteract
981 engine torque.

982 [Figure 61.23](#) illustrates the (helicopter) pitch angle during transitions as well as
983 the throttle control deflections. In forward flight, the pitch angle is approximately
984 -75° and varies in hover due to reduced control effectiveness and the presence
985 of a steady wind. Additionally, [Fig. 61.24](#) shows the position trajectory during
986 transitions, whereas [Fig. 61.25](#) is a snapshot of the aircraft during the maneuver.

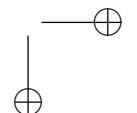
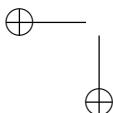
987 61.5.5 Implementation of Concurrent-Learning Adaptive Controller 988 on a VTOL UAV

989 Flight-test results of the concurrent-learning adaptive law described in [Sect. 61.3](#)
990 are presented in this section. The test vehicle is the GTMax rotorcraft UAV.
991 The modification to the adaptive controller described in [Sect. 61.2.2](#) include
992 the concurrent-learning adaptive law of equations ([Eq. 61.27](#)) for a nonlinearly
993 parameterized SHL NN. Data points were selected online based on [Eq. 61.30](#) and
994 were stored in a history stack limited to carrying 20 points. Once the history stack
995 was full, a new data point was added by replacing the oldest data point. A fixed point
996 smoother was used to estimate \dot{x}_j for a recorded data point using both forward and a
997 backward Kalman filter ([Chowdhary and Johnson 2011b; Gelb 1974](#)). Typically this
998 induced a selectable time delay introduced by the time required for the smoother to
999 converge, however, this does not affect the instantaneous tracking error.

1000 61.5.5.1 Repeated Forward-Step Maneuvers

1001 The repeated forward-step maneuvers are chosen in order to create a relatively
1002 simple situation in which the controller performance can be compared over several
1003 similar maneuvers. By using concurrent-learning NN, improved performance
1004 is expected through repeated maneuvers and a faster convergence of weights.
1005 [Figure 61.26](#) shows the body frame states from recorded flight data for a chain of
1006 forward-step inputs. [Figure 61.27a, b](#) shows the evolution of inner- and outer-loop
1007 errors. These results assert the stability (in the ultimate boundedness sense) of the
1008 combined concurrent and online learning approach.

1009 [Figure 61.28b, d](#) show the evolution of NN W and V weights as the rotor-
1010 craft performs repeated step maneuvers and the NN is trained using the



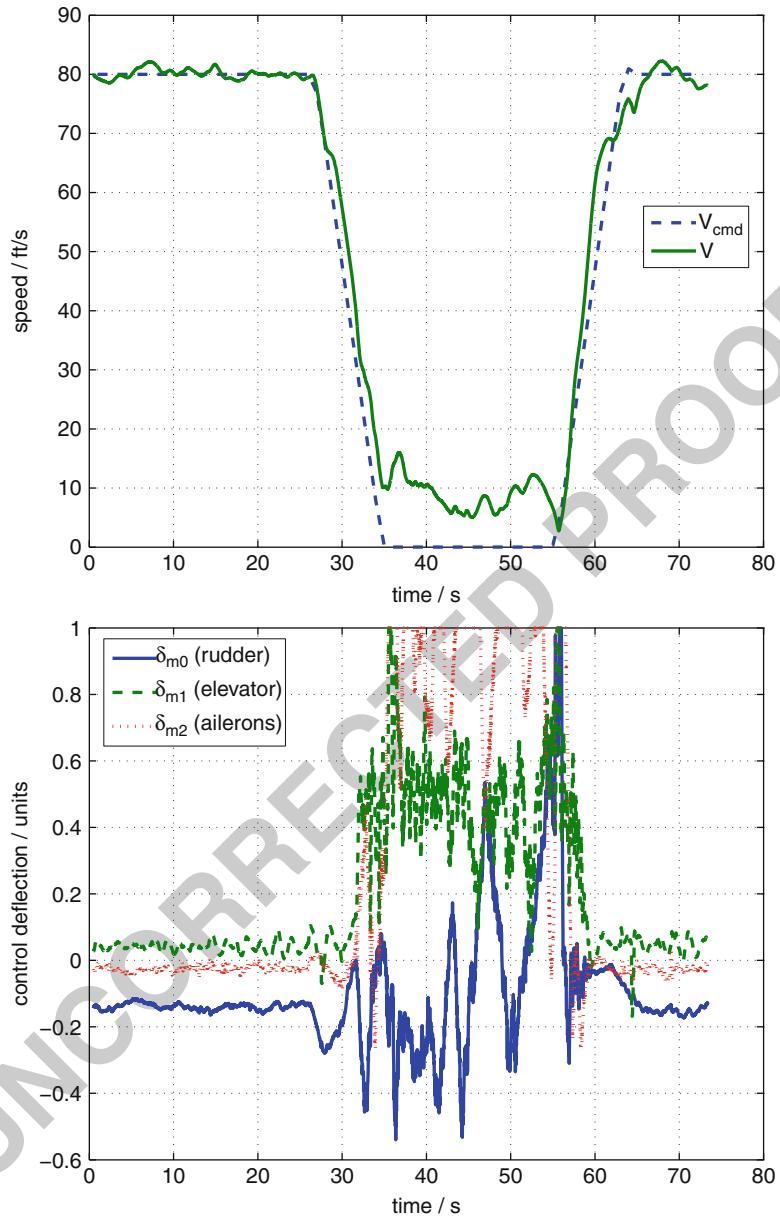
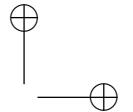
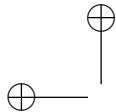
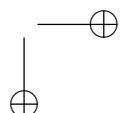
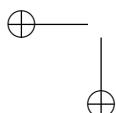


Fig. 61.22 GTEdge speed profile and control deflections during transitions between hover and forward flight



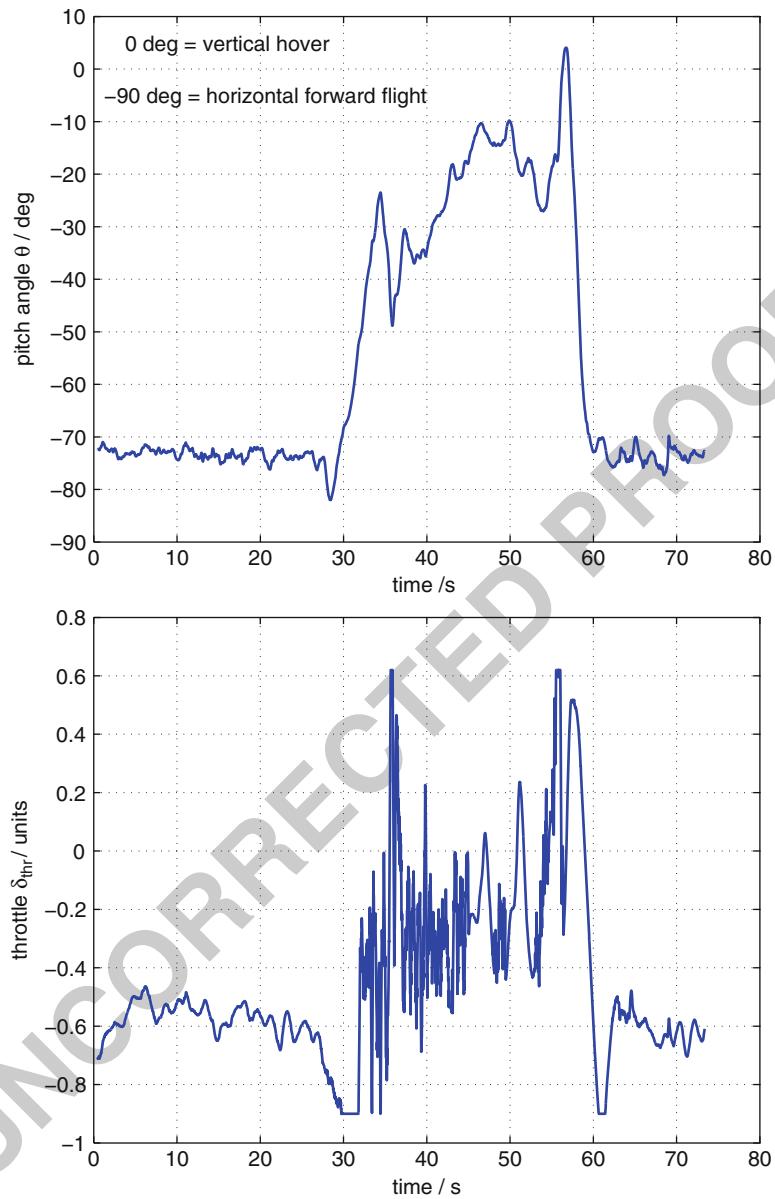
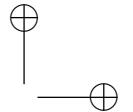
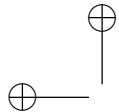
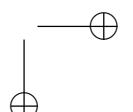
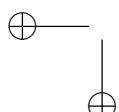
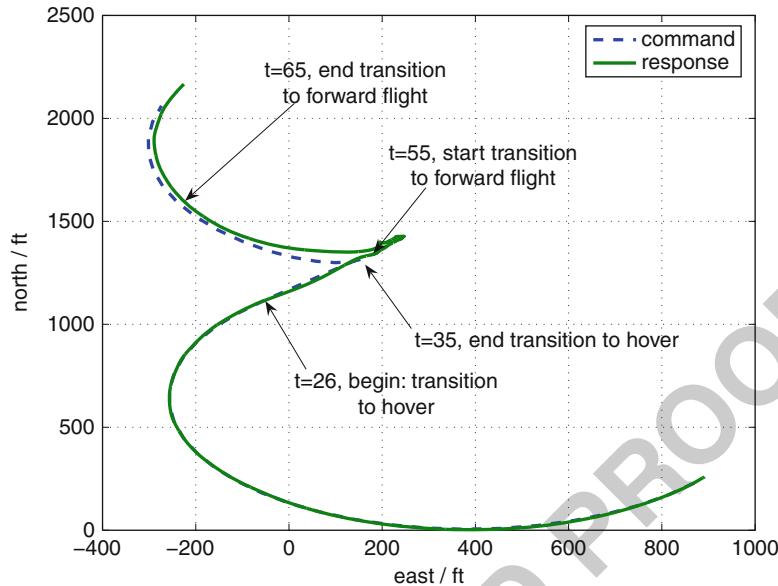
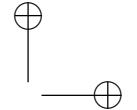
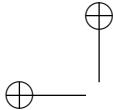
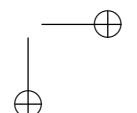
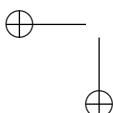


Fig. 61.23 GTEdge pitch angle, throttle profile during transitions between hover and forward flight



**Fig. 61.24** GTEdge trajectory during transitions**Fig. 61.25** GTEdge during a transition

1011 concurrent-learning method of Theorem 61.3. The NN V weights (Fig. 61.28b)
 1012 appear to go to constant values when concurrent-learning adaptation is used, this
 1013 can be contrasted with Fig. 61.28a which shows the V weight adaptation for a
 1014 similar maneuver without concurrent-learning. NN W weights for both cases remain
 1015 bounded, however it is seen that with concurrent learning adaptation the NN W
 1016 weights seem to separate, this indicates alleviation of the rank-1 condition experi-
 1017 enced by the baseline adaptive law relying only on instantaneous data (Chowdhary



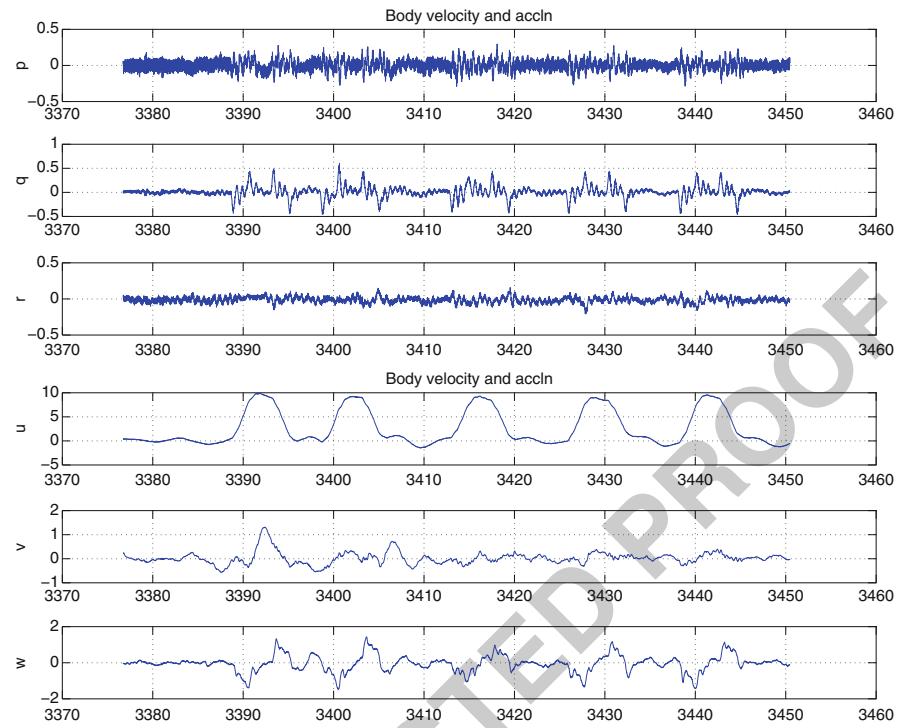
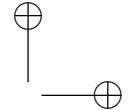
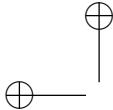
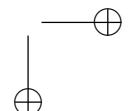
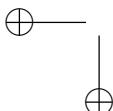


Fig. 61.26 Recorded body frame states for repeated forward steps

and Johnson 2011b). The flight test results indicate a noticeable improvement in the error profile. In Fig. 61.26 it is seen that the UAV tends not to have a smaller component of body lateral velocity (ν) through each successive step. This is also seen in Fig. 61.27b where it is noted that the error in ν (body y axis velocity) reduces through successive steps. These effects in combination indicate that the combined online and concurrent-learning system is able to improve performance over the baseline controller through repeated maneuvers, indicating long-term learning. These results are of particular interest, since the maneuvers performed were conservative, and the baseline adaptive MRAC controller had already been extensively tuned.

61.5.5.2 Aggressive Trajectory Tracking Maneuvers

Flight-test results are presented for concurrent-learning adaptive controllers while tracking repeatedly an elliptical trajectory with aggressive velocity (50 ft/s) and acceleration (20 ft/s²) profile. Since these maneuvers involve state commands in more than one system state it is harder to visually inspect the data and see whether an improvement in performance is seen, therefore the Euclidian norm of



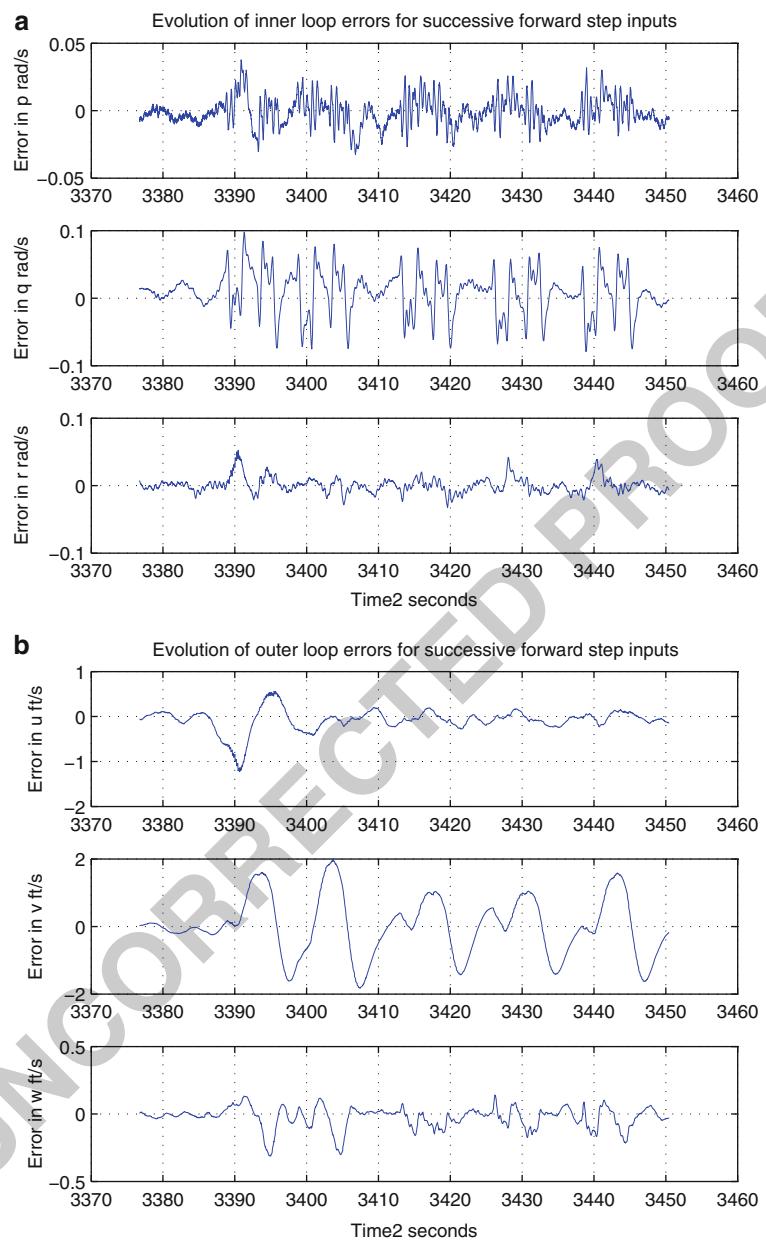


Fig. 61.27 GTMax recorded tracking errors for successive forward-step inputs with concurrent learning. (a) Evolution of inner loop errors with concurrent adaptation. (b) Evolution of outer loop errors with concurrent adaptation

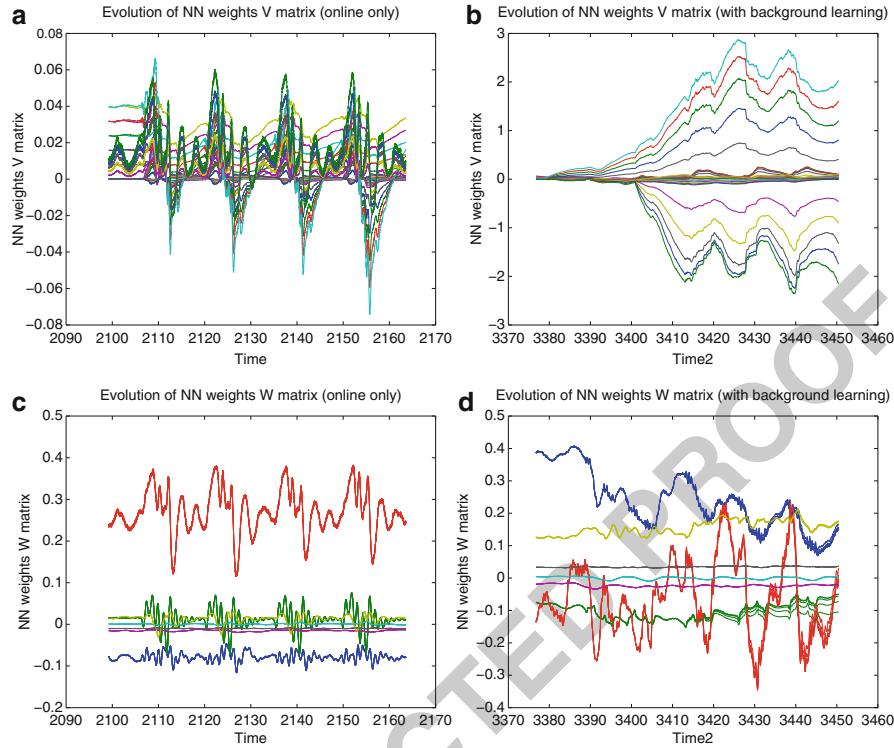


Fig. 61.28 Comparison of weight convergence on GTMax with and without concurrent learning. (a) Evolution of V matrix weights with only online adaptation. (b) Evolution of V matrix weights with concurrent adaptation. (c) Evolution of W matrix weights with only online adaptation. (d) Evolution of W matrix weights with concurrent adaptation

1034 the error signal at each time step is used as a rudimentary metric. [Figure 61.29](#)
 1035 shows the recorded inner- and outer-loop states as the rotorcraft repeatedly tracks
 1036 an oval trajectory pattern. In this flight, the first two ovals (until $t=5415\text{ s}$) are
 1037 tracked with a commanded acceleration of 30 ft/s^2 , while the rest of the ovals are
 1038 tracked at 20 ft/s^2 . In the following both these parts of the flight test are discussed
 1039 separately.

1040 61.5.5.3 Aggressive Trajectory Tracking with Saturation in the 1041 Collective Channel

1042 Due to the aggressive acceleration profile of 30 ft/s^2 the rotorcraft collective
 1043 channels were observed to saturate while performing high velocity turns. This leads
 1044 to an interesting challenge for the adaptive controller equipped with pseudocontrol
 1045 hedging. [Figure 61.30](#) shows the evolution of the innerloop and outerloop tracking
 1046 error. It can be clearly seen that the tracking error in the u (body x axis velocity)

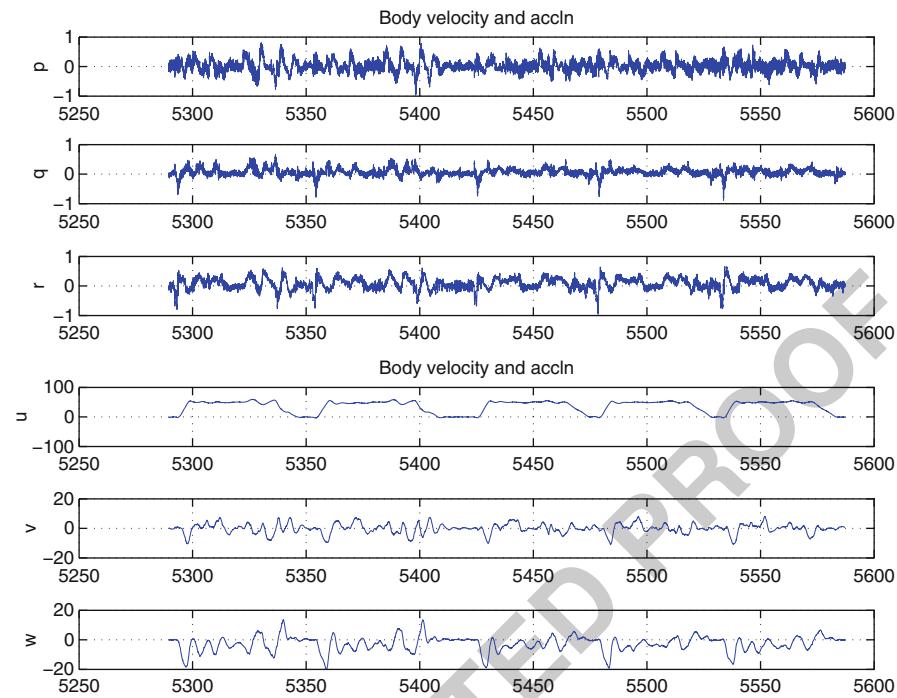
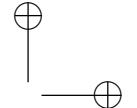
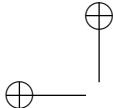


Fig. 61.29 Recorded body frame states for repeated oval maneuvers

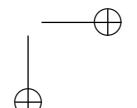
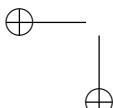
1047 channel reduces in the second pass through the ellipse indicating long-term learning
 1048 by the combined online and concurrent-learning adaptive control system. This result
 1049 is further characterized by the noticeable reduction in the norm of the tracking error
 1050 at every time step as shown in Fig. 61.31.

1051 **61.5.5.4 Aggressive Trajectory Tracking Maneuver**

1052 For the results presented in this section, the acceleration profile was reduced to
 1053 20 ft/s^2 . At this acceleration profile, no saturation in the collective input was noted.
 1054 Figure 61.32 shows the evolution of tracking error, and Fig. 61.33a shows the plot
 1055 of the norm of the tracking error at each time step.

1056 **61.5.5.5 Aggressive Trajectory Tracking Maneuvers with Only Online 1057 Learning NN**

1058 The performance of the concurrent-learning adaptive controller is compared with
 1059 the traditional instantaneous update-based adaptive controllers for the maneuvers
 1060 described in Sect. 61.5.5.3.



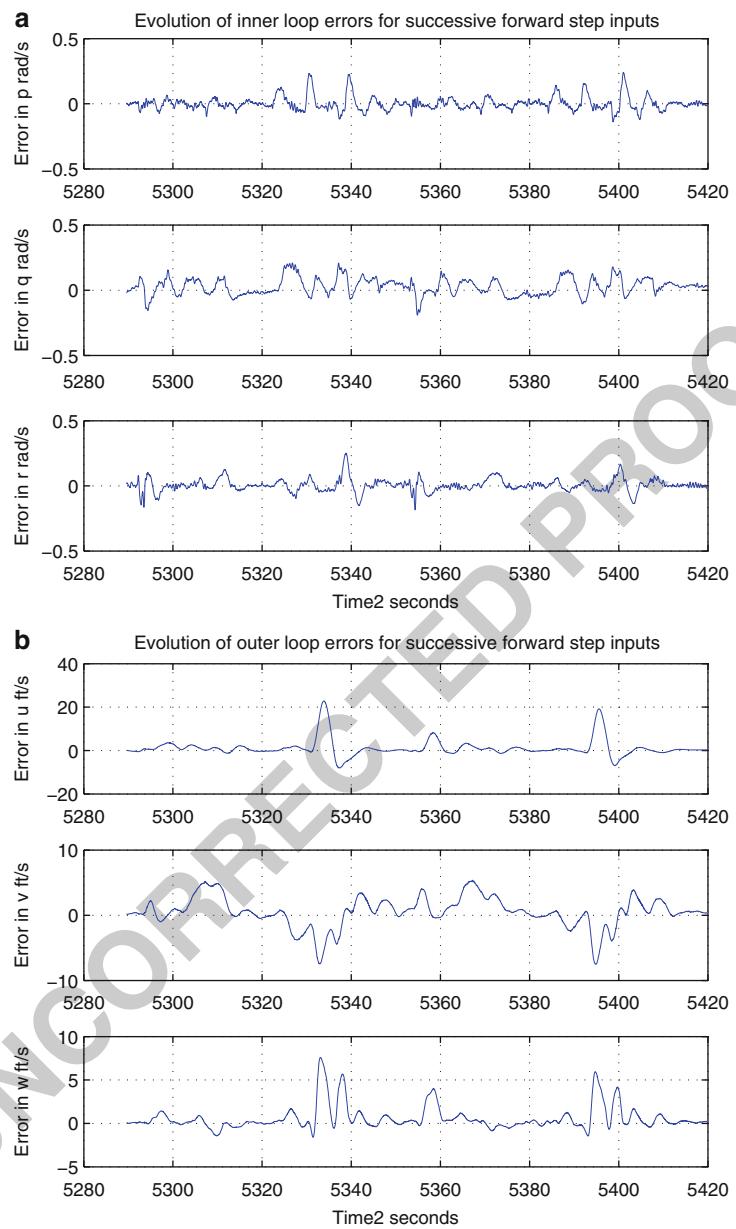
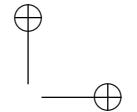
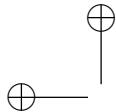
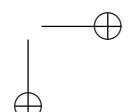
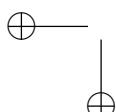


Fig. 61.30 GTMax recorded tracking errors for aggressive maneuvers with saturation in collective channels with concurrent learning. **(a)** Evolution of inner-loop errors with concurrent adaptation. **(b)** Evolution of outer-loop errors with concurrent adaptation



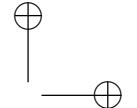
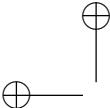
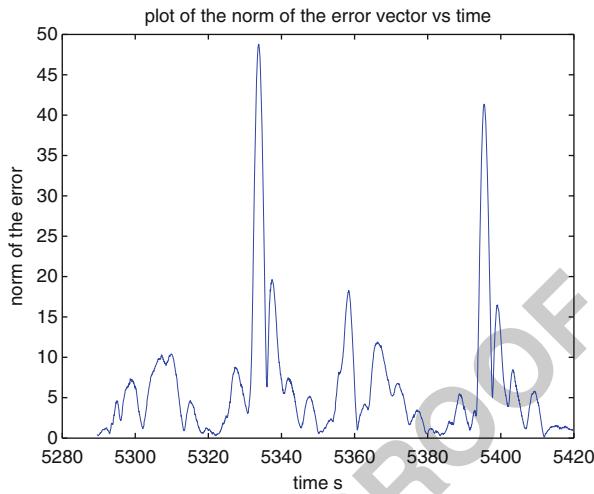


Fig. 61.31 Plot of the norm of the error at each time step for aggressive trajectory tracking with collective saturation

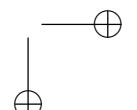
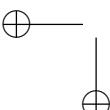


1061 It is instructive to compare Fig. 61.34b, d which show the evolution of the NN
1062 weights with only instantaneous learning with Fig. 61.34a, c which show evolution
1063 of the NN weights with concurrent learning. Although absolute convergence of
1064 weights is not seen, as expected due to Theorem 61.3, it is interesting to see that
1065 when concurrent learning is on, the weights tend to be less oscillatory than when
1066 only instantaneous learning is used. Also, with combined online and concurrent
1067 learning, the weights do not tend to go to zero as the rotorcraft hovers between
1068 two successive tracking maneuvers. Figure 61.33b shows the plot of the tracking
1069 error norm as a function of time without concurrent learning. Comparing this
1070 figure with Fig. 61.33a, it can be clearly seen that the norm of the error vector is
1071 much higher when only online learning is used. This indicates that the combined
1072 online and concurrent-learning adaptive controller has improved trajectory tracking
1073 performance.

1074 In summary, the flight test results ascertain an expected improvement in tracking
1075 performance. Furthermore, the evolution of the neural network W and V matrix
1076 weights were observed to have different characteristics when concurrent learning
1077 was employed, including weight separation, a tendency toward weight convergence
1078 in some cases, and different numerical values of the adaptive weights. This
1079 difference in neural network weight behavior demonstrates the effect of overcoming
1080 the rank 1 condition.

1081 61.6 Summary

1082 The objective in this chapter has been to provide an affordable control de-
1083 sign solution that uses minimal prior knowledge of the vehicle dynamics. This
1084 is accomplished by relying on adaptation to cover the flight envelope of the



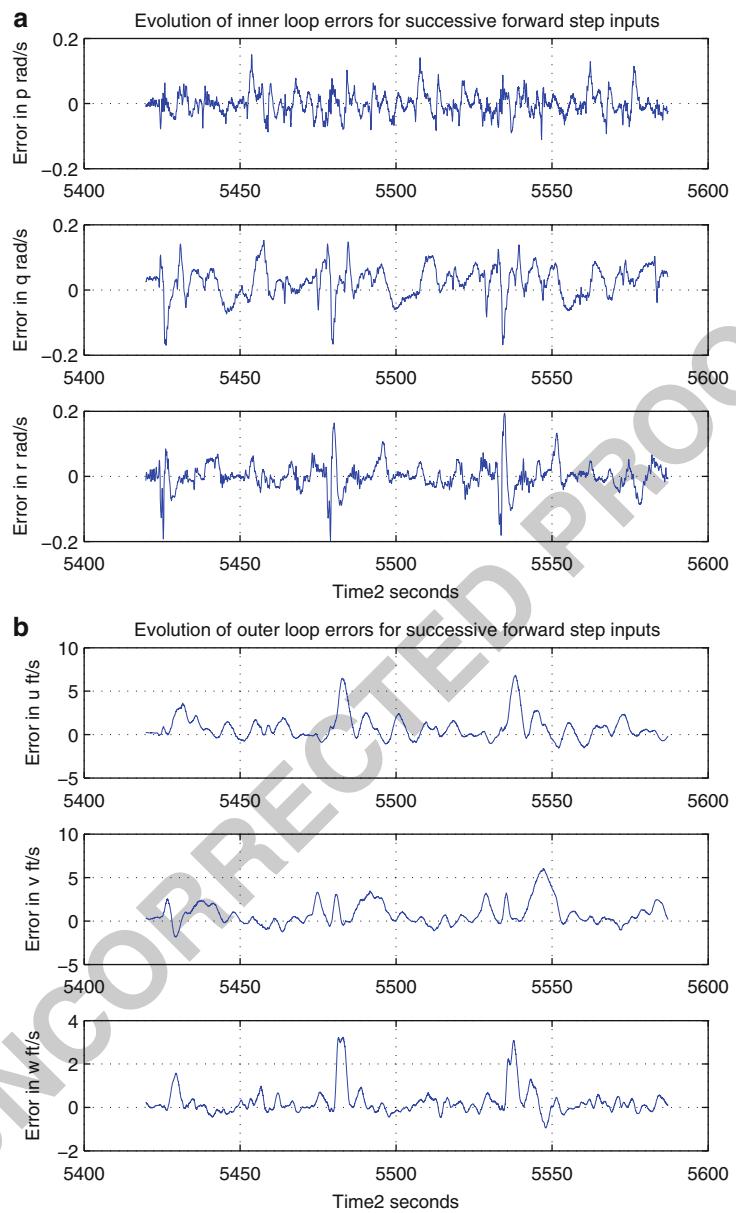
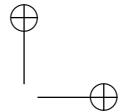
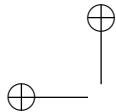
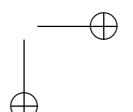
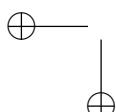


Fig. 61.32 GTMax recorded tracking errors for aggressive maneuvers with concurrent learning. (a) Evolution of inner-loop errors with concurrent adaptation. (b) Evolution of outer-loop errors with concurrent adaptation



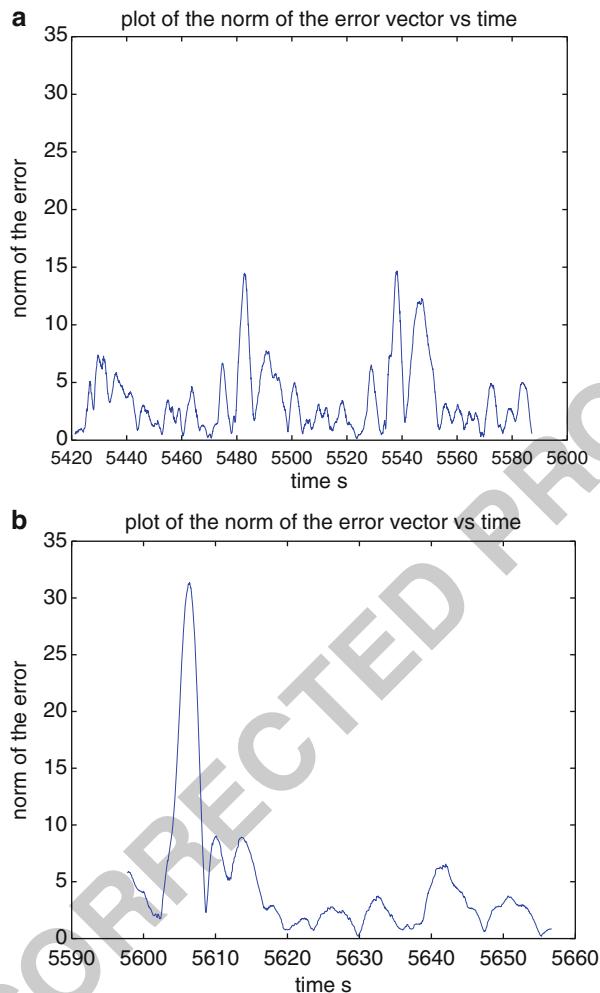
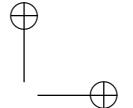
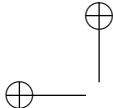
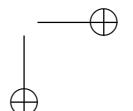
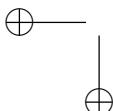


Fig. 61.33 Comparison of the norm of GTMax recorded tracking errors for aggressive maneuvers. (a) Evolution of the norm of the tracking error with concurrent adaptation. (b) Evolution of the norm of the tracking error with only online adaptation

1085 helicopter under nominal conditions. Under mission-specific variations in the
1086 environment and system dynamics due to payload changes or damage, adap-
1087 tation allows little or no human intervention after deployment. This approach
1088 is also in agreement with the DoD UAS Roadmap which subscribes to the
1089 following view on UAVs: “...affordability will be treated as a key performance
1090 parameter (KPP) equal to, if not more important than, schedule and technical
1091 performance...”.



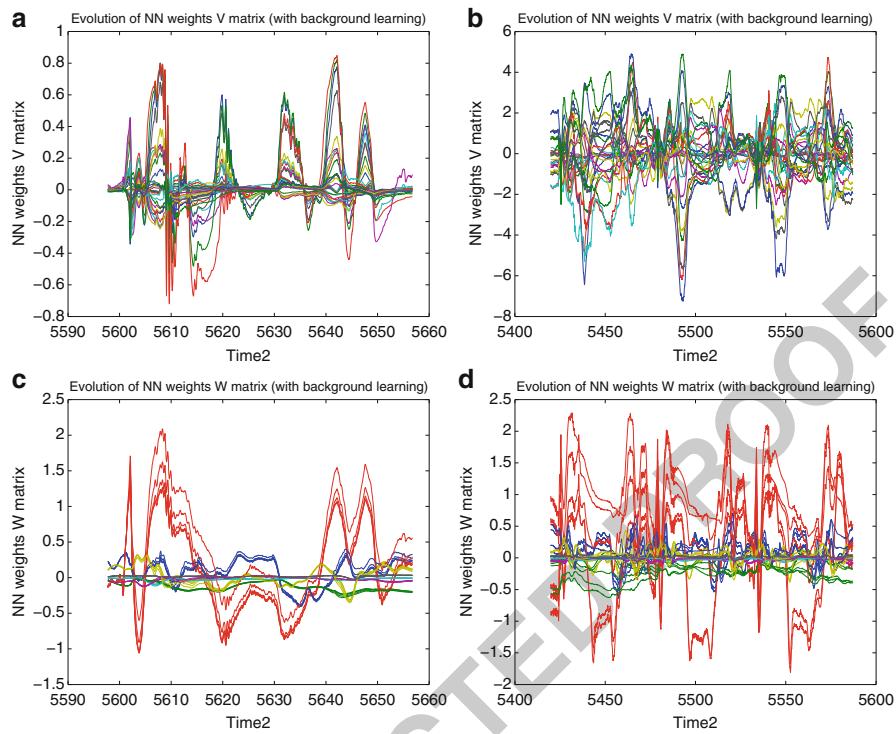
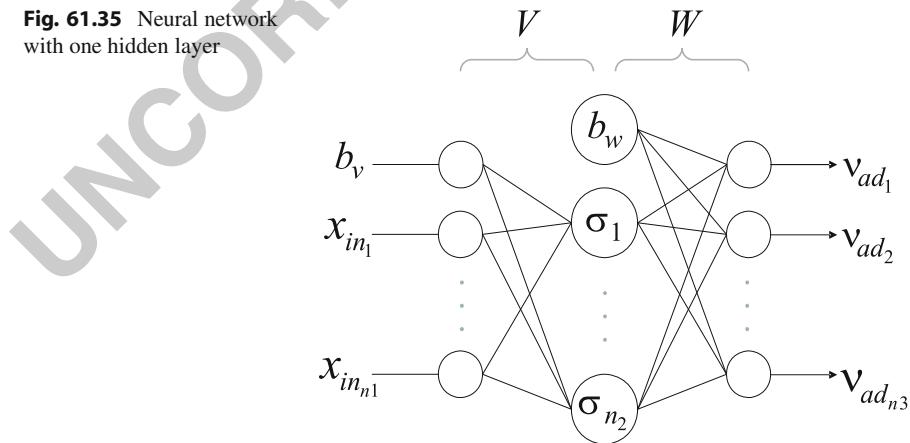
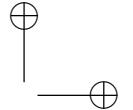
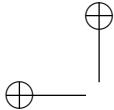


Fig. 61.34 Comparison of weight convergence as GTMax tracks aggressive trajectory with and without concurrent learning. (a) Evolution of V matrix weights with only online adaptation. (b) Evolution of V matrix weights with concurrent adaptation. (c) Evolution of W matrix weights with only online adaptation. (d) Evolution of W matrix weights with concurrent adaptation

Fig. 61.35 Neural network with one hidden layer





1092 **A Adaptive Element**

1093 Single hidden layer (SHL) perceptron NNs are universal approximators (Hornik
1094 et al. 1989b; Spooner et al. 2002; Lewis 1999). Hence, given a sufficient number
1095 of hidden layer neurons and appropriate inputs, it is possible to train the network
1096 online to cancel model error.

1097 Figure 61.35 shows the structure of a generic single hidden layer network whose
1098 input-output map may be expressed as

1099
$$v_{ad_k} = b_w \theta_{w_k} + \sum_{j=1}^{n_2} w_{jk} \sigma_j(z_j), \quad (61.51)$$

1100 where $k = 1, \dots, n_3$, b_w is the outer layer bias, θ_{w_k} is the k th threshold. w_{jk}
1101 represents the outer layer weights, z_j is the input to the neurons, and the scalar
1102 σ_j is a sigmoidal activation function

1103
$$\sigma_j(z_j) = \frac{1}{1 + e^{-az_j}}, \quad (61.52)$$

1104 where a is the so-called activation potential and may have a distinct value for each
1105 neuron. z_j is the input to the j th hidden layer neuron and is given by

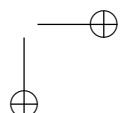
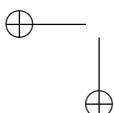
1106
$$z_j = b_v \theta_{v_j} + \sum_{i=1}^{n_1} v_{ij} x_{\text{in}_i}, \quad (61.53)$$

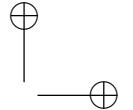
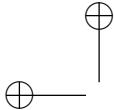
1107 where b_v is the inner layer bias and θ_{v_j} is the j th threshold. Here, n_1, n_2 , and
1108 n_3 are the number of inputs, hidden layer neurons, and outputs, respectively. x_{in_i} ,
1109 $i = 1, \dots, n_1$, denotes the inputs to the NN. For convenience, define the following
1110 weight matrices:

1111
$$V \triangleq \begin{bmatrix} \theta_{v,1} & \dots & \theta_{v,n_2} \\ v_{1,1} & \dots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \dots & v_{n_1,n_2} \end{bmatrix}, \quad (61.54)$$

1112
$$W \triangleq \begin{bmatrix} \theta_{w,1} & \dots & \theta_{w,n_3} \\ w_{1,1} & \dots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \dots & w_{n_2,n_3} \end{bmatrix}, \quad (61.55)$$

1113
$$Z \triangleq \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix}. \quad (61.56)$$





1115 Additionally, define the $\sigma(z)$ vector as

1116
$$\sigma^T(z) \triangleq [b_w \ \sigma(z_1) \cdots \sigma(z_{n_2})], \quad (61.57)$$

1117 where $b_w > 0$ allows for the thresholds, θ_w , to be included in the weight matrix W .
 1118 Also, $z = V^T \bar{x}$, where

1119
$$\bar{x}^T = [b_v \ x_{in}^T], \quad (61.58)$$

1120 where $b_v > 0$ is an input bias that allows for thresholds θ_v to be included in the
 1121 weight matrix V . The input-output map of the SHL network may now be written in
 1122 concise form as

1123
$$v_{ad} = W^T \sigma(V^T \bar{x}). \quad (61.59)$$

1124 The NN may be used to approximate a nonlinear function, such as $\Delta(\cdot)$. The
 1125 universal approximation property (Hornik et al. 1989b) of NNs ensures that given
 1126 an $\bar{\epsilon} > 0$, then $\forall \bar{x} \in \mathcal{D}$, where \mathcal{D} is a compact set, \exists an \bar{n}_2 , and an ideal set of
 1127 weights (V^*, W^*) brings the output of the NN to within an ϵ -neighborhood of the
 1128 function approximation error. This ϵ is bounded by $\bar{\epsilon}$ which is defined by

1129
$$\bar{\epsilon} = \sup_{\bar{x} \in \mathcal{D}} \|W^T \sigma(V^T \bar{x}) - \Delta(\bar{x})\|. \quad (61.60)$$

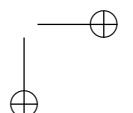
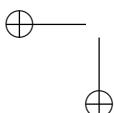
1130 The weights (V^*, W^*) may be viewed as optimal values of (V, W) in the sense
 1131 that they minimize $\bar{\epsilon}$ on \mathcal{D} . These values are not necessarily unique. The universal
 1132 approximation property thus implies that if the NN inputs x_{in} are chosen to reflect
 1133 the functional dependency of $\Delta(\cdot)$, then $\bar{\epsilon}$ may be made arbitrarily small given a
 1134 sufficient number of hidden layer neurons, n_2 .

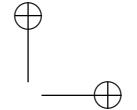
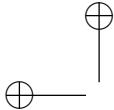
AU8

1135 References

- 1136 Aeronautical Design Standard, *Handling Qualities Requirements for Military Rotorcraft, ADS-33E*
 1137 (United States Army Aviation and Missile Command, Redstone Arsenal, AL, March 2000)
- 1138 A. Bemporad, A. Casavola, E. Mosca, Nonlinear control of constrained linear systems via
 1139 predictive reference management. *IEEE Trans. Autom. Control* **42**(3), 340–349 (1997)
- 1140 D.S. Bernstein, A.N. Michel, A chronological bibliography on saturating actuators. *Int. J. Robust
 1141 Nonlinear Control* **5**, 375–380 (1995)
- 1142 S. Boyd, S. Sastry, Necessary and sufficient conditions for parameter convergence in adaptive
 1143 control. *Automatica* **22**(6), 629–639 (1986)
- 1144 A.J. Calise, N. Hovakimyan, M. Idan, Adaptive output feedback control of nonlinear systems using
 1145 neural networks. *Automatica* **37**, 1201–1211 (2001)
- 1146 G. Chowdhary, Concurrent learning for convergence in adaptive control without persistency of
 1147 excitation. Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, 2010
- 1148 G. Chowdhary, E.N. Johnson, Concurrent learning for convergence in adaptive control without
 1149 persistency of excitation, in *49th IEEE Conference on Decision and Control*, Atlanta, 2010,
 1150 pp. 3674–3679
- 1151 G. Chowdhary, E.N. Johnson, A singular value maximizing data recording algorithm for
 1152 concurrent learning, in *American Control Conference*, San Francisco, June 2011a

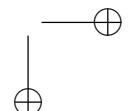
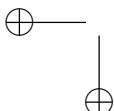
AU9

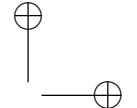
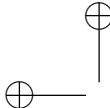




- 1153 G. Chowdhary, E.N. Johnson, Theory and flight test validation of a concurrent learning adaptive
1154 controller. *J. Guid. Control Dyn.* **34**(2), 592–607 (2011b)
- 1155 G. Chowdhary, E.N. Johnson, S.M. Kimbrell, R. Chandramohan, A.J. Calise, Flight test results of
1156 adaptive controllers in the presence of significant aircraft faults, in *AIAA Guidance Navigation*
1157 and *Control Conference*, Toronto, Canada, 2010 Invited
- 1158 G. Chowdhary, R. Chandramohan, J. Hur, E.N. Johnson, A.J. Calise, Autonomous guidance and
1159 control of an airplane under severe damage, in *AIAA@INFOTECH*, St. Louis, MO, March
1160 2011a. invited paper
- 1161 G. Chowdhary, M. Sobers, C. Pravitra, C. Christmann, A. Wu, H. Hashimoto, C. Ong, R. Kalghatgi,
1162 E.N. Johnson, Integrated guidance navigation and control for a fully autonomous indoor uas,
1163 in *AIAA Guidance Navigation and Control Conference*, Portland, OR, Aug 2011b
- 1164 G. Chowdhary, E.N. Johnson, R. Chandramohan, S.M. Kimbrell, A. Calise, Autonomous guidance
1165 and control of airplanes under actuator failures and severe structural damage. *J. Guid. Control*
1166 *Dyn.* (2012 in-press)
- 1167 H. Christophersen, R.W. Pickell, J.C. Neidhoefer, A.A. Koller, S.K. Kannan, E.N. Johnson, A
1168 compact guidance, navigation, and control system for unmanned aerial vehicles. *J. Aerosp.*
1169 *Comput. Inf. Commun.* **3**(5), 187–213 (2006)
- 1170 M. La Civita, W.C. Messner, T. Kanade, Modeling of small-scale helicopters with integrated
1171 first-principles and system-identification techniques, in *Proceedings of the 58th Forum of the*
1172 *American Helicopter Society*, Montreal, Canada, June 2002a, vol. 2, pp. 2505–2516
- 1173 M. La Civita, G. Papageorgiou, W.C. Messner, T. Kanade, Design and flight testing of a high
1174 bandwidth \mathcal{H}_∞ loop shaping controller for a robotic helicopter, in *AIAA Guidance, Navigation*
1175 and *Control Conference*, number AIAA-2002-4846, Monterey, CA, Aug 2002b
- 1176 M. La Civita, G. Papageorgiou, W.C. Messner, T. Kanade, Design and flight testing of a gain-
1177 scheduled \mathcal{H}_∞ loop shaping controller for wide-envelope flight of a robotic helicopter, in
1178 *Proceedings of the 2003 American Control Conference*, Denver, CO, June 2003, pp. 4195–
1179 4200
- 1180 J.E. Corban, A.J. Calise, J.V.R. Prasad, G. Heynen, B. Koenig, J. Hur, Flight evaluation of an
1181 adaptive velocity command system for unmanned helicopters, in *AIAA Guidance, Navigation,*
1182 and *Control Conference and Exhibit*, Austin, TX, Aug 2003
- 1183 B. Etkin, *Dynamics of Atmospheric Flight* (Wiley, New York, 1972)
- 1184 E. Frazzoli, M.A. Dahleh, E. Feron, Real-time motion planning for agile autonomous vehicles.
1185 *AIAA J. Guid. Control Dyn.* **25**(1), 116–129 (2002)
- 1186 V. Gavrilets, B. Mettler, E. Feron, Nonlinear model for a small-sized acrobatic helicopter, in *AIAA*
1187 *Guidance, Navigation and Control Conference*, number 2001-4333, Montréal, QC, Canada,
1188 Aug 2001
- 1189 V. Gavrilets, B. Mettler, E. Feron, Control logic for automated aerobatic flight of miniature
1190 helicopters, in *AIAA Guidance, Navigation and Control Conference*, number AIAA-2002-4834,
1191 Monterey, CA, Aug 2002
- 1192 A. Gelb, *Applied Optimal Estimation* (MIT, Cambridge, 1974)
- 1193 S. Haykin, *Neural Networks a Comprehensive Foundation*, 2nd edn. (Prentice Hall, Upper Saddle
1194 River, 1998)
- 1195 K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators.
1196 *IEEE Trans. Neural Netw.* **2**(5), 359–366 (1989a)
- 1197 K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators.
1198 *Neural Netw.* **2**(5), 359–366 (1989b)
- 1199 E.N. Johnson, Limited authority adaptive flight control. Ph.D. thesis, Georgia Institute of
1200 Technology, 270 Ferst Drive, Atlanta GA 30332, USA, 2000
- 1201 E.N. Johnson, S.K. Kannan, Adaptive trajectory control for autonomous helicopters. *J. Guid.*
1202 *Control Dyn.* **28**(3), 524–538 (2005)
- 1203 S.K. Kannan, Adaptive control of systems in cascade with saturation. Ph.D. thesis, Georgia
1204 Institute of Technology, 270 Ferst Drive, Atlanta GA 30332, USA, Dec 2005
- 1205 S.K. Kannan, E.N. Johnson, Nested saturation with guaranteed real poles, in *American Control*
1206 *Conference*, Denver, CO, June 2003, pp. 497–502

AU10





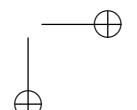
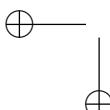
- 1207 S.K. Kannan, E.N. Johnson, Adaptive control of systems in cascade with saturation, in *IEEE Conference on Decision and Control*, Atlanta, GA, Dec 2010a
- 1208 S.K. Kannan, E.N. Johnson, Model reference adaptive control with a constrained linear reference model, in *IEEE Conference on Decision and Control*, Atlanta, GA, Dec 2010b
- 1209 S.K. Kannan, A.A. Koller, E.N. Johnson, Simulation and development environment for multiple heterogeneous uavs, in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Providence, RI, Aug 2004
- 1210 S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **30**, 846–894 (2011)
- 1211 F. Kendoul, A survey of advances in guidance, navigation and control of unmanned rotorcraft systems. *J. Field Robot.* **29**, 315–378 (2012). To appear
- 1212 F. Kendoul, I. Fantoni, R. Lozano, Modeling and control of a small autonomous aircraft having two tilting rotors. *IEEE Trans. Robot.* **22**(6), 1297–1302 (2006)
- 1213 F. Kendoul, L. David, I. Fantoni, R. Lozano, Real-time nonlinear embedded control for an autonomous quad-rotor helicopter. *AIAA J. Guid. Control Dyn.* **30**(4), 1049–1061 (2007)
- 1214 F. Kendoul, Z. Yu, K. Nonami, Guidance and nonlinear control system for autonomous flight of mini-rotorcraft unmanned aerial vehicles. *J. Field Robot.* **27**(3), 311–334 (2010)
- 1215 Y.H. Kim, F. Lewis, *High-Level Feedback Control with Neural Networks*. Robotics and Intelligent Systems, vol. 21 (World Scientific, Singapore, 1998)
- 1216 H.A. Kingravi, G. Chowdhary, P.A. Vela, E.N. Johnson, Reproducing kernel hilbert space approach for the online update of radial bases in neuro-adaptive control. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(7), 1130–1141 (2012)
- 1217 F.L. Lewis, Nonlinear network structures for feedback control (survey paper). *Asian J. Control* **1**(4), 205–228 (1999)
- 1218 Y. Lin, E.D. Sontag, Universal formula for stabilization with bounded controls. *Syst. Control Lett.* **16**, 393–397 (1991)
- 1219 A.M. Lipp, J.V.R. Prasad, Synthesis of a helicopter nonlinear flight controller using approximate model inversion. *Math. Comput. Model.* **18**, 89–100 (1993)
- 1220 B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft* (Kluwer, Boston/London, 2002)
- 1221 C. Munzinger, Development of a real-time flight simulator for an experimental model helicopter. Master’s thesis, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, GA 30332, USA, 1998
- 1222 G.J. Pappas, Avoiding saturation by trajectory reparameterization, in *IEEE Conference on Decision and Control*, Kobe, 1996
- 1223 J. Park, I. Sandberg, Universal approximation using radial-basis-function networks. *Neural Comput.* **3**, 246–257 (1991)
- 1224 D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. *Nature* **323**(6088), 533 (1986)
- 1225 R.T. Rysdyk, A.J. Calise, Robust nonlinear adaptive flight control for consistent handling qualities. *IEEE Trans. Control Syst. Technol.* **13**(6), 896–910 (2005). Robust nonlinear ad
- 1226 S.N. Singh, W. Yim, W.R. Wells, Direct adaptive control of wing rock motion of slender delta wings. *J. Guid. Control Dyn.* **18**(1), 25–30 (1995)
- 1227 J.T. Spooner, M. Maggiore, R. Ordóñez, K.M. Passino, *Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques* (Wiley, New York, 2002)
- 1228 B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation* (Wiley, New York, 2003a)
- 1229 B.L. Stevens, F.L. Lewis, *Aircraft Control and Simulation* (Wiley, Hoboken, 2003b)
- 1230 J.A. Suykens, J.P. Vandewalle, B.L.D. Moor, *Artificial Neural Networks for Modelling and Control of Non-Linear Systems* (Kluwer, Norwell, 1996)
- 1231 A. Teel, A nonlinear small gain theorem for the analysis of control systems with saturation. *IEEE Trans. Autom. Control* **41**(9), 1256–1270 (1996)
- 1232 A. Teel, Semi-global stabilization of linear systems with position and rate-limited actuators. *Syst. Control Lett.* **30**, 1–11 (1997)
- 1233 Unmanned Aircraft Systems Roadmap 2011–2036, Technical report, Office of the Secretary of Defense, 2011

AU11

AU12

AU13

AU14

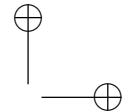
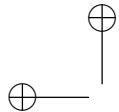


Author Query Form

Handbook of Unmanned Aerial Vehicles

Chapter No. 61

Query Refs.	Details Required	Author's response
AU1	Please check if author affiliations are okay.	
AU2	Please check if edit to sentence starting “In a helicopter, . . . ” is okay.	
AU3	Please check if “R-Max” should be changed to “RMAX” throughout text for consistency.	
AU4	Please check if edit to sentence starting “This universal approximation. . . ” is okay.	
AU5	Variables “pd, cr, . . . ” are used italics for both online and subscript throughout the chapter. Please check if we can retain them.	
AU6	Please check sentence starting “If the reference. . . ” for completeness.	
AU7	Please check if edit to sentence starting “For an air. . . ” is okay.	
AU8	Please check if edit to sentence starting “The universal approximation. . . ” is okay.	
AU9	Please check if updated conference location for Chowdhary and Johnson (2010) is okay.	
AU10	Please provide volume number and page range for Chowdhary et al. (2012).	
AU11	Please check if updated volume number and page range for Kendoul (2012) is okay.	



AU12	Please check if updated publisher location for Mettler (2002) and Spooner et al. (2002) are okay.	
AU13	Please check if updated author name “Hinton” for Rumelhart and Williams (1986) is okay.	
AU14	Please check if updated page range for Rysdyk and Calise (2005) is okay.	

UNCORRECTED PROOF

