

Adaptive Control of Unmanned Aerial Vehicles - Theory and Flight Tests

Suresh K. Kannan
Systems Department
United Technologies Research Center
East Hartford, CT 06118

Girish V. Chowdhary
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139

Eric N. Johnson
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332

May 24, 2012

Abstract



Typically, unmanned aerial vehicles are underactuated systems i.e., they have fewer independent control inputs than degrees of freedom. In a helicopter for example, the body-axis roll, pitch, yaw and altitude axis are fully actuated. However, lateral and longitudinal motion is only possible by tilting the thrust vector. Thus, roll and pitch attitude now act as virtual actuators. Such underactuated but controllable dynamics appear as cascaded nonlinear systems and are in general not feedback linearizable. An alternative is to feedback linearize the inner-loop (attitude dynamics) and outer-loop (translational dynamics) separately using approximate inversion models for each. An adaptive element can then be introduced to cancel approximation errors allowing for a control design that can function reliably in the presence of significant modeling uncertainties. A key challenge in realizing an adaptive control design on real aircraft is dealing with actuator magnitude and rate saturation. This chapter develops an adaptive control scheme for control of rotorcraft UAVs. It also introduces a mechanism called pseudocontrol hedging that is used to protect a single hidden layer neural network adaptive element from undesirable saturation effects by removing artifacts of the input

dynamics from the error dynamics used to drive the adaptation. Nonlinear reference models are introduced to mitigate the risks of the closed loop system leaving the null-controllable region. 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 concurrent-learning adaptive law is also presented that improves the long-term learning capabilities of the adaptive controller. The architecture is validated through flight-tests on several fixed wing and rotorcraft UAVs, including a 145 *lb* helicopter UAV (Yamaha R-Max), a scale model fixed-wing aircraft (GTEdge), and a small ducted-fan (GTSpy).

1 Introduction



The US Department of Defense Integrated Unmanned Systems Roadmap [1] 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.

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

Currently, fast algorithms to plan routes around obstacles are available[18, 32]. To be truly useful, these routes would include high-speed dashes, tight turns around buildings, avoiding dynamic obstacles and other required aggressive maneuvers. Allowing control saturation and adaptation allows higher level planning algorithms to provide optimistic trajectories which are then tracked by the flight controller to the best of the vehicles ability. Multiple factors affect flight control stability and performance. They may be loosely categorized based

on how they appear in the system equations.

1.1 Challenges

Parametric Uncertainty arises from uncertainty in the mass, inertia properties, and aerodynamic characteristics. This limits the operational envelope of the vehicle to flight regimes where control designs are valid and parametric uncertainty is small. The effects of parametric uncertainty and unmodeled dynamics can be handled using a combination of system identification [19, 35, 41] and robust control techniques [36, 20, 37]. However, system identification is expensive. Parametric uncertainty in this chapter is considered to be unstructured and handled using an adaptive element that is a nonlinear function approximator (such as a neural network).

Unmodeled Dynamics arise when the vehicle model used for control design neglects parts of the real system's dynamics. Examples include the Bell-Hillier stabilizer bar and the flapping dynamics of the helicopter rotor blades. For most autonomous flight control work, the flapping dynamics may be safely neglected, whereas the Bell-Hillier stabilizer bar found on small rotorcraft cannot be ignored when high-bandwidth attitude control is desired. This chapter assumes state-feedback of rigid-body states keeping the control design simpler by leveraging the control design's robustness to unmodeled dynamics. Related adaptive designs that use an output-feedback controller formulation to explicitly deal with unmodeled dynamics are described in [6] with experimental results in [16].

Actuator Magnitude and Rate Saturation limit control authority and thus the closed loop stability and performance. Addressing input dynamics constitutes an important class of control design methods for systems with bounded actuation and include Sontag's universal formula approach using control Lyapunov functions [39] and others [3, 43]. Avoiding saturation, however, usually results in either conservative or highly complex control laws leading to possibly very conservative domains of attraction and slow convergence. See [4] and the references therein for a survey of early work on constrained control.

1.2 Approach

Helicopters have 6 degrees of freedom when considering just the rigid body modes and 4 independent controls are available to control them. Traditionally, the control variables lateral stick, δ_{lat} , longitudinal stick, δ_{lon} and pedal, δ_{ped} , control moments around the roll, pitch and yaw axes respectively. Finally, the collective input, δ_{coll} produces thrust along the main rotor shaft. The rotational dynamics are fully actuated whereas the translational dynamics are underactuated, but controllable. The rotor thrust has to be oriented using the aircraft's pitch and roll attitude to produce translational accelerations.

An overall architecture of the approach is shown in Fig. 1 with details in Fig. 6. The outer loop is responsible for tracking desired translational accelerations. It generates δ_{coll} to vary rotor thrust along the main shaft and also generates the desired roll and pitch angles to orient the thrust vector to generate linear accelerations in these two underactuated degrees of freedom. Note here that the desired pitch and roll angles are commands to the inner-loop

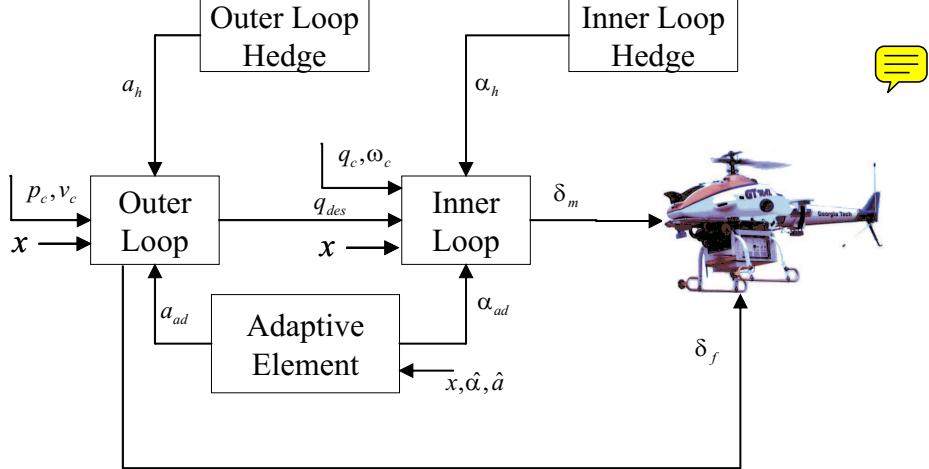


Figure 1: Overall Architecture

controller. In this respect the inner-loop acts like a (virtual) actuator as far as the outer-loop is concerned. Similarly, the inner-loop generates the actuator deflections necessary to control the rotational dynamics. Of course, here the inner-loop’s output actuation signal is subject to the real actuator dynamics of the physical aircraft. In both loops, approximate models of the rotational (inner-loop) and translational (outer-loop) dynamics are dynamically inverted to produce the actuator deflections (and desired pitch and roll) necessary to achieve the desired angular and linear accelerations. These desired accelerations are generated using reference models dictating the desired ideal closed loop response.

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

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

An underlying assumption of this work is that the nonlinear uncertainty can be approximated by a parameterized continuous function over the flight domain of an aircraft. The goal is to capture an approximation of the uncertainty using universal approximators such as neural networks. This universal approximation property guarantees that given a sufficient number neurons, there exists and 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 [26, 30, 27]. 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 [12]. 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 [7].

The adaptive controller described in this chapter has been extensively validated in flight on several aircraft regularly since 2002. The range of aircraft types include the Yamaha RMAX (GTMax) helicopter (Fig. 2), a 11-inch ducted-fan, the GTSpy (Fig. 3), a tail-less fixed-wing aircraft, the D6, and a high thrust-to-weight ratio aircraft, the GTEdge (Fig. 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 the GT Twinstar (Figure 5) a foam built twin-engine aircraft, the GT Logo a small rotorcraft of about 1 meter rotor diameter, and the GTQ [14], a miniature quadrotor. On the GT Twinstar, a variant of the algorithms presented here was used for flight with 25% right wing missing [13, 9].

2 Control of an Air Vehicle

2.1 Vehicle Dynamics

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



$$\dot{p} = v \quad (1)$$

$$\dot{v} = a(p, v, q, \omega, \delta_f, \delta_m) \quad (2)$$

$$\dot{q} = \dot{q}(q, \omega) \quad (3)$$

$$\dot{\omega} = \alpha(p, v, q, \omega, \delta_f, \delta_m), \quad (4)$$

where, $p \in \mathcal{R}^3$ is the position vector, $v \in \mathcal{R}^3$ is the velocity of the vehicle, $q \in \mathcal{R}^4$ is the attitude quaternion and $\omega \in \mathcal{R}^3$ is the angular velocity. Eqn (2) represents translational dynamics and Eqn (4) represents the attitude dynamics. Together, they represent rigid body dynamics and flat-earth kinematics as given in [17] and [50]. Eqn (3) represents the



Figure 2: The GTMax Helicopter



Figure 3: The GTSpy 11-inch ducted fan



Figure 4: The GTEdge aircraft with a high (greater than 1) thrust-to-weight ratio

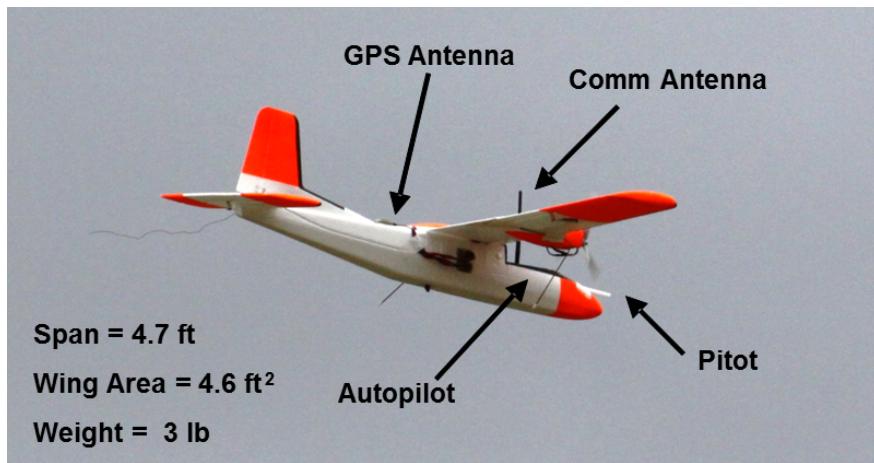


Figure 5: The GTTwinstar foam built twin engine aircraft equipped for fault-tolerant control work (see e.g. [9])

quaternion propagation equations [50]. The use of quaternions, though not a minimal representation of attitude, avoids numerical and singularity problems that Euler-angles-based representations have. This enables the control system to be all attitude capable as required for aggressive maneuvering. The state vector x may now be defined as $x \triangleq [p^T \ v^T \ q^T \ \omega^T]^T$.

Remark 1. *The objective is to design a control system that can track a given position, velocity, attitude and angular rate trajectory. The consolidated trajectory command is given by $[p_c^T \ v_c^T \ q_c^T \ \omega_c^T]^T$.*

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

The consolidated control vector δ is defined as

$$\delta \triangleq [\delta_f^T \ \delta_m^T]^T,$$

the actuators themselves may have dynamics represented by

$$\dot{\delta} = \begin{bmatrix} \dot{\delta}_m \\ \dot{\delta}_f \end{bmatrix} = \begin{bmatrix} g_m(x, \delta_m, \delta_{m_{des}}) \\ g_f(x, \delta_f, \delta_{f_{des}}) \end{bmatrix} = g(x, \delta, \delta_{des}), \quad (5)$$

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

Remark 2. *It is possible to extend the architecture in order to treat actuator dynamics as simply another system in cascade with the translational and attitude dynamics and the control design would include an outer, inner and actuator loop with the actuator loop being the lowest block in the cascade. However unless the physical actuators need to be stabilized, their internal dynamics may be assumed to be asymptotically stable. In this chapter, rate and higher order dynamics are ignored, but magnitude-saturation will be handled explicitly. It can be shown that such an assumption is possible because the control design is robust to the unmodeled dynamics[27].*

2.2 Control Design

The control architecture is based on a model reference adaptive control architecture (see Fig. 6). Noting that Eqn (1) and Eqn (3) represent exactly known kinematics, approximate models for translational acceleration, \hat{a} and a model for angular acceleration, $\hat{\alpha}$ need to be established.

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

Here, a_{des} and α_{des} are commonly referred to as the pseudocontrol and represent desired accelerations. Additionally, $\delta_{f_{des}}$, $\delta_{m_{des}}$, q_{des} are the control inputs and attitude expected to achieve the desired pseudo-control. This form assumes that translational dynamics are coupled strongly with attitude dynamics, as is the case for a helicopter. From the outer-loop's point of view, q (attitude), is like a *virtual actuator* that generates translational accelerations and q_{des} is the desired attitude that the outer-loop inversion expects will contribute towards achieving the desired translational acceleration, a_{des} . The dynamics of q appears like actuator dynamics to the outer loop.

Remark 3. *Although the models are approximate, their functional dependence on vehicle rigid body and actuator states are stated accurately for completeness. It is likely that a specific approximate model that is introduced might drop some of this dependency.*

Remark 4. *The attitude quaternion q_{des} will be used to augment the externally commanded attitude q_c to achieve the desired translational accelerations. Ideally, the trajectory generator would generate a commanded attitude q_c that is consistent with the translational acceleration profile needed to track $x_c(t)$ and $v_c(t)$. If not, the outer-loop inverse takes care of correcting it by an amount necessary to achieve the desired translational accelerations in the longitudinal and lateral directions.*

The models have a functional dependence on current actuator position. Because actuator positions are often not measured on small unmanned aerial vehicles, estimates of the actuator positions $\hat{\delta}_m$, $\hat{\delta}_f$ can be used. When the actuator positions are directly measured, they may be regarded as known $\hat{\delta}_m = \delta_m$ and $\hat{\delta}_f = \delta_f$. In fact, in the outer loop's case, good estimates of the roll and pitch attitude virtual actuators are available using inertial sensors and a navigation filter. The approximate models may now be inverted to obtain the desired control and attitude

$$\begin{bmatrix} \delta_{f_{des}} \\ q_{des} \end{bmatrix} = \begin{bmatrix} \hat{a}_{\delta_f}^{-1}(p, v, a_{des_{\delta_f}}, \omega, \hat{\delta}_m) \\ \hat{a}_q^{-1}(p, v, a_{des_q}, \omega, \hat{\delta}_m) \end{bmatrix} \quad (6)$$

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

with $a_{des_{\delta_f}} + a_{des_q} = a_{des}$, \hat{a}_{δ_f} , \hat{a}_q formulated to be consistent with Eqn (6) and where actuator estimates are given by actuator models

$$\dot{\hat{\delta}} = \begin{bmatrix} \dot{\hat{\delta}}_f \\ \dot{\hat{\delta}}_m \end{bmatrix} = \begin{bmatrix} \hat{g}_f(x, \hat{\delta}_f, \delta_{f_{des}}) \\ \hat{g}_m(x, \hat{\delta}_m, \delta_{m_{des}}) \end{bmatrix} = \hat{g}(x, \hat{\delta}, \delta_{des}). \quad (7)$$

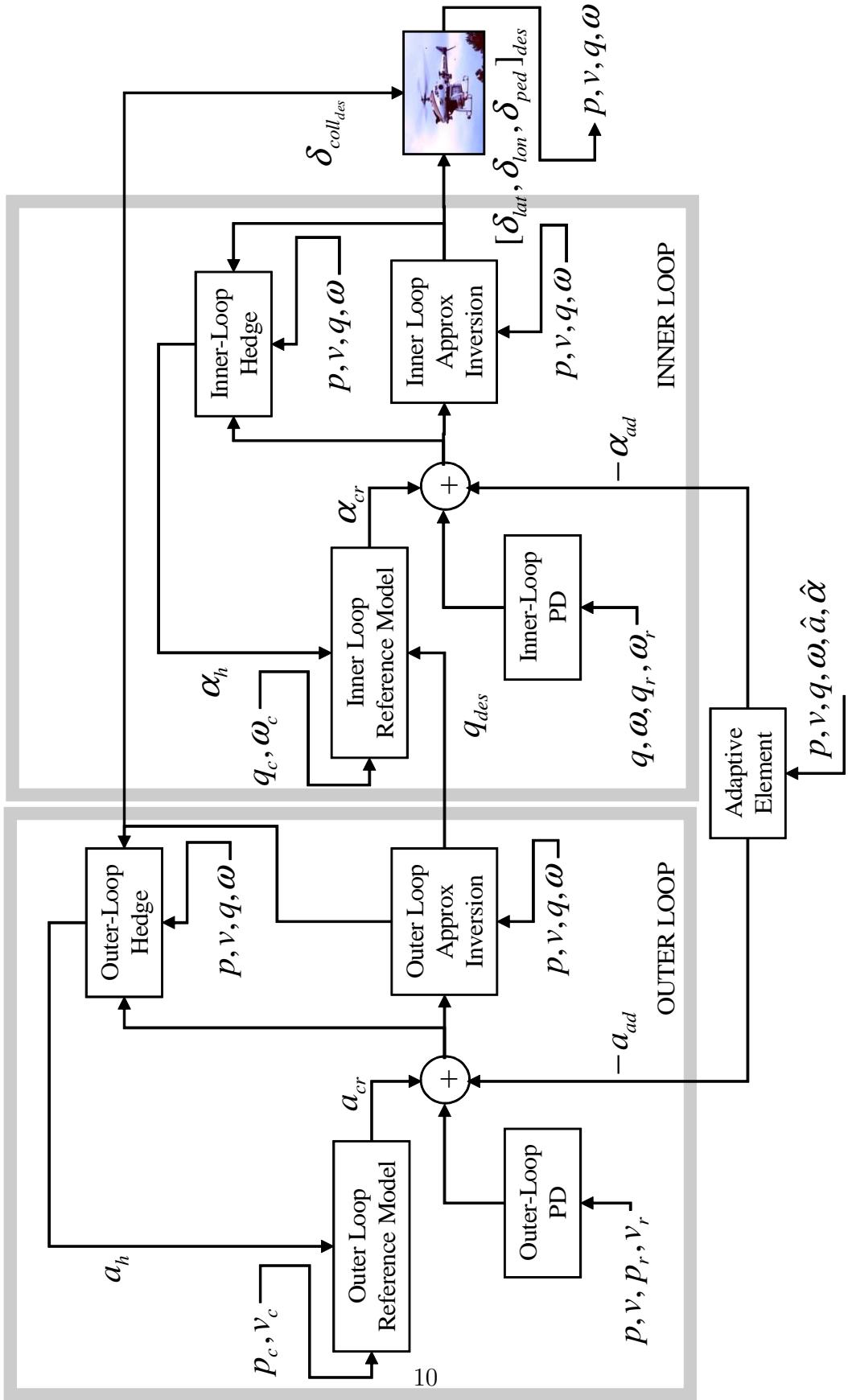


Figure 6: Detailed inner and outer loop controller architecture for an autonomous helicopter.

Introducing the inverse control law Eqn (6) into Eqn (2) and Eqn (4) results in the following closed-loop translational and attitude dynamics

$$\begin{aligned}\dot{v} &= a_{des} + \bar{\Delta}_a(x, \delta, \hat{\delta}) - a_h \\ \dot{\omega} &= \alpha_{des} + \bar{\Delta}_\alpha(x, \delta, \hat{\delta}) - \alpha_h,\end{aligned}\quad (8)$$

where

$$\bar{\Delta}(x, \delta, \hat{\delta}) = \begin{bmatrix} \bar{\Delta}_a(x, \delta, \hat{\delta}) \\ \bar{\Delta}_\alpha(x, \delta, \hat{\delta}) \end{bmatrix} = \begin{bmatrix} a(x, \delta) - \hat{a}(x, \hat{\delta}) \\ \alpha(x, \delta) - \hat{\alpha}(x, \hat{\delta}) \end{bmatrix}, \quad (9)$$

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

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

$$\begin{aligned}a_{des} &= a_{cr} + a_{pd} - \bar{a}_{ad} \\ \alpha_{des} &= \alpha_{cr} + \alpha_{pd} - \bar{\alpha}_{ad},\end{aligned}\quad (10)$$

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

2.3 Reference Model and Hedging

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

If the reference model is designed as

$$\begin{aligned}\dot{v}_r &= a_{cr}(p_r, v_r, p_c, v_c) \\ \dot{\omega}_r &= \alpha_{cr}(q_r, \omega_r, q_c \oplus q_{des}, \omega_c),\end{aligned}\quad (11)$$

where p_r and v_r are the outer-loop reference model states and q_r, ω_r , are the inner-loop reference model states. The external command signal is $x_c = [p_c^T \ v_c^T \ q_c^T \ \omega_c^T]^T$. The attitude rotation desired by the outer loop is now added to the commands for the inner loop controller. Here, $q_c \oplus q_{des}$ denotes quaternion multiplication[49] and effectively concatenates the two rotations.

If tracking error dynamics is computed by subtracting Eqn (10) from Eqn (11), the unachievable acceleration a_h, α_h will appear in the tracking error dynamics. When an adaptive element such as a neural network or integrator is introduced, these effects of input dynamics propagate into the training signal and eventually result in the adaptive element attempting to correct for them leading to incorrect adaptation.

Tackling this issue involves redesigning the reference model by subtracting the deficit accelerations (pseudo-control hedging)

$$\dot{v}_r = a_{cr}(p_r, v_r, p_c, v_c) - a_h \quad (12)$$

$$\dot{\omega}_r = \alpha_{cr}(q_r, \omega_r, q_c \oplus q_{des}, \omega_c) - \alpha_h, \quad (13)$$

a_h and α_h are the differences between commanded pseudocontrol and an estimate of the achieved pseudocontrol. It is an estimate because actual actuator positions may not be known. Additionally, the aircraft state vector p, v, q, ω are estimated using a Kalman Filter [15, 14]. However, for purposes of control design, they are assumed to be known and thus the virtual actuators such as attitude may be assumed to be known in the PCH computation. This assumption may have to be revisited in the case where the control/observer pair is not assumed to be separable, perhaps in a tough localization problem where the control inputs directly affect the observability of the aircraft states.

The PCH signals are given by

$$\begin{aligned} a_h &= \hat{a}(p, v, q_{des}, \omega, \delta_{f_{des}}, \hat{\delta}_m) - \hat{a}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m) \\ &= a_{des} - \hat{a}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m) \end{aligned} \quad (14)$$

$$\begin{aligned} \alpha_h &= \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \delta_{m_{des}}) - \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m) \\ &= \alpha_{des} - \hat{\alpha}(p, v, q, \omega, \hat{\delta}_f, \hat{\delta}_m). \end{aligned} \quad (15)$$

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

The command tracking error may now be defined as e_r

$$e_r \triangleq \begin{bmatrix} p_c - p_r \\ v_c - v_r \\ \tilde{Q}(q_c, q_r) \\ \omega_c - \omega_r \end{bmatrix}, \quad (16)$$

with corresponding command tracking error dynamics given by

$$\dot{e}_r = \begin{bmatrix} v_c - v_r \\ a_c - (a_{cr} - a_h) \\ \omega_c - \omega_r \\ \alpha_c - (\alpha_{cr} - \alpha_h) \end{bmatrix}, \quad (17)$$

The particular form of the reference model dynamics chosen for the translational dynamics, a_{cr} , and attitude dynamics, α_{cr} , has profound effects on the overall response and controllability of the system. This is fully expounded in Chapter 4 of [27] and in [29]. Also see [28] for a discussion on the effects of reference model poles when various elements saturate.

As a summary, three reference models were considered.

- A *Linear Reference Model* will attempt to elicit a linear response in the plant when no such response is possible (peaking) as the plant is nonlinear, especially with the magnitude saturation of actuators.
- The *Nested Saturation-based Reference Model* is an alternative to the linear reference model containing saturations functions appearing in a nested form and is based on the work by Teel[53, 52]. This form allows one to restrict the evolution of states in a prescribable manner.
- The *Constrained Linear Reference Model* is a special case of the nested saturation-based reference model, that is locally linear near the origin.

For the quadratic candidate Lyapunov functions chosen in [27], only the nested-saturation and constrained linear reference models have their Lyapunov derivative bounds on the PCH signals a_h, α_h . In this chapter the constrained reference model is used with equations given later in Section 4.2.

2.4 Tracking error dynamics

The tracking error vector is defined as, e , as

$$e \triangleq \begin{bmatrix} p_r - p \\ v_r - v \\ \tilde{Q}(q_r, q) \\ \omega_r - \omega \end{bmatrix}, \quad (18)$$

where, $\tilde{Q} : \mathcal{R}^4 \times \mathcal{R}^4 \mapsto \mathcal{R}^3$, is a function [25] that, given two quaternions results in an error angle vector with three components. An expression for \tilde{Q} is given by

$$\begin{aligned} \tilde{Q}(p, q) = 2\operatorname{sgn}(q_1 p_1 + q_2 p_2 + q_3 p_3 + q_4 p_4) \times \\ \begin{bmatrix} -q_1 p_2 + q_2 p_1 + q_3 p_4 - q_4 p_3 \\ -q_1 p_3 - q_2 p_4 + q_3 p_1 + q_4 p_2 \\ -q_1 p_4 + q_2 p_3 - q_3 p_2 + q_4 p_1 \end{bmatrix}. \end{aligned} \quad (19)$$

The output of the PD compensators may be written as

$$\begin{bmatrix} a_{pd} \\ \alpha_{pd} \end{bmatrix} = \begin{bmatrix} R_p & R_d & 0 & 0 \\ 0 & 0 & K_p & K_d \end{bmatrix} e, \quad (20)$$

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 whose choice is discussed below. The tracking error dynamics may be found by directly differentiating Eqn (18)

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

Considering \dot{e}_2 ,

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

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

$$\dot{e} = Ae + B \left[\bar{\nu}_{ad} - \bar{\Delta}(x, \delta, \hat{\delta}) \right], \quad (21)$$

where, $\bar{\Delta}$ is given by Eqn (9),

$$\bar{\nu}_{ad} = \begin{bmatrix} \bar{a}_{ad} \\ \bar{\alpha}_{ad} \end{bmatrix}, A = \begin{bmatrix} 0 & I & 0 & 0 \\ -R_p & -R_d & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -K_p & -K_d \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix}. \quad (22)$$

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

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

The adaptive signal $\bar{\nu}_{ad}$ contains two terms

$$\bar{\nu}_{ad} = \nu_{ad} + \nu_r = \begin{bmatrix} a_{ad} + a_r \\ \alpha_{ad} + \alpha_r \end{bmatrix},$$

where ν_{ad} is the output of the Single Hidden Layer (SHL) Neural Network (NN) described in Section A. For an air vehicle with adaptation in all degrees of freedom, $\nu_{ad} \in \mathcal{R}^6$, where the first three outputs, a_{ad} , approximates Δ_a and the last three outputs, α_{ad} , approximate Δ_α and is consistent with the definition of the error in Eqn (18). The term, $\nu_r = [a_r^T, \alpha_r^T]^T \in \mathcal{R}^6$ is a robustifying signal that arises in the proofs of boundedness found in [27].

2.5 Boundedness

Noting that the plant states are given by

$$x(t) = x_r(t) - e(t), \quad (23)$$

boundedness of the reference model states $x_r(t)$ is sufficient to establish boundedness of the plant states $x(t)$. However, the reference model dynamics now includes the PCH signal which could be arbitrary and large. The problem of actuator saturation has effectively been moved from affecting the tracking error dynamics to affecting the command-tracking error dynamics of the reference model. If boundedness of (17) can be established then an assumption that the external command $x_c(t)$ is bounded is sufficient to establish boundedness of the overall system. The following assumptions are required to guarantee boundedness

Assumption 1. *The external command x_c is bounded,*

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

Assumption 2. *The NN approximation $\Delta(x, \hat{\delta}) = \nu_{ad}(x, \hat{\delta}) + \epsilon$ holds in a compact 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 assumption is required to leverage the universal approximation property of SHL NN [23].*

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

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

where $\|\cdot\|_F$ denotes the Frobenius norm. This is justified due to the universal approximation property of SHL NN if the previous assumption holds [23].

Assumption 4. *Note that, Δ depends on ν_{ad} through the pseudocontrol ν , whereas $\bar{\nu}_{ad}$ has to be designed to cancel Δ . Hence the existence and uniqueness of a fixed-point-solution for $\nu_{ad} = \Delta(x, \nu_{ad})$ is assumed. Sufficient conditions[6] for this assumption are also available.*

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

The adaptive element training signal, r , adaptive element output, ν_{ad} , and robustifying term, ν_r , are given by

$$\begin{aligned} r &= (e^T PB)^T \\ \bar{\nu}_{ad} &= \nu_{ad} + \nu_r \\ \nu_{ad} &= W^T \sigma(V^T \bar{x}) \\ \nu_r &= -K_r (\|Z\|_F + \bar{Z}) r \frac{\|e\|}{\|r\|}. \end{aligned}$$

Theorem 1. Consider the system given by (1,2,3,4), with the inverse law (6), reference models (35,36) which is consistent with (12,13), where the gains are the same as those selected such that the system matrix in (21) is Hurwitz and assumptions (1,2,3,4,5) are met. If $K_r > 0 \in \mathcal{R}^{k \times k}$ is chosen sufficiently large with lower-limit stated in the proof, and adaptive element weights W, V satisfy the adaptation laws

$$\begin{aligned}\dot{W} &= -[(\sigma - \sigma' V^T \bar{x})r^T + \kappa \|e\|W] \Gamma_W \\ \dot{V} &= -\Gamma_V [\bar{x}(r^T W^T \sigma') + \kappa \|e\|V],\end{aligned}\tag{24}$$

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

Proof. See proof of Theorem 4 in [27]. \square

Remark 5. The update laws $\dot{W}(t), \dot{V}(t)$, closely resembles the backpropagation method of tuning neural network weights [45, 51, 22, 33]. However, it is important to note that the training signal r is different from that of the backpropagation based learning laws.

3 Concurrent Learning

The single hidden layer Neural-Network based adaptive elements used in this chapter are known to have the universal approximation property [22, 23], i.e., given sufficient number of hidden-layer neurons there exists a set of ideal weights W^*, V^* that brings the neural network output to within an ϵ neighborhood of the modeling error $\bar{\Delta}(x, \delta)$ (uncertainty). The adaptive laws in Eqn (24) are designed to minimize the instantaneous tracking error e . Although Theorem 1 guarantees boundedness of the tracking error e , it cannot be guaranteed that the adaptive weights will approach the ideal weights over the long term during a normal course of operation. It is useful to drive the weights closer towards their ideal values, as the resulting NN representation forms a good approximation of the uncertainty, which can result in improved performance, and can be used for planning and health-monitoring purposes.

One limitation of the adaptive laws in Eqn (24) (without the e -modification term) is that at any instant of time, they are constrained to search for the ideal weights only in the direction of instantaneous tracking error reduction. In that sense these adaptive laws are equivalent to a gradient-descent or a greedy update. Therefore, the adaptive weights may not approach the ideal weights unless all directions in which the weights can evolve to reduce the tracking error are explored infinitely often during the course of operation. Intuitively, this explains why *Persistency of Excitation* is required to guarantee weight convergence for most adaptive laws (see e.g. [5]). The idea in concurrent learning is to use specifically selected and online recorded data to ensure parameter convergence without requiring persistent excitation. If data is recorded when the system states are exciting, and if invariant system properties, such as modeling error information, can be inferred from the recorded

data, then weight convergence can be guaranteed without requiring persistent excitation [8]. In an implementation of a concurrent learning adaptive controller, each measured data point is evaluated to determine whether it should be added to a “history stack”. The maximum number of recorded data points is limited, and when this number is reached, new data points replace old points. Note that the history stack is not intended to be a buffer of last p states. The approximation modeling error at a recorded data point, which is an invariant system property, is inferred from the recorded data point by noting that $\Delta(x_i, \delta_i) \approx \dot{\hat{x}}_i - \nu(x_i, \delta_i)$ where \hat{x}_i is the smoothed estimate of x_i [12, 21]. Adaptation happens concurrently on recorded and current data such that the instantaneous tracking error and the modeling error at all recorded data points simultaneously reduces [12, 8, 10].

It was shown in [8] and [10] that for linearly parameterized uncertainties the requirement on persistency of excitation can be relaxed if online recorded data is used concurrently with instantaneous data for adaptation. If the uncertainty can be linearly parameterized, then

$$\bar{\Delta}(x, \delta) = W^* \phi(x, \delta) + \epsilon(x, \delta) \quad (25)$$

where $W^* \in \mathbb{R}^l$ denotes the ideal weights that guarantee for a given basis function $\phi(x, \delta) \in \mathbb{R}^l$ $\sup_{\delta} \|\epsilon(x, \delta)\| \leq \bar{\epsilon}$ for some positive constant $\bar{\epsilon}$. In this case, the adaptive element can also be linearly parameterized in the form $\nu_{ad} = W^T \phi(x, \delta)$. In certain UAV applications, the basis functions for the modeling error are known (see for example the problem of wingrock control [47]), in which case, the existence of an unknown ideal weight vector W^* can be established such that $\bar{\epsilon} = 0$. The representation in (25) can also be guaranteed for any continuous modeling error approximated over a compact domain if elements of ϕ consist of set of Gaussian radial bases functions and a scalar bias term b_w (see [44, 22]). For either of these linearly parameterized representations of the uncertainty, the following theorem can be proven [7, 10, 11]:

Theorem 2. Consider the system given by (1,2,3,4), with the inverse law (6), reference models (35,36) which is consistent with (12,13), where the gains are the same as those selected such that the system matrix in (21) is Hurwitz. Assume further that the uncertainty is linearly parameterizable using an appropriate set of bases over a compact domain D , and that assumptions (4,5) hold. For each recorded data point j , let, $\epsilon_j(t) = W^T(t) \phi(x_i, \delta_i) - \hat{\Delta}(x_i, \delta_i)$, with $\hat{\Delta}(x_i, \delta_i) = \dot{\hat{x}}_i - \nu(x_i, \delta_i)$. Now consider the following update law for the weights of the RBF NN

$$\dot{W} = -\Gamma_W \sigma(z) e^T P B - \sum_{j=1}^p \Gamma_W \sigma(x_i, \delta_i) \epsilon_j^T, \quad (26)$$

and assume that $Z = [\phi(z_1), \dots, \phi(z_p)]$ and $\text{rank}(Z) = l$. Let B_α be the largest compact ball in D , and assume $\zeta(0) \in B_\alpha$, define $\delta = \max(\beta, \frac{2\|PB\|\bar{\epsilon}}{\lambda_{\min}(Q)} + \frac{p\bar{\epsilon}\sqrt{l}}{\lambda_{\min}(\Omega)})$, and assume that D is sufficiently large such that $m = \alpha - \delta$ is a positive scalar. If the states x_{rm} of the bounded input bounded output reference model of (11) remains bounded in the compact ball $B_m = \{x_{rm} : \|x_{rm}\| \leq m\}$ for all $t \geq 0$ then the tracking error e and the weight error $\tilde{W} = W - W^*$ are uniformly ultimately bounded. Furthermore, if the representation in (25) is exact over the entire operating domain, that is $\bar{\epsilon} = 0$, then the tracking error and weight

error converge exponentially fast to a compact ball around the origin for arbitrary initial conditions, with the rate of convergence directly proportional to the minimum singular value of the history stack matrix Z .

Remark 1. The size of the compact ball around the origin where the weight and tracking error converge is dependent on the representation error $\bar{\epsilon}$ and the estimation error $\check{\epsilon} = \max_i \|\dot{x}_i - \hat{x}_i\|$. The former can be reduced by choosing appropriate number of RBFs across the operating domain, and the latter can be reduced by an appropriate implementation of a fixed point smoother. Note that $\dot{x}(t)$ is not needed at a current instant t . Therefore, an appropriate implementation of a fixed point smoother alleviates several issues faced in estimating $\dot{x}(t)$ by using recorded data before and after a data point is recorded to form very accurate estimates of \dot{x}_i [21, 12].

The history stack matrix $Z = [\phi(z_1), \dots, \phi(z_p)]$ is not a buffer of last p states. It can be updated online by including data points that are of significant interest over the course of operation. In the linearly parameterized case, convergence is guaranteed as soon as the history stack becomes full ranked. New data points could replace existing data points once the history stack reaches a pre-determined size. It was shown in [11] that the rate of convergence of the tracking error and weights is directly proportional to the minimum singular value of Z . This provides a useful metric to determine which data points are most useful for improving convergence. Consequently, an algorithm for adding points that improve the minimum singular value of Z for the case of linearly parameterizable uncertainty was presented in [11]. The main limitation of the linearly parameterized RBF NN representation of the uncertainty is that the RBF centers need to be preallocated over an estimated compact domain of operation D . Therefore, if the system evolves outside of D all benefits of using adaptive control are lost. This can be addressed by evolving the RBF basis to reflect the current domain of operation, a reproducing kernel Hilbert space approach for accomplishing this was presented in [34].

On the other hand, the nonlinearly parameterized NN described in Section A is more flexible: it only requires the uncertainties to be bounded over a compact set, but does not require that the domain of operation be known. However, it is typically more difficult to analyze due to the nonlinear parameterizations. In [12] a concurrent learning adaptive law was proposed for SHL NN, and was validated in flight on the GTMax rotorcraft (see Section 5.5). In particular, the following theorem can be proven [12, 8]

Theorem 3. Consider the system given by (1,2,3,4), with the inverse law (6), reference models (35,36) which is consistent with (12,13), where the gains are the same as those selected such that the system matrix in (21) is Hurwitz and assumptions (1,2,3,4,5) are met. Let $i \in \mathbb{N}$ denote the index of an online recorded data point z_i , define $r_{b_i}(t) = \nu_{ad}(z_i) - \hat{\Delta}(z_i)$, where $\hat{\Delta}(z) = \hat{x}_i - \nu_i$ and \hat{x}_i is the smoothed estimate of \dot{x}_i , and consider the following

adaptive law

$$\begin{aligned}\dot{W}(t) &= -(\sigma(V^T(t)\bar{x}(t)) - \sigma'(V^T(t)\bar{x}(t))V^T(t)\bar{x}(t))r^T(t)\Gamma_w - k\|e(t)\|W(t) \\ &\quad - W_c(t) \sum_{i=1}^p (\sigma(V^T(t)\bar{x}_i) - \sigma'(V^T(t)\bar{x}_i)V^T(t)\bar{x}_i)r_{b_i}^T(t)\Gamma_w,\end{aligned}\tag{27}$$

$$\begin{aligned}\dot{V}(t) &= -\Gamma_V\bar{x}(t)r^T(t)W^T(t)\sigma'(V^T(t)\bar{x}(t)) - k\|e(t)\|V(t) - \\ &\quad V_c(t) \sum_{i=1}^p \Gamma_V\bar{x}_i r_{b_i}^T(t)W^T(t)\sigma'(V^T(t)\bar{x}_i),\end{aligned}\tag{28}$$

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

$$\begin{aligned}W_c &= I - \frac{(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})^T}{(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})^T(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})}, \\ V_c &= I - \frac{\Gamma_V\bar{x}\bar{x}^T\Gamma_V}{\bar{x}^T\Gamma_V\Gamma_V\bar{x}}.\end{aligned}\tag{29}$$

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

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

$$\frac{\|x(t) - x_k\|^2}{\|x(t)\|} \geq \bar{\theta},\tag{30}$$

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

4 Helicopter Specific Design

Consider the application of the combined inner-outer-loop adaptive architecture to the trajectory control of a helicopter. The dynamics [42, 41, 19] of the helicopter may be modeled in the same form as Eqns. (1-4). Most small helicopters include a Bell-Hiller stabilizer bar, which provides provide lagged rate feedback, and is a source of unmodeled dynamics. The nonlinear model used for simulation in this work included the stabilizer bar dynamics. Additionally, blade flapping and other aspects such as gear and engine dynamics were also modeled.

4.1 Approximate Model

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

$$\alpha_{des} = \hat{A}_1 \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \hat{A}_2 \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \hat{B} \left(\underbrace{\begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix}}_{\text{des}} - \underbrace{\begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix}}_{\text{trim}} \right), \quad (31)$$

or,

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

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

$$\delta_{m_{des}} = \hat{B}^{-1}(\alpha_{des} - \hat{A}_1 \omega_B - \hat{A}_2 v_B) + \delta_{m_{trim}}.$$

The translational dynamics may be modeled as a point mass with a thrust vector that may be oriented in a given direction as illustrated in Fig. 7. More involved inverses [40] may be used, but the simple relationships between thrust, attitude and accelerations suffice when used with adaptation

$$a_{des} = \begin{bmatrix} 0 \\ 0 \\ Z_{\delta_{coll}} \end{bmatrix} (\delta_{coll_{des}} - \delta_{coll_{trim}}) + L_{bv} g, \quad (32)$$

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

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

The required collective input may be evaluated as

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

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

$$\Delta\Phi_1 = \frac{a_{des2}}{f_{sf}}, \quad \Delta\Phi_2 = -\frac{a_{des1}}{f_{sf}}, \quad \Delta\Phi_3 = 0, \quad (33)$$

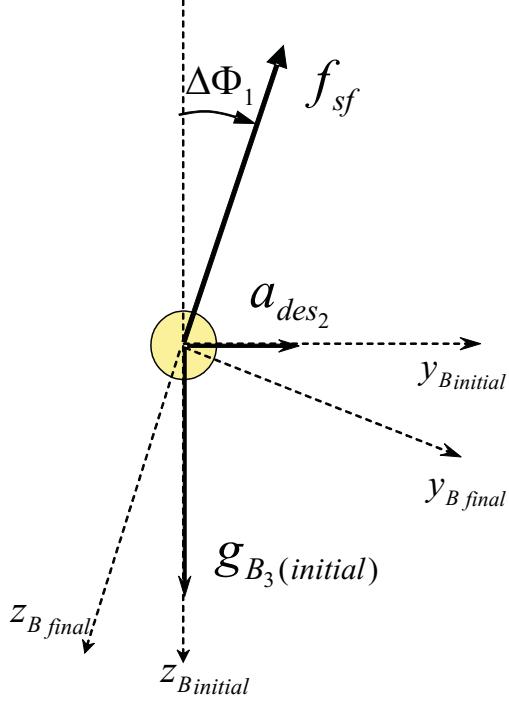


Figure 7: Point mass model for outerloop inversion.

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

$$q_{des} = q(\Delta\Phi_1, \Delta\Phi_2, \Delta\Phi_3), \quad (34)$$

where, $q(\cdot)$ is a function[49] that expresses an euler-angles-based rotation as a quaternion. The overall detailed controller architecture is shown in Fig. 6.

Remark 2. If the desired specific force f_{sf} is close to zero, which occurs when the desired acceleration in the body z axis is the same as the component of gravity vector along that axis, then, Equation (33) is undefined. To overcome this problem, one can impose a restriction where (33) is only computed if $|f_{sf}| > \bar{f}_{sf}$, where $\bar{f}_{sf} > 0$ and is a lower limit. Essentially it means, do not bother using attitude unless the desired specific force is greater than \bar{f}_{sf} .

4.2 Reference Model

Using a linear a_{cr} and α_{cr} in Eqn (12) and Eqn (13) results in the following reference model dynamics

$$\begin{aligned} \dot{v}_r &= R_p(p_c - p_r) + R_d(v_c - v_r) - a_h \\ \dot{\omega}_r &= K_p(\tilde{Q}(q_c \oplus q_{des}, q_r)) + K_d(\omega_c - \omega_r) - \alpha_h, \end{aligned}$$

where, R_p, R_d, K_p, K_d are the same gains used for the PD compensator in Eqn (20). If limits on the angular rate or translational velocities are to be imposed, then they may be easily included in the reference model dynamics by choosing the following constrained linear reference for a_{cr} and α_{cr} .

$$a_{cr} = R_d[v_c - v_r + \sigma(R_d^{-1}R_p(p_c - p_r), v_{lim})] \quad (35)$$

$$\alpha_{cr} = K_d[\omega_c - \omega_r + \sigma(K_d^{-1}K_p\tilde{Q}(q_c \oplus q_{des}, q_r), \omega_{lim})]. \quad (36)$$

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

Remark 3. Note that there are no limits placed on the externally commanded position, velocity, angular rate or attitude. For example, in the translational reference model, if a large position step is commanded, $p_c = [1000, 0, 0]^T \text{ft}$ and $v_c = [0, 0, 0]^T \text{ft/s}$, the speed at which this large step will be achieved is v_{lim} . On the other hand if $p_c = \int v_c dt$ and $v_c = [60, 0, 0]^T \text{ft/s}$, the speed of the vehicle will be 60ft/s . Similarly, ω_{lim} dictates how fast large attitude errors will be corrected. Additionally, aggressiveness with which translational accelerations will be pursued by tilting the body may be governed by limiting the magnitude of q_{des} to the scalar limit q_{lim} .

4.3 Choice of Gains Linear Dynamics

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

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

$$\ddot{x} = a_{des} = \ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x) \quad (37)$$

$$\ddot{\theta} = \alpha_{des} = \ddot{\theta}_g + K_d(\dot{\theta}_g - \dot{\theta}) + K_p(\theta_g - \theta), \quad (38)$$

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

When contributions of $\dot{\theta}_g(s)$ and $\ddot{\theta}_g(s)$, are ignored, the pitch dynamics Eqn (38) may be rewritten in the form of a transfer function as

$$\theta(s) = \frac{\theta(s)}{\theta_g(s)}\theta_g(s) = \frac{K_p}{s^2 + K_d s + K_p}\theta_g(s). \quad (39)$$

If the outer-loop linearizing transformation used to arrive at Eqn (37) has the form $\ddot{x} = f\theta$, where $f = -g$ and g is gravity, it may be written as

$$s^2x(s) = f\theta(s). \quad (40)$$

The outer-loop attitude command may be generated as

$$\theta_{des} = \frac{\ddot{x}_{des}}{f} = \frac{a_{des}}{f}. \quad (41)$$

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

$$\theta_g = \theta_{des} = \frac{1}{f} [\ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x)]. \quad (42)$$

When Eqn (39) and Eqn (42) are used in Eqn (40)

$$s^2x(s) = \frac{K_p [s^2x_c + R_d s(x_c - x) + R_p(x_c - x)]}{s^2 + K_d s + K_p}, \quad (43)$$

Rearranging the above equation results in the following transfer function

$$\frac{x(s)}{x_c(s)} = \frac{K_p s^2 + K_p R_d s + K_p R_p}{s^4 + K_d s^3 + K_p s^2 + K_p R_d s + K_p R_p}. \quad (44)$$

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

$$\begin{aligned} \Upsilon(s) &= (s^2 + 2\zeta_o\omega_o + \omega_o^2)(s^2 + 2\zeta_i\omega_i + \omega_i^2) \\ &= s^4 + (2\zeta_i\omega_i + 2\zeta_o\omega_o)s^3 \\ &\quad + (\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2)s^2 \\ &\quad + (2\zeta_o\omega_o\omega_i^2 + 2\omega_o^2\zeta_i\omega_i)s + \omega_o^2\omega_i^2, \end{aligned} \quad (45)$$

where, the subscripts i, o , represent the inner and outerloop values respectively.

Comparing the coefficients of the poles of Eqn (44) and Eqn (45) allows the gains to be expressed as a function of the desired pole locations for each axis in turn

$$\begin{aligned} R_p &= \frac{\omega_o^2\omega_i^2}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2} \\ R_d &= 2 \frac{\omega_o\omega_i(\zeta_o\omega_i + \omega_o\zeta_i)}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2} \\ K_p &= \omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2 \\ K_d &= 2\zeta_i\omega_i + 2\zeta_o\omega_o. \end{aligned} \quad (46)$$

Additionally, the zeros of the transfer function given by Eqn (44) affect the transient response. Thus, $\omega_i, \zeta_i, \omega_o, \zeta_o$ must be selected such that performance is acceptable.

4.4 Imposing Response Characteristics

The methods presented in this chapter do not contain assumptions that limit its application to unmanned helicopters. Manned rotorcraft normally have to meet standards, such as those specified in the Aeronautical Design Standard-33 [2] handling qualities specifications. Control system performance[36, 46] may be evaluated by imposing response requirements and computing metrics prescribed in the ADS-33. When there is no saturation, the hedging signals a_h, α_h are zero. When it is assumed that the adaptation has reached its ideal values of (V^*, W^*) , then

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

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

5 Experimental Results

The proposed guidance and control architecture was applied to the Georgia Institute of Technology Yamaha R-Max helicopter (GTMax) shown in Fig. 2. The GTMax helicopter weighs about $157lb$ and has a main rotor radius of $5.05ft$. Nominal rotor speed is 850 revolutions per minute. Its practical payload capability is about $66lbs$ with a flight endurance of greater than 60 minutes. It is also equipped with a Bell-Hillier stabilizer bar. Its avionics package includes a Pentium 266 flight control computer, an inertial measurement unit (IMU), a global positioning system, a 3-axis magnetometer and a sonar altimeter. The control laws presented in this chapter were first implemented in simulation [31] using a nonlinear helicopter model that included flapping and stabilizer bar dynamics. Wind and gust models were also included. Additionally, models of sensors with associated noise characteristics were implemented. Many aspects of hardware such as the output of sensor model data as serial packets was simulated. This introduced digitization errors as would exist in real-life and also allowed testing of many flight specific components such as sensor drivers. The navigation system [15] consists of a 17-state Kalman filter to estimate variables such as attitude, and terrain altitude. The navigation filter was executed at $100Hz$ and corresponds to the highest rate at which the IMU is able to provide data. Controller calculations occurred at $50Hz$. The control laws were first implemented as C-code and tested in simulation. Because almost all aspects specific to flight-testing were included in the simulation environment, a subset of the code from the simulation environment was implemented on the main flight computer. During flight, ethernet and serial-based data links provided a link to the ground station computer that allowed monitoring and uploading of way-points. A simple kinematics-based trajectory generator (with limits on accelerations) was used to generate smooth consistent trajectories $(p_c, v_c, q_c, \omega_c)$ for the controller. Various moderately aggressive maneuvers were performed

during flight to test the performance of the trajectory-tracking controller. Controller testing began with simple hover followed by step responses and way-point navigation. Following initial flight tests, aggressiveness of the trajectory was increased by relaxing acceleration limits in the trajectory generator and relaxing ω_{lim} and v_{lim} in the reference models. Tracking error performance was increased by increasing the desired bandwidth of the controllers. Selected results from these flight tests are provided in the following sections.

5.1 Parameter Selections

The controller parameters for the inner loop involved choosing K_p, K_d based on a natural frequency of $2.5, 2, 3 \text{ rad/s}$ for the roll, pitch and yaw channels respectively and damping ratio of 1.0. For the outer loop, R_p, R_d were chosen based on a natural frequency of $2, 2.5, 3 \text{ rad/s}$ for the x, y and z body axis all with a damping ratio of unity. The NN was chosen to have 5 hidden layer neurons. The inputs to the network included body axis velocities and rates as well as the estimated pseudocontrols i.e, $x_{in} = [v_B^T, \omega_B^T, \hat{a}^T, \dot{\hat{a}}^T]$. The output layer learning rates Γ_W were set to unity for all channels and a learning rate of $\Gamma_V = 10$ was set for all inputs. Limits on maximum translation rate and angular rate in the reference model dynamics were set to $v_{lim} = 10 \text{ ft/s}$ and $\omega_{lim} = 2 \text{ rad/s}$. Additionally, attitude corrections from the outer loop, q_{des} were limited to 30 degrees.

With regard to actuator magnitude limits, the helicopter has a radio-control transmitter that the pilot may use to fly the vehicle manually. The full deflections available on the transmitter sticks in each of the channels were mapped as $\delta_{lat}, \delta_{lon}, \delta_{ped} \in [-1, 1]$ corresponding to the full range of lateral tilt and longitudinal tilt of the swash plate and full range of tail rotor blade pitch. The collective was mapped as $\delta_{coll} \in [-2.5, 1]$, corresponding to the full range of main rotor blade pitch available to the human pilot. The dynamic characteristics of the actuators were not investigated in detail. Instead, conservative rate limits were artificially imposed in software. Noting that $\delta = [\delta_{coll}, \delta_{lat}, \delta_{lon}, \delta_{ped}]^T$, the actuator model used for PCH purposes as well as artificially limiting the controller output has form

$$\dot{\hat{\delta}} = \lim_{\lambda \rightarrow +\infty} \sigma \left(\lambda(\sigma(\delta_{des}, \delta_{min}, \delta_{max}) - \hat{\delta}), \dot{\delta}_{min}, \dot{\delta}_{max} \right), \quad (47)$$

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

$$\hat{\delta}[k+1] = \sigma \left(\hat{\delta}[k] + \sigma \left(\sigma(\delta_{des}, \delta_{min}, \delta_{max}) - \hat{\delta}[k], \Delta T \dot{\delta}_{min}, \Delta T \dot{\delta}_{max} \right), \delta_{min}, \delta_{max} \right), \quad (48)$$

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

$$\begin{aligned} \delta_{min} &= [-2.5, -1, -1, -1]^T \\ \delta_{max} &= [1, 1, 1, 1]^T \end{aligned} \quad (49)$$

units, and the rate limits were set to

$$\begin{aligned} \dot{\delta}_{min} &= [-4, -2, -2, -2]^T \\ \dot{\delta}_{max} &= [4, 2, 2, 2]^T \end{aligned} \quad (50)$$

units per second.

5.2 Flight Test

Finally, the controller was flight tested on the GTMax helicopter shown in Fig. 2. A lateral position step response is shown in Fig. 8. The vehicle heading was regulated due-north during this maneuver. Lateral control deflections during the maneuver were recorded and are also shown. A step heading command response and pedal control history is shown in Fig. 9. It should be noted that during flight tests, states were sampled at varying rates in order to conserve memory and datalink bandwidth. The trajectory commands p_c, v_c, q_c, ω_c were sampled at 1Hz, actuator deflections $\delta_{coll}, \delta_{lon}, \delta_{lat}$ and δ_{ped} were sampled at 50Hz, vehicle position and speed was sampled at 50Hz. Since the command vector is sampled at a low rate (1Hz), a step command appears as a fast ramp in figures.

During takeoff and landing phases a range sensor (sonar) is used to maintain and update the estimated local terrain altitude in the navigation system. The sonar is valid up to 8ft above the terrain, sufficient for landing and takeoff purposes. Fig. 10 illustrates the altitude and collective profile during a landing. The vehicle starts at an initial hover at 300ft, followed by a descent at 7ft/s until the vehicle is 15ft above the estimated terrain. The vehicle then descends at 0.5ft/s until weight-on-skids is automatically detected at which point the collective is slowly ramped down. Automatic takeoff (Fig. 11) is similar where the collective is slowly ramped up until weight-on-skids is no longer detected. It should be noted that NN adaptation is active at all times except when weight-on-skids is active. Additionally, when weight is on skids, the collective ramp-up during takeoff and ramp-down during landing is open-loop.

The approximate model used to compute the dynamic inverse (Eqn (32) and Eqn (31)) is based on a linear model of the dynamics in hover. To evaluate controller performance at different points of the envelope, the vehicle was commanded to track a trajectory that accelerated up to a speed of 100ft/s. To account for wind, an upwind and downwind leg were flown. In the upwind leg the vehicle accelerated up to 80ft/s and during the downwind leg the vehicle accelerated up to a speed of 97ft/s as shown in Fig. 12. Collective and longitudinal control deflections are also shown. In the upwind leg, the collective is saturated and the vehicle is unable to accelerate further. The longitudinal control deflections behave nominally as the vehicle accelerates and decelerates through a wide range of the envelope. The NN is able to adapt to rapidly changing flight conditions, from the baseline inverting design at hover through to the maximum speed of the aircraft. A conventional proportional-integral-derivative design would have required scheduling of gains throughout the speed range. More significantly, classical design would require accurate models at each point, unlike this design which does not. In addition to flight at high speeds, tracking performance was evaluated at moderate speeds, where a square pattern was flown at 30ft/s for which position tracking is shown in Fig. 13. External command position tracking errors are shown in Fig. 14 with a peak total position error 3.3ft and standard deviation of 0.8ft.

Many maneuvers such as high-speed flight are quasi steady, in the sense that once in the maneuver, control deflection changes are only necessary for disturbance rejection. To

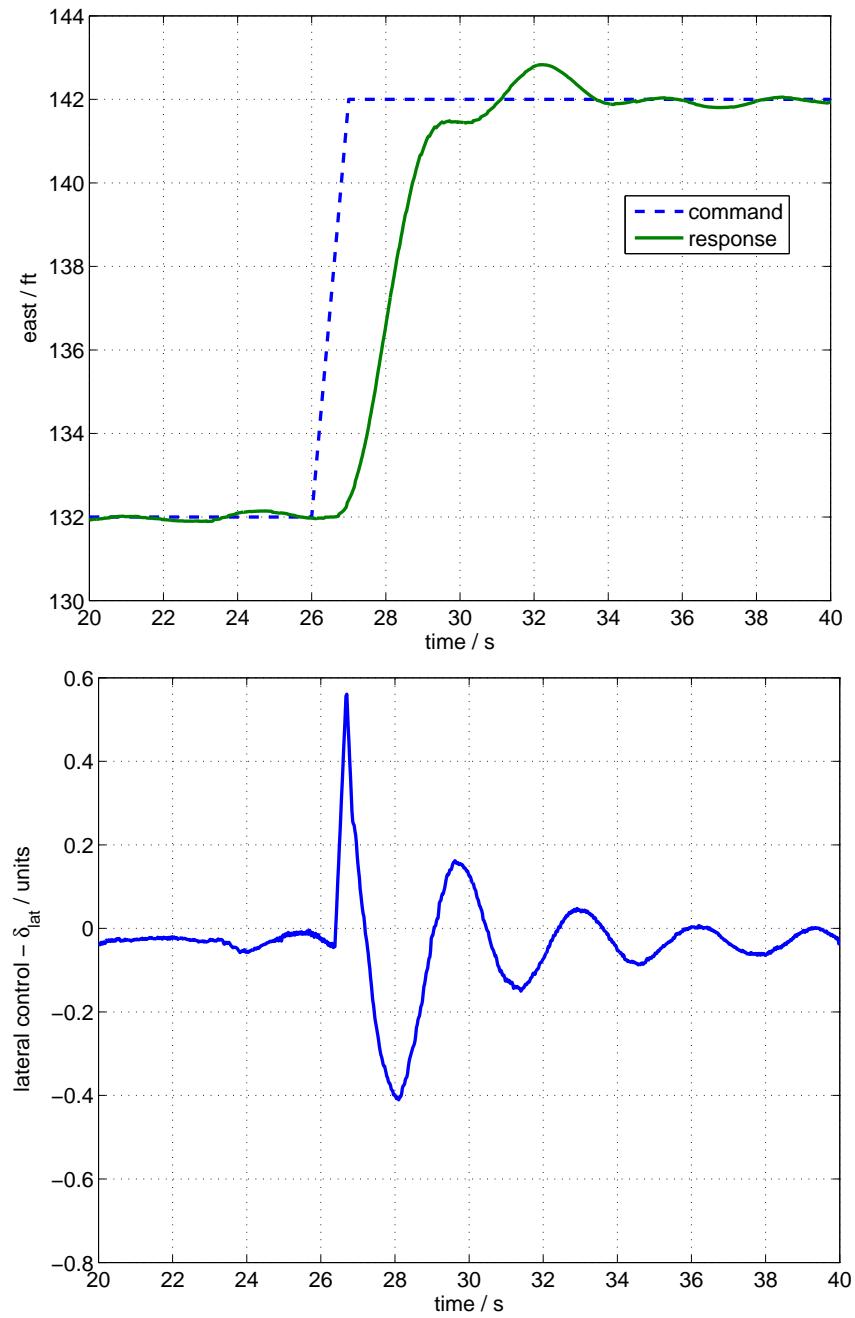


Figure 8: Response to a 20ft step in the lateral direction.

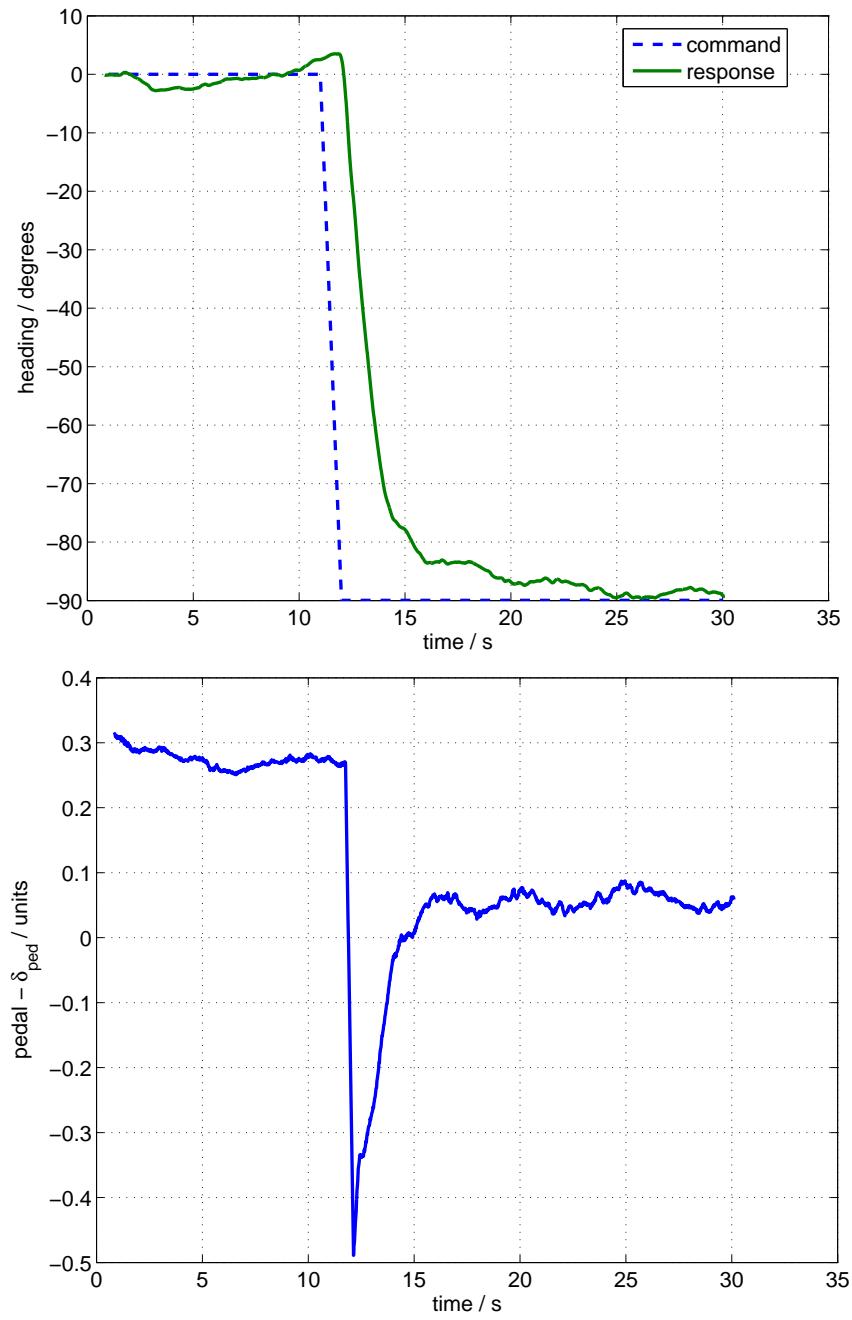


Figure 9: Response to a 90 degree heading command.

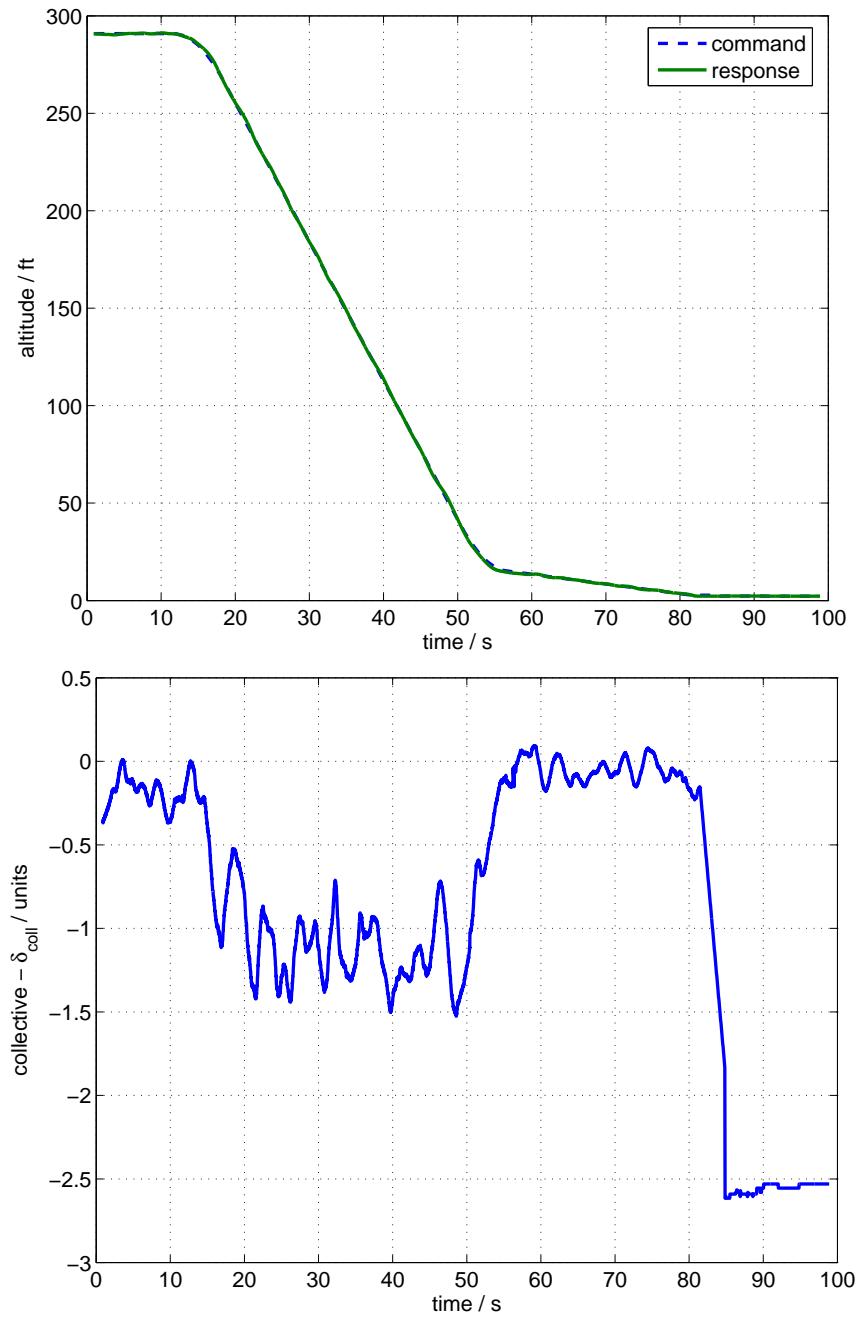


Figure 10: Automatic landing maneuver.

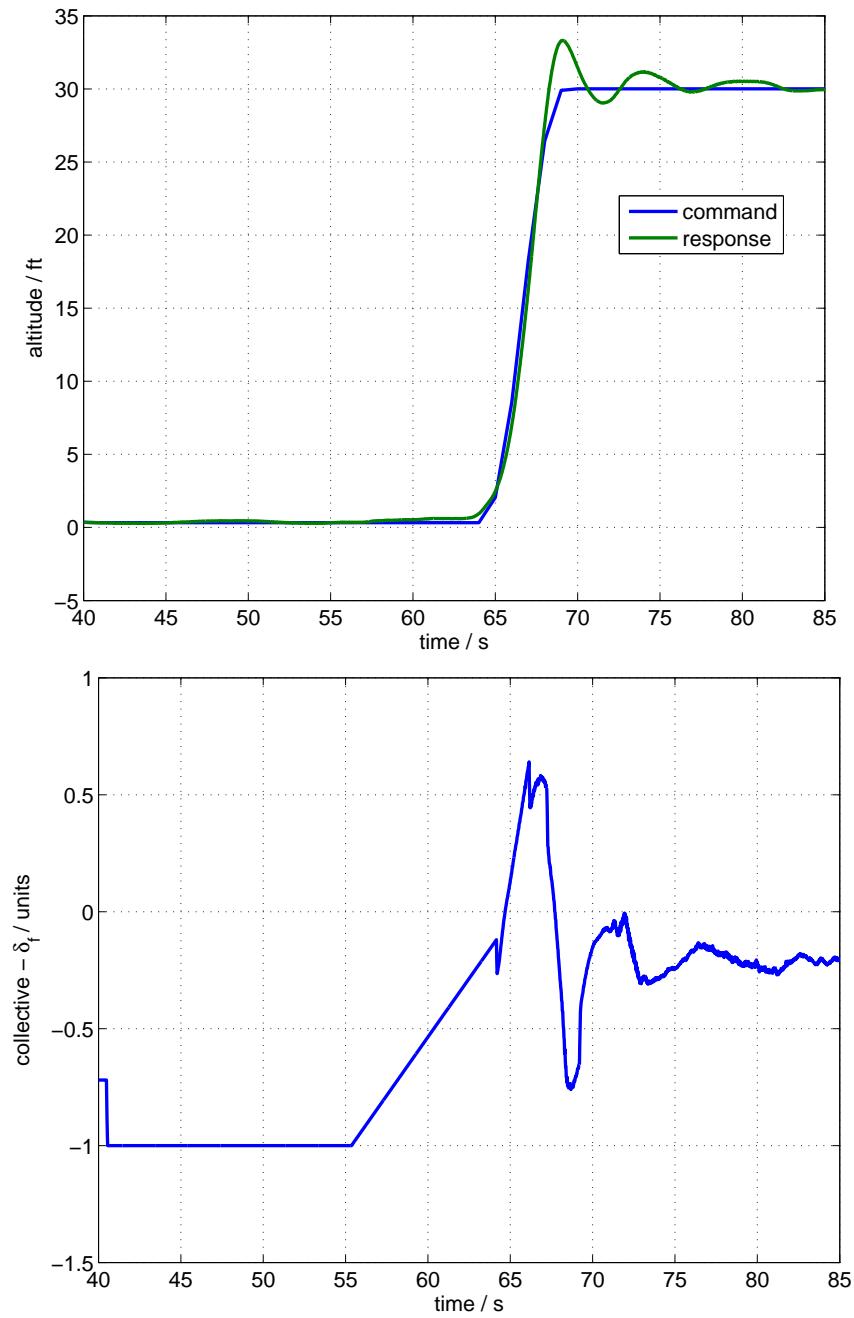


Figure 11: Automatic take-off maneuver.

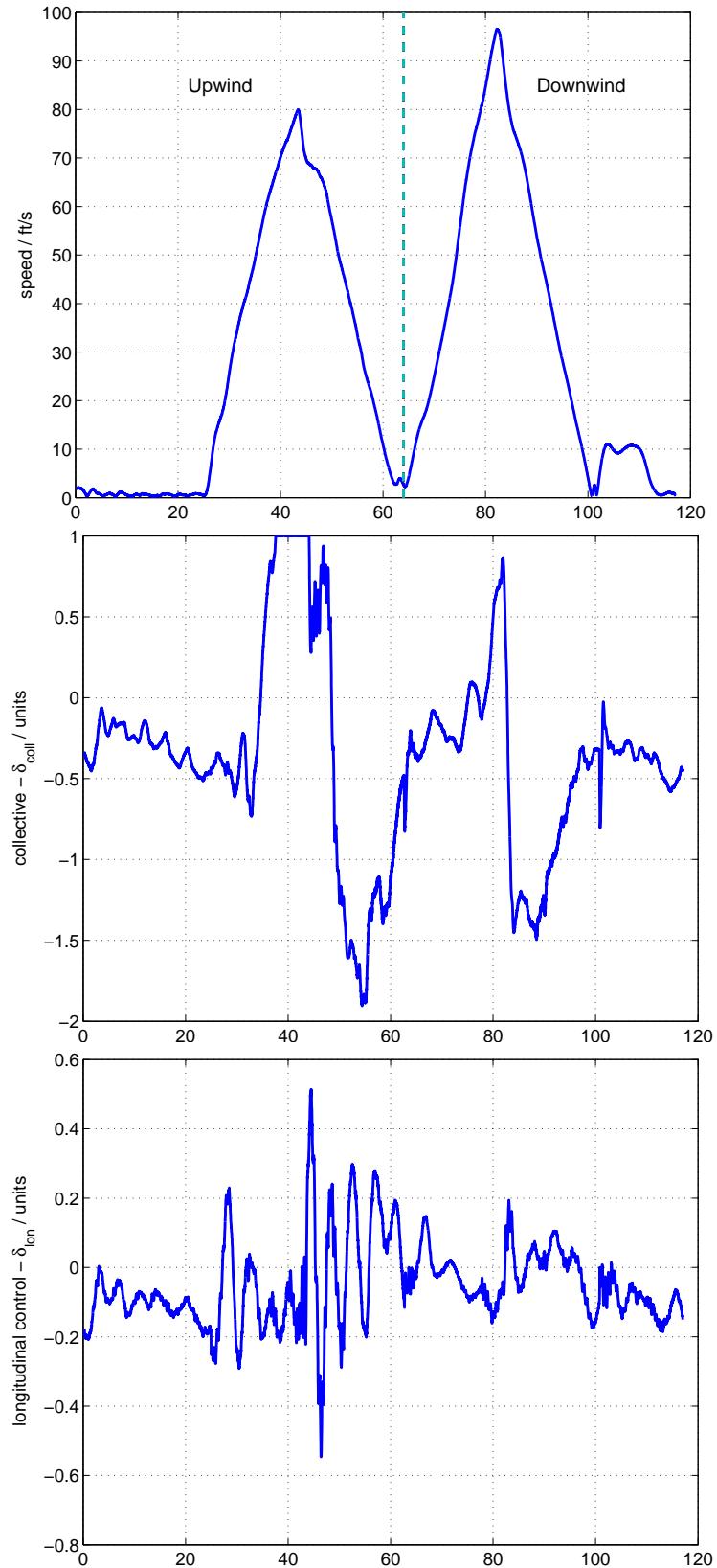


Figure 12: High speed forward flight up to 97 ft/s.
31

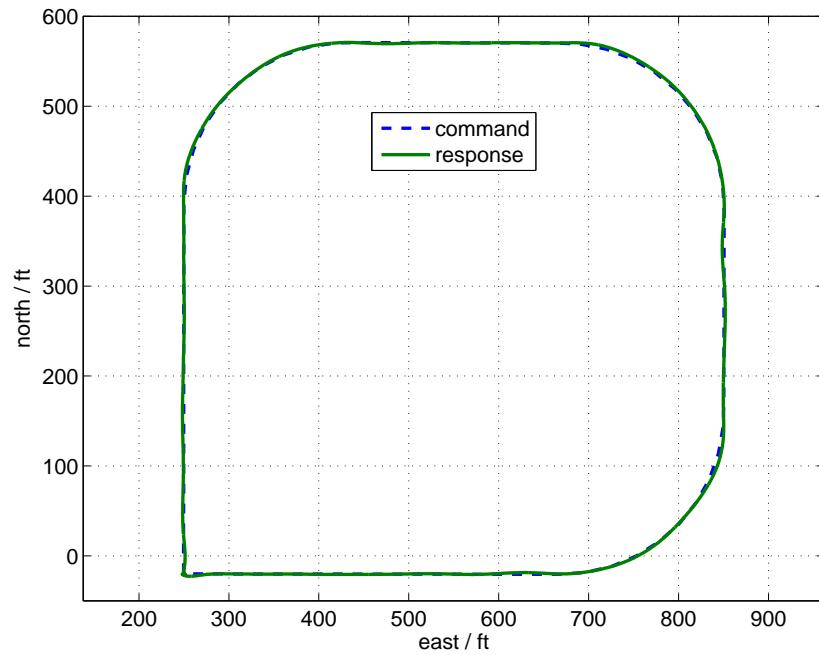


Figure 13: Flying a square pattern at 30ft/s .

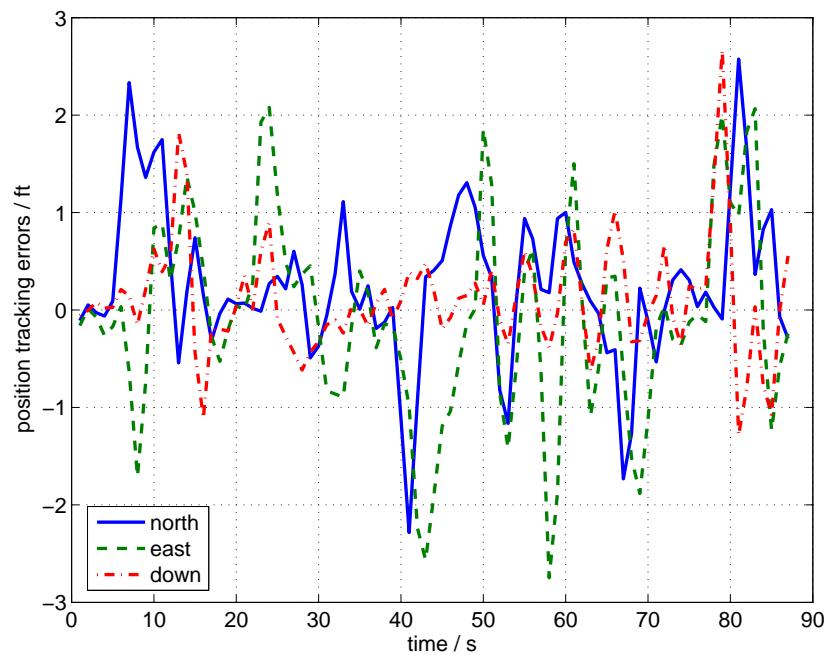


Figure 14: Command tracking errors while flying a square pattern at 30ft/s .

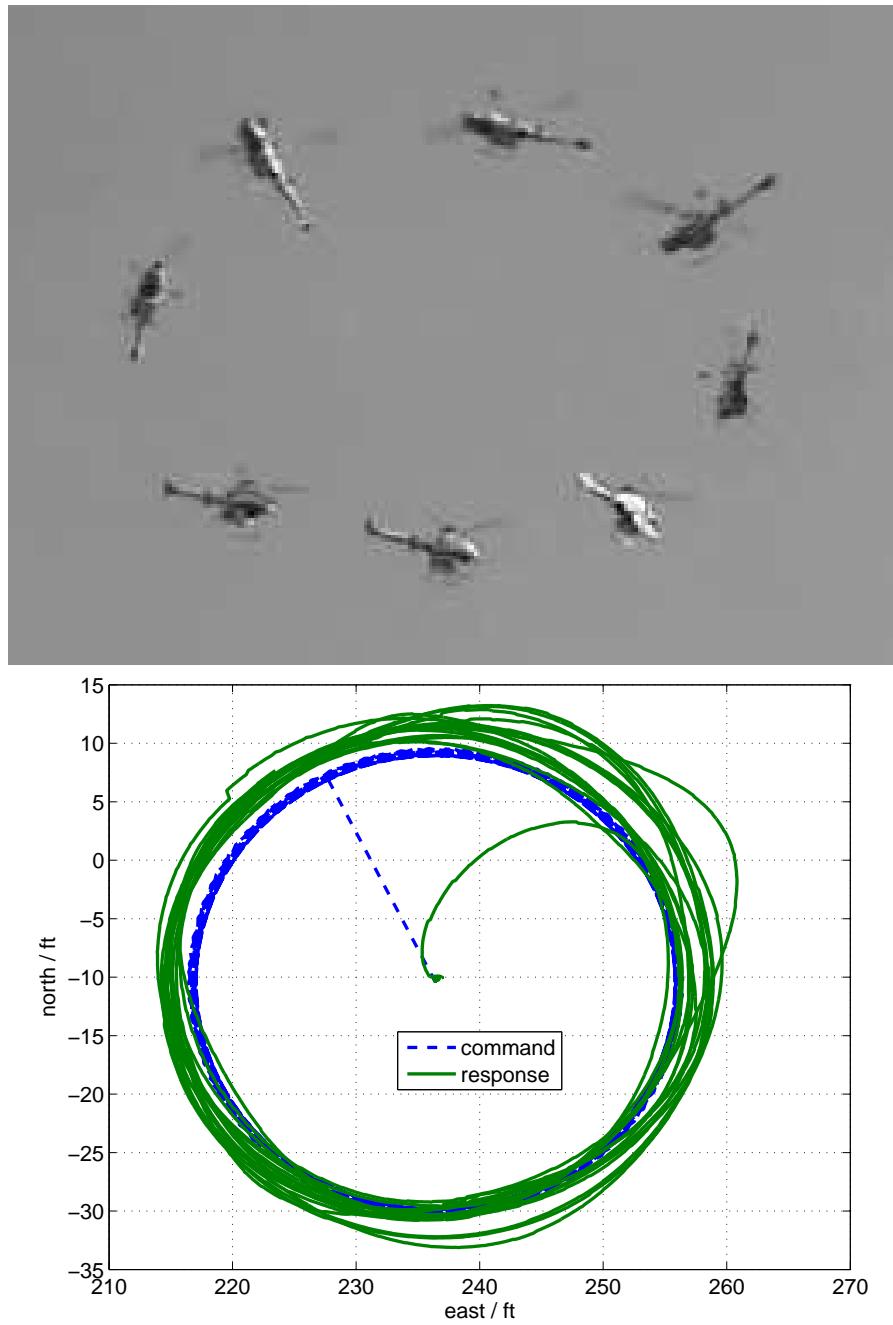


Figure 15: Circular maneuver, with 360° heading changes during the circuit.

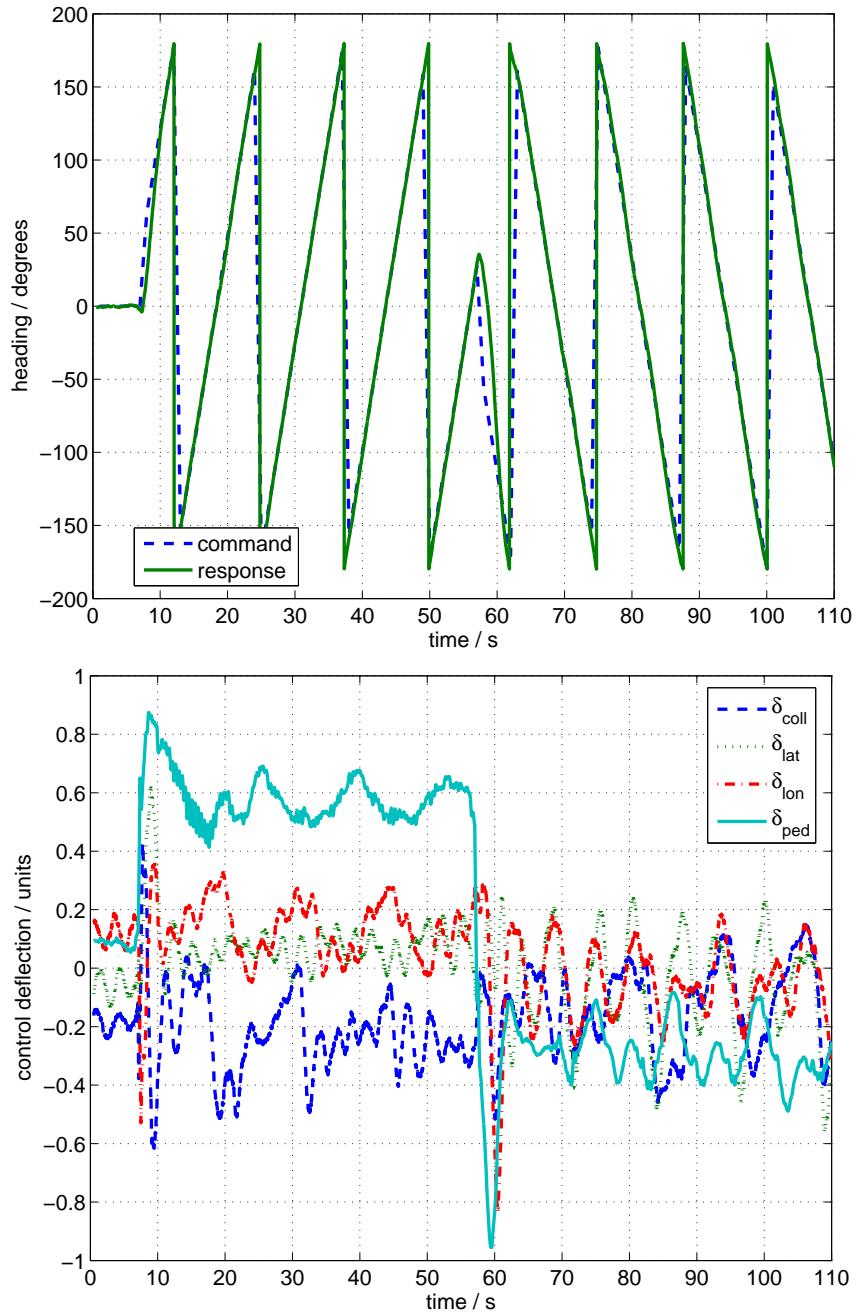


Figure 16: Heading tracking during circular maneuver and control time history.

evaluate performance where the controls have to vary significantly in order to track the commanded trajectory, the helicopter was commanded to perform a circular maneuver in the north-east plane with constant altitude and a constantly changing heading. The trajectory equations for this maneuver are given by

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

$$\psi_c = \omega t f,$$

where, t is current time and h is a constant altitude command. V is speed of the maneuver, ω is angular speed of the helicopter around the maneuver origin, and f is number of 360° changes in heading to be performed per circuit. If $\omega = \pi/2 \text{ rad/s}$, the helicopter will complete the circular circuit once every 4 seconds. If $f = 1$, the helicopter will rotate anticlockwise 360° once per circuit. Fig. 15 shows the response to such a trajectory with parameters $\omega = 0.5 \text{ rad/s}$, $f = 1$, $V = 10 \text{ ft/s}$. After the initial transition into the circular maneuver, the tracking is seen to be within 5 ft. To visualize the maneuver easily, superimposed still images of the vehicle during the circular maneuver are shown. Both anticlockwise and clockwise heading changes during the maneuver were tested by changing the parameter from $f = 1$ (anticlockwise) to $f = -1$ (clockwise) at $t = 55 \text{ s}$. Fig. 16 shows that heading tracking is good in both cases. The time history of the pedal input δ_{ped} and all other controls during the maneuver is also shown and illustrates how the vehicle has to exercise all of its controls during this maneuver.

Next, the ability of the controller to track a previous manually-flown maneuver was tested. First, a human pilot flew a figure eight, 3-dimensional pattern with the vehicle. Vehicle state was recorded and was then played back as commands to the adaptive controller. A 3D plot of the pilot and controller flown trajectories are shown in Fig. 17 along with projected ground track. Overall, the tracking in position was measured to be within 11.3 ft of the desired pilot flown trajectory with a standard deviation of 4.7 ft .

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. 18 along with the pitch angle. A time history of the altitude and the collective control deflection is shown in Fig. 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. 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

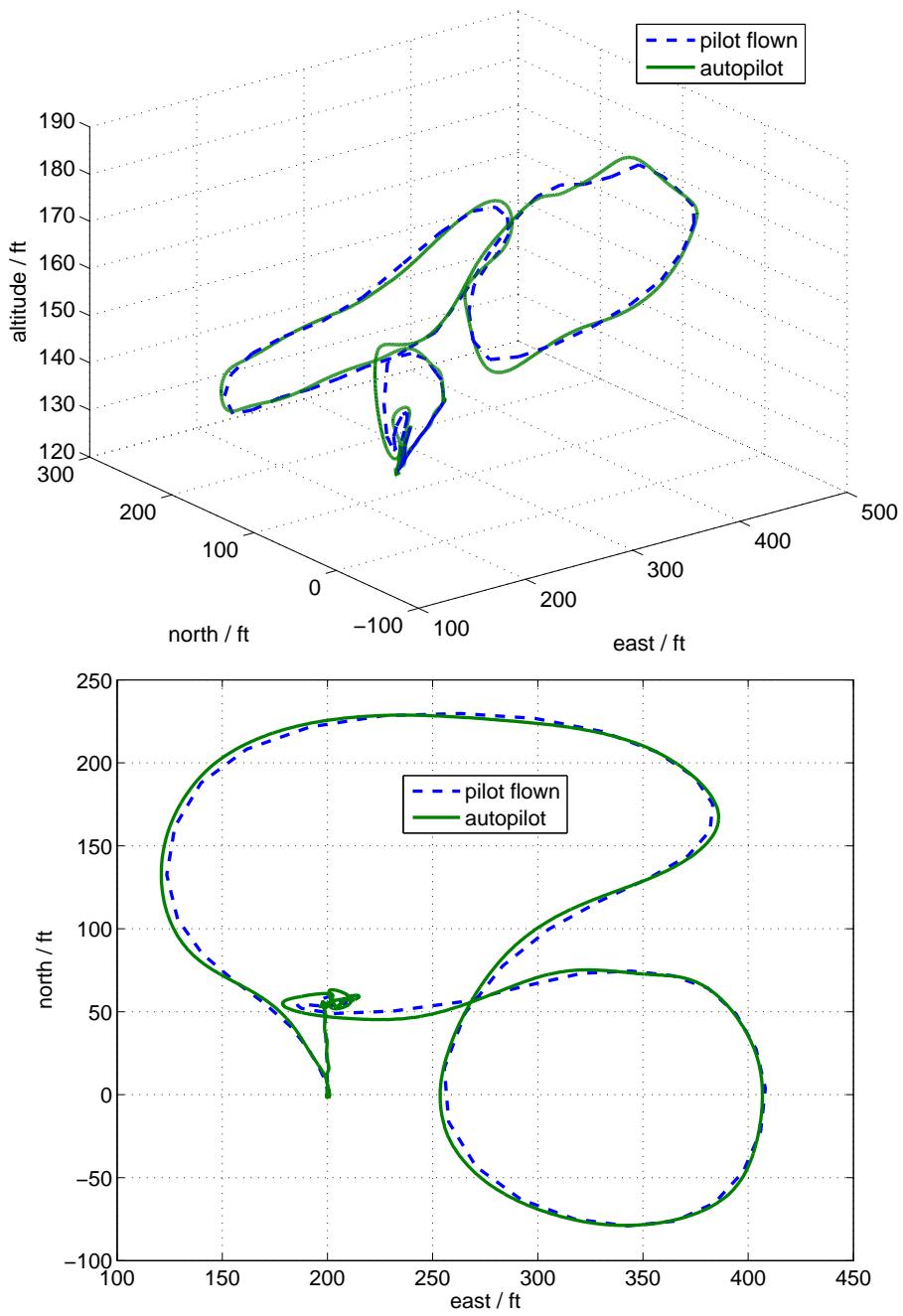


Figure 17: A 3D view and ground track view, of a trajectory initially flown manually by a pilot and then tracked by the controller.

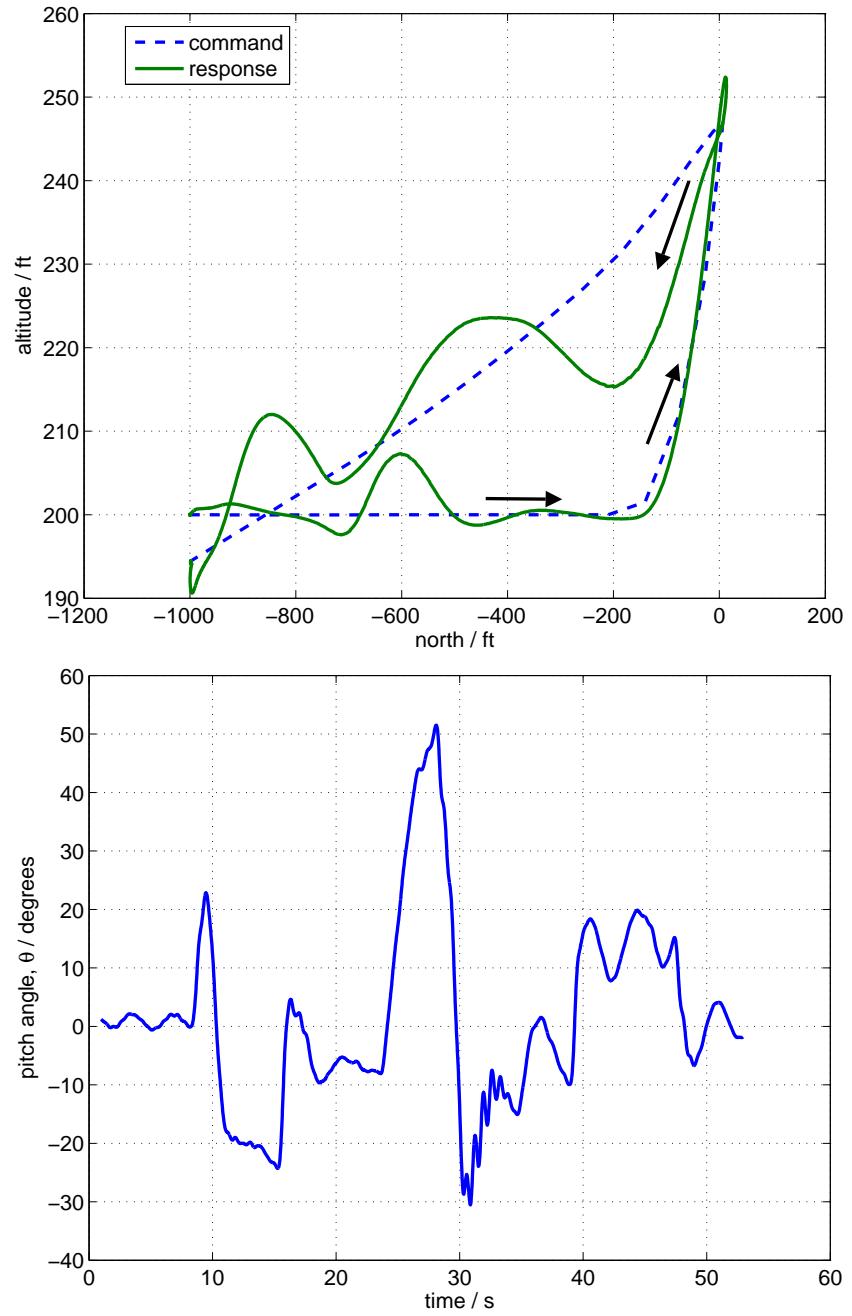


Figure 18: North-Altitude and pitch angle profile during a 180° velocity change maneuver.
Note: North axis and Altitude axis scales are not equal.

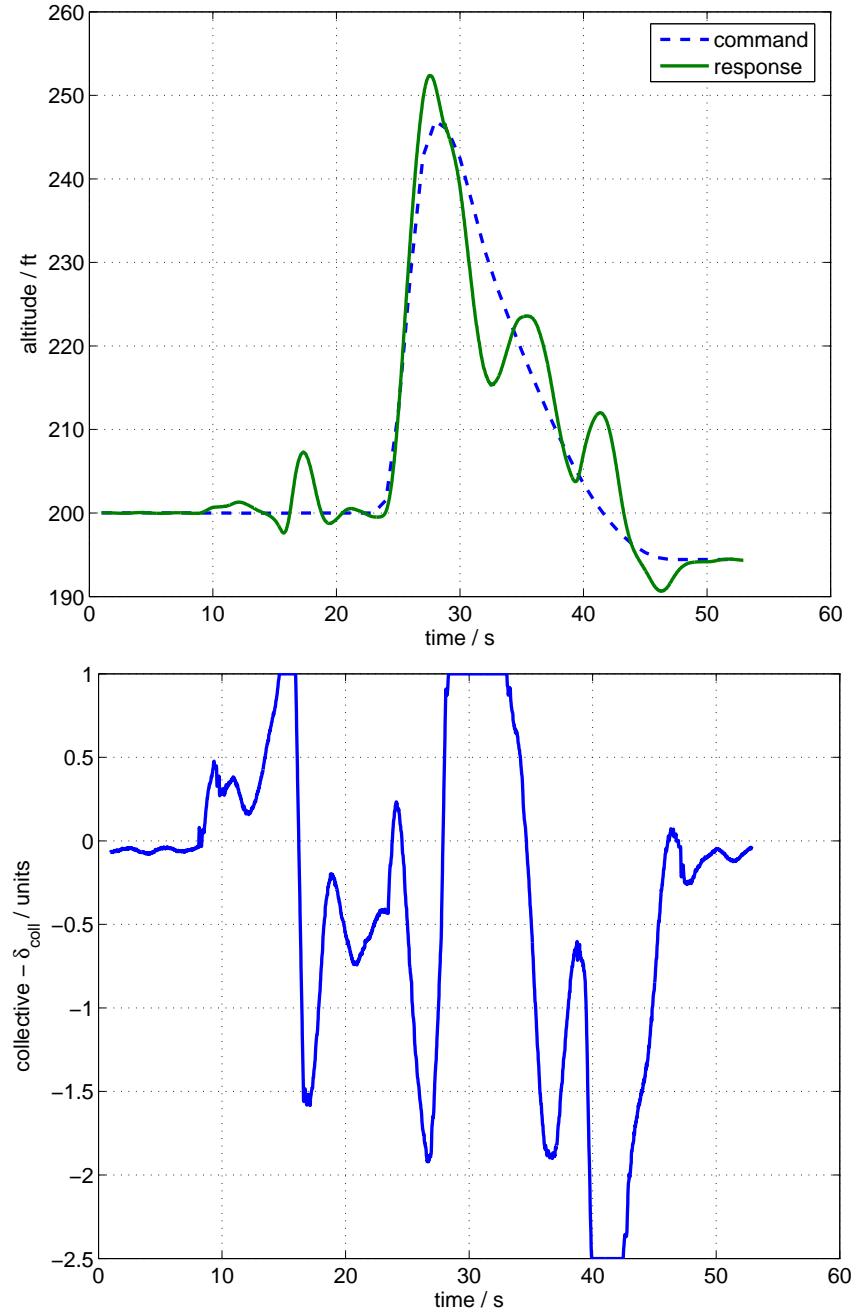


Figure 19: Altitude and collective control history during a 180° velocity change maneuver.

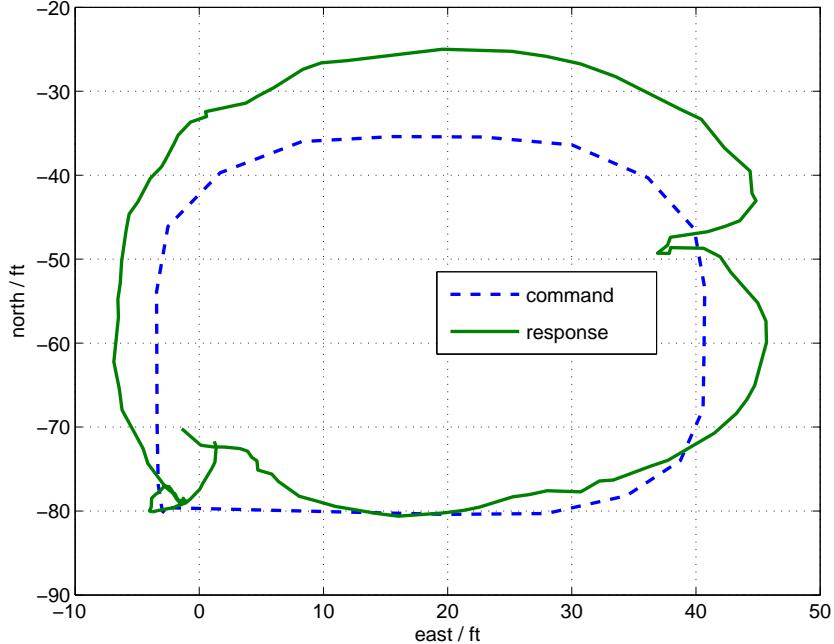


Figure 20: The GTSpy performing a box maneuver

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.

5.3 Application to a Ducted Fan

Following tests on the GTMax helicopter, the control method presented in this chapter was applied to other smaller aircraft. The algorithms were ported to a custom DSP/FPGA hardware device (the FCS20) along with a small sensor board that contained gyroscopes and accelerometers for inertial sensing and a GPS. The avionics package weighed less than *1lb* and fell within the payload capacity of the 11-inch ducted fan (GTSpy). The GTSpy has a maximum take-off weight of *5.5lbs* and is driven by a two-bladed fixed-pitch propeller. The propeller is enclosed in an annular wing duct with an outer diameter of *11inches*. Vanes located directly beneath the propeller move in order to provide yaw control about the propeller axis. Two sets of control surfaces located further below the propeller move in order to provide pitch and roll moments. Maneuvering is accomplished by tilting the thrust vector with the control surfaces relying primarily on inflow for dynamic pressure during hover. Following satisfactory tethered tests, the vehicle was untethered and allowed to fly simple missions. Fig. 20 shows a plan view of a small *50ft* box maneuver and the GTSpy's tracking. The large deviation on the eastern side of the box is most likely due to a wind gust. Another maneuver performed was the mid-air deployment of the GTSpy. The GTSpy was



Figure 21: Deployment of the GTSpy ducted fan from the GTMax helicopter

mounted on the GTMax helicopter with its engine on and then deployed from a safe altitude. The GTSpy was able to recover from the initial deployment transient and maintain attitude and position within 5 seconds of launch. Fig. 21 shows the GTSpy and GTMax during 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.

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 $2.5, 1.5, 2.5 \text{ rad/s}$ for the (helicopter) roll, pitch and yaw axis respectively. Outer-loop gains were based on $1.5, 1.0, 0.7 \text{ rad/s}$ for the x, y and z helicopter-body-axis respectively. The output-layer learning rates Γ_W was set to unity on all channels and a learning rate of Γ_V was set for all inputs. Reference model

parameters were set to $v_{lim} = 10 \text{ ft/s}$ and $\omega_{lim} = 1.0 \text{ rad/s}$. 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 gradually lowering airspeed until hover. The reverse, transition to forward flight was accomplished by a more aggressive command into forward flight.

The following figures illustrate the response of the aircraft during transitions between hover and forward flight. Fig. 22 shows the vehicle in forward flight at 80 ft/s performing a circular orbit. At $t = 26\text{s}$ a transition to hover is initiated by supplying external trajectory commands that lower the vehicle's speed. Transition is completed at $t = 35\text{s}$ with a low residual speed of approximately 5 ft/s . At $t = 55\text{s}$ a transition back to forward flight at 80 ft/s is initiated and completed at $t = 65\text{s}$. During hover, $t \in [35, 55]$, the control deflections are seen to be significantly higher due to the lower effectiveness at lower speeds. The ailerons are saturated for significant intervals in a particular direction in order to counteract engine torque.

Fig. 23 illustrates the (helicopter) pitch angle during transitions as well as the throttle control deflections. In forward flight, the pitch angle is approximately -75deg and varies in hover due to reduced control effectiveness and the presence of a steady wind. Additionally, Fig. 24 shows the position trajectory during transitions whereas Fig. 25 is a snapshot of the aircraft during the maneuver.

5.5 Implementation of Concurrent Learning Adaptive Controller on a VTOL UAV

Flight-test results of the concurrent learning adaptive law described in Section 3 are presented in this section. The test vehicle is the GTMax rotorcraft UAV. The modification to the adaptive controller described in Section 2.2 include the concurrent learning adaptive law of Equations 27 for a nonlinearly parameterized SHL NN. Data points were selected online based on Equation 30 and were stored in a history stack limited to carrying 20 points. Once the history stack was full, a new data point was added by replacing the oldest data point. A fixed point smoother was used to estimate \dot{x}_j for a recorded data point using both forward and a backward Kalman filter [12, 21]. Typically this induced a selectable time-delay introduced by the time required for the smoother to converge, however, this does not affect the instantaneous tracking error.

5.5.1 Repeated Forward Step Maneuvers

The repeated forward step maneuvers are chosen in order to create a relatively simple situation in which the controller performance can be compared over several similar maneuvers. By using concurrent learning NN improved performance is expected through repeated maneuvers and a faster convergence of weights. Figure 26 shows the body frame states from recorded flight data for a chain of forward step inputs. Figure 27(a) and Figure 27(b) shows the evolution of inner and outer loop errors. These results assert the stability (in the ultimate boundedness sense) of the combined concurrent and online learning approach.

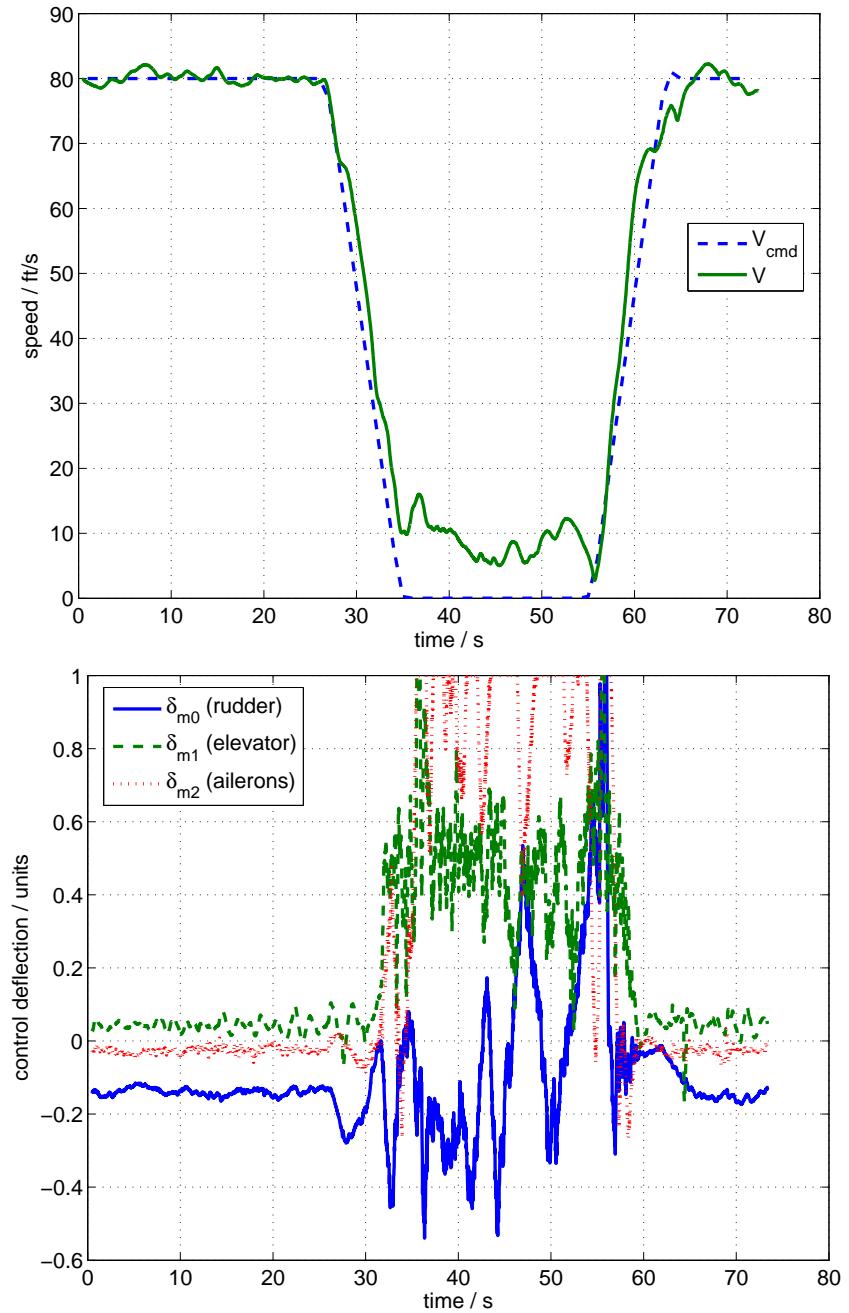


Figure 22: GTEdge speed profile and control deflections during transitions between hover and forward flight

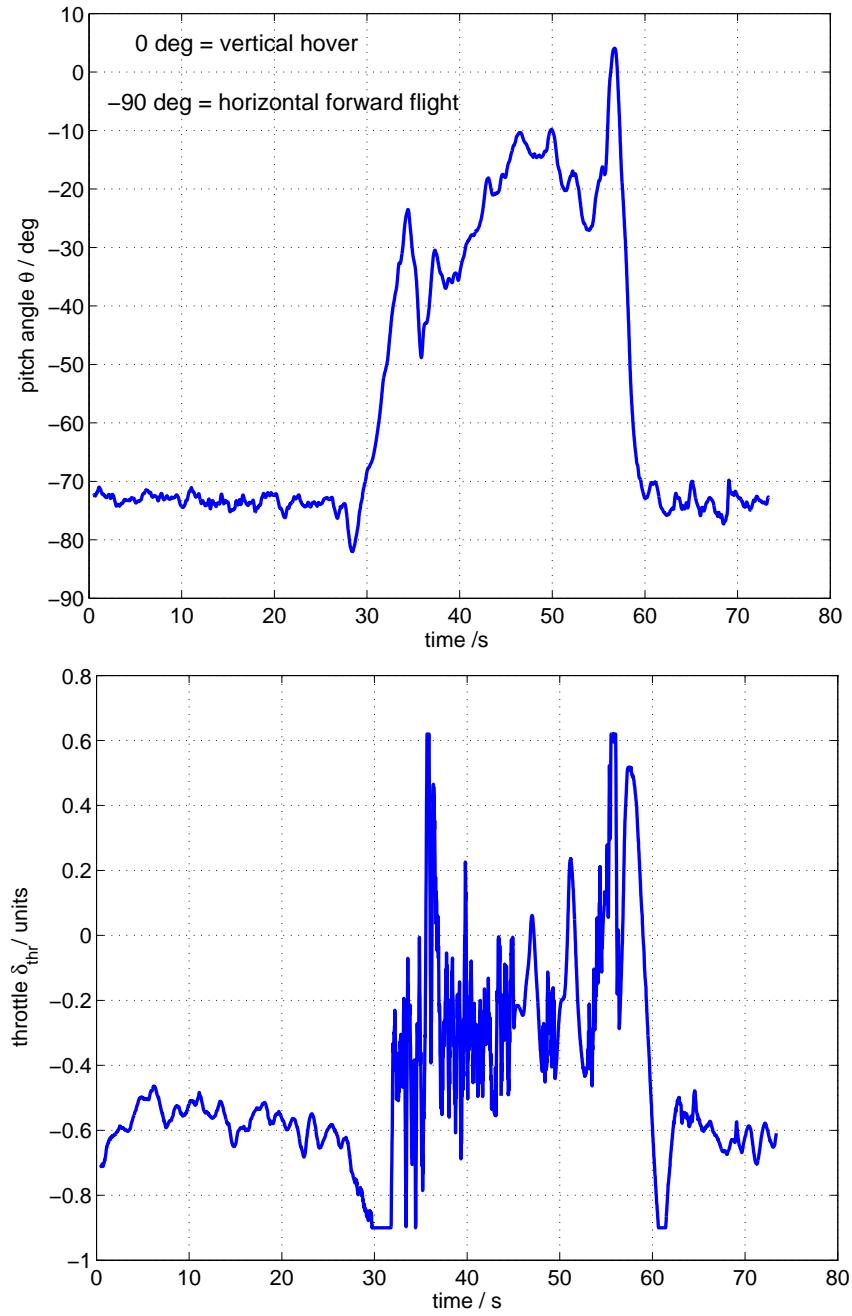


Figure 23: GTEdge pitch angle, throttle profile during transitions between hover and forward flight

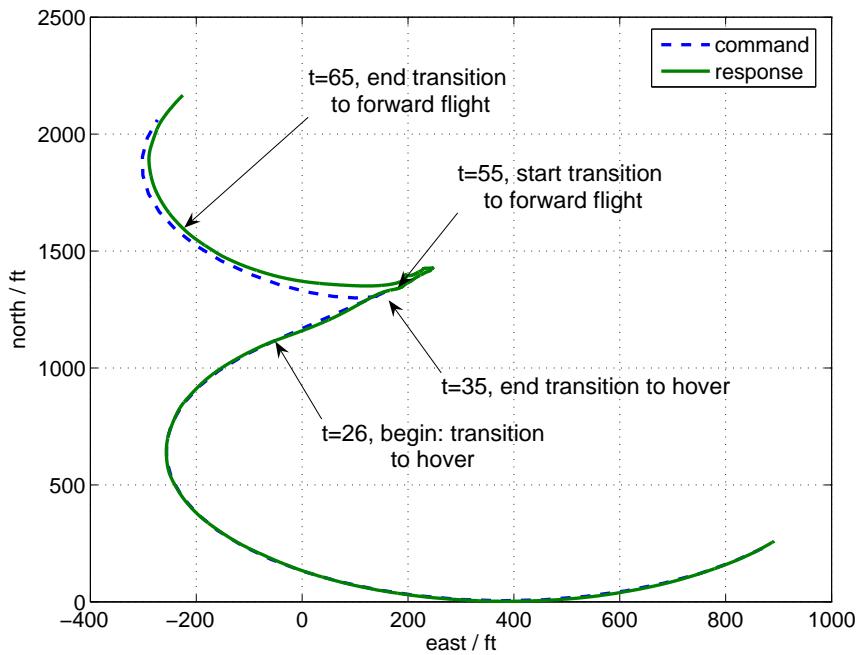


Figure 24: GTEdge trajectory during transitions



Figure 25: GTEdge during a transition

Figure 28(d) and Figure 28(b) show the evolution of NN W and V weights as the rotorcraft performs repeated step maneuvers and the NN is trained using the concurrent learning method of Theorem 3. The NN V weights (28(b)) appear to go to constant values when concurrent learning adaptation is used, this can be contrasted with Figure 28(a) which shows the V weight adaptation for a similar maneuver without concurrent learning. NN W weights for both cases remain bounded, however it is seen that with concurrent learning adaptation the NN W weights seem to separate, this indicates alleviation of the rank-1 condition experienced by the baseline adaptive law relying only on instantaneous data [12]. The flight test results indicate a noticeable improvement in the error profile. In Figure 26 it is seen that the UAV tends not to have a smaller component of body lateral velocity (v) through each successive step. This is also seen in Figure 27(b) where it is noted that the error in v (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.

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 (50ft/s) and acceleration (20ft/s^2) 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 the error signal at each time step is used as a rudimentary metric. Figure 29 shows the recorded inner and outer loop states as the rotorcraft repeatedly tracks an oval trajectory pattern. In this flight, the first two ovals (until $t = 5415\text{ s}$) are tracked with a commanded acceleration of 30ft/sec^2 , while the rest of the ovals are tracked at 20ft/sec^2 . In the following both these parts of the flight test are discussed separately.

5.5.3 Aggressive Trajectory Tracking with Saturation in the Collective Channel

Due to the aggressive acceleration profile of 30ft/s^2 the rotorcraft collective channels were observed to saturate while performing high velocity turns. This leads to an interesting challenge for the adaptive controller equipped with pseudo-control hedging. Figure 30 shows the evolution of the innerloop and outerloop tracking error. It can be clearly seen that the tracking error in the u (body x axis velocity) channel reduces in the second pass through the ellipse indicating long term learning by the combined online and concurrent learning adaptive control system. This result is further characterized by the noticeable reduction in the norm of the tracking error at every time step as shown in Figure 31 .

5.5.4 Aggressive Trajectory Tracking Maneuver

For the results presented in this section, the acceleration profile was reduced to 20ft/sec^2 . At this acceleration profile, no saturation in the collective input was noted. Figure 32 shows

the evolution of tracking error, and Figure 33(a) shows the plot of the norm of the tracking error at each time step.

5.5.5 Aggressive Trajectory Tracking Maneuvers with Only Online Learning NN

The performance of the concurrent learning adaptive controller is compared with the traditional instantaneous update based adaptive controllers for the maneuvers described in Section 5.5.3.

It is instructive to compare Figure 34(b), and Figure 34(d) which show the evolution of the NN weights with only instantaneous learning with Figure 34(a), and Figure 34(c) which show evolution of the NN weights with concurrent learning. Although absolute convergence of weights is not seen, as expected due to Theorem 3 it is interesting to see that when concurrent learning is on, the weights tend to be less oscillatory than when only instantaneous learning is used. Also, with combined online and concurrent learning, the weights do not tend to go to zero as the rotorcraft hovers between two successive tracking maneuver. Figure 33(b) shows the plot of the tracking error norm as a function of time without concurrent learning. Comparing this figure with Figure 33(a) it can be clearly seen that the norm of the error vector is much higher when only online learning is used. This indicates that the combined online and concurrent learning adaptive controller has improved trajectory tracking performance. In summary, the flight test results ascertain an expected improvement in tracking performance. Furthermore, the evolution of the neural network W and V matrix weights were observed to have different characteristics when concurrent learning was employed, including, weight separation, a tendency towards weight convergence in some cases, and different numerical values of the adaptive weights. This difference in neural network weight behavior demonstrates the effect of overcoming the rank-1 condition.

6 Summary

The objective in this chapter has been to provide an affordable control design solution that uses minimal prior knowledge of the vehicle dynamics. This is accomplished by relying on adaptation to cover the flight envelope of the helicopter under nominal conditions. Under mission specific variations in the environment and system dynamics due to payload changes or damage, adaptation allows little or no human intervention after deployment. This approach is also in agreement with the DoD UAS Roadmap which subscribes to the following view on UAV's ...*affordability will be treated as a key performance parameter (KPP) equal to, if not more important than, schedule and technical performance....*

A Adaptive Element

Single hidden layer (SHL) perceptron NNs are universal approximators[24, 48, 38]. Hence, given a sufficient number of hidden layer neurons and appropriate inputs, it is possible to

train the network online to cancel model error. Fig. 35 shows the structure of a generic single hidden layer network whose input-output map may be expressed as

$$\nu_{ad_k} = b_w \theta_{w_k} + \sum_{j=1}^{n_2} w_{jk} \sigma_j(z_j), \quad (51)$$

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

$$\sigma_j(z_j) = \frac{1}{1 + e^{-az_j}}, \quad (52)$$

where, a is the so called *activation potential* and may have a distinct value for each neuron. z_j is the input to the j^{th} hidden layer neuron, and is given by

$$z_j = b_v \theta_{v_j} + \sum_{i=1}^{n_1} v_{ij} x_{in_i}, \quad (53)$$

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

$$V \triangleq \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & v_{n_1,n_2} \end{bmatrix}, \quad (54)$$

$$W \triangleq \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix}, \quad (55)$$

$$Z \triangleq \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix}. \quad (56)$$

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

$$\sigma^T(z) \triangleq [b_w \ \sigma(z_1) \ \cdots \ \sigma(z_{n_2})], \quad (57)$$

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

$$\bar{x}^T = [b_v \ x_{in}^T], \quad (58)$$

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

$$\nu_{ad} = W^T \sigma(V^T \bar{x}). \quad (59)$$

The NN may be used to approximate a nonlinear function, such as $\Delta(\cdot)$. The universal approximation property[24] of NN's ensures that given 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 weights (V^*, W^*) , that brings the output of the NN to within an ϵ -neighborhood of the function approximation error. This ϵ is bounded by $\bar{\epsilon}$ which is defined by

$$\bar{\epsilon} = \sup_{\bar{x} \in \mathcal{D}} \|W^T \sigma(V^T \bar{x}) - \Delta(\bar{x})\|. \quad (60)$$

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

References

- [1] Unmanned aircraft systems roadmap 2011-2036. Technical report, Office of the Secretary of Defense, 2011.
- [2] Aeronautical Design Standard. *Handling Qualities Requirements for Military Rotorcraft, ADS-33E*. United States Army Aviation and Missile Command, Redstone Arsenal, Alabama, March 2000.
- [3] A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, March 1997.
- [4] D. S. Bernstein and A. N. Michel. A chronological bibliography on saturating actuators. *International Journal of Robust and Nonlinear Control*, 5:375–380, 1995.
- [5] S. Boyd and S. Sastry. Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica*, 22(6):629–639, 1986.
- [6] A. J. Calise, N. Hovakimyan, and M. Idan. Adaptive output feedback control of nonlinear systems using neural networks. *Automatica*, 37:1201–1211, August 2001.
- [7] G. Chowdhary. *Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2010.
- [8] G. Chowdhary. *CONCURRENT LEARNING FOR CONVERGENCE IN ADAPTIVE CONTROL WITHOUT PERSISTENCY OF EXCITATION*. PhD thesis, Georgia Institute of Technology, 270 Ferst Drive, Atlanta GA 30332, U.S.A., November 2010.

- [9] G. Chowdhary, R. Chandramohan, J. Hur, E. N. Johnson, and A. J. Calise. Autonomous guidance and control of an airplane under severe damage. In *AIAA@INFOTECH*, St. Louis, MO, March 2011. invited paper.
- [10] G. Chowdhary and E. N. Johnson. Concurrent learning for convergence in adaptive control without persistency of excitation. In *IEEE Conference on Decision and Control*, pages 3674–3679, 2010.
- [11] G. Chowdhary and E. N. Johnson. A singular value maximizing data recording algorithm for concurrent learning. In *American Control Conference*, San Francisco, June 2011.
- [12] G. Chowdhary and E. N. Johnson. Theory and flight test validation of a concurrent learning adaptive controller. *Journal of Guidance Control and Dynamics*, 34(2):592–607, March 2011.
- [13] G. Chowdhary, E. N. Johnson, S. M. Kimbrell, R. Chandramohan, and A. J. Calise. Flight test results of adaptive controllers in the presence of significant aircraft faults. In *AIAA Guidance Navigation and Control Conference*, Toronto, Canada, 2010. *Invited*.
- [14] G. Chowdhary, M. Sobers, C. Pravitra, C. Christmann, A. Wu, H. Hashimoto, C. Ong, R. Kalghatgi, and E. N. Johnson. Integrated guidance navigation and control for a fully autonomous indoor uas. In *AIAA Guidance Navigation and Control Conference*, Portland, OR, August 2011.
- [15] H. Christophersen, R. W. Pickell, J. C. Neidhoefer, A. A. Koller, S. K. Kannan, and E. N. Johnson. A compact guidance, navigation, and control system for unmanned aerial vehicles. *Journal of Aerospace Computing, Information, and Communication*, 3(5):187–213, May 2006.
- [16] J. E. Corban, A. J. Calise, J. V. R. Prasad, G. Heynen, B. Koenig, and J. Hur. Flight evaluation of an adaptive velocity command system for unmanned helicopters. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, August 2003.
- [17] B. Etkin. *Dynamics of Atmospheric Flight*. John Wiley & Sons, New York, 1972.
- [18] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.
- [19] V. Gavrilets, B. Mettler, and E. Feron. Nonlinear model for a small-sized acrobatic helicopter. In *AIAA Guidance, Navigation and Control Conference*, number 2001-4333, Montréal, Quebec, Canada, Aug. 2001.
- [20] V. Gavrilets, B. Mettler, and E. Feron. Control logic for automated aerobatic flight of miniature helicopter. In *AIAA Guidance, Navigation and Control Conference*, number AIAA-2002-4834, Monterey, CA, August 2002.

- [21] A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, 1974.
- [22] S. Haykin. *Neural Networks a Comprehensive Foundation*. Prentice Hall, USA, Upper Saddle River, 2nd edition, 1998.
- [23] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *IEEE Transactions on Neural Networks*, 2(5):359–366, 1989.
- [24] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [25] E. N. Johnson. *Limited Authority Adaptive Flight Control*. PhD thesis, Georgia Institute of Technology, 270 Ferst Drive, Atlanta GA 30332, U.S.A., 2000.
- [26] E. N. Johnson and S. K. Kannan. Adaptive trajectory control for autonomous helicopters. *Journal of Guidance Control and Dynamics*, 28(3):524–538, 2005.
- [27] S. K. Kannan. *Adaptive Control of Systems in Cascade with Saturation*. PhD thesis, Georgia Institute of Technology, 270 Ferst Drive, Atlanta GA 30332, U.S.A., December 2005.
- [28] S. K. Kannan and E. N. Johnson. Nested saturation with guaranteed real poles. In *American Control Conference*, pages 497 – 502, Denver, CO, June 2003.
- [29] S. K. Kannan and E. N. Johnson. Adaptive control of systems in cascade with saturation. In *IEEE Conference on Decision and Control*, Atlanta, GA, December 2010.
- [30] S. K. Kannan and E. N. Johnson. Model reference adaptive control with a constrained linear reference model. In *IEEE Conference on Decision and Control*, Atlanta, GA, December 2010.
- [31] S. K. Kannan, A. A. Koller, and E. N. Johnson. Simulation and development environment for multiple heterogeneous uavs. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Providence, Rhode Island, August 2004.
- [32] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. Journal of Robotics Research*, 30:846–894, June 2011.
- [33] Y. H. Kim and F. Lewis. *High-Level Feedback Control with Neural Networks*, volume 21 of *Robotics and Intelligent Systems*. World Scientific, Singapore, 1998.
- [34] H. Kingravi, G. Chowdhary, P. Vela, and E. Johnson. A reproducing kernel hilbert space approach for the online update of radial bases in neuro-adaptive control. *IEEE Transactions on Neural Networks*, 2012. submitted.

- [35] M. La Civita, W. C. Messner, and T. Kanade. Modeling of small-scale helicopters with integrated first-principles and system-identification techniques. In *Proceedings of the 58th Forum of the American Helicopter Society*, volume 2, pages 2505–2516, Montreal, Canada, June 2002.
- [36] M. La Civita, G. Papageorgiou, W. C. Messner, and T. Kanade. Design and flight testing of a high bandwidth \mathcal{H}_∞ loop shaping controller for a robotic helicopter. In *AIAA Guidance, Navigation and Control Conference*, number AIAA-2002-4846, Monterey, CA, August 2002.
- [37] M. La Civita, G. Papageorgiou, W. C. Messner, and T. Kanade. Design and flight testing of a gain-scheduled \mathcal{H}_∞ loop shaping controller for wide-envelope flight of a robotic helicopter. In *Proceedings of the 2003 American Control Conference*, pages 4195–4200, Denver, CO, June 2003.
- [38] F. L. Lewis. Nonlinear network structures for feedback control (survey paper). *Asian Journal of Control*, 1(4):205–228, 1999.
- [39] Y. Lin and E. D. Sontag. Universal formula for stabilization with bounded controls. *Systems & Control Letters*, 16:393–397, 1991.
- [40] A. M. Lipp and J. V. R. Prasad. Synthesis of a helicopter nonlinear flight controller using approximate model inversion. *Mathematical and Computer Modeling*, 18:89100, 1993.
- [41] B. Mettler. *Identification Modeling and Characteristics of Miniature Rotorcraft*. Kluwer Academic Publishers, 2002.
- [42] 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, U.S.A., 1998.
- [43] G. J. Pappas. Avoiding saturation by trajectory reparameterization. In *IEEE Conference on Decision and Control*, Kobe, 1996.
- [44] J. Park and I. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246–257, 1991.
- [45] D. E. Rumelhart, H. G. E., and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.
- [46] R. T. Rysdyk and A. J. Calise. Robust nonlinear adaptive flight control for consistent handling qualities. *IEEE Transactions on Control Systems Technology*, 13(6):Robust nonlinear ad, November 2005.

- [47] S. N. Singh, W. Yim, and W. R. Wells. Direct adaptive control of wing rock motion of slender delta wings. *Journal of Guidance Control and Dynamics*, 18(1):25–30, Feb. 1995.
- [48] J. T. Spooner, M. Maggiore, R. Ordóñez, and K. M. Passino. *Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques*. Wiley, 2002.
- [49] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, New York, 2003.
- [50] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley and Sons, 111 River Street, Hoboken, NJ 07030-5774, 2003.
- [51] J. Suykens, J. Vandewalle, and B. D. Moor. *Artificial Neural Networks for Modeling and Control of Non-Linear Systems*. Kluwer, Norwell, 1996.
- [52] A. Teel. A nonlinear small gain theorem for the analysis of control systems with saturation. *IEEE Transactions on Automatic Control*, 41(9):1256–1270, 1996.
- [53] A. Teel. Semi-global stabilization of linear systems with position and rate-limited actuators. *Systems and Control Letters*, 30:1–11, 1997.

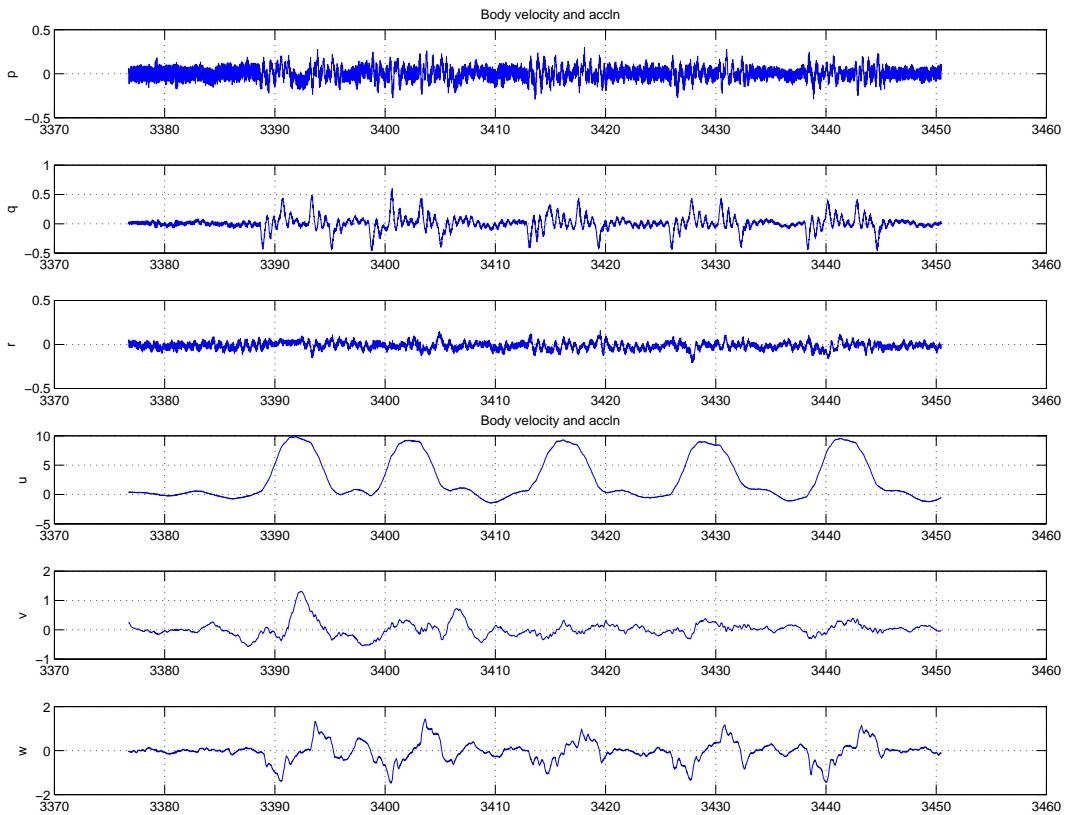
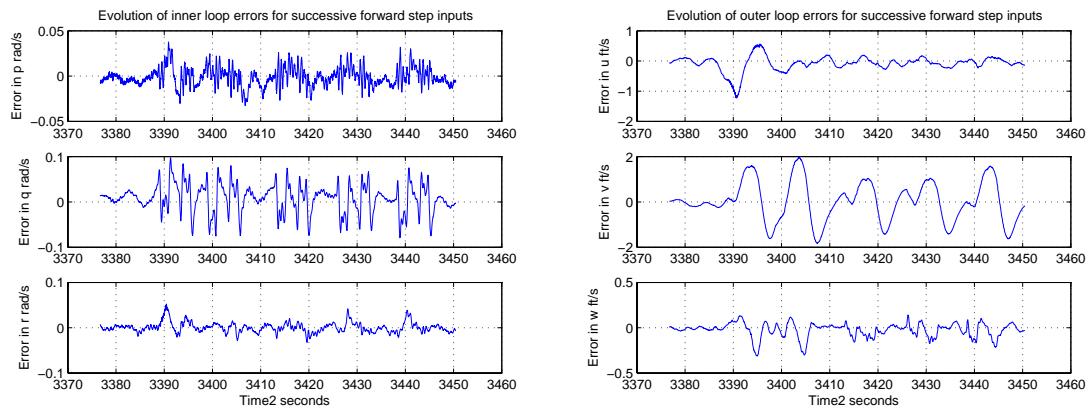
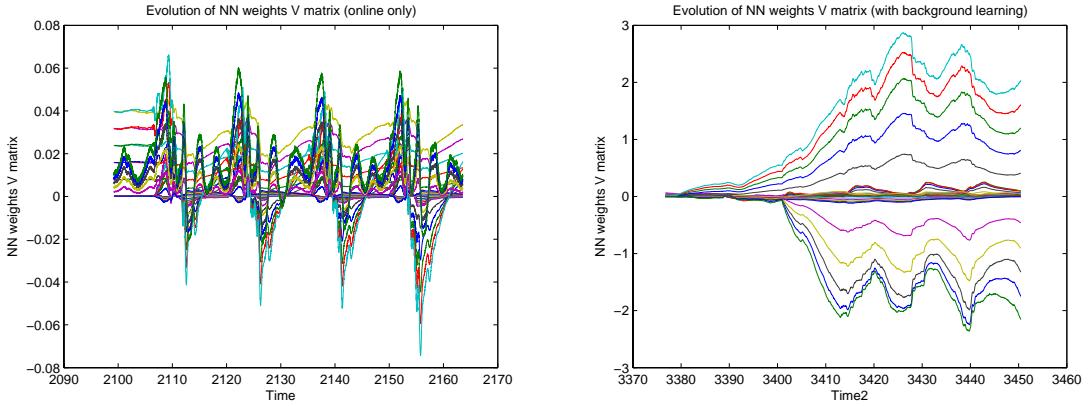


Figure 26: Recorded Body Frame States for Repeated Forward Steps

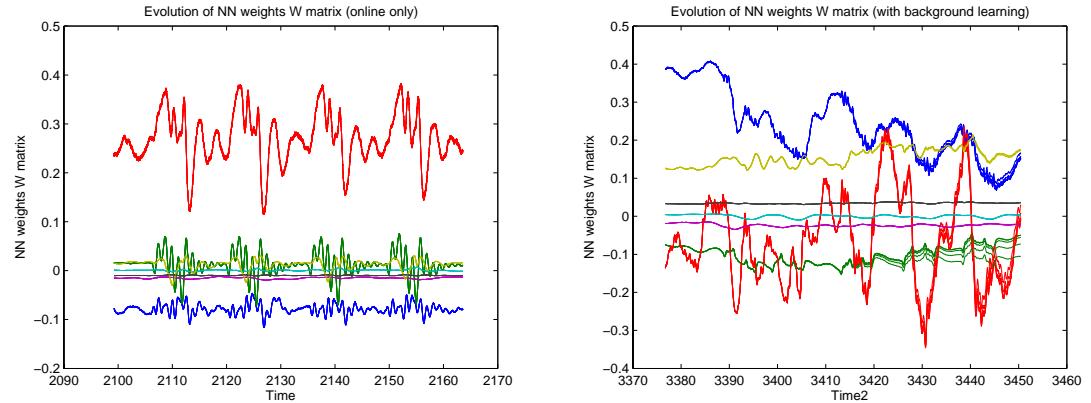


(a) Evolution of inner loop errors with concurrent Adaptation (b) Evolution of outer loop errors with concurrent Adaptation

Figure 27: GTMax Recorded Tracking Errors for Successive Forward Step Inputs with concurrent Learning



(a) Evolution of V matrix weights with Only On-line 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

Figure 28: Comparison of Weight Convergence on GTMax with and without concurrent Learning

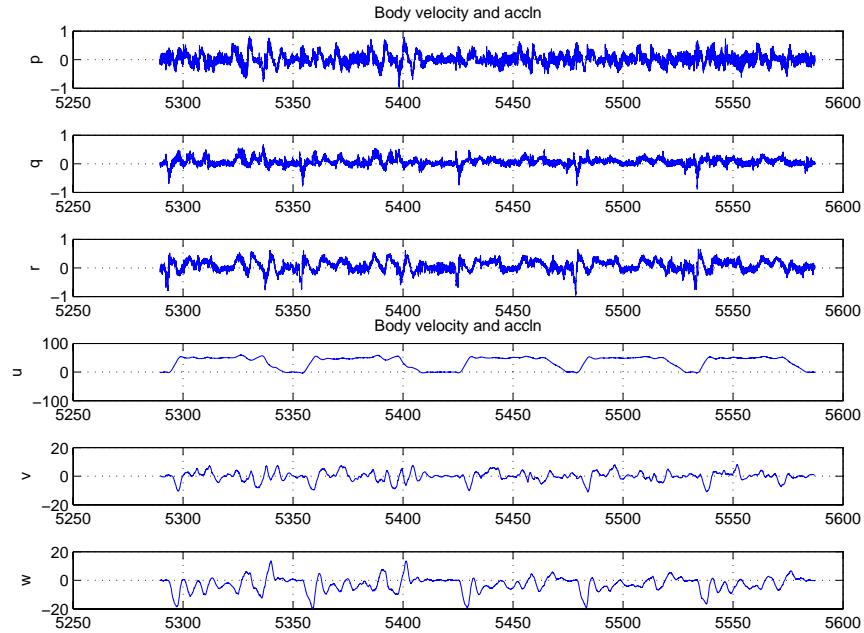
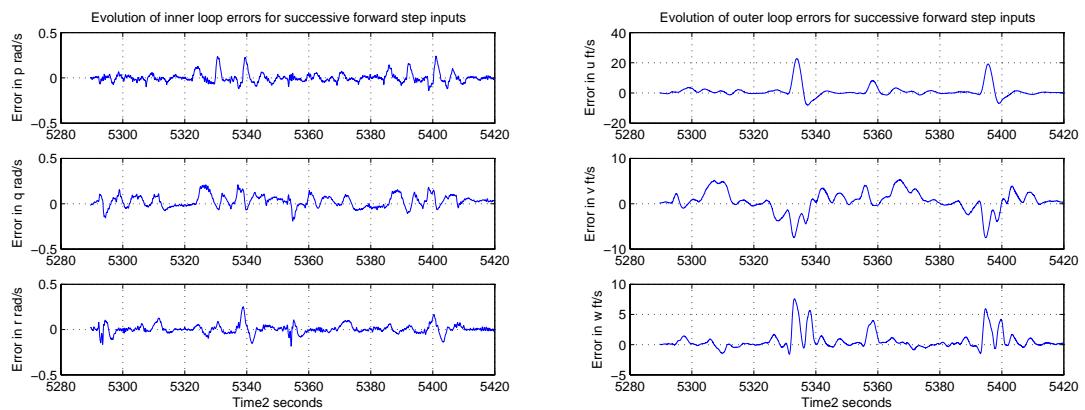


Figure 29: Recorded Body Frame States for Repeated Oval Maneuvers



(a) Evolution of inner loop errors with concurrent Adaptation (b) Evolution of outer loop errors with concurrent Adaptation

Figure 30: GTMax Recorded Tracking Errors for Aggressive Maneuvers with Saturation in Collective Channels with concurrent Learning

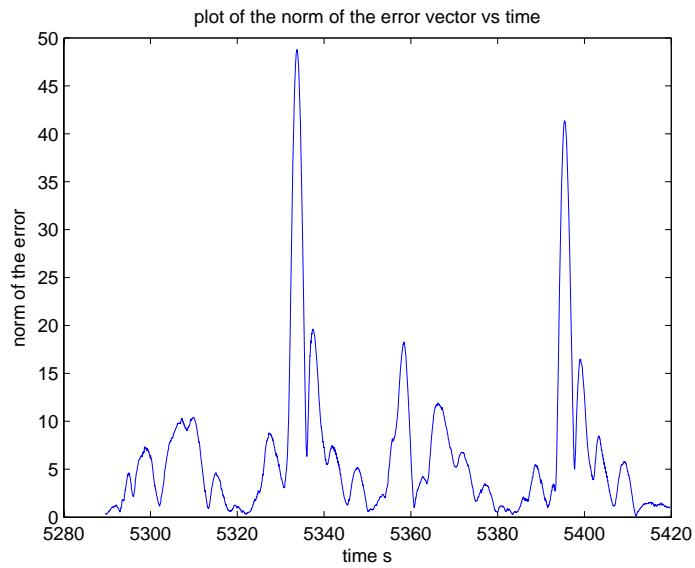
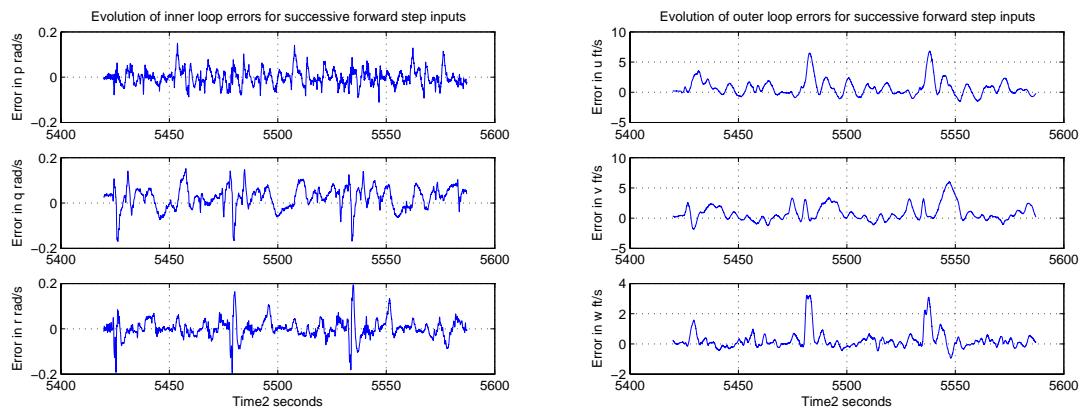
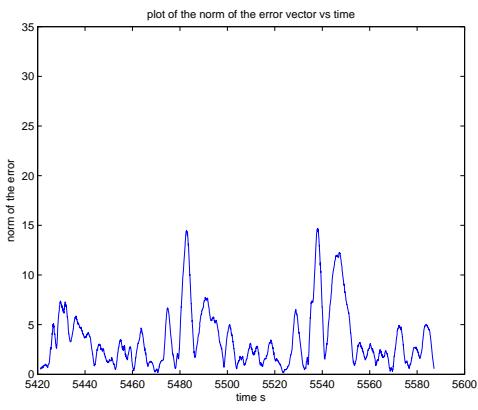


Figure 31: Plot of the norm of the error at each time step for aggressive trajectory tracking with collective saturation

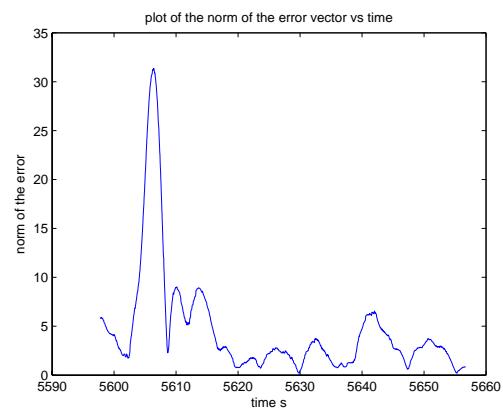


(a) Evolution of inner loop errors with concurrent Adaptation (b) Evolution of outer loop errors with concurrent Adaptation

Figure 32: GTMax Recorded Tracking Errors for Aggressive Maneuvers with concurrent Learning

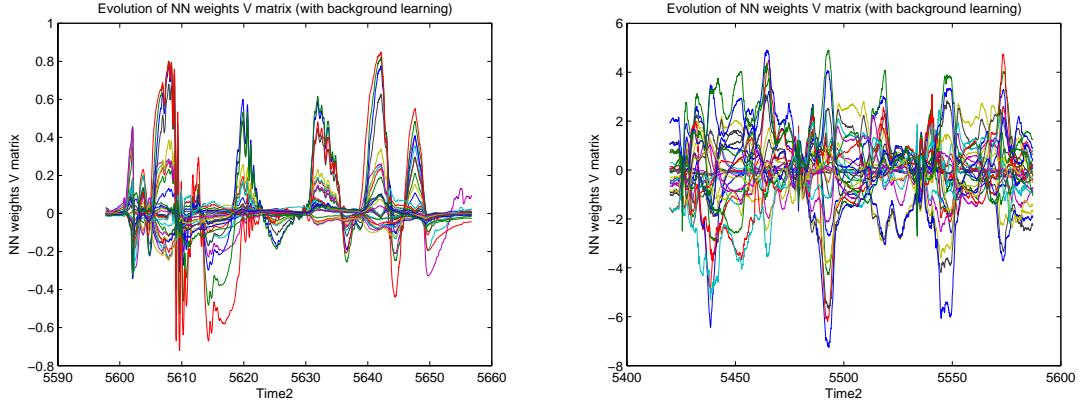


(a) Evolution of the norm of the tracking error
with concurrent Adaptation

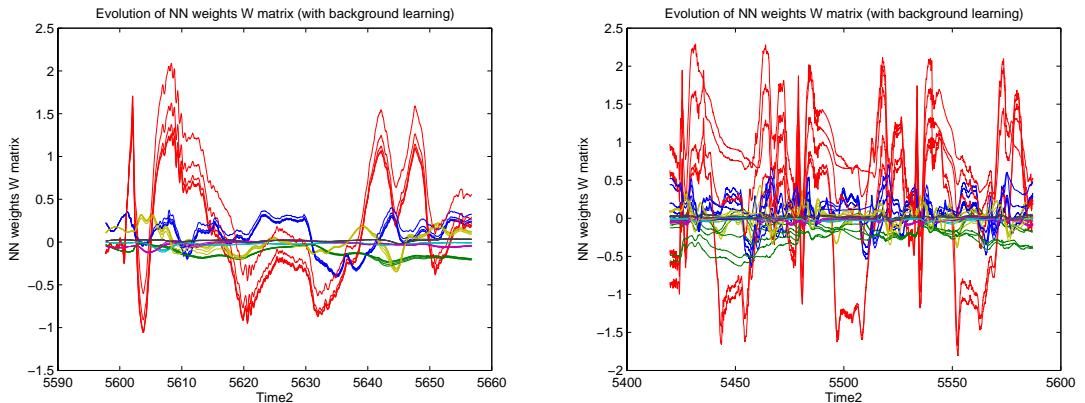


(b) Evolution of the norm of the tracking error
with only online Adaptation

Figure 33: Comparison of norm of GTMax Recorded Tracking Errors for Aggressive Maneuvers



(a) Evolution of V matrix weights with Only On-line 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

Figure 34: Comparison of Weight Convergence as GTMax tracks aggressive trajectory with and without concurrent Learning

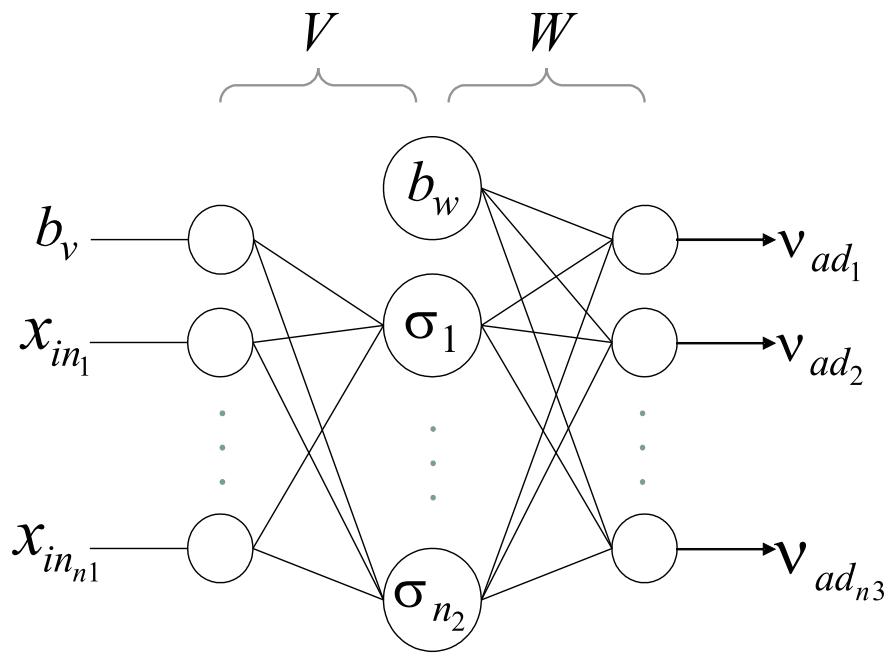


Figure 35: Neural Network with one hidden layer.