AIAA Guidance, Navigation, and Control Conference and Exhibit
5-8 August 2002, Monterey, California

AIAA 2002-4439

Adaptive Flight Control for an Autonomous Unmanned Helicopter

Eric N. Johnson[*] and Suresh K. Kannan[†]
School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332

For autonomous helicopter flight, it is common to separate the flight control problem into an innerloop that controls attitude and an outerloop that controls the trajectory of the helicopter. The outerloop generates attitude commands that orient the main rotor forces appropriately to generate required translational accelerations. Recent work in Neural Network based adaptive flight control may be applied to control a helicopter where the reference commands include position, velocity, attitude and angular rate. The outerloop is used to correct the commanded attitude in order to follow position and velocity commands. This however generally requires a model of the translational dynamics which has some model error. This paper introduces adaptation in the outerloop using Pseudo Control Hedging in a way that prevents adaptation to the innerloop dynamics. Additionally, hedging is used in the innerloop to avoid incorrect adaptation while at control limits. Such an approach along with correct placement of the combined poles of the linearized system mitigates inner/outer loop interaction problems and allows one to increase bandwidth in the outerloop, thus, improving tracking performance further.

Nomenclature

$\oplus$	quaternion addition operator
α	angular acceleration, rad/s^2
a	acceleration, ft/s^2
Δ	function approximation error
δ	control vector / actuator deflections
NN	neural network
ν	pseudo-control vector
ω	angular velocity, rad/s
p	position, ft
PCH	pseudo-control hedging
q	attitude quaternion
V, W	neural network input, output weights
v	velocity, ft/s
x	state vector

Subscripts

ad	adaptive signal
c	commanded
d	derivative
des	desired
f	force
h	hedge
m	moment
ol	outerloop correction
p	proportional
r	robustifying term
rm	reference model

Introduction

Helicopters are versatile machines that can perform aggressive maneuvers. This is evident from the wide range of acrobatic maneuvers executed by expert pilots. Helicopters have a distinct advantage over fixed-wing aircraft especially in an urban environment, where hover capability is required. There is increased interest in the deployment of autonomous helicopters for military and civilian applications. Some of these include, reconnaissance of an urban area and search and rescue missions. Autonomous helicopters must have the capability of planning routes (perhaps optimal in some sense) and executing them. In order to be truly useful, these routes would include high speed dashes, tight turns around buildings, avoiding dynamic obstacles and other aggressive maneuvers. In planning[1] these routes however, the tracking capability of the flight control system is a limiting factor as current control systems are still unable to leverage the full-flight-envelope of small helicopters.

Although stabilization and autonomous flight[2] have been achieved, their performance is modest compared to a human pilot. This is largely due to assumptions made during control design. One such example is the assumption of time-scale separation[3] between the innerloop attitude control and outerloop trajectory control systems. Other problems arise when using a model based robust control approach[4] where the model used is no longer valid during aggressive maneuvers, effectively reducing the bandwidth of the system. Researchers have tackled this by introducing adaptation in the innerloop.[3] This adaptation however only addresses tracking performance of the attitude dynamics.

Another significant problem in controlling a small-

[*] Assistant Professor. AIAA Member, eric.johnson@ae.gatech.edu
[†] Research Assistant. AIAA Student Member suresh_kannan@ae.gatech.edu

helicopter is the presence of the control rotor or stabilizer bar. At higher bandwidths, the innerloop begins to interact with the control rotor dynamics whose states are not normally directly measurable. One approach has been to treat the problem in an output feedback[5] setting that allows high bandwidth control of the attitude dynamics. Even if the innerloop provides high bandwidth attitude tracking performance, an outerloop that can take advantage of this performance is necessary.

This paper is concerned with the development of a combined inner-outer loop adaptive architecture. Neural Network (NN) based direct adaptive control has recently emerged as an enabling technology for practical flight control systems. This technology has been successfully applied[6] to the recent USAF Reconfigurable Control for Tailless Fighter Aircraft (RESTORE) culminating in a successful flight demonstration[7] of the adaptive controller on the X-36. A combined inner-outer loop architecture was also applied for guidance and control of the X-33[8] and evaluated successfully in simulation for various failure cases.

In synthesizing a controller (Figure 1), the conventional conceptual separation between inner and outer loops is made. The inner loop controls the moments acting on the aircraft by changing the *longitudinal stick, lateral stick and pedal* inputs. The outer loop controls the forces acting on the aircraft by varying the magnitude of the rotor thrust using the *collective* input. The thrust vector is oriented in the desired direction by commanding changes to the attitude of the helicopter using the inner-loop.

The attitude and translational dynamics are input-state feedback linearized separately using dynamic inversion and linear controllers designed for their linearized dynamics. The effect of *parametric uncertainty* arising due to approximate inversion is minimized using an adaptive element. The adaptive element could be something as simple as an integrator or a neural network. In this paper, a nonlinearly parameterized neural network will be used to provide on-line adaptation. In introducing an adaptive system however, a new problem arises by way of unwanted adaptation to plant input characteristics such as actuator dynamics.

For example, the innerloop sees actuator limits, rate saturation and associated dynamics. In order to alleviate this problem, Pseudo-Control-Hedging[8, 9] (PCH), is used to modify the innerloop reference model dynamics in a way that allows continued adaption in the presence of these system characteristics. This same technique, PCH, is used to "hedge" the outerloop to prevent adaptation to innerloop dynamics.

In this paper, the combined inner-outer loop architecture is first described followed by a description of the Neural Network and selection of linear compensator gains. The controller is then applied to the trajectory and attitude control of an unmanned helicopter. Practical discussions on the choice of parameters and reference model dynamics are provided. Finally, simulation and flight test results are presented.

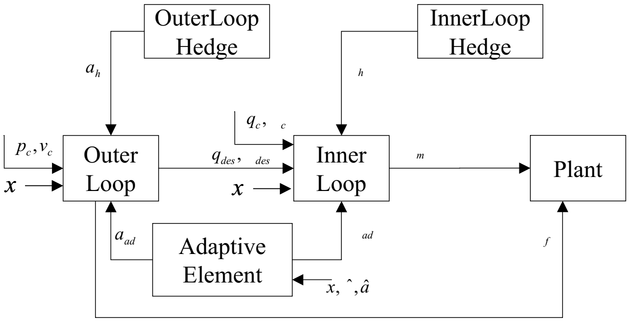

Fig. 1 Overall Inner and Outerloop with Adaptation and Hedging

Controller Development

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

$$\begin{aligned} \dot{p} &= v & (1)\\ \dot{v} &= a(p, v, q, \omega, \delta_f, \delta_m) & (2)\\ \dot{q} &= \dot{q}(q, \omega) & (3)\\ \dot{\omega} &= \alpha(p, v, q, \omega, \delta_f, \delta_m) & (4) \end{aligned}$$

where, $q \in \mathcal{R}^4$ represents the attitude quaternion, $\omega \in \mathcal{R}^3$ is the angular velocity, $p \in \mathcal{R}^3$ is the position vector and $v \in \mathcal{R}^3$ is the velocity of the vehicle. Eq. (3) represents the quaternion propagation equations.[10] Eq. (4) represents the attitude dynamics and Eq. (2) represents translational dynamics. The state vector x may now be defined as $x \triangleq [p, v, q, \omega]$.

The control vectors are denoted by δ_f and δ_m, where δ_f denotes the main force generating controls and δ_m denotes the main moment generating controls. The consolidated control vector δ may be defined as $\delta = [\delta_f, \delta_m]$. This classification of moment and force generating controls is an artefact of the design strategy. In general, both control inputs δ_f and δ_m may each produce forces *and* moments.

Approximate feedback linearization of the system (1) is achieved by introducing the following transformation

$$\nu = \begin{bmatrix} a_{des} \\ \alpha_{des} \end{bmatrix} = \begin{bmatrix} \hat{a}(p, v, q, \omega, q_{des}, \delta_m, \delta_{f_{des}}) \\ \hat{\alpha}(p, v, q, \omega, \delta_{m_{des}}, \delta_f) \end{bmatrix} \qquad (5)$$

where, ν is commonly referred to as the pseudo-control. $\hat{a}$, $\hat{\alpha}$ represent the best available approximation of $a(\cdot)$ and $\alpha(\cdot)$. Additionally, $\delta_{f_{des}}, \delta_{m_{des}}, q_{des}$ are the control inputs and attitude that is expected to achieve ν. This form assumes that translational dynamics is coupled strongly with attitude. Hence, q_{des} will be used to make corrections to the nominal commanded attitude q_c in order to achieve the desired translational accelerations. Choosing $\hat{a}$ and $\hat{\alpha}$ such that they are invertible, the desired control and attitude may be written as

$$\begin{bmatrix}\delta_{f_{des}} \\ q_{des}\end{bmatrix} = \hat{a}^{-1}(p, v, q, \omega, a_{des}, \delta_m)$$
$$\delta_{m_{des}} = \hat{\alpha}^{-1}(p, v, q, \omega, \delta_f, \alpha_{des}) \tag{6}$$

In computing $\delta_{f_{des}}$, $\delta_{m_{des}}$and q_{des} using the approximate inversion $\hat{a}^{-1}, \hat{\alpha}^{-1}$, any dynamics in the actuators are temporarily ignored. If we define the model error $\bar{\Delta}$ as

$$\bar{\Delta}(x, \delta, q_{des}, \delta_{des}) = \begin{bmatrix}\bar{\Delta}_a \\ \bar{\Delta}_\alpha\end{bmatrix} = \begin{bmatrix}a - \hat{a} \\ \alpha - \hat{\alpha}\end{bmatrix} \tag{7}$$

then, the dynamics in Eq. (1) may be written as

$$\dot{v} = a_{des} + \bar{\Delta}_a$$
$$\dot{\omega} = \alpha_{des} + \bar{\Delta}_\alpha \tag{8}$$

The pseudo-controls a_{des} and α_{des} may now be designed to satisfy closed-loop performance and stability characteristics. Choosing,

$$a_{des} = a_{crm} + a_{pd} - \hat{a}_{ad}$$
$$\alpha_{des} = \alpha_{crm} + \alpha_{pd} - \hat{\alpha}_{ad} \tag{9}$$

where a_{crm} and α_{crm} are the *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, $\hat{a}_{ad}$ and $\hat{\alpha}_{ad}$ are the outputs of an adaptive element.

Reference Model and PCH

Normally, the reference model dynamics may be designed as

$$\dot{v}_{rm} = a_{crm}(p_{rm}, v_{rm}, p_c, v_c)$$
$$\dot{\omega}_{rm} = \alpha_{crm}(q_{rm}, \omega_{rm}, q_c \oplus q_{des}, \omega_c) \tag{10}$$

where p_{rm} and v_{rm} denote the outerloop reference model states whereas q_{rm}, ω_{rm}, denote the innerloop reference model states. $x_c = [p_c, v_c, q_c, \omega_c]$ is the external command signal. Note that the attitude desired by the outerloop is now added to the commands for the innerloop controller. Here, $q_c \oplus q_{des}$ denotes quaternion addition.[10]

This form however, does not account for actuator dynamics, nor does it account for the innerloop dynamics. If the actuators are saturated, the reference models will continue to demand tracking as though full authority were still available. A similar problem exists when the outerloop demands attitude corrections from the innerloop without consideration of its dynamics. If left uncorrected, it may cause incorrect adaptation in the adaptive element.

Pseudo-Control Hedging is used to prevent the adaptive element from trying to adapt to selected system input characteristics. One way to describe the PCH method is: *move the reference model in the opposite direction (hedge) by an estimate of the amount the plant did not move due to system characteristics the control designer does not want the adaptive element to see.*[9] This will prevent the characteristic from appearing in the model tracking error dynamics to be developed in the sequel.

The reference model dynamics may be redesigned to include hedging as follows

$$\begin{aligned}\dot{v}_{rm} &= a_{crm} - a_h \\ \dot{\omega}_{rm} &= \alpha_{crm} - \alpha_h \qquad (11)\\ a_{crm} &= a_{crm}(p_{rm}, v_{rm}, p_c, v_c) \\ \alpha_{crm} &= \alpha_{crm}(q_{rm}, \omega_{rm}, q_c \oplus q_{des}, \omega_c) \qquad (12)\end{aligned}$$

where a_h and α_h are the difference between commanded pseudo-control and achieved pseudo-control.

$$\begin{aligned}a_h &= \hat{a}(x, q_{des}, \delta_{f_{des}}, \delta_m) - \hat{a}(x, \delta_f, \delta_m) \\ &= a_{des} - \hat{a}(x, \delta_f, \delta_m) \\ \alpha_h &= \hat{\alpha}(x, \delta_f, \delta_{m_{des}}) - \hat{\alpha}(x, \delta_f, \delta_m) \\ &= \alpha_{des} - \hat{\alpha}(x, \delta_f, \delta_m)\end{aligned} \tag{13}$$

Note that the hedge signals a_h, α_h, affect the reference model output a_{crm}, α_{crm}, only through changes in the reference model states. In computing the hedge signal, knowledge of q, ω, δ_f and δ_m is assumed. Although it is reasonable to assume knowledge of the attitude and angular rate which may be measured, actuator positions are not normally available for feedback. Results that use estimates of actuator positions may be found in existing work.[9, 11]

Tracking error dynamics

One may define the tracking error vector, e, as

$$e \triangleq \begin{bmatrix}p_{rm} - p \\ v_{rm} - v \\ \tilde{Q}(q_{rm}, q) \\ \omega_{rm} - \omega\end{bmatrix} \tag{14}$$

where, $\tilde{Q} : \mathcal{R}^4 \times \mathcal{R}^4 \mapsto \mathcal{R}^3$, is a function[9] which, given two quaternions results in an error angle vector with three components. If error angles angles are kept small, an expression for $\tilde{Q}$ is given by

$$\tilde{Q}(p, q) = 2sgn(q_1p_1 + q_2p_2 + q_3p_3 + q_4p_4)\times \begin{bmatrix}-q_1p_2 + q_2p_1 + q_3p_4 - q_4p_3 \\ -q_1p_3 - q_2p_4 + q_3p_1 + q_4p_2 \\ -q_1p_4 + q_2p_3 - q_3p_2 + q_4p_1\end{bmatrix} \tag{15}$$

The output of the PD compensators may be written as

$$\begin{bmatrix}a_{pd} \\ \alpha_{pd}\end{bmatrix} = \begin{bmatrix}R_p & R_d & 0 & 0 \\ 0 & 0 & K_p & K_d\end{bmatrix} e \tag{16}$$

where, $R_p, R_d \in \mathcal{R}^{3\times3}$, $K_p, K_d \in \mathcal{R}^{3\times3}$ are diagonal and positive definite matrices. The tracking error dynamics may be found by directly differentiating

Eq. (14).

$$\dot{e} = \begin{bmatrix} v_{rm} - v \\ \dot{v}_{rm} - \dot{v} \\ \omega_{rm} - \omega \\ \dot{\omega}_{rm} - \dot{\omega} \end{bmatrix} \tag{17}$$

Considering $\dot{e}_2$,

$$\begin{aligned} \dot{e}_2 &= \dot{v}_{rm} - \dot{v} \\ &= a_{crm} - a_h - a(x,\delta) \\ &= a_{crm} - a_{des} + \hat{a}(x,\delta) - a(x,\delta) \\ &= a_{crm} - a_{pd} - a_{crm} + \hat{a}_{ad} + \hat{a}(x,\delta) - a(x,\delta) \\ &= -a_{pd} - (a(x,\delta) - \hat{a}(x,\delta) - \hat{a}_{ad}) \\ &= -a_{pd} - (\Delta_a(x,\delta) - \hat{a}_{ad}) \end{aligned} \tag{18}$$

$\dot{e}_4$ may be found similarly. Then,

$$\Delta(x,\delta) \triangleq \begin{bmatrix} \Delta_a \\ \Delta_\alpha \end{bmatrix} = \begin{bmatrix} a(x,\delta) - \hat{a}(x,\delta) \\ \alpha(x,\delta) - \hat{\alpha}(x,\delta) \end{bmatrix} \tag{19}$$

Hence, the overall tracking error dynamics may now be expressed as

$$\dot{e} = Ae + B\left[\hat{\nu}_{ad} - \Delta(x,\delta)\right] \tag{20}$$

where,

$$A = \begin{bmatrix} 0 & I & 0 & 0 \\ -R_p & -R_d & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -K_p & -K_d \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \tag{21}$$

Now, $\hat{\nu}_{ad}$ remains to be designed in order to cancel the effect of Δ. In order to cancel Δ using an adaptive element, it should have the functional form $\hat{\nu}_{ad}(x,\delta)$.

Remark 1. *Note that commands, $\delta_{m_{des}}, \delta_{f_{des}}, q_{des}$, do not appear in the tracking error dynamics. Pseudo-control hedging allows adaptation to continue when the actual control signal has been replaced by any arbitrary signal. 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.*

Remark 2. *A departure from previous MRAC work when PCH is included is the modification of the (stable) reference model dynamics by a_h and α_h. Considering just the attitude dynamics reference model, the un-hedged reference model dynamics α_{crm}, represents the desired response of the system to command tracking errors. Taking,*

$$\begin{aligned} \dot{\omega}_{rm} &= \alpha^*_{ad} + \hat{\alpha}(p, v, q_{rm}, \omega_{rm}, \delta_f, \delta_m) \\ \delta_{m_{des}} &= \hat{\alpha}^{-1}(p, v, q_{rm}, \omega_{rm}, \delta_f, \alpha_{des}) \\ &= \hat{\alpha}^{-1}(p, v, q_{rm}, \omega_{rm}, \delta_f, \alpha_{crm} - \alpha^*_{ad}) \end{aligned} \tag{22}$$

where, α^*_{ad} is an ideal post adaptation output of the adaptive element and tracking error is zero, i.e., $q = q_{rm}$ and $\omega = \omega_{rm}$. If adaptation is capable of exactly cancelling for model error. Then Eq. (22) may be written as

$$\begin{aligned} \dot{\omega}_{rm} &= \alpha(p, v, q_{rm}, \omega_{rm}, \delta_f, \delta_m) \\ \delta_{m_{des}} &= \alpha^{-1}(p, v, q_{rm}, \omega_{rm}, \delta_f, \alpha_{crm}) \end{aligned} \tag{23}$$

Eq. (23) now becomes the non-adaptive design synthesis problem for selecting α_{crm}. In this paper, the reference model dynamics are chosen as

$$\alpha_{crm} = K_p\tilde{Q}(q_c, q_{rm}) + K_d(\omega_c - \omega_{rm}) \tag{24}$$

where K_p and K_d are the same as the PD compensator gains and will achieve the desirable responses for permissible plant and actuator dynamics. If actuators and adaptation are perfect, the closed loop attitude system will exhibit linear dynamics. A similar argument may be made for the translational dynamics reference model.

Adaptive Element

Single hidden layer perceptron Neural Networks (NNs) are universal approximators.[12] Hence, given a sufficient number of hidden layer neurons and appropriate inputs, it should be possible to train the network online to cancel model error. Figure 2 shows the structure of a single hidden layer network whose input-output map may be expressed as

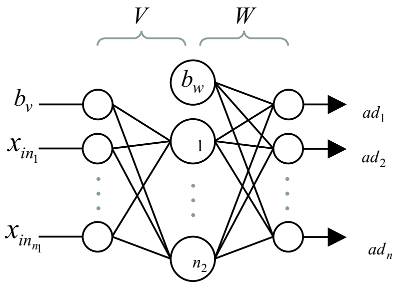

Fig. 2 Neural Network with one hidden layer

$$\nu_{ad_k} = b_w\theta_{wk} + \sum_{j=1}^{n_2} w_{jk}\sigma_j(z_j) \tag{25}$$

where, $k = 1, ..., n_3$ and,

$$\sigma_j(z_j) = \sigma(b_v\theta_{vj} + \sum_{i=1}^{n_1} v_{ij}x_{in_i}) \tag{26}$$

Here, n_1, n_2 and n_3 are the number of inputs, hidden layer neurons and outputs respectively. $x_{in_i}, i = 1..n_1$, denote the inputs to the NN. The scalar σ_j is a sigmoidal activation activation function,

$$\sigma(z) = \frac{1}{1 + e^{-az}} \tag{27}$$

where, a is the so called *activation potential.* For convenience, define the following weight matrices

$$V = \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & v_{n_1,n_2} \end{bmatrix} \tag{28}$$

$$W = \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix} \tag{29}$$

$$Z = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \tag{30}$$

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

$$\sigma^T(z) = \begin{bmatrix} b_w & \sigma(z_1) & \cdots & \sigma(z_{n_2}) \end{bmatrix} \tag{31}$$

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

$$\bar{x}^T = \begin{bmatrix} b_v & x_{in}^T \end{bmatrix} \tag{32}$$

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

$$\nu_{ad} = W^T\sigma(V^T\bar{x}) \tag{33}$$

The NN may be used to approximate a nonlinear function, such as $\Delta(.)$. The universal approximation property[12] 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-neighbourhood of the function approximation error. This ϵ is bounded by $\bar{\epsilon}$ which is defined by

$$\bar{\epsilon} = \sup_{\bar{x}\in\mathcal{D}} \left\| W^T\sigma(V^T\bar{x}) - \Delta(\bar{x}) \right\| \tag{34}$$

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.

Associated with the tracking error dynamics given in Eq. (20), is the Lyapunov function.

$$A^TP + PA + Q = 0 \tag{35}$$

Choosing[9]

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \frac{1}{\frac{1}{4}n_2 + b_w^2} \tag{36}$$

where,

$$Q_1 = \begin{bmatrix} R_dR_p^2 & 0 \\ 0 & R_dR_p \end{bmatrix} > 0 \tag{37}$$

$$Q_2 = \begin{bmatrix} K_dK_p^2 & 0 \\ 0 & K_dK_p \end{bmatrix} > 0 \tag{38}$$

Making use of the property that $R_p, R_d, K_p, K_d > 0$ and diagonal, results in a positive definite solution for P. Hence,

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \frac{1}{\frac{1}{4}n_2 + b_w^2} \tag{39}$$

where,

$$P_1 = \begin{bmatrix} R_p^2 + \frac{1}{2}R_pR_d^2 & \frac{1}{2}R_pR_d \\ \frac{1}{2}R_pR_d & R_p \end{bmatrix} > 0 \tag{40}$$

$$P_2 = \begin{bmatrix} K_p^2 + \frac{1}{2}K_pK_d^2 & \frac{1}{2}K_pK_d \\ \frac{1}{2}K_pK_d & K_p \end{bmatrix} > 0 \tag{41}$$

Assumption 1. *The norm of the ideal weights (V^*, W^*) is bounded by a known positive value.*

$$0 < \|Z^*\|_F \leq \bar{Z} \tag{42}$$

where, $\|\cdot\|_F$ denotes the Frobenius norm.

Assumption 2. *The external command x_c is bounded.*

$$\|x_c\| \leq \bar{x}_c \tag{43}$$

Assumption 3. *The states of the reference model, selected in context of Remark 2 remain bounded for permissable plant and actuator dynamics.*

Assumption 4. *Note that, Δ depends on ν_{ad} through ν, whereas ν_{ad} has to be designed to cancel Δ. Hence the existence and uniqueness of a fixed-point-solution for $\nu_{ad} = \Delta(x, \nu_{ad})$ must be assumed. A sufficient condition is to ascertain that the map $\nu_{ad} \mapsto \Delta(x, \nu_{ad})$ is a contraction over the entire input domain of interest, or $\|\partial\Delta/\partial\nu_{ad}\| < 1$. This condition is equivalent to the following condition on $\hat{f}$, where $\hat{f}, f, \Delta, \nu_{ad}, \nu, \delta$ represent the elements corresponding to the angular or translational dynamics.*

$$\begin{aligned} \left\| \frac{\partial\Delta}{\partial\nu_{ad}} \right\| &= \left\| \frac{\partial\Delta}{\partial\delta} \frac{\partial\hat{f}^{-1}}{\partial\nu} \frac{\partial\nu}{\partial\nu_{ad}} \right\| \\ &= \left\| \left(\frac{\partial f}{\partial\delta} - \frac{\partial\hat{f}}{\partial\delta} \right) \frac{\partial\hat{f}^{-1}}{\partial\nu} \right\| \\ &= \left\| \frac{\partial f}{\partial\delta} \frac{\partial\hat{f}^{-1}}{\partial\nu} - I \right\| < 1 \end{aligned} \tag{44}$$

For a SISO system, condition (44) is equivalent to

$$sgn(\partial f/\partial\delta) = sgn(\partial\hat{f}/\partial\delta) \tag{45}$$

and,

$$|\partial\hat{f}/\partial\delta| > |\partial f/\partial\delta|/2 \tag{46}$$

Condition (45) states that unmodeled control reversal is not permissable and (46) places a lower bound on the estimate of control effectiveness.

Theorem 1. *Consider the system given by (1) together with the inverse law (6) and the above assumptions, where,*

$$r = (e^T PB)^T \tag{47}$$
$$\hat{\nu}_{ad} = \nu_{ad} + \nu_r \tag{48}$$
$$\nu_{ad} = W^T\sigma(V^T\bar{x}) \tag{49}$$
$$\nu_r = -K_r(\|Z\|_F + \bar{Z})r \tag{50}$$

with diagonal $K_r > 0 \in \mathcal{R}^{6\times 6}$*, and where* W, V *satisfy the adaptation laws*

$$\dot{W} = -\left[(\sigma - \sigma' V^T\bar{x})r^T + \kappa\|e\|W\right]\Gamma_W \tag{51}$$
$$\dot{V} = -\Gamma_V\left[\bar{x}(r^T W^T\sigma') + \kappa\|e\|V\right] \tag{52}$$

with, $\Gamma_W, \Gamma_V > 0$ *and* $\kappa > 0$*, guarantees that reference model tracking error (e) and NN weights (W, V) are uniformly ultimately bounded.*

Proof. See appendix. □

Corollary 1. *All plant states* $[p, v, q, \omega]$ *are uniformly ultimately bounded.*

Proof. If the ultimate boundedness e, W, V from Theorem 1 is taken together with Assumption 3, the uniform ultimate boundedness of the plant states is immediate following the definition of the reference model tracking error in Eq. (14). □

Application to an Autonomous Helicopter

Consider the application of the combined inner and outerloop adaptive architecture to the trajectory control of a helicopter. The dynamics[13–15] of the helicopter may be modelled in the same form as Eq. (1). Most small helicopters include the control rotor *(also known as a Hillier stabilizer bar)* which has two main functions. a) provide lagged rate feedback and b) aiding servos change main rotor blade pitch by providing additional aerodynamic moment. The nonlinear model used for simulation included these control rotor dynamics. Additionally, blade flapping and other aspects such as gear and engine dynamics were also modelled.

For a helicopter, the main force effector is the rotor thrust which is controlled by changing main rotor collective δ_{coll}. Hence $\delta_f \in \mathcal{R}^1 = \delta_{coll}$. There are three primary moment control surfaces, the longitudinal cyclic δ_{lon}, lateral cyclic δ_{lat} and tail rotor pitch, also called the pedal input δ_{ped}. Hence, $\delta_m \in \mathcal{R}^3 = \begin{bmatrix}\delta_{lon} & \delta_{lat} & \delta_{ped}\end{bmatrix}^T$.

Reference Model

A reasonable choice for the reference model dynamics is given by

$$\begin{aligned} a_{crm} &= R_p(p_c - p_{rm}) + R_d(v_c - v_{rm}) \\ \dot{v}_{rm} &= a_{crm} - a_h \end{aligned} \tag{53}$$
$$\begin{aligned} \alpha_{crm} &= K_p(\tilde{Q}(q_c \oplus q_{des}, q_{rm})) + K_d(\omega_c - \omega_{rm}) \\ \dot{\omega}_{rm} &= \alpha_{crm} - \alpha_h \end{aligned} \tag{54}$$

where, R_p, R_d, K_p, K_d can be the same gains used for the PD compensator. 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 modifying a_{crm} and α_{crm} to the following form

$$\begin{aligned} a_{crm} &= R_d[v_c - v_{rm} + sat(R_d^{-1}R_p(p_c - p_{rm}), v_{lim})] \\ \alpha_{crm} &= K_d[\omega_c - \omega_{rm} + sat(K_d^{-1}K_p\tilde{Q}, \omega_{lim})] \end{aligned} \tag{55}$$

where the functional dependence of $\tilde{Q}$ has been dropped for clarity and is the same as in Eq. (54). The function *sat* is the saturation function and v_{lim}, ω_{lim} are the translational and angular rate limits respectively.

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.

$$\alpha_{des} = A_1\begin{bmatrix}p\\q\\r\end{bmatrix} + A_2\begin{bmatrix}u\\v\\w\end{bmatrix} + B\left(\underbrace{\begin{bmatrix}\delta_{lat}\\\delta_{lon}\\\delta_{ped}\end{bmatrix}}_{des} - \underbrace{\begin{bmatrix}\delta_{lat}\\\delta_{lon}\\\delta_{ped}\end{bmatrix}}_{trim}\right)$$

or,

$$\alpha_{des} = A_1\omega_B + A_2 v_B + B(\delta_{m_{des}} - \delta_{m_{trim}}) \tag{56}$$

where, A_1 and 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. v_B, is the body velocity vector with respect to the earth expressed in the body frame. $\delta_{m_{trim}}$ is the trim control vector that is consistent with the linear model.

Choosing the control matrix B such that it is invertible, the moment controls may be evaluated as

$$\delta_{m_{des}} = B^{-1}(\alpha_{des} - A_1\omega_B - A_2 v_B) + \delta_{m_{trim}} \tag{57}$$

The translational dynamics were modelled as a point mass with a thrust vector that may be oriented in a given direction as illustrated in Figure 3.

$$a_{des} = \begin{bmatrix}0\\0\\Z_{\delta_{coll}}\end{bmatrix}(\delta_{coll_{des}} - \delta_{coll_{trim}}) + L_{bv}g \tag{58}$$

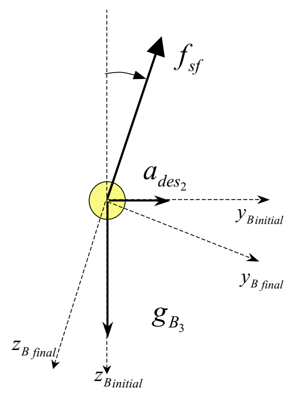

Fig. 3 Point mass model for outerloop inversion

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

$$f_{sf} = (a_{des} - L_{bv} g)_3 \tag{59}$$

The required collective input may be evaluated as

$$\delta_{f_{des}} = \delta_{coll_{des}} = \frac{f_{sf}}{Z_{\delta_{coll}}} + \delta_{coll_{trim}} \tag{60}$$

The attitude correction required in order to orient the thrust vector to attain the desired translational accelerations are given by the following small angle corrections

$$\Delta\phi = -\frac{a_{des_2}}{f_{sf}}, \quad \Delta\theta = \frac{a_{des_1}}{f_{sf}}, \quad \Delta\psi = 0 \tag{61}$$

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

$$q_{des} = q(\Delta\phi, \Delta\theta, \Delta\psi) \tag{62}$$

where, $q(.)$ is a function[10] that expresses an euler angles based orientation as a quaternion. The overall detailed controller architecture is shown in Figure 4.

Remark 3. *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 (61) is undefined. In order to overcome this problem, one can impose a restriction where (61) 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 $\bar{f}_{sf}$".*

Choice of Gains R_p, R_d, K_p, K_d

When implementing the combined adaptive inner-outerloop controller for position and attitude control, the poles for the combined error dynamics must be selected appropriately. In the following analysis we examine the situation where inversion model error is compensated for accurately by the neural-network and assume that the system is exactly feedback linearized. The inner loop and outer loop each represent a second order system and the resulting position dynamics $\frac{x(s)}{x_c(s)}$ is a fourth order system.

Considering the closed loop longitudinal dynamics, near hover, and acknowledging an abuse of notation, it may be written as

$$\ddot{x} = a_{des} = \ddot{x}_c + K_d(\dot{x}_c - \dot{x}) + K_p(x_c - x) \tag{63}$$

$$\ddot{\theta} = \alpha_{des} = \ddot{\theta}_g + K_d(\dot{\theta}_g - \dot{\theta}) + K_p(\theta_g - \theta) \tag{64}$$

where, R_p, R_d, K_p, K_d are the PD compensator gains for the innerloop (pitch angle) and outerloop (fore-aft position). x is the position, θ the attitude and θ_g the attitude command. Normally, $\theta_g = \theta_c + \theta_{ol}$ where θ_c is the external command and θ_{ol} the outerloop generated attitude command. Here, we assume that the external attitude command and its derivatives are zero, hence, $\theta_g = \theta_{ol}$. In the following development, we will find the transfer function $\frac{x(s)}{x_c(s)}$ and use that to place the poles of the combined inner-outer loop system in terms of the PD compensator gains R_p, R_d, K_p, K_d.

Ignoring contributions of $\dot{\theta}_g(s)$ and $\ddot{\theta}_g(s)$, the pitch dynamics Eq. (64), may be rewritten in the form of a transfer function as

$$\theta(s) = \frac{\theta(s)}{\theta_g(s)}\theta_g(s) = \frac{K_p}{s^2 + K_d s + K_p}\theta_g(s) \tag{65}$$

If the outerloop linearizing transformation used to arrive at Eq. (63) has the form $\ddot{x} = f\theta$, where $f = -g$ and g is gravity, it may be written as,

$$s^2 x(s) = f\theta(s) \tag{66}$$

The outerloop attitude command may be generated as

$$\theta_{ol} = \frac{\ddot{x}_{des}}{f} = \frac{a_{des}}{f} \tag{67}$$

Noting that $\theta_g = \theta_{ol}$, if, $\theta_c = 0$,

$$\theta_g = \theta_{ol} = \frac{1}{f}\left[\ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x)\right] \tag{68}$$

Using Eq. (65) and Eq. (68) in Eq. (66),

$$s^2 x(s) = \frac{K_p\left[s^2 x_c + R_d s(x_c - x) + R_p(x_c - x)\right]}{s^2 + K_d s + K_p} \tag{69}$$

Rearranging the above equation results in the following transfer function

$$\frac{x(s)}{x_c(s)} = \frac{K_p s^2 + K_p R_d s + K_p R_p}{s^4 + K_d s^3 + K_p s^2 + K_p R_d s + K_p R_p} \tag{70}$$

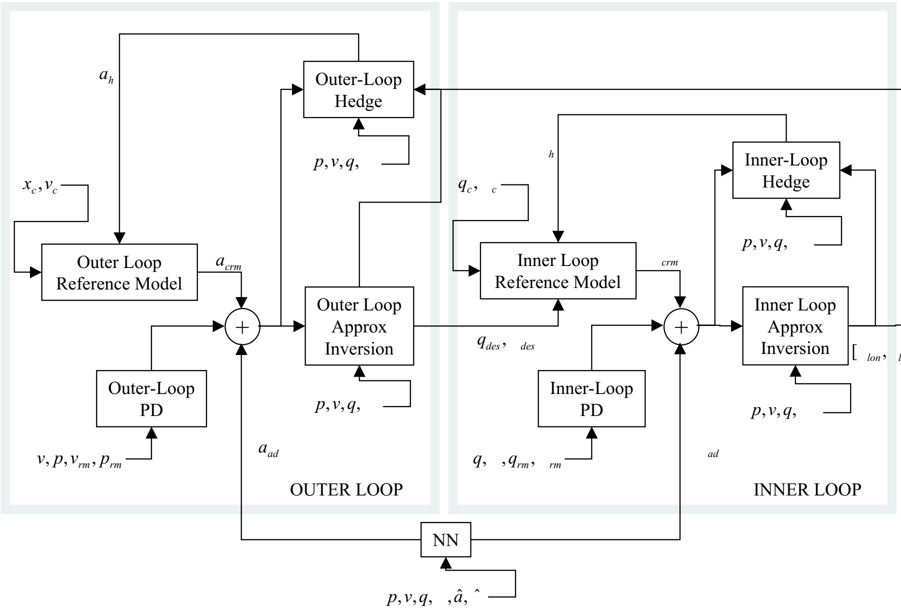

Fig. 4 Detailed inner and outer loop controller architecture for an autonomous helicopter

A 4^{th} order characteristic polynomial written as the product of two second order systems may be expressed as.

$$\begin{aligned}\Upsilon(s) &= (s^2 + 2\zeta_o\omega_o + \omega_o^2)(s^2 + 2\zeta_i\omega_i + \omega_i^2)\\ &= s^4 + (2\zeta_i\omega_i + 2\zeta_o\omega_o)s^3\\ &+ (\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2)s^2\\ &+ (2\zeta_o\omega_o\omega_i^2 + 2\omega_o^2\zeta_i\omega_i)s + \omega_o^2\omega_i^2 \qquad (71)\end{aligned}$$

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

Comparing the coefficients of the poles of Eq. (70) and Eq. (71) allows the gains to be expressed as a function of the desired pole locations

$$\begin{aligned}R_p &= \frac{\omega_o^2\omega_i^2}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2}\\ R_d &= 2\frac{\omega_o\omega_i(\zeta_o\omega_i + \omega_o\zeta_i)}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2}\\ K_p &= \omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2\\ K_d &= 2\zeta_i\omega_i + 2\zeta_o\omega_o \qquad (72)\end{aligned}$$

Numerical Results

The proposed guidance and control architecture was applied to a nonlinear simulation of a Yamaha R - Max helicopter, using a simulation tool called *ESim*. The R - Max helicopter weighs about 128 lbs (empty) and has a main rotor radius of 5.05 ft. Nominal rotor speed is 700 revolutions per minute. Its practical payload capability is about 66 lbs with a flight endurance of greater than 60 minutes. It is also equipped with a control rotor.

The helicopter was commanded to perform a circular maneuver in the North-East plane with constant altitude. Constant speed was commanded around the circuit at the same time commanding heading by declaring the number of pirouettes to be performed per loop. The trajectory equations are given by

$$\begin{aligned}p_c &= \begin{bmatrix}\frac{V}{\omega}\cos(\omega t)\\ \frac{V}{\omega}\sin(\omega t)\\ -h\end{bmatrix}, \qquad v_c= \begin{bmatrix}-V\sin(\omega t)\\ V\cos(\omega t)\\ 0\end{bmatrix}\\ \psi_c &= \omega t f \qquad (73)\end{aligned}$$

where, t is simulation time and h is a constant altitude command. V is speed of the maneuver, ω is angular speed of the helicopter around the local frame origin, and f is number of pirouettes to be performed per circuit. If $\omega = \pi/2\ rad/s$, the helicopter will go around once every 4 seconds. If $f = 1$, the helicopter will rotate once every revolution.

Parameters

The controller parameters for the innerloop involved choosing K_p, K_d based on a natural frequency of $3, 3, 5\ rad/s$ for the roll, pitch and yaw channels respectively and damping ratio of 0.9. For the outerloop, R_p, R_d were chosen based on a natural frequency of $1, 1, 1.5\ rad/s$ for the x, y and z body axis all with a damping ratio of unity. The neural network was chosen to have 5 hidden layer neurons. The inputs to the network included body axis velocities and rates as well as the estimated pseudo controls i.e, $x_{in} = [v_B, \omega_B, \hat{a}, \hat{\alpha}]$. The output layer learning rates[9] Γ_W were set to unity for all channels and a learning rate of $\Gamma_V = 10$ was set for all inputs. Limits were imposed on the maximum angular rate and translational velocity that the reference model dynamics may demand. The limits were set to $v_{lim} = 50\ ft/s$ and $\omega_{lim} = 3\ rad/s$ in the reference model dynamics. Maneuver parameters involved setting the speed of the maneuver to $V = 10\ ft/s$,

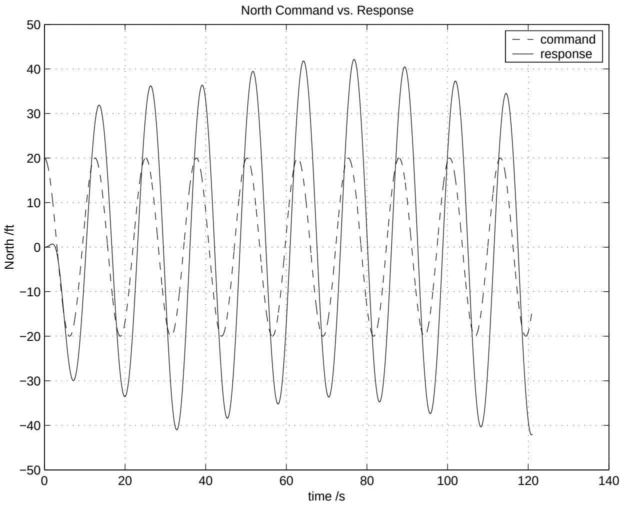

Fig. 5 North Position with innerloop adaptation only

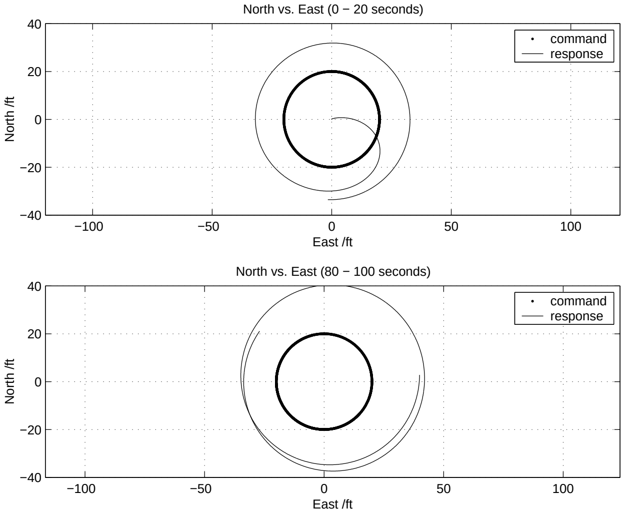

Fig. 6 x-y plane trajectory plot of helicopter position with innerloop adaptation only

$\omega = 0.5\ rad/s$ and frequency of pirouette to once every revolution i.e., $f = 1$.

Simulation

Maneuvers with the above parameters were simulated with and without outerloop adaptation. Innerloop adaptation was enabled in all cases. Figure 5 shows the position tracking in the x-axis of the local frame (North). Figure 6 illustrates the position plot of the helicopter in the x-y plane during the first 30 seconds of simulation and the latter 80-100 seconds of simulation at which point the innerloop may be assumed to be fully trained. Tracking without adaptation in the outerloop is quite poor due to the simplistic point-mass model used for outerloop inversion.

Figure 7 and Figure 8 shows similar plots but with adaptation in both the inner and outer loops. The network is able to compensate for approximations in both the innerloop dynamics and the outerloop dynamics.

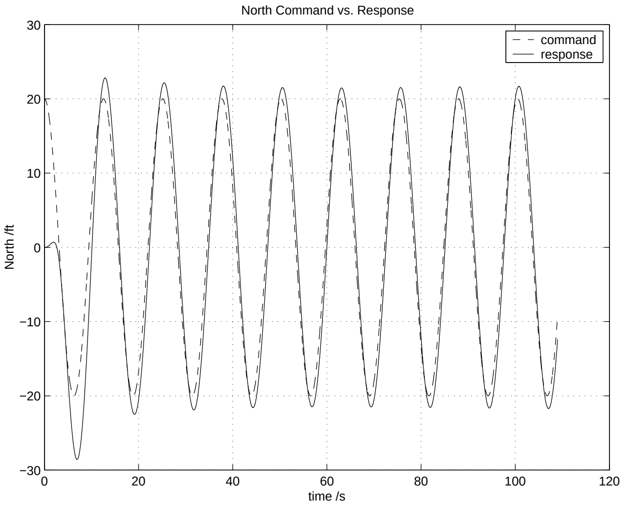

Fig. 7 North Position with both inner and outer loop adaptation

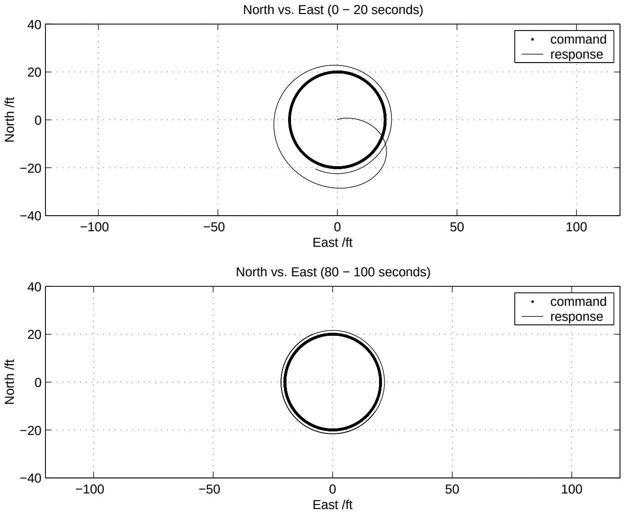

Fig. 8 x-y plane trajectory plot of helicopter position with both inner and outerloop adaptation

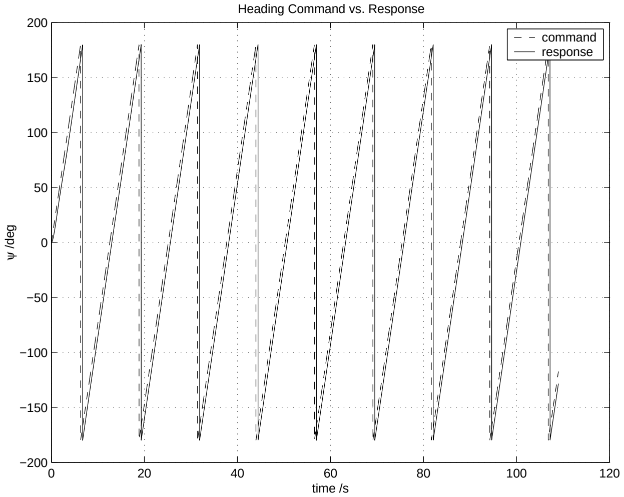

Fig. 9 Heading command and response for inner and outer loop adaptation case

Fig. 10 The Yamaha R-Max Helicopter

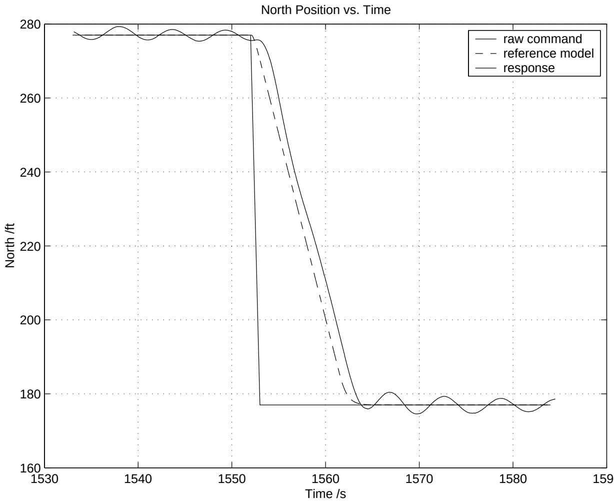

Fig. 11 North position step command response during flight test

In Figure 8, during the latter parts of the simulation, the network appears to be fully trained with accurate tracking to within 2 ft. Additionally, Figure 9 illustrates the heading command and response of the helicopter for the full adaptation case.

Flight Test

Finally, the controller was flight tested on the Yamaha R-Max helicopter shown in Figure 10. Figure 11 shows the *flight-test* response of the helicopter to a step position command of $100ft$, where, the outerloop bandwidth was set to $2rad/s$ with unity damping for all channels. The innerloop bandwidths were set to $3rad/s$ with unity damping, together with an angular rate limit of $1rad/s$. The graph shows the raw step command together with the reference model output and helicopter response. Additionally, a velocity limit of $10ft/s$ was included in the outerloop reference model, this is evident from the almost linear response in position during the maneuver.

Conclusions

This paper presents results where the inner loop attitude control and outer loop trajectory control loops have been combined using Pseudo Control Hedging (PCH) in a way that does not affect adaptation. Both loops can now use the same adaptive element to cancel model errors. Additionally PCH was also used to prevent unwanted adaptation to actuator limits and dynamics in the innerloop. Using this architecture, a consolidated reference command x_c that includes position, velocity, attitude and angular rate may be provided to the control system. This reference signal is automatically augmented with any attitude corrections required to follow a trajectory. Using PCH along with expressions for the poles of the combined inner-outer loop error dynamics alleviates bandwidth separation requirements. This allows increased bandwidth in the outerloop leading to better overall tracking performance.

Appendix : Proof of Theorem 1

Considering the error dynamics of Eq. (20), using Eq. (48) for $\hat{\nu}_{ad}$ and noting that the NN reconstruction error with optimal weights is given by

$$\epsilon = \nu^*_{ad} - \Delta = W^{*T}\sigma(V^{*T}\bar{x}) - \Delta$$

the error dynamics may be written as

$$\dot{e} = Ae + B\left[W^T\sigma(V^T\bar{x}) - W^{*T}\sigma(V^{*T}\bar{x}) + \epsilon + \nu_r\right] \tag{74}$$

For clarity, hereafter, $\sigma = \sigma(V^T\bar{x})$, $\sigma^* = \sigma(V^{*T}\bar{x})$ and $\sigma' = \sigma'(V^T\bar{x})$. Using a Taylor-series expansion for σ with respect to W and V, Eq. (74) may be rewritten as

$$\dot{e} = Ae + B\left[\tilde{W}^T(\sigma - \sigma' V^T\bar{x}) + W^T\sigma'\tilde{V}^T\bar{x} + w + \nu_r\right]$$

where, after defining, $\tilde{W} \triangleq W - W^*, \tilde{V} \triangleq V - V^*$,

$$w = \epsilon - W^{*T}\left[\sigma^* - \sigma + \sigma'\tilde{V}^T\bar{x}\right] + \tilde{W}^T\sigma' V^{*T}\bar{x}$$

The norm of w may bounded as[9]

$$\|w\| \leq \bar{\epsilon}+2\bar{Z}(b_w+n_2)+2\bar{Z}\bar{a}(b_w+n_2)(1+b_w+n_2)\|\tilde{Z}\|\|\bar{x}\|$$

where $\tilde{Z} = Z - Z^*$, $\bar{a}$ is the maximum activation potential of Eq. (27). A bound on the norm of the NN inputs $\bar{x}$ may be expressed as

$$\|\bar{x}| \leq b_v + \bar{x}_c + \bar{Z} + \|e\| + (1 + b_w + n_2)\|\tilde{Z}\|$$

Hence, the upper bound on the norm of w may finally be expressed as

$$\|w\| \leq c_0 + c_1\|\tilde{Z}\| + c_2\|e\|\|\tilde{Z}\| + c_3\|\tilde{Z}\|^2$$

where, c_0, c_1, c_2, c_3 are known constants. Consider the Lyapunov candidate function,

$$L(e, \tilde{W}, \tilde{V}) = \frac{1}{2}\left[e^T Pe + tr(\tilde{W}\Gamma_w^{-1}\tilde{W}) + tr(\tilde{V}\Gamma_v^{-1}\tilde{V})\right]$$

Using the NN weight update laws of Eq. (51) and Eq. (52), the time derivative of L along the state trajectories may be expressed as

$$\dot{L} = -\frac{1}{2}e^T Qe + r^T(w + \nu_r) - \kappa\|e\|tr(\tilde{Z}^T Z)$$

Using $-tr(\tilde{Z}^TZ) \le \|\tilde{Z}\|_F\|Z^*\|_F - \|\tilde{Z}\|_F^2$ and using Eq. (50) for the robustifying term ν_r, results in

$$\begin{aligned}\dot{L} \le & -\frac{1}{2}e^TQe + \|r\|\|w\| - r^TK_rr(\|Z\|_F + \bar{Z}) \\ & - \kappa\|e\|\|\tilde{Z}\|_F^2 + \kappa\|e\|\|\tilde{Z}\|_F\bar{Z}\end{aligned}$$

Using the bound on w and requiring that

$$\lambda_{min}(K_r) \ge \frac{c_2}{\|PB\|} \qquad \text{and} \qquad \kappa > \|PB\|c_3$$

$\dot{L}$ may be further bounded as

$$\begin{aligned}\dot{L} \le & -\frac{1}{2}\lambda_{min}(Q)\|e\|^2 - (\kappa - \|PB\|c_3)\|e\|\|\tilde{Z}\|_F^2 \\ & + a_0\|e\| + a_1\|e\|\|\tilde{Z}\|_F\end{aligned}$$

where,

$$\begin{aligned}a_0 &= (\bar{\epsilon} + 2\bar{Z}(b_w + n_2))\|PB\| \\ a_1 &= 2\bar{Z}\bar{a}(b_w + n_2)(1 + b_w + n_2) \\ & \quad (b_v + \bar{x}_c + \bar{x}_{rm} + \bar{Z})\|PB\| + \kappa\bar{Z}\end{aligned}$$

Now, $\dot{L} \le 0$ when,

$$\|\tilde{Z}\|_F \ge Z_m = \frac{a_1 + \sqrt{a_1^2 + 4a_0(\kappa - \|PB\|c_3)}}{\kappa - \|PB\|c_3}$$

or

$$\|e\| \ge \frac{a_0 + a_1Z_m}{\frac{1}{2}\lambda_{min}(Q)}$$

By selecting $\lambda_{min}(Q), \kappa, \Gamma_W$ and Γ_v, $\dot{L} \le 0$ everywhere outside a compact set that is entirely within the largest level set of L, which in turn lies entirely within the compact set D.[9] Thus, for initial conditions within D, the tracking error e and weights $\tilde{Z}$ are uniformly ultimately bounded.

Acknowledgements

This work was supported in part by DARPA's Software Enabled Control Program under Contracts #33615-98-C-1341 and #33615-99-C-1500 with William Koenig of the Air Force Research Laboratory (AFRL) as contract monitor. We also acknowledge the contributions of Eric Corban, Jeong Hur, Wayne Pickell, Joerg Dietrich and Henrik Christophersen who made the flight tests possible.

References

[1] Frazzoli, E., Dahleh, M. A., and Feron, E., "Robust Hybrid Control for Autonomous Vehicles Motion Planning," Tech. Rep. LIDS-P-2468, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, 1999.

[2] Sanders, C., DeBitetto, P. D., Vuong, H. F., and Levenson, N., "Hierarchical Control of Small Autonomous Helicopters," *IEEE Conference on Decision and Control*, Tampa, Florida, 1998.

[3] Rysdyk, R. T. and Calise, A. J., "Nonlinear Adaptive Flight Control Using Neural Networks," *IEEE Controls Systems Magazine*, Vol. 18, No. 6, dec 1998.

[4] Shim, D. H., Koo, T. J., Hoffmann, F., and Sastry, S., "A comprehensive study of control design for an autonomous helicopter," *IEEE Conference on Decision and Control*, Dec. 1999.

[5] Calise, A. J., Lee, H., and Kim, N., "High Bandwidth Adaptive Flight Control," *AIAA Guidance, Navigation and Control Conference*, Denver, CO, aug 2000.

[6] Calise, A. J., Lee, S., and Sharma, M., "Development of a Reconfigurable Flight Control Law for Tailless Aircraft," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 896–902.

[7] Brinker, J. and Wise, K., "Flight testing of a reconfigurable flight control law on the X-36 tailless fighter aircraft," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 903–909.

[8] Johnson, E. N., Calise, A. J., and Corban, J. E., "Reusable Launch Vehicle Adaptive Guidance and Control Using Neural Networks," *AIAA Guidance, Navigation and Control Conference*, No. 4381, 2001.

[9] Johnson, E. N., *Limited Authority Adaptive Flight Control*, Ph.D. thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, dec 2000.

[10] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulaion*, John Wiley & Sons, New York, 1992.

[11] Johnson, E. N., Calise, A. J., El-Shirbiny, H. A., and Rysdyk, R. T., "Feedback Linearization with Neural Network Augmentation applied to X-33 Attitude Control," *AIAA Guidance, Navigation and Control Conference*, Denver, CO, aug 2000.

[12] K.Hornik, M.Stinchombe, and H.White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, 1989, pp. 359–366.

[13] Munzinger, C., *Development of a Real-Time Flight Simulator for An Experimental Model Helicopter*, Master's thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, jul 1997.

[14] Johnson, E. N. and DeBitetto, P. D., "Modelling and Simulation for Small Autonomous Helicopter Development," *AIAA Modelling & Simulation Technologies Conference*, Monterey, California, 1997.

[15] Gavrilets, V., Mettler, B., and Feron, E., "Nonlinear Model for a Small-Sized Acrobatic Helicopter," *AIAA Guidance, Navigation and Control Conference*, No. 2001-4333, Montréal, Quebec, Canada, Aug. 2001.