

ADAPTIVE TRAJECTORY BASED CONTROL FOR AUTONOMOUS HELICOPTERS

Suresh K. Kannan¹ and Eric N. Johnson²

School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332

Abstract

For autonomous helicopters it is common to design a high-performance tracking controller for the attitude dynamics (innerloop) followed by a simpler, lower bandwidth design that tracks commanded position and velocity (outerloop). Separating these two designs places restrictions on the maximum bandwidth of the outerloop. This paper continues to make a conceptual separation between inner and outerloop designs, but, the final choice of compensator gains is made by treating both loops together. The controller design for both loops use feedback linearization with an adaptive element (neural network) to account for model inversion error. Pseudo Control Hedging is used to protect the adaptive element from actuator saturation nonlinearities and also from inner-outerloop interaction. The resulting control system allows position, velocity, attitude and angular rate commands. The outerloop however augments an attitude correction, that allows tracking of position and velocity in addition to attitude and angular rate.

Nomenclature

\oplus	quaternion addition operator
α	angular acceleration, rad/s^2
a	translational acceleration, ft/s^2
Δ	model error
δ	control vector / actuator deflections
NN	neural network
v	pseudocontrol
ω	angular velocity, rad/s
p	position, ft
PCH	pseudocontrol 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 term
des	desired
f	force
h	hedge
m	moment
ol	outerloop correction
p	proportional
r	robustifying term
rm	reference model

Introduction

Unmanned helicopters are versatile machines that can perform aggressive maneuvers. They also have a distinct advantage over fixed-wing aircraft especially in urban environments where hover-capability is required. Additionally, the ability to perform tight aggressive turns around buildings is a requirement for the operation of unmanned aerial vehicles in an urban low-altitude setting. Many planning algorithms[1] have been developed to generate aggressive trajectories in an obstacle rich environment. The realization of these trajectories is however, limited by the tracking capability of the flight controller.

The most common control design approach for autonomous helicopters is a two-loop design. First an innerloop that provides high-performance tracking of attitude commands is designed, subsequent to which an outerloop that tracks translational variables is designed. This two loop

¹ Graduate Research Assistant, suresh_kannan@ae.gatech.edu

² Assistant Professor, eric.johnson@ae.gatech.edu

approach makes the design conceptually simpler, but, generally requires the assumption of time-scale separation, where the outerloop bandwidth has to be considerably lower than that of the innerloop in order to prevent interaction between the two loops. Researchers have also used model based robust control approaches [2,3] to tackle the flight control problem. A model based approach however has problems in areas of the flight envelope where the model used is no longer valid, although success in this area has been reported recently in [3].

This paper takes a different approach and is based on direct adaptive control, which has recently proven itself as an enabling technology for practical flight control systems. This technology has been successfully applied [4] to the recent USAF Reconfigurable Control for Tailless Fighter Aircraft (RESTORE) culminating in a successful flight demonstration [5] 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 [6] and evaluated successfully in simulation for various failure cases.

This paper employs dynamic inversion to approximately linearize the attitude and translational dynamics separately. The resulting set of 2nd order integrator systems (innerloop and outerloop) are stabilized using a linear proportional-derivative controller and desired response imposed using reference models in each of the degrees of freedom. Hence, in this respect the conceptual separation between innerloop and outerloop is used.

Given attitude and angular rate commands, q_c, ω_c , the innerloop, which controls all the moment producing actuators δ_m , (*longitudinal stick, lateral stick and pedal*) is first designed. Next, given position and velocity commands, p_c, v_c , the outerloop, that controls the main force producing actuator δ_m (*collective*) is designed. Additionally, the outerloop also generates an attitude correction q_{ol} , which is augmented with the external attitude command q_c . It is instructive to note here that from the outerloop's point of view the attitude correction and collective input are essentially actuator commands. Any innerloop dynamics or dynamics in the collective actuator

now appears like actuator dynamics to the outerloop. Similarly, any limits imposed on the attitude of the aircraft appears like actuator saturation to the outerloop. This architecture is illustrated in Figure 1. A detailed view is presented in Figure 15.

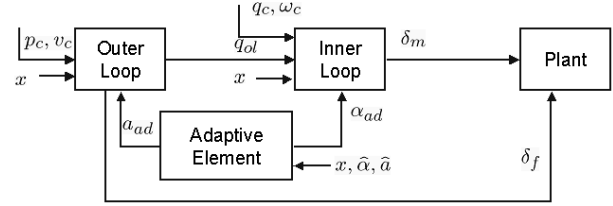


Figure 1: Overview of the inner and outerloop controller architecture

The effect of parametric uncertainty arising due to approximate inversion is minimized using an adaptive element. This adaptive element could be a simple set of integrators or an elaborate function approximator such as a neural network. The neural network is updated online using a stable adaptive law derived from Lyapunov stability analysis [7]. Using a neural network function approximator allows one to treat the model error as unstructured uncertainty that needs to be cancelled. Introducing adaptation however leaves the adaptive element vulnerable to unwanted adaptation to plant input dynamics such as moment actuator saturation and associated dynamics as seen by the innerloop. A similar problem occurs with respect to the outerloop due to unwanted adaptation to the innerloop and collective actuator dynamics.

This problem is alleviated using Pseudocontrol-Hedging (PCH) [6,8], which modifies the inner and outerloop reference model dynamics in a way that allows continued adaptation in the presence of actuator saturation. Finally, both loops are considered *together* by examining the position response transfer function, $\frac{x(s)}{x_c(s)}$. This

allows a choice of inner and outerloop compensator gains that minimizes the requirement of bandwidth separation.

Controller Development

A 6-degree-of-freedom helicopter may be represented as a rigid body using the following dynamic and kinematic equations:

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

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

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

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

where, $q \in R^4$ represents the attitude quaternion, $\omega \in R^3$ is the angular velocity, $p \in R^3$ is the position vector and $v \in R^3$ is the velocity of the vehicle. Equation 3, represents the quaternion propagation equations[9]. $\dot{\omega}$ represents the attitude dynamics and \dot{v} represents translational dynamics. The state vector x may now be defined as $x \equiv [p \ v \ q \ \omega]$. The main force generating control (*collective*) is denoted by δ_f , and the main moment generating control vector (*longitudinal stick, lateral stick and pedal*) is denoted by δ_m .

Introducing our best available estimate of the translational and attitude dynamics denoted by \hat{a} and $\hat{\alpha}$ respectively, the control deflections and attitude that is expected to achieve the desired accelerations may be represented by

$$\begin{bmatrix} \delta_{f_{des}} \\ q_{ol} \end{bmatrix} = \hat{a}^{-1}(p, v, q, \omega, a_{des}, \delta_m) \quad (5)$$

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

where, a_{des}, α_{des} , represent desired acceleration values (pseudocontrol) and have yet to be designed. $\delta_{m_{des}}$, represents the actuator deflections that are expected to achieve the desired angular accelerations, $\delta_{f_{des}}$, represents the collective input that will achieve the desired z-axis acceleration in the body axis and q_{ol} , represents the attitude correction that will generate the desired translational accelerations in the body x and y axis and will be augmented with the external command

using quaternion addition. Hence, the desired attitude may be represented as

$$q_{des} = q_c \oplus q_{ol} \quad (6)$$

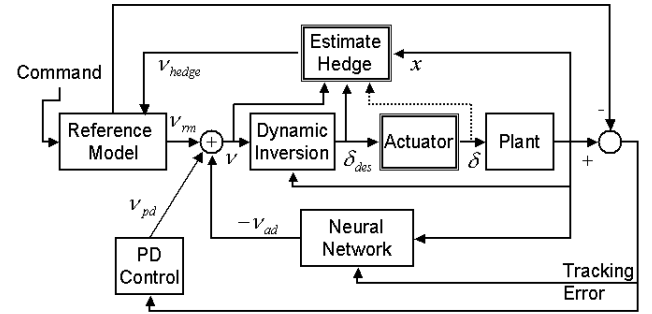


Figure 2: Internals of the inverting controller, representative of both the innerloop and outerloop

In this development it is assumed that actual actuator positions are known. For situations where only estimates of actuator position are available see [8]. Additionally, the state variables $x \equiv [p \ v \ q \ \omega]$ are available from a Kalman Filter and separation between observer and controller is assumed. Substituting Equation 5 into the equations of motion result in two sets of double integrator systems with an additional model error term Δ , and may be written as

$$\dot{v} = a_{des} + \Delta_a \quad (7)$$

$$\dot{\omega} = \alpha_{des} + \Delta_\alpha$$

Here,

$$\Delta = \begin{bmatrix} \Delta_a \\ \Delta_\alpha \end{bmatrix} = \begin{bmatrix} a - \hat{a} \\ \alpha - \hat{\alpha} \end{bmatrix} \quad (8)$$

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

$$a_{des} = a_{crm} + a_{pd} - \hat{a}_{ad} \quad (9)$$

$$\alpha_{des} = \alpha_{crm} + \alpha_{pd} - \hat{\alpha}_{ad}$$

where, a_{crm}, α_{crm} are outputs of a reference model that dictate the shape of the desired response.

a_{pd}, α_{pd} are outputs of a Proportional-Derivative (PD) compensator that stabilizes the double

integrator system in Equation 7 and shapes the error dynamics. Finally, $\hat{a}_{ad}, \hat{\alpha}_{ad}$ represents the output of an adaptive element that is designed to cancel the effects of model error Δ_a and Δ_α . This model inverting control architecture is illustrated in Figure 2 and is representative of both the innerloop and outerloop designs. Following is a discussion on each of the terms that constitute the pseudocontrol signals in Equation 9.

Reference Model and Pseudo Control Hedging

If the reference model dynamics are designed without consideration of plant input dynamics such as control saturation, incorrect and unwanted adaptation to these characteristics would take place. For example, 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 tracking without consideration of the innerloop dynamics. The technique of pseudocontrol hedging (PCH) is used to prevent the adaptive element from trying to adapt to selected system input characteristics. One way to describe PCH 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*. The reference model dynamics of the inner and outerloops may be expressed as

$$\dot{v}_{rm} = a_{crm}(p_{rm}, v_{rm}, p_c, v_c) - a_h \quad (10)$$

$$\dot{\omega}_{rm} = \alpha_{crm}(q_{rm}, \omega_{rm}, q_c \oplus q_{ol}, \omega_c) - \alpha_h$$

where a_h and α_h are the hedge signals and represent the difference between commanded pseudocontrol and achieved pseudocontrol. They may be expressed as

$$a_h = a_{des} - \hat{a}(x, \delta_f, \delta_m) \quad (11)$$

$$\alpha_h = \alpha_{des} - \hat{\alpha}(x, \delta_f, \delta_m)$$

In computing these estimates, the actuator positions are assumed to be known. Results that use estimates of actuator position may be found in [8]. The functions a_{crm}, α_{crm} are chosen to represent the desired response and are usually designed to

include limits on translational and angular rates. A specific choice for the helicopter will be provided later.

The tracking error vector e , may now be defined as

$$e \equiv \begin{bmatrix} p_{rm} - p \\ v_{rm} - v \\ \tilde{Q}(q_{rm}, q) \\ \omega_{rm} - \omega \end{bmatrix} \quad (12)$$

where, $\tilde{Q} : R^4 \times R^4 \rightarrow R^3$, is a function which, given two quaternions results in an error angle vector with three components [7].

PD Compensator

The output of the PD compensator may be written as

$$\begin{bmatrix} a_{pd} \\ \alpha_{pd} \end{bmatrix} = \begin{bmatrix} R_p & R_d & 0 & 0 \\ 0 & 0 & K_p & K_d \end{bmatrix} e \quad (13)$$

where, $R_p, R_d, K_p, K_d \in R^{3 \times 3}$ are diagonal and positive definite matrices. Such a choice of gains stabilizes the double integrator system of Equation 7, the actual values of which will be designed later. Now, the tracking error dynamics may be found by directly differentiating Equation 12 and after rearranging may be expressed as

$$\dot{e} = Ae + B[\hat{v}_{ad} - \Delta(x, \delta)] \quad (14)$$

where,

$$A = \begin{bmatrix} 0 & I & 0 & 0 \\ -R_p & -R_d & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -K_p & -K_d \end{bmatrix} \quad (15)$$

$$B = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix}$$

Now, only the adaptive term \hat{v}_{ad} remains to be designed to cancel the effect of inversion error.

Adaptive Element

Δ represents function approximation error. Given a sufficient number of hidden layer neurons and appropriate inputs, it should be possible to train a single hidden layer neural network online to cancel the effect of Δ .

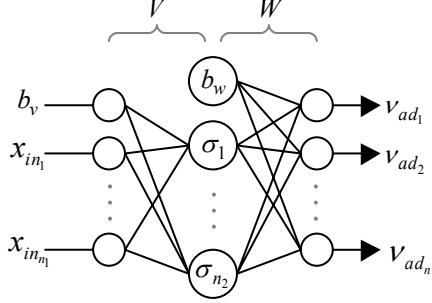


Figure 3: Neural Network with one hidden layer

Figure 3 illustrates the structure of a single hidden layer network (SHL), the input-output map of which may be expressed as

$$v_{ad_k} = b_w \theta_{w,k} + \sum_{j=1}^{n_2} w_{j,k} \sigma_j \quad (16)$$

where $k = 1, \dots, n_3$ and

$$\sigma_j = \sigma \left(b_v \theta_{v,j} + \sum_{i=1}^{n_1} v_{i,j} x_i \right) \quad (17)$$

Here n_1 , n_2 , and n_3 are the number of input nodes, hidden layer nodes, and outputs respectively. The scalar function σ is a sigmoidal activation function that represents the ‘‘firing’’ characteristics of the neuron, e.g.

$$\sigma(z) = \frac{1}{1 + e^{-az}} \quad (18)$$

The factor a is the activation potential, and should normally be a distinct value for each neuron. For convenience, define the two weight matrices

$$V = \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & v_{n_1,n_2} \end{bmatrix} \quad (19)$$

$$W = \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix} \quad (20)$$

and define a new sigmoid vector

$$\bar{\sigma}(z) = [b_w \ \sigma(z_1) \ \sigma(z_2) \ \cdots \ \sigma(z_{n_1})]^T$$

where $b_w \geq 0$ allows for the threshold θ_w to be included in the weight matrix W . Define

$$\bar{x} = [b_v \ x_1 \ x_2 \ \cdots \ x_{n_1}]^T$$

$b_v \geq 0$ is an input bias that allows for the threshold θ_v to be included in the weight matrix V . With the above definitions, the input-output map of the SHL NN in the controller architecture can be written in matrix form as

$$v_{ad} = W^T \sigma(V^T \bar{x}) \quad (21)$$

Boundedness

It can be shown that the system given by Equations 1-4 with the inverse law given by Equation 5, a few assumptions described in [7] along with the neural network weight update equations given by

$$\dot{W} = -\left[(\sigma - \sigma' V^T \bar{x}) r^T + \kappa \|e\| W \right] \Gamma_W \quad (22)$$

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

where,

$$r = (e^T P B)^T \quad (23)$$

$$\hat{v}_{ad} = v_{ad} - v_r \quad (24)$$

$$v_{ad} = W^T \sigma(V^T \bar{x}) \quad (25)$$

$$v_r = -K_r (\|Z\|_F + \bar{Z}) r \quad (26)$$

with $K_r > 0 \in R^{6 \times 6}$, and with

$\Gamma_W, \Gamma_V > 0$ and $\kappa > 0$, guarantees that the reference model tracking error (e) and Neural Network (W, V) weights are uniformly ultimately bounded. With the additional assumption that the

reference model dynamics are stable, it can be shown that all signals in the closed loop system including plant states are uniformly ultimately bounded. Γ_W, Γ_V are usually referred to as the network learning rates. An outline of the proof of boundedness and descriptions of all the variables associated with the Neural Network may be found in [7].

Trajectory tracking for an autonomous helicopter

The proposed trajectory-tracking controller was applied to a Yamaha R-Max helicopter. The reference model for both the innerloop and outerloop were chosen to be 2nd order and were chosen to have the same characteristics as their corresponding error dynamics.

$$a_{crm} = R_p(p_c - p_{rm}) + R_d(v_c - v_{rm}) \quad (27)$$

$$\dot{v} = a_{crm} - a_h$$

$$\alpha_{crm} = K_p \tilde{Q}(q_{des}, q_{rm}) + K_d(\omega_c - \omega_{rm}) \quad (28)$$

$$\dot{\omega}_{rm} = \alpha_{crm} - \alpha_h$$

In practice however, velocity and angular rate limits are placed on the response of the system. This may be achieved by modifying the above equations to the form below

$$a_{crm} = R_d[v_c - v_{rm} + sat(R_d^{-1}R_p(p_c - p_{rm}), v_{lim})]$$

$$\alpha_{crm} = K_d[\omega_c - \omega_{rm} + sat(K_d^{-1}K_p \tilde{Q}, \omega_{lim})]$$

where, the function *sat* is the saturation function with v_{lim} and ω_{lim} representing the translational and angular rate limits to be imposed.

Approximate Model

For purposes of model inversion, simple linear attitude and translational dynamics models were developed. For the attitude dynamics ($\hat{\alpha}$), a nonlinear helicopter model was linearized around hover and any coupling between the attitude and translational dynamics were ignored. The corresponding approximate model may be expressed as

$$\alpha_{des} = A_1 \begin{bmatrix} p \\ q \\ r \end{bmatrix} + A_2 \begin{bmatrix} u \\ v \\ w \end{bmatrix} + B \left(\underbrace{\begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix}}_{des} - \underbrace{\begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix}}_{trim} \right) \quad (29)$$

or,

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

where, A_1, A_2 are system matrices representing angular accelerations due to angular and translational velocities respectively. B is the control derivative matrix and $\delta_{m_{trim}}$ is the trim control vector consistent with the linear model. The inverse may be represented as

$$\delta_{m_{des}} = B^{-1}(\alpha_{des} - A_1 \omega_B - A_2 v_B) + \delta_{m_{trim}} \quad (30)$$

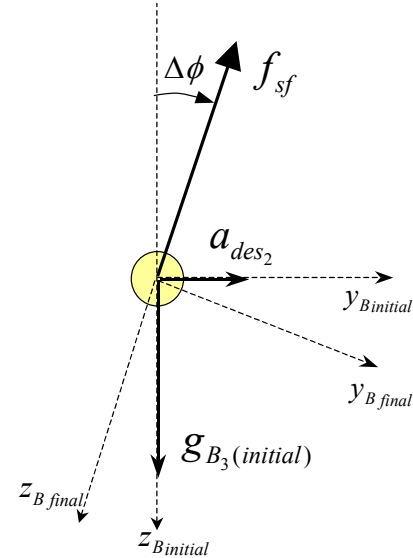


Figure 4: Approximate model of the translational dynamics

The translational dynamics (\hat{a}) were modeled as a point mass with a thrust vector that may be oriented in a given direction as illustrated in Figure 4. The corresponding approximate model may be expressed as

$$a_{des} = \begin{bmatrix} 0 \\ 0 \\ Z_{\delta_{coll}} \end{bmatrix} (\delta_{coll_{des}} - \delta_{coll_{trim}}) + L_{bv} g \quad (31)$$

Where $Z_{\delta_{coll}}$ is the control derivative for acceleration in the body z-axis. L_{bv} is the direction cosine matrix that transforms from the vehicle carried or local earth frame to the body frame. Finally, \mathbf{g} represents the gravity vector. Hence, given a desired acceleration a_{des} , the desired specific force along the body z-axis may be expressed as

$$f_{sf} = (a_{des} - L_{bv}\mathbf{g})_3 \quad (32)$$

allowing the calculation of the required collective control as

$$\delta_{f_{des}} = \delta_{coll_{des}} = \frac{f_{sf}}{Z_{\delta_{coll}}} + \delta_{coll_{trim}} \quad (33)$$

The attitude correction required to orient the thrust vector to attain the desired x-y accelerations are given by the following small angle corrections

$$\Delta\phi = -\frac{a_{des_2}}{f_{sf}}, \quad \Delta\theta = \frac{a_{des_1}}{f_{sf}}, \quad \Delta\psi = 0 \quad (34)$$

In the case of the simplified helicopter model, a heading change has no effect on accelerations in the x,y plane. The correction quaternion may now be expressed as

$$q_{ol} = q(\Delta\phi, \Delta\theta, \Delta\psi) \quad (35)$$

where $q(\cdot)$ is a function that converts Euler angles to a quaternion rotation.

Choice of gains R_p, R_d, K_p, K_d

Having chosen the form of the model inversion, simple linear analysis will yield a choice of gains that minimizes the requirement of bandwidth separation. One of the main goals of the trajectory controller is precise position tracking. For simplicity, considering just the longitudinal position dynamics, and assuming the neural network has reached its 'ideal' set of weights, thus canceling Δ exactly and also assuming that the *external* pitch attitude command is zero. One may write the position response transfer function as

$$\frac{x(s)}{x_c(s)} = \frac{K_p s^2 + K_p R_d s + K_p R_p}{s^4 + K_d s^3 + K_p s^2 + K_p R_d s + K_p R_p} \quad (36)$$

If the compensator gains for the outerloop R_p, R_d and innerloop K_p, K_d , were chosen independently, the gains may be expressed as

$$\begin{aligned} R_p &= \omega_o^2 \\ R_d &= 2\zeta_o \omega_o \\ K_p &= \omega_i^2 \\ K_d &= 2\zeta_i \omega_i \end{aligned} \quad (37)$$

where, the subscripts, i, o , represent the desired inner and outerloop values of the error dynamics bandwidth ω and damping ζ . The transfer function of Equation 36 however, indicates that the poles of the combined system will not reside at the desired locations.

Instead, we choose to place the poles of the Equation 36, which considers both loops together. The 4th order characteristic polynomial of the product of two 2nd order systems may be written as

$$\begin{aligned} Y(s) &= (s^2 + 2\zeta_o \omega_o + \omega_o^2)(s^2 + 2\zeta_i \omega_i + \omega_i^2) \\ &= s^4 + (2\zeta_i \omega_i + 2\zeta_o \omega_o)s^3 \\ &\quad + (\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2)s^2 \\ &\quad + (2\zeta_o \omega_o \omega_i^2 + 2\omega_o^2 \zeta_i \omega_i)s + \omega_o^2 \omega_i^2 \end{aligned} \quad (38)$$

Comparing the coefficients of the poles of Equation 36 and Equation 38 allows the gains to be expressed as

$$\begin{aligned} R_p &= \frac{\omega_o^2 \omega_i^2}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \\ R_d &= 2 \frac{\omega_o \omega_i (\zeta_o \omega_i + \omega_o \zeta_i)}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \\ K_p &= \omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2 \\ K_d &= 2\zeta_i \omega_i + 2\zeta_o \omega_o \end{aligned} \quad (39)$$

This choice of gains keeps poles close to desired locations even when ω_i and ω_o are close.

Numerical Results



Figure 5: Yamaha RMax Helicopter

The RMax helicopter (Figure 5) 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 flight endurance of approximately 60 minutes. It is also equipped with a control rotor. Its avionics package includes a Pentium 266 flight control computer, an IMU, a GPS unit and a sonar altimeter. Additionally, a radar altimeter is also included to complement the sonar unit for estimating the terrain height.

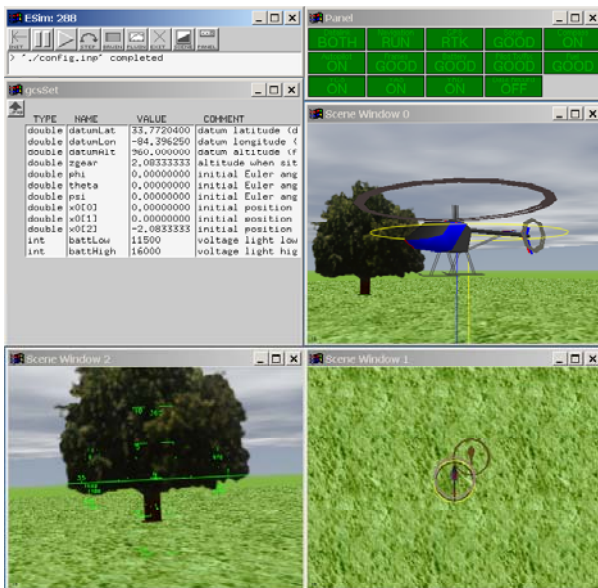


Figure 6: Simulation environment

A nonlinear model of the Yamaha-RMax helicopter was implemented in an existing

simulation environment called *Esim* shown in Figure 6. The helicopter model included flapping and control rotor 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.

Parameters

The innerloop gains K_p , K_d were based on a natural frequency of 2, 2, 2.5 rad/s for the roll, pitch and yaw channels respectively and a damping ratio of 1 in all channels. The outerloop gains were based on a natural frequency of 1, 1, 1.5 rad/s for the x, y and z body axis respectively, all with a damping ratio of 1. The neural network had 5 hidden layer neurons and the inputs to the network included body axis velocities, angular rates as well as the estimated pseudocontrols, which implicitly includes the actuator positions. This choice reflects the functional dependence of the model error $\Delta(x, \delta)$, which the neural network is trying to approximate. The output layer learning rates (Γ_w) were set to one in all channels and the input layer learning (Γ_v) rates were set to 10.

Simulation

A trajectory generator using simple kinematics with limits on acceleration was used to generate moderately aggressive trajectories for the helicopter. The controller was tested extensively in simulation using various maneuvers and also yielded the choice of compensator and network parameters presented earlier. One set of maneuvers used to evaluate the benefits of adaptation included a pirouetting circle. Here the helicopter was commanded to move in a circle at 10 ft/s and frequency of 0.5 rad/s. During this maneuver the helicopter was also commanded to pirouette once per revolution.

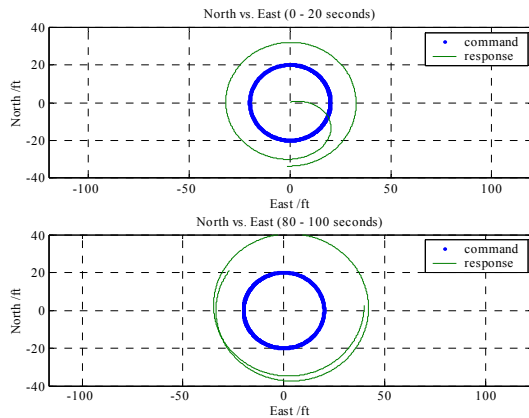


Figure 7: Pirouetting circle with only innerloop adaptation

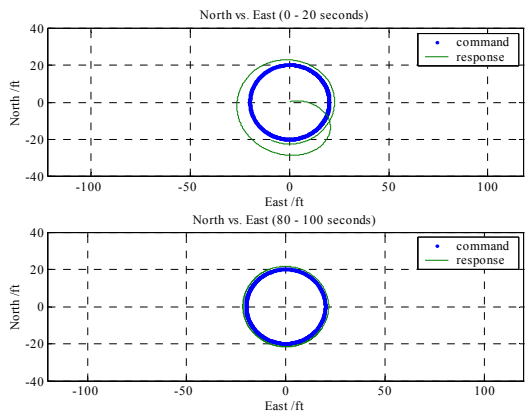


Figure 8: Pirouetting circle with both innerloop and outerloop adaptation

Figure 7 illustrates the maneuver when only the innerloop adaptation is turned. No improvement in position tracking is observed over time. This is however expected because the model used for outerloop inversion is simple and no integrators are present to subdue position errors. Figure 8 shows the tracking when adaptation is enabled for both loops. The network in this case compensates and begins to track the position command well.

Since 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 using the QNX operating system. Various maneuvers were performed during flight to test the performance of the trajectory-tracking controller. Selected results from these flight tests are provided in the following section.

Flight Tests

The adaptive controller was flown on the Yamaha helicopter following extensive testing of the Navigation system. The vehicle states including attitude quaternion were available through a Kalman filter. Testing of the controller began with simple hover followed by step responses and waypoint navigation.

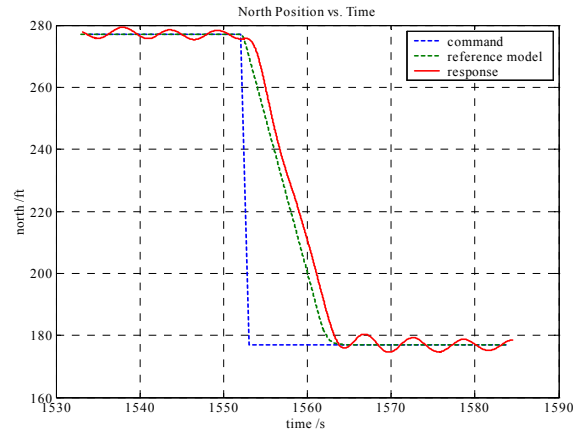


Figure 9: North position step response of 100 ft

Figure 9 illustrates a step response to a command of 100ft South in the local earth frame. The dashed lines represent the command and the solid line represents the response of the system. Also shown in this plot is the reference model output. Notice that the reference model essentially filters the raw command and also imposes a 10ft/s limit at which the position command is achieved.

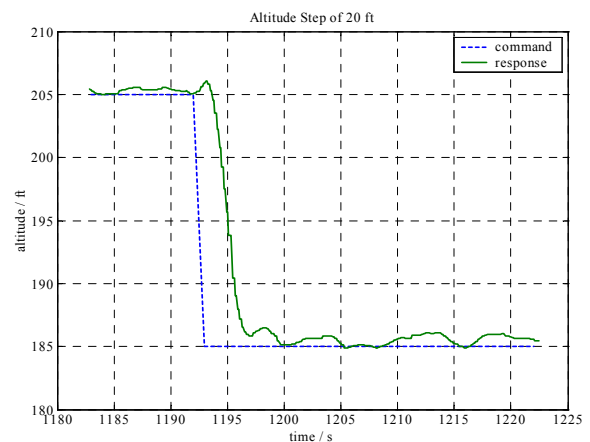


Figure 10: Altitude descent step response of 20 ft

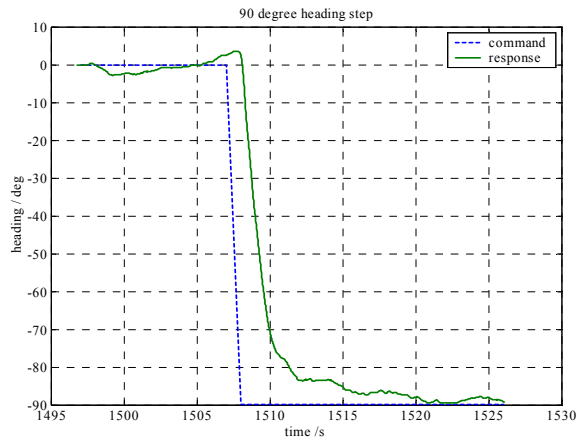


Figure 11: Heading step response of 90 degrees

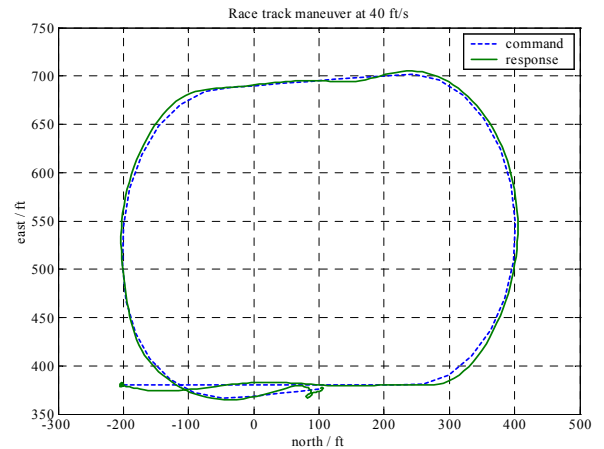


Figure 14: Flying a racetrack maneuver at 40 ft/s

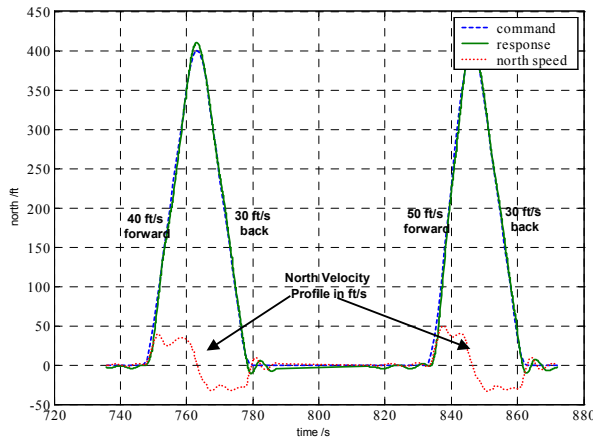


Figure 12: Fast forward and backward flight at 10 ft/s^2 . First hump is at 40 ft/s and second hump at 50 ft/s. Both maneuvers use 30 ft/s for backward flight

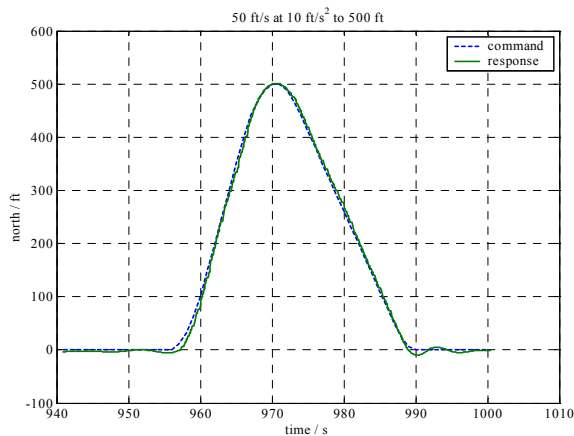


Figure 13: Forward flight at 50 ft/s and backwards at 30 ft/s, both at 10 ft/s^2

Conclusions

This paper presents the architecture and flight test results of an adaptive trajectory-tracking controller. It also demonstrates the advantages of using adaptation in the outerloop. Adaptation allows the use of simple models for inversion purposes and the neural network learns to cancel model inversion error. Since this is an online process present at all times, any changes in the dynamics with flight condition are automatically accommodated without any requirement of redesign. This is in contrast to model-based approaches where the controller is usually designed to be robust to changes in flight condition or require the scheduling of controllers or controller parameters.

The use of pseudocontrol hedging allows the adaptive element to adapt in the presence of actuator saturation. PCH was also used to hedge out pilot signals when the helicopter was in manual flight, thus setting up a novel situation where the Neural Network is adapting even when it is not in control. The combined inner-outerloop design together with PCH and careful choice of gains allows the innerloop and outerloop bandwidths to be closer together.

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. Shim, D. H., Koo, T. J., Hofmann, F., and Sastry, S., "A comprehensive study of control design for an autonomous helicopter," *IEEE Conference on Decision and Control*, Dec. 1999.
3. Civita, M. L., Papageorgiou, G., Messner, W. C., and Kanade, T., "Design and Flight Testing of a High Bandwidth H_∞ Loop Shaping Controller for a Robotic Helicopter," *AIAA Guidance, Navigation and Control Conference*, No. AIAA-2002-4846, Monterey, CA, Aug 2002.
4. 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.
5. 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.
6. 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.
7. Johnson, E. N. and Kannan, S., "Adaptive Flight Control for an Autonomous Unmanned Helicopter," *AIAA Guidance, Navigation and Control Conference*, No. AIAA-2002-4439, Monterey, CA, aug 2002.
8. Johnson, E. N., *Limited Authority Adaptive Flight Control*, Ph.D. thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, dec 2000.
9. Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, John Wiley & Sons, New York, 1992.

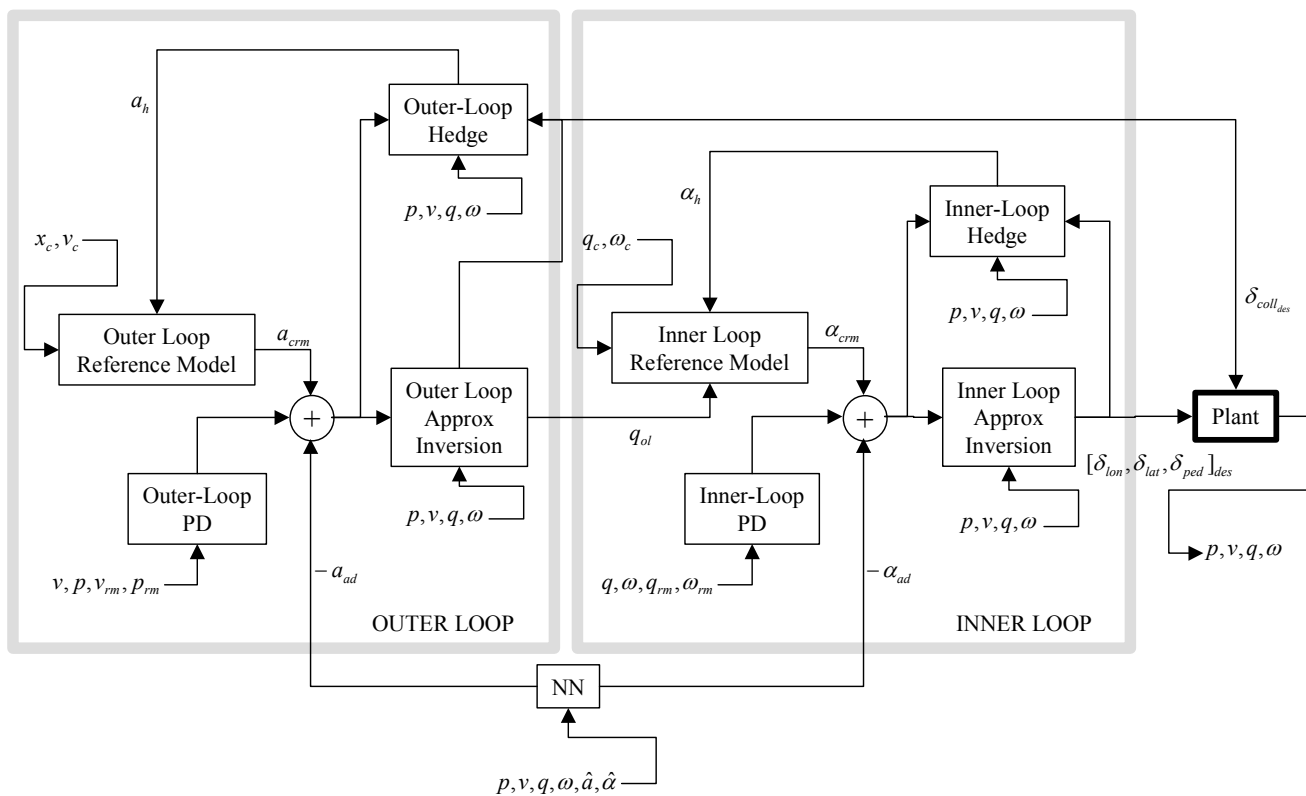


Figure 15: Detailed view of the adaptive trajectory controller