

FLIGHT EVALUATION OF AN ADAPTIVE VELOCITY COMMAND SYSTEM FOR UNMANNED HELICOPTERS

J. Eric Corban^{*}

Guided Systems Technologies, Inc., P.O. Box 1453, McDonough, GA 30253

Anthony J. Calise,[†] J. V. R. Prasad,[‡] Gerhard Heynen,^{††} Benedikt Koenig,^{††} and Jeong Hur^{‡‡}
Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332-0150

ABSTRACT

This paper examines the general requirements for fully-autonomous unmanned air vehicle in-flight mission planning and management, sensor and data fusion, synthesis of situation awareness, path planning, guidance, and control in order to define a reasonable set of near term guidance and control system requirements for a U.S. Army unmanned helicopter test-bed. Development of a near term operational capability dictates simplification of the postulated architecture down to the level of “intelligent” way point-following combined with high-level interaction with a human “supervisor.” Practical implementation leads to the development of a velocity command system driven by a relatively simple rule-based command generator. The velocity loop is based on inversion of a very simple helicopter dynamic model at hover, and produces a set of attitude commands. This outer loop is augmented with a neural network to provide adaptation to parameter and environmental uncertainty. A previously developed high-bandwidth adaptive attitude command system is employed for the inner loop. In both the inner and outer loops, pseudo control hedging is used to enable correct adaptation despite periods of control saturation. A coupled analysis of the inner and outer loops is provided for gain selection. The developed velocity command system is implemented in real-time, and evaluated both in high-fidelity nonlinear simulation and in flight test.

INTRODUCTION

For many years now, researchers and engineers have been working toward the design and development of unmanned air vehicles (UAVs) that are sufficiently automated to enable management by a remote human operator who is relieved of actual piloting duties. A number of UAV systems, most

notably the US Department of Defense *Predator*, have achieved this level of automation and are now being operated on a regular basis¹.

There is much research being conducted to achieve ever greater levels of “autonomy” in UAV operations, with the ultimate goal being large numbers of fully-autonomous UAV systems performing a variety of complex tasks given only very high-level mission objectives and supervision. Systems under development are both large and small, including representatives such as DARPA’s unmanned combat air vehicle (UCAV) and DARPA’s micro air vehicles. The large systems such as UCAV are being developed to carry sophisticated sensor and weapon systems great distances, and warrant the development of the software required to achieve full autonomy. This task is aided by the fact that these systems are designed for very specific and well-defined missions. However, great need exists for very simple low-cost systems to serve the small unit or even the individual soldier. In this case the applications will be quite varied (and thus are poorly defined at the outset), and sufficient resources will not be available to develop software tailored to each possible scenario. The payload for these systems is most often a simple imaging package for surveillance and reconnaissance tasks. Rotorcraft offer unique flight capabilities that make them especially well suited to this type of work in a very constrained or cluttered environment, and are also receiving a lot of attention. For example the U.S. Army has near-term plans to evaluate the use of unmanned rotorcraft in a variety of applications using several unmanned helicopter test-beds. Operations of interest include tasking and supervision of the unmanned rotorcraft by the crew of manned flight systems.

In this paper we consider the elements of a generic guidance and control system for an unmanned rotorcraft that can be configured at varying levels of sophistication to support a wide variety of potential tasks. We then identify a specific subset of tasks we would like to perform in the near term, and the associated guidance and control system requirements. The paper then proceeds with development of an adaptive velocity command system to meet these

* President, Member AIAA

† Professor, Fellow AIAA

‡ Professor, Member AIAA

†† Graduate Student

‡‡ Research Engineer

requirements. The velocity command system is driven by a relatively simple rule-based command generator. The resulting velocity command system is combined with a previously developed adaptive attitude command system, and then evaluated both in nonlinear simulation and flight test.

AUTOMOUS UAV ARCHITECTURE

Figure 1 presents a generic architecture for an autonomous UAV mission planning, mission management, path planning, guidance, and control system, along with associated elements for synthesizing what is often referred to as “situation awareness.” There are three basic functional groups depicted. The upper portion (shown in green) represents elements designed to fuse information and requirements into a plan of action. The middle blocks (shown in blue) are associated with path planning and guidance. The lower blocks (shown in grey) are for aircraft-specific control functions. Also shown in the figure (in yellow) are some of the vehicle dynamic and sensor models required to simulate operations. The components of this system and their interconnections are purely notional, and were not derived from any current specification or requirement.

At the very top is a block that represents a command and/or supervisor interface. This could represent the combination of a preplanned set of mission objectives and a traditional UAV operator. Alternately this block could include a natural language interpreter/translator and an associated intelligent agent to enable direct supervision of the UAV in an interactive manner by personnel whose primary responsibility is other than UAV operation (e.g. a crew member on a manned aircraft, or a soldier on the ground). In the case of a fully-autonomous UAV, operating either alone or cooperating with other autonomous assets in some fashion, this block may represent the interface to other hardware/software systems.

The next layer down is a high-level software component for in-flight mission planning and updating, the latter process being referred to as mission management. This block translates all of the inputs into a consistent set of objectives, requirements/constraints that are representative of the pre-mission plan and/or supervisor’s intent. Depending on the specifics of the system, data from external sources and on-board sensors is next fused and integrated into a current world model, which is combined with the interpreted mission objectives and constraints to produce some degree of “situation awareness.” The output of this process is a current plan for motion of the vehicle/sensor(s), including consideration of dynamic constraints. The motion requirements are then passed to one or more

guidance laws that employ the attitude command system to produce the commanded motion.

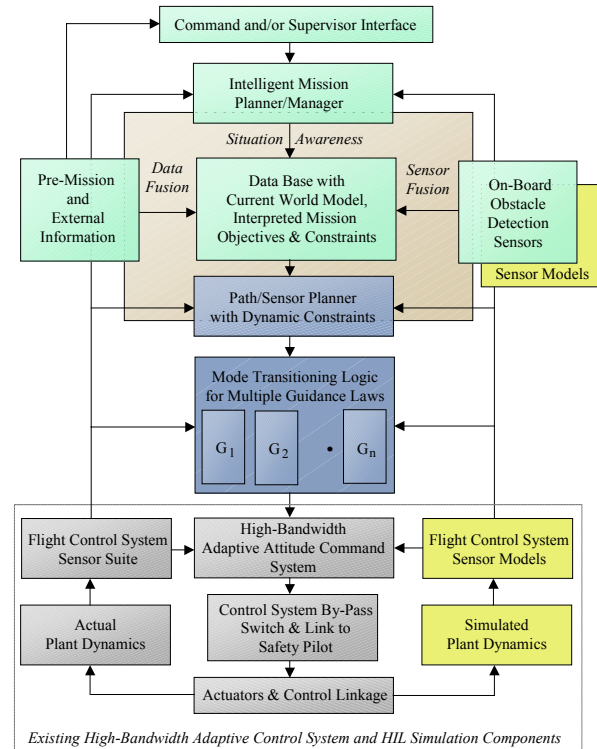


Figure 1 Notional UAV Architecture that supports varying levels of autonomy.

This architecture is not unlike that of the “Rotorcraft Pilot’s Associate” which was developed over a period of many years at a cost of many millions of dollars, and its full functionality is unlikely to be produced in the course of developing a low-cost unmanned helicopter testbed. Instead, we seek to implement and/or emulate the functions laid out in this architecture as appropriate to near term tasks assigned to the testbed. Specifically, we seek a system that will accept a set of preplanned waypoints along with a general set of operational guidelines and data (i.e. constraints, threat locations, terrain data, mission goals, behavior criteria), that will continuously and automatically generate a plan of action and flight path, and which can be redirected at any time either through interaction with the supervisor, other systems, or due to availability of new information (such as due to on-board sensors for obstacle detection). It was determined through experiment in simulation that a velocity command system driven by a simple rule base could perform well in this capacity, and it is the development of this velocity command system and

command generation rule base that is the subject of the remainder of this paper.

COMMAND GENERATION

Consider the requirement to continuously generate a prescribed flight path and heading that satisfies the requirements set forth in the previous section. One might choose to employ an optimal or near-optimal control solution given a path length of significant dimension and criteria or constraints that dictate solutions other than straight lines. However, for many candidate mission scenarios, a simple string of straight-line path segments is more than adequate. We will concentrate on this case. Now consider the requirement for a rotorcraft to hover at specific points in space or times, and to translate between these hover locations. Some form of acceleration and deceleration must be prescribed. There is also the question of the speed to be employed along each straight-line segment, the choice of vehicle heading, obstacle and terrain avoidance, and the management of payload sensors such as an imaging package. We also seek to implement a mode of operation in which a human supervisor or team mate is able to explore the environment by remote viewing of on-board sensor data and by giving high level commands to the vehicle for very simple tasks (e.g. stop, go forward slowly, move to the right 100 feet, go up). A rule base and associated logic is being evolved for this purpose, and is described in the following.

Consider first command of the horizontal components of velocity. At any given instant in time, let there be a point in 3-D space which defines the current goal of flight. The point, referred to as the current waypoint, is designated as either a “fly-through” point or a “hover” point. Unless and until the current waypoint has been achieved, the baseline velocity command is directed from the current position to the current waypoint. The current waypoint is achieved by satisfying criteria associated with that specific waypoint. This criteria can consist of any number of factors, the most common for hover being maintaining proximity to the waypoint for some duration of time, and for a fly-through waypoint simply passing within a prescribed distance from the waypoint. A list of waypoints defines the overall flight program, and the next waypoint on the list is selected once the criteria for the current waypoint are met.

Transition between hover and translation is managed by predefined acceleration and deceleration profiles that prescribe the magnitude of velocity command as a function of relative distance. The choice of maximum airspeed at any given time is governed by a rule base which takes into account such features as the distance to the next waypoint, and the overall goals of

the mission (i.e. proceed quickly, or proceed cautiously), and may also be dictated by the performance of payload sensors or sensors employed for obstacle detection. A similar rule base may be employed to determine the radius of curvature (i.e. rate of rotation of the horizontal component of the velocity vector) at the intersection of two straight-line path segments (i.e. at flight thru waypoint)

Consider next the generation of the heading command. Except in certain circumstances, the heading of the vehicle is prescribed to remain aligned with the commanded velocity vector. The baseline heading command is relieved of this requirement if (1) the destination waypoint is a hover waypoint and the distance to the destination waypoint is such that the magnitude of the current velocity command is below a given threshold. For example, if hovering, and a requirement arises to move a small distance (i.e. a distance too small to warrant buildup of significant airspeed), then there is no requirement to reorient the heading command. (2) the vehicle is hovering. At any time a perturbation of the heading command can be superimposed on the baseline heading command. When the perturbation is removed, the heading command returns to its baseline value which is slaved to the velocity vector. Also, after satisfying the criteria associated with a hover waypoint, one can specify a delay in which to complete a heading change before accelerating and proceeding to next waypoint. Options exist also to suspend slaving of the heading command to the velocity vector entirely and to use some other rule, like heading being fixed, or heading always being directed toward a target of interest. The altitude command is managed in a similar fashion, but with special emphasis on terrain avoidance.

In all cases, command and command rate limiting are employed to ensure both the stability and the “quality of behavior” of the overall guided system. These features of design are currently ensured through comprehensive simulation. Methods for design and analysis that guarantee stability and performance are of future research interest.

VELOCITY COMMAND SYSTEM

The authors have prepared a series of papers on the design and flight test of a neural network adaptive nonlinear attitude command system for unmanned helicopters^{2,3}. Recent improvements of this controller to allow for high-bandwidth design using adaptive output feedback are reported in references 4 and 5. The authors have also previously presented the design of a companion outer loop control system for tracking of position commands³. In the discussion that follows, we first present the design of an outer loop for tracking of a velocity command. An on-line neural

network is used to compensate a model-inverting controller for errors that arise due to the difference between a simple model of translational dynamics being used for inversion, and the actual plant dynamics. This outer loop generates attitude commands that are processed by an inner loop, which is discussed in the subsection that follows. Presentation of a coupled analysis of the inner and outer loop for gain selection in a manner similar to that reported in reference 6 completes this section of the paper.

Adaptive Outer Loop for Tracking Horizontal Velocity Commands

Figure 2 presents an overall block diagram of the velocity command system. The command generator is located at the far left. The features of this command generator were discussed in the previous section. The velocity commands are first filtered, and then passed to a linear compensator that acts on the velocity error formed by comparing the filtered command with a velocity measurement. The output of the compensator, the command acceleration generated by the command filter, and the output of a neural network are combined to form the pseudo control signal u . The pseudo control feeds a simple model of the helicopter translational dynamics that is inverted to obtain corresponding attitude commands. Pseudo control hedging of attitude command limits and the inner loop dynamics is used to protect the outer loop adaptive process from saturation and from adapting to the inner loop dynamics. Design of the inverting control law is discussed further in the following. Details of the outer loop neural network update law and pseudo control hedging are as given in references 2 and 3.

The relationship between the components of acceleration of the vehicle mass center and components of external forces acting on the vehicle can be expressed as

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = L_{vb}(\phi, \theta, \psi) \begin{bmatrix} F_x / m \\ F_y / m \\ F_z / m \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (1)$$

where X , Y and Z are the position components of the vehicle mass center in the earth-fixed (North, East, Down) coordinate system, L_{vb} is the transformation matrix from body axes to earth-fixed axes, ϕ , θ and ψ are the Euler roll, pitch and yaw attitudes, respectively, F_x , F_y and F_z are total aerodynamic force components along the body axes, m is vehicle mass and g is the constant of acceleration due to gravity. Following the trajectory controller synthesis described

in⁶, a set of pseudo-controls, U_1 , U_2 , and U_3 , are formulated as

$$U_1 = K_x(X_c - X) + K_{\dot{x}}(\dot{X}_c - \dot{X}) + \ddot{X}_c \quad (2)$$

$$U_2 = K_y(Y_c - Y) + K_{\dot{y}}(\dot{Y}_c - \dot{Y}) + \ddot{Y}_c \quad (3)$$

$$U_3 = K_z(Z_c - Z) + K_{\dot{z}}(\dot{Z}_c - \dot{Z}) + \ddot{Z}_c \quad (4)$$

where the subscript c denotes commanded value. Next, the left-hand side of Eq. (1) is replaced by the values of the pseudo-controls U_1 , U_2 and U_3 computed from Eqs. (2)–(4). This will result in a set of algebraic equations which can be rearranged as

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 - g \end{bmatrix} = L_{vb}(\phi, \theta, \psi) \begin{bmatrix} F_x / m \\ F_y / m \\ F_z / m \end{bmatrix} \quad (5)$$

Equation (5) can be used to compute the values of pitch and roll attitudes required for horizontal (both X and Y position, velocity and acceleration) command tracking. In the process, certain approximations are made in order to simplify the resulting computations. First, the magnitudes of cyclic and pedal control forces are assumed to be much smaller compared to the collective control force and hence, they are neglected. Second, the F_x and F_y (i.e., body x -axis and y -axis aerodynamic force components) are assumed to be small in magnitude compared to F_z (body z -axis force component) and hence, they are neglected in Eq. (5). With these approximations, closed-form expressions for the required pitch and roll attitudes can be obtained as

$$\bar{\theta} \approx \tan^{-1} \left(\frac{U_1 \cos \Psi_c + U_2 \sin \Psi_c}{U_3 - g} \right) + \theta_{trim} \quad (6)$$

$$\bar{\phi} \approx \sin^{-1} \left(\frac{-U_1 \sin \Psi_c + U_2 \cos \Psi_c}{\sqrt{U_1^2 + U_2^2 + (U_3 - g)^2}} \right) + \phi_{trim} \quad (7)$$

where $\bar{\phi}$ and $\bar{\theta}$ are the required roll and pitch attitudes, respectively, and Ψ_c is the commanded yaw attitude. The values of $\bar{\phi}$ and $\bar{\theta}$ computed using Eqs. (6) and (7) are used as command inputs to the inner loop attitude control system shown in Fig.2, which in turn will determine inputs to the cyclic control actuators.

Loops for Control of Altitude and Heading

An adaptive heading loop is employed as described in references 2 and 3. An altitude control loop is constructed as reported in reference 6.

Adaptive Inner Loop for Tracking of Pitch and Roll Attitude Commands

As noted in the introduction to this section, evolution and evaluation of the inner loop design has been documented in previous papers. An adaptive state-feedback design was presented in References 2 and 3, but was bandwidth-limited by the presence of unmodeled control rotor dynamics and system time delays. In Reference 7, the authors presented a direct output feedback approach to design of the attitude command system that is tolerant of unmodeled dynamics. The term “direct” in this context refers to an output feedback control architecture that avoids the use of a state observer. In references 4 and 5, an alternate output feedback architecture was reported that is based on *Observing the Tracking Error Dynamics*. The attitude command system of References 4 and 5, which is based on the error observer approach described above, was employed in this work.

Figure 3 presents a block diagram of this inner loop. In essence, a simple inverting controller is augmented with the output of a neural network. The network, a multilayer perceptron, employs a single hidden layer with only five nodes. Attitude commands are received on the left and passed through a third-order command filter. The filtered commands are employed in a linear compensator that operates on attitude and attitude rate errors. The output of the compensator, u_{pd} , the command acceleration, and the output of the neural network are combined to produce the pseudo control, u . A simple linear model of the attitude dynamics at hover is inverted to obtain the helicopter controls (main rotor longitudinal swashplate angle) as a function of the pseudo control. The control effector manager simply computes the action required of each of the three main rotor actuators to produce the desired swashplate tilt. Sensed attitude and attitude rates are then fed back as shown. The weights of the neural network are updated to produce an output signal, u_{ad} , that compensates for the difference between the simple dynamic model being inverted and the dynamics of the actual plant. This modeling uncertainty is represented in the diagram as Δ . Uniform ultimate boundedness of the error signals in the loop has been shown through Lyapunov-like stability analysis. Pseudo control hedging is used to enable correct adaptation even during periods of control saturation. Details regarding design of the error observer, design of the PD compensator, design of the hedge estimator, and the design of the adaptive law used to train the neural network weights, as well as the application of the method to attitude control of an unmanned helicopter are presented in References 4. Initial flight evaluation of this controller is discussed in Ref. 5..

Coupled Analysis of the Inner and Outer Loops for Gain Selection

Design of the inner and outer loops has thus far been discussed independently. This independence is achieved by ensuring adequate time scale separation between the loops. In this section a coupled analysis is presented for gain selection. Figure 2 provides an overview of the outer loop structure, where \ddot{x}_c and \dot{x}_c are generated by the first order command filter. The desired acceleration is \ddot{x}_{des} is identified in the figure as the outer loop pseudo control, u . The error dynamics of the outer loop are first order and can be written as follows:

$$\ddot{X}_{des} = \ddot{X}_c + R_d(\dot{X}_c - \dot{X}) \quad (15)$$

$$s^2 X_{des} = s^2 X_c + sR_d(X_c - X) \quad (16)$$

Coupling between the inner and outer loops is made via the outer loop inversion law. The outer loop inversion law block is labeled “Approximate Inverse Transformation” in Figure 2 and identified here as f_{ol}^{-1} .

$$\theta_{ol_c} \approx \arctan \frac{(v_1 \cos \psi_c + v_2 \sin \psi_c)}{v_3 - g} + \theta_{trim} \quad (17)$$

The pitch channel outer loop inversion law, equation (17) will be simplified to make the analysis easier. For constant altitude, U_3 will be approximately zero and for small angles θ , the arctan function, can be linearized with a straight line. If we also assume that the heading angle ψ is zero we get the following approximation of equation (17):

$$\theta_{ol_c} \approx \frac{\ddot{X}_{des}}{-g} = \frac{s^2 X_{des}}{-g} \quad (18)$$

Now equation (15) is substituted in equation (18) and the input command to the inner loop θ_{ol_c} can be written:

$$\theta_{ol_c}(s) = \frac{1}{-g} [s^2 X_c + sR_d(X_c - X)] \quad (12)$$

A diagram of the inner loop is presented in Figure 3. A proportional plus derivative controller is employed as the linear compensator, but with one shifted pole, p ,

introduced. The error dynamics of the inner loop are third order:

$$\theta = \frac{p}{s^2(s+p)}v$$

$$v = \frac{1}{p}(\ddot{\theta}_c + p\dot{\theta}_c) + K_d(\dot{\theta}_c - \dot{\theta}) + K_p(\theta_c - \theta) \quad (20)$$

$$\frac{s^3}{p}\theta + s^2\theta = \frac{s^3}{p}\theta_c + s^2\theta_c + sK_d(\theta_c - \theta) + K_p(\theta_c - \theta)$$

For this third order system with one shifted pole, p, the transfer function can be written in the form:

$$\theta(s) = \frac{\theta(s)}{\theta_c(s)}\theta_c(s) = \frac{pK_p}{s^3 + ps^2 + pK_d s + pK_p}\theta_c(s) \quad (21)$$

The inner loop input θ_{olc} is generated by the outer loop (see equation 19). For $\theta_{olc} = \theta_c$ equation (21) can be written:

$$\theta(s) = \frac{pK_p}{s^3 + ps^2 + pK_d s + pK_p} \frac{s^2 X_c + sR_d(X_c - X)}{-g} \quad (22)$$

The left side of equation (22) represents the real output of the inner loop. To feed this back to the outer loop an equation which is also derived from the inversion law is used:

$$\ddot{X} = s^2 X = -g\theta(s) \quad (23)$$

To get the transfer function for the whole system equation (23) is substituted in equation (22):

$$\frac{s^2 X}{-g} = \frac{pK_p}{s^3 + ps^2 + pK_d s + pK_p} \frac{s^2 X_c + sR_d(X_c - X)}{-g} \quad (24)$$

Equation (24) can be solved for X / X_c and represents now the transfer function for the coupled system:

$$\frac{X}{X_c} = \frac{pK_p s + pR_d K_p}{s^4 + ps^3 + pK_d s^2 + pK_p s + pR_d K_p} \quad (25)$$

Finding a generic system which is chosen to represent the desired characteristics, a fourth order system is set up as a product of a first and a third order system. For this system the natural frequency, damping ratio, ratio of inner loop to outer loop bandwidth, and time constant of the first order command filter, $\omega_i, \zeta_i a$ and τ , are prescribed:

$$\omega_i = 9.0 \text{ rad/sec} \quad (26)$$

$$\zeta_i = 0.8$$

The parameter a is chosen later as the only free parameter to characterize the system. To ensure time scale separation between the outer and the inner loop the inner loop is set to be three times faster than the outer loop. Therefore τ is defined as a function of ω_i and ζ_i :

$$\tau = \frac{3}{\zeta_i \omega_i} \quad (27)$$

For the fourth order system the characteristic polynomial can be expressed as:

$$X(s) = (\tau s + 1)(as + 1)(s^2 + 2\zeta_i \omega_i s + \omega_i^2)$$

$$= s^4 + \left(2\zeta_i \omega_i + \frac{\tau + a}{\tau a}\right)s^3 + \left(\omega_i^2 + \frac{2(\tau + a)\zeta_i \omega_i}{\tau s} + \frac{1}{\tau a}\right)s^2$$

$$+ \left(\frac{(\tau + a)\omega_i^2}{\tau a} + \frac{2\zeta_i \omega_i}{\tau a}\right)s + \frac{\omega_i^2}{\tau a} \quad (28)$$

By comparing the coefficients of equation (25) and equation (28) the gains and the shifted pole for the coupled system can be expressed in dependence of $\tau, \zeta_i \omega_i$ and a :

$$p = 2\zeta_i \omega_i + \frac{1}{a} + \frac{1}{\tau}$$

$$K_p = \frac{(r + a)\omega_i^2}{rap} + \frac{2\zeta_i \omega_i}{rap} \quad (29)$$

$$K_d = \frac{\omega_i^2 + 2(r + a)\zeta_i \omega_i}{rap} + \frac{1}{rap}$$

$$R_d = \frac{\omega_i}{(r + a)\omega_i + 2\zeta_i}$$

UNMANNED HELICOPTER RESEARCH FACILITY

A facility for simulating and flight testing advanced guidance and control algorithms on unmanned helicopters has been evolved jointly by the Georgia Institute of Technology and Guided Systems Technologies, Inc (GST).

Unmanned Helicopter Research Testbed

The helicopter testbed used in this work is based on the airframe of the Yamaha Model R50

production unmanned helicopter⁸. It features a roughly ten foot diameter rotor driven by a twelve horsepower liquid cooled engine, and has an approximate useful payload of forty-four pounds. The helicopter has been outfitted with a general-purpose aluminum enclosure between the landing gear that houses the experimental control system avionics and is pictured in Figure 6.



Figure 6. Joint Unmanned Helicopter Control Systems Testbed.

The flight vehicle is supported by a PC-based ground station. Communication between the flight control and ground station computers is maintained using a two-way digital radio link. This link is used to transmit GPS differential corrections, operator commands, and various control system configuration parameters to the vehicle, and to obtain from the vehicle data on system operation during flight. A safety pilot is employed at all times. The safety pilot is outfitted with an independent hand-held radio transmitter. With the flip of a switch, he can bypass the flight control computer and pilot the helicopter directly by line of sight¹⁰.

Flight Control System Implementation

The guidance and control system formulation presented in the previous sections has been coded in the C programming language. It runs in real-time in double precision with a nominal update rate of 100 Hertz on a 200 MHz Pentium-based single board computer. Select components of the high-bandwidth attitude command system (the neural network update laws, the filters at the inputs to the neural networks and the actuator dynamic models used for hedging) are updated at 200 Hz. The computer employs a custom real-time operating system. The sensor suite consists of a tactical-grade GPS-aided inertial navigation solution, augmented with a 3-axis magnetometer for heading initialization and ultrasonic ranging for precision geometric altitude measurement during take-off and landing⁸.

Helicopter Simulation Model

A nonlinear simulation model of the testbed helicopter has been developed for controller validation and software testing based on the math model given in

Reference 9. It includes a six degrees-of-freedom fuselage rigid body model, as well as a first-order representation of main rotor flapping and quasi-steady representations of main and tail rotor inflows. The simulation also includes the simple control rotor model developed in Reference 10, models of first order actuator dynamics, control linkages and limits, sensor noise, bias and latency, transport delays, and simple engine and rotor RPM dynamics. The model has been validated against flight test data for a few key parameters¹¹. It is simple enough to run in real-time on a typical desktop PC, and is coupled to the real-time flight control system to produce a hardware-in-the-loop simulation capability. Response of the actuators is presently simulated, not measured. Because adaptation is used to overcome model deficiencies, the utility of the real-time simulation is primarily software test and validation. All of the numerical results presented in the next section were generated on the flight system hardware using the software that has been prepared for flight evaluation. All results include actuator, rotor and sensor models, noise, system time delays, and gust inputs.

SIMULATION RESULTS

The flight system hardware and software described in the previous sections was employed to develop and study the performance of the rule-based command generator, and to evaluate performance of the coupled inner and outer loops.

Consider for example the impact of command generation logic and control system bandwidth on simulation of flight about a set of four waypoints in the horizontal plane as shown in Figure 1. To characterize the impact of design choices one can examine features such as the time required to complete one circuit around the square, or the amount of time flight exceeds a prescribed corridor about the straight-line path segments.

Figure 1 presents a plot of the waypoints and the shortest path connecting them (red circles and lines), a prescribed corridor of width 10 feet (gray area around the path) and the simulated flight trajectory (black line). The plotted trajectory was flown with a speed limit of 30 ft/sec. The overshoot occurring at each turn can be altered by modification of the command generation logic, but the logic remains fixed for the results presented herein.

Time test results are presented in Figure 2 as the inner loop bandwidth is varied. A fixed ratio of 5 between the inner to outer loops is maintained. Increasing bandwidth results in decreased times in both evaluation categories despite holding the maximum allowable speed constant.

Figure 3 presents a similar evaluation given a waypoint pattern selected to navigate through a field of

obstacles (circles). The simple command generation logic employed results in only one major overshoot. The associated time test results are shown in Figure 4, wherein the benefit of increased control system bandwidth is again evident when using time as a criteria. Similar improvement in time can be obtained by tuning the command generation logic. However, increasing the controller bandwidth enables one to obtain a high performance time result without having to specialize the command generation logic for the case at hand.

FLIGHT TEST RESULTS

Flight validation of the interactive command generator and velocity command system is on-going. Example flight test results of the response of the velocity command system are presented in Figure 7. Heading of the helicopter was commanded manually during this test case such that the longitudinal body axis of the helicopter is approximately aligned with the inertial x axis during most of the maneuver. Step commands for velocity in the inertial x direction were generated, while command of the inertial y velocity was held at zero. The time histories of the hedged inertial x-axis velocity command (blue trace) and measured inertial x-axis velocity response (green trace) are plotted in units of feet per second at the top of Figure 7. Hedging of the command was employed, though in this result adaptation is not employed in the outer loop. Excellent tracking of the command is evident. The design natural frequency of the adaptive inner loop for pitch attitude tracking is 6 radians per second. The time scale separation between the inner and outer loops is prescribed by a ratio of 5 (i.e. the outer loop design is 5 times slower than the inner loop, resulting in an outer loop design natural frequency of 1 rad/sec).

Tracking of the corresponding pitch attitude command is shown in the middle of Figure 7. The light blue trace is the hedged pitch attitude command (in degrees) generated by the outer loop, the dark blue trace is the corresponding pitch attitude measurement. Very good tracking of the pitch attitude command is evident from examination of the figure.

At the bottom of Figure 7 we present the values of several flags that have been set up in the software to log saturation of various control system limits. The flags for the three limits shown, from top to bottom are as follows

(1) “theta cmd limit” – (red trace) magnitude limit placed on the pitch attitude command generated by the outer loop. The unsaturated value of this flag was scaled to 4, and switches to 5 when saturation is present. This limit was set to 25 degrees for the test case shown. Note however, that this limit is applied to the command generated by the outer loop prior to hedging, and that the pitch attitude command presented

at the center of Figure 7 has been hedged. This is why saturation of the attitude command signal corresponding to those periods of time when the flag is set high is not visible on the plot.

(2) “actuator pos limited” – (blue trace) magnitude limit (travel) associated with the dynamic model of main rotor swashplate pitch-axis actuator used for hedging of the inner loop. The unsaturated value of the flag is scaled to 2 for plotting purposes, and switches to 3 when position saturation occurs. Note that no saturation of the actuator endpoints occurs during this maneuver.

(3) “actuator rate limited” – (purple trace) rate limit associated with the dynamic model of main rotor swashplate pitch-axis actuator used for hedging of the inner loop. The unsaturated value of the flag is scaled to 0 for plotting purposes, and switches to 1 when rate saturation occurs. Rate saturation is evident in the actuator model used for hedging each time the velocity command is stepped in one direction or the other.

SUMMARY AND CONCLUSIONS

A general architecture to support varying levels of autonomy in future unmanned air vehicles was postulated, and then reduced to a velocity command system driven by the combination of a sequential waypoint list, a variable-level supervisor interface, and a simple rule-based command generator. The proposed solution is adequate to support a wide variety of near term operational scenarios, including direction of the unmanned system by the crew of a piloted vehicle, and is realized without requiring a formal algorithm for on-board trajectory optimization. Design of an adaptive velocity command system was presented, and coupling with a previously developed adaptive attitude command system discussed. The capabilities and features of an existing unmanned helicopter research facility were briefly discussed. Finally the results of evaluation of the developed velocity command system in both simulation and flight test were presented. One of the demonstrated benefits of high-bandwidth control system design is the ability to obtain good tracking performance of a simply generated trajectory based only on straight line segments. Excellent tracking of the prescribed velocity commands is demonstrated in flight test using the high-bandwidth design despite significant periods of control saturation.

ACKNOWLEDGEMENTS

This work is supported in part by the U.S. Army Research Office under Contract NCC 2-945.

REFERENCES

- ¹"Predator Attrition is Continuing", World News Roundup, *Aviation Week and Space Technology*, p. 22, January 27, 2003.
- ²Calise, A. J., Prasad, J.V.R., and Corban, J. E., Flight Evaluation of an Adaptive Neural Network Flight Controller of an Uninhabited Helicopter. Paper No P6, *Twenty-Fifth European Rotorcraft Forum*, September 14-16, 1999, Rome, Italy.
- ³Prasad, J.V.R., Calise, A.J., Pei, Y. and Corban, J.E., "Adaptive Nonlinear Controller Synthesis and Flight Test Evaluation on an Unmanned Helicopter," Proceedings of the IEEE Conference on Control Applications, Hawaii, August 1999.
- ⁴Hovakimyan, N., N. Kim, and A. J. Calise, "Adaptive Output Feedback for High-Bandwidth Control of an Unmanned Helicopter," *AIAA Guidance, Navigation, and Control Conference*, 2001.
- ⁵Corban, J.E., et.al., "Flight Evaluation of Adaptive High-Bandwidth Control Methods for Unmanned Helicopters," *AIAA Guidance, Navigation, and Control Conference*, 2002.
- ⁶Prasad, J.V.R. and Lipp, A.M., "Synthesis of a Helicopter Nonlinear Flight Controller Using Approximate Model Inversion", *Mathl. Comput. Modelling* Vol. 18, No.3-4, pp. 89-100, 1993.
- ⁷Calise, A.J., Lee, Hungu, and Kim, Nakwan, "High Bandwidth Adaptive Flight Control," *AIAA Guidance, Navigation, and Control Conference*, 2000.
- ⁸J.E. Corban, A.J. Calise, and J.V.R. Prasad, "Implementation of Adaptive Nonlinear Control for Flight Test on an Unmanned Helicopter", Proceedings on the 37th IEEE Conference on Decision & Control, 1998.
- ⁹Heffley, R. K., M. A. Mnich, "Minimum-Complexity Helicopter Simulation Model Math Model, NASA CR-177476, USAAVSCOM Technical Report 87-A-7, April 1988.
- ¹⁰Perhinschi, M. and Prasad, J.V.R., "A Simulation Model of an Autonomous Helicopter," Proceedings of the RPV/UAV Systems 13th Digital International Conference, April 1998.
- ¹¹Munzinger, Christian, "Development of a Real-Time Flight Simulator for and Experimental Model Helicopter," Diploma Thesis, December 1998, Georgia Institute of Technology/University of Stuttgart.
- ¹²Suresh K. Kannan and Eric N. Johnson, "Adaptive trajectory based flight control for autonomous helicopters," *AIAA Digital Avionics Conference*, number 358, Irvine, CA, Oct 2002.

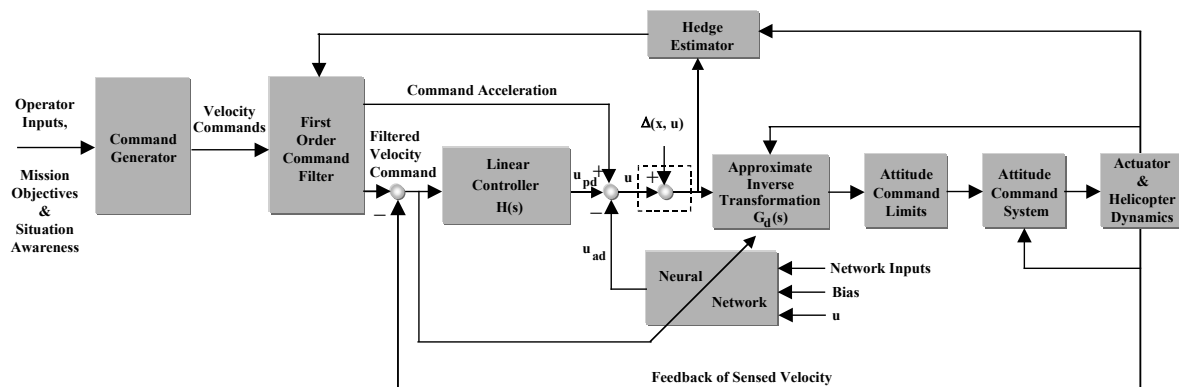


Figure 2 Block Diagram of Velocity Command System

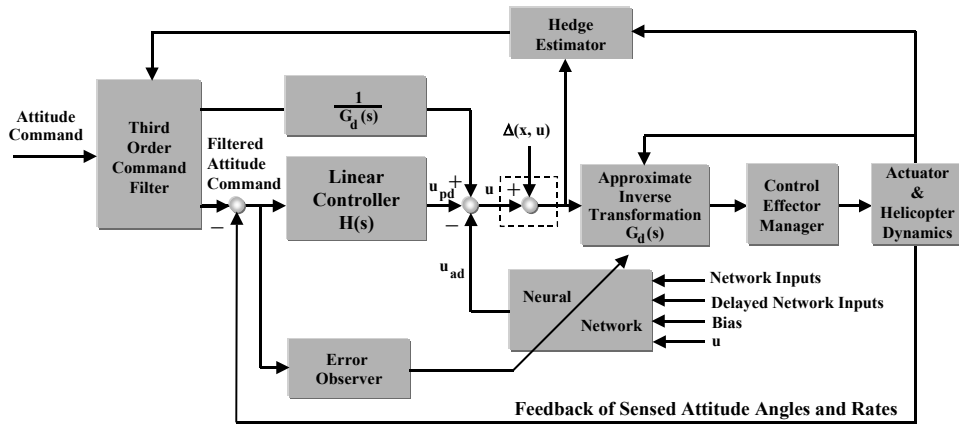


Figure 3 Block diagram of the neural network adaptive high-bandwidth nonlinear attitude command system.

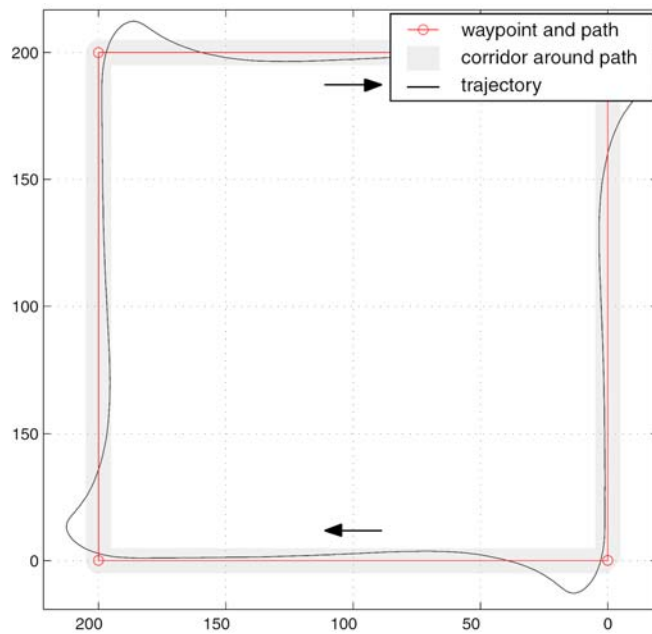


Figure 1 Square waypoint pattern evaluated in simulation

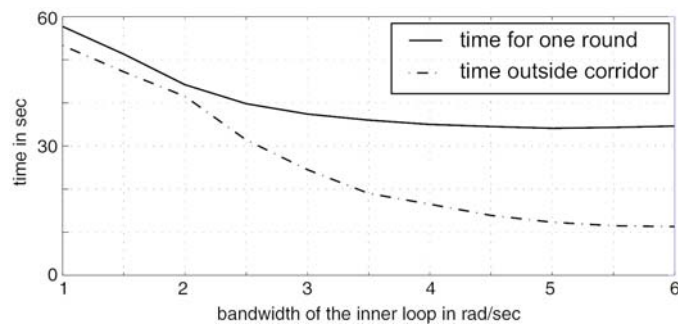


Figure 2 Time metrics for square pattern as inner loop bandwidths is varied

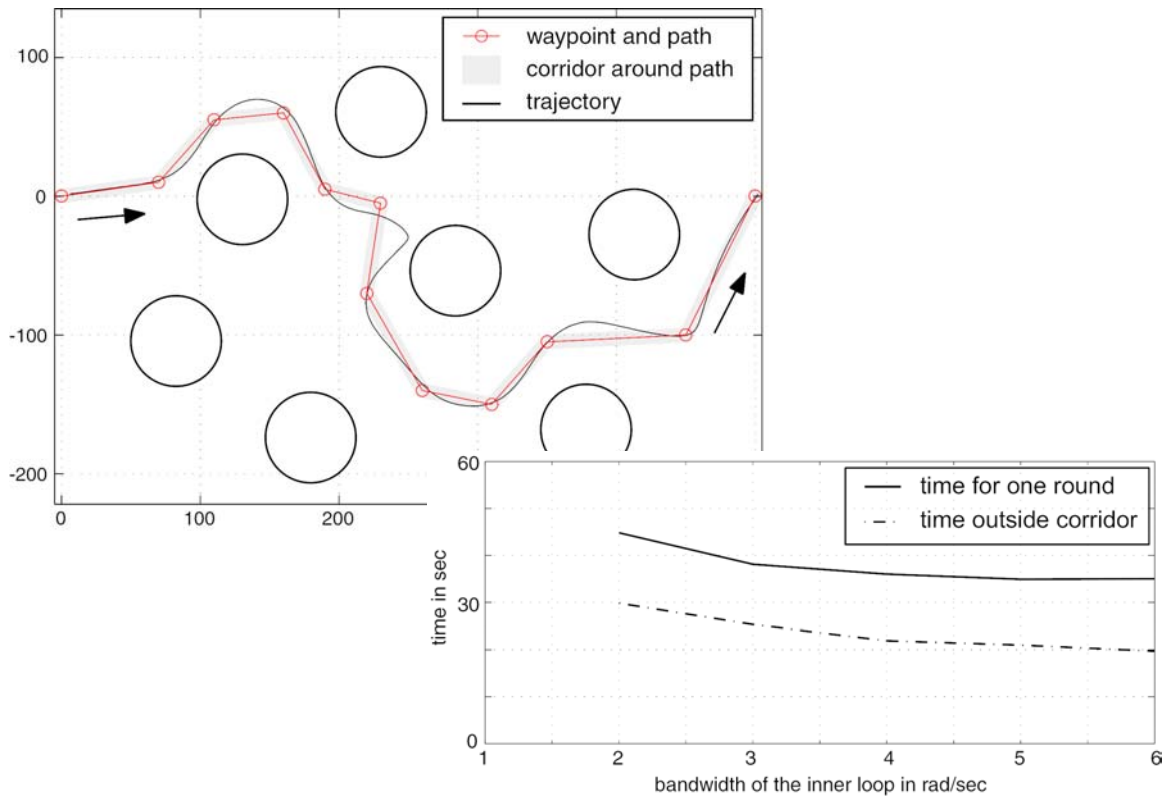


Figure 4 Navigating an obstacle field and associated time metrics as a function of controller bandwidth

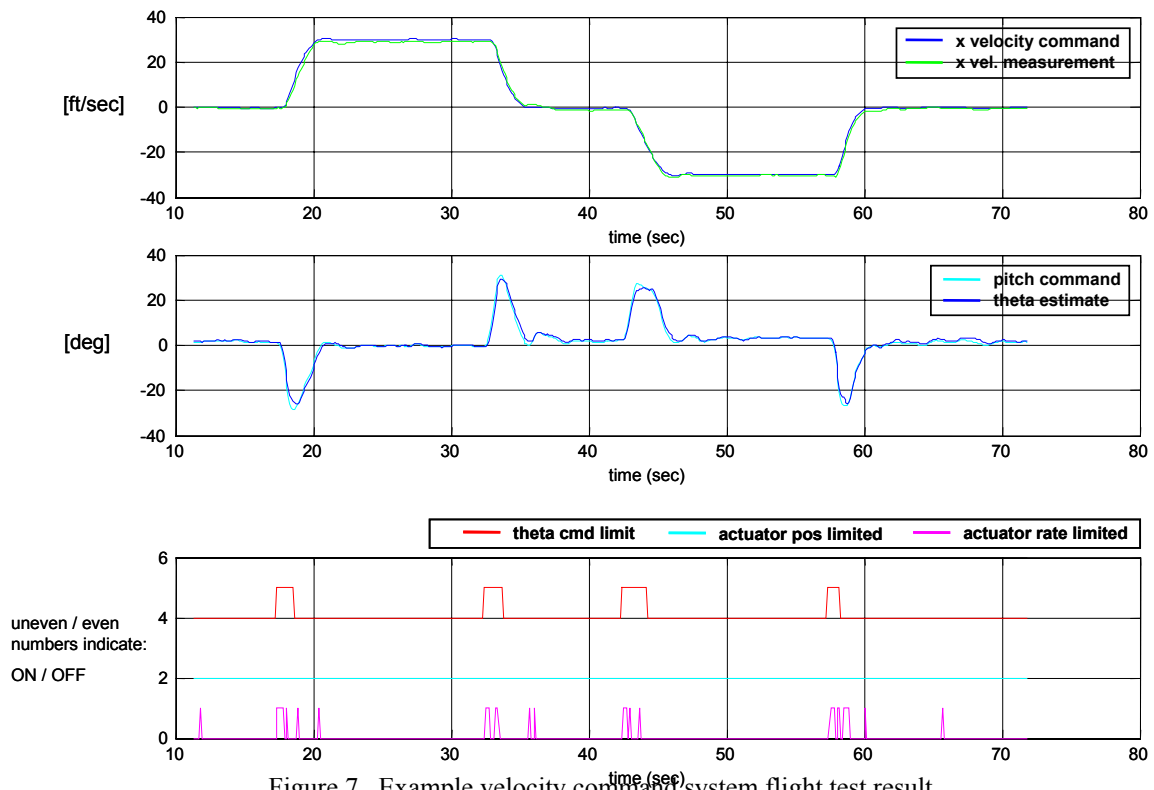


Figure 7. Example velocity command system flight test result.