JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 17, No. 6, November–December 1994

Automatic Guidance and Control for Helicopter Obstacle Avoidance

Victor H. L. Cheng*
NASA Ames Research Center, Moffett Field, California 94035-1000
and
T. Lam†
Sterling Software, Palo Alto, California 94303

The helicopter nap-of-the-earth flight problem has previously been discussed in various papers. The concept of automatic guidance involving obstacle avoidance requires a sophisticated obstacle detection system to provide three-dimensional obstacle and terrain data in flight in a hostile and unknown environment. Passive imaging sensors, augmented by selective use of low-detectability active sensors, will likely be needed to maximize covertness and safety, thus necessitating data fusion. The sensor data, limited by intervisibility constraints of the environment, suggest the use of heuristic arguments in flight-path planning over conventional analytic techniques. In addition, explicit consideration of vehicle capability is essential in the autopilot design to assure safe flight in such close proximity to the ground. This paper describes the automatic obstacle avoidance guidance and control functions and the implementation of these functions and a mock obstacle detection system in a graphical simulation for evaluation.

I. Introduction

IN order to minimize enemy fire in high-threat battle environments, military rotorcraft execute nap-of-the-earth (NOE) flight to seek masking with surrounding objects. This flight mode tends to be ground hugging and below tree tops, with predominantly lateral maneuvers around objects to achieve the masking, hence the characteristics of varying airspeed and heading. The complexity of rotorcraft missions involving NOE flight results in high pilot work load. This is especially true for single-pilot vehicles, such as the case originally intended for the RAH-66 Comanche. The Automated NOE Flight Program was initiated at NASA Ames Research Center to establish the feasibility and development of requisite technologies for automating NOE flight to reduce pilot work load.

The Automated NOE Program has identified obstacle detection and avoidance as the two functions requiring further research and development.[1] Obstacle detection techniques using passive imaging sensors are being pursued most vigorously by the Automated NOE Flight Program due to the desirable attributes of covertness and wide field of view.

The problem of obstacle avoidance was first investigated in Ref. 2 in a two-dimensional (2D) framework representative of the lateral maneuvers typical in NOE flight. The kind of NOE flight considered in this research for fully automatic flight involves the vehicle moving generally forward along some given trajectory. This part of the development effort excludes the stage of enemy engagement, which involves mostly dash/stop and bob-up/down maneuvers. This exclusion results from the premise that the guidance and control decisions at such critical juncture involve situation-dependent tactics and an element of pilot preference, which cannot be precisely captured in an automation system. Issues encountered when the 2D techniques are extended to the full 3D case were discussed in Ref. 3. Reference 3 also identified certain anticipated limitations of obstacle detection techniques based solely on passive sensors and recommended selective use of low-detectability active sensors such as laser range finders to augment the passive sensors. The block diagram in Fig. 1 illustrates the feedback structure involving the obstacle detection and obstacle avoidance guidance components and selective use of the guidance command to drive the active sensor. The use of multiple sensors necessitates the inclusion of a sensor-data-fusion function in the overall system.

The purpose of this paper is to report on the latest development in the obstacle avoidance effort to consummate the first full-function automatic obstacle avoidance guidance and control implementation in a simulation for the full 3D case. In view of the preliminary guidance concept previously explored in Refs. 2 and 3 and the adoption of an existing autopilot design,[4,5] modifications and additions to complete the guidance and control system can be summarized as follows, with details explained in Sec. II: 1) extension of previous 2D guidance concept in Ref. 2 to a fully automatic 3D concept via definition of an inertial database to store range measurements or estimates from a forward-looking sensor and a scheme to extract obstacle information from the inertial database for making lateral guidance decisions, augmentation of the guidance logic to allow hover-turn and climb above obstacles, and examination of the inertial database to generate an altitude profile along the selected ground track for altitude tracking and 2) modification of the autopilot design in Refs. 4 and 5 to improve safety in NOE flight via determination of acceleration command to emphasize 3D trajectory tracking due to the close proximity to the ground and objects by compromising 4D tracking, which also emphasizes time, and modification of the dynamic inverse model of Ref. 4 to properly resolve multiple solutions. Section II contains a description of all the individual modules of the guidance and control laws in the same order as they appear in the guidance and control structure. Although new modules will be discussed in some detail, modules that have been explained in previous publications will be included solely for completeness and their description will be brief. This complete guidance and control system is implemented in the graphical simulation described in Sec. III for evaluation. Besides the usual simulation components, this simulation includes the computer implementation of a mock obstacle detection system needed for realistic evaluation of the guidance and control modules. Section IV illustrates the performance with some examples and results, and Sec. V provides some concluding remarks.

II. Automatic Guidance and Control

When compared to the guidance and control of robotic vehicles, the automatic guidance and control for obstacle avoidance in NOE flight may differ significantly in requirements and capabilities be-

Received October 31, 1992; revision received May 20, 1994; accepted for publication May 25, 1994.

*Senior Research Scientist. Associate Fellow AIAA.
†Member of Technical Staff.

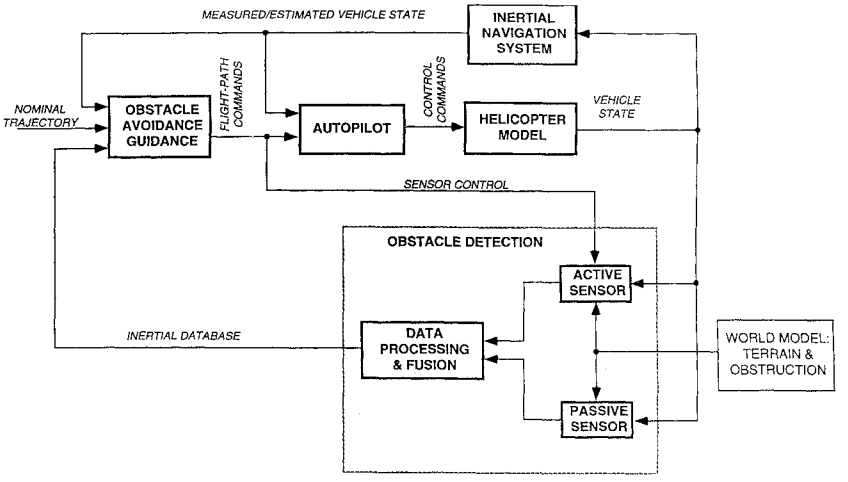

Fig. 1 Automatic obstacle avoidance guidance structure.

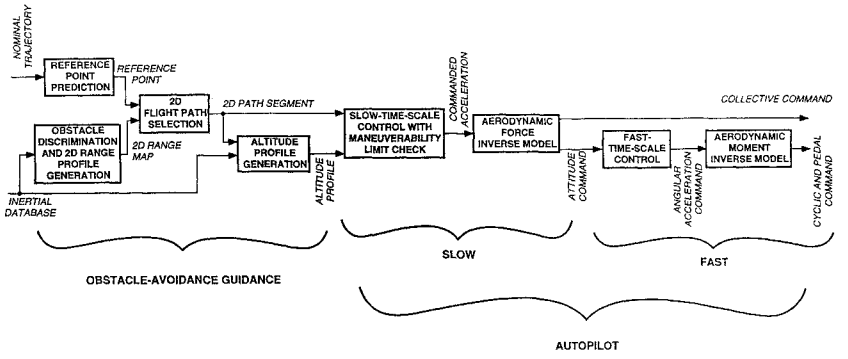

Fig. 2 Automatic obstacle avoidance guidance and control subsystems (feedback of state assumed but omitted).

cause the helicopters are manned. Certain requirements may need to be more stringent to provide a wider safety margin and a more comfortable ride. Other requirements may be slightly relaxed since the automatic guidance no longer needs to be able to handle every situation, as some situations can be delegated back to the pilot: for instance, when the automatic system concludes that it is stuck in a dead end with no reasonable way of moving forward without resorting to climbing above the obstacles, it can bring the helicopter to a hover while alerting the pilot of the situation. Of course, a lack of response from the pilot may have to be construed as the pilot being incapacitated, in which case fully automatic flight becomes necessary. A complete return-to-base capability will involve automatic switching between different flight modes including NOE, contour, low level, cruise, and landing, based on the on-board information on enemy and terrain threats. Our initial treatment is to develop a fully automatic guidance system for NOE flight. Issues related to human interaction with the automatic guidance will not be addressed in this paper.

The current implementation of the automatic guidance and control functions assumes that obstacle data are accumulated by the sensor and stored in the form of an inertial database. The objective of the guidance function is to determine an NOE path segment around the obstacles. The autopilot will take the path-segment information, including altitude profile, to generate the required stick control commands: pitch and roll cyclic, collective, and directional. Roughly speaking, the pitch, roll cyclic, and directional controls in helicopters cause rotation about the lateral, longitudinal, and vertical axes, respectively, and the collective causes changes in the main-rotor thrust.

The modules involved in these functions are depicted in Fig. 2. The feedback of vehicle state estimates to the modules is assumed and omitted from the diagram.

A. Obstacle-Avoidance Guidance

The input to the obstacle avoidance guidance subsystem includes a nominal trajectory provided by a higher level guidance function and an inertial database. The function of this system is to follow the nominal trajectory, to extract obstacle information from the inertial database, and to deviate from the nominal trajectory if needed to avoid detected obstacles. The guidance function consists of the first four blocks in Fig. 2. These four modules are described in the following. Only the first and third of these modules originated from the 2D guidance scheme of Ref. 2.

Reference Point Prediction

Tracking of the nominal trajectory is accomplished by continuously predicting a "reference point" on the nominal trajectory and tracking this reference point. The nominal trajectory and the reference point are both defined in the horizontal plane. The nominal trajectory is provided by a higher level guidance function over a long look-ahead time interval to perform terrain masking based on prestored terrain profile data.[1] It is generally a curved trajectory determined using a reduced helicopter dynamic model.[6] For our current evaluation, however, it is piecewise linear in the absence of integration with the higher level guidance. Prediction of the reference point is a function of the helicopter's position and velocity. Under normal situations, the reference point is computed on the nominal trajectory at a certain distance away from the helicopter, corresponding to the distance traveled at a constant speed for a period of a few seconds.[2] If, however, the helicopter is too far away from the nominal trajectory, the reference point may lie somewhere between the helicopter and the nominal trajectory so as to lead the helicopter back to the nominal trajectory.

Two-Dimensional Range Profile Generation

In parallel with the reference point prediction, the inertial database is scanned to identify the obstacles for the computation of a 2D range map. The 2D range map contains range data to obstacles as a function of azimuthal look angle.[2] It allows the guidance algorithm to identify a 2D flight path among the obstacles. To extract the 2D range map, the minimum range to obstacles is determined for each

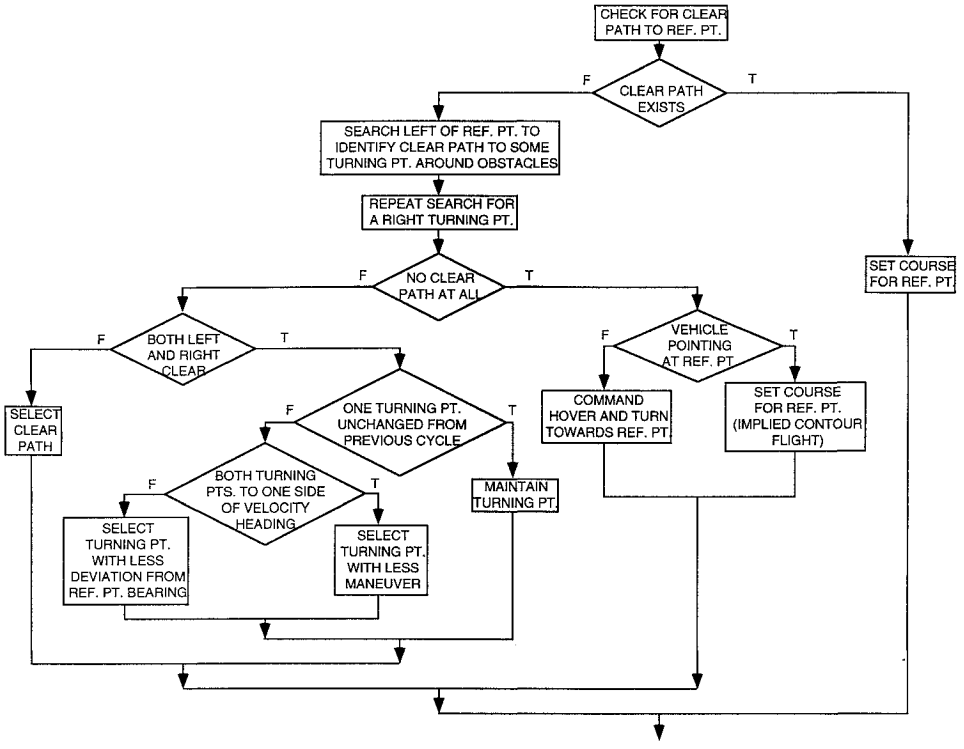

Fig. 3 Flowchart for 2D flight-path selection.

azimuth angle in a discretized set defined over a sector about the heading of the vehicle. For any given azimuth bearing, the technique described in Ref. 3 would scan the range data, which are measured as a function of elevation angle, for a sufficiently large local slope to indicate the existence of an obstacle. In the current approach where the inertial database stores altitude as a function of (x, y) location, however, the local-slope criterion reduces to one that detects a sufficiently large increase in altitude along the given azimuth bearing.

The sector comprising the 2D range map can be wider than the field of view of the sensor because the inertial database, being updated continuously, would contain data from previous sensor inputs and hence would still have useful information outside the sensor's current field of view.

Two-Dimensional Flight-Path Selection

When both the reference point and the 2D range map are available, the guidance proceeds to select a 2D flight path based on the concept studied in Ref. 2. The flow chart in Fig. 3 describes in detail the logic involved at every guidance cycle. The guidance begins by checking for a clear path directly to the reference point. If the path exists, it would be selected. Here a "clear path" takes into account the physical dimensions of the helicopter and additional clearance for safety margins.

If the path to the reference point is blocked, the guidance starts searching first to the left for a clear path to some turning point where the vehicle has the potential to go around some obstacle. The turning points are identified on the 2D range map as azimuthal bearings with sufficiently large range discontinuities. This search is followed by a similar search for a clear path to the right of the reference point. The search for these paths are constrained to an envelope about the vehicle's centerline, subject to some maximum deviation from the reference point bearing to prevent the vehicle from wandering too far off the nominal trajectory.

If neither path can be located, the guidance commands a hover-and-turn maneuver toward the reference point bearing. If there is still no clear path when the vehicle is pointing directly at the reference point, it simply commands a path directly to the reference point. Under this situation, since the path to the reference point is obstructed and the guidance would automatically track the altitude profile as described below, the guidance is effectively commanding a switch to contour flight.[1] This logic to switch to contour flight was not a part of the 2D guidance concept studied in Ref. 2 but is included in the current 3D scheme for the sake of completeness in a fully automatic system. For the case where a pilot is available, it would be preferable as discussed above to have the automatic system alert the pilot during the hover-turn of potential blockage and wait for the pilot's response before assuming contour flight.

If one, and just one, clear path exists to a turning point, that path is selected. When both left and right paths exist, the guidance logic would first test if any of these turning points coincides with the path selected in the previous guidance cycle. It would attempt to continue on this path if that is the case; otherwise, it would test if the current ground-track angle (i.e., velocity heading) falls in between the two paths. If it does, the path with less deviation from the reference point bearing is selected; otherwise, the path with less deviation from the current ground-track angle is selected.

The guidance was designed to prefer contour flight in difficult spots over tenacious search for open paths as in the maze problem. This is more appropriate for helicopter NOE flight than maze search techniques due to time and fuel constraints inherent in such flight mode.

Altitude Profile Generation

The last module in the obstacle avoidance guidance subsystem examines the inertial database to generate an altitude profile along the selected 2D path. The current technique is to command the vehicle to track the terrain altitude plus a preset clearance altitude, with a commanded altitude rate sufficiently large to clear everything along the selected path (Fig. 4a). The commanded altitude and altitude rate are subject to a maximum-altitude-rate constraint applied to the terrain profile defined over a distance corresponding to a few seconds of flight over the 2D path. As illustrated in Fig. 4b, the commanded altitude is allowed to deviate from the terrain-plus-clearance altitude if the look-ahead altitude profile slope exceeds the flight-path angle associated with the maximum altitude rate. Possible future improvements may involve a curved 2D path for smoother flight as well as more sophisticated altitude-tracking techniques[7] if proven necessary by the evaluation.

B. Autopilot

The autopilot function in helicopter NOE flight is substantially more complicated than other autopilot designs, which generally

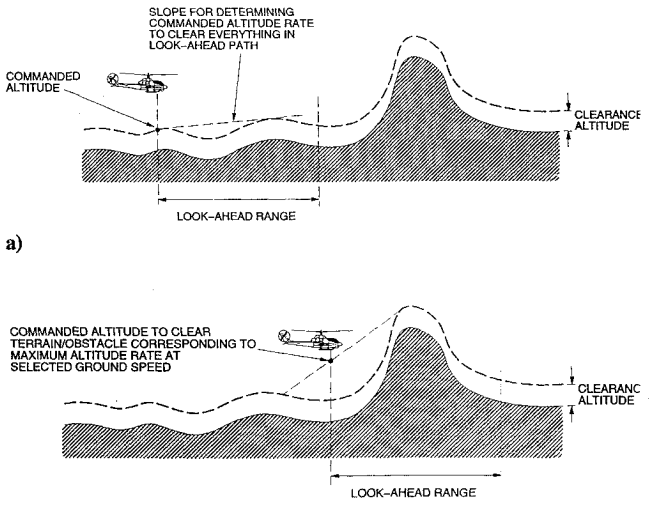

Fig. 4 Commanded altitude and altitude rate: a) determination of commanded altitude and altitude rate and b) effect of maximum altitude on commanded altitude.

need to maintain constant altitude, heading, and airspeed. In NOE flight, the altitude, heading, airspeed, as well as flight-path angle all vary interdependently to safely track the ever-changing flight path commanded by the guidance subsystem. It controls the helicopter through the usual commands: pitch and roll cyclic, collective, and directional. Our autopilot design follows closely the one presented in Refs. 4 and 5. It is a two-time-scale controller with the outer, slower loop modeled by the positional state for controlling position and velocity and the inner, faster loop modeled by the angular state for controlling attitude and angular velocity. The controller for each of these loops consists of an inverse model[8] to cancel the aerodynamic and kinematic nonlinearities and a linear feedback control law to control the resulting linear time-invariant model. The two loops, each with the inverse model and linear control law, result in the four modules depicted in Fig. 2 comprising the autopilot.

Slow-Time-Scale Controller

Based on the assumption that the inverse model and the fast-time-scale loop described below will be adequate in tracking reasonably well any commanded acceleration within the vehicle's maneuverability envelope, the resulting system can be studied as a linear time-invariant one with vehicle position and velocity as the state variables. Consequently, design of the controller for generating the commanded acceleration can be easily accomplished with simple state-feedback techniques. Validity of the assumption would certainly affect the performance of the autopilot. This assumption is reasonable in most aircraft applications and has been verified in the simulation described in Sec. III below. In applications where the assumption appears less reasonable, robust controller designs instead of simple state feedback should be considered to improve performance.

The remaining issue is to constrain the desirable acceleration within the maneuverability envelope. Indeed, examples in Ref. 5 have shown that control laws for tracking position and velocity as functions of time may call for maneuvers requiring bank angles close to 90 deg when the position or velocity error is large. Such unrealistic maneuvers would easily cause abrupt altitude loss, which is extremely dangerous in NOE flight. The remedy adopted by Ref. 5 is to smooth the commanded trajectory with low-pass filters so that the commanded maneuvers are realizable. Another straightforward remedy is to subject the maneuvers to predefined limits. Either of these techniques may keep the maneuvers within the vehicle's capabilities, but they may also slow down the maneuvers and affect the trajectory-tracking performance.

We believe that in NOE flight it is more important to track the trajectory in 3D space than to precisely track the trajectory defined as a function of time (i.e., 4D trajectory). To this end, we apply the constraints of the maneuverability envelope by down-playing time in the command: If the commanded maneuver exceeds the predefined limits, we seek another value within the limits by adjusting the commanded position or velocity along the desired trajectory. With adjustments of this sort, if the original commanded acceleration has an exceedingly large horizontal component due to a large change in commanded heading, the adjusted command would dedicate most of the control authority to slowing down the vehicle in the direction of the current heading and put less emphasis in accelerating toward the desired heading. (Remark: An even better solution is to have an integrated guidance/controller design where the performance capabilities are built into the guidance logic, and the necessary maneuvers will be determined directly and guaranteed to be achievable. This integrated approach, however, is not pursued in this paper.)

Let the inertial position, velocity, and acceleration be denoted by the variables r, v, and a, respectively, and let the subscripts H denote the two horizontal components and z the vertical component in the downward sense. With such conventions, let $\Delta r_H \in \mathbb{R}^2$ be the relative horizontal position defining the end of the 2D path segment commanded by the guidance. The end of the 2D path segment corresponds to either the reference point or a turning point, as described above and in Fig. 3. For safety reasons, the predefined nominal speed may be reduced according to some safe speed limit depending on the situation: The 2D range map is re-examined beyond the path segment to determine how much more room is available in that direction before an obstacle is encountered; this additional margin establishes a safe speed limit, which in turn constrains the desired speed $V_d \in \mathbb{R}_+$. The desired horizontal velocity is defined by

$$v_{Hd} := V_d \frac{\Delta r_H}{\|\Delta r_H\|} \tag{1}$$

The notation := indicates a definition or assignment and is different from =, which defines an equation. With $\gamma_d \in (-90 \text{ deg}, 90 \text{ deg})$ denoting the desired flight-path angle calculated with the altitude profile, the desired altitude rate is

$$v_{zd} := -V_d \tan \gamma_d \tag{2}$$

The desired horizontal acceleration is

$$a_{Hd} := \frac{1}{\tau_v}(v_{Hd} - v_H) \tag{3}$$

where τ_v is a time constant representing the horizontal velocity control loop bandwidth.

Let r_{zd} be the desired vertical position calculated with the altitude profile (Fig. 4). The desired vertical acceleration, including that needed to overcome gravity, is

$$a_{zd} := k_r(r_{zd} - r_z) + k_v(v_{zd} - v_z) - g \tag{4}$$

where g is the acceleration due to gravity and k_r and k_v are two gains defined by the control law. The commanded acceleration

$$a_c = \begin{bmatrix} a_{Hc} \\ -- \\ a_{zc} \end{bmatrix} \tag{5}$$

is generally defined by the desired acceleration a_d with components given in Eqs. (3) and (4).

To assure that the commanded acceleration is realizable, we bound the magnitude of a_c by some limits $A_{\min}$ and $A_{\max}$ and the angle made by a_c with the vertical by some limit $\varphi_{\max}$, with the notion that most of a_c is generated by the main-rotor thrust. For NOE flight, the helicopter is not likely allowed to flip over, so $\varphi_{\max} < \pi/2$. For a_c defined in Eq. (5), the two conditions are summarized as follows:

1. For some $A_{\min}$, $A_{\max}$ satisfying $0 \le A_{\min} \le A_{\max}$,

$$A_{\min} \le \|a_c\| \le A_{\max} \tag{6}$$

2. For some $\varphi_{\max} \in [0, \pi/2)$,

$$0 \le \varphi_c \le \varphi_{\max} \tag{7}$$

where

$$\varphi_c := \tan^{-1} \frac{\|a_{Hc}\|}{-a_{zc}} \tag{8}$$

Our procedure achieves the above conditions by first assuring that the vertical component a_{zc} would have a high priority not to be adjusted to increase in the downward sense, due to the close proximity to the ground. Then it adjusts the horizontal velocity a_{Hc} through adjustment of the magnitude of the desired horizontal velocity: viz, it tries to determine some $k \ge 0$ such that a new choice for commanded horizontal acceleration given by

$$a_{Hc} := \frac{1}{\tau_v}(k v_{Hd} - v_H) \tag{9}$$

would satisfy conditions 1 and 2. If determination of such a k is impossible, it would first find a k that would minimize the violation of the conditions and then subject the resulting acceleration to hard limits. The procedure is based on the following formulation: Since conditions 1 and 2 imply that the helicopter is not allowed to flip over and the main rotor is not allowed to exert negative thrust, it follows that $a_{zc} < 0$. Under this condition, the upper bound of condition (6) implies

$$-a_{zc} \le A_{\max} \tag{10}$$

and so this condition will be imposed on a_{zc}. Furthermore, the lower bound of condition (6) can be easily satisfied by requiring

$$A_{\min} \le -a_{zc} \tag{11}$$

Although condition (11) is not a necessary condition for (6), it is not a severe compromise in the envelope of a_c because $A_{\min}$ is supposed to be a very small quantity close to zero. Under these conditions, the upper limits of conditions (6) and (7) can be satisfied by limiting the magnitude of the commanded horizontal acceleration

$$\|a_{Hc}\| \le \min\left\{\sqrt{A_{\max}^2 - a_{zc}^2}, (-a_{zc}) \tan \varphi_{\max}\right\} \tag{12}$$

In view of Eq. (9), the condition of (12) is equivalent to requiring

$$\|k v_{Hd} - v_H\| \le V_{H,\max} := \tau_v \min\left\{\sqrt{A_{\max}^2 - a_{zc}^2}, (-a_{zc}) \tan \varphi_{\max}\right\} \tag{13}$$

Here $V_{H,\max}$ represents the maximum magnitude of the allowable change in horizontal velocity regardless of velocity-direction change, subject to the commanded vertical acceleration a_{zc} and the constraints imposed by $A_{\max}$ and $\varphi_{\max}$. When such a $k \ge 0$ exists to satisfy the inequality in (13), we pick the k that is closest to the

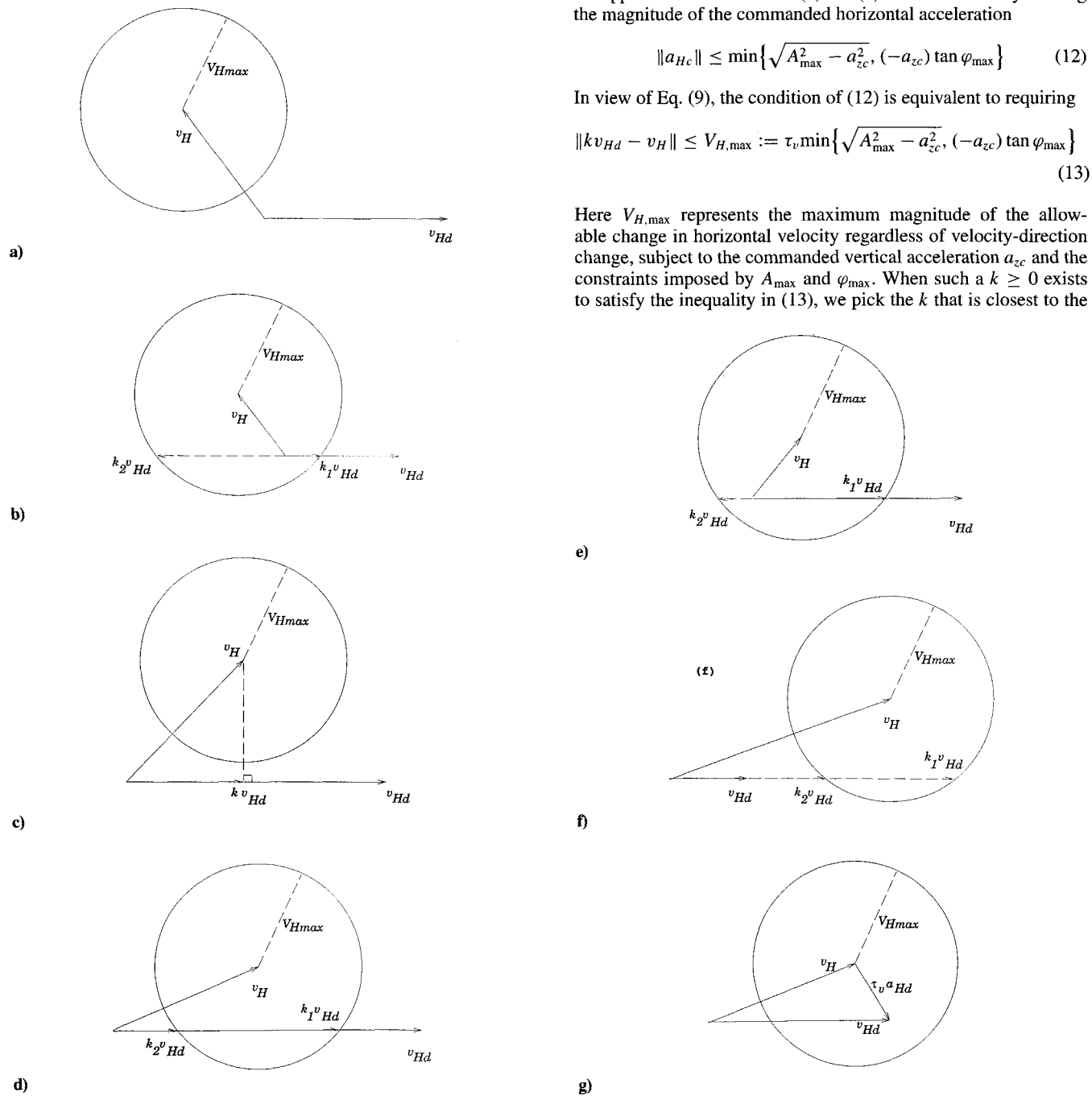

Fig. 5 Scaling desired horizontal velocity to adjust commanded horizontal acceleration.

originally desired value of 1. This solution of k occurs when the above inequality reduces to the equality

$$\|kv_{Hd} - v_H\| = V_{H,\max} \tag{14}$$

By completing squares, the solution occurs at

$$k = \frac{(v_{Hd} \cdot v_H) \pm \Delta}{\|v_{Hd}\|^2} \tag{15}$$

where

$$\Delta = \sqrt{(v_{Hd} \cdot v_H)^2 - \|v_{Hd}\|^2\left(\|v_H\|^2 - V_{H,\max}^2\right)} \tag{16}$$

and there may be two possible roots. Let these be denoted by k_1 and k_2, with $k_1 \geq k_2$. The various possible choices of k are explained in the following cases and their geometric interpretations are illustrated in Fig. 5.

Case 1: $v_{Hd} \cdot v_H < 0$ (i.e., change in ground-track angle more than 90 deg)

If (1) $\|v_H\| > V_{H,\max}$, it is obvious that, for any $k \geq 0$,

$$\|kv_{Hd} - v_H\|^2 = k^2\|v_{Hd}\|^2 - 2k(v_{HD} \cdot v_H) + \|v_H\|^2$$

$$\geq \|v_H\|^2 > V_{H,\max}^2.$$

i.e., no $k \geq 0$ can satisfy Eq. (13), and minimum violation is achieved with $k = 0$ (see Fig. 5a); hence Eq. (9) is reduced to

$$a_{Hc} := -\frac{1}{\tau_v} v_H$$

(i.e., use maximum available horizontal acceleration to decelerate along the current ground track until enough excess acceleration capability is available to modify ground track).

Else (2) for the two roots in Eq. (15), $k_2 \leq 0 \leq k_1 < 1$, and so the choice is (Fig. 5b)

$$k := k_1$$

Case 2: $v_{Hd} \cdot v_H \geq 0$

If (1) $\Delta^2 < 0$, no real root exists for k. Choose k where the error between $\|kv_{Hd} - v_H\|$ and $V_{H,\max}$ is minimized (Fig. 5c):

$$k = \frac{v_{Hd} \cdot v_H}{\|v_{Hd}\|^2}$$

Else (2) if $k_1 < 1$, $k := k_1$ (Fig. 5d or e), else $k := k_2 > 1$ (Fig. 5f).

Observe that none of the cases with $k_2 \leq 1 \leq k_1$ is included here, because they imply that $k = 1$ is a solution within the range (Fig. 5g) and so it would have been unnecessary to adjust the magnitude of v_{Hd}.

With these formulas, the procedure is as follows.

Step 1. Define a_{Hc} temporarily by a_{Hd} in Eq. (3), and define a_{zc} by clipping a_{zd} of Eq. (4) such that

$$A_{\min} \leq -a_{zc} \leq A_{\max}$$

Step 2. With the temporary a_{Hc} and the clipped a_{zc}, define φ_c according to Eq. (8). If $\|a_c\| \leq A_{\max}$ and $\varphi_c \leq \varphi_{\max}$, stop.

Step 3. With $V_{H,\max}$ computed according to Eq. (13), find k according to Cases 1 or 2, parts 1 or 2, as appropriate, and define a_{Hc} as in Eq. (9).

Step 4. If k has been determined based on Case 1, part 1, or, Case 2, part 1, adjust a_{Hc} by

$$a_{Hc} := \frac{a_{Hc}}{\|a_{Hc}\|} \min\left\{\sqrt{A_{\max}^2 - a_{zc}^2},\ (-a_{zc}) \tan \varphi_{\max}\right\}$$

Aerodynamic Force Inverse Model

This model is fundamentally the same as the one described in Ref. 4. It involves calculating the slow-time-scale controls, viz, collective and attitude commands, from the acceleration commands via an inverse model of the aerodynamic forces. The only substantial difference between the implementation described in Ref. 4 and the current implementation can be summarized as follows: To uniquely determine the commanded attitude, Ref. 4 introduced the wind frame, a coordinate system defined by the commanded velocity vector, so that the angle of side slip can be selected to be zero. Under this condition, Eqs. (18) and (19) of Ref. 4 are combined to compute the angle of attack α and the wind-axis roll angle ϕ_W, which in turn can be used to determine the commanded attitude. The implementation described in Ref. 4 used two of the three component equations given by Eqs. (18) and (19) to solve individually for α and ϕ_W using an iterative scheme. The current implementation, however, uses the same two component equations to solve for all of the multiple solutions of α and ϕ_W via a closed-form formulation and uses the third component equation to resolve the multiple solutions to identify the only correct solution pair.

Fast-Time-Scale Controller and Inverse Model

These two modules are the same as the ones described in Ref. 4 They involve 1) a linear control law to determine the Euler angular acceleration based on the attitude command from the slow-time-scale controller and feedback of the angular state and 2) an inverse model to circumvent the nonlinearities imposed by the kinematics governing the angular state.

III. Implementation of Simulation

The automatic guidance and control laws discussed in the last section have been implemented in a 3D computer graphics simulation for evaluation. The simulation employs 3D graphics to depict the helicopter, terrain, and objects such as trees, houses, and telephone poles to represent different kinds of obstacles. Such 3D models facilitate the visualization of the helicopter flight in real time, taking into account its physical dimensions and six-degree-of-freedom dynamics. Reference 9 contains a detailed description of the simulation functionalities. In this section we would cover the unique features that affect the simulation of the structure depicted in Fig. 1.

A. Helicopter Model

The helicopter is modeled as a six-degree-of-freedom point-mass system. The aerodynamics are specified by stability derivatives corresponding to a generic helicopter[10] with a built-in stability and control augmentation system (SCAS). The input consists of conventional cyclic, collective, and directional commands.

B. Sensor Model

The current sensor model uses a unique approach that exploits the special computer graphics hardware to generate the required sensor data. The integrated obstacle detection system in Fig. 1 is assumed to be able to provide range data from a sensor suite located at the nose of the helicopter. The data are computed over a 128 × 128 array structure, corresponding to a 60 deg field of view in both azimuth and elevation. The sensor model does not represent any combination of any particular active or passive sensors. It is merely devised to capture the essential capabilities of the sensors required for automatic NOE flight, striking a balance between optimism and realism of anticipated sensor capabilities. The current implementation assumes range measurements over the entire field of view. Future evaluations will address the problem where range information is available only sparsely,[11-14] and the integration of hybrid sensors will be examined.[3]

Since the 3D objects and terrain are all modeled by polygons, it is trivial to determine the range measurements given the sensor position and attitude. Performing these calculations by software, however, would be overly time consuming, and it would preclude any chance of real-time simulation on today's computer workstations. Our approach is to exploit the special computer hardware that is designed to perform hidden-surface removal in real time by saving and comparing range-related data. To this end, a 128 × 128 "window"

is generated on the computer display screen to model the sensor's viewpoint. The depth buffer, which is the memory containing the range-related data for hidden-surface removal, is examined and the quantities are transformed back into range for subsequent use.

C. Inertial-Database Model

To facilitate asynchronous fusion of data from possibly hybrid sensors, an inertial-database format has been selected for storing the data. The range measurements are converted into 3D Cartesian coordinates with respect to some predefined reference datum and saved in the inertial database in the form of altitude as a function of the horizontal components. This database format will have to be modified in the future if NOE flight beneath overhanging objects is necessary.[3]

The database is defined over a horizontal grid of 700 × 700 m. This grid is partitioned into a 7 × 7 regular subgrid structure, each measuring 100 × 100 m. The grid-cell resolution is 2 × 2 m, which is fine enough to permit a reasonably tight control tolerance. The database is continuously adjusted so that the helicopter will always be in the center subgrid. As the helicopter leaves this center subgrid, the database will shift by 100 m to bring the helicopter back into the center subgrid. The shift of 100 m according to the vehicle's position saves computation over the approach to continuously maintain the vehicle in the dead center of the database. As the database shifts, some data is erased and a new area is opened up. The new area is first initialized with interpolated DMA data and subsequently modified by sensor data. Since the database can store only one altitude value per cell, the new sensor data will replace the old value only if the new value represents a higher altitude.

For real-time simulation, the inertial database is stored as the depth buffer in a 350 × 350 window on the graphical display of the workstation, much like the idea for the sensor simulation described above. The only difference is that this window is based on an orthographically projected top view, instead of the perspective view of a sensor.

IV. Examples

The guidance and control functions discussed in Sec. II have been through substantial evaluation and improvement with the simulation described in Sec. III. Figure 6 shows an example of the automatic NOE flight. The piecewise linear white line represents the nominal trajectory to be followed. The retrace of the helicopter positions in Fig. 6 illustrates how the helicopter flies in proximity to the nominal trajectory and how it deviates from it to avoid all the obstacles, which consist only of trees in this example. The obstacle database is generated by random placement of the objects in user-selected regions. This serves to create general but varying situations for evaluating the guidance and control algorithms. This and the rest of the examples are based on a nominal helicopter speed of 15 knots and a nominal ground clearance of 10 ft.

Figures 7–9 demonstrate the different behaviors of the obstacle avoidance scheme via purposeful placement of the obstacles. Figure 7 shows that the helicopter, entering from the bottom of the figure, approaches a group of trees and goes around them after passing by two trees on its right. Observe that, when the helicopter is still at the lower half of the figure, it may not know if there exists an opening behind the two trees. Figure 8 shows the same situation with the exception that a few trees have been intentionally placed to cut off the opening between the two trees and the rest of the group. The resulting situation comprises a dead end and the helicopter is able to detect this early on. This is evident by the decision to go around the whole group of trees, and this maneuver is initiated when the helicopter is still at the bottom of the figure before it gets close to the

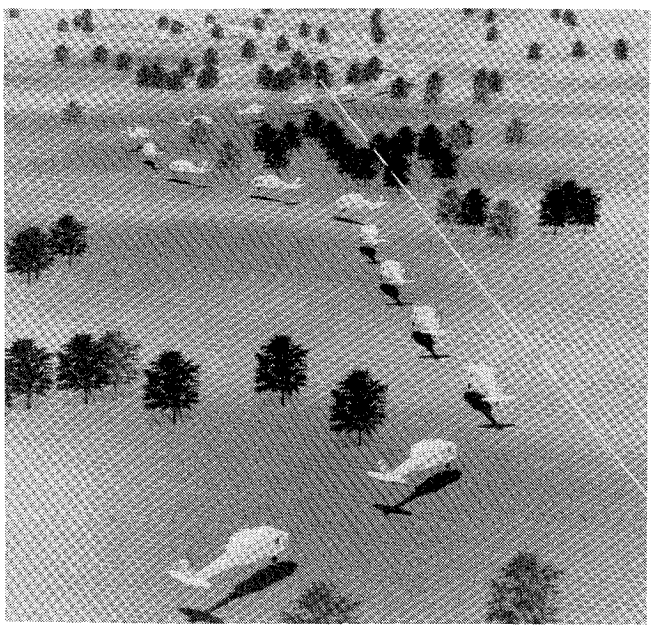

Fig. 6 Simulation example of automatic NOE flight.

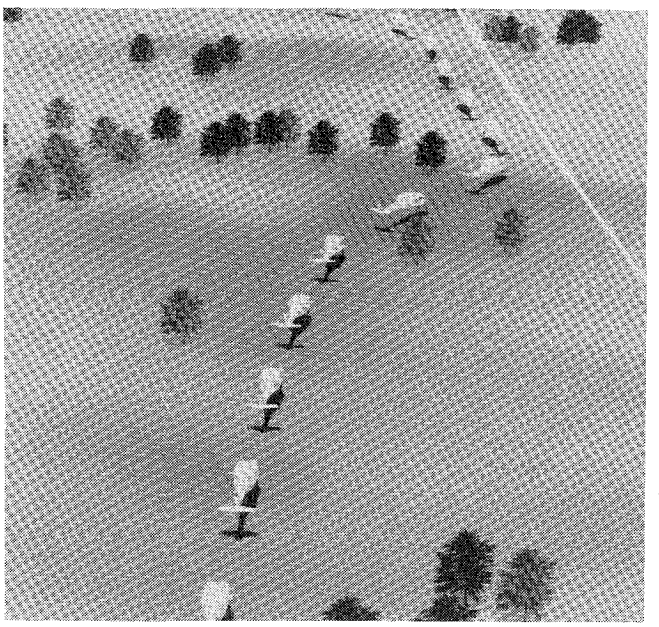

Fig. 7 Baseline example of helicopter obstacle avoidance maneuver.

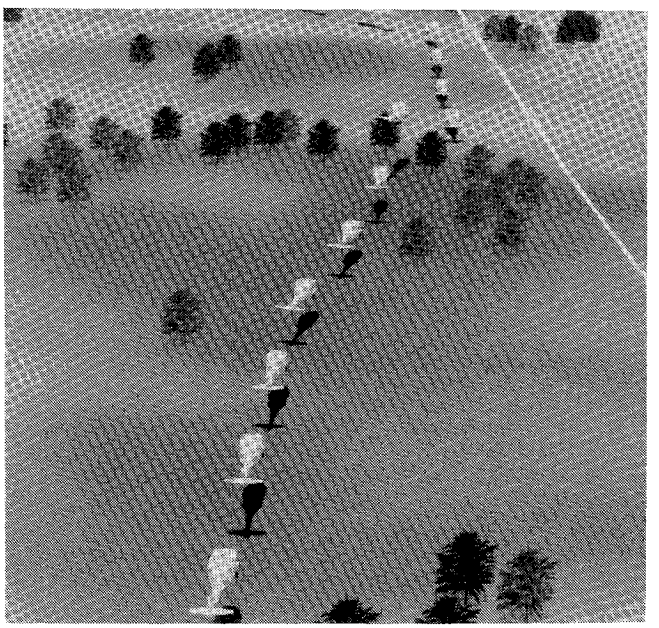

Fig. 8 Example of helicopter deviation when opening in Fig. 7 becomes visibly blocked.

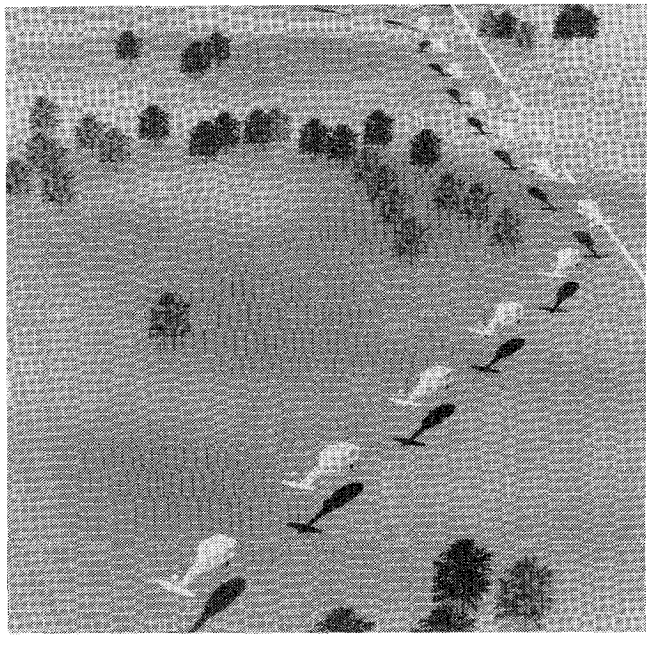

Fig. 9 Example of helicopter climb when blockage of potential opening is not visible early enough for lateral maneuver.

obstacles. Figure 9 shows the situation where the few added trees are moved toward the right so that the blockage is hidden behind the two trees mentioned in Fig. 7. Here the helicopter cannot conclude that the potential opening has been blocked before it enters the dead end, and so it enters the area similar to the situation in Fig. 7. By the time it detects the dead end, it would mean too big a deviation from the nominal direction to try to come back out to go around the trees, and so it decides to fly over the trees ahead of it.

The guidance algorithm discussed in this paper is an improved version of the one mentioned in Ref. 15, which compares the maneuvers for the earlier version with a guidance and control simulation developed for pilot evaluation.[16] Much of the overly aggressive maneuver of the earlier version discussed in Ref. 15 has been eliminated by the improved slow-time-scale controller discussed in Sec. II.B:

1) The version in Ref. 15 commands a deceleration to anticipate a hover at the point where it has to turn around some object to get it to the reference point, whereas the current version is less conservative as it commands a deceleration to a speed just sufficiently low to guarantee enough room for subsequent deceleration-to-hover by checking the forward path for remaining clearance.

2) The version in Ref. 15 did not include the provision for resolving multiple solutions of α and ϕ_W as explained above in the aerodynamic force inverse model; hence inappropriate and unnecessary maneuvers were possible in Ref. 15.

V. Conclusions

This paper covers the design and implementation of a full-function guidance and control system for automatic obstacle avoidance in helicopter NOE flight. The guidance function assumes that the helicopter is sufficiently responsive so that the flight path can be readily adjusted at NOE speeds. The controller, basically an autopilot for following the derived flight path, was implemented with parameter values to control a generic helicopter model used in the simulation.

Evaluation of the guidance and control system with a 3D graphical helicopter simulation suggests that the guidance has the potential for providing meaningful flight trajectories required for masking and obstacle avoidance in NOE flight. Improvements for safety assurance and ride comfort are desirable. Issues on pilot interaction with the automatic guidance and control system will need to be addressed before piloted simulations and flight tests can be carried out.

References

[1]Cheng, V. H. L., and Sridhar, B., "Considerations for Automated Nap-of-the-Earth Flight," *AHS Journal*, Vol. 36, No. 2, 1991, pp. 61–69.

[2]Cheng, V. H. L., "Concept Development of Automatic Guidance for Rotorcraft Obstacle Avoidance," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 2, 1990, pp. 252–257.

[3]Cheng, V. H. L., and Sridhar B., "Integration of Active and Passive Sensors for Obstacle Avoidance," *IEEE Control Systems Magazine*, Vol. 10, No. 4, 1990, pp. 43–50.

[4]Heiges, M. W., Menon, P. K. A., and Schrage, D. P., "Synthesis of a Helicopter Full-Authority Controller," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, 1992, pp. 222–227.

[5]Heiges, M. W., "A Helicopter Flight Path Controller Design via a Nonlinear Transformation Technique," Ph.D. Dissertation, Georgia Inst. of Technology, Atlanta, GA, March 1989.

[6]Swenson, H. N., "Computer Aiding for Low-Altitude Helicopter Flight," NASA TM 103861, May, 1991.

[7]Hess, R. A., and Jung, Y. C., "An Application of Generalized Predictive Control to Rotorcraft Terrain-Following Flight," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-19, No. 5, 1989, pp. 955–962.

[8]Meyer, G., and Cicolani, L., "Application of Nonlinear Systems Inverses to Automatic Flight Control Design—System Concepts and Flight Evaluation," *Theory and Applications of Optimal Control in Aerospace Systems*, edited by P. Kant, AGARDograph 251, 1980, pp. 10-1–10-29.

[9]Lam, T., and Cheng, V. H. L., "Simulation of Automatic Rotorcraft Nap-of-the-Earth Flight in Graphics Workstation Environment," *Proceedings of the AIAA/AHS Flight Simulation Technologies Conference* (Hilton Head, SC), AIAA, Washington, DC, 1992, pp. 8–15.

[10]Lewis, M. S., and Aiken, E. W., "Piloted Simulation of One-on-One Helicopter Air Combat at NOE Flight Levels," NASA TM 86686, April 1985.

[11]Sridhar, B., Cheng, V. H. L., and Phatak, A. V., "Kalman Filter Based Range Estimation for Autonomous Navigation Using Imaging Sensors," *Proceedings of the 11th IFAC Symposium on Automatic Control in Aerospace* (Tsukuba, Japan), 1989, pp. 45–50.

[12]Sridhar, B., Suorsa, R., and Smith, P. N., "Vision-Based Techniques for Low Altitude Flight," *Proceedings of the International Symposium on Intelligent Robotics* (Bangalore, India), 1991, pp. 27–37.

[13]Menon, P. K. A., and Sridhar, B., "Image Based Range Determination," *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (Portland, OR), AIAA, Washington, DC, 1990, pp. 786–796; also *Journal of Guidance, Control, and Dynamics* (to be published).

[14]Barniv, Y., "Application of Velocity Filtering to Optical Flow Passive Ranging," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 28, No. 4, 1992, pp. 957–969.

[15]Coppenbarger, R. A., and Cheng, V. H. L., "Status of Automatic Nap-of-the-Earth Rotorcraft Guidance," *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (New Orleans, LA), AIAA, Washington, DC, 1991, pp. 1436–1450.

[16]Clement, W. F., Gorder, P. J., Jewell, W. F., and Coppenbarger, R. A., "Real-Time Piloted Simulation of Fully Automatic Guidance and Control for Rotorcraft Nap-of-the-Earth (NOE) Flight Following Planned Profiles," *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (Portland, OR), AIAA, Washington, DC, 1990, pp. 507–517.