

# Software-Enabled Receding Horizon Control for Autonomous Unmanned Aerial Vehicle Guidance

Tamás Keviczky\* and Gary J. Balas†  
 University of Minnesota, Minneapolis, Minnesota 55455

**This paper describes autonomous unmanned aerial vehicle (UAV) guidance technologies developed and demonstrated in a flight test sponsored by the DARPA Software Enabled Control program. The flight experiment took place in June 2004 using a Boeing UAV testbed and demonstrated important autonomy capabilities enabled by a receding horizon guidance controller and fault detection filter. The receding horizon controller (RHC) design process is presented in detail as well as demonstration scenarios which were designed to exercise and evaluate the primary functionalities of the control system. Simulation results of the key capabilities are shown and compared with recorded flight data for evaluation purposes. Hardware-in-the-loop simulations and other high-fidelity test run results illustrate secondary capabilities such as controller reconfiguration due to actuator fault and maneuvering limit enforcement using output constraints in the receding horizon approach.**

## Nomenclature

$A, B, C, D$	= linear-time-invariant (LTI) prediction model state matrices	$\Delta r$	= commanded input signal changes of the prediction model
$A_k, B_k, C_k, D_k$	= linear-time-varying (LTV) prediction model state matrices	$\delta H_c$	= blocking parameter
$\tilde{A}_k, \tilde{B}_k, \tilde{C}_k, \tilde{D}_k$	= augmented LTV prediction model state matrices	$\varepsilon$	= slack variable for soft constraint formulation
$A_z, b_z$	= matrix and vector characterizing linearized maneuvering limits	$\zeta$	= local north coordinate
$\tilde{A}, \tilde{B}$	= state matrices of linearized kinematics	$\eta$	= local east coordinate
$d$	= additive output disturbance estimate	$\Lambda$	= longitude
$g$	= gravitational constant	$\lambda$	= latitude
$H_c$	= control prediction horizon	$\lambda_0, \Lambda_0, h_0$	= RHC engagement point in WGS-84 coordinates
$H_p$	= output prediction horizon	$\rho$	= slack variable coefficient in cost function
$\dot{h}$	= altitude rate	$\varrho$	= scheduling variable of parameter-varying prediction model
$J$	= quadratic cost function	$\phi$	= bank angle
$K_h$	= coordinate frame for altitude navigation	$\chi$	= heading
$K_{\zeta\eta}$	= coordinate frame for north-east navigation	$\dot{\chi}$	= turn rate
$n_x$	= longitudinal acceleration	( $\cdot$ )	= deviations from trim values
$n_y$	= lateral acceleration	( $\hat{\cdot}$ )	= predicted variables
$n_z$	= normal acceleration	( $\bar{\cdot}$ )	= upper limit on variables
$\underline{Q}, R$	= receding horizon control (RHC) weighting matrices	( $\underline{\cdot}$ )	= lower limit on variables
$r$	= commanded input signals of the prediction model		
$T_s$	= sampling time	<i>Subscripts</i>	
$u$	= control inputs of prediction model with hard constraints	cmd	= command signals
$V$	= ground speed	DS	= discrete-time LTI identified DemoSim state matrices
$w$	= augmented linear prediction model outputs	DStrim	= DemoSim trim values
$y$	= outputs of prediction model for tracking	DS0	= output trim adjusted DemoSim state matrices
$z$	= soft constrained outputs of prediction model	$d$	= input-delay adjusted DemoSim state matrices
$\alpha$	= angle of attack	$f$	= faulty state matrices and signals
$\beta$	= sideslip	$k$	= time step
$\gamma$	= flight-path angle	ref	= reference tracking signals
		0	= trim variables of kinematics linearization
		<i>Superscript</i>	
		o	= DemoSim state matrices including soft constrained outputs

Received 14 January 2005; revision received 9 July 2005; accepted for publication 4 August 2005; presented as Paper at the AIAA Guidance, Navigation, and Control Conference, San Francisco, CA, 15–18 August 2005. Copyright © 2005 by Tamás Keviczky. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/06 \$10.00 in correspondence with the CCC.

\*Graduate Research Assistant, Control Science and Dynamical Systems Center; keviczky@aem.umn.edu.

†Professor, Department of Aerospace Engineering and Mechanics; balas@aem.umn.edu. Associate Fellow AIAA.

## I. Introduction

INTEREST in the use of unmanned aerial vehicles (UAVs) has grown significantly in the last decade. Currently most UAVs are used for military surveillance and reconnaissance in practice, but the range of possible civilian applications is also promising. Current UAV systems are either remotely operated by a human pilot or rely on rudimentary guidance technologies that limit the number of vehicles used and the flexibility needed to accomplish complex mission tasks in the near future.<sup>1</sup>

One of the main technological goals of improving the capabilities of currently operational UAVs is to increase their autonomy by providing more advanced guidance systems: in particular, to enable the application of advanced control design methods that respect constraints of the vehicle dynamics and allow systematic design of contingencies (e.g., reconfiguration) in case a fault occurs in the system. These technologies are essential elements of a high-performance, high-risk autonomous aerial vehicle system.

Advances in software and computational power have provided the basis that is critical in implementing state-of-the-art control algorithms. To this date, very few advanced guidance and fault-detection concepts have been evaluated in flight tests on actual vehicles.<sup>2-5</sup>

The technologies described in this paper were developed and implemented as part of the Defense Advanced Research Projects Agency (DARPA) Software Enabled Control (SEC) program.<sup>6</sup> The focus of our efforts during the five-year program was online control customization for UAVs. Advances in this field have enabled a dramatic increase in military effectiveness by increasing the level of autonomy in UAVs, probability of mission success and survivability, and expanding the range of UAV missions while reducing air vehicle fatigue and life-cycle costs. The benefits to the military include use of extremely aggressive maneuvering of UAVs to achieve mission directives, accommodation of goal changes in real-time, life-extending control, and a reduced need for hardware redundancy while allowing more complex control strategies without increased software production and verification costs. A key component of the project was the development of the Open Control Platform (OCP) middleware,<sup>7</sup> which enabled seamless integration of our control algorithms in a real-time embedded environment, and the demonstration of advanced guidance capabilities on a full-scale aircraft equipped with actual future UAV avionics and software.

The paper is organized as follows. First, a brief description of Boeing's UAV testbed is given in Sec. II, followed by the receding horizon guidance controller design in Sec. III. Section IV explains the experimental scenario that formed the basis of the flight demonstration. It presents high-fidelity simulation results, hardware-in-the-loop, and actual flight-test data as well.

## II. UAV Testbed and OCP Software Environment

The aircraft used for the flight-test experiments is a modified T-33 two-seat jet trainer equipped with components of the Boeing X-45 UCAV unmanned aerial vehicle avionics and an autopilot. The autopilot provides several control functionalities such as altitude, velocity, heading, and turn-rate tracking. These different command types and functionalities are accessible through Boeing's OCP.<sup>7</sup> The OCP served as a software integration framework for flight code development and desktop simulations as well, besides hosting and interfacing the final flight code implementation.

A standard laptop computer served as the onboard, real-time flight controller hosting the OCP and the guidance algorithms. The guidance system produces the velocity, and turn-rate and altitude commands of the aircraft to follow a reference trajectory.

In the testing and control design phase, simulations were performed using a nonlinear black-box executable model of the T-33 aircraft integrated with the autopilot provided by Boeing. This open vehicle simulation model represents the nonlinear closed-loop aircraft dynamics and will be referred to as DemoSim.

The receding horizon control (RHC) based guidance system was designed based on identified linear time-invariant closed-loop vehicle dynamics using DemoSim and the techniques presented in Ref. 8. Reference 9 contains a detailed description of the identification process. The nonlinear operational constraints of the vehicle were approximated by linear expressions as described in Sec. II.B.

### A. DemoSim Modeling

The inputs to the identified linear-time-invariant (LTI) DemoSim model are  $V_{cmd}$ ,  $\dot{\chi}_{cmd}$ , and  $\dot{h}_{cmd}$ . The model outputs are  $V$ ,  $\chi$ , and  $\gamma$ . A representation of these variables is depicted in Fig. 1 using a fixed local coordinate frame.

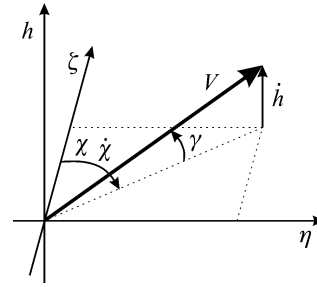


Fig. 1 Representation of DemoSim model input and output variables in a local coordinate frame.

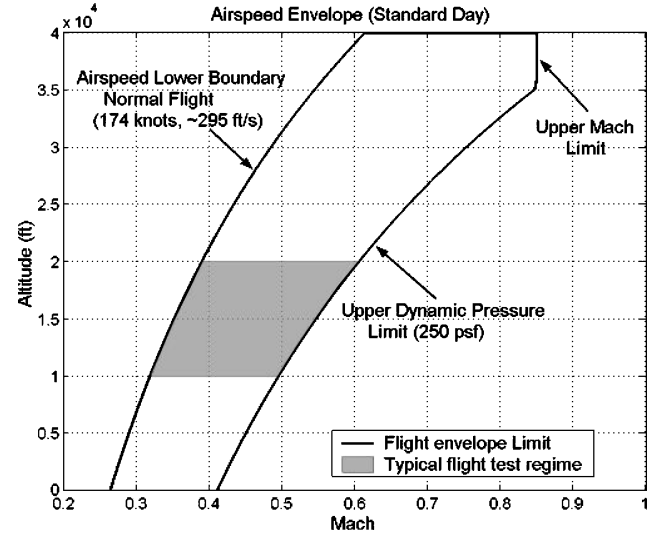


Fig. 2 T-33/UCAV airspeed flight envelope in terms of altitude and Mach.

The linear DemoSim dynamics was identified at the following input-output trim values:

$$\begin{aligned} V_{D\text{trim}} &= 505 \text{ ft/s} \\ \dot{\chi}_{D\text{trim}} &= 0 \text{ rad/s}, & \chi_{D\text{trim}} &= (\pi/2) \text{ rad} \\ \dot{h}_{D\text{trim}} &= 0 \text{ ft/s}, & \gamma_{D\text{trim}} &= 0 \text{ rad} \end{aligned} \quad (1)$$

### B. Characterization of Flight Envelope Constraints

The approximate nonlinear flight envelope limits of the DemoSim model were characterized and represented by a collection of linear constraints that could be accommodated by the RHC formulation. The most important constraints to be characterized and enforced by any chosen control approach were found to be the following: 1) flight envelope limits in terms of altitude and airspeed, 2) limits on maximum bank angle, 3) minimum and maximum vertical and longitudinal acceleration limits, and 4) any other important limitations induced by the autopilot in DemoSim.

Figure 2 shows the flight envelope of the T-33 aircraft augmented with the UCAV avionics and autopilot. The actual flight envelope limits were restricted to the shaded area for the purpose of the flight-test demonstration.

These flight regime constraints were respected by choosing a reference trajectory that lies between the minimum and maximum altitude limits. The constraints on minimum and maximum velocity were imposed by limiting the allowable deviations ( $\pm 100$  ft/s) from the trim velocity value (approximately 500 ft/s). These limits were incorporated into the controller in terms of ground speed.

The main objective of characterizing the remaining three types of constraints was to arrive at formulas that could be approximated with linear expressions of the DemoSim model output variables and their derivatives.

Based on the limited information available about the autopilot incorporated into DemoSim, the following assumptions were

reasonable to make:

1) Angle of attack can be considered zero  $\alpha \approx 0$ .

2) The aircraft performs coordinated turns; therefore, the side force and sideslip angles are approximately zero  $n_y \approx 0, \beta \approx 0$ . Hence in level flight ( $\gamma = 0$ )

$$V \dot{\chi} / g = \tan \phi \quad (2)$$

Using these assumptions, the following three equations represent different types of constraints

1) Longitudinal acceleration constraint:

$$n_x g = \dot{V} + g \sin \gamma \quad (3)$$

2) Turn constraint equation:

$$0 = \cos \phi \cdot V \dot{\chi} \cos \gamma - \sin \phi (V \dot{\gamma} + g \cos \gamma) \quad (4)$$

3) Vertical acceleration constraint equation:

$$\sqrt{(V \dot{\chi} / g)^2 + (V \dot{\gamma} / g \cos \gamma + 1)^2} = -n_z / \cos \gamma \quad (5)$$

Reference 9 contains a detailed derivation of these approximate nonlinear flight envelope limits.

By studying the behavior of DemoSim and running extensive simulations, the following set of limitations on velocity, vertical acceleration, and bank angle were determined:

$$\begin{aligned} V_{\max} &= 630 \text{ ft/s}, & V_{\min} &= 340 \text{ ft/s} \\ n_{z,\max} &= 1.4 g, & n_{z,\min} &= 0.6 g \\ \phi_{\max} &= 32 \text{ deg}, & \phi_{\min} &= -32 \text{ deg} \end{aligned} \quad (6)$$

The flight-path angle  $\gamma$  was also limited to approximately  $\pm 2$  deg.

With these limits in mind and using the small angle approximation  $\cos \gamma \approx 1$ , we arrive at the nonlinear expressions that serve as the basis for characterizing the important constraints to be enforced by the controller. Using Eqs. (2), (4), (5) and the limits in Eq. (6) leads to the following inequalities in the three variables of  $V$ ,  $\dot{\chi}$ , and  $\dot{\gamma}$ :

$$V_{\min} < V < V_{\max} \quad (7a)$$

$$\frac{g \tan \phi_{\min}}{V_{\min}} < \dot{\chi} < \frac{g \tan \phi_{\max}}{V_{\min}} \quad (7b)$$

$$(n_{z,\min} g)^2 < (V \dot{\chi})^2 + (V \dot{\gamma} + g)^2 < (n_{z,\max} g)^2 \quad (7c)$$

$$\tan \phi_{\min} < \frac{V \dot{\chi}}{V \dot{\gamma} + g} < \tan \phi_{\max} \quad (7d)$$

[The constraint on longitudinal acceleration ( $n_x$ ) in Eq. (3) is respected only indirectly by limiting the change of velocity command.]

Figure 3 represents the different types of constraints that were considered. Constraint type 1 represents bank angle  $\phi$  limits based on the inequalities in Eq. (7d), and constraint type 2 shows vertical acceleration limitations imposed by Eq. (7c). Constraint type 3 was discovered after running extensive simulations with DemoSim. We could not explain this phenomenon using mathematical formulas; however, its limiting effect could be approximated with straight lines very well in the given coordinate frame. This limitation is likely induced by the autopilot in situations when the aircraft is trying to pitch down ( $n_z$  approaches its minimum), while turning hard (large  $\dot{\chi}$ ).

Because the coordinate axes each depend on two variables and both axes share velocity as one of the variables, the shape of nonlinear constraints can be visualized in a three-dimensional plot. The individual axes in this case are the three variables  $V$ ,  $\dot{\chi}$ , and  $\dot{\gamma}$ . This three-dimensional characterization of constraints is illustrated in Fig. 4. The “front” and “back” facets of the shape representing  $V_{\min}$  and  $V_{\max}$  limits are not shown in this or any subsequent plots for better visibility. Notice that the  $\dot{\chi}_{\min}$  and  $\dot{\chi}_{\max}$  limits (7b) become “active” in forming the boundary of this nonconvex region near the edges at lower velocity and positive  $\dot{\gamma}$  values.

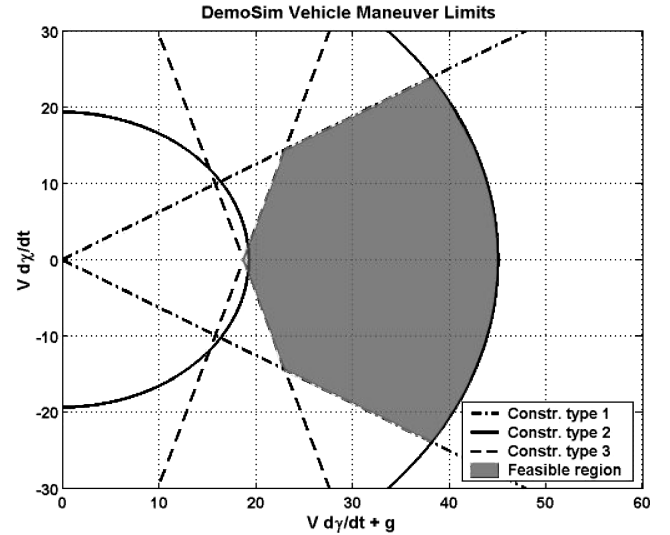


Fig. 3 Characterization of nonlinear constraints (2D).

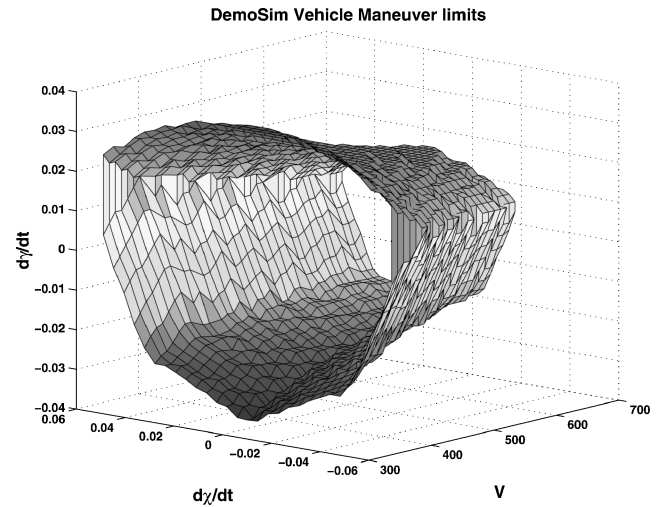


Fig. 4 Characterization of nonlinear constraints (3D).

The quadratic programming (QP)-based RHC framework described later in Sec. III requires constraints to be formulated as linear inequalities involving the decision variables. The objective of representing the nonlinear constraints by linear expressions using the aforementioned three variables was achieved by finding a polyhedron that is an inner approximation of the nonconvex feasible region shown in Fig. 4. The resulting polytope, which is constructed from a small number of halfspaces compromising approximation accuracy, is depicted in Fig. 5. (Note that the bounding planes associated with  $V_{\min}$ ,  $V_{\max}$  limits are not displayed. Hyperplanes representing  $\dot{\chi}_{\min}$  and  $\dot{\chi}_{\max}$  limits are also not shown, because other facets of the polytope rendered these redundant.) The number of half-spaces that characterize the polytope and represent linear constraints in terms of the three variables were kept small to allow the complexity of the problem to be maintained at a manageable size.

Using the half-space representation of the inner polyhedral approximation to the nonconvex constraint set, the original nonlinear maneuver limits can be characterized by the following linear inequality:

$$A_z \begin{bmatrix} V \\ \dot{\chi} \\ \dot{\gamma} \end{bmatrix} \leq b_z \quad (8)$$

where  $A_z \in \mathbb{R}^{10 \times 3}$  and  $b_z \in \mathbb{R}^{10}$ . This inequality represents 10 half-space constraints, six of which are depicted in Fig. 5. The remaining

four are the maximum and minimum limits on velocity  $V$  and turn rate  $\dot{\chi}$ .

**C. Prediction Model**

The prediction model was chosen to accommodate two important requirements. It has to provide a reasonably accurate description of the dynamic relationship between the control inputs of the test platform and the output signals of interest, which include position coordinates for tracking performance and other variables used for describing maneuvering constraints. At the same time, it has to be simple enough to limit the complexity of the optimization problem that is solved online in a receding horizon fashion.

These objectives were captured by constructing the prediction model from the identified LTI DemoSim dynamics and a flat-Earth kinematic model. Using linearized kinematics that were updated at every time step, a discrete-time linear time-invariant prediction model was derived. Although this model was fixed throughout the prediction horizon of the optimization problem, the updates to the linearized kinematics part rendered the prediction model a linear parameter-varying system, which depended on the current velocity, heading, and flight-path angle values.

The continuous-time nonlinear prediction model, composed of the LTI dynamics and the nonlinear kinematics, is depicted in Fig. 6.

The tilded input and output variables of the LTI DemoSim model represent deviations from the trim values:

$$V_{cmd} = V_{DStrim} + \tilde{V}_{cmd}, \quad V = V_{DStrim} + \tilde{V} \quad (9a)$$

$$\dot{\chi}_{cmd} = \dot{\chi}_{DStrim} + \tilde{\chi}_{cmd}, \quad \chi = \chi_{DStrim} + \tilde{\chi} \quad (9b)$$

$$\dot{h}_{cmd} = \dot{h}_{DStrim} + \tilde{h}_{cmd}, \quad \gamma = \gamma_{DStrim} + \tilde{\gamma} \quad (9c)$$

A schematic diagram of the linearized prediction model is shown in Fig. 7. Note the nonlinear kinematics are linearized around fixed  $V_0, \chi_0, \gamma_0$  values that represent current measurements. The inputs fed into the linearized kinematics model represent the differences between the true values predicted by the LTI DemoSim model (after addition of trim values) and the current measurements used for linearization:

$$\tilde{V}_0 = (V_{DStrim} + \tilde{V}) - V_0 \quad (10a)$$

$$\tilde{\chi}_0 = (\chi_{DStrim} + \tilde{\chi}) - \chi_0 \quad (10b)$$

$$\tilde{\gamma}_0 = (\gamma_{DStrim} + \tilde{\gamma}) - \gamma_0 \quad (10c)$$

The linearized model was discretized using the discrete-time identified DemoSim dynamics and approximating the continuous-time nonlinear kinematics around fixed  $V_0, \chi_0, \gamma_0$  values with the following forward-Euler discretized linear system:

$$\begin{bmatrix} \tilde{\zeta}(k+1) \\ \tilde{\eta}(k+1) \\ \tilde{h}(k+1) \end{bmatrix} = \tilde{A} \begin{bmatrix} \tilde{\zeta}(k) \\ \tilde{\eta}(k) \\ \tilde{h}(k) \end{bmatrix} + \tilde{B} \begin{bmatrix} \tilde{V}_0(k) \\ \tilde{\chi}_0(k) \\ \tilde{\gamma}_0(k) \end{bmatrix} \quad (11)$$

where

$$\tilde{A} = I_3, \quad \tilde{B} = T_s \cdot \begin{bmatrix} C_{\chi_0} C_{\gamma_0} & -V_0 S_{\chi_0} C_{\gamma_0} & -V_0 C_{\chi_0} S_{\gamma_0} \\ S_{\chi_0} C_{\gamma_0} & V_0 C_{\chi_0} C_{\gamma_0} & -V_0 S_{\chi_0} S_{\gamma_0} \\ S_{\gamma_0} & 0 & V_0 C_{\gamma_0} \end{bmatrix} \quad (12)$$

In Eq. (12), the sampling time  $T_s$  is 0.5 s, and  $C_{(\cdot)}, S_{(\cdot)}$  are short notation for  $\cos(\cdot)$  and  $\sin(\cdot)$ , respectively.

The addition of DemoSim trim values and the subtraction of the current measurement values at the output of the LTI DemoSim model were implemented by augmenting the LTI dynamics with extra states (integrators with zero inputs), which were subtracted from

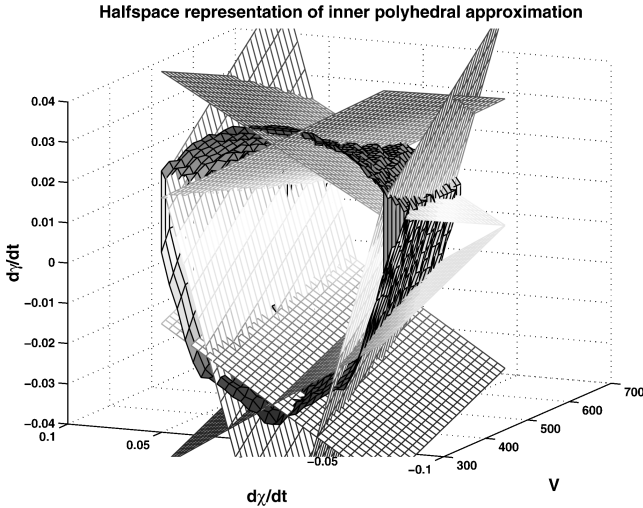


Fig. 5 Inner polyhedral approximation of nonlinear constraints (3D).

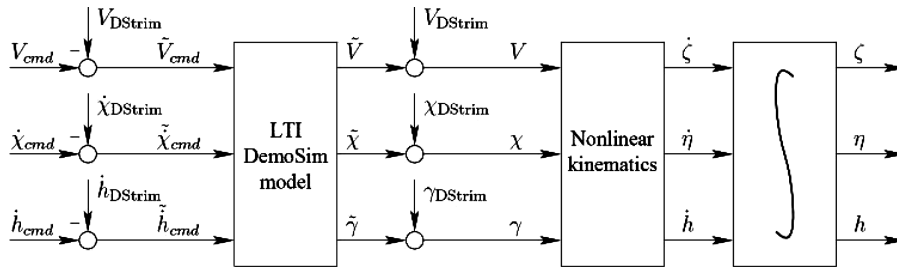


Fig. 6 Nonlinear prediction model.

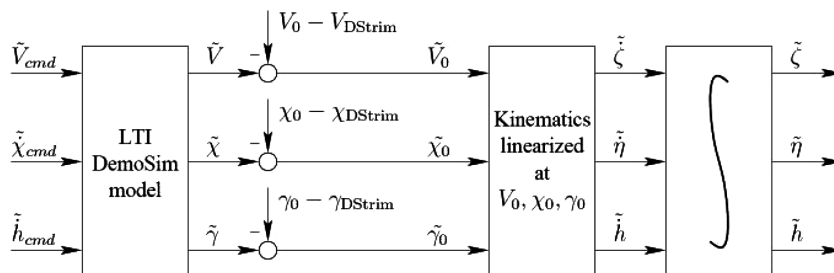


Fig. 7 Linearized prediction model.

the model outputs. The state values were updated every time step based on the difference between DemoSim trim values and current measurements. Denote the original discrete-time LTI DemoSim matrices with  $A_{DS}$ ,  $B_{DS}$ ,  $C_{DS}$ ,  $D_{DS}$ , and using  $A_{DS0}$ ,  $B_{DS0}$ ,  $C_{DS0}$ ,  $D_{DS0}$  to denote the augmented dynamics that adjust the output values based on current measurements ( $V_0$ ,  $\chi_0$ ,  $\gamma_0$ ) leads to

$$\begin{aligned} A_{DS0} &= \begin{bmatrix} A_{DS} & 0 \\ 0 & I_3 \end{bmatrix}, & B_{DS0} &= \begin{bmatrix} B_{DS} \\ 0 \end{bmatrix} \\ C_{DS0} &= [C_{DS} \quad -I_3], & D_{DS0} &= D_{DS} \end{aligned} \quad (13)$$

The estimated time delays associated with the pilot model and data processing by the onboard avionics were accounted for by augmenting the discrete-time DemoSim dynamics with extra states. The command input time delays were characterized as integer multiples of the sampling time  $T_s$ . Assuming a time delay of  $n_d$  samples affecting each input channel, the state-space matrices of the DemoSim dynamics were augmented as

$$\begin{aligned} A_d &= \begin{bmatrix} A_{DS0} & B_{DS0} & 0 \\ 0 & 0 & I_{3(n_d-1)} \\ 0 & 0 & 0 \end{bmatrix}, & B_d &= \begin{bmatrix} 0 \\ 0 \\ I_3 \end{bmatrix} \\ C_d &= [C_{DS0} \quad D_{DS0} \quad 0], & D_d &= 0 \end{aligned} \quad (14)$$

Since the nonlinear kinematics part of the prediction model was always linearized around the current measurements of  $V_0$ ,  $\chi_0$ ,  $\gamma_0$ , the outputs of this augmented, modified LTI DemoSim model could be fed directly into the linearized kinematics model. In other words, the complete linear prediction model could be obtained by the following augmentation of the state-space matrices:

$$\begin{aligned} A &= \begin{bmatrix} A_d & 0 \\ \tilde{B}C_d & \tilde{A} \end{bmatrix}, & B &= \begin{bmatrix} B_d & 0 \\ \tilde{B}D_d & 0 \end{bmatrix} \\ C &= [0 \quad I_3], & D &= [0 \quad I_3] \end{aligned} \quad (15)$$

Additional disturbance inputs were added in the preceding formulas (15) to model additive output disturbance that affects the plant.

The states associated with the DemoSim dynamics part of the prediction model were updated using a linear-time-invariant observer that relied on the control inputs sent from the RHC controller to the plant ( $V_{cmd}$ ,  $\dot{\chi}_{cmd}$ ,  $\dot{h}_{cmd}$ ) and measurements of ground speed  $V$ , heading angle  $\chi$ , and flight-path angle  $\gamma$ . Note the flight-path angle could not be measured directly on the test platform, and so a conversion from ground speed and altitude rate measurements was performed to obtain flight-path angle according to  $\gamma = \arcsin(\dot{h}/V)$ . The input-output scheme of the RHC controller and the observer are shown in Fig. 8.

The T-33 test platform avionics provides global-positioning-system (GPS) position measurements in terms of latitude  $\lambda$ , longitude  $\Lambda$ , and altitude  $h$  using the WGS-84 system. These measurements were converted to north  $\zeta$ , east  $\eta$ , and altitude  $h$  values to be compatible with the coordinate frame of the reference trajectory. The north and east coordinates were obtained by the transformation of the geodetic measurements (GPS) into an NEU (north-east-up) local Cartesian coordinate frame that had its origin at the point in space where the RHC controller is engaged ( $\lambda_0$ ,  $\Lambda_0$ ,  $h_0$ ). This

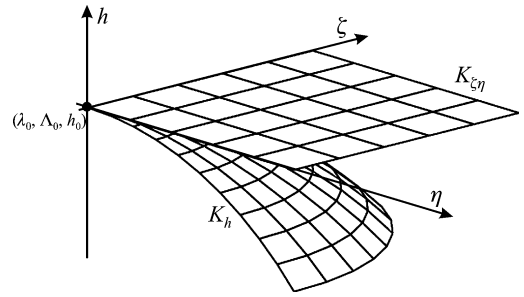


Fig. 9 Navigation frames for north, east, and altitude coordinates.

flat-Earth coordinate frame for north-east navigation is depicted in Fig. 9 as  $K_{\zeta\eta}$ . The geodetic GPS altitude measurements however were used directly, without any further conversion. This geodetic coordinate frame for altitude navigation is also illustrated in Fig. 9 and denoted by  $K_h$ . The states associated with the position integrators of the linearized prediction model were always initialized at zero (i.e., current position).

The output signals of the entire linear prediction model were assigned to three objective groups denoted by  $u$ ,  $z$ , and  $y$ . These correspond to hard actuator or other input constraints, maneuvering limits, and tracking performance, respectively. The commanded input signals are denoted by  $r$ . The prediction model in Eq. (15) has only  $y$  outputs for tracking performance.

$$\begin{aligned} y &= \begin{bmatrix} \tilde{\zeta} \\ \tilde{\eta} \\ \tilde{h} \end{bmatrix}, & z &= A_z \begin{bmatrix} V \\ \dot{\chi} \\ \dot{y} \end{bmatrix} \\ u = r &= \begin{bmatrix} \tilde{V}_{cmd} \\ \tilde{\chi}_{cmd} \\ \tilde{h}_{cmd} \end{bmatrix}, & \Delta r &= \begin{bmatrix} \Delta \tilde{V}_{cmd} \\ \Delta \tilde{\chi}_{cmd} \\ \Delta \tilde{h}_{cmd} \end{bmatrix} \end{aligned} \quad (16)$$

Note that outputs  $z$  used to define maneuvering limits were not included in the flight code. These output constraints were implemented only for testing in high-fidelity simulations, as described in more detail in Secs. III.D and IV.

The parameter variance of the prediction model, because of linearization of nonlinear kinematics, is characterized by the nominal velocity  $V_0$ , heading  $\chi_0$ , and flight-path angle  $\gamma_0$ , around which the kinematic model was linearized. The parameter dependence is present in the  $\tilde{B}$  matrix of the linearized kinematics model (12). Denoting the vector of parameters with  $\varrho(k) = [V_0(k) \quad \chi_0(k) \quad \gamma_0(k)]^T$ , the linearized discrete-time prediction models have the form

$$x(k+1) = A_k x(k) + B_k r(k), \quad \begin{bmatrix} y(k) \\ z(k) \\ u(k) \end{bmatrix} = C_k x(k) + D_k r(k) \quad (17)$$

where the parameter dependency of the prediction model is indicated by the subscript  $k$ , meaning

$$\begin{aligned} A_k &= A(\varrho(k)), & B_k &= B(\varrho(k)) \\ C_k &= C(\varrho(k)), & D_k &= D(\varrho(k)) \end{aligned}$$

### III. RHC-Based Guidance Design

Receding horizon control techniques, also known as model predictive control (MPC) or model based predictive control methods, have been in the limelight of significant research efforts, motivated by several successful industrial applications.<sup>10–13</sup> The process industry provided a perfect fit for these algorithms that respected critical process constraints to achieve safer and more efficient operation of industrial plants. These applications were not only “well suited”

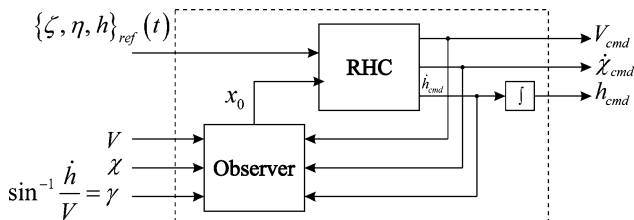


Fig. 8 RHC controller and observer.

for RHC methods but because of their relatively slow dynamics (large time constants) the significant computational effort of repetitive optimization, which is inherently involved in receding horizon approaches, could be accommodated by the relatively infrequent updates of the control signal.

Online optimization performed within RHC schemes allows control engineers to incorporate in the problem formulation and address physical process constraints explicitly. Furthermore, the flexibility of this solution approach lends itself to online reconfiguration and adaptation to changes in the environment such as updates to process models and constraints. These characteristics make receding horizon control schemes attractive for potential application in areas, in which process constraints and performance are critical, yet currently addressed with classical control techniques. In the past few decades it became apparent that predictive control methods possess qualities that could be utilized in more complex, nonlinear applications, possibly with much faster dynamics such as autopilot flight controls<sup>14</sup> or autonomous UAV guidance. A main consideration of RHC schemes is real-time implementation, that is, whether sufficient computational resources are available to accommodate repetitive solution of the optimization problem within each sampling time interval. The software infrastructure mentioned in Secs. II and III.B enabled us to implement the chosen RHC formulation in real time, which will be described in Sec. III.A.

The objective of the RHC control design was to track a time-stamped three-dimensional position reference trajectory in the presence of constraints that limit the maneuverability of the aircraft. Vehicle guidance and trajectory tracking had to be performed using the identified guidance-level vehicle model described in the preceding section, while explicitly accounting for the actual vehicle maneuvering capabilities such as flight envelope and dynamics constraints. Secondary capabilities included output constraint enforcement and reconfiguration based on actuator fault detection, which were tested in high-fidelity simulations only.

### A. Problem Formulation

The position reference trajectory was specified in terms of north, east, and altitude coordinates in a local NEU coordinate frame relative to the point in space where the controller is engaged. The reference position vector elements were placed 0.5 s apart from each other in time. Denote the reference position trajectory values by  $\zeta_{\text{ref}}(k)$ ,  $\eta_{\text{ref}}(k)$ ,  $h_{\text{ref}}(k)$  at time step  $k$ . The LTI prediction model based RHC controller was required to track a linearized position reference trajectory  $\tilde{\zeta}_{\text{ref}}(k)$ ,  $\tilde{\eta}_{\text{ref}}(k)$ ,  $\tilde{h}_{\text{ref}}(k)$ . This was generated by subtracting the simulated output of the nonlinear kinematics from the original reference trajectory. The nominal simulated trajectory was calculated using fixed  $V_0$ ,  $\chi_0$ ,  $\gamma_0$  values based on the current measurements used for linearization. The linear reference trajectory was therefore obtained by

$$(\tilde{\zeta}, \tilde{\eta}, \tilde{h})_{\text{ref}}(k) = (\zeta, \eta, h)_{\text{ref}}(k) - (\zeta, \eta, h)_0(k) \quad (18)$$

where the nominal simulated trajectories were calculated by

$$\zeta_0(k) = \sum_{i=1}^k T_s \cdot \dot{\zeta}_0(V_0, \chi_0, \gamma_0) \quad (19a)$$

$$\eta_0(k) = \sum_{i=1}^k T_s \cdot \dot{\eta}_0(V_0, \chi_0, \gamma_0) \quad (19b)$$

$$h_0(k) = \sum_{i=1}^k T_s \cdot \dot{h}_0(V_0, \chi_0, \gamma_0) \quad (19c)$$

The state values that represent the constant additive output disturbance in the prediction model were updated every time step based on the following disturbance filter:

$$d(k+1) = 0.99d(k) + 0.01d_{\text{in}}(k) \quad (20)$$

where the input  $d_{\text{in}}(k)$  was determined from the following error equation:

$$d_{\text{in}}(k) = \underbrace{\begin{bmatrix} \zeta(k) \\ \eta(k) \\ h(k) \end{bmatrix}}_{\text{pos. meas.}} - \underbrace{\begin{bmatrix} \zeta_0(k|k-1) \\ \eta_0(k|k-1) \\ h_0(k|k-1) \end{bmatrix}}_{\text{nom. pos. pred.}} - \underbrace{\begin{bmatrix} \tilde{\zeta}(k|k-1) \\ \tilde{\eta}(k|k-1) \\ \tilde{h}(k|k-1) \end{bmatrix}}_{\text{pred. lin. output}} \quad (21)$$

where  $\zeta_0(k|k-1)$ ,  $\eta_0(k|k-1)$ ,  $h_0(k|k-1)$  are one-step ahead position predictions based on the nonlinear kinematics model and  $V_0(k-1)$ ,  $\chi_0(k-1)$ ,  $\gamma_0(k-1)$  measurements at time  $k-1$ .

The optimization problem setup is based on the linear MPC formulation of Ref. 13 with some modifications. In most linear predictive controllers, the performance is specified by the following quadratic cost function to be minimized, which will also be adopted here:

$$J(k) = \sum_{i=1}^{H_p} \|\hat{y}(k+i|k) - y_{\text{ref}}(k+i|k)\|_Q^2 + \sum_{i=0}^{H_c-1} \|\Delta r(k+i|k)\|_R^2 + \rho \varepsilon \quad (22)$$

where  $\hat{y}(k+i|k)$  is the  $i$ -step ahead prediction of the outputs based on data up to time  $k$ .  $H_p$  denotes the number of steps in the output prediction horizon. These predictions of the outputs are functions of future control increments  $\Delta r(k+i|k)$  for  $i = 0, \delta H_c, 2\delta H_c, \dots, H_c - 1$ . The integer number of samples  $H_c$  is called the control horizon; the control signal is allowed to change only at integer multiples of  $\delta H_c$  samples and is set to be constant for all  $i \geq H_c$ . This means that the future control signal has the form of a stair-step function with steps occurring at  $\delta H_c$  intervals. The reference signal  $y_{\text{ref}}$  represents the desired outputs, and  $Q$  and  $R$  are suitably chosen weighting matrices. For this specific application, the optimization problem was specified using the following parameters:

$$H_c = 1, \quad \delta H_c = 1, \quad H_p = 40$$

$$Q = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 \cdot 10^6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The slack variable  $\varepsilon$  and its weight  $\rho$  are used for softening constraints. The exact purpose of the slack variable and weight in the problem formulation will be clarified shortly.

To obtain the predictions for the signals of interest, a model of the process is needed. By using a linear model, the resulting optimization problem of minimizing  $J(k)$  will be a QP problem, for which fast and numerically reliable algorithms are available. The linearized prediction model, introduced in Sec. II.C, is augmented with extra states to fit the formulation in this RHC scheme. Three integrators are added to convert the control changes  $\Delta r$  into actual control commands  $r$ , each one associated with the command inputs of velocity, turn rate, and altitude rate. A simple disturbance model is incorporated to the state-space description of the prediction model in Eq. (23), which assumes constant disturbances are acting on outputs. The constant disturbance estimates are obtained by filtering the difference between measured and predicted outputs, as described by Eq. (21).

The disturbance model also serves to mitigate the effect of model mismatch. The augmented linear prediction model has the following

form:

$$\begin{aligned} \begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \\ r(k) \end{bmatrix} &= \begin{bmatrix} A_k & 0 & B_k \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \\ r(k-1) \end{bmatrix} + \begin{bmatrix} B_k \\ 0 \\ I \end{bmatrix} \Delta r(k) \\ \begin{bmatrix} \hat{y}(k) \\ \hat{z}(k) \\ \hat{u}(k) \end{bmatrix} &= \begin{bmatrix} I \\ C_k & 0 & D_k \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \\ r(k-1) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ D_k \end{bmatrix}}_{\mathcal{D}_k} \Delta r(k) \quad (23) \\ \underbrace{\begin{bmatrix} \hat{y}(k) \\ \hat{z}(k) \\ \hat{u}(k) \end{bmatrix}}_{\hat{w}(k)} & \quad \underbrace{\begin{bmatrix} I \\ C_k & 0 & D_k \\ 0 \\ 0 \end{bmatrix}}_{\mathcal{C}_k} \quad \underbrace{\begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \\ r(k-1) \end{bmatrix}}_{\hat{\xi}(k)} \end{aligned}$$

As in most applications, there are level and rate limits on control inputs. These are enforced as hard constraints

$$\underline{u} \leq \hat{u}(k+1|k), \dots, \hat{u}(k+H_p|k) \leq \bar{u} \quad (24)$$

$$\underline{\Delta r} \leq \Delta r(k), \Delta r(k+\delta H_c), \dots, \Delta r(k+H_c) \leq \bar{\Delta r} \quad (25)$$

because the RHC algorithm has almost direct control over them. (The optimization variables are the changes in control inputs.) Hence there is no modeling uncertainty associated with this aspect of the prediction model. Another type of constraint is also considered in this specific application example represented by certain maneuvering limits on the aircraft. The controller has to be versatile enough to handle these limits that might be system-state dependent or change according to different stages of a mission. The most important maneuvering constraints of the T-33 testbed were characterized based on the DemoSim open vehicle executable model. These constraints arose mainly from vertical acceleration and bank angle limits. Although the relationship between the control inputs and these variables is nonlinear, it could be approximated reasonably well with linear expressions as described in Sec. II.B. This allowed their incorporation into the RHC problem formulation as output constraints. It is vital that these limits are treated as soft constraints because disturbances and model mismatch can easily lead to infeasibility problems if hard constraints are included on these type of output signals. Details of the soft output constraint formulation are presented in Sec. III.D and can also be found in Ref. 9. Note that the actual flight code was flown without the inclusion of these type of constraints.

The numerical values of the  $u$  output signal limits (representing hard constraints) were specified as

$$\underline{u} = \begin{bmatrix} -100 \text{ ft/s} \\ -0.035 \text{ rad/s} \\ -1000 \text{ ft/s} \end{bmatrix}, \quad \bar{u} = \begin{bmatrix} 100 \text{ ft/s} \\ 0.035 \text{ rad/s} \\ 1000 \text{ ft/s} \end{bmatrix} \quad (26)$$

Hard constraints were put on the optimization variables as well, specifically,

$$\underline{\Delta r} = \begin{bmatrix} -4 \text{ ft/s}^2 \\ -0.01 \text{ rad/s}^2 \\ -1000 \text{ ft/s}^2 \end{bmatrix}, \quad \bar{\Delta r} = \begin{bmatrix} 4 \text{ ft/s}^2 \\ 0.01 \text{ rad/s}^2 \\ 1000 \text{ ft/s}^2 \end{bmatrix} \quad (27)$$

Constraint softening for the outputs  $z$  is accomplished by introducing an additional slack variable, which allows some level of constraint violation if no feasible solution exists

$$\underline{z} - \varepsilon \leq \hat{z}(k+1|k), \dots, \hat{z}(k+H_p|k) \leq \bar{z} + \varepsilon, \quad 0 \leq \varepsilon \quad (28)$$

It is beneficial to use an  $\infty$ -norm (maximum violation) penalty on constraint violations [as shown in Eqs. (22) and (28)] because it gives an “exact penalty” method if the weight  $\rho$  is large enough. This means that constraint violations will not occur unless no feasible solution exists to the original “hard” problem. If a feasible solution exists, the same solution will be obtained as with the hard formulation.

## B. Implementation Within the OCP

The RHC guidance algorithm was implemented using an application programming interface developed specifically for receding horizon control algorithms (RHC API) in a real-time environment using the OCP. The RHC API provides an interface to an online mathematical program solver by formulating a generic optimization problem for receding horizon applications. Implementation of the control algorithms is performed using three separate threads. One of these is used to enable anytime scheduling of the online optimization solver while satisfying real-time requirements with the other two hard real-time tasks. The reader is referred to Ref. 15 for further details.

After creating the prediction model and formulating the RHC problem, the optimization problem was translated to the formulation used by the RHC API. To gain insight to the RHC problem, a standard quadratic programming solution based on substitution is shown next. Although this derivation does not represent the actual process that was used within the RHC API to solve the problem, some parts of this development were used in the flight code to obtain predictions of the system for the disturbance estimator.

By using successive substitution, it is straightforward to derive that the prediction model of inner-loop outputs (signals of interest) over the prediction horizon is given by Eq. (29).

Denote parts of the state matrices  $\mathcal{C}_k$  and  $\mathcal{D}_k$  in Eq. (29) that correspond to the predicted outputs  $\hat{y}(k)$ ,  $\hat{z}(k)$ ,  $\hat{u}(k)$  in  $\hat{w}(k)$ , with an additional  $y, z, u$  subscript:

$$\mathcal{C}_k = \begin{bmatrix} \mathcal{C}_{ky} \\ \mathcal{C}_{kz} \\ \mathcal{C}_{ku} \end{bmatrix}, \quad \mathcal{D}_k = \begin{bmatrix} \mathcal{D}_{ky} \\ \mathcal{D}_{kz} \\ \mathcal{D}_{ku} \end{bmatrix}$$

Consider only those predicted outputs that appear in the performance index:

$$\hat{y}(k) = \mathcal{C}_{ky} \hat{\xi}(k) + \mathcal{D}_{ky} \Delta r(k)$$

$$\mathcal{Y}(k) = [\hat{y}(k+1|k), \dots, \hat{y}(k+H_p|k)]^T$$

$$\underbrace{\begin{bmatrix} \hat{w}(k+1|k) \\ \hat{w}(k+2|k) \\ \vdots \\ \hat{w}(k+H_c|k) \\ \hat{w}(k+H_c+1|k) \\ \vdots \\ \hat{w}(k+H_p|k) \end{bmatrix}}_{\mathcal{W}(k)} = \underbrace{\begin{bmatrix} \mathcal{C}_k \mathcal{A}_k \\ \mathcal{C}_k \mathcal{A}_k^2 \\ \vdots \\ \mathcal{C}_k \mathcal{A}_k^{H_c} \\ \mathcal{C}_k \mathcal{A}_k^{H_c+1} \\ \vdots \\ \mathcal{C}_k \mathcal{A}_k^{H_p} \end{bmatrix}}_{\Psi_k} \hat{\xi}(k)$$

$$+ \underbrace{\begin{bmatrix} \mathcal{C}_k \mathcal{B}_k & \mathcal{D}_k & \dots & 0 \\ \mathcal{C}_k \mathcal{A}_k \mathcal{B}_k & \mathcal{C}_k \mathcal{B}_k & \mathcal{D}_k & \vdots \\ \vdots & \vdots & \ddots & \ddots \\ \mathcal{C}_k \mathcal{A}_k^{H_c-1} \mathcal{B}_k & \mathcal{C}_k \mathcal{A}_k^{H_c-2} \mathcal{B}_k & \dots & \mathcal{C}_k \mathcal{B}_k \\ \mathcal{C}_k \mathcal{A}_k^{H_c} \mathcal{B}_k & \mathcal{C}_k \mathcal{A}_k^{H_c-1} \mathcal{B}_k & \dots & \mathcal{C}_k \mathcal{A}_k \mathcal{B}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}_k \mathcal{A}_k^{H_p-1} \mathcal{B}_k & \mathcal{C}_k \mathcal{A}_k^{H_p-2} \mathcal{B}_k & \dots & \mathcal{C}_k \mathcal{A}_k^{H_p-H_c} \mathcal{B}_k \end{bmatrix}}_{\Theta_k}$$

$$\times \underbrace{\begin{bmatrix} \Delta r(k|k) \\ \vdots \\ \Delta r(k+H_c-1|k) \end{bmatrix}}_{\Delta \mathcal{R}(k)} \quad (29)$$

using only the corresponding  $C_{ky}$  and  $D_{ky}$  matrices in expression (22). The prediction for these outputs has the form

$$\mathcal{Y}(k) = \Psi_{ky} \hat{\xi}(k) + \Theta_{ky} \Delta \mathcal{R}(k) \quad (30)$$

Substituting the predicted output in Eq. (30) into the cost function of Eq. (22), we get a quadratic expression in terms of the control changes  $\Delta \mathcal{R}(k)$ :

$$J(k) = \Delta \mathcal{R}(k)^T \mathcal{H}_k \Delta \mathcal{R}(k) - \Delta \mathcal{R}(k)^T \mathcal{G}_k + \text{const} + \rho \varepsilon \quad (31)$$

where

$$\begin{aligned} \mathcal{H}_k &= \Theta_{ky}^T Q_e \Theta_{ky} + R_e, & \mathcal{G}_k &= 2\Theta_{ky}^T Q_e \mathcal{E}(k) \\ \text{const} &= \mathcal{E}^T(k) Q_e \mathcal{E}(k) \end{aligned}$$

and  $\mathcal{E}(k)$  is defined as a tracking error between the future target trajectory and the free response of the system, that is,  $\mathcal{E}(k) = \mathcal{Y}_{\text{ref}}(k) - \Psi_{ky} \hat{\xi}(k)$ .  $Q_e$  and  $R_e$  are block diagonal matrices of appropriate dimensions with  $Q$  and  $R$  on the main diagonal, respectively. (These could be chosen parameter dependent also.)

Using the linear prediction model in Eq. (29), all of the constraints in Eqs. (24) and (28) can be posed as linear constraints on the optimization variables  $\Delta \mathcal{R}$  and  $\varepsilon$ . Finally, the QP to be solved at each time step has the following form:

$$\begin{aligned} \min_{\Delta \mathcal{R}, \varepsilon} & \Delta \mathcal{R}^T \mathcal{H}_k \Delta \mathcal{R} + \Delta \mathcal{R}^T \mathcal{G}_k + \text{const} + \rho \varepsilon \\ \text{s.t.} & \begin{bmatrix} \Omega_{k,\text{hard}} \\ \Omega_{k,\text{soft}} \end{bmatrix} \Delta \mathcal{R} \leq \begin{bmatrix} \omega_{k,\text{hard}} \\ \omega_{k,\text{soft}} \end{bmatrix} + \begin{bmatrix} 0 \\ \varepsilon \end{bmatrix} \\ & 0 \leq \varepsilon \end{aligned} \quad (32)$$

Note: The optimization yields  $\Delta \tilde{V}_{\text{cmd}}, \Delta \tilde{\chi}_{\text{cmd}}, \Delta \tilde{h}_{\text{cmd}}$  values, which were integrated to get  $\tilde{V}_{\text{cmd}}, \tilde{\chi}_{\text{cmd}}, \tilde{h}_{\text{cmd}}$ . The obtained  $\tilde{h}_{\text{cmd}}$  value was further integrated to get  $\tilde{h}_{\text{cmd}}$ . Trim values were added to arrive at  $V_{\text{cmd}}, \dot{\chi}_{\text{cmd}}, h_{\text{cmd}}$  values, which could be directly implemented on the T-33 autopilot.

### 1. Remarks

The problem formulation in the preceding section is a natural extension of a fixed LTI model-based RHC. The prediction at a certain time step is based on a linear model that best describes the plant at the actual flight condition, assuming that flight-condition-dependent linear models are available for prediction. A fixed LTI model is used over the entire prediction horizon, but it is updated according to the values of the scheduling parameters  $\rho(k)$  every time the horizon is propagated, and the optimization is resolved based on new measurement data. This approach leads to the QP problem in Eq. (32) and the state matrices  $A_k, B_k, C_k, D_k$  describing the internal model change in each implementation cycle. Usually a flight-condition-dependent description of the plant dynamics can be obtained either by freezing the scheduling parameters of a quasi-LPV model<sup>16</sup> or interpolating over a database of linearized models. In other cases the nonlinear prediction model is simple enough to lend itself to analytic linearization, while still retaining a reasonable prediction accuracy. This latter approach was followed in our control solution.

We note that if an accurate prediction of the parameters that the linear models depend on is available, this would allow for the prediction model to vary over the prediction horizon. The optimization problem could still be formulated as a quadratic program using different state matrices of the internal model at each time step. Obtaining a reasonable prediction of the scheduling parameters is not always easy; one could experiment with solving the problem first with the fixed LTI model based RHC method and use the solution as the prediction for the scheduling parameters. Our investigations indicate that this extra effort does not lead to significant improvement for the specific application example and horizon lengths considered. Moreover, even though the optimization problem complexity is retained the additional computational overhead could undermine real-time implementation of these ideas if the parameter-dependent

models are calculated using interpolation over a collection of linear systems. Theoretical aspects of using LTV models for prediction in an RHC-based tracking strategy are treated in Ref. 17. Treatment of sufficient stability conditions for LPV model based RHC goes beyond the scope of this paper and can be found in Refs. 18–20. These techniques typically make use of linear matrix inequalities, and the results are published in Ref. 21. A practical application example is reported in Ref. 22.

### C. RHC Reconfiguration Based on FDI Output

A detailed description of the fault-detection (FD) filter design for the flight demonstration can be found in Refs. 9, 23, and 24. For the purpose of describing the RHC reconfiguration process based on FD output, it suffices to say that at some point in the flight experiment a simulated aileron actuator fault is inserted. This is implemented in the flight control software by corrupting the turn rate command output of the controller with the output of a fault model. The corrupted control signal is then sent to the healthy aircraft; however, for the controller it appears as if a fault has occurred in the aircraft. The fault-detection filter was designed to detect this artificially inserted fault when the aircraft reaches a certain segment of the reference trajectory designed to excite the lateral fault dynamics and facilitate the detection process.

The fault model is depicted in Fig. 10, which illustrates how the turn rate control command is corrupted in the fault scenario.

The receding horizon controller was reconfigured when the threshold residual of the detection filter indicated a fault after the insertion of the simulated fault model at the appropriate segment of the flight experiment. The reconfiguration process involved two major changes to the RHC controller. The prediction model was updated and augmented with additional dynamics to reflect the faulty behavior of the aircraft. At the same time, more restrictive constraints were put on the turn rate control command to counteract the corruption caused by the fault model and represent a less aggressive control strategy. The  $u$  control signal limits given in Eq. (26) were updated accordingly using  $\pm 0.027$  rad/s turn rate limits associated with the faulty model.

A description of modifications to the prediction model based on incorporation of the fault dynamics follows. Based on modeling considerations described in Ref. 9, the turn rate  $\dot{\chi}$  input to heading  $\chi$  output channel of DemoSim was identified as a single-input single-output (SISO) system, decoupled from other inputs and outputs. This means that the state matrices describing the identified DemoSim dynamics have the following structure:

$$\begin{aligned} A_{\text{DS}} &= \begin{bmatrix} A_{VV} & 0 & A_{Vh} \\ 0 & A_{\chi\dot{\chi}} & 0 \\ A_{\gamma V} & 0 & A_{\gamma h} \end{bmatrix}, & B_{\text{DS}} &= \begin{bmatrix} B_{VV} & 0 & B_{Vh} \\ 0 & B_{\chi\dot{\chi}} & 0 \\ B_{\gamma V} & 0 & B_{\gamma h} \end{bmatrix} \\ C_{\text{DS}} &= \begin{bmatrix} C_{VV} & 0 & C_{Vh} \\ 0 & C_{\chi\dot{\chi}} & 0 \\ C_{\gamma V} & 0 & C_{\gamma h} \end{bmatrix}, & D_{\text{DS}} &= \begin{bmatrix} D_{VV} & 0 & D_{Vh} \\ 0 & D_{\chi\dot{\chi}} & 0 \\ D_{\gamma V} & 0 & D_{\gamma h} \end{bmatrix} \end{aligned} \quad (33)$$

Because of this decoupled structure, the faulty lateral DemoSim dynamics can be represented by the interconnection shown in Fig. 11, where  $A_f, B_f, C_f, D_f$  stand for the state matrices of the SISO fault model. The faulty lateral dynamics can be described by the following

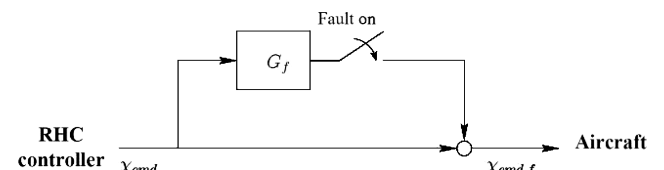


Fig. 10 Fault injection diagram.



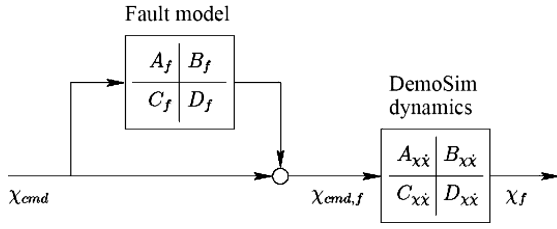


Fig. 11 Lateral DemoSim dynamics after fault activation.

state matrices based on the interconnection in Fig. 11:

$$A_{\chi\dot{\chi},f} = \begin{bmatrix} A_{\chi\dot{\chi}} & B_{\chi\dot{\chi}}C_f \\ 0 & A_f \end{bmatrix}, \quad B_{\chi\dot{\chi},f} = \begin{bmatrix} B_{\chi\dot{\chi}}(1 + D_f) \\ B_f \end{bmatrix}$$

$$C_{\chi\dot{\chi},f} = [C_{\chi\dot{\chi}} \quad D_{\chi\dot{\chi}}C_f], \quad D_{\chi\dot{\chi},f} = [D_{\chi\dot{\chi}}(1 + D_f)] \quad (34)$$

Using this representation, we can construct the modified prediction model incorporating the faulty lateral dynamics by replacing  $A_{\chi\dot{\chi}}, B_{\chi\dot{\chi}}, C_{\chi\dot{\chi}}, D_{\chi\dot{\chi}}$  in the DemoSim state matrices with  $A_{\chi\dot{\chi},f}, B_{\chi\dot{\chi},f}, C_{\chi\dot{\chi},f}, D_{\chi\dot{\chi},f}$ . The obtained faulty state matrices of  $A_{DS,f}, B_{DS,f}, C_{DS,f}, D_{DS,f}$  are then used to build the complete prediction model as described in Eqs. (13–15).

#### D. Flight Envelope Limits as Output Constraints

Flight envelope limits were characterized in Eq. (8) of Sec. II.B as a collection of linear inequalities involving  $V, \dot{\chi}, \dot{\gamma}$ . This means that predictions for these variables are needed in order to implement these constraints in the RHC framework.

State matrices of the original linear discrete-time identified DemoSim model can be used to define the additional prediction model outputs that are needed for constraint specifications. The identified DemoSim model is described with state matrices  $A_{DS}, B_{DS}, C_{DS}, D_{DS}$  corresponding to  $\tilde{V}, \dot{\chi}, \dot{\gamma}$  outputs that represent deviations from trim values. Specifying a true ground speed  $V$  output of the complete prediction model requires the addition of the trim value to the identified LTI DemoSim model output variable:  $V = V_{DStrim} + \tilde{V}$ . Since the identified model has  $\chi$  and  $\gamma$  outputs, their derivatives can be approximated using the discrete-time state matrices in the following way. Assume that a discrete-time LTI system is described with state matrices  $A, B, C, D$ . If the  $D$  matrix is all zero, then the derivative of the output signal  $y$  can be approximated by

$$\dot{y} \sim T_s^{-1} \Delta y = T_s^{-1} (y_{k+1} - y_k) = T_s^{-1} C(A - I)x_k + T_s^{-1} C B u_k \quad (35)$$

where  $T_s$  is the sampling time. Because the  $D_{DS}$  matrix of the identified DemoSim model is all zero, we can apply this method to formulate new  $\dot{\chi}, \dot{\gamma}$  outputs using the original identified state matrices.

Partition the original  $C_{DS}, D_{DS}$  identified DemoSim matrices with respect to the individual outputs in the following way:

$$C_{DS} = \begin{bmatrix} C_{DS,v} \\ C_{DS,\chi} \\ C_{DS,\gamma} \end{bmatrix}, \quad D_{DS} = \begin{bmatrix} D_{DS,v} \\ D_{DS,\chi} \\ D_{DS,\gamma} \end{bmatrix} \quad (36)$$

Then form the matrices

$$C_{DS}^o = \begin{bmatrix} C_{DS,v} \\ T_s^{-1} C_{DS,\chi} (A_{DS} - I) \\ T_s^{-1} C_{DS,\gamma} (A_{DS} - I) \end{bmatrix}, \quad D_{DS}^o = \begin{bmatrix} D_{DS,v} \\ T_s^{-1} C_{DS,\chi} B_{DS} \\ T_s^{-1} C_{DS,\gamma} B_{DS} \end{bmatrix} \quad (37)$$

to create new augmented DemoSim matrices of

$$A_{DS0}^o = A_{DS0}, \quad B_{DS0}^o = B_{DS0}$$

$$C_{DS0}^o = \begin{bmatrix} C_{DS0} \\ C_{DS}^o \quad 0 \end{bmatrix}, \quad D_{DS0}^o = \begin{bmatrix} D_{DS0} \\ D_{DS}^o \end{bmatrix} \quad (38)$$

which now have six outputs of the variables

$$[\tilde{V}_0 \quad \tilde{\chi}_0 \quad \tilde{\gamma}_0 \quad \tilde{V} \quad \dot{\chi} \quad \dot{\gamma}]^T$$

including the newly defined last three. The matrices in Eq. (38) will replace the original  $A_{DS0}, B_{DS0}, C_{DS0}, D_{DS0}$  matrices of Eq. (13) in constructing the complete prediction model. This means that the matrices  $A_d, B_d, C_d, D_d$  are constructed exactly as in Eq. (14), but using the newly defined  $A_{DS0}^o, B_{DS0}^o, C_{DS0}^o, D_{DS0}^o$  from Eq. (38). The complete prediction model state matrices are then formed by

$$A^o = \begin{bmatrix} A_d & 0 \\ \tilde{B}C_{d1} & \tilde{A} \end{bmatrix}, \quad B^o = \begin{bmatrix} B_d & 0 \\ \tilde{B}D_{d1} & 0 \end{bmatrix}$$

$$C^o = \begin{bmatrix} 0 & I_3 \\ A_z C_{d2} & 0 \end{bmatrix}, \quad D^o = \begin{bmatrix} 0 & I_3 \\ A_z D_{d2} & 0 \end{bmatrix} \quad (39)$$

where  $C_{d1}, D_{d1}$  correspond to the original three outputs and  $C_{d2}, D_{d2}$  to the newly created second three outputs of the matrices  $C_d, D_d$ . The complete prediction model of Eq. (39) describes the dynamic relationship between the control  $r$  and disturbance  $d$  inputs and the tracking  $y$  and soft constraint  $z$  outputs as defined in Eq. (16):

$$x_{k+1} = A^o x_k + B^o \begin{bmatrix} r_k \\ d_k \end{bmatrix}, \quad \begin{bmatrix} y_k \\ z_k \end{bmatrix} = C^o x_k + D^o \begin{bmatrix} r_k \\ d_k \end{bmatrix} \quad (40)$$

The flight envelope limits can then be represented by lower and upper limits on the  $z$  output signals

$$\underline{z} = -\infty, \quad \bar{z} = b_z - A_z \begin{bmatrix} V_{DStrim} \\ 0 \\ 0 \end{bmatrix} \quad (41)$$

to be used in the soft constraint formulation of Eq. (28).

## IV. Simulation and Experimental Results

### A. Demonstration Scenarios

Figure 12 shows a bird's eye view of the flight-test range and the basic demonstration scenario elements. The timeline of events starts with engaging the RHC controller in a prespecified area near the ingress point once the starting conditions are met. The controller tracks a time-stamped position reference trajectory while respecting constraints on the vehicle dynamics. At a certain point along the reference trajectory, a pop-up threat can be invoked by a ground operator, which results in a switch to an alternative reference trajectory that avoids the threat. After the target is reached with a specified heading, a simulated fault was inserted into the system, which is detected shortly thereafter at a trajectory segment designed specifically for this purpose. After detection, the fault is removed and the aircraft returned to the egress point.

In a second, more ambitious scenario, the fault is not removed from the system after detection, and the RHC controller is reconfigured to adapt to the faulty vehicle dynamics. Constraints are also adjusted to restrict the aircraft's maneuvers as described in Sec. III.D. Description of the fault-detection filter design and test results are omitted in this paper and can be found in Refs. 9, 23, and 24.

The receding-horizon guidance controller was tested in simulation with different wind conditions and showed excellent robustness. Figure 13 shows how the true airspeed was adjusted according to wind conditions to achieve the necessary ground speed required for accurate tracking of the position reference. Constraint enforcement under various wind conditions is also demonstrated in this figure by the saturating turn rate command. The main limitations of good performance at higher wind velocities are the flight envelope constraints.

The top two plots in Fig. 14 show simulated tracking performance in the north-east coordinate frame and in terms of altitude under different wind conditions. These simulations were performed in

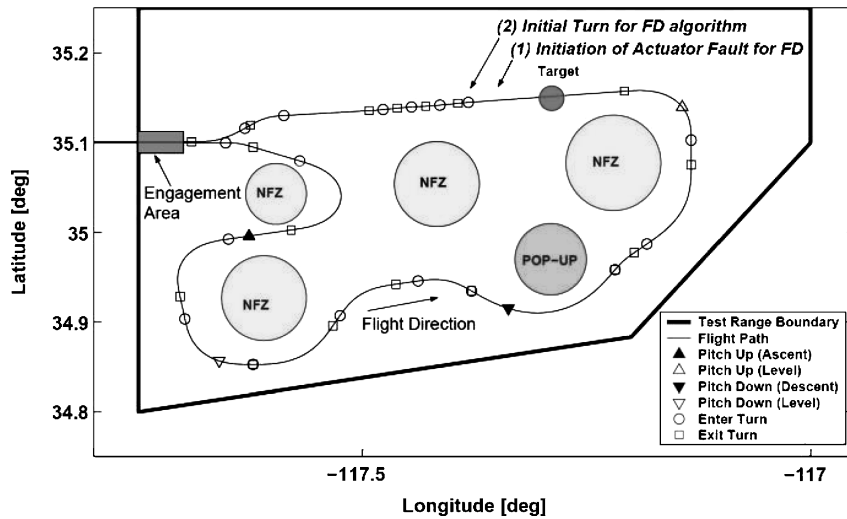


Fig. 12 Illustration of the reference trajectory in the flight-test area with the target, pop-up threat, and no-fly-zone locations of the experiment scenario.

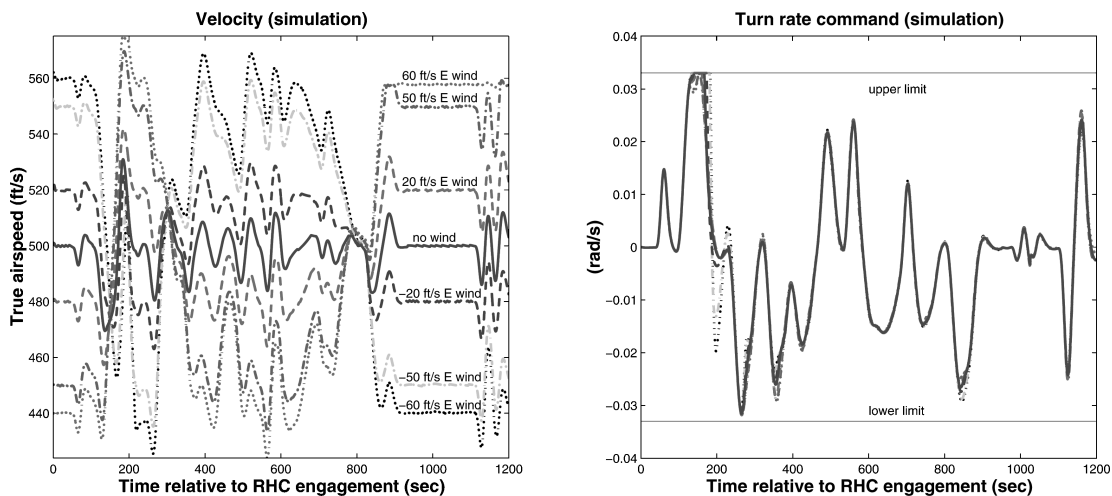


Fig. 13 True airspeed and turn rate command simulation results under different RHC wind conditions.

MATLAB<sup>®</sup> using the high-fidelity DemoSim nonlinear executable model.

### B. Hardware-in-the-Loop Simulations

Besides coordinating and executing the flight tests, the Boeing Company was also responsible for integrating the individual flight experiments and controller code developed for DemoSim and OCP by the different teams participating in the final demo. Preparations for the flight tests culminated in a final system check and testing phase of code development, which included hardware-in-the-loop (HIL) testing of flight experiments in a realistic, high-fidelity simulation environment at the Boeing facilities, in St. Louis, Missouri.

The hardware-in-the-loop simulation environment relied on high-fidelity nonlinear closed-loop aircraft dynamics represented by the DemoSim model. It also made use of the actual avionics pallet flown in the flight tests with simulated sensors and electronic command data processing.

The middle two plots in Fig. 14 show HIL tracking performance in the north-east coordinate frame and in terms of altitude.

### C. Experimental Flight-Test Results

The flight tests took place at Edwards Air Force Base in the Mojave desert in June 2004. Several universities and aerospace companies were involved in the two-week long flight demonstrations of

the DARPA SEC project, each having their own experimental flight scenario to be tested. Because of difficulties with asset scheduling at the base, our team eventually had only two flight tests that could be evaluated.

The bottom two plots in Fig. 14 illustrate the flight-test results. The online optimization-based guidance algorithm required not more than 50–100-ms computational time in each cycle of the 2-Hz sampling rate, leading to a 5–10 times faster execution than real time. The main reason for deviations from the reference trajectory and degraded tracking performance during flight test was that automatic speed control was not available on the test platform. Airspeed was controlled using manual adjustments to the throttle by the pilot, who was cued by a three-state light-emitting-diode (LED) indicator whether to increase, decrease, or maintain the velocity of the aircraft based on commanded and actual speed measurements. The dead zone of the “maintain speed” status indicator was approximately 30 ft/s. Another factor influencing the outcome of experiments was that flight tests were conducted in the presence of strong winds (25–30 kn).

Figure 15 suggests an average delay of 50–100 s in the velocity command channel, which was not modeled in the RHC controller. The controller was tested only up to 10–20 samples (5–10 s) of unmodeled additional delay compared to the prediction model and showed acceptable degradation of performance.

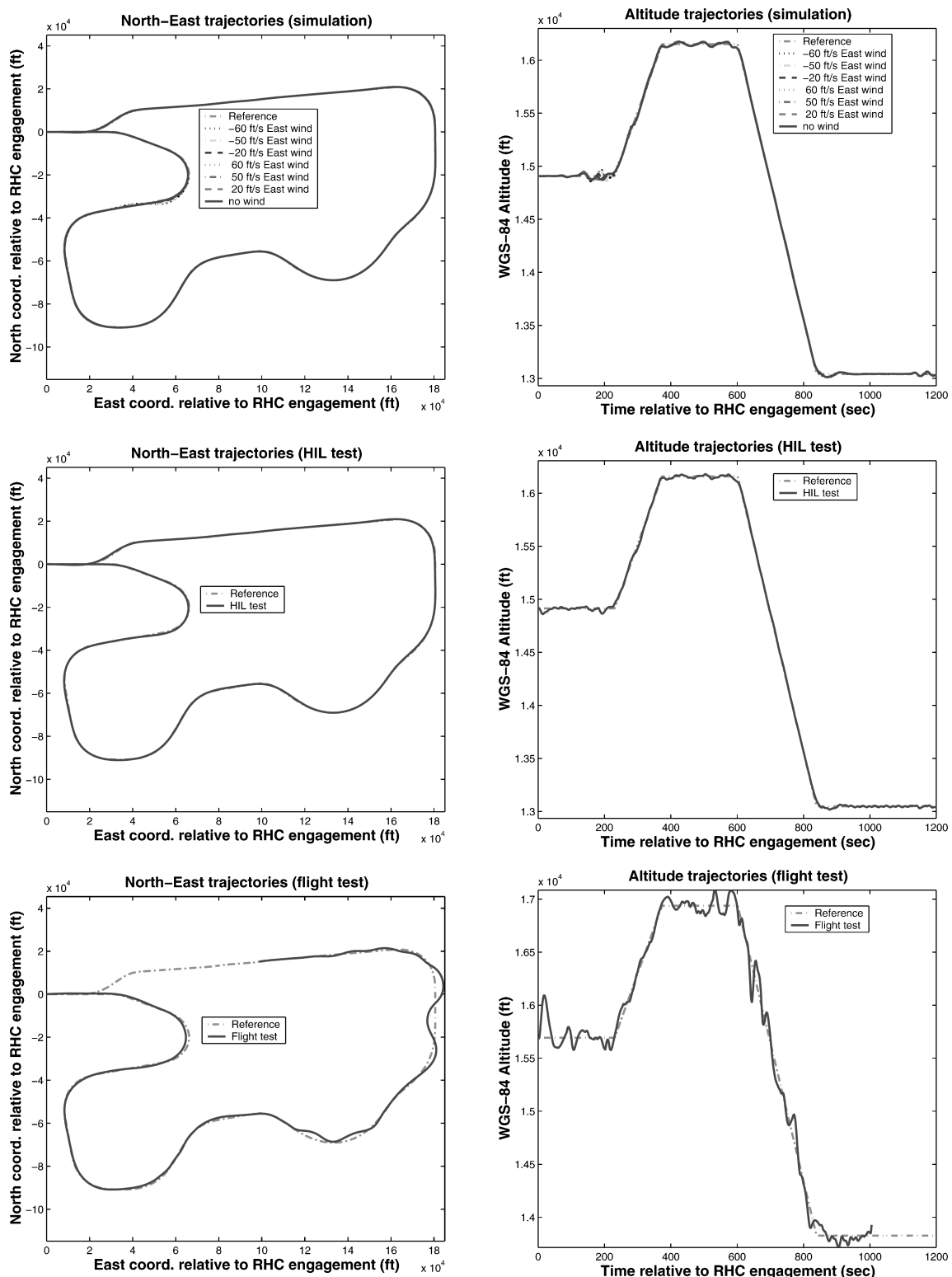


Fig. 14 North, east, and altitude tracking performance. The top two plots show simulation results under different wind conditions, the middle two plots depict a hardware-in-the-loop test run and the bottom two plots represent flight-test data.

The “jagged,” sawtooth-like nature of the velocity signal in the HIL test run is as a result of the pilot model embedded into DemoSim, which represents pilot reaction to speed commands by a quantizer, which is probably acting on speed tracking error. This pilot quantization effect could be switched off in the MATLAB simulation environment, so that this phenomenon was not present in the results presented in the top plot of Fig. 15. The HIL test environment was constructed in such a way that this quantization effect could not be switched off.

#### D. Reconfiguration Based on Fault Detection

A second, more ambitious experiment was constructed to test RHC controller reconfiguration based on fault detection and to provide a longer, more versatile reference trajectory for evaluating the guidance controller. This second scenario is conducted exactly the same way as the basic scenario until the point of fault insertion, where the fault is not removed from the system after detection. Instead, the prediction model and constraints of the RHC controller are updated as described in Sec. III.C to adapt to the faulty vehicle

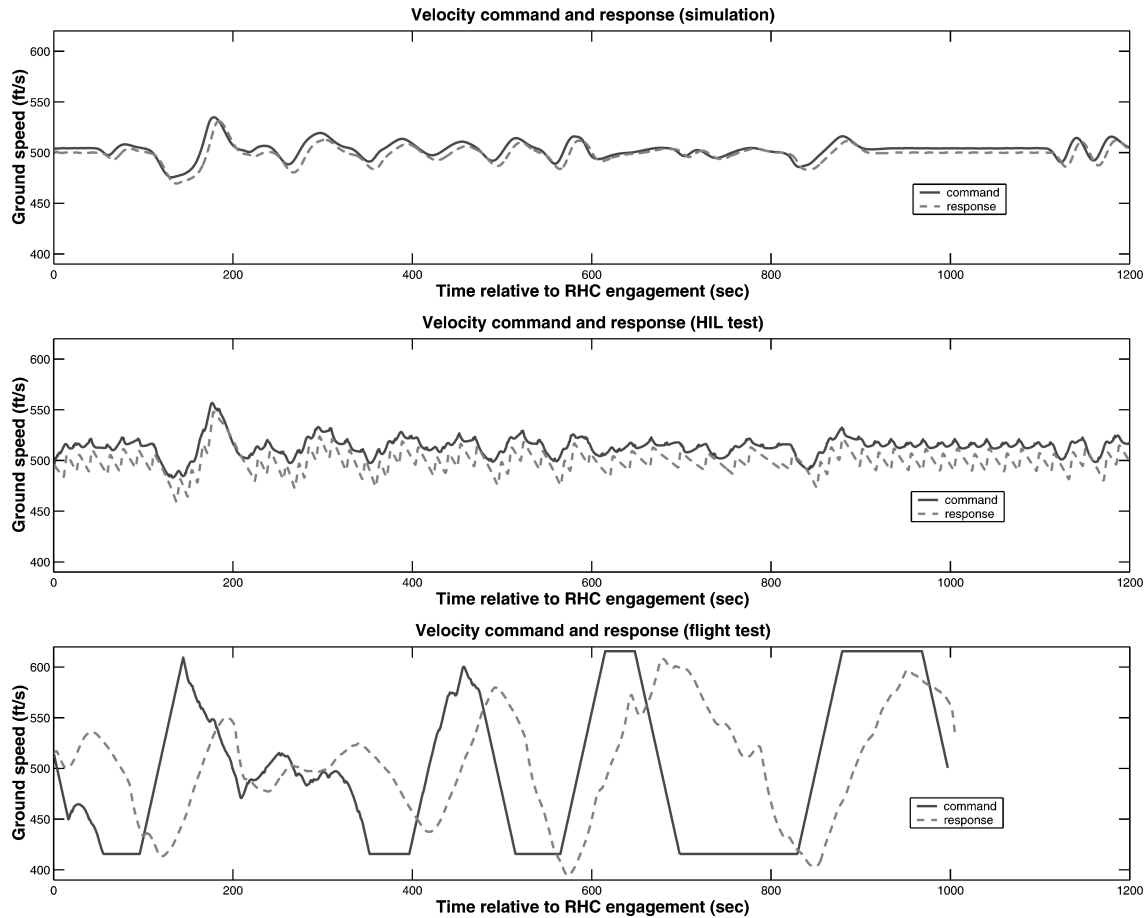


Fig. 15 Commanded and actual ground speed in simulations, hardware-in-the-loop test, and during the flight test.

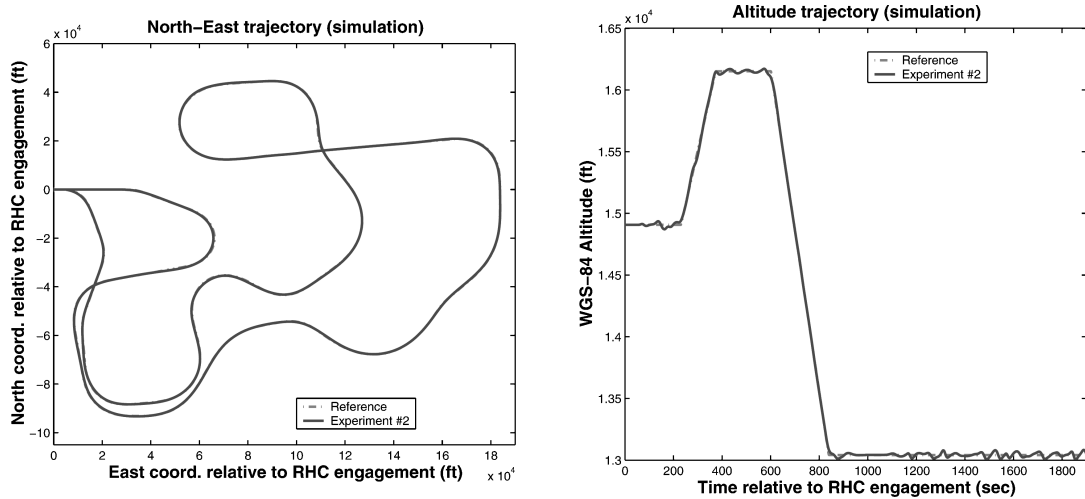


Fig. 16 Simulated north, east, and altitude tracking performance in the second scenario involving RHC reconfiguration.

dynamics and restrict the aircraft’s maneuvers. Unfortunately, this experiment could not be flight tested because of difficulties with asset scheduling at the base and shortage of time. This section presents simulation results conducted using the high-fidelity DemoSim model.

Figure 16 shows position trajectories of the second scenario simulation indicating good tracking performance after a successful reconfiguration of the RHC controller.

Figure 17 illustrates one effect of reconfiguration by plotting the turn rate command output of the RHC controller, which was corrupted by the output of the fault simulator, and the faulty command sent to DemoSim. Note that bounds on turn rate command

became more restrictive after fault detection, as part of the controller reconfiguration. Note also that constraints were formulated for the RHC controller output and not the actual autopilot command, which includes the corruption by the fault simulator. This implies that although the RHC controller restricted its turn rate commands to respect the specified tighter limits the actual transient commands, because of the additive effect of the fault model, could reach values larger than the constraint limits. This is simply a consequence of the chosen fault modeling, and does not have any practical significance. If the fault model represents an actual physical system component, then soft output constraints could be specified on the corrupted control signal as well.

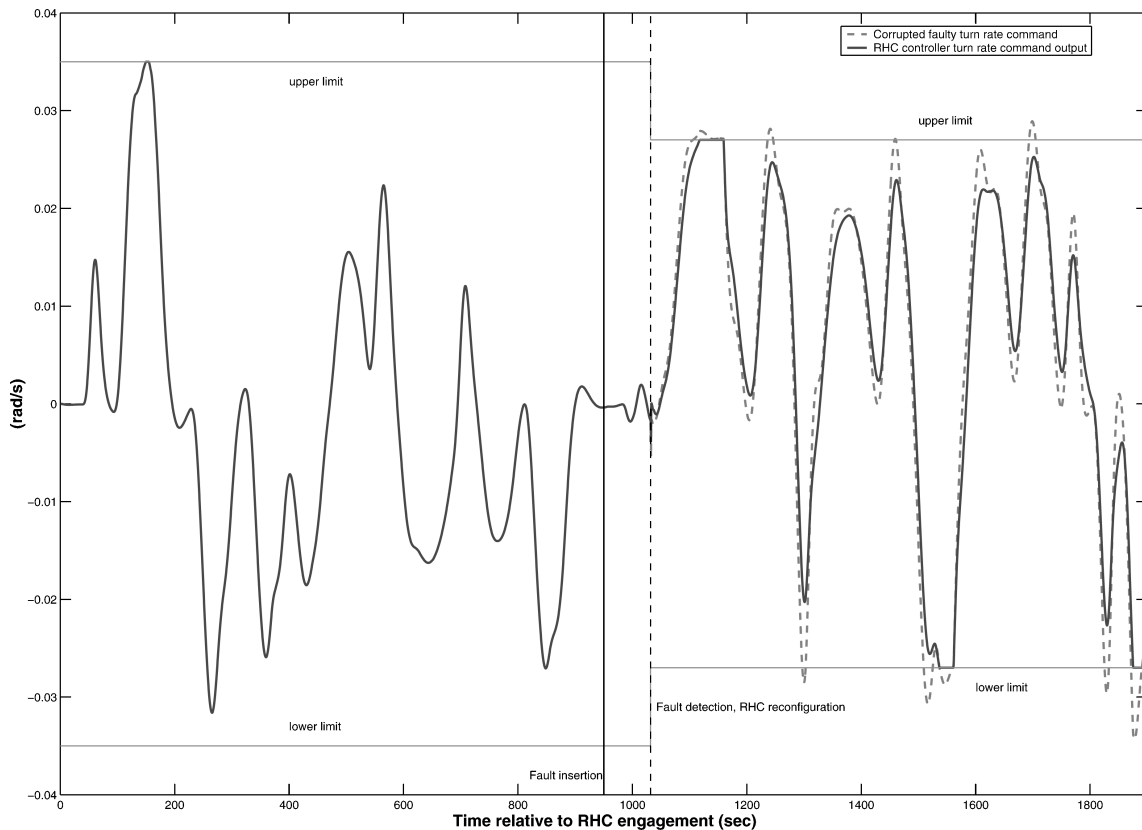


Fig. 17 Commanded and corrupted (faulty) turn rate commands in the second scenario. RHC reconfiguration includes updating turn rate command constraints after fault detection.

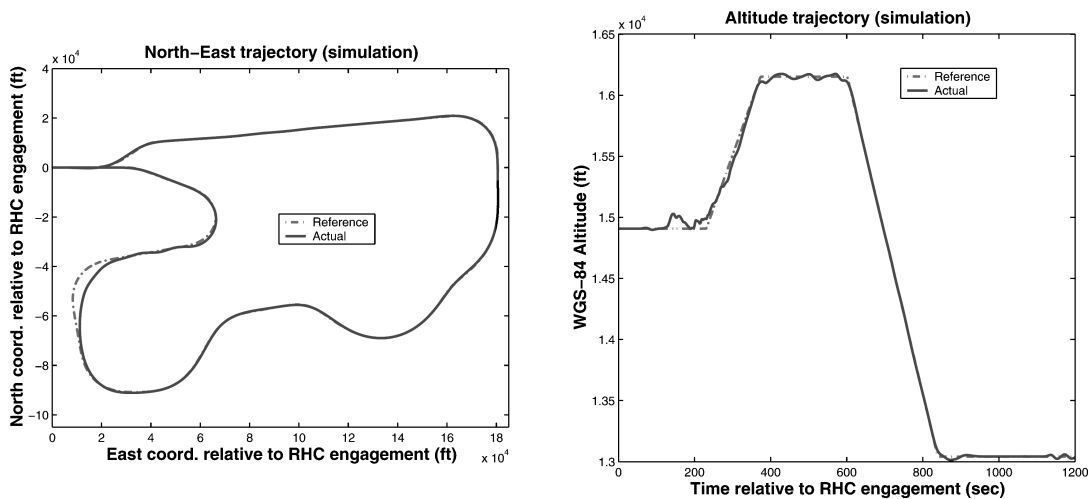


Fig. 18 North, east, and altitude tracking performance with linearized maneuvering limits as soft constraints.

**E. Flight Envelope Limits as Output Constraints**

This section presents simulation results using maneuvering and flight envelope limits characterized in Sec. II.B as soft output constraints in the RHC formulation according to the description provided in Sec. III.D.

Figure 18 shows north, east, and altitude position trajectories, which indicate poorer tracking accuracy compared to the performance obtained without output constraint enforcement. The reason is that the inner polyhedral approximation of such constraints results in more restrictive limits on bank angle and consequently on turn rate command than before. The reference trajectory includes tighter turns than what these new linearized approximate limits allow, so that the guidance controller is unable to perform very precise tracking given these restrictions on maneuverability. On the other

hand, the controller tries to keep up with the time-stamped position reference by velocity control and accelerating to “catch up” with the reference point that “got ahead” of the aircraft in the tighter turns.

Maneuvering and flight envelope constraint enforcement is illustrated in Fig. 19 by plotting the aircraft trajectory in the output constraint space defined by  $V, \dot{\chi}, \dot{\gamma}$ . These plots show the nonlinear feasible region from different viewing angles together with its inner polyhedral approximation as introduced in Sec. II.B. The aircraft trajectory was plotted in terms of the constraint-space coordinate axes and overlaid on these images to show that the soft output constraints are respected. Slight violations can be noticed on the facets of the constraint-polyhedron corresponding to bounds on bank angle. These violations are possible because of the soft constraint formulation.

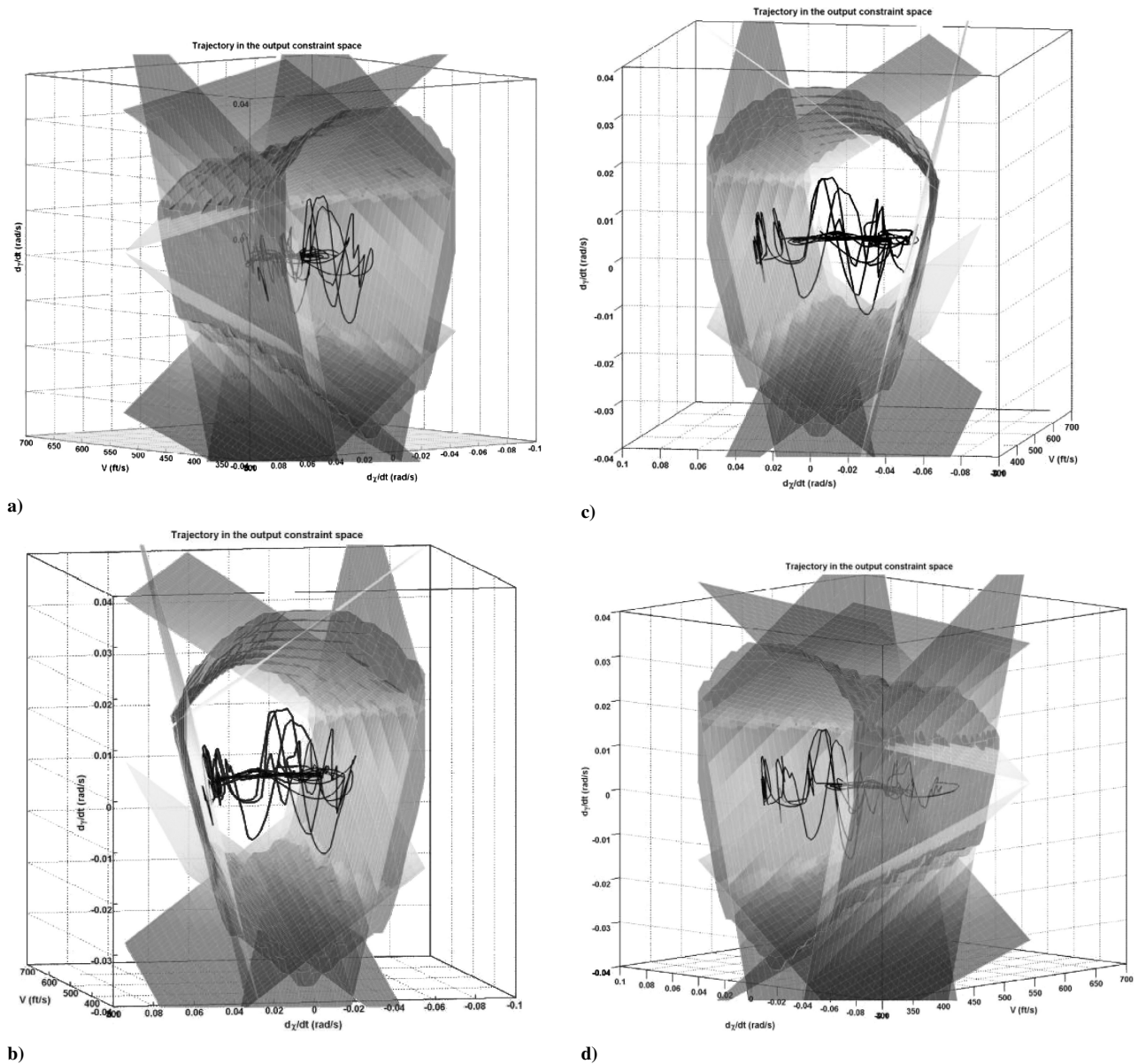


Fig. 19 Output constraints.

## V. Conclusions

The RHC-based guidance system showed amenable robustness properties in spite of the dramatic difference between the velocity tracking behavior of the prediction model and the real system. Performance analysis of flight-test data and simulation results with varying wind conditions suggest that the RHC guidance law would have excellent tracking performance using an autonomous speed control system. Further high-fidelity simulations showed that output constraints could be also accommodated using soft constraint formulation. This, along with the observed successful reconfiguration experiments, suggests that the proposed approach provides a high-performance, versatile guidance technology for future unmanned aircraft systems.

## Acknowledgments

The authors gratefully acknowledge discussions with George Pappageorgiou at Honeywell Labs on the particular receding horizon control formulation to be used in the final flight test and derivation of approximate nonlinear flight envelope limits. The polyhedral constraint characterization was obtained with the instrumental help of Ryan Ingvalson.

The authors would also like to acknowledge Brian Mendel, Tim Espey and Jared Rosson at the Boeing Company for their help and

support with the integration of the RHC API into the Open Control Platform.

This work was funded by the Defense Advanced Research Projects Agency under the Software Enabled Control program, with John Bay as program manager. The contract number is USAF/AFMC F33615-99-C-1497; Dale Van Cleave is the technical contract monitor.

## References

- <sup>1</sup>Office of the Under Secretary of Defense for Acquisition Technology and Logistics, "Unmanned Aerial Vehicles Roadmap 2002-2027," Tech. Rept. A809414, Dept. of Defense, Washington, DC, Dec. 2002.
- <sup>2</sup>Calise, A. J., and Rysdyk, R. T., "Nonlinear Adaptive Flight Control Using Neural Networks," *IEEE Control Systems Magazine*, Vol. 18, No. 6, 1998, pp. 14–25.
- <sup>3</sup>Calise, A., Lee, S., and Sharma, M., "Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft," AIAA Paper 1998-4108, Aug. 1998.
- <sup>4</sup>Brinker, J. S., and Wise, K. A., "Flight Testing of Reconfigurable Control Law on the X-36 Tailless Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 903–909.
- <sup>5</sup>Eberhardt, R. L., and Ward, D. G., "Indirect Adaptive Flight Control of a Tailless Fighter Aircraft," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, 1999, pp. 466–476.
- <sup>6</sup>Bay, J. S., and Heck, B. S., "Special Section: Software-Enabled Control," *IEEE Control Systems Magazine*, Vol. 23, No. 1, Feb. 2003.

- <sup>7</sup>Paunicka, J., Mendel, B., and Corman, D., "The OCP—An Open Middleware Solution for Embedded Systems," *Proceedings of the American Control Conference*, Vol. 5, Inst. of Electrical and Electronics Engineers, IEEE Publications, Piscataway, NJ, 2001, pp. 3345–3350.
- <sup>8</sup>Keviczky, T., and Balas, G. J., "Receding Horizon Control of an F-16 Aircraft: A Comparative Study," *Control Engineering Practice* (to be published).
- <sup>9</sup>Keviczky, T., Ingvalson, R., Rotstein, H., Packard, A., Natale, O. R., and Balas, G. J., "An Integrated Multi-Layer Approach to Software Enabled Control: Mission Planning to Vehicle Control," Dept. of Aerospace Engineering and Mechanics, Univ. of Minnesota, Tech. Rept., Minneapolis; Dept. of Mechanical Engineering, Univ. of California, Berkeley, Nov. 2004.
- <sup>10</sup>Camacho, E. F., and Bordons, C., "Model Predictive Control in the Process Industry," *Advances in Industrial Control*, Springer-Verlag, London, 1995.
- <sup>11</sup>Bemporad, A., and Morari, M., "Robust Model Predictive Control: A Survey," *Robustness in Identification and Control*, Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, 1999, pp. 207–226.
- <sup>12</sup>Allgöwer, F., and Zheng, A. Z., "Nonlinear Model Predictive Control: Assessment and Future Directions for Research," *Progress in Systems and Control Series*, Birkhäuser Verlag, Basel, 2000.
- <sup>13</sup>Maciejowski, J. M., *Predictive Control with Constraints*, Prentice-Hall, Upper Saddle River, NJ, 2002.
- <sup>14</sup>Papageorgiou, G., Glover, K., Huzmezan, M., and Maciejowski, J. M., "A Combined MBPC/ $H_\infty$  Automatic Pilot for a Civil Aircraft," *Proceedings of the American Control Conference*, Vol. 1, Inst. of Electrical and Electronics Engineers, 1997, pp. 118–122.
- <sup>15</sup>Keviczky, T., Packard, A., Natale, O. R., and Balas, G. J., "Application Programming Interface for Real-Time Receding Horizon Control," *Proceedings of the 44th IEEE Conference on Decision and Control, European Control Conference*, IEEE Publications, Piscataway, NJ, 2005, pp. 1689–1694.
- <sup>16</sup>Huzmezan, M., and Maciejowski, J. M., "Reconfiguration and Scheduling in Flight Using Quasi-LPV High-Fidelity Models and MBPC Control," *Proceedings of the American Control Conference*, Vol. 6, Inst. of Electrical and Electronics Engineers, IEEE Publications, Piscataway, NJ, 1998, pp. 3649–3653.
- <sup>17</sup>Kim, K. B., "Implementation of Tracking Controls for Constrained Discrete Time-Varying Systems via Receding Horizon Strategy," *Proceedings of the 40th IEEE Conference on Decision and Control*, Vol. 5, Inst. of Electrical and Electronics Engineers, IEEE Publications, Piscataway, NJ, 2001, pp. 4883, 4884.
- <sup>18</sup>Giannelli, M., and Primbs, J. A., "An Analysis Technique for Optimization Based Control Applied to Quasi-LPV Plants," *Proceedings of the American Control Conference*, Vol. 3, Inst. of Electrical and Electronics Engineers, IEEE Publications, Piscataway, NJ, 2000, pp. 1909–1913.
- <sup>19</sup>Wada, N., Saito, K., and Saeki, M., "Model Predictive Control for Linear Parameter Varying Systems Using Parameter Dependent Lyapunov Function," *47th IEEE International Midwest Symposium on Circuits and System, MWSCAS'04*, Vol. 3, 2004, pp. iii-133–iii-136.
- <sup>20</sup>Lu, Y., and Arkun, Y., "Quasi-Min-Max MPC Algorithms for LPV Systems," *Automatica*, Vol. 36, No. 4, 2000, pp. 527–540.
- <sup>21</sup>Kothare, M. V., Balakrishnan, V., and Morari, M., "Robust Constrained Model Predictive Control Using Linear Matrix Inequalities," *Automatica*, Vol. 32, No. 10, 1996, pp. 1361–1379.
- <sup>22</sup>Jadbabaie, A., and Hauser, J., "Control of the Caltech Ducted Fan in Forward Flight: A Receding Horizon-LPV Approach," *Proceedings of the American Control Conference*, Vol. 2, Inst. of Electrical and Electronics Engineers, IEEE Publications, Piscataway, NJ, 2001, pp. 1333–1338.
- <sup>23</sup>Rotstein, H. P., Ingvalson, R., Keviczky, T., and Balas, G. J., "Input-Dependent Threshold Function for an Actuator Fault Detection Filter," *16th IFAC World Congress*, Elsevier, Amsterdam, 2005.
- <sup>24</sup>Ingvalson, R., Rotstein, H. P., Keviczky, T., and Balas, G. J., "Fault Detection Design for Uninhabited Aerial Vehicles," AIAA Paper 2005-6251, 2005.