A Generic Guidance and Control Structure for Six-Degree-of-Freedom Conceptual Aircraft Design

M. Christopher Cotting[*] and Timothy H. Cox[†]
NASA Dryden Flight Research Center, Edwards, California, 93523

A control system framework is presented for both real-time and batch six-degree-of-freedom simulation. This framework allows stabilization and control with multiple command options, from body rate control to waypoint guidance. Also, pilot commands can be used to operate the simulation in a pilot-in-the-loop environment. This control system framework is created by using direct vehicle state feedback with nonlinear dynamic inversion. A direct control allocation scheme is used to command aircraft effectors. Online *B*-matrix estimation is used in the control allocation algorithm for maximum algorithm flexibility. Primary uses for this framework include conceptual design and early preliminary design of aircraft, where vehicle models change rapidly and a knowledge of vehicle six-degree-of-freedom performance is required. A simulated airbreathing hypersonic vehicle and a simulated high performance fighter are controlled to demonstrate the flexibility and utility of the control system.

Nomenclature

a	=	angular acceleration, deg/sec^2
B	=	control matrix
BEST	=	*B*-matrix estimation algorithm
b	=	reference wing span, ft
C_l	=	nondimensional rolling moment
C_m	=	nondimensional pitching moment
C_n	=	nondimensional yawing moment
CV	=	control variable
CV_{cmd}	=	control variable commanded
$\bar{c}$	=	reference mean aerodynamic chord
D	=	vehicle total drag force, lbf
dB	=	decibels
deg	=	degrees
dt	=	integration time step, s
EOM solver	=	equations of motion solver
F_T	=	engine thrust force, lbf
g	=	acceleration constant due to gravity, ft/sec^2
h	=	altitude above sea level, ft
$\dot{h}$	=	change in altitude above sea level, ft
I	=	moment of inertia, ft^4
K	=	control system gain
K_B	=	control law response shaping gain
K_{in}	=	control law stick input gain
l	=	body axis rolling moment, $\text{ft} \cdot \text{lbf}$
L	=	lift, lbf
M	=	dimensional moment, $\text{ft} \cdot \text{lbf}$
m	=	body axis pitching moment, $\text{ft} \cdot \text{lbf}$

[*] Aerospace Engineer, Dynamics and Controls Branch, P.O. Box 273/Mail Stop 4840D, AIAA Member.

[†] Aerospace Engineer, Dynamics and Controls Branch, P.O. Box 273/Mail Stop 4840D.

NDI	=	nonlinear dynamic inversion
n	=	body axis yaw moment, $\text{ft} \cdot \text{lbf}$
PLA	=	power lever angle, percent
p	=	body axis roll rate, deg/sec
$\dot{p}$	=	body axis roll acceleration, deg/sec^2
p_w	=	wind axis roll rate, deg/sec
$\dot{p}_w$	=	wind axis roll acceleration, deg/sec^2
q	=	body axis pitch rate, deg/sec
$\dot{q}$	=	body axis pitch acceleration, deg/sec^2
$\overline{q}$	=	dynamic pressure, psf
r	=	body axis yaw rate, deg/sec
$\dot{r}$	=	body axis yaw acceleration, deg/sec^2
rad	=	radians
S	=	wing reference area, ft^2
SDI	=	simplified dynamic inversion
s	=	Laplace operator
sec	=	seconds
V	=	total vehicle velocity
$\dot{V}$	=	change in total vehicle velocity
WGS84	=	World Geodetic System 1984 - a worldwide datum
x	=	latitude command in flat plane frame, ft
$\dot{x}$	=	change in latitude command in flat plane frame, ft/sec
y	=	longitude command in flat plane frame, ft
$\dot{y}$	=	change in longitude command in flat plane frame, ft/sec
z	=	altitude command in flat plane frame, ft
6-DOF	=	six-degree-of-freedom
α	=	angle of attack, deg
$\dot{\alpha}$	=	change in angle of attack, deg/sec
β	=	angle of sideslip, deg
$\dot{\beta}$	=	change in angle of sideslip, deg/sec
γ	=	flightpath angle, deg
$\dot{\gamma}$	=	change in flightpath angle, deg/sec
μ	=	mass, lbm
ϕ	=	angle of roll (body axis), deg
$\dot{\phi}$	=	change in angle of roll (body axis), deg/sec
ϕ_w	=	angle of roll (wind axis), deg
ψ	=	heading angle, deg
$\dot{\psi}$	=	change in heading angle, deg/sec

I. Introduction

THE early stages of aircraft design are typically characterized by a highly conceptual approach. A set of requirements are defined and a system, or aircraft mold line, or propulsion arrangement (depending on the particular focus of interest) is established. Trade studies using analysis and simulation of the entire system are used to determine the best design to meet the requirements. Also typical of this stage of aircraft design are frequent configuration changes, which translate to frequent and significant changes to various subsystem models such as aerodynamics, propulsion, and actuators. Unfortunately, these characteristics of aircraft design can make using a six-degree-of-freedom (6-DOF) simulation cumbersome. Changes to the various subsystem models are often significant and require modification and sometimes redesign of the control laws. Yet, the frequent changes in the models provide little time to make these modifications.

Developing a generic set of control laws that are independent of some of the most influential models on aircraft dynamics, such as aerodynamics and propulsion, could eliminate some of the problems associated with using 6-DOF simulation in the early design phase. These control laws could allow a 6-DOF simulation to be used as a rapid prototyping tool, supplying usable control laws quickly and with minimal modification to any envisioned aircraft configuration. The simulation could then be quickly adapted to typical early design phase tasks such as actuator bandwidth sizing, aerodynamic uncertainty sensitivity studies, performance trade studies, and mission planning.

Dynamic inversion[1,2] is a control law methodology that can be developed into a desired generic set of control laws. Figure 1 shows the utility of this methodology across linearized aerodynamic models of aircraft ranging from lifting bodies (for example, X-38) to fighter aircraft to airbreathing hypersonic vehicles, as well as across flight conditions ranging from subsonic to hypersonic. The figure shows that regardless of the aircraft type or flight condition, the pitch rate response to a stick input is the same. This result occurs because the dynamic inversion methodology incorporates an aerodynamic model of the aircraft within its architecture. The model, which can be stable or unstable, is inverted and used to cancel the bare airframe dynamics, allowing the control law designer to define the desired dynamics. In this sense, the control laws and the desired vehicle response are independent of the aerodynamics, provided the model used in the control law algorithm is valid.

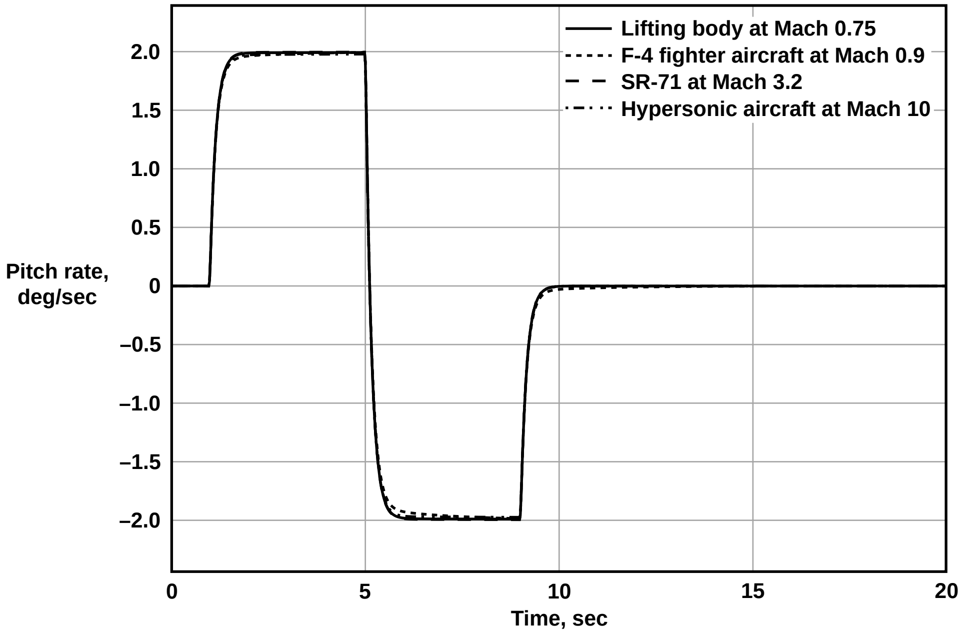

Figure 1. Response of step inputs to four different aircraft linearized aerodynamic models.

Researchers at the NASA Dryden Flight Research Center (Edwards, California) have recently evaluated generic dynamic inversion control laws in a 6-DOF simulation environment. Two types of dynamic inversion methodologies were investigated. One type, nonlinear dynamic inversion (NDI), incorporates full state, nonlinear equations[1,2] into the control law architecture. Although dynamic inversion is primarily an inner-loop function, outer-loop functions such as guidance in relation to the NDI were also investigated.[3]

The other type of dynamic inversion methodology investigated is a simplified approach to NDI. This simplified dynamic inversion approach (SDI)[4] utilizes state acceleration feedback and estimates of the surface effectiveness of the aerodynamics to perform the dynamic inversion cancellation of the bare airframe dynamics. An additional feature of SDI is that the control variable the pilot commands can be easily and quickly modified. The primary focuses of the SDI investigation were inner-loop dynamics and pilot-in-the-loop control.[5]

This report describes the generic implementation of the NDI into 6-DOF real-time simulations of an airbreathing hypersonic vehicle at a Mach 6 flight condition and a twin-engine carrier class high performance fighter aircraft at a subsonic flight condition. Included in the generic control law architecture is a direct allocation scheme to command aircraft effectors. With minimal change to the control laws, the generic utility of NDI is demonstrated through comparison of inner-loop and outer-loop dynamics between the two simulations. The NDI control laws were primarily developed for offline batch simulation development, with the ability to also use the real-time, pilot-in-the-loop simulation environment.

The motivation for this report was to create a tool that would be useful for conceptual design and evaluation of preliminary design of aircraft by utilizing work previously completed by several different researchers. This tool would be used to specify subsystem requirements for actuator bandwidth and performance, control surface effectiveness, control system time delays, and bare airframe aerodynamics. In conceptual design, aircraft performance is often evaluated with simplifying assumptions, ignoring the rotational effects on an aircraft. The tool presented here will enable the conceptual design of an aircraft to be accomplished while taking into account the full nonlinear equations of motion for that aircraft. To build on the successful work of others that is publicly available, control law architecture was based on work done by Honeywell[3] (Minneapolis, Minnesota) researchers, and control allocation was based on work done at the Virginia Polytechnic Institute (Blacksburg, Virginia).[6]

II. Simulation and Aircraft Description

The NASA Dryden Flight Research Center has developed a common simulation platform that is used for all 6-DOF simulations originating at NASA Dryden. This simulation platform is capable of simulating wide ranges of aircraft, and includes basic analysis tools integrated into the simulation environment. The simulation platform has the ability to run both in real-time and batch modes so that piloted cockpit evaluation can be performed as well as analysis from the engineer's desktop. Included in the simulation is a standard architecture for subsystems. The control system described in this report makes use of this architecture standard. As new simulations are developed, the control system will not require modification. Specifically, common access to the simulation aerodynamic database and toolset for linearizing aerodynamic models[7] are needed to support the control system.

Two radically different aircraft models were used to test control system utility. An airbreathing hypersonic vehicle in the conceptual design phase was evaluated, as well as a well-established twin-engine supersonic aircraft. The simulation of the supersonic aircraft is a high fidelity simulation used for preflight evaluation of flight experiments on that aircraft. Both models were capable of pilot-in-the-loop simulation. The hypersonic aircraft was evaluated in its high-speed regime, where the simulation models were deemed relevant. The supersonic aircraft was evaluated across its full flight envelope (subsonic, transonic, and supersonic) as well as in normal and unusual attitudes to fully test the robustness of the control system. These two aircraft models were selected because they present a wide range in aircraft characteristics (as shown in table 1) and a wide range in simulation fidelity.

Table 1. Comparison of basic aircraft characteristics.

	Fighter aircraft	Hypersonic aircraft
Total weight, lb	32,000	4,800
Length, ft	56	16
Reference wing area, ft^2	400	74
Speed range, Mach number	0.3 to 1.8	4 to 8

III. Control Law Architecture

The control law framework was designed primarily with the objective of minimizing the number of changes required to move from one aircraft simulation to the next. Further, it was desired that vehicle performance could be specified. Gain changes are to be minimized and if possible eliminated as simulation models are updated during conceptual or preliminary design.

A. Architecture Overview

Dynamic inversion was chosen as a basic architecture for the control system. A control allocation scheme was implemented to allow any number of control effectors to be used.[6,8,9,10] To accommodate outer-loop control, a scalable design was also desired. Specifically, a control architecture template was selected that could be used for both inner- and outer-loop control. Using the same design template in both inner- and outer-loop control enabled rapid implementation of a complex outer-loop control. After the inner-loop control was proven, outer-loop design could be rapidly achieved.[3]

Figure 2 shows an overview of the control architecture. The control system can be broken into four major sections. These sections are the input and output interfaces, the NDI control section, the *B*-matrix estimator (BEST), and the control allocator.

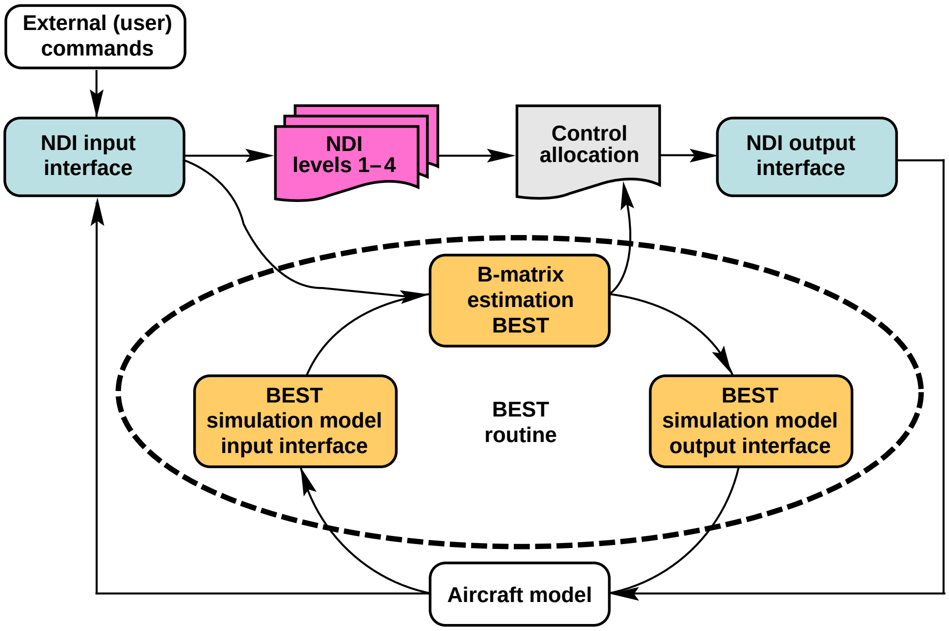

Figure 2. Control system overview.

B. Input-Output Interfaces

The input-output interfaces are used to connect the control system with the simulation and to access simulation models required by dynamic inversion. All required simulation parameters, as well as outputs from the control system, must pass through this interface. External commands, either from a pilot command interface or external guidance algorithms, must pass through this interface as well. The interfaces were established in this manner to allow the control law subroutines to be used with any NASA Dryden simulation with minimal changes.

C. Nonlinear Dynamic Inversion

Figure 3 shows the four levels of the NDI control system. The four control levels allow varying degrees of user input control. External inputs can be interjected at each level. For example, during real-time piloted simulation operation, only level four is functional, accepting body pitch, roll, and yaw rate commands (*p, q,* and *r*) from piloted input. Levels one through three allow varying levels of augmentation, depending on the desired guidance level for the system. The level one function accepts waypoints from an external source, such as a mission manager, allowing autonomous aircraft control. Final outputs of the NDI controller are nondimensional body moment commands in pitch, roll, and yaw, which are passed to the control allocator.

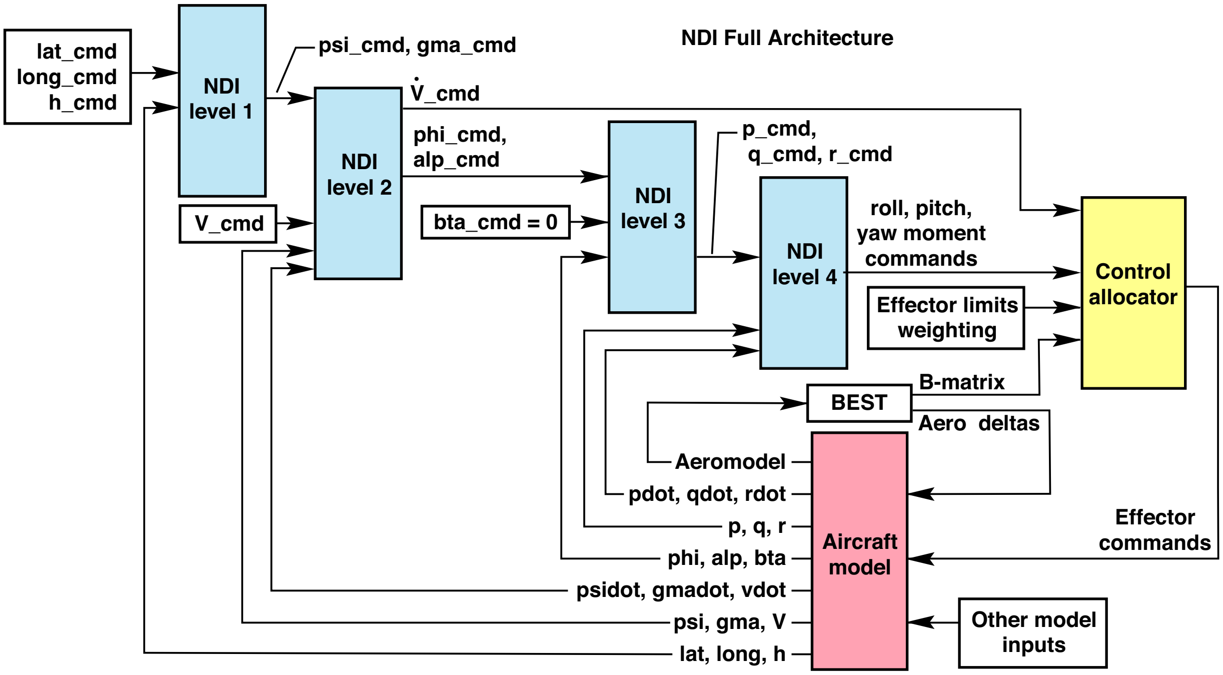

Figure 3. NDI control section.

Figure 4 is a breakdown of one NDI control level. Commands from either external inputs or the upper levels of NDI control (internal commands) are accepted. These commands are manipulated as necessary in the command generation block to become inputs for the dynamic algorithm. Aircraft states and the state derivatives are then combined with the generated commands in the dynamic algorithm, which generates a desired dynamic response for the control system. Figure 5 shows a detailed diagram of the dynamic algorithm.[2] Figure 6 shows an example dynamic response from the system in Fig. 5. The control law response shaping gain (K_B) can be changed to shorten or lengthen the desired response time, as well as increase or decrease the rate command from the input command to the system.[11] The dynamic algorithm selected can also be represented as the transfer function found below.

$$\frac{CV}{CV_{cmd}} = \frac{0.5K_B}{s + 0.5K_B} \tag{1}$$

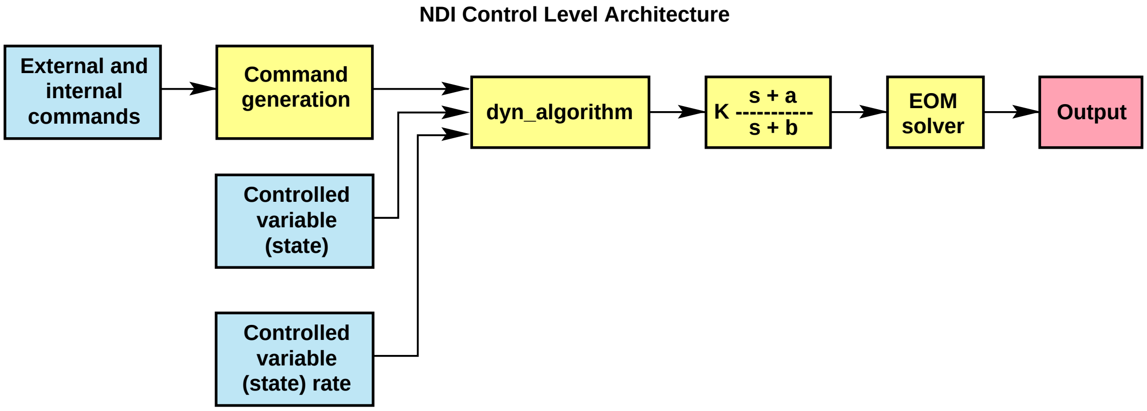
NDI Control Level Architecture
External and internal commands
Command generation
Controlled variable (state)
Controlled variable (state) rate
dyn_algorithm
K (s + a)/(s + b)
EOM solver
Output

Figure 4. NDI individual level breakdown.

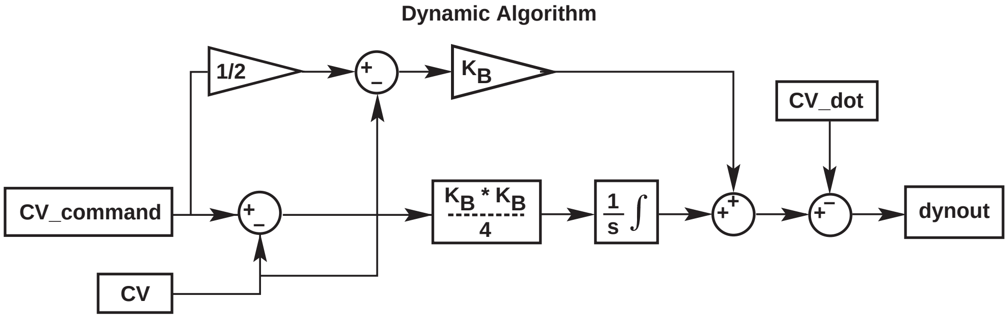

Figure 5. Dynamic algorithm for NDI controller.

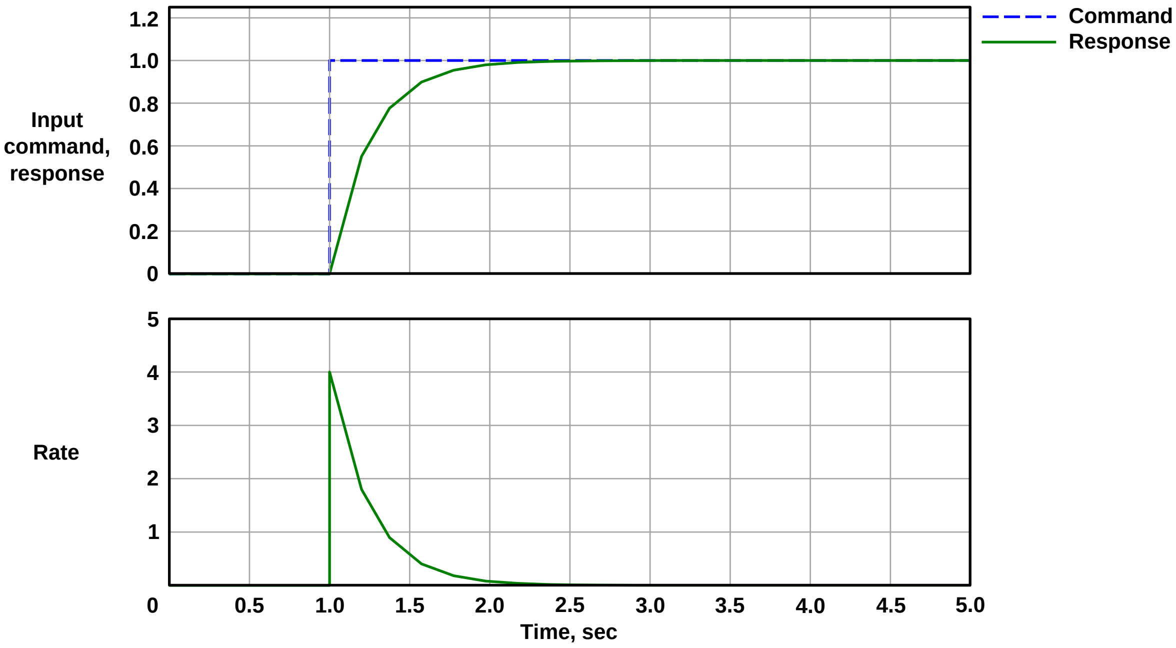

Figure 6. Dynamic response from NDI dynamic algorithm.

The outputs of the dynamic algorithm can then be shaped with a filter to further modify response dynamics. The equations of motion (EOM) solver found in Fig. 4 then converts outputs from the shaping filter. This solver consists of equations derived from the equations of motion to convert the outputs into the necessary inputs for the next level of control. An example of this algorithm would be an angle of attack (α) command coming into NDI level three. The output of the dynamic algorithm would be a command of change in angle of attack ($\dot{\alpha}$). This would be converted in the EOM solver to a body axis pitch rate (q) command, which would then be sent to NDI level four.

Connecting the different NDI control levels requires manipulation of the equations of motion for an aircraft. Since a state derivative is the output of each level of control, that state derivative must be converted into the required input for the next level of control. The output of each level and the corresponding conversion is described below.

1. Level Four

The level four controller is used to directly invert the three rotationally derived equations of motion. The commanded inputs to this controller are wind axis roll rate (p_w), body axis pitch rate (q), and body axis yaw rate (r). The outputs of this controller are wind axis roll acceleration ($\dot{p}_w$), body axis pitch acceleration ($\dot{q}$), and body axis yaw acceleration ($\dot{r}$). The control allocation command requires conversion of wind axis roll acceleration ($\dot{p}_w$) to body axis roll acceleration ($\dot{p}$). Commanded moments are calculated from these accelerations and are used to find nondimensional moment coefficients using the equations below.[12]

$$M = I * a \tag{2}$$

$$C_l = \frac{l}{\bar{q}Sb} \tag{3}$$

$$C_m = \frac{m}{\bar{q}S\bar{c}} \tag{4}$$

$$C_n = \frac{n}{\bar{q}Sb} \tag{5}$$

The nondimensionalized coefficients are then passed to the control allocator for effector allocation.

2. Level Three

The level three controller is an outer-loop controller for the rotationally derived equations of motion found in level four. The inputs to this controller are angle of roll (ϕ), angle of attack (α), and angle of sideslip (β). The outputs of the level three dynamic algorithm are wind axis roll angle rate command ($\dot{\phi}_w$), angle of attack rate command ($\dot{\alpha}$), and sideslip rate command ($\dot{\beta}$). These rates must be converted using the EOM solver into the commanded values for the level four controller, namely p, q, and r. Assuming perfect knowledge of the aircraft states, the following equations can be solved for p, q, and r, which can then be passed as commands to the level four controller.[3]

$$\dot{\phi} = \frac{\cos(\alpha)}{\cos(\beta)} \cdot p + \frac{\sin(\alpha)}{\cos(\beta)} \cdot r + \tan(\beta)\cos(\phi)\dot{\gamma} + \left[\sin(\gamma) + \tan(\beta)\sin(\phi)\cos(\gamma)\right] \cdot \dot{\psi} \tag{6}$$

$$\dot{\alpha} = -\cos(\alpha)\tan(\beta) \cdot p + q - \sin(\alpha)\tan(\beta) \cdot r - \frac{\cos(\phi)}{\cos(\beta)} \cdot \dot{\gamma} - \sin(\phi)\frac{\cos(\gamma)}{\cos(\beta)} \cdot \dot{\psi} \tag{7}$$

$$\dot{\beta} = \sin(\alpha) \cdot p - \cos(\alpha) \cdot r - \sin(\phi) \cdot \dot{\gamma} + \cos(\phi) \cdot \cos(\gamma) \cdot \dot{\psi} \tag{8}$$

3. Level Two

The level two controller directly inverts the three position or "force" derived equations of motion. The inputs to this controller are heading angle (ψ) command, flightpath angle (γ), and total vehicle velocity (V). The outputs of this controller are change in heading angle rate ($\dot{\psi}$), change in flightpath angle rate ($\dot{\gamma}$), and change in total vehicle velocity ($\dot{V}$). When connecting the level two controller to the level three controller, the controller commands a zero β to accommodate turn coordination. The level two controller can be connected to the level three controller with the understanding that the level two controller $\dot{V}$ command is connected directly to a control allocation scheme. The control allocator will control the power lever angle command (PLA) to the aircraft for level two and level one control schemes. By commanding β to be zero at all times, and assuming that the sideslip controller in level three does an adequate job of maintaining zero β, the equations relating level two to level three can be significantly reduced to save computational time. The resulting equations can be found below.[3,12,13,14]

$$F_T = \dot{V} \cdot \mu - D \tag{9}$$

$$\sin(\phi) = \dot{\psi}\left[\frac{V \cdot \mu \cdot \cos(\gamma)}{L}\right] \tag{10}$$

$$L = \left[\dot{\gamma} + \frac{\cos(\gamma)}{V} \cdot g\right] \cdot V \cdot \mu \cdot \frac{1}{\cos(\phi)} \tag{11}$$

These equations can be solved for commanded engine thrust (F_T), commanded angle of roll (ϕ), and commanded lift (L). F_T can be used to find a corresponding PLA command for the aircraft engine. If a simplified engine model is assumed (such as usually exists in conceptual design), a direct correlation between PLA command and engine thrust can be easily developed. If a more mature engine model is present, a digital engine controller can be used to simplify the engine response as a function of power setting to again achieve a correlation between PLA and engine thrust. This relation should be kept simple with minimal dynamics to minimize throttle command movements. The ϕ found from the equation can be directly passed into the level three controller. The solution for L, however still must be reduced further to create an α command for the level three controller. An algorithm similar to the B-matrix estimation algorithm (BEST) must be employed to find the aircraft lift coefficient (C_L) as a function of α. Assuming a linear relation (true for all but near-stalled flight) between lift and angle of attack, α can be determined from the value of L calculated from the above equations.

4. Level One

The inputs to the level one controller are latitude, longitude, and altitude command. This controller is an outer-loop control to the level two controller. Inputs to the level one controller must first be transformed from an oblate earth coordinate frame, the World Geodetic System 1984 (WGS84) to a flat plane, for control purposes. Once in a flat plane coordinate frame, the vehicle can be commanded to fly to a commanded x, y, and z coordinate, related to latitude, longitude, and altitude, respectively. The output of the controller is rate of change in x, y, and z ($\dot{x}, \dot{y}, \dot{z}$). The rate outputs can then be mapped to the level two inputs by the equations below.[3,13]

$$\sin(\gamma) = \frac{\dot{h}}{V} \tag{12}$$

$$\tan(\psi) = \frac{\dot{y}}{\dot{x}} \tag{13}$$

Velocity must be commanded in the level two controller; if it was allowed to be commanded in level one, zero V would be commanded when the waypoint (latitude, longitude, and altitude) was reached. Further, the equations listed above have numerous numerical singularities that must be taken into account during the controller design. These can be overcome by defining proper sign relationships between x, y, and ψ.

Another numerical problem that must be resolved relates to the ψ command being sent to the level two controller. If standard polar coordinates are used, there will be a large transient between 359 deg and 0 deg. The solution that was implemented involved always considering the current ψ to be 0 deg, and then commanding a delta from the "true" heading value. Numerical problems again appeared when trying to command 180-deg heading changes. This was solved by not allowing the delta command to grow larger than 160 deg. When commands were larger than 160 deg, the command was allowed to saturate until the aircraft turned enough to reduce the heading error. A third problem that was solved was commanding the aircraft to turn the proper direction for the smallest heading error.

5. Command Limiting

While the desire in designing this control system is to create a framework that is independent of aircraft type or configuration, the performance differences of each class of aircraft must be taken into account. The turning radius of a hypersonic aircraft, for example, is much larger than that of a modern fighter. To this end, command limits were placed on each path of each level of the controller.[3,4] These limits are aircraft-specific and must be set by the user to ensure that performance limits inherent to a given class of aircraft are not exceeded.

D. *B*-Matrix Estimation Algorithm

The BEST algorithm outputs a linearized estimate of the effectiveness of each control effector at the aircraft's current state in time as a function of control deflection. The BEST routine was made compatible with current simulation architecture and to fit within a real-time simulation frame. Several algorithms were tested to find the optimal balance between accuracy and speed. The final algorithm choice calculates a forward and backward increment in effectiveness from the current effector position.[5] Only one effector per control system frame is updated.

E. Control Allocation

The control allocator[6,8,9] uses a model matching technique to compute a set of effector commands from the nondimensional moment command input generated from the level four controller. The allocator requires a model input for the aircraft; currently that is accomplished with a BEST algorithm. A change in nondimensional moment from the current aircraft state is commanded as input along with a control effector matrix and effector limits. A direct allocation method[10] is then used to determine a change in effector output command, which is then summed with the current aircraft effector position to create a new effector position command to the simulation actuator model. Direct allocation was chosen for several reasons. A computationally simple algorithm that could fit within the real-time simulation frame was desired. While direct allocation may not yield the most optimized result for control allocation, it does produce an acceptable[10] result that allowed more computational resources to be spent in the BEST algorithm, where they were needed. Further, an algorithm for direct allocation was available, and using this algorithm allowed the researchers to spend less time developing the control allocation portion of the system and to focus on areas that required more maturation for the design to function properly.

IV. Results and Analysis

An example of the ability to shape response in the controller is given. Open-loop inputs are made to the inner-loop controller for each aircraft described earlier. The simulation responses are then compared. The utility of the outer-loop control is also demonstrated for the fighter aircraft.

The utility of the dynamic algorithm shown in Fig. 5 allows the user to shape the dynamic response of the aircraft to a desired form. During the design of the controller, the dynamic response was selected for a given aircraft. Figure 7 is an example of various dynamic responses tested during design. This figure illustrates the changes in roll rate response from the hypersonic vehicle, although the same responses can be shown for the fighter aircraft. Modulating the K_B gain in Fig. 5 can make these changes. Response 1 in this figure shows a two-tiered response from the roll command. This response can be explained by the aircraft reaching its performance limit, and the actuators saturating during the roll rate response. Responses 2 and 3 in Fig. 5 exhibit no undesired characteristics.

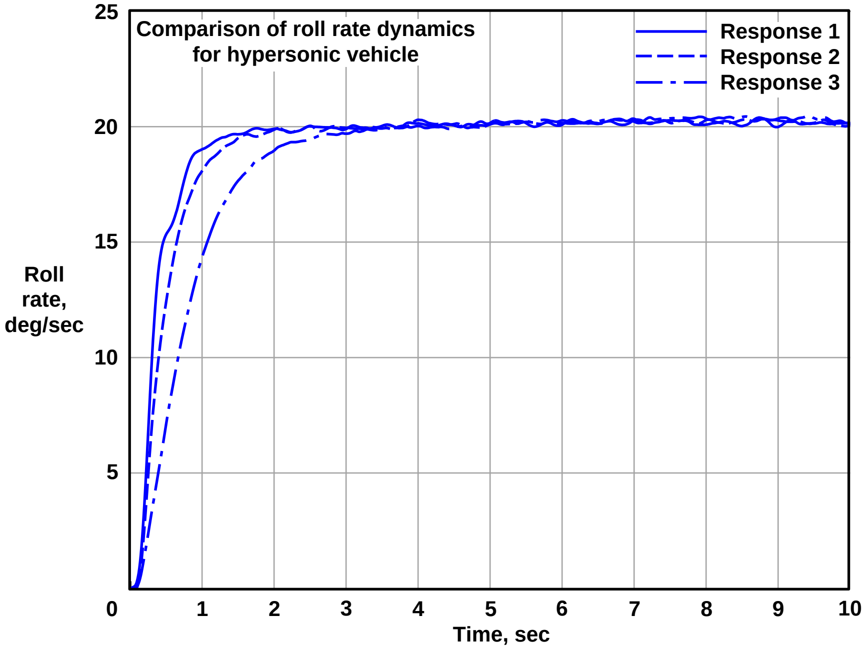

Figure 7. Various dynamic responses for hypersonic aircraft roll command.

Upon selection of a desired response for the controller in each axis, the open-loop response characteristics were compared for the level four (rate command) inner-loop controller. The desired response for this controller is that it produce the same command response, irrespective of the aircraft. Using the fighter and hypersonic aircraft described above, various open-loop commands were analyzed to determine the vehicle response to open-loop commands. Theoretically, the dynamic inversion control law will produce the same response for both aircraft, at any flight condition. To demonstrate this characteristic, open-loop steps are commanded to the different simulations in each axis. The hypersonic vehicle and the fighter are trimmed at a Mach 5.78 and a Mach 0.6 flight condition, respectively. Figure 8 contains six time histories, generated by three different simulation runs for each aircraft. Each run commanded a step response in one axis, and the vehicle response was recorded. Roll and yaw rate ($\dot{\phi}, \dot{\psi}$) commands show very similar responses, whereas the pitch axis shows a difference in response. This difference is related to the engine performance of the hypersonic aircraft; this factor was not included in the aircraft control law modeling. Engine modeling was intentionally excluded from the BEST algorithm because of the computational resources that would be required to include engine performance estimation. Further minor differences in the responses can be attributed to the simulations using different actuator models, which for simplicity were not included in the dynamic inversion controller.

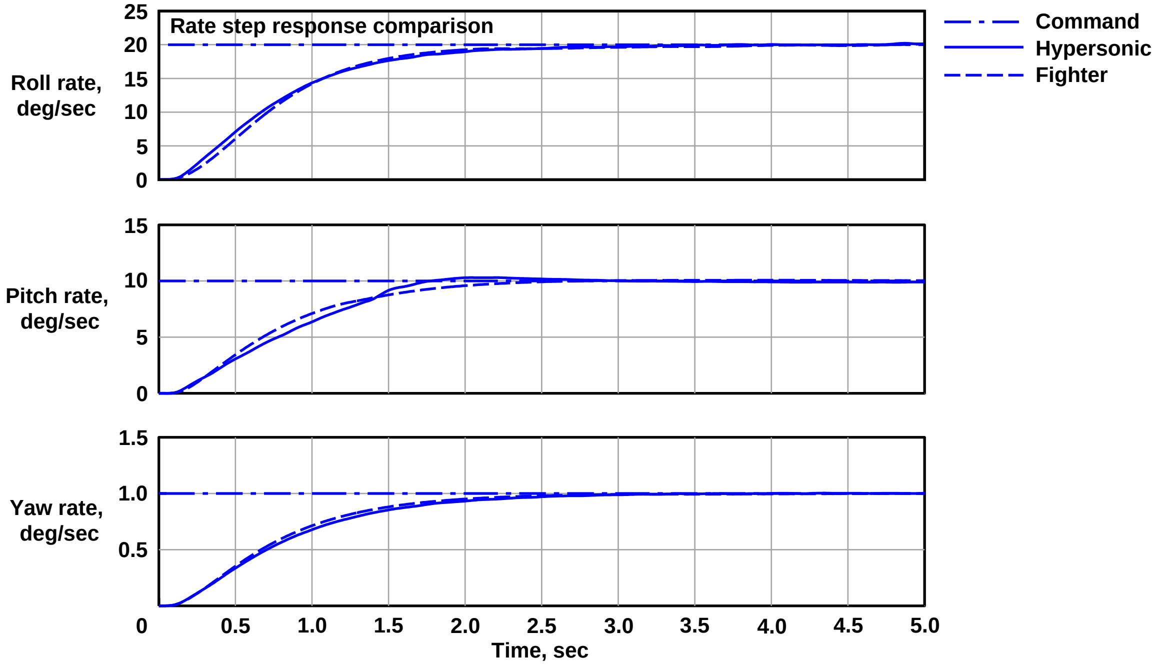

Figure 8. Open-loop step responses for both aircraft.

Investigation into outer-loop control response similarity between the two aircraft was also performed. This investigation was limited to only the level three controller, since aircraft performance issues would not allow further reasonable comparisons. Figure 9 contains an α command and ϕ command for each simulation. The ϕ command shows very similar responses for both aircraft. The α command responses are not exact, but show similarity. The differences in the α command are related to installed engine performance in the hypersonic vehicle, which again were not simulated in the aircraft model used in the dynamic inversion controller. Although the level three traces were not exact, their similarity, especially across the very different classes of aircraft and flight conditions, show that the controller is capable of making repeatable outer-loop control responses for various classes of aircraft.

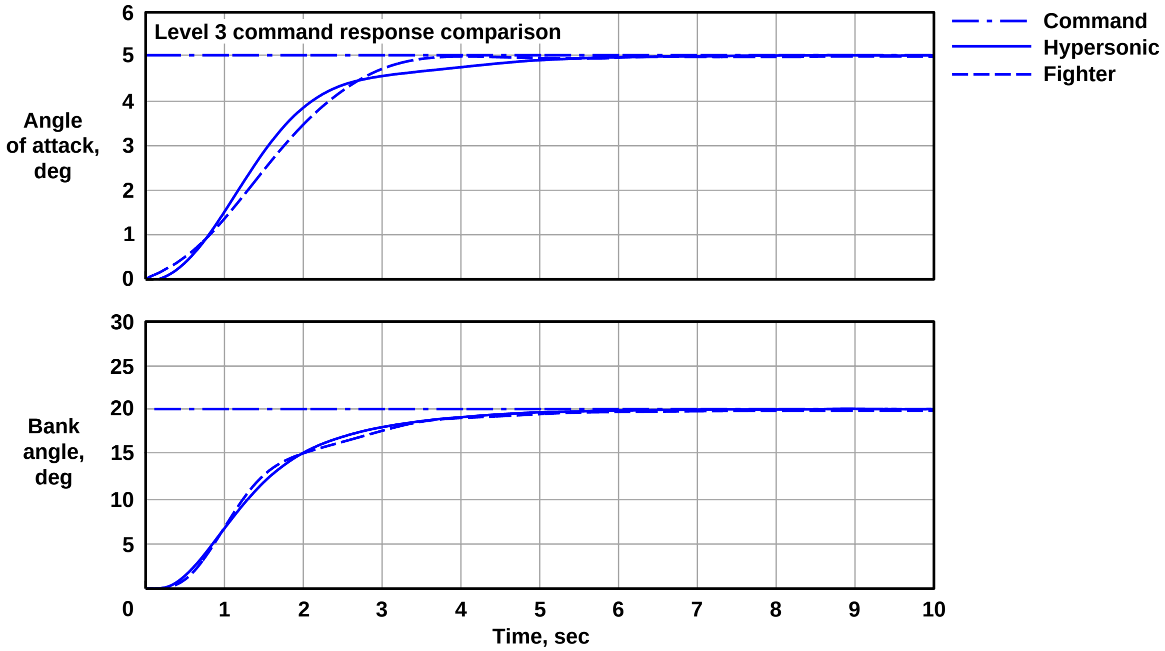

Figure 9. Level three command responses for both aircraft.

To document the phase and gain margins of this controller, the level four controller was examined using a linearization tool built into the simulation framework.[7] Figures 10, 11, and 12 demonstrate the controller margins across both simulations. The trimmed flight conditions selected were Mach 5.8 for the hypersonic aircraft and Mach 0.6 for the fighter. While the phase and gain margin plots for both simulations show differences between the two simulations, they both show significant gain and phase margins.

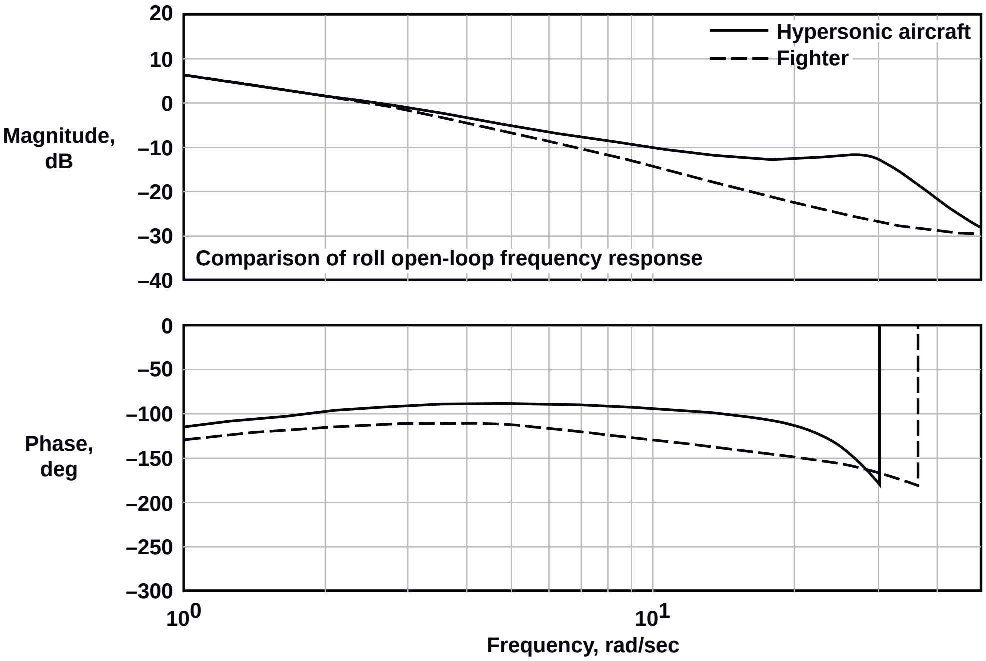

Figure 10. Roll channel phase and gain margins.

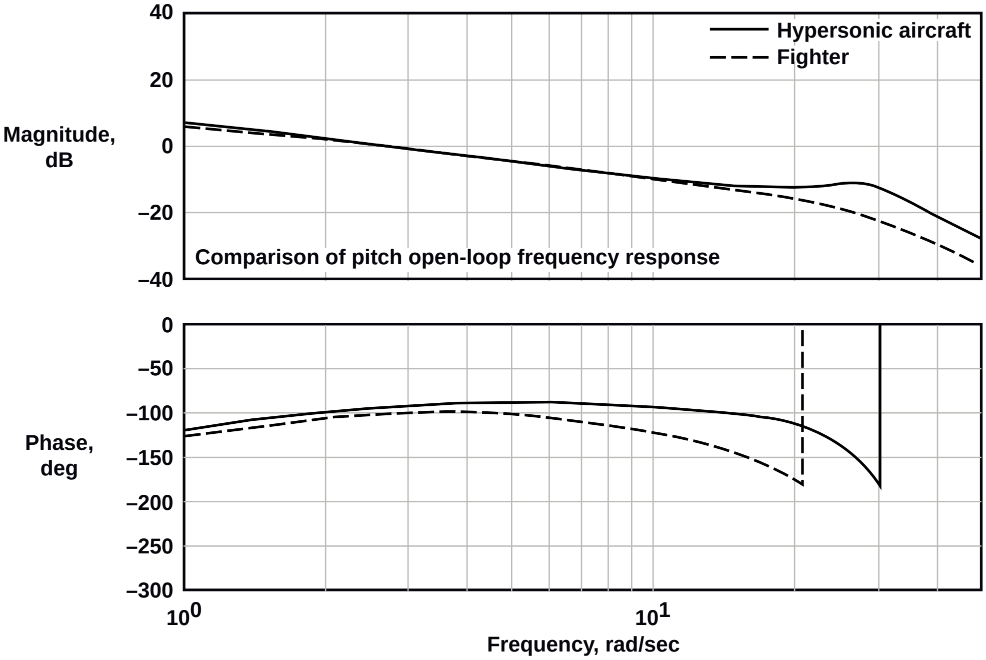

Figure 11. Pitch channel phase and gain margins.

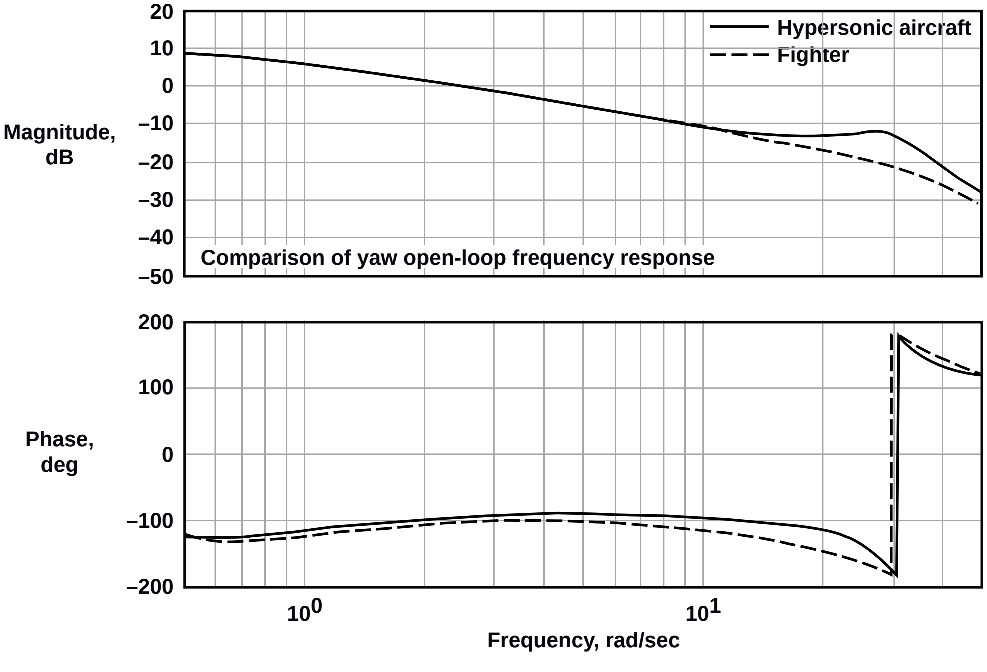

Figure 12. Yaw channel phase and gain margins.

The level one and two controllers were only examined for the fighter aircraft because comparison of a hypersonic aircraft and a fighter aircraft is not possible due to the performance differences between them. While the hypersonic aircraft can be controlled with the level one and two controller, only the fighter aircraft is presented here. Figure 13 contains data from a simulation run with the level two controller engaged. During this run, aircraft velocity was controlled at 650 kn. The flightpath angle (γ) was changed twice, both times as step inputs. The heading angle (ψ) was also changed twice, both as step inputs. Figure 13 illustrates some lightly damped oscillations in command tracking. This can be attributed to being three integrators away from the direct control of the aircraft, and the manifestation of a phasing issue. Interactions between γ and ψ can be observed. Since both channels rely on the lift vector for control, some cross coupling can be seen. Figure 14 illustrates the ground track of the aircraft during the simulated maneuver. The commands to the flight controller cause a moderate right turn, followed by a larger left turn. As can be found in Fig. 13, the aircraft is capable of maintaining climbing and level flight during the turning maneuver.

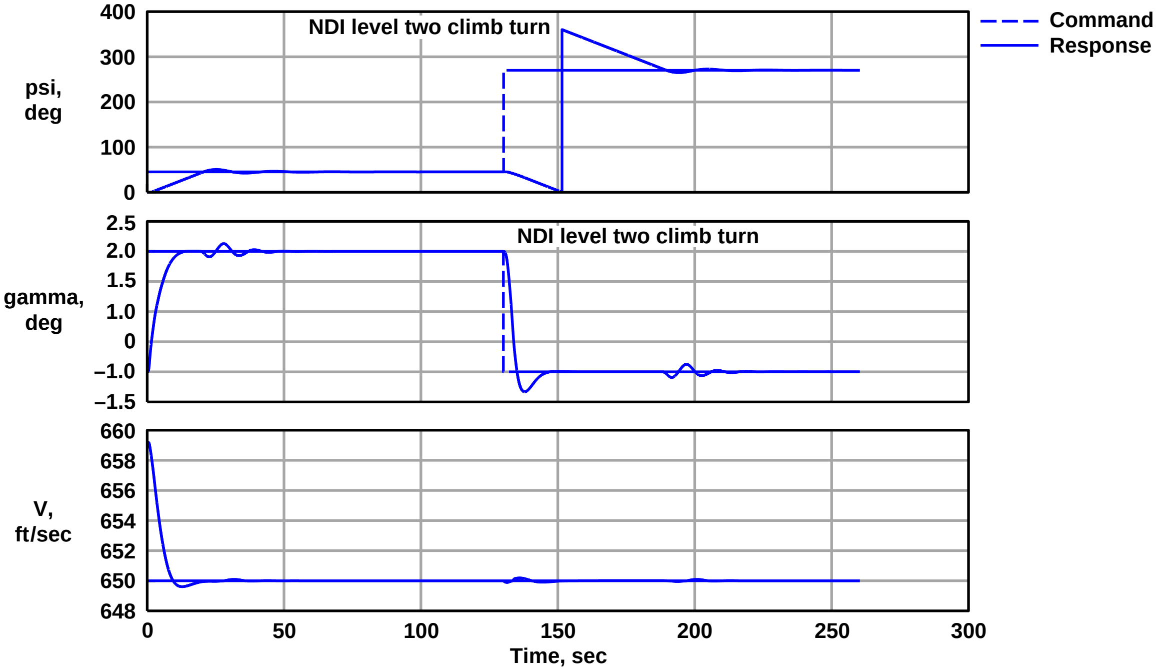

Figure 13. Level two controller simulation run.

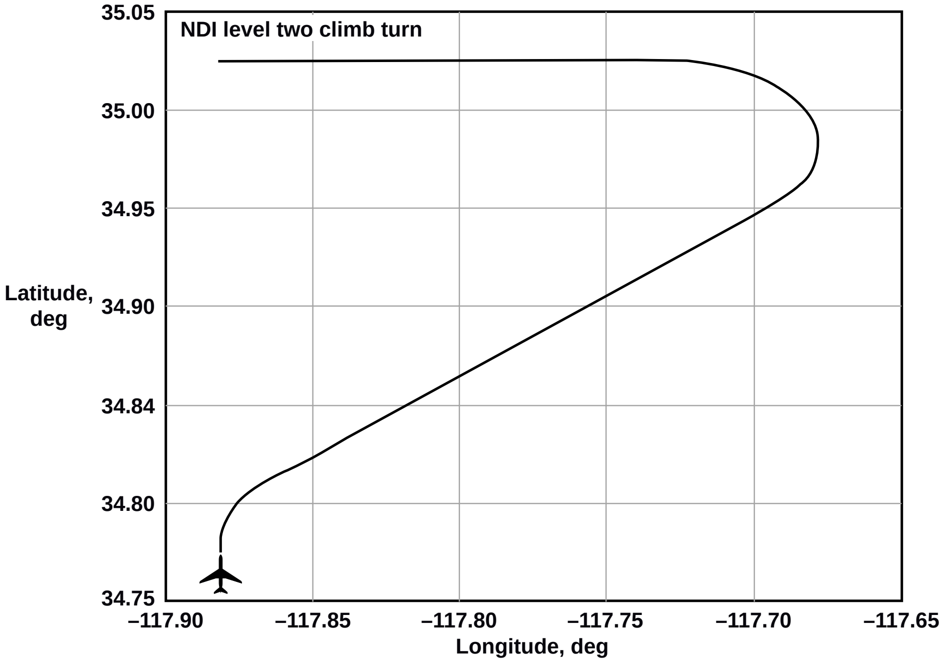

Figure 14. Level two controller ground track.

The level one controller enables the aircraft to be flown to specific waypoints. The simulations in this study use oblate earth models, and therefore the user commands for latitude, longitude, and altitude must be converted into a coordinate frame relative to the oblate earth used in the simulation. During this conversion, precision is lost in this controller because all variables were calculated using single precision numbers instead of double precision. Single precision is used to keep within the simulation framework. This precision loss resulted in some minor inaccuracies in the controller response. Figure 15 contains the trace from a simulation run with the fighter aircraft. The aircraft does intercept the waypoint twice, and the errors due to precision loss discussed above can be seen in the altitude trace in Fig. 15. Figure 16 shows the ground track of the fighter. The fighter was allowed to intercept the waypoint, and then turn and reacquire the waypoint for a second pass. After the second pass, the simulation was reset. The aircraft would continue to follow the "clover leaf" pattern until the simulation was reset. Limits were placed on the commanded ϕ to limit the interaction between γ and ϕ required for a tighter turn.

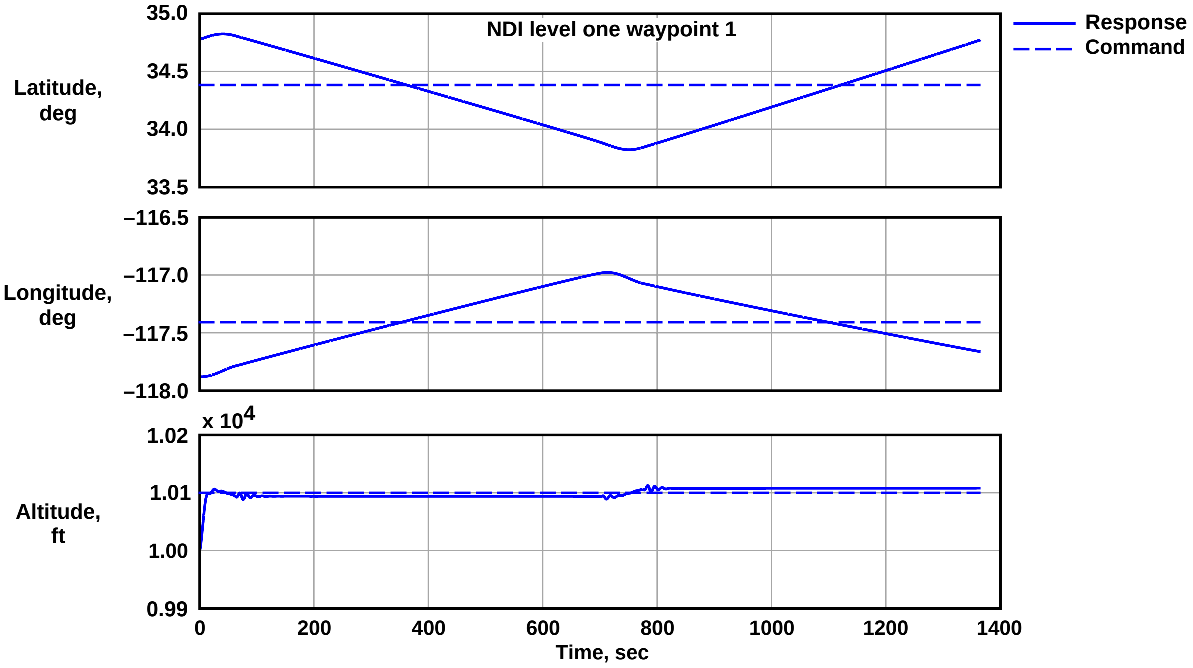

Figure 15. Level one controller simulation run.

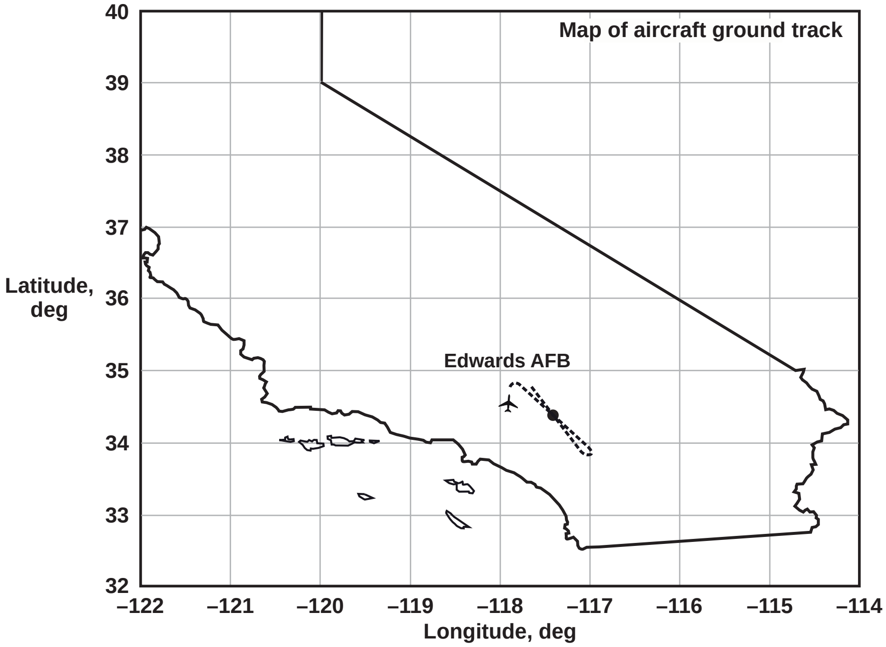

Figure 16. Level one controller ground track.

V. Conclusions

Researchers at the NASA Dryden Flight Research Center (Edwards, California) recently developed a set of generic dynamic inversion control laws for a six-degree-of-freedom simulation environment to aid in the rapid development of aircraft design in the early, conceptual stages. The generic implementation utilizes a nonlinear dynamic inversion approach, which uses state acceleration feedback and estimates of the surface effectiveness of the aerodynamics to perform the dynamic inversion cancellation of the bare airframe dynamics. Several outer-loop guidance algorithms were developed to aid in aircraft design, and to aid in the performance measurement of an aircraft with the six-degree-of-freedom simulation.

Current plans do not include flight test and evaluation of the control system described, however, it has proven to be a very useful tool in conceptual design and early preliminary design of aircraft. Future work may include the addition of a mission manager that would interface with the level one control algorithm and allow an aircraft to follow a series of waypoints. Also included in future work may be a tool to allow variance of actuator bandwidths and sensor fidelity changes. Adding these features would allow investigation into subsystem requirements definition for a simulated airframe. Determining performance envelopes and flight maneuvers can, however, be determined very easily with the algorithm described in this report.

This report describes the generic implementation of the nonlinear dynamic inversion into two simulations: an airbreathing hypersonic vehicle and a twin-engine high performance fighter aircraft. With only changes to the performance limits in the control laws, their generic utility is validated by showing that for a given open-loop input, a match between the hypersonic simulation response at Mach 5.78 and the fighter aircraft simulation response at Mach 0.6 is achieved. For this study, stability analysis for both simulations indicate the control laws are robust, with large amounts of stability margin.

References

[1] Enns, D., Bugajski, D., Hendrick, R., and Stein, G., "Dynamic Inversion: an Evolving Methodology for Flight Control Design," *International Journal of Control*, Vol. 59, No. 1, 1994, pp. 71-91.

[2] Honeywell Technology Center, Lockheed Martin Skunk Works, and Lockheed Martin Tactical Aircraft Systems, "Application of Multivariable Control Theory to Aircraft Control Laws, Final Report: Multivariable Control Design Guidelines," WL-TR-96-3099, Flight Dynamics Directorate, Wright Laboratory, Air Force Materiel Command, Wright-Patterson Air Force Base, Ohio, May 1996.

[3] Bugajski, D. J., and Enns, D. F., "Nonlinear Control Law with Application to High Angle-of-Attack Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, 1992, p. 761.

[4] Smith P. R., "A Simplified Approach to Nonlinear Dynamic Inversion Based Flight Control," AIAA-98-4461, 1998.

[5] Cox, T. H., and Cotting, M. C., "A Generic Inner-Loop Control Law Structure for Six-Degree-of-Freedom Conceptual Aircraft Design," AIAA-2005-0031, 2005.

[6] Durham, W. C., and Bordignon, K. A., "Closed-Form Solutions to Constrained Control Allocation Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 1000-1007.

[7] Bosworth, J. T., "Flight-Determined Longitudinal Stability Characteristics of the X-29 Airplane Using Frequency Response Techniques," NASA Technical Memorandum 4122, 1989.

[8] Durham, W. C., "Attainable Moments for the Constrained Control Allocation Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1994, pp. 1371-1373.

[9] Durham, W. C., Bolling, J. G., and Bordignon, K. A., "Minimum Drag Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1996, p. 190.

[10] Glaze, M. L., "Design and Implementation of a GUI-Based Control Allocation Toolbox in the MATLAB® Environment," Masters Thesis, Aerospace and Ocean Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1998.

[11] Edwards, J. W., and Deets, D. A., "Development of a Remote Digital Augmentation System and Application to a Remotely Piloted Research Vehicle," NASA TN-D-7941, 1975.

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

[13] Schindeler, N. J., "Phugoid Damping Control," Masters Thesis, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, 2001.

[14] Vihn, N. X., *Flight Mechanics of High Performance Aircraft*, Cambridge University Press, Cambridge, England, UK, 1993.