Quaternion-Based Control Architecture for Determining Controllability/Maneuverability Limits

Barton J. Bacon[*]

NASA Langley Research Center, Hampton, VA, 23681

Dynamic inversion has often been used in the simulation environment to rapidly prototype controls for the full flight envelope, because of its capacity for assessing a vehicle's maneuver performance and proper sizing of control surfaces. Generally, the architectures involve either a direct inversion of the entire set of equations of motion or a sequential set of inversions exploiting time scale separation in the vehicle dynamics where faster parameters are considered as controls for slower varying parameters. The proposed architecture builds on the latter using a quaternion formulation that provides singularity free tracking. Of interest, the proposed architecture simplifies the sequential approach by exploiting a simpler kinematic inversion in place of a more difficult inversion typically used. This kinematic relationship accurately describes the angular rate required to drive some reference frame of interest to a desired attitude at some desired quaternion error rate. A simple PID control is used to define the desired quaternion error rate. The paper develops the theoretical framework for the approach, and shows results in tracking a desired trajectory.

I. Introduction

A recently developed quaternion-based control for a notional launch abort system is reformulated as an analysis tool to address aircraft safety applications in assessing recovery maneuver control power requirements, providing a maneuvering envelop based on control power limitations of the damaged vehicle, and characterizing reachable response dynamics for control adaptation. The quaternion formulation provides a concise singularity-free description of the orientation of one reference frame to another. For crew escape, the control provided singularity-free tracking of the vehicle with respect to the wind axis system (defined with the angle of attack and side-slip angle) with coordinated moment commands from pitch-over and coast-to-reorientation and heat-shield-forward flight. For aircraft, the extended approach presented in this paper includes control of wind axis bank angle along with angle-of-attack and sideslip angle—all three angles are drivers in determining the orientation of the vehicle's lift vector and consequently the movement of the vehicle's inertial velocity vector (i.e. maneuvering). In both cases the control exploits the two-timescale nature of the controlled dynamics. Here, the slower loop, utilizing a simple quaternion relationship, controls the orientation of the vehicle with respect to, not the wind-axis, but the *N*-frame (separated from the wind-axis by the wind-axis bank angle and the inertial axis by heading and flight path angle), and generates a set of commanded angular body rates. The angular rate commands are followed by a faster dynamic-inversion-based inner loop control that determines the required vehicle's control moments. To follow candidate trajectories, an outer guidance loop, based on inverted point mass equations, is added to generate the necessary angle of attack, side-slip angle, and wind-axis bank angle commands along with required throttle commands. A control allocation algorithm will determine whether the required control moments can be realized or whether the candidate recovery trajectory needs to be modified.

At this point in the tool's development, however, no candidate recovery trajectories have been modified or envelopes of acceptable maneuvering have been derived. In this paper, the control architecture is described and implemented in the simulation of a subscale vehicle. An illustrative example of its potential for following a desired trajectory is presented. It should be mentioned that there are many options available both in academia and in

[*] Senior Research Engineer, Dynamic Systems and Control Branch, NASA LaRC, MS 308, Hampton, VA 23681.

industry regarding control architectures capable for rapidly prototyping full-flight-envelope controls, determining vehicle maneuver performance and in the selection of control surface sizing. Most are based on dynamic inversion, involving direct manipulation of the equations of motions. Some use acceleration feedback, acceptable in the simulation environment—not necessarily acceptable in flight controls—to perform the inversion and realize lower-order responses across the flight envelop [1],[2],[3]. Others exploit the two-time-scale nature of the vehicle discussed above, and perform a sequential inversion treating faster-varying parameters as controls for slower parameters [4]. The proposed architecture builds on the latter, specifically the work of [4]. Interestingly, the closing remarks of [4] included a comment that a quaternion implementation could possibly overcome some of the singularities of the approach. It turns out that not only many of the singularities are eliminated, but quaternions can be used to eliminate one of the sequential inversions involving wind-axis bank angle, sideslip angle and angle of attack and replace it with a much simpler kinematic inversion producing a more accurate relationship between the desired attitude and the angular rate required to achieve it.

To facilitate the discussion, the paper is organized as follows. First a tutorial is provided regarding quaternions, quaternion operations and properties. A simple quaternion-based PID control is then developed and demonstrated via an example with features that will be used in both the control and guidance portions of proposed architecture. Discussions follow concerning the outer-loop *N*-frame attitude control, the inner-loop body-rate controls, the control allocator algorithm, and finally, the guidance law. An illustrative example will follow demonstrating both the benefits and the areas needing more work.

To avoid any confusion, the following nomenclature is adopted. Let $\left\{\omega_B^A\right\}_C$ denote the angular rate of frame *B* with respect to frame *A* expressed in the coordinates of frame *C*. Similarly, let $\left\{V_B^A\right\}_C$ denote the velocity of the origin of frame *B* with respect to frame *A* expressed in the coordinates of frame *C*. Moreover let $\left\{{}^D\dot{V}_B^A\right\}_C$ denote the time derivative taken with respect to frame *D* of V_B^A, the result expressed in coordinates of frame *C*. More nomenclature will be provided in the next section on quaternions.

II. Quaternion Definition, Operations and Properties

The quaternion defining the attitude of frame *B* with respect to frame *A* can be expressed as $q_{A2B} = \begin{bmatrix} \cos(\varepsilon_{A2B}/2) \\ \sin(\varepsilon_{A2B}/2)\left\{n_{A2B}\right\}_A \end{bmatrix}$ where frame *A* undergoes a right-handed rotation about unit vector $\left\{n_{A2B}\right\}_A$ through an angle ε_{A2B} to orient itself to coincide with frame *B*. Since frame *A* rotates about $\left\{n_{A2B}\right\}_A$ to obtain *B*, the coordinates of n_{A2B} remain the same for both frames, i.e. $\left\{n_{A2B}\right\}_A = \left\{n_{A2B}\right\}_B$. For our purposes, the quaternions are all of unit length, $q_{A2B}^T q_{A2B} = 1$.

The relative attitude of one frame to another is often expressed by successive right-handed rotations about different axes using intermediate reference frames. Let q_{A2C} and p_{C2B} respectively designate quaternions defining the attitude of frame *C* with respect to frame *A* and the attitude of frame *B* with respect to frame *C*. Successive rotations can be accomplished with quaternion multiplication ($*$) defined as

$$q_{A2B} = q_{A2C} * p_{C2B} = \begin{bmatrix} q_0 \\ \left\{q\right\}_A \end{bmatrix} * \begin{bmatrix} p_0 \\ \left\{p\right\}_C \end{bmatrix} = \begin{bmatrix} q_0 p_0 - \left\{q\right\}_C \cdot \left\{p\right\}_C \\ q_0\left\{p\right\}_C + p_0\left\{q\right\}_C + \left\{q\right\}_C \times \left\{p\right\}_C \end{bmatrix} \tag{1}$$

where $q_0 \triangleq \cos(\varepsilon_{A2C}/2)$, $p_0 \triangleq \cos(\varepsilon_{C2B}/2)$, $\{q\}_A \triangleq \sin(\varepsilon_{A2C}/2)\{n_{A2C}\}_A$, $\{p\}_C \triangleq \sin(\varepsilon_{C2A}/2)\{n_{C2A}\}_C$ and as noted previously, $\{q\}_A = \{q\}_C$. Note although the lower vector portion of the multiplication definition is in coordinates of the frame *C*, it still satisfies

$$\{n_{A2B}\}_A = \{n_{A2B}\}_B = \frac{(q_0\{p\}_C + p_0\{q\}_C + \{q\}_C \times \{p\}_C)}{\sin(\varepsilon_{A2B}/2)}$$

with identical coordinate values in frames *A* and *B*. A useful property satisfied by quaternion multiplication is the associative property, or

$$q_{A2D} = q_{A2C} * p_{C2B} * r_{B2D} = q_{A2C} * \left(p_{C2B} * r_{B2D}\right) = \left(q_{A2C} * p_{C2B}\right) * r_{B2D} .$$

Quaternion multiplication, however, does not commute, i.e. $q_{A2C} * p_{C2B} \neq p_{C2B} * q_{A2C}$, except when one quaternion is the inverse of the other.

The inverse of quaternion q_{A2C}, defining the attitude of frame *A* with respect to frame *C*, is defined as

$$q_{C2A} = q_{A2C}^{-1} = \begin{bmatrix} q_0 \\ -\{q\}_A \end{bmatrix} \tag{2}$$

and satisfies $q_{A2C} * q_{A2C}^{-1} = q_{A2C}^{-1} * q_{A2C} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$. Of course, the two multiplications correspond to two different relative attitudes: the former is the attitude of frame *A* with itself and the latter is the attitude of frame *C* with itself.

For control, there is one ambiguity in the quaternion description to be noted. Both q_{A2B} and $-q_{A2B}$ are equivalent quaternions in that they provide the same relative attitude between the two frames. The negated one, however, utilizes a rotation in the opposite direction through a complementary angle ε_{A2B}' satisfying $\varepsilon_{A2B} - \varepsilon_{A2B}' = \pm 2\pi$ where $|\varepsilon_{A2B}'| \leq 2\pi$. This ambiguity will be important in a quaternion-based control where one frame is to be driven to a commanded one using the shortest angular path possible.

Consider now frame *B* rotating with an angular rate with respect to frame *A*, ω_B^A. The quaternion q_{A2B} evolves according to [5]

$$\dot{q}_{A2B} = \frac{1}{2} q_{A2B} * \begin{bmatrix} 0 \\ \{\omega_B^A\}_B \end{bmatrix} = \frac{1}{2} q_{A2B} * \{\bar{\omega}_B^A\}_B . \tag{3}$$

Note that although the lower vector portion of q_{A2B} has equivalent coordinates in frame *A* and *B*, the angular rate is expressed in coordinates of *B*, or $\{\omega_B^A\}_B$, not as $\{\omega_B^A\}_A$. Let us expand (3) into a matrix-vector multiplication using the definition of quaternion multiplication in (1) where $q_{A2B} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$ and $\{\omega_B^A\}_B = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$, so

$$\dot{q}_{A2B}=\frac{1}{2}\begin{bmatrix}-q_1 & -q_2 & -q_3\\ q_0 & -q_3 & q_2\\ q_3 & q_0 & -q_1\\ -q_2 & q_1 & q_0\end{bmatrix}\left\{\omega_B^A\right\}_B=\frac{1}{2}Q_1\left\{\omega_B^A\right\}_B\,. \tag{4}$$

From (4), the derivative $\dot{q}_{A2B}$ is always orthogonal to q_{A2B} implying that the trajectory of q_{A2B} moves on the surface of a 4-dimensional hyper sphere of unit radius. This restriction will be required of any desired $\dot{q}_{A2B}$ that a control may be asked to realize.

A last property to be considered is using quaternions to transform vector v expressed in coordinates of frame *A* to coordinates of frame *B*. Defining $q_{A2B}=\begin{bmatrix}q_0 & \{q\}_A^T\end{bmatrix}^T$, the operation to obtain $\{v\}_B$ from $\{v\}_A$ is

$$\{\overline{v}\}_B=\begin{bmatrix}0\\ \{v\}_B\end{bmatrix}=q_{A2B}^{-1}*\begin{bmatrix}0\\ \{v\}_A\end{bmatrix}*q_{A2B}=q_{A2B}^{-1}*\{\overline{v}\}_A*q_{A2B}\;\;. \tag{5}$$

This expression may be expanded with the definitions of quaternion multiplication and inverse to produce a simpler, equivalent result

$$\{v\}_B=\left[2\{q\}_A(\{q\}_A)^T+(q_0^2-(\{q\}_A)^T\{q\}_A)I-2q_0\left\{\tilde{Q}\right\}_A\right]\{v\}_A\,. \tag{6}$$

The cross product matrix $\left\{\tilde{Q}\right\}_A$ is given by

$$\left\{\tilde{Q}\right\}_A=\begin{bmatrix}0 & -q_3 & q_2\\ q_3 & 0 & -q_1\\ -q_2 & q_1 & 0\end{bmatrix}.$$

The matrix in (6) is equivalent to a direction cosine matrix[5].

III. Simple PID Quaternion Control Example

At the heart of both the guidance and control loops in the proposed quaternion-based control is a PID controller that directly manipulates two quaternions representing the actual and desired attitudes of some key reference frame relative to a second reference frame. Due to the commonality of both loops, it is advantageous to look at a simple example that contains all the features/algorithms of the PID quaternion controller. To define the problem, suppose one wants to control the attitude of object *B* relative to frame *A*. Let frame *B* be fixed to the object. Let the attitude of frame *C* with respect to *A* be the desired attitude of the object. The actual and desired attitudes are defined by quaternions q_{A2B} and q_{A2C} respectively. Define the attitudes also by a successive angular rotation sequence $\theta_x\to\theta_z\to\theta_y$, abbreviated as 1-3-2, (first about the *x*-axis in the *A* frame and then about subsequent *z* and *y* axes in the intermediate frames) and equivalently expressed with quaternion multiplication as

$$q_{A2B}=q_{\theta x}*q_{\theta z}*q_{\theta y} \tag{7}$$

where the intermediate quaternions have the form

$$q_{\theta x}=\begin{bmatrix}\cos(\theta_x/2)\\ \sin(\theta_x/2)\\ 0\\ 0\end{bmatrix},\ q_{\theta z}=\begin{bmatrix}\cos(\theta_z/2)\\ 0\\ 0\\ \sin(\theta_z/2)\end{bmatrix},\text{ and }\ q_{\theta y}=\begin{bmatrix}\cos(\theta_y/2)\\ 0\\ \sin(\theta_y/2)\\ 0\end{bmatrix}. \tag{8}$$

A similar set of angles ($\theta_{x\_c}\rightarrow\theta_{z\_c}\rightarrow\theta_{y\_c}$) specify the object's desired attitude where $q_{A2C}=q_{\theta x\_c}*q_{\theta z\_c}*q_{\theta y\_c}$. For simplicity, assume that a first-order system governs angular rate $\left\{\omega_B^A\right\}_B$ of the object to some commanded angular rate response $\left\{\omega_B^A\right\}_{B\_CMD}$ with a diagonal transfer matrix with elements $\dfrac{1}{\tau_k s+1}$ for $k=x,y,z$. From $\left\{\omega_B^A\right\}_B$ and q_{A2B}, equation (3) governs the object's resulting attitude q_{A2B}.

For control purposes, the attitude change required of the object to move to the desired attitude is

$$q_e=q_{A2B}^{-1}*q_{A2C}=q_{B2C} \tag{9}$$

The attitudes of frames *B* and *C* coincide when $q_e=\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$. To develop an expression governing the motion of q_e, note first that q_e satisfies $q_{A2B}*q_e=q_{A2C}$. From (1), it is easily shown (using an equivalent matrix-vector product expression) that the derivative operator distributes over quaternion multiplication, or

$$\dot{q}_{A2B}*q_e+q_{A2B}*\dot{q}_e=\dot{q}_{A2C} \tag{10}$$

Substituting for $\dot{q}_{A2B}$ and $\dot{q}_{A2C}$ using (3),

$$\frac{1}{2}q_{A2B}*\left\{\bar{\omega}_B^A\right\}_B*q_e+q_{A2B}*\dot{q}_e=\frac{1}{2}q_{A2C}*\left\{\bar{\omega}_C^A\right\}_C \tag{11}$$

Multiplying through by q_{A2B}^{-1} from the left,

$$\dot{q}_e=\frac{1}{2}(q_{A2B}^{-1}*q_{A2C}*\left\{\bar{\omega}_C^A\right\}_C-\left\{\bar{\omega}_B^A\right\}_B*q_e)\ \text{ or} \tag{12}$$

$$\dot{q}_e=\frac{1}{2}(q_e*\left\{\bar{\omega}_C^A\right\}_C-\left\{\bar{\omega}_B^A\right\}_B*q_e)\,. \tag{13}$$

To drive $\dot{q}_e$ to some desired $\dot{q}_{e\_des}$, this kinematic expression can be inverted to obtain the required commanded angular rate

$$\left\{\bar{\omega}_B^A\right\}_{B\_CMD}=-(2\dot{q}_{e\_des}-q_e*\left\{\bar{\omega}_C^A\right\}_C)*q_e^{-1} \tag{14}$$

An equivalent simple matrix-vector expression can be obtained by noting $q_e*\left\{\bar{\omega}_C^A\right\}_C=Q_{e1}\left\{\omega_C^A\right\}_C$ and $\left\{\bar{\omega}_B^A\right\}_B*q_e=Q_{e2}\left\{\omega_B^A\right\}_B$ where

$$Q_{e2} = \begin{bmatrix} -q_{e1} & -q_{e2} & -q_{e3} \\ q_{e0} & q_{e3} & -q_{e2} \\ -q_{e3} & q_{e0} & q_{e1} \\ q_{e2} & -q_{e1} & q_{e0} \end{bmatrix} \tag{15}$$

and Q_{e1} is defined in (4). Since the columns of Q_{e2} are orthonormal, (13) can be solved for $\left\{\omega_B^A\right\}_{B\_CMD}$ as

$$\left\{\omega_B^A\right\}_{B\_CMD} = -Q_{e2}^T(2\dot{q}_{e\_des} - Q_{e1}\left\{\omega_C^A\right\}_C). \tag{16}$$

To construct a corrective $\dot{q}_{e\_des}$, define the tracking error as

$$\Delta q_e = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T - \left(\overset{A}{q_e}\right) \tag{17}$$

where $\left(\overset{A}{q_e}\right)$ denotes that a sign change is applied to q_e to make the first entry $q_{e0} \geq 0$. While it is true that quaternions q_e and $-q_e$ correspond to the same relative attitude, using the one with $q_{e0} < 0$ will produce an error direction that will force the object to rotate the long way around to the target attitude. In the control, we will exclusively use this 'aligned' q_e for all occurrences of q_e (e.g., definitions of Q_{e1} and Q_{e2}) up to the calculation of $\left\{\omega_B^A\right\}_{B\_CMD}$. To simplify nomenclature, q_e will denote $\left(\overset{A}{q_e}\right)$.

A PID gain set will set the desired $\dot{q}_{e\_des}$ as

$$\dot{q}_{e\_des} = Q_{e1}Q_{e1}^T(k_p\Delta q_e + k_i\int \Delta q_e dt + k_{de}\Delta\dot{q}_e) \tag{18}$$

where the orthogonal projection $Q_{e1}Q_{e1}^T$ assures that $\dot{q}_{e\_des}$ is directionally realizable since it is always orthogonal to q_e, i.e. $\Re(Q_{e1}) \perp q_e$. From (13), the rate term can be expanded as $\Delta\dot{q}_e = -\dot{q}_e = -\frac{1}{2}(q_e * \left\{\bar{\omega}_C^A\right\}_C - \left\{\bar{\omega}_B^A\right\}_B * q_e)$.

The control is summarized in figure 1 and utilizes (16) and (17). The control is implemented digitally using a trapezoidal integration scheme for the integrated error. The three scalar gains in (18) can be selected using any PID gain optimizer, such as the one available in the MATLAB Control Design Toolbox.

One controller input not discussed thus far is the angular rate of the commanded attitude $\left\{\omega_C^A\right\}_C$. Since this is usually not available, the input must be constructed from signal q_{A2C}. To do this, care must be taken when the sign changes arbitrarily between the previous and current samples, $q_{A2C}(k-1)$ and $q_{A2C}(k)$ respectively. The approach taken here is to first define the two possibilities: $\dot{q}_M = (q_{A2C}(k) - q_{A2C}(k-1))/\Delta t$ or $\dot{q}_P = (q_{A2C}(k) + q_{A2C}(k-1))/\Delta t$ where Δt is the sample time. Then, simply choose the estimate with the smallest inner product, e.g. if $\dot{q}_M \cdot \dot{q}_M < \dot{q}_P \cdot \dot{q}_P$, then $\tilde{\dot{q}}_{A2C} = \dot{q}_M$. Using the definition of Q_1 from (4) with the current sample $q_{A2C}(k)$ defining the matrix elements, the estimate of $\left\{\omega_C^A\right\}_C$ can be obtained as

$$\left\{\tilde{\omega}_C^A(k)\right\}_C = 2Q_1^T(k)\tilde{\dot{q}}_{A2C}(k). \tag{19}$$

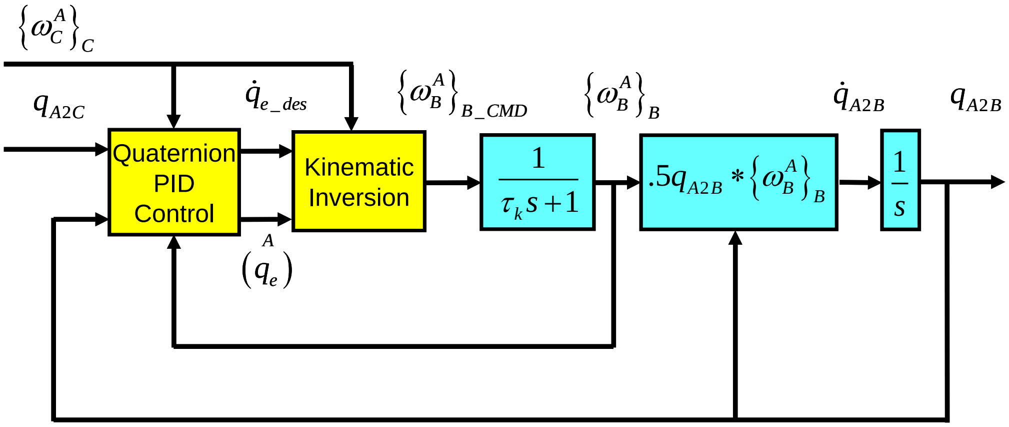


Figure 1. Quaternion-based PID control example

Tracking results, with q_{A2B} and q_{A2C} decomposed into 1-3-2 Euler angles, are shown in figure 2 for two different choices of gain k_{de} on $\Delta\dot{q}_e$. The simple example presented in this section demonstrates the potential of a quaternion-based control for good angular tracking with the characteristics of a low-order response for each attitude angle. Regarding the Euler angles in figure 2, the reader is directed to [5] for a method of extracting the Euler angles from the terms of the corresponding direction cosine matrix which equals the matrix in equation (6).

As stated, key features of the example will appear in both the guidance and the control loops of the proposed quaternion-based control architecture. Those features include

1) constructing q_e that defines the attitude change required to drive the actual to the desired frame,

2) using a PID gain set to define a desired rate $\dot{q}_{e\_des}$ to drive q_e to $[1 \quad 0 \quad 0 \quad 0]^T$ representing perfect tracking, and

3) using $\dot{q}_{e\_des}$ in an inverse kinematic expression to define a desired angular rate $\{\omega_B^A\}_{B\_CMD}$ of the actual frame *B* to move it towards the commanded frame *C*.

Note this angular rate is expressed with respect to frame *A*. Interestingly, there is no restriction in the development thus far that frame *A* is stationary. Frame *A* could be rotating with respect to some other frame *D* and in fact, will be in the next section.

IV. Quaternion-Based Control Architecture

The architecture developed in this section reflects the notion that maneuvering flight is a matter of controlling the vehicle speed along with the lift vector's magnitude and direction in the inertial frame (fixed earth). As many have done [4], the control exploits the two-time-scale nature of the controlled dynamics. Here, the slower loop, utilizing the simple quaternion relationship developed above, controls the orientation of the vehicle with respect to a slower moving frame *N*, generating a set of commanded body rates with respect to frame *N*, $\{\omega_B^N\}_{B\_CMD}$. The attitude of frame *N* is defined by the direction of the vehicle's velocity vector in the inertial frame *I*, i.e., heading and flight path angles. The attitude of the body frame *B* with respect to *N* is defined by three angles: wind-axis bank angle, sideslip

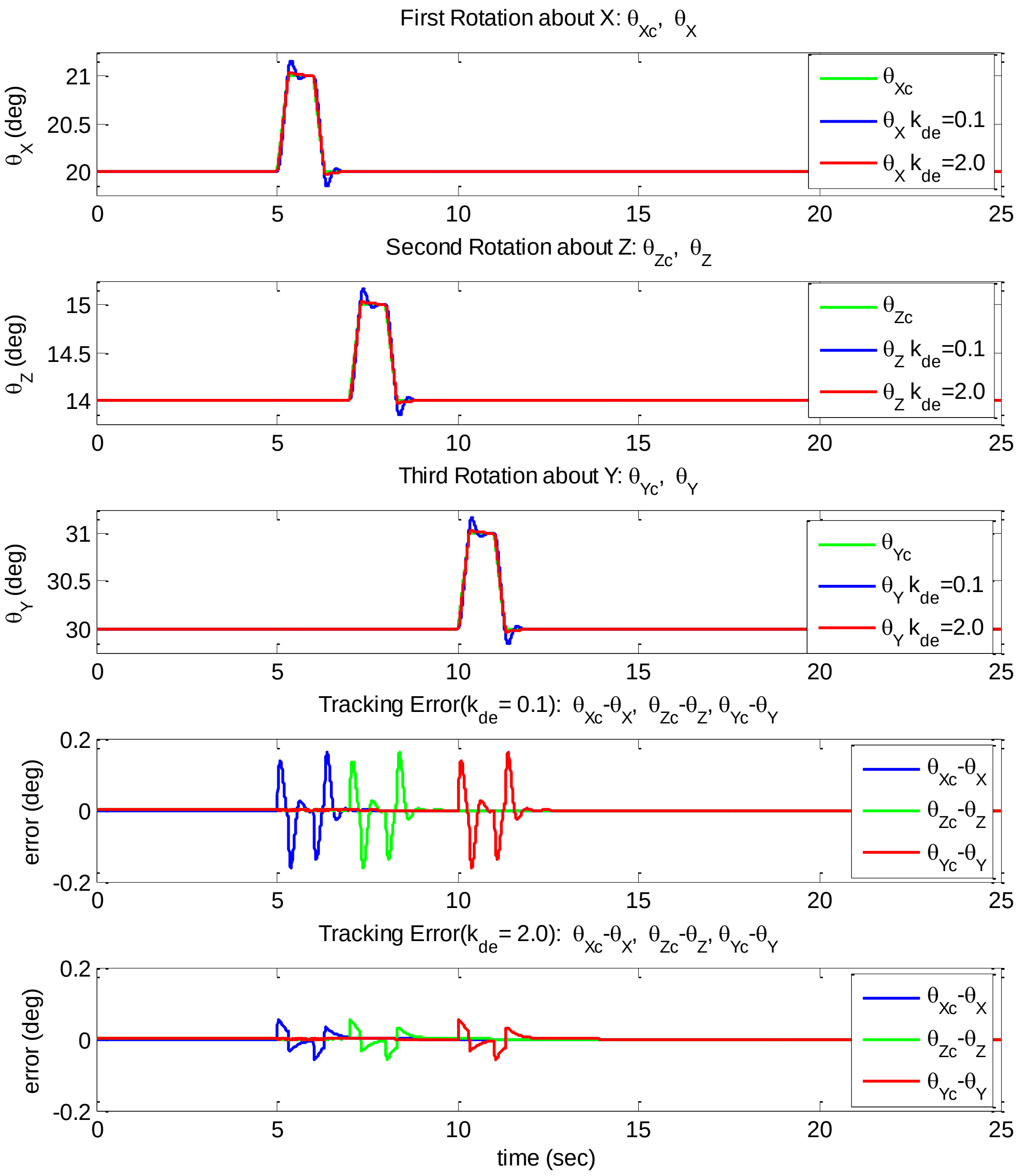


Figure 2. Tracking Results of Quaternion-Based PID Control

angle, and angle of attack. The resulting commanded body rate $\left\{\omega_B^N\right\}_{B\_CMD}$ is then combined with an estimate of the angular rate of *N* with respect to *I* (in body coordinates) to produce the command body rate with respect to *I*, or

$$\left\{\omega_B^I\right\}_{B\_CMD} = \left\{\omega_N^I\right\}_B + \left\{\omega_B^N\right\}_{B\_CMD} . \qquad (20)$$

This angular rate command drives a faster dynamic-inversion-based inner-loop control that determines the required vehicle control moments. To follow candidate trajectories, an outer guidance loop, based on inverted point mass equations with inputs from a second quaternion based PID control, is added to generate the necessary angle of attack, side-slip, and wind-axis bank angle commands along with the required thrust commands. The overall

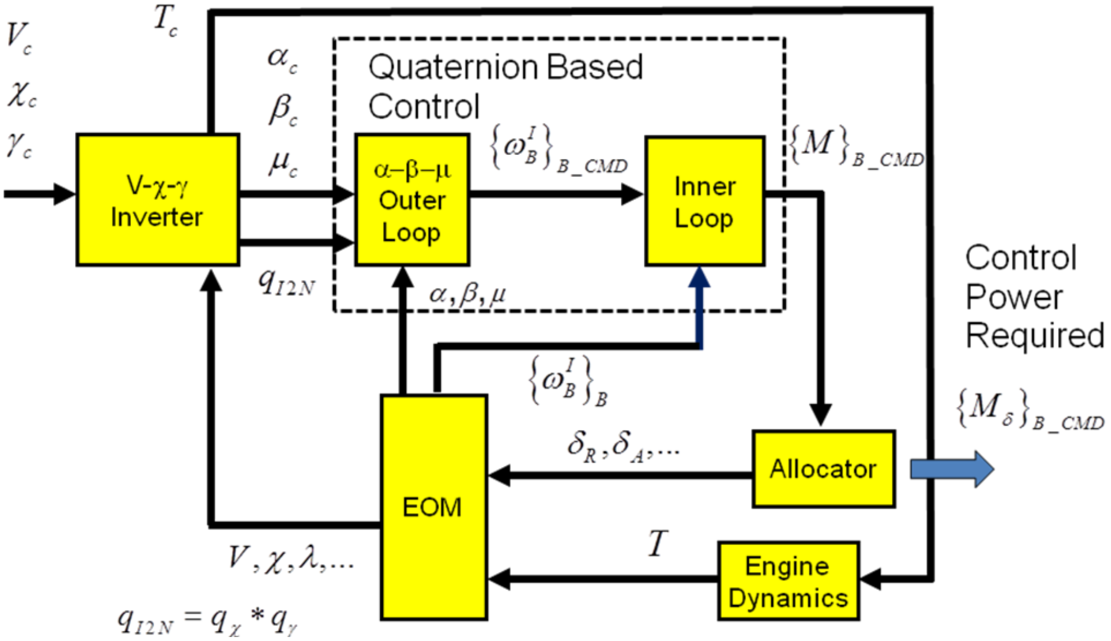


Figure 3. Quaternion-Based Control Architecture

architecture is summarized in figure 3. Subject to limitations in control power, a control allocation algorithm is used to realize the desired control moments if possible.

In the combined guidance and control system, the attitude of the body with respect to frame *N* and the attitude of frame *N* with respect to the inertial frame are expressed as quaternions satisfying

$$q_{N2B} = q_\mu * q_{-\beta} * q_\alpha \tag{21}$$

$$q_{I2N} = q_\chi * q_\gamma \tag{22}$$

where $\mu, \beta, \alpha, \chi,$ and γ respectively denote the angles of wind axis bank, side-slip, angle of attack, heading and flight path. A 1-3-2 set of rotations defines q_{N2B} where it is noted that a negative side-slip rotation is specified about the intermediate *z* axis. The sequence of successive rotations defining q_{I2N} is 3-2. Generally, the vehicle's equations of motion already yield the quaternion of the body with respect to the inertial frame, q_{I2B} . This quaternion enables a more efficient calculation of q_{N2B}, or

$$q_{N2B} = q_{I2N}^{-1} * q_{I2B} \tag{23}$$

where one quaternion multiplication operation is replaced by the simpler inverse operation (changing sign on the rotation vector part of q_{I2N}).

Turning first to the control of q_{N2B}, the quaternion representing the commanded attitude of the body with respect to N, q_{N2C}, is assembled from the commanded wind-axis bank angle, side-slip angle, and angle of attack as

$$q_{N2C} = q_{\mu_c} * q_{-\beta_c} * q_{\alpha_c} \; . \tag{24}$$

The commanded angles will be supplied by the guidance algorithm. The commanded angular rate $\left\{\tilde{\omega}_C^N(k)\right\}_C$ is obtained from (19). As in the example, the relative attitude of the command to the actual attitude is defined as

$$q_e = q_{N2B}^{-1} * q_{N2C} = q_{B2C} \tag{25}$$

with a tracking error expressed as

$$\Delta q_e = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T - (\overset{A}{q_e}) \; . \tag{26}$$

The aligned version of q_e used above will replace q_e in all subsequent calculations leading up to the generation of $\left\{\omega_B^N\right\}_{B\_CMD}$. These calculations include

$$\dot{q}_{e\_des} = Q_{e1} Q_{e1}^T (k_p \Delta q_e + k_i \int \Delta q_e dt + k_{de} \Delta \dot{q}_e) \;\; \text{where} \tag{27}$$

$$\Delta \dot{q}_e = -\dot{q}_e = -\frac{1}{2}(q_e * \left\{\bar{\omega}_C^N\right\}_C - \left\{\bar{\omega}_B^N\right\}_B * q_e) \; . \tag{28}$$

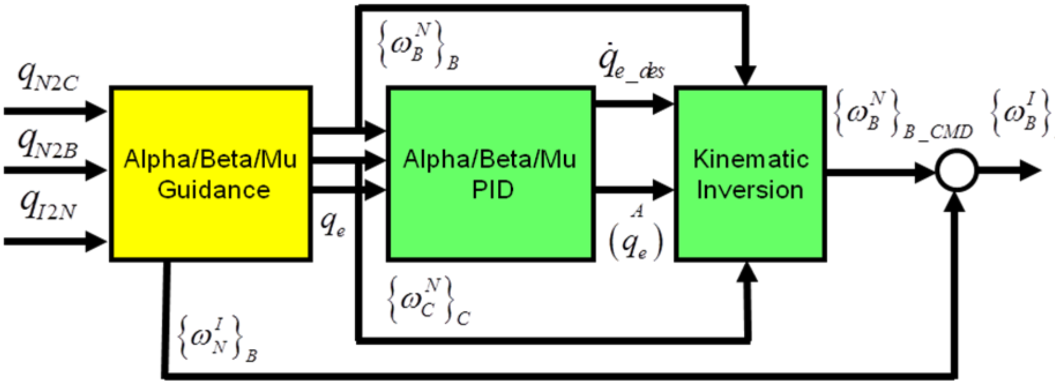
Here, the required actual and commanded angular rates, $\left\{\omega_B^N(k)\right\}_B$ and $\left\{\omega_C^N(k)\right\}_C$, are estimated using (19). The command $\left\{\omega_B^N\right\}_{B\_CMD}$ is defined by the inverse kinematic relation (5),

$$\left\{\omega_B^N\right\}_{B\_CMD} = -Q_{e2}^T (2\dot{q}_{e\_des} - Q_{e1} \left\{\omega_C^N\right\}_C) \tag{29}$$

where the elements of q_e define Q_{e1} and Q_{e2} as in (4) and (15). The equations of motion governing angular acceleration determine $\left\{\omega_B^I\right\}_B$ from the external moments applied to the vehicle. Moments required to drive $\left\{\omega_B^I\right\}_B$ to some commanded $\left\{\omega_B^I\right\}_{B\_CMD}$ will be determined by inverting these equations and replacing the angular acceleration $\left\{{}^B\dot{\omega}_B^I\right\}_B$ with a desired one defined by a stabilizing inner-loop control. The commanded rate $\left\{\omega_B^I\right\}_{B\_CMD}$ is defined as the sum of $\left\{\omega_B^N\right\}_{B\_CMD}$ and the slower-changing $\left\{\omega_N^I\right\}_B$, corresponding to the angular rate of the vehicle's velocity vector with respect to the inertial frame. Here, $\left\{\omega_N^I(k)\right\}_N$ can be estimated from the current and previous value of $q_{I2N}(k)$ using (19) and transformed to the appropriate coordinates via (5), or

$$\left\{\bar{\omega}_N^I\right\}_B = q_{N2B}^{-1} * \left\{\bar{\omega}_N^I\right\}_N * q_{N2B} \tag{30}$$

where the over bar signifies $\left\{\bar{\omega}_B^A\right\}_B = \begin{bmatrix} 0 & \left\{\omega_B^A\right\}_B^T \end{bmatrix}^T$. Technically, one could obtain $\left\{\omega_N^I\right\}_B$ as $\left\{\omega_B^I\right\}_B - \left\{\tilde{\omega}_B^N\right\}_B$ since $\left\{\omega_B^I\right\}_B$ is from the equations of motion. This is not recommended, though, because $\left\{\omega_B^I\right\}_B$ and $\left\{\tilde{\omega}_B^N\right\}_B$ are not exactly in sync and their difference can produce undesirable transients not found in the estimated $\left\{\tilde{\omega}_N^I\right\}_B$. The elements of the $\alpha - \beta - \mu$ (q_{N2B}) outer-loop are summarized in figure 4.



$$q_{N2C} = q_{\mu_c} * q_{-\beta_c} * q_{\alpha_c} \qquad q_{N2B} = q_{\mu} * q_{-\beta} * q_{\alpha} \qquad q_{I2N} = q_{\chi} * q_{\gamma}$$

Figure 4. Quaternion-based outer loop control architecture

The architecture of the quaternion-based control exploits the slow/fast time scales separating the vehicle's responses of α, β, μ and $\{\omega_B^I\}_B$. Deployment of the vehicle's effectors mainly create moments that are integrated to yield $\{\omega_B^I\}_B$. A second integration on angular rate $\{\omega_B^I\}_B$ (actually, integrands dominated by angular rate) produce changes in α, β and μ. The outer loop assumes that the commanded $\{\omega_B^I\}_{B\_CMD}$ can be quickly realized by $\{\omega_B^I\}_B$ in order to drive α, β and μ to their commanded values.

Here, the faster inner-loop control of $\{\omega_B^I\}_B$ results directly from equations of motion governing the vehicle's angular acceleration

$$\left\{{}^{I}\dot{H}_B^I\right\}_B = I\left\{{}^{B}\dot{\omega}_B^I\right\}_B + \left\{\omega_B^I\right\}_B \times I\left\{\omega_B^I\right\}_B = \left\{M_T + M_A + \sum_k^{n_\delta} M_{\delta k}\right\}_B . \tag{31}$$

Above, the angular momentum $\{H_B^I\}_B$ about the vehicle's center of mass satisfies $\{H_B^I\}_B = I\{\omega_B^I\}_B$ where I, approximated as a constant, is the inertia matrix defined about the mass center. M_T, M_A and $M_{\delta k}$ denote the external moments about the vehicle's mass center due to thrust, baseline aerodynamics, i.e. with zero control deflections, δ_k, and the aerodynamic increment of each effector δ_k. Independent parameters for the moments include angle of attack, side-slip angle, dynamic pressure, effector position, and center of mass locations.

Dynamic inversion can be used to define $\sum_k^{n_\delta} M_{\delta k}$ such that $\left\{{}^{B}\dot{\omega}_B^I\right\}_B = \left\{{}^{B}\dot{\omega}_B^I\right\}_{B\_des}$,

$$\left\{\sum_k^{n_\delta} M_{\delta k}\right\}_B = I\left\{{}^{B}\dot{\omega}_B^I\right\}_{B\_des} + \left\{\omega_B^I\right\}_B \times I\left\{\omega_B^I\right\}_B - \left\{M_T + M_A\right\}_B . \tag{32}$$

A stabilizing control defines $\left\{{}^{B}\dot{\omega}_{B}^{I}\right\}_{B\_des}$ as $\left\{{}^{B}\dot{\omega}_{B}^{I}\right\}_{B\_des} = \mathrm{K}_{i}(\left\{\omega_{B}^{I}\right\}_{B\_CMD} - \left\{\omega_{B}^{I}\right\}_{B})$ where $\mathrm{K}_{i} = diag\,(\begin{bmatrix} k_{pb} & k_{qb} & k_{rb} \end{bmatrix})$. Assuming there is sufficient control power, or there exists a $\sum_{k}^{n_{\delta}} M_{\delta k}$ that satisfies equation (32), the closed-loop response for this faster inner-loop is

$$\left\{\omega_{B}^{I}\right\}_{B} \Big/ \left\{\omega_{B}^{I}\right\}_{B\_CMD}(s) = diag(\begin{bmatrix} \dfrac{k_{pb}}{s+k_{pb}} & \dfrac{k_{qb}}{s+k_{qb}} & \dfrac{k_{rb}}{s+k_{rb}} \end{bmatrix}) \,. \tag{33}$$

Limits on control power include position and rate limits for the vehicle's various effectors and are dealt with in the next section on control allocation.

V. Control Allocation

Since the inner-loop control and control allocation problem deal solely with vectors in body axis coordinates, the bracket designation is dropped and (32) is expressed as

$$\sum_{k}^{n_{\delta}} M_{\delta k} = \sum_{k=1}^{n_{1}} M_{\delta k}(\bar{q}^{0}, \alpha^{0}, \beta^{0}, \delta_{k}) + \sum_{j=n_{1}+1}^{n_{\delta}} M_{\delta k}(\bar{q}^{0}, \alpha^{0}, \beta^{0}, \delta_{l}, \delta_{j}) = M_{\delta\_des} \tag{34}$$

where $1 \le l \le n_{1}$, $\bar{q}$ denotes dynamic pressure , and M_{δ_des} denotes the target moment corresponding to the right side of equation (32) and superscript (^{0}) denotes the current value. The problem is to find δ_{k} that satisfies (34), subject to rate and position limits, with the slower varying $\bar{q}, \alpha, \beta$ parameters set to their current values. The control moment increments considered are one of two types: One is a function of a single effector deflection and the other is a function of two, modeling the interference effects of one effector on another. For this development, assume that the current values of $\bar{q}, \alpha, \beta,$ and δ_{k} are available. Expand (34) as a Taylor Series approximation about the current value as

$$\sum_{k}^{n_{\delta}} M_{\delta k}^{0} + \sum_{k=1}^{n_{1}} b_{\delta k}^{0} \Delta\delta_{k} + \sum_{k=n_{1}+1}^{n_{\delta}} b_{\delta k}^{0} \Delta\delta_{k} \approx M_{\delta\_des} \quad \text{where} \tag{35}$$

$$b_{\delta k}^{0} = \frac{\partial M_{\delta k}}{\partial \delta_{k}}(\bar{q}^{0}, \alpha^{0}, \beta^{0}, \delta_{k}^{0}) + \sum_{j=1}^{n_{3}} \frac{\partial M_{\delta j}}{\partial \delta_{k}}(\bar{q}^{0}, \alpha^{0}, \beta^{0}, \delta_{k}^{0}, \delta_{j}^{0}) \quad \text{for} \;\; 1 \le k \le n_{1},\; j > n_{1}$$

$$b_{\delta k}^{0} = \frac{\partial M_{\delta k}}{\partial \delta_{k}}(\bar{q}^{0}, \alpha^{0}, \beta^{0}, \delta_{l}^{0}, \delta_{k}^{0}) \;\; \text{for} \;\; 1 \le l \le n_{1},\; k > n_{1}, \text{ and}$$

$$\sum_{k}^{n_{\delta}} M_{\delta k}^{0} = \sum_{k=1}^{n_{1}} M_{\delta k}(\bar{q}^{0}, \alpha^{0}, \beta^{0}, \delta_{k}^{0}) + \sum_{k=n_{1}+1}^{n_{\delta}} M_{\delta k}(\bar{q}^{0}, \alpha^{0}, \beta^{0}, \delta_{k}^{0}, \delta_{j}^{0}) \,. \tag{36}$$

With this approximation, the control allocation problem can be expressed on an incremental level, with simpler notation, as one of finding some optimal $\Delta u = \begin{bmatrix} \Delta\delta_{1} & \Delta\delta_{2} & \cdots & \Delta\delta_{n_{\delta}} \end{bmatrix}^{T}$ satisfying

$$B^{0} \Delta u = m_{des} = M_{\delta\_des} - \sum_{k}^{n_{\delta}} M_{\delta k}^{0} \tag{37}$$

where $B^0 = \begin{bmatrix} b_{\delta 1} & b_{\delta 2} & \cdots & b_{\delta n_\delta} \end{bmatrix}$, subject to all the limitations of rate and position on the elements of Δu. The resulting control is defined as $u = u^0 + \Delta u$.

In [3], the standard minimum norm solution of (37), that essentially minimizes control rate, is replaced by one that minimizes both control rate and position. The quadratic cost function is

$$J = .5\left(\Delta u/\Delta t\right)^T W_{rr}\left(\Delta u/\Delta t\right) + .5\left(u^0 + \Delta u\right)^T W_{pp}\left(u^0 + \Delta u\right) \tag{38}$$

where W_{rr} and W_{pp} are rate and position diagonal weighting matrices, respectively. To minimize (38) subject to (37), define the Lagrangian equation

$$L = .5\Delta u^T W_r \Delta u + .5\left(u^0 + \Delta u\right)^T W_{pp}\left(u^0 + \Delta u\right) + \lambda^T\left(B^0 \Delta u - m_{Des}\right) \tag{39}$$

where $W_r = W_{rr}/\Delta t^2$. The necessary condition for optimal Δu [6] yields

$$\begin{aligned}\Delta u &= \bar{W}^{-1} B^{0T}\left(B^0 \bar{W}^{-1} B^{0T}\right)^{-1} m_{des} - \\ &\quad \left[I_{n\delta} - \bar{W}^{-1} B^{0T}\left(B^0 \bar{W}^{-1} B^{0T}\right)^{-1} B^0\right]\bar{W}^{-1} W_{pp} u^0 \\ &= B^{0\#}_{\bar{W}} m_{des} - \left[I_{n\delta} - B^{0\#}_{\bar{W}} B^0\right]\bar{W}^{-1} W_{pp} u^0\end{aligned} \tag{40}$$

where $\bar{W} = W_r + W_{pp}$ and $B^{0\#}_{\bar{W}} = \bar{W}^{-1} B^{0T}\left(B^0 \bar{W}^{-1} B^{0T}\right)^{-1}$. If $W_{pp} = 0$, (40) yields the minimum norm solution. The first term in (40) satisfies (37). The second term in (40) is in the null space of $col\left\{B^0\right\}$ and provides the correction to reduce $\left\|u^0 + \Delta u\right\|^2_{W_{pp}}$.

Although (40) minimizes effector rate and position activity, tending to zero counterbalanced effectors, it does not guarantee the commanded controls will be within their saturation limits. A multi-pass strategy is added to the allocator solution to reduce violations of limits.

To satisfy both position and rate constraints of the effectors, the incremental control is constrained prior to commanding the hardware as

$$\begin{gathered} u_{l,\lim} - u_0 \le \Delta u \le u_{u,\lim} - u_0 \\ \dot{u}_{l,\lim}\Delta t \le \Delta u \le \dot{u}_{u,\lim}\Delta t \end{gathered} \tag{41}$$

where $\left(u_{l,\lim}, u_{u,\lim}\right)$ and $\left(\dot{u}_{l,\lim}, \dot{u}_{u,\lim}\right)$ are respectively the lower and upper position and rate limits of the effectors applied to the control command $u_{cmd} = u^0 + \Delta u$. Let $k = 1$ correspond to the first pass solution given by (40), i.e., $\Delta u^{(1)} = \Delta u$, and assume that at least one control effector command $\Delta u_i^{(1)}$ violates a saturation limit. For any $k \ge 1$, define $\Delta u_{\lim}^{(k)} \in R^{n\delta}$ as follows. If $\Delta u_i^{(k)}$, $i = 1, n_\delta$, violates one of the constraints in (41), denoted as $\Delta \bar{u}_i$, set $\Delta u_{\lim,i}^{(k)} = \Delta \bar{u}_i$. Otherwise, set $\Delta u_{\lim,i}^{(k)} = 0$.

The portion of m_{des} achieved by the constrained effectors is

$$m_{des,\lim}^{(k)} = B^0 \Delta u_{\lim}^{(k)} . \tag{42}$$

For the (*k*+1)-pass solution, solve first for $\Delta u^{(k+1)}$

$$B^0 \Delta u_0^{(k+1)} = m_{des} - m_{des,\lim}^{(k)} = f_{des}^{(k)} \tag{43}$$

using (40) with $f_{des}^{(k)}$ replacing m_{des} and the diagonal elements of $\bar{W}^{-1}$ corresponding to constrained elements (all *i* such that $\Delta u_{\lim,i}^{(k)} \neq 0.$) set to zero. From (40), only the unconstrained controls are used in the solution: elements of $\Delta u_0^{(k+1)}$ corresponding to constrained effectors are zero. The incremental control solution for the $(k+1)$ -pass is

$$\Delta u^{(k+1)} = \Delta u_0^{(k+1)} + \Delta u_{\lim}^{(k)} . \tag{44}$$

If $\Delta u^{(k+1)}$ satisfies (41), set $\Delta u = \Delta u^{(k+1)}$. Otherwise, set $k = k+1$ and repeat. To avoid driving all controls into saturation when excessive control power is required, limit $k \leq 2$. Note that the first pass corresponds to $\Delta u_{\lim,i}^{(0)} = 0$ for all $i = 1, n_\delta$.

Implementation of the control allocator requires tables describing both force and moment coefficients generated by the baseline aerodynamics and the increments due to effector position. Furthermore, tables are required of the partial derivatives of the force and moment increments to effector position. In the simulation, the aero database is linearly interpolated across all independent parameters. The simplest approach to constructing the partial derivative tables is to double the effector's intermediate breakpoints $\delta_k^{b,i} \rightarrow \delta_k^{b,i} - \varepsilon, \delta_k^{b,i} + \varepsilon$, providing the required support to the constant slope between the original breakpoints indicative of linear interpolation. All the tabular data express the moment about a specific aerodynamic reference, so these must typically be translated to vehicle's center of mass using the force data to create $M_{\delta k}^0$ and B^0 in (37).

It is worth noting that the control allocation algorithm produces an approximate solution to (34), even if the simulation uses linear interpolation. Over the multi-pass iterations, B^0 does not change. As a result, the solution generated by the control allocator in the simulation is exact only when no control solution passes a table breakpoint due to the change in slope --even if no control violates its saturation limits. Since many effectors have different slopes on either side of zero deflection, differences can be expected in this neighborhood of small effector deflections. Only if the effectors effectiveness are strictly linear (not piecewise) will there be no control allocation error.

VI. Guidance

In figure 3, the closed-loop guidance law is responsible for providing the command $q_{N2C} = q_{\mu_c} * q_{-\beta_c} * q_{\alpha_c}$ and a thrust command T_c required to follow a desired trajectory. The actual trajectory is specified by both the speed, $V \triangleq \left\{ V_N^I \right\}_{N,x}$ (the origin of *N* is attached to vehicle's center of mass) and the orientation of the *N* with respect to *I* specified by $q_{I2N} = q_\chi * q_\gamma$ as a function of time. The desired trajectory is defined by time histories of the commanded $q_{I2G} = q_{\chi_c} * q_{\gamma_c}$ and $V_{CMD} \triangleq \left\{ V_G^I \right\}_{G,x}$.

To obtain the desired guidance outputs, consider the point-mass equation of motion expressed in the *N*-frame

$$M\left\{{}^{I}\dot{V}_{N}^{I}\right\}_{N}=M(\left\{{}^{N}\dot{V}_{N}^{I}\right\}_{N}+\left\{\omega_{N}^{I}\right\}_{N}\times\left\{V_{N}^{I}\right\}_{N})=\left\{F_{A}\right\}_{N}+\left\{F_{T}\right\}_{N}+\left\{F_{G}\right\}_{N} \tag{45}$$

where F_A, F_T, and F_G respectively denote the aerodynamic, thrust, and gravitational force, and M is the vehicle mass. Let the aerodynamic forces be expressed in the wind-axis frame *W* in terms of lift (*L*), drag (*D*), and side force (*Y*) as $\left\{F_A\right\}_W = -D\hat{w}_x + Y\hat{w}_y - L\hat{w}_z$. Wind axis bank angle orients *W* from *N* as $q_{N2W} = q_\mu$ requiring q_μ^{-1} to transform $\left\{F_A\right\}_W$ to $\left\{F_A\right\}_N$. The thrust force is generally expressed in body axis as $\left\{F_T\right\}_B = T_x\hat{b}_x + T_y\hat{b}_y + T_z\hat{b}_z$ requiring q_{N2B}^{-1} to transform it to frame *N*. Assuming flat earth, the gravitational force is expressed in the inertial frame as $\left\{F_G\right\}_I = Mg\hat{i}_z$ and transformed to frame *N* with q_{I2N}.

Transforming the forces to the *N* frame (using direction cosine matrices to avoid manipulating half-angle sines and cosines [5]) and noting $\left\{V_N^I\right\}_N = V\hat{n}_x$, (45) can be expressed in component form as

$$M\dot{V}=-D+T_x\cos(\beta)\cos(\alpha)+T_y\sin(\beta)+T_z\cos(\beta)\sin(\alpha)-Mg\sin(\gamma) \tag{46}$$

$$\begin{aligned}MV\left\{\omega_N^I\right\}_{N,z}=&\,Y\cos(\mu)+L\sin(\mu)\\&+T_x\left(\sin(\mu)\sin(\alpha)-\cos(\mu)\sin(\beta)\cos(\alpha)\right)\\&+T_y\left(\cos(\mu)\cos(\beta)\right)\\&-T_z\left(\sin(\mu)\cos(\alpha)+\cos(\mu)\sin(\beta)\sin(\alpha)\right)\end{aligned} \tag{47}$$

$$\begin{aligned}-MV\left\{\omega_N^I\right\}_{N,y}=&\,Y\sin(\mu)-L\cos(\mu)+Mg\cos(\gamma)\\&-T_x\left(\cos(\mu)\sin(\alpha)+\sin(\mu)\sin(\beta)\cos(\alpha)\right)\\&+T_y\left(\sin(\mu)\cos(\beta)\right)\\&+T_z\left(\cos(\mu)\cos(\alpha)-\sin(\mu)\sin(\beta)\sin(\alpha)\right)\end{aligned} \tag{48}$$

Assume that Y, T_y and T_z are zero and that the control design will keep β close to zero so that (46-48) simplify to

$$M\dot{V}=-D+T_x\cos(\alpha)-Mg\sin(\gamma) \tag{49}$$

$$MV\left\{\omega_N^I\right\}_{N,z}=L\sin(\mu)+T_x\left(\sin(\mu)\sin(\alpha)\right) \tag{50}$$

$$-MV\left\{\omega_N^I\right\}_{N,y}=-L\cos(\mu)+Mg\cos(\gamma)-T_x\left(\cos(\mu)\sin(\alpha)\right) \tag{51}$$

Required T_x, α and μ commands will be obtained by inverting equations (49-51) to satisfy desired values $\dot{V}_{des}$, $\left\{\omega_N^I\right\}_{N,z\_des}$ and $\left\{\omega_N^I\right\}_{N,y\_des}$ designed to drive the actual trajectory to the desired one. The desired $\dot{V}_{des}$ will be defined as

$$\dot{V}_{des}=k_v\left(V_{CMD}-V\right). \tag{52}$$

The desired angular rate will be obtained as $\left\{\omega_N^I\right\}_{N\_CMD}$ from a quaternion-based control directly taken from the simple example utilizing q_{I2N} and the commanded q_{I2G}. Assume for the inversion, that the slower varying parameters V and γ are set to their current values, $V=V^0$ and $\gamma=\gamma^0$.

To obtain μ_c, subtract the gravity term in (51) from both sides of the equation and divide the result into (50) to yield

$$\tan\left(\mu_c\right)=\frac{MV^0\left\{\omega_N^I\right\}_{N,z\_des}}{MV^0\left\{\omega_N^I\right\}_{N,y\_des}+Mg\cos\left(\gamma^0\right)} \tag{53}$$

Here it is assumed that V^0 and $\cos\left(\gamma^0\right)$ are not both zero.[4]

To obtain the thrust command, T_c, and α_c, first multiply (50) by $\sin\left(\mu\right)$ and subtract from it (51) multiplied by $\cos\left(\mu\right)$ to yield

$$\begin{aligned} L\left(V^0,\alpha_c,\delta_k^0\right)+T_c\sin\left(\alpha_c\right)=MV^0\left\{\omega_N^I\right\}_{N,z\_des}\sin\left(\mu_c\right)& \\ +M\left(V^0\left\{\omega_N^I\right\}_{N,y\_des}+g\cos\left(\gamma^0\right)\right)\cos\left(\mu_c\right)& \end{aligned} \tag{54}$$

Equation (54) shows the dependency of lift on angle of attack, airspeed (through dynamic pressure) and effector position. Lift is also dependent on side-slip which is assumed to be commanded to zero. Equation (54) along with (49), or

$$D\left(V^0,\alpha_c,\delta_k^0\right)-T_c\cos\left(\alpha_c\right)=-M\dot{V}_{des}-Mg\sin\left(\gamma^0\right) \tag{55}$$

provide the two equations to be solved for T_c and α_c. Due to the nonlinear dependency of L and D on α_c an iterative procedure is used to solve for T_c and α_c. Since μ_c is set by (53) and the current values γ^0 and V^0 remain constant over the iterative process, the following shorthand for (54-55) is adopted.

$$\begin{aligned} f_L\left(\alpha_c,T_c\right)&=L\left(V^0,\alpha_c,\delta_k^0\right)+T_c\sin\left(\alpha_c\right)=C_{lift} \\ f_D\left(\alpha_c,T_c\right)&=D\left(V^0,\alpha_c,\delta_k^0\right)-T_c\cos\left(\alpha_c\right)=C_{drag} \end{aligned} \tag{56}$$

Here, an iterative Newton method [7], based on the Taylor series expansion of the (56) at current iterative values $\alpha_c\left(k\right)$ and $T_c\left(k\right)$, is used to update the command as $\alpha_c\left(k+1\right)=\alpha_c\left(k\right)+s_\alpha\Delta\alpha_c\left(k\right)$ and $T_c\left(k+1\right)=T_c\left(k\right)+s_T\Delta T_{\,c}(k)$ where s_α and s_T control the step size (<1) with the correction satisfying the linear equations

$$\frac{\partial f_L\left(\alpha,T\right)}{\partial\alpha}\Big|_{\substack{\alpha_c(k)\\T_c(k)}}\Delta\alpha+\frac{\partial f_L\left(\alpha,T\right)}{\partial T}\Big|_{\substack{\alpha_c(k)\\T_c(k)}}\Delta T=C_{lift}-f_L\left(\alpha_c\left(k\right),T_c\left(k\right)\right) \tag{57}$$

$$\frac{\partial f_D\left(\alpha,T\right)}{\partial\alpha}\Big|_{\substack{\alpha_c(k)\\T_c(k)}}\Delta\alpha+\frac{\partial f_D\left(\alpha,T\right)}{\partial T}\Big|_{\substack{\alpha_c(k)\\T_c(k)}}\Delta T=C_{drag}-f_D\left(\alpha_c\left(k\right),T_c\left(k\right)\right) \tag{58}$$

The iteration begins by setting $\alpha_c\left(0\right)=\alpha^0$ and $T_c\left(0\right)=T_x^0$.

Implementation requires additional tables for lift and drag coefficients and their partial derivatives with respect to angle of attack. Generally aerodynamic force coefficients are given with respect to the body axis, and these must be converted to the wind axis for use in the guidance algorithm, i.e.

$$C_D=-\cos\left(\beta\right)\cos\left(\alpha\right)C_X-\sin\left(\beta\right)C_Y-\cos\left(\beta\right)\sin\left(\alpha\right)C_Z \tag{59}$$

$$C_L=\sin\left(\alpha\right)C_X-\cos\left(\alpha\right)C_Z \tag{60}$$

Since linear interpolation is used in the simulation, the partial derivative tables are constructed consistently with those for the control derivatives in the control allocation algorithm. Here, however, there are many more break points for angle of attack than for the effector position. In closing, it should be mentioned that the Newton method may not be the best way to solve (56). The rough approximation of the Newton method restricted the process to two iterations where $s_\alpha=.1$ and $s_T=.4$ to prevent destabilizing the guidance loop. Evidently the linear dependency on thrust in (56) allows a larger correction for T_c.

Having described the quaternion-based control, an example is presented in the next section. To be noted, the two PID controllers for guidance and control use six gains, the inner loop controlling body rate $\left\{\omega_B^I\right\}_B$ uses three gains, and the definition of $\dot{V}_{des}$ uses one for a total of ten gains to be selected.

VII. Example

To demonstrate both the guidance and control aspects of the algorithm, the subscale GTM vehicle [8] with 17 effectors is flown first in simulation with the guidance loop open. Commands for angle of attack , wind-axis bank angle and sideslip angle are direct inputs to the $\alpha-\beta-\mu$ outer-loop control. A flight path trajectory is obtained from the resulting maneuver and then used to fly the system with the guidance system engaged. At this stage in the tool's development, the guidance law is set to follow a known-reachable trajectory. For our purposes, the vehicle is trimmed in a level turn to the pilot's right with the vehicle banked 40 degrees right wing down (trimmed $\alpha=4.34^\circ$, $V_{eas}=$ 95 knots). At 5 seconds, the pilot pulls up one degree of angle of attack and at 10 seconds reverses the bank angle to -40 degrees producing a S-shaped maneuver. The throttle setting is unchanged, although thrust changes due to changing flight conditions.

This flight path generation maneuver provides an initial check on the performance of the outer loop $\alpha-\beta-\mu$ control, the inner-loop control of $\left\{\omega_B^I\right\}_B$, and the control allocator. The GTM has an expanded suite of 17 effectors resulting from split elevator, spoilers, and flaps, each with four surfaces designated as right/left and inboard/outboard. The rudder is split into upper and lower surfaces, but the right and left ailerons are not split. The stabilizer has also been actuated, though this capability does not exist in the flight test vehicle, to exercise the allocator ability to handle surfaces that influence one another. In this case it is the effectiveness of the elevator that is changed due to the position of the stabilizer. The actuators driving these surfaces are all modeled as first-order systems with 5 Hz bandwidth and a rate limit of 300 deg/sec. The position limits vary for the surfaces and are given in table 1. The GTM example includes effectors that can be deployed positively or negatively as well as some (spoilers and flaps) that do not have negative deflections, providing a good test of the proposed allocation algorithm. All aerodynamics associated with the baseline and the surface increments were obtained from wind tunnel testing, with the exception of the flaps, which were modeled with a constant control derivative.

The tracking results for the outer loop $\alpha-\beta-\mu$ control are shown in figure 5. The largest overshoots occur in response to most aggressive portion of the maneuver, the 80 degree change in wind-axis bank angle. But the overshoot of μ is less than two degrees. Side-slip angle is maintained close to zero, while the angle of attack is slightly perturbed from its one degree increase during the roll. The inner-loop tracking results for body angular rate are shown in figure 6. The largest overshoot occurs in response to the commanded roll rate: the vehicle's dynamics will not follow transients of that magnitude. The commanded spike is, however, relatively well damped as a result of the PID gains selected.

Effector	Designation	Position Limit	Rate Limit
		(deg)	(deg/sec)
Aileron	L, R	[-20, 20]	300
Stabilizer		[-12, 4]	300
Elevator	LOB,LIB,RIB,ROB	[-30, 20]	300
Rudder	U,L	[-30,30]	300
Spoiler	LOB,LIB,RIB,ROB	[0,[45 15 15 45]]	300
Flap	LOB,LIB,RIB,ROB	[0, 30]	300

Table 1. Effector Position and Rate Limits

To check the performance of the control allocator, the desired moment change required of the effector *k* is defined as $b_k^0\Delta\delta_k$ where b_k^0 is the *k* column of B^0 and $\Delta\delta_k$ is the commanded incremental change in current effector position δ_k^0. Assuming that $\Delta\delta_k$ does not violate any limit on the last pass of the control allocator iterative algorithm, $b_k^0\Delta\delta_k$ is the desired change in moment from the *k* effector. In the example, the linear solution produced the requested moment. To determine how well the control derivative reflected the actual effector control effectiveness, consider the error

$$M_{\delta,e}\left(\alpha^0,\beta^0,\delta_k^0,\Delta\delta_k\right)=b_k^0\Delta\delta_k-\left(M_{\delta_k}\left(\alpha^0,\beta^0,\delta_k^0+\Delta\delta_k\right)-M_{\delta_k}\left(\alpha^0,\beta^0,\delta_k^0\right)\right). \qquad (61)$$

For this maneuver, the largest error was the pitching moment due to the right aileron shown in figure 7. Note that as the right aileron crossed the breakpoint at $\delta_{ail,r}=0$ the slope changed enough to cause a four lbf-ft error in pitching moment. As expected there is no error for the linear flap model in figure 8.

One concern regarding the allocator is illustrated in figure 8. The surfaces tend to end the maneuver at some non-zero deflection with no apparent tendency to retract. This not only occurred for the flaps, but the spoilers as well. The algorithm is formulated to reduce the deflections of counter-balanced effectors, but the simulation result does not indicate success in this example. Evidently more work needs to be done on the allocator method.

To test the guidance law, airspeed, heading and flight path angle data from the previous run are now used as guidance commands with the guidance loop closed. Recall, the solution involved satisfying a the point mass equation of motion expressed in the *N* frame to generate μ_c, α_c and T_c. The guided results in figure 9 reveal good tracking of airspeed, heading and flight path angle with airspeed error within ± 0.2 knots and heading and flight path angle error within ± 0.25 deg. The commanded α_c and μ_c in figure 10 have roughly the same character as the unguided results, but there is definitely more activity due to the guidance-generated α_c. The outer loop $\alpha-\beta-\mu$ control still follows the commanded signals. Interestingly the desired $\left\{\omega_N^I\right\}_{N\_CMD}$ used in point mass equation is realized by the actual N frame in figure 11. Not shown, the effector deployment well approximates that of the unguided result.

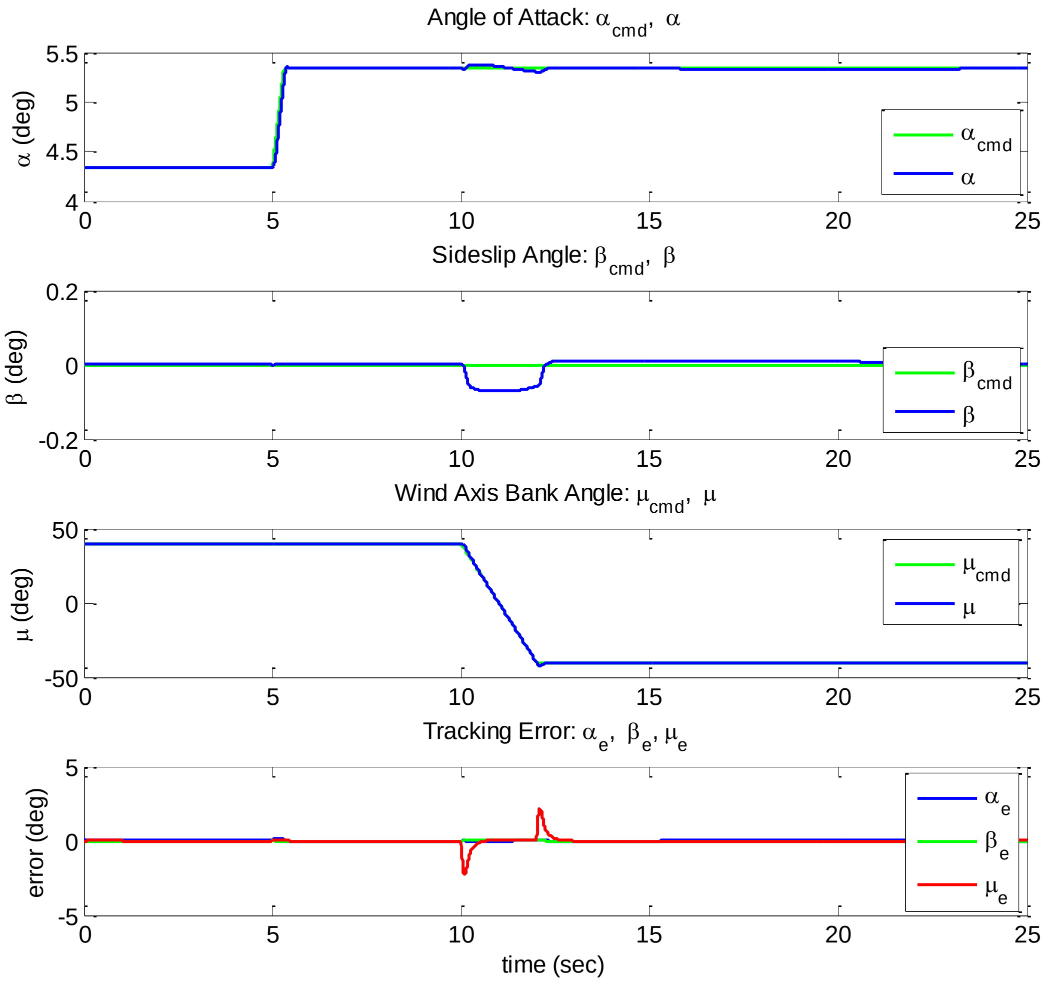


Figure 5. Maneuver generating response

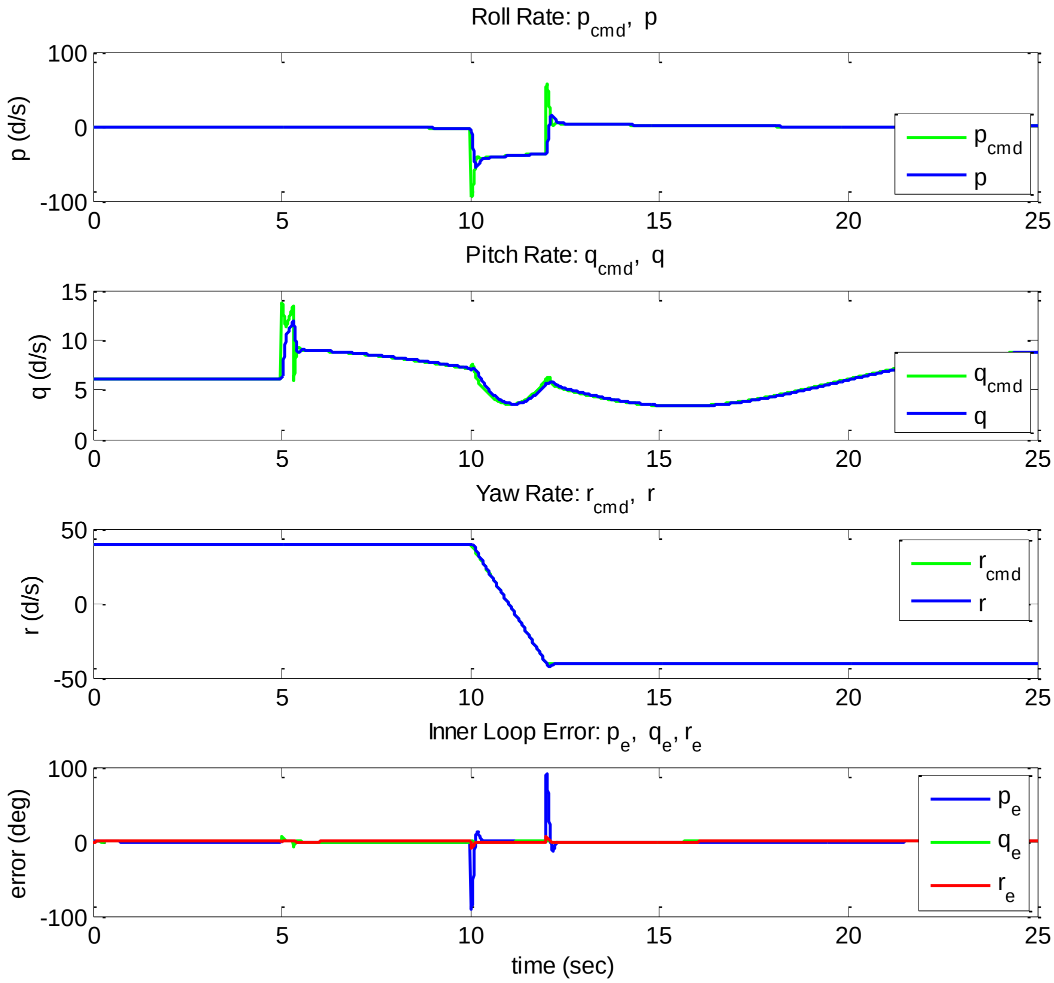


Figure 6. Inner loop tracking

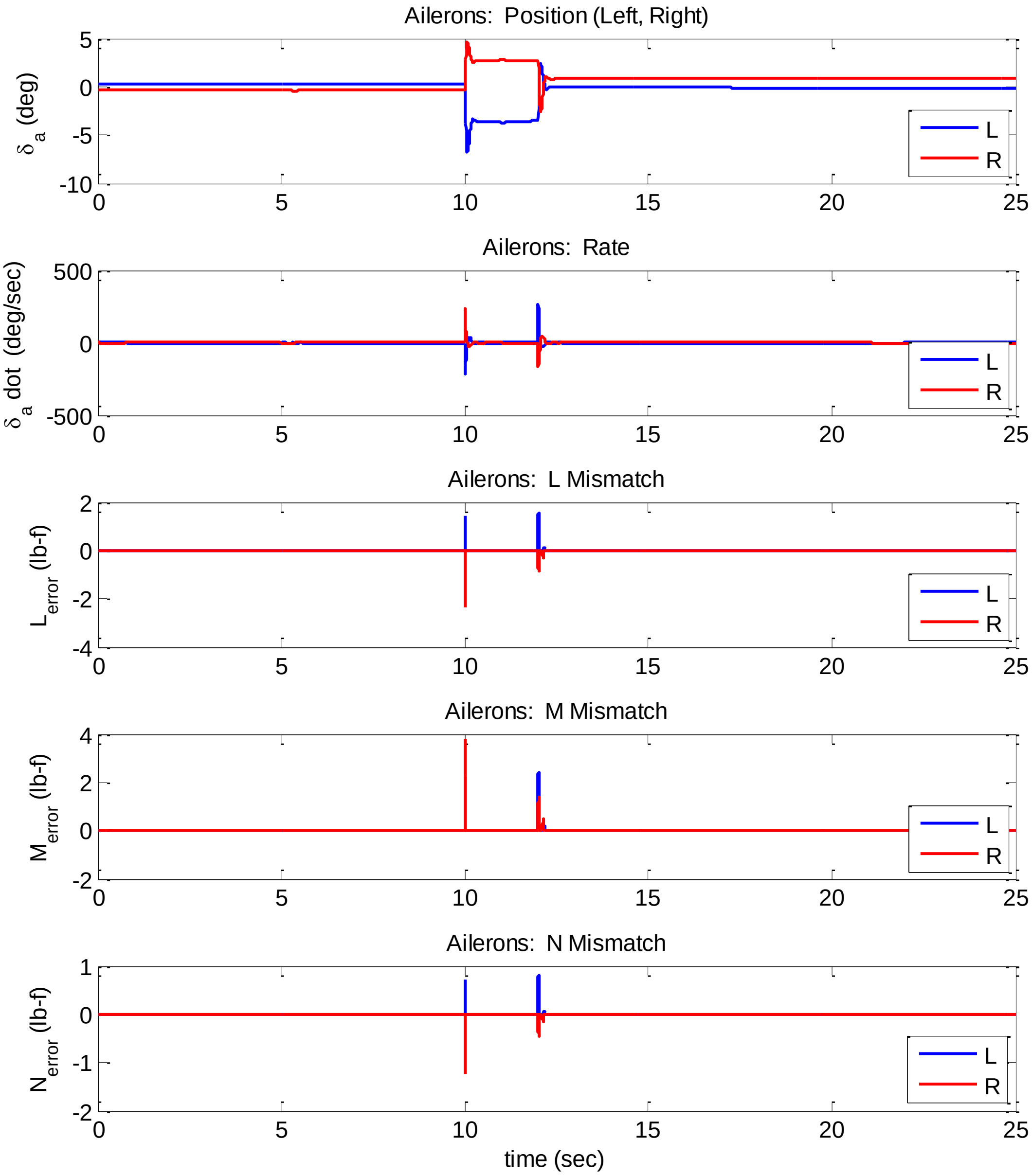


Figure 7. Outer loop control: ailerons

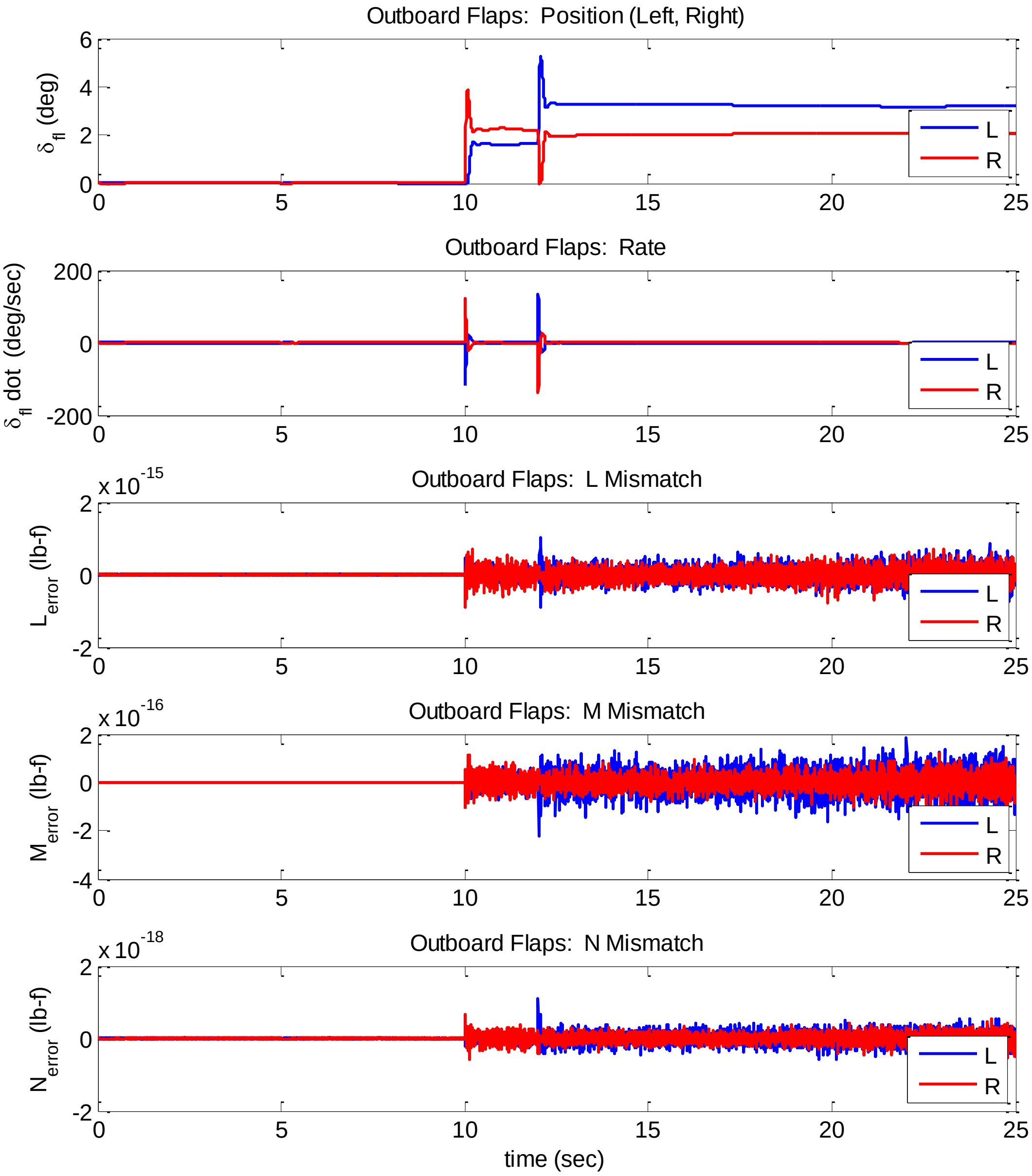


Figure 8. Outer loop control: outboard flaps

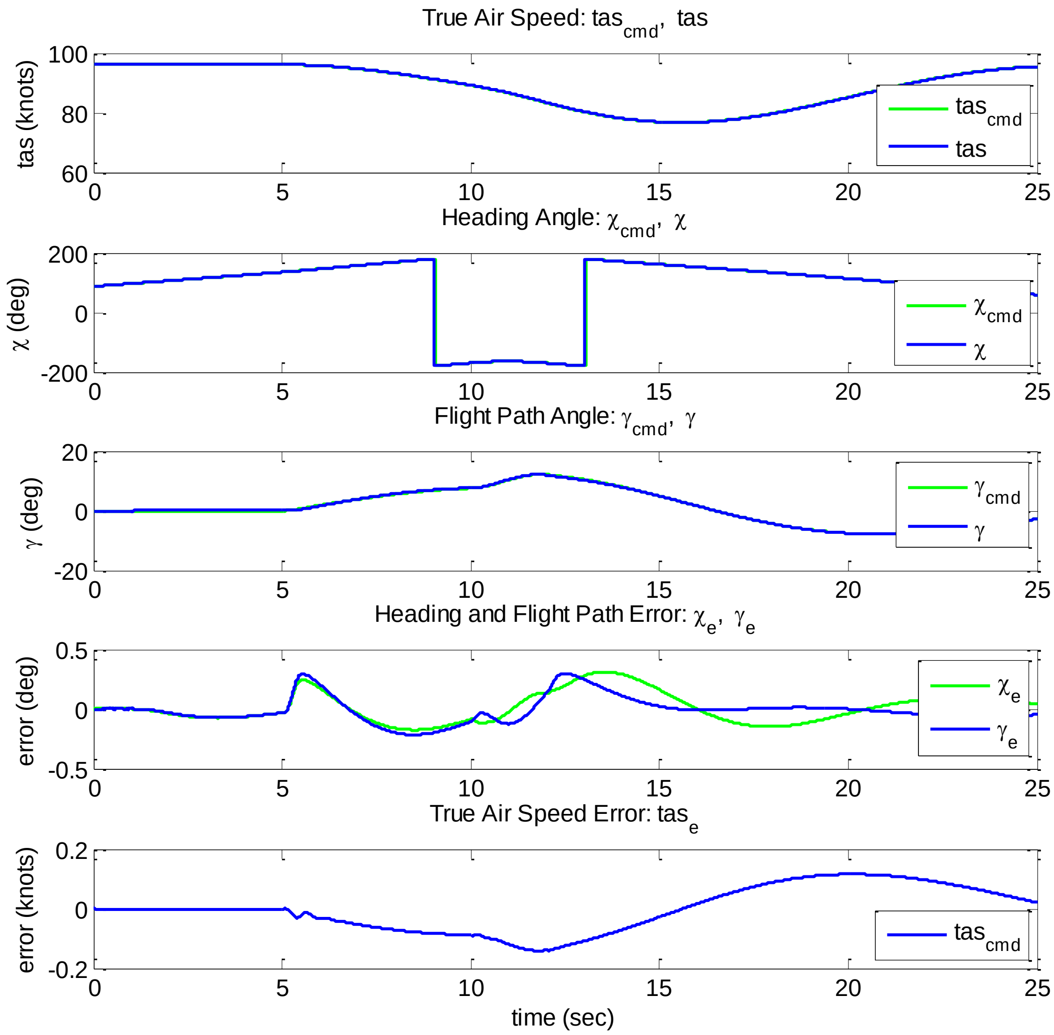


Figure 9. Guided results: velocity vector tracking

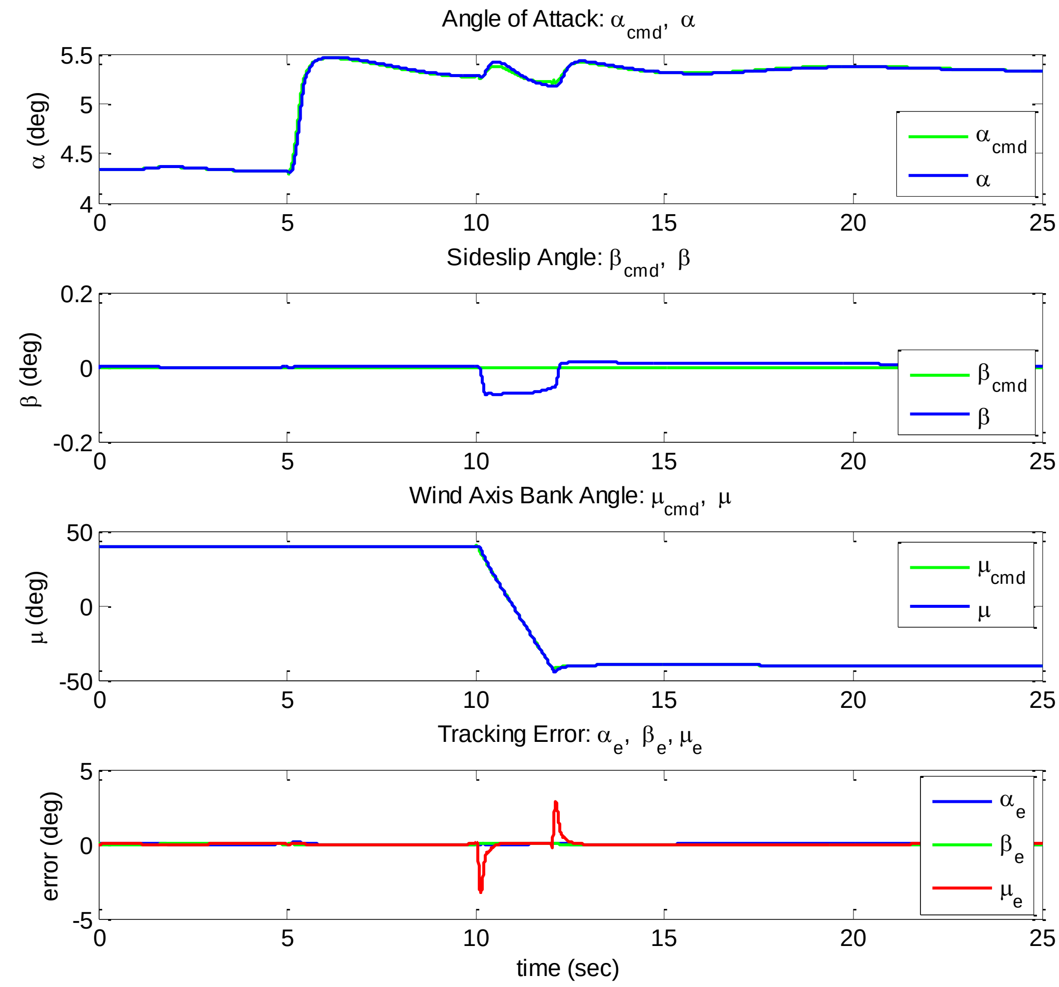


Figure 10. Guided results: angle of attack, sideslip, and wind axis bank angle

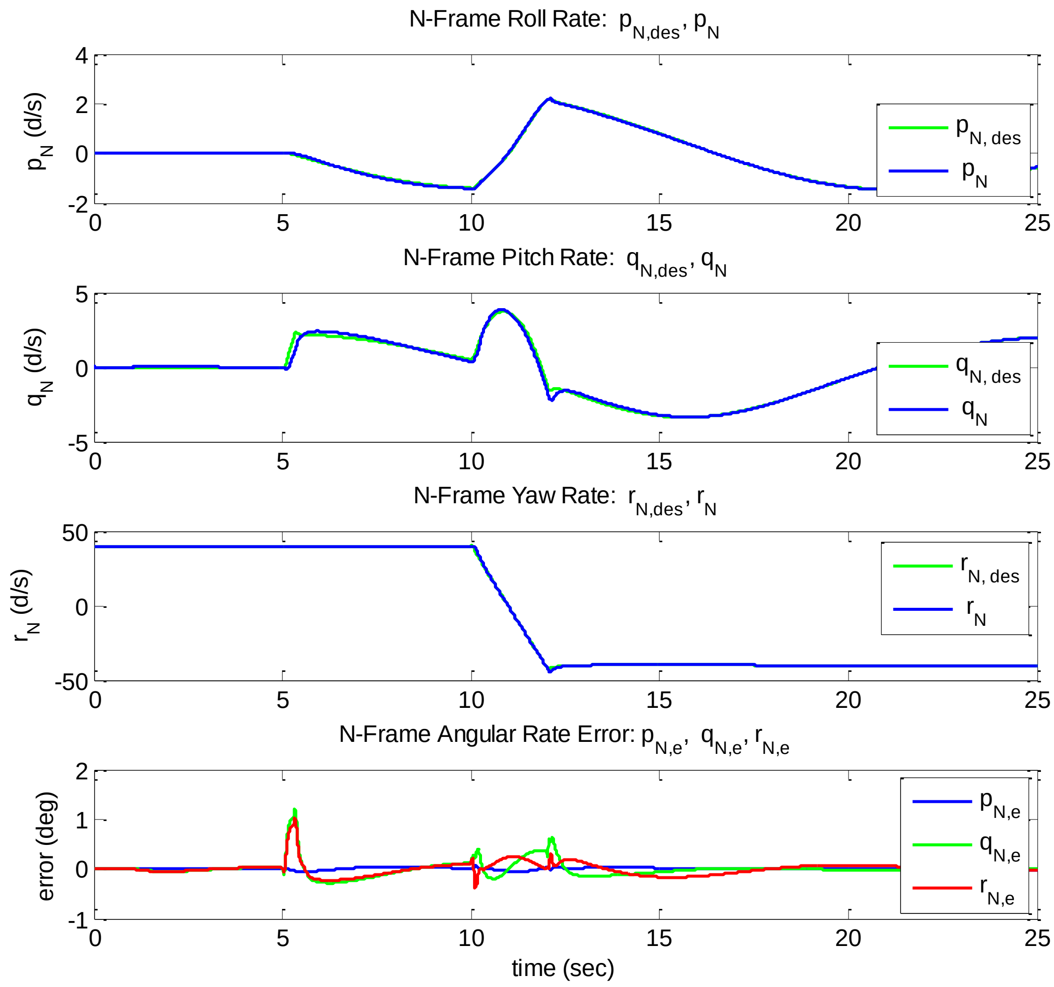


Figure 11. Guided results: angular rate of the N-frame with respect to I-frame

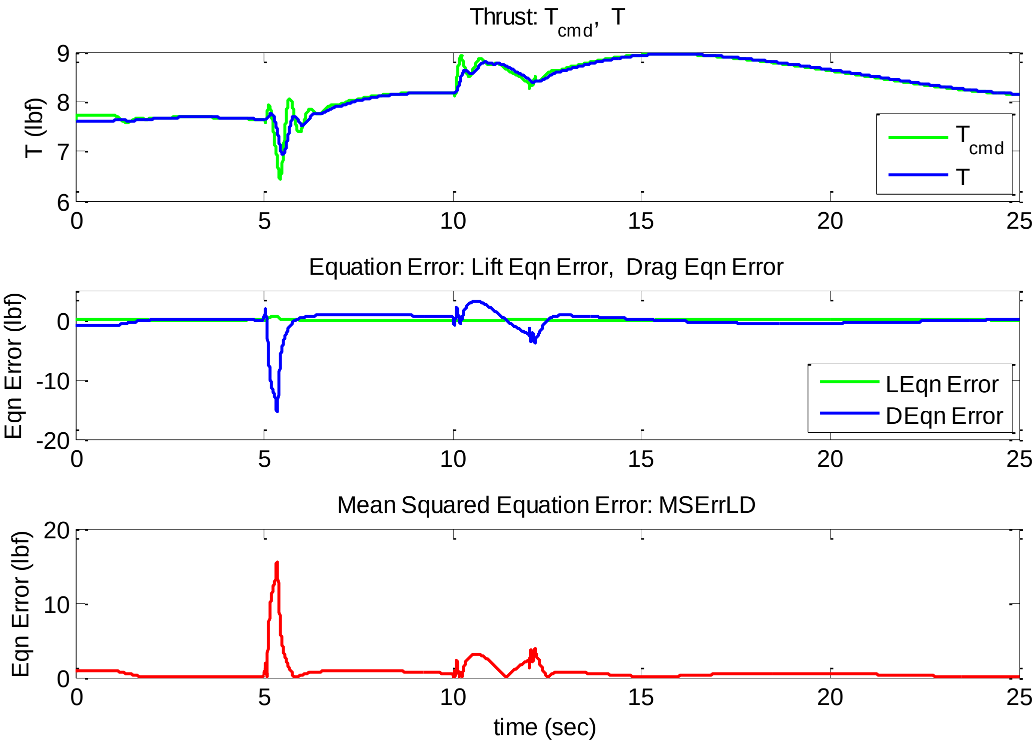


Figure 12. Guided results: commanded thrust and equation error in guidance solution

There probably is, however, a better method for satisfying the point mass equation of motion. The present guidance law tends to destabilize the system if the number of iterations between control sample time is three or greater. Another deficiency can be seen in the equation error of the drag equation in figure 12. It dominates the mean squared error of the two equations (54-55) and causes undesirable oscillatory thrusting commands during the periods of commanded angle of attack changes. The one degree change in angle of attack produced a drag equation error spike of -15 lbf at 5 seconds. Perhaps one cause could be that the dynamic derivative models, producing moments due to angular body rates, were neglected in the build up for *L* and *D*.

VIII. Concluding Remarks

This paper has described and shown the potential of a quaternion-based aircraft control architecture for following candidate trajectories in simulation. The use of kinematic inversion coupled with the quaternion PID control was shown to produce the same level of tracking in all attitude angles. With the guidance loop open, the outer loop $\alpha-\beta-\mu$ provided excellent tracking of α_c, β_c and μ_c, when the effectors did not violate their position and rate limits. The maneuver used did not demonstrate the potential of the control allocation to address the problem effectors in saturation. The control allocation also failed to demonstrate that it would retract counterbalanced controls in the example given. Instead, the allocation algorithm ended the maneuver with all surfaces deployed—perhaps indicating an error in the implementation. The quaternion-based guidance showed good tracking of the desired velocity vector. Moreover, the commanded wind axis bank angle resulting from the guidance law appeared to duplicate the unguided command. More work, however, is needed to address the equation error problem during angular body rate changes in defining acceptable thrust and angle of attack commands. While the example demonstrated the potential of the proposed architecture to follow a commanded trajectory, it remains to be seen whether this architecture can be used define acceptable candidate trajectories based on the control power limitations of the vehicle.

IX. References

[1]Smith, P.R.: Functional Control Law Design Using Exact Non-Linear Dynamic Inversion. *AIAA Atmospheric Flight Mechanics Conference,* AIAA-94-3516, pp. 481-489, August 1994.

[2]Bacon, B., and Ostroff, A.: Reconfigurable Flight Control Using a Special Accelerometer Implementation. *Proc*. AIAA *Guidance, Navigation, and Control Conference*, Denver, CO, AIAA 2000-4565, Aug. 2000.

[3]Ostroff, A., and Bacon, B.: Enhanced NDI Strategies for Reconfigurable Flight Control. *Proc. 2002 American Control Conference*, Anchorage, AK, FAO7-4, May 2002.

[4]Snell, S.A.: Nonlinear Dynamic-Inversion Flight Control of Supermaneuverable Aircraft. Doctoral Dissertation, University of Minnesota, October 1991.

[5]Stevens, B.L., and Lewis, F.L.: Aircraft Control and Simulation, 2nd Edition. Wiley, 2003.

[6]Bryson, A.E., and Ho, Y.C.: Applied Optimal Control. Hemisphere Publishing, 1975.

[7]Conte, S.D., and de Boor,C.: Elementary Numerical Analysis. McGraw-Hill, 1972.

[8]Cunningham, K., Foster, J.V., Morelli, E.A., and Murch A.M.: Practical Application of a Subscale Transport Aircraft for Flight Research in Control Upset and Failure Conditions. AIAA Atmospheric Flight Mechanics Conference, AIAA 2008-6200, Honolulu, HI, 2008.