NASA
Technical
Paper
3695

October 1997

Dynamic Forms Part II: Application to Aircraft Guidance

George Meyer and G. Allan Smith

National Aeronautics and
Space Administration

NASA Technical Paper 3695

1997

Dynamic Forms Part II: Application to Aircraft Guidance

George Meyer and G. Allan Smith, *Ames Research Center, Moffett Field, California*

National Aeronautics and Space Administration

Ames Research Center
Moffett Field, California 94035-1000

CONTENTS

NOMENCLATURE v

SUMMARY 1

1 INTRODUCTION 2

2 AIRCRAFT MODEL 9

2.1 Rigid Body 9

2.2 Force and Moment Generators 10

2.3 Simulation Model 13

3 OUTPUT TRAJECTORY COMMAND GENERATOR 16

3.1 Aircraft Maneuvers 16

3.2 Polynomial Segments 21

3.3 Control Cards 25

3.4 Coordinate Patching 29

3.5 Transition Dynamics 34

3.6 Output Mode 36

4 DYNAMIC INVERSE 44

4.1 Dynamic Inverse of Pure Feedback Systems 48

4.2 Aircraft Model for Guidance Purposes 53

4.3 Intermediate Axis System 54

4.4 Control Modes 57

4.5 Guidance Servo 66

5 EXAMPLES 68

5.1 Maneuver: Move Forward 300 Feet 68

5.2 Maneuver: Frisbee 70

6 CONCLUSION 74

REFERENCES 75

NOMENCLATURE

Acronyms

ATC	air traffic control
CCS	control card sequence
CTOL	conventional takeoff and landing
DCC	direction cosine attitude control mode
DFC	direct force control mode
EAC	Euler angle control mode
HAT	horizontal attitude coordinatization mode, $E_1E_2E_3$
RCS	reaction control system
RPA	runway path axes coordinatization mode
SCS	status card sequence
STOL	short takeoff and landing
VAT	vertical attitude control mode, $E_3E_2E_3$
VATOL	vertical attitude takeoff and landing
VMS	vehicle management system
V/STOL	vertical short takeoff and landing
VTOL	vertical takeoff and landing
WCS	waypoint card sequence
WPA	wind path axes coordinatization mode

Symbols

a_r	runway coordinates of acceleration (ft/sec^2)
$\overleftrightarrow{b}$	body-fixed axis system (right-handed orthonormal)
c_k	k^{th} control point
C_{br}	direction cosine matrix relating $\overleftrightarrow{b}$ to $\overleftrightarrow{r}$ (e.g., $x_b = C_{br}x_r$)
$E_i(\theta)$	elementary Euler rotation about axis i through angle θ
$\vec{F}$	force
$\vec{f}$	specific force ($\vec{F}/m$)
F_b	force and moment generating algorithm, $F_b = (f_b^f, f_b^m)$
	also, the expanded algorithm, $\left(F_b^{mb}, F_b^{mr}, F_b^{fb}, F_b^{ar}, F_b^{vb}\right)$
F_t	$= \left(F_t^{mb}, F_t^{m\alpha}, F_t^{mr}, F_t^{ft}, F_t^{fb}, F_t^{ar}, F_t^{vb}\right)$
$\hat{F}_t$	algorithm for the multimode static inverse of F_t
$F_t^\dagger$	generalized $\hat{F}_t$ to include a switch (ϵ) to suppress zero dynamics
f_b	body coordinates of specific force
f_b^f	specific force generating function in body coordinates: $f_b = f_b^f(z_b)$
f_b^m	specific moment generating function in body coordinates: $m_b = f_b^m(z_b)$

f_r	runway coordinates of specific force
f_t	t-coordinates of specific force, $C_{tr} f_r$
f_t^e	specific force error of the pure feedback approximation
f_μ	state flow for mode μ (e.g., $\dot{x} = f_\mu$)
g	acceleration of gravity
h	altitude, $h = -r_{r3}$
h_μ	state-to-output function in mode μ [e.g., $y = h_\mu(x)$]
I_k	time interval of segment k
J_b	moment of inertia matrix
M_1	macro maneuver: step forward a given distance in a given time
M_{ij}	maxima of a unit step maneuver in i^{th} derivative
m	mass
m_b	body coordinates of the specific moment, $m_b = n_b + J_b^{-1} S(\omega_{brb}) J_b \omega_{brb}$
$\mathcal{M}_\mu$	state space model for mode μ
n_b	body coordinates of the specific torque, $n_b = J_b^{-1} N_b$
P_μ	parameter space for mode μ
$P[t, T, x(0), x(T)]$	polynomial segment with given boundary conditions
p	parameter $p = (w_r, W, \theta_T, \ldots)$
R^n	set of all real n-tuples
$\overleftrightarrow{r}$	runway-fixed axis system
$\vec{r}_i$	i^{th} basis vector of $\overleftrightarrow{r}$
r_r	runway coordinates of position vector $\vec{r}$
$S(x)$	matrix of vector cross product operator
T	algorithm for the exact dynamic inverse
T_k	duration of the kth segment
T^0	algorithm for the pure feedback dynamic inverse
$\hat{T}$	algorithm for the inverse of T
$\hat{T}^0$	algorithm for the inverse of T^0
t	time in seconds
U_μ	active control space for mode μ
u	control $u = (u_m, u_p, u_f)$
u^g	guidance control
u^p	plant control
u_f	configuration control (flap, aileron droop, speed brake)
u_m	moment control (roll, pitch, yaw)
u_p	power control (throttle, side force, nozzle)
$\hat{u}^p$	input to plant control servos
$\vec{v}^a$	relative air velocity vector
v_r	runway coordinates of velocity vector $\vec{v}$
W	weight of aircraft
$\vec{w}$	wind vector
w_r	runway coordinates of wind
X_μ	state space of aircraft for mode μ
x	state

x^e	state error
x^g	guidance state
x^p	plant state
$\hat{x}^p$	estimated plant state
x_ω	rigid-body state $x_\omega = (r_r, v_r, C_{br}, \omega_{brb})$
Y_μ	output space for mode μ
y	output
y_b	output of the force and moment algorithm, $y_b = (f_b, m_b)$
y^g	desired output
z_b	input to the force and moment algorithm
$()^T$	transpose
$()^{-1}$	inverse
$()^{(k)}$	k^{th} time derivative
$\vec{()}$	vector
δ_i	three-dimensional column of zeros except row i, which is 1
ϵ	switch for the suppression zero dynamics
ε	trim error threshold
θ	pitch angle
θ_T	air temperature
μ	operating mode, $\mu = (\mu_\alpha, \mu_y, \mu_u)$
μ_α	coordinatization mode
μ_c	control mode
μ_y	output mode
σ_j^k	elementary step maneuver in j^{th} derivative
ω_{brb}	angular velocity, so that $\dot{C}_{br} = S(\omega_{brb})C_{br}$

Functions of Dynamic Forms

AF_q^k	Euler angle form to order k for sequence q
$f[SF^m(x)]$	scalar form of $f(x)$, namely $\left[f(x), \dot{f}(x), \ldots, f^{(m)}(x)\right]$
$MF^k(M)$	matrix form, $\left[M, \dot{M}, \ldots, M^{(k)}\right]$
$MF \succ RF$	transformation taking matrix forms to rotational forms
$MF \prec RF$	transformation taking rotational forms to matrix forms
$MF \succ AF_q$	transformation taking matrix forms to Euler angle forms in sequence q
$MF \prec AF_q$	transformation taking Euler angle forms in sequence q to matrix forms
$MF \succ PF$	transformation taking matrix forms to Euler parameter forms
$MF \prec PF$	transformation taking Euler parameter forms to matrix forms
RF_{br}^k	rotation form, $\left[C_{br}, \omega_{brb}, \ldots, \omega_{brb}^{(k-1)}\right]$
$SF^k(x)$	scalar form to order k of $x \in R$, $= \left[x, \dot{x}, \ldots, x^{(k)}\right]$
$SF^k(x) \star SF^k(y)$	scalar form of xy, namely, $\left[xy, (xy)^{(1)}, \ldots, (xy)^{(k)}\right]$
$VF^k(x_b)$	vector form to order k of b-coordinates of $\vec{x}$

Dynamic Forms Part II: Application to Aircraft Guidance

George Meyer and G. Allan Smith
Ames Research Center

SUMMARY

The paper describes a method for guiding a dynamic system through a given set of points. The paradigm is a fully automatic aircraft subject to air traffic control (ATC). The ATC provides a sequence of waypoints through which the aircraft trajectory must pass. The waypoints typically specify time, position, and velocity. The guidance problem is to synthesize a system state trajectory that satisfies both the ATC and aircraft constraints. Complications arise because the controlled process is multidimensional, multiaxis, nonlinear, highly coupled, and the state space is not flat. In addition, there is a multitude of operating modes, which may number in the hundreds. Each such mode defines a distinct state space model of the process by specifying the state space coordinatization, the partition of the controls into active controls and configuration controls, and the output map. Furthermore, mode transitions are required to be smooth.

The proposed guidance algorithm is based on the inversion of the pure feedback approximation, followed by correction for the effects of zero dynamics. The paper describes the structure and major modules of the algorithm, and the performance is illustrated by several example aircraft maneuvers.

1 INTRODUCTION

Development of a methodology for the design and verification of aviation operations systems is a NASA goal. (See ref. 1.) For the purposes of the present report, several interacting components of such a system may be identified as shown in figure 1. They include air traffic control (ATC) and a varying number of flights, each consisting of a pilot crew, the aircraft, and the flight vehicle management system (VMS). Finally, the ever-present environmental conditions, including weather and hardware failures, affect all the other components.

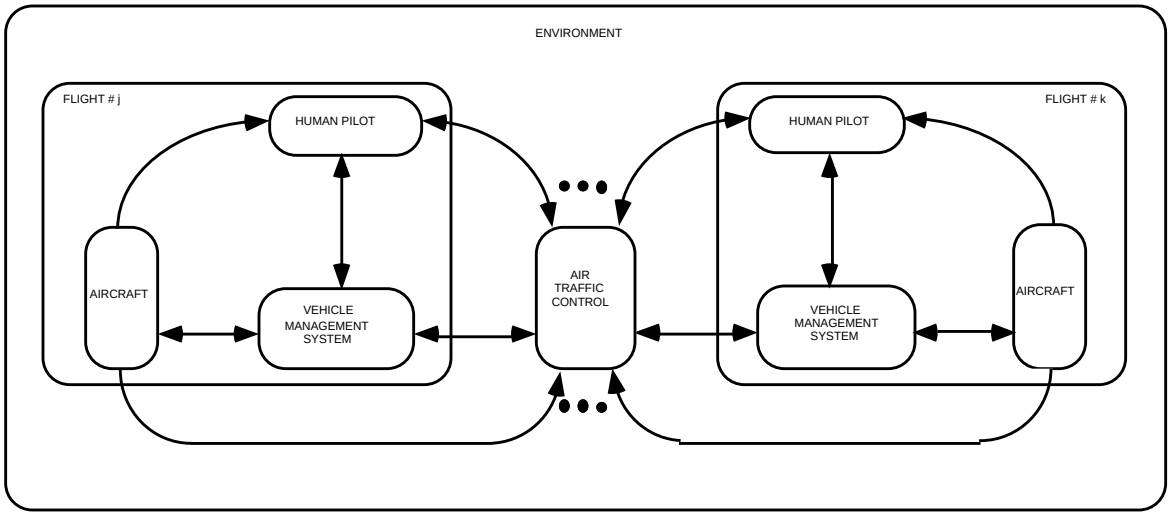

Figure 1. Major components of an aviation operations system.

Consider the vehicle management system; its purpose is to provide interfaces between the human pilot, the ATC, and the aircraft. The major subsystems of VMS and their interactions are shown in figure 2. (See ref. 2.) The human pilot interacts with the VMS at three levels. At the highest level, which is knowledge-based (termed strategic in the upper left-hand corner of the diagram), the pilot in command generates the flight plan and proposes it to ATC, accounting for weather, route, fuel reserves, and other parameters. ATC then modifies this proposed flight plan as necessary to provide separation from other traffic, and issues a clearance to the flight. On board the aircraft, the clearance is checked for legality and feasibility, and either accepted or renegotiated with ATC, as necessary. The clearance is then translated into an executable reference flightpath, and appropriate modes are selected to enable the aircraft to fly the accepted clearance.

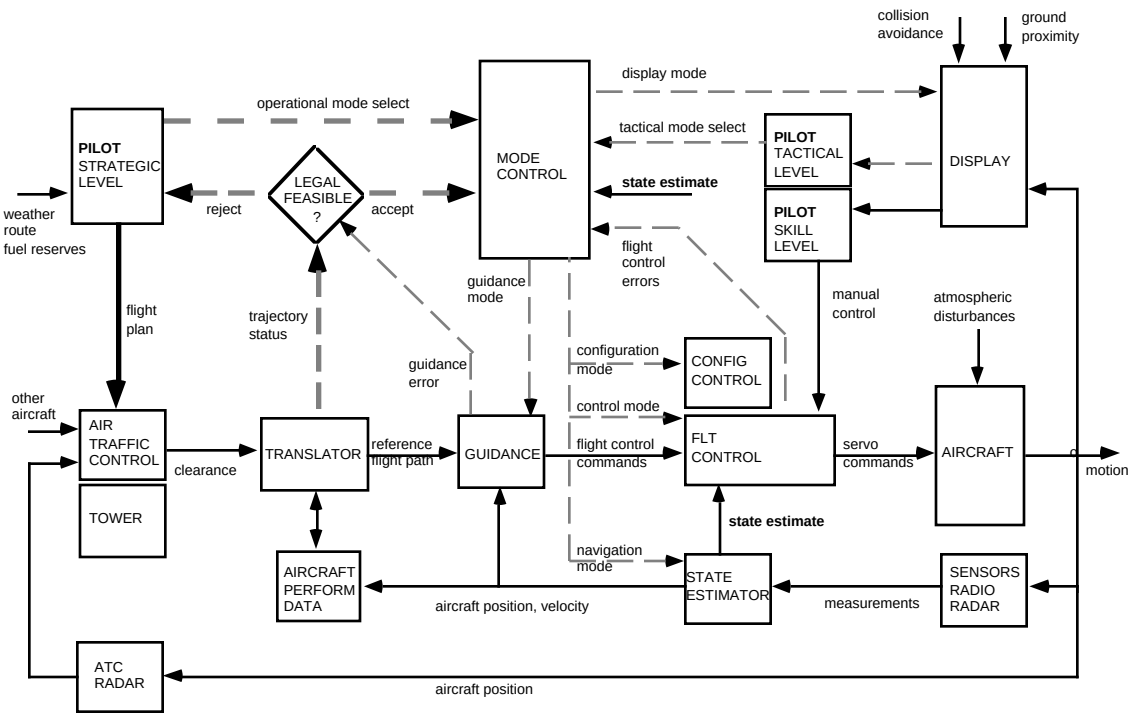

Figure 2. Major components of a VMS.

In addition to operational modes such as takeoff, climb, cruise, approach, and landing that are selected by the pilot at the strategic level, lower-level modes are selected by the pilot at the tactical level. These tactical modes include guidance, configuration, flight control, navigation (state estimation), and display modes.

At the skill level, the pilot is manually controlling the aircraft with little assistance from the automatics.

It may be noted that there is a mixture of both computation and message types of variables. Some computations, particularly at the high levels, are discrete or sentential in that they are based on a formal logic and they manipulate sentences in that logic, while other computations are arithmetic or numeric in the sense that they employ mathematical methods that manipulate real variables. Similarly, the communication between subsystems may be by means of both sentences and numbers. In the diagram, dashed lines represent flows of sentential or boolean variables; solid lines represent flows of numbers. For the present discussion, continuous variables and sampled data need not be distinguished between, but both can be referred to as analog variables. Systems with mixed variables, both discrete and analog, are referred to as *hybrid*. It is apparent that a VMS is a hybrid system.

In the diagram, the aircraft represents the airframe, including engines, servo actuators for all controls, and all sensors. In this view, the aircraft is the main source of analog variables. It is modeled mathematically with differential equations, which are multidimensional, multiaxis, highly coupled, nonlinear, and time varying.

The interaction between the sentential and analog components of a VMS is further illustrated by the diagram in figure 3. In this model, the VMS communicates with ATC by means of sentences and with the controlled aircraft by means of analog signals. The virtue of the diagram is that it shows that there must be an interface, called herein a dynamics translator, which translates sentential commands into functions of time, and conversely, analog measurements into sentential descriptions of system behavior.

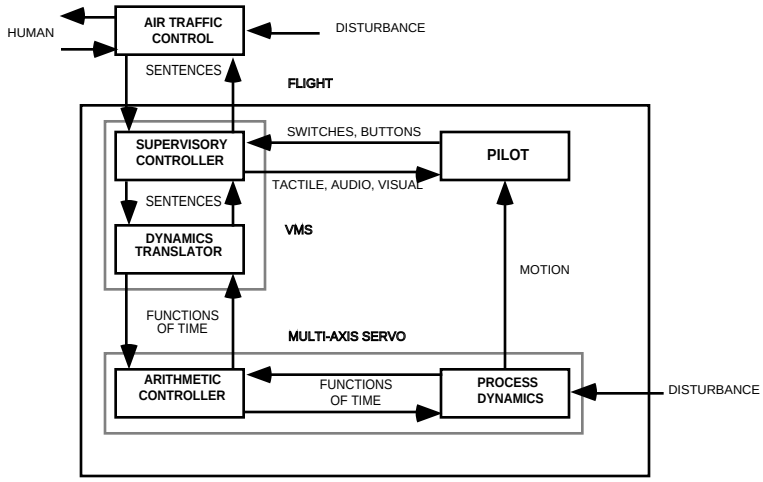

Figure 3. Interface between sentential and analog layers.

The multiaxis servo is the analog layer. The arithmetic controller closes the loops around aircraft disturbances and modeling errors, thereby making the analog layer robust, and it provides accurate tracking of input signals.

Another representation of the aviation operations system is shown in figure 4. For simplicity, only the fully automatic mode of operation, in which the flight is controlled completely by ATC and flight computers, is shown. A hierarchy of subsystems or layers is in effect, namely, (top to bottom) ATC, flight, VMS, path generator, guidance, and regulator. The hierarchy operates in terms of the horizon width, persistence, detail, and type. In planning a sequence of way points, ATC has the widest horizon, the plan changes infrequently (minutes to hours), it is very coarse, and the planning algorithm is of the discrete type. The path generator, in converting the waypoints into control points (described later in the report) has a narrower horizon than does ATC; replanning occurs more frequently (seconds to minutes); there is a moderate amount of detail in the plan; and planning employs a mix of discrete and continuous methods. The guidance, which transforms the path into the corresponding reference motion of aircraft state and control (x^g, u^g), has a narrow horizon with only several seconds of "look-ahead" capability; replanning occurs frequently (several seconds), and the planning employs mainly continuous methods. Finally, at the bottom of the hierarchy, the regulator transforms the tracking error $x^e = \hat{x}^p - x^g$, which is the difference between the estimated plant state and the guidance state, into corrective control that is combined with the guidance control u^g to produce the plant control u^p. (The negative of the standard servo error is chosen so that x^g may be treated as the origin.) The regulator has the narrowest horizon,

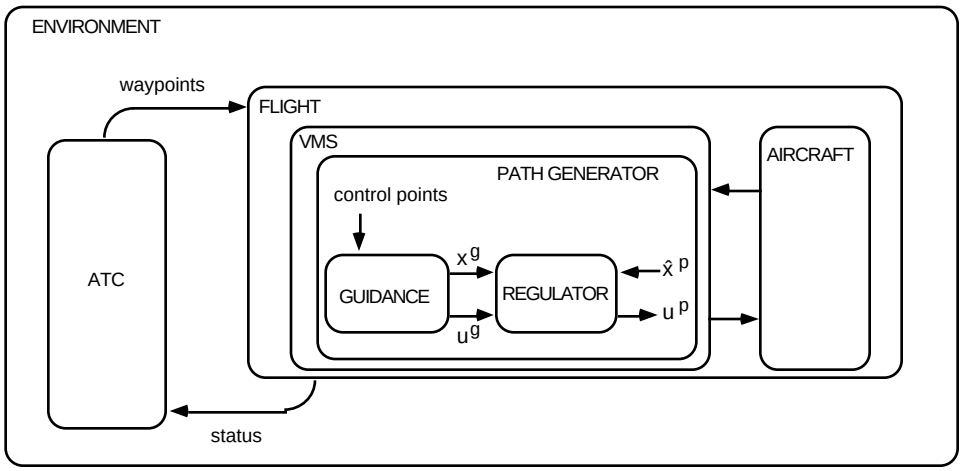

Figure 4. Multilayer model of aviation operations system.

without any look-ahead capability; the shortest persistence (milliseconds); and the greatest amount of detail. The algorithm is based on continuous (and sampled-data) methods.

The environment injects disturbances into the aviation operations system at many levels. The standard procedure for reducing the effects of these disturbances is to employ feedback. Centralized feedback in which the hierarchy is ignored is always possible, but it leads, in all but the simplest cases, to very complex and largely unsolvable performance analysis problems. It seems that a more promising approach is to structure the feedback hierarchically and to match at each level the variable type of the feedback with the algorithm type of the level. If the feedback can be layered in this way, then the complete system will consist of a hierarchy of closed-loop subsystems. Each subsystem has the servo structure in the sense that its function is to closely follow its command in spite of disturbances. In such a completely layered structure, the type of command and disturbance depends on the type of the layer. The expected analytical advantage of such a structure is that each servo has a relatively simple input/output description, and communication between servos is simpler. This is not passive modeling—it is active design to make it so. Part of the freedom available in the design process is used to structure the system; the remainder is used for control. *The present working hypothesis is that such complete layering is possible for the aviation operations system.* Consider the block diagram shown in figure 5. In the figure, the ATC waypoints are represented by a waypoint card sequence (WCS) that describes the waypoints. The discrete event control system operates on the waypoint card sequence, the pilot input, and the system behavior represented by the status card sequence (SCS), and produces the control card sequence (CCS). The control cards are the only means for controlling the rest of the system below this level. That is, at this level the CCS *is* the control variable. Furthermore, the SCS is the only means for observing the behavior of the system below this level.

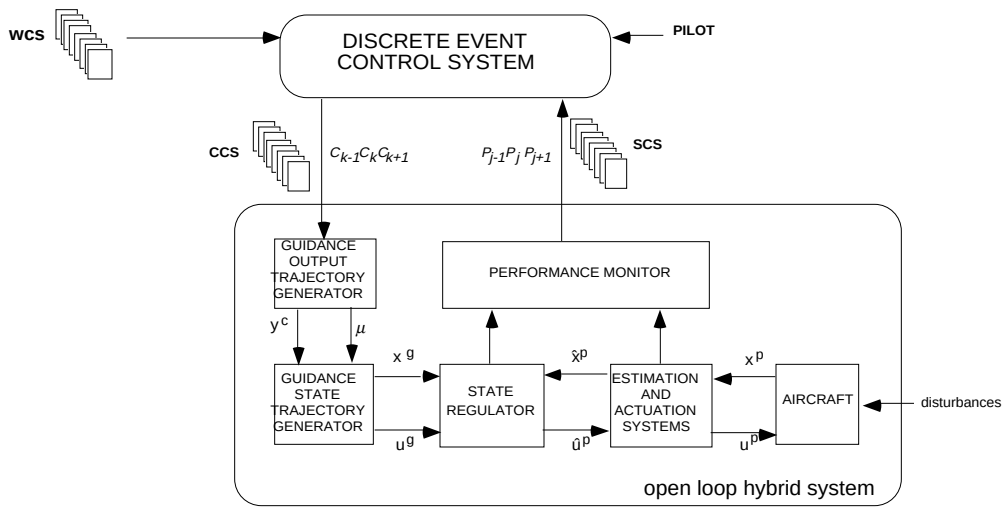

Figure 5. Closed-loop hybrid flight control model.

The guidance output trajectory generator translates the control cards into appropriate continuous functions of time that define the commanded motion of system output y^c, which may include, for example, the runway coordinates of aircraft position vector, the position of flaps, and other variables. This generator also specifies the operation mode μ, which specifies some of the details (such as control mode) of just how the maneuver given by y^c is to be executed.

The guidance state trajectory generator expands y^c into a complete state trajectory and corresponding control (x^g, u^g) that satisfies the state equation of the aircraft model and the output map. The aircraft is then regulated to that reference state. Finally, the performance monitor reports any deviations from the expected motion by means of the SCS.

This report is concerned only with the transformation of CCS into the corresponding guidance (x^g, u^g). The guidance algorithms that are developed are quite general; they apply to both civilian and military operations, including remotely piloted vehicles, and to a variety of aircraft types such as conventional takeoff and landing (CTOL), short takeoff and landing (STOL), vertical takeoff and landing (VTOL), helicopters, and tilt rotors.

The guidance problem considered in the present report may be described as follows. The control point table describes the control points. For example, figure 6 shows a 90-degree turn that is composed of several segments, which are specified by control points. The straight west-to-east line segment terminates on the control point c_k. The next segment is a variable curvature transition to a circle at c_{k+1}. The circular arc, terminating at c_{k+2}, is followed by a variable curvature transition terminating on a north/south line at c_{k+3}. The CCS provides the x,y coordinates at each control point. In addition, it may specify the duration or the arc length of the segment, as well as a set of consistent terminal conditions for some of the many additional variables such as altitude, climb rate, ground speed, airspeed, heading, thrust level, and flap deployment. The CCS also provides higher level instructions such as the type of segment (linear, circular, or helical), and it specifies the operating mode for the segment. The operating mode provides such high-level instructions as (1) which of several possible state space representations is to be used; (2) which of several possible state-to-output maps is to be used; and

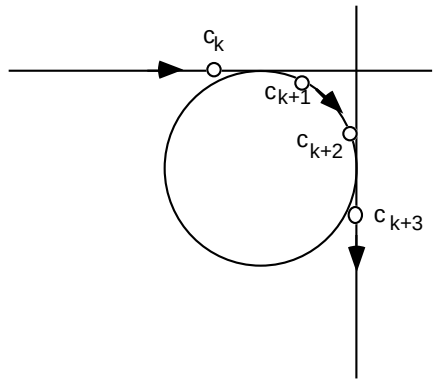

Figure 6. Typical maneuver specified by means of control points.

(3) which of several control modes is to be used. The operating mode is discussed in detail in the main body of the present report. At this point we merely note that there may easily be over a hundred possible operating modes.

Each such mode (denoted by μ) defines a different state space model $\mathcal{M}_\mu$ of the same underlying physical process. That is, each operating mode defines a particular state space (X_μ), control space (U_μ), parameter space (P_μ), output space (Y_μ), state flow (f_μ), and output map (h_μ):

$$\begin{array}{rcl} t & \in & R,\ x \in X_\mu,\ u \in U_\mu,\ p \in P_\mu,\ y \in Y_\mu \\ \dot{x} & = & f_\mu(x, u, p) \\ y & = & h_\mu(x) \end{array} \tag{1}$$

The dimension of the state space X_μ remains constant; the meaning of the coordinates is mode-dependent. All controls are distinguished into two types: The controls used by the regulator are called the active controls (or controls, for short) u; the rest of the controls are called parameters p. The dimensions of the active controls and output spaces, U_μ and Y_μ, respectively, while always equal to each other, may be mode-dependent. For example, in one control mode the throttle may be an active control that is used by the control system to execute a particular maneuver; in that case the throttle is included in U_μ. On the other hand, occasions arise where the throttle is not available for active control, as when set to idle. In that case the throttle is considered to be a parameter coordinate in P_μ and not a control coordinate in U_μ. Similarly, both airspeed and climb rate may be controlled outputs, in which case both coordinates are in Y_μ. But there is a mode in which the climb rate is not controlled explicitly; then it is not one of the coordinates in Y_μ. Some variables, such as the wind, are always in P_μ.

In general, each such model $\mathcal{M}_\mu$ is multidimensional, multiaxis, cross-coupled, nonlinear, and, because of p, time-varying. The problem is to find for every segment k the evolution of state and control,

$$(x^g, u^g)(t),\ t \in I_k \tag{2}$$

such that they satisfy both the state and output equations. That is, for the duration of the segment $t \in I_k$, and for the given desired output y^g,

$$\begin{array}{rcl} \dot{x}^g(t) & = & f_\mu[x^g(t), u^g(t), p(t)] \\ y^g(t) & = & h_\mu[x^g(t)] \end{array} \tag{3}$$

Furthermore, the segments must be patched smoothly together. A solution of this class of problems is the subject of the present report.

The report consists of three major sections. First, a general model of an aircraft is developed, and the typical structure of the force and moment generators is described. The Harrier VTOL fighter aircraft is used as an example. Next, the command generator in which the aircraft maneuvers are defined is developed. The maneuvers are represented by a sequence of control points that are linked with polynomials. Several useful representations of aircraft attitude are also presented. Finally, the multimode inverse algorithm, which generates the initial state and the control time history that produce the desired maneuver, is developed. The methodology is illustrated by means of several numerical examples.

The aircraft model used in the report is quite realistic. As a result, it is relatively complex. The state space is high-dimensional and, because of rotations, not Euclidean. The model is multiaxis (up to six), cross-coupled, and nonlinear. Furthermore, there is a need to employ several representations, output maps, and control strategies. The methodology of dynamic forms developed in "Dynamic Forms Part I: Functions" (ref. 3) provides effective means for the management of this complexity. A dynamic form of order k of a variable of time is simply the variable together with its first k time derivatives. The essential aspects of the methodology are reviewed and employed throughout the present report.

2 AIRCRAFT MODEL

In this section a state equation describing the open-loop dynamic behavior of an aircraft is developed in several steps. This equation forms the basis for a variety of state space models that are developed and used in the remainder of the report.

2.1 Rigid Body

The core of the model is a rigid body moving in three-dimensional, inertial space, subject to the gravitational, aerodynamic, and propulsion forces. In this report the dynamic effects due to Earth rotation and nonflatness are ignored. This approximation causes less than $0.004\ g$ acceleration error for speeds up to 600 mph, and such errors are consistent with the assumed modeling accuracy of the aerodynamic and propulsion forces. Similarly, the aircraft mass, location of the center of mass, and the moments of inertia are assumed to be quasistatic in the sense that their time derivatives do not appear in the model. Furthermore, the effects of rotating parts of the aircraft are ignored. All such translational and angular acceleration errors that are of the order of $0.004\ g$ and 0.002 radians/sec^2, respectively, are ignored since their effects can be easily controlled by means of a realistic regulator.

Let $\overleftrightarrow{r}$ be a local (runway) right-handed, orthonormal axis system in which $\vec{r}_3$ points in the direction of the local vertical (down), $\vec{r}_1$ points in the direction of the runway, and $\vec{r}_2 = \vec{r}_3 \times \vec{r}_1$. The $\overleftrightarrow{r}$ system is attached to the runway. The runway coordinates of vectors are denoted by the subscript r.

Several variables are fundamental to the following discussion. The position, velocity, and acceleration of the aircraft center of mass relative to the runway are denoted by $\vec{r}$, $\vec{v}$, and $\vec{a}$, respectively, the average wind velocity at $\vec{r}$ by $\vec{w}$, and the relative air velocity by $\vec{v}^a = \vec{v} - \vec{w}$. The aircraft mass is denoted by m, the total nongravitational force acting on the aircraft by $\vec{F}$, the specific force by $\vec{f} = \vec{F}/m$, and the gravitational force by $mg\vec{r}_3$ with scalar, constant g. Since $\overleftrightarrow{r}$ is assumed to be inertial, the standard form of the translational equations of motion follows:

$$\begin{aligned} \dot{r}_r &= v_r \\ \dot{v}_r &= a_r = f_r + g\delta_3 \end{aligned} \tag{4}$$

where δ_i for $i = 1, 2, 3$ is a column matrix with zero entries except in row i, where the entry is 1.

Another frequently used axis system is attached to the aircraft. In this (body) system, $\overleftrightarrow{b}$, $\vec{b}_1$ points tail to nose, $\vec{b}_3$ is in the aircraft plane of symmetry and perpendicular to $\vec{b}_1$, pointing down, and $\vec{b}_2 = \vec{b}_3 \times \vec{b}_1$ points along the right wing. The $\overleftrightarrow{b}$ system is attached to the aircraft center of mass.

The attitude of the aircraft relative to the runway is defined by the direction cosine matrix C_{br} so that, for example, the body coordinates of the aircraft velocity $\vec{v}$ are given by

$$v_b = C_{br} v_r \tag{5}$$

The inverse of the direction cosine matrix equals its transpose: $C_{br}^{-1} = C_{br}^T = C_{rb}$. The body coordinates of the angular velocity of $\overleftrightarrow{b}$ relative to $\overleftrightarrow{r}$ are denoted by ω_{brb}, so that

$$\dot{C}_{br} = S(\omega_{brb}) C_{br} \tag{6}$$

where, for any three-dimensional column matrix x,

$$S(x) = \begin{pmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{pmatrix} \tag{7}$$

The function S represents the vector cross product. For example, $\vec{x} = \vec{y} \times \vec{z}$ is represented in $\overleftrightarrow{r}$ as $x_r = -S(y_r)z_r$.

The angular momentum and the total torque about the aircraft center of mass are denoted by $\vec{L}$ and $\vec{N}$, respectively. Since $\overleftrightarrow{r}$ is assumed to be inertial,

$$\dot{L}_r = N_r \tag{8}$$

where the overdot denotes the time derivative of the (scalar) elements of L_r. But $L_b = C_{br}L_r$, so that after applying equation (6),

$$\dot{L}_b = N_b + S(\omega_{brb})L_b \tag{9}$$

The body coordinates of the moment-of-inertia operator about the center of mass are denoted by J_b so that, ignoring rotating components,

$$L_b = J_b\omega_{brb} \tag{10}$$

and, since the time derivatives of J_b are ignored, the Euler equations (refs. 5 and 6) become

$$J_b\dot{\omega}_{brb} = N_b + S(\omega_{brb})J_b\omega_{brb} \tag{11}$$

or, equivalently,

$$\dot{\omega}_{brb} = n_b + J_b^{-1}S(\omega_{brb})J_b\omega_{brb} \tag{12}$$

where the specific torque $n_b = J_b^{-1}N_b$. For convenience, let

$$m_b = n_b + J_b^{-1}S(\omega_{brb})J_b\omega_{brb} \tag{13}$$

The parameter m_b is the specific moment.

In summary, the equations of motion of a rigid body in inertial space follow:

$$\begin{aligned} \dot{r}_r &= v_r \\ \dot{v}_r &= C_{rb}f_b + g\delta_3 \\ \dot{C}_{br} &= S(\omega_{brb})C_{br} \\ \dot{\omega}_{brb} &= m_b \end{aligned} \tag{14}$$

Next a model of the specific force f_b and moment m_b is developed.

2.2 Force and Moment Generators

In this section the structure of the function relating the aerodynamic, propulsion, and reaction control forces and moments to other process variables is developed. The special case of the Harrier aircraft is considered, but the formalism applies to other aircraft types.

The YAV-8B Harrier (ref. 4) is a powered-lift vertical/short takeoff and landing (V/STOL) transonic, light, attack aircraft. The engine is a single turbofan, whose thrust may be diverted by means of a nozzle system. The directed thrust supports the aircraft in low-speed flight. Aerodynamic lift supports the aircraft in high-speed, conventional flight. The general features are shown in figure 7. The engine thrust vector is controllable in the longitudinal plane by means of the throttle, which controls the magnitude, and the nozzle, which controls the direction. The control of aircraft rotation depends on the flight condition. At high speed, the torque vector is generated aerodynamically by means of the conventional ailerons, elevator, and rudder. At low speed, when the aerodynamics is ineffective, the torque vector is generated by means of a reaction control system (RCS). In the RCS, a portion of the engine gas is ducted to the extremities of the aircraft, where the directed flow produces reaction forces whose magnitude is controlled by means of valves. The valves are connected to the aircraft roll, pitch, and yaw controls. In the transition regime, there is a smooth blend of aerodynamics, powered lift, and RCS. Note that, since a request for moment may produce unwanted forces as a side effect, there may be zero dynamics.

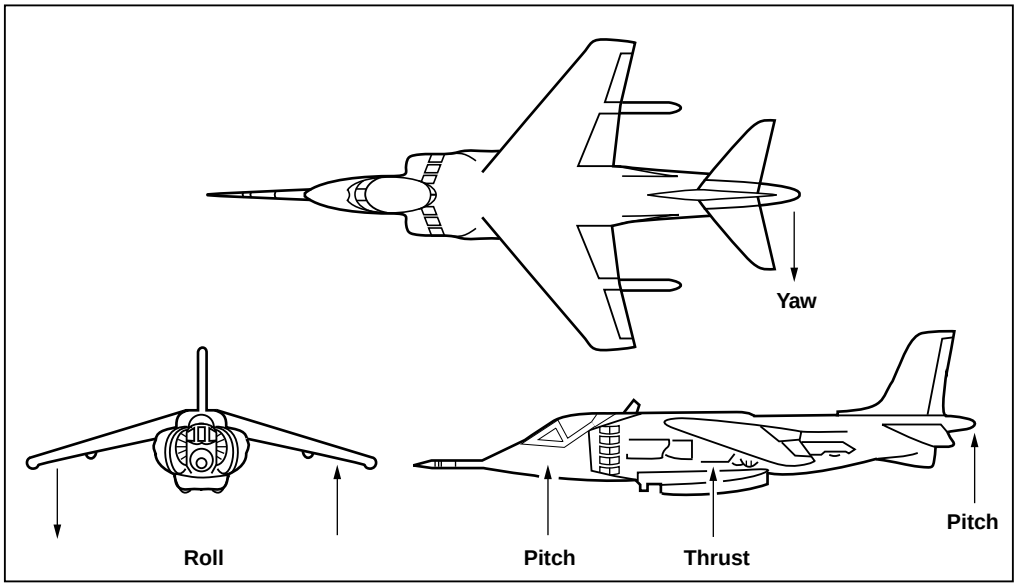

Figure 7. YAV-8B Harrier aircraft.

The algorithm $YAV8B$ adapted from reference 4 for computing the total force and moment is shown in figure 8. At the top level, the algorithm implements the function

$$YAV8B: \; R^{19} \rightarrow R^3 \times R^3 \tag{15}$$

The input consists of moment controls $u_m \in R^3$, power controls $u_p \in R^3$, configuration controls $u_f \in R^3$, air velocity $v_b^a \in R^3$, angular velocity $\omega_{brb} \in R^3$, pitch angle θ, altitude h, air temperature θ_T, and aircraft weight W. The output consists of the specific force $f_b \in R^3$ and the specific moment $m_b \in R^3$. In this case of the Harrier, the power control u_p consists of the throttle, empty slot, and nozzle. In general, the middle term could be a side-force controller. The configuration control u_f includes the flaps, aileron droop, and the speed brake.

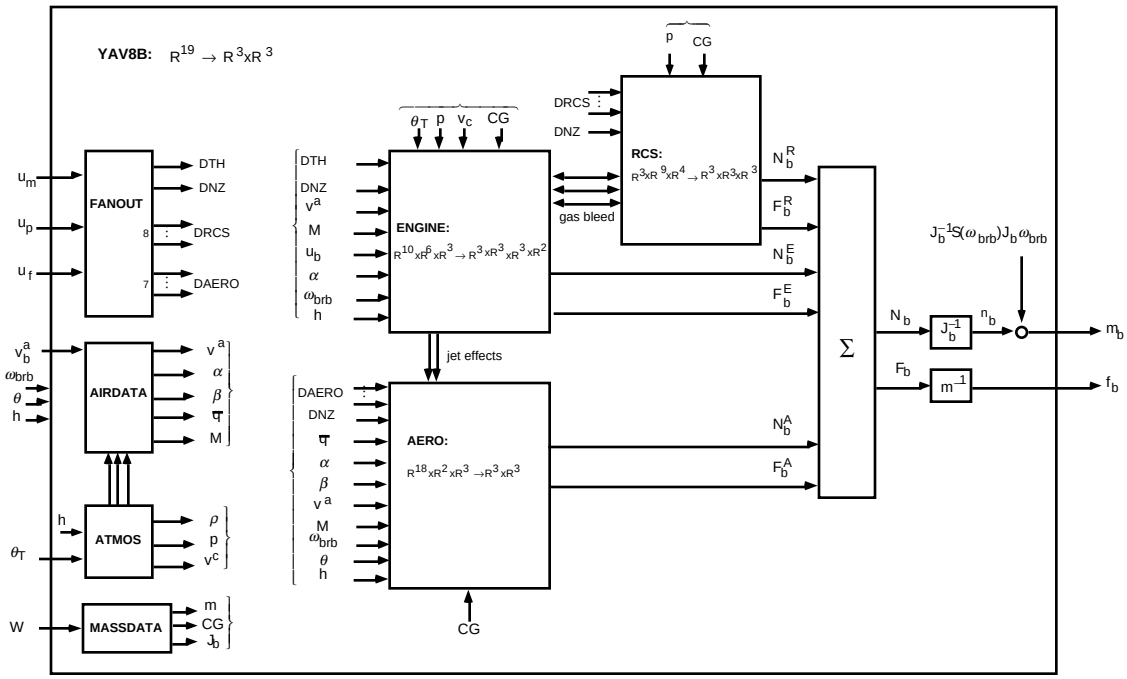

Figure 8. Force and moment generation algorithm for the YAV-8B Harrier aircraft.

At the next level down are seven major subalgorithms. Secondary variables are computed first from the input. The aircraft mass, the location of the aircraft center of mass, and the moment of inertia matrix are computed in $MASSDATA$. The air density, pressure, and speed of sound are the outputs of $ATMOS$, which models the atmosphere. The airspeed, angles of attack and sideslip, dynamic pressure, and Mach number are provided by $AIRDATA$. The preliminary computations are completed in the $FANOUT$ algorithm, in which the controls (u_m, u_p, u_f) are interpreted in terms of the engine throttle, the nozzle angle, the eight reaction control valve openings, and the seven aerodynamic control surfaces.

The details of the actual physics of the process are represented by the three large algorithms $AERO$, $ENGINE$, and RCS, which compute the aerodynamic, engine, and reaction control system torque and force. In the case of $YAV8B$, there is a one-way interaction between engine and aerodynamics through the jet effects and a two-way interaction between the engine and the RCS due to bleed gas ejected through the reaction control valves. The pitch angle θ and altitude h are used in $AERO$ to compute the ground effect. Other aircraft types, such as other VTOL types, powered-lift STOLs, helicopters, tilt rotors, and CTOL aircraft have similar algorithms describing the corresponding force and moment generators. The action of the algorithm is denoted by

$$y_b = F(z_b) \tag{16}$$

where the input and output are given by, respectively,

$$\begin{aligned} z_b &= (\, u_m, u_p, u_f, v_b^a, \omega_{brb}, \theta, h, W, \theta_T \,) \in (R^3)^5 \times (R)^4 \\ y_b &= (\, f_b, m_b \,) \in (R^3)^2 \end{aligned} \tag{17}$$

Henceforth it is assumed that the algorithm is given for the particular aircraft being considered.

In general, the body coordinates v_b^a of the air velocity, which is an entry in the input z_b, may be computed as follows

$$v_b^a = C_{br}(v_r - w_r) \tag{18}$$

where $w_r \in R^3$ represents the runway coordinates of wind. Another entry, the altitude h, is the negative of the third coordinate of position:

$$h = -r_{r3} \tag{19}$$

and the pitch angle θ is the middle Euler angle in the following expansion of the direction cosine matrix

$$C_{br} = E_1(\phi)E_2(\theta)E_3(\psi) \tag{20}$$

Therefore, z_b may be computed from (u, x_ω, p), say

$$z_b = Z\,(\,u, x_\omega, p\,) \tag{21}$$

where u, x_ω, p are the combined control, rigid body state, and parameters, respectively, defined as follows:

$$\begin{array}{rcl} u & = & (\,u_m, u_p, u_f\,) \in U \subset (R^3)^3 \\ x_\omega & = & (\,r_r, v_r, C_{br}, \omega_{brb}\,) \in X_\omega \subset (R^3)^2 \times SO(3) \times R^3 \\ p & = & (\,w_r, W, \theta_T\,) \in P \subset R^3 \times R \times R \end{array} \tag{22}$$

The combined algorithm is denoted by $F_b = (f_b^f, f_b^m)$. Consequently, equations (14) may be written as the rigid-body state equation, as follows:

$$\begin{array}{rcl} \dot{r}_r & = & v_r \\ \dot{v}_r & = & C_{rb} f_b^f(u, x_\omega, p) + g\delta_3 \\ \dot{C}_{br} & = & S(\omega_{brb})C_{br} \\ \dot{\omega}_{brb} & = & f_b^m(u, x_\omega, p) \end{array} \tag{23}$$

with the arguments defined by equations (22).

2.3 Simulation Model

In general, the nine controls (u_m, u_p, u_f) are not directly accessible. Rather, intervening dynamics, including position and rate limits, exist between the accessible servo inputs (u_m^c, u_p^c, u_f^c) and the outputs (u_m, u_p, u_f). However, often the control servos may be adequately modeled by the usual second-order systems of the form

$$\frac{u}{u^c} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{24}$$

as shown figure 9. The servo controls (u_m^2, u_p^2, u_f^2) are connected to the inputs (u_m^c, u_p^c, u_f^c) by the usually decoupled servo control laws:

$$\begin{array}{rcl} u_m^2 & = & \sigma_m(u_m, u_m^2, u_m^c) \\ u_p^2 & = & \sigma_p(u_p, u_p^2, u_p^c) \\ u_f^2 & = & \sigma_f(u_f, u_f^2, u_f^c) \end{array} \tag{25}$$

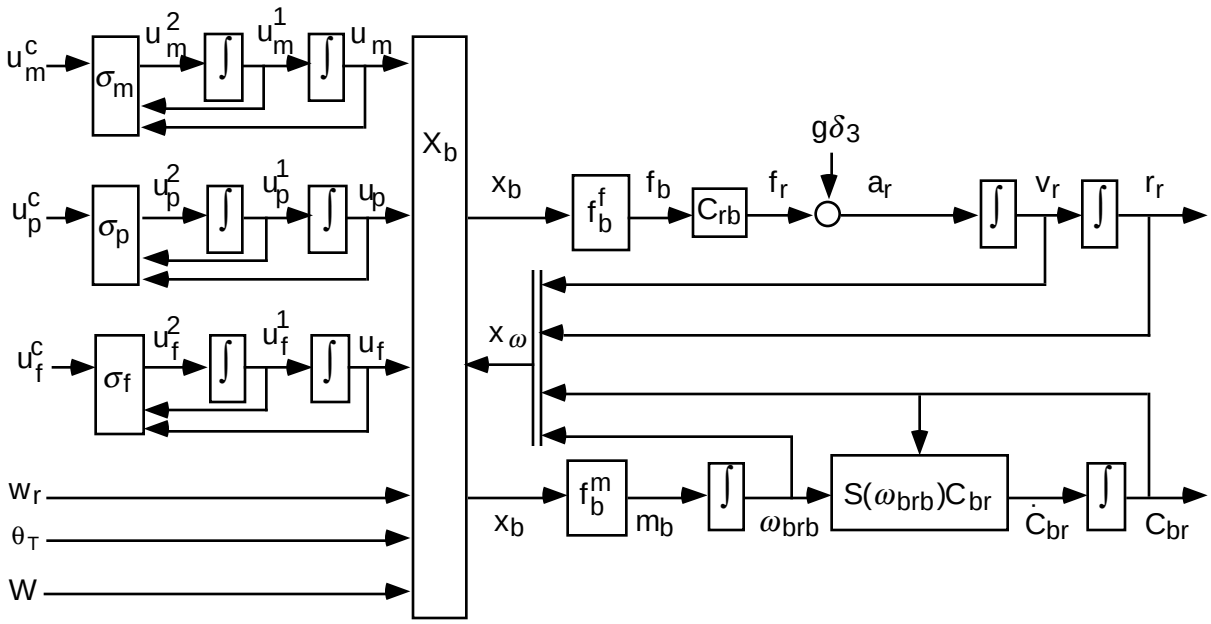

Figure 9. Block diagram of a typical simulation.

such that there is good tracking for slowly varying inputs that remain within their position limits:

$$(u_m, u_p, u_f) \approx (u_m^c, u_p^c, u_f^c) \tag{26}$$

The other inputs to the model are the parameters, namely, runway coordinates of wind w_r, aircraft weight W, ambient temperature θ_T, and the rigid-body state x_ω. The block X_b produces the variable x_b, which is the input to the force f_b^f and moment f_b^m algorithm. The body coordinates of force f_b are transformed into the runway coordinates f_r and the effects of gravity are added, resulting in the aircraft acceleration a_r, which is integrated to produce aircraft velocity v_r, and another integration produces aircraft position r_r.

In the rotational part, the body coordinates of the angular acceleration of the aircraft about its center of mass, $\dot{\omega}_{brb} = m_b$, are integrated to yield the body coordinates of angular velocity relative to the runway, ω_{brb}, and another integration according to equation (6) results in the direction cosine matrix C_{br}, locating the body axes $\overleftrightarrow{b}$ relative to the runway axes $\overleftrightarrow{r}$.

It should be noted that the presence of the direction cosine matrix C_{br} causes the state space to be nonflat, but a small angle patch of the state space is a subset of R^{30}. The analysis of the large angle behavior of the system is complicated by this relatively large dimension and type of the state space. The analysis is further complicated by the nature of the force and moment generator, which is generally high-dimensional, multiaxis, cross-coupled, and nonlinear.

In a typical real-time, piloted simulation, the algorithms shown in figure 9 are connected to several other systems, such as a cab that provides to the pilot translational and rotational motion cues, and a cockpit that provides the pilot access to the controls and visual, tactile, audio, and electronic displays. The simulations are quite realistic. In fact, in many cases the control software is loaded into the simulation computers, where it is checked out by the test pilot and then loaded without changes and in exactly the same way into the flight computer and flight tested.

The remainder of the report assumes a high-fidelity force and moment algorithm $F_b = (f_b^f, f_b^m)$.

3 OUTPUT TRAJECTORY COMMAND GENERATOR

The purpose of the command generator is to provide the interface between the discrete and analog levels. As already noted, the signals in the discrete level are best described as sentences, which compose the current plan of action generated by the automatic reasoning system. The signals in the analog level are real functions of time. At the detailed level these signals are, of course, not analog but sampled data. However, since the present variables are relatively slowly varying, while the usual sampling rate is high, sampling is largely ignored, and the signals are analog. The high-level plan to be executed by the aircraft may be considered to be a series of maneuvers. Furthermore, there is a hierarchy of maneuvers in the sense that a particular maneuver may be further parsed into a more-detailed concatenation of simpler maneuvers. The maneuvers have both qualitative and quantitative components. In this chapter, a method for converting the commands of the sentential (discrete) level to the analog level is developed. The interface is realized by means of a sequence of records (control cards) that specify the complete maneuver. In the next chapter, these maneuvers are converted into detailed guidance commands, which specify the nominal evolution of system state and controls.

3.1 Aircraft Maneuvers

The type of maneuvers that the automatic system should be able to execute is illustrated by the following examples.

Maneuver $M_1 = M_x(\Delta x, \Delta t)$. In this maneuver, the aircraft is required to step forward a given distance Δx in a given amount of time Δt. Furthermore, the aircraft is in hover at both the beginning and the end of the maneuver. This relatively high-level maneuver may be parsed into a sequence of more elementary submaneuvers, as follows.

$$M_1 = hover \bullet accelerate \bullet coast \bullet decelerate \bullet hover$$

The specific case of stepping forward 300 feet is shown in figure 10, where motion is from right to left. In panel (a) time is zero and the aircraft is in hover with zero pitch attitude and with nozzle at 90 degrees. The control mode is VTOL, in which both the throttle and the nozzle are available for longitudinal force control. The next submaneuver is to accelerate to 8.2 ft/sec^2, holding pitch attitude at zero degrees. At (b), 4.5 seconds later, the position, velocity, and acceleration are, respectively, 16 ft, 15 ft/sec, and 8 ft/sec^2; the nozzle angle changed to 74 degrees to provide the horizontal force needed for the specified acceleration. The next submaneuver (panel c) is to coast at 30 ft/sec for two seconds. Finally, the aircraft is commanded to decelerate to zero velocity. It may be noted that, while the motion of the aircraft center of mass is symmetric about the midpoint (c), the control is not, because the available nozzle range is from 5 to 100 degrees. Therefore, the control mode has to be switched to one in which the nozzle becomes a parameter (in the present example driven to 90 degrees), and the control action flows through the pitch axis. This control scheme of tipping the thrust is typical in the control of helicopters. For this reason this control mode is denoted as HEL. In panel (e), at time 18 sec, the aircraft is hovering over the point 300 ft from the starting point, as specified by the macromaneuver $M_x(300, 18)$.

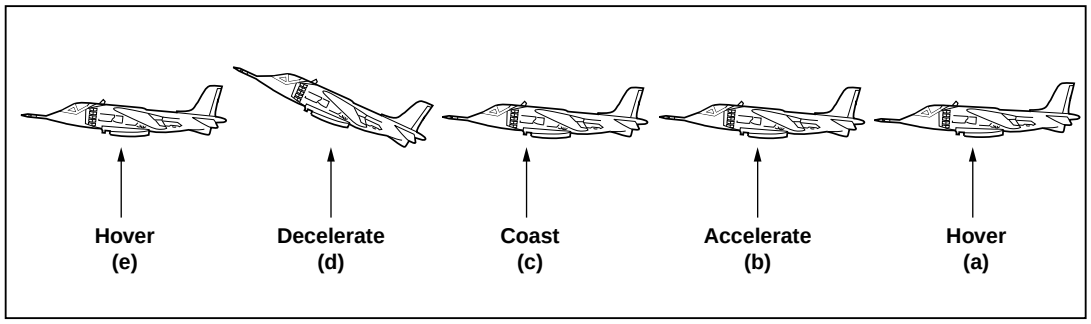

Figure 10. The step forward maneuver, $M_x(300, 18)$.

Similar maneuvers in the lateral $M_y(\Delta y, \Delta t)$ and vertical $M_z(\Delta z, \Delta t)$ directions are, of course, also of practical interest, as are simultaneous combinations, $M_r(\Delta \vec{r}, \Delta t)$.

Maneuver $M_2 = M_{\boldsymbol{\theta}}(\Delta \boldsymbol{\theta}, \Delta t)$. In this maneuver, the aircraft center of mass remains fixed while the pitch angle is changed by the amount $\Delta\theta$. The duration of the maneuver is Δt. The case of $M_\theta(81, 12)$ is shown in figure 11, where the drawings are offset for clarity. Initially, the aircraft is in the VTOL mode and the pitch is zero. During the maneuver the control mode changes to vertical attitude takeoff and landing (VATOL), that is, a "tail sitter," in which the force control flows through thrust, pitch, and yaw channels. The nozzle becomes a parameter driven to 9 degrees, and the maneuver includes a change of coordinates. The near-horizontal attitudes are usually represented by means of the Euler angles in the standard sequence of roll-pitch-yaw. That representation becomes singular in the vertical attitude. On the other hand, the yaw-pitch-yaw sequence is nonsingular in the neighborhood of the vertical attitude but becomes singular for near-horizontal attitudes. During the maneuver the control system must smoothly patch the two representations.

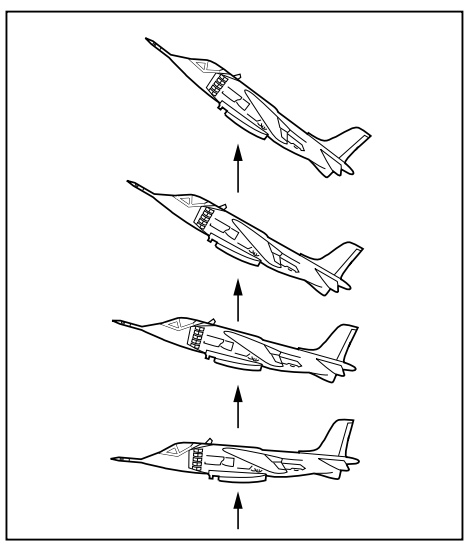

Figure 11. Transition from VTOL to VATOL maneuver, M_2.

Similar maneuvers may be defined for the other two axes, namely, roll and yaw. Whether or not a given maneuver is executable depends on the available control degrees of freedom. If an aircraft, unlike the Harrier, does have a direct three-axis force controller, then translation and rotation become independently controllable. In any case, a macromaneuver may consist of simultaneous translation and rotation submaneuvers. For example, a vertical step together with a large pitch change would look as shown in figure 11.

Maneuver $M_3 = FRISBEE$. This only partly whimsical example illustrates the multiaxis, multisegment nature of the maneuvers. It is assumed that an accurate force and moment model exists in the unusual flight conditions. As shown in figures 12 and 13, the aircraft is taken from steady hover, through a climbing transition, to intermediate speed flight. In addition, the aircraft executes a flat, 360-degree yaw maneuver. As before, the motion is from right to left in the figures. Initially, in panel (a), the Harrier is in steady hover, at which time the command is given to accelerate speed, altitude rate, and yaw. Some of the variables at 7 sec into the maneuver are shown in panel (b). For example, the yaw and yaw rate are 90 degrees and 30 degrees/sec, respectively; the nozzle is in vertical position, the required acceleration being provided by tipping the thrust by means of the roll angle of -6 degrees. In panel (c), the aircraft is flying backward, the acceleration being provided by the forward position of the nozzle at 96 degrees, and the roll angle is back at zero. The speed and climb rate are, respectively, 26 and 15 ft/sec. In panel (d), the roll angle is not quite symmetric in comparison with panel (b) because of growing aerodynamic effects. By 18 sec, panel (e), the yaw maneuver has been completed; the speed is 50 ft/sec, and the nozzle is at 82 degrees. The Harrier is in longitudinal accelerated, climbing flight. The transition from VTOL to the CTOL mode is shown in figure 13. Up to this point the control mode was VTOL, the pitch angle was held at zero, and the flap was set to a maximum of 61 degrees. At 30 sec into the maneuver, panel (g), the control is changed from VTOL to CTOL, in which the lift is generated mainly by means of the angle of attack. The nozzle is commanded to its minimum, and the flaps are commanded to retract to 25 degrees. The flaps have to be partially retracted to satisfy the flap placard, which limits the allowable aerodynamic forces on the flaps for structural reasons. Finally, in panel (i), the aircraft has transitioned to conventional flight at 215 ft/sec speed and 15 ft/sec climb rate.

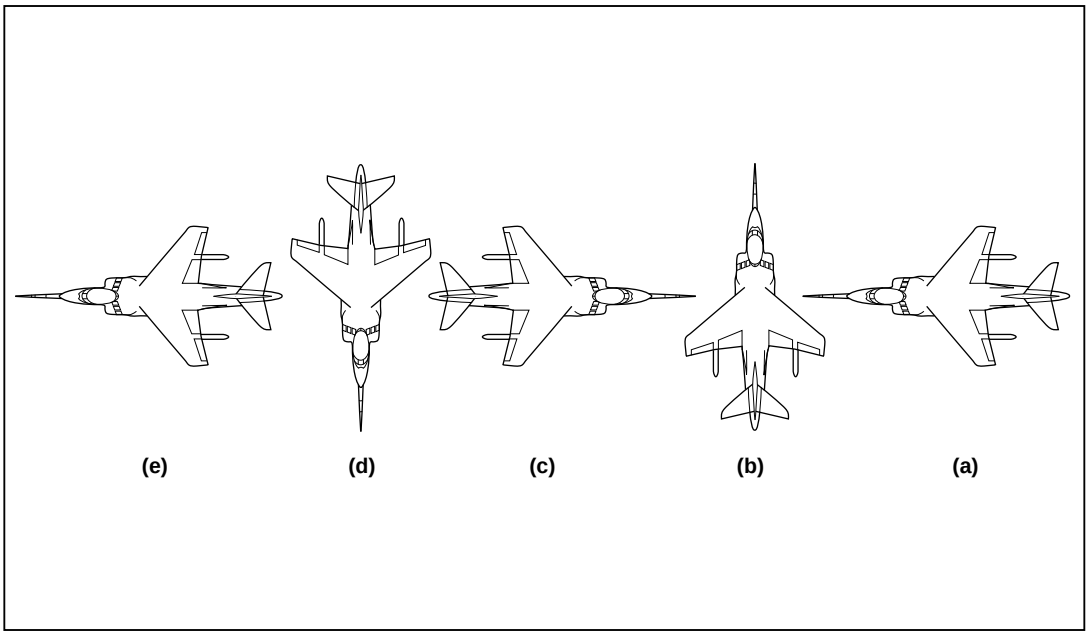

Figure 12. Low-speed part of the Frisbee maneuver, M_3.

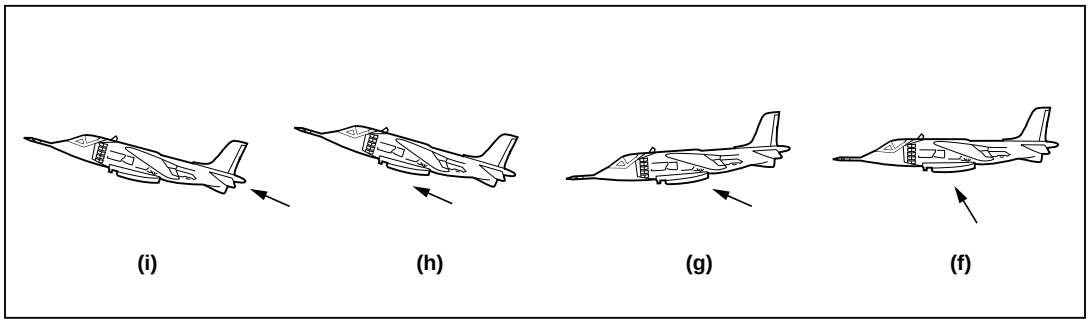

Figure 13. Conventional-flight part of the Frisbee maneuver, M_3.

Maneuver $M_4 = ROLL$. This maneuver, shown in figure 14, illustrates the case in which the definition of the system output changes during the maneuver. In panel (a) the fighter is shown in horizontal, high-speed flight and in the CTOL control mode. The output is the triple of Cartesian coordinates of velocity. The aircraft is commanded to acquire a given vertical velocity, and then to switch to the output mode VYZ in which the variables being controlled are the speed (v) and the lateral (Y) and normal (Z) components of force. Then the command is given to unload the force by driving its component perpendicular to the flight path to zero, as in panel (c), at which point the aircraft is in near free fall, with thrust being used for speed control. Meanwhile, the maneuver analogous to M_1 is executed in the roll axis:

$$M_\phi = hold \bullet accelerate \bullet coast \bullet decelerate \bullet hold$$

to accomplish the 360-degree roll. Finally, the force is reloaded and level flight is recovered, as shown in panel (g).

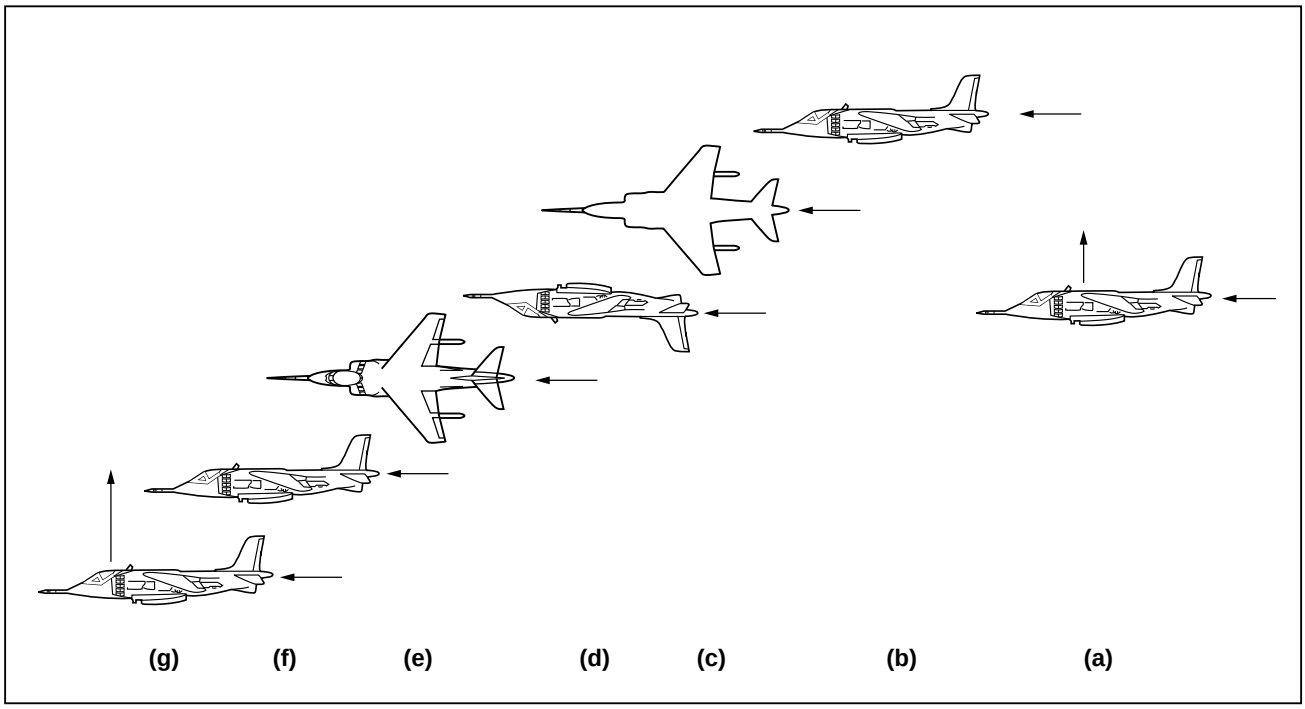

Figure 14. The high-speed roll maneuver, M_4.

Several characteristics may be extracted from the preceding examples:

- There appears to be a hierarchy of maneuvers so that a maneuver at one level may be parsed into a sequence of simpler maneuvers from the next lower level.
- In general, the parsing may not be unique.
- The maneuvers are described both qualitatively and quantitatively.
- It is useful to characterize the qualitative behavior of the aircraft by means of operation modes, such as control modes, coordinates modes, and output modes.

The parsing problem, which is, as yet, unsolved, is the subject of current research. The present report is concerned only with the representation and generation of the elementary maneuvers.

3.2 Polynomial Segments

Consider the example maneuver M_1 shown in figure 10; the objective is construction of a time history of position $y(t)$ such that the aircraft accomplishes the step forward in the allotted time and such that it is flyable, that is, the aircraft motion is consistent with the existing dynamic and saturation constraints. Position is, of course, linked to acceleration by a pair of integrators. Usually, acceleration is not a directly accessible control variable. For example, in panel (d) of figure 10, acceleration is accomplished by means of rotation, which inserts two more integrators in the control path. Angular acceleration is essentially directly accessible because the servos controlling the torque generators have a high bandwidth. However, since the servos are rate limited, one more integrator is inserted in the control path to obtain continuity of the control variable. Similar considerations apply to other controls. As a result, in order to be a possible motion of aircraft position, y and four of its derivatives must be continuous at all times. Consequently, the special case of a string of integrators shown in figure 15 with $y = x_1$ is a model of the single-axis-motion command generator. The kth time derivative of x_1 is

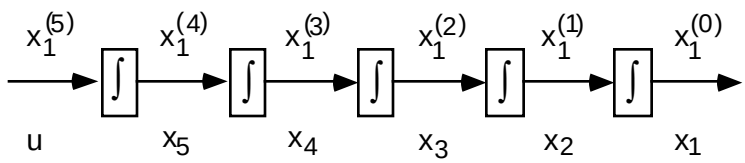

Figure 15. Single string of integrators as model for single-axis position generator.

denoted by $x_1^{(k)}$. Let the state and control of this system be denoted by, respectively,

$$\begin{aligned} x &= \left[x_1^{(0)}, x_1^{(1)}, x_1^{(2)}, x_1^{(3)}, x_1^{(4)}\right] \\ u &= x_1^{(5)} \end{aligned} \tag{27}$$

Suppose that the boundary conditions $[x(0), x(T)]$ are to be connected by a polynomial satisfying the structure in figure 15. For n integrators, the boundary conditions require $2n$ values, which are provided by a polynomial of degree $2n-1$. Consequently,

$$x(t) = E_1(t)c_1 + E_2(t)c_2 \tag{28}$$

where the two square matrices are given by

$$E_1(t) = \begin{pmatrix} 1 & t & t^2/2! & \dots & t^{n-1}/(n-1)! \\ 0 & 1 & t & \dots & t^{n-2}/(n-2)! \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & 1 \end{pmatrix} \tag{29}$$

and

$$E_2(t) = \begin{pmatrix} t^n/n! & t^{n+1}/(n+1)! & \dots & t^{2n-1}/(2n-1)! \\ t^{n-1}/(n-1)! & t^n/n! & \dots & t^{2n-2}/(2n-2)! \\ \dots & \dots & \dots & \dots \\ t & t^2/2! & \dots & t^n/n! \end{pmatrix} \tag{30}$$

Since $E_1(0) = I$, $E_2(0) = 0$, and $E_2(t)$ is nonsingular for $t > 0$,

$$\begin{array}{rcl} c_1 & = & x(0) \\ c_2 & = & E_2^{-1}(T)[x(T) - E_1(T)x(0)] \end{array} \tag{31}$$

Therefore, the polynomial with n continuous derivatives that satisfies the boundary condition $[x(0), x(T)]$ is given by

$$P[t, T, x(0), x(T)] = [E_1(t) - E_2(t)E_2^{-1}(T)E_1(T)]x(0) + E_2(t)E_2^{-1}(T)x(T) \tag{32}$$

which is the optimal solution for a string of n integrators with the quadratic penalty on the input:

$$J = \int_0^T \left[x^{(n+1)}\right]^2 (\tau) d\tau \tag{33}$$

The polynomial solution has the following two computationally desirable properties: Suppose that it is necessary to insert a new control point $0 < T_1 < T$ inside the current interval. That will not change the solution, since for $t > T_1$

$$P[t, T, x(0), x(T)] = P\{t - T_1, T_2 - T_1, P[T_1, T, x(0), x(T)], x(T)\} \tag{34}$$

Furthermore, as can be easily verified, the time variable scales, so that,

$$E_1(t/T) = D_T E_1(t) D_T^{-1} \quad , \qquad E_2(t/T) = T^{-n} D_T E_2(t) D_T^{-1} \tag{35}$$

where

$$D_T = diag\left[1, T, \ldots, T^{n-1}\right] \tag{36}$$

and, therefore,

$$P[t, T, x(0), x(T)] = D_T^{-1} P[t/T, 1, D_T x(0), D_T x(T)] \tag{37}$$

Of particular practical interest is the case of a step of size $x_1(T)$:

$$\begin{array}{rcl} x(0) & = & 0 \\ x(T) & = & x_1(T)\delta_1 \end{array} \tag{38}$$

where δ_i is a column of zeros except in row i, where it is one. The corresponding solution is

$$\psi_1[t, T, x_1(T)] = x_1(T) D_T^{-1} P(t/T, 1, 0, \delta_1) \tag{39}$$

Consequently, properties of the function $t \to \psi[t, T, x_1(T)]$ can be obtained from the normalized function $\tau \to \psi(\tau, 1, 1)$. For example, the maximum absolute value of each coordinate for $t \in [0, T]$, which is of considerable practical interest, is related to that over $\tau \in [0, 1]$ by the simple relation

$$max|\psi_i[t, T, x_1(T)]| = x_1(T) T^{1-i} max|\psi_i(\tau, 1, 1)| \tag{40}$$

So, a unit step in position for $T = 4$, which is shown in figure 16, may serve as a standard. The upper panel shows $x_1 = x^{(0)}$ (solid) and $x_2 = \dot{x}_1 = x_1^{(1)}$ (dotted); similarly, the middle panel shows $(x_3, x_4) = \left[x_1^{(2)}, x_1^{(3)}\right]$, and the bottom panel shows $(x_5, u) = \left[x_1^{(4)}, x_1^{(5)}\right]$. The absolute maxima

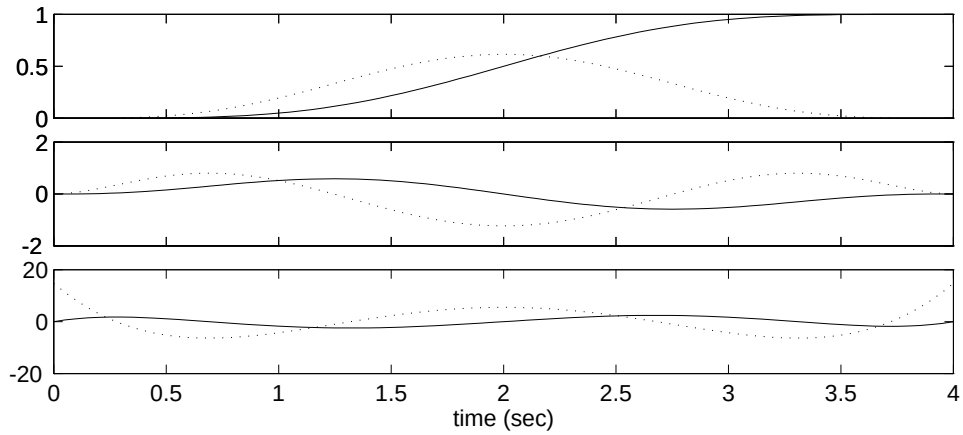

Figure 16. Unit position maneuver for $T = 4$.

$\left[|x^{(j)}|_{max},\ j = 0, \ldots, 5\right]$ for a unit step in position for any duration T may be computed from the corresponding maxima for the case of $T = 4$ by means of equation (40). The first row $M_{0j}(T)$ in table 1 gives these maxima for a unit step in position x_1 that is accomplished in time T. The second row in table 1 represents the maxima for the unit velocity step. For this maneuver the boundary conditions are given by $x_1^{(1)}(T) = 1$, $x_i(T) = 0,\ i = 2, ...5$ and $x_1(T)$ is free—it is not matched. Similarly, for the other rows in the table the state coordinates to the left of 1.000 are not matched; they are "don't cares."

Table 1. Maxima $M_{i,j}(T)$ of elementary maneuvers

	x	v	a	$\dot{a}$	$\ddot{a}$	$a^{(3)}$
x	1.000	$2.461T^{-1}$	$9.371T^{-2}$	$78.74T^{-3}$	$622.5T^{-4}$	$15120T^{-5}$
v	$0.500T$	1.000	$2.188T^{-1}$	$7.511T^{-2}$	$52.50T^{-3}$	$840.0T^{-4}$
a	$0.143T^2$	$0.500T^1$	1.000	$1.875T^{-1}$	$5.773T^{-2}$	$60.00T^{-3}$
$\dot{a}$	$0.033T^3$	$0.150T^2$	$0.500T^1$	1.000	$1.500T^{-1}$	$6.000T^{-2}$
$\ddot{a}$	$0.008T^4$	$0.042T^3$	$0.167T^2$	$0.500T^1$	1.000	$1.000T^{-1}$

The unit velocity v maneuver for $T = 3$ is shown in figure 17. The absolute maxima in this case are given by the second row (first derivative) $M_{1j}(3)$. For example, $|x^{(5)}|_{max} = 840/81$. The unit acceleration a maneuver for $T = 2$ is shown in figure 18.

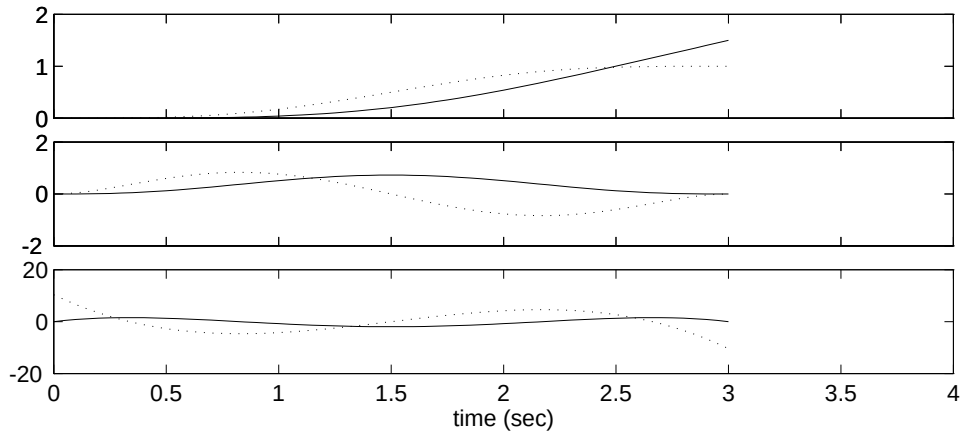

Figure 17. Unit velocity maneuver for $T = 3$.

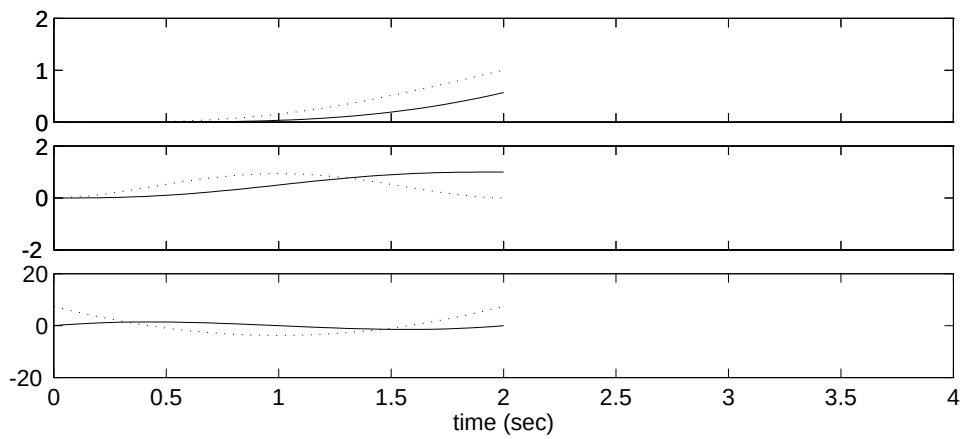

Figure 18. Unit acceleration maneuver for $T = 2$.

So far the states were assumed to be unconstrained, but in practice, there may be constraints. Thus, it may be necessary to stay within a speed limit, $v \leq v_{max}$, and impose limits on the acceleration and higher derivatives. For example, in the case of the Harrier in the helicopter (HEL) control mode, forward acceleration is accomplished by pitching the aircraft forward, so that, approximately, the forward acceleration $a_{r1} = -g\theta$, where g is the acceleration of gravity and θ is the pitch angle in radians. Situations arise when it is desirable to limit the pitch angle and three of its time derivatives to some given values. Pitch acceleration is limited by the capabilities of the pitch moment generator, and pitch acceleration rate must be consistent with the rate limits, which may be imposed on the pitch channel actuator servo to prevent rapid hard-over failures. That means that the acceleration and higher derivatives

should be constrained as follows:

$$\begin{pmatrix} |x^{(2)}| \\ |x^{(3)}| \\ |x^{(4)}| \\ |x^{(5)}| \end{pmatrix} \leq g \begin{pmatrix} \theta_{max} \\ \dot{\theta}_{max} \\ \ddot{\theta}_{max} \\ \theta^{(3)}_{max} \end{pmatrix} \tag{41}$$

For example, typical constraints, including a speed limit for the Harrier in the low-speed HEL mode, may be

$$\begin{pmatrix} |x^{(1)}| \\ |x^{(2)}| \\ |x^{(3)}| \\ |x^{(4)}| \\ |x^{(5)}| \end{pmatrix} \leq \begin{pmatrix} 30 \\ 12 \\ 12 \\ 12 \\ 24 \end{pmatrix} \tag{42}$$

These limits would be violated if, for example, the maneuver $M_1 =$ step forward in 18 seconds were to be executed as a single segment as shown in figure 19. The maneuver is very smooth, with small $x_1^{(2)}$ and $x_1^{(3)}$, but the speed $v = x_1^{(1)}$ exceeds its limit of 30 ft/sec. Next an approach for satisfying constraints is considered.

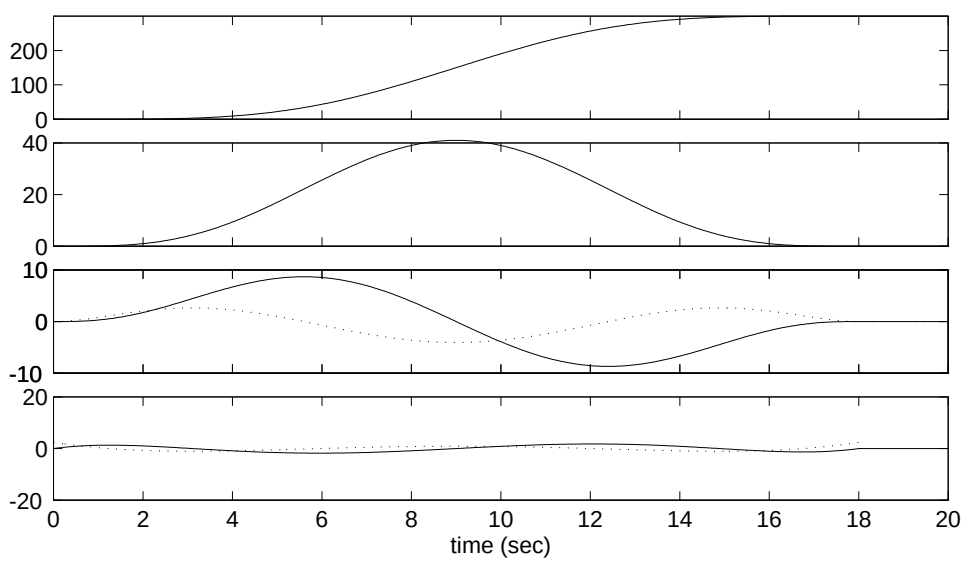

Figure 19. Unconstrained realization of the maneuver M_1.

3.3 Control Cards

As is well known, for reasonable situations with constraints, the optimal trajectory is a concatenation of segments that are either interior or on the constraining boundary. Strict optimality is not pursued further since the structure of the overall control system is of primary concern. However, the insight that the optimal solution consists of a sequence of control points that are connected by polynomials is used.

In the present case of a string of integrators, in order to stay on the (constant) boundary $x_i = x_{imax}$, the higher derivatives of x_i must be zero. Consequently, the unconstrained elementary maneuvers listed in table 2 are considered.

Table 2. Elementary maneuvers

$\sigma_0^k(x,T)$	get steady position	x in time T
$\sigma_1^k(v,T)$	get steady velocity	v in time T
$\sigma_2^k(a,T)$	get steady acceleration	a in time T
$\sigma_3^k(\dot{a},T)$	get steady accel. rate	$\dot{a}$ in time T
$\sigma_4^k(\ddot{a},T)$	get steady accel. accel.	$\ddot{a}$ in time T

The lower index on σ indicates the first derivative that is not to be ignored; the upper index indicates the highest derivative that must be matched. Thus, σ_2^5 indicates that position and velocity are to be ignored (not matched), and all derivatives of acceleration up to order 3 must be zero at the end of the interval. Many maneuvers of practical interest may be parsed into the these elementary maneuvers. For example, the overall goal of the step forward maneuver M_1 may be achieved in several different ways:

(1) The maneuver may be achieved as a single segment (shown in figure 19) that is given by

$$M_1 = \sigma_0^5(\Delta x, T) \tag{43}$$

The maxima are given by the first row in table 1: $|x_1^{(j)}|_{max} = \Delta x M_{0j}(T)$, $\Delta x = 300, T = 18$.

(2) The maneuver may be achieved as a symmetric concatenation of three velocity segments (shown in figure 20 for $T_1 = 8,\ T_2 = 2$) given by

$$M_1 = \sigma_1^5(v,T_1)\sigma_1^5(v,T_2)\sigma_1^5(0,T_1) \tag{44}$$

where the duration of the maneuver is

$$T = 2T_1 + T_2 \tag{45}$$

and size of the position step is the product

$$x = (T_1 + T_2)v \tag{46}$$

The maxima of velocity and higher derivatives for this maneuver are given by the second row in table 1,

$$|v^{(k)}|_{max} = vM_{1,k+1}(T_1) \tag{47}$$

(3) The maneuver may also be synthesized as a symmetric concatenation of seven elementary acceleration segments (shown in figure 21 for $T_1 = 4,\ T_2 = 0,\ T_3 = 2$) as follows:

$$M_1 = \sigma_2^5(a,T_1)\sigma_2^5(a,T_2)\sigma_2^5(0,T_1) \bullet \sigma_2^5(0,T_3) \bullet \sigma_2^5(-a,T_1)\sigma_2^5(-a,T_2)\sigma_2^5(0,T_1) \tag{48}$$

The duration of the complete maneuver is

$$T = 4T_1 + 2T_2 + T_3 \tag{49}$$

The maximum velocity is

$$v = a(T_1 + T_2) \tag{50}$$

and the final position is

$$x = (T_1 + T_2)(2T_1 + T_2 + T_3)a \tag{51}$$

The absolute maxima of acceleration and its derivatives are given by

$$|a^{(k)}|_{max} = aM_{2,k+2}(T_1),\ k = 0, 3 \tag{52}$$

The maxima for the three realizations are compared in table 3.

Table 3. Absolute maxima for the three realizations of the maneuver M_1

	x	v	a	$\dot{a}$	$\ddot{a}$	$a^{(3)}$
$\sigma_0^{(5)}$	300.0	41.0	8.7	4.1	1.8	2.4
$(\sigma_1^{(5)})^3$	300.0	30.0	8.2	3.5	3.1	6.2
$(\sigma_2^{(5)})^7$	300.0	30.0	7.5	3.5	2.7	7.0

As noted previously, the position maneuver is smooth but violates the speed limit. The other two maneuvers are quite similar and, as shown in figures 20 and 21, step the aircraft forward 300 feet in 18 seconds without violating the constraints. It may be noted that, as is to be expected, the satisfaction of constraints requires greater activity in the control u. Of course, the increase may be expected to be smaller for the optimum solution.

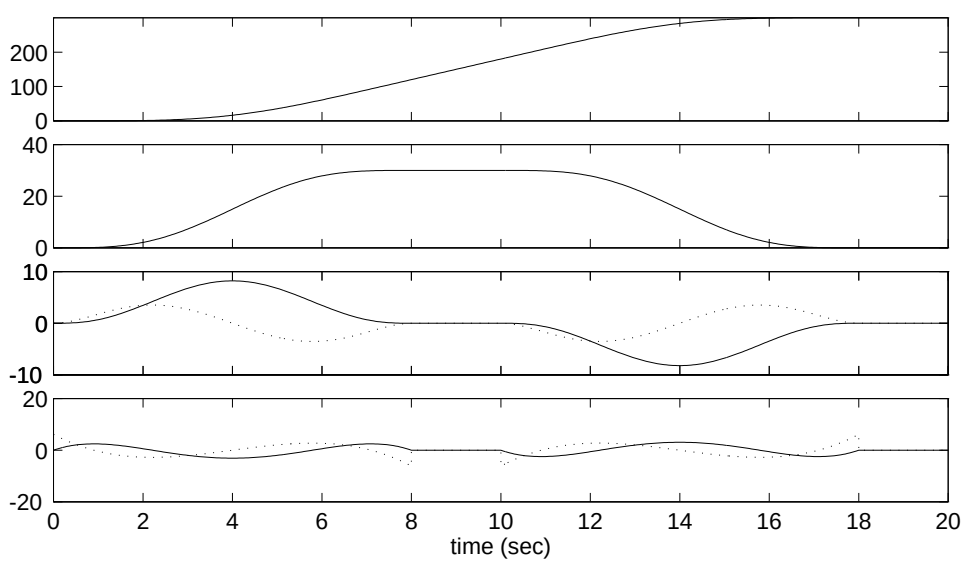

Figure 20. The three-segment maneuver $M_{13} = (\sigma_1^{(5)})^3$.

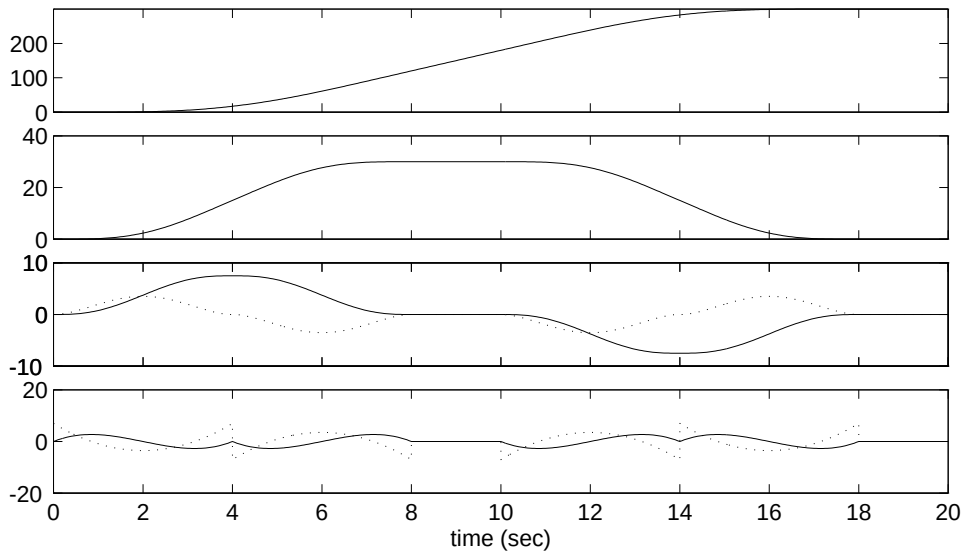

Figure 21. The seven-segment maneuver $M_{17} = (\sigma_2^{(5)})^7$.

It is helpful to think of a composite maneuver as a sequence of *control cards*. For example, the three-segment maneuver $M_1 = (\sigma_1^{(5)})^3$ may be represented by the sequence $\mathcal{C} = C_0 C_1 C_2 C_3$ of four control cards, as follows:

$$C_0 = \begin{bmatrix} T = 0 & & & & & & \\ x = & 0 & 0 & 0 & 0 & 0 & ** \end{bmatrix}$$

$$C_1 = \begin{bmatrix} T = 8 & & & & & & \\ x = & * & 30 & 0 & 0 & 0 & 14 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} T = 2 & & & & & & \\ x = & * & 30 & 0 & 0 & 0 & 14 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} T = 8 & & & & & & \\ x = & 300 & 0 & 0 & 0 & 0 & 04 \end{bmatrix}$$

The first card C_0 specifies the initial condition. The second card C_1 commands that the segment is to last 8 seconds and that the terminal velocity $\dot{x} = 30$ ft/sec (variable 1), higher derivatives up to fourth (variable 4) are to be zero, while the position is to be ignored. The third card C_2 commands a 2-second coast at 30 ft/sec. The last card commands a steady 300 ft (variable 0).

This representation becomes particularly helpful when x is multidimensional. Complex, multiaxis maneuvers may require many control cards, each with many (e.g., 90) rows of data. In the present structure for the flight vehicle management system, these control cards are the interface from the discrete levels to the analog levels. Before expanding on this point, consider the multiaxis case.

3.4 Coordinate Patching

To be applicable to aircraft, the preceding discussion must be generalized to include multiple axes and rotations. A slight detour into the methodology of dynamic forms is made to considerably simplify the discussion that follows. Much more detail is given in reference 3.

A scalar form of order k of a scalar variable of time is defined by the k-tuple consisting of the variable and k of its time derivatives:

$$SF^k(x) = [x, \dot{x}, x^{(2),\ldots,x^{(k)}}],\ x \in R \tag{53}$$

For example, the system in figure 15 with $n = 4$ may be represented by $SF^5(x_1)$. Similarly, vector and matrix forms are given, respectively, by

$$VF^k(v) = \left(v, \dot{v}, v^{(2)}, \ldots, v^{(k)} \right),\ v \in R^n \tag{54}$$

and

$$MF^k(M) = \left(M, \dot{M}, M^{(2)}, \ldots, M^{(k)} \right),\ M \in R^{n_1 \times n_2} \tag{55}$$

As described in reference 3, sums, products, and other elementary functions are easily extendible to dynamic forms. Then, more-complicated functions may be built out of the elementary set. Thus, for example, the possibly time-varying affine map

$$v = Mu + z \tag{56}$$

may be raised from order zero to order k as follows:

$$VF^k(v) = MF^k(M) \star VF^k(u) + VF^k(z) \tag{57}$$

Let $\overleftrightarrow{a}$ be a three-dimensional, right-handed, orthonormal axis system. The $\overleftrightarrow{a}$-coordinates of a vector $\vec{x}$ will be denoted by the column x_a. The attitude of another axis system $\overleftrightarrow{b}$ may be defined relative to $\overleftrightarrow{a}$ by the direction cosine matrix C_{ba}. The coordinates transform according to $x_b = C_{ba} x_a$. Let the angular velocity of $\overleftrightarrow{b}$ relative to $\overleftrightarrow{a}$ be denoted by the vector $\vec{\omega}_{ba}$; its body coordinates are ω_{bab}. Then the time derivative of the direction cosine matrix,

$$\dot{C}_{ba} = S(\omega_{bab}) C_{ba} \tag{58}$$

where, as before, the skew-symmetric matrix, given for any $z \in R^3$ by

$$S(z) = \begin{pmatrix} 0 & z_3 & -z_2 \\ -z_3 & 0 & z_1 \\ z_2 & -z_1 & 0 \end{pmatrix} \tag{59}$$

represents the cross-product operator. The ability to generate direction cosine matrices by a structure of the type shown in figure 22 is of interest. It is convenient to represent this structure as the rotation form:

$$RF^k(C_{ba}) = \left[C_{ba}, \omega_{bab}, \dot{\omega}_{bab}, \omega_{bab}^{(2)}, \ldots, \omega_{bab}^{(k-1)}\right] \tag{60}$$

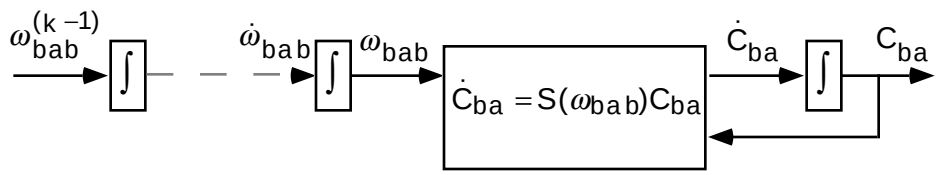

Figure 22. Rotational form $RF^k(C_{ba})$.

The closely related matrix form is given by

$$MF^k(C_{ba}) = \left[C_{ba}, \dot{C}_{ba}, C_{ba}^{(2)}, \ldots, C_{ba}^{(k)}\right] \tag{61}$$

The kinematic equation (58) and

$$S(\omega_{bab}) = \dot{C}_{ba} C_{ba}^T \tag{62}$$

where $()^T$ denotes the transpose, provide a nonsingular link between the two forms. The subroutines for transforming them into each other are denoted as

$$\begin{aligned} MF^k(C_{ba}) &= MF \prec RF[RF^k(C_{ba})] \\ RF^k(C_{ba}) &= MF \succ RF[MF^k(C_{ba})] \end{aligned} \tag{63}$$

The generalization of the Coriolis' derivative to order k is given by

$$VF^k(v_b) = MF^k(C_{ba}) \star VF^k(v_a) = MF \prec RF[RF^k(C_{ba})] \star VF^k(v_a) \tag{64}$$

This subroutine outputs the b-components v_b of $\vec{v}$ and the k time derivatives of v_b for given a-components v_a and its k time derivatives and the attitude C_{ba} and angular velocity and its derivatives.

The transpose of a matrix form,

$$[MF^k(M)]^T = MF^k(M^T) \tag{65}$$

Consequently, the inverse of a rotation form is given by

$$RF^k(C_{ab}) = MF \succ RF\{[MF^k(C_{ba})]^T\} \tag{66}$$

It is often necessary to represent rotations in terms of Euler angles in some desired sequence. That is,

$$C_{ba} = E_{q_1}(\alpha_1) E_{q_2}(\alpha_2) E_{q_3}(\alpha_3) \tag{67}$$

where the sequence $q = (q_1, q_2, q_3)$ with $q_2 \neq q_1, q_3$, and E_i is an elementary rotation about axis i. This representation has a gimbal lock (the Jacobian matrix is singular) at $\alpha_2 = \pi/2$ for a nonrepeating sequence, and at $|\alpha_2| = 0$ for a repeating sequence. Since derivatives of Euler angles are needed, an angle form is defined as follows:

$$AF_q^k(\alpha) = \begin{pmatrix} VF^k(\alpha) \\ q \end{pmatrix} \tag{68}$$

and the subroutine for the forward and inverse parameterizations (away from the singularities) is defined as:

$$\begin{aligned} MF^k(C_{ba}) &= MF \prec AF_q[AF_q^k(\alpha_{ba})] \\ AF_q^k(\alpha_{ba}) &= MF \succ AF_q[MF^k(C_{ba})] \end{aligned} \tag{69}$$

This algorithm provides a bidirectional link between a direction cosine matrix and the corresponding angular velocity and its derivatives on the one hand, with the corresponding Euler angles and their time derivatives on the other.

All these easily coded algorithms are very useful in practice. Now consider a maneuver such as the transition from VTOL to VATOL, as shown in figure 11. The standard representation by the nonrepeating sequence $q = (1, 2, 3)$:

$$C_{tr} = E_{123}(\gamma) \equiv E_1(\gamma_1)E_2(\gamma_2)E_3(\gamma_3) \tag{70}$$

which is nonsingular for near horizontal attitudes, becomes singular in the vertical attitude. A repeating sequence such as $p = (3, 2, 3)$

$$C_{tr} = E_{323}(\beta) \equiv E_3(\beta_1)E_2(\beta_2)E_3(\beta_3) \tag{71}$$

is nonsingular near vertical attitudes. To achieve large-angle maneuvers, it is necessary to patch coordinates smoothly. That is easily done by means of the algorithms defined previously.

$$AF_p^k(\beta) = H_q^p\left[AF_q^k\right] \equiv MF \succ AF\left\{MF \prec AF\left[AF_q^k(\gamma)\right]\right\} \tag{72}$$

Let the coordinate-designating mode variable be denoted by μ_α. This mode variable may take on many values, but for the VTOL to VATOL maneuver, two are noted: horizontal attitude (HAT) and vertical attitude (VAT). Switching from one coordinate system to another is a discrete event initiated by a change in μ_a. In the horizontal mode $\mu_a = HAT$, both sets of coordinates are computed according to

$$\begin{pmatrix} AF_q^k(\gamma) \\ AF_p^k(\beta) \end{pmatrix} = \begin{pmatrix} AF_q^k(\gamma) \\ H_q^p[AF_q^k(\gamma)] \end{pmatrix} \tag{73}$$

That is, γ is free to move in any preassigned manner while β is slaved to the image of γ. On the other hand, in the vertical mode $\mu_a = VAT$, β is free while γ is slaved:

$$\begin{pmatrix} AF_q^k(\gamma) \\ AF_p^k(\beta) \end{pmatrix} = \begin{pmatrix} H_p^q[AF_p^k(\beta)] \\ AF_p^k(\beta) \end{pmatrix} \tag{74}$$

The coordinate switch may be commanded by means of control cards. For example, consider a sequence of control cards $\mathcal{C} = \ldots C_i C_{i+1} \ldots$, where

$$C_i = \begin{bmatrix} T = 2 & & & & & & & \\ \mu_a = HAT & & & & & & & \\ \gamma_1 & f & 0 & 0 & 0 & 0 & 0 & 04 \\ \gamma_2 & f & 0 & 10 & 0 & 0 & 0 & 14 \\ \gamma_3 & f & 0 & 11 & 0 & 0 & 0 & 14 \\ \beta_1 & s & 0 & 0 & 0 & 0 & 0 & 04 \\ \beta_2 & s & 0 & 0 & 0 & 0 & 0 & 04 \\ \beta_3 & s & 0 & 0 & 0 & 0 & 0 & 04 \end{bmatrix}$$

and

$$C_{i+1} = \begin{bmatrix} T = 7 & & & & & & & \\ \mu_a = VAT & & & & & & & \\ \gamma_1 & s & 0 & 0 & 0 & 0 & 0 & 04 \\ \gamma_2 & s & 0 & 0 & 0 & 0 & 0 & 04 \\ \gamma_3 & s & 0 & 0 & 0 & 0 & 0 & 04 \\ \beta_1 & f & 0 & 0 & 0 & 0 & 0 & 04 \\ \beta_2 & f & 90 & 0 & 0 & 0 & 0 & 04 \\ \beta_3 & f & 0 & 11 & 0 & 0 & 0 & 14 \end{bmatrix}$$

Control card C_i commands a segment of 2 seconds duration in the horizontal attitude coordinatization mode $\mu_a = HAT$, in which the γs are free to satisfy the indicated terminal conditions: zero roll ($\gamma_1 = 0$), constant pitch rate $\dot{\gamma}_2 = 10$ deg/sec, and constant yaw rate $\dot{\gamma}_3 = 11$ deg/sec. The βs are to be slaved to the image of the γs. The next control card C_{i+1} commands a segment of 7 seconds duration in the vertical attitude coordinatization mode $\mu_a = VAT$. The initial condition to be satisfied by the βs are given by the slaved βs of the previous card C_i. The terminal conditions are constant $\beta_1 = 0$, $\beta_2 = 90$, and $\dot{\beta}_3 = 11$. The γs are slaved to the image of the βs.

As an example of this type of coordinate patching, consider a large-angle maneuver, as shown in figure 23. The maneuver lasting 40 seconds pitches the time-axis system from zero to 90 degrees and then back to zero while yawing through the complete 360 degrees. In the vertical position, ($\vec{t}_1 \approx -\vec{r}_3$), $\overleftrightarrow{t}$ is tilted about $\vec{t}_3$ from zero to 3 degrees and back to zero. The complete maneuver is synthesized with eight control cards. The evolutions of γ and $\dot{\gamma}$ are shown in the first two panels of the figure. The Euler angle forms are set to zero within one degree of their respective singularities. The evolutions of β and $\dot{\beta}$ are shown in the bottom two panels. The 3-degree tilt is shown in the top panel of figure 24.

The corresponding evolution of the direction cosine matrix C_{tr} may be computed as discussed previously by means of the algorithm

$$RF^k(C_{tr}) = \begin{cases} MF \succ RF\{MF \prec AF[AF^k_{123}(\gamma)]\} & \text{if } \mu_a = HAT \\ MF \succ RF\{MF \prec AF[AF^k_{323}(\beta)]\} & \text{if } \mu_a = VAT \end{cases} \tag{75}$$

The angular velocity, angular acceleration, and their derivatives are shown in in the bottom panels of the figure. The components of ω_{trt} are measures of the rate gyros attached to $\overleftrightarrow{t}$. Note that $\dot{\gamma}$, which looks like yaw rate, becomes roll rate as $\overleftrightarrow{t}$ becomes vertical and then again yaw rate at the end of

the maneuver. The 3-degree tilt of $\vec{t}_1$ submaneuver has no effect on the roll channel but activates both pitch and yaw. Finally, it may be noted that, as required, both angular velocity and acceleration are continuous throughout the maneuver, despite segmentation and coordinate changes.

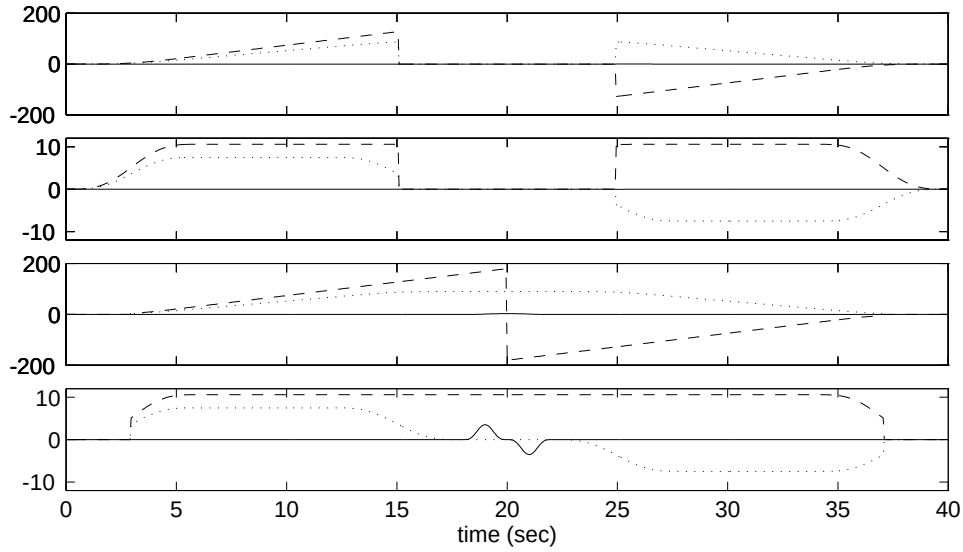

Figure 23. Vertical attitude maneuver: Euler angles.

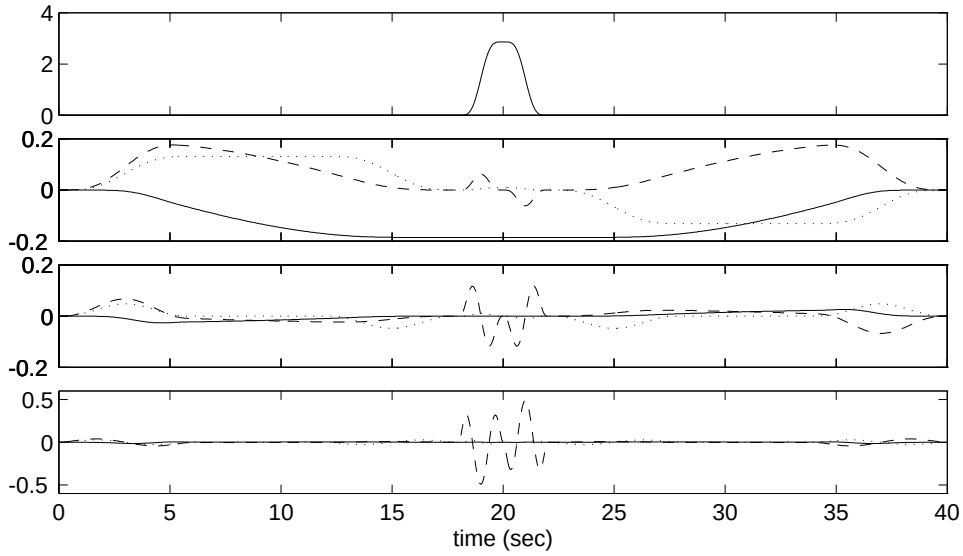

Figure 24. Vertical attitude maneuver: angular velocity.

3.5 Transition Dynamics

In the preceding discussion of the coordinate change $\gamma \leftrightarrow \beta$, there was enough freedom to patch continuously across the mode switch. That is not always the case. Two very useful coordinatizations of C_{tr}, namely the runway path axes RPA and wind path axes WPA, are given in terms of the trajectory of the origin of $\overleftrightarrow{t}$, as follows. Consider an axis system $\overleftrightarrow{v}$, defined so that the first unit vector $\vec{v}_1$ points along the velocity $\vec{v}$ of $\overleftrightarrow{v}$ relative to the runway-fixed axes $\overleftrightarrow{r}$. In addition, let the total specific force

$$\vec{f} = \dot{\vec{v}} - g\vec{r}_3 \tag{76}$$

be in the $(\vec{v}_1, \vec{v}_3)$ plane, pointing up. That is,

$$\begin{array}{rcl} \vec{v}_1 & = & \vec{v}/|\vec{v}| \\ \vec{v}_2 & = & \vec{v}_1 \times \vec{f}/|\vec{v}_1 \times \vec{f}| \\ \vec{v}_3 & = & \vec{v}_1 \times \vec{v}_2 \end{array} \tag{77}$$

Then the attitude of $\overleftrightarrow{v}$ relative to $\overleftrightarrow{r}$ is given by the direction cosine matrix C_{vr}, whose rows are the r-coordinates of $\overleftrightarrow{v}$:

$$C_{vr} = (\, v_{1r}, v_{2r}, v_{3r}\,)^T \tag{78}$$

This zero-order computation can be easily raised to order k by means of the methodology of dynamic forms:

$$MF^k(C_{vr}) = MF \prec VR[VF^{k+1}(v_r)] \tag{79}$$

which may then be transformed into the corresponding rotation form $RF^k(C_{vr})$. In the runway path axis mode ($\mu_a = RPA$), $MF^k(C_{tr}) = MF^k(C_{vr})$.

Another frequently used axis system $\overleftrightarrow{w}$ is defined in terms of the relative wind. The velocity relative to the average wind is given by

$$\vec{v}^a = \vec{v} - \vec{w} \tag{80}$$

where $\vec{w}$ is the average wind velocity with respect to the runway. The axis system is given by

$$\begin{array}{rcl} \vec{w}_1 & = & \vec{v}^a/|\vec{v}^a| \\ \vec{w}_2 & = & \vec{w}_1 \times \vec{f}/|\vec{w}_1 \times \vec{f}| \\ \vec{w}_3 & = & \vec{w}_1 \times \vec{w}_2 \end{array} \tag{81}$$

so that the first basis vector points in the direction of the relative wind $\vec{v}^a$ rather than in the direction of $\vec{v}$. In the wind path axis coordinatization mode ($\mu_a = WPA$), $MF^k(C_{tr}) = MF^k(C_{wr})$. Other variations on this theme are, of course, possible. The problem is that, while it is possible to switch smoothly from a path-defined mode to an Euler angle mode, the change from Euler angles to path may be discontinuous because the path is generally not a free variable that can be slaved to an attitude. However, the discontinuity may be smoothed by means of transition dynamics, as follows. The idea is to factor the attitude into two rotations, one of which is relaxed to the identity. For example, suppose that the mode is switched at $t = t_1$ from $\mu_a =$ HAT to $\mu_a =$ RPA:

$$C_{tr} = \begin{cases} E_{123}(t), & \text{for } t \le t_1 \\ C_{vr}(t), & \text{for } t_1 < t \end{cases} \tag{82}$$

Then, without smoothing, the discontinuity at t_1 is

$$D(t_1) = E_{123}(t_1) C_{vr}^T(t_1) \tag{83}$$

On the other hand, if we define

$$C_{tr} = \begin{cases} E_{123}(t), & \text{for } t \leq t_1 \\ D(t) C_{vr}(t), & \text{for } t_1 < t \end{cases} \tag{84}$$

and relax $D(t) \to I$, then the transition can be made smooth. This zero-order procedure can be easily lifted to order k:

$$MF^k(C_{tr}) = \begin{cases} MF^k(E_{123}), & \text{for } t \leq t_1 \\ MF^k(D) \star MF^k(C_{vr}), & \text{for } t_1 < t \end{cases} \tag{85}$$

There are many ways to relax $MF^k(C) \to (\, I, 0, \ldots, 0\,)$. One possibility is to express D in terms of Euler angles δ in some sequence q and then drive $AF_q(\delta)(t_1) \to (0, \ldots, 0)$ by means of polynomial segments. Another way is to apply a zero-seeking regulator structure. For example, suppose that D is represented in terms of Euler parameters $(\epsilon, \eta) \in R^3 \times R$:

$$D = I + 2\eta S(\epsilon) + 2S^2(\epsilon) \tag{86}$$

and conversely,

$$\begin{aligned} \eta &= \tfrac{1}{2}[tr(D) + 1]^{\frac{1}{2}} \\ \epsilon &= \tfrac{1}{2} axis(D)/\eta \end{aligned} \tag{87}$$

where for any $x \in R^3$,

$$axis[S(x)] = x \tag{88}$$

The constraint is $\epsilon^2 + \eta^2 = 1$, and the representation is singular for $|\epsilon| = 1$. These zero-order computations are easily raised to order k, as follows:

$$\begin{aligned} (\, VF^k(\epsilon), SF^k(\eta)\,) &= MF \succ PF[MF^k(D)] \\ MF^k(D) &= MF \prec PF[VF^k(\epsilon), SF^k(\eta)] \end{aligned} \tag{89}$$

Associated with the vector part of the parameter form is the three-dimensional string of k integrators, each

$$\epsilon^{(k+1)} = u \tag{90}$$

The initial condition is given by

$$\left[\epsilon(t_1), \ldots, \epsilon^{(k)}(t_1)\right] = VF^k(\epsilon)(t_1) \tag{91}$$

Choose a stable regulator law

$$u = k\left[\epsilon, \ldots, \epsilon^{(k)}\right] \tag{92}$$

Then, provided that $|\epsilon| < 1$, the offset ϵ will decay to zero. The evolution of C_{tr} for $t_1 < t$ is then constructed as follows

$$\begin{aligned} SF^k(\eta) &= [1 - VF^k(\epsilon)^2]^{\frac{1}{2}} \\ MF^k(D) &= MF \prec PF[VF^k(\epsilon), SF^k(\eta)] \\ MF^k(C_{tr}) &= MF^k(D) \star MF^k(C_{vr}) \end{aligned} \tag{93}$$

By construction there is no discontinuity in C_{tr} or k of its time derivatives, and C_{tr} coincides with C_{vr} after the transient has died down.

Several useful ways have been developed to coordinatize large-angle attitudes. The generation of three-dimensional reference trajectories is now considered.

3.6 Output Mode

The coordinate mode, μ_a, discussed in the preceding section, affects the meaning of the system state and control coordinates. In this section the output mode, μ_y, which determines the meaning of the system output (that is, which variables are to be tracked) is considered. Some of the many possibilities are the Cartesian, cylindrical, or spherical coordinates of position, or airspeed and path, or, as in the 360-degree roll maneuver shown in figure 14, speed and lateral and normal forces. Many more output maps are of practical interest. Specifying and switching of output maps may be handled in much the same way as the Euler angles. Thus, for example, consider a 90-degree horizontal turn maneuver with 100 ft/sec initial and final speeds. The maneuver may be realized by means of the Cartesian coordinates, as follows:

M_{xy}: 90-degree Cartesian maneuver

$$
\begin{array}{rcrrrrrr}
T = & 0, \mu_a = RPA, \mu_y = CAR & & & & & & \\
 & VF_1(r_r) & f & 0 & 100 & 0 & 0 & 0 \\
 & VF_2(r_r) & f & 0 & 0 & 0 & 0 & 0 \\
 & VF_3(r_r) & f & 0 & 0 & 0 & 0 & 0 \\
T = & 16.5, \mu_a = RPA, \mu_y = CAR & & & & & & \\
 & VF_1(r_r) & f & 1000 & 0 & 0 & 0 & 0 \quad 04 \\
 & VF_2(r_r) & f & 1000 & 100 & 0 & 0 & 0 \quad 04 \\
 & VF_3(r_r) & f & 0 & 0 & 0 & 0 & 0 \quad 04
\end{array}
$$

The turn is accomplished with one segment in 16.5 sec. The path is shown in figure 25. The attitude of $\overleftrightarrow{t}$ is slaved to the runway path axis system ($\mu_a = RPA$). The corresponding Euler angles α in the standard sequence $q = (1, 2, 3)$ may be extracted from C_{tr}. Figure 26 shows the Euler angles, angular velocity, and angular acceleration of the $\overleftrightarrow{t}$ system relative to the runway $\overleftrightarrow{r}$. Figure 27 shows the evolution of the ground speed v (ft/sec, upper panel), acceleration $\dot{v}$ (ft/sec^2, second panel), t-coordinates of force f_t' (third panel), and curvature κ (in inverse thousand feet), where

$$
\begin{array}{rcl}
v & = & \dot{s} = \dot{s} = t_{1r}^T v_r \\
\dot{v} & = & \ddot{s} \\
f_t' & = & C_{tr}(f_r + g\delta_3)/g \\
\kappa & = & \|d/dt(\frac{v_r}{v})\|/v
\end{array}
\tag{94}
$$

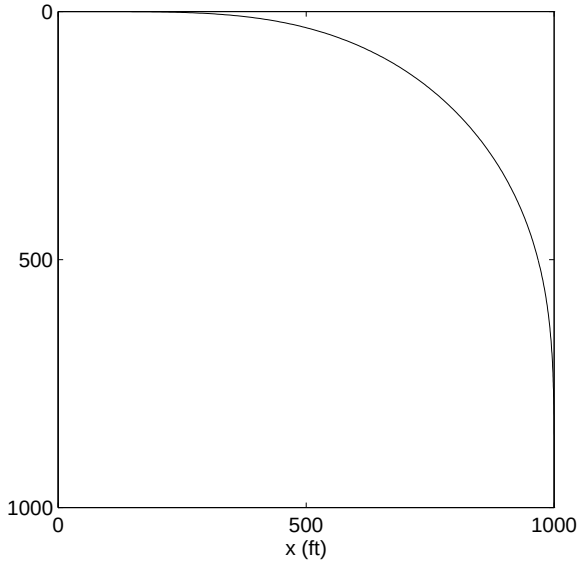

Figure 25. Cartesian maneuver: path.

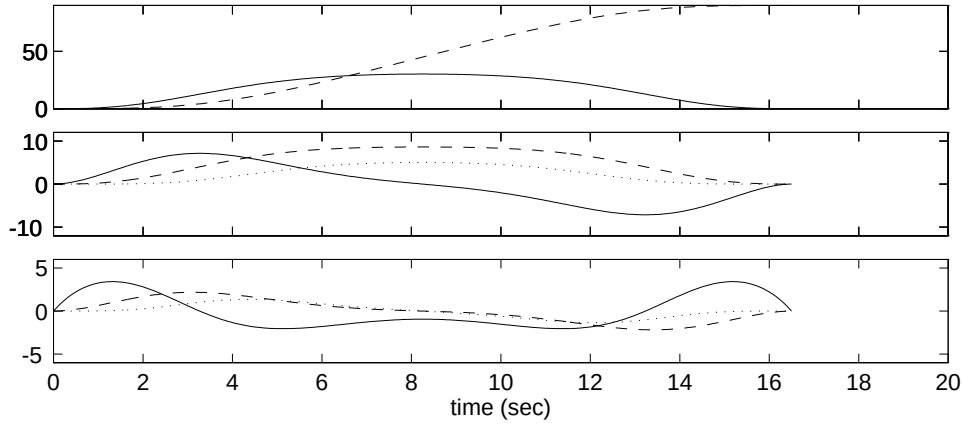

Figure 26. Cartesian maneuver: α_{tr} (deg), ω_{trt} (deg/sec), $\dot{\omega}_{trt}$ (deg/sec^2).

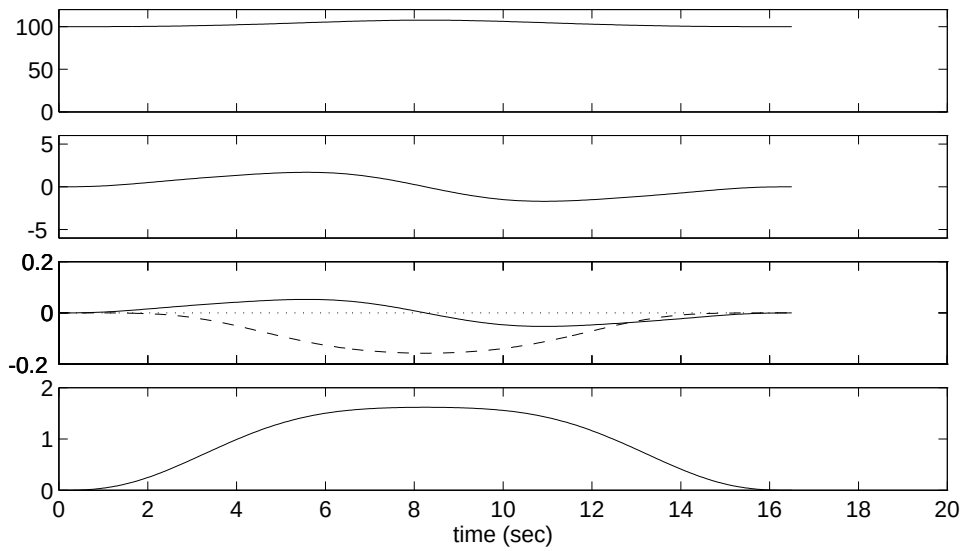

Figure 27. Cartesian maneuver: v (ft/sec), $\dot{v}$ (ft/sec^2), f_t' (g's), κ (1/kft).

and where g is the acceleration of gravity. It may be noted that the ground speed of this maneuver is not constant; there is a small acceleration into the turn and a small deceleration out of the turn (second panel). In some cases a constant speed maneuver is preferable. That may be accomplished by means of the cylindrical coordinates $(v, \psi, \dot{h})$ of velocity:

$$v_r = \begin{pmatrix} v\cos\psi \\ v\sin\psi \\ -\dot{h} \end{pmatrix} \tag{95}$$

where $h = -r_{r3}$ is the altitude. The Cartesian coordinates of position and the arc length are, of course,

$$\begin{aligned} r_r(t) &= r_r(0) + \int_0^t v_r(\sigma)d\sigma \\ s(t) &= s(0) + \int_0^t v(\sigma)d\sigma \end{aligned} \tag{96}$$

Often it is easier to generate the maneuver in two steps. First the path is "drawn" with some convenient speed, and the path parameter is changed from time to arc length. Then the path is traversed at some other speed. This may be accomplished as follows:

First draw the path, as a function of time

$$(s, \psi) = [s_1(t), \psi_1(t)],\ t \in [0, T] \tag{97}$$

where $\dot{s}_1 > 0$. Next consider s as the independent parameter:

$$(t, \psi) = [\alpha(s), \beta(s)],\ s \in [s_1(0), s_1(T)] \tag{98}$$

where $s_1[\alpha(s)] = s$, and $\beta(s) = \psi_1[\alpha(s)]$. Finally, traverse it with a different schedule

$$s = s_2(t),\ t \in [t_0, t_f] \tag{99}$$

namely

$$(s, \psi) = [s_2(t), \psi_2(t)] = \{s_2(t), \beta[s_2(t)]\},\ t \in [t_0, t_f] \tag{100}$$

where $t_0 = \alpha[s_1(0)]$ and $t_f = \alpha[s_1(T)]$. The problem is to raise this zero-order computation to order m. That is, the problem is to construct

$$SF^k(\psi) = \psi_1\{\alpha[SF^k(s)]\} \tag{101}$$

Suppose that for given $s_1 = a_0$ the corresponding t_1 and $SF^k(s_1)$ are known. Then the evolution of s_1 near the base point (a_0, t_1) is given by the Taylor series in τ,

$$s_1(t_1 + \tau) = s_1 + \dot{s}_1\tau + \ldots + s_1^{(k)}\tau^k/k! \tag{102}$$

This is a polynomial of degree k in τ with constant coefficients

$$a = \left(s_1, \dot{s}_1, \ldots, s_1^{(k)}/k! \right) \tag{103}$$

This zero-order polynomial may be raised easily to order k:

$$SF^k(s_1) = poly[a, SF^k(\tau)] \tag{104}$$

The input is $SF^k(\tau_1) = (\tau, 1, 0, \ldots, 0)$ and the output is $SF^k(s_1)$. The present objective is to invert the process: given some other evolution $SF^k(s_2)$, find the corresponding $SF^k(\tau_2)$. Having found that, the new evolution of ψ is given by the polynomial

$$SF^k(\psi_2) = poly[b, SF^k(\tau_2)] \tag{105}$$

with constant coefficients

$$b = \left(\psi_1, \dot{\psi}_1, \ldots, \psi_1^{(k)}/k! \right) \tag{106}$$

The inverse $SF^k(\tau_2)$ may obtained in the standard way. (See ref. 3.)

$$\begin{aligned}
&SF^k(\tau) = (0, \ldots, 0)\\
&g = (\dot{s})^{-1}\\
&do\ i = 1, k\\
&\quad w = \{poly[a, SF^k(\tau)]\}_k\\
&\quad \tau^{(k)} = g(s^{(k)} - w)\\
&end\ do
\end{aligned}$$

In the special but useful case when both the initial drawing and final traverse are done at constant but possibly different speeds, $\dot{s}_1 = v_1$ and $\dot{s}_2 = v_2 = \mu v_1$, respectively, simple scaling suffices:

$$SF^k(\psi_2) = \left(\psi_1, \mu\dot{\psi}_1, \ldots, \mu^k\psi^{(k)} \right) \tag{107}$$

For example, consider the following abbreviated control card schedule.

$M_{v\psi}$: 90-degree cylindrical maneuver

$$
\begin{array}{rrrrrrrr}
T = & 0, \mu_y = CYL & & & & & & \\
 & SF^4(s) & f & 0 & 100 & 0 & 0 & 0 \\
 & SF^4(\psi) & f & 0 & 0 & 0 & 0 & 0 \\
 & SF^4(h) & f & 100 & 0 & 0 & 0 & 0 \\
T = & 5, \mu_y = CYL & & & & & & \\
 & SF^4(s) & f & 0 & 100 & 0 & 0 & 0 \quad 14 \\
 & SF^4(\psi) & f & 0 & 6 & 0 & 0 & 0 \quad 14 \\
 & SF^4(h) & f & 100 & 0 & 0 & 0 & 0 \quad 04 \\
T = & 10, \mu_y = CYL & & & & & & \\
 & SF^4(s) & f & 0 & 100 & 0 & 0 & 0 \quad 14 \\
 & SF^4(\psi) & f & 0 & 6 & 0 & 0 & 0 \quad 14 \\
 & SF^4(h) & f & 100 & 0 & 0 & 0 & 0 \quad 04 \\
T = & 5, \mu_y = CYL & & & & & & \\
 & SF^4(s) & f & 0 & 100 & 0 & 0 & 0 \quad 14 \\
 & SF^4(\psi) & f & 90 & 0 & 0 & 0 & 0 \quad 14 \\
 & SF^4(h) & f & 100 & 0 & 0 & 0 & 0 \quad 04
\end{array}
$$

The speed is held constant throughout the maneuver at $v = 100$ ft/sec. At the end of the third segment, the displacement $r_r(20) - r_r(0) = (1208, 1208, 0)^T$. A displacement of $(1000, 1000, 0)$ may be obtained by scaling time by a factor of $1000/1208$. The resulting path is shown solid in figure 28, and it may be compared with the Cartesian path (dotted). The behaviors of the attitude, angular velocity, and

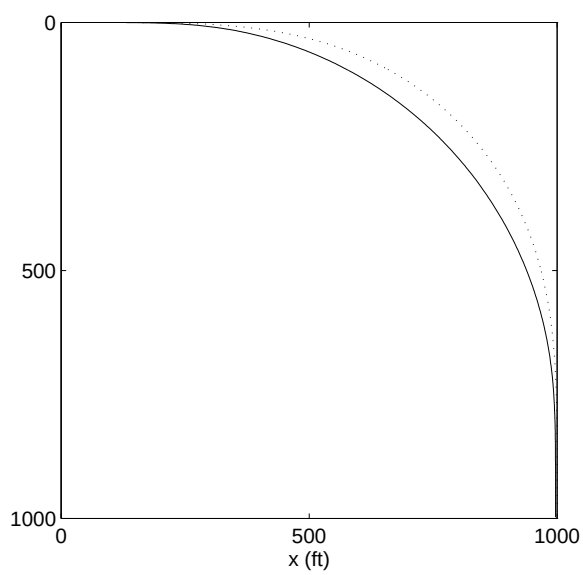

Figure 28. Cylindrical maneuver: path.

acceleration are shown in figure 29. Figure 30 shows that the speed (top panel) is constant, as expected. The curvature is smaller for this case than for the Cartesian maneuver shown in figure 27.

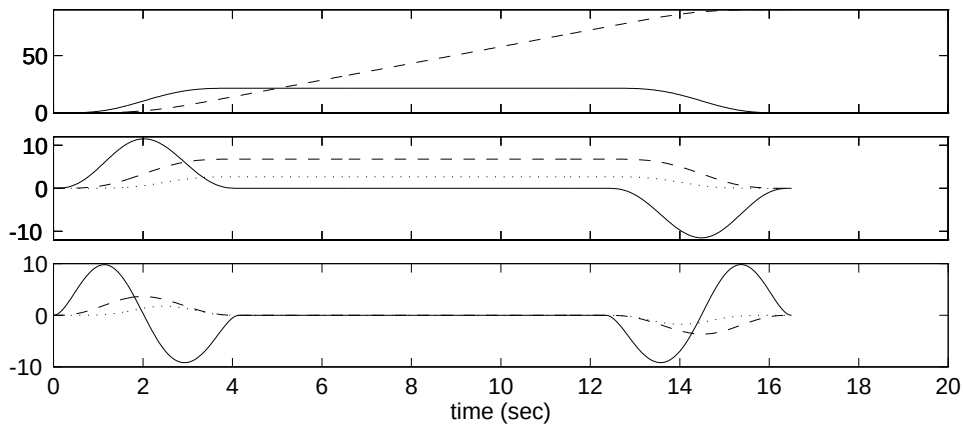

Figure 29. Cylindrical maneuver: α_{tr} (deg), ω_{trt} (deg/sec), $\dot{\omega}_{trt}$ (deg/sec^2).

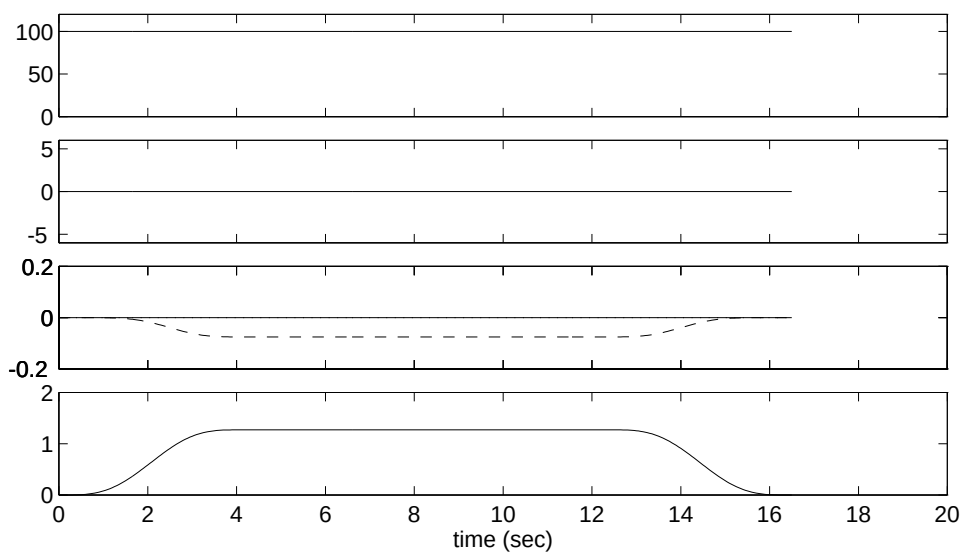

Figure 30. Cylindrical maneuver: v (ft/sec), $\dot{v}$ (ft/sec^2), f_t' (g's), κ (1/kft).

Often ground speed is of lesser interest than the airspeed. The air velocity is given by

$$\vec{v}^a = \vec{v} - \vec{w} = \dot{s}\vec{t} - \vec{w} \tag{108}$$

where $\vec{t}$ is the path tangent. Therefore, the ground speed needed to maintain a desired airspeed $v^a = a$, which may be a function of time, is given by

$$\dot{s} = \vec{t} \cdot \vec{w} + \sqrt{(\vec{t} \cdot \vec{w})^2 + a^2 - w^2} \tag{109}$$

This equation can be used iteratively to raise the order to $SF^k(s)$. Often it is desirable to point the $\overleftrightarrow{t}$ axes into the wind. This maneuver may be done by choosing $\vec{t}_1$ in the direction of the air velocity $\vec{v}^a$ rather than, as before, in the direction of the ground velocity $\vec{v}$.

Figure 31 shows the 90-degree turn in which the airspeed is held constant at $v_a = 100$ ft/sec in the presence of a constant wind $w_r = (-20, 20, 0)$ ft/sec. The maneuver is executed in the wind path axis mode $\mu_a = WPA$. It may be noted that, while the ground speed v (solid line in panel 1) varies, the airspeed v_a (dotted) is constant at the set value of 100 ft/sec. The second panel shows f'_t. It may be noted that only the normal component f'_{t3} (dashed) is active. The Euler angles α_{tr} degrees are shown in the third panel. Note that, since the axis is pointing into $v_r^a = v_r - w_r$, the yaw angle α_{tr3} (dashed) starts the turn with -11.5 degrees and finishes the turn with 78.6 degrees. Furthermore, the roll angle (solid) is not constant in the constant radius segment. The bottom panel shows the angular acceleration $\dot{\omega}_{trt}$ degrees/sec^2. Note the smaller roll acceleration at the beginning of the maneuver and the larger value at the end in comparison with the no-wind case shown in figure 29. Finally, the duration of the maneuver is a little longer, namely 17.35 seconds.

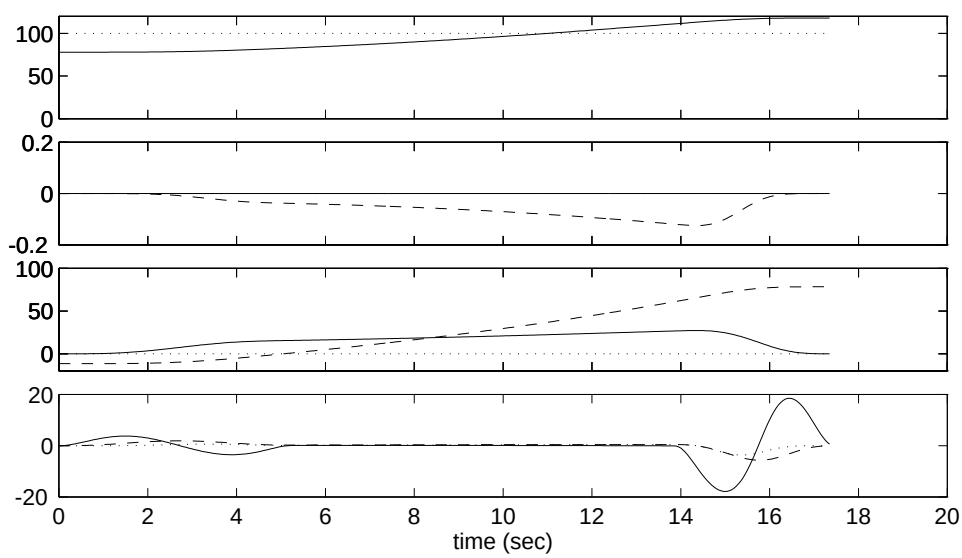

Figure 31. Fixed air speed cylindrical maneuver $(v, v_a), f'_t, \alpha_{tr}, \dot{\omega}_{trt}$.

Other possibilities are of practical interest. For example, spherical coordinates of velocity $(\mu_y = SPH)$

$$v_r = (\, v \cos\psi \cos\gamma, v \sin\psi \cos\gamma, -v \sin\gamma \,)^T \tag{110}$$

are frequently used for guiding an aircraft to the airport.

The methodology developed in the present chapter allows one to generate segments in any one of these many useful representations, and to patch the segments into a smooth, executable maneuver. Thus, the Guidance Output Trajectory Generator, which is the first block in figure 5, is realized. The input is a sequence of control cards. The output is the commanded definition and evolution of the system output y^c and the evolution of the operating mode μ. So far, the operating mode had two coordinates, namely μ_a, which defines the coordinatization of the system state space, and μ_y, which defines the state-to-output map. In the next chapter, in which the Guidance State Trajectory Generator is considered, the third coordinate, the control mode μ_c, which specifies how the maneuver is to be executed, is introduced, and an approach for constructing the guidance state and control (x^g, u^g), which produce the desired output $y^g \approx y^c$, is given.

4 DYNAMIC INVERSE

In the previous section, a procedure for converting control points into the desired motion of the output $\underline{y}_d$ is developed. (The underline denotes a function of time.) In this section, an inversion procedure that converts $\underline{y}_d$ into the corresponding motions of the state and control $(\underline{x}^g, \underline{u}^g)$ is developed. It will be useful to clearly distinguish between two types of inverses, namely the static inverse and the dynamic inverse.

For the *static* inverse problem, the following terminology is employed. A direct algorithm f is given:

$$\begin{array}{rcl} u & \in & U \subset R^m,\ p \in P \subset R^q,\ y \in Y \subset R^m \\ f & : & U \times P \to Y \\ y & = & f(u,p) \end{array} \tag{111}$$

where u, p, and y are, respectively, the control, parameter, and output. The problem is to construct the relative inverse algorithm $\hat{f}$,

$$\begin{array}{rcl} \hat{f} & : & Y \times P \to U \\ u^c & = & \hat{f}(y_d, p) \end{array} \tag{112}$$

so that, for any desired output $y_d \in Y$ and parameter $p \in P$, the output error

$$e_y = y_d - f(u^c, p) \tag{113}$$

is sufficiently close to zero.

In contrast, the *dynamic* inverse problem may be summarized as follows: Suppose that a state equation and an output map are given:

$$\begin{array}{rcl} t & \in & I_t \subset R,\ x \in X \subset R^n,\ u \in U \subset R^m,\ y \in Y \subset R^m,\ p \in P \subset R^q \\ f & : & X \times U \times P \to X \\ h & : & X \times P \to Y \\ \dot{x} & = & f(x,u,p) \\ y & = & h(x,p) \end{array} \tag{114}$$

Here, x, u, p, and y are, respectively, the state, control, parameter, and output; f is the flow, and h is the output map. In this case, the inversion problem is to construct an algorithm $\hat{T}$, which transforms any given, suitably restricted, motion of the output $\underline{y}_d$ and parameter $\underline{p}$, respectively, into the corresponding motion of the state $\underline{x}^g$ and control $\underline{u}^g$, which satisfy both the state equation and the output map. That is, the state error equation and output equation error, respectively,

$$\begin{array}{rcl} e_x(t) & = & \dot{x}^g(t) - f[x^g(t), u^g(t), p(t)] \\ e_y(t) & = & y_d(t) - h[x^g(t), p(t)] \end{array} \tag{115}$$

must be sufficiently close to zero on the interval of interest I_t.

In the case of linear systems, the dynamic inverse is given by the particular solution. For example, consider a linear, time invariant, scalar, n^{th}-order system,

$$\begin{array}{rcl} x & \in & R^n,\ u \in R,\ y \in R \\ \dot{x} & = & Ax + bu \\ y & = & cx \end{array} \tag{116}$$

As is well known, if the system is controllable and observable, then there exists a transformation of state and control coordinates

$$\begin{array}{rcl} z & = & Tx \\ v & = & wu + rx \end{array} \tag{117}$$

with nonsingular T and w, which turns the state equation into a simple string of integrators (Brunowsky form with the Kronecker index $\kappa = n$)

$$\begin{array}{rcl} \dot{z}_k & = & z_{k+1},\ 1 \le k \le n-1 \\ \dot{z}_n & = & v \end{array} \tag{118}$$

For the system considered here, the rows of T and the row vector r are given by

$$\begin{array}{rcl} t_i & = & t_1 A^{i-1},\ 2 \le i \le n \\ r & = & t_1 A^n \end{array} \tag{119}$$

while the first row of T is found from the n equations

$$\begin{array}{rcl} t_1 A^{i-1} b & = & 0,\ 1 \le i \le n-1 \\ t_1 A^{n-1} b & = & w \end{array} \tag{120}$$

Then, for $cT^{-1} = (h_0, \ldots, h_{n-1})$, the output is given by

$$y = \sum_{k=0}^{n-1} h_k z_1^{(k)} \tag{121}$$

But for a given $\underline{y}_d$, that is a driven differential equation

$$\sum_{n-1}^{0} h_k z_1^{(k)} = y_d(t) \tag{122}$$

This equation is referred to as the driven zero dynamics of system (116). Its general solution is given by the sum of the particular solution z_{1p} and the homogeneous solution z_{1h}. Rather than the solution of the usual initial value problem, the "steady-state" (bounded-input-bounded-output) solution of this equation, which is just z_{1p}, is desired. Thus, if the desired output is

$$y_d(t) = \sin(\omega t),\ t \in R \tag{123}$$

then, of course, the particular solution is given by

$$z_{1p}(t) = A \sin(\omega t + \phi),\ t \in R \tag{124}$$

where A and ϕ are the gain and phase of the transfer function,

$$Ae^{j\phi} = g(j\omega) = h(j\omega)^{-1} \tag{125}$$

The roots of $h(\lambda) = 0$ are assumed to be far enough from the $j\omega$-axis: $|Re(\lambda)| > \sigma$ for some given $\sigma > 0$.

The particular solution for the more general case of the forcing term, $\underline{y}_d$, may be solved by means of a convolution. Let $g(t)$ denote the Fourier inverse transform of the transfer function, $g(j\omega)$. Then

$$z_{1p}(t) = \int_{-\infty}^{\infty} g(t-\tau) y_d(\tau) d\tau \tag{126}$$

In practice, the integral may be approximated by

$$z_{1p}(t) = \int_{-a}^{a} g(t-\tau) y_d(\tau) d\tau \tag{127}$$

where a is only several times the longest time constant $1/\sigma$ of g.

The corresponding state and control $[z_p(t), v_p(t)]$ are given by the appropriate derivatives of $z_{1p}(t)$, and the solution to the inverse problem is given by

$$\begin{array}{rcl} x^g(t) & = & T^{-1} z_p(t) \\ u^g(t) & = & w^{-1}[v_p(t) - rT^{-1} z_p(t)] \end{array} \tag{128}$$

Note that, since the homogeneous solution is chosen to be zero, it does not matter whether the differential equation (122) is stable or not, that is, whether the zeros dynamics are or are not minimum phase. On the other hand, the computation may be noncausal with respect to $\underline{y}_d$, requiring a "look-ahead" at t of a units of time:

$$y_d(t+\tau), \ -a \le \tau \le a \tag{129}$$

The computation of the particular solution for the multiaxis case ($u, y \in R^m$) is the same if the Brunowsky form is rectangular, that is, if all m strings of integrals are of the same length $\kappa = n/m$. Otherwise, the computation is similar but a bit more complicated, requiring some additional bookkeeping.

In many cases, it is not necessary to obtain a particular solution that satisfies equation (122) because often the zero dynamics are caused by small, essentially parasitic, effects so that the zeros of equation (122) are far away from the spectrum of the forcing term y_d. In this report, the cases where the h_0 term is nonsingular and dominant are of interest, while the rest of the terms represent a small, parasitic perturbation. This distinction is made explicit by means of the parameter ϵ:

$$h(D, \epsilon) = \epsilon(h_{n-1} D^{n-1} + \ldots + h_1 D^1) + h_0 \tag{130}$$

For the case of $\epsilon = 0$, the particular solution is just

$$z_{1p}(t) = h_0^{-1} y_d(t) \tag{131}$$

Then, in view of equation (118), and because h_0 is constant, (z_p, v_p) may be obtained by differentiation of the desired output y_d:

$$\begin{array}{rcl} z_{kp}(t) & = & h_0^{-1} y_d^{(k-1)}(t), \ 1 \le k \le n-1 \\ v_p(t) & = & h_0^{-1} y_d^{(n)}(t) \end{array} \tag{132}$$

The corresponding evolution of the natural state and control coordinates (x^0, u^0) is obtained by inverting equations (117); that is,

$$\begin{array}{rcl} x_p^0(t) & = & T^{-1} z_p(t) \\ u_p^0(t) & = & w^{-1}\left[v_p(t) - r x_p^0(t)\right] \end{array} \tag{133}$$

This solution is referred to as the *pure feedback approximation*. Suppose that

$$\begin{aligned} x^g(t) &= x^0(t) \\ u^g(t) &= u^0(t) \end{aligned} \tag{134}$$

Then the state equation and output errors are given by the linear version of (115)

$$\begin{aligned} e_x(t) &= \dot{x}^g(t) - [Ax^g(t) + bu^g(t)] \\ e_y(t) &= y_d(t) - cx^g(t) \end{aligned} \tag{135}$$

In the absence of zero dynamics, that is when $h_k = 0$ for $k > 0$, the pure feedback approximation is exact in that it satisfies both the state equation ($e_x = 0$) and the output ($e_y = 0$). Otherwise, there will be errors in either one or the other or both equations. Often in practice these errors are completely negligible. However, when the zeros are closer in and the desired maneuver is relatively fast, the errors due to the neglected zero dynamics may become significant. In that case, the zero dynamics must be accounted for.

This approach may be extended to nonlinear, time-varying, multiaxis systems as described in references 7–10. As in the linear case, it is useful to introduce ϵ into the state equation and the output map

$$\begin{aligned} \dot{x} &= f(x, u, \epsilon) \\ y &= h(x, \epsilon) \end{aligned} \tag{136}$$

so that the zero dynamics, if any, may be suppressed from the model by setting $\epsilon = 0$. The pure feedback solution $(\underline{x}^o, \underline{u}^o)$ obtained for the model with $\epsilon = 0$ may be taken as the first approximation of the inverse. The errors due to the neglected dynamics are given by

$$\begin{aligned} e_x^o(t) &= \dot{x}^o(t) - f[x^o(t), u^o(t), 1] \\ e_y^o(t) &= y_d(t) - h[x^o(t), 1] \end{aligned} \tag{137}$$

An iterative (Picard) procedure for reducing these errors is described in reference 9 and applied in reference 10. The procedure is based on the convolution (127). In the present report the computation of the pure feedback approximation, and controlling the resulting errors, is of interest.

A very easy and practical way to satisfy the state equation so that $e_x = 0$ follows: Consider the servo structure shown in figure 32. The left block represents the approximate inverse $\hat{T}^0$ such as the pure feedback algorithm in which the zero dynamics are suppressed $(\epsilon = 0)$, or the pure feedback approximation corrected by only one Picard step. The right block represents the simulation of the exact model ($\epsilon = 1$). The input to this block is the control u^g; the output is the state x^g. So, by construction, $e_x^g = 0$. The tracking error e_y^g is regulated by the middle block. That is, a regulator law

$$u^{gr} = k(x^o, x^g) \tag{138}$$

is designed and the loop is closed:

$$u^g = u^o + u^{gr} \tag{139}$$

Then it may be expected that, for a reasonable regulator, sufficiently weak zero dynamics, and suitably restricted class of inputs $\underline{y}_d$, the tracking error will be small.

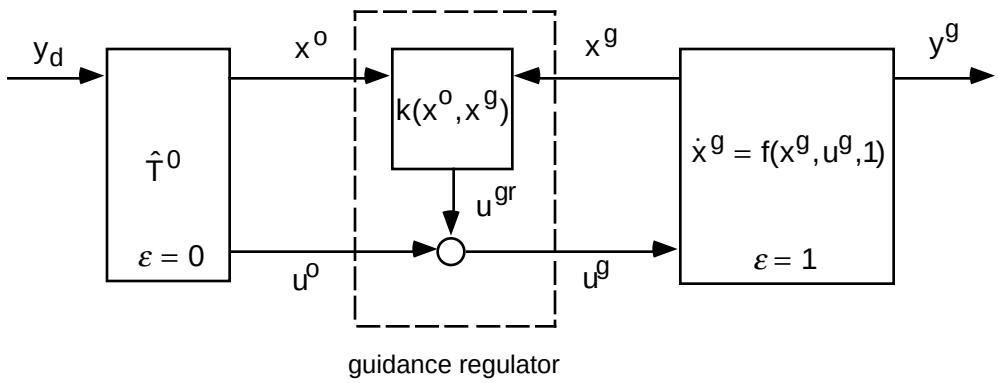

Figure 32. Servo structure satisfying the state equation.

Next, the key problem of computing the pure feedback approximation is considered.

4.1 Dynamic Inverse of Pure Feedback Systems

In the preceding section, the dynamic inverse of a linear system (116) was considered. The key steps in the construction of the pure feedback approximation were the static inversion, as in equation (131), and the differentiation of that inverse, as in equation (132). The computations may be generalized to nonlinear cases as follows: Consider again a function

$$\begin{array}{rcl} x & \in & X \subset R^m,\ p \in P \subset R^q,\ y \in Y \subset R^m \\ f & : & X \times P \to Y \\ y & = & f(x,p) \end{array} \tag{140}$$

and suppose that this (zero-order) algorithm f has been raised to arbitrary order k, which is a key step in what follows. The corresponding algorithm is denoted as follows:

$$VF^k(y) = f[VF^k(x), VF^k(p)] \tag{141}$$

That is, equation (141) is an algorithm whose input consists of x and k of its time derivatives, namely the vector form

$$VF^k(x) = \left(x, \dot{x}, \ldots, x^{(k)} \right) \tag{142}$$

and similarly for the parameter p. The output of the algorithm is y and k of its derivatives

$$VF^k(y) = [y, \dot{y}, \ldots, y^{(k)}] \tag{143}$$

Inversion of this algorithm is desired, so that for given evolution of the output $VF^k(y)$ and parameter $VF^k(p)$ the new algorithm produces the required input $VF^k(x)$.

Now, suppose that for f in equation (140), the Jacobian matrix f_x is nonsingular in the region of interest, and that the inverse algorithm $\hat{f}$

$$\begin{array}{rcl} \hat{f} & : & Y \times P \to X \\ x^c & = & \hat{f}(y_d, p) \\ e_y & = & y_d - f(x^c, p) = 0 \end{array} \tag{144}$$

is available. This zero-order inverse may be raised to order k, as follows. For $k = 0$, the zero-order inverse (144) computes the base point:

$$x = \hat{f}(y, p) \tag{145}$$

At that base point, the first derivative of y is given by

$$\dot{y} = f_x \dot{x} + f_p \dot{p} = \{f[(x, 0), (p, \dot{p})]\}_1 + f_x \dot{x} \tag{146}$$

where $(x, 0)$ denotes the vector form of order one in which the first derivative is zero. The subscript 1 denotes the first derivative of the variable, namely,

$$\{VF^k(y)\}_1 = \dot{y} \tag{147}$$

and it is evaluated by the algorithm given in equation (141), the vector form of $f(x, p)$. Similarly, for the second derivative

$$\ddot{y} = \dot{f}_x \dot{x} + f_x \ddot{x} + \dot{f}_p \dot{p} + f_p \ddot{p} = \{f[(x, \dot{x}, 0), (p, \dot{p}, \ddot{p})]\}_2 + f_x \ddot{x} \tag{148}$$

and so on for higher derivatives. Hence, the following algorithm for passing derivatives through an inverse:

$$\begin{array}{l} x = \hat{f}(y, p) \\ VF^k(x) = (x, 0, \ldots, 0) \\ do\ j = 1, k \\ \quad x^{(j)} = f_x^{-1} \left[y^{(j)} - \{f[VF^j(x), VF^j(p)]\}_j \right] \\ end\ do \end{array} \tag{149}$$

It should be noted that, remarkably, only the base point x and the Jacobian f_x at the base point are needed to generate the derivatives of x from the derivatives of y and p.

Next, consider the construction of a dynamic inverse as in equation (114) for the special case of pure feedback systems. Later in the present report it will be shown that, for the purposes of guidance, the following model may be used to approximate many aircraft types. (See figure 36.) In this model, the structure of the multiaxis state, control, and parameter is given by

$$x_1, x_2 \in R^{n_1},\ x_3, x_4, u \in R^{n_2},\ p \in R^{n_p} \tag{150}$$

with $n_2 \geq n_1$. The state equation is of the following form with sufficiently smooth flows f_i.

$$\begin{array}{l} \dot{x}_1 = f_1(x_1, \quad x_2, \quad 0, \quad 0, \quad 0, \quad p) \\ \dot{x}_2 = f_2(x_1, \quad x_2, \quad x_3, \quad 0, \quad 0, \quad p) \\ \dot{x}_3 = f_3(x_1, \quad x_2, \quad x_3, \quad x_4, \quad 0, \quad p) \\ \dot{x}_4 = f_4(x_1, \quad x_2, \quad x_3, \quad x_4, \quad u, \quad p) \end{array} \tag{151}$$

The structure of the multiaxis output is given by

$$y_1 \in R^{n_1},\ y_2 \in R^{n_2 - n_1} \tag{152}$$

with the sufficiently smooth output map

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} h_1(x_1, \quad 0, \quad 0, \quad 0, \quad 0, \quad p) \\ h_2(x_1, \quad x_2, \quad x_3, \quad 0, \quad 0, \quad p) \end{pmatrix} \tag{153}$$

The block diagram of the model is shown in the upper part of figure 33. The heavy lines are n_2–dimensional. Note that, disregarding the parameter p, there are no lines bypassing integrators in the forward direction. Hence there are no zeros. There is a new (canonical) set of coordinates of state and control related to the natural state and control (x, u) by the nonsingular transformation T, in which the system takes the particularly simple (Brunowsky) form shown in the bottom of the figure. There is a bundle of n_1 strings of integrators, each four integrators long, and another $n_2 - n_1$ dimensional bundle

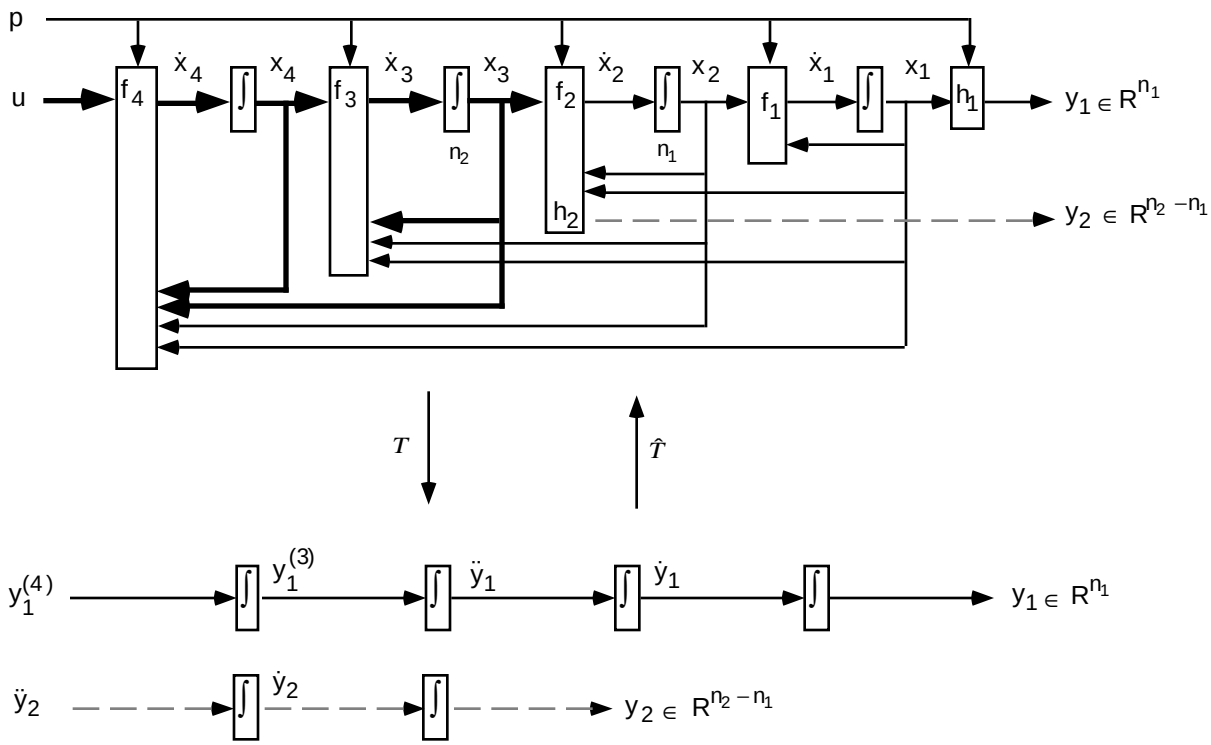

Figure 33. Pure feedback model of aircraft.

of strings, each of length two. The inverse transformation $\hat{T}$ converts the Brunowsky form back into the natural representation. The construction of the dynamic inverse is equivalent to the construction of $\hat{T}$: the evolution of the output $[VF^4(y_1), VF^2(y_2)]$ and parameter $VF^4(p)$ is transformed into the corresponding natural state and control (x, u). Both the inverse $\hat{T}$ and the direct T transformations are constructed as follows:

For the construction of $\hat{T}$, assume that $VF^k(y_1)$ and $VF^{k-2}(y_2)$ are given and that all the functions and their inverses have been raised to sufficient order, as discussed previously. Then from the inverse $\hat{h}_1$ of h_1, the derivatives of x_1 are:

$$VF^k(x_1) = \hat{h}_1[VF^k(y_1), 0, 0, 0, 0, VF^k(p)] \tag{154}$$

and consequently $k-1$ derivatives of $\dot{x}_1$, namely $VF^{k-1}(\dot{x}_1)$. Then, inverting the first line of the state equation (151) gives the derivatives of x_2:

$$VF^{k-1}(x_2) = \hat{f}_1\left[VF^{k-1}(x_1), VF^{k-1}(\dot{x}_1), 0, 0, 0, VF^{k-1}(p)\right] \tag{155}$$

and $VF^{k-2}(\dot{x}_2)$. Next, since there are n_1 equations from f_2 and $n_2 - n_1$ equations from h_2, the determination of the n_2 elements of x_3 requires the simultaneous inverse of (f_2, h_2) with respect to x_3 to produce the derivatives of x_3:

$$VF^{k-2}(x_3) = (\widehat{f_2, h_2}) \left[VF^{k-2}(x_1), VF^{k-2}(x_2), VF^{k-2}(\dot{x}_2), 0, 0, VF^{k-2}(p)\right] \tag{156}$$

and $VF^{k-3}(\dot{x}_3)$. Similarly for x_4 and u:

$$\begin{aligned} VF^{k-3}(x_4) &= \hat{f}_3[VF^{k-3}(x_1), VF^{k-3}(x_2), VF^{k-3}(x_3), VF^{k-3}(\dot{x}_3), 0, VF^{k-3}(p)] \\ VF^{k-4}(u) &= \hat{f}_4[VF^{k-4}(x_1), VF^{k-4}(x_2), VF^{k-4}(x_2), VF^{k-4}(x_3), VF^{k-4}(\dot{x}_4), VF^{k-4}(p)] \end{aligned} \tag{157}$$

At this point, the function

$$\hat{\mathcal{T}} : \ (R^{n_1})^{k+1} \times (R^{n_2-n_1})^{k-1} \times (R^{n_p})^{k+1} \rightarrow (R^{n_1})^{k} \times (R^{n_2})^{k-1} \times (R^{n_2})^{k-2} \times (R^{n_2})^{k-3} \times (R^{n_1})^{k-4} \tag{158}$$

which maps the desired output and some of its derivatives into the corresponding state and control, and some of their derivatives has been constructed:

$$\hat{\mathcal{T}} \left[VF^{k}(y_1), VF^{k-2}(y_2), VF^{k}(p)\right] = \left[VF^{k}(x_1), VF^{k-1}(x_2), VF^{k-2}(x_3), VF^{k-3}(x_4), VF^{k-4}(u)\right] \tag{159}$$

The multiaxis dynamic inverse algorithm $\hat{T}$ with $k = 4$ that is desired is contained in $\hat{\mathcal{T}}$. The state and control (x, u) are given by the zero-order entries of the output of $\hat{\mathcal{T}}$, namely,

$$(\, x, u \,) = \hat{T}[VF^4(y_1), VF^2(y_2), VF^4(p)] = \left\{\hat{\mathcal{T}} \left[VF^4(y_1), VF^2(y_2), VF^4(p)\right]\right\}_0 \tag{160}$$

It is also very useful to be able to go the other way, that is, to have the direct dynamic map T which transforms the state and control (x, u) into the Brunowsky canonical coordinates $VF^4(y_1)$ and $VF^2(y_2)$. This algorithm may be constructed as follows: From the given state and control $[VF^0(x), VF^0(u)]$, parameter $VF^4(p)$, and the state equation is obtained

$$VF^1(x_i) = f_i[VF^0(x_1), VF^0(x_2), VF^0(x_3), VF^0(x_4), VF^0(u), VF^0(p)], \ i = 1, 4 \tag{161}$$

then

$$VF^2(x_i) = f_i[VF^1(x_1), VF^1(x_2), VF^1(x_3), VF^1(x_4), 0, VF^1(p)], \ i = 1, 3 \tag{162}$$

then

$$VF^3(x_i) = f_i[VF^2(x_1), VF^2(x_2), VF^2(x_3), 0, 0, VF^2(p)], \ i = 1, 2 \tag{163}$$

and finally,

$$VF^4(x_1) = f_1[VF^3(x_1), VF^3(x_2), 0, 0, 0, VF^3(p)] \tag{164}$$

The outputs are given by

$$\begin{aligned} VF^4(y_1) &= h_1 \left[VF^4(x_1), 0, 0, 0, 0, VF^4(p)\right] \\ VF^2(y_2) &= h_2 \left[VF^2(x_1), VF^2(x_2), VF^2(x_3), 0, 0, VF^2(p)\right] \end{aligned} \tag{165}$$

Thus, the function

$$\mathcal{T}: \ (R^{n_1})^2 \times (R^{n_2})^3 \times (R^{n_p})^4 \rightarrow (R^{n_1})^4 \times (R^{n_2-n_1})^2 \times (R^{n_1})^4 \times (R^{n_1})^3 \times (R^{n_2})^2 \times R^{n_2} \quad (166)$$

which transforms the natural state and control (x, u) and the evolution of the parameter $VF^4(p)$ into the output and state rate and some of their derivatives, has been constructed.

$$\mathcal{T}\,(\,x, u, VF^4(p)\,) = (\,VF^4(y_1), VF^2(y_2), VF^3(\dot{x}_1), VF^2(\dot{x}_2), VF^1(\dot{x}_3), VF^0(\dot{x}_4)\,) \quad (167)$$

The transformation T from the natural state and control (x, u) to the canonical control is just the projection $P_{1,2}$ of $\mathcal{T}$ on the first two slots of its output;

$$\left[VF^4(y_1), VF^2(y_2)\right] = T\left[x, u, VF^4(p)\right] = P_{1,2}\mathcal{T}\left[x, u, VF^4(p)\right] \quad (168)$$

It should be noted that the time derivatives of the outputs y_1 and y_2 are obtained without differentiating either the state x or the control u. Both the forward map T and the inverse $\hat{T}$ differentiate known functions; neither differentiates signals.

The preceding discussion may be summarized by means of the block diagram in figure 34, as follows. The system $\dot{x} = f(x, u, p)$, which appears multiaxis, nonlinear, and time-dependent in its natural coordinates (x, u), becomes a simple collection of decoupled strings of integrators when viewed through the coordinate transformations $\hat{T}$ and T.

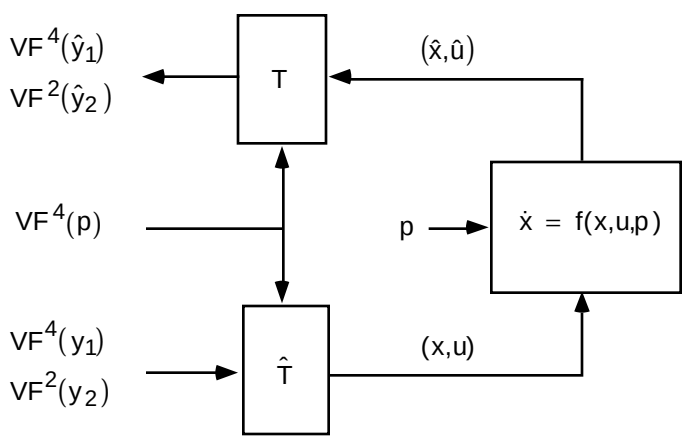

Figure 34. Linearization of pure feedback systems.

In the following discussion an aircraft model that will be considered basic for the purposes of guidance is developed. Next the notion of aircraft control modes is introduced. Then a procedure for computing the pure feedback solution (x^o, u^o) for any of the many possible control modes is presented. Finally, a general structure of the regulator is developed. The discussion is illustrated by applications to the Harrier aircraft. It should be noted that the procedure is general and applies to many types of aircraft and helicopters.

4.2 Aircraft Model for Guidance Purposes

The model developed in the second chapter is summarized and extended as follows. As before, let the runway-fixed axes and body-fixed axes be denoted by $\overleftrightarrow{r}$ and $\overleftrightarrow{b}$, respectively. Let r_r and v_r denote the runway coordinates of the aircraft position and velocity relative to the runway, respectively. Let the attitude of the aircraft relative to the runway be denoted by the direction cosine matrix C_{br}, and let the body coordinates of the angular velocity of $\overleftrightarrow{b}$ relative to $\overleftrightarrow{r}$ be denoted by ω_{brb}. The structure of the rigid-body state is chosen to be

$$\begin{array}{rcl} x_\omega & \in & X_\omega \subset (R^3)^2 \times \left[SO(3) \times R^3\right] \\ x_\omega & = & (\, r_r, v_r, C_{br}, \omega_{brb}\,) \end{array} \tag{169}$$

The structure of the control is

$$\begin{array}{rcl} u & \in & U \subset (R^3)^3 \\ u & = & (\, u_m, u_p, u_f\,) \end{array} \tag{170}$$

where u_m, u_p, and u_f are the moment control, power control, and configuration control. In addition, there are the parameters with the following structure and interpretation.

$$\begin{array}{rcl} p & \in & P \subset R^3 \times R \times R \\ p & = & (\, w_r, W, \theta_T\,) \end{array} \tag{171}$$

where w_r represents the wind in runway coordinates, W is the aircraft weight, and θ_T is the temperature of the air. The runway and body coordinates of the relative air velocity are, respectively,

$$\begin{array}{rcl} v_r^a & = & v_r - w_r \\ v_b^a & = & C_{br} v_r^a \end{array} \tag{172}$$

The input to the force and moment algorithm has the following structure and interpretation

$$\begin{array}{rcl} x_b & \in & X_b \subset U \times X_\omega \times P \\ x_b & = & (\, u, x_\omega, p\,) = (\, u_m, u_p, u_f, r_r, v_r, C_{br}, \omega_{brb}, w_r, W, \theta_T\,) \end{array} \tag{173}$$

The output will be expanded to include several other variables of interest:

$$\begin{array}{rcl} y_b & \in & Y_b = (R^3)^5 \\ y_b & = & (\, m_b, m_r, f_b, a_r, v_b^a\,) \end{array} \tag{174}$$

In the interpretation, the entries are body and runway coordinates of the generated specific moment, body coordinates of the specific force, runway coordinates of acceleration, and the body coordinates of the air velocity. Note that the coordinates of y_b are not independent. Ultimately, an independent set will be associated with each operation mode.

The algorithm specific to a particular aircraft is the function

$$\begin{array}{rcl} F_b & : & X_b \to Y_b \\ y_b & = & F_b(x_b) \end{array} \tag{175}$$

On occasion the three-dimensional (3-D) components of the algorithm F_b,

$$F_b = \left(F_b^{mb}, F_b^{mr}, F_b^{fb}, F_b^{ar}, F_b^{vb} \right) \tag{176}$$

will be needed. Next a pair of integrators are added in front of power and configuration controls (u_p, u_f) in order to enforce continuity in these controls and their rates. The result is the aircraft model shown in figure 35.

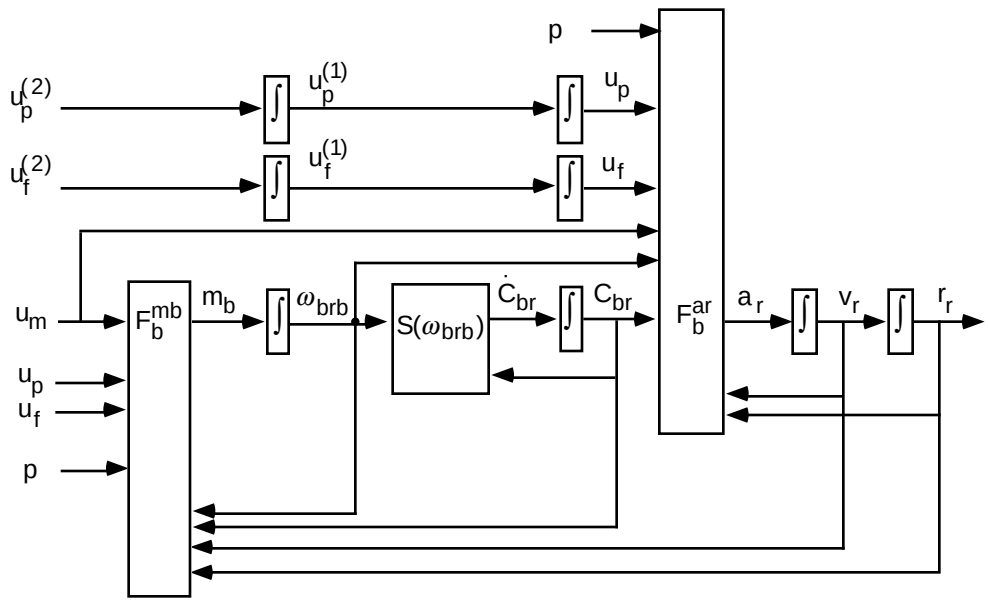

Figure 35. Block diagram of the basic aircraft model.

The model is quite an accurate and general representation for the purposes of designing aircraft guidance systems. The representation of attitude by means of the direction cosine matrix C_{br} is not limited to small angles. On the other hand, the presence of the direction cosine matrix makes the state space non-Euclidean. Furthermore, for certain aircraft maneuvers, and in certain patches of the state space, it is necessary to represent the aircraft attitude in terms of Euler angles in some given sequence. Now such state space coordinate changes are considered.

4.3 Intermediate Axis System

The usual procedure is followed; an intermediate axis system, denoted here by $\overleftrightarrow{t}$, is introduced, and the aircraft attitude, C_{br}, is factored into two rotations,

$$C_{br} = C_{bt} C_{tr} \tag{177}$$

where C_{tr} may be a large-angle rotation, but C_{bt} is small enough so that the following Euler expansion may be always assumed to be nonsingular:

$$C_{bt} = E_2(\alpha_{bt1}) E_3(\alpha_{bt2}) E_1(\alpha_{bt3}) \tag{178}$$

Several examples of parameterizations of C_{tr} have been given in section 3. There the mode variable μ_α was introduced and the horizontal attitude mode HAT, vertical attitude mode VAT, runway path axes mode RPA, and wind path axes mode WPA were discussed.

With respect to the small-angle rotation C_{bt}, the angles α_{bt1}, α_{bt2}, and α_{bt3} are pitch, yaw, and roll angles of the $\overleftrightarrow{b}$-axes relative to the $\overleftrightarrow{t}$-axes. When $\overleftrightarrow{t}$ and $\overleftrightarrow{b}$ are selected to be the wind axes and body axes, respectively, these angles become the angle of attack, negative of the sideslip angle, and the roll angle of $\overleftrightarrow{b}$.

Whatever the interpretation of the two factors of C_{br}, the force and moment generators are needed in terms of the Euler angles α_{bt} of the small rotation C_{bt} instead of the composite rotation C_{br}.

The Euler angles α_{bt} and direction cosine matrices C_{br} and C_{tr} as well as their derivatives to order k are easily related to each other by means of the dynamic forms. The angle form

$$AF_{bt}^k = \left(\alpha_{bt}, \dot{\alpha}_{bt}, \ldots, \alpha_{bt}^{(k)} \right) \tag{179}$$

and the rotation form

$$RF_{tr}^k = \left(C_{tr}, \omega_{trt}, \ldots, \omega_{trt}^{(k-1)} \right) \tag{180}$$

may be converted into RF_{br}^k by first converting the angle form into the corresponding rotation form by means of the algorithm $RF \prec AF$ and then multiplying by RF_{tr}^k:

$$RF_{br}^k = RF \prec AF_q(AF_{bt}^k) * RF_{tr}^k \tag{181}$$

Conversely, suppose that RF_{br}^k and RF_{tr}^k are given. Then the corresponding angle form is given by

$$AF_{bt}^k = RF \succ AF_q \left[RF_{br}^k * RF_{rt}^k \right] \tag{182}$$

where RF_{rt}^k is the rotation form for $C_{rt} = C_{tr}^T$, the transpose of C_{tr}.

Consider now a new representation of aircraft rigid-body state.

$$\begin{aligned} x_\alpha &\in X_\alpha \subset (R^3)^4 \\ x_\alpha &= \left[r_r, v_r, \alpha_{bt}, \alpha_{bt}^{(1)} \right] \end{aligned} \tag{183}$$

which is obtained from x_ω by replacing (C_{br}, ω_{brb}) by $(\alpha_{bt}, \dot{\alpha}_{bt})$. The two sets of variables are related by the special case of the algorithm described previously:

$$\begin{aligned} \left(\alpha_{bt}, \alpha_{bt}^{(1)} \right) &= RF \succ AF_q[RF_{br}^1 * RF_{rt}^1] \\ \left(C_{br}, \omega_{brb} \right) &= RF \prec AF_q(AF_{bt}^1) * RF_{tr}^1 \end{aligned} \tag{184}$$

The main advantage of the new state space X_α is that it is (locally) unconstrained, whereas the old state space X_ω has the orthogonality constraint $C_{br}C_{br}^T = I$.

It is also desirable to expand the force and moment generator output as follows.

$$\begin{array}{rcl} y_t & \in & Y_t \subset (R^3)^7 \\ y_t & = & (\, m_b, m_\alpha, m_r, f_t, f_b, a_r, v_b^a \,) \end{array} \tag{185}$$

where m_α is the second time derivative of α_{bt}, and f_t is the specific force in the $\overset{\leftrightarrow}{t}$ coordinate system, namely,

$$\begin{array}{rcl} m_\alpha & = & \ddot{\alpha} = \left\{ RF \succ AF_{231} \left[RF_{br}^2 * RF_{rt}^2 \right] \right\}_2 \\ f_t & = & C_{bt}^T f_b \end{array} \tag{186}$$

Note that, whereas m_b and m_r represent the specific moment vector, m_α does not. Also, the subscript 231 represents the Euler sequence in equation (178). The new model of the force and moment model algorithm will be denoted by F_t:

$$\begin{array}{rcl} x_t & \in & X_t \subset U \times X_\alpha \times \left[SO(3) \times (R^3)^2 \right] \times P \\ x_t & = & (\, u, x_\alpha, RF_{tr}^2, p \,) \\ F_t & : & X_t \to Y_t \\ y_t & = & F_t(x_t) \end{array} \tag{187}$$

The term in square brackets is the domain of the second-order dynamic form RF_{tr}^2 of the intermediate axis system. The (3-D) components of F_t will be grouped and denoted as follows.

$$F_t = \left(F_t^{mb}, F_t^{m\alpha}, F_t^{mr}, F_t^{ft}, F_t^{fb}, F_t^{ar}, F_t^{vb} \right) \tag{188}$$

The rigid-body state equation in the new coordinates becomes

$$\begin{array}{rcl} \dot{r}_r & = & v_r \\ \dot{v}_r & = & F_t^{ar}(u, x_\alpha, RF_{tr}^2, p) \\ \dot{\alpha}_{bt} & = & \alpha_{bt}^{(1)} \\ \dot{\alpha}_{bt}^{(1)} & = & F_t^{m\alpha}(u, x_\alpha, RF_{tr}^2, p) \end{array} \tag{189}$$

The system is shown in block diagram form in figure 36. The computations of the body direction cosine matrix C_{br}, body angular velocity, and acceleration RF_{br}^2 are also shown in the diagram.

The following list summarizes the interpretation of the variables:

$$\begin{array}{lllllllllll} u & = (u_m, & u_p, & u_f) & & & & & & & \\ x_\omega & = (r_r, & v_r, & C_{br}, & \omega_{brb}) & & & & & & \\ x_\alpha & = (r_r, & v_r, & \alpha_{bt}, & \alpha_{bt}^{(1)}) & & & & & & \\ x_b & = (u_m, & u_p, & u_f, & r_r, & v_r, & C_{br}, & \omega_{brb}, & p) & & \\ x_t & = (u_m, & u_p, & u_f, & r_r, & v_r, & \alpha_{bt}, & \alpha_{bt}^{(1)}, & C_{tr}, & \omega_{trt}, & \dot{\omega}_{trt}, \quad p) \\ y_b & = (m_b, & m_r, & f_b, & a_r, & v_b^a) & & & & & \\ y_t & = (m_b, & m_\alpha, & m_r & f_t, & f_b, & a_r, & v_b^a) & & & \end{array} \tag{190}$$

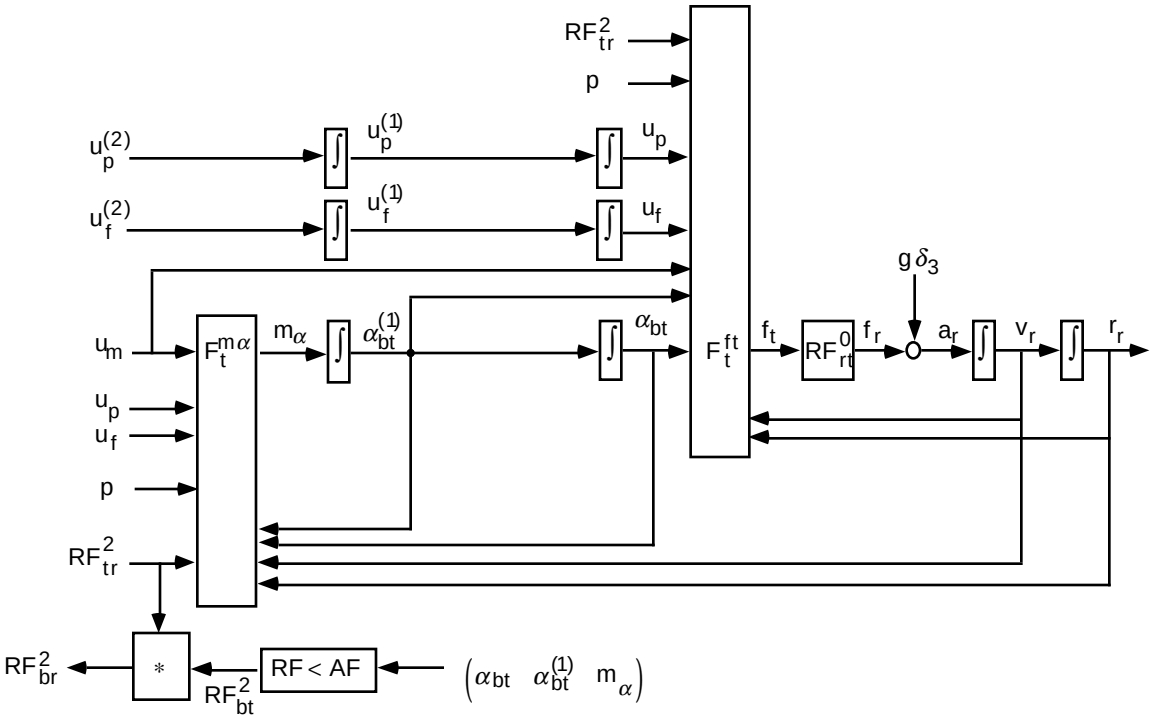

Figure 36. Alpha guidance model of the aircraft.

In a simulation, the initial conditions on the integrators in figure 36 and an evolution of all the inputs are given, and the simulation output is the evolution of all the states and various functions of the states. In a typical guidance problem, the evolution of the aircraft position r_r is given, and the problem is to fill in all the rest of the variables in the diagram. There are many ways to do that. Just which of the solutions is actually wanted and what additional information is to be provided is defined by the control mode. This mode is discussed next.

4.4 Control Modes

Now, the objective in this report is to construct pure feedback approximations of the inverses of the aircraft equations of motion. Consider, again, the model shown in figure 35. Suppose that at each instant of time t the desired evolution of aircraft position $VF^4(r_r)$ is given. Going upstream, that is to the left in the diagram in analogy with equations (155) and (156), the desired velocity $VF^3(v_r)$ and the desired acceleration $VF^2(a_r)$ are obtained. The force generator F_b^{ar} is analogous to f_2 in figure 33, except that F_b^{ar} may not be of the pure feedback type. To proceed further to the left, it is necessary to obtain a relative inverse of F_b^{ar}. But relative to what? That is defined by a new discrete variable, namely the control mode, μ_c. For example, suppose that the test aircraft, unlike the Harrier, has in addition to throttle and nozzle a side force controller, so that the power control u_p has complete, 3-D control of force. Then the aircraft rotation and translation may be decoupled within reasonable limit. That is, in addition to specifying the motion of the center of mass by $VF^4(r_r)$, the desired rotation $RF^4(C_{br})$ can also be specified. Then, assuming that the configuration $VF^2(u_f)$ and other parameters

$VF^2(p)$ are given, the inverse of F_b^{ar} relative to $VF^2(u_p)$ and the inverse of the moment generator $F_b^{m_b}$ relative to the moment control $VF^2(u_m)$ can be generated. In particular, the controls (u_m, u_p) are given by the six-dimensional static inverse of both moment and force functions. That is,

$$\begin{aligned} \hat{u}_m &= \hat{F}_b^{mb}(m_b, a_r, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) \\ \hat{u}_p &= \hat{F}_b^{ar}(m_b, a_r, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) \end{aligned} \tag{191}$$

so that

$$\begin{aligned} m_b &= F_b^{mb}(\hat{u}_m, \hat{u}_p, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) \\ a_r &= F_b^{ar}(\hat{u}_m, \hat{u}_p, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) \end{aligned} \tag{192}$$

Inverses such as those in equation (191) are sometimes referred to as trim maps. This particular zero-order trim map can be raised to any order by means of dynamic forms, as discussed previously in the algorithm (149). At this point the dynamic inversion of the model in figure 35 has been completed by obtaining $VF^2(u_m)$ and $VF^2(u_p)$. The inverse model is shown in figure 37. This control mode will be designated by the mnemonic $\mu_c = DFC$ for "direct force control."

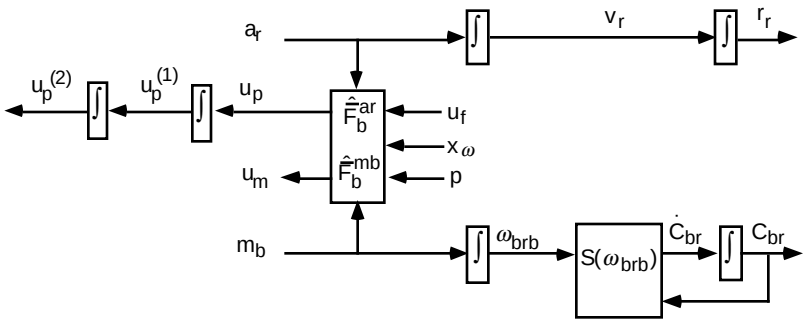

Figure 37. Direct force control (DFC) mode.

Another very useful control mode is the direction cosine control, $\mu_c = DCC$. For example, the first step in the recovery from a severe atmospheric disturbance may be to recover the aircraft attitude and, only after the attitude stabilizes, to proceed to recover the trajectory. In this mode the desired rotation is specified at each instant by the rotation form $RF^2(C_{br})$; the power and configuration controls $VF^2(u_p)$ and $VF^2(u_f)$ are driven according to a preassigned schedule. In this mode the trajectory $VF^2(r_r)$ is a dependent variable. The corresponding trim map is simply the 3-D moment trim:

$$\hat{u}_m = \hat{F}_b^{mb}(m_b, u_p, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) \tag{193}$$

so that

$$\begin{aligned} m_b &= F_b^{mb}(\hat{u}_m, u_p, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) \\ a_r &= F_b^{ar}(\hat{u}_m, u_p, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) \end{aligned} \tag{194}$$

The composite algorithm producing force subject to moment trim will be denoted with an overbar, as follows:

$$a_r = \bar{F}_b^{ar}(m_b, u_p, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) = F_b^{ar} \circ \hat{F}_b^{mb} \tag{195}$$

where the last term means that the algorithm $\hat{F}_b^{mb}$ is called first and then algorithm F_b^{ar}. The DCC control mode is useful, not only for attitude maneuvers but also for the design of other guidance laws. This mode may be interpreted as a nonsingular change of control coordinates from the natural moment control variable u_m to the new control variable m_b, namely, angular acceleration. The corresponding aircraft model is shown in figure 38. The natural moment control u_m is a dependent variable, which is

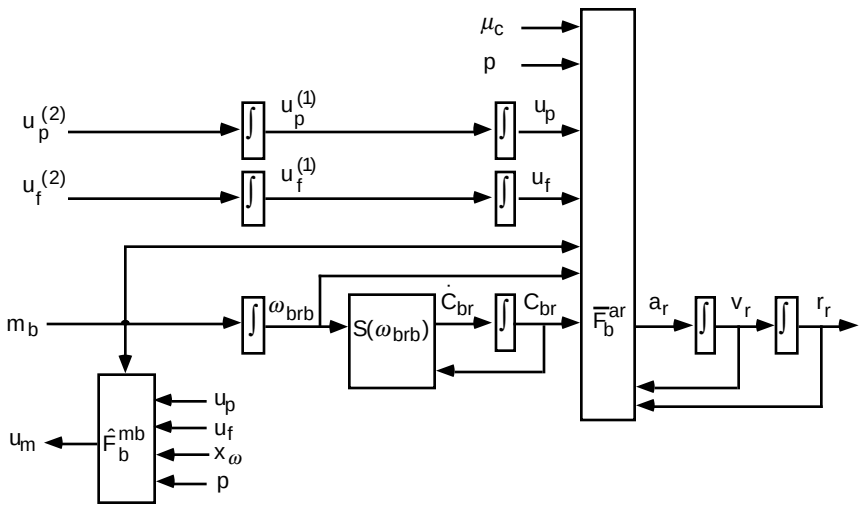

Figure 38. Direction cosine attitude control (DCC) mode.

computed by the moment trim algorithm $\hat{F}_b^{mb}$. The DCC mode provides a very convenient and realistic procedure for the suppression of zero dynamics. A pure feedback approximation of the force model may be obtained by evaluating the moment-trimmed force with $m_b = \omega_{brb} = 0$, namely,

$$a_r^0 = \bar{F}_b^{ar}(0, u_p, u_f, r_r, v_r, C_{br}, 0, p) \tag{196}$$

The error due to the suppression of the effects on a_r of u_m and ω_{brb} is

$$e_{ar} = \bar{F}_b^{ar}(m_b, u_p, u_f, r_r, v_r, C_{br}, \omega_{brb}, p) - \bar{F}_b^{ar}(0, u_p, u_f, r_r, v_r, C_{br}, 0, p) \tag{197}$$

The present report is concerned with cases for which this error is small.

Still another very useful mode is based on the alpha control mode shown in figure 39. Consider the Euler angle control (EAC) mode $\mu_c = EAC$, in which the moment function $F_t^{m\alpha}$ is inverted with respect to the moment control u_m.

$$u_m = \hat{F}_t^{m\alpha}(m_\alpha, u_p, u_f, r_r, v_r, \alpha_{bt}, \dot{\alpha}_{bt}, C_{tr}, \omega_{trt}, \dot{\omega}_{trt}, p) \tag{198}$$

The result is shown in figure 39. The composite algorithm producing the moment-trimmed force is denoted, as before, with an overbar

$$\bar{F}_t^{m_\alpha} = F_t^{ft} \circ \hat{F}_t^{m\alpha} \tag{199}$$

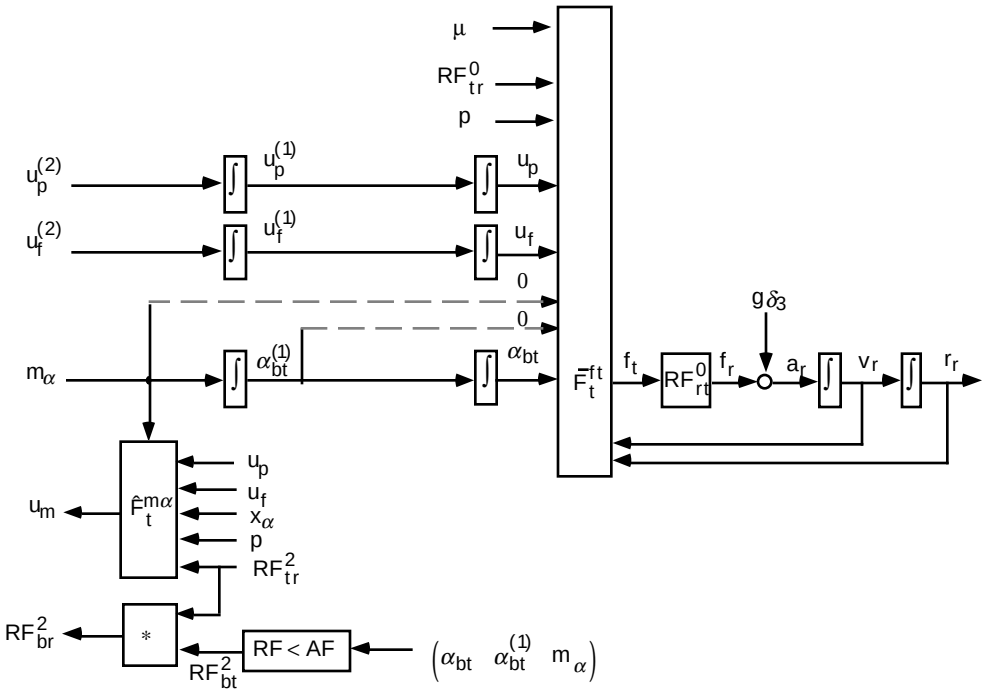

Figure 39. EAC mode.

The pure feedback approximation of the force is

$$f_t^0 = \bar{F}_t^{m_\alpha}(0, u_p, u_f, r_r, v_r, \alpha_{bt}, 0, C_{tr}, 0, 0, p) \tag{200}$$

The action of the control mode may be described in general as follows. Consider the expanded force and moment algorithm F_t, as defined in equation (188), with input x_t and output y_t,

$$y_t = F_t(x_t) \tag{201}$$

Suppose that some coordinates of y_t are labeled as independent 'i' and the remainder as dependent 'd.' Similarly, let some coordinates of x_t be labeled as active controls 'c' and the remainder as parameters 'p.' Each such labeling defines a control mode, provided that the number of independent coordinates equals the number of active control coordinates, which is the dimension of the control mode. The algorithm $\hat{F}_t$ for computing the corresponding partial inverses is given by

$$\begin{aligned} (\hat{x}_t, \hat{y}_t) &= \hat{F}_t(x_t^*, y_t^*, \mu_c) \\ \|y_t^* - F_t(\hat{x}_t)\|_{\mu_c} &\leq \varepsilon \end{aligned} \tag{202}$$

The first input x_t^* contains the initial values for the coordinates of x_t that are designated as active controls; the remainder of x_t contains the parameters p. The second input y_t^* contains the target (desired) values for the independent coordinates of y_t; the remainder of y_t containing the dependent coordinates is ignored. The third input μ_c, the control mode, designates which coordinates are which. The algorithm

fills in the locations of $\hat{x}_t$ designated as controls and locations of $\hat{y}_t$ designated as dependent. Table 4 contains several practical examples of control modes.

Table 4. Example control modes

μ_c	m_b	m_α	m_r	f_t	f_b	a_r	v_b^a	u_m	u_p	u_f	α
SIM	ddd	ddd	ddd	ddd	ddd	ddd	ddd	ppp	ppp	ppp	ppp
DCC	iii	ddd	ddd	ddd	ddd	ddd	ddd	ccc	ppp	ppp	ppp
EAC	ddd	iii	ddd	ddd	ddd	ddd	ddd	ccc	ppp	ppp	ppp
DFC	iii	ddd	ddd	ddd	ddd	iii	ddd	ccc	ccc	ppp	ppp
HAR	ddd	iii	ddd	iii	ddd	ddd	ddd	ccc	cpc	ppp	ppc
VAT	ddd	iii	ddd	iii	ddd	ddd	ddd	ccc	cpp	ppp	ccp
HEL	ddd	iii	ddd	iii	ddd	ddd	ddd	ccc	cpp	ppp	cpc
CFY	ddd	iii	ddd	iii	did	ddd	ddd	ccc	cpp	ppp	ccc
CFB	ddd	iii	ddd	iii	ddd	ddd	did	ccc	cpp	ppp	ccc
CFF	ddd	iii	ddd	iii	did	ddd	ddd	ccc	cpp	cpp	pcc
CFX	ddd	iii	ddd	iid	did	ddd	ddd	ccc	ppp	ppp	ccc
CFZ	ddd	iii	ddd	dii	did	ddd	ddd	ccc	ppp	ppp	ccc
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$

Example 1: $\mu_c = SIM$, the simulation mode. In this mode all coordinates of input x_t are parameters, and all the output y_t is dependent. There is nothing to invert. That is, SIM simply invokes the forward map F_t.

Example 2: $\mu_c = DCC$, the previously described direction cosine attitude control mode. In this mode, m_b is designated as independent and u_m is the active control.

Example 3: $\mu_c = EAC$, the previously described Euler angle attitude control mode. In this mode, the independent output is m_α and the active control is u_m.

Example 4: $\mu_c = DFC$, direct force control mode. As already described, this mode requires a six-dimensional inversion. Both angular and translational acceleration $(\, m_b, a_r \,)$ are given, and the inverter finds the required (trim) values of the moment and power controls $(\, u_m, u_p \,)$.

If these four modes were the only ones of interest, the trim problem would be solved. But, in fact, most control modes of practical interest lead to more-complicated inversion problems. Several such cases are described.

Example 5: $\mu_c = HAR$. This is the Harrier mode, in which the direct side force in not available. It is treated as a parameter, $u_{p2} = 0$. The active control is five-dimensional: full moment control, plus throttle and nozzle. The active control has direct access to first and third components of the body coordinates of the total force, f_b. Full control of f_r is achieved by tipping the aircraft. That is, one of the attitude degrees of freedom is used for control. In the HAR mode, that degree of freedom is the

roll (α_{bt3}) channel, while the pitch (α_{bt1}) and yaw (α_{bt2}) channels remain free parameters. The relative attitude is represented by $C_{bt} = E_2(\alpha_{bt1})E_3(\alpha_{bt2})E_1(\alpha_{bt3})$.

Example 6: $\mu_c = VAT$. This is the vertical attitude control mode, in which the aircraft is sitting on its tail. The force is controlled by the thrust and the pitch and yaw channels. The force balance must be achieved by means of the throttle u_{p1} and the relative pitch and yaw angles α_{bt1} and α_{bt2}, while the relative roll angle α_{bt3} acts as a parameter: in the VAT control mode the aircraft may execute an independent roll schedule $\underline{\alpha}_{bt3}$. The relative attitude is represented by $C_{bt} = E_2(\alpha_{bt1})E_3(\alpha_{bt2})E_1(\alpha_{bt3})$.

Example 7: $\mu_c = HEL$. This is the helicopter mode, in which neither the direct side force controller nor the nozzle is available for active control. Only the full moment control and throttle (collective in helicopters) are active. Two additional degrees of freedom for complete force control are provided by roll and pitch channels, while the yaw channel remains free. The force balance is achieved by manipulating the throttle (or collective) u_{p1} and the relative pitch and roll angles α_{bt1} and α_{bt3}, respectively, while the relative yaw angle α_{bt2} is treated as a given parameter.

Example 8: $\mu_c = CFY$. This is the conventional (high-speed) cruise control mode. The thrust is mainly in the direction of the longitudinal axis $\vec{b}_1$ of the aircraft. The direct force control is the throttle. All three attitude channels are used for control purposes: pitch and roll to provide the two additional degrees of freedom for complete force control. Yaw is used to maintain the body side force f_{b2} near zero. The seven-dimension inversion computes the moment control, throttle control, and the three relative Euler angles: $(u_m, u_{p1}, \alpha_{bt})$.

Example 9: The CFB mode is also seven-dimensional, but it is the side air velocity v^a_{b2} that is externally specified (usually zero).

Example 10: In the CFF mode, flap u_{f1} is manipulated to obtain a given angle of attack α_{bt1}.

The last two entries in the table are modes for which the throttle is not available for control; it is being moved between its limits or sitting on one of them.

Example 11: The CFX, conventional flight X force control mode, is essentially an airspeed control mode. The independent force components are

$$(f_{t1}, f_{t2}, f_{b2}) \tag{203}$$

The normal force channel f_{t3} is not controlled; it is a dependent variable so that the climb rate $\dot{h}$ is not controlled directly.

Example 12: On the other hand, in the CFZ mode the independent force components are

$$(f_{t2}, f_{t3}, f_{b2}) \tag{204}$$

while the longitudinal channel f_{t1} is not controlled. This is a climb rate control mode in which a given altitude rate $\dot{h}(t)$ schedule is being executed, while the airspeed is not directly controlled.

These examples illustrate the fact that there may be many control modes to consider in a complete design of the guidance subsystem. Each mode generates a specific inversion problem. Furthermore, it

is desirable to be able to easily add new control modes, without having to redesign the whole algorithm. In the present approach, a new mode may be added by simply entering a new line in the control mode table. We use a standard variable dimension Newton-Raphson inversion algorithm. The labels in the control mode table are used to rearrange the input to this standard algorithm by storing the coordinates of y_t^* labeled as independent in the head y_I^* of the standard output, y^*; the dependent part is placed in the tail y_D^* of y^*. Similarly, the control coordinates of x_t^* are placed in the head x_C^* of the standard input x^*, and the parameters in the tail x_P^*. For a valid control mode, the Jacobian $\partial y_I/\partial x_C$ is always nonsingular in the region of interest. When the standard inverter reaches the threshold, then the output $(\hat{x}, \hat{y})$ is redistributed according to the labels into $(\hat{x}_t, \hat{y}_t)$. If the partition functions are defined by

$$\begin{array}{ll} (x_C, x_P) = \pi_x(x_t, \mu_c), & x_t = \pi_x^{-1}(x_C, x_P, \mu_c) \\ (y_I, y_D) = \pi_y(x_t, \mu_c), & y_t = \pi_y^{-1}(y_I, y_D, \mu_c) \end{array} \tag{205}$$

Then the multimode (direct) function for the standard inverter is given by

$$F(x, \mu_c) = \pi_y \left\{ F_t \left[\pi_x^{-1}(x, \mu_c) \right], \mu_c \right\} \tag{206}$$

In this fashion a single static inversion algorithm for any valid control mode is obtained,

$$(\hat{x}_t, \hat{y}_t) = \hat{F}_t(x_t^*, y_t^*, \mu_c) \tag{207}$$

The parameter ϵ is introduced in order to be able to suppress zero dynamics, if any, as follows.

$$F_t^\dagger(x_t^*, y_t^*, \mu_c, \epsilon) = \begin{cases} \hat{F}_t(x_t^*, y_t^*, \mu_c) & \text{if } \epsilon = 1 \\ \hat{F}_t(x_t^*, y_t^*, \mu_c)_{m_b = \omega_{brb} = 0} & \text{if } \epsilon = 0 \end{cases} \tag{208}$$

That is, $F_t^\dagger(x_t^*, y_t^*, \mu_c, 0)$ computes the force subject to the constraint that both angular acceleration ω_{brb} and angular velocity ω_{brb} of the aircraft relative to the runway are zero.

The $F_t^\dagger$ algorithm is quite powerful. For example,

$$(\hat{x}_t, \hat{y}_t) = F_t^\dagger(x_t^*, y_t^*, SIM, 1) \tag{209}$$

reproduces the exact simulation. In this trivial case, there is only one call to F_t and no inversion. The input x_t^* is copied without change into $\hat{x}_t$. The input y_t^* is ignored, while the output $\hat{y}_t = F_t(x_t)$.

The pure feedback approximation of the forward map F_t corresponding to equation (200) may be obtained with two calls:

$$\begin{array}{rcl} (\hat{x}_t, \hat{y}_t) & = & F_t^\dagger(x_t^*, y_t^*, SIM, 1) \\ (x_t^0, y_t^0) & = & F_t^\dagger(\hat{x}_t, \hat{y}_t, EAC, 0) \end{array} \tag{210}$$

The m_α component of $\hat{y}_t$ is the exact second derivative of the Euler angles, $\hat{m}_\alpha$, while the f_t component of y_t^0 is the pure feedback approximation f_t^0. The resulting force error

$$f_t^e = \hat{f}_t - f_t^0 \tag{211}$$

In the absence of zero dynamics, that error is bounded by the trim error bound ε. This inverter error will henceforth be largely ignored.

The pure feedback inverse of the force generator is obtained with two calls:

$$\begin{aligned}(x_t^0, y_t^0) &= F_t^\dagger(x_t^*, y_t^*, \mu_c, 0)\\ x_t^\dagger &= x_t^0,\ m_\alpha^\dagger = m_\alpha^*\\ (\hat{x}_t, \hat{y}_t) &= F_t^\dagger(x_t^\dagger, y_t^*, EAC, 0)\end{aligned} \tag{212}$$

The first call trims the aircraft to the desired force $f_t^0 = f_t^*$ by manipulating the active force controls as specified by the control mode μ, but with angular acceleration and angular velocity held at zero. The resulting moment control is u_m^0, and the power and configuration controls are (u_p^0, u_f^0). Then x_t^0 is copied into $x_t^\dagger$, except that $x_t^\dagger$ contains the true angular velocity $\dot{\alpha}_{bt}$. The second call trims the moment control for the specified angular acceleration m_α^* and angular velocity $\dot{\alpha}_{bt}^*$, disregarding force balance. The overall result is the exact moment control $\hat{u}_m$ for the given angular acceleration and angular velocity, the pure feedback values u_p^0, u_f^0, and α_{bt}^0 for the given f_t^*, and the exact $\hat{f}_t$. The force error due to the neglected zero dynamics is, again,

$$f_t^e = \hat{f}_t - f_t^0 \tag{213}$$

assuming a perfect inverter.

The final step in the construction of the pure feedback approximation of the multimode dynamic inverse $\hat{T}^0$ and direct T^0 is to raise the (zero-order) $F_t^\dagger$ algorithm to order two. As described previously, that step may be done routinely by means of dynamic forms. The result is shown in the following block digrams. Figure 40 shows the pure feedback approximation $\hat{T}^0$ of the exact inverse $\hat{T}$. The function

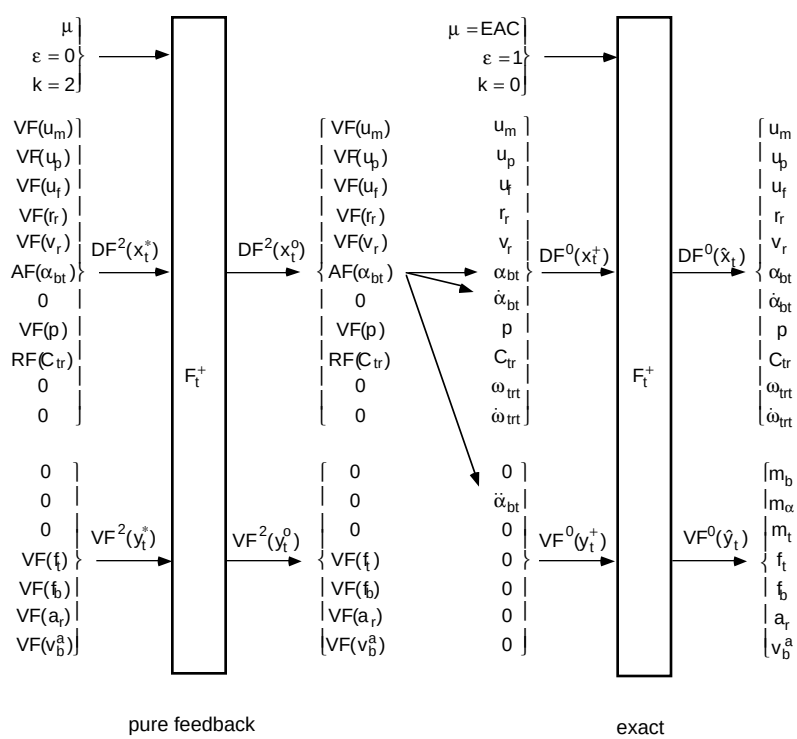

Figure 40. Pure feedback approximation $\hat{T}^0$ of the inverse map $\hat{T}$.

$F_t^\dagger$ is the multimode static inverter, as defined by equation (208) raised to arbitrary order. The input to the inverse map is the mode μ, $\epsilon = 0$, which cuts the zero dynamics, order $k = 2$, and second-order dynamic form of x_t^*, namely $DF^2(x_t^*)$, and second-order vector form $VF^2(y_t^*)$. The output of this call to $F^\dagger$ is the filled-in $DF^2(x_t^o)$ and $VF^2(y^o)$. The next step is to copy the zero-order components $DF^0(x_t^o)$ into $DF^0(x_t^\dagger)$, and copy the first and second derivatives of α_{bt} into the appropriate slots of $x_t^\dagger$, and similarly with $\dot{\omega}_{trt}$ and ω_{trt}. Finally, the last call to $F^\dagger$ performs the exact moment trim. At this point the pure feedback inverse $\hat{T}^o$ has been constructed, since the given dynamic forms have been transformed into the corresponding state and control.

The pure feedback direct algorithm T^o, shown in figure 41, is the reversal of the preceding algorithm. The first step is to compute the exact angular acceleration $\ddot{\alpha}_{bt}$ from the given state and control. Next, the second-order angle form $AF^2(\alpha_{bt})$ is assembled, and the pure feedback static inverter is called first with order zero. The resulting acceleration $\hat{a}_r$ is loaded into $VF^2(r_r)$ and $VF^1(v_r)$. Then $F^\dagger$ is called again, but to order $k = 1$, and the process is repeated once more. Finally, the output, $DF^2(x_t^0)$ and $VF^2(y_t^0)$, is computed. At this point T^o has been constructed, since the state and control have been transformed into the corresponding dynamic forms. Note that the approximate transformation T^o suppresses (blocks) the effects of zero dynamics. The effects of these two transformations may be summarized with the help of the block diagram in figure 42.

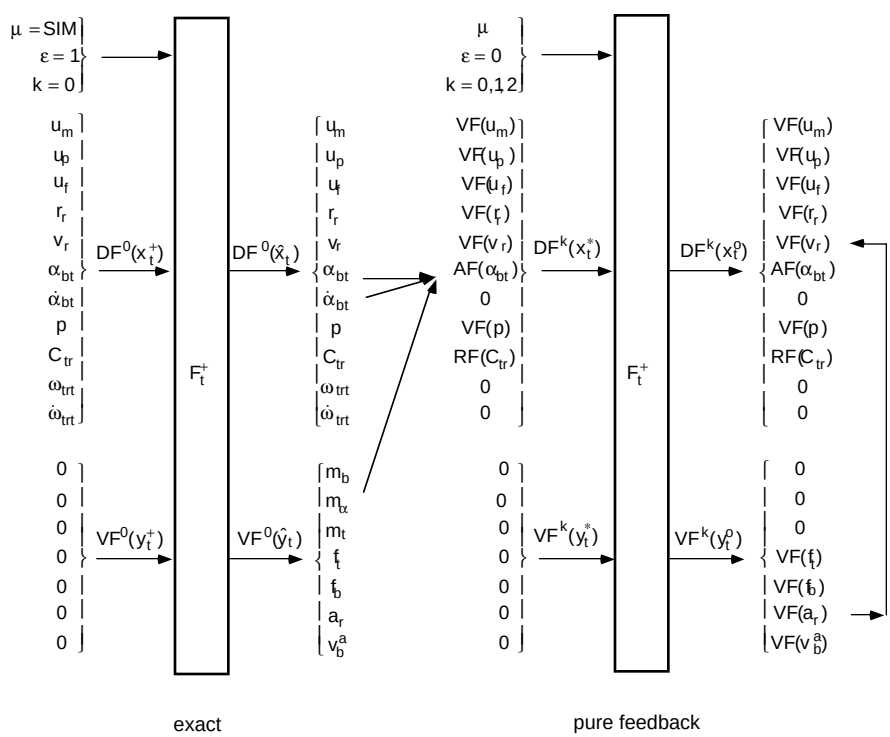

Figure 41. Pure feedback approximation T^0 of the direct map T.

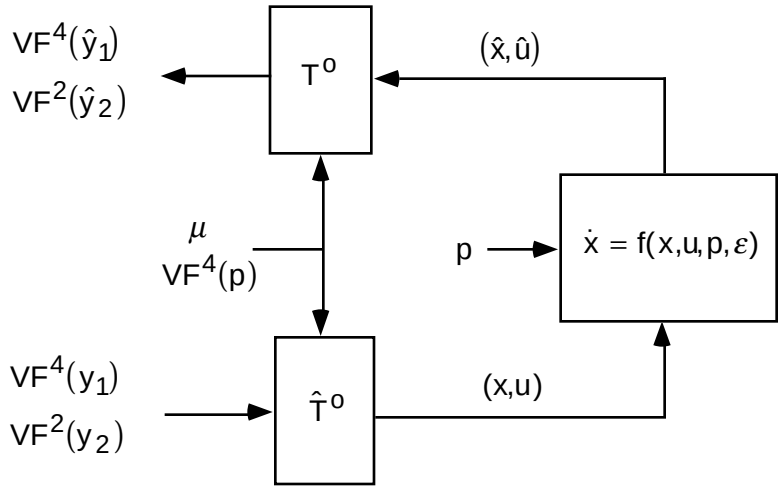

Figure 42. Pure feedback approximation for aircraft guidance.

The system

$$\dot{x} = f(x, u, p, 0) \tag{214}$$

is viewed through $\hat{T}^0$ and T^0 as a decoupled string of integrators (see bottom of figure 33). There will be deviations from this Brunowsky form if there are zero dynamics present. The control of such errors is considered next.

4.5 Guidance Servo

As discussed previously (see figure 32), a simple and direct way to control the errors caused by the neglected zero dynamics is by means of the servo structure shown in figure 43. The right-hand block

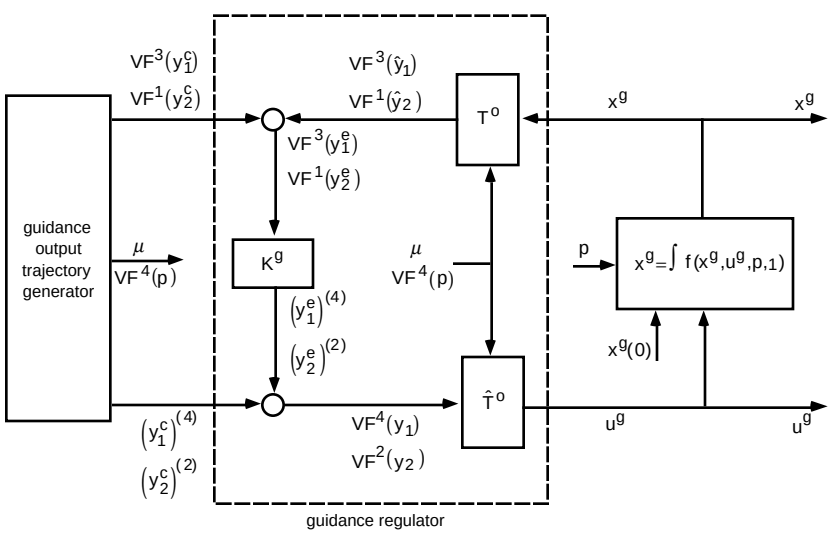

Figure 43. Guidance servo in canonical coordinates.

represents the exact simulation, whose state and control (x^g, u^g) are taken as the solution to our guidance problem. By construction, (x^g, u^g) always satisfies the state equation. The tracking error is given by the difference between the commanded output produced by the guidance output trajectory generator, as shown in figure 5 and discussed in Chapters 1 and 3 , and the actual output of the simulation:

$$VF^3(y_1^e)VF^1(y_2^e) \tag{215}$$

For the pure feedback case, these errors are identically zero. In any case, the loop on that error is closed by assuming that the system being regulated is a pure Brunowsky form. The corrective feedback is injected into the fourth derivative of y_1^c and the second derivative of y_2^c, which are the canonical controls for the pure feedback case:

$$\begin{aligned} (y_1^e)^{(4)} &= K_1^g(y_1^e, \dot{y}_1^e, \ddot{y}_1^e, (y_1^e)^{(3)}) \\ \ddot{y}_2^e &= K_2^g(y_2^e, \dot{y}_2^e) \end{aligned} \tag{216}$$

The resulting modified signals $VF^4(y_1)$ and $VF^2(y_2)$ are then transformed as before by the pure feedback dynamic inverse $\hat{T}^o$ into the natural state and control. The state is used only to initialize the simulation while the control drives the simulation model.

The design of the regulator law K^g is greatly simplified by the fact that the system to be controlled, when viewed through the transformations $\hat{T}^o$ and T^o, is, except for the corrupting influence of the zero dynamics, a collection of simple, decoupled strings of integrators. Linear techniques such as pole placement are quite effective. Of course, the regulator may be nonlinear or include dynamic elements such as filters to obtain smoothing. No gain scheduling is needed. That function is being provided automatically by the linearizing transformations.

In the following numerical examples, a linear regulator was selected and all the gains were held constant throughout the maneuvers. For each fourth-order string of integrators, the closed-loop poles, including a first-order filter with time constant of $\tau = 0.18$, were placed at

$$\begin{aligned} (\omega_1, \zeta_1) &= (0.5, 0.7) \\ (\omega_2, \zeta_2) &= (2.0, 0.7) \\ \sigma &= 2.0 \end{aligned} \tag{217}$$

For each second-order string of integrators, the closed-loop poles were placed at

$$(\omega_1, \zeta_1) = (2.0, 0.7) \tag{218}$$

and filters were not used.

5 EXAMPLES

In this section we present two numerical examples.

5.1 Maneuver: Move Forward 300 Feet

Consider again the maneuver M_1 described in figure 10. The commanded motion of the center of mass, $VF^4(r_r^c)$, which transfers the Harrier 300 feet forward in 18 seconds, from hover to hover, is shown in figure 44. There are four segments. 1) In the first segment of the maneuver, $0 \leq t < 8$, the aircraft is accelerated to 30 ft/sec in the Harrier mode, namely $\mu = HAR$, while the pitch angle, which is a parameter in this mode, is held at zero degrees. 2) In the second segment, $8 \leq t < 10$, the aircraft coasts for two seconds at 30 ft/sec with $\mu = HAR$. 3) In the third segment, $10 \leq t \leq 18$, the aircraft decelerates to hover with $\mu = HEL$, while the nozzle is commanded to 90 degrees. In this mode the pitch axis is used for control, while the nozzle angle becomes a parameter. 4) In the last segment, $18 \leq t < 24$, the command is to hover in the HEL mode.

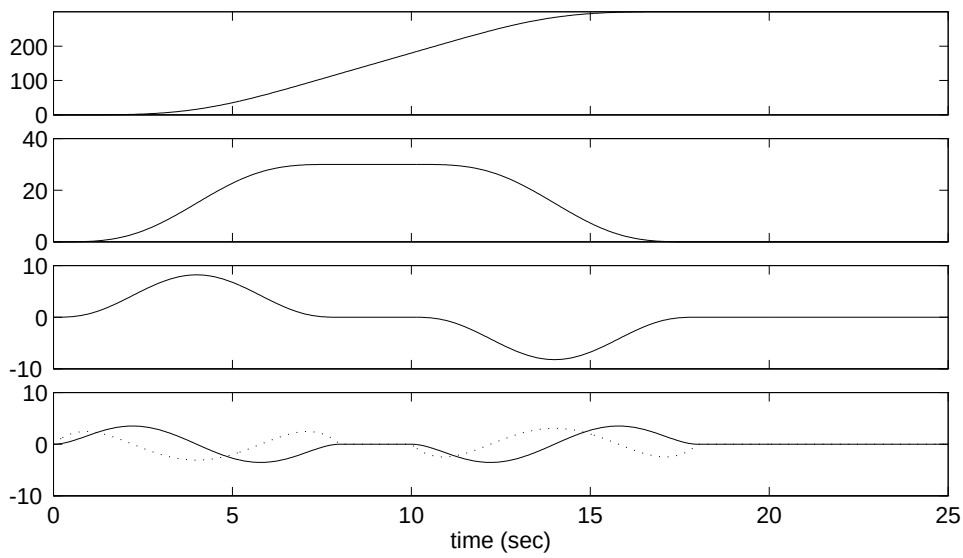

Figure 44. Forward step: commanded position $VF^4(r_c)$.

The behavior of the open-loop controls is shown in figure 45. The first panel shows the behavior of the pitch moment control u_{m2}. The solid curve is u_{m2}^o, which is the position of the pitch moment control assuming zero angular velocity and zero angular acceleration. The dotted curve is $\hat{u}_{m2}$, which is the position of the pitch moment control for the given angular velocity and acceleration without correcting for the effects of zero dynamics on the resulting force. The second panel shows the throttle and two of its derivatives, $VF^2(u_{p1})$. The third panel shows the nozzle angle and two of its derivatives, $VF^2(u_{p3})$. The last panel shows the motion of the relative pitch angle and its derivatives, $VF^2(\alpha_{bt1})$. This is also the total pitch angle θ of the aircraft body axes since in this case the t-axes were chosen to coincide with the runway axes, namely, $C_{tr} = I$.

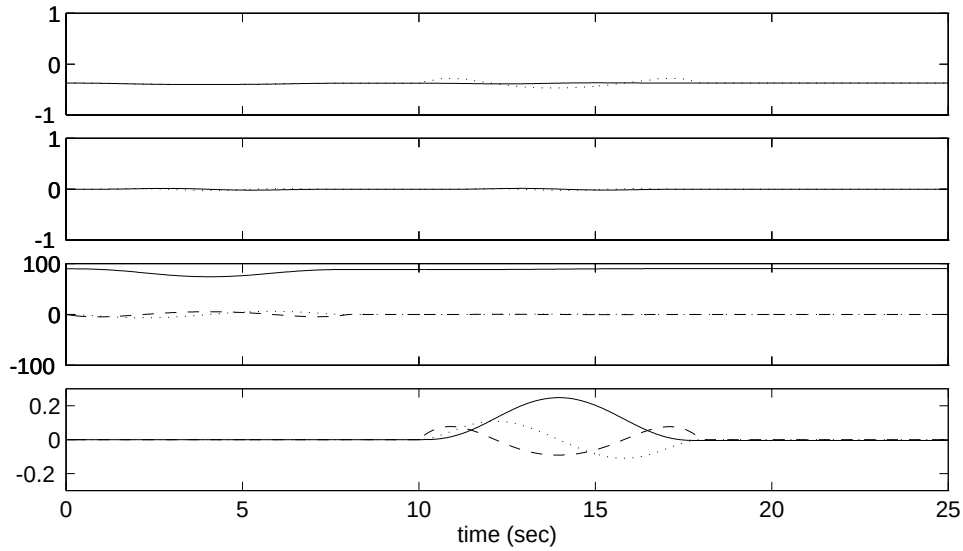

Figure 45. Forward step: open-loop controls.

The t-coordinates of the force error caused by the zero dynamics is the difference

$$f_t^e = f_t - f_t^c \tag{219}$$

between the exact force f_t, computed with the exact ($\epsilon = 1$) force algorithm F_t^{fr}, and the commanded force

$$f_t^c = C_{tr}(\ddot{y}_1/g - \delta_3) \tag{220}$$

This force error, shown in the first panel of figure 46, is quite small. The small jagged sections are inverter errors. The rest is due to zero dynamics. Some cross coupling from the longitudinal to normal channels is evident in the HEL mode. The RCS jets that generate the pitch moment also generate normal force. The inverse map by design does not take this effect into account. The resulting translational acceleration errors are bounded by 0.25 ft/sec.

Thus far in the present example the open-loop behavior has been considered. Now consider the closed-loop behavior, as in figure 43. The closed-loop tracking error r_r^e is shown in the second panel of figure 46. The maximum error for the entire maneuver is only 0.2 feet along track, and 0.6 feet in altitude. The acceleration error $a_r^e = \ddot{r}_r^e$, shown in the third panel, is bounded by the small value of 0.25 ft/sec^2. The corrective feedback, $(y_1^e)^{(4)} = (r_r^e)^{(4)}$ (see figure 43), shown in the fourth panel, is bounded by the small value 0.25 ft/sec^4. So, the weak zero effects, in the present case of the step forward maneuver, are well controlled by the simple and rather crude (guidance) regulator.

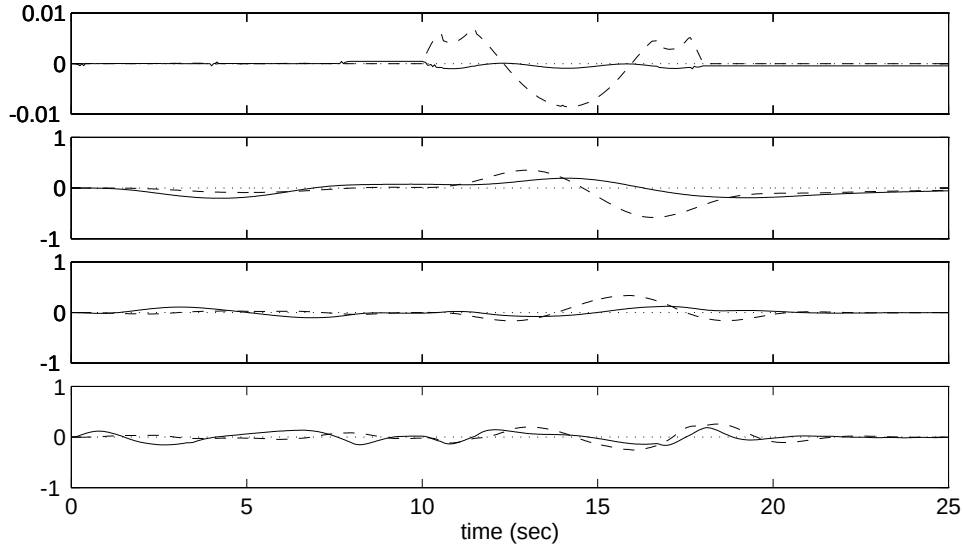

Figure 46. Forward step: force errors.

5.2 Maneuver: Frisbee

As a second example, consider the Frisbee maneuver described in figures 12 and 13. This maneuver is synthesized by means of 14 segments. During this complex maneuver, both the control and the attitude coordinatization modes change. As shown in the first panel of figure 47, the trajectory transfers the center of mass from hover at the origin to 4000 feet along the runway and 667 feet in altitude. The second panel shows the behavior of the velocity v_r^c. The duration of the maneuver is 50 seconds. The horizontal velocity changes from zero to 215 ft/sec, and the vertical from 0 to 15 ft/sec climb rate. The lateral velocity remains at zero. The third panel shows the yaw angle. Note the 360-degree yaw maneuver. The large-angle maneuver is accomplished by means of an intermediate axis system $\overleftrightarrow{t}$, as described before. In the first part of the maneuver, C_{tr} is parameterized by means of Euler angles in the standard sequence:

$$C_{tr} = E_1(\phi)E_2(\theta)E_3(\psi) \tag{221}$$

Then during the second, higher speed part, the wind path axis (WPA) parameterization is used.

The last panel shows the behavior of the body coordinates of the relative air velocity v_{bc}^a. It may be noted that during the initial, slow part of the maneuver, there is significant side velocity, and, in fact, for a time the aircraft is flying backward, but in the high-speed condition the speed is along $\vec{t}_1$ with no side velocity.

The behavior of the open-loop controls is shown in figure 48. The first panel shows the moment controls u_m. There are three pairs of curves. The difference between members is due to the zero dynamics. The next two panels show the throttle and nozzle and two of their derivatives. The last panel shows the flap and aileron droop.

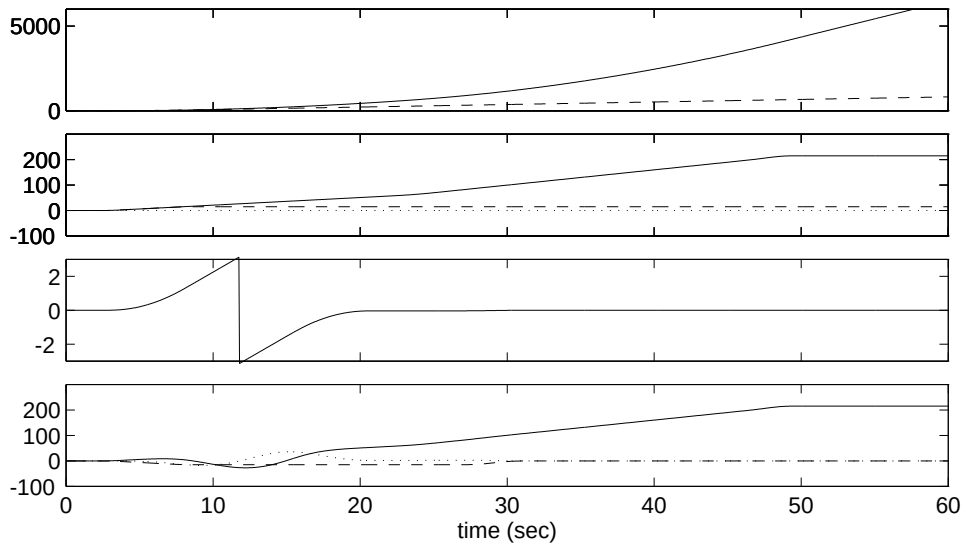

Figure 47. Frisbee: trajectory $VF^4(r_{rc})$.

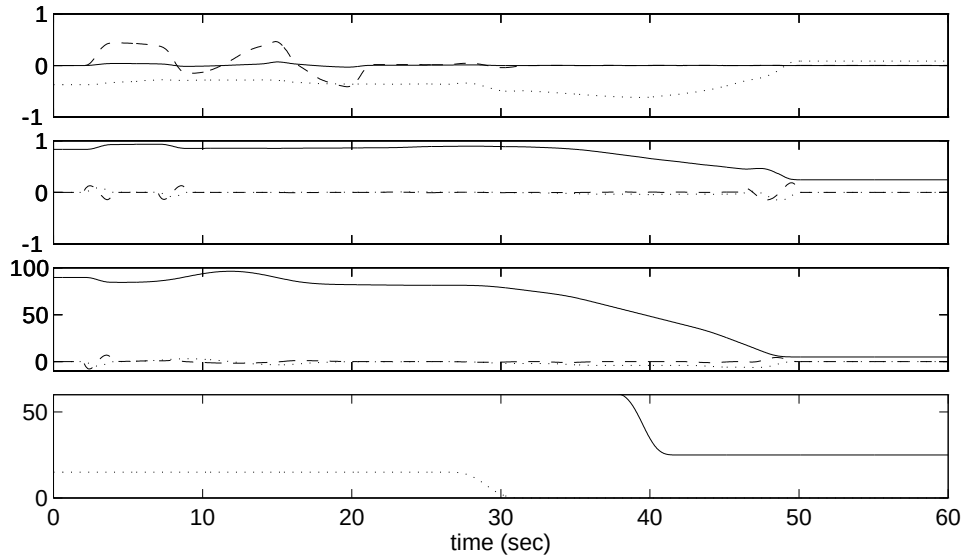

Figure 48. Frisbee: open-loop controls.

The open-loop behavior of the angular variables is shown in figure 49. The first part (the first 20 seconds) of the maneuver is accomplished in the HAR mode, and the rest in the HEL mode. The first panel shows the motion of the relative angles α_{bt} locating the body axes $\overleftrightarrow{b}$ relative to the intermediate axes $\overleftrightarrow{t}$. In the first part of the maneuver, the roll angle α_{bt3} rolls the force vector to maintain along-track acceleration. In the second part, the relative pitch angle α_{bt1} tips the force vector forward to maintain along-track acceleration. Later in the maneuver, this angle becomes the angle of attack (producing lift) because the coordinatization of aircraft attitude is switched to the wind path axes.

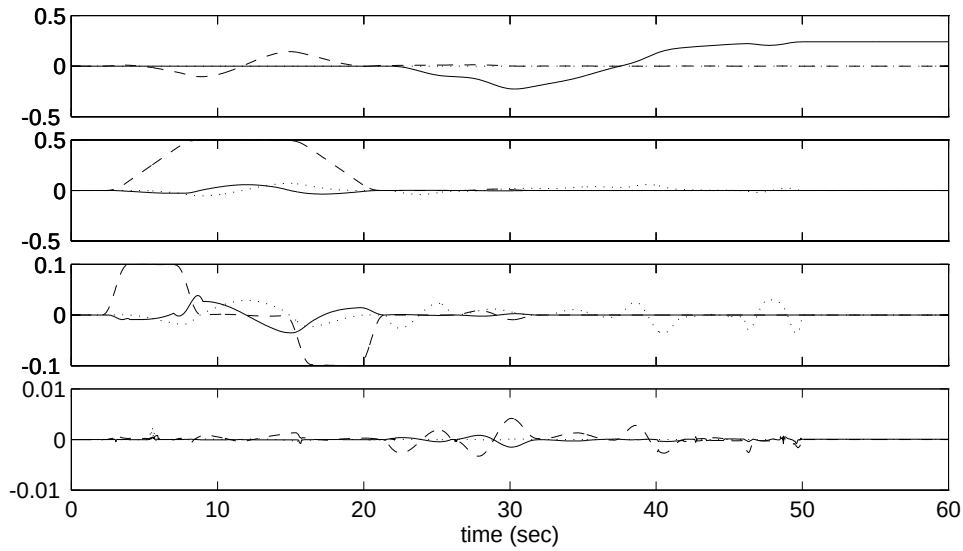

Figure 49. Frisbee: open-loop rotation and force error.

The second and third panels in figure 49 show the first and second derivatives of α_{bt}. Both vectors are well behaved. The force error f_t^e [see eq. (219)], due to zero dynamics, is shown in the bottom panel. It is quite small, being bounded by 0.005 g's. The activity in the HAR mode is due to the RCS, whereas the activity in the HEL mode is due to both RCS and the lift of the elevator.

The closed-loop position error r_r^e, shown in the first panel of figure 50, is bounded by 1 ft throughout this complex maneuver. The acceleration error a_r^e, shown in the second panel, is quite well behaved, being bounded by the small value of 0.3 ft/sec^2. The corrective feedback $(y_1^e)^{(4)}$ (see fig. 43) shown in the third panel is active, but bounded by the small value of 1 ft/sec^4. The final panel shows the closed-loop behavior of the moment control u_m.

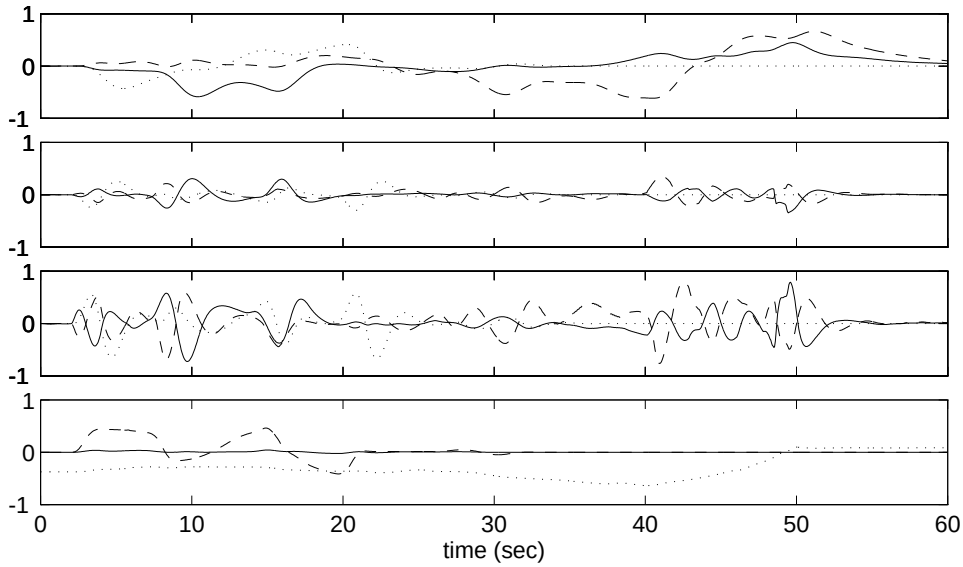

Figure 50. Frisbee: closed-loop behavior.

In summary, the two fairly complex examples show that the proposed approach based on inversion of the pure feedback approximation, followed by a regulator in canonical coordinates, is feasible for maneuvers that are slow relative to the aircraft zero dynamics.

6 CONCLUSION

An effective method has been described for guiding a dynamic system, such as an aircraft, through complicated maneuvers. The maneuvers are specified by means of a control point table, which is the discrete input to the guidance algorithm. The output is the continuous reference motion of the aircraft state and all the controls. The state is maintained continuous across the control points. Furthermore, this reference motion is flyable exactly by the aircraft simulation model. Therefore, the structure of the regulator for the actual aircraft may be taken to be that of an exact model follower. This feature has the great practical advantage that this (plant) regulator is subjected to process uncertainties only and is not loaded down by *apriori* known signals. Consequently, the full authority of the regulator is available for the control of disturbances and other uncertainties. The proposed multimode dynamic inversion procedure, consisting of the pure feedback approximation followed by a (guidance) regulator, is effective for maneuvers that are slow relative to the the plant zero dynamics. For stronger zero dynamics, the method described in reference 10 for the improvement of the pure feedback approximation may be employed. The methodology of the dynamic forms is very effective for organizing and drastically simplifying the coding of the proposed algorithms.

REFERENCES

1. NASA Strategic Plan, National Aeronautics and Space Administration, Washington, D.C., Feb. 1996.

2. Sherry, Lance; Youssefi, David; and Hynes, Charles: HSR FD G&C Task 4—VMS Structure Development: A Formalism for the Specifications of Operationally Embedded Reactive Avionics Systems. Honeywell Publication C69-5370-001, Oct. 1995.

3. Meyer, George; and Smith, G. Allan: Dynamic Forms Part I: Functions. NASA TP-3397, Aug. 1993.

4. The YAV-8B Simulation and Modeling. Volume 1: Aircraft Description and Program Summary. NASA CR-170397, 1983.

5. Goldstein, H.: Classical Mechanics. Second ed., Addison-Wesley, 1980.

6. Arnold, V. I.: Mathematical Methods of Classical Mechanics. Second ed., Springer-Verlag, 1989.

7. Isidori, Alberto; and Byrnes, Christopher I.: Output Regulation of Nonlinear Systems. IEEE Trans. on Automatic Control, vol. 35, no. 2, Feb. 1990, pp. 131–140.

8. Devasia, S.; and Paden, B.: Exact Output Tracking for Nonlinear Time-Varying Systems. Paper TP-11 3:20, Proc. 33rd Conf. on Decision and Control, Lake Buena Vista, Fla., Dec. 14–16, 1994, pp. 2346–2355.

9. Devasia, Santosh; Chen, Degang; and Paden, Brad: Nonlinear Inversion-Based Output Tracking, IEEE Trans. on Automatic Control, vol. 41, no. 7, July 1996, pp. 930–942.

10. Meyer, G.; Hunt, L. R.; and Su, R.: Nonlinear System Guidance in the Presence of Transmission Zero Dynamics. NASA TM-4661, Jan. 1995.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 1997

3. REPORT TYPE AND DATES COVERED
Technical Paper

4. TITLE AND SUBTITLE
Dynamic Forms Part II:
Application to Aircraft Guidance

5. FUNDING NUMBERS
548-40-12

6. AUTHOR(S)
George Meyer and G. Allan Smith

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER
A-976688

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TP-3695

11. SUPPLEMENTARY NOTES
Point of Contact: George Meyer, Ames Research Center, MS 210-3, Moffett Field, CA 94035-1000
(650) 604-5750

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified — Unlimited
Subject Category 31

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The paper describes a method for guiding a dynamic system through a given set of points. The paradigm is a fully automatic aircraft subject to air traffic control (ATC). The ATC provides a sequence of waypoints through which the aircraft trajectory must pass. The waypoints typically specify time, position, and velocity. The guidance problem is to synthesize a system state trajectory that satisfies both the ATC and aircraft constraints. Complications arise because the controlled process is multidimensional, multiaxis, nonlinear, highly coupled, and the state space is not flat. In addition, there is a multitude of operating modes, which may number in the hundreds. Each such mode defines a distinct state space model of the process by specifying the state space coordinatization, the partition of the controls into active controls and configuration controls, and the output map. Furthermore, mode transitions are required to be smooth.

The proposed guidance algorithm is based on the inversion of the pure feedback approximation, followed by correction for the effects of zero dynamics. The paper describes the structure and major modules of the algorithm, and the performance is illustrated by several example aircraft maneuvers.

14. SUBJECT TERMS
Automatic control, Algorithms, Differentiation

15. NUMBER OF PAGES
82

16. PRICE CODE
A05

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT