OPTIMAL HYPersonic PURSUIT EVASION

by

William Scripps Beebee

B.S.E., Princeton University, 1964
M.S. in E.E., University of Pennsylvania, 1969

Submitted in Partial Fulfillment
of the Requirements for the
Degree of Doctor of Science
in Instrumentation

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1975

Signature of Author

Certified by ________________________________
Thesis Supervisor

Certified by ________________________________
Thesis Supervisor

Certified by ________________________________
Thesis Supervisor

Accepted by ________________________________
Chairman, Departmental Graduate Committee
OPTIMAL HYPersonic PURSUIT EVASION

by

William Scripps Beebee

Submitted to the Department of Aeronautics and Astronautics on September 30, 1974 in partial fulfillment of the requirements for the degree of Doctor of Science in Instrumentation.

ABSTRACT

This thesis concerns optimal evasion from a pursuer possessing closed-loop guidance. The evader (E) is a maneuvering reentry body descending in a vertical plane at hypersonic speed towards a desired impact point on the ground. The pursuer (P) is a Sprint-type interceptor missile launched before the combat begins and flying in the same plane. E is "blind" to P but also has a particular, desired ground-impact point, the location of which P does not know. P is assumed to have perfect current information about E's state, mass and aerodynamic coefficients. The only thing P does not know is what E plans to do next.

The E model experiences gravity and aerodynamic force but has instantaneous response to the commanded control angle-of-attack. P is a thrusting vehicle with a nontrivial, lateral-acceleration-commanded autopilot and rapid acceleration through the atmosphere. Realistic dynamic constraints are made on the control allowed in each case.

The overall objective is to find an "open-loop" time history for E control which avoids interception (by an unseen P using an unknown pursuit law) yet assures impact on the desired ground target. As a practical approach to shed light on the larger problem, three pursuit laws are postulated and in optimal evasion maneuver is found numerically against each. An optimal maneuver maximizes the intercept miss yet hits the ground target. The most elementary pursuit scheme studied is proportional navigation. The other two schemes, developed by the author, extrapolate future P and E flight paths via approximate solutions to aerodynamic motion equations. They differ in the types of future control possibilities postulated for E. The P control is chosen using gradient search techniques in both cases, with the optimization process updated at discrete time points during the encounter. For the first of these two steering laws, the optimal evasion is also found for the cases of two off-nominal P lateral acceleration limits and a "head-on" intercept geometry.
The results of this study indicate that a maneuver that is optimal against one type of pursuit law is not necessarily successful against another. Optimal evasion from proportional navigation calls for a high-drag, weaving flight. However, the latter two steering laws intercept this maneuver. These laws predict high-drag E trajectories and are best thwarted by a steep, low-drag dive, a type of flight which proportional navigation intercepts with ease. When the P acceleration limits are raised or lowered, the optimal evasion is less or more successful, respectively. Changing the combat geometry from the nominal, right-angle intercept to a head-on encounter, optimal evasion requires a sudden turn from E at the end. This opposes an assumption of straight-line flight for E postulated in the steering law used. In all cases studied evasion succeeds by "fooling" the prediction algorithm of the pursuit law via an unexpected type of maneuver.

Thesis Supervisors:

Wallace E. VanderVelde, Sc.D.
Title: Professor Aeronautics and Astronautics

Theodore N. Edelbaum, M.S.
Title: Lecturer in Department of Aeronautics and Astronautics

James E. Potter, Ph.D.
ACKNOWLEDGEMENTS

This thesis would not have been completed without the help of a number of people. Professor Wallace VanderVelde was instrumental in providing many, many hours of important discussion and advice. His guidance in overcoming numerous numerical difficulties in the research and his careful reading and criticism of the thesis drafts are especially appreciated. The author is also grateful to Mr. T. N. Edelbaum and Dr. James Potter for insightful, constructive criticism and suggestions in the course of the thesis writing and research. The contributions of the thesis committee have played a major part in making the degree program a rewarding experience.

Great amounts of time and money were spent in writing and debugging the several large computer programs which provide the numerical results for the thesis. The author is especially indebted to Ernest Sabine of the Charles Stark Draper Laboratory for providing quick, accurate answers to numerous programming questions. Stephen Duggan, David Lynde and Arnold Soltz of the same organization were also of great assistance in the programming effort.

No graduate level thesis research is possible without a financial sponsor. The effort was supported under DOD contract number N00030-75-C-0038. The author's supervisor, Mr. Peter Howard of the Draper Laboratory, was most cooperative over a
period of several years in leaving the author virtually free from non-thesis duties and arranging payment for the rather large amount of computation time required. The author is also grateful to Mr. Howard for proposing the thesis topic.

No graduate level thesis research is possible for a married man without the personal sponsorship of his wife. The author is indebted to Shirley Beebee for relieving him of family duties on nights and weekends over the last several years. Anyone who has tried to raise two small children without a husband will appreciate the sacrifices involved.

The publications of this report does not constitute approval by MIT, the C. S. Draper Laboratory, Inc. or the Department of Defense of the findings or the conclusions contained therein. It is published only for the exchange and stimulation of ideas.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapters</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 INTRODUCTION AND SYNOPSIS</td>
<td>8</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>8</td>
</tr>
<tr>
<td>1.2 Synopsis</td>
<td>13</td>
</tr>
<tr>
<td>2 PREDICTION TECHNIQUES</td>
<td>17</td>
</tr>
<tr>
<td>2.0 Summary</td>
<td>17</td>
</tr>
<tr>
<td>2.1 Basic Explanation</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Approximate Solution to the Evader Equations of Motion</td>
<td>21</td>
</tr>
<tr>
<td>2.3 Approximate Solution to the Pursuer Equations of Motion</td>
<td>28</td>
</tr>
<tr>
<td>3 INTERCEPTION TECHNIQUES</td>
<td>41</td>
</tr>
<tr>
<td>3.0 Summary</td>
<td>41</td>
</tr>
<tr>
<td>3.1 Proportional Navigation</td>
<td>41</td>
</tr>
<tr>
<td>3.2 Minimum-Miss-Time Guidance</td>
<td>44</td>
</tr>
<tr>
<td>3.3 Least Risk Guidance</td>
<td>50</td>
</tr>
<tr>
<td>4 OPTIMAL EVASION</td>
<td>56</td>
</tr>
<tr>
<td>4.0 Summary</td>
<td>56</td>
</tr>
<tr>
<td>4.1 Searching for the Optimum: Modified Gradient Projection and Conjugate Gradient Methods</td>
<td>56</td>
</tr>
<tr>
<td>4.2 The State Equation and Its Derivatives</td>
<td>70</td>
</tr>
<tr>
<td>4.3 Optimization Against Proportional Navigation</td>
<td>80</td>
</tr>
<tr>
<td>4.4 Optimization Against Minimum-Miss-Time (MMT) Interception</td>
<td>84</td>
</tr>
<tr>
<td>4.5 Optimization Against Least Risk (LR) Interception</td>
<td>88</td>
</tr>
<tr>
<td>5 NUMERICAL EXAMPLES</td>
<td>93</td>
</tr>
<tr>
<td>5.0 Summary</td>
<td>93</td>
</tr>
<tr>
<td>5.1 Parameter Values Assumed</td>
<td>93</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (Cont)

<table>
<thead>
<tr>
<th>Chapters</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2 Search Behavior in Seeking the Optimum...</td>
<td>94</td>
</tr>
<tr>
<td>5.3 Optimal Evasion vs. Proportional Navigation</td>
<td>97</td>
</tr>
<tr>
<td>5.4 Optimal Evasion vs. Minimum-Miss-Time Guidance</td>
<td>112</td>
</tr>
<tr>
<td>5.5 Optimal Evasion vs. Least Risk Guidance...</td>
<td>149</td>
</tr>
<tr>
<td>5.6 Summary of Results</td>
<td>164</td>
</tr>
<tr>
<td>6 CONCLUSIONS AND RECOMMENDATIONS</td>
<td>169</td>
</tr>
<tr>
<td>6.1 Conclusions</td>
<td>169</td>
</tr>
<tr>
<td>6.2 Recommendations</td>
<td>172</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendices</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A EVADER AND PURSUER MODELS</td>
<td>174</td>
</tr>
<tr>
<td>B FIRST AND SECOND DERIVATIVE INFORMATION</td>
<td>189</td>
</tr>
<tr>
<td>C P AND E PREDICTION ACCURACY</td>
<td>313</td>
</tr>
<tr>
<td>D APPROXIMATE AUTOPILOT RISE TIME CALCULATION</td>
<td>315</td>
</tr>
</tbody>
</table>

References

Biography
CHAPTER 1
INTRODUCTION AND SYNOPSIS

1.1 Introduction

This thesis concerns optimal evasion from a pursuer possessing closed-loop guidance.* The evader (E) is a maneuvering reentry body descending at hypersonic speed towards a desired impact point on the ground. The pursuer (P) is a Sprint-type interceptor missile launched before the combat begins. All motions of P and E are assumed to occur in a vertical plane. As is the case for an actual ICBM warhead, E is "blind" to P but also has a particular, desired ground-impact point. It is assumed that P is defending a number of targets. Thus, P does not know E's destination. This scenario is illustrated in Figure 1.1-1.

The evader is postulated to be a point-mass, experiencing axial and normal aerodynamic forces as well as gravity.† For the relatively short flight times and distances, an exponential atmosphere, a flat earth and a constant gravity field are assumed. To keep down the number of dynamic states of the problem, autopilot dynamics are neglected for E. Steering is accomplished by varying the E angle-of-attack (the E control).

*In a sense, the title could as well be "Optimal Hypersonic Evasion from Hypersonic Pursuit."
†See Appendix A for complete E and P model descriptions.
Figure 1.1-1  P vs E Combat Scenario
Unlike E, the pursuer is a thrusting vehicle with nonzero rotational inertia and a nontrivial, time-varying autopilot. P steers by generating appropriate net force using aerodynamics and a component of engine thrust. The fuel-burning process in P causes the thrust, mass, center-of-mass and moment-of-inertia to be time-varying. These functions are thus predetermined by the solid-fuel engine characteristics.

During the encounter, P is assumed to have perfect current information about E. In addition to noiseless current position and velocity data, P knows E's mass and aerodynamic coefficients. The only thing P does not know is what E plans to do next. It is the effect of this pursuit dilemma which is being studied. Inclusion of noise in the state observations and imperfect knowledge of the E mass, etc., would make the problem more realistic but would also obscure this basic question. Since it is not desired to create a trivial problem, realistic dynamic constraints are made on the two parties. E is limited in angle-of-attack magnitude while P has a finite lateral acceleration capability as well as nonzero response time.

The overall problem addressed can be stated very simply:

how does E maneuver to

a. impact the desired ground target and

b. avoid interception

when the pursuit steering law is unknown and E cannot observe P during the flight?
What is desired is a preprogrammed control history for E to fly which will be tolerably successful against any pursuit scheme P would be likely to use. A practical approach to solving this problem has been adopted. Three pursuit policies are postulated and an optimal evasion is obtained numerically against each. The first scheme studied is proportional navigation, which postulates straight-line future E and P trajectories. The other two steering laws were developed by the author. They incorporate sophisticated techniques for extrapolating E and P flight paths. The future control for P is found in both cases to satisfy stated optimal conditions, and this optimal control is updated at discrete points in time in the light of information current at those times. Time and computation expense prohibited experimentation with additional pursuit schemes. However, the laws studied span a wide enough range of interception philosophies to be of interest as a first step in a series of efforts to solve the overall problem.

The author has seen only two references to prior work on the problem of optimal evasion from a known pursuit law. Both deal with proportional navigation for P. The most recent is contained in course notes given at the USAF Academy by Scott, Preyss and Willes.¹ The authors investigate air-to-air combat in the horizontal plane with no objective for the evader other

than to dodge interception. Highly idealized, linear, constant-coefficient equations of motion are assumed and the relative closing velocity is treated as a constant. Under the further assumption of control-limited P and E and no P autopilot lag, the optimum maneuver in a tail-chase is found to be a hard pull-right or pull-left. The authors acknowledge that in the presence of such an autopilot lag, the optimum is probably a weaving, "scissors" maneuver. Earlier work is discussed by Julich and Borg in an engineering note to the *Journal of Spacecraft*. In their paper, the authors also consider planar air-to-air combat without a desired terminal point for E. Constant speeds are assumed for both parties and the control in each case is assumed to be the azimuth angle rate. P is postulated to have an autopilot with saturation and a first-order lag imposed on the commanded rate. E has a control limit but no lags. With this scenario, Julich and Borg find that the optimal maneuver structures a head-on or a tail-chase situation if the geometry permits. Thereafter, E either breaks away with a hard turn if the encounter is head-on or weaves back and forth in the scissor maneuver for a tail-chase. As before, the scissors maneuver is caused by and dependant on the P autopilot time-constant.

---

Contrasting with the above references, this work not only deals with a range of interception laws but also carries much more realistic assumptions about the P and E models. Both vehicles have nearly true aerodynamic characteristics with drag a function of lift and a nontrival atmospheric model assumed. No constant-velocity constraints are made. The P model has an almost-realistic autopilot controlling the time-varying response characteristics associated with a fuel-consuming, thrusting vehicle. As will be shown in Chapter 5, the encounters studied feature realistic, rapidly changing velocity and control environments in which P and E must perform. The results of the study are more useful for drawing conclusions about actual ABM-ICBM contests.

1.2 Synopsis

Chapter 2 contains discussions of the problems of predicting future P and E trajectories. The aerodynamic equations of motion are presented for E with constant control history assumed. With gravity ignored (small compared to aerodynamic forces), these equations are solved approximately using a partly analytic, iterative technique developed by the author. An expanded version of the same approximation method is used to "solve" the P equations. In this case the control is assumed to be a biased ramp in time. Again, gravity is neglected although the effect of P control saturation is explicitly dealt with.
The three postulated interception laws are presented in Chapter 3. The first, proportional navigation, appears in a conventional form. In the second scheme the E trajectory is predicted forward in time assuming a constant flight-path-angle but drag associated with a maximum pull-up or pull-down. A gradient search is used along with P trajectory prediction to choose the biased ramp control resulting in an "intercept" of the predicted E path. The third interception law generates hard pull-up and hard pull-down predictions for E and finds, via a more complicated gradient search, a P control to guard against both trajectories. The hope in doing this is that the true E trajectory will lie in between and thus be intercepted.

In Chapter 4 the optimum-seeking routines are developed for evading interception by the three pursuit steering laws. The objective of E is couched in mathematical terms. The optimum methods to be used are then defined and their implementations discussed. Finally, the state equation and its partial derivatives are presented, noting the particular features associated with each interception law.

In Chapter 5 some sample interception scenarios are presented. The proportional navigation interceptor is first set against three different E maneuvers, including the optimal one. Radically different E control histories result in a broad range of miss distances. The best results from E's view are found by performing a high-drag, weaving maneuver. The worst come from
a low-drag dive. While E is pitted against the second and third interception schemes, the overall best maneuver is the low-drag dive which was the worst before. The weaving maneuver seen to be successful against proportional navigation fairs poorly against the more sophisticated interception laws. Three distinct P lateral acceleration limits are tested using the second steering law. The results are generally what one would expect: the higher the limit, the closer the interception miss. Finally, an entirely different encounter geometry is postulated in which E is forced to fly over the initial P location and an almost head-on conflict results. The second pursuit scheme is also assumed in this case and, contrasting to previous experience, the optimal E maneuver features a sudden turn just before the intercept time.

The final chapter contains conclusions and recommendations for future work. The observation is made that in all cases tested the optimal evasion seeks to increase the component of relative position prediction error "cross-track" to P's flight path. This is accomplished by flying a maneuver which is the opposite in character to that expected by P. Thus, for right-angle encounters E uses a high-drag maneuver when a low-drag one is predicted and vice-versa. In the altered-geometry, head-on situation P is "fooled" by a sharp turn when a straight flight path is predicted. An unexpected result of the study is that the optimal maneuver against the most effective interception law happens to be much less successful against the least
effective law. This leads one to believe that more work is needed by studying other interception schemes before a best open-loop strategy can be found for E when the pursuit law is unknown. Other suggested future work includes extending the analysis into the third (out-of-plane) dimension and using more realistic (classified) models for P and E.
CHAPTER 2
PREDICTION TECHNIQUES

2.0 Summary

Advanced interception schemes require prediction of the future trajectories of the pursuer and the evader. P needs to be able to predict the E trajectory (with an assumed E control) in order to steer to an intercept. P also needs to know its own flight path for a given control. This chapter describes new applications of some old numerical techniques in deriving methods to predict both trajectories.

2.1 Basic Explanation

In the research following it is essential to be able to predict the E and P equation-of-motion solutions with a minimum of computation. A general technique has been evolved for doing this. Although the author derived it without reference to previous work, the approach is in fact a combination of two time-honored techniques: the method of successive substitutions\(^3\) and Newton-Cotes integration.\(^4\)

Consider the general nonlinear, time-varying scalar differential equation

\[ \dot{X} = f(x, t), \quad X(t_0) = X_0 \]  \hspace{1cm} (2.1-1)

---


\(^4\) IBID, p. 71.
to be solved over $[t_0, t_f]$. If the right-hand-side of the equation is analytically integrable or falls into some other convenient category, standard procedures yield the solution. Suppose this is not the case but that $f(x,t)$ is continuous in $\chi$ and "well-behaved"* in $t$. Then, we fit $f(x,t)$ with a polynomial in $t$. To expedite this, assume initially that

$$f(x,t) \approx f(x_0,t) \quad (2.1-2)$$

The curve fit implies that

$$f(x_0,t) \approx c_0 + c_1 (t-t_0) + \cdots + c_n (t-t_0)^n$$

$$\approx c_0 + c_1 \tau + \cdots + c_n \tau^n \quad (2.1-3)$$

with $\tau \equiv t-t_0$.

To solve for $\{c_i\}$ let us first evaluate $f(x_0,t)$ at $n+1$ points on $t \in [t_0, t_f]$. Using $t_0$ and $t_n$ as the first and last points, the remaining $\{t_1, \ldots, t_{n-1}\}$ can be selected to be concentrated in regions of rapid time variability of $f(x_0,t)$ or can be located using some other criterion based on known properties of $f(x,t)$. Solving the equations in matrix form, we obtain

$$c_0 = f(x_0,t_0)$$

$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} t_1, \ldots, t_1 \\ \vdots \\ t_n, \ldots, t_n \end{bmatrix}^{-1} \begin{bmatrix} f(x_0,t_1) - f(x_0,t_0) \\ \vdots \\ f(x_0,t_n) - f(x_0,t_0) \end{bmatrix} \quad (2.1-4)$$

* "Well-behaved" is here taken to mean "can be accurately fit by a low-order polynomial over the range of interest."

† From this point on, we assume $\tau \equiv t-t_0$. 

where $T_i' = t_i' - t_0$.

We now proceed to integrate the differential equation \( (2.1-1) \) with $f(x,t)$ replaced by the polynomial approximation.

Integrating

$$\dot{x} = f(x,t) \approx c_0 + c_1 t + \cdots + c_n t^n$$

yields

$$x'(t) = x_0 + \int_0^t \left[ c_0 + c_1 \eta + \cdots + c_n \eta^n \right] d\eta$$

$$= x_0 + c_0 t + \frac{1}{2} c_1 t^2 + \cdots + \frac{1}{n+1} c_n t^{n+1} \quad (2.1-5)$$

where $x'(t)$ indicates the first approximation to the true $x(t)$.

Defining

$$x_{i}^{(0)} \equiv x^{(0)}(t_i), \ldots, x_{n}^{(0)} \equiv x^{(0)}(t_n)$$

we obtain a refined polynomial fit. Solving for a new set of \( \{ c_i \} \) values, called \( \{ c_i^{(1)} \} \), requires simultaneous solution of

$$C_{0}^{(1)} = f(x_0, t_0)$$

$$\begin{bmatrix} C_{1}^{(1)} \\ \vdots \\ C_{n}^{(1)} \end{bmatrix} = \begin{bmatrix} t_1 & \cdots & t_1^n \\ \vdots & \ddots & \vdots \\ t_n & \cdots & t_n^n \end{bmatrix}^{-1} \begin{bmatrix} f(x_{1}^{(0)}, t_1) - f(x_0, t_0) \\ \vdots \\ f(x_{n}^{(0)}, t_n) - f(x_0, t_0) \end{bmatrix} \quad (2.1-6)$$

Following equation \( (2.1-5) \), the new function approximating $x(t)$ is

$$x^{(2)}(t) = x_0 + c_0^{(1)} t + \cdots + \frac{1}{n+1} c_n^{(1)} t^{n+1} \quad (2.1-7)$$

We are now in a position to write the $L^h$ iteration step:
\[ C^{(\ell-1)}_\ell = f(x_\ell, t_\ell) \]

\[
\begin{bmatrix}
  C^{(\ell-1)}_1 \\
  \vdots \\
  C^{(\ell-1)}_n \\
\end{bmatrix} = 
\begin{bmatrix}
  t_1, \ldots, t_n \\
  \vdots \\
  t_n \\
\end{bmatrix}^{-1} 
\begin{bmatrix}
  f(x_i^{(\ell-1)}, t_i) - f(x_0, t_0) \\
  \vdots \\
  f(x_n^{(\ell-1)}, t_n) - f(x_0, t_0) \\
\end{bmatrix} \tag{2.1-8}
\]

with

\[
\chi^{(\ell)}(t) = x_0 + C^{(\ell-1)}_\ell t + \frac{1}{2} C^{(\ell-1)}_i t^2 + \cdots + \frac{1}{\nu+1} C^{(\ell-1)}_n t^{\nu+1} \tag{2.1-9}
\]

Remember that the initial function has been assumed to be

\[
\chi^{(1)}(t) \equiv x_0
\]

A better guess at \( \chi(t) \) can be made in some cases, with the result that fewer iterations are required to obtain convergence. A convenient check for convergence is

Largest value over all \( j \):

\[ \left| \chi^{(\ell)}_j - \chi^{(\ell-1)}_j \right| < \varepsilon \]

for appropriate \( \varepsilon \).

In the above explanation the curve fit followed by analytic integration [equations (2.1-8) and (2.1-9)] is known as Newton-Cotes integration. The process of repeated substitution of \( \{\chi^{(\ell)}_1, \ldots, \chi^{(\ell-1)}_n\} \) into \( f(x,t) \) to obtain

\[ \{\chi^{(\ell)}_1, \ldots, \chi^{(\ell)}_n\} \]

is an example of the method of successive substitutions. For brevity in later discussions, we denote this combination of old methods by the new name, SEMI-ANALYTIC APPROXIMATION (SAA). Note an interesting feature of SAA:

since each iteration describes \( \chi \) over the entire time span, the general expression for \( \chi^{(\ell)}(t) \) can be written as an expli-
cit, closed-form function of $X_0, t_e$ and $t_f$. Of course if "i" is large, this function will be extremely complicated.

An important question to be dealt with later is the accuracy of this method in obtaining solutions to actual nonlinear differential equations. The versions of SAA to be used are described in the following sections. They are more complicated than the simple example given above. Because of this complexity, these applications of SAA can only be checked for accuracy by comparison with numerical integration results. These comparisons are given in Appendix C.

2.2 Approximate Solution to the Evader Equations of Motion

William McFarland, of the Charles Stark Draper Laboratory, Inc., has worked with the equations of motion of a hypersonic re-entry body in the vertical plane. Following the approach of Busemann, et al, gravity is neglected (small compared to aerodynamic forces), an exponential atmosphere is assumed and the flight-path-angle is made the independent variable. The flat-earth assumption is also used. A constant angle of attack is postulated which, with the hypersonic assumption, forces the lift and drag coefficients to be constant in time. If

$$X = \text{range}$$
$$Z = \text{altitude}$$

---


\( \gamma = \text{flight-path-angle} \)
\( \rho = \text{air density} = \rho_0 e^{-\frac{\gamma}{\sigma}} \), \( \sigma = 24000 \text{ feet} \)
\( V = \text{velocity magnitude} \)
\( g = \text{acceleration due to gravity} \)
\( C_L, C_D, C_D^0 = \text{lift, drag and zero-lift drag coefficients, respectively} \)
\( A = \text{cone base area (aerodynamic reference area)} \)
\( m = \text{vehicle mass} \)
and \( \beta_0 = mg/(C_D^0 A) \) (the hypersonic ballistic coefficient), the vertical-plane equations of motion are
\[
\begin{align*}
\dot{x} &= V \cos \gamma \\
\dot{z} &= V \sin \gamma \\
\dot{V} &= -\left( \frac{3}{\rho_0} \right) \frac{C_L}{C_D^0} \left( \frac{F V^2}{2} \right) \\
\dot{\gamma} &= \left( \frac{3}{\rho_0} \right) \frac{C_L}{C_D^0} \left( \frac{F V}{2} \right)
\end{align*}
\]
(2.2.1)

The Evader dynamics are shown in Figure 2.2-1.

At this point, McFarland follows the lead of Busemann, et al, and defines
\[
\begin{align*}
\mathcal{U} &= \ln \left( \frac{V}{V_0} \right) \\
\mathcal{W} &= \left( \frac{\rho_0 \eta g}{2 \beta_0} \right) e^{-\frac{3}{\sigma}}
\end{align*}
\]
(2.2.2)

Going to \( \gamma \) as the independent variable,
\[
\frac{\dot{x}}{\dot{z}} = \frac{d^2 z}{dV^2} = \frac{3 V N \gamma}{(C_L/C_D^0) \left( \frac{2 \beta_0}{\beta_0} \right)}; \quad \frac{x}{\dot{z}} = \frac{dV}{dx} = \frac{C_D^0 \cos \gamma}{C_L} 2 \beta_0 \left( \frac{2 \beta_0}{\beta_0} \right) + \frac{V}{\dot{z}} = \frac{dV}{dx} = -\frac{C_D^0}{C_L} V.
\]

Using the above definitions he writes the complete set as
Figure 2.2-1 The Evader Model Dynamics
\[
\begin{align*}
\frac{dx}{d\gamma} &= \frac{\sigma C_{\infty}}{C_L} \frac{\cos \gamma}{W} \\
\frac{dW}{d\gamma} &= -\frac{C_{\infty}}{C_L} \sin \gamma \\
\frac{d\gamma}{d\xi} &= -\frac{C_{\infty}}{2C_L}
\end{align*}
\]

(2.2-3)

Defining

\[\alpha \equiv \left(\frac{C_L}{C_{\infty}}\right)W_0 - \cos \gamma_0\]

the solutions are

\[U(\gamma) = U_0 - \frac{C_{\infty}}{C_L} (\gamma - \gamma_0)\]

\[W(\gamma) = W_0 + \frac{C_{\infty}}{C_L} (\cos \gamma - \cos \gamma_0)\]

\[X(\gamma) = X_0 + \sigma \begin{cases} 
\gamma_0 - \delta + \frac{\alpha}{\sqrt{1 - \alpha^2}} \ln \left[\frac{(\sqrt{\frac{\alpha^2 + \tan^2 \frac{\delta}{2} + \alpha + 1}{\alpha^2 + \tan^2 \frac{\delta}{2} - \alpha - 1}}\sqrt{\frac{\alpha^2 + \tan^2 \frac{\delta}{2} - \alpha + 1}{\alpha^2 + \tan^2 \frac{\delta}{2} + \alpha + 1}}}\right] \\
\text{if } |\alpha| < 1 \\
\gamma_0 - \delta + \frac{2\alpha}{\sqrt{1 - \alpha^2}} \left[\tan^{-1}\left(\frac{\alpha^2 + \tan^2 \frac{\delta}{2}}{1 + \alpha^2}\right) - \tan^{-1}\left(\frac{\alpha^2 + \tan^2 \frac{\delta}{2}}{1 - \alpha^2}\right)\right] \\
\text{if } |\alpha| > 1
\end{cases}\]

(2.2-4)

Unfortunately, the independent variable is now \(\gamma\) and \(\gamma(t)\) is not known. The equation to be solved for \(\gamma\) is

\[\dot{\gamma} = k \left[ a + \cos \gamma \right] e^{\frac{C_{\infty}}{C_L} \gamma}\]

(2.2-5)

where

\[k \equiv -\frac{1}{\sigma} V_0 e^{(U_0 - \frac{C_{\infty}}{C_L} \gamma_0)}\]

and \(\sigma\) is previously defined. As far as the author has been
able to determine, equation (2.2-5) cannot be solved in closed form. Numerical integration can be employed with accuracy, but a faster and less complicated solution algorithm is desired for use in an interceptor guidance law. The SAA approach outlined in Section 2.1 satisfies the demands for simplicity and speed while providing acceptable accuracy in this application. First, we write the above equation in the notation of equation (2.1-1):

\[ \dot{\gamma} = f(\gamma, t) \], \[ \gamma(t_0) = \gamma_0 \]

(Of course, in this case \( f = f(\gamma) \) only). From numerical integrations of (2.2-5), it appears that a cubic polynomial with equally spaced data points will fit \( f(\gamma, t) \) to the desired accuracy. From knowledge of the speeds and distances involved in the intercepts for the assumed geometries, we estimate that the E vs. P encounter will be finished within six seconds after it begins. Thus,

\[ t_f = 6 \]

Other geometries can change this number, but for present purposes, we will use this one. Although the encounter starts at \( t = 0 \) by definition, the present time is called \( t_0 \) for purposes of deciding the appropriate intercept control to be used in the future. Referring to equations (2.1-2) and (2.1-3), \( n = 3 \) and

\[ f(\gamma, t) \equiv f(\epsilon, t) \equiv C_0 + C_1 \tau + C_2 \tau^2 + C_3 \tau^3 \quad (2.2-6) \]

where \( \tau = t - t_0 \) as in Section 2.1.
If the equally spaced time points are \( t_0, t_1, t_2 \) and 6 sec., it is required that

\[
t_1 \equiv t_0 + \frac{6-t_0}{3} \quad \text{and} \quad t_2 \equiv t_0 + \frac{2}{3}(6-t_0)
\]

Thus, from equation (2.1-4) we write

\[
C_0 = f(y_0, t_0)
\]

\[
\begin{bmatrix}
C_1 \\
C_2 \\
C_3
\end{bmatrix} = \begin{bmatrix}
T_1, T_1^2, T_1^3 \\
T_2, T_2^2, T_2^3 \\
T_3, T_3^2, T_3^3
\end{bmatrix}^{-1} \begin{bmatrix}
f(x_0, t_1) - f(x_0, t_0) \\
f(x_0, t_2) - f(x_0, t_1) \\
f(x_0, t_3) - f(x_0, t_2)
\end{bmatrix} = \begin{bmatrix}
\alpha \\
\alpha \\
\alpha
\end{bmatrix} \tag{2.2-7}
\]

where \( T_1 \equiv t_1 - t_0 = \frac{6-t_0}{3}, T_2 \equiv \frac{2}{3}(6-t_0) \) and \( T_3 \equiv (6-t_0). \)

The first estimate of \( \gamma(t) \) is

\[
\gamma^{(1)}(t) = y_0 + C_0 \tau \tag{2.2-8}
\]

in accord with (2.1-5). Finally, from equations (2.1-8) and (2.1-9), the \( \ell^4 \) estimate is generated by

\[
C_0^{(\ell^4)} = f(y_0, t_0)
\]

\[
\begin{bmatrix}
C_1^{(\ell^4)} \\
C_2^{(\ell^4)} \\
C_3^{(\ell^4)}
\end{bmatrix} = \begin{bmatrix}
T_1, T_1^2, T_1^3 \\
T_2, T_2^2, T_2^3 \\
T_3, T_3^2, T_3^3
\end{bmatrix}^{-1} \begin{bmatrix}
f(x_0^{(\ell^4)}, t_1) - f(x_0, t_0) \\
f(x_0^{(\ell^4)}, t_2) - f(x_0, t_1) \\
f(x_0^{(\ell^4)}, t_3) - f(x_0, t_2)
\end{bmatrix} \tag{2.2-9}
\]

with

\[
\gamma^{(2)}(t) = y_0 + C_0^{(\ell^4)} \tau + \frac{1}{2} C_1^{(\ell^4)} \tau^2 + \frac{1}{3} C_2^{(\ell^4)} \tau^3 + \frac{1}{4} C_3^{(\ell^4)} \tau^4 \tag{2.2-10}
\]

Note from equation (2.2-4) that \( \gamma(t) \) defines the entire planar trajectory.
The above analysis is invalid for the case when \( \dot{y} = 0 \) throughout the flight. The \( \ddot{y} \) equation becomes \( \ddot{y} = 0 \) in the equation set (2.2-1), leaving us with

\[
\dot{x} = V \cos y; \quad \dot{z} = V \sin y; \quad \dot{V} = -\left( \frac{g}{c_d} \right) \left( \frac{C_D}{2} \right) \left( \frac{L^2}{2} \right)
\]  

(2.2-11)

Recall the definition of \( W \) in the equation set (2.2-2): \( W = \frac{\sigma q}{2 \rho_o} \rho \).

Differentiating this and substituting for \( \dot{Z} \) yields

\[
\dot{W} = \frac{\sigma q}{2 \rho_o} \rho = -\frac{1}{\tau} WV \sin x_o
\]  

(2.2-12)

The \( W \) definition is solved for \( \rho \). Using this expression in the \( \dot{V} \) equation gives us

\[
\dot{V} = -\frac{1}{\tau} \frac{C_D}{2} W V^2
\]  

(2.2-13)

Equation (2.2-12) is now divided by equation (2.2-13) to give

\[
\frac{dW}{dV} = \left( \frac{C_D}{2} \right) \left( \sin x_o \right) \frac{1}{V}
\]  

(2.2-14)

with the solution

\[
W = W_o + \left( \frac{C_D}{2} \right) \sin x_o \ln \left( \frac{V}{V_o} \right)
\]  

(2.2-15)

Substituting this into equation (2.2-13) results in

\[
\dot{V} = -\frac{1}{\tau} \frac{C_D}{2} \left[ W_o + \left( \frac{C_D}{2} \sin x_o \right) \ln \left( \frac{V}{V_o} \right) \right] V^2
\]  

(2.2-16)

This equation is "solved" by the SAA technique in the same manner as equation (2.2-5). The resulting polynomial for \( V \) is substituted into the \( \dot{x} \) and \( \dot{z} \) equations above. The result is two quadratures for \( x \) and \( z \).
In practice SAA convergence has never required more than three iterations for either equation (2.2-5) or equation (2.2-16). This increases the desirability of writing a single, closed-form expression for the converged $\mathbf{y}$ or $\mathbf{v}$ as described at the end of the previous section.

2.3 Approximate Solution to the Pursuer Equations of Motion

The Busemann, Vinh and Kelley motivated approach seen in analyzing the evader equations of motion cannot be applied to the pursuer. Although it is possible to write the aerodynamic equations of a point-mass and change the independent variable from "t" to "$\mathbf{y}$", the resulting equations are not even partially amenable to closed-form solution. The reason for this is the presence of the thrust term in the $\dot{\mathbf{v}}$ equation. Accordingly, no change-of-variable is attempted in the following. First, we write the equations for the point-mass $\mathbf{p}$. (The effect of rotational dynamics in the autopilot loop is inserted later.) If $\mathbf{T}$ = engine thrust, $\mathbf{A}_l$ = net lateral acceleration in the negative lift direction, $\mathbf{D}$ = drag force, and all other symbols are defined as in Section 2.1, we have

\[
\begin{align*}
\dot{x} &= v \cos y \\
\dot{z} &= v \sin y \\
\dot{v} &= \frac{1}{M} (T - D) \\
\dot{\gamma} &= - \frac{A_l}{v}
\end{align*}
\]  

(2.3-1) (2.3-2) (2.3-3) (2.3-4)

Although a component of $\mathbf{T}$ does contribute to $\mathbf{A}_l$, the "cosine
effect" of a small angle between the thrust and wind directions allows the approximation to be made that the full \( T \) drives \( \dot{V} \) in equation (2.3-3).

The drag term is not expressed in terms of \( \rho, V, \) etc., because we want to compute the SAA approximation to the trajectory by fitting a polynomial to the \( D \) time history. If \( m_o = \) the mass of \( P \) at \( t=0 \) (launch), we write

\[
\frac{D}{m_o} = \Delta_L + \Delta_S t + \Delta_d t^2 \tag{2.3-5}
\]

for \( t \equiv t-t_o \) and \( t \in [t_o, \infty] \) as in Section 2.2. From Appendix A,

\[
m \approx m_o \left( 1 - 0.0016602 \left( 484 t + \frac{2.25 \times 10^5}{6} t^2 \right) \right) \]
\[
\approx m_o \left( 1 - 1.84 t - 0.02283 t^2 \right) \text{ slugs} \tag{2.3-6}
\]
\[
T = 1549.6 \left( 484 t + 27.5 t^2 + 0.396 t^2 \right) \text{ lbs.} \tag{2.3-7}
\]
\[
m_o = 233.1 \text{ slugs}
\]

Thus, from equation (2.3-3),

\[
\dot{V} = \frac{1}{m} (T-D) = \frac{1549.6 \left( 484 t + 27.5 t^2 + 0.396 t^2 \right) - m_o (\Delta_L + \Delta_S t + \Delta_d t^2)}{m_o \left( 1 - 0.8035 t - 0.02283 t^2 \right)}
\]

After some algebraic manipulation,

\[
V(t) = V_o + \int_{t_0}^{t} \frac{1}{m} (T-D) \, dt' = V_o - 438.60 (2.5968 - \Delta_d) (t-t_0)
\]
\[
+ \left[ -217.298 (182.81-\Delta_S + 2\Delta_d t_0) + 7733.1 (2.5968 - \Delta_d) \right] \gamma
\]
\[
+ \left[ 8.0115 (321.75 + \Delta_S t_0 - \Delta_L - \Delta_d t_0^2) - 141.25 (182.81
\]
\[-\Delta_S + 2\Delta_d t_0) + 8495.6 (2.5968 - \Delta_d) \right] \omega \tag{2.3-8}
\]
where
\[ Y = \ln \left( \frac{1 - 0.0804 t - 0.0228 t^2}{1 - 0.0504 t - 0.0228 t^2} \right) \]
\[ Z = \ln \left[ \frac{(-0.0456 t - 0.2052)(-0.0456 t + 0.4944)}{(-0.0456 t + 0.4944)(-0.0456 t - 0.4944)} \right] \]

We now have \( V(V_0, t_0; \Delta_L, \Delta_S, \Delta_S, t) \) as an approximate solution to (2.3-3) based on guessed values for \( \Delta_L, \Delta_S \) and \( \Delta_S \).

If we can obtain a closed-form approximation to equation (2.3-4) as well as to (2.3-3), we can integrate (2.3-1) and (2.3-2) directly. Thus, we now concentrate on (2.3-4). Using our newfound knowledge of \( V(t) \), we first attempt to integrate directly to obtain
\[ Y = Y_0 - \int_{t_0}^{t} \frac{A_L}{V(V_0, t_0; \Delta_L, \Delta_S, \Delta_S, t)} \, dt' \]

Much time spent working on this integral has convinced the author of the futility of this approach. Once again, we rely on a polynomial fit. Previous experience with this integrand indicates that a quadratic fit will suffice. We hypothesize that
\[ \frac{A_L(t)}{V(V_0, t_0; \Delta_L, \Delta_S, \Delta_S, t)} = \frac{5}{6} + \frac{7}{6} 2 + \frac{5}{2} 2^2 \quad (2.3-9) \]

for \( t \in [t_0, 6] \) and \( \tau \equiv t - t_0 \). Defining
\[ A_{\pm} \equiv A_c(t')/t' \]
\[ V_c \equiv V(V_o, t_o, \Delta c_1, \Delta c_2, \Delta c_3; t_c) \]
\[ t_c = \frac{t_o + t_c}{2}; t_2 = \Delta c_1; t_3 = \frac{t_o - t_c}{2}; t_4 = 2t_c \]

the values of \( \xi_{c_2} \) are found from

\[
\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \left[ \begin{array}{cc} 1 & t_2^2 \\ 2t_1 & (2t_2)^2 \end{array} \right]^{-1} \begin{pmatrix} A_{c_2} - A_{c_0} \\ V_{c_2} - V_{c_0} \end{pmatrix}
\]

(2.3-10)

with the result that

\[
\frac{A_c}{V} \approx \frac{A_{c_0}}{V_0} + \left[ \frac{2}{n_1} (\frac{A_{c_2}}{V_1} - \frac{A_{c_0}}{V_0}) - \frac{1}{2n_1} (\frac{A_{c_2}}{V_2} - \frac{A_{c_0}}{V_0}) \right] t
\]
\[ + \left[ \frac{1}{2n_1} (\frac{A_{c_2}}{V_2} - \frac{A_{c_0}}{V_0}) - \frac{1}{2n_1} (\frac{A_{c_2}}{V_1} - \frac{A_{c_0}}{V_0}) \right] t^2 \]

(2.3-11)

The approximation for \( \gamma(t) \) results directly:

\[
\gamma(t) = \gamma_0 - \int_{t_0}^{t} \frac{A_c(t')}{V(t')} \, dt'
\]

\[
\approx \gamma_0 - \frac{A_{c_0}}{V_0} \left[ \frac{t - \frac{3}{4n_1} t^2 + \frac{1}{6n_1^2} t^3}{4n_1^2} \right] - \frac{A_{c_1}}{V_1} \left[ \frac{t^2 - \frac{1}{3n_1} t^3}{3n_1^2} \right] - \frac{A_{c_2}}{V_2} \left[ \frac{t^2 + \frac{1}{6n_1} t^3}{6n_1^2} \right]
\]

(2.3-12)

At this point, we make a simplifying assumption about \( A_c(t) \):
This permits the control to be specified by two scalars, \( A_{\text{lo}} \) and \( A_{\text{lo}'} \), and brings within reasonable bounds of complexity the algorithm to pick a suitable interception control. (See Chapter 3.) Using this assumption in (2.3-12), we see that

\[
A_{\text{lo}1} = A_{\text{lo}} + A_{\text{lo}'} \tau,
\]

\[
A_{\text{lo}2} = A_{\text{lo}} + 2A_{\text{lo}'} \tau,
\]

so that

\[
\gamma(t) = \gamma_0 - A_{\text{lo}} \left[ \frac{1}{V_0} \left( t - \frac{3}{4} t^3 + \frac{1}{6} \frac{g}{g^2} t^3 \right) + \frac{1}{V_1} \left( \frac{1}{g^2} t^2 - \frac{1}{g^2} \frac{g^2}{g^2} t^3 \right) \right. \\
\left. + \frac{1}{V_2} \left( -\frac{1}{4} g^2 t^3 + \frac{1}{6} \frac{g^3}{g^2} t^3 \right) \right] - A_{\text{lo}'} \left[ \frac{1}{V_1} \left( \frac{1}{g^2} t^2 - \frac{1}{g^2} \frac{g^2}{g^2} t^{3/2} \right) + \frac{1}{V_2} \left( \frac{1}{g^2} t^2 - \frac{1}{g^2} \frac{g^2}{g^2} t^3 \right) \right]
\]

As mentioned earlier, the rotational dynamics of P have not been considered up to this point. We must correct (2.3-13) for their presence since they imply that the autopilot cannot deliver the desired \( A_L \) instantaneously. If we assume that \( V \leq V_e \) during the autopilot transient and define

\[
\eta(t) \equiv (A_L) \text{ commanded} - (A_L) \text{ delivered},
\]

it is easily shown that

\[
\gamma(t)_{\text{true}} \approx \gamma(t)_{\text{of (2.3-13)}} + \frac{1}{V_0} \int_{t_0}^{t_0 + \text{autopilot rise time}} \eta(t') \, dt'
\]

In Appendix D it is shown that a straight-line approximation to the autopilot transient yields an effective rise time of
\[ r_t = \frac{\frac{5}{\omega_n}}{1 - \frac{(A_{lo}-A_d)(\frac{5}{\omega_n})}{(A_{lo}-A_{lo})}(1-G_{HF})} \]

where

\( \omega_n \) = natural frequency of closed autopilot loop

\( A_{lo} \) = used prior to \( \tau_0 \)

\( A_{lo} \) = used prior to \( \tau_0 \), evaluated at \( \tau_0 \)

\( G_{HF} \) = high frequency closed-loop gain giving initial wrong-way autopilot response.

It is further shown in the same appendix that, given the above response,

\[
\int_{\tau_0}^{\tau_0 + \tau_t} \eta(t) dt = \frac{1}{2} \left( 1 - G_{HF} \right) \left( A_{lo} - A_{lo} \right) r_t \tag{2.3-15}
\]

Applying (2.3-14) and (2.3-15) to (2.3-13) yields the corrected approximation to \( \gamma(t) \).

\[
\gamma(t) \approx \gamma_0 - A_{lo} \left[ \int_0^t \left( \frac{1}{\tau_0} \left( -\frac{3}{2} \tau^2 + \frac{1}{5} \tau^3 \right) + \frac{1}{\tau_2} \left( \frac{2}{5} \tau^2 - \frac{1}{3} \tau^3 \right) \right] dt \right] - A_{lo} \left[ \int_0^t \left( \frac{2}{3} \tau^2 - \frac{1}{3} \tau^3 \right) \right] + \int_0^t \left( \frac{1}{2V_0} \left( 1 - G_{HF} \right) \left( A_{lo} - A_{lo} \right) \right) r_t \tag{2.3-16}
\]

Now, we are in a position to give approximate solutions to (2.3-1) and (2.3-2). If \( \gamma(t) \equiv \gamma(t) \) (equation 2.3-16) and \( V(t) \equiv V(t) \) (equation (2.3-8), we use quadratic fits to obtain

\[
V \cos \gamma \approx V_0 \cos \gamma_0 + \left[ \frac{1}{\tau_0} \left( V_1 \cos \gamma_1 - V_0 \cos \gamma_0 \right) - \frac{1}{2\tau_0^2} \left( V_2 \cos \gamma_2 - V_0 \cos \gamma_0 \right) \right] t + \left[ \frac{1}{2\tau_0^3} \left( V_2 \cos \gamma_2 - V_0 \cos \gamma_0 \right) - \frac{1}{4\tau_0^2} \left( V_1 \cos \gamma_1 - V_0 \cos \gamma_0 \right) \right] t^2
\]
and

\[ V_{\sin \theta} = V_0 \sin \theta_0 + \left[ \frac{2}{\eta_1} \left( V_1 \sin \theta_1 - V_0 \sin \theta_0 \right) - \frac{1}{\eta_1} \left( V_0 \sin \theta_0 - V_0 \sin \theta_0 \right) \right] T \\
+ \left[ \frac{1}{\eta_1} \left( V_2 \sin \theta_2 - V_0 \sin \theta_0 \right) - \frac{1}{\eta_1} \left( V_1 \sin \theta_1 - V_0 \sin \theta_0 \right) \right] T^2 \]

Integrating these yields

\[ X = x_0 + V_0 \cos \theta_0 \left( t - \frac{3}{2} \frac{1}{\eta_1} T^2 + \frac{1}{6 \eta_1} T^3 \right) + V_1 \cos \theta_1 \left( \frac{1}{\eta_1} T^2 - \frac{1}{3 \eta_1} T^3 \right) \\
+ V_2 \cos \theta_2 \left( \frac{1}{2 \eta_1} T^2 + \frac{1}{6 \eta_1} T^3 \right) \tag{2.3-17} \]

\[ Z = z_0 + V_0 \sin \theta_0 \left( t - \frac{3}{2} \frac{1}{\eta_1} T^2 + \frac{1}{6 \eta_1} T^3 \right) + V_1 \sin \theta_1 \left( \frac{1}{\eta_1} T^2 - \frac{1}{3 \eta_1} T^3 \right) \\
+ V_2 \sin \theta_2 \left( - \frac{1}{2 \eta_1} T^2 + \frac{1}{6 \eta_1} T^3 \right) \tag{2.3-18} \]

Reviewing the above results, we first assume a quadratic fit to the drag history with literal coefficients \( \Delta_l, \Delta_s \) and \( \Delta_k \). This permits us to integrate the velocity equation (2.3-8) and obtain \( V(V_0, t_0, \Delta_l, \Delta_s, \Delta_k, t) \). Proceeding to the equation for the flight-path angle, we use another quadratic fit to approximate the right-hand side of (2.3-4). Constraining \( A_l(t) \) to be composed of a bias \( A_{\alpha_0} \) plus a ramp of slope \( A_{\alpha_1} \), we integrate the quadratic fit to obtain (2.3-13). The effect of finite bandwidth of the P autopilot needs to be included, so we add a correction term to get (2.3-16). At this point we have

\[ \delta = \delta(V_0, t_0 ; V_1, V_2 ; \gamma_k, \gamma_s, \gamma_s, A_{\alpha_0}, A_{\alpha_1}, t) \]
or, ignoring the initial conditions and other constants and expanding \( V_1 \) and \( V_2 \),

\[
\dot{y} = \gamma(A_u, A_s, A_d; A_w, A_w'; t) \tag{2.3-19}
\]

The remaining equations for \( \dot{x} \) and \( \dot{z} \) are integrated using polynomial fits to the right-hand sides of equations (2.3-1) and (2.3-2). Since the estimates for \( V(t) \) and \( \gamma(t) \) are the only data used in these curve fits, \( x \) and \( z \) (ignoring constants as before) are now written as

\[
x = X(A_u, A_s, A_d; A_w, A_w'; t) \tag{2.3-20}
\]

\[
z = Z(A_u, A_s, A_d; A_w, A_w'; t) \tag{2.3-21}
\]

With estimates of \( x, z \) and \( V \) available, we are now in a position to compute the drag as a function of time and obtain estimates of \( A_u, A_s \) and \( A_d \).

Note from Section 2.1 that in the motivating discussion of SAA, a single polynomial fit is assumed in obtaining \( X^{(i)}(t) \) from \( X^{(i-1)}(t) \). In the above discussion the notion of SAA is expanded: the three curve fits are used in cascade to generate new trajectory estimates from the "input" variables \( A_u, A_s \) and \( A_d \). This procedure may introduce error in general unless the variables used in obtaining the curve fits are "well-behaved" as discussed in Section 2.1. Fortunately, prior experience with \( D(t) \), \( V(t) \) and \( \gamma(t) \) indicates that for the "bias plus ramp" P control assumed they are well represented by quadratic functions of time. The resulting predictions of P motion are accurate enough
to give us further confidence in the quadratic fit assumption (see Appendix C).

Although $X, Z, V, A_{\alpha \theta}$ and $A_{\alpha \theta}'$, are all the implicit variables needed to solve for $DL$, the drag is an explicit function of $Z, V$ and the angle-of-attack $\alpha$. The angle-of-attack is a function of $Z, V, A_{\alpha \theta}$ and $A_{\alpha \theta}'$. To solve for it we again examine the rotational dynamics of $P$. Let $C_L$ be the lift coefficient and $C_N$ and $C_M$ the normal force and moment coefficients, respectively; all other symbols are previously defined or seen in Figure 2.3-1. It is easily shown that the net acceleration in the negative lift direction, $A_L$, is found from

$$A_L = -\frac{1}{2} \rho \frac{V^2 A}{m} C_L - \frac{T}{m} \sin(\alpha - \delta) + g \cos \delta \quad (2.3-22)$$

$$= A_{\alpha \theta} + A_{\alpha \theta}' T \quad \text{(when the autopilot is in equilibrium)}$$

Furthermore, in equilibrium the aerodynamic moment tending to force $\alpha$ to zero must be balanced by the engine torque needed to generate a nonzero $\alpha$. Accordingly,

$$\frac{1}{2} \rho \frac{V^2 A}{I} (c_M - \frac{c_M}{c_N} d) C_N = \frac{T c_m}{I} \sin \delta \quad (2.3-23)$$

By the definition of $C_M$, $\frac{c_M}{c_N} d$ ( = distance between the cone base and the center of pressure) is a constant. However

*The moment coefficient $C_M$ is defined by the equation $M = \frac{1}{2} \rho V^2 d C_M A$ where $M$ is the moment about the cone base caused by the normal force acting through the center of pressure.
Figure 2.3-1 The Pursuer Model Dynamics
\( C_l \) and \( C_N \) are functions of \( \alpha \) and are given for \( P \) in Appendix A. For \( \alpha \) in the expected range,

\[
\begin{align*}
C_l &\approx 1.9824 \alpha - 0.40134 \ \text{SGN}(\alpha) \ \alpha^2 \\
C_N &\approx 1.9843 \alpha - 0.07297 \ \text{SGN}(\alpha) \ \alpha^2
\end{align*}
\]

(2.3-24) (2.3-25)

As seen in Appendix A, the thrust angle \( \delta \) is limited between \( \pm 5^\circ \). This allows a small-angle approximation to the sine function in (2.3-23), resulting in

\[
\delta = \frac{1}{2} \rho V^2 A \frac{C_m - \frac{C_N d}{\alpha}}{T m} C_N
\]

(2.3-26)

Using the same approximation on the sine function in (2.3-22) gives

\[
A_c = -\frac{1}{2} \frac{\rho V^2 A}{m} C_l - \frac{T}{m} (\alpha - \delta) + g \cos \gamma
\]

(2.3-27)

Now, we substitute (2.3-26) into (2.3-27) and use the expressions for \( C_l \) and \( C_N \) in (2.3-24) and (2.3-25). Collecting terms in \( \alpha^2 \), \( \alpha' \) and \( \alpha^0 \) we obtain a polynomial in \( \alpha \) alone:

\[
p_2 \alpha^2 + p_1 \alpha + p_0 = 0
\]

(2.3-28)

where

\[
\begin{align*}
p_2 &\equiv \frac{1}{2} \frac{\rho V^2 A}{m} \text{SGN}(\alpha) \left[ (1 - \frac{C_N d}{C_m}) (-0.07297) + 0.40134 \right] \\
p_1 &\equiv \frac{1}{2} \frac{\rho V^2 A}{m} \left[ 1.9843 (1 - \frac{C_N d}{C_m}) - 1.9824 \right] - \frac{T}{m} \\
p_0 &\equiv g \cos \gamma - A_c
\end{align*}
\]

The solution to this polynomial gives the value of \( \alpha \) which
provides \( A_L \), the negative of the desired net lift, and also stabilizes the vehicle from pitch rotation. Physical requirements for this stabilization indicate that \( \text{sgn}(\alpha) = -\text{sgn}(A_L) \). Thus,

\[
p_2 = -\frac{1}{2} \frac{\rho V^2 A}{m} \text{sgn}(A_L) \left[ -0.07297 (1 - \frac{\beta}{\gamma_{an}}) + 0.0137 \right] \tag{2.3-29}
\]

The solution to (2.3-28) is

\[
\alpha = -\frac{p_1}{2p_2} + \text{sgn}(A_L) \sqrt{\left( \frac{p_1}{2p_2} \right)^2 - \frac{p_0}{p_2}} \tag{2.3-30}
\]

where the \( \text{sgn}(A_L) \) term dictates the sign of \( \alpha \) in accordance with the above stabilization requirements.

The solution for \( \Delta_L, \Delta_S \), and \( \Delta_\alpha \) now follows directly. From Appendix A we have

\[
C_D = 0.01519 + 0.001854 \text{sgn}(\alpha) \alpha + 2.9114 \alpha^2 \tag{2.3-31}
\]

The drag force is \( \frac{1}{2} \rho V^2 A C_D \). Thus,

\[
\frac{D}{m_0} = \frac{1}{2} \rho V^2 A \left( \frac{0.01519 + 0.001854 \text{sgn}(\alpha) \alpha + 2.9114 \alpha^2}{m_0} \right) \tag{2.3-32}
\]

Fitting this function with the polynomial given in (2.3-5), the solution for \( \Delta_L, \Delta_S \), and \( \Delta_\alpha \) is

\[
\begin{align*}
\Delta_L &= \left. \frac{D}{m_0} \right|_{t_e} \\
\left[ \Delta_S \right] &= \left[ \frac{T_1}{27}, \frac{T_1^2}{(27)^2} \right]^{-1} \left[ \left. \frac{D}{m_0} \right|_{t_e}, - \left. \frac{D}{m_0} \right|_{t_0} \right] \\
\left[ \Delta_\alpha \right] &= \left[ \frac{T_1}{27}, \frac{T_1^2}{(27)^2} \right]^{-1} \left[ \left. \frac{D}{m_0} \right|_{t_e}, - \left. \frac{D}{m_0} \right|_{t_0} \right]
\end{align*}
\tag{2.3-33}
\]

where \( \left. \frac{D}{m_0} \right|_{t_1} = \frac{1}{2} \rho V^2 A \left( \frac{0.01519 + 0.001854 \text{sgn}(\alpha) \alpha + 2.9114 \alpha^2}{m_0} \right) \)
\[ \rho' = \text{air density at altitude } Z', \]
\[ \alpha' = \alpha(h') \]

We now have \( A_1, A_2 \) and \( A_c \) as functions of \( \rho_1, \rho_2, \rho_3, V_0, V_1, V_2, \alpha(A_1, \rho_1), V_0, \delta \), \( \alpha(A_2, \rho_2), V_1, \delta \), and \( \alpha(A_c, \rho_3), V_2, \delta \). With our previous knowledge about \( V, \delta \) and \( A_1, A_2 \), we can write out the functional dependencies on variables changing each iteration.

\[ \Delta_L = \Delta_L(A_1, \rho_1) \]
\[ \Delta_S = \Delta_S(\Delta_L, \Delta_S, \Delta_c; A_1, A_1') \]
\[ \Delta_c = \Delta_c(\Delta_L, \Delta_S, \Delta_c; A_1, A_1') \]

Given \( A_1 \) and \( A_1' \), these dependencies permit us to establish an SAA cycle to approximately solve the \( P \) equations of motion.

If \( A_1 \) and \( A_1' \) are changed, the SAA algorithm must be rerun to obtain a new approximate solution. Extensive experience with this algorithm shows convergence in usually less than three iterations. As seen in Appendix C, the accuracy in predicting the true trajectory is acceptable for the present purposes.
CHAPTER 3
INTERCEPTION TECHNIQUES

3.0 Summary

The three interception techniques to be studied are described in this chapter. The first scheme considered is proportional navigation (PN), a classical technique useful as approximate base-line for putting the other schemes in perspective. A basic weakness of PN is the neglect of evader or pursuer accelerations, a deficiency the other schemes attempt to avoid. The Minimum-Miss-Time (MMT) and Least Risk (LR) pursuit steering laws both predict accelerated P motion forward in time, differing in their estimates of E motion. In both these cases E is assumed to be using maximum angle-of-attack at all times. LR simultaneously considers predicted "pull-up" and "pull-down" maneuvers and steers P to intercept both trajectories with one pass. The MMT law simplifies this approach by assuming a straight-line E trajectory, continuing on with the present flight-path-angle and drag associated with maximum angle-of-attack. Although such a trajectory could only arise from rapid oscillation between maximum positive and negative angle-of-attack, it is a useful compromise between a pull-up and a pull-down maneuver.

3.1 Proportional Navigation

Although we have developed some powerful prediction tools
in Chapter 2, we are first going to discuss an interception steering law which makes no use of them. Proportional navigation is a well known technique; its application to missile interception has been discussed in numerous technical articles.\textsuperscript{7,8,9} The pursuer and the evader are assumed to be flying at constant speed and heading with intercept geometry as illustrated in Figure 3.1-1. The commanded negative-lift-direction acceleration $A_L$ is chosen to have a component perpendicular to the line of sight (l.o.s.) tending to drive the l.o.s. rate $\phi$ to zero. If the closing velocity $V_C$ becomes large, it is desirable to have $A_L$ respond with high sensitivity to $\phi$. Thus, the gain is made proportional to $V_C$; the desired steering law is

$$A_L \cos(\gamma-\phi) = -\lambda V_C \dot{\phi}$$

or

$$A_L = - (\lambda V_C \dot{\phi}) / \cos(\gamma-\phi) \quad (3.1-1)$$


$\gamma$ = flight-path-angle
$\phi$ = line-of-sight angle

$A_\perp \cos(\gamma - \phi)$ is applied along the ⊥ to line-of-sight
to drive $\phi$ to zero.

Steering Law: $A_\perp \cos(\gamma - \phi) = -\lambda V_\perp \phi$

where $V_\perp$ = closing velocity
$\lambda$ = navigation constant, chosen = 2

Figure 3.1-1 Proportional Navigation Intercept Geometry
where $\lambda$ is the navigation ratio, set = 2 here.*

The geometrical assumptions behind proportional navigation are severely violated in the hypersonic interception scenario to be discussed in Chapter 5. Due to aerodynamics, the E velocity changes rapidly even if no evasive maneuver is made. Furthermore, P is driven by an engine of $100+9$ thrust acceleration capability. Both of these factors introduce strong relative accelerations to which proportional navigation cannot immediately respond, since proportional navigation is derived from constant velocity assumptions. However, as a simple and often-used technique, proportional navigation provides a baseline for evaluating the effectiveness of more sophisticated techniques to be discussed in later sections.

3.2 Minimum-Miss-Time Guidance

Due to the basic nature of SAA prediction we are always left with an approximate analytic expression for the predicted trajectory as a function of the control. As seen in Sections 2.2 and 2.3, the $X(t)$ and $Z(t)$ approximations for the P trajectory are functions of $A_{4\omega}$ and $A_{6\omega}$. If the E (for an assumed

*Studies of the optimality of PN in references 8 and 9 conclude that $\lambda \geq 3$ (except very near the intercept). However, the most sophisticated of these studies assumes only a second-order, constant-coefficient autopilot. The present autopilot, described in Appendix A, is fourth-order with coefficients rapidly varying in time. Determining the optimal $\lambda$ in this case is beyond the scope of the thesis, if it can be done at all. The values of $\lambda = 3$ and 4 with this autopilot give severe instability in simulations so $\lambda = 2$ is chosen as a useful value.
$C_L$ and $C_D$ trajectory) is obtained for the same time span as $P$, the approximate miss distance $d_{\text{miss}}$ and miss time $t_{\text{miss}}$ can be calculated. The $P$ and $E$ positions are evaluated at four equally-spaced time points between the present time and $t=6.\ast$
Comparing the first three points with the last three, a choice is made as to which set is more desirable for interpolating the time and distance of closest approach. This decision made, a parabolic curve fit of position vs. time is made to each set of three points for $P$ range, $P$ altitude, $E$ range and $E$ altitude. With these functions the analytic expression for $P-E$ distance vs. time is computed. Finding the minimum of this expression yields the distance $(d_{\text{min}})$ and time $(t_{\text{min}})$ of closest approach. Since the $d_{\text{min}}$ and $t_{\text{min}}$ are estimated from functions of $A_{00}$ and $A_{00}'$, they themselves are functions of $A_{00}$ and $A_{00}'$. Accordingly, with all $P$ and $E$ initial conditions held constant, we can compute

$$g_{d_{\text{min}}} = \frac{d_{d_{\text{min}}}}{dA_L}$$

$$g_{t_{\text{min}}} = \frac{d_{t_{\text{min}}}}{dA_L}$$

with $A_L \equiv \left[ \frac{A_{00}}{A_{00}'} \right]$. These gradients are analytic functions of all parameters, controls and initial conditions composing the $P$ and $E$ trajectory approximations.

\ast Recall that in Section 2.3 the 6 sec. point is considered to be a practical upper bound on the intercept time.
Minimum-Miss-Time (MMT) guidance makes use of $d_{min}$ and $t_{min}$ to steer P to an interception of E. Given the present position and velocity of E, it is possible to use the results of Section 2.2 to predict a region of space through which E is most likely to fly. Referring to Figure 3.2-1, the ($\pm 30^\circ$) maximum and minimum angles of attack for E cause a hard pull-up and a steep dive which form a fan-shaped region. It is intrinsic in the interception problem that P cannot know which trajectory E will fly in this region. Lacking such knowledge, we assume a straight-line, no lift E path as a compromise. To account for the probability that E will do some evasive maneuver and thus lose an extra amount of velocity, we make the drag equal to the value seen when $\alpha = \pm 30^\circ$. Assuming that E follows the no-lift trajectory from the present moment on, it is possible to adopt a two-element policy. The present choice of $(A_{lo}, A_{lo})$ produces a unique $d_{min}$ and $t_{min}$ between the P and E trajectories. We want to minimize $d_{min}$ to bring P within warhead kill-radius of E. We choose to also minimize $t_{min}$ so that E is intercepted as soon as possible. The earlier the intercept time, the more narrow the fan-shaped region about the no-lift trajectory. This width partly indicates the error due to our inability to predict the true E trajectory. Minimizing this error increases the chances of interception.

Summarizing the above ideas, MMT guidance seeks to minimize the unconstrained cost
Figure 3.2-1 P vs E Intercept Scenario Showing Possible E Trajectories
\[ J = \frac{1}{2} d_{\min}^2 + p t_{\min} \]  
\[ \text{(3.2-3)} \]

where \( P \) is a penalty function.* The minimization is performed against the no-lift evader trajectory using a simple gradient search. The cost gradient for the search is

\[ \frac{\partial J}{\partial A_x} = d_{\min} g_{\min} + p g_{t_{\min}} \]  
\[ \text{(3.2-4)} \]

and the stopping condition is

\[ \| \frac{\partial J}{\partial A_x} \| < \varepsilon \]

for an appropriate \( \varepsilon \). At regular time intervals in the flight this process is repeated to obtain a new optimal \( (A_{\infty}, A_{\infty}^\prime) \) point. The result is a piecewise-continuous control history guiding \( P \). The search process is seen in Figure 3.2-2.

Note the advantages to be found in MMT guidance compared to proportional navigation. The interception geometry can be entirely general with less degradation in accuracy due to strong relative acceleration and/or highly non-straight-line flight paths. Also, the SAA prediction includes the effect of the autopilot transient. The obvious deficiency in MMT of non-continuous control is removed if the \( (A_{\infty}, A_{\infty}^\prime) \) update occurs sufficiently often. (Here, the guidance is updated four times each second). However, both PN and MMT are postulating particular aerodynamic coefficients for \( E \) (PN assumes \( C_e = C_d = 0 \)). Thus, both laws suffer if \( E \) has far different values.

*The square of the miss distance is used to avoid the slope discontinuity seen in \( d_{\min}(A_x) \) near \( d_{\min} = 0 \).
Figure 3.2-2: Steps in Obtaining the MMT Response at $t_0 = 3.5$ seconds

Note: Beside each step label is the predicted miss against the predicted E trajectory.
3.3 Least Risk Guidance

In Section 3.2 SAA trajectory-prediction is used to minimize the miss distance and miss time between the P flight path and the zero-lift E flight path. Figure 3.2-1 shows the region of likely E trajectories, bounded by the maximum and minimum angle-of-attack (+30°) cases. Least Risk (LR) guidance goes one step further than MMT in steering P with respect to this region. The two +30° boundaries are treated as two possible E trajectories to be intercepted by one P pass. (See Figure 3.3-1.) Specifically, LR guidance computes $d_{min}^+$ and $d_{min}^-$ and steers P to minimize the maximum of the two miss distances. By steering as close as possible to both boundaries, P seems more likely to intercept any given path in between -- thus, the name "Least Risk". This control allows the pursuer to guard against the most aerodynamically extreme maneuvers the evader is capable of.

To obtain the minimax of the ($d_{min}^+$, $d_{min}^-$) pair, a "twin gradient" minimization scheme is used. Assuming an $A_e$, the process seen in Section 3-2 for finding $d_{min}$ is used for the +30° trajectory to obtain $d_{min}^+$, $A_{min}^+$ and then for the -30° trajectory to find $d^-_{min}$, $A_{min}^-$. With this data the larger of the two $d_{min}$ values is selected and labelled $d_{min}^+$; the smaller is called $d_{min}^-$. The desired change in cost is $\Delta MN$ (a negative number). In deciding whether to drive down $d_{min}^+$ by itself or $d_{min}^+$ and $d_{min}^-$ together, it is important to have an estimate of the
Figure 3.3-1 Least Risk Interception With P Driving the $(d_{min}, d_{max})$ Pair to a Minimax
effect of each action. Thus, the $d_{\text{min}}^L$-reducing trial step

$$\delta A_L = \text{ADMIN} \frac{1}{\langle d_{\text{min}}^L, d_{\text{min}}^L \rangle} d_{\text{min}}^L$$  \hspace{1cm} (3.3-1)

is computed. If this step is applied, the results to first order are

$$d_{\text{min}}^L \rightarrow d_{\text{min}}^L + \text{ADMIN} = (d_{\text{min}}^L)_{\text{new}}$$

$$d_{\text{min}}^S \rightarrow d_{\text{min}}^S + \langle d_{\text{min}}^S, \delta A_L \rangle = (d_{\text{min}}^S)_{\text{new}}$$

If

$$(d_{\text{min}}^L)_{\text{new}} > (d_{\text{min}}^S)_{\text{new}}$$

stepping to reduce $d_{\text{min}}^L$ alone is justified and the trial $\delta A_L$ in (3.3-1) is used. In case the inequality does not hold, a step is taken to drive down both costs. The desired result is

$$d_{\text{min}}^L \rightarrow d_{\text{min}}^L + \text{ADMIN}$$

$$d_{\text{min}}^S \rightarrow d_{\text{min}}^L + \text{ADMIN}$$

so that, to first order, both distances are driven to a common lower level. The step to obtain this result is

$$\delta A_L = \left[ \begin{array}{c} (d_{\text{min}}^L)^T \\ (d_{\text{min}}^S)^T \end{array} \right] \frac{1}{\text{ADMIN}} \left[ \begin{array}{c} \text{ADMIN} \\ (d_{\text{min}}^L + \text{ADMIN}) - d_{\text{min}}^L \end{array} \right]$$  \hspace{1cm} (3.3-2)

Thus, the algorithm drives down the larger cost whenever there is a substantial size difference, but, if $d_{\text{min}}^L$ and $d_{\text{min}}^S$ are "close" in the above sense, the step is taken to reduce both
to a common value. Two stopping conditions are used which reflect the two possible minimax solution points. First, the algorithm stops whenever $d_{min}^L$ reaches a relative min, i.e. whenever

$$
\| g_{min} \| < \varepsilon_1
$$

The second solution point is not at a relative min, but is characterized by

$$
d_{min}^s = d_{min}^L
$$

and

$$
< g_{min}^s, g_{min}^L > = -k
$$

for $K$ some positive constant. In other words the two gradients point in opposite directions so that any attempt at driving down either cost will raise the other above its current minimax value. The stopping conditions corresponding to this situation are

$$
| d_{min}^L - d_{min}^s | < \varepsilon_2
$$

and

$$
\frac{< g_{min}^L, g_{min}^s >}{\| g_{min}^L \| \| g_{min}^s \|} < -(1-\varepsilon_3)
$$

The search is seen in Figure 3.3-2 for the case of $t_0=3.5$ sec.

As with MMT guidance, the LR guidance algorithm is repeated at regular intervals to update the P control function. The intervals are as close together as computation costs allow.
Figure 3.3-2: Steps in Obtaining the LR Response at $t_0 = 3.5$ seconds

Note: Beside each step label is the predicted miss against the pull-up and pull-down predicted trajectories, respectively.
LR represents the most sophisticated steering law to be treated in this work. Not only does it use SAA techniques to accurately predict possible P and E trajectories, it deals with the region of likely E flight paths. As will be shown in Chapter 5, the critical stages of the intercept from the P viewpoint are the first few seconds after launch. In this time span the P guidance must decide the general direction in which P must fly to be in the best position for a successful interception. Later, due to the rapid acceleration of P the general direction of flight cannot be changed before E is passed and the pursuit-evasion conflict is over. In this light LR guidance is seen to be a more complete approach: its consideration of the region of future E flight paths is intended to lessen the effect of surprise maneuvers by E. However, the same weakness is apparent in LR as in the other schemes. The two E trajectories are for particular assumed aerodynamics. Though other E maneuvers fall within the fan-shaped region of Figure 3.3-1, their positions at any given time are not as well defined. The results in Chapter 5 present a successful low-drag maneuver which violates the high-drag assumptions used in obtaining the region boundaries.
CHAPTER 4
OPTIMAL EVASION

4.0 Summary

In this chapter the optimum seeking routines are developed for evading interception by the three pursuit steering laws. The evasion problem of avoiding interception while minimizing terminal impact error is defined. Then, the rationale for using the gradient projection optimum seeking method is presented, followed by derivations of the required function-space gradients. The implementation of gradient projection is explained as used here. A conjugate gradient approach for faster terminal convergence is also developed. (Small constraint violations arising from the small steps taken are removed by corrections in the post-intercept time). Finally, the state equation and its partial derivatives with respect to control and state are presented, noting the particular features associated with each interception law.

4.1 Searching for the Optimum: Modified Gradient Projection and Conjugate Gradient Methods

In optimizing the Evader performance two goals are sought simultaneously:

a. force the intercept miss distance (miss) to be as large as possible, and

b. keep the impact error (IE) on the ground as small as possible.
A challenging aspect of the evasion problem is that satisfying these two criteria is not a trivial task. If it were not important to hit the ground target, E could fly intricate maneuvers and "run away" until P ran out of fuel. On the other hand a simple maneuver dropping square on the target would almost certainly be intercepted. Trial and error methods for solving this problem would probably require much computation time and more than a little luck to achieve even passable results. A systematic search technique is obviously a preferred approach.

Examination of the physical geometry of the problem reveals some features which tend to make the Gradient Projection Method a likely choice. Experience has shown that the IE is much more sensitive to changes in the E control function $\alpha$ than is the miss. The reason is intuitively apparent from Figure 4.1-1 and knowledge of the pursuit goal. In this example a constant positive $\delta\alpha$ is added to $\alpha(\gamma)$ over the time of flight. Since the intercept altitude is reached sooner than ground impact, there is less time for the perturbation in specific force resulting from $\delta\alpha$ to be integrated to a change in horizontal position. There is generally increased drag at lower altitudes due to increased air density so that, as seen in the examples of Chapter 5, the time from intercept altitude to impact is often two or three times the time from start to

---

a) Effect on Range at Intercept Altitude

b) Effect on IE at the Ground

Note: "Intercept Altitude" is known only approximately, but the concept is useful for purposes of illustration.

Figure 4.1-1 Effects of Constant Perturbation in $\alpha_h$ on the Miss and IE.
intercept altitude. The pursuit goal of minimizing the miss also contributes to decreasing the miss sensitivity to \( \alpha \); the pursuer tends to adapt to changing E maneuvers and counter the increase in miss otherwise obtainable. Finally, in the present problem it is desireable to drive the miss into the thousands of feet whereas the IE should be in hundreds of feet. All of these facts point to a steep-sided ravine problem resulting from applying the penalty function approach to consider the zero IE constraint. (In fact, this method was used initially with discouraging results.) The other technique applicable in the initial phase of this problem is the gradient-projection method, where steps are taken which, to first order, deal with constraints independently from cost.

The present problem is formulated as follows. We desire to find \( \alpha(t) \) to maximize

\[
\phi(x(t)) = \text{miss}
\]

subject to the constraint

\[
y(x(t)) = IE = \sigma
\]

and the system equations of motion

\[
\dot{x} = f(x, \alpha, t); \ x(t_0) = x_0
\]

where

\[
x(t) = \text{total state vector including E and P states at time } t.
\]

\[
t_0 = \text{start time for the problem (fixed)}
\]

\[
t_i = \text{intercept time}
\]

\[
t_f = \text{final impact time}
\]
\[ \chi_0 = \text{assumed initial condition at } t_0. \]

It is advantageous to divide the problem into two parts -- one concerned with maximizing \( \phi \) and the other with minimizing \( \psi \).

First we want to find the functional gradient of \( \phi \) with respect to \( \alpha \). We write

\[ \Theta(x(t)) = \phi(x(t)) + \int_{t_0}^{t_1} (f(x, \alpha, t) - \bar{x}) \, dt \]  \hspace{1cm} (4.1-2)

where \( \lambda_\phi \) is an unspecified function of \( t \) and the integral term must be identically zero. The perturbation of this equation is

\[ d\Theta(x(t)) = (\phi_x s_x + \phi f_{,x} \, dt) \bigg|_{t=t_i} + \lambda_{\phi}^T (f(x, \alpha, t) - \bar{x}) \bigg|_{t=t_i} \]

\[ + \int_{t_0}^{t_i} \left[ \lambda_{\phi}^T (f_x s_x) + f_{,\alpha} s_x + f_{,x} s_{\alpha} - \bar{x} \right] \, dt \]  \hspace{1cm} (4.1-3)

Note that

\[ \int_{t_0}^{t_i} \lambda_{\phi}^T s_x \, dt = [\lambda_{\phi}^T s_x]_{t_0}^{t_i} - \int_{t_0}^{t_i} \lambda_{\phi}^T s_x \, dt \]  \hspace{1cm} (4.1-4)

Substituting \( \dot{x} = f(x, \alpha, t) \) and equation (4.1-4) into (4.1-3) gives us

\[ d\Theta(x(t)) = \left[ (\phi_x - \lambda_{\phi}^T) s_x + \phi f_{,x} f_{,t} \right] \bigg|_{t=t_i} \]

\[ + \int_{t_0}^{t_i} \left[ (\lambda_{\phi}^T f_x + \lambda_{\phi}^T s_x + \lambda_{\phi}^T f_{,x} s_{\alpha}) dt + (\lambda_{\phi}^T s_x) \right]_{t_0}^{t_i} \]  \hspace{1cm} (4.1-5)
The last term is identically zero because \( \chi(t_0) = X_o \) is given. A stopping condition must be introduced to determine when \( t_i \) occurs. The derivative of the distance between \( P \) and \( E \) is used since it goes to zero at \( t_i \). Labelling this quantity \( \Sigma \), we have

\[
\Sigma (x(t)) \equiv 0
\]  \hspace{1cm} (4.1-6)

The stopping condition is used to obtain a relationship between the differentials of \( x(t) \) and \( t_i \), since

\[
\Sigma (x(t)) \equiv 0 \implies d\Sigma (x(t)) = 0
\]

Thus,

\[
d\Sigma = \sum_x (\delta x(t) + f dt_i) = 0
\]

or

\[
 dt_i = -\frac{\sum_x \delta x(t)}{\sum_x f} \bigg|_{t=t_i}
\]  \hspace{1cm} (4.1-7)

We now use this relationship to express that part of the perturbation at time \( t_i \) solely as a function of \( \delta x(t) \).

\[
\delta \theta(\chi(t)) = \left[ -\frac{\lambda_T^{\phi} f_x + \phi_x - \left( \frac{\lambda_T^{\phi} f}{\sum_x f} \right) \sum_x f x \right] \delta x(t) \]

\[
+ \int_{t_0}^{t_i} \left[ \left( \lambda_T^{\phi} f_x + \lambda_T^{\phi} \right) \delta x + \lambda_T^{\phi} f_x \delta x \right] dt \]  \hspace{1cm} (4.1-8)

Let a new definition be stated. Let

\[
\lambda_T^{\phi} (x, \alpha, t) \equiv \sum_x f (x, \alpha, t). \]

(4.1-9)

Since \( \lambda_T^{\phi} = \lambda_T^{\phi}(t) \),
\[ H_{\phi_x} = \lambda_\phi^T f_x \quad \text{and} \quad H_{\phi_\alpha} = \lambda_\phi^T f_\alpha \]

We substitute the relations into the above to obtain

\[ d\Theta(x(t_i)) = \left[-\lambda_\phi^T + \phi_x - \left(\frac{\partial f_x}{\partial x_\alpha}\right)\Sigma_x \right] \mid_{t=t_i}^t S_x(t_i) \]

\[ + \int_{t_0}^{t_i} \left[ (H_{\phi_x} + \lambda_\phi^T) S_x + H_{\phi_\alpha} S_\alpha \right] dt \quad (4.1-10) \]

With \( \lambda_\phi(t) \) an arbitrary function of time, we let

\[ \lambda_\phi(t_i) = \left[ \phi_x - \left(\frac{\partial f_x}{\partial x_\alpha}\right)\Sigma_x \right]^T \mid_{t=t_i} \quad (4.1-11) \]

\[ \dot{\lambda}_\phi = -H_{\phi_x} \quad (4.1-12) \]

and obtain

\[ d\Theta(x(t_i)) = \int_{t_0}^{t_i} H_{\phi_\alpha} S_\alpha dt \]

Since the integral term of (4.1-2) is identically zero,

\[ d\Theta(x(t_i)) = \Theta(x(t_i)) \]

\[ = \int_{t_0}^{t_i} H_{\phi_\alpha} S_\alpha dt \quad (4.1-13) \]

Now, the inner product in the function space defined on \([t_0, t_i]\) is defined by the following identity\(^{11}\)

\[ \langle a, b \rangle \equiv \int_{t_0}^{t_i} a(t)b(t) dt \quad (4.1-14) \]

Moreover, the general gradient $\frac{\partial \mathcal{C}}{\partial u}$ of cost $\mathcal{C}(u)$ with respect to $u$ is defined by\textsuperscript{*}

$$d\mathcal{C} = \langle \frac{\partial \mathcal{C}}{\partial u}, du \rangle \quad (4.1-15)$$

Using equation (4.1-14) and 4.1-15) to interpret (4.1-13) we can write

$$d\phi(x(t)) = \langle H_{t_0}, dx \rangle \quad (4.1-16)$$

and the functional gradient of $\phi$ with respect to $x$ is

$$\Phi = \{ H_{t_0}(t) \mid t \in [t_0, t_i] \} \quad (4.1-17)$$

where $H_{t_0}(t)$ for each $t$ value is considered a different component of the function space vector $\Phi$. To summarize these results, the function space gradient $\Phi$ is computed by solving the following set of equations for a given $\alpha(t)$:

$$\dot{X} = f(x, \alpha, t) ; \quad X(t_0) = x_0$$

$$\Sigma(X(t)) = 0 \quad (to\;obtain\;t_i)$$

$$\Phi = -H_{t_0}^T \dot{X} \lambda = [\Phi - \left(\frac{\partial \Phi}{\partial X_0}\right)^T \Sigma x] |_{t=t_i} \quad (4.1-18)$$

$$\Phi = \{ H_{t_0}(t) \mid t \in [t_0, t_i] \}$$

where

$$H_{t_0}(X, x, \alpha, t) \equiv \frac{\partial^T}{\partial \alpha} f(x, \alpha, t)$$

Now, the function-space gradient of $\psi$ with respect to $x$ is found by reference to the previous derivation. Note that $\psi(x(t))$ and $\phi(x(t))$ are both functions of $X$ but at the dif-

\textsuperscript{*}By interpretation of theorem 1' on p. 125 of Brockett's book previously referenced.
different times $t_f$ and $t_i$. The stopping condition for determining $t_f$, the time of impact, is that the altitude of $E$ goes to zero. We denote this stopping condition

$$\Omega(X(t)) = 0 \quad (4.1-19)$$

As with $\Psi$ and $\phi$, $\Omega(X(t))$ and $\Sigma(X(t))$ are functions of $X$ only, although at different times. The same system equation of motion is used for both $\Psi$ and $\phi$, although the time spans are $[t_0, t_f]$ and $[t_0, t_i]$, respectively. Finally, the initial condition $X_o$ is the same. Defining a general function $\lambda = \lambda(\mathbf{x}, t)$ and the function $H_\Psi = \lambda^T f(x, \mathbf{t})$ and repeating the previous derivation, we arrive at a similar set of equations for $\Psi$. Thus, given some $a(t)$, we solve the following:

$$\dot{X} = \frac{\partial f(x, \mathbf{t})}{\partial x}; \quad X(t_0) = X_0$$

$$\Omega(X(t)) = 0 \quad \text{(to obtain } t_f)$$

$$\Sigma_\Psi = -H_{\Psi_x}; \quad \Delta \Psi(t_f) = \left[ \Psi_X - \frac{\partial f(X,t)}{\partial X} \right]_t = t_f$$

$$\Psi = \begin{Bmatrix} H_{\Psi_x}(t) \mid t \in [t_0, t_f] \end{Bmatrix} \quad (4.1-20)$$

Remember that $[t_0, t_f] \supset [t_0, t_i]$.

We are now in a position to discuss the present use of the gradient projection method. In essence, a step is taken which, to first order, simultaneously:

a. reduces the constraint function $\Psi$ by a desired amount $\Delta \Psi_{\text{red}}$

and

b. steps orthogonal to $\Psi$ so as to complete the increase
in φ by the desired amount. The first part is achieved by stepping in the direction of -gφ.

Letting

\[ \Delta \alpha_\psi = \text{step along the } -g_\psi \text{ direction } = -C_\psi g_\psi \]

where \( C_\psi \) is a step-length constant, the resulting change in constraint function is

\[ \Delta \psi_{\text{pred}} = \langle g_\psi, \Delta \alpha_\psi \rangle = -C_\psi \langle g_\psi, g_\psi \rangle \]  \hspace{1cm} (4.1-21)

Thus,

\[ C_\psi = -\frac{\Delta \psi_{\text{pred}}}{\langle g_\psi, g_\psi \rangle} \]  \hspace{1cm} (4.1-22)

The component of \( g_\phi \) which is orthogonal to \( g_\psi \) is given by

\[ g_{\phi, \text{orth}} = g_\phi - \langle g_\phi, g_\psi \rangle g_\psi \]  \hspace{1cm} (4.1-23)

where \( g_\psi \) is the unit length direction of \( g_\psi \). Since

\[ g_\psi = \frac{g_\psi}{||g_\psi||} \]

we write that

\[ g_{\phi, \text{orth}} = g_\phi - \frac{\langle g_\phi, g_\psi \rangle}{\langle g_\psi, g_\psi \rangle} g_\psi \]  \hspace{1cm} (4.1-24)

We choose to step in this direction by the amount \( C_\phi \). This step does not effect \( \psi \). Thus,

\[ \Delta \alpha_{\phi, \text{orth}} = \text{step along the } g_{\phi, \text{orth}} \text{ direction } = C_\phi g_{\phi, \text{orth}} \]

The total step is

\[ \Delta \alpha = \Delta \alpha_\psi + \Delta \alpha_{\phi, \text{orth}} \]
or
\[ \Delta \alpha = -c_\psi g_\psi + c_\phi g_\phi \text{ORTHogonal} \]  \hspace{1cm} (4.1-25)

We want this step to increase \( \phi \) by \( \Delta \phi_{\text{PRED}} \). Thus, we set
\[ \Delta \phi_{\text{PRED}} = \langle g_\phi, \Delta \alpha \rangle = -c_\psi \langle g_\psi, g_\psi \rangle + c_\phi \langle g_\phi, g_\phi \rangle \text{ORTHogonal} \]

Substitution for \( c_\psi \) from (4.1-22) and \( g_\phi \text{ORTHogonal} \) from (4.1-24) gives
\[ \Delta \phi_{\text{PRED}} = \Delta \psi_{\text{PRED}} \frac{\langle g_\psi, g_\psi \rangle}{\langle g_\psi, g_\psi \rangle} + c_\phi \left\{ \frac{\langle g_\psi, g_\psi \rangle}{\langle g_\psi, g_\psi \rangle} - \frac{\langle g_\psi, g_\psi \rangle^2}{\langle g_\psi, g_\psi \rangle} \right\} \]

from which there results
\[ c_\phi = \frac{\Delta \phi_{\text{PRED}} - \Delta \psi_{\text{PRED}} \frac{\langle g_\psi, g_\psi \rangle}{\langle g_\psi, g_\psi \rangle}}{\langle g_\psi, g_\psi \rangle - \frac{\langle g_\psi, g_\psi \rangle^2}{\langle g_\psi, g_\psi \rangle}} \]  \hspace{1cm} (4.1-26)

If use of equation (4.1-26) gives a negative number for \( c_\phi \), this quantity is set to zero. The explanation for this policy is that negative \( c_\phi \) implies that the \( c_\psi \) length step of \( \Delta \alpha \) is itself sufficient to generate more than \( \Delta \phi_{\text{PRED}} \) change in \( \phi \).

To summarize the above, the following step is taken:
\[ \Delta \alpha = -c_\psi g_\psi + c_\phi \left( g_\phi - \frac{\langle g_\psi, g_\psi \rangle}{\langle g_\psi, g_\psi \rangle} g_\psi \right) \]  \hspace{1cm} (4.1-27)

where
\[ c_\psi = -\frac{\Delta \psi_{\text{PRED}}}{\langle g_\psi, g_\psi \rangle} \]

and
\[ C \phi = \frac{\Delta \phi_{\text{PRE}} - \Delta \psi_{\text{PRE}}}{\langle g_{\phi} g_{\phi} \rangle - \langle g_{\psi} g_{\psi} \rangle} \]

or 0, whichever is larger.

Note that the particular choices of \( \Delta \psi_{\text{PRE}} \) and \( \Delta \phi_{\text{PRE}} \) remain to be specified. Here, we choose

\[ \Delta \psi_{\text{PRE}} = -IE \]

in an effort to drive the impact error to zero on each step. The desired increase in miss \( \Delta \phi_{\text{PRE}} \) is chosen based on the local nonlinearity seen in the latest step. This nonlinearity is measured using the quantity

\[ \eta_{k+1} = \frac{|\Delta \phi_{\text{PRE},k} - \Delta \phi_{k}|}{|\Delta \phi_{\text{PRE},k}|} \]

The quantity \( \Delta \phi_{k} \equiv \phi_{k} - \phi_{k-1} \) is the achieved change in miss between the \( k-1 \)th and \( k \)th step. The following decision rule is used for the \( k+1 \)th step:

\[
\text{If} \quad \eta_{k+1} \begin{cases} 
\geq .8 & \text{then make } \Delta \phi_{\text{PRE},k+1} = \frac{1}{2} \Delta \phi_{\text{PRE},k} \\
\leq .2 & \text{then make } \Delta \phi_{\text{PRE},k+1} = 2 \Delta \phi_{\text{PRE},k} \\
& \text{but } .2 < \eta_{k+1} < .8 & \text{then make } \Delta \phi_{\text{PRE},k+1} = \Delta \phi_{\text{PRE},k} 
\end{cases}
\]

Note that a "moderate" amount of nonlinearity causes no change in the desired increase. This rule tends to keep \( \eta \) in the middle range as some nonlinearity indicates desirable progress in the optimum search. "Too much" nonlinearity may cause
negative $\Delta \phi$, however, and is to be avoided. When $\Delta \phi$ goes negative, the step is rejected and a half-step is taken using the same gradient information.

Thus far in this section the gradient projection method used appears totally conventional in its concept and its execution. However, the term "modified" appears in the title and this word refers to a feature not mentioned before.

Remember the set inclusion statement $[t_o, t_f] \supset [t_o, t_i]$. This statement refers to the fact that the function-space gradient $g_\psi$ is defined in a space of higher dimension or "time of greater duration" than the gradient $g_\phi$. This implies that it is possible to search in the subspace $[t_i, t_f]$ of $[t_o, t_f]$, where $g_\phi$ is not defined but $g_\psi$ is. Thus, the constraint $\psi$ can be reduced, in general, without changing the miss $\phi$.

This property is used to advantage after an acceptable (increasing $\phi$) projected gradient step is taken. Specifically, in this case the resultant IE is checked for size. If it is larger in magnitude than an acceptable amount (900 feet here), a "cleanup" step is taken using those components of $g_\psi$ outside of $[t_o, t_i]$. These steps are repeated until IE is sufficiently small. Physically, $E$ is made to try different post-intercept maneuvers until one reaches a suitable small IE magnitude. Usually, one "cleanup" step is sufficient.

Near the optimum, the Modified Gradient Projection Method becomes unsatisfactory. Even though the $\Delta \phi_{req}$ values used
become small (less than 100 feet in some cases), numerous steps are taken which decrease the miss distance and must be rejected. The conjugate gradient method is then used. This search algorithm generates estimates of second-order information from gradient values and uses these estimates to step with greater accuracy and success near the optimum. The algorithm starts at the latest, "best" control point with a search in the local gradient direction \( g_{\phi_0}(t) \). Thus, the initial search direction \( d_0(t) \) is defined by the equation
\[
d_0(t) = g_{\phi_0}(t)
\]
The control along this direction is defined by the equation
\[
\alpha(t) = \alpha_0(t) + C_0 d_0(t)
\]
and \( C_0 \) is varied until that value \( C_0^* \) is found giving the largest miss. This control is labelled \( \alpha_f ( = \alpha_0 + C_0^* d_0 ) \). The next search direction \( d_i \) uses \( \alpha_f \) and the local gradient \( g_{\phi_i} \). For the Kth iteration the pertinent quantities are
\[
\alpha_K = \alpha_{K-1} + C_{K-1} d_{K-1}
\]
and
\[
d_K = g_K - \frac{\langle g_K, g_K \rangle}{\langle g_{K-1}, g_{K-1} \rangle} d_{K-1}
\]

---


+ Searching as few as four values of \( C_0 \) to find \( C_0^* \) is excessively expensive, consuming 2-4 min. of IBM 370 machine time for each value. Instead, a single step is taken and a cubic interpolation is made for \( C_0^* \) using the miss and the miss derivative, projected along \( d_f \), at each end of the step.
where $C_{k-1}^*$ is the "best" $C_{k-1}$ value.

In practice four iterations are usually sufficient to come within 30 feet of the optimal. Due to the expense of the runs, this is considered to be close enough.

4.2 The State Equation and Its Derivatives

A description of the optimum-seeking algorithm is presented in the previous section without mentioning the elements of the state equations

$$\dot{X} = \mathbf{f} (X, \alpha, t)$$

or defining the state vector $X$. As shown in this section, the $\mathbf{f}$ vector contains a complete description of the P and E dynamics as well as the particular interception scheme being used. The reader should now turn to Appendix A for background material on the P and E equations of motion. Each particular interception policy is discussed in one of the next three sections along with the unique effect of each scheme on the $\mathbf{f}$ vector.

The state vector is composed of the states of E and the states of P as follows

$$\chi = \begin{bmatrix}
X_E \\
\dot{X}_E \\
S \\
x_1 \\
x_2 \\
x_3 \\
x_0 \\
\dot{X}_0
\end{bmatrix}
\begin{array}{c}
\{ \text{4 E states} \\
\{ \text{12x1} \} \\
\{ \text{8 P states} \}
\end{array}$$

(4.2-1)
As seen in Appendix A,

\[ \dot{X}_E = E \text{ position vector}; \]
\[ \dot{X}_{DOT} = \frac{d}{dt} X_E = E \text{ velocity vector}; \]
\[ \dot{X}_P = P \text{ position vector}; \]
\[ \dot{X}_{DOT_P} = P \text{ velocity vector}. \]

For convenience later we define two subvectors \( X_E \) and \( P_A \), separating terms appropriate to \( E \) from those of \( P \).

\[
\begin{bmatrix}
X_E \\
\dot{X}_{DOT_E}
\end{bmatrix}
= 
\begin{bmatrix}
X_E \\
\dot{X}_{DOT_E}
\end{bmatrix}
\begin{pmatrix}
4 \\
1
\end{pmatrix}
\]

\[
\begin{bmatrix}
\dot{X}_P \\
\dot{X}_{DOT_P}
\end{bmatrix}
= 
\begin{bmatrix}
\dot{X}_P \\
\dot{X}_{DOT_P}
\end{bmatrix}
\begin{pmatrix}
8 \\
1
\end{pmatrix}
\]

The \( \dot{f} \) vector is now written, again reference Appendix A.

\[
\begin{bmatrix}
\dot{X}_E \\
\dot{X}_{DOT_E}
\end{bmatrix}
= 
\begin{bmatrix}
\dot{X}_E \\
\dot{X}_{DOT_E}
\end{bmatrix}
\begin{pmatrix}
1 - \frac{\text{LEVER}}{2 \text{ EMAS}} \rho_{e} - \frac{X_{DOT_E}}{\text{EMAS}} \\
-157.1 \delta + 157.1 (-C_3(A_{\text{POSAT}} - A_L) + C_4 X_3 - C_2 X_2)
\end{pmatrix}
\]

\[
\begin{bmatrix}
\dot{X}_P \\
\dot{X}_{DOT_P}
\end{bmatrix}
= 
\begin{bmatrix}
\dot{X}_P \\
\dot{X}_{DOT_P}
\end{bmatrix}
\begin{pmatrix}
Q_3 \sin(\delta) - Q_4 \alpha_P \\
-C_3 (A_{\text{POSAT}} - A_L) + C_4 \dot{X}_{DOT_P}
\end{pmatrix}
\]

(Note that the \( \dot{X}_{DOT_E} \) expression here is equivalent to the one in equation (A-7).)
where

\[ \alpha_{SAT} = \text{E control after limiting} \]

\[ \text{EAREA}, \text{PAREA} = \text{E and P cone reference areas} \]

\[ C_{AE}, C_{AP} = \text{E and P axial force coefficients} \]

\[ C_{NE}, C_{NP} = \text{E and P normal force coefficients} \]

\[ \text{EMASS}, \text{PMass} = \text{E and P masses} \]

\[ g = \text{Magnitude of acceleration due to gravity} = 32.17 \text{ feet/sec}^2 \]

\[ \rho_0 = \text{Air density factor} = 0.02377 \text{ slug/ft}^3 \]

\[ \alpha_p, \gamma_p = \text{P angle-of-attack and flight-path-angle} \]

\[ A_L = \text{The achieved acceleration in the negative lift direction} \]

\[ \Lambda_{DGM} = \text{The desired negative-lift-direction acceleration after limiting} \]

\[ \phi_{SAT} = \text{The achieved engine angle in the body frame after limiting} \]

\[ \text{THRUST} = \text{The engine thrust} \]

\[ \alpha_1, \alpha_3, \alpha_3, \alpha_4 = \text{Time-varying autopilot parameters.} \]

\[ X_E = (X_{E1}, X_{E2}); \quad \dot{X}_E = (\dot{X}_{E1}, \dot{X}_{E2}); \quad Y_p = (Y_{p1}, Y_{p2}); \quad \dot{Y}_p = (\dot{Y}_{p1}, \dot{Y}_{p2}) \]

Note in the above that the first two vector equations, expressing

\[ \dot{X}_E = \begin{bmatrix} \dot{X}_E \\ \dot{Y}_E \end{bmatrix} \]

are themselves functions only of \( X_E \) and the control \( \alpha_{SAT} \). The last six equations, four scalar and two vector, are functions of \( P_A \) and \( \Lambda_{DGM} \) since the other variables \( \alpha_p, \gamma_p, \phi_{SAT}, A_L \) are themselves functions of \( P_A \). The desired negative-lift-direction acceleration (after limiting) \( \Lambda_{DGM} \) is the input to the autopilot from the interception steering
law. Through this law, \( x_E, x_\rho \) and \( x_{dot\rho} \), determine \( A_{losa-t} \). Thus, \( \rho \) is functionally related to \( x_E \) because of the interception policy.

Two derivatives of the \( f \) vector are used in the previous section. We now derive the expression for the first of these, \( \dot{f}_x \). For convenience, the derivatives of each component of \( f \) are taken separately with respect to \( x_E \) and \( \rho \). Since \( x_{dotE} \) is part of the state vector \( x \), \( \frac{2}{x_E^2} \) is trivial to evaluate and is not considered further. No derivatives of parts of \( x \) are derived below. Since \( x_{dotE} \) is part of \( x_E \), \( \frac{1}{x_{dotE}} x_{dotE} = 0 \). The second vector component (\( \dot{x}_{dotE} \)) of \( f \) is differentiated with less ease.

\[
\frac{d}{dx_E} \dot{x}_{dotE} = \frac{1}{2} \frac{E_{max}}{Emass} \rho e \frac{2E_2}{x_E^3} \frac{x_{dotE}}{\|x_{dotE}\|} \begin{bmatrix}
-x_{dotE} \sin(\gamma_{sat}) - x_{dotE} \cos(\gamma_{sat}) + x_{dotE} \cos(\gamma_{sat}) \\
x_{dotE} \cos(\gamma_{sat}) - x_{dotE} \sin(\gamma_{sat}) - x_{dotE} \sin(\gamma_{sat}) \\
-x_{dotE} \cos(\gamma_{sat}) + x_{dotE} \sin(\gamma_{sat})
\end{bmatrix}
\]

\[
x = \begin{bmatrix}
\dot{x}_{dotE} \\
-x_{dotE} \frac{2x_{dotE}}{24000} + \frac{x_{dotE}}{\|x_{dotE}\|^2} \frac{2}{x_E^2} x_{dotE} \left( + \frac{1}{x_{dotE}} \frac{d}{dx_E} x_{dotE} \right)
\end{bmatrix}
\]

\[(4.2-4)\]

Since \( \dot{x}_{dotE} \) is composed of \( \gamma_{sat} \) and components of \( x_E \), \( \frac{d}{\rho} \dot{x}_{dotE} = 0 \). Before continuing, we note the equation defining the achieved negative-lift-direction acceleration \( A_L \) in Appendix A.

\[
A_L = -G \sin(\rho - \gamma_{sat}) + G \cos \rho - G \frac{x}{2} \alpha \rho \quad (4.2-5)
\]
From the same source we note that

\[
\begin{align*}
\alpha_p &= \tan^{-1} \left( \frac{X \Delta T_{p,2}}{X \Delta T_{p,1}} \right) \\
\xi_{3 \text{at}} &= \left( \frac{\delta}{5^\circ \text{ in rad.}} \right) \tanh \left( \frac{\delta}{5^\circ \text{ in rad.}} \right)
\end{align*}
\]

Differentiating with respect to \(X \Delta T\) and \(\alpha_p\) gives

\[
\frac{d}{dx_{\Delta T}} \alpha_p = \frac{d}{dx_{\Delta T}} \xi_{3 \text{at}} = 0 \quad (4.2-7)
\]

and

\[
\begin{align*}
\frac{d}{d\alpha_p} \alpha_p &= \frac{1}{\|X \Delta T_{p,1}\|^2} \left( X \Delta T_{p,1} \frac{d}{d\alpha_p} X \Delta T_{p,2} - X \Delta T_{p,2} \frac{d}{d\alpha_p} X \Delta T_{p,1} \right) \\
\frac{d}{d\alpha_p} \xi_{3 \text{at}} &= \frac{1}{\left( \cosh \left( \frac{\delta}{5^\circ \text{ in rad.}} \right) \right)^2} \frac{d}{d\alpha_p} \delta
\end{align*}
\]

The derivatives of \(A_L\) can now be written

\[
\begin{align*}
\frac{d}{dx_{\Delta T}} A_L &= 0 \\
\frac{d}{d\alpha_p} A_L &= -Q_4 \cos(\alpha_p - \xi_{3 \text{at}}) \left( \frac{d}{d\alpha_p} \alpha_p - \frac{d}{d\alpha_p} \xi_{3 \text{at}} \right) - 9 \sin(\alpha_p) \frac{d}{d\alpha_p} \delta - Q_2 \frac{d}{d\alpha_p} \delta
\end{align*}
\]

We are now in a position to finish differentiating \(f\).

\[
\begin{align*}
\frac{d}{dx_{\Delta T}} f &= -157.1 C_3 \frac{d}{dx_{\Delta T}} A_{\text{sat}} \\
\frac{d}{d\alpha_p} f &= -157.1 \frac{d}{d\alpha_p} f - 157.1 C_3 \left( \frac{d}{d\alpha_p} A_{\text{sat}} - \frac{d}{d\alpha_p} A_L \right) + 157.1 C_3 \frac{d}{d\alpha_p} A_{\text{sat}}^3 - 157.1 C_3 \frac{d}{d\alpha_p} A_{\text{sat}}^2 \frac{d}{d\alpha_p} A_{\text{sat}} \frac{d}{d\alpha_p} A_{\text{sat}} \\
&\quad + 157.1 C_4 \frac{d}{d\alpha_p} X^3 - 157.1 C_4 \frac{d}{d\alpha_p} X^2
\end{align*}
\]

\((4.2-10)\)
The terms \( \frac{\partial}{\partial x_k} A_{LDSAT} \) and \( \frac{\partial}{\partial \alpha_{PA}} A_{LDSAT} \) relate to the particular interception law used. They are discussed in the next three sections. The derivatives of \( \dot{x}_1 \) are \( \frac{\partial}{\partial x_k} x_2 = 0 \) and \( \frac{\partial}{\partial \alpha_{PA}} x_2 \). Differentiating \( \dot{x}_2 \) gives us

\[
\begin{align*}
\frac{\partial}{\partial x_k} x_2 &= 0 \\
\frac{\partial}{\partial \alpha_{PA}} x_2 &= Q_3 \cos(S_{SAT}) \frac{\partial}{\partial \alpha_{PA}} S_{SAT} - Q_1 \frac{\partial}{\partial \alpha_{PA}} \alpha_P \quad (4.2-11)
\end{align*}
\]

Also,

\[
\frac{\partial}{\partial x_k} x_3 = -C_3 \frac{\partial}{\partial x_k} A_{LDSAT}; \quad \frac{\partial}{\partial \alpha_{PA}} x_3 = -C_3 \left( \frac{\partial}{\partial \alpha_{PA}} A_{LDSAT} - \frac{\partial}{\partial \alpha_{PA}} A_L \right) \quad (4.2-12)
\]

The derivatives of \( \dot{x}_p \) are \( \frac{\partial}{\partial x_k} \dot{x}_p = 0 \) and \( \frac{\partial}{\partial \alpha_{PA}} \dot{x}_p = 0 \).

The derivative of \( \dot{x}_{DOTP} \) is rather lengthy. (The reader will hopefully forgive the following plethora of algebra: six computer statements encompassing 46 lines of MAC language code are required to express it.) First, we define some shorthand notation:

\[
\begin{align*}
C_{PA} &= \cos(\alpha_P); \quad S_{PA} = \sin(\alpha_P); \quad C_{2PA} = \cos(2\alpha_P); \quad S_{2PA} = \sin(2\alpha_P) \\
C_{PADS} &= \cos(\alpha_P-S_{SAT}); \quad C_{DS} = \cos(S_{SAT}); \quad S_{DS} = \sin(S_{SAT}) \\
S_{PGA} &= \sin(\alpha_P); \quad S_{PAPC} = \sin(\alpha_P+\alpha_P); \quad C_{PAPC} = \cos(\alpha_P+\alpha_P) \quad (4.2-13)
\end{align*}
\]

Also, we need to differentiate the air density, velocity magnitude and normal and axial force coefficients for \( P \). Let

\[
\rho_P \equiv \rho_0 e^{-\frac{N_P^2}{2 \sigma_0}}, \quad \text{and} \quad \text{PVELO} \equiv || \dot{x}_{DOTP} || \quad (4.2-14)
\]

with \( C_{N_P} \) and \( C_{A_P} \) defined earlier. Then
\[
\frac{1}{\Delta E} \beta = 0; \quad \frac{\partial}{\partial P} \frac{\beta}{\Delta A} - \frac{1}{29000} \rho P \frac{2}{\Delta A} \chi_{p,2} \]
\[
\frac{\partial}{\partial E} \Delta E = 0; \quad \frac{\partial}{\partial E} \Delta E = \frac{1}{\rho E} \frac{\partial}{\partial E} \Delta E \frac{\partial}{\partial E} \Delta E \frac{\partial}{\partial E} \Delta E
\]
\[
\frac{\partial}{\partial E} \Delta E = 0; \quad \frac{\partial}{\partial E} \Delta E = 2 \Delta E \sin(\beta) - \frac{\partial E}{\Delta A} \sin^2(\beta) \frac{\partial E}{\Delta A} \sin^2(\beta)
\]
\[
\frac{\partial}{\partial E} \Delta E = 0; \quad \frac{\partial}{\partial E} \Delta E = 2 \Delta E \cos^2(\beta) \frac{\partial E}{\Delta A} \cos^2(\beta)
\]

where \( \beta \) is the cone vertex angle of \( \rho \). The derivatives of \( \Delta E \) follow:
\[
\frac{\partial}{\partial E} \Delta E \frac{\partial}{\partial E} \Delta E = 0
\]
\[
\frac{\partial}{\partial E} \Delta E \frac{\partial}{\partial E} \Delta E = \frac{\text{THRU} \text{ST}}{\text{PMASS}} \left[ \begin{array}{c}
\text{CPA}_E, -\text{SPA}_E \\
\text{SPA}_E, \text{CPA}_E
\end{array} \right] \left[ \begin{array}{c}
\text{SDS} \\
\text{OPS}
\end{array} \right] \frac{\partial}{\partial E} \left( \Delta E \cos^2(\beta) - \Delta E \sin^2(\beta) \right)
\]
\[
- \frac{1}{2} \frac{\partial}{\partial E} \frac{\rho E}{\Delta A} \left[ \begin{array}{c}
\text{CPA}_E, -\text{SPA}_E \\
\text{SPA}_E, -\text{CPA}_E
\end{array} \right] \frac{\partial}{\partial E} \Delta E \frac{\partial}{\partial E} \Delta E
\]
\[
- \frac{\rho E}{\Delta A} \text{PAREA} \left[ \begin{array}{c}
\text{CPA}_E, -\text{SPA}_E \\
\text{SPA}_E, -\text{CPA}_E
\end{array} \right] \frac{\partial}{\partial E} \Delta E \frac{\partial}{\partial E} \Delta E
\]
\[
- \frac{1}{2} \rho E \frac{\rho E}{\Delta A} \text{PAREA} \left[ \begin{array}{c}
\text{CPA}_E, -\text{SPA}_E \\
\text{SPA}_E, -\text{CPA}_E
\end{array} \right] \frac{\partial}{\partial E} \Delta E \frac{\partial}{\partial E} \Delta E
\]

These expressions are as compact as possible without losing a clear representation of the functional dependencies. We now
summarize these results in a more readable form.

As a convenience, we decompose the \( \mathbf{f} \) vector into two parts, separating E and P states. Thus,

\[
\mathbf{f} = \begin{bmatrix}
\dot{x}_E \\
\dot{x}_{\text{dot} E} \\
x_1 \\
x_2 \\
x_3 \\
\dot{x}_P \\
\dot{x}_{\text{dot} P}
\end{bmatrix} \\
\equiv \begin{bmatrix}
\mathbf{f}_E \\
\mathbf{f}_P
\end{bmatrix}_{(4 \times 1)}
\]

with

\[
f_E \equiv \begin{bmatrix}
\dot{x}_E \\
\dot{x}_{\text{dot} E}
\end{bmatrix} \quad \text{and} \quad f_P \equiv \begin{bmatrix}
\dot{x}_1 \\
x_2 \\
x_3 \\
\dot{x}_P \\
\dot{x}_{\text{dot} P}
\end{bmatrix}
\]  \hspace{1cm} (4.2-17)

From equations (4.2-10) and (4.2-12) recall that \( f_P \) depends (via \( A_{LPSAT} \)) only on \( x_E \) in the states \( s \) and \( x_3 \). Thus, we write

\[
\frac{\partial f_P}{\partial x_E} = \begin{bmatrix}
\frac{\partial f_P}{\partial x_E} \\
0 \\
0 \\
\frac{\partial f_P}{\partial x_3} \\
\frac{\partial f_P}{\partial \dot{x}_P} \\
0
\end{bmatrix}_{(8 \times 4)} \hspace{1cm} (4.2-18)
\]

where \( 0 \) denotes an \( 8 \times 4 \) matrix of zeroes. Collecting previous derivative equations into the appropriate groups, we obtain

\[
\frac{\partial f_E}{\partial x_E} = \begin{bmatrix}
\frac{\partial f_E}{\partial x_E} \\
\frac{\partial f_E}{\partial \dot{x}_{\text{dot} E}}
\end{bmatrix}_{(4 \times 4)} \hspace{1cm} (4.2-19)
\]
\[
\frac{\partial f_e}{\partial \alpha} = \begin{bmatrix}
\frac{\partial}{\partial \alpha} \dot{v}_e \\
\frac{\partial}{\partial \alpha} \dot{x}_{1e} \\
\frac{\partial}{\partial \alpha} \dot{x}_{2e} \\
\frac{\partial}{\partial \alpha} \dot{x}_{3e} \\
\frac{\partial}{\partial \alpha} \dot{x}_{pe} \\
\frac{\partial}{\partial \alpha} \dot{X}_{dotp}
\end{bmatrix} = 0 \quad (4.2-20)
\]

\[
\frac{\partial f_p}{\partial \alpha} = \begin{bmatrix}
\frac{\partial}{\partial \alpha} \dot{g} \\
\frac{\partial}{\partial \alpha} \dot{x}_1 \\
\frac{\partial}{\partial \alpha} \dot{x}_2 \\
\frac{\partial}{\partial \alpha} \dot{x}_3 \\
\frac{\partial}{\partial \alpha} \dot{x}_{pe} \\
\frac{\partial}{\partial \alpha} \dot{X}_{dotp}
\end{bmatrix} \quad (8 \times 8) \quad (4.2-21)
\]

and the desired \( f_x \) matrix follows:

\[
f_x = \begin{bmatrix}
\frac{\partial f_e}{\partial x_e} & 0 \\
\frac{\partial f_p}{\partial x_e} & \frac{\partial f_p}{\partial \alpha}
\end{bmatrix} \quad (12 \times 12) \quad (4.2-22)
\]

The other derivative of interest is \( \frac{\partial f}{\partial \alpha} \). Since \( \alpha \) is the control for \( E \) and chosen independently from the state,

\[
\frac{\partial f}{\partial \alpha} = \frac{\partial}{\partial \alpha} \begin{bmatrix}
\dot{v}_e \\
\dot{x}_{1e} \\
\dot{x}_{2e} \\
\dot{x}_{3e} \\
\dot{x}_{pe} \\
\dot{X}_{dotp}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial}{\partial \alpha} \dot{v}_e \\
\frac{\partial}{\partial \alpha} \dot{x}_{1e} \\
\frac{\partial}{\partial \alpha} \dot{x}_{2e} \\
\frac{\partial}{\partial \alpha} \dot{x}_{3e} \\
\frac{\partial}{\partial \alpha} \dot{x}_{pe} \\
\frac{\partial}{\partial \alpha} \dot{X}_{dotp}
\end{bmatrix} \quad (4.2-23)
\]
Recall from equation (4.2-3) that \( \dot{X}^{\text{DOT_E}} \) is a function of \( \alpha_{\text{SAT}} \).

The relationship between \( \alpha \) and \( \alpha_{\text{SAT}} \) is, from Appendix A:

\[
\alpha_{\text{SAT}} = \left(30^\circ \text{ in rad} \right) \tanh \left( \frac{\alpha}{\left(30^\circ \text{ in rad} \right)} \right) \quad (4.2-24)
\]

Thus, the differential relationship is

\[
\delta \alpha_{\text{SAT}} = \frac{1}{\cosh^2 \left( \frac{\alpha}{\left(30^\circ \text{ in rad} \right)} \right)} \delta \alpha \quad (4.2-25)
\]

Finally,

\[
\frac{d}{d\alpha} \dot{X}^{\text{DOT_E}} = -\frac{1}{2} \text{EAREA} \text{FCE} \frac{X_e \dot{e}}{\|X^{\text{DOT_E}}\|} \left[ \frac{\partial (\text{CNS} + \text{CAC})}{\partial \alpha_{\text{SAT}}} \right] \frac{1}{\partial \alpha_{\text{SAT}}} \left( \frac{\partial (\text{CNC} - \text{CAS})}{\partial \alpha_{\text{SAT}}} \right) \frac{d}{d\alpha} \dot{X}^{\text{DOT_E}} \quad (4.2-26)
\]

where

\[
\text{CNS} \equiv G_a E \sin (\alpha_{\text{SAT}}); \quad \text{CAC} \equiv G_a E \cos (\alpha_{\text{SAT}})
\]

\[
\text{CAS} \equiv G_a E \sin (\alpha_{\text{SAT}}); \quad \text{CAC} \equiv G_a E \cos (\alpha_{\text{SAT}})
\]

and

\[
\frac{\partial}{\partial \alpha_{\text{SAT}}} \text{CNS} = 1.973 \sin (\alpha_{\text{SAT}}) + \text{CNC}
\]

\[
\frac{\partial}{\partial \alpha_{\text{SAT}}} \text{CAC} = 1.973 \cos (\alpha_{\text{SAT}}) - \text{CNS}
\]

\[
\frac{\partial}{\partial \alpha_{\text{SAT}}} \text{CAS} = 1.357 \sin (\alpha_{\text{SAT}}) \sin (\alpha_{\text{SAT}}) + \text{CAC}
\]

\[
\frac{\partial}{\partial \alpha_{\text{SAT}}} \text{CAC} = 1.357 \sin (\alpha_{\text{SAT}}) \cos (\alpha_{\text{SAT}}) - \text{CAS}
\]
The three interception schemes have one characteristic in common: they compute the desired negative-lift-direction-acceleration $A_{LAT}$ as a function of $X_E$, $X_{DOT_E}$, $X_P$ and $X_{DOT_P}$. The derivatives $\frac{2}{3e} A_{LAT}$ and $\frac{2}{3a} A_{OSAT}$ appearing in equations (4.2-10) and (4.2-12) link $P_A$ to $X_E$ through the interception laws. The derivation of these derivatives for each case is discussed in the balance of this chapter.

4.3 Optimization Against Proportional Navigation

Section 3.1 contains a brief summary of the proportional navigation interception law as it applies to this problem. However, the equations for line-of-sight angle $\phi$, its derivative $\dot{\phi}$ and the closing speed $V_C$ are not presented there. Before deriving expressions for $\frac{2}{3e} A_{OSAT}$ and $\frac{2}{3a} A_{OSAT}$ as promised in the previous section, it is necessary to write down all the equations for computing $\phi$, $\dot{\phi}$, $V_C$ and $A_{OSAT}$.

First, we define two quantities: RANGEVECTOR and RANGE.

$$\text{RANGEVECTOR} = X_E - X_P = \text{the vector position difference of the E position from the P position}. \quad (4.3-1)$$

$$\text{RANGE} = ||\text{RANGEVECTOR}||$$

For a briefer notation, we define $R = \text{RANGE}$ and $RV = \text{RANGEVECTOR}$. The time derivatives of these quantities are needed.
The line-of-sight $\phi$ is the angle of the range vector from the horizontal. Thus,

$$\phi = \tan^{-1}\left( \frac{RV_2}{RV_1} \right)$$

(4.3-3)

and

$$\phi = \frac{1}{R^2} \left( RV_1 RV_2 - RV_2 RV_2 \right)$$

(4.3-4)

The closing speed $\nu_C$ is expressed by the following:

$$\nu_C = |\dot{R}|$$

(4.3-5)

Equation (3.1-1) is repeated here to give the expression for $A_{LD}$.

$$A_{LD} = - \frac{\nu_C \phi}{\cos(\gamma_p - \phi)}$$

(3.3-1)

where $\gamma_p$, the pursuer flight-path-angle, comes from equation (4.2-6). As seen in Appendix A, the physical acceleration limits on $P$ require a maximum value to be set on the magnitude of $A_{LD}$. This is denoted by the symbol $g_{\text{limit}}$ (in feet/sec$^2$). The value of $A_{LD}$ after this limit has been taken into account and is termed $A_{LDSAT}$ and computed from this expression:

$$A_{LDSAT} = g_{\text{limit}} \tanh \left( \frac{A_{LD}}{g_{\text{limit}}} \right)$$

(4.3-6)

The differential relationship between $A_{LD}$ and $A_{LDSAT}$ follows the equation

$$\frac{\partial A_{LDSAT}}{\partial A_{LD}} = \frac{1}{\cosh^2 \left( A_{LD}/g_{\text{limit}} \right)}$$

(4.3-7)
We now seek to find \( \frac{\partial \text{Altitude}}{\partial \alpha_{E}} \) and \( \frac{\partial \text{Altitude}}{\partial \alpha_{A}} \). The derivatives of \( \dot{RV} \) and \( \ddot{RV} \) with respect to \( \alpha_{E} \) and \( \alpha_{A} \) are trivial and are not given here. The derivative \( \frac{\partial \delta}{\partial \alpha_{E}} \) is zero by equation (4.2-7) and \( \frac{\partial \delta}{\partial \alpha_{A}} \) is found in equation (4.2-8). Differentiating the equation for \( \dot{\phi} \) gives us

\[
\frac{\partial \dot{\phi}}{\partial \alpha_{E}} = \frac{1}{R^2} \left( RV_{1} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{2} - RV_{2} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{1} \right) \tag{4.3-8}
\]

and

\[
\frac{\partial \phi}{\partial \alpha_{A}} = \frac{1}{R^2} \left( RV_{1} \frac{\partial}{\partial \alpha_{A}} \dot{RV}_{2} - RV_{2} \frac{\partial}{\partial \alpha_{A}} \dot{RV}_{1} \right) \tag{4.3-9}
\]

The derivatives of \( \dot{\phi} \) are

\[
\frac{\partial \dot{\phi}}{\partial \alpha_{E}} = \frac{1}{R^4} \left( R^2 \left( RV_{1} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{2} + RV_{2} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{1} \right) - (RV_{1} \dot{RV}_{2}) 2R \frac{\partial}{\partial \alpha_{E}} \dot{R} \right)
- \frac{1}{R^4} \left( R^2 \left( RV_{1} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{2} + RV_{2} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{1} \right) - (\dot{RV}_{1} \dot{RV}_{2}) 2R \frac{\partial}{\partial \alpha_{E}} \dot{R} \right) \tag{4.3-10}
\]

and

\[
\frac{\partial \dot{\phi}}{\partial \alpha_{A}} = \frac{1}{R^4} \left( R^2 \left( RV_{1} \frac{\partial}{\partial \alpha_{A}} \dot{RV}_{2} + RV_{2} \frac{\partial}{\partial \alpha_{A}} \dot{RV}_{1} \right) - (RV_{1} \dot{RV}_{2}) 2R \frac{\partial}{\partial \alpha_{A}} \dot{R} \right)
- \frac{1}{R^4} \left( R^2 \left( RV_{1} \frac{\partial}{\partial \alpha_{A}} \dot{RV}_{2} + RV_{2} \frac{\partial}{\partial \alpha_{A}} \dot{RV}_{1} \right) - (\dot{RV}_{1} \dot{RV}_{2}) 2R \frac{\partial}{\partial \alpha_{A}} \dot{R} \right) \tag{4.3-11}
\]

The derivatives of RANGE are

\[
\frac{\partial R}{\partial \alpha_{E}} = \frac{1}{R} \left( RV_{1} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{2} \right) \text{ and } \frac{\partial R}{\partial \alpha_{A}} = \frac{1}{R} \left( RV_{1} \frac{\partial}{\partial \alpha_{A}} \dot{RV}_{2} \right) \tag{4.3-12}
\]

Differentiating \( \dot{R} \) gives us

\[
\frac{\partial \dot{R}}{\partial \alpha_{E}} = \frac{1}{R^2} \left( R \left( RV_{1} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{2} + RV_{2} \frac{\partial}{\partial \alpha_{E}} \dot{RV}_{1} \right) - (\dot{RV}_{1} \dot{RV}_{2}) \frac{\partial}{\partial \alpha_{E}} \dot{R} \right) \tag{4.3-13}
\]
and
\[
\frac{\partial \hat{R}}{\partial \rho_{PA}} = \frac{1}{R^2} \left( R (RV \frac{\partial}{\partial \rho_{PA}} RV + RV \frac{\partial}{\partial \rho_{PA}} RV) - (RV \cdot RV) \frac{\partial}{\partial \rho_{PA}} R \right)
\]

Since \( VC = |\hat{r}| \),
\[
\frac{\partial VC}{\partial XE} = \text{SGN}(\hat{r}) \frac{\partial \hat{r}}{\partial XE}
\]

and
\[
\frac{\partial VC}{\partial \rho_{PA}} = \text{SGN}(\hat{r}) \frac{\partial \hat{r}}{\partial \rho_{PA}}
\]

We now have all the necessary derivatives for obtaining the \( A_{\text{LOAD}} \) derivatives. First, we differentiate \( A_{\text{LOAD}} \).

\[
\frac{\partial A_{\text{LOAD}}}{\partial XE} = -\frac{\lambda \phi}{\cos(\lambda \phi)} \frac{\partial VC}{\partial XE} - \frac{\lambda VC}{\cos^2(\lambda \phi)} \frac{\partial \phi}{\partial XE} + \frac{\lambda VC \phi}{\cos^2(\lambda \phi)} \frac{\partial \sin(\lambda \phi)}{\partial XE} \frac{\partial \phi}{\partial XE}
\]

\[
\frac{\partial A_{\text{LOAD}}}{\partial \rho_{PA}} = -\frac{\lambda \phi}{\cos(\lambda \phi)} \frac{\partial VC}{\partial \rho_{PA}} - \frac{\lambda VC}{\cos(\lambda \phi)} \frac{\partial \phi}{\partial \rho_{PA}} - \frac{\lambda VC \phi}{\cos(\lambda \phi)} \frac{\partial \sin(\lambda \phi)}{\partial \rho_{PA}} \left( \frac{\partial \phi}{\partial \rho_{PA}} - \frac{\partial \phi}{\partial XE} \right)
\]

Note that a derivative of \( \phi \) appears in \( \frac{\partial A_{\text{LOAD}}}{\partial \rho_{PA}} \) but not in \( \frac{\partial A_{\text{LOAD}}}{\partial XE} \).

Finally, we use equation (4.3-7) to obtain
\[
\frac{\partial A_{\text{LOAD}}}{\partial XE} = \frac{1}{\cosh^2 \left( \frac{A_{\text{LOAD}}}{\lambda_{\text{limit}}} \right)} \frac{\partial A_{\text{LOAD}}}{\partial XE}
\]

and
\[
\frac{\partial A_{\text{LOAD}}}{\partial \rho_{PA}} = \frac{1}{\cosh^2 \left( \frac{A_{\text{LOAD}}}{\lambda_{\text{limit}}} \right)} \frac{\partial A_{\text{LOAD}}}{\partial \rho_{PA}}
\]
Inserting the expressions for the $\mathbf{A_{L,S,T}}$ derivatives into equations (4.2-10) and (4.2-12) of the previous section completes the process of finding $f_x$ for the proportional navigation case.

4.4 Optimization Against Minimum-Miss-Time (MMT) Interception

As in the previous section, we desire formulae for evaluating $\frac{\partial \mathbf{A_{L,S,T}}}{\partial \mathbf{x}}$ and $\frac{\partial \mathbf{A_{L,S,T}}}{\partial \mathbf{p}}$. Referring to Section 3.2, MMT interception guidance causes the cost function

$$J = \frac{1}{2} d_{\text{min}}^2 + \rho t_{\text{min}}$$

(3.2-3)

to be minimized by appropriate choice of the P control vector $\mathbf{A_L} = \begin{bmatrix} A_{L_0} \\ A_{L_0} \end{bmatrix}$. This process occurs at update points regularly spaced in time. At each such point, the necessary conditions for an optimum are

$$\frac{\partial J}{\partial \mathbf{A_L}} = d_{\text{min}} \frac{\partial d_{\text{min}}}{\partial \mathbf{A_L}} + \rho \frac{\partial t_{\text{min}}}{\partial \mathbf{A_L}} = 0$$

(4.4-1)

Once these conditions are achieved, keeping at the optimum means they must be maintained to first order regardless of perturbations in the current $\mathbf{x}_E, x_\mathbf{D}, x_\mathbf{P}, x_\mathbf{D}, x_\mathbf{P}$ and $\mathbf{A_L}$ vectors. (The necessary conditions are assumed to be insensitive to changes in the current autopilot states $\mathbf{s}, \mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$ as these quantities have 2nd order effect on the predicted quantities $d_{\text{min}}$ and $t_{\text{min}}$.) This fact allows us to write differential relationships between $\mathbf{A_L}$ and $\mathbf{x}_E$ and $\mathbf{p}_\mathbf{A}$. 
The derivatives \( \frac{\partial \chi}{\partial \Delta_L} \) and \( \frac{\partial \chi_{\Delta}}{\partial \Delta_L} \) are functions of \( \Delta_L \), \( \chi_E \), \( \chi_{\Delta E} \), \( \chi_P \) and \( \chi_{\Delta \Delta P} \). Recalling from equation (4.2-2) that

\[
\chi_E \equiv \begin{bmatrix} \chi_E \\ \chi_{\Delta E} \end{bmatrix}
\] (4.2-2)

we group the \( P \) position and velocity in a similar manner. Let

\[
\chi_P \equiv \begin{bmatrix} \chi_P \\ \chi_{\Delta \Delta P} \end{bmatrix}
\] (4.4-2)

Thus, we can write the perturbation expansions on \( \frac{\partial \chi}{\partial \Delta_L} \) and \( \frac{\partial \chi_{\Delta}}{\partial \Delta_L} \).

\[
\frac{\partial \chi}{\partial \Delta_L} = \frac{\partial \chi}{\partial \chi_P} \chi_P + \frac{\partial \chi}{\partial \chi_{\Delta E}} \chi_{\Delta E} + \frac{\partial \chi}{\partial \chi_{\Delta \Delta L}} \chi_{\Delta \Delta L}
\] (4.4-3)

and

\[
\frac{\partial \chi_{\Delta}}{\partial \Delta_L} = \frac{\partial \chi_{\Delta}}{\partial \chi_P} \chi_P + \frac{\partial \chi_{\Delta}}{\partial \chi_{\Delta E}} \chi_{\Delta E} + \frac{\partial \chi_{\Delta}}{\partial \chi_{\Delta \Delta L}} \chi_{\Delta \Delta L}
\] (4.4-4)

Maintaining the necessary conditions of equation (4.4-1) implies that

\[
\frac{\partial \chi}{\partial \Delta_L} = \frac{\partial \chi}{\partial \Delta_L} \Delta + \Delta \frac{\partial \chi}{\partial \Delta_L} \Delta + \rho \frac{\partial \chi}{\partial \Delta_L} \rho = 0
\] (4.4-5)

Since \( \Delta \) is a function of the same variables as \( \frac{\partial \chi}{\partial \Delta_L} \), we write

\[
\frac{\partial \chi}{\partial \Delta_L} = \frac{\partial \chi}{\partial \chi_P} \chi_P + \frac{\partial \chi}{\partial \chi_{\Delta E}} \chi_{\Delta E} + \frac{\partial \chi}{\partial \chi_{\Delta \Delta L}} \chi_{\Delta \Delta L}
\] (4.4-6)

Substituting equations (4.4-3), (4.4-4) and (4.4-6) into equation (4.4-5), we obtain
\[ S \left( \frac{\partial^2}{\partial A_l} \right) = \frac{\partial A_m}{\partial x_p} \left( \frac{\partial A_m}{\partial x_p} S_{x_p} + \frac{\partial A_m}{\partial x_e} S_{x_e} + \frac{\partial A_m}{\partial A_l} S_{A_l} \right) \\
+ \frac{\partial A_m}{\partial x_e} \left( \frac{\partial^2 A_m}{\partial x_p \partial A_l} S_{x_p} + \frac{\partial^2 A_m}{\partial x_e \partial A_l} S_{x_e} + \frac{\partial^2 A_m}{\partial A_l^2} S_{A_l} \right) \\
+ p \left( \frac{\partial^2 A_m}{\partial x_p \partial A_l} S_{x_p} + \frac{\partial^2 A_m}{\partial x_e \partial A_l} S_{x_e} + \frac{\partial^2 A_m}{\partial A_l^2} S_{A_l} \right) \\
= 0 \]

or

\[ S \left( \frac{\partial^2}{\partial A_l} \right) = \left( \frac{\partial A_m}{\partial x_p} \frac{\partial A_m}{\partial x_p} + \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_e} + p \frac{\partial^2 A_m}{\partial A_l^2} \right) S_{x_p} \\
+ \left( \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_p} + \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_e} + p \frac{\partial^2 A_m}{\partial A_l^2} \right) S_{x_e} \\
+ \left( \frac{\partial A_m}{\partial A_l} \frac{\partial A_m}{\partial A_l} + \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_e} + p \frac{\partial^2 A_m}{\partial A_l^2} \right) S_{A_l} \\
= 0 \quad (4.4-7) \]

This equation lets us solve for \( \frac{\partial A_m}{\partial x_e} \) and \( \frac{\partial A_m}{\partial x_p} \). Thus,

\[ \frac{\partial A_m}{\partial x_e} = - \left( \frac{\partial A_m}{\partial x_p} \frac{\partial A_m}{\partial x_p} + \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_e} + p \frac{\partial^2 A_m}{\partial A_l^2} \right)^{-1} \times \left( \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_p} + \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_e} + p \frac{\partial^2 A_m}{\partial A_l^2} \right) \quad (4.4-8) \]

and

\[ \frac{\partial A_m}{\partial x_p} = - \left( \frac{\partial A_m}{\partial x_p} \frac{\partial A_m}{\partial x_p} + \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_e} + p \frac{\partial^2 A_m}{\partial A_l^2} \right)^{-1} \times \left( \frac{\partial A_m}{\partial x_p} \frac{\partial A_m}{\partial x_p} + \frac{\partial A_m}{\partial x_e} \frac{\partial A_m}{\partial x_e} + p \frac{\partial^2 A_m}{\partial A_l^2} \right) \quad (4.4-9) \]

Since the output of the MMT guidance law is \( A_l \) at discrete points in time, the control supplied to the \( P \) dynamics equation
is
\[ A_{LD} = A_{LO} + \frac{\partial A_{L}}{\partial \xi} \tau \] (4.4-10)
where \( \tau \) is the time since \( A_L \) was last supplied.
Thus,
\[ \frac{\partial A_{LD}}{\partial \xi_e} = \left[ 1, \tau^2 \right] \frac{\partial A_{L}}{\partial \xi_e} \] (4.4-11)
and
\[ \frac{\partial A_{LD}}{\partial \xi_p} = \left[ 1, \tau^2 \right] \frac{\partial A_{L}}{\partial \xi_p} \] (4.4-12)
From the definition of \( \xi_P \) in equation (4.4-2) it is clear that
\[ \xi_P = \begin{bmatrix} 0 & I \\ 4 \times 1 & 4 \times 1 \end{bmatrix} P_A \] (4.4-13)
and
\[ \frac{\partial \xi_P}{\partial P_A} = \begin{bmatrix} 0 & I \\ 4 \times 1 & 4 \times 1 \end{bmatrix} \]
Thus,
\[ \frac{\partial A_{LD}}{\partial P_A} = \left[ 1, \tau^2 \right] \frac{\partial A_{L}}{\partial \xi_P} \frac{\partial \xi_P}{\partial P_A} = \left[ 1, \tau^2 \right] \frac{\partial A_{L}}{\partial \xi_P} \begin{bmatrix} 0 & I \\ 4 \times 1 & 4 \times 1 \end{bmatrix} \] (4.4-14)
From the previous section,
\[ \frac{\partial A_{LD_{OUT}}}{\partial \xi_e} = \frac{1}{\cosh^2 \left( \frac{A_{LD_{OUT}}}{Q_{LIMIT}} \right)} \frac{\partial A_{LO}}{\partial \xi_e} \] (4.3-19)
and
\[ \frac{\partial A_{LD_{OUT}}}{\partial P_A} = \frac{1}{\cosh^2 \left( \frac{A_{LD_{OUT}}}{Q_{LIMIT}} \right)} \frac{\partial A_{LO}}{\partial P_A} \] (4.3-20)
Thus, the desired derivatives are found, assuming equations (4.4-8) and (4.4-9), from the following:

$$\frac{\partial A_{losm}}{\partial \theta E} = \frac{1}{\cos(\theta \theta_{mm})} [I, I] \frac{\partial A_{\theta}}{\partial \theta E}$$  \hspace{1cm} (4.4-15)

$$\frac{\partial A_{losm}}{\partial \theta A} = \frac{1}{\cosh^2(\theta \theta_{mm})} [I, I] \frac{\partial A_{\theta}}{\partial \theta \theta} [I, I]$$  \hspace{1cm} (4.4-16)

Substituting into equations (4.2-10) and (4.2-12) gives for the MMT interception law. Note that unlike the proportional navigation case, this $f_\theta$ is computed using functions $\frac{\partial A_{\theta}}{\partial \theta E}$ and $\frac{\partial A_{\theta}}{\partial \theta \theta}$ which are constant between guidance update times. This reflects the discontinuous nature of the control given to $P$ by the MMT law.

4.5 Optimization Against Least-Risk (LR) Interception

The final and most complicated interception steering law will now be discussed. Least-Risk is similar to MMT in that a piecewise-continuous $f_\theta$ matrix results: the required first and second derivative information is only available at discrete update times. The differences between LR and MMT steering result from the distinct definitions of optimality in the search for $A_{\alpha}$.

As seen in Section 3.3, LR guidance has two sets of necessary conditions corresponding to the two possible types of optima. In the first case, $d_{\alpha min}$, the larger of the two
predicted minimum distances, must be at a stationary point. Thus,

$$\mathcal{L}_{d_{\text{min}}} \equiv \left( \frac{\partial \mathcal{L}_{d}}{\partial A_L} \right)^T = 0 \quad (4.5-1)$$

The derivatives $\frac{\partial \mathcal{L}}{\partial X^E}$ and $\frac{\partial \mathcal{L}}{\partial X^P}$ for this case are derived in a similar manner to those in the case of MMT guidance. Following the form of equation (4.4-3), we write

$$\mathcal{S} \mathcal{L}_{d_{\text{min}}} = \mathcal{S} \left( \frac{\partial \mathcal{L}_{d_{\text{min}}}}{\partial A_L} \right)^T = \frac{\partial^2 \mathcal{L}_{d_{\text{min}}}}{\partial X^P \partial A_L} \delta X^P + \frac{\partial \mathcal{L}_{d_{\text{min}}}}{\partial X^E \partial A_L} \delta X^E + \frac{\partial^2 \mathcal{L}_{d_{\text{min}}}}{\partial A_L^2} \delta A_L \quad (4.5-2)$$

Since the necessary conditions of (4.5-1) must be held to first order at the optimum, we require that

$$\mathcal{S} \mathcal{L}_{d_{\text{min}}} = 0 \quad (4.5-3)$$

This permits solution for the desired derivatives. Thus,

$$\frac{\partial A_L}{\partial X^E} = - \left( \frac{\partial^2 \mathcal{L}_{d_{\text{min}}}}{\partial A_L^2} \right)^{-1} \frac{\partial \mathcal{L}_{d_{\text{min}}}}{\partial X^E \partial A_L} \quad (4.5-4)$$

and

$$\frac{\partial A_L}{\partial X^P} = - \left( \frac{\partial^2 \mathcal{L}_{d_{\text{min}}}}{\partial A_L^2} \right)^{-1} \frac{\partial \mathcal{L}_{d_{\text{min}}}}{\partial X^P \partial A_L} \quad (4.5-5)$$

The second type of optimum seems to arise more often in
practice. Again, reference to Section 3.3 gives us the necessary conditions:

\[ d_{\text{miss}}^S = d_{\text{miss}}^L \]  
\[ (\text{the predicted miss distances are equal}) \]  
\[ (4.5-6) \]

\[ g_{\text{miss}}^S \cdot g_{\text{miss}}^L = -1 \]  
\[ (\text{the miss gradients point in opposing directions}) \]  
\[ (4.5-7) \]

To maintain these conditions to first order regardless of perturbations in \( \chi^P \), \( \chi^E \) and \( A_L \) we form these constraints:

\[ S (d_{\text{miss}}^S - d_{\text{miss}}^L) = 0 \]  
\[ (4.5-8) \]

\[ S (g_{\text{miss}}^S \cdot g_{\text{miss}}^L) = 0 \]  
\[ (4.5-9) \]

The perturbation expansions for these quantities are

\[ S (d_{\text{miss}}^S - d_{\text{miss}}^L) = \left( \frac{\partial d_{\text{miss}}^L}{\partial \chi^P} - \frac{\partial d_{\text{miss}}^S}{\partial \chi^P} \right) S \chi^P + \left( \frac{\partial d_{\text{miss}}^L}{\partial \chi^E} - \frac{\partial d_{\text{miss}}^S}{\partial \chi^E} \right) S \chi^E + \left( \frac{\partial d_{\text{miss}}^L}{\partial A_L} - \frac{\partial d_{\text{miss}}^S}{\partial A_L} \right) S A_L \]  
\[ (4.5-10) \]

and

\[ S (g_{\text{miss}}^S \cdot g_{\text{miss}}^L) = g_{\text{miss}}^L \cdot S g_{\text{miss}}^S + g_{\text{miss}}^S \cdot S g_{\text{miss}}^L \]

\[ = g_{\text{miss}}^L^T \left( \frac{\partial \chi^P}{\partial A_L} S \chi^P + \frac{\partial \chi^E}{\partial A_L} S \chi^E + \frac{\partial \chi^E}{\partial A_L} S A_L \right) \]

\[ + g_{\text{miss}}^S^T \left( \frac{\partial \chi^P}{\partial A_L} S \chi^P + \frac{\partial \chi^E}{\partial A_L} S \chi^E + \frac{\partial \chi^E}{\partial A_L} S A_L \right) \]

\[ (4.5-11) \]

Equations (4.5-10) and (4.5-11) can be substituted into (4.5-8) and
(4.5-9), respectively. The result is two simultaneous equations for $\frac{\partial A_l}{\partial x}$ and $\frac{\partial A_l}{\partial \phi}$. Writing them in vector-matrix form, we solve for $\frac{\partial A_l}{\partial x}$ and then for $\frac{\partial A_l}{\partial x}$ and $\frac{\partial A_l}{\partial \phi}$. The results follow:

$$\frac{\partial A_l}{\partial x} = -\left[ G^L_{mm} - G^S_{mm} \right]^{-1} \left[ \frac{\partial A_l}{\partial X} - \frac{\partial A_l}{\partial \phi} \right]$$

and

$$\frac{\partial A_l}{\partial \phi} = -\left[ G^L_{mm} - G^S_{mm} \right]^{-1} \left[ \frac{\partial A_l}{\partial X} - \frac{\partial A_l}{\partial \phi} \right]$$

(4.5-12)

(4.5-13)

where

$$G^L_{mm} = \frac{\partial^2 x}{\partial x^2}$$

$$G^S_{mm} = \frac{\partial^2 x}{\partial \phi^2}$$

$$G^L_{ml} = \frac{\partial^2 x}{\partial x \partial \phi}$$

$$G^S_{ml} = \frac{\partial^2 x}{\partial \phi \partial x}$$

Once we have $\frac{\partial A_l}{\partial x}$ and $\frac{\partial A_l}{\partial \phi}$ for a particular LR optimum, the derivations for $\frac{\partial A_{l1}}{\partial x}$ and $\frac{\partial A_{l2}}{\partial \phi}$ are identical to those found in the section on MMT guidance. Without further comment, we repeat those results here.

$$\frac{\partial A_{l1}}{\partial x} = \frac{1}{\cosh^2 \frac{A_l}{2}} \left[ 1, 1 \right] \frac{\partial A_l}{\partial x}$$

(4.4-15)
\[
\frac{\Delta A_{\text{EXT}}}{\Delta A} = \frac{1}{\cosh^2 \left( \frac{A_{\text{EXT}}}{2 A_{\text{LIM}} - A_{\text{EXT}}} \right)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{\partial A_{\text{L}}}{\partial x_p} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}
\] (4.4-16)

As in the case of MMT guidance, these results are substituted (for the particular optimum found) into equations (4.2-10) and (4.2-12) to give the desired \( \bar{f}_k \). The same observations about constant \( \frac{\partial A_{\text{L}}}{\partial x_E} \) and \( \frac{\partial A_{\text{L}}}{\partial x_p} \) between guidance update times can be made here as were made in the MMT case in the previous section.
5.0 Summary

Starting with given nominal evader maneuvers, the local optimal maneuver is found against each interception scheme. Each optimal is not necessarily the global optimal; the entire maneuver control space cannot be searched for reasons of cost. However, the word "optimal" is used instead of "local optimal" for compactness. For the case of Minimum-Miss-Time guidance, three situations are investigated. The maximum lateral acceleration capability is lowered to 100g from 200g to test evasion against a weakened pursuer. Then this quantity is raised to 300g for examination of performance against a stronger pursuer. Finally, the pursuer g limit is returned to 200g and the desired impact point is changed to force E to fly over the initial pursuer position. A completely different intercept geometry results.

5.1 Parameter Values Assumed

Appendix A describes the E and P models in detail. That data is not repeated here. This section deals with the pertinent initial conditions and environment assumed in the simulation. The values of interest follow.

\[
g = \text{acceleration due to gravity} = 32.17 \text{ ft/sec}^2 \text{ constant in direction and magnitude (flat earth assumed).}
\]
\[ \rho = \text{air density} = \rho_0 e^{-\frac{h}{2400}} \text{ in slugs/feet}^3 \text{ with} \]
\[ \rho_0 = 0.002377 \text{ slugs/ft}^3 \] (The coordinate frame has altitude for the ordinate and range for the abscissa. Altitude is measured from sea level and range from the P initial point.)

\[ x_e = \text{initial E position} = (-52905, 81584) \text{ in feet} \]
\[ v_e = \text{initial E velocity} = (7492.1, -18543.7) \text{ in feet/sec.} \]
\[ x_p = \text{initial P position} = (0, 10000) \text{ in feet} \]
\[ v_p = \text{initial P velocity} = (-3023.1, 1565.8) \text{ in feet/sec.} \]

Nominal desired impact point is 25000 feet uprange of the P initial point at (-25000, 0).

5.2 Search Behavior in Seeking the Optimum

The steps of the search procedure used to find the optimum maneuver against PN are presented below in Table 5.2-1 as an example of the technique. Note that simple gradient steps are initially used to help in estimating the correct step size (desired change in miss distance) with which to start the modified gradient projection search. In the inner product \( \langle d_{new}, g_p \rangle \), \( d_{new} \) is the latest conjugate search direction. Note that the term \( \langle g_p, g_p \rangle \), representing the length of the miss gradient, shrinks dramatically. Also, no cleanup steps (steps to bring the impact to the desired position) are required and few half-steps (due to rejecting a bad step) need to be taken. For some reason probably associated with the numerical approximation errors, the minimum of \( \langle g_p, g_p \rangle \) appears to correspond to a miss about 10 feet smaller than the best miss found. This difference is not considered significant by the author.
<table>
<thead>
<tr>
<th>Step</th>
<th>Miss</th>
<th>Deltamiss</th>
<th>Predicted Deltamiss</th>
<th>Norm of Proj. Grad</th>
<th>$\langle d_{new}, \delta \rangle$</th>
<th>$\langle \delta, \delta \rangle$</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10336</td>
<td>-</td>
<td>-</td>
<td>3600</td>
<td>-</td>
<td>13656069</td>
<td>Nominal maneuver</td>
</tr>
<tr>
<td>1</td>
<td>10714</td>
<td>378</td>
<td>500</td>
<td>2843</td>
<td>-</td>
<td>6481180</td>
<td>Simple grad. step</td>
</tr>
<tr>
<td>2</td>
<td>10796</td>
<td>82</td>
<td>100</td>
<td>2246</td>
<td>-</td>
<td>5283134</td>
<td>&quot;</td>
</tr>
<tr>
<td>3</td>
<td>11012</td>
<td>216</td>
<td>300</td>
<td>1640</td>
<td>-</td>
<td>2783290</td>
<td>Projected grad.</td>
</tr>
<tr>
<td>4</td>
<td>11204</td>
<td>192</td>
<td>300</td>
<td>1109</td>
<td>-</td>
<td>1249408</td>
<td>&quot;</td>
</tr>
<tr>
<td>5</td>
<td>11367</td>
<td>163</td>
<td>300</td>
<td>659</td>
<td>-</td>
<td>435078</td>
<td>&quot;</td>
</tr>
<tr>
<td>6</td>
<td>11490</td>
<td>122</td>
<td>300</td>
<td>373</td>
<td>-</td>
<td>138881</td>
<td>&quot;</td>
</tr>
<tr>
<td>7</td>
<td>11459</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Bad proj. grad.</td>
</tr>
<tr>
<td>7</td>
<td>11490</td>
<td>1</td>
<td>150</td>
<td>390</td>
<td>-</td>
<td>151135</td>
<td>step - cut in half</td>
</tr>
<tr>
<td>8</td>
<td>11527</td>
<td>36</td>
<td>75</td>
<td>565</td>
<td>-</td>
<td>319086</td>
<td>&quot;</td>
</tr>
<tr>
<td>9</td>
<td>11501</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Bad proj. grad.</td>
</tr>
<tr>
<td>9</td>
<td>11527</td>
<td>.3</td>
<td>37.5</td>
<td>306</td>
<td>-</td>
<td>92878</td>
<td>step - cut in half</td>
</tr>
<tr>
<td>10</td>
<td>11528</td>
<td>.05</td>
<td>18.75</td>
<td>300</td>
<td>-</td>
<td>90327</td>
<td>Projected grad.</td>
</tr>
<tr>
<td>11</td>
<td>11782</td>
<td>-</td>
<td>-</td>
<td>91</td>
<td>6735</td>
<td>8322</td>
<td>Conj. grad. search.</td>
</tr>
<tr>
<td>12</td>
<td>11782</td>
<td>-</td>
<td>-</td>
<td>91</td>
<td>2918</td>
<td>8328</td>
<td>2nd part is the opt</td>
</tr>
<tr>
<td>13</td>
<td>11497</td>
<td>-</td>
<td>-</td>
<td>314</td>
<td>-16889</td>
<td>98776</td>
<td>&quot;</td>
</tr>
<tr>
<td>14</td>
<td>11780</td>
<td>-</td>
<td>-</td>
<td>77</td>
<td>5055</td>
<td>5991</td>
<td>&quot;</td>
</tr>
<tr>
<td>15</td>
<td>11791</td>
<td>-</td>
<td>-</td>
<td>384</td>
<td>-2013</td>
<td>147824</td>
<td>&quot;</td>
</tr>
<tr>
<td>16</td>
<td>11791</td>
<td>-</td>
<td>-</td>
<td>324</td>
<td>104963</td>
<td>104963</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

Table 5.2-1: Search for Optimum Against PN
<table>
<thead>
<tr>
<th>Step</th>
<th>Miss</th>
<th>Deltamiss</th>
<th>Predicted Deltamiss</th>
<th>Norm of Proj. Grad.</th>
<th>$\langle A_{new} \rangle$</th>
<th>$\langle B_{Opt} \rangle$</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>11791</td>
<td>-</td>
<td>-</td>
<td>195</td>
<td>62932</td>
<td>38094</td>
<td>Conj. grad. search</td>
</tr>
<tr>
<td>18</td>
<td>11791</td>
<td>-</td>
<td>-</td>
<td>293</td>
<td>8282</td>
<td>85675</td>
<td>2nd part is the opt</td>
</tr>
<tr>
<td>19</td>
<td>11791</td>
<td>-</td>
<td>-</td>
<td>163</td>
<td>4635</td>
<td>26484</td>
<td>&quot;</td>
</tr>
<tr>
<td>20</td>
<td>11791</td>
<td>-</td>
<td>-</td>
<td>253</td>
<td>58587</td>
<td>63939</td>
<td>&quot;</td>
</tr>
<tr>
<td>21</td>
<td>11790</td>
<td>-</td>
<td>-</td>
<td>112</td>
<td>25230</td>
<td>12566</td>
<td>&quot;</td>
</tr>
<tr>
<td>22</td>
<td>11791</td>
<td>-</td>
<td>-</td>
<td>222</td>
<td>58588</td>
<td>49439</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

Table 5.2-1 (Continued)
5.3 Optimal Evasion vs. Proportional Navigation

Three E maneuvers along with the corresponding P responses are pictured in Figure 5.3-1. The "nominal" E maneuver is the same nominal used for the other cases described in this chapter. Note that the "optimal" maneuver is nearly the same trajectory. To show the range of possibilities a little better, the optimal E maneuver against the Minimum-Miss-Time steering law is also given. The latter two maneuvers and their P responses are compared below. Note the miss for the optimal maneuver is 11791 feet while for the "MMT optimal" it is only 1749 feet. The significance of this difference and speculation about why it occurs are discussed below.

The proportional navigation concept contains a constant-velocity assumption. In particular the E and P vehicles are assumed to continue in the future with the present velocities. From Section 3.1 we see that PN attempts to drive the line-of-sight rate to zero. When it reaches zero, continuation of both parties with their current velocities assures an intercept. Unfortunately for PN, P is a thrusting vehicle and E has variable drag and lift. As the two E maneuvers unfold in time, the interception scheme continuously predicts future motion for both parties. These predictions are presented for both maneuvers at particular points in time in Figure 5.3-2. Note that the line-of-sight rate \( \dot{\phi} \), also presented, gives a measure of how close PN is to the desired condition \( \dot{\phi} \approx 0 \). In part
Figure 5.3-1b: Evader Control Histories
(After Intercept Time)
Figure 5.3-2a: The Predicted E and P Trajectories at $t_0 = 1$ second (Dashed Lines).
Figure 5.3-2b: The PN-Predicted E and P Trajectories at t₀ = 2 seconds (Dashed Lines).
Figure 5.3-2c. The PN-Predicted E and P Trajectories at $t_0 = 3$ seconds (Dashed Lines).
Figure 5.3-2d: The PN-Predicted E and P Trajectories at $t_0=3.5$ seconds (Dashed Lines).
Figure 5.3-2c: The PN-Predicted E and P Trajectories at $t_0=4$ seconds (Dashed Lines).
Figure 5.3-2f: The PN-Predicted E and P Trajectories at $t_0 = 4.5$ seconds (Dashed Lines).
5.3-2a the interception appears impossible because the initial P velocity is too small. (Of course, because of the 100g+ thrust of the P engine, this velocity increases rapidly). The relatively large also reflects the fact that the P direction points at too large an angle from the uprange horizon. This angle is seen to decrease in parts 5.3-2(b-d). By the time of part 5.3-2c, is under control and an intercept is predicted against both the optimal and the MMT optimal maneuvers.

However, the maneuvers exhibit aerodynamic forces giving rise to accelerations ignored by PN. The PN constant velocity prediction of the MMT optimal maneuver is reasonably accurate in terms of altitude vs. time because E is using a relatively small angle of attack and the aerodynamic force is not excessive. This condition persists through to the intercept as the P feedback seems to correct with ease the small E predicting errors. Apparently, the inability of PN to predict the P velocity is the major reason this intercept is not perfect. The large aerodynamic forces on E in the optimal maneuver account for poor altitude vs. time predictions. P is sent out on a low, nearly horizontal path by the early PN velocity estimates which, as they predict too low a velocity for P, also foresee a low-drag, high velocity vertical dive into the ground for E. The optimal maneuver, which continuously weaves E back and forth away from the straight-line prediction, also causes E to fall much more slowly than expected by PN. The result starts
to appear at 3.5 sec. in part 5.3-2d. There, P is predicted to fly below the E trajectory and be well up-range before E descends to the altitude region in which an intercept can occur. In the latter parts 5.3-2e and 5.3-2f, it is apparent that P "sees its mistake" and attempts to climb back up to a better altitude. However, the P velocity is too great and even a lateral acceleration of 200g's is insufficient to gain P enough altitude before it passes beneath E. Note in parts 5.3-2(d-f) that as the predicted 6 sec. point rises due to the P pull-up to catch E, the 6 sec. point for E also rises, maintaining at least 7000 feet altitude separation all the way to intercept time.*

Figures 5.3-3 and 5.3-4 are departures from conventional reporting practice. They depict three-dimensional views of the pursuit evasion, with time as the third dimension. The reader is invited to unfold parts b and c of each figure along the axis lines and make the fold angle 90° to form three adjacent faces of a cube. This assists in visualizing each figure. Note that in this space a perfect intercept occurs only if the P trajectory touches the E trajectory. The dashed lines on the figures are trajectories which PN predicts at the starting times of 1 sec., 2 sec., and 3.5 sec. First consider the MMT optimal pursuit evasion in Figure 5.3-3. At early times the E

*"Intercept time" is taken to mean the time of closest approach even though the minimum approach distance may be too large for the encounter to be labelled an interception.
Figure 5.3-3: Range vs. Altitude vs. Time for MMT-Optimal Evasion Against PN (Predictions for $t_0=1, 2, 3, 5$ seconds in Dashed Lines).

Figure 5.3-3a

Figure 5.3-3b

Figure 5.3-3c
Figure 5.3-4: Range vs. Altitude vs. Time for Optimal Evasion Against PN (Predictions for $t_0 = 1, 2, 3, 5$ seconds in Dashed Lines).

Figure 5.3-4a

Figure 5.3-4b
Figure 5.3-4d: Isometric View
trajectory is predicted to extend farther uprange and to lower altitudes than it actually does. However, the difference between predicted and actual flight is apparently not enough for E to "fool" the PN law. An isometric sketch of the 3 dimensional encounter is presented in 5.3-3d to aid in visualization. The optimal evasive maneuver of Figure 5.3-4 also features predictions placing the E trajectory farther downrange than it turns out to be. This error is much greater than in the previous figure. However, as part 5.3-4b indicates, P is well able to sweep over all these predictions. E turns toward P at the end so that P passes it before the intercept time can be reached.

Another reason E is able to dodge P is seen in the altitude vs. time plane (Part 5.3-4c). Note how far below the true E trajectory are the predicted curves. P is mislead by these bad altitude predictions into flying way too low. Starting at 3.5 sec., the error becomes apparent but it is then too late. A maximum-g pull-up by P is insufficient to intercept the E trajectory before P passes under it, continuing in the uprange direction.

5.4 Optimal Evasion vs. Minimum-Miss-Time Guidance

This section can be divided in three parts - all concerned with MMT guidance. The first part shows how the MMT guided pursuer responds to the nominal evasion of Section 5.3. The optimal evasion against MMT guidance is also presented. Figures
5.4-1 through 5.4-4 illustrate both encounters in the same format as found in Section 5.3. The next part illustrates how changes in P lateral acceleration capability affect the nominal and optimal intercept. This discussion includes Figures 5.4-5 and 5.4-6. A different desired impact point is assumed for E in the last part. Instead of (-25K, 0) the point (+25K, 0) is used. The evader is forced to fly over the initial P position and impact on the other side. Nominal and optimal E maneuvers are presented for this scenario also using the format of the previous section.

Unlike the PN guidance, MMT steering considers acceleration models for both parties. The model for P is quite accurate since the P control is known. The E acceleration model must include assumptions about E's control, however. For MMT, E is assumed to have zero lift force but a drag force corresponding to maximum control angle-of-attack.* The resulting flight-path is a straight line with deceleration appropriate to the drag force. Referring to Figure 5.4-1 it can be seen that the nominal E trajectory uses large amounts of control right up to intercept. A nearly perfect intercept is scored by P against this maneuver. On the other hand, the successful optimal evasion features decreased control until after intercept. The

*Physically, this set of forces would result from rapid oscillation or "chattering" between + maximum angle-of-attack values (see Section 3.2).
Figure 5.4-1a: Evasive Maneuvers Against MMT Guidance

-60 K Range -50 K in -40 K Feet -30 K -20 K -10 K
decisions made by P at several points in time are detailed in Figure 5.4-2. Initially (part 5.4-2a), a rapidly decelerating trajectory is predicted with E coming further downrange than actually occurs. Against this, MMT orders P to fly a lofted flight path to achieve interception in minimum time. A perfect intercept is predicted. As time advances in Figure 5.4-2, the predicted E trajectories drop lower in altitude and become more steep. The predictions on the optimal trajectory lie at lower altitudes than corresponding predictions for the nominal since the optimal is flying with less drag than expected. (The optimal maneuver itself extends to lower altitudes than the nominal at corresponding time points.) Against these developments, the two P responses and their pursuit predictions begin to pitch over and perfect intercepts are still predicted. At 3.5 sec. (part 5.4-2d) the predicted P response to the optimal is required to pitch over with maximum lateral acceleration and a perfect intercept is no longer predicted. At succeeding time points the predicted optimal path drops lower and lower in altitude. Since P is already slated to pull max-g's downward, no tighter turn is possible. The predicted intercept miss against the predicted optimal trajectory increases with time. Finally, P passes over E and the intercept is broken. The nominal evasion features a much slower drop in altitude and does not force a max-g turn from P until very near the intercept. As a result, P has little
Figure 5.4-2a: The MMT-Predicted E and P Trajectories at $t_0=1$ second (Dashed Lines).

central after limiting for both P trajectories:

predict miss = 6 feet for both cases.
Figure 5.4-2b: The MMT-Predicted E and P Trajectories at $t_0=2$ seconds (Dashed Lines).
Figure 5.4-2c: The MMT-Predicted E and P Trajectories at $t_0 = 3$ seconds (Dashed Lines).
Figure 5.4-2d: The MMT-Predicted E and P Trajectories at \( t_0 = 3.5 \) seconds (Dashed Lines).
Figure 5.4-2e: The MMT-Predicted E and P Trajectories at $t_0=4$ seconds (Dashed Lines).
Figure 5.4-2f: The MMT-Predicted E and P Trajectories at $t_0 = 4.5$ seconds (Dashed Lines).

Nominal predicted miss = 198 feet
Optimal $n_1 = 4519$ feet
difficulty "catching" E.

Figures 5.4-3 and 5.4-4 represent three-dimensional views of the nominal and optimal cases, respectively, in the same manner as seen in Section 5.3. The predicted trajectories are for 1 sec. and 3.5 sec. starting times. Recall that 3.5 sec. is the approximate time when MMT first predicts that the intercept will not be perfect against optimal evasion. First consider the nominal evasion and P response in Figure 5.4-3. In the range vs. time plane the fact that P is not required to make any sharp turns out of the plane leaves it plenty of uprange acceleration for chasing E. E is also most obliging by not making any sharp turns after about 3 sec. Examining the altitude vs. time plane, we see that the initial predicted E path is a little higher than the actual one because the full control is not being used. However, the later prediction is well matched by E and only a gentle pitch-over is required to obtain a good intercept. The optimal evasion is much harder on P, of course. Since P must make a max-g turn in the altitude vs. time plane starting half-way through the encounter, the acceleration available in the range-time plane (part 5.4-4b) is not as great as before. As a result, when E turns away in this plane at about 4 sec., P cannot change its flight-path to catch up in time. The reason why P must make a max-g turn is apparent in the altitude-time frame of part 5.4-4c. E is deliberately holding back on its control in order to violate
Figure 5.4-3: Range vs. Altitude vs. Time for Nominal Evasion Against MMT (Predictions for $t_0=1$ and 3.5 seconds in Dashed Lines).

Figure 5.4-3a

Figure 5.4-3b

Figure 5.4-3c
Figure 5.4-4: Range vs. Altitude vs. Time for Optimal Evasion Against MMT (Predictions for t = 1 and 3.5 seconds in Dashed Lines).
the "maximum maneuver" assumptions of MMT. By 3.5 sec. P has started to pitch over but the E prediction then still calls for too lofted a trajectory. Thereafter, P is unable to pitch over fast enough to avoid flying over the rapidly descending evader. The turn E makes in the range-time plane delays interception until this crossing over can occur.

A certain penalty is attached to the optimal evasive maneuver since high acceleration requirements are made on E soon after the intercept time. Because a relatively low-drag flight is chosen, E finds itself at 20000 feet altitude and 17000 down-range from desired impact with a velocity of over 12000 feet/sec. pointing down and slightly towards the uprange direction. This is most undesirable and a very sharp turn (330 g's) is required to avoid hitting the ground well uprange of the target. The situation is illustrated in Figure 5.4-1. Note how much slower E is travelling between 6 and 7 sec. compared with the 4-5 sec. period. The turn redirects and severely reduces E's momentum (perhaps introducing motion-dependent errors in onboard inertial sensors).

The effect of varying the maximum lateral P acceleration is illustrated in Figures 5.4-5 and 5.4-6*. In the first figure the maximum acceleration is 300g, instead of the nominal

*Predictions of motion at various times and 3-D plots are not given for these cases as the encounters are similar to the 200g acceleration case given before.
Figure 5.4-5a: Evasive Maneuvers Against MMT Guidance With a 300g Pursuer
Figure 5.4-5b: Evader Control Histories
(After Intercept Time)
Figure 5.4-6b: Evader Control Histories
(After Intercept Time)
200g limit. The new miss distance against the nominal maneuver is essentially zero, while the new optimal evasion miss is 3641 feet, closer but not an interception. Note the sharpness of the P turn required at 4 sec. against the optimal evasion. The E control prior to intercept is even smaller than in the 200g case. Apparently, when the pursuer has a high acceleration tolerance it becomes even more important to "fool" the MMT guidance by using low-drag maneuvers. The low drag is called for earlier, too, indicating the necessity for beginning the "fooling" process earlier in the encounter. The low acceleration capability pursuit of Figure 5.4-6 is hardly capable of intercepting even the previously unsuccessful nominal evasion maneuver. Note the very gradual bending of the P trajectory which a 100g lateral acceleration limit permits. The response to the nominal evasion pushes P near the 100g limit; the optimal evasion response elicits only a slightly sharper bend in the flight-path. The optimal evasion also calls for E to use more control than in the other cases. Apparently, with this pursuer it is more desirable for E to pull down away from a threatened intercept than it is to attempt to fool the prediction capability of MMT guidance. Perhaps in this case, P cannot be lead as far into an intercept-breaking situation and is thus more capable of responding later to the detected prediction errors. Thus, not the "intelligence" of P (prediction accuracy) so much as the "strength" (acceleration capability) of P is
being tested by this evasion.

The last case considered in this section concerns an altered encounter geometry, with the evader forced to fly over the initial position of P. As shown in Figure 5.4-7, the nominal control is a simple, constant angle-of-attack pull up followed by a zero-lift coast to impact. P does not do very well against this nominal maneuver because of the somewhat undesirable initial P pointing angle compounded by the inaccurate early E predictions given by the straight-line assumption in MMT guidance. The miss against the nominal is 348 feet. The optimal evasion uses less control than the nominal for roughly 80% of the time before intercept. This causes prediction errors because of the decreased true E drag compared with what MMT "expects". The E flight path is initially straighter, drawing P farther uprange than is advisable for interception. In the last second, E abandons the "low drag" philosophy and does a hard pull-up. P's velocity is such that this maneuver cannot be successfully countered before the parties pass and the contest ends.

Figure 5.4-8 details the interception decision points in time. Although the initial predicted trajectories for E cause P to swing out, perfect intercepts are predicted. Note that the E altitude vs. time is predicted with good accuracy. This quality proved useful for intercepting the nominal maneuvers in the previous parts of this section. Such is not the case here, however. Previously, E was descending in a ver-
Figure 5.4-7b: E Flight to the New Impact Point After Intercept Time
Figure 5.4-7c: Evader Control Histories
(After Intercept Time)
Figure 5.4-8a: The Predicted E and P Trajectories at $t_0=1$ second (Dashed Lines)
Figure 5.4-8b: The Predicted E and P Trajectories at $t_0=2$ seconds (Dashed Lines)
Figure 5.4-8c: The Predicted E and P Trajectories at $t_0 = 3$ seconds (Dashed Lines)
Figure 5.4-8d: The Predicted E and P Trajectories at $t_0 = 3.5$ seconds (Dashed Lines)

Nominal predicted miss = 154 feet
Optimal $n = 670$ feet
Figure 5.4-8e: The Predicted E and P Trajectories at $t_0=4$ seconds (Dashed Lines)

- Nominal predicted miss = 718 feet
- Optimal $\eta = 2154$ feet

-60 K Range - 50 K in -40 K Feet -30 K -20 K -10 K
tical direction, roughly at right angles to the P flight path. As a result, knowledge of accurate altitude vs. time data on E "cross-track" to the P trajectory was most of what was required to insure interception.* Small errors in estimating the position of E downrange ("downdown" to P) were compensated for by P with slight pull-ups or pull-downs which altered the intercept time. In the present case the interception is "head on" with a more vertical P flight path. The largest errors in predicting E and also the most damaging to P are the now horizontal cross-track errors, due to inaccurate knowledge of the shape of the E path. MMT predicts straight-line paths while the actual curves pull up. Thus, the straight-line assumption made by MMT is a real liability for the first time even though the altitude estimates along this line are accurate. As time goes on, the predicted E paths swing up towards the downrange direction and the P trajectories bend up to meet them. Starting at the 3 second point, the predicted intercepts are no longer perfect.

Optimal evasion is at first closer to the appropriate predicted E path, taking a less severe pull-up initially. This strategy keeps P pointing in a more uprange direction and also contributes some error to the altitude vs. time part of the

*To assure intercept, P must align the relative velocity vector in opposition to the relative position vector. If they are approximately aligned in this case, both are nearly parallel to the downdown direction. With this orientation, greater angle perturbations occur in the relative position vector for a given change in crosstrack component than for the same change in downdown component. This perturbation in angle must be matched by rotating the relative velocity vector via lateral acceleration. Thus, crosstrack relative position errors require more P effort to null than downdown track errors.
prediction. Starting at 3 seconds, the optimal maneuver begins a maximum pull-up. This increases the error in the predicted shape of the E path, and P is constantly "told" to pull up less than the amount really required. By 4 seconds the new maneuver has been discerned by MMT and the prediction of E motion is again accurate. Too late, P attempts an unsuccessful pull-up in the remaining time. E crosses its path to the right, increasing the necessary pull-up acceleration beyond the g limit available to P.

Figures 5.4-9 and 5.4-10 present the three-dimensional views of the two encounters in the same format as before. The path of the nominal maneuver is a steady pull-up in Figure 5.4-9. After the first prediction, seen to be a particularly bad representation of the E flight in the range-time frame, P recovers the correct direction of flight to intercept. Later predictions are more representative of the true E trajectory. The optimal maneuver has a similar range-time plot, seen as part 5.4-10b, although E crosses over P a little bit earlier before the intercept time. The evasion strategy is first seen in part 5.4-10c, the altitude-time plane. There, E is seen to effectively delay the intercept by doing an unpredicted hard pull up. P attempts to pull up also but reaches max-g's. The purpose of E's pull-up and the resulting delay is apparent back in the range-time plane. Little P acceleration is left to be used in this plane, so that P cannot turn sharply to avoid passing E. The time delay postpones the intercept time until after the two paths cross. P is considerably uprange of E when the intercept time is finally reached. Thus, the miss distance is maximized.
Figure 5.4-9a: Range vs. Altitude vs. Time for Nominal Maneuver With Shaded Impact Point Against MMT (Predictions for $t_0=1$ and 3.5 seconds in Dashed Lines)
Figure 5.4-9d: Isometric View
Figure 5.4-10a: Range vs. Altitude vs. Time for Optimal Maneuver With Shifted Impact Point Against MMT (Predictions for $t_0 = 1$ and 3.5 seconds in Dashed Lines)
Figure 5.4-10d: Isometric View
5.5 Optimal Evasion vs. Least-Risk Guidance

The LR results are presented in the same format as used for MMT and PN guidance. They are seen in Figures 5.5-1 through 5.5-4. The optimal and nominal $E$ and $P$ trajectories are shown in Figure 5.5-1. Note that the nominal is the same as in the previous sections, consisting of a pull-down followed by a pull-up using plenty of control. Recall that the LR guidance predicts maximum pull-up and pull-down maneuvers for $E$ and picks $P$ controls to "intercept" both trajectories in a single pass. The miss distance for the nominal case is only 297 feet when MMT guidance is used (see Figure 5.4-1a) compared to 1226 feet with the LR law. This result is discussed later. The LR optimal $E$ control is somewhat similar to the optimal $E$ control for the MMT case in that relatively small control is used and a trajectory with less-than-expected aerodynamic loading results. Comparing Figure 5.5-1a with Figure 5.4-1a shows the optimal $E$ trajectories to be similar in shape. The $P$ responses to the optimals are also similar in shape although it will be noted that the LR response is slightly lofted compared to the MMT case. The principal difference first seen between the MMT and LR evasive maneuvers is that the dive is initially more steep with LR, resulting in a shorter downrange excursion before the intercept time. As before, the optimal evasion requires a high-$g$ turn after this time in order to impact on the target. The particular maneuver shown to do this is more complicated and
Figure 5.5-1a: Evasive Maneuvers Against LR Guidance
Figure 5.5: Evader Control Histories (After Intercept Time)
time-consuming than necessary.* Certainly something similar to
the MMT optimal maneuver could be substituted after intercept
to achieve the desired impact point.

Figure 5.5-2 contains the P and E trajectory predictions
at several decision times. Initially, a lofted P response is
called for much as with MMT guidance. The two initially
predicted maximum maneuvers are avoided by both E paths, and
the optimal one in particular drops much faster than expected.
As time continues, the nominal curve lies roughly along the
pull-down and then the pull-up maximum-maneuver predictions
with the switch occurring at 4 sec.

If E were to stay on either the pull-up or the pull-down
maximum-maneuver prediction, the nominal interception should be
more accurate. It may be that due to the switch, the overall
altitude vs. time prediction leads P to expect a higher-
alitude intercept than actually occurs. Thus, P flies too
high and is forced against its g limit in a final pull down.
In the MMT case the same type of E trajectory is predicted all
of the time: a straight-line flight with maximum drag. Since
the MMT and the LR evasive maneuvers are both basically steep
dives, the upturning of later pull-up and pull-down LR predic-
tions causes a higher altitude to be expected at a given time
than predicted by the vertical (in this case) MMT predictions.

*The gradient searches used only attempt to cause E to hit the
target after the intercept time. No constraint is made in the
searches on what type of maneuver is to be used.
Figure 5.5-2a: The Predicted E and P Trajectories at $t_0 = 1$ second (Dashed Lines)
Figure 5.5-2b: The Predicted E and P Trajectories at $t_0 = 2$ seconds (Dashed Lines)
Figure 5.5-2c: The Predicted E and P Trajectories at $t_0 = 3$ seconds (Dashed Lines)
Figure 5.5-2d: The Predicted E and P Trajectories at $t_0 = 3.5$ seconds (Dashed Lines)
Figure 5.5-2e: The Predicted E and P Trajectories at $t_0 = 4$ seconds (Dashed Lines)
Figure 5.5-2f: The Predicted E and P Trajectories at $t_0=4.5$ seconds (Dashed Lines)
Since the MMT estimate is closer to the truth when E uses pull-ups and pull-downs, the MMT response to the nominal is more successful than the corresponding LR response. The LR optimal maneuver stays completely away from the maximum-maneuver predictions by flying an almost vertical path of relatively low aerodynamic drag. Thus, the shape of the E path is completely unpredicted and the altitude vs. time prediction is far from the truth. P, in response to the incorrect instructions from LR, flies too high and does not detect the rapid fall of E until it is too late. As in the response to the MMT optimal, P attempts an unsuccessful pull-down at the end.

As in the previous section, three-dimensional plots are presented to clarify the evasive strategy. The nominal case description, Figure 5.5-3, indicates that P has the situation well in hand in the range vs. time plane. There is little difference there between the actual and predicted P trajectories, and the E trajectory is well within the maximum-maneuver predictions at all times. The altitude vs. time plot (Figure 5.5-3c) shows why the intercept is not very good. As in the MMT case, a lofted P trajectory is initially requested, but the altitude of the predicted E paths are higher for LR. Look at the 6 sec. point of the initial predicted E path in Figure 5.4-3c and compare it to the initial predicted maximum-maneuver 6 sec. points of the present figure. The 6 sec. point for the MMT case is at 28,500 feet, but the 6 sec. points for the LR case
Figure 5.5-3a: Range vs. Altitude vs. Time for Nominal Evasion Against LR (Predictions for \( t_0 = 1 \) and 3.5 seconds in Dashed Lines)
Figure 5.5-3d: Isometric View
Figure 5.5-4d: Isometric View
are at 28,600 feet and 42,000 feet. Remember that LR instructs P to intercept both of these trajectories! This situation is repeated for the 3.5 sec. predictions. Unlike the range vs. time plane situation, the actual trajectory falls below the pairs of maximum-maneuver predictions. It is little wonder the interception performs so poorly compared to MMT. The optimal evasion makes the above problems worse for P. Examination of the altitude-time plane in Figure 5.5-4c reveals still larger discrepancies between the predicted maximum-maneuver pairs and the E flight-path actually flown. This is produced by the more rapid fall of E due to less control use in the optimal evasion. P is called up too high and then cannot turn down again fast enough to prevent E from slipping underneath. This situation is aggravated by a turn away from P which E starts at about 3.5 sec in the range-time plane. This turn postpones the intercept time and allows E to drop farther below P in the altitude-time plot before minimum distance is reached.

5.6 Summary of Results

Apparently, the first general rule for successful evasion is to "do the unexpected". In particular the maneuver should have the opposite character to that assumed by P. If a low-drag straight-line path is expected as with PN, fly a high-drag, weaving path. If a high drag straight-line path is expected as in MMT, fly a low-drag trajectory with some turn(s). If in addition to high drag, sharp turns are expected as in LR, fly
in a straight line or at least in a series of turns with as low a drag as possible. Obviously, if one type of interception law is expected and an opposing type is encountered, the evasion can be singularly unsuccessful.

In choosing the particular maneuver for E, a gradient search as described in earlier chapters will yield local optima. But, the search can be started much closer to an optimum by careful choice of a trial maneuver.* All of the cases tested indicate that E should constantly try to put P's cross-track relative position estimates in serious error. Cross-track error is harder for P to null out due to the speed advantage over E. For almost all of the cases, when right-angle encounters occur, this cross-track error generation is achieved by violating the drag assumptions discussed above. When E is forced to a "head on" encounter against MMT guidance, the cross-track error is brought about by a pull-up at the end. In this case, the turn disobeys the expected straight-line flight. Another tool in E's "bag of tricks" is the last-second turn to change the intercept time. This increases the along-track relative position error.

Coordinated with the cross-track technique, the intercept time can be delayed or brought forward in time to maximize the miss. Against PN, the evader flies slower than expected, making the cross-track error negative in sign. Then, a turn towards P

*With the cost of each step on the order of 2-4 min. IBM 370 computation time, this goal is quite important.
forces an earlier intercept time and P crosses the future evader path before E gets there.* When MMT or LR guidance is the threat, the cross-track distance is less than believed, giving a positive cross-track error. The proper move for E is to turn away from the intercept, delaying it. This allows E to cross the future pursuer path before P gets there.

These general conclusions hold when the P lateral acceleration limit is decreased or increased in the MMT guidance case. However, the relative emphasis between cross-track and along-track error generation depends on the acceleration limit. The high limit seems to encourage E to expand cross-track error at the expense of intercept-time alterations. On the other hand, the low P acceleration optimal calls for substantial change in intercept time while permitting P to more correctly estimate the cross-track relative position. Changing the approach geometry may require modifying this conclusion, however.

The performance of E in terms of nominal and optimal maneuvers is shown for each case in Figure 5.6-1. Note that the optimal maneuver against MMT is more successful in the original geometry than it is in the case of altered impact

* As mentioned before, PN cannot predict the effect of P's acceleration. From P's view the intercept could be improved by shutting off the engine in the last second.
Figure 5.6-1 The Miss Distance Achieved for (Nominal, Optimal) Maneuvers in the Cases Tested

*The nominal in this case is the MMT optimal.
*The nominal here is different than for the other MMT cases
(4832 feet vs. 2433 feet). Yet, if the long flight-time could be endured, E could fly the optimal maneuver of the original geometry until the intercept time and have enough momentum afterwards to fly far downrange to the shifted impact point. The terminal velocity would probably be subsonic and the flight time multiplied by at least a factor of three, but a better miss could be achieved in this way. This illustrates that the optimum found for the altered impact case is local but not global, although the other choice would probably be rejected on the grounds given.
6.1 Conclusions

It appears from the results of the previous chapter that there is an overall pattern in the different evasion maneuvers. In every case the near-intercept relative velocity between P and E has a larger component in the direction of P's flight (along track) than orthogonal to it (cross-track). This means that position prediction errors in the cross-track direction are more difficult for the P guidance to remove. Each evasion attempts to some degree to generate large cross-track errors. For near right-angle crossings, this must be done by E generating a different drag history than P expects. In the case of a pursuer guided by PN, the evader does a hard pull-down followed by a hard pull-up. Since PN "expects" low-drag, straight-line flight from E, the predicted positions for E are much lower in altitude than seen for the actual flight path. (In most cases the cross-track direction is vertical.) For the nominal MMT pursuer, the cross-track error is generated by flying a low-drag dive. This continuously fools the high-drag expectations assumed in the MMT predictions for E. Much the same maneuver works well against the LR pursuer, which expects even slower altitude changes than in the former case. When the nominal MMT pursuer is given first higher and then lower lateral acceleration capability than before, the results are as expected.
The "slow" P is more easily dodged and the "fast" P is harder to escape from. Another interesting point is that before intercept, evading the P with the largest lateral g limit (300) is done by a trajectory with the smallest (144 g's) peak acceleration. When the pursuit-evasion geometry is altered by moving the E impact point 50K feet downrange, the conflict is more nearly head-on. Then, the evasive maneuver against MMT requires a hard turn just before intercept to produce the desired cross-track prediction errors. Here, the lift is incorrectly predicted instead of the drag.

Generally, MMT guidance seems most difficult to evade of the three schemes.* LR guidance is more sophisticated but falls down by not trying to "seal the bottom" of the fan shaped region of Figure 3.3-1. It fails to guard against a low-drag maneuver as well as it guards against hard pull-ups and pull-downs. (MMT guidance does better than LR, even though it also neglects the low drag case, because it predicts straight flight paths which reach farther down into this region.) Proportional navigation appears to be the easiest for E to get away from. Although LR and MMT can be deceived into permitting large cross-track errors, both schemes predict the P acceleration nicely. Unfortunately for PN, it always expects P to continue with the present speed and consistently overestimates the time-to-go. However, the

*See Figure 5.6-1
best maneuver against the best of these interception schemes
does not necessarily do better or even as well against a less
effective law. The optimal maneuver against MMT yields a 4832
foot miss. The same maneuver against PN barely escapes with
a 1749 foot miss. Although an overall "best" evasion against
any interception is the ultimate goal in this research, prudent
advice at this point is "Know Thy Pursuer".

In addition to the insight gained about evasive maneuvers,
other thesis contributions are worth noting. The idea of SAA
approximate integration has not appeared in any of the litera-
ture seen to date by the author. It offers a straightforward
method for generating simple and accurate iterative solutions.
The stability and speed of SAA algorithms in solving aerodynamic
motion equations of thrusting and nonthrusting bodies has been
seen in countless instances in optimal evasion computations.
Such solutions allow useful predictions of future E and P posi-
tions, if the appropriate control functions are known or can be
assumed. Intelligent use is made of these predictions in two
new planar interception schemes: Minimum-Miss-Time and Least
Risk. Both laws use gradient searches to find optimum inter-
ception control histories of a particular form. (They differ in
the interception cost functions assumed.) In optimizing the
evasions against these schemes as well as against proportional
navigation, a partial comparative evaluation can be made of all
three interception policies.
6.2 Recommendations

In the author's opinion the primary thrust of future optimal evasion work should be to extend the study into the 3rd spatial dimension ("out-of-plane" maneuvers). PN and MMT guidance can be dealt with in 3 dimensions without too much difficulty. However, it may not be worthwhile to try to find a 3-dimensional interception scheme analogous to LR. The "fan shaped region" of Figure 3.3-1 would become a "cone shaped volume". Designing a guidance law to guard against the boundaries of such a figure would be an extremely complicated task and a computational nightmare. It may be a good idea to look into interception schemes (other than PN) designed to work in 3 dimensions. The ultimately-sought open-loop evasion should certainly be looked for in this space. Of course, the P and E models could at some point be brought closer to reality, making the research classified, and uncertainty in the knowledge of E and P states should be included (somehow) in the study. Also, the SAA technique for prediction of future E and P trajectories should be investigated. There may be better ways to do the same thing.

Looking at the pursuit problem alone, the evasive maneuvers found indicate how critical knowledge of the E drag characteristics is to the success of the interception. It is conceivable that a "super interception" policy could be designed to measure these characteristics after observation of E during the first second or so of reentry flight. By "learning" the E aerodynamics,
the scheme would make more accurate E predictions. It would be less likely to be fooled in the ways the optimal evasions fooled all three pursuit laws. However, the learning process would introduce a significant extra lag if carried on during the encounter. Only by simulations can it be shown whether or not this idea is of practical value.
APPENDIX A

EVADER AND PURSUER MODELS

For convenience, Figure 2.2-1 from Chapter 2 is repeated here. The evader body is assumed to be a blunted, right cone of revolution with a bluntness ratio $\psi$ of

$$\psi = \frac{\text{nose curvature radius}}{\text{base radius}} = .255$$

and a semiapex angle of $10^\circ$. Empirical data for blunted cones of this semiapex angle appear in a Masters degree thesis by Julius E. Harris. For this bluntness ratio the following empirical formulae fit Harris' data quite well (even though theoretically, $\frac{\partial \pi}{\partial \alpha} = 0$ at $\alpha = 0$ for a symmetrical body).

$$Q_n = 1.973 \alpha$$  \hspace{1cm} (A-1)

$$Q_4 = 1.12 + .357/\alpha$$  \hspace{1cm} (A-2)

Note the lack of Mach number dependencies for hypersonic flight.

The axial and normal accelerations can now be written.

$$A_a = -\frac{1}{2} \frac{\text{AREA}}{\text{EMASS}} \rho \frac{\partial v_e}{\partial t} |x_e| \frac{v}{|x_e|}$$

$$A_n = \frac{1}{2} \frac{\text{AREA}}{\text{EMASS}} \rho \frac{\partial v_e}{\partial t} |x_e| \frac{v}{|x_e|}$$

where

$$\text{AREA} \equiv \text{E base area} = 4 \text{ square feet}$$

$$\text{EMASS} \equiv \text{E mass} = 18.68 \text{ slugs}.$$}

$$X_e \equiv [X_{e1}, X_{e2}] = \text{range-altitude E position vector}$$

$$X_{dot} \equiv [X_{dot}, X_{dot}^2] = \text{range-altitude E velocity vector}$$

---


14 IBID, p.36.
Figure 2.2-1 The Evader Model Dynamics
The angle of attack is limited to 30°. The \( \alpha \) actually used in all the E equations is \( \alpha_{SAT} \), where

\[
\alpha_{SAT} = (30^\circ \text{ in rad}) \cdot \tanh\left(\alpha/(30^\circ \text{ in rad})\right)
\]  

(A-5)

The transformation from the body frame to the inertial frame is expressed by the matrix

\[
\begin{bmatrix}
\cos(\alpha + \phi_0) & -\sin(\alpha + \phi_0) \\
\sin(\alpha + \phi_0) & \cos(\alpha + \phi_0)
\end{bmatrix}
\]

where \( \alpha = \text{E flight-path angle} = \tan^{-1}\left(\frac{\dot{Y}}{\dot{X}}\right) \). To keep to a minimum the number of states in the problem, E is assumed to have no rotational inertia or autopilot lags. Thus, the total forces acting on E are due to gravity \( g \) and the aerodynamics described above. The state equation for E can now be written.

\[
\dot{X}_E = XDOTE
\]

(A-6)

\[
\dot{XDOTE} = \frac{1}{2} \frac{FAKEA}{MASS} X_E - \frac{X_E}{2900} \frac{1}{\|XDOTE\|^2} \begin{bmatrix}
\cos(\alpha + \phi_0) & -\sin(\alpha + \phi_0) \\
\sin(\alpha + \phi_0) & \cos(\alpha + \phi_0)
\end{bmatrix} \begin{bmatrix}
C_L \\
C_N
\end{bmatrix}
\]

\[-\begin{bmatrix}
\delta \\
g
\end{bmatrix}
\]  

(A-7)

The pursuer model is based on Nike-X Sprint missile data from a 1967 Aviation Week and Space Technology article.\(^{15}\)

Several assumptions are made in addition which, as the author has never seen classified data on this subject, are a compromise between reasonableness and simplicity. The four pieces of data gleaned from the referenced article are that the weight is 7500 lbs. at takeoff, the base diameter is 4.5 feet, the length is 27 feet and the thrust acceleration is in excess of 100 g's.

Assuming the Sprint to be a perfect, pointed, right cone of revolution, the cone semiapex angle \( \theta = \tan^{-1} \left( \frac{2.25}{57} \right) \approx 5.25^\circ \).

The P model used has a semiapex angle \( \theta / 2 \) of \( 2.6^\circ \), a length of 27 feet, a takeoff weight of 7500 lbs, and a takeoff thrust acceleration of 100 g's. The model is shown in cross-section in Figure A-1.* It is assumed that the inner cone length \( L_2 \) is initially 22 feet and grows linearly with time to equal \( L_1 \) at 8 seconds after launch. (Remember that all interceptions in Chapter 5 occur within 6 seconds from launch.) Thus,

\[
L_2 = \begin{cases} 
22 + \frac{5}{8} t & \text{for } t \leq 8 \\
27 & \text{for } t > 8 
\end{cases} \quad (A-8)
\]

The area of the inner cone is proportional to \( L_2^2 \). It is assumed that the thrust is proportional to this area and is thus also proportional to \( L_2^2 \). The thrust at takeoff of 100 g's added to the above results gives us a formula for the thrust.

\[
T = \begin{cases} 
1549.6 (22 + \frac{5}{8} t)^2 & \text{for } t \leq 8 \\
0 & \text{for } t > 8 
\end{cases} \quad (A-9)
\]

The takeoff mass \( M_0 \) is \( \frac{7500}{32.72} \) slugs. With the above \( L_2 \) model and the mass distribution seen in Figure A-1, the instantaneous mass \( \rho_{\text{mass}} \) is seen to follow this relationship:

\[
\rho_{\text{mass}} = \begin{cases} 
M_0 (1 - 0.600 \times 10^{-2} (484 t + (\frac{20}{3} x^2)) \frac{t^2}{2} + (\frac{8}{3})^2 \frac{t^2}{2}) & \text{for } t \leq 8.87 \\
0.2 M_0 & \text{for } t > 8 \text{ sec.} 
\end{cases} \quad (A-10)
\]

The mass is distributed about a center-of-mass with distance \( r_{\text{com}} \) from the base of the cone.

\[
r_{\text{com}} = \left( \frac{M_0}{\rho_{\text{mass}}} \right) \left( 2 + \frac{L_2^4 - L_1^4}{4 (L_2 - L_1)} \frac{\rho_{\text{mass}}}{\rho_0} - 2 \right) \quad (A-11)
\]

*This model is of course not realistic, although the basic concepts of thrust and mass (magnitude and distribution) variations with burning time are represented.
Note: Mass of Fuel = .8 \( M_p \)
Mass of Shell = .1 \( M_p \)
Point Mass = .1 \( M_p \)

Figure A-1: Pursuer Model in Cross-Section,
Noting Mass Distribution
The moment of inertia $I_p$ about this center-of-mass is now given.

$$I_p = M_p \left\{ 1.7428 \left[ \frac{3}{20} (\tan \frac{\beta}{2} + \frac{\pi}{6}) (\frac{L_4^2 - L_2^2}{L_4^2}) - \frac{L_2^2 (L_4 - L_2)}{16 L_4^2 L_2^2} \right] + \right. \\
(\frac{PMass}{M_p} - .3) \left[ p_{com} - (\frac{L_4^2 - L_2^2}{4 L_4^2 L_2^2}) \right]^2 + \\
.1 \left[ L_4^2 \left( \frac{1}{18} + \frac{1}{2} \tan \frac{\beta}{2} \right) + \left( \frac{L_4^2 - L_2^2}{p_{com}} \right)^2 \right] + .1 \left( 11 - p_{com} \right)^2 \right\}$$

(A-12)

The values of $T$ and $PMass$ are plotted vs. time in Figure A-2. Figure A-3 depicts $I_p$ and $p_{com}$ over time. The aerodynamic coefficients for the P cone have been derived using the Newtonian drag assumptions valid for hypersonic flight. They are first expressed for the case when $\alpha < \frac{\beta}{2}$ and there is no "shading" of one part of the cone surface by another.

$$PCA = \text{axial force coefficient} = 2 \left( \sin \frac{\beta}{2} \cos \alpha \right)^2 + \left( \cos \frac{\beta}{2} \sin \alpha \right)^2$$

(A-13)

$$PCN = \text{normal force coefficient} = \cos^2 \frac{\alpha}{2} \sin 2\alpha$$

(A-14)

$$p_{cm} = \text{pitch moment coefficient} = \frac{1}{8} \cos^3 \frac{\alpha}{2} \left( \frac{L_4^2 - L_2^2}{\tan \frac{\beta}{2}} \right) \sin 2\alpha$$

(A-15)

In the case when $\alpha > \frac{\beta}{2}$ and shading does occur, the formulae are much more complicated.

$$PCA = \frac{2}{\pi} \left\{ \sqrt{1 - \tan^2 \frac{\beta}{2}} \left( \frac{3}{8} \sin \beta \sin 2\alpha \pi \right) + \left( \sin^2 \frac{\alpha}{2} \cos^2 \alpha \pi + \frac{1}{8} \sin^3 \alpha \pi \cos^2 \frac{\alpha}{2} \right) \right\} \left[ \pi - \cos^{-1} \left( \frac{\tan \frac{\beta}{2}}{\tan \frac{\alpha}{2}} \right) \right]$$

(A-16)

$$PCN = \frac{2}{\pi} \left\{ \frac{1}{2} \sin \beta \sin 2\alpha \pi \left[ \pi - \cos^{-1} \left( \frac{\tan \frac{\beta}{2}}{\tan \frac{\alpha}{2}} \right) \right] + \sqrt{1 - \tan^2 \frac{\beta}{2}} \sin \beta \sin 2\alpha \pi \right\} \left( \frac{\alpha}{2} \cos^2 \frac{\beta}{2} \sin^2 \alpha \pi + \frac{1}{8} \sin^3 \alpha \pi \cos^2 \frac{\alpha}{2} \right)$$

(A-17)
Figure A-2: Thrust, Mass and Specific Force in g's Over the Time Range of Interest for the Pursuer
Figure A-3: Moment of Inertia and Center-of-Mass vs. Time for the Pursuer

Note: When the center of pressure is ahead of the center-of-mass, P is aerodynamically unstable.
\[
\begin{align*}
\text{PCM} &= S_{\text{C}} \tan(\alpha_p) \frac{(1 - 2 \tan^2 \frac{\alpha_p}{2})}{\tan \tan^2 \frac{\alpha_p}{2}} \left[ \sqrt{1 - \frac{\tan^2 \frac{\alpha_p}{2}}{\tan^2 \alpha_p}} \left( \frac{2 \cos^2 \frac{\alpha_p}{2} \sin^2 \alpha_p}{\tan \alpha_p} \right) \right. \\
& \quad + \frac{1}{3} \sin^2 \frac{\alpha_p}{2} \cos^2 \alpha_p \right] + \frac{1}{4} \sin^2 \alpha_p \sin \frac{\alpha_p}{2} \left[ \pi - \cos^{-1} \left( \frac{\tan \frac{\alpha_p}{2}}{\tan \alpha_p} \right) \right] \\
& \quad \left. \right] 
\end{align*}
\]

(A-18)

The three coefficients are plotted against \( \alpha_p \) in Figure A-4.

The lift and drag coefficients, computed from the equations

\[
\begin{align*}
\text{PCL} &= \text{PCN} \cos \alpha_p - \text{PCA} \sin \alpha_p \\
\text{PCD} &= \text{PCN} \sin \alpha_p + \text{PCA} \cos \alpha_p
\end{align*}
\]

(A-19a)

(A-19b)

are also plotted. Note that the approximations

\[
\begin{align*}
\text{PCN} &\approx 1.9843 \alpha_p - 0.07297 \alpha_p^2 (\text{Chapter 2}) \\
\text{PCL} &\approx 2 (\cos^2 \frac{\alpha_p}{2}) \alpha_p - 1.9898 \alpha_p (\text{autopilot model}) \\
\text{PCD} &\approx 0.01517 + 0.08174 \alpha_p + 2.911 \alpha_p^2 (\text{Chapter 2}) \\
\text{PCL} &\approx 1.9824 \alpha_p - 0.0713 \alpha_p^2 (\text{Chapter 2}) \\
\text{PCD} &\approx \text{PCN}
\end{align*}
\]

(A-20a)

(A-20b)

(A-20c)

(A-20d)

(A-20e)

are accurate for small \( \alpha_p \).

Figure 2.3-1 is repeated here as a convenience for deriving the \( P \) state equations. The pitch moment expressed by \text{PCM} is measured about the cone base and is caused by the normal force acting through the lever arm from the base to the center-of-pressure.\textsuperscript{16} The moment \( M \) is expressed in terms of the normal force \( F_N \) and the lever arm,

\[
M = 9 \text{\_a} \text{AREA} \ d \text{\_PCM} = (F_N) (\text{lever arm})
\]

(A-21)

where \( \text{\_a} \) is the free-stream dynamic pressure. But,

\[
F_N = 9 \text{\_a} \text{AREA} \text{\_PCN}.
\]

Thus,

\[
\text{lever arm} = \frac{M}{F_N} = \frac{\text{PCN} \ d}{\text{PCD}}
\]

(A-22)

This is a constant \( \text{\_a} (\frac{2 \tan^2 \frac{\alpha_p}{2}}{\tan \frac{\alpha_p}{2}}) \). Two torques operate in opposition to give pitch angle acceleration. They

\textsuperscript{16} \textit{IBID}, p.2,3.
Figure A-4: Pursuer Aerodynamic Coefficients and Approximations to Them vs. $\alpha_p$
Figure 2.3-1 The Pursuer Model Dynamics
arise from the aerodynamic moment and the engine thrust, \( T \). Calculating the net moment about the P center-cf-mass, we obtain

\[
T \text{ cone } \sin \delta - \frac{1}{2} \rho V^2 \text{PAREA } (\text{cone } - \frac{\text{Pcm }}{\text{Pcm } d}) \text{PCN} = I_p (\delta_p + \dot{\delta}_p)
\]  
(A-23)

where \( \text{PAREA } = \text{P cone base area} \), \( \gamma_p \) is the flight-path-angle, the air density \( \rho = \rho_0 e^{-\frac{Y_p^2}{2\pi c_0}} \) and \( V \) is the current velocity. The net acceleration in the negative lift direction is \( A_L \).

It is computed by summing the effects of engine thrust, aerodynamics and gravity.

\[
A_L = -\frac{1}{2} \rho V^2 \text{PAREA } \text{PCL} - \frac{T}{\text{Pmass }} \sin (\delta_p - \delta) + g \cos \delta_p
\]  
(A-24)

The engine gimbal angle \( \delta \) is limited in size to \( \pm 5^\circ \), and there is a 25 Hz. lag modelling the gimbal response. This frequency is chosen as a compromise between the desires for simulation accuracy and frequency separation between the gimbal lag and the overall, closed-loop response. The result is

\[
\delta_{SAT} = (5^\circ \text{ in rad.}) \tanh (\dot{\delta}_{OUT} / (5^\circ \text{ in rad.})
\]  
(A-25)

where

\[
\dot{\delta}_{OUT} = 157.1 (-\dot{\delta}_{OUT} + \delta)
\]  
(A-26)

The commanded lateral acceleration is also limited in magnitude. If the limiting quantity is denoted by \( GLIMIT \), the commanded acceleration to the autopilot is

\[
A_{LPSAT} = GLIMIT \tanh \left( \frac{A_{LPSAT}}{GLIMIT} \right)
\]  
(A-27)

\( (GLIMIT = 200 \text{ g's nominally}) \). The system block diagram appears in Figure A-5. Note that the linearized versions of the \( \text{PCN} \) and \( \text{PCL} \) formulae are used. The compensation gains \( C_1, C_3 \) and \( C_4 \) are picked via a linear analysis of the system. \( C_4 \) is chosen to
where 

\[ Q_1 \equiv -\frac{\rho V^2 PAcA}{I_p} \left( \frac{C_{d1}}{C_{d1}} d - \text{pcom} \right) \cos^2 \beta \]

\[ Q_2 \equiv \frac{\rho V^2 PAcA}{\text{PMass}} \cos^2 \beta \]

\[ Q_3 \equiv \frac{1}{I_p} \text{pcom} \]

\[ Q_4 \equiv \frac{T}{\text{PMass}} \]

Figure A-5: Pursuer Autopilot Diagram
damp the inner loop oscillation in the pitch control. The gains \( C_3 \) and \( C_4 \) form an integral-bypass network to drive steady-state following error to zero. Ignoring the high-frequency gimbal lag and the low-frequency pole-zero pair of the integral-bypass network, the autopilot loop is approximately second-order. The \( C_j \) are adjusted to give an equivalent damping ratio \( \zeta \) of .707. The equivalent natural frequency \( \omega_n \) is 13.3333 rad/sec for \( Q_j < 4 \pi \) and \( 2\sqrt{Q} \) otherwise. \( C_4 \) is chosen to give \(-90^\circ\) phase shift at \( \omega_n \). The formulae for achieving these ends follow.

\[
C_3 = \frac{1}{\left( \frac{Q_3 (Q_2 + Q_4)}{\omega_n^2 - Q_1} + Q_4 \right)} \quad (A-28)
\]
\[
C_4 = \frac{\omega_n}{\tan 80^\circ} \quad (A-29)
\]
\[
C_1 = 2J\omega_n \left( \frac{1 - Q_4 C_3}{Q_3} \right) \quad (A-30)
\]

By inspection of Figure A-5, the equations for the first four states follow directly.

\[
\begin{align*}
\dot{\delta} &= 157.1 \left[ -S - C_3 (A_{3100} + A_{3}) + C_4 X_3 - C_1 X_2 \right] \quad (A-31) \\
\dot{X}_1 &= X_2 \quad (A-32) \\
\dot{X}_2 &= Q_3 SIN S_{5AT} - Q_4 \alpha P \quad (A-33) \\
\dot{X}_3 &= -C_3 (A_{3100} - A_{3}) \quad (A-34)
\end{align*}
\]

Inspection of Figure 2.3-1 yields the non-field specific force equations through the center-of-mass in body coordinates. The body-to-inertial transformation is expressed by the matrix

\[
\begin{bmatrix}
\cos(\alpha P + \delta P), -\sin(\alpha P + \delta P) \\
\sin(\alpha P + \delta P), \cos(\alpha P + \delta P)
\end{bmatrix}
\]
Premultiplying the non-field specific force by this matrix and adding in the gravity term, we obtain the rest of the \( P \) state equations

\[
\dot{x}_p = X_{\text{DOT}P} \quad \text{(A-35)}
\]

\[
X_{\text{DOT}P} = \begin{bmatrix}
\cos (\alpha_p + \omega) , -\sin (\alpha_p + \omega) \\
\sin (\alpha_p + \omega) , \cos (\alpha_p + \omega)
\end{bmatrix}
\begin{bmatrix}
\cos \theta_{\text{SRF}} \\
\sin \theta_{\text{SRF}}
\end{bmatrix}^T

\frac{1}{\text{FMASS}}

+ \frac{1}{2} \rho V^2 \frac{\text{AREA}}{\text{FMASS}} \begin{bmatrix}
-\rho A \\
\rho A
\end{bmatrix}

+ \begin{bmatrix}
\mathbf{g}
\end{bmatrix} \quad \text{(A-36)}
\]

where \( x_p \) is the \( P \) position vector \( = [x_{p1}, x_{p2}] \) and \( X_{\text{DOT}P} \) is the \( P \) velocity vector \( = [x_{p1} \dot{p}, x_{p2} \dot{p}] \). The full \( \rho A \) and \( \rho A \) relationships are used in these equations.
B.0 Summary

This Appendix is the only one considered sufficiently lengthy to require a summary section. The optimal evasion program against MMT guidance requires a first-order description of how $A_L$ varies with changes in the current $P$ and $E$ states. (In this case the autopilot states have small effect and are thus ignored.) This is supplied by the subroutine MMTSRCH. The same information in the case of LR pursuit guidance is provided by the LRSRCH subroutine. Each of these subroutines calls other subroutines in turn, with a family of such programs supporting each optimal evasion program. Figure B-1 shows the respective "family trees".

![Diagram](image-url)

Figure B-1: Subroutines Needed for Evasion Optimization Against MMT and LR Guidance
Note that each subroutine is treated in a distinct section in this Appendix. Each section is broken into "blocks" and some of the blocks are divided into "parts". In some cases a part has "subparts". Comments identify these logical units. In addition each section is headed by a DIMENSION statement describing all arrays which are not (3X3), (1X3) or (3X1).

The subroutines are written in the MAC language, developed for the C. S. Draper Laboratory. This language is used because of the convenience in writing statements containing arrays and array operations. The statements are more easily interpreted if it is understood that

a. overbars (—) represent vectors.

b. overstars (*) represent matrices.

c. array indices start with zero (0).

d. superscript T and -1 mean matrix transpose and inversion, respectively.

e. statements are continued over onto following lines without special flags indicating the continuation.

f. as much as possible, the variable names are self descriptive.
B.1 Subroutine MMTSRCH

This subroutine accepts inputs from the MMT evader optimization program. It supplies the predicted E maneuver and the P predicted response as well as the required derivatives.

```
DIMENSION (GRADIEN T, 2) , (TGRADIEN T, 2) ,
(COSTSLOPE, 2), (COSTSLOPEC, 2) ,
(GRADIEN TGRADIEN T, 2X4) , (TGRADIEN TGRADIEN T , 2X4) , (COSTSLOPEGRADIEN T, 2X4) ,
(DDDXE0, 4), (DDDXP0, 4),
(DDMINDSTATE, 4) ,
(COSTSLOPEGRADIEN T , 2X4) , (D2COS T DALDALC, 2) , (D2AL0CDALDAL0, 2) ,
(D2AL0PRIMECDALDAL0, 2) , (D2AL0PRIMECDALDAL0PRIME, 2) ,
(DD2COSTDAL2, 2), (DALDXP0, 2X4) , (DALDXE0, 2X4) ,
(DGGDALTRANSPOSE, 2X2) , (DALCDAL, 2)
```

First, the parameter GLIMIT (= maximum magnitude of allowed $A_L$) is supplied, along with the current time $t_0$ and the P and E states. Then, the subroutines ETRAJNC and GDTSUB are called to get AL0NOW ($= A_L$ selected by MMT), AL0PRIMENOW ($= A_L'$ selected by MMT), DMIN (= predicted miss), GRADIENT ($= \frac{\partial M}{\partial A_L}$), GRADIEN TGRADIEN T ($= \frac{\partial^2 M}{\partial A_L^2}$), TGRADIEN TGRADIEN T ($= \frac{\partial M}{\partial A_L^2 \text{ miss time}}$). The next block computes the relationship between $A_L$ downstream and the "upstream" quantity.

$AL0NOWP=AL0NOW$

$SATFCN1=GLIMIT \tanh(AL0NOWP/GLIMIT)$
\[
\begin{align*}
\text{DSATFCN1DAL0} &= (1/\cosh(\text{ALONOWP/GLIMIT})) \\
\text{D2SATFCN1DAL0}^2 &= -(2/\text{GLIMIT}) \sinh(\text{ALONOWP/GLIMIT})/(\cosh(\text{ALONOWP/GLIMIT})) \\
\text{ALT6} &= \text{ALONOW} + \text{ALOPRIMENOW} (6-T0) \\
\text{ALT6P} &= \text{ALT6} \\
\text{SATFCN2}=\text{GLIMIT TANH(ALT6P/GLIMIT)}
\end{align*}
\]

\[
\begin{align*}
\text{DSATFCN2DAL0} &= (1/\cosh(\text{ALT6P/GLIMIT})) \\
\text{DSATFCN2DAL0PRIME} &= (6-T0) \\
\text{DSATFCN2DAL0}^2 &= -(2/\text{GLIMIT}) \sinh(\text{ALT6P/GLIMIT})/(\cosh(\text{ALT6P/GLIMIT})) \\
\text{D2SATFCN2DAL0} &= (6-T0) \text{ D2SATFCN2DAL02} \\
\text{D2SATFCN2DAL0PRIME} &= (6-T0) \text{ D2SATFCN2DAL02} \\
\text{DALCDAL} &= (\text{DSATFCN1DAL0}, 0, (1/(6-T0)) \text{ (DSATFCN2DAL0-DSATFCN1DAL0)}, \\
\text{DSATFCN2DAL0PRIME} &= (6-T0)) \\
-\text{D2AL0CDALDAL0} &= (\text{D2SATFCN1DAL02}, 0) \\
-\text{D2AL0PRIMECDALDAL0} &= (1/(6-T0)) \text{ (D2SATFCN2DAL02-D2SATFCN1DAL02),} \\
\text{D2SATFCN2DAL0PRIME} &= (6-T0) \text{ (D2SATFCN2DAL0DAL0PRIME,} \\
-\text{D2AL0PRIMECDALDAL0PRIME} &= (1/(6-T0)) \text{ (D2SATFCN2DAL0DAL0PRIME,} \\
-\text{D2SATFCN2DAL0PRIME2}) \\
\text{In this block the cost slope is computed for the "upstream"} \quad \tilde{A}_L \\
\text{COSTSLOPEC} &= \text{DMIN GRADIENT} + P \text{ TGRADIENT}
\end{align*}
\]
COSTSLOPE=COSTSLOPEC DALCDAL

This block computes the derivative of the downstream costslope with respect to P (position-velocity) state, E state and control. It is repeated for I=0,1,2.

IF I=0, DDMINSTATE=DDDXPO

IF I=1, DDMINSTATE=DDDXEO

IF I=2, DDMINSTATE=(GRADIENT,0,0)

COSTSLOPEGRADIENTC = DMIN GRADIENTGRADIENT

*GRADIENT DDMINSTATE

*P TGRADIENTGRADIENT

This block computes the upstream gradients of \( \frac{\partial \text{cost}}{\partial A_L} \) with respect to P (position-velocity) state, E state and upstream \( A_L \).

COSTSLOPEGRADIENT = DALCDAL COSTSLOPEGRADIENTC + DALCDAL

COSTSLOPEGRADIENTC

COSTSLOPEGRADIENT = DALCDAL COSTSLOPEGRADIENTC + DALCDAL

COSTSLOPEGRADIENTC

D2COSTDLDALC=(COSTSLOPEGRADIENTC +COSTSLOPEGRADIENTC )

16 17
This block uses the above results to compute the derivatives of upstream $A_L$ with respect to $P$ (position-velocity) state and $E$ state.

$$D2\text{CO}ST\text{DALD}A\text{LC} = D2\text{CO}ST\text{DALD}A\text{LC} \quad \text{DALCDAL}$$

$$D2\text{CO}ST\text{DALD}2 = \text{DALCDAL} \quad D2\text{CO}ST\text{DALD}A\text{LC} + \text{COSTSLOPEC} \quad (D2\text{ALOCDALDALO},$$

$$0, 0) + \text{DALCDAL} \quad D2\text{CO}ST\text{DALD}A\text{LC} + \text{COSTSLOPEC} \quad (D2\text{ALOPRIMECDALDALO},$$

$$2, 2) - \text{D2ALOPRIMECDALDAL0PRIME})$$

$$\text{COSTSLOPEGRADIENT} = (D2\text{CO}ST\text{DALD}2, 0, 0, D2\text{CO}ST\text{DALD}2, 0, 0)$$

$$16, 0, 2$$

$$D\text{GIALTRANSPOSE} = (\text{COSTSLOPEGRADIENT}, \text{COSTSLOPEGRADIENT})$$

$$16, 17$$

$$D\text{ALDXP}0 = - (D\text{GIALTRANSPOSE}) \quad \text{COSTSLOPEGRADIENT}$$

$$0$$

$$D\text{ALDXE}0 = - (D\text{GIALTRANSPOSE}) \quad \text{COSTSLOPEGRADIENT}$$

$$8$$

END OF MMTSRCH
B.2 Subroutine ETRAJNC

This subroutine accepts the (current) E state from MMTSRCH and estimate the desired straight-line E trajectory and its derivatives.

\[
\text{DIMENSION (EPSILON,4), (ONEZERO,4), (DVXEO,4),}
\]
\[
(DVDEPSILON,4), (DFCNDW0,4), (DFCNDV,4), (DEPSILONDFCN,4), (DVDO,4)
\]
\[
(DVODXEO,4), (DGMAMADOXEO,4), (DWOXEO,4), (DEPSILONDXEO,4),
\]
\[
(DXDEPSILON,4), (DZDEPSILON,4), (DXDV,4), (DZDV,4), (DXDGMAMAO,4),
\]
\[
(DZDGMAMAO,4), (DXDXEO,4), (DZDXEO,4), (DX123DXEO,3X4),
\]
\[
(DZ123DXEO,3X4), (DFCNDK1,4), (DK1DXEO,4)
\]

The given variables are GAMMAZERO (= \( \chi_{E} \)), INITIALALTITUDE (= E range), T0 (=present time) and VZERO (E velocity). The initialization prior to the SAA iteration follows.

\[
\text{G=32.17, MASS=18.68, AREA=4}
\]
\[
CD0=.120
\]
\[
CD=.785
\]
\[
SIGMA=24000
\]
\[
TAU3=6-T0, TAU1=TAU3/3, TAU2=2 \text{ TAU1}
\]
\[
* \text{ INVMATRIX=} \text{INVERSE(TAU1, TAU1, TAU1, TAU2, TAU2, TAU2, TAU3, TAU3, TAU3, TAU3, TAU3, TAU3)}
\]
\[
\text{TAU3)}
\]
BETAZERO = G MASS / (C0 AREA)
INITIALRHO = .002377 EXP (-INITIALALTITUDE / SIGMA)
WZERO = SIGMA (G / BETAZERO) (INITIALRHO / 2)
W0 = WZERO, V0 = VZERO, GAMMAO = GAMMAZERO
K1 = -SIN(GAMMAZERO) / SIGMA
K2 = -CD / (SIGMA C0)
2
EPSILONL = K2 VZERO WZERO
EPSILONS = 0, EPSILONA = 0, EPSILONJ = 0

This block is the SAA approximate integration of the \( \mathbf{v} \) equation for \( \mathbf{E} \). It is repeated until
\[
\left| \frac{\mathbf{v}^{3\text{NEW}} - \mathbf{v}^{3\text{OLD}}}{\mathbf{v}^{3\text{OLD}}} \right| < .001
\]

\( \mathbf{v}^{3\text{OLD}} = \mathbf{v}^{3\text{NEW}} \)

\( \mathbf{v}^{1} = \mathbf{v}^{0} + \mathbf{EpsilonL} \mathbf{Tau}^{1} + (1/2) \mathbf{EpsilonO} \mathbf{Tau}^{1} + (1/3) \mathbf{EpsilonO} \mathbf{Tau}^{3} + (1/4) \mathbf{EpsilonO} \mathbf{Tau}^{3} \)

IF \( \mathbf{V}^{1} < 0, \mathbf{V}^{1} = .1 \)

\( \mathbf{v}^{2} = \mathbf{v}^{0} + \mathbf{EpsilonL} \mathbf{Tau}^{2} + (1/2) \mathbf{EpsilonO} \mathbf{Tau}^{2} + (1/3) \mathbf{EpsilonO} \mathbf{Tau}^{4} + (1/4) \mathbf{EpsilonO} \mathbf{Tau}^{4} \)

IF \( \mathbf{V}^{2} < 0, \mathbf{V}^{2} = .1 \)

\( \mathbf{v}^{3} = \mathbf{v}^{0} + \mathbf{EpsilonL} \mathbf{Tau}^{3} + (1/2) \mathbf{EpsilonO} \mathbf{Tau}^{3} + (1/3) \mathbf{EpsilonO} \mathbf{Tau}^{4} + (1/4) \mathbf{EpsilonO} \mathbf{Tau}^{4} \)

IF \( \mathbf{V}^{3} < 0, \mathbf{V}^{3} = .1 \)

\( \text{FCN}^{1} = K2 \ V^{1} \ (WZERO + (K1 / K2) \ \text{LOG}(V^{1} / VZERO)) \)
FCN2=K2 V2  (WZERO+(K1/K2) LOG(V2/VZERO))

FCN3=K2 V3  (WZERO+(K1/K2) LOG(V3/VZERO))

- BOTTOMEPSILON=INVMATRIX (FCN1-EPsILONL,FCN2-EPsILONL,FCN3-
EPSILONL)

- EPSILON=(EPSILONL,BOTTOMEPSILON)

EPSILONS=EPSILON

EPSILONA=EPSILON

EPSILONJ=EPSILON

V3NEW=V3

Initialize the next block.

TREL=0

Repeat this part for TREL=0, TAU1, TAU2 and TAU3 to generate X0, X1, X2, X3, Z0, Z1, Z2 and Z3.

X=INITIALRANGE+COS(GAMMAZERO) (VZERO TREL+(1/2) EPSILONL TREL +
(1/6) EPSILONS TREL +(1/12) EPSILONA TREL +(1/20) EPSILONJ TREL )

Z=INITIALALTITUDE+SIN(GAMMAZERO) (VZERO TREL+(1/2) EPSILONL
2 TREL +(1/6) EPSILONS TREL +(1/12) EPSILONA TREL +(1/20) EPSILONJ
3 TREL )

IF TREL=0, X0=X, Z0=Z, TREL=TAU1

IF TREL=TAU1, X1=X, Z1=Z, TREL=TAU2
IF TREL=TAU1, X2=X, Z2=Z, TREL=TAU3
IF TREL=TAU3, X3=X, Z3=Z

In this block the derivatives of X1, X2, X3, and Z1, Z2, Z3 are computed with respect to the (present) E state vector.

T1=TAU1, T2=TAU2, T3=TAU3

* DVDEPSILON=(0, 0, 0, 0, T1, T1 /2, T1 /3, T1 /4, T2 /2, T2 /3, T2 /4,
2 3 4
T3, T3 /2, T3 /3, T3 /4)

- 2 2 2 2
DFCNW0=K2 (V0, V1, V2, V3)

* DFCNVDV= (2 K2 V0 W0, 0, 0, 0, -K1 V1 /V0, 2 K2 V1 (W0+(K1/K2) LOG (V1/V0)) +
2
K1 V2, 0, -K1 V2 /V0, 0, 2 K2 V2 (W0+(K1/K2) LOG (V2/V0)) +
K1 V3)

- DGAMMAODXEO=(0, 0, -SIN(GAMMAZERO), COS(GAMMAZERO)) /V0

- 2 2
DFCNKD1= (0, V1, LOG (V1/V0), V2, LOG (V2/V0), V3, LOG (V3/V0))

- DK1DXEO=- (COS(GAMMA)/SIGMA) DGAMMAODXEO

* ONEZERO=(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)

* DEPSILONDFCN=(1, 0, 0, 0, 0, INVMATRIX, 0, INVMATRIX, 0, INVMATRIX, 0, INVMATRIX, 0, INVMATRIX)

-(0, 0, 0, 0, 0, INVMATRIX, 0, INVMATRIX, 0, INVMATRIX)

- DVODXEO= (0, 0, COS(GAMMAZERO), SIN(GAMMAZERO))

- DWDV= (1, 1, 1, 1)
DW0DXE0 = (0, -W0/\sigma, 0, 0)

* DEPSILONDXE0 = ((1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1) - DEPSILONDFCN

* * -1 * - - -

DPCNDV DVDEPSILON) (DEPSILONDFCN DPCNDW0 DW0DXE0+)

* DEPSILONDFCN DPCNDV DV0DV0 DV0DXE0+DEPSILONDFCN DPCNDK1 DK1DXE0)

* - - * * *

DV0DXE0 = DV0DV0 DV0DXE0+DVDEPSILON DEPSILONDXE0

* DXDEPSILON = \cos(Gamma0) (0, 0, 0, 0, T1/2, T1/6, T1/12, T1/20, T2/2, T2/6, T2/12, T2/20, T3/2, T3/6, T3/12, T3/20)

* DZDEPSILON = \tan(Gamma0) DXDEPSILON

* DXDV = \cos(Gamma0) (0, 0, 0, 0, T1, 0, 0, 0, T2, 0, 0, 0, T3, 0, 0, 0)

* DZDV = \tan(Gamma0) DXDV

- DXDGAMMA0 = \tan(Gamma0) (0, -(X1-X0), -(X2-X0), -(X3-X0))

- DZDGAMMA0 = \cot(Gamma0) (0, (Z1-Z0), (Z2-Z0), (Z3-Z0))

* DXDXF0 = (1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0) *

* * * * *

DXDEPSILON DEPSILONDXE0+DXDV DV0DXE0+DXDGAMMA0 DGAMMA0DXE0

* DZDXE0 = (0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0) *

* * * * *

DZDEPSILON DEPSILONDXE0+DZDV DV0DXE0+DZDGAMMA0 DGAMMA0DXE0

* DX123DXE0 = (DXDXF0, DXDXE0, DXDXE0) 4 8 12

* DZ123DXE0 = (DZDXF0, DZDXE0, DZDXE0) 4 8 12
END OF ETRAJNC
B.3 Subroutine GDTSUB

This subroutine is called by and given inputs by LRSRCH. It predicts the P response to the latest predicted E trajectory (from ETARJNC) and supplies necessary first and second derivatives.

```
DIMENSION (GAMMAMATRIX, 2X4), (VMATRIX, 2X3), (DELTAMATRIX, 2X7), 
         (D, 2X7), (A, 3X8), 
         (ZMATRIX, 2X4), (AMATRIX, 3X8), (PCMATRIX, 8X8), (PMATRIX, 2X4), 
         (APCPGAMMAMATRIX, 3X4), (DELTAPAPCPGAMMAMATRIX, 2X4), 
         (VALOMATRIX, 2X4), (VAL01MATRIX, 2X2), (VAL02MATRIX, 2X2), 
         (DVDMATRIX, 2X2), (DXVECTOR, 3X4), (DZVECTOR, 3X4), 
         (DELTAAAS, 2), (GAMMADVDMATRIX, 4X2), 
         (GTEMP1, 4X4), (GTEMP2, 4X2), (GRADIENT, 2), 
         (TGRADIENT, 2), (TRUEGRADIENT, 2), (TRUEGRADIENT, 2), 
         (DDXZMATRIX, 3X4), (DELTAPRIMEMATRIX, 4X9), (APRIMEMATRIX, 3X9), 
         (PPRIMEMATRIX, 9X11), (TF1, 2), (TF2, 2), (TF3, 2), (XMATRIX, 2X7), 
         (ZPRIMEMATRIX, 2X7), (VMATRIX, 2X6), (VDROP, 9X9), (VER, 2X9), 
         (APRP, 11X11), (APR, 3X11), (VDRIAPRP, 9X11), (VDRIAPRP, 2X11), 
         (XIVG, 2X7), (ZIVG, 2X7), (VDRIAPRZIVGVGI, 2X7), (VI, 7X2), (DVSTATE, 
         2X5), (DXDSTATE, 2X6), (DZDSTATE, 2X6), (DXPDSTATE, 2X4), (DXDESTATE, 
         3X4), (DXPDAL, 2X4), (DZPDSTATE, 2X4), (DZPDAL, 2X4), (DVSTATECOMP, 
         3X5), (DVDPSTATE, 3X4), (DVDESTATE, 3X4), (DVDESTATE, 3X4), (DVDESTATE, 3X4), 
         (TEMPZ, 3X6), (DZDSTATE, 3X4), (DZDSTATE, 3X4), (DZDSTATE, 3X4), 
         (TEMFEG, 3X6), (DXDPSTATE, 3X4), (DXDSTATE, 3X4), 
```
The following variables are given by MMTSRCH: $T_0$ = present time; $ALOLODT_0$ = present control at $T_0$; $ALOPRIMEOLD$ = previous control slope; $ALOPRIMETHEN$ = initial guess at new control slope; $ALOTHEN$ = initial guess at new control bias; $GLIMIT$ = $\hat{A}_{SP}$ max size; $COMPOCOEF$ = $C_3$ (P computation gain) and $WCLOSEDLOOP$ = $\omega_n$, natural frequency autopilot and $Q_4$ = autopilot gain (See Appendix A for these terms.);

$X_0$, $Z_0$, $V_0$, $GAMMA_0$ = initial P position, velocity and flight-path-angle; $TAU_1$, $TAU_2$ and $TAU_3$ = future times (referenced from $T_0$) at which $E$ and $P$ positions will be predicted; $XE_1$, $XE_2$, $XE_3$, $ZE_1$, $ZE_2$, and $ZE_3$ = future $E$ positions (predicted); $DELXE_0$ = derivative of $XE_1$ (for I and even number) or derivative of $ZE_1$ (for I and odd number) with respect to the $E$ state.

This block initializes the SAA iteration for integrating the $\dot{V}$ equation.

$$G = 32.17, AREA = 17.53, SIGMA = 24000$$

$$RHO SUBZERO = .002377, M_0 = 233.1$$

$$T_2 = 6, T_1 = (T_2 + T_0) / 2, \text{CAPY}_1 = log((1 - .0804 T_1 - .00228 T_1) / (1 - .0804 T_0 - .00228 T_0))$$,

$$\text{CAPY}_2 = log((1 - .0804 T_2 - .00228 T_2) / (1 - .0804 T_0)$$
\[ \text{CAPZ1} = \log\left( -0.00456 \ T1 - 2.052 \right) - 0.00456 \ T0 + 0.0444 \]
\[ \text{CAPZ2} = \log\left( -0.00456 \ T2 - 2.052 \right) - 0.00456 \ T0 + 0.0444 \]
\[ \text{L1} = 27 \]

Do this part for I=0,1,2

IF I=0, T=T0
IF I=1, T=T1
IF I=2, T=T2

\[ \text{L2} = 22 + \frac{5}{8} T \]

\[ \text{THRUST} = 1549.5 \left( 22 + \frac{5}{8} T \right) \]

\[ \text{PCCM} = \left( \frac{2}{4} \right)^{\frac{2}{3}} \left( \frac{L1 - L2}{4(L1 - L2)} \right) \left( -0.00016602 \right) \]
\[ \left( 484 \ T + \left( \frac{22}{5/8} T + \frac{1}{3} \left( \frac{5}{8} \right) T \right) \right) \left( 1 - 0.00016602 \left( 484 \ T + \right) \left( 22 \frac{5}{8} T + \frac{1}{3} \left( \frac{5}{8} \right) T \right) \right) \]

\[ \text{PMASS} = \text{M0} \left( 1 - 0.00016602 \left( 484 \ T + \left( \frac{22}{5/8} T + \frac{1}{3} \left( \frac{5}{8} \right) T \right) \right) \right) \]

\[ \text{ALI} = \text{ALOTHEN} + \text{ALOPRIME} \text{THEN} (T-T0) \]

\[ \text{ALPHAI} = 1 - \text{SGN}(\text{ALI}) \]

IF I=0, MASS0 = PMASS, THRUST0 = THRUST, CM0 = PCCM, ALPHAO = ALPHAI
IF I=1, MASS1 = PMASS, THRUST1 = THRUST, CM1 = PCCM, ALPHAI1 = ALPHAI
IF I=2, MASS2 = PMASS, THRUST2 = THRUST, CM2 = PCCM, ALPHAI2 = ALPHAI

Continue with the initialization. Note that AL0 and ALOPRIME are converted from "upstream" to "downstream" values. (See Appendix 3.1 for definitions of "upstream" and "downstream").
RH00=RHOSUBZERO*EXP(-Z0/SIGMA)
SINHALFB=SIN(10 DEGTORAD/2), COSHALFB=COS(10 DEGTORAD/2),
TANHALFB=SINHALFB/COSHALFB
D=2 L1 TANHALFB
L2=22+(5/8)T0
HFGAIN=-(Q4 COMPCOEF3/(1-Q4 COMPCOEF3))/(S/2)
TRISE=S/WCLOSEDLOOP
ALT6=ALO+ALOPRIME(6-T0)
ALO=GLIMIT TANH(ALO/GLIMIT)
ALOPRIME=GLIMIT TANH(ALT6/GLIMIT-AL0)/(6-T0)
RT=TRISE/(1-(ALOPRIME-ALOPRIMEOLD)*TRISE/((ALO-AOLDT0)*(1-HFGAIN )))
GAMMADOTAREA=(1/2)*(1-HFGAIN)*(ALO-AOLDT0)*RT

This block performs the SAA integration. It is repeated until

\[ \left| \frac{X2-X2OLD}{X2OLD} \right| < .001 \]

V1=VO-438.6(2.5968-DELTA4A)(T1-T0)+(-219.298(182.81-DELTA4S+2
DELTA4 T0)+7733.1(2.5968-DELTA4A))CAPY1+(8.0115(3217.5+DELTA4 T0
2
-DELTA4-DELTA4 T0)-141.256(182.81-DELTA4S+2 DELTA4 T0)+8494.97
(2.5968-DELTA4A))CAPZ1
V2=VO-438.60(2.5968-DELTA4A)(T2-T0)+(-219.298(182.81-DELTA4S+2
DELTA4 T0)+7733.1(2.5968-DELTA4A))CAPY2+(8.0115(3217.5+DELTA4
2
T0-DELTA4-DELTA4 T0)-141.256(182.81-DELTA4S+2 DELTA4 T0)+
8494.97(2.5968-DELTA4A))CAPZ2
F1=T1-T0, F2=T2-T0
GAMMA1=GAMMA0-AL0*(5/(12 V0)+2/(3 V1)-1/(6 V2))F1+ALOPRIME(2/(3
2
V1)-1/(6 V2))F1+GAMMADOTAREA/V0
GAMMA2 = GAMMA0 - ALO \left( \frac{1}{3} V0 + 4 \left( \frac{1}{3} V1 + 1 \left( \frac{1}{3} V2 \right) \right) F1 - ALOPRIME \left( \frac{4}{3} V1 + 2 \left( \frac{1}{3} V2 \right) F1 \right) \right) + GAMMADOTAREA / V0

In this part of the block, X1, X2, Z1 and Z2 are computed. Repeat this part for I=1, 2.

IF I=1, T=F1
IF I=2, T=F2
IF I>2, T=INTEGER(T0) + I - 2 - T0

\[
X = X0 + V0 \cos(Gamma0) \left( (1/3/4 F1) T + (1/(6 F1)) T \right) + V1 \cos(Gamma1) \left( (1/3 F1) T - (1/(3 F1)) T \right) + V2 \cos(Gamma2) \left( (1/(4 F1)) T + (1/(6 F1)) T \right)
\]

\[
Z = Z0 + (V0 \sin(Gamma0) \left( (1/3/4 F1) T + (1/(6 F1)) T \right) + V1 \sin(Gamma1) \left( (1/3 F1) T - (1/(3 F1)) T \right) + V2 \sin(Gamma2) \left( (1/(4 F1)) T + (1/(6 F1)) T \right)
\]

ABSTIME = T0 + T

IF I=1, X1 = X, Z1 = Z
IF I=2, X2OLD = X2, X2 = X, Z2 = Z

Continue with the block.

RHOC = RHOSUBZERO \exp(-Z0/\Sigma)
RHOC1 = RHOSUBZERO \exp(-Z1/\Sigma), RHOC2 = RHOSUBZERO \exp(-Z2/\Sigma)
P00 = - (-G \cos(Gamma0) + ALO)
P01 = - (-G \cos(Gamma1) + ALO + ALOPRIME F1)
P02 = - (-G \cos(Gamma2) + ALO + ALOPRIME F2)
P10 = (1/2) (RHOC V0 AREA/MASS0) \left( (1 - 8.362 / CM0) 1.7473 - 1.773B \right) -
\[ P_{11} = \frac{1}{2} \left( \frac{\rho_0}{v_1} \right) \frac{\text{AREA/MASS}}{(1-8.862/cm)1.7473-1.7738} \] - \[ \text{THUST1/MASS1} \]

\[ P_{12} = \frac{1}{2} \left( \frac{\rho_0}{v_2} \right) \frac{\text{AREA/MASS2}}{(1-8.862/cm)1.7473-1.7738} \] - \[ \text{THUST2/MASS2} \]

\[ P_{20} = +\left( \frac{1}{2} \right) \left( \frac{\rho_0}{v_0} \right) \frac{\text{AREA/MASS0}}{\text{SGN}(\alpha_0)} \left( 2.1908 \left( 1-8.862/cm \right) - 1.5885 \right) \]

\[ P_{21} = +\left( \frac{1}{2} \right) \left( \frac{\rho_1}{v_1} \right) \frac{\text{AREA/MASS1}}{\text{SGN}(\alpha_1)} \left( 2.1908 \left( 1-8.862/cm \right) - 1.5885 \right) \]

\[ P_{22} = +\left( \frac{1}{2} \right) \left( \frac{\rho_2}{v_2} \right) \frac{\text{AREA/MASS2}}{\text{SGN}(\alpha_2)} \left( 2.1908 \left( 1-8.862/cm \right) - 1.5885 \right) \]

IF \( P_{10}/(2 P_{20}) \) < \( p_{00}/P_{20} \), \( \alpha_0 = -P_{10}/(2 P_{20}) \), \( A_{20} = 0 \), GO TO ASKP1, OTHERWISE

\[ \alpha_{0i} = -(\frac{P_{10}}{2 P_{20}}) + \sqrt{((\frac{P_{10}}{2 P_{20}}) - p_{00}/P_{20})} \]

\[ \alpha_{0j} = -(\frac{P_{10}}{2 P_{20}}) - \sqrt{((\frac{P_{10}}{2 P_{20}}) - p_{00}/P_{20})} \]

IF \( \text{ABS}(\alpha_{0i}) > \text{ABS}(\alpha_{0j}) \), \( \alpha_0 = \alpha_{0j}, A_{20} = -1 \), OTHERWISE \( \alpha_0 = \alpha_{0i}, A_{20} = +1 \)

location ASKP1

\[ \text{DOCVERMO} = \frac{1}{2} \rho_0 v_0 \frac{\text{AREA/(1/M0)}}{(0.01519 - \text{SGN}(\alpha_0)0.0149119 \text{ALPHA}0 + 3.07089 \text{ALPHA}0) \]}

\[ \text{IF} \ (P_{11}/(2 P_{21})) < p_{01}/P_{21}, \alpha_{11} = -P_{11}/(2 P_{21}), \ A_{21} = 0, \]

GO TO ASKP2, OTHERWISE
\[ \text{ALPHA1I} = - \frac{(P_1 (1/2) P_2)}{P_0} + \text{SQRT} \left( \left( \frac{P_1 (1/2) P_2}{P_0} \right)^2 - 1 \right) \]

\[ \text{ALPHA1J} = - \frac{(P_1 (1/2) P_2)}{P_0} - \text{SQRT} \left( \left( \frac{P_1 (1/2) P_2}{P_0} \right)^2 - 1 \right) \]

\[ \text{IF ABS(ALPHA1I)} > \text{ABS(ALPHA1J)}, \text{ALPHA1=ALPHA1I, AS1=-1, OTHERWISE ALPHA1=ALPHA1J, AS1=+1} \]

\[ \text{location ASKp2} \]

\[ \text{IF (P_1 (1/2) P_2) < P_0, \text{ALPHA2=-P_1 (1/2) P_2, AS2=0, GO TO ASKp3, OTHERWISE} } \]

\[ \text{ALPHA2I} = - \frac{(P_1 (1/2) P_2)}{P_0} + \text{SQRT} \left( \left( \frac{P_1 (1/2) P_2}{P_0} \right)^2 - 1 \right) \]

\[ \text{ALPHA2J} = - \frac{(P_1 (1/2) P_2)}{P_0} - \text{SQRT} \left( \left( \frac{P_1 (1/2) P_2}{P_0} \right)^2 - 1 \right) \]

\[ \text{IF ABS(ALPHA2I)} > \text{ABS(ALPHA2J), ALPHA2=ALPHA2I, AS2=-1, OTHERWISE ALPHA2=ALPHA2J, AS2=+1} \]

\[ \text{location ASKp3} \]

\[ \text{D1OVERM0 = (1/2) RH01 V1 AREA(1/M0) (0.0159-ABS(ALPHA1) .0149119} \]

\[ + 3.07089 \ \text{ALPHA1})} \]

\[ \text{D2OVERM0 = (1/2) RH02 V2 AREA(1/M0) (0.0159-ABS(ALPHA2) .0149119} \]

\[ + 3.07089 \ \text{ALPHA2})} \]

\[ - \text{DELTAAS=INVERSE(F1,F1,F2,F2) (D1OVERM0-D2OVERM0, D2OVERM0-D2OVERM0)} \]

\[ \text{DELTALOLD=DELTAL, DELTASOLD=DELTAS, DELTAAOLD=DELTAA} \]
This block initializes the first derivative computation.

\[ L_1 = \tau_1, L_2 = \tau_2, L_3 = \tau_3 \]

\[ \Lambda_{\text{matrix}} = (L_1, L_1, L_1, L_2, L_2, L_3, L_3, L_3) \]

\[ \Lambda_{\text{DAI}} = \text{inverse} \left( \Lambda_{\text{matrix}} \right) \]

\[ A_{\text{DENOM}} = (1 - (A_{\text{OLPRIME}} - A_{\text{OLPRIMEOLD}}) \text{ TRISE} / (A_{\text{OL}} - A_{\text{OLDTON}}) (1 - 2 H_{\text{F Gain}})) \]

\[ D_{\text{TRDALO}} = -(1 / A_{\text{DENOM}}) \text{ TRISE} (A_{\text{OLPRIME}} - A_{\text{OLPRIMEOLD}}) / ((2 A_{\text{OL}} - A_{\text{OLDTON}}) (1 - H_{\text{F Gain}})) \]

\[ D_{\text{TRDALOPRIME}} = (1 / A_{\text{DENOM}}) \text{ TRISE} / ((A_{\text{OL}} - A_{\text{OLDTON}}) (1 - H_{\text{F Gain}})) \]

\[ D_{\text{READALO}} = (1 / 2) (1 - H_{\text{F Gain}}) (RT + (A_{\text{OL}} - A_{\text{OLDTON}}) D_{\text{TRDALO}}) \]

\[ D_{\text{READALOPRIME}} = (1 / 2) (1 - H_{\text{F Gain}}) (A_{\text{OL}} - A_{\text{OLDTON}}) D_{\text{TRDALOPRIME}} \]

\[ M_{\text{sbo}} = M_0, P_{C1} = C_1, P_{CM2} = C_2, P_{A0} = \alpha_0, P_{A1} = \alpha_1, P_{A2} = \alpha_2 \]

\[ X_{P0} = X_0, Z_{P0} = Z_0, \gamma_{\text{ZERO}} = \gamma_{\text{MA}} \]

\[ P_{\text{FEA}} = \text{AREA} \]

This block is repeated for \( I = 1, 2, 3 \) to compute \( X_{P1}, X_{P2}, X_{P3}, Z_{P1}, Z_{P2} \) and \( Z_{P3} \).

\[ X_{P1} = X_0 + V_0 \cos(\gamma_{\text{MA}}) \left( L_1 - (3 / (4 F_1)) L_1 + (1 / (6 F_1)) L_1 \right) \]

\[ + V_1 \cos(\gamma_{\text{MA}}) \left( (1 / F_1) L_1 - (1 / (3 F_1)) L_1 \right) + V_2 \cos(\gamma_{\text{MA}}) \left( \left( 2 - (1 / (4 F_1)) L_1 + (1 / (6 F_1)) L_1 \right) \right) \]
$Z_P = Z_C + V_0 \sin(\gamma_0) \left( \left( \frac{1}{3} \left( \frac{1}{6} \right) \right) \left( \frac{1}{3} \right) \right) + \left( \frac{1}{6} \left( \frac{1}{3} \right) \right) \left( \frac{1}{3} \right)$

$V_1 \sin(\gamma_1) \left( \left( \frac{1}{3} \left( \frac{1}{6} \right) \right) \left( \frac{1}{3} \right) \right) + V_2 \sin(\gamma_2) \left( \left( \frac{1}{3} \left( \frac{1}{6} \right) \right) \left( \frac{1}{3} \right) \right)$

IF $i = 1$, $X_P = X_P^I$, $Z_P = Z_P^I$, $D_1 = D_1^I$

IF $i = 2$, $X_P = X_P^I$, $Z_P = Z_P^I$, $D_2 = D_2^I$

IF $i = 3$, $X_P = X_P^I$, $Z_P = Z_P^I$, $D_3 = D_3^I$

At this point the subroutine DISTFIT is called. The miss and miss-time are computed along with appropriate first and second derivatives. The variables returned follow:

$T_{MINDEL}$, $T_{TRUEDMIN}$, $D_{TMINDXE}$, $D_{TMINDXP}$, $D_{TMINDZE}$, $D_{TMINDZP}$

$D_{DMINDXE}$, $D_{DMINDXP}$, $D_{DMINDZE}$, $D_{DMINDZP}$, $D_{2TMINDXP2}$, $D_{2TMINDZPDXP}$

$D_{2TMINDXPDZP}$, $D_{2TMINDZP2}$, $D_{2DMINDXP2}$, $D_{2DMINDZPDXP}$, $D_{2DMINDZP2}$

The desired first derivatives are computed in this block.

$D_{2TMINDXEDXP} = -D_{2TMINDXP2}$, $D_{2TMINDZEDXP} = -D_{2TMINDZPDXP}$

$D_{2TMINDXEDZP} = -D_{2TMINDXPDZP}$, $D_{2TMINDZEDZP} = -D_{2TMINDZP2}$

$D_{2DMINDXEDXP} = -D_{2DMINDXP2}$, $D_{2DMINDZEDXP} = -D_{2DMINDZPDXP}$

$D_{2DMINDXEDZP} = -D_{2DMINDXPDZP}$, $D_{2DMINDZEDZP} = -D_{2DMINDZP2}$
TRUETMIN = TMINREL + T0

* Gammamatix = \(-5/(12 \ V0) + 2/(3 \ V1) - 1/(12 \ V2)\) \(F1 + DAREADAL0/\ V0,\)

\(-2/(3 \ V1) - 1/(6 \ V2)\) \(\ F1 + DAREADAL0PRIME/\ V0,\)

\((2/(3 \ V1)) (AL0 F1 + AL0PRIME F1), -1/(12 \ V2)\)

\((2/(3 \ V1)) (AL0 F1 + 2 AL0PRIME F1), -1/(3 \ V0) + 4/(3 \ V1) + 1/(3 \ V2)\)

\(F1 + DAREADAL0/\ V0, -(4/(3 \ V1) + 2/(3 \ V2)) \(\ F1 + DAREADAL0PRIME/\ V0,\)

\((4/(3 \ V1)) (AL0 F1 + AL0PRIME F1), +1/(3 \ V2)\) (AL0 F1 + 2 AL0PRIME F1))

* VMATRIX = \(-8.0115 \ CAPZ1, 219.298 \ CAPY1 + (8.0115 \ T0 + 141.256 \ CAPZ1, 438.60 \ F1 - (2 \ T0 219.298 + 7733.1) \ CAPY1 - (8.0115 \ T0 + 141.256 + 8494.97) \ CAPZ2, 219.298 \ CAPY2 + (8.0115 \ T0 + 141.256) \ CAPZ2, 438.60 \ F2 - (2 \ T0 219.298 + 7733.1) \ CAPY2 - (8.0115 \ T0 + 141.256 + 8494.97) \ CAPZ2)\)

DDM0DRHO0 = (1/2) \(\ V0 \ PAREA (1/MSUB0) (.01519 - SGN(\ PA0)) .0149119 \ PA0 + 3.0709 \ PA0 \))

DDM0DRHO1 = (1/2) \(\ V1 \ PAREA (1/MSUB0) (.01519 - SGN(\ PA1)) .0149119 \ PA1 + 3.0709 \ PA1 \))

DDM0DRHO2 = (1/2) \(\ V2 \ PAREA (1/MSUB0) (.01519 - SGN(\ PA2)) .0149119 \ PA2 + 3.0709 \ PA2 \))

DDM0DALPHAO = (1/2) \(\ RHOO \ V0 \ PAREA (1/MSUB0) (-.0149119 \ SGN(\ PA0))\)
\[ +2(3.0709) PA_0 \]

\[ \text{DDMODALPHA}_1 = (1/2) \text{RHO}_1 \text{ V1} \cdot \text{PAREA} (1/\text{MSUB}_0) (-0.0149119 \text{ SGN(PA1)} +2 3.0709 \text{ PA1}) \]

\[ \text{DDMODALPHA}_2 = (1/2) \text{RHO}_2 \text{ V2} \cdot \text{PAREA} (1/\text{MSUB}_0) (-0.0149119 \text{ SGN(PA2)} +2 3.0709 \text{ PA2}) \]

\[ \text{DDMODV0} = (2/\text{V0}) \text{DOOVERMO} \]

\[ \text{DDMODV1} = \text{RHO}_1 \text{ V1} \cdot \text{PAREA} (1/\text{MSUB}_0) (0.01519 - \text{SGN(PA1)} 0.0149119 \text{ PA1} +2 3.0709 \text{ PA1}) \]

\[ \text{DDMODV2} = \text{RHO}_2 \text{ V2} \cdot \text{PAREA} (1/\text{MSUB}_0) (0.01519 - \text{SGN(PA2)} 0.0149119 \text{ PA2} +2 3.0709 \text{ PA2}) \]

* \[ \text{DELTAMATRIX} = ((4/2) \text{DDMODRHO}_1, (-1/2) \text{DDMODRHO}_2, (-3/2) \text{DDMODALPHA}_0, (4/2) \text{DDMODALPHA}_1 \]

\[ \text{DDMODALPHA}_2, (4/2) \text{DDMODV1}, (-1/2) \text{DDMODV2}, (-4/2) \text{DDMODRHO}_1, (2/2) \text{DDMODRHO}_2, (2/2) \text{DDMODALPHA}_0, (-4/2) \text{DDMODALPHA}_1, (2/2) \text{DDMODALPHA}_2, (-4/2) \text{DDMODV1}, (2/2) \text{DDMODV2} \]

* \[ \text{ZMATRIX} = ((2/3) F_1 \text{ SIN(GAMMA}_1), -(1/12) F_1 \text{ SIN(GAMMA}_2), (2/3) F_1 \text{ V1 COS(GAMMA}_1), -(1/12) F_1 \text{ V2 COS(GAMMA}_2), \text{SIN(GAMMA}_1) ((F_2 \text{ F}_1) \text{ F}_2 / (3 \text{ F}_1)) \]

\[ 3 \text{ F}_1 \text{ F}_2, \text{SIN(GAMMA}_2) ((-1/4 \text{ F}_1) F_2 + (1/6 \text{ F}_1) F_2 ), \]

\[ \text{V1 COS(GAMMA}_1) ((1/3 \text{ F}_1) F_2 + (1/6 \text{ F}_1) F_2 ) \]

\[ \text{V2 COS(GAMMA}_2) \]

\[ ((-1/4 \text{ F}_1) F_2 + (1/6 \text{ F}_1) F_2 ) \]
\[ PSQRTTERM0 = \text{SORT}\left( \frac{P10}{(2 \ P20)} - \frac{P00}{P20} \right) \]

\[ PSQRTTERM1 = \text{SORT}\left( \frac{P11}{(2 \ P21)} - \frac{P01}{P21} \right) \]

\[ PSQRTTERM2 = \text{SORT}\left( \frac{P12}{(2 \ P22)} - \frac{P02}{P22} \right) \]

\[ \text{AMATRIX} = \frac{1}{2} A0 \left( \frac{-P10}{(2 \ P20)} \right) + \frac{1}{2} \left( \frac{P00}{P20} \right) / PSQRTTERM0, \left( \frac{P10}{(2 \ P20)} \right) + \frac{1}{2} \left( \frac{P11}{(2 \ P21)} \right) / PSQRTTERM1, \]

\[ \left( \frac{P11}{(2 \ P21)} \right) + A1 \left( \frac{P11}{(2 \ P21)} \right) / PSQRTTERM1, \]

\[ \left( \frac{P12}{(2 \ P22)} \right) + A2 \left( \frac{P12}{(2 \ P22)} \right) / PSQRTTERM2, \]

\[ \left( \frac{P12}{(2 \ P22)} \right) + A1 \left( \frac{P12}{(2 \ P22)} \right) / PSQRTTERM2, \]

\[ \text{PCMATRIX} = -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, -F1, 0, 0, 0, 0, -\sin(\gamma1), \]

\[ 0, 0, 0, 0, 0, 0, 0, -1, -F1, 0, 0, 0, 0, -\sin(\gamma1), \]

\[ 0, 0, (1/2) (\text{PAREA} \ V1 / \text{MASS1}) \left( (1 - 8.862/\text{PCM1}) 1.7473 - 1.7738 \right), \]

\[ 0, (\text{RHO1} \ V1 \text{PAREA/MASS1}) \left( (1 - 8.862/\text{PCM1}) 1.7473 - 1.7738 \right), 0, 0, 0, 0, \]

\[ 0, (1/2) (\text{V1 PAREA/MASS1}) \text{SGN} (\alpha1) (2.1908 (1 - 8.862/\text{PCM1}) - 1.5885), 0, (\text{RHO1} \ V1 \text{PAREA/MASS1}) \text{SGN} (\alpha1) (2.1908 (1 - 8.862/\text{PCM1}) - 1.5885), 0, 0, 0, -1, -F2, 0, 0, 0, 0, 0, \]

\[ -\sin(\gamma2), 0, 0, 0, (1/2) (V2 \text{PAREA}) \]
MASS2 \((1 - 8.862/\text{PCM2}) 1.7473 - 1.7738\), 0, (RHO2 V2 PAREA/MASS2)

\((1 - 8.862/\text{PCM2}) 1.7473 - 1.7738\), 0, 0, 0, 0, 0, (+1/2) (V2 PAREA/

MASS2) SGN(ALPHA2) (2.1908 (1 - 8.862/\text{PCM2}) - 1.5885),

0, (RHO2 V2 PAREA/MASS2) SGN(ALPHA2) (2.1908 (1 -

8.862/\text{PCM2}) - 1.5885), 0, 0)

* PMATRIX=-(1/SIGMA) RHSUBZERO (EXP(-Z1/SIGMA), 0, 0, EXP(-Z2/

SIGMA)) ZMATRIX (0, 0, 1, 0, 0, 0, 0, 1, GAMMAMATRIX)

* * * * APCPGAMMAMATRIX=AMATRIX PCMATRIX (1, 0, 0, 0, 0, 1, 0, 0, PMATRIX,

* 0, 0, 1, 0, 0, 0, 0, 1, GAMMAMATRIX)

* DELTAPAPCPGAMMAMATRIX=DELTAMATRIX (PMATRIX, APCPGAMMAMATRIX, 0, 0,

1, 0, 0, 0, 0, 1)

* VALOMATRIX=VMATRIX (1/2) RHO0 V0 (PAREA/KSUB0) (-.0149119 SGN(

PA0 + 2 3.0709 PA0) (1/2) AS0 (-1/P20) (1/SQRT ((P10/(2 P20)))

-(P00/P20)) (-1), 0, 0, 0, DELTAPAPCPGAMMAMATRIX)

* VAL01MATRIX=(VALOMATRIX, VALOMATRIX, VALOMATRIX, VALOMATRIX)

* 0 1 4 5

* VAL02MATRIX=(VALOMATRIX, VALOMATRIX, VALOMATRIX, VALOMATRIX)

* 2 3 6 7

* DVDALMATRIX=-INVERSE(VAL02MATRIX-(1, 0, 0, *)) VAL01MATRIX

* GTEMP1 = (0, 0, 1, 0, 0, 0, 0, 1, GAMMAMATRIX)

* GTEMP2=(1, 0, 0, 1, DVDALMATRIX)

* * *

GAMMADVDALMATRIX=GTEMP1 GTEMP2
This part is repeated for I=0,1,2

\[
\begin{align*}
\text{IF } I &= 0, \quad \text{LI} = L_1 \\
\text{IF } I &= 1, \quad \text{LI} = L_2 \\
\text{IF } I &= 2, \quad \text{LI} = L_3 \\
\end{align*}
\]

\[
\begin{align*}
\text{DXVECT} &= (\cos(\Gamma_{\text{MA}1}) \left( (1/F_1)\text{LI} - (1/(3 \ F_1 ))\text{LI} \right) , \cos(\Gamma_{\text{MA}2}) \\
& \quad \left( (-1/(4 \ F_1))\text{LI} + (1/(6 \ F_1 ))\text{LI} \right) , -V_1 \sin(\Gamma_{\text{MA}1}) \left( (1/F_1)\text{LI} - \\
& \quad (1/(3 \ F_1 ))\text{LI} \right) , -V_2 \sin(\Gamma_{\text{MA}2}) \left( (-1/(4 \ F_1))\text{LI} + (1/(6 \ F_1 ))\text{LI} \right) \\
\end{align*}
\]

\[
\begin{align*}
\text{DZVECT} &= (\sin(\Gamma_{\text{MA}1}) \left( (1/F_1)\text{LI} - (1/(3 \ F_1 ))\text{LI} \right) , \sin(\Gamma_{\text{MA}2}) \\
& \quad \left( (-1/(4 \ F_1))\text{LI} + (1/(6 \ F_1 ))\text{LI} \right) , V_1 \cos(\Gamma_{\text{MA}1}) \left( (1/F_1)\text{LI} - (1/( \\
& \quad (3 \ F_1 ))\text{LI} \right) , V_2 \cos(\Gamma_{\text{MA}2}) \left( (-1/(4 \ F_1))\text{LI} + (1/(6 \ F_1 ))\text{LI} \right) \\
\end{align*}
\]

Continue the block.

\[
\begin{align*}
\text{TRUEGRAD} &= (\text{DDMINDXP} \ (0,0,0,0,\text{DXVECT}) + \text{DDMINDZP} \ (0,0,0,0, \text{DZVECT})) \text{ GAMMAADVDMATRIX} \\
\text{TRUEGRAD} &= (\text{DTMINDXP} \ (0,0,0,0,\text{DXVECT}) + \text{DTMINDZP} \ (0,0,0,0, \text{DZVECT})) \text{ GAMMAADVDMATRIX} \\
\text{GRADIENT} &= \text{TRUEGRAD}, \text{TGRADIENT} = \text{TRUEGRAD} \\
\text{MIN} &= \text{TRUEMIN}, \text{DMIN} = \text{TRUEDMIN}
\end{align*}
\]
This block contains second derivative calculations.

* *
D=DELMATRIX
*
DELTAPRIMEMATRIX=

\( (0, 0, 0, 0, 0, 1, 0, 0, \text{DDMOCRHO0}, 0, 0, \text{DDMODALPHA0}, 0, 0, \text{DDMODV0}, 0, 0, \) \\
\( -(3/F2) \text{DDMOCRHO0}, D, D, D, D, -(3/F2) \text{DDMODV0}, D, D, (2/F2) \text{DDMOCRHO0}, D, D, 2 \) \\
\( 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, (2/F2) \text{DDMODV0}, D, D, 12, 13 \)

* *
A=AMATRIX
*
APRIMEMATRIX=(A , -(1/(2 P20)) +ASO  

\( 0 \) \\
\( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 \) \\
\( 17, 18, 19, 20, 21, 22, 23 \)

* 
PRIMEMATRIX=(0, 0, 0, -G \sin(\text{GAMMA ZERO}), 0, 0, 0, 0, 0, -1, 0, (1/2) (V0  
(1-8.862/CM0) 1.7473-1.7738), 0, 0, 0, 0, 0, (RHO0 V0  
PAREA/MASS0) (1-8.862/CM0) 1.7473-1.7738), 0, 0, 0, 0, P20/RHOO0,  
0, 0, 0, 0, 2 P20/V0, 0, 0, 0, 0, 0, 0, 0, 0, PCMATRIX, 0, 0, 0, 0, -1, -F1, 0,  
22 

PCMATRIX ,0, 0, 0, 0, 0, PCMATRIX ,0, 0, 0, PCMATRIX ,0, 0, 0, 0, 0, 0, 0, 26  
28  
34 

PCMATRIX ,0, 0, 0, 0, 0, 0, PCMATRIX ,0, 0, 0, PCMATRIX ,0, 0, -1, -F2, 0, 0,  
36 
47 

PCMATRIX ,0, 0, 0, 0, 0, PCMATRIX ,0, 0, 0, PCMATRIX ,0, 0, 0, 0, 0, 0, 0,  
51 
53 
59 

PCMATRIX ,0, 0,  
61 

* 
RHCMATRIX=(-1/SIGMA) (RHO0, 0, 0, 0, RHO1, 0, 0, 0, RHO2)
Do this part for $i=0,1$

IF $i=0$, $T=F_1$, OTHERWISE $T=F_2$

\[
T_{F_1} = T - 3 \frac{T}{(4F_1) + T} \quad (6F_1)
\]

\[
T_{F_2} = T \frac{F_1 - T}{(3F_1)} \quad (6F_1)
\]

\[
T_{F_3} = -T \frac{F_1 - T}{(4F_1) + T} \quad (6F_1)
\]

Continue the block.

$S_0 = \sin(G_0), S_1 = \sin(G_1), S_2 = \sin(G_2), C_0 = \cos(G_0), C_1 = \cos(G_1), C_2 = \cos(G_2)$

\[
X_{\text{MATRIX}} = \begin{pmatrix}
1 & -V_0 & S_0 & T_{F_1} & -V_1 & S_1 & T_{F_2} & -V_2 & S_2 & T_{F_3} & C_0 & T_{F_1} & C_1 & T_{F_2} & C_2 & T_{F_3}
\end{pmatrix}
\]

\[
Z_{\text{PRIME MATRIX}} = \begin{pmatrix}
1 & V_0 & C_0 & T_{F_1} & Z_{\text{MATRIX}} & Z_{\text{MATRIX}} & S_0 & T_{F_1} & 0 & 2 & 3 & 0
\end{pmatrix}
\]

\[
Z_{\text{MATRIX}} = Z_{\text{MATRIX}} \begin{pmatrix}
1 & V_0 & C_0 & T_{F_1} & Z_{\text{MATRIX}} & Z_{\text{MATRIX}} & S_0 & T_{F_1} & 0 & 1 & 1 & 6 & 7 & 1
\end{pmatrix}
\]

\[
Z_{\text{MATRIX}} = Z_{\text{MATRIX}} \begin{pmatrix}
4 & 5 & 2 & 2
\end{pmatrix}
\]

\[
V_{\text{GAMMA MATRIX}} = \begin{pmatrix}
1, (5/12)A_0 & F_1/V_0 & -\text{GAMMADOTAREA}/V_0 & \text{GAMMAMATRIX}
\end{pmatrix}
\]

\[
\text{GAMMAMATRIX} \quad \text{GAMMAMATRIX} \quad \text{GAMMAMATRIX} \quad 1, (1/3)A_0 & F_1/V_0 & 2
\]

\[
-\text{GAMMADOTAREA}/V_0 \quad \text{GAMMAMATRIX} \quad \text{GAMMAMATRIX} \quad \text{GAMMAMATRIX}
\]
GAMMAMATRIX )
5

* VDRP = (1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0)
VDRP = RHOMATRIX , VDRP = RHOMATRIX , VDRP = RHOMATRIX
0 0 10 4 20 8

* VDR = (1, VMATRIX , 1, VMATRIX ) DELTAPRIMEMATRIX VDRP
0 3

* APRP = (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)
APRP = RHOMATRIX , APRP = RHOMATRIX , APRP = RHOMATRIX
0 0 12 4 24 8

* APR = APRIMEMATRIX PPRIMEMATRIX APRP

* VDRIAPRP = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0)

* * *
VDRIAPR = VDR VDRIAPRP

* * *
XIVG = XMATRIX (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
VGMATRIX , 0,
VGMATRIX , 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0)

* * *
ZIVG = ZPRIMEMATRIX (1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, VGMATRIX , 0
0, VGMATRIX, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)

* * *
VDRIAPRZIVGVGI=VDRIAPR (1, 0, 0, 0, 0, 0, 0, ZIVG, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,emendation
* * *
DZPDAL=DZDSTATE (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0)
0)

* * *
DVSTATECOMP=(0,0,1,0,0,0,DVSTATE)

* * *
DVDPSTATE=DVSTATECOMP (0,1,0,0,0,0,-SG0/VO,CG0/VO,0,0,0,CG0,SG0,
0,0,0,0,0,0,0,0,0,0)

* * *
DVDESTATE=0

* * *
DVDCONTROL=DVSTATECOMP (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,
0,0)

* * *
DZDPSTATE=(0,1,0,0,0,DZPDSTATE)

* * *
DZDESTATE=0

* * *
TEMPZ=(0,1,0,0,0,0,0,DZDSTATE)

* * *
DZDCONTROL=TEMPZ (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,1,0,0,0,0,0,
0,0)

* * *
DXDPSTATE=(1,0,0,0,0,0,DXPDSTATE)

* * *
DXDCONTROL=(0,0,0,0,0,0,DXPDAL)

* * *
TEMPG=(1,0,0,0,0,0,0,0,VGMATRIX)

* * *
DGDPSTATE=TEMPG (0,0,-SG0/VO,CG0/VO,DVDPSTATE,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

* * *
DGDESTATE=0

* * *
DGDCONTROL=TEMPG (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,
0,0,0,1,0,0,0,0)
DPDPSTATE = PPRIMEMATRIX (RHOMATRIX DZDPSTATE, DGDPSTATE,
DPDPSTATE, 0, 0, 0, 0, 0, 0, 0, 0)

DPDDESTATE = 0

DPDCONTROL = PPRIMEMATRIX (RHOMATRIX DZDCONTROL, DGDCONTROL,
DPDCONTROL, 1, 0, 0, 0, 0, 1, 0, 0)

DADPSTATE = APRIMEMATRIX DPDPSTATE

DADESTATE = 0

DADCONTROL = APRIMEMATRIX DPDCONTROL

TRD1 = ALOPRIME - ALOPRIMEOLD, TRD2 = 1 - HFGAIN, TRD3 = ALO - ALOLDT0

D2TRDALO2 = 2 TRISE (1/ADENOM) \[\frac{3}{2}\] (TRD1 TRISE/(TRD2 TRD3)) \[\frac{2}{2}\]

+ 2 TRISE TRD1 (1/ADENOM) / (TRD2 TRD3)

D2TRDALODALOP = -2 TRISE (1/ADENOM) TRD1/(TRD2 TRD3)

- TRISE (1/ADENOM) / (TRD2 TRD3)

D2TRDALOP2 = 2 TRISE (1/ADENOM) \[\frac{3}{2}\] \[\frac{2}{2}\]

D2AREDALO2 = (1/2) TRD2 (2 DTRDALO + TRD3) D2TRDALO2

D2AREDALOP2 = (1/2) TRD2 TRD3 D2TRDALOP2

D2AREDALODALOP = (1/2) TRD2 (DTRDALOPRIME + TRD3) D2TRDALODALOP

- \( C \) \[\begin{array}{c}
\frac{2}{2} \\
\frac{0}{2}
\end{array}\]

GAMMADERIV = (5 F1/(12 V0) - DAREDALO/V0, 2 F1/(3 V1), -F1/(12 V0)

\[\begin{array}{c}
2 \\
V2
\end{array}\] , D2AREDALO2/V0, D2AREDALODALOP/V0)

- \( C \) \[\begin{array}{c}
\frac{2}{2} \\
\frac{2}{2}
\end{array}\]

GAMMADERIV = (-DAREDALOPRIME/V0, 2 F1/(3 V1), F1/(-6 V2), 2

1
D2AREA DALODALOP/VO, D2AREA DALODALOP2/VO)
- \(C \cdot \text{GAMMADERIV} = (0, -4(\text{ALO} F1 + \text{ALOPRIME} F1)^2) / (3 \text{ V1}^2), 0, 2 F1 / (3 \text{ V1}^2), 2^2 \text{ F1} / (3 \text{ V1}^2))
- \(C \cdot \text{GAMMADERIV} = (0, 0, (\text{ALO} F1 + 2 \text{ALOPRIME} F1)^2) / (6 \text{ V2}^2), -F1 / (12 \text{ V2}^2), 2^2 \text{ F1} / (-6 \text{ V2}^2))
- \(C \cdot \text{GAMMADERIV} = (F1 / (3 \text{ V0}^2) - \text{DAREA DALO/VO}, 2^2 \text{ F1} / (3 \text{ V1}^2), 2^2 \text{ F1} / (3 \text{ V2}^2), 4 \text{ F1} / (3 \text{ V1}^2), 2^2 \text{ F1} / (3 \text{ V2}^2), 2^2 \text{ F1} / (3 \text{ V1}^2))
- \(C \cdot \text{GAMMADERIV} = (0, -8(\text{ALO} F1 + \text{ALOPRIME} F1)^2) / (3 \text{ V1}^2), 0, 4 F1 / (3 \text{ V1}^2), 2^2 \text{ F1} / (3 \text{ V2}^2), 2^2 \text{ F1} / (3 \text{ V1}^2)
- \(C \cdot \text{GAMMADERIV} = (0, 0, -2(\text{ALO} F1 + 2 \text{ALOPRIME} F1)^2) / (3 \text{ V2}^2), F1 / (3 \text{ V2}^2), 2^2 \text{ F1} / (3 \text{ V2}^2))

\text{DMOALOTERM = DDMODALPHA A50 / (2 P20 PSQRTTERMO)}
- \(\text{DMODALODERIV} = (\text{DMOALOTERM / RH00, 2 DMOALOTERM / VO, 2 3.0709}
\text{DMOALOTERM / (-0.0149119 SGN(PAO) + 2 3.0709 PAO)}, \text{DMOALOTERM / (2 P20 PSQRTTERMO), -P10 DMOALOTERM / (4 P20 PSQRTTERMO),}
- (\text{DMOALOTERM / P20}) - ((1/2)
((-1/2)(\text{P10 / P20 }) + \text{P00/P20}) / PSQRTTERMO } \text{ DM0ALOTERM})
DELTADERIV = (0,0,0,0, (2/V1) DDMODPHO1,0,0, (1/2)V1^2 (PAREA/MSUB0)
0
(-.0149119 SGN(PA1)+2 3.0709 PA1),0)

- C
DELTADERIV = (0,0,0,0, (2/V2) DDMODRHO2,0,0, (1/2)V2^2 (PAREA/MSUB0)
1
(-.0149119 SGN(PA2)+2 3.0709 PA2))

- C
DELTADERIV = (DDMODALPHAO/RHO0,0,0,2 DDMODALPHAO/V0,0,0,3.0709
2
RHO0 V0 PAREA/MSUB0,0,0)

- C
DELTADERIV = (0,DDMODALPHA1/RHO1,0,0,2 DDMODALPHA1/V1,0,0,3.0709
3
RHO1 V1 PAREA/MSUB0,0)

- C
DELTADERIV = (0,0,DDMODALPHA2/RHO2,0,0,2 DDMODALPHA2/V2,0,0,
4
3.0709 RHO2 V2 PAREA/MSUB0)

- C
DELTADERIV = (0,DDMODV1/RHO1,0,0,DDMODV1/V1,0,0,RHO1 V1 (PAREA/MSUB0)
5
(-.0149119 SGN(PA1)+2 3.0709 PA1),0)

- C
DELTADERIV = (0,0,DDMODV2/RHO2,0,0,DDMODV2/V2,0,0,RHO2 V2 (PAREA
6
/MSUB0) (-.0149119 SGN(PA2)+2 3.0709 PA2))

- C
DELTADERIV = (-4/F2 )DELTADERIV,DELTADERIV = (2/F2 )DELTADERIV
7
0 8
- C
DELTADERIV = (2/F2 )DELTADERIV,DELTADERIV = (-4/F2 )DELTADERIV
9
2 10
- C
DELTADERIV = (2/F2 )DELTADERIV,DELTADERIV = (-4/F2 )
11
4 12
- C
DELTA_DERIV = (2/F2) DELTA_DERIV
5  6

- C
DELTA_DERIV = (4/F2) DELTA_DERIV
0  1

- C
DELTA_DERIV = (-3/F2) DELTA_DERIV
2  3

- C
DELTA_DERIV = (-1/F2) DELTA_DERIV
4  5

- C
DELTA_DERIV = (-1/F2) DELTA_DERIV
6  6

- C
Z_DERIV = (CG1, TF2, 0, 0, 0) 0

- C
Z_DERIV = (CG1, TF2, 0, 0, 0) 4 1

- C
Z_DERIV = (0, CG2, TF3, 0, 0) 1 0

- C
Z_DERIV = (0, CG2, TF3, 0, 0) 5 1

- C
Z_DERIV = (-V1, SG1, 0, CG1, 0) TF2 2 0

- C
Z_DERIV = (-V1, SG1, 0, CG1, 0) TF2 6 1

- C
Z_DERIV = (0, -V2, SG2, 0, CG2) TF3 3 0

- C
Z_DERIV = (0, -V2, SG2, 0, CG2) TF3 7 1

SGN0 = AS0, SGN1 = AS1, SGN2 = AS2

* ADERIV = 0
Do this part for I=0,1,2.

IF I=0, P0=P00, P1=P10, P2=P20, PST=PSQRTTERM0, SG=SGN0
IF I=1, P0=P01, P1=P11, P2=P21, PST=PSQRTTERM1, SG=SGN1
IF I=2, P0=P02, P1=P12, P2=P22, PST=PSQRTTERM2, SG=SGN2

\[ \text{AP0DERIV} = \left( -\frac{1}{4} \right) \text{SG} \frac{P2}{PST} \left( \frac{1}{8} \right) \text{SG} \frac{P1}{(PST \ P2)} \left( -\frac{1}{4} \right) \text{SG} \frac{P0}{(P2 \ PST)} \right)^3 \]

\[ \text{AP1DERIV} = \left( \frac{1}{8} \right) \text{SG} \frac{P1/P2}{PST} \left( -\frac{1}{16} \right) \text{SG} \frac{P1/P2}{PST} \left( +\frac{1}{4} \right) \text{SG} \]

\[ \left( \frac{1}{P2} \right) /PST, \left( \frac{1}{(2 \ P2)} \right) + \left( -\frac{1}{9} \right) \text{SG} \frac{P1/P2}{(P1/P2)} \left( -\frac{1}{2} \right) \frac{P1/P2}{PST} + \left( \frac{1}{4} \right) \text{SG} \left( -2 \frac{P1/P2}{PST} \right) \]

\[ \text{AP2DERIV} = \left( \frac{1}{4} \right) \text{SG} \left( -\frac{1}{2} \right) \frac{P1/P2}{PST} \left( +\frac{P0/P2}{PST} \right) \left( -\frac{1}{2} \right) \frac{P1/P2}{PST} \left( +\frac{1}{4} \right) \text{SG} \frac{P2}{PST}, \frac{1}{(2 \ P2)} \right) \left( -\frac{1}{8} \right) \text{SG} \frac{(P1/P2)}{(P1/P2)} \left( -\frac{1}{2} \right) \frac{P1/P2}{PST} \left( +\frac{(P0/P2)}{(P0/P2)} \right) /PST \]

\[ + \left( \frac{1}{2} \right) \text{SG} \left( -\frac{P1/P2}{PST} \right) \left( -\frac{P1/P2}{PST} \right) \left( -\frac{1}{4} \right) \text{SG} \left( -\frac{1}{2} \right) \frac{P1/P2}{PST} \left( +\frac{(P0/P2)}{(P0/P2)} \right) /PST \]

\[ + \left( \frac{2}{2} \right) \text{SG} \left( -\frac{3}{2} \right) \frac{P1/P2}{(P1/P2)} \left( -2 \frac{P0/P2}{PST} \right) \]

IF I=0, ADERIV = (AP0DERIV, 0, 0, 0, 0, 0, 0), ADERIV = (AP2DERIV, 0, 0, 0, 0, 0, 0)

IF I=1, ADERIV = (0, 0, 0, APODERIV, 0, 0, 0), ADERIV = (0, 0, 0, 0, 0, 0, 0)

IF I=2, ADERIV = (0, 0, 0, 0, 0, 0, APODERIV), ADERIV = (0, 0, 0, 0, 0, 0, 0)
AP1DERIV, ADERIV = (0, 0, 0, 0, 0, 0, AP2DERIV)

Continue the block

*       *
PCDERIV1=0, PCDERIV2=0

Do this part for I=0,1

IF I=0, RHO=RHO1, GAMMA=GAMMA1, V=V1, T=T1, PCM=PCM1, MASSI=MASS1, ALPHAI=ALPHA1
IF I=1, RHO=RHO2, GAMMA=GAMMA2, V=V2, T=T2, PCM=PCM2, MASSI=MASS2, ALPHAI=ALPHA2
POGDERIV=-G COS(GAMMA)
P1DERIV=(V PAREA/MASSI) ((1-8.862/PCM) 1.7473-1.7738)
P2DERIV=(V PAREA/MASSI) SGN(ALPHAI) (2.1908 (1-8.862/PCM)-1.5885)
P1VDERIVA=(V PAREA/MASSI) ((1-8.862/PCM) 1.7473-1.7738)
P1VDERIVB=(RHO PAREA/MASSI) ((1-9.862/PCM) 1.7473-1.7738)
P2VDERIVA=(V PAREA/MASSI) SGN(ALPHAI) (2.1908 (1-8.862/PCM)-1.5885)
P2VDERIVB=(RHO PAREA/MASSI) SGN(ALPHAI) (2.1908 (1-3.862/PCM)-1.5885)

IF I=0, PCDERIV1 = (0, 0, POGDERIV, 0, 0, 0),

PCDERIV1 = (0, 0, 0, 0, P1DERIV, 0),

PCDERIV1 = (P1VDERIVA, 0, 0, 0, P1VDERIVB, 0),

PCDERIV1 = (0, 0, 0, 0, P1DERIV, 0),
- \text{C}\n\text{PCDERIV2} = (0,0,0,0,\text{P2RDERIV},0),
\frac{2}{2}
- \text{C}\n\text{PCDERIV2} = (\text{P2VDERIVA},0,0,0,\text{P2VDERIVB},0)
\frac{4}{4}
- \text{C}\n\text{PCDERIV2} = (0,0,0,\text{POGDERIV},0,0),
\frac{15}{15}
- \text{C}\n\text{PCDERIV2} = (0,0,0,0,\text{P1RDERIV}),
\frac{19}{19}
- \text{C}\n\text{PCDERIV2} = (0,\text{P1VDERIVA},0,0,0,\text{P1VDERIVB}),
\frac{21}{21}
- \text{C}\n\text{PCDERIV2} = (0,0,0,0,\text{P2RDERIV}),
\frac{27}{27}
- \text{C}\n\text{PCDERIV2} = (0,\text{P2VDERIVA},0,0,0,\text{P2VDERIVB})
\frac{29}{29}

Do this part for I=0,1,2.

\text{IF I=0, T=TAU1,XE=XE1,XP=XP1,ZE=ZE1,ZP=ZP1,DI=D1}
\text{IF I=1, T=TAU2,XE=XE2,XP=XP2,ZE=ZE2,ZP=ZP2,DI=D2}
\text{IF I=2, T=TAU3,XE=XE3,XP=XP3,ZE=ZE3,ZP=ZP3,DI=D3}
\text{TT1}=T-(3/(4 \text{ F1}))T+(1/(6 \text{ F1}))T,\text{TT2}=(1/\text{F1})T-(1/(3 \text{ F1} ))T
\frac{2}{2} \frac{2}{2} \frac{3}{3} \frac{2}{2} \frac{2}{2} \frac{3}{3}
\text{TT3}=(-1/(4 \text{ F1} ))T+(1/(6 \text{ F1} ))T
\frac{2}{2} \frac{2}{2} \frac{3}{3}
- \text{C}\n\text{DXDERIV} = (-\text{SG1 TT2},0,0,0)
\frac{4}{4} \text{I}
- \text{C}\n\text{DXDERIV} = (\text{CG1 TT2},0,0,0)
\frac{4}{4} \text{I}
- \text{C}\n\text{DXDERIV} = (0,-\text{SG2 TT3},0,0)
\frac{4}{4} \text{I+1}
- C
  DZDEIV = (0, CG2 TT3, 0, 0)
  4 I+1
- C
  DXDERIV = (-V1 CG1, 0, -SG1, 0) TT2
  4 I+2
- C
  DZDEIV = (-V1 SG1, 0, CG1, 0) TT2
  4 I+2
- C
  DZDEIV = (0, -V2 SG2, 0, CG2) TT3
  4 I+3
- C
  DXDERIV = (0, -V2 CG2, 0, -SG2) TT3
  4 I+3
- C
  XPDDEIV = (-V0 SG0 TT1, -V1 SG1 TT2, -V2 SG2 TT3, CG0 TT1, CG1 TT2
  , CG2 TT3)
- C
  ZPDDEIV = (V0 CG0 TT1, V1 CG1 TT2, V2 CG2 TT3, SG0 TT1, SG1 TT2,
  I
  SG2 TT3)

Do this part for K=0,1, 2

* * * * * * * *

IF K=0, DVDU=DVDPSTATE, DZDU=DZDPSTATE, DGDU=DGDSTATE, DPDU=
* * * * * * * *
DPDPSTATE, DADU=DADPSTATE,DALDU=0, DXEDU=0, DZEDU=0, DXEDDU=0,
-
DZEDDU=0,
-
DXPDU=(1, 0, 0, 0), DZPDU=(0, 1, 0, 0), DZFGSTATEDU=(0, 0, 0, 1, 0, 0,
C, 0, 0, 0, 0, 0, 0, 0, 0)
-
* * * * * * * *

IF K=1, DVDU=0, DZDU=0, DGDU=0, DPDU=0, DADU=0, DALDU=0, DZEDU=(
-
DELXEO, DELXEO, DELXEO), DZEDUT=(DELXEO, DELXEO, DELXEO),
  0 8 16 4 12 20
DXEDU=(1, 0, 0, 0), DZEDU=(0, 1, 0, 0),

DXPODU=0, DZPODU=0, DXEDU=DXEDUT, DZEDU=DZEDUT, DZEROSTATEDU=
(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

IF K=2, DVDU=DVDCONTROL, DZDU=DZDCONTROL, DGDU=DGDCONTROL, DADU=

DADDCONTROL, DPDU=DPDCONTROL, DALDU=(1, 0, 0, 0, 0, 1, 0, 0),

DXEDU=0, DZEDU=0, DXEDU=0, DZEDU=0, DXPODU=0, DZPODU=0,

DZEROSTATEDU=0

DRHODU=PHOMATRIX DZDU

DVPDUALDUT = (DVDU, DALDU)

DGAMMAMATRIXDU=DVPDUALDUT GAMMADERIV

DRVAPDUT = (DRHODU, DVDU, DADU, DPDU, DPDU, DPDU)

0 0 0 0 4 8

DDMDALOQU=DRVAPDUT DMDALADERIV

DRVATUT = (DRHODU, DVDU, DADU)

DDDELTA MadnessDU=DRVATUT DELTADERIV

DGVDUT = (DGDU, DGDU, DVDU, DVDU)

4 8 4 8

DZN MATRIXDU=DGVGUT ZDERIV

DAMATRIXDU=DPDU ADERIV

DRGVGUT = (DRHODU, DRHODU, DGDU, DGDU, DVDU, DVDU)

4 8 4 8 4 8
** DPCMATRIXDU1=DRGVDUT  PCDERIV1
** DPCMATRIXDU2=DRGVDUT  PCDERIV2
** DPC1=DPCMATRIXDU1, DPC2=DPCMATRIXDU2
*
** DDXDU=DRGVDUT  DXDERIV
** DDZDU=DRGVDUT  DZDERIV
*
GVUDERIVT=(DGDU,DVDU), DXPOT=(DXPDU,DXPODU,DXPODU)
*
- DZPOT=(DZPODU,DZPODU,DZPODU)
*
** DXPDU=GVUDERIVT  XPDERIV+DXPOT
** DZPDU=GVUDERIVT  ZPDERIV+DZPOT
*

Do this subpart for I=0, 1, 2

** DXZEPDUT = (DXEDU, DXPDU, DZEDU, DZPDU)
** I I I I
- C ** C - C - C

** D2DDXPDU = DXZEPDUT  DDERIVX
** I I
- C ** T - C

** D2DDZPDU = DXZEPDUT  DDERIVZ
** I I
- C ** T - C

Continue the part.

** D2DMINudu dxp = D2DMINuxedxp (DXEDU, DXEDU) + D2DMINuxexp
** (DXPDU, DXPDU) + D2DMINuzdexp (DZEDU, DZEDU)
*
- D2DMINuzpxp (DZPODU, DZPODU)
\[
\begin{align*}
D2DMINDUDZP &= D2DMINDXEDZP (DXEDU, DXEDU) + D2DMINDXPDXP \\
& - \quad * \quad T \quad * \quad T
\end{align*}
\]

\[
\begin{align*}
(DXEDU, DXEDU) + D2DMINDZEDZP (DZEDU, DZEDU) +
\end{align*}
\]

\[
\begin{align*}
* \quad T \quad * \\
D2DMINDZP2 (DZEDU, DZEDU)
\end{align*}
\]

\[
\begin{align*}
D2TMINDUDX = D2TMINDXEDX (DXEDU, DXEDU) + D2TMINDXP2 \\
& - \quad * \quad T \quad * \quad T
\end{align*}
\]

\[
\begin{align*}
(DXEDU, DXEDU) + D2TMINDZEDX (DZEDU, DZEDU) +
\end{align*}
\]

\[
\begin{align*}
* \quad T \quad *
D2TMINDZPDX (DZEDU, DZEDU)
\end{align*}
\]

\[
\begin{align*}
D2TMINDUDZP &= D2TMINDXEDZP (DXEDU, DXEDU) + D2TMINDXPDXP \\
& - \quad * \quad T \quad * \quad T
\end{align*}
\]

\[
\begin{align*}
(DXEDU, DXEDU) + D2TMINDZEDZP (DZEDU, DZEDU) +
\end{align*}
\]

\[
\begin{align*}
* \quad T \quad *
D2TMINDZP2 (DZEDU, DZEDU)
\end{align*}
\]

\[
\begin{align*}
D2DMINDUDXPT = D2DMINDUDX, D2DMINDUDZPT = D2DMINDUDZP \\
* \quad T \quad * \quad T
\end{align*}
\]

\[
\begin{align*}
D2TMINDUDXPT = D2TMINDUDX, D2TMINDUDZPT = D2TMINDUDZP
\end{align*}
\]

Do this subpart for I=0,1,2.
Do this for J=0,...,7

\[
\text{DMATRIXGAMMADU} = \text{DGAMMAMATRIXDU} \\
\text{J} \quad 8 \quad \text{I+J}
\]

Do this for J=0,1,...,13

\[
\text{DMATRIXDELTADU} = \text{DDELTAMATRIXDU} \\
\text{J} \quad 14 \quad \text{I+J}
\]

Do this for J=0,1,...,7

\[
\text{DMATRIXZDU} = \text{DZMATRIXDU} \\
\text{J} \quad 8 \quad \text{I+J}
\]
Do this for J=0,1,...,23

\[
\text{DMATRIXADU} = \text{DMATRIXDU} \\
J \quad 24 \ I+J
\]

Do this for J=0,1,...,63

\[
\text{IF } J<32, \text{DMATRIXPCDU} = \text{DPC1} \\
J \quad 4 \ J+1
\]

\[
\text{IF } J>31, \text{DMATRIXPCDU} = \text{DPC2} \\
J \quad 4(J-32)+1
\]

Do this for J=0,1,...,11

\[
\text{DDXVECTORDU} = \text{DDXDU} \\
J \quad 12 \ I+J
\]

\[
\text{DDZVECTORDU} = \text{DDZDU} \\
J \quad 12 \ I+J
\]

Continue the subpart.

\[
\text{RTEMP= (EXP (-Z1/SIGMA), 0, 0, EXP (-Z2/SIGMA)) (-RHOSUBZERO/SIGMA)}
\]

\[
\text{DRTMPDU= (EXP (-Z1/SIGMA) DZDU , 0, 0, EXP (-Z2/SIGMA) DZDU )} \\
4+I \quad 8+I
\]

(RHOSUBZERO/SIGMA)

\[
\text{DMATRIXPDU= DRTMPDU ZMATRIX (0, 0, 1, 0, 0, 0, 0, 1, GAMMAMATRIX)}
\]

\[
\text{DMATRIXPDU= DMATRIXPDU + RTEMP DMATRIXZDU (0, 0, 1, 0, 0, 0, 0, 1,} \\
\text{ GAMMAMATRIX)}
\]

\[
\text{DMATRIXPDU= DMATRIXPDU + RTEMP ZMATRIX (0, 0, 0, 0, 0, 0, 0,} \\
\text{0, DMATRIXGAMMADU)}
\]
DAPCPGAMMADU=DMATRIXADU PCMATRIX(1,0,0,0,1,0,0,PMATRIX,0,0,
1,0,0,0,0,1,GAMMAMATRIX) +AMATRIX DMATRIXPDU(1,0,0,0,1,0,0,
PMATRIX,0,0,1,0,0,0,0,1,GAMMAMATRIX) +AMATRIX PCMATRIX(0,0,0,
0,0,0,0,DMATRIXPDU,0,0,0,0,0,0,0,0,DMATRIXGAMMADU)

DELTAPAPCPGAMMADU=DMATRIXDELTADU(PMATRIX,APCPGAMMAMATRIX,0,
0,1,0,0,0,0,1) +DELTAMATRIX (DMATRIXPDU,DAPCPGAMMADU,0,0,0,0,
0,0,0,0)

DVALODU=VMATRIX (DDMODALODU,0,0,0,DDELTAPAPCPGAMMADU)

DVAL01DU=(DVALODU,DVALODU,DVALODU,DVALODU)

DVAL02DU=(DVALODU,DVALODU,DVALODU,DVALODU)

DDVDALDU=-(VAL02MATRIX-(1,0,0,1)) (DVAL02DU DVDALMATRIX+

DVAL01DU)

DG1=(0,0,0,0,0,0,0,0,DMATRIXGAMMADU)

DG2=(1,0,0,1,DVDALMATRIX)

DG3=(0,0,1,0,0,0,0,1,GAMMAMATRIX)

DG4=(0,0,0,0,DDVDALDU)

DGAMMADVDALDU=DG1 DG2+DG3 DG4

TRUEGRADIENTDUI=(D2DMINUDUPTXPT (0,0,0,0,DXVECTOR)

2 I 4 I
End of the subpart. Continue the part.

\[ \text{TRUEGRADIENTGRADIENT}^T = \text{DTRUEGRADIENTDUI} \]
\[ \text{TRUEETGRADIENTGRADIENT}^T = \text{DTRUEETGRADIENTDUI} \]

Continue the block.

\[ \text{GRADIENTGRADIENT} = \text{TRUEGRADIENTGRADIENT} \]
\[ \text{TRUEGRADIENTGRADIENT} = \text{TRUEGRADIENTGRADIENT} \]
\[ \text{TGRADIENTGRADIENT} = \text{TRUEGRADIENTGRADIENT} \]
TRUETGRADIENT = 16

TRUEDDDXEO = DDMINDXE (1, 0, 0, 0, DELXEO, DELXEO, DELXEO) +

DDMINDZE (0, 1, 0, 0, DELXEO, DELXEO, DELXEO)

TRUEDTDXEO = DTMINDXE (1, 0, 0, 0, DELXEO, DELXEO, DELXEO) +

DTMINDZE (0, 1, 0, 0, DELXEO, DELXEO, DELXEO)

GVUERIVT = (DGDPSTATE, DVDPSTATE)

DXPDU = GVUERIVT XPDERIV*(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)

DZPDU = GVUERIVT ZPDERIV*(0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0)

TRUEDDDXPO = DDMINDXP (1, 0, 0, 0, DXPDU, DXPDU, DXPDU) +

DDMINDZP (0, 1, 0, 0, DZPDU, DZPDU, DZPDU)

TRUEDTDXPO = DTMINDXP (1, 0, 0, 0, DXPDU, DXPDU, DXPDU) +

DTMINDZP (0, 1, 0, 0, DZPDU, DZPDU, DZPDU)

DDDXE0 = TRUEDDDXEO, DTDXE0 = TRUEDTDXEO, DDDXPO = TRUEDDDXPO

END OF GDTSUB
B.4 Subroutine LRSRCH

This subroutine accepts E and P states from the LR evader optimization program. It supplies the predicted E motion and P response as well as the desired derivatives.

\[
\text{DIMENSION } (\text{PGRADIENT}^2, 2), (\text{N\text{GRADIENT}}^2, 2), \\
(\text{PGRADIENTGRADIENT}, 2 \times 4), (\text{N\text{GRADIENTGRADIENT}}, 2 \times 4), \\
(\text{PDDD\text{DXE}0}, 4), (\text{PDDD\text{DXP}0}, 4), (\text{NDDD\text{XEO}}, 4), (\text{NDDD\text{XP}0}, 4), (\text{P\text{MULT}}, 2), \\
(\text{NG\text{MULT}}, 2), (\text{DAL\text{MULTIPLIER}}, 2 \times 2), (\text{DX\text{P}\text{MULTIPLIER}}, 2 \times 4), \\
(\text{DXE0\text{MULTIPLIER}}, 2 \times 4), (\text{D\text{ALDXP}0}, 2 \times 4), (\text{D\text{ALDXE}0}, 2 \times 4), \\
(\text{DP\text{GRADIENTDAL}}, 2 \times 2), (\text{DNG\text{RADIENTDAL}}, 2 \times 2), (\text{PCOSTSLOPE}, 2), \\
(\text{NCOSTSLOPE}, 2), (\text{PCOSTSLOPEGRADIENT}, 2 \times 4), (\text{NCOSTSLOPEGRADIENT}, 2 \times 4), \\
(\text{DAL\text{CDAL}}, 2), (\text{D2PCOSTDALDALC}, 2), (\text{D2ALOCDA\text{DALDALO}}, 2), \\
(\text{D2NCOSTDALDALC}, 2), \\
(\text{D2ALOPRIMEDALDALALO}), 2), (\text{D2ALOPRIMEDALDALALO\text{PRIME}}, 2), \\
(\text{D2PCOSTdal2}, 2), (\text{D2NCOSTDAL2}, 2)
\]

First, the parameter GLIMIT (=maximum $A_\alpha$ magnitude) is supplied along with current E and P states and the present time $T_0$. The subroutines ETRAJSUB and GDMINSUB are called to get AL0NOW and ALOPRIMENOW, PGRADIENT, NGRADIENT; PGRADIENTGRADIENT, NGRADIENTGRADIENT, PDDD\text{DXE}0, NDDD\text{XEO}, PDDD\text{DXP}0, and NDDD\text{XP}0. (These quantities are identical to corresponding ones in 8.1 except that the P and N prefixes are indicative of positive-$\alpha$ and negative-$\alpha$ predicted E trajectories.)
This block computes the relationship between the "downstream" of the saturating nonlinearity and the "upstream" quantity.

\[ \text{ALONWP} = \text{ALONOW} \]
\[ \text{SATFCN1} = \text{GLIMIT} \tanh \left( \frac{\text{ALONOWP}}{\text{GLIMIT}} \right) \]
\[ \text{DSATFCN1DAL0} = \frac{1}{\cosh \left( \frac{\text{ALONOWP}}{\text{GLIMIT}} \right)} \]
\[ \text{D2SATFCN1DAL0} = - \frac{2}{\text{GLIMIT}} \sinh \left( \frac{\text{ALONOWP}}{\text{GLIMIT}} \right) \cosh \left( \frac{\text{ALONOWP}}{\text{GLIMIT}} \right) \]
\[ \text{ALT6} = \text{ALONOW} + \text{AL0PRIMENOW} \times (6 - T) \]
\[ \text{ALT6P} = \text{ALT6} \]
\[ \text{SATFCN2} = \text{GLIMIT} \tanh \left( \frac{\text{ALT6P}}{\text{GLIMIT}} \right) \]
\[ \text{DSATFCN2DAL0} = \frac{1}{\cosh \left( \frac{\text{ALT6P}}{\text{GLIMIT}} \right)} \]
\[ \text{DSATFCN2DAL0} = (\text{DSATFCN2DAL0PRI}ME= (6 - T0)) \]
\[ \text{D2SATFCN2DAL0} = - \frac{2}{\text{GLIMIT}} \sinh \left( \frac{\text{ALT6P}}{\text{GLIMIT}} \right) \cosh \left( \frac{\text{ALT6P}}{\text{GLIMIT}} \right) \]
\[ \text{D2SATFCN2DAL0DAL0PRI}ME= (6 - T0) \]
\[ \text{D2SATFCN2DAL0PRI}ME= (6 - T0) \]
\[ \text{D2SATFCN2DAL0} \]
\[ \times (\text{D2SATFCN2DAL0PRI}ME= (6 - T0)) \]
\[ \text{D2AL0CDALDAL0} = (\text{DSATFCN1DAL0, 0, (1/(6 - T0)) (DSATFCN2DAL0-DSATFCN1DAL0), DSATFCN2DAL0PRI}ME/(6 - T0)) \]
\[ \text{D2AL0CDALDAL0} = (\text{D2SATFCN1DAL0DAL0PRI}ME, 0) \]
\[ \text{D2AL0PRI}MECDALDAL0 = (1/(6 - T0)) \]
\[ \text{D2AL0PRI}MECDALDAL0 = (\text{D2SATFCN2DAL0DAL0PRI}ME) \]
\[ \text{D2AL0PRI}MECDALDAL0PRI}ME= (1/(6 - T0)) \]
This block computes the upstream gradient for P and N predicted miss distances.

\begin{align*}
- & - * \\
PCOSTSLOPE & = PGRADE/I\ DCDAL \\
- & - * \\
NCOSTSLOPE & = NGRADE/I\ DCDAL \\
\end{align*}

This block computes the upstream gradients of \( \frac{\partial \text{Cost}}{\partial A_L} \) with respect to P (position-velocity) state, E state and upstream \( A_L \).

\begin{align*}
* & \\
PCOSTSLOPEGRADIENT & = DCDAL \quad PGRADE/I\GRADIENT \quad DCDAL \\
0 & 0 0 2 \\
- & \\
PGRADE/I\GRADIENT & \\
4 & \\
* & \\
PCOSTSLOPEGRADIENT & = DCDAL \quad PGRADE/I\GRADIENT \quad DCDAL \\
8 & 0 8 2 \\
- & \\
PGRADE/I\GRADIENT & \\
12 & \\
* & \\
NCOSTSLOPEGRADIENT & = DCDAL \quad NGRADE/I\GRADIENT \quad DCDAL \\
0 & 0 0 2 \\
- & \\
NGRADE/I\GRADIENT & \\
4 & \\
* & \\
NCOSTSLOPEGRADIENT & = DCDAL \quad NGRADE/I\GRADIENT \quad DCDAL \\
8 & 0 8 2 \\
- & \\
NGRADE/I\GRADIENT & \\
12 & \\
\end{align*}
D2PCOSTDALDALC = \( \begin{pmatrix} \text{PGRADIENT} & \text{PGRADIENT} \\ 16 & 17 \end{pmatrix} \) 

D2NCOSTDALDALC = \( \begin{pmatrix} \text{NGRADIENT} & \text{NGRADIENT} \\ 16 & 17 \end{pmatrix} \) 

D2PCOSTDALDALC = D2PCOSTDALDALC \( \mathbf{T} \) DALCDAL 

D2NCOSTDALDALC = D2NCOSTDALDALC \( \mathbf{T} \) DALCDAL 

D2PCOSTDAL2 = DALCDAL \ D2PCOSTDALDALC + \text{PGRADIENT} \ (D2AL0CDALDAL0, 0, 0, 0) 

D2NCOSTDAL2 = DALCDAL \ D2NCOSTDALDALC + \text{NGRADIENT} \ (D2AL0CDALDAL0, 0, 0, 0) 

P\text{COSTSLOPE} = (D2PCOSTDAL2, 0, 0, D2PCOSTDAL2, 0, 0) 

N\text{COSTSLOPE} = (D2NCOSTDAL2, 0, 0, D2NCOSTDAL2, 0, 0) 

This block computes further ingredients for the final, desired derivatives when \( P \) and \( N \) misses are the same value.

PG\text{MULT} = \text{PCOSTSLOPE} (0, -1, 1, 0) 

NG\text{MULT} = \text{NCOSTSLOPE} (0, -1, 1, 0)
This block computes final, desired derivatives when the P miss or the N miss is larger (ONEMIN=1).

If ONEMIN=1 and PDMIN>NDMIN, DALMULTIPLIER=-DPGRADIENTDAL,

DXPMULTIPLIER=PCOSTSLOPEGRADIENT, DXEOMULTIPLIER=PCOSTSLOPEGRADIENT
IF ONEMIN=1 AND NDMIN>PDMIN, DALMULTIPLIER=-DGGRADIENTDAL, * 
* DXPMULTIPLIER=NCOSTSLOPEGRADIENT, DXEMULTIPLIER= 
* NCOSTSLOPEGRADIENT 8

This block computes the derivatives of upstream $\mathbf{A}_L$ with respect to P (position-velocity) and E states.

* * -1 *
DALDXP0=DALMULTIPLIER DXPMULTIPLIER

* * -1 *
DALDXEO=DALMULTIPLIER DXEMULTIPLIER

END OF LRSRCH
B.5 Subroutine ETRAJSUB

This subroutine accepts the current E state from LRSRCH and returns the estimated positive-$\chi$ and negative-$\chi$ trajectories along with their derivatives.

(dimensions)

DIMENSION (TSTATE, 3X4), (K, 2X3), (KF, 3X2), (EGAMMA, 3X4), (YG, 3X4),
(DELXEO, 2X4), (AXGWA, 2X3), (AXVECT, 2), (AX, 3), (AXGWATS, 2X4),
(DZDZZERO, 3)

The variables given to ETRAJSUB are identical to those given to ETRAJNC. (See Appendix B.2) The initialization prior to the SAA iteration follows.

\[ \text{GAMMAO=GAMMAZERO} \]
\[ \text{G=32.17, EMASS=18.68, EAREA=4} \]
\[ \text{CDO=.120} \]
\[ \text{SIGMA=24000} \]
\[ \text{BETAZERO=EMASS G/(CDO EAREA)} \]
\[ \text{CL=.740 SGN(EANGLEOPATTACK)} \]
\[ \text{CD=.785} \]
\[ \text{TAU3=6-TO, TAU1=TAU3/3, TAU2=2 TAU1} \]
INVMATRIX=INVERSE(TAU1,TAU1,TAU1,TAU2,TAU2,TAU2,
TAU3,TAU3,TAU3)

INITIALRHO=.002377 EXP(-INITIALALTITUDE/SIGMA)
WZERO=SIGMA (G/BETAZERO) (INITIALRHO/2)
K1=WZERO EXP((CD/CL)GAMMAZERO) /SIGMA
K2=WZERO (CL/CD0) -COS(GAMMAZERO)
EPSILONL=K1(K2+COS(GAMMAZERO)) EXP((-CD/CL)GAMMAZERO)
EPSILON=0,EPSILONA=0,EPSILONJ=0,GAMMA3NEW=.0001

Following is the SAA approximate integration of the \( \phi \) equation for \( E \).
This block is repeated until

\[
\left| \frac{\text{GAMMA}_3\text{NEW}-\text{GAMMA}_3\text{OLD}}{\text{GAMMA}_3\text{OLD}} \right| < .0001
\]

GAMMA3OLD=GAMMA3NEW

GAMMA1=EPSILONL TAU1+(1/2) EPSILONS TAU1+(1/3) EPSILONA
TAU1+(1/4) EPSILONJ TAU1
GAMMA1=GAMMA1+GAMMAZERO

GAMMA2=GAMMAZERO+EPSILONL TAU2+(1/2) EPSILONS TAU2+
(1/3) EPSILONA TAU2+(1/4) EPSILONJ TAU2

GAMMA3=GAMMAZERO+EPSILONL TAU3+(1/2) EPSILONS TAU3+
(1/3) EPSILONA TAU3+(1/4) EPSILONJ TAU3
GAMMA3NEW=GAMMA3

FCN1=K1(K2+COS(GAMMA1)) EXP((-CD/CL)GAMMA1)
FCN2=K1(K2+COS(GAMMA2)) EXP((-CD/CL)GAMMA2)
FCN3=K1(K2+COS(GAMMA3)) EXP((-CD/CL)GAMMA3)

- * 
EPSILON=INVMATRIX (FCN1-EPSILONL,FCN2-EPSILONL,FCN3-EPSILONL)
\[ \text{EPSILON} = 0 \]
\[ \text{EPSILON} = 1 \]
\[ \text{EPSILON} = 2 \]

Initialize the next block.
TREL = 0
Repeat this block for TREL = 0, TAU1, TAU2 and TAU3 to get X0, Z0, X1, Z1, X3 and Z3.

\[
\begin{align*}
\text{GAMMA} &= \text{GAMMA}_0 + \text{EPSILON}_L \cdot \text{TREL} + \left( \frac{1}{2} \right) \text{EPSILON}_S \cdot \text{TREL} + \\
&\quad \left( \frac{1}{3} \right) \text{EPSILON}_O \cdot \text{TREL} + \left( \frac{1}{4} \right) \text{EPSILON}_J \cdot \text{TREL} \\
\text{VELOCITY} &= \text{VZERO} \cdot \exp \left( - \left( \frac{CD}{CL} \right) (\text{GAMMA} - \text{GAMMA}_0) \right) \\
W &= \text{WZERO} + \left( \frac{CD}{CL} \right) (\cos(\text{GAMMA}) - \cos(\text{GAMMA}_0)) \\
\text{RHO} &= 2 \cdot \text{BETA}_0 \cdot W / (G \cdot \text{SIGMA}) \\
Z &= \text{INITIAL ALTITUDE} - \text{SIGMA} \cdot \log(\text{RHO} / \text{INITIAL RHO}) \\
A &= \left( \frac{CL}{CD} \right) \cdot \text{WZERO} - \cos(\text{GAMMA}_0) \\
\text{IF ABS}(A) > 1, \text{GO TO XFIRST} \\
\text{LOG ARGUMENT}_0 &= \frac{\left( \sqrt{1 - A} \right) \tan(\text{GAMMA}_0 / 2) + A + 1}{\left( \sqrt{1 - A} \right) \tan(\text{GAMMA}_0 / 2) - A - 1} \\
\text{LOG ARGUMENT} &= \frac{\left( \sqrt{1 - A} \right) \tan(\text{GAMMA} / 2) + A + 1}{\left( \sqrt{1 - A} \right) \tan(\text{GAMMA} / 2) - A - 1} \\
\text{GO TO XSECOND location XFIRST} \\
\text{ARCTAN ARGUMENT}_0 &= \sqrt{A - 1} \cdot \tan(\text{GAMMA}_0 / 2) / (A + 1) \\
\text{ARCTAN ARGUMENT} &= \sqrt{A - 1} \cdot \tan(\text{GAMMA} / 2) / (A + 1)
\end{align*}
\]
GO TO XTHIRD
  location XSECOND
  X = INITIAL RANGE - SIGMA (GAMMA ZERO - GAMMA + A LOG (LOG ARGUMENT / 2 LOG ARGUMENT ZERO) / SQRT (1 - A ))
  Go To XFOURTH
  location XTHIRD
  X = INITIAL RANGE - SIGMA (GAMMA ZERO - GAMMA + 2 A (ARCTAN (ARCTAN ARGUMENT 2) - ARCTAN (ARCTAN ARGUMENT ZERO)) / SQRT (A - 1 ))
  location XFOURTH
  IF TREL = 0, XE0 = X, ZE0 = Z, TREL = TAU 1
  IF TREL = TAU 1, XE1 = X, ZE1 = Z, TREL = TAU 2
  IF TREL = TAU 2, XE2 = X, ZE2 = Z, TREL = TAU 3
  IF TREL = TAU 3, XE3 = X, ZE3 = Z

In this block the derivatives of XE1, XE2, ZE1, ZE2, etc. are computed with respect to the E state.

* TSTATE = (0, - W ZERO / SIGMA, 0, 0, 0, 0, - SIN (GAMMA ZERO) / W ZERO, COS (GAMMA ZERO) / W ZERO, 0, 0, COS (GAMMA ZERO), SIN (GAMMA ZERO))

* K = (0, CD K1 / CL, K1 / W ZERO, CL / CD0, SIN (GAMMA ZERO), 0)

  EL = (EPSILONL / K1, K1 EXP (- CD GAMMA ZERO / CL))

  EL = EL + (0, K1 SIN (GAMMA ZERO) EXP (- CD GAMMA ZERO / CL) - CD)

EPSILONL / CL, 0)

* KF = (FCN1 / K1, K1 EXP (- CD GAMMA 1 / CL), FCN2 / K1, K1 EXP (- CD GAMMA 2 / CL), FCN3 / K1, K1 EXP (- CD GAMMA 3 / CL))

* GAMMA F = (- K1 SIN (GAMMA 1) EXP (- CD GAMMA 1 / CL) - CD FCN1 / CL, 0, 0, 0,
-k1 sin(gamma2) exp(-cd gamma2/cl) - cd fcn2/cl, 0, 0, 0, -k1
sin(gamma3) exp(-cd gamma3/cl) - cd fcn3/cl

* 
2 3 4 2
egamma = (tau1, (1/2) tau1, (1/3) tau1, (1/4) tau1, tau2, (1/2) tau2,
3 4 2
(1/3) tau2, (1/4) tau2, tau3, (1/2) tau3, (1/3) tau3, (1/4) tau3 )

-gwa = (cl/cd0, sin(gammazero), 0)

* 

gx = 0

* 

gx0 = 0

* 

gz = 0

* 

gz0 = 0

Do this part for i = 0, 1, 2

if i = 0, gamma = gamma1
if i = 1, gamma = gamma2
if i = 2, gamma = gamma3
if abs(a) > 1, go to secondx

dlogargument dgamma = -(a+1) sqrt(1-a) (sec(gamma/2)) ^ 2

2
1-a tan(gamma/2) - a-1

2
logargument = (sqrt(1-a) tan(gamma/2) + a+1) / (sqrt(1-a) tan(

gamma/2) - a-1)

gx = -sigma(-1+a dlogargument dgamma / (logargument sqrt(1-a))

4 i

gx0 = -sigma(1+(a(a+1)) (sec(gammazero/2)) / (sqrt(1-a) tan(}

3 i+1

2
2
) / (sqrt(1-a) tan(}
GAMMAZERO/2 + A + 1) \cdot (\text{SQRAT}(1 - A) \cdot \text{TAN}(\text{GAMMAZERO/2} - A - 1))

\text{ASN} = \text{SQRAT}(1 - A)

\text{DLOGARGUMENTDA} = 2(A + 1) \cdot \text{TAN}(\text{GAMMA/2}) / (\text{ASN} \cdot \text{TAN(\text{GAMMA/2} - A - 1))}

\text{DLOGARGUMENTDA} = 2(A + 1) \cdot \text{TAN}(\text{GAMMA0/2}) / (\text{ASN} \cdot \text{TAN(\text{GAMMA0/2} - A - 1))}

\text{LALAO} = \text{LOGARGUMENT/LOGARGUMENTZERO}

\text{LOGLALAO} = \text{LOG(LALAO)}

\text{THREEHALFTERM} = (\text{SQRAT}(1 - A))

\text{TEMPV1} = -\text{SIGMA} \cdot \text{LOGLALAO/THREEHALFTERM}

\text{TEMPV2} = -\text{SIGMA} ((A/\text{ASN}) \cdot (\text{DLOGARGUMENTDA/LOGARGUMENT}))

\text{TEMPV3} = +\text{SIGMA} ((A/\text{ASN}) \cdot (\text{DLOGARGUMENTDA/LOGARGUMENTZERO}))

\text{AX} = \text{TEMPV1} + \text{TEMPV2} + \text{TEMPV3}

\text{GO TO ENDX}

\text{location SECONDX}

\text{ASP} = \text{SQRAT}(A - 1)

\text{DARCTANARGUMENTDGMMA} = (1/2) \cdot \text{SQRAT}(A - 1) \cdot \text{SEC}(\text{GAMMA/2}) / (A + 1)

\text{ARCTANARGUMENT} = \text{SQRAT}(A - 1) \cdot \text{TAN}(\text{GAMMA/2}) / (A + 1)

\text{GX} = -\text{SIGMA} (-1 + (2 \cdot A \cdot \text{DARCTANARGUMENTDGMMA}) / (\text{ASP} \cdot \text{ARCTANARGUMENT}))

\text{GX0} = -\text{SIGMA} (1 - ((A/(A + 1)) \cdot \text{SEC}(\text{GAMMAZERO/2})) / (1 + 3 \cdot I + 1)

\text{ARCTANARGUMENTZERO})}
DARCTANARGUMENTDA = TAN(GAMMA/2) / ((A+1) ASP)
DARCTANARGUMENTODA = TAN(GAMMA0/2) / ((A+1) ASP)

AX = -SIGMA (-2(ARCTAN(ARCTANARGUMENT) - ARCTAN(ARCTANARGUMENTZERO))
   I
   3
   /ASP + (2 A/ASP) ((1/(1+ARCTANARGUMENT)) DARCTANARGUMENTDA
   2
   -(1/(1+ARCTANARGUMENTZERO)) DARCTANARGUMENTODA))

location ENDX

GZ = SIGMA (CDO/CL) SIN(GAMMA) / (WZERO+(CDO/CL) (COS(4 I
GAMMA) - COS(GAMMAZERO))

GZO = -SIGMA (CDO/CL) SIN(GAMMAZERO) / (WZERO+(CDO/CL) (COS(3 I+1
GAMMA) - COS(GAMMAZERO))

DZDZERO = WZERO / (WZERO+(CDO/CL) (COS(GAMMA) - COS(GAMMAZERO))
I

Continue this block.

* - -
ELMATRIX = (EL, EL, EL)

* *
EGIKFKEI = EGAMMA (0,0,0,0,0) INVMATRIX (RF K-ELMATRIX)

* *
EGIKFKEI = EGIKFKEI + EGAMMA (EL,0,0,0,0,0,0,0,0,0)

* *
EGIKFKEI = EGIKFKEI + (0,1,0,0,0,1,0,0,1,0)

* *
XG = INVERSE(IDMATRIX-EGAMMA (0,0,0,0,0,0,0,0,0,0,0,0)

* *
EGIKFKEI TSTATE

Do this part for I=0,1,2

* *
DELXEO = (GX , GZ ) XG
8 I 3 I 3 I
* DPLXEO = DELXEO + (GX0, GZ0) TSTATE
  8 I   8 I   3 I   3 I

AXVECT = (AX, 0)
  I

AYGWA = (AX, (CL/CD0), AX, SIN(GAMMAZERO), 0, 0, 0, 0)
  I   I

AXGWATS = AXGWA TSTATE

* DEXLEO = DEXLEO + AXGWA TSTATE
  8 I   8 I

* DEXLEO = DEXLEO + (1, 0, 0, 0, 0, 0, 0, 0)
  8 I   8 I

END OF ETRAJSUB
B.6 Subroutine GDMINSUB

This subroutine is called by LRSRCH. It predicts P response to predicted E motion (from ETRAJSUB) and supplies the required first and second derivatives.

```
DIMENSION (GAMMAMATRIX, 2X4), (VMATRIX, 2X3), (DELTAMATRIX, 2X7),
           (D, 2X7), (A, 3X9),
           (ZMATRIX, 2X4), (AMATRIX, 3X8), (PCMATRIX, 8X8), (PMATRIX, 2X4),
           (APCPGAMMAMATRIX, 3X4), (DELTAPAPCPGAMMAMATRIX, 2X4),
           (VALOMATRIX, 2X4), (VALOMATRIX, 2X2), (VALOMATRIX, 2X2),
           (DVDALMATRIX, 2X2), (DXVECTOR, 3X4), (DZVECTOR, 3X4),
           (DELTAS2, 2), (GAMMAVDALMATRIX, 4X2),
           (GTEMP1, 4X4), (GTEMP2, 4X2), (GRADIENT, 2), (TRUEGRADIENT, 2),
           (DDXDZMATRIX, 3X4), (DELTAPRIMEMATRIX, 4X9), (APRIMEMATRIX, 3X9),
           (PPRIMEMATRIX, 9X11), (TP1, 2), (TP2, 2), (TP3, 2), (XMATRIX, 2X7),
           (ZPRIMEMATRIX, 2X7), (VGMATRIX, 2X6), (VDROP, 9X9), (VDR, 2X9),
           (APR, 11X11), (APR, 3X11), (VDRIAPRP, 9X11), (VDRIAPRP, 2X11),
           (XIVG, 2X7), (ZIVG, 2X7), (VDRIAPRZIVGVI, 2X7), (V1, 7X2), (DVSTATE, 2X5), (DXDSTATE, 2X6), (DZDSTATE, 2X6), (DXPSTATE, 2X4), (DXDESTATE, 3X4), (DXPDAL, 2X4), (DZPSTATE, 2X4), (DZPDAL, 2X4), (DVSTATECOMP, 3X5), (DVDPSTATE, 3X4), (DVDESTATE, 3X4), (DVDCONTROL, 3X4),
           (DZDPSTATE, 3X4), (DZDESTATE, 3X4), (TEMPZ, 3X6), (DZDCONTROL, 3X4),
           (TEMPG, 3X6), (DXPSTATE, 3X4), (DZDCONTROL, 3X4),
           (DGDPSTATE, 3X4), (DGDESTATE, 3X4), (DGDCONTROL, 3X4), (DPDPSTATE,
```
9x4), (dpdestate, 9x4), (dpdcontrol, 9x4), (dadpstate, 3x4),
(dadestate, 3x4), (dadcontrol, 3x4), (gammaderv, 5x8),
(dmoderalderiv, 6), (deltaderiv, 9x14), (zderiv, 4x8), (aderiv, 9x24),
(pcederiv, 1, 6x32),
(pcederiv2, 6x32), (xnderiv, 4x12), (dnderiv, 4x12),
(xnderiv, 6x3), (znderiv, 6x3), (dvdU, 3x4), (dzdu,
3x4), (dgdU, 3x4), (dpdu, 9x4), (dadu, 3x4), (daldu, 2x4), (dxedu, 4x3),
(dxedu, 3x4), (dzedu, 3x4), (dvdudaldut, 5x4), (drvapdu, 6x4),
(dravadut, 9x4), (dgvdud, 4x4), (drsvdud, 6x4), (dpc1, 32x4),
(dpc2, 32x4), (gvduderiv, 6x4), (dexe, 3x4),
(dzvpt, 3x4), (dxzepdu, 4x4),
(dzepdu, 4x3), (dxpodu, 4), (dzeodu, 4), (dgmamatrixdu, 4x8),
(dzerostatedu, 4x4), (dxpodu, 4), (dzpodu, 4),
(dmdomalodu, 4), (dmdeltamatrixdu, 4x14), (dzmamatrixdu, 4x8),
(dpmamatrixdu, 4x32), (dpmamatrixdu2, 4x32), (drhodu, 3x4),
(damamatrixdu, 4x24), (dxdzdu, 4x12), (ddzdu, 4x12),
(dxpdu, 4x3), (dzpdu, 4x3), (d2dxpdu, 4x3), (d2zzpdu, 4x3),
(dmmamadmad, 2x4), (dmmatrixdelta, 2x7), (dmmamad, 2x4),
(dmmadmad, 3x9), (dmmamapcdum, 8x8), (ddxvectorsdu, 3x4),
(ddxvectorsdu, 3x4), (dmmamapudu, 2x4),
(dtemp, 2x2), (drtmdenu, 2x2),
(danmamadmad, 3x4), (ddmamamatapcdum, 2x4), (dvalomdu, 2x4),
(dg1, 4x4), (dg2, 4x2), (dg3, 4x4), (dg4, 4x2),
(dvalodu, 2x2), (dvalo2du, 2x2), (dvdaldud, 2x2), (dmmamadvdaldud, 4x2),
(dgradgrad, 2x4),
(ddxiv, 4), (ddxiv, 4), (doderiv, 3x4), (ddonzereostate, 3x4),
(dtmindx, 4), (dtmindexp, 4), (dtmindex, 4), (dtmindex, 4), (dmmindz, 4),
(dmmindz, 4), (d2tmindexp2, 4),
(dmmindz, 4), (d2tmindexp2, 4),
(dmmindz2, 4), (d2tmindexp2, 4).
(D2TMINDDZPDXP, 4), (D2TMINDDXPDZP, 4), (D2TMINDDZP2, 4), (D2DMINDDXP2, 4),
(D2DMINDDZPDXP, 4), (D2DMINDDXPDZP, 4), (D2DMINDDZP2, 4),
(D2TMINDDZEDXP, 4), (D2TMINDDZEDXP, 4), (D2TMINDDZEDXP, 4),
(D2TMINDDZEDZP, 4), (D2DMINDDXEDXP, 4), (D2DMINDDXEDXP, 4),
(D2DMINDDXEDZP, 4), (D2DMINDDZEDZP, 4), (D2DMINDDUDXP, 4X4),
(D2DMINDDUDZP, 4X4), (D2DMINDDUDXP, 4X4), (D2DMINDDUDZP, 4X4),
(DTRUEGRADIENTDUI, 4X2), (TREUGRAFIENTGFLAIENT, 2X4), (TRUEEDDXXEO, 4)
), (TRUEEDDXXPO, 4)

The variables given by LRSRCHE to GDMINSUB are the same variables given by MMTSRCHE to GDTSUB (see Appendix B, 3). This block initializes the SAA iteration for integrating the \( \dot{V} \) equation.

G=32.17, AREA=17.53, SIGMA=24000
RHOSUBZERO=.002377, M0=233.1

T2=6, T1=(T2+T0)/2, CAPY1=LOG((1-.0804 T1-.00228 T1)/(1-.0804
2
T0-.00228 T0 )), CAPY2=LOG((1-.0804 T2-.00228 T2)/(1-.0804 T0
2
-.00228 T0 )), CAPZ1=LOG((-0.00456 T1-.2052) (-0.00456 T0+.0444)
2
/((-0.00456 T1+.0444) (-0.00456 T0-.2052)) , CAPZ2=LOG((-0.00456 T2
2
-.2052) (-0.00456 T0+.0444) /((-0.00456 T2+.0444) (-0.00456 T0-.2052)
}))
L1=27

Do this part for I=0,1,2

IF I=0, T=T0
IF I=1, T=T1
IF I=2, T=T2
L2=22+(5/8)T

\[ \text{THRUST}=1549.6 (22+(5/8)T) \]

\[ \text{PCOM} = \left( 2^4 + \frac{(L1 - L2)}{4(L1-L2)} \right) \left( \frac{3}{3} \right) \left( \frac{8}{0.00016602} \right) \]

\[ = \left( \frac{484 T + (22 \times 5/8) T + (1/3) (5/8) T}{2^2} \right) \left( \frac{2^3}{2^3} \right) \left( \frac{2}{2} \right) \left( \frac{0.00016602 (484 T + (22 \times 5/8) T + (1/3) (5/8) T)}{2} \right) \]

PMASS=M0 \left( 1 - 0.00016602 (484 T + (22 \times 5/8) T + (1/3) (5/8) T) \right)^2 \left( 2 \right) \left( 3 \right)

ALI=ALOTHEN+ALOPPRIMETHEN(T-T0)

ALPHA1=-.1 SGN(ALI)

IF I=0, MASS0=PMASS, THRUST0=THRUST, CM0=PCOM, ALPHA0=ALPHA1
IF I=1, MASS1=PMASS, THRUST1=THRUST, CM1=PCOM, ALPHA1=ALPHA1
IF I=2, MASS2=PMASS, THRUST2=THRUST, CM2=PCOM, ALPHA2=ALPHA1

Continue with the initialization. Note that AL0 and ALOPRIME are converted from "upstream" to "downstream" values (see Appendix B.1).

RH00=RHOSUBZERO \times \text{EXP}(-Z0/\text{SIGMA})

SINHALFB=SIN(10 \text{ DEGTORAD}/2), COSHALFB=COS(10 \text{ DEGTORAD}/2),
TANHALFB=SINHALFB/COSHALFB
D=2 L1 TANHALFB
L2=22+(5/8)T0

\[ \text{HFGAIN}=-(Q4 \text{ COMPCOEF3}/(1-Q4 \text{ COMPCOEF3})) / 2 \]
TRISE=5/WCLOSEDLOOP
ALT6=AL0+ALOPRIME(6-T0)
AL0=GLIMIT TANH(AL0/GLIMIT)
ALOPRIME=(GLIMIT TANH(ALT6/GLIMIT)-AL0)/(6-T0)
RT=TRISE/(1-(ALOPRIME-ALOPRIMEOLD)TRISE/((AL0-ALOLDT0)(1-HFGAIN)))
GAMMADOTAREA=(1/2)(1-HFGAIN)(AL0-ALOLDT0)RT

This block performs the SAA integration. It is repeated until
\[
\left| \frac{X2-X2CLD}{X2CLD} \right| < .001
\]

V1=V0-438.6(2.5968-DELTA A)(T1-T0)+(-219.298(182.81-DELTA S+2
DELTA A T0)+7373.1(2.5968-DELTA A))CAPY1+(8.0115(3217.5+DELTA S T0
2
-DELTA L-DELTA A T0 )-141.256(182.81-DELTA S+2 DELTA A T0)+8494.97
(2.5968-DELTA A))CAPZ1
V2=V0-438.60(2.5968-DELTA A)(T2-T0)+(-219.298(182.81-DELTA S+2
DELTA A T0)+7373.1(2.5968-DELTA A))CAPY2+(8.0115(3217.5+DELTA S
2
T0-DELTA L-DELTA A T0 )-141.256(182.81-DELTA S+2 DELTA A T0)+
8494.97(2.5968-DELTA A))CAPZ2
F1=T1-T0, F2=T2-T0
GAMMA1=GAMMA0-ALO(5/(12 V0)+2/(3 V1)-1/(12 V2))F1-ALOPRIME(2/(3
2
V1)-1/(6 V2))F1+GAMMADOTAREA/V0
GAMMA2=GAMMA0-ALO(1/(3 V0)+4/(3 V1)+1/(3 V2))F1-ALOPRIME(4/(3
2
V1)+2/(3 V2))F1+GAMMADOTAREA/V0

In this part of the block, X1, X2, Z1, and Z2 are computed. Repeat this
part for I=1,2.

IF I=1, T=F1
IF I=2, T=F2

IF I>2, T=INTEGER(T0)+I-2-T0

\[ X = X0 + V0 \cos(\gamma_0) \left( T - \frac{3}{(4 \ F1)} T + \frac{1}{(6 \ F1)} T \right) + V1 \cos(\gamma_1) \left( \frac{1}{(4 \ F1)} T - \frac{1}{(3 \ F1)} T \right) + V2 \cos(\gamma_2) \left( \frac{1}{(4 \ F1)} T + \frac{1}{(6 \ F1 \ F1)} T \right) \]

\[ Z = Z0 + (V0 \sin(\gamma_0) \left( T - \frac{3}{(4 \ F1)} T + \frac{1}{(6 \ F1)} T \right) + V1 \sin(\gamma_1) \left( \frac{1}{(4 \ F1)} T - \frac{1}{(3 \ F1)} T \right) + V2 \sin(\gamma_2) \left( \frac{1}{(4 \ F1)} T + \frac{1}{(6 \ F1 \ F1)} T \right) \]

ABSTIME=T0+T

IF I=1, X1=X, Z1=Z

IF I=2, X2OLD=X2, X2=X, Z2OLD=Z2, Z2=Z

Continue with the block

RH00=RHOSUBZERO EXP(-Z0/SIGMA)

RH01=RHOSUBZERO EXP(-Z1/SIGMA), RH02=RHOSUBZERO EXP(-Z2/SIGMA)

P00=-(-G COS(\gamma_0) + ALO)

P01=-(-G COS(\gamma_1) + ALO + ALOPRIME F1)

P02=-(-G COS(\gamma_2) + ALO + ALOPRIME F2)

2

\[ P10 = \frac{1}{2} \left( RH00 \ V0 \ AREA/MASS0 \right) (1-8.862/CM0) 1.7473-1.7738 - \]

\[ \text{THRUST0/MASS0} \]

2

\[ P11 = \frac{1}{2} \left( RH01 \ V1 \ AREA/MASS1 \right) (1-8.862/CM1) 1.7473-1.7738 - \]

\[ \text{THRUST1/MASS1} \]

2

\[ P12 = \frac{1}{2} \left( RH02 \ V2 \ AREA/MASS2 \right) (1-8.862/CM2) 1.7473-1.7738 - \]
THRUST2/MASS2

\[ P_{20} = + \left( \frac{1}{2} \right) \frac{\text{RHO0}}{V_0} \text{AREA/MASS0} \text{SGN}(\text{ALPHA0}) \left( 2.1908(1-8.862/\text{CM0}) - 1.5885 \right) \]

\[ P_{21} = + \left( \frac{1}{2} \right) \frac{\text{RHO1}}{V_1} \text{AREA/MASS1} \text{SGN}(\text{ALPHA1}) \left( 2.1908(1-8.862/\text{CM1}) - 1.5885 \right) \]

\[ P_{22} = + \left( \frac{1}{2} \right) \frac{\text{RHO2}}{V_2} \text{AREA/MASS2} \text{SGN}(\text{ALPHA2}) \left( 2.1908(1-8.862/\text{CM2}) - 1.5885 \right) \]

IF \( \frac{P_{10}}{2 \ P_{20}} < P_{00}/P_{20}, \text{ALPHA0} = -P_{10}/(2 \ P_{20}), \text{AS0} = 0, \)

GO TO ASKP1, OTHERWISE

\[ \text{ALPHA0I} = \left( \frac{P_{10}}{2 \ P_{20}} \right) + \sqrt{\left( \frac{P_{10}}{2 \ P_{20}} \right) - P_{00}/P_{20}}, \]

\[ \text{ALPHA0J} = \left( \frac{P_{10}}{2 \ P_{20}} \right) - \sqrt{\left( \frac{P_{10}}{2 \ P_{20}} \right) - P_{00}/P_{20}} \]

IF ABS(ALPHA0I) > ABS(ALPHA0J), \text{ALPHA0} = ALPHA0J, \text{AS0} = -1, OTHERWISE

\[ \text{ALPHA0} = \text{ALPHA0I}, \text{AS0} = +1 \]

location ASKP1

\[ \text{DOOVERM0} = \left( \frac{1}{2} \right) \text{RHO0} \ V_0 \ \text{AREA}(1/\text{M0}) \left( .01519 - \text{SGN}(\text{ALPHA0}) \cdot .0149119 \right) \]

\[ \text{ALPHA0} + 3.07089 \ \text{ALPHA0} \]

IF \( \frac{P_{11}}{2 \ P_{21}} < P_{01}/P_{21}, \text{ALPHA1} = -P_{11}/(2 \ P_{21}), \text{AS1} = 0, \)

GO TO ASKP2, OTHERWISE

\[ \text{ALPHA1I} = \left( \frac{P_{11}}{2 \ P_{21}} \right) + \sqrt{\left( \frac{P_{11}}{2 \ P_{21}} \right) - P_{01}/P_{21}}, \]

\[ \text{ALPHA1J} = \left( \frac{P_{11}}{2 \ P_{21}} \right) - \sqrt{\left( \frac{P_{11}}{2 \ P_{21}} \right) - P_{01}/P_{21}} \]
IF AES(ALPHA1I) > ABS(ALPHA1J), ALPHA1 = ALPHA1J, AS1 = -1, OTHERWISE
ALPHA1 = ALPHA1I, AS1 = +1

location ASKP2

2
IF (P12/(2 P22)) < P02/P22, ALPHA2 = -P12/(2 P22), AS2 = 0,
GO TO ASKP3, OTHERWISE

ALPHA2I = -(P12/(2 P22)) + SQRT((P12/(2 P22))
-P02/P22),

ALPHA2J = -(P12/(2 P22)) - SQRT((P12/(2 P22))
-P02/P22)

IF AES(ALPHA2I) > ABS(ALPHA2J), ALPHA2 = ALPHA2J, AS2 = -1, OTHERWISE
ALPHA2 = ALPHA2I, AS2 = +1

location ASKP3

D1OVERR0 = (1/2) RHO1 V1, AREA(1/M0) (.01519 - ABS(ALPHA1) .0149119
+3.07089 ALPHA1 )

D2OVERR0 = (1/2) RHO2 V2, AREA(1/M0) (.01519 - ABS(ALPHA2) .0149119
+3.07089 ALPHA2 )

- DELTAAS = INVERSE(P1, F1, F2, P2) (D1OVERR0 - D2OVERR0,
D2OVERR0 - D2OVERR0)
DELTALOLD = DELTAL, DELTASOLD = DELTAS, DELTAAOLD = DELTAA
DELTAL = D2OVERR0, DELTAS = DELTAAS, DELTAA = DDELTAAS

This block initializes the first derivative computation.
L1=TAU1, L2=TAU2, L3=TAU3

\[
\begin{align*}
\text{LAMBDAMATRIX} &= (L1, L1, L1, L2, L2, L3, L3, L3) \\
\text{ADENOM} &= (1 - (ATLPRIME-ATLPRIMEOLD) TRISE / ((AT0-ATOLD) (1 - 2 HFGAIN)))) \\
\text{DTRDALO} &= (1/ADENOM) TRISE (ATLPRIME-ATLPRIMEOLD) / ((2 AT0-ATOLD) (1 - HFGAIN)) \\
\text{DTRDALOPRIME} &= (1/ADENOM) TRISE / ((AT0-ATOLD) (1 - HFGAIN)) \\
\text{DAREADALO} &= (1/2) (1 - HFGAIN) (RT + (AT0-ATOLD) DTRDALO) \\
\text{DAREADALOPRIME} &= (1/2) (1 - HFGAIN) (AT0-ATOLD) DTRDALOPRIME \\
\text{MSUB0} &= MO, PCM1 = CM1, PCM2 = CM2, PA0 = ALPHA0, PA1 = ALPHA1, PA2 = ALPHA2 \\
\text{XP0} &= X0, ZP0 = ZO, GAMMAZERO = GAMMA0 \\
\text{AREA} &= AREA
\end{align*}
\]

This block is repeated for I=1, 2, 3 to compute XP1, XP2, XP3, ZP1, ZP2 and ZP3.

\[
\begin{align*}
\text{XPI} &= X0 + V0 \cos(GAMMA0) (LI - (3/(4 F1)) LI + (1/(6 F1)) LI) \\
&\quad + V1 \cos(GAMMA1) ((1/F1) LI - (1/(3 F1)) LI) + V2 \cos(GAMMA2) ((2 LI) LI + (1/(6 F1)) LI) \\
\text{ZPI} &= Z0 + V0 \sin(GAMMA0) (LI - (3/(4 F1)) LI + (1/(6 F1)) LI) \\
&\quad + V1 \sin(GAMMA1) ((1/F1) LI - (1/(3 F1)) LI) + V2 \sin(GAMMA2) ((-1/(4 F1)) LI + (1/(6 F1)) LI) \\
\text{IF } I &= 1, \text{ XP1=XPI, ZP1=ZPI, D1=D1} \\
\end{align*}
\]
IF \( i = 2 \), \( x_2 = x_1, z_2 = z_1, d_2 = d_1 \)
\[ \text{IF } i = 3, \ x_3 = x_1, z_3 = z_1, d_3 = d_1 \]

At this point the subroutine DISTFIT is called. The miss and miss time are computed along with appropriate first and second derivatives. The variables returned follow:

\[
\text{TMINREL, TRUEDMIN, DTMINDXE, DTMINDXP, DTMINDZE, DTMINDZP,}
\]
\[
\text{DDMINDXE, DDMINDXP, DDMINDZE, DDMINDZP, D2TMINDXP2, D2TMINDZPDXP,}
\]
\[
\text{D2TMINDXPDPZP, D2TMINDZP2, D2DMINDXP2, D2DMINDZPDXP, D2DMINDXPDPZP,}
\]
\[
\text{D2DMINDZP2}
\]

The desired first derivatives are computed in this block.

\[
\text{D2TMINDXEDXP} = -D2TMINDXP2, D2TMINDZEDXP = -D2TMINDZPDXP
\]
\[
\text{D2TMINDXEDZP} = -D2TMINDXPDPZP, D2TMINDZEDZP = -D2TMINDZP2
\]
\[
\text{D2DMINDXEDXP} = -D2DMINDXP2, D2DMINDZEDXP = -D2DMINDZPDXP
\]
\[
\text{D2DMINDXEDZP} = -D2DMINDXPDPZP, D2DMINDZEDZP = -D2DMINDZP2
\]

\[
\text{TRUETMIN} = \text{TMINREL} + 0
\]

\[
\text{GAMMAMATRIX} = \left( -\frac{5}{12} v_0 + 2/3 \ v_1 - 1/(12 \ v_2) \right) F_1 + \text{DAREADAL0}/v_0,
\]
\[
\left( -2/(3 \ v_1) - 1/(6 \ v_2) \right) F_1 + \text{DAREADAL0PRIME}/v_0,
\]
\[
\left[ (2/(3 \ v_1)) (A_{lo} F_1 + A_{loprime} F_1) \right] - \left( 1/(12 \ v_2) \right) (2) F_1 + \text{DAREADAL0}/v_0,
\]
\[
\left( -4/(3 \ v_1) + 2/(3 \ v_2) \right) F_1 +
\]
\[
\text{DAREADALOPRIME}/V_0, *((4/(3 \ V_1)) (A_0 \ F_1 + ALOPRIME \\
F_1^2, *(1/(3 \ V_2)) (A_0 \ F_1 + 2 \ ALOPRIME \ F_1))
\]

\* 

\[
\text{VMATRIX}=(-8.0115 \ CAPZ_1, 219.298 \ CAPY_1 + (8.0115 \ T0 + 141.256 \\
\text{CAPZ_1}, 438.60 \ F1 - (2 \ T0 + 219.298 + 7733.1) \ CAPY_1 - (8.0115 \ T0 + 2 \ T0 \\
141.256 + 8494.97) \ CAPZ_1, 219.298 \ CAPY_2 + (8.0115 \ T0 + 141.256) \ CAPZ_2, 438.60 \ F2 - (2 \ T0 + 219.298 + 7733.1) \ CAPY_2 - (8.0115 \\
T0 + 2 \ T0 + 141.256 + 8494.97) \ CAPZ_2)
\]

\[
\text{DMDODRHO}_0=(1/2) V_0 \ \text{PAREA (1/MSUB0)} (.01519 - SGN(PA0) .0149119 \\
PA0 + 3.0709 \ PA0
\]

\[
\text{DMDODRHO}_1=(1/2) V_1 \ \text{PAREA (1/MSUB0)} (.01519 - SGN(PA1) .0149119 \\
PA1 + 3.0709 \ PA1
\]

\[
\text{DMDODRHO}_2=(1/2) V_2 \ \text{PAREA (1/MSUB0)} (.01519 - SGN(PA2) .0149119 \\
PA2 + 3.0709 \ PA2
\]

\[
\text{DMDODALPHA}_0=(1/2) RHOO \ V_0 \ \text{PAREA (1/MSUB0)} (-.0149119 \ SGN(PA0) \\
+ 2 (3.0709) \ PA0)
\]

\[
\text{DMDODALPHA}_1=(1/2) RHOO \ V_1 \ \text{PAREA (1/MSUB0)} (-.0149119 \ SGN(PA1) \\
+ 2 3.0709 \ PA1)
\]

\[
\text{DMDODALPHA}_2=(1/2) RHOO \ V_2 \ \text{PAREA (1/MSUB0)} (-.0149119 \ SGN(PA2) \\
+ 2 3.0709 \ PA2)
\]

\[
\text{DMDODV}_0=(2/V_0) \ \text{DOOVERMO}
\]

\[
\text{DMDODV}_1=RHOO \ V_1 \ \text{PAREA (1/MSUB0)} (.01519 - SGN(PA1) .0149119 \ PA1}
\]
+ 3.0709 PA1 )

DDMODV2 = RHC2 V2 PAREA (1/MSUBO) (0.01519 - SGN(PA2) .0149119 PA2

+ 3.0709 PA2 )

* DELTAMATRIX = ((4/F2) DDMODRHO1, (-1/F2) DDMODRHO2, (-3/F2)

DDMODALPHA0, (4/F2) DDMODALPHA1

2

DDMODRHO1, (2/F2) DDMODRHO2, (2/F2) DDMODALPHA0, (-4/F2 )

2

DDMODALPHA1, (2/F2 )

2

DDMODALPHA2, (-4/F2 ) DDMODV1, (2/F2 ) DDMODV2)

* ZMATRIX = ((2/3) F1 SIN(GAMMA1), -(1/12) F1 SIN(GAMMA2), (2/3) F1

V1 COS(GAMMA1), -(1/12) F1 V2 COS(GAMMA2), SIN(GAMMA1) ((F2 /F1) -

3 2

(F2 / (3 F1 ))), SIN(GAMMA2) ((-1/(4 F1 ))F2 + (1/(6 F1 ))F2 ),

2

V1 COS(GAMMA1) ((1/F1 )F2 - (1/(3 F1 ))F2 ), V2 COS(GAMMA2)

2

((-1/(4 F1 ))F2 + (1/(6 F1 ))F2 ))

PSQRTTERM0 = SQRT((P10/(2 P20)) - (P00/P20))

PSQRTTERM1 = SQRT((P11/(2 P21)) - (P01/P21))

PSQRTTERM2 = SQRT((P12/(2 P22)) - (P02/P22))

* AMATRIX = ((1/2) AS0 (-1/P20) /PSQRTTERM0, (P10/(2 P20)) - (1/2)

2

AS0 (-(P10/(2 P20)) + (P00/P20)) /PSQRTTERM0,0,0,0,0,0,0,
0, (1/2) AS1(-1/P21)/SQRTERM1,

(-1/(2 P21)) + AS1(1/2) (P11/(2 P21)) /SQRTERM1

2

+ (P11/(2 P21)) + AS1(1/2) (- (P11/(2 P21))

2

+(P01/P21)) /SQRTERM1, 0, 0, 0, 0, 0, 0, 0, 0, 0, (1/2) AS2(-1/P22)

/PSQRTERM2, (-1/(2 P22)) + (1/2) AS2(P12/(2 P22))

2

2

1/PSQRTERM2, (P12/(2 P22)) + AS2(1/2) (- (P12/(2

2

3

P22)) + (P02/P22)) /PSQRTERM2)

* PCMATRIX= (-1, 0, 0, 0, 0, 0, 0, 0, 0.

0, 0, 0, 0, 0, 0, 0, -1, -F1, 0, 0, 0, 0, -G*SIN(GAMMA1),

2

0, 0, 0, 0, (1/2) (PAREA V1/MASS1) ((1-8.862/PCM1) 1.7473-1.7738),

0, (RHO1 V1 PAREA/MASS1) ((1-8.862/PCM1) 1.7473-1.7738), 0, 0, 0, 0

0, (1/2) (V1 PAREA/MASS1) SGN(ALPHA1) (2.1908 (1

-8.862/PCM1) -1.5885), 0, (+RHO1 V1 PAREA/MASS1) SGN(ALPHA1

) (2.1908 (1-8.862/PCM1) -1.5885), 0, 0, 0, -1, -F2, 0, 0, 0, 0, 0,

-2

-G*SIN(GAMMA2), 0, 0, 0, (1/2) (V2 PAREA

/MASS2) ((1-8.862/PCM2) 1.7473-1.7738), 0, (RHO2 V2 PAREA/MASS2)

((1-8.862/PCM2) 1.7473-1.7738), 0, 0, 0, 0, 0, (+1/2) (V2 PAREA/

MASS2) SGN(ALPHA2) (2.1908 (1-8.862/PCM2) -1.5885),

0, (RHO2 V2 PAREA/MASS2) SGN(ALPHA2) (2.1908 (1-

8.862/PCM2) -1.5885), 0, 0)

* PMATRIX=-(1/SIGMA)PHOSUBZPRO (EXP(-Z1/SIGMA), 0, 0, EXP(-Z2/
* SIGMA)) ZMATRIX (0,0,1,0,0,0,0,1,GAMMAMATRIX)

* 

* ACPGAMMAMATRIX=AMATRIX PCMATRIX (1,0,0,0,0,1,0,0,PMATRIX,

* 0,0,1,0,0,0,0,1,GAMMAMATRIX)

* 

* DELTAPAPCPGAMMAMATRIX=DELTAMATRIX (PMATRIX, ACPGAMMAMATRIX, 0,0,
1,0,0,0,1)

* 

* VALOMATRIX=VMATRIX ((1/2) RHO0 V0 (PAREA/MSUB0) (-0.0149119 SGN(PA0)+2 3.0709 PA0) (1/2)AS0 (-1/P20) (1/SQRT ((F10/(2 P20)) 2

(P00/P20)) (-1),0,0,0,DELTAPAPCPGAMMAMATRIX)

* 

* VAL01MATRIX=(VALOMATRIX ,VALOMATRIX ,VALOMATRIX ,VALOMATRIX )

0 1 4 5

* 

* VAL02MATRIX=(VALOMATRIX ,VALOMATRIX ,VALOMATRIX ,VALOMATRIX )

2 3 6 7

* 

* DMDALMATRIX=-INVERSE(VAL02MATRIX- (1,0,0,1)) VAL01MATRIX

* 

* GTEMP1 = (0,0,1,0,0,0,0,1,GAMMAMATRIX)

* 

* GTEMP2=(1,0,0,1,DMDALMATRIX)

* 

* GAMMADMDALMATRIX=GTEMP1 GTEMP2

This part is repeated for I=0,1,2

IF I=0, LI=L1
IF I=1, LI=L2
IF I=2, LI=L3

- DXVECTOR = (COS (GAMMA1) ((1/F1) LI -(1/(3 F1)) LI ), COS (GAMMA2)
264

((-1/(4 F1))LI + (1/(6 F1))LI), -V1 \sin(\text{GAMMA1}) ((1/F1)LI -
2 \quad 3
(1/(3 F1))LI), -V2 \sin(\text{GAMMA2}) ((-1/(4 F1))LI + (1/(6 F1))LI)
)

DZVECTOR = (\sin(\text{GAMMA1}) ((1/F1)LI - (1/(3 F1))LI), \sin(\text{GAMMA2})
4 \quad I

((-1/(4 F1))LI + (1/(6 F1))LI), V1 \cos(\text{GAMMA1}) ((1/F1)LI - (1/(
2 \quad 3
3 F1))LI), V2 \cos(\text{GAMMA2}) ((-1/(4 F1))LI + (1/(6 F1))LI))

Continue the block.

** TRUEGRADIENT = (DDMINDXP (0, 0, 0, 0, DZVECTOR) + DDMINDZP (0, 0, 0, 0,
* DZVECTOR)) GAMMACVDBALMATRIX

- GRADIENT = TRUEGRADIENT

TMIN = TRUEMIN, DMIN = TRUEDMIN

This block contains second derivative calculations.

* * *
D = DELTAMATRIX

* DELTAPRIMEMATRIX = 

(0, 0, 0, 0, 0, 1, 0, 0, DDMODRHOO, 0, 0, DDMODALPHA0, 0, 0, DDMODV0, 0, 0, 0,
-(3/F2) DDMODRHOO, D, D, D, D, -(3/F2) DDMODV0, D, D, (2/
0 1 2 3 4 5 6
F2) DDMODRHOO, D, D, D, D, (2/F2) DDMODV0, D, D )

* *
A = AMATRIX
* APRIMEMATRIX= (A, -(1/(2 P20)) + AS0 P10/(2 P20) PSQRTTERM0
  
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
), A, A, A, A, A, A, A, 
  17 18 19 20 21 22 23

* PPRIMEMATRIX= (0, 0, 0, -G SIN(GAMMAZERO), 0, 0, 0, 0, -1, 0, (1/2) (V0
  PARFA/MASSO) ((1-8.862/CMO) 1.7473-1.7738), 0, 0, 0, 0, (RHO0 V0
  PARFA/MASSO) ((1-8.862/CMO) 1.7473-1.7738), 0, 0, 0, 0, P20/RHOO,
  0, 0, 0, 0, 0, 2 P20/V0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, PCMATRIX
  0, 0, 0, 0, -1, -F1, 0, 0,
  22
PCMATRIX ,0, 0, 0, 0, 0, PCMATRIX ,0, 0, 0, 0, PCMATRIX ,0, 0, 0, 0,
  26 28 34
PCMATRIX ,0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  36 47
PCMATRIX ,0, 0, 0, 0, 0, PCMATRIX ,0, 0, 0, 0, PCMATRIX ,0, 0, 0, 0,
  51 53 59
PCMATRIX ,0, 0)
  61

* RHCMATRIX= (-1/SIGMA) (RHOO, 0, 0, 0, RHO1, 0, 0, 0, RHO2)

Do this part for I=0, 1

IF I=0, T=F1, OTHERWISE T=F2

IF I

TF1 = T-3 T/(4 F1)+T/(6 F1)

TF2 = T/F1-T/(3 F1)

TF3 = -T/(4 F1)+T/(6 F1)

Continue the block.
\[ \begin{align*}
SG_0 &= \sin(\gamma_0), \quad SG_1 = \sin(\gamma_1), \quad SG_2 = \sin(\gamma_2), \quad CG_0 = \cos(\gamma_0), \\
&\quad \quad CG_1 = \cos(\gamma_1), \quad CG_2 = \cos(\gamma_2) \\
\text{**} \quad XMATRIX &= \begin{pmatrix}
1 & -v_0 & SG_0 & TF_1 \\
0 & 0 & 0 & 0 \\
CG_1 & TF_2 & SG_2 & TF_3 \\
0 & 0 & 1 & -v_1 \\
CG_0 & TF_1 & CG_1 & TF_2 \\
1 & 1 & 1 & 1
\end{pmatrix} \\
\text{**} \quad ZPIMEMATRIX &= \begin{pmatrix}
1 & v_0 & CG_0 & TF_1 \\
0 & 2 & 3 & 0 \\
ZMATRIX & ZMATRIX & 1 & v_0 \\
0 & 1 & 6 & 7 \\
ZMATRIX & ZMATRIX \\
4 & 5
\end{pmatrix} \\
\text{**} \quad VGMATRIX &= \begin{pmatrix}
1, (5/12) \cdot \text{ALO} & P_1 / v_0 \\
2 & 2 \\
\text{GAMMAMATRIX} & \text{GAMMAMATRIX} \\
3 & 0 \\
\text{GAMMAMATRIX} \\
2 & 1
\end{pmatrix} \\
\text{**} \quad VDSP &= \begin{pmatrix}
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
\end{pmatrix}
\text{**} \quad \text{VDSP} &= \text{RHOMATRIX} \cdot \text{VDSP} = \text{RHOMATRIX} \cdot \text{VDSP} = \text{RHOMATRIX} \\
0 & 10 & 4 & 20 & 8 \\
\text{**} \quad \text{VDR} &= \begin{pmatrix}
1, VMATRIX \\
0 & 3
\end{pmatrix} \cdot \text{DELTAPRIMEMATRIX} \cdot \text{VDSP}
\end{align*} \]
*        APRP=(1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,
    0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
    0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1)

APRP = RHMATRIX, APRP = RHMATRIX, APRP = RHMATRIX

0    0    12    4    24    8

*        *        *        *        *        *

APR = APRIMEMATRIX PPRIMEMATRIX APRP

*        *

VDRIAPRP = (1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,

*        *

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0

,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0

*        *

VDRIAPR = VDR VDRIAPRP

*        *

XIVG = XMATRIX (1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

-        -

VGMATRIX, 0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,1,0,0

-        -

ZIVG = ZPRIMEMATRIX (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

-        -

0, VGMATRIX, 0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0

*        *

VDRIAPR ZIVG VGMATRIX = VDRIAPR (1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

-        -

VGMATRIX, 0, VGMATRIX, 0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0

0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0

*        *

VI = (0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0)
* * 
DUSTATE=INVERSE ((1,0,0,1)-VDRIAVRZIVGVGI VDRIAVRZIVGVGI 
(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0, 
0,0,0,1) 
* * 
DXDSTATE=XIVG (1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0, 
- 
DUSTATE ,0,DUSTATE ,0,0,0,0,1,0,0,0,0,0,0,0,1) 
5 
* * 
DZDSTATE=ZIVG (0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0, 
- 
DUSTATE ,0,0,0,0,1,0,0,0,0,0,0,0,1) 
5 
* * 
DXPDSTATE=DXDSTATE (1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0, 
- 
SIN(GAMMAO)/V0, 
COS(GAMMAO)/V0,0,0,COS(GAMMAO),SIN(GAMMAO),0,0,0,0,0,0,0,0,0) 
* 
DXDESTATE=0 
* * 
DXPDAL=DXDSTATE (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0, 
0,0,0) 
* * 
DZPDSTATE=DZDSTATE (1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0, 
- 
SIN(GAMMAO)/V0, 
COS(GAMMAO)/V0,0,0,COS(GAMMAO),SIN(GAMMAO),0,0,0,0,0,0,0,0,0) 
* * 
DZPDAL=DZDSTATE (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0, 
0,0,0) 
* 
DUSTATECOMP=(0,0,1,0,0,DUSTATE) 
* * 
DVPSTATE=DUSTATECOMP (0,1,0,0,0,0,0,-SG0/V0,CG0/V0,0,0,CG0,SG0, 
0,0,0,0,0,0,0,0) 
* 
DVESTATE=0
* * *
DVDCONTROL=DVSTATECOMP (0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0)
*
DZDPSTATE=(0,1,0,0,DZPDSTATE)
*
DZDESTATE=0
*
TEMPZ=(0,1,0,0,0,0,0,DZDESTATE)
*
DZDCONTRL=TEMPZ (0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0)
*
DZDPSTATE=(1,0,0,0,DZPDSTATE)
*
DZDCONTRL=(0,0,0,0,DZPDAL)
*
TEMPG=(1,0,0,0,0,0,VGMATRIX)
*
DGDPSTATE=TEMPG (0,0,-SG0/V0,CG0/V0,DVDPSTATE,0,0,0,0,0,0,0)
*
DGDESTATE=0
*
DGDCONTRL=TEMPG (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
*
DPDPSTATE=PRIMEMATRIX (RHMATRIX DZDPSTATE, DGDPSTATE,
*
DVDPSTATE,0,0,0,0,0,0,0,0,0,0,0)
*
DPDESTATE=0
*
DPDCONTRL=PRIMEMATRIX (RHMATRIX DZDCONTRL,DGDCONTRL,
*
DVDCONTRL,1,0,0,0,0,1,0,0)
DADPSTATE=APRIMEMATRIX DPDPSTATE

DADESTATE=0

DADDCONTROL=APRIMEMATRIX DPDCONTROL

TRD1=ALOPRIME-ALOPRIMEOLD, TRD2=1-HFGAIN, TRD3=AL0-ALOLDT0

D2TRDAL02= 2 TRISE (1/ADENOM) (TRD1 TRISE/(TRD2 TRD3 ))

+2 TRISE TRD1 (1/ADENOM) /(TRD2 TRD3 )

D2TRDALODALOP=-2 TRISE (1/ADENOM) TRD1/(TRD2 TRD3 )

- TRISE (1/ADENOM) /(TRD2 TRD3 )

D2TRDAL0P2=2 TRISE (1/ADENOM) (1/(TRD2 TRD3 ))

D2ARFADAL02=(1/2) TRD2 (2 DTRDAL0+TRD3 D2TRDAL02)

D2ARFADAL0P2=(1/2) TRD2 TRD3 D2TRDAL0P2

D2ARFADALODALOP=(1/2) TRD2 (DTRDALOPRIME+TRD3 D2TRDALODALOP)

\[
\text{GAMMADERIV } = (5 F1/(12 V0) - \text{DAREADAL0/VO} , 2 F1/(3 V1) , -F1/(12 V2) , D2AREADAL0/V0, D2AREADALODALOP/V0)
\]

\[
\text{GAMMADERIV } = (\text{DAREADALOPRIME/VO} , 2 F1/(3 V1) , F1/(-6 V2) , D2AREADALODALOP/V0, D2AREADALOP2/V0)
\]

\[
\text{GAMMADERIV } = (0,-4(\text{AL0 F1+ALOPRIME F1})/(3 V1) , 0, 2 F1/(3 V1) , 2 F1/(3 V1))
\]

\[
\text{GAMMADERIV } = (0,0, (\text{AL0 F1+2 ALOPRIME F1})/(6 V2) , -F1/(12 V2))
\]
\[ \text{Gammadef} = \frac{F_1}{(3 \cdot V_0)} \]
\[ \text{Gammaderv} = \frac{2}{4} \left( \frac{F_1}{(3 \cdot V_1)}, \frac{F_1}{(3 \cdot V_2)} \right) \]
\[ \text{Gammaderv} = \frac{2}{5} \left( \frac{F_1}{(3 \cdot V_1)}, \frac{2}{2} \frac{F_1}{(3 \cdot V_2)} \right) \]
\[ \text{Gammaderv} = \frac{2}{6} \left( \frac{2}{4} \frac{F_1}{(3 \cdot V_1)} \right) \]
\[ \text{Gammaderv} = \frac{2}{7} \left( \frac{2}{2} \frac{F_1}{(3 \cdot V_2)} \right) \]

\[ \text{DMoaloter} = \text{DDmodalpha} \frac{A_0}{(2 \cdot P_20 \cdot \text{Psqrttermo})} \]
\[ \text{Dmodaloderiv} = \left( \frac{\text{DMoaloter}}{\rho_{00}}, 2 \cdot \text{DMoaloter} / V_0, 2 \cdot 3.0709 \right) \]
\[ \text{DMoaloter} / (-0.0149119 \cdot \text{sgn}(P_{a0}) + 2 \cdot 3.0709 \cdot P_{a0}) \]
\[ \text{DMoaloter} / (4 \cdot P_{20} \cdot \text{Psqrttermo}) \]
\[ \left( \frac{2}{2} \frac{2}{2} \frac{2}{2} \frac{2}{2} \right) \]
\[ \left( \frac{2}{2} \frac{3}{2} \frac{2}{2} \frac{2}{2} \right) \]
\[ \left( \frac{2}{2} \frac{3}{2} \frac{2}{2} \frac{2}{2} \right) \]

\[ \text{Deltaderiv} = \frac{0, 0, 0, 0}{(2 / V_1)} \frac{\text{DModrhom}, 0, 0, (1 / 2) V_1}{\text{Parea/MSUB0}} \]
\[ -0.0149119 \cdot \text{sgn}(P_{a1}) + 2 \cdot 3.0709 \cdot P_{a1}, 0) \]
\[ \text{Deltaderiv} = \frac{0, 0, 0, 0, 0}{(2 / V_2)} \frac{\text{DModrhom}, 0, 0, (1 / 2) V_2}{\text{Parea/MSUB0}} \]
\[ -0.0149119 \cdot \text{sgn}(P_{a2}) + 2 \cdot 3.0709 \cdot P_{a2}) \]
\[ \text{DELTADERIV} = (D \text{D}\text{MODALPHA}0/\text{RHO}0,0,0,2 \text{D}\text{D}\text{MODALPHA}0/\text{V}0,0,0,3,0709 \]
\[ 2 \]
\[ \text{RHO0 V0 PAREA/MSUB0,0,0) \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (0, \text{D}\text{D}\text{MODALPHA}1/\text{RHO}1,0,0,2 \text{D}\text{D}\text{MODALPHA}1/\text{V}1,0,0,3,0709 \]
\[ 2 \]
\[ \text{RHO1 V1 PAREA/MSUB0,0) \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (0,0, \text{D}\text{D}\text{MODALPHA}2/\text{RHO}2,0,0,2 \text{D}\text{D}\text{MODALPHA}2/\text{V}2,0,0,3,0709 \]
\[ 2 \]
\[ \text{RHO2 V2 PAREA/MSUB0) \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (0, \text{D}\text{D}\text{MODV1/\text{RHO}1,0,0,\text{D}\text{D}\text{MODV1/\text{V}1,0,0,\text{RHO1 V1 (PAREA/}} \]
\[ 5 \]
\[ \text{MSUB0) (-0.0149119 SGN(PA1)+2 3.0709 PA1),0) \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (0,0, \text{D}\text{D}\text{MODV2/\text{RHO}2,0,0,\text{D}\text{D}\text{MODV2/\text{V}2,0,0,\text{RHO2 V2 (PAREA}} \]
\[ 6 \]
\[ \text{MSUB0) (-0.0149119 SGN(PA2)+2 3.0709 PA2)) \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (-4/F2)\text{DELTADERIV,DELTADERIV} = (2/F2)\text{DELTADERIV \[ 7 \]
\[ 8 \]
\[ 9 \]
\[ 10 \]
\[ 11 \]
\[ 12 \]
\[ 13 \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (2/F2)\text{DELTADERIV,DELTADERIV} = (-4/F2) \]
\[ 4 \]
\[ 12 \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (2/F2)\text{DELTADERIV,DELTADERIV} = (2/F2) \]
\[ 5 \]
\[ 13 \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (4/F2)\text{DELTADERIV,DELTADERIV} = (-1/F2) \]
\[ 0 \]
\[ 1 \]
\[ \text{C} \]
\[ \text{DELTADERIV} = (-3/F2)\text{DELTADERIV,DELTADERIV} = (4/F2) \]
\[ 2 \]
\[ 3 \]
\[ 4 \]
\[ 5 \]
\[ 6 \]
DELTADEIV = (-1/F2) DELTADEIV
        C   C
        4   5

DELTADEIV = (-1/F2) DELTADEIV
        C   C
        6   6

ZDERIV = (CG1, TF2, 0, 0, 0)
        0

ZDERIV = (CG1, TF2, 0, 0, 0)
        4

ZDERIV = (0, CG2, TF3, 0, 0)
        1

ZDERIV = (0, CG2, TF3, 0, 0)
        5

ZDERIV = (-V1, SG1, 0, CG1, 0) TF2
        2

ZDERIV = (-V1, SG1, 0, CG1, 0) TF2
        6

ZDERIV = (0, -V2, SG2, 0, CG2) TF3
        3

ZDERIV = (0, -V2, SG2, 0, CG2) TF3
        7

SGN0 = AS0, SGN1 = AS1, SGN2 = AS2

* ADERIV = 0

Do this part for I=0,1,2

IF I=0, P0=P00, P1=P10, P2=P20, PST=PSQRTTERM0, SG=SGN0
IF I=1, P0=P01, P1=P11, P2=P21, PST=PSQRTTERM1, SG=SGN1
IF I=2, P0=P02, P1=P12, P2=P22, PST=PSQRTTERM2, SG=SGN2
AP0DERIV = ((-1/4)SG P2 /PST, (1/8)SG P1/(PST P2), (-1/4)SG P0/(P2 PST))

AP1DERIV = ((1/8)SG (P1/P2)) /PST, (-1/16)SG (P1 /P2 ) /PST + (1/4)SG (1/P2 ) /PST, (1/(2 P2 ))+ (-1/8)SG (P1/P2 ) (-1/2) (P1 /P2 ) + (P0/P2 ) /PST + (1/4)SG (-2 (P1/P2 )) /PST

AP2DERIV = ((1/4)SG ((-1/2) (P1 /P2 ) + (P0 /P2))) /PST + (1/2)SG/(P2 PST), 1/(2 P2 ) + (-1/8)SG (P1/P2 ) ((-1/2) (P1 /P2 ) + (P0/P2 )) /PST + (1/2)SG(-P1/P2 ) /PST, - (P1/P2 ) + (-1/4)SG ((-1/2) (P1 /P2 ) + (P0/P2 )) /PST + (1/2)SG((3/2) (P1 /P2 ) -2 (P0/P2 )) /PST

IF I=0, ADERIV = (AP0DERIV, 0, 0, 0, 0, 0), ADERIV = (AP2DERIV, 0, 0, 0, 0, 0)

IF I=1, ADERIV = (0, 0, 0, AP0DERIV, 0, 0, 0), ADERIV = (0, 0, 0, AP2DERIV, 0, 0, 0)

IF I=2, ADERIV = (0, 0, 0, 0, 0, AP0DERIV), ADERIV = (0, 0, 0, 0, AP2DERIV)

AP1DERIV), ADERIV = (0, 0, 0, 0, 0, AP2DERIV)

Continue the block.

PCDERIV1=0, PCDERIV2=0
Do this part for \( I=0,1 \)

\[
\begin{align*}
\text{IF } I=0, & \text{ RHO=RHO1, GAMMA=GAMMA1, } V=V1, T=T1, PCM=PCM1, \text{ MASSI=MASS1, } \\
& \text{ALPHA1=ALPHA1} \\
\text{IF } I=1, & \text{ RHO=RHO2, GAMMA=GAMMA2, } V=V2, T=T2, PCM=PCM2, \text{ MASSI=MASS2, } \\
& \text{ALPHA1=ALPHA2} \\
\text{POGDERIV}=& G \cos(\text{GAMMA}) \\
\text{P1 DERIV}=& (V \text{ PAREA/MASSI}) ((1-8.862/\text{PCM}) 1.7473-1.7738) \\
\text{P2 DERIV}=& (V \text{ PAREA/MASSI}) \text{ SGN(ALPHA1)} (2.1908 (1-8.862/ \\
& \text{PCM}) -1.5885) \\
\text{P1 VDERIVA}=& (V \text{ PAREA/MASSI}) ((1-8.862/\text{PCM}) 1.7473-1.7738) \\
\text{P1 VDERIVB}=& (\text{RHO PAREA/MASSI}) ((1-8.862/\text{PCM}) 1.7473-1.7738) \\
\text{P2 VDERIVA}=& (V \text{ PAREA/MASSI}) \text{ SGN(ALPHA1)} (2.1908 (1- \\
& 8.862/\text{PCM}) -1.5885) \\
\text{P2 VDERIVB}=& (\text{RHO PAREA/MASSI}) \text{ SGN(ALPHA1)} (2.1908 (1- \\
& 8.862/\text{PCM}) -1.5885) \\
- & \text{ C} \\
\text{IF } I=0, & \text{ PCDERIV1} = (0,0,0,0,0,0,0) \text{,} \\
& 22 \\
- & \text{ C} \\
\text{PCDERIV1} = (0,0,0,0,0,0,0,0), \quad \text{PCDERIV1} = (0,0,0,0,0,0,0,0), \quad \text{PCDERIV1} = (0,0,0,0,0,0,0,0), \quad \text{PCDERIV1} = (0,0,0,0,0,0,0,0) \\
- & \text{ C} \\
\text{PCDERIV2} = (0,0,0,0,0,0,0,0), \quad \text{PCDERIV2} = (0,0,0,0,0,0,0,0) \\
- & \text{ C} \\
\text{PCDERIV2} = (0,0,0,0,0,0,0,0) \\
\text{IF } I=1, & \text{ PCDERIV2} = (0,0,0,0,0,0,0,0) \text{,} \\
& 15
\end{align*}
\]
PCDERIV2 = (0, 0, 0, 0, p1RDERIV),

PCDERIV2 = (0, p1VDERIVA, 0, 0, 0, p1VDERIVB),

PCDERIV2 = (0, 0, 0, 0, p2RDERIV),

PCDERIV2 = (0, p2VDERIVA, 0, 0, 0, p2VDERIVB)

Do this part for I=0, 1, 2

IF I=0, T=TAU1, XE=EX1, XP=XP1, ZE=ZE1, ZP=ZP1, DI=D1
IF I=1, T=TAU2, XE=EX2, XP=XP2, ZE=ZE2, ZP=ZP2, DI=D2
IF I=2, T=TAU3, XE=EX3, XP=XP3, ZE=ZE3, ZP=ZP3, DI=D3

TT1 = T - (3/(4 F1)) T + (1/(6 F1)) T
TT2 = (1/F1) T - (1/(3 F1)) T
TT3 = (-1/(4 F1)) T + (1/(6 F1)) T

DXDERIV = (-SG1 TT2, 0, 0, 0)

DZDERIV = (CG1 TT2, 0, 0, 0)

DXDERIV = (0, -SG2 TT3, 0, 0)

DZDERIV = (0, CG2 TT3, 0, 0)

DXDERIV = (-V1 CG1, 0, -SG1, 0) TT2

DZDERIV = (-V1 SG1, 0, CG1, 0) TT2
DZDERIV = (0, -V2, SG2, 0, CG2) TT3 4 I+3

DXDERIV = (0, -V2, CG2, 0, -SG2) TT3 4 I+3

XPDERIV = (-V0 SG0 TT1, -V1 SG1 TT2, -V2 SG2 TT3, CG0 TT1, CG1 TT2, I, CG2 TT3)

ZPDERIV = (V0 CG0 TT1, V1 CG1 TT2, V2 CG2 TT3, SG0 TT1, SG1 TT2, I, SG2 TT3)

Do this part for K = 0, 1, 2

IF K = 0, DVPDU = DVPSTATE, DZDU = DZPSTATE, DGDU = DGDPSTATE, DPDU =

DZPSTATE, DADU = DADPSTATE, DALDU = 0, DXEDU = 0, DZEDU = 0, DXEDOU = 0,

DZFODU = 0,

DXPDU = (1, 0, 0, 0), DZPDU = (0, 1, 0, 0), DZEROSTATEDU = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)

IF K = 1, DVPDU = 0, DZDU = 0, DGDU = 0, DPDU = 0, DADU = 0, DALDU = 0, DXEDU = (DELXEO, DELXEO, DELXEO, 0, 8, 16, 4, 12, 20)

DXEDU = (1, 0, 0, 0), DZEDU = (0, 1, 0, 0),

DXPDU = 0, DZPDU = 0, DXEDU = DXEDUT, DZEDU = DZEDUT, DZEROSTATEDU = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

IF K = 2, DVPDU = DVPSTATE, DZDU = DZPSTATE, DGDU = DGDPSTATE, DADU =
DATDCONTROL, DPDU=DPDCONTROL, DALDU=(1, 0, 0, 0, 0, 1, 0, 0),

- XPDU=0, ZPDU=0,

* DXEDU=0, DZEDU=0, DXEDUDU=0, DZEDUDU=0, DZEROSTATEDU=0

* DRHODU=PHOMATRIX, DZDU

* DUDUALDUT = (DVDU, DALDU)

* T

DGAMMAMATRIXDU=DVDUDALDUT, GAMMADERIV

* DRVAPDUT = (DRHODU, DVDU, DADU, DPDU, DPDU, DPDU)

0 0 0 4 8

- T

DDMODALODU=DRVAPDUT, DMODALODERIV

* DRVADUT = (DRHODU, DVDU, DADU)

* T

DELTAMATRIXDU=DRVADUT, DELTADERIV

* DGVDUT = (DGDU, DGDU, DVDU, DVDU)

4 4 8 8

* T

DZMATRIXDU=DGVDUT, ZDERIV

* T

DAMATRIXDU=DPDU, ADERIV

* DRGVDUT = (DRHODU, DRHODU, DGDU, DGDU, DVDU, DVDU)

4 8 4 8 4 8

* T

DPCMATRIXDU1=DRGVDUT, PCDERIV1

* T

DPCMATRIXDU2=DRGVDUT, PCDERIV2

* T

DPC1=DPCMATRIXDU1, DPC2=DPCMATRIXDU2

* T

DDXDU=DGDVDUT, DXDERIV
DDZDU=DGVDUT  DZDERIV

GVUDEPVT=(DGDU,DVDU), DXPOT=(DXPDU,DXPDU,DXPDU)

DZPOT=(DZPDU,DZPDU,DZPDU)

DXPDU=GVUDEPVT  XPDERIV+DXPOT

DZPDU=GVUDEPVT  ZPDERIV+DZPOT

Do this subpart for I=0,1,2

DXZEPDUT=(DXEDU,DXPDU,DZEDU,DZPDU)

- C T C C

I I I I

D2DDXPDU=DXZEPDUT  DDERIVX

I

D2DDZPDU=DXZEPDUT  DDERIVZ

I

Continue the part

D2DMINUDXP=D2DMINDXEDXP (DXEDODU,DXEDU)+D2DMINDXP2

- T - T

D2DMINDZPDXP (DZPDU,DZPDU)

- T - T

D2DMINUDZP=D2DMINDXEDZP (DXEDODU,DXEDU)+D2DMINDXPDZP

- T - T

D2DMINZP2 (DZPDU,DZPDU)

D2DMINUDXPT=D2DMINUDXP , D2DMINUDZPT=D2DMINUDZP
Do this subpart for $I=0,1,2$
Do this for $J=0,\ldots,7$

$$ \text{DMATRIXGAMMADU} = DGAMMAMATRIXDU_{J}^{8I+J} $$

Do this for $J=0,\ldots,13$

$$ \text{DMATRIXDELTADU} = DDELTAMATRIXDU_{J}^{14I+J} $$

Do this for $J=0,\ldots,7$

$$ \text{DMATRIXZDU} = DZMATRIXDU_{J}^{8I+J} $$

Do this for $J=0,\ldots,23$

$$ \text{DMATRIXADU} = DAMATRIXDU_{J}^{24I+J} $$

Do this for $J=0,\ldots,11$

$$ \text{IF } J<32, \text{DMATRIXPCDU} = DPC1_{J}^{4I+J} $$

$$ \text{IF } J>31, \text{DMATRIXPCDU} = DPC2_{J}^{4(J-32)+I} $$

Do this for $J=0,\ldots,11$

$$ \text{DDXVECTORDU} = DDXDU_{J}^{12I+J} $$

$$ \text{DDZVECTORDU} = DDZDU_{J}^{12I+J} $$

Continue the subpart

* \[ \text{RTEMP} = (\exp(-Z1/\text{SIGMA}), 0, 0, \exp(-Z2/\text{SIGMA})) (-\text{PHOSUBZERO}/\text{SIGMA}) \]

* \[ \text{DRTEMPDU} = (\exp(-Z1/\text{SIGMA}) \text{ DZDU}, 0, 0, \exp(-Z2/\text{SIGMA}) \text{ DZDU})^{4+I} \]
(RHOSUBZERO/SIGMA)

*   *   *
DMATRIXPDU=DRTEMPDU ZMATRIX (0,0,1,0,0,0,0,1,GAMMAMATRIX)

*   *   *
DMATRIXPDU=DMATRIXPDU*RTEMP DMATRIXZDU (0,0,1,0,0,0,0,1,
*  GAMMAMATRIX)

*   *   *
DMATRIXPDU=DMATRIXPDU*RTEMP ZMATRIX (0,0,0,0,0,0,0,0,
*  DMATRIXGAMMADU)

*   *   *
DAPCPGAMMADU=DMATRIXADU PCMATRIX (1,0,0,0,0,1,0,0,PMATRIX,0,0,
*   *   *
1,0,0,0,0,1,GAMMAMATRIX) + AMATRIX DMATRIXPCDU (1,0,0,0,0,1,0,0,
*   *   *
PMATRIX,0,0,1,0,0,0,0,1,GAMMAMATRIX) + AMATRIX PCMATRIX (0,0,0,
*   *   *
0,0,0,0,0,DMATRIXPDU,0,0,0,0,0,0,0,0,DMATRIXGAMMADU)

*   *   *
DDELTAAPPAPCPGAMMADU=DMATRIXDELTADU (PMATRIX,AAPCPGAMMAMATRIX,0,
*   *   *
0,1,0,0,0,0,0,1) + DELTAMATRIX (DMATRIXPDU,DAPCPGAMMADU,0,0,0,0,
0,0,0,0)

*   *   *
DVALODU=VMATRIX (DDMODALODU,0,0,0,DDELTAAPPAPCPGAMMADU)

*
DVALO1DU=(DVALODU,DVALODU,DVALODU,DVALODU)
      0   1   4   5

*  
DVALO2DU=(DVALODU,DVALODU,DVALODU,DVALODU)
      2   3   6   7

*  *
DDVDALDU=-(V2MATRIX-(1,0,0,1)) (DVALO2DU DVDALMATRIX+
*  
DVALO1DU)
*  
DG1 = (0, 0, 0, 0, 0, 0, 0, 0, DMATRIXGAMMA DU)  
*  
DG2 = (1, 0, 0, 1, DV DALM ATRIX)  
*  
DG3 = (0, 0, 1, 0, 0, 0, 1, GAMMA M ATRIX)  
*  
DG4 = (0, 0, 0, 0, DDV DALDU)  
*  
DGAMMADV DALDU = DG1 DG2 + DG3 DG4  
  
D TRUEGRADIENTDUI = (D2DMIN DUXPT (0, 0, 0, 0, DXVECTOR)  
                  2 I  4 I  
+ D2DMIN DUXPT (0, 0, 0, 0, DZVECTOR)) GAMMA MADV DALT M ATRIX+  
                  4 I  
(DDMIN DXP (0, 0, 0, 0, DDXVECTORDU) + DDMIN DZP (0, 0, 0, 0,  
*  
DDZVECTORDU)) GAMMA MADV DALT M ATRIX+ (DDMIN DXP (0, 0, 0, 0,  
*  
DXVECTOR) + DDMIN DZP (0, 0, 0, 0, DZVECTOR)) DGAMMADV DALDU  
  
End of the subpart. Continue the part.  

*  
TRUEGRADIENTGRADIENT = TRUEGRADIENTDUI T  
                  8 K  
  
Continue the block.  

*  
GRADIENTGRADIENT = TRUEGRADIENTGRADIENT ,  
               0 0  8  
GRADIENTGRADIENT =  
*  
TRUEGRADIENTGRADIENT ,  
               9 16  16  
TRUEGRADIENTGRADIENT ,  
              9 16  16  
  
TRUEDDDXXE = DDMIN DXP (1, 0, 0, 0, DELXEO , DELXEO , DELXEO ) +  
             0 8 16
DDMINDZ (0,1,0,0,DELXEO,DELXEO,DELXEO)

* * GVDIVERIVET=(DGDPSTATE,DVDPSTATE)

* * DXPDU=GVUDERIVT XPDERIV+(1,1,1,0,0,0,0,0,0,0,0,0)

* * DZPDU=GVUDERIVT ZPDERIV+(0,0,0,1,1,1,0,0,0,0,0,0)

- C - C - C TRUEDDXPO=DDMINDXP (1,0,0,0,DXPDU,DXPDU,DXPDU)+

- C - C - C DDMINDZP (0,1,0,0,DZPDU,DZPDU,DZPDU)

- TRUEDDXPO,DDDXEO=TRUEDDXEO

END OF GDMINSUB
B.7 Subroutine DISTFIT

This subroutine is called by GDTSUB or GDMINSUB. It accepts predicted E and P position points at future times and interpolates the miss and miss time. First and second derivatives of these quantities are also computed.

\[
\text{DIMENSION 4, (INTDTECT1, 2), (INTDTECT2, 2), (K1VECT, 2), (K2VECT, 2),}
\]
\[
\text{(TAUMATRIX, 3), (CXE, 3), (CXP, 3), (CZE, 3), (CZP, 3), (DCX, 3), (DCZ, 3),}
\]
\[
\text{(D3STATE4STATE, 3X4), (DDCXDXE, 3X4), (DDCXDXP, 3X4), (DDCZDZE, 3X4),}
\]
\[
\text{(DDCZDXP, 3X4), (DPDDCX, 3), (DDENOMDDCX, 3), (DDENOMDDCZ, 3),}
\]
\[
\text{(DENUMDDCX, 3), (DEQNUMDDCZ, 3), (DEQNUMDDCX, 3), (DEQNUMDDCZ, 3),}
\]
\[
\text{(DFNUMDDCX, 3), (DFQNUMDDCZ, 3), (DFQNUMDDCX, 3), (DFQNUMDDCZ, 3),}
\]
\[
\text{(DRNUMDDCX, 3), (DRQNUMDDCZ, 3), (DRQNUMDDCX, 3), (DRQNUMDDCZ, 3),}
\]
\[
\text{(D2PDDCX2, 3), (DPDDCZ, 3), (D2PDDCZ2, 3), (D2QDDCZ2, 3), (D2QDDCZ2, 3),}
\]
\[
\text{(D2QDDCZ2, 3), (D2QDDCZ2, 3), (DRDDCZ, 3), (DRDDCZ, 3), (D2RDDCZ, 3),}
\]
\[
\text{(D2RDDCZ, 3), (D2RDDCZ, 3), (D2RDDCZ, 3), (D2RDDCZ, 3), (D2RDDCZ, 3),}
\]
\[
\text{(D2RDDCZ, 3), (DDCXDSTATE1, 3X4), (DDCXDSTATE2, 3X4),}
\]
\[
\text{(DDCZDSTATE1, 3X4), (DDCZDSTATE2, 3X4),}
\]
\[
\text{(DPOLYTMINVECTORDSTATE1, 3X4), (DPOLYTMINVECTORDSTATE2, 3X4)}
\]

In this block the particular set of points is chosen for interpolation, INTFL AG=1 or 2 means the first three or second three are to be used.

\[
x=0, x^1=0, x^2=0, x^3=0, y=0, y^1=0, y^2=0, y^3=0
\]

DECIDE1=0, DECIDE2=0

* INTDTECT1=(XE2-XE0,- (XP2-XP0), ZE2-ZE0,- (ZP2-ZP0))
INTDETECT2=(XE3-XE1, (XP3-XP1), ZE3-ZE1, (ZP3-ZP1))

IF DET(INTDETECT1) NZ, INTFLAG=1, K1VECT=INTDETECT1
(XPO-XEO, ZPO-ZEO), DECIDE1=1

IF DET(INTDETECT2) NZ, INTFLAG=2, K2VECT=INTDETECT2
(XP1-XE1, ZP1-ZE1), DECIDE2=1

IF DECIDE1=0 OR DECIDE2=0, GO TO ONEZERO

IF ABVAL(K1VECT-(1/2,1/2)) < ABVAL(K2VECT-(1/2,1/2)),
INTFLAG=1, OTHERWISE INTFLAG=2

location ONEZERO

IF INTFLAG=1, EXO=XE0, EZ0=ZE0, EX1=XE1, EZ1=ZE1, EX2=XE2, EZ2=ZE2,
PX0=XP0, PZ0=ZP0, PX1=XP1, PZ1=ZP1, PX2=XP2, PZ2=ZP2, TAU1=T1-T0, TAU2 =T2-T0

IF INTFLAG=2, EX0=XE1, EZ0=ZE1, EX1=XE2, EZ1=ZE2, EX2=XE3, EZ2=ZE3,
PX0=XP1, PZ0=ZP1, PX1=XP2, PZ1=ZP2, PX2=XP3, PZ2=ZP3, TAU1=T2-T1,
TAU2=T3-T1

A and B are now computed from the cubic fit to the derivative of P-E
distance function.

TAUMATRIX=(1, 0, 0, 1, TAU1, TAU1, 1, TAU2, TAU2)

CXE=TAUMATRIX (EXO, EX1, EX2)

CXP=TAUMATRIX (PX0, PX1, PX2)

CZE=TAUMATRIX (EZ0, EZ1, EZ2)

CZP=TAUMATRIX (PZ0, PZ1, PZ2)
DCX =CXE -CXP
   0   0   0

DCX =CXE -CXP
   1   1   1

DCX =CXE -CXP
   2   2   2

DCZ =CZE -CZP
   0   0   0

DCZ =CZE -CZP
   1   1   1

DCZ =CZE -CZP
   2   2   2

IF DCX2=0 AND DCZ2=0, DCX2=.001, DCZ2=.001

2
QNUMERATOR=DCX +DCZ +2 DCX DCX +2 DCZ DCZ
1   1   0   2   0   2

RNUMERATOR=DCX DCX +DCZ DCZ
0   1   0   1

PNUMERATOR=DCX DCX +DCZ DCZ + 2 DCX DCX +2 DCZ DCZ
1   2   1   2   1   2   1   2

2
DENOMINATOR=2( DCX +DCZ )
2   2

P=PNUMERATOR/DENOMINATOR
Q=QNUMERATOR/DENOMINATOR
R=RNUMERATOR/DENOMINATOR

2
A=(3 Q-P )/3

3
B=(2 P -9 P Q+27 R)/27

This block uses A and B to compute the desired root of the cubic. The root can fall into one of three categories: ROOTFLAG=1, 2 or 3.

2
IF A<0,IF B/4-((ABS(A)) )/27<0,GO TO NEGZERO
ABSQT=SQR(T(3/B +A/27))

IF -B/2+ABSQT<0, CAPA=-(ABS(-B/2+ABSQT))

OTHERWISE CAPA=(-B/2+ABSQT)

IF -B/2-ABSQT<0, CAPB=-(ABS(-B/2-ABSQT))

OTHERWISE CAPB=(-B/2-ABSQT)

X=CAPA+CAPB, Y=X-(P/3)

ROOTFLAG=+1

location NEGZERO

IF B/4+A/27=0, CAPA=(-B/2), CAPB=CAPA,

X1=CAPA+CAPB, X2=-(CAPA+CAPB)/2, X3=X2, ROOTFLAG=2

IF B/4+A/27<0, PHI=ARCCOS((-B/2)/SQRT(-A/27)),

X1=2 SQR(T(-A/3) COS(PHI/3), X2=2 SQR(T(-A/3) COS(PHI/3+

120 DEGTORAD), X3=2 SQR(T(-A/3) COS(PHI/3+240 DEGTORAD),

ROOTFLAG=3

Y1=X1-(P/3), Y2=X2-(P/3), Y3=X3-(P/3)

In this block the minimum distance is computed for the particular ROOTFLAG value.

IF ROOTFLAG=1, POLYTMIN=Y, DELTAX=DCX+DCX POLYTMIN+

DCX POLYTMIN, DELTAZ=DCZ+DCZ POLYTMIN+DCZ POLYTMIN,

POLYDMIN=SQR(T(DELTAX+DELTAZ)), GO TO ANS
IF ROOTFLAG=3, GO TO RF3

\[
\begin{align*}
\text{DELTAX1} &= \text{DCX} + \text{DCX} Y1 + \text{DCX} Y1, \\
\text{DELTAZ1} &= \text{DCZ} + \text{DCZ} Y1, \\
\text{DELTAX2} &= \text{DCX} + \text{DCX} Y2 + \text{DCX} Y2, \\
\text{DELTAZ2} &= \text{DCZ} + \text{DCZ} Y2, \\
\text{DIST1} &= \text{SQRT} (\text{DELTAX1}^2 + \text{DELTAZ1}^2), \\
\text{DIST2} &= \text{SQRT} (\text{DELTAX2}^2 + \text{DELTAZ2}^2), \\
\text{POLYDMIN} &= 10000000
\end{align*}
\]

IF DIST1<POLYDMIN, POLYDMIN=DIST1, POLYTMIN=Y1

IF DIST2<POLYDMIN, POLYDMIN=DIST2, POLYTMIN=Y2

GO TO ANS

location RF3

\[
\begin{align*}
\text{DELTAX1} &= \text{DCX} + \text{DCX} Y1 + \text{DCX} Y1, \\
\text{DELTAZ1} &= \text{DCZ} + \text{DCZ} Y1, \\
\text{DELTAX2} &= \text{DCX} + \text{DCX} Y2 + \text{DCX} Y2, \\
\text{DELTAZ2} &= \text{DCZ} + \text{DCZ} Y2, \\
\text{DELTAX3} &= \text{DCX} + \text{DCX} Y3 + \text{DCX} Y3, \\
\text{DELTAZ3} &= \text{DCZ} + \text{DCZ} Y3, \\
\text{DIST1} &= \text{SQRT} (\text{DELTAX1}^2 + \text{DELTAZ1}^2), \\
\text{DIST2} &= \text{SQRT} (\text{DELTAX2}^2 + \text{DELTAZ2}^2), \\
\text{DIST3} &= \text{SQRT} (\text{DELTAX3}^2 + \text{DELTAZ3}^2), \\
\text{POLYDMIN} &= 10000000
\end{align*}
\]

IF DIST1<POLYDMIN, POLYDMIN=DIST1, POLYTMIN=Y1

IF DIST2<POLYDMIN, POLYDMIN=DIST2, POLYTMIN=Y2

IF DIST3<POLYDMIN, POLYDMIN=DIST3, POLYTMIN=Y3

location ANS

This block provides 1st and 2nd derivative computations common to all values of ROOTFLAG.
IF INTFLAG=2, TMIN=POLYMIN+(T1-T0), OTHERWISE TMIN=POLYMIN

* IF INTFLAG=1, D3STATED4STATE=(1,0,0,0,0,1,0,0,0,0,1,0), OTHERWISE

* D3STATED4STATE=(0,1,0,0,0,0,1,0,0,0,0,1)

* -1 *
DDCXDXE=TAUMATRIX D3STATED4STATE

* -1 *
DDCXDXP=-TAUMATRIX D3STATED4STATE

* -1 *
DDCZDZE=TAUMATRIX D3STATED4STATE

* -1 *
DDCZDZP=-TAUMATRIX D3STATED4STATE

- 
DPDDCX=(0, 3 DCX /DENOMINATOR), (3 DCX DENOMINATOR-PNUMERATOR
2 2 1
4 DCX )/DENOMINATOR
2

- 
DDENOMDDCX=(0, 0, 4 DCX )
2

- 
DDENOMDDCZ=(0, 0, 4 DCZ )
2

- 
DPNUMDDCX=(0, 3 DCX , 3 DCX )
2 1

- 
DPNUMDDCZ=(0, 3 DCZ , 3 DCZ )
2 1

- 
DQNUMDDCX=(2 DCX , 2 DCX , 2 DCX )
2 1 0

- 
DQNUMDDCZ=(2 DCZ , 2 DCZ , 2 DCZ )
2 1 0

- 
DRNUMDDCX=(DCX , DCX , 0)
1 0
\[
- \text{DPNUMDDCZ} = (\text{DCZ},\text{DCZ},0) \\
\]

\[
- \text{D2PDDCZ2} = 0 \\
\]

\[
- \text{D2PDDCZ2} = (3/\text{DENOMINATOR}) (0,0,1) - 3 (\text{DCZ} / \text{DENOMINATOR}) \\
\]

\[
- \text{DDENOMDDCZ} \\
\]

\[
- \text{D2PDDCZ2} = (\text{DENOMINATOR}) (3 \text{ DENOMINATOR} (0,1,0) + 3 \text{ DCZ} \\
\]

\[
- \text{DDENOMDDCZ-4 PNUMERATOR} (0,0,1) - 4 \text{ DCX DPNUMDDCZ} \\
\]

\[
- (3 \text{ DCX} \text{ DENOMINATOR-PNUMERATOR} 4 \text{ DCX}) 2 \text{ DENOMINATOR} \\
\]

\[
- \text{DDENOMDDCZ}/\text{DENOMINATOR} \\
\]

\[
- \text{DPDDCZ} = (0,3 \text{ DCZ} / \text{DENOMINATOR}) , (3 \text{ DCZ} \text{ DENOMINATOR-PNUMERATOR} \\
\]

\[
4 \text{ DCZ} / \text{DENOMINATOR} \\
\]

\[
- \text{D2PDDCZ2} = (0,0,0) \\
\]

\[
- \text{D2PDDCZ2} = (3/\text{DENOMINATOR}) (0,0,1) - 3 (\text{DCZ} / \text{DENOMINATOR}) \\
\]

\[
- \text{DDENCMDDCZ} \\
\]

\[
- \text{D2PDDCZ2} = (\text{DENOMINATOR}) (3 \text{ DENOMINATOR} (0,1,0) + 3 \text{ DCZ} \\
\]

\[
- \text{DDENOMDDCZ-4 PNUMERATOR} (0,0,1) - 4 \text{ DCX DPNUMDDCZ} \\
\]
\[- \frac{4}{\text{DENOMINATOR}} - \frac{(3 \text{ DCZ})}{\text{DENOMINATOR}} - \frac{2}{\text{DDENOMDDCX}} \]

\[- \frac{2}{\text{DDENOMDDCX}} = 0 \]

\[- \frac{2}{\text{DDENOMDDCX}} = \frac{(-3 \text{ DCZ})}{\text{DENOMINATOR}} \]

\[- \frac{2}{\text{DDENOMDDCX}} = \frac{(\text{DENOMINATOR}) (3 \text{ DCZ})}{\text{DENOMINATOR}} \]

\[- \frac{2}{\text{DDENOMDDCX}} = (3 \text{ DCZ}) (\text{DENOMINATOR}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = \frac{(2 \text{ DCX})}{\text{DENOMINATOR}} \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]

\[- \frac{2}{\text{DDENOMDDCX}} = (2 \text{ DCX}) \]
292

- \((2 \text{ DCX} \text{ DENOMINATOR\text{-QNUMERATOR}} +4 \text{ DCX})\) \text{ DENOMINATOR}

- \(\text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCZ}=(2 \text{ DCZ} / (\text{DENOMINATOR}), 2 \text{ DCZ} / (\text{DENOMINATOR}), (2 \text{ DCZ}

- \text{DENOMINATOR\text{-QNUMERATOR}} +4 \text{ DCZ})/\text{DENOMINATOR})

- \(\text{DQDDCZ2}=(\text{DENOMINATOR} 2 (0,0,1)-2 \text{ DCZ} \text{ DDENOMDDCZ})/\)

- \(\text{DDENOMDDCZ-4 QNUMERATOR} (0,0,1)-4 \text{ DCZ} \text{ DQNUMDDCZ})

- \(\text{DENOMINATOR-4 DCZ}) \text{ DENOMINATOR}

- \(\text{DDENOMDDCX)}/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ}

- \text{DDENOMDDCX})/\text{DENOMINATOR}

- \(\text{DQDDCXXDCC}=-2 \text{ DCZ} \text{ DDENOMDDCXX} / (\text{DENOMINATOR})

- \(\text{DQDDCXXDCC}=(\text{DENOMINATOR} 2 \text{ DCZ} \text{ DDENOMDDCXX-4}

- \(\text{DCZ DQNUMDDCXX})-(2 \text{ DCZ} \text{ DENOMINATOR-4 DCZ
QNUMERATOR) 2 DENOMINATOR DDENOMDDCX) / DENOMINATOR
-
DRDDCX = (DCX / (DENOMINATOR), DCX / (DENOMINATOR), -(RNUMERATOR / 1

2
DENOMINATOR) 4 DCX ) 2

-
D2RDDCX2 = (DENOMINATOR (0,1,0) - DCX 2 DDENOMDDCX) / DENOMINATOR 0

-
D2RDDCX2 = (DENOMINATOR (1,0,0) - DCX 2 DDENOMDDCX) / DENOMINATOR 3

-
D2RDDCX2 = -(DENOMINATOR (4 DCX DNUMDDCX + 4 RNUMERATOR 6

2
(0,0,1)) 4 DCX RNUMERATOR 2 DENOMINATOR DDENOMDDCX) / 2

4
DENOMINATOR
-
DRDDCZ = (DCZ / (DENOMINATOR), DCZ / (DENOMINATOR), -(RNUMERATOR / 1

2
DENOMINATOR) 4 DCZ ) 2

-
D2RDDCZ2 = (DENOMINATOR (0,1,0) - DCZ 2 DDENOMDDCZ) / DENOMINATOR 0

-
D2RDDCZ2 = (DENOMINATOR (1,0,0) - DCZ 2 DDENOMDDCZ) / DENOMINATOR 3

-
D2RDDCZ2 = -(DENOMINATOR (4 RNUMERATOR (0,0,1) + 4 DCZ DRNUMDDCZ 6

2
\}) -(4 DCZ RNUMERATOR) 2 DENOMINATOR DDENOMDDCZ) / DENOMINATOR 2

-
D2RDDCXDDCZ = -(DCZ / (DENOMINATOR)) DDENOMDDCX 0

2
D2RDCXDDCZ = -(DCZ / (DENOMINATOR)) DDENOMDDCX

D2RDCXDDCZ = -4 DCZ (DENOMINATOR DPNUMDDCX - 2 RNUMERATOR

DENOMINATOR DDENOMDDCX) / DENOMINATOR

* * * T * *
D2PDXE2 = DDCXDXE

* * * T * *
D2PDXX2 = DDCXDXE D2PDDCX2 DDCXDXE

* * * T * *
D2PDXXP = DDCXDXP D2PDDCX2 DDCXDXE

* * * T * * T *
D2PDXXEP = DDCXDXP D2PDDCXDCZ DDCXDZE

* * * T * *
D2PDXXP2 = DDCXDXP D2PDDCX2 DDCXDXP

* * * T * * T *
D2PDZPDXP = DDCXDXP D2PDDCXDCZ DDCXDZP

* * * T * *
D2PDZEP = DDCXDXP D2PDDCXDCZ DDCXDZE

* * * T * *
D2PDZEP2 = DDCXDXP D2PDDCX2 DDCXDXP

* * * T * *
D2PDZPDZP = DDCXDP D2PDDCXDCZ DDCXDZP

* * * T * *
D2PDZPDZP2 = DDCXDP D2PDDCX2 DDCXDP

* * * T * *
DQDXE = DQDPCX DDCXDXE
* * T * T *
D2RDZPDXP=DCXDZXP D2RDCXDDCZ DDCZDP

- - *
DRDZE=DRDCCZ DDCZDZE

* * T * *
D2RDZE2=DCZDZE D2RDCCZ2 DDCZDZE

- - *
DRDZP=DRDCCZ DDCZDZE

* * T * *
D3PDXEDZP=DDCZDZP D2RDCXDDCZ DDCXDXE

* * T * *
D2RDZEDZP=DDCZDZP D2RDCZ2 DDCZDZE

* * T *
D2RDZPDZP=D2RDZPDXP

* * T *
D2RDZP2=DDCZDZP D2RDCCZ2 DDCZDZP

DADP=-(2/3)P, DADQ=1

D2ADP2=-(2/3)

2
DBDP=(1/27) (6 P -9 Q), DBDQ=-P/3, DBDR=1

D2BDP2=(12/27) P, D2BDPDQ=-1/3

- - *
DADXE=DADP DPDXE+DADQ DQDXE

* - *
D2ADXE2=-(2/3) DPDXE DPDXE+DADP D2PDXE2+D2QDXE2

- - *
DBDXE=DBDP DPDXE*DBQ DQDXE+DBDR DRDXE

* - *
D2BDXE2=(12/27) P DPDXE DPDXE-(1/3)DPDXE DQDXE

* * *
+DBDP D2DPDXE2-(1/3)DQDXE DPDXE+DBDQ D2QDXE2+DBDR D2RDXE2

- - *
DADXP=DADP DPDXP+DADQ DQDXP

* - *
D2ADXEDXP=D2ADP2 (DPDXP DPDXE)+DADP D2PDXEDXP+DADQ D2QDXEDXP

* - *
D2ADZEDXP=D2ADP2 (DPDXP DPDXE)+DADP D2PDZEDXP+DADQ D2QDZEDXP
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D2ADXP2 = (2/3)</td>
<td>DPDXZ</td>
<td>DPDXZ + DADP</td>
<td>D2PDXP2 + DADQ</td>
<td>D2QDXP2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2ADZPD XP = D2ADP2</td>
<td>(DPDXZ</td>
<td>DPDZP) + DADP</td>
<td>D2PDZPD XP + DADQ</td>
<td>D2QDZPD XP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This block computes derivatives needed when ROOTFLAG=1 and 2 only.

\[ DCAPADA = \frac{2}{(3 \ CAPA)} \left( \frac{1}{(2 \ ABSQRT)} \right) \left( \frac{1}{9} \right) A \]

\[ DCAPADB = \frac{2}{(3 \ CAPA)} \left( \frac{-1/2}{(2 \ ABSQRT)} \right) \left( \frac{1/2}{9} \right) B \]

\[ DCAPEDA = \frac{2}{(3 \ CAPB)} \left( \frac{1}{(2 \ ABSQRT)} \right) \left( \frac{1}{9} \right) A \]

\[ DCAPBDB = \frac{2}{(3 \ CAPB)} \left( \frac{-1/2}{(2 \ ABSQRT)} \right) \left( \frac{1}{9} \right) B \]

Repeat this part for \( L=1, 2, 3, 4 \)

\[ IF \ COEFSINDEX=1, ADFER=DADXE, BDFER=DBDXE \]

\[ IF \ COEFSINDEX=2, ADFER=DADX, BDFER=DBDYP \]
IF COEFINDEX=3, ADER=DADZE, BDER=DBDZE

IF COEFINDEX=4, ADER=DADZP, BDER=DBDZP

CAd = (1/3) CAPA \left( (1/2) (-1 + (1/2) ABSQRT B) BDER + (A / 18) \right)

CAd = (1/3) CAPB \left( (-1/2) (1 + (1/2) B ABSQRT ) BDER - (A / 18) \right)

IF ROOTFLAG=1, DXDSTATE=CAPADER+CAPBDER

IF ROOTFLAG=2 AND POLYTMN=Y1, DXDSTATE=CAPADER+CAPBDER

IF ROOTFLAG=2 AND POLYTMN=Y2, DXDSTATE=-(CAPADER+CAPBDER)/2

IF COEFINDEX=1, DCAPADXE=CAPADER, DCAPBDXE=CAPBDER, DXDXE=

IF COEFINDEX=2, DCAPADXP=CAPADER, DCAPBDXP=CAPBDER, DXDXP=

IF COEFINDEX=3, DCAPADZ=CAPADER, DCAPBDZ=CAPBDER, DXDZ=

IF COEFINDEX=4, DCAPADZP=CAPADER, DCAPBDZP=CAPBDER, DXDZP=

DXDSTATE

Continue the block.
\[
D2\text{CAPADA}_2 = \frac{-2}{9} \text{ CAPA } \left( \frac{1}{2} \text{ ABSQRT } \frac{1}{9} \text{ A } \right) \\
\quad -2 \quad -3 \quad 2 \quad 2 \\
+ \frac{1}{3} \text{ CAPA } \left( \frac{-1}{4} \text{ ABSQRT } \left( \frac{1}{9} \text{ A } \right) \right) \\
\quad -2 \quad -1 \\
+ \frac{1}{3} \text{ CAPA } \left( \frac{1}{2} \text{ ABSQRT } \frac{2}{9} \text{ A } \right) \\
\quad -5 \quad -1 \quad 2 \quad -1 \quad 2 \\
D2\text{CAPADB}_2 = \frac{-2}{9} \text{ CAPA } \left( \frac{-1}{2} + \left( \frac{B}{4} \right) \text{ ABSQRT } \right) + \\
\quad -2 \quad -1 \quad 2 \quad -3 \\
\quad \left( \frac{1}{3} \text{ CAPA } \left( \frac{1}{4} \text{ ABSQRT } - \left( \frac{B}{16} \right) \text{ ABSQRT } \right) \right) \\
\quad -5 \quad -1 \quad 2 \\
D2\text{CAPADADB} = \left( \frac{-1}{81} \right) \text{ CAPA } \left( \text{ ABSQRT } \frac{A}{\left( \frac{1}{2} \right) + \left( \frac{1}{2} \right) \text{ ABSQRT } \left( \frac{B}{2} \right)} \right) + \\
\quad -2 \quad -3 \quad 2 \quad 2 \\
\quad \left( \frac{-1}{216} \right) \text{ CAPA } \left( \text{ ABSQRT } \frac{B}{A} \right) \\
\quad -5 \quad 2 \quad -1 \quad 2 \\
D2\text{CAPBDA}_2 = \left( \frac{-2}{9} \right) \text{ CAPB } \left( \left( \frac{A}{18} \right) \text{ ABSQRT } \right) + \\
\quad -2 \quad -3 \quad 2 \\
\quad \left( \frac{1}{3} \right) \text{ CAPB } \left( \frac{1}{36} \text{ ABSQRT } \frac{A}{\left( \frac{1}{9} \right) \text{ A } \right) \right) \\
\quad -2 \quad -1 \\
\quad \left( \frac{-1}{27} \right) \text{ CAPB } \left( \text{ ABSQRT } \frac{A}{A} \right) \\
\quad -5 \quad -1 \quad 2 \\
D2\text{CAPBDB}_2 = \left( \frac{-2}{9} \right) \text{ CAPB } \left( \left( \frac{-1}{2} - \left( \frac{B}{4} \right) \text{ ABSQRT } \right) + \\
\quad -2 \quad -1 \quad 2 \quad -3 \\
\quad \left( \frac{1}{3} \right) \text{ CAPB } \left( \left( \frac{-1}{4} \text{ ABSQRT } + \left( \frac{B}{16} \right) \text{ ABSQRT } \right) \right) \\
\quad -5 \quad -1 \quad 2 \\
D2\text{CAPBDADB} = \left( \frac{1}{81} \right) \text{ CAPB } \left( \left( \frac{-1}{2} \right) \text{ ABSQRT } \left( \frac{E}{4} \right) \right) \text{ ABSQRT } \frac{A}{\left( \frac{1}{2} \right) + \left( \frac{1}{2} \right) \text{ ABSQRT } \left( \frac{B}{2} \right) \right) + \\
\quad -2 \quad -3 \quad 2 \\
\quad \left( \frac{1}{216} \right) \text{ CAPB } \left( \text{ ABSQRT } \frac{B}{A} \right) \\
\quad + \text{D2CAPADXE}_2 = \text{D2CAPADA}_2 \text{ DADXE DADXE+D2CAPADADB DADXE DBDXE} \\
\quad + \text{DACAPADA D2ADXE}_2+\text{D2CAPADB}_2 \text{ DBDXE DBDXE+D2CAPADADB DBDXE DADXE} \\
\quad + \text{D2CAPBDXE}_2 = \text{D2CAPBDA}_2 \text{ DADXE DADXE+D2CAPBDADB DADXE DBDXE} \\
\quad + \text{DACAPBDA D2DADXE}_2+\text{D2CAPBDDB}_2 \text{ DBDXE DBDXE+D2CAPBDADB DBDXE DADXE} \\
\quad *
D2CAPBDXP2=D2CAPBDA2 DADXP DADXP+D2CAPBDADB DADXP DBDXP

+DCAPBDA D2ADXP2+D2CAPBDB2 DBDXP DBDXP+D2CAPBDADB DBDXP DADXP

+DCAPDB D2BDXP2

D2CAPBDZPDXP=D2CAPBDA2 DADXP DADZP+D2CAPBDADB DADXP DBDZP

+DCAPBDA D2ADZPDXP+D2CAPBDB2 DBDXP DBDZP+D2CAPBDADB DBDXP DADZP

+DCAPDB D2BDZPDXP

D2CAPADZ2=E2 D2CAPADA2 DADZE DADZE+D2CAPADADB DADZE DBDZE

+DCAPADA D2ADZ2+D2CAPAD2 DBDZE DBDZE+D2CAPADADB DBDZE DADZE

+DCAPADB D2BDZ2

D2CAPBDZ2=D2CAPBDA2 DADZE DADZE+D2CAPBDADB DADZE DBDZE

+DCAPBDA D2ADZ2+D2CAPBDB2 DBDZE DBDZE+D2CAPBDADB DBDZE DADZE

+DCAPDB D2BDZ2

D2CAPADXEDZP=D2CAPADA2 DADZP DADXE+D2CAPADADB DADZP DBDXE

+DCAPADA D2ADXEDZP+D2CAPADB2 DBDXZ DBDXE+D2CAPADADB DBDXZ DADX

+DCAPADB D2BDXEDZP

D2CAPADZEDZP=D2CAPADA2 DADZP DADXE+D2CAPADADB DADZP DBDZE

+DCAPADA D2ADZEDZP+D2CAPADB2 DBDZP DBDZE+D2CAPADADB DBDZP DADZE

+DCAPADB D2BDZEDZP

D2CAPADXPDPZP=D2CAPADZPDXP
D2CAPADZP2=D2CAPADA2 DADZP DADZP+D2CAPADADB DADZP DEDZP

* +DCAPADA D2ADZP2+D2CAPADB2 DBDZP DBDZP+D2CAPADADB DBDZP DADZP

* +DCAPADB D2BDZP2

* D2CAPBDXEDZP=D2CAPBDA2 DADZP DADXE+D2CAPBDADB DADZP DBDXE

* +DCAPBDA D2ADXEDZP+D2CAPBDB2 DBDZP DBDXE+D2CAPBDADB DBDZP DADXE

* +DCAPBDB D2BDXEDZP

* D2CAPBDZEDZP=D2CAPBDA2 DADZP DADZE+D2CAPBDADB DADZP DBDZE

* +DCAPBDA D2ADZEDZP+D2CAPBDB2 DBDZP DBDZE+D2CAPBDADB DBDZP DADZE

* +DCAPBDB D2BDZEDZP

* * *
D2CAPBDXPDZP=D2CAPBDZPDXP

* D2CAPBDZP2=D2CAPBDA2 DADZP DADZP+D2CAPBDADB DADZP DBDZP

* +DCAPBDA D2ADZP2+D2CAPBDB2 DBDZP DBDZP+D2CAPBDADB DBDZP DADZP

* +DCAPBDB D2BDZP2

* * * * * DRVDEXE2=D2CAPADX2+D2CAPBDXE2, DRVDEXEDXP=D2CAPADXEDXP

* * * * * +D2CAPBDXEDXP, DRVDEXEDXP=D2CAPADZEDXP+D2CAPBDZEDXP, DRVDEXP2=

* D2CAPADXP2+

* * * * * D2CAPBDXP2, DRVDEXPDXP=D2CAPADZPDXP+D2CAPBDZPDXP, DRVDEXZ2=

* * * * D2CAPADZ2+D2CAPBDZ2, DRVDEXZP=

* * * * * D2CAPADXEDZP+D2CAPBDXEDZP, DRVDEXEDZP=D2CAPADZEDZP+D2CAPBDZEDZP,
* * * T * * *

DRVDZP(DP) = DRVDZP(DX) + DRVDZP(DY) + DRVDZP(DZ)

IF ROOTFLAG = 1, DELTAX = DCX + DCX Y + DCX Y +

2

DCZ Y, D2XDXE2 = DRVDXEP, D2XDXEP = DRVDXEP, D2XDXEDXP = DRVDZEDXP

2

, D2XDP2 = DRVDZP2, D2XDPDP = DRVDZPDP, D2XDE2 = DRVDZ2,

* * * T

D2XDXEDZP = DRVDXEDZP, D2XDEDP = DRVDZDP, D2XDPDP = D2XDPDP

* *

D2XDP2 = DRVDZP2

IF ROOTFLAG = 2 AND POLYMIN = Y1, DELTAX = DELTAX1, DELTAZ = DELTAZ1

IF ROOTFLAG = 2 AND POLYMIN = Y2, DELTAX = DELTAX2, DELTAZ = DELTAZ2

IF ROOTFLAG = 2, D2XDXE2 = -DRVDXEP/2, D2XDXEP = -DRVDZDP/2,

* * * T

D2XDXEDXP = -DRVDZEDXP/2, D2XDP2 = -DRVDZPDP/2, D2XDPDP = -DRVDZPDP/

2, D2XDE2 = -DRVDZ2/2, D2XDEDP = -DRVDZDP/2, D2XDPDP =

* ** T

-DRVDZDP/2, D2XDPDP = D2XDPDP, D2XDP2 = -DRVDZP2/2

This block computes derivatives needed when ROOTFLAG = 3

IF POLYMIN = Y1, ANGLEARG = PHI/3, DELTAX = DELTAX1, DELTAZ = DELTAZ1

IF POLYMIN = Y2, ANGLEARG = PHI/3 + 120 DEGTOPAD, DELTAX = DELTAX2,

DELTAX = DELTAX2

IF POLYMIN = Y3, ANGLEARG = PHI/3 + 240 DEGTOPAD, DELTAX = DELTAX3,

DELTAX = DELTAX3

2

DPHIDA = (A / 36) (1/SIN(PHI)) ((ABS(A)) / 27)
DPHIDB = \frac{1}{2} \sin(\phi) \sqrt{-(A/27)}

D2PHIDA2 = \left(\frac{A B}{18}\right) \left(\frac{\text{ABS}(A)}{27}\right) + \left(\frac{1}{216}\right) A B

D2PHIDDB2 = -\left(\frac{1}{\tan(\phi)}\right) DPHIDB

D2PHIDADB = \left(\frac{A}{36}\right) \left(\frac{\text{ABS}(A)}{27}\right) - \frac{3}{2} \cos(\phi) DPHIDA DPHIDB

// SIN(\phi)

Do this part for I=1, ..., 9

IF I=1, DADSTATE1=DADXE, DADSTATE2=DADXE, DBDSTATE1=DBDXE,

DBDSTATE2=DBDXE, D2ADSTATE2=D2ADXE2, D2BDSTATE2=D2BDXE2

IF I=2, DADSTATE1=DADXP, DADSTATE2=DADXE, DBDSTATE1=DBDXP,

DBDSTATE2=DBDXE, D2ADSTATE2=D2ADXE2, D2BDSTATE2=D2BDXE2

IF I=3, DADSTATE1=DADXP, DADSTATE2=DADZE, DBDSTATE1=DBDXP,

DBDSTATE2=DBDZE, D2ADSTATE2=D2ADZEDXP, D2BDSTATE2=D2BDZEDXP

IF I=4, DADSTATE1=DADXP, DADSTATE2=DADXP, DBDSTATE1=DBDXP,

DBDSTATE2=DBDXP, D2ADSTATE2=D2ADXP2, D2BDSTATE2=D2BDXP2

IF I=5, DADSTATE1=DADXP, DADSTATE2=DADZP, DBDSTATE1=DBDXP,

DBDSTATE2=DBDZP, D2ADSTATE2=D2ADZPDXP, D2BDSTATE2=D2BDZPDXP

IF I=6, DADSTATE1=DADZE, DADSTATE2=DADZE, DBDSTATE1=DBDZE,
DBDSTATE2 = DBDZE, D2ADSTATE2 = D2ADZE2, D2RDSTATE2 = D2BDZE2

- - * * * *

IF I = 7, DADSTATE1 = DADZP, DADSTATE2 = DADXE, DBDSTATE1 = DBDZP,

- * * * * * *

DBDSTATE2 = DBDZE, D2ADSTATE2 = D2ADXEDZP, D2BDSTATE2 = D2BDXEDZP

- - * * * *

IF I = 8, DADSTATE1 = DADZP, DADSTATE2 = DADZE, DBDSTATE1 = DBDZP,

- * * * * * *

DBDSTATE2 = DBDZE, D2ADSTATE2 = D2ADZEDZP, D2BDSTATE2 = D2BDZEDZP

- - * * * *

IF I = 9, DADSTATE1 = DADZP, DADSTATE2 = DADZP, DBDSTATE1 = DBDZP,

- * * * * * *

DBDSTATE2 = DBDZP, D2ADSTATE2 = D2ADZP2, D2BDSTATE2 = D2BDZP2

- 

DPHIDSTATE1 = DPHIDA DADSTATE1 + DPHIDB DBDSTATE1

- 

DPHIDSTATE2 = DPHIDA DADSTATE2 + DPHIDB DBDSTATE2

- 

D2PHIDSTATE1 = D2PHIDA2 DADSTATE2 + D2PHIDADB DBDSTATE2

- 

D2PHIDSTATE2 = D2PHIDADB DADSTATE2 + D2PHIDB2 DBDSTATE2

* * * *

D2PHIDSTATE2 = DADSTATE1 D2PHIDSTATE1 + DPHIDA D2ADSTATE2 +

- 

DBDSTATE1 D2PHIDSTATE2 = + DPHIDB D2BDSTATE2

- 

D2XSTATE = (1/(3 SQRT(-A/3))) COS(ANGLEARG) DADSTATE1 -

(2/3) SQRT(-A/3) SIN(ANGLEARG) DPHIDSTATE1

* 

D2XSTATE2 = (1/(18 (-A/3))) COS(ANGLEARG) DADSTATE1 -

DADSTATE2

+ (1/(9 (-A/3))) SIN(ANGLEARG) DADSTATE1 DPHIDSTATE2 -

1/2 * (1/(3 (-A/3))) COS(ANGLEARG) D2ADSTATE2 + (1/9) (-A/3)
\[ \sin(\text{ANGLEARG}) \ \text{DPHIDSTATE1} \ \text{DADSTATE2} - (2/9) (-A/3) \ \cos(\text{ANGLEARG}) \ \text{DPHIDSTATE1} \ \text{DPHIDSTATE2} - (2/3) (-A/3) \ \sin(\text{ANGLEARG}) \ \text{D2PHIDSTATE2} \]

* 

IF \( I=1 \), \( DXDXE=DXDSTATE, D2DXE2=DXDSTATE2 \)

* 

IF \( I=2 \), \( D2DXEDXP=DXDSTATE2 \)

* 

IF \( I=3 \), \( D2DZEDXP=DXDSTATE2 \)

* 

IF \( I=4 \), \( DXDXP=DXDSTATE, D2DXP2=DXDSTATE2 \)

* 

IF \( I=5 \), \( D2DZPDXP=DXDSTATE2 \)

* 

IF \( I=6 \), \( DXDZE=DXDSTATE, D2DXZ2=DXDSTATE2 \)

* 

IF \( I=7 \), \( D2DXZEDZP=DXDSTATE2 \)

* 

IF \( I=8 \), \( D2DXZEDZP=DXDSTATE2 \)

* 

IF \( I=9 \), \( DXDZP=DXDSTATE, D2DXZP2=DXDSTATE2 \)

Continue the block.

* 

\( D2DXPDZP=DXDZPDXP \)

This block contains the final derivative computations. It is used for all 3 values of \text{ROOTFLAG}. Do this part for \( I=1, \ldots, 9 \).

* 

IF \( I=1 \), \( DPDSTATE1=DPDXE, DPDSTATE2=DPDXE, DXDSTATE1=DXDXE, DXDSTATE2=DXDXE, DDCXSTATE1=DDCXDXE, DDCXSTATE2=DDCXDXE, \)
* * * *
DDCZDSTAT01=0, DDCZDSTAT02=0, D2PDSTAT02=D2PDXE2, *
* *
D2XDSTAT02=D2DXEDXP

IF I=2, DPDSTATE1=DPDXP, DPDSTATE2=DPDXE, DXDSTATE1=DXDXP,
*
DXDSTATE2=DXDXE, DDCXSTATE1=DDCXDXP, DDCXSTATE2=DDCXDXE,
*
DDCZDSTATE1=0, DDCZDSTATE2=0, D2PDSTATE2=D2PDXE2, *
* *
D2XDSTATE2=D2DXEDXP

IF I=3, DPDSTATE1=DPDXP, DPDSTATE2=DPDZE, DXDSTATE1=DXDXP,
*
DXDSTATE2=DXDZE, DDCXSTATE1=DDCXDXP, DDCXSTATE2=0, DDCZDSTATE1
*=0, DDCZDSTATE2=DDCZDZE, D2PDSTATE2=D2PDXE2, D2XDSTATE2=
* D2DXEDXP

IF I=4, DPDSTATE1=DPDXP, DPDSTATE2=DPDXP, DXDSTATE1=DXDXP,
*
DXDSTATE2=DXDXP, DDCXSTATE1=DDCXDXP, DDCXSTATE2=DDCXDXP,
*
DDCZDSTATE1=0, DDCZDSTATE2=0, D2PDSTATE2=D2PDXP2, D2XDSTATE2=
* D2DXDP2

IF I=5, DPDSTATE1=DPDXP, DPDSTATE2=DPDZP, DXDSTATE1=DXDXP,
*
DXDSTATE2=DXDZP, DDCXSTATE1=DDCXDXP, DDCXSTATE2=0,
* *
DDCZDSTATE1=0, DDCZDSTATE2=DDCZDZP, D2PDSTATE2=D2PDZPDXP,
* *
D2XDSTATE2=D2DXDPDXP

IF I=6, DPDSTATE1=DPDXE, DPDSTATE2=DPDZE, DXDSTATE1=DXDZE,
D2XSTATE2=D2XZPE

IF I=7,DPDSTATE1=DPDZP,DPDSTATE2=DPDZE,DXDSTATE1=DXDZP,

DXDSTATE2=DXDZE,DDCXDSTATE1=0,DDCXDSTATE2=DDCXDZE,

DDCZDSTATE1=DDCZDZP,DDCZDSTATE2=0,D2PDSTATE2=D2PDZEDZP,

D2XSTATE2=D2XZEDZP

IF I=8,DPDSTATE1=DPDZP,DPDSTATE2=DPDZE,DXDSTATE1=DXDZP,

DXDSTATE2=DXDZE,DDCXDSTATE1=0,DDCXDSTATE2=0,DDCZDSTATE1=

DDCZDZP,DDCZDSTATE2=DDCZDZE,D2PDSTATE2=D2PDZEDZP,

D2XSTATE2=D2XZEDZP

IF I=9,DPDSTATE1=DPDZP,DPDSTATE2=DPDZP,DXDSTATE1=DXDZP,

DXDSTATE2=DXDZP,DDCXDSTATE1=0,DDCXDSTATE2=0,DDCZDSTATE1=

DDCZDZP,DDCZDSTATE2=DDCZDZP,D2PDSTATE2=D2PDZP2,D2XSTATE2=

D2XZP2

D2MINDSTATE1=DXDSTATE1-DPDSTATE1/3

D2MINDSTATE2=DXDSTATE2-DPDSTATE2/3

DPOLYTMINVECTORSTATE1=(0,0,0,0,D2MINDSTATE1,2 POLYTMIN

D2MINDSTATE1)
DPOLYTMINVECTORDSTATE2=(0,0,0,0,DTMINDSTATE2,2 POLYTM
-
DTMINDSTATE2)

* * *
D2TMINDSTATE2=D2XSTATE2-D2PDSTATE2/3
-

DDeltaXSTATE1=(1,POLYTMIN,POLYTMIN) DDCXSTATE1+DCX

* D POLYTMINVECTORDSTATE1
-

2 *
DDeltaXSTATE2=(1,POLYTMIN,POLYTMIN) DDCXSTATE2+DCX

* D POLYTMINVECTORDSTATE2

* * T *
D2DELTAZSTATE2=DDCZSTATEE1 D POLYTMINVECTORDSTATE2+

* T *
D POLYTMINVECTORDSTATE1 DDCZSTATE2+DCZ DTMINDSTATE2+
-

2 DCX (DTMINDSTATE1 DTMINDSTATE2+POLYTMIN DTMINDSTATE2)
-

DDeltaZSTATE1=(1,POLYTMIN,POLYTMIN) DDCZSTATE1+DCZ

* D POLYTMINVECTORDSTATE1
-

2 *
DDeltaZSTATE2=(1,POLYTMIN,POLYTMIN) DDCZSTATE2+DCZ

* D POLYTMINVECTORDSTATE2

* T *
D2DELTAZSTATE2=DDCZSTATEE1 D POLYTMINVECTORDSTATE2+

* T *
D POLYTMINVECTORDSTATE1 DDCZSTATE2+DCZ DTMINDSTATE2+
-

2 DCZ (DTMINDSTATE1 DTMINDSTATE2+POLYTMIN DTMINDSTATE2)
DDMINSTATE1 = (DELTAX DDELTAXDSTATE1 + DELTAZ DDELTAZDSTATE1) / POLYDMIN

DDMINSTATE2 = (DELTAX DDELTAXDSTATE2 + DELTAZ DDELTAZDSTATE2) / POLYDMIN

* D2DMINDSTATE2 = (POLYDMIN (DDELTAXDSTATE1 - DDELTAXDSTATE2 + DELTAZ D2DELTAZDSTATE2 + DDELTAZDSTATE1 DDELTAZDSTATE2 - DELTAZ DDELTAZDSTATE1 DDMINDSTATE2)) / POLYDMIN

IF I = 1, DTMINDXE = DTMINDSTATE1, DDMINDXE = DDMINDSTATE1,

* d2tmindx2 = d2tmindstate2, d2dmindxe2 = d2dmindstate2

IF I = 2, DTMINDXEDXP = DTMINDSTATE2, D2DMINDXEDXP = D2DMINDSTATE2

IF I = 3, D2DMINDZEDXP = D2TMINDSTATE2, D2DMINDZEDXP = D2DMINDSTATE2

IF I = 4, DTMINDXP = DTMINDSTATE1, DDMINDXP = DDMINDSTATE1,

* d2tmindxp2 = d2tmindstate2, d2dmindxp2 = d2dmindstate2

IF I = 5, D2DMINDZPDXP = D2TMINDSTATE2, D2DMINDZPDXP = D2DMINDSTATE2

IF I = 6, DTMINDZE = DTMINDSTATE1, DDMINDZE = DDMINDSTATE1,

* d2tmindze2 = d2tmindstate2, d2dmindze2 = d2dmindstate2

IF I = 7, D2TMINDXEDZP = D2TMINDSTATE2, D2DMINDXEDZP = D2DMINDSTATE2

IF I = 8, D2DMINDZEDZP = D2TMINDSTATE2, D2DMINDZEDZP = D2DMINDSTATE2
IF I=9, DTMINDZP=DTMINDSTATE1, DDMINDZP=DDMINDSTATE1,
*       *       *       *
D2TMINDZP2=D2TMINDSTATE2, D2DMINDZP2=D2DMINDSTATE2

Continue the block.

*       *       T       *       T
D2TMINDXPDPZP=D2TMINDZPDZP , D2DMINDXPDPZP=D2DMINDZPDZP

END OF DISTFIT
APPENDIX C

P AND E PREDICTION ACCURACY

The SAA technique as seen in Chapter 2 is used to approximately solve the evader and pursuer equations of motion. The accuracy of those solutions is explored in this appendix. Turn to Figure C-1. Starting at $t_0=2$ sec., the P trajectory and three E trajectories are integrated forward in time using a fine ($\Delta t = .005$) time step. At the times 3.33, 4.67 and 6 sec. the predicted position is marked and the error noted. The max-$\alpha$ pulldown, max-$\alpha$ pull-up and the P response are from Figure 5.5-2b in the section of Chapter 5 describing evasion from LR guidance. The constant flight-path-angle path is from Figure 5.4-2b in the section on MMT guidance.

Note that the SAA predictions are more accurate for E than they are for P. There are three likely reasons for this. First and probably most important, the E predictions use cubic polynomials while quadratics are used in the P case. Also, each SAA iteration for E uses a single polynomial fit operation while the P computation cycle involves several such operations in cascade. Finally, the drag history with the P model may be more complicated in general due to the more complicated control histories provided; this complexity may make an accurate polynomial fit of drag vs. time more difficult.
Figure C-1: SAA Prediction Accuracy in Four Cases

Note: □ = P predicted point with error noted.
± = E predicted point with error noted.
The three Evader trajectories are a max-α pull-down, a maximum drag path with constant flight-path-angle and a max-α pull-up, respectively.
In this appendix we derive an approximation to the autopilot output-acceleration time response. This approximation is used to estimate the integral of the acceleration error when an update is made in the P control. This error integral is part of a correction term accounting for the effects of autopilot dynamics on the P flight-path-angle prediction made by SAA in Chapter 2.

From Appendix A the P autopilot loop response can be partially described by the equivalent closed-loop natural frequency $\omega_n$. This is possible because over a certain frequency band the autopilot transfer function is approximately that of a second-order system. At high frequencies (though still before the engine gimbal break-point) a negative closed-loop gain is seen. This corresponds to the "wrong way" initial response of tail-controlled rockets when a step input change occurs. From experience with simulations of the autopilot response, the step rise time $t_{\text{rise}}$ is approximately $5/\omega_n$.

When the P guidance provides the $A_{\phi}$ and $\dot{A}_{\phi}$ to be used until the next decision point, the control history has a step discontinuity. However, since the old control is a bias+ramp and the new one is also a bias+ramp, a pure step (with no initial or
final velocity) does not occur. The effects of the old and new ramps must be included in the response model. The viewpoint in deriving the approximate response is illustrated in Figure D-1. If \( A_{old} = A_{new}^0 \), the response is to a step and the rise time is \( t_{rise} = \frac{5}{\omega_n} \). If \( A_{old}^0 \neq 0 \) then there is an initial velocity. The response is assumed to "coast" on this.

\[
A_{new} = \text{new level}
\]

\[
A_{old} = \text{level at step}
\]

\[
A_{old} = A_{old}^0 + \text{step}
\]

\[
t_{rise} = \frac{5}{\omega_n}
\]

\[
t = \text{time}
\]

\[
G_{hf} = \frac{A_{new} - A_{old}^0}{A_{old} - A_{old}^0}
\]

Note: \( G_{hf} < 0 \)

\[
\text{area neglected if perfect response is assumed.}
\]

Figure D-1: The Approximate Response to a Bias+Ramp

velocity with the response to the new bias \( A_{new} \) and the new slope \( A_{new} \) added on. The slope of the step response to a new bias only is

\[
\frac{(A_{new} - A_{old}) - (A_{new}^0 - A_{old}^0)}{t_{rise}} G_{hf}
\]
Assuming that, to first order, the same rate of increase applies on top of the rate of increase due to coasting, the slope when \( A_{\text{new}} \neq 0 \) is

\[
\frac{(A_{\text{new}} - A_{\text{coast}}) \Delta t + (A_{\text{new}} - A_{\text{coast}}) \Delta t}{\Delta t} = \frac{(A_{\text{new}} - A_{\text{coast}}) (1 - \eta_{\text{rise}})}{\eta_{\text{rise}}} \tag{D-1}
\]

Solving for the new rise time and substituting \( \eta_{\text{rise}} = \frac{5}{\omega_n} \),

\[
\eta t = \frac{5/\omega_n}{1 - \frac{(A_{\text{new}} - A_{\text{coast}}) (5/\omega_n)}{(A_{\text{new}} - A_{\text{coast}}) (1 - \eta_{\text{rise}})}} \tag{D-2}
\]

If this response is compared to the "perfect" response in which the called-for bias+ramp is reached instantaneously, the integral of the difference \( \delta t \) is simply the area of the shaded triangle in Figure D-1. By inspection,

\[
\int_{t_0}^{t_0 + \eta t} s(t') dt' = \frac{1}{2} (1 - \eta_{\text{rise}}) (A_{\text{new}} - A_{\text{coast}}) \eta t \tag{D-3}
\]

This integral is subtracted from the integral of the commanded acceleration to give the approximate integral of acceleration used in the solution of the pursuer's equations of motion.
REFERENCES


USAF Academy, Colorado, 1972.

BIOGRAPHY

William Scripps Beebee was born in Centralia, Illinois on June 13, 1942. Soon after, his family moved to East Orange, New Jersey. He attended Columbian grammar school, Clifford Scott High School and Newark Academy. He received the BSE with Honors in Electrical Engineering from Princeton University in June, 1964.

After graduation, Mr. Beebee worked for the CALSPAN Corp. supervising electronics construction of an aircraft variable-handling-stability control system and designed electronic equipment at the Pratt and Whitney Division of the United Aircraft Corporation. He entered the University of Pennsylvania on a Ford Foundation fellowship in September, 1965, specializing in automatic control and communications theory. After a year of graduate study, he joined the Hamilton Standard Division of the United Aircraft Corporation (formerly, the UAC Corporate Systems Center), instituting an extensive statistical study of random errors in strapdown gyros. In October, 1967, he began work at The Analytic Sciences Corporation. He applied optimal estimation techniques to estimate gravity anomaly from Ships Inertial System outputs but soon shifted his efforts to studying dynamic responses of pulse-rebalanced gyro loops using describing function techniques and analog computer simulations.

Mr. Beebee entered MIT as a full-time student in September, 1969. Initially, he worked on optimal time-to-turn aircraft maneuvers under a Research Assistantship in the MIT Measurement Systems Laboratory but in June, 1970 transferred to the Trident group in the Charles Stark Draper Laboratory.

Mr. Beebee lives in Reading, Massachusetts, with his wife Shirley and two children, Eve and Peter.