IMPROVED CONVERGENCE FOR OPTIMIZATION OF EVASIVE MANEUVERING

by

Niall J. Duffy

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
Aerospace Engineering

APPROVED:

Dr. E. M. Cliff, Chairman

Dr. F. H. Lutze

Dr. H. L. Stalford

December, 1988
Blacksburg, Virginia

Improved Convergence for Optimization of Evasive Maneuvering

by

Niall J. Duffy

Committee Chairman: Eugene M. Cliff

(ABSTRACT)

Consider the problem of developing an algorithm that computes optimal pre-programmed evasive maneuvers for a Maneuvering Reentry Vehicle (MaRV) attacking a target defended with Anti-Ballistic Missiles (ABMs). The problem is large in terms of the number of optimization parameters, and perhaps in terms of the number of nonlinear constraints. Since both MaRV and ABM trajectories are expensive to compute, rapid convergence of the optimization algorithm is of prime concern. This paper examines a discontinuity in the cost function that degrades both the speed and the reliability of optimizer convergence. A solution is offered, proposing that the optimization algorithm be operated in a new parameter space, in which the discontinuity occurs at infinity. Effectively, the mapping prevents the optimization algorithm from crossing the discontinuity thereby improving optimizer convergence. Results comparing convergence with and without the parameter mapping demonstrate the effectiveness of the procedure.

Acknowledgements

I would like to thank for his guidance, advice, and heroic efforts at cutting through red tape. I am also grateful for the persistent encouragement and support provided by the management of Lincoln Laboratory, in particular by .

Contents

1 Introduction 1

1.1 Engagement Simulation 4

1.2 Optimization Algorithm 9

2 The Discontinuity 15

3 Early Approaches 25

3.1 Angle to be Rolled Reduction 25

3.2 Commanded Acceleration Reduction 27

3.3 LOS Rate Time Lag 30

3.4 Roll Direction Bias 31

4 Guidance Parameter Mapping 33

5 Other Modifications 39

6 Results 41

7 Conclusions 52

References 53

A Two Parameter Problem Input Files 54

B Six Parameter Problem Input Files 61

1 Introduction

The maneuvering capability of terminally guided Maneuvering Reentry Vehicles (MaRVs) produces extremely accurate target strikes, and in addition, it provides an evasive maneuvering capability against Anti-Ballistic Missiles (ABMs). The MaRV design uses a lifting body configuration to provide lateral acceleration that can be directed by rolling the vehicle's lift vector. In contrast to earlier designs, the magnitude of the lift vector is controllable by guidance. The vehicle is equipped with sensors and an inertial navigation system that provide accurate position and velocity information. However, no feedback is provided regarding ABM progress. As such, evasive maneuvers are conducted in an open loop fashion by flying the MaRV at a sequence of preselected aim points. This report addresses the problem of selecting the aim point locations and the aim point switching times that maximize ABM miss distance, yet still provide an adequate MaRV-target impact.

Solving the full optimization problem requires a powerful computer and a large budget of computer time. To properly shape the trajectory, over 5 aim points may be desired. Since each aim point has a switching time and 3 coordinates, the optimization problem may have over 20 parameters. The problem is further compounded by the implementation of nonlinear constraints to ensure that the optimal trajectory provides the MaRV with sufficient accuracy, speed

and path angle for effective target impact. As such, good convergence characteristics for the program are extremely important. As long as the cost function is continuous in the first and second derivatives, the variable metric optimization procedure implemented in the program provides the excellent convergence properties described in reference 1. However, the presence of discontinuities in the cost function degrades optimizer convergence and, even worse, reduces the reliability of finding any sort of optimum. This study investigates discontinuities in ABM miss distance and solves the convergence problem with a parameter mapping.

The discontinuity may appear in any version of the problem including complex versions with many optimization parameters and simple versions with only a few optimization parameters. To limit CPU usage, the solution offered in this study is developed for a relatively simple problem. A single ABM is assumed to defend against a MaRV programmed to use only 3 aim points. The MaRV pierces the earth's atmosphere at an altitude of 150,000 feet, 150,000 feet uprange (X coordinate) and 0 feet cross range (Y coordinate). The MaRV velocity vector is pointed directly at the target location $(X, Y, Z) = (0,0,0)$. The first aim point is fixed to the target, and is tracked for 6 seconds causing the velocity vector to continue pointing at the target. The second aim point is used beginning 6 seconds after reentry, until 8 seconds after reentry. The object of this opti-

mization problem is to compute optimal second aim point X and Y coordinates. The Z coordinate of the second aim point is fixed to 0. The third aim point is fixed to the target location (0,0,0), and is used from 8 seconds after reentry on through ground impact. The code allows for the implementation of a number of nonlinear constraints to ensure MaRV lethality including MaRV-target miss distance, MaRV impact velocity, and MaRV impact path angle. However, the only constraint imposed in this problem is MaRV-target miss distance, which is not allowed to exceed 200 feet. As demonstrated in Section 5, both the discontinuity, and the solution of this reduced scope problem apply to more complex problems.

Understanding the discontinuity problem requires detailed knowledge of certain aspects of the program, in particular, the MaRV guidance algorithm, the implementation of MaRV guidance commands, and the optimization algorithm. These descriptions make up the remainder of this introduction. For descriptions of other aspects of the program see reference 2. The discontinuity and the related optimizer convergence problems, which have been understood for some time, are described in section 2. Section 3 describes some of the earlier approaches taken to remove the convergence problems. Section 4 describes a parameter mapping that effectively removes convergence problems. Section 5 proposes other modifications recommended in conjunction with the parameter mapping. Results

comparing optimizer performance for the various approaches are presented in Section 6. Conclusions are presented in Section 7. Lastly, input files used for the study are listed in appendices A and B.

1.1 Engagement Simulation

This section describes the MaRV model and the ABM model used to compute trajectories. As the complete descriptions are involved, the models are only outlined here, with special attention being paid to those aspects of the simulation that are relevant to this study. In particular, the MaRV proportional navigation guidance scheme and the implementation of MaRV guidance commands are described in some detail. For a complete description of the engagement modeling for this program see reference 2.

The models are based on the 'equations of motion'. These are the differential equations that describe the relations between acceleration, velocity, and position vectors. For a six degree of freedom model, additional equations relating vehicle body moments, rotations and angles are also included. However, this simulation uses a point mass approximation, with time lags for approximating body rotational dynamics. The equations that model the MaRV flight are listed below.

$$
\begin{aligned}
\dot{V} &= -g\left(\frac{D}{W} + \sin\gamma\right) \\
\dot{\gamma} &= -g\left(\frac{L}{W}\cos\phi + \cos\gamma\right)/V \\
\dot{\chi} &= -g\left(\frac{L}{W}\sin\phi\right)/V\cos\gamma \qquad (1) \\
\dot{x} &= V\cos\gamma\cos\chi \\
\dot{y} &= V\cos\gamma\sin\chi \\
\dot{h} &= -\dot{z} = V\sin\gamma
\end{aligned}
$$

where V = velocity

γ = flight path angle

χ = heading angle with respect to x direction

ϕ = roll angle (zero when lift is down)

x, y, z = earth fixed coordinates

h = altitude

W = weight (constant)

L = aerodynamic lift

D = aerodynamic drag

In principle, these equations along with suitable initial conditions, are solved with a fourth order Runge-Kutta procedure with Gill coefficients (see reference 3) that is implemented with an extensive event trapping system to provide for the occurrence of discrete events not described in the equations of motion. Of course,

the complete description requires that one define the aerodynamic functions L and D, as well as the guidance procedure.

An understanding of the L and ϕ calculations is particularly important for this study. The process is outlined as follows. First, the guidance algorithm computes the desired or 'commanded acceleration' for the MaRV in earth fixed coordinates ($\overline{a}_{cc}$ of equation 3 to follow). This acceleration is converted to the wind axes coordinate system to determine the lift magnitude and the roll angle that will best achieve the commanded acceleration. These are termed 'commanded lift' and 'commanded roll angle.' The commanded lift value is tested to determine whether it is physically possible for the vehicle to achieve such an acceleration, that is, whether the vehicle can generate sufficient lift and whether the airframe can withstand such an acceleration. If not, the commanded lift is truncated to the maximum achievable value.

The next step is to implement the commanded roll angle and the commanded lift. For a six degree of freedom simulation, this process begins with the activation of the fin deflectors. The fins produce moments that result in pitch angular acceleration and in roll angular acceleration. The resulting pitching motion adjusts the angle of attack of the vehicle to give the commanded lift. The resulting roll motion aligns the actual roll angle with commanded roll angle. For this point mass simulation the lags in implementing the commanded lift and and

commanded roll angle are approximated by time lags on the lift coefficient and the roll angle respectively. The time lag equations are listed below.

$$\begin{aligned} \dot{C}_L &= \frac{(C_{L_c} - C_L)}{\tau_n} \\ \dot{\phi} &= \frac{\Delta\phi}{\tau_r} \\ \Delta\phi &= \phi_c - \phi \qquad \pi \geq \Delta\phi > -\pi \end{aligned} \tag{2}$$

where ϕ_c = commanded roll angle

C_{L_c} = commanded lift coefficient

ϕ = currently achieved roll angle

C_L = currently achieved lift coefficient

τ_r, τ_n = roll and pitch time constants

$L = C_L \frac{1}{2} \rho V^2 S$

ρ = density of air

S = reference area

The roll direction depends on the sign of the 'angle-to-be-rolled', $\Delta\phi$. If $\Delta\phi$ is positive, the MaRV will roll in a positive direction ($\dot{\phi} > 0$). If $\Delta\phi$ is negative, the MaRV will roll in a negative direction ($\dot{\phi} < 0$). Effectively, this arrangement causes the MaRV to always roll the 'short way' around. As will be described in more detail in Section 2, the source of the discontinuity lies in the fact that neighboring aim point locations may yield MaRV trajectories with different roll

directions. Consider an aim point location that yields $\Delta\phi = \pi$. A slight change in aim point location may change ϕ_c a slight amount so that $\Delta\phi = -\pi$. The new aim point will have the MaRV rolling in a negative direction rather than a positive direction, resulting in a significantly different trajectory.

The simulation has available two guidance schemes, dive-line guidance with cross product steering, and a proportional navigation law. This study addresses the proportional navigation law which is described with the following equations.

$$\begin{aligned} \overline{R} &= \overline{R}_T - \overline{R}_{RV} \\ \overline{V} &= -\overline{V}_{RV} \\ \overline{\omega}_s &= \frac{\overline{R} \times \overline{V}}{\overline{R} \cdot \overline{R}} \\ \overline{a}_{cc} &= K_p \frac{\overline{R} \cdot \overline{V}}{\mid \overline{R} \mid\mid \overline{R} \mid} \overline{\omega}_s \times \overline{V}_{RV} \end{aligned} \tag{3}$$

where K_p = navigation gain

$\overline{V}_{RV}$ = MaRV velocity vector

$\overline{R}_{RV}$ = MaRV position vector

$\overline{R}_T$ = target position vector

$\overline{\omega}_s$ = Line Of Site (LOS) rate vector

In simple terms, the guidance law specifies that the MaRV turn towards the target with a commanded acceleration proportional to the LOS rate. The simplicity of this guidance law is used to develop the parameter mapping, discussed

in Section 4, that removes the convergence problems caused by the discontinuity.

The ABM model is considerably more sophisticated than the MaRV model. Moreover, the details of the model are not particularly relevant to this study. For this reason, the description presented here is brief. The ABM is modeled in a similar manner to the MaRV, except it is a staged, thrusting skid-to-turn vehicle rather than a nonthrusting roll-to-turn vehicle. The ABM uses point mass equations with time lags to model pitch and yaw dynamics. The equations of motion are written in Euler parameters rather that Euler angles to avoid the singularity that occurs with vertical flight. The guidance scheme is a blend of proportional navigation and pursuit guidance designed to keep the MaRV in front of the ABM. For a description of these features see reference 1.

1.2 Optimization Algorithm

In contrast to Soviet literature, in Western literature, optimization problems are generally set up as minimization problems. In this problem, $f(x) = 1/(\text{ABM-MaRV closest approach})^2$ is minimized to find a maximum of ABM-MaRV closest approach distance. The various parameters that define a trajectory are all fixed except for the vector of optimization parameters, denoted x. This vector may consist of any combination of the aim point coordinates, the aim point switching criteria, and the parameters defining the initial state of the MaRV. Not all values

of x give acceptable trajectories. In particular, the trajectory should satisfy requirements for target miss distance, final velocities, and final path angle. Such requirements are met by imposing equality constraints, denoted $g_j(x) = 0, j = 1, ..., m$, and inequality constraints, denoted $g_j(x) \geq 0, j = m+1, ..., m+p$. The goal is to select, from the values of x that satisfy the constraint equations, those that minimize the function f.

For this problem, the cost function is somewhat expensive to compute as trajectory integrations are required. Moreover, derivatives cannot be computed analytically, so numerous function evaluations are required to compute finite difference derivative approximations. As such, there is considerable motivation for an efficient optimization algorithm with good convergence characteristics. The Kelley-Speyer algorithm used in this program exploits the smoothness properties of the cost function to provide excellent convergence characteristics. The theoretical basis of the algorithm is described by Gill, et. al. in reference 4, Kelley and Speyer in reference 5, and Fletcher in reference 6. This section describes the algorithm and the consequences of not having a continuous cost function $f(x)$.

The Kelley-Speyer optimization algorithm can perform unconstrained optimization, constrained optimization using the penalty-function approximation, and constrained optimization using Variable-Metric Projection. This study is concerned with the third option, which is generally best at performing con-

strained optimization and also relies most heavily on smoothness properties of both the cost function and the constraint functions. The general approach is an iterative one, in which a sequence of optimization cycles are performed, each consisting of a search direction calculation followed by a one-dimensional search. If the one-dimensional search leads to constraint violations, a correction step is performed to restore the parameters to the feasible set. The cycles repeat until either convergence, or until the optimizer gets 'hung-up' on a discontinuity.

The search step is given by the equation

$$\Delta x \;=\; -\alpha H f_x. \tag{4}$$

The one-dimensional search determines the $\alpha > 0$ that minimizes f along the search direction, $-Hf_x$. The metric matrix H is initialized to be positive definite, commonly as a diagonal matrix. With H initialized to the identity matrix, the procedure starts out like a first order method. However, with metric updates using the formula

$$\begin{aligned} H + \Delta H \;&=\; H + \frac{\Delta x \Delta x^T}{\Delta x^T \Delta f_x} - \frac{H \Delta f_x^T H}{\Delta f_x^T H \Delta f_x} + \phi \nu \nu^T \\ 0 \le \phi \le 1 \end{aligned} \tag{5}$$

where

$$\nu \;=\; (\Delta f_x^T H \Delta f_x)^{1/2} \frac{\Delta x}{\Delta x^T \Delta f_x} - \frac{H \Delta f_x}{\Delta f_x{}^T H \Delta f_x}, \tag{6}$$

the process rapidly approximates a second-order procedure. The reason being that this update causes the metric to approach the inverse Hessian, $[f_{xx}]^{-1}$.

Applying this process to a positive definite quadratic cost function gives a metric matrix H that exactly equals the inverse Hessian in at most n updates, where n is the number of parameters. At this point, the one-dimensional search becomes trivial as $\alpha = 1$ automatically gives the optimum. This performance occurs to some extent with any function as long as it is smooth. This is because smooth functions have Taylor Series expansions about the optimum in which the terms of third order and higher are insignificant, thereby approximating a quadratic.

The description presented above covers several versions of metric updates. Setting ϕ to zero gives the original version, the Davidon-Fletcher-Powell (DFP) procedure. Setting ϕ to 1 gives the Broyden-Fletcher-Goldfarb-Shanno (BFGS) procedure. For reasons that are not fully understood, better performance is achieved using the (BFGS) update and that is the update used in this study.

The performance of variable-metric procedures surpasses that of first order methods and that of second order procedures by combining the best characteristics of each. Early in the process, the variable-metric procedure produces good progress in a fashion similiar to a first order process. Later, as the metric accumulates curvature information, the process resembles a second order process.

Moreover, by updating the metric only when

$$\Delta x^T(\Delta f_x + \Delta g_x \lambda) > 0, \tag{7}$$

the metric remains positive definite, thereby ensuring that the search direction has the descent property.

Equality constraint functions, denoted $g_j(x) = 0, j = 1, ..., m$, and active inequality constraint functions, denoted $g_j(x) \geq 0, j = m + 1, ..., m + p$, are adjoined to the function to be minimized with Lagrange multipliers to make a new cost function $f + g\lambda$ where $\lambda = -({g_x}^T H g_x)^{-1} {g_x}^T H f_x$. The new function, possibly augmented with quadratic terms in g, is optimized as before except that along the one-dimensional search, tests are performed to determine whether any constraints have been violated sufficiently to warrant a correction step.

The correction procedure employed by the Kelley-Speyer algorithm is essentially a separate minimization problem, in which the function to be minimized is a sum of constraint function terms, with a penalty-function type constraint to ensure that f doesn't get too large. This minimization process uses a metric update similiar to the DFP procedure, with the exception that it incorporates an elaborate system to weight each constraint term based on the derivatives of the constraints. Moreover, the metric is updated for each constraint weighting, taken one at a time.

The theoretical basis of the algorithm assumes smoothness in the cost function

and in each of the constraint functions. If the functions are smooth, the procedure delivers the quadratic convergence described in reference 1. On the other hand, if the functions are not smooth the procedure may not converge. The usual problem is that the one-dimensional search ends on a discontinuity, an unfortunate event that occurs frequently because in one dimension the discontinuity may be a local minimum. Finite difference derivatives computed across the discontinuity are invalid and lead to a bad search direction in which no improvement is possible. The optimizer is 'hung-up.'

Problems are also caused by 'cusps' in the cost function. These are discontinuities in first derivatives that look like kinks in the cost function surface. When the one-dimensional search ends on a minimum that is a cusp, the new search direction will be such that no improvement is possible. In actual fact, if the optimizer were modified to follow the cusp, improvement could be found. As there is no such modification, the optimizer gets 'hung-up.'

If discontinuities are present in either constraint functions or the cost function, these same problems may plague the constraint restoration process, causing constraint violation recovery to get 'hung-up.' However, this problem was never observed in the course of this study.

2 The Discontinuity

The discontinuity addressed here is a real life phenomena inherent in the roll-to-turn MaRV configuration; it is not just a flaw in the modelling process. When a pair of MaRV trajectories are computed for different aim point locations, the trajectories are generally similar, as long as the aim points are near each other. However this is not always the case. In rare instances, no matter how near the aim points are made to each other, significant differences result in the trajectories.

At first glance, the idea that results of a numerical integration would have a discontinuity seems unlikely. However, it should be emphasized that the discontinuity exists over the domain of guidance parameters, not over the independent variable of integration, time. The discontinuity presented here is related to the MaRV roll-to-turn configuration, in which the airframe can provide lift in only one direction. As mentioned in section 1.1, commanded accelerations, generated by guidance, are implemented by computing a commanded roll angle, and rolling the airframe to this angle. Both positive and negative roll directions can achieve the commanded roll angle; however, the direction taken is that which gives the shortest 'angle-to-be-rolled', $\Delta\phi$.

The discontinuity occurs at aim point coordinates that result in a trajectory in which at some time $\Delta\phi = \pm\pi$. At this point, both roll directions give equal angles-to-be-rolled. On one side of the discontinuity, the MaRV rolls right. On

the other side of the discontinuity, the MaRV rolls left. The differences in the roll right and roll left trajectories result in a discontinuity in ABM-MaRV miss distance, and in any other function of the MaRV trajectory. In general, the commanded roll angle varies smoothly, and the airframe responds to changes quickly, seldom falling far behind the commanded roll angle, except at control switches. It is the control switches that cause the large jumps in commanded roll angle associated with the discontinuity.

From the simple, conceptual understanding of proportional navigation to a fixed target presented in Section 1.1, it is possible to "back out" the locations of the discontinuity in aim point coordinates. In terms of commanded roll angle, the discontinuity occurs at the commanded roll angle that is π radians from the current roll angle. As the guidance law gives acceleration commands that turn the MaRV velocity vector towards the aim point, it is clear that the discontinuity consists of aim points in the plane of the velocity vector $\overline{V}$ and the lift vector $\overline{L}$. The line containing the MaRV velocity vector divides this plane into two half planes. Denoting the vector from the MaRV to the aim point as $\overline{P}$, the discontinuity lies in the half plane where $\overline{P} \cdot \overline{L} < 0$. For the reduced scope problem of this paper, the Z coordinate is fixed to 0, so the discontinuity is restricted to the half line defined by the intersection of the discontinuity half plane with the $Z = 0$ plane. (In special cases, where $\overline{V}_z = 0$, the intersection

with the ground plane may not exist or the intersection may give a line.) An example of a discontinuity half plane and the corresponding discontinuity half line is illustrated in figure 1.

Figure 2 shows a plot of $\Delta\phi$ at 6 seconds into the engagement as a function of aim point coordinates. A discontinuity exists along the line from (X, Y)=$(0, 0)$ to the edge of the plot, (X, Y)=$(1540, 3000)$. As the aim point is moved from $(X, Y) = (769, 1500)$ to $(X, Y) = (771, 1500)$ the angle-to-be-rolled jumps from $-\pi$ to $+\pi$. The corresponding ABM-MaRV miss distances, plotted in figure 3, indicate that the surface has a number of discontinuities. One of these corresponds to the discontinuity in $\Delta\phi$ at 6 seconds shown in figure 2. The other discontinuities correspond to jumps in $\Delta\phi$ at 8 seconds. The surface of MaRV-target miss distance, plotted in figure 4, illustrates related discontinuities in the constraint function.

The relationship between the jumps in $\Delta\phi$ and the discontinuities in the cost and constraint functions is more clearly illustrated by cuts taken through these surfaces. Figures 5 and 6 show cuts along $Y1 = 2500$ feet, for ABM-MaRV miss distance, MaRV-target miss distance, $\Delta\phi$ at 6 seconds, and $\Delta\phi$ at 8 seconds. Figure 5 is an example of where a jump in $\Delta\phi$ at 6 seconds is the cause of the discontinuity, and figure 6 is an example of where a jump in $\Delta\phi$ at 8 seconds is the cause of the discontinuity. Other functions that are based on the MaRV

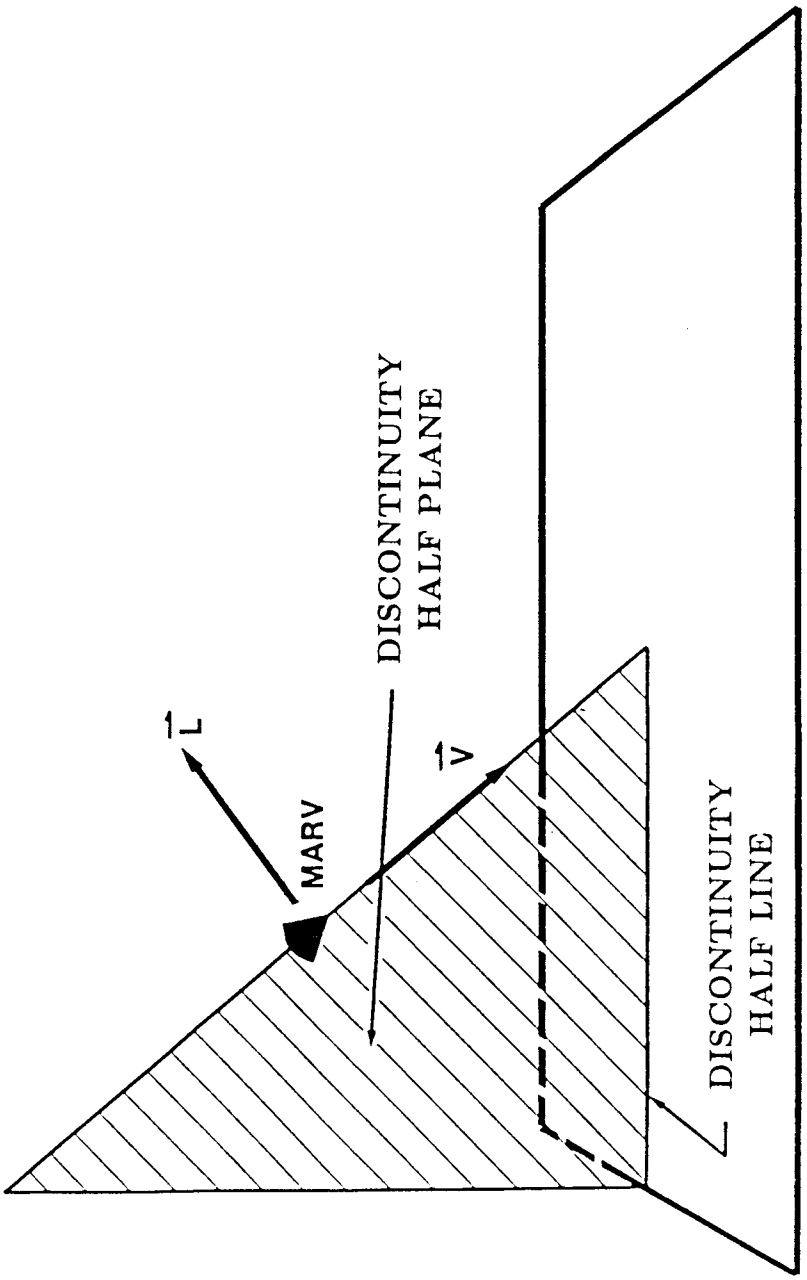

Figure 1: The shaded region of this figure depicts the discontinuity half plane corresponding to the illustrated MaRV position, velocity, and roll angle. For this example the intersection of the half plane with the ground, $Z = 0$, gives a half line.

trajectory, such as MaRV impact velocity and MaRV final path angle also show related discontinuities.

This type of discontinuity is inherent in the roll to turn type of vehicle configuration because there are always at least two ways to roll to achieve the commanded roll angle. As long as both roll directions are allowed, there will be a condition in which the roll direction jumps. If a restriction is made to roll direction permitting only one direction, then there will be a condition at which the angle-to-be-rolled jumps by 2π. Both of these effects lead to discontinuities in trajectory. Attempts to smooth this type of discontinuity also lead to undesireable results. The result may be to delay implementation of the unmodified commanded roll angle, thereby delaying the time at which the roll direction is defined. However the discontinuity remains.

A final point regarding the discontinuity is that even though the discontinuity occupies only a small fraction of parameter space, the odds of the optimization algorithm encountering the discontinuity are quite high. This is because the optimizer doesn't wander about randomly, but rather, it actively homes on features that look like optima. Frequently, a discontinuity has such features. Thus, even the single discontinuity described here may have a devastating effect on convergence.

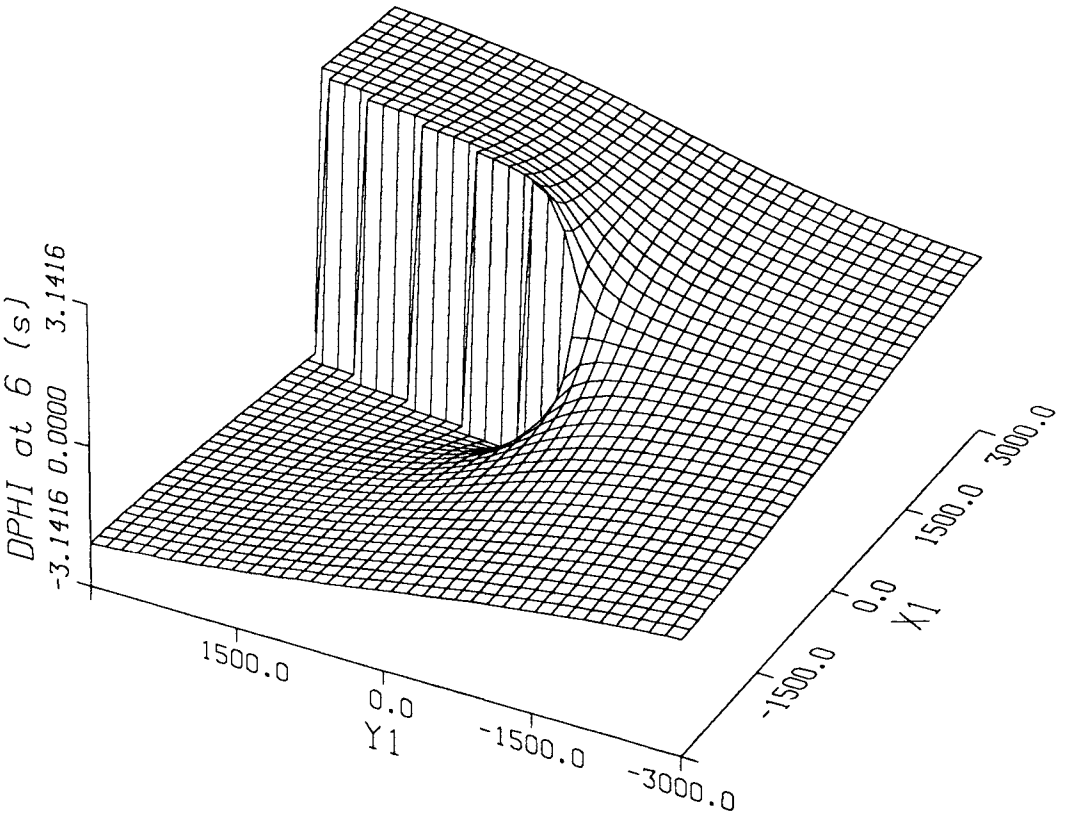

Figure 1: This figure illustrates how the angle-to-be-rolled can jump from $-\pi$ to π. The $\Delta\phi$ commanded at the 6 second control switch is plotted over the surface of aim point coordinates X and Y. The shape of the discontinuity is characteristic for the roll-to-turn MaRV configuration flying with proportional navigation. It is always a half line with a starting point and direction that depend on the MaRV position, velocity, and roll angle at the control switch.

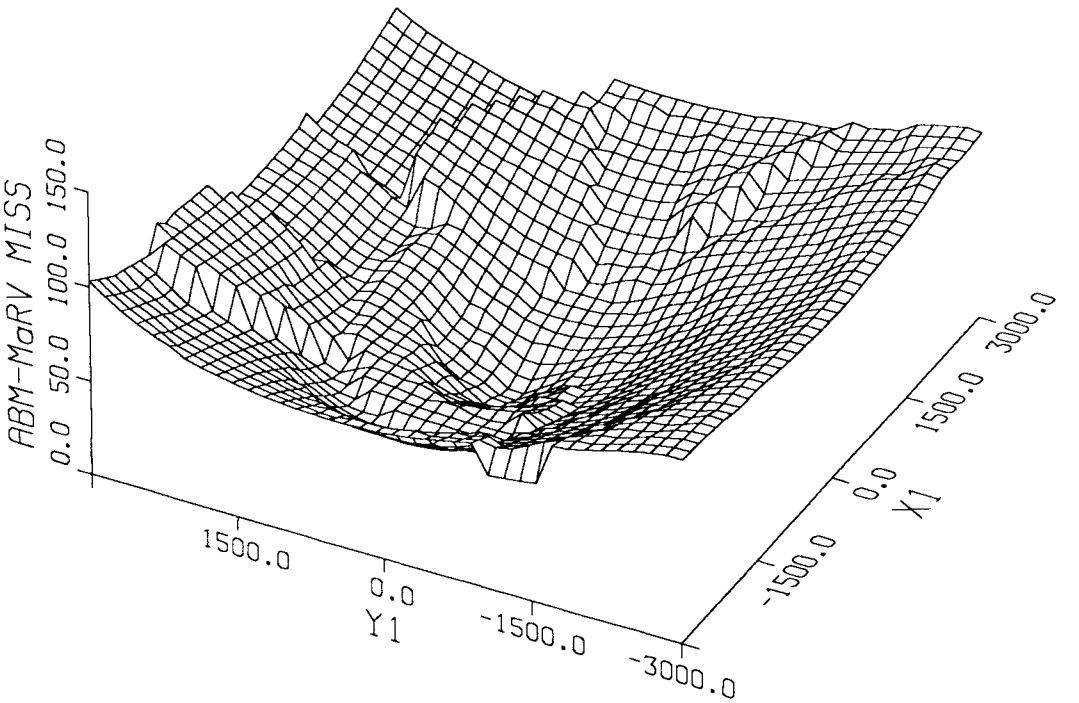

Figure 2: This plot shows the surface of ABM-MaRV miss distance over aim point coordinates implemented at the first control switch. Numerous discontinuities are visible. The discontinuity in $\Delta\phi$ of figure 1 causes a corresponding discontinuity in ABM-MaRV miss distance. The other discontinuities are caused by jumps in $\Delta\phi$ at the 8 second control switch.

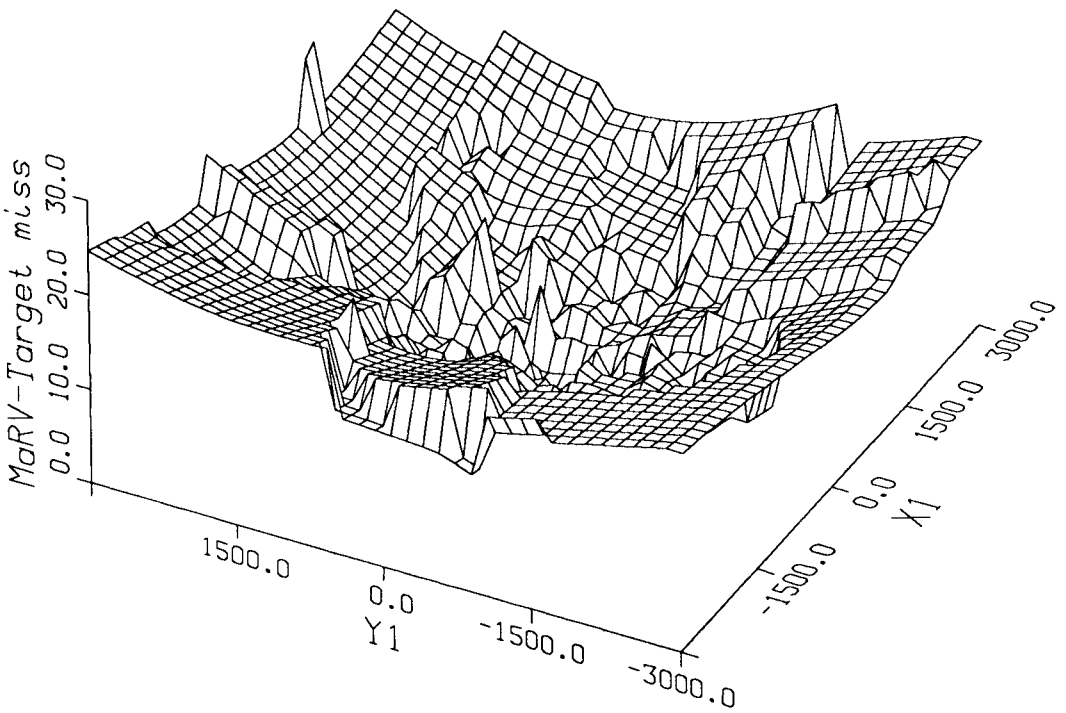

Figure 3: This plot shows the surface of MaRV-target miss distance over aim point coordinates implemented at the first control switch. Numerous discontinuities are visible, and all are attributed to the discontinuities in $\Delta\phi$ for either 6 seconds, shown in figure 2.

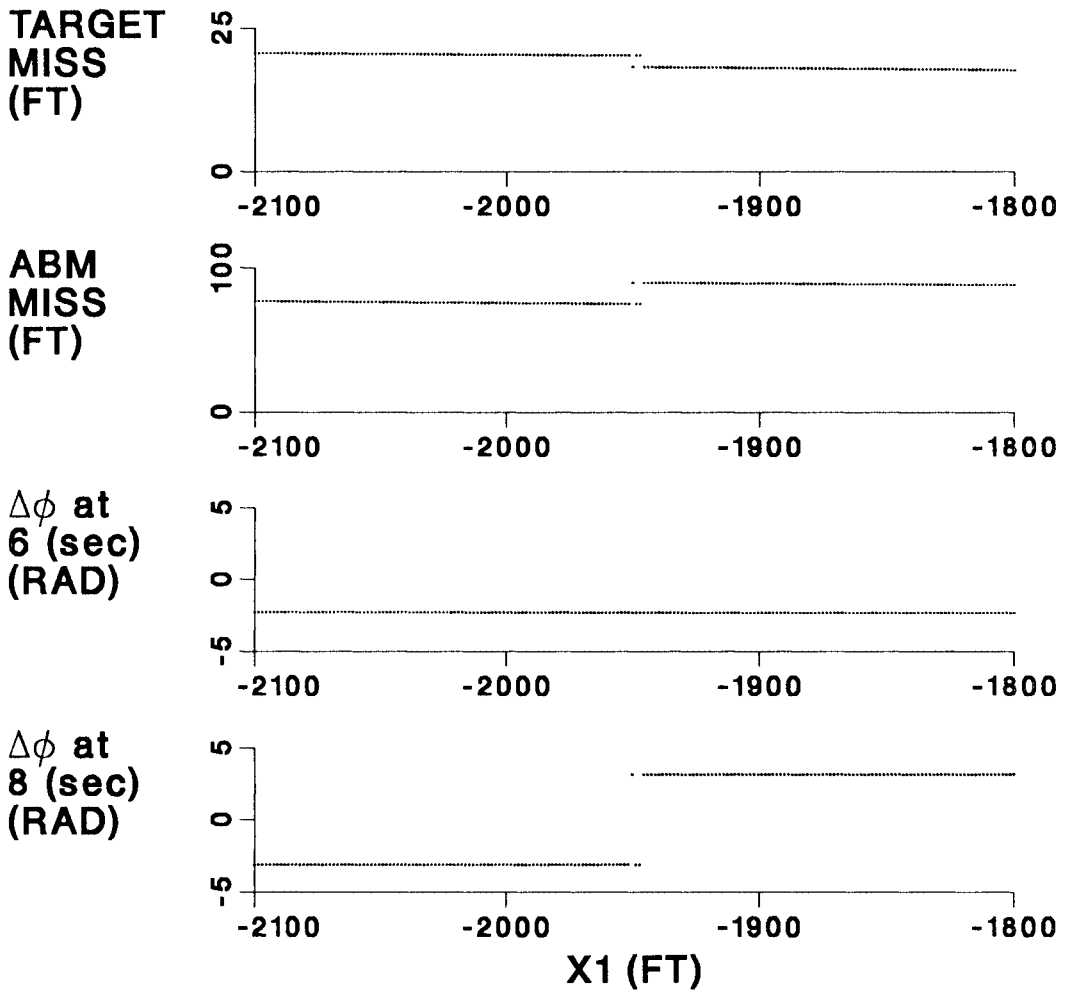

Figure 4: These curves are cuts from the surfaces of figures 2, 3 and 4 taken along $Y1 = 2500$. The discontinuities in ABM-MaRV miss distance and MaRV-target miss distance are caused by the jump in $\Delta\phi$ for the 8 second control switch. These jumps are not clean, that is, the values alternate between the high and the low values. This behavior is caused by numerical error that may perturb a trajectory so that it behaves like a run for the other side of the discontinuity.

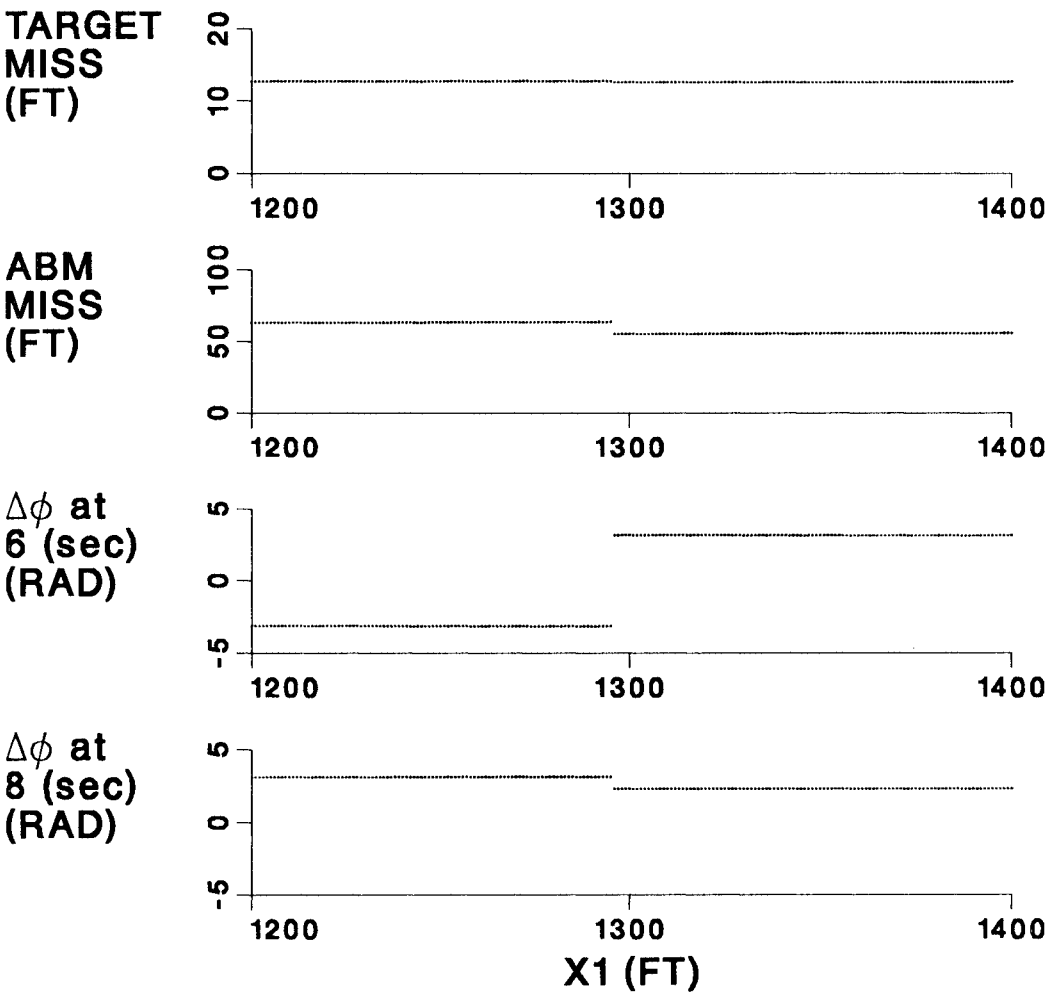

Figure 5: These curves are cuts from the surfaces of figures 2, 3 and 4 taken along $Y1 = 2500$. The discontinuities in ABM-MaRV miss distance and MaRV-target miss distance are caused by the jump in $\Delta\phi$ for the 6 second control switch.

3 Early Approaches

The object of this study is to remove the convergence problems caused by the discontinuity, without significantly changing the problem. The early attempts in this regard focused on slight modifications to the engagement simulation that retained the basic characteristics of the true engagement, yet provided smoother cost and constraint functions. This section briefly describes some of these approaches, listing them in the order they were considered. While none of the approaches described here produced a satisfactory solution, the descriptions, reasoning and faults of the various approaches do shed some light on the problem. The successful approach, the parameter mapping, is presented in Section 4.

3.1 Angle to be Rolled Reduction

The first attempt at smoothing the discontinuity was a modification to the implementation of MaRV guidance that reduces an actual $\Delta\phi$ in the neighborhood of π to an effective command closer to zero. The effect of this modification is to cause the MaRV to roll a very small amount if $\Delta\phi$ is near π or $-\pi$, thereby reducing the differences on the two sides of the discontinuity. The relations that

describe the modification are expressed by

$$\begin{aligned} \phi_c &= \phi + \epsilon_2 && \varepsilon < \delta \\ \epsilon_1 &= \varepsilon/\delta \\ \epsilon_2 &= [\epsilon_1(2-\epsilon_1) - \varepsilon] sign(\pi, \Delta\phi) \\ \varepsilon &= \frac{\pi - |\Delta\phi|}{\pi} \\ \delta &= 0.1 \end{aligned} \tag{8}$$

where ϕ_c and $\Delta\phi$ have the same meaning as in equation 2. Figure 7 shows plots of $(\phi_c - \phi)$ versus $\Delta\phi$ defined on the interval $(0, 2\pi)$ comparing the modified version with the unmodified version. The unmodified ϕ_c has a discontinuity at $\Delta\phi = \pi$ whereas the modified ϕ_c is continuous.

By rolling a very small amount if $\Delta\phi$ is near $\pm\pi$, the MaRV continues accelerating in the wrong direction, away from the unmodified commanded acceleration. The result is that all MaRV trajectories where the actual $\Delta\phi$ is reduced end up with noticeably larger target miss distances than those trajectories without reduced $\Delta\phi$. This modification does tend to match MaRV-target miss distances on both sides of the discontinuity by giving both sides large miss distances, however, it also produces a cusp, or discontinuity in the derivatives of the cost function. Unfortunately, the cusp is almost as damaging to optimization performance as the basic discontinuity. Cusps produced by this modification appear in the surface of MaRV-target miss distance shown in figure 8. This plot may be compared with the corresponding plot for the unmodified version shown in figure 4. This

modification was not included in subsequent versions of the code.

3.2 Commanded Acceleration Reduction

This modification is based on the observation that as long as the MaRV is flying with zero lift, it doesn't matter which way the MaRV rolls to achieve a $\Delta\phi = \pm\pi$. Both roll directions give the same ballistic trajectory, thereby eliminating the discontinuity. For obvious reasons $\overline{a}_{cc}$ can't be set to zero everywhere, but it does make sense to reduce $\overline{a}_{cc}$ when the MaRV roll angle is poorly aligned with the commanded roll angle. Apart from the smoothing benefits, such a modification improves the match of the commanded acceleration with achieved acceleration, thereby giving smaller MaRV-target miss distances.

The first attempt at implementing this rule used a hard cutoff for commanded acceleration. If the angle-to-be-rolled is larger than 60 degrees, commanded acceleration is reset to zero. Otherwise, the commanded acceleration was implemented as before. The abruptness of the cutoff produced a discontinuity of its own. One trajectory might see the full cutoff while a neighboring trajectory may see no acceleration reduction. To avoid this problem, the rule was modified to smooth the cutoff using a $\cos\Delta\phi$ multiplier according to the relation

$$\begin{aligned} a_{cc} &= a_{cc}\cos\Delta\phi \quad |\Delta\phi| \leq \tfrac{\pi}{2} \\ a_{cc} &= 0. \qquad\qquad\quad \Delta\phi > \tfrac{\pi}{2} \end{aligned} \tag{9}$$

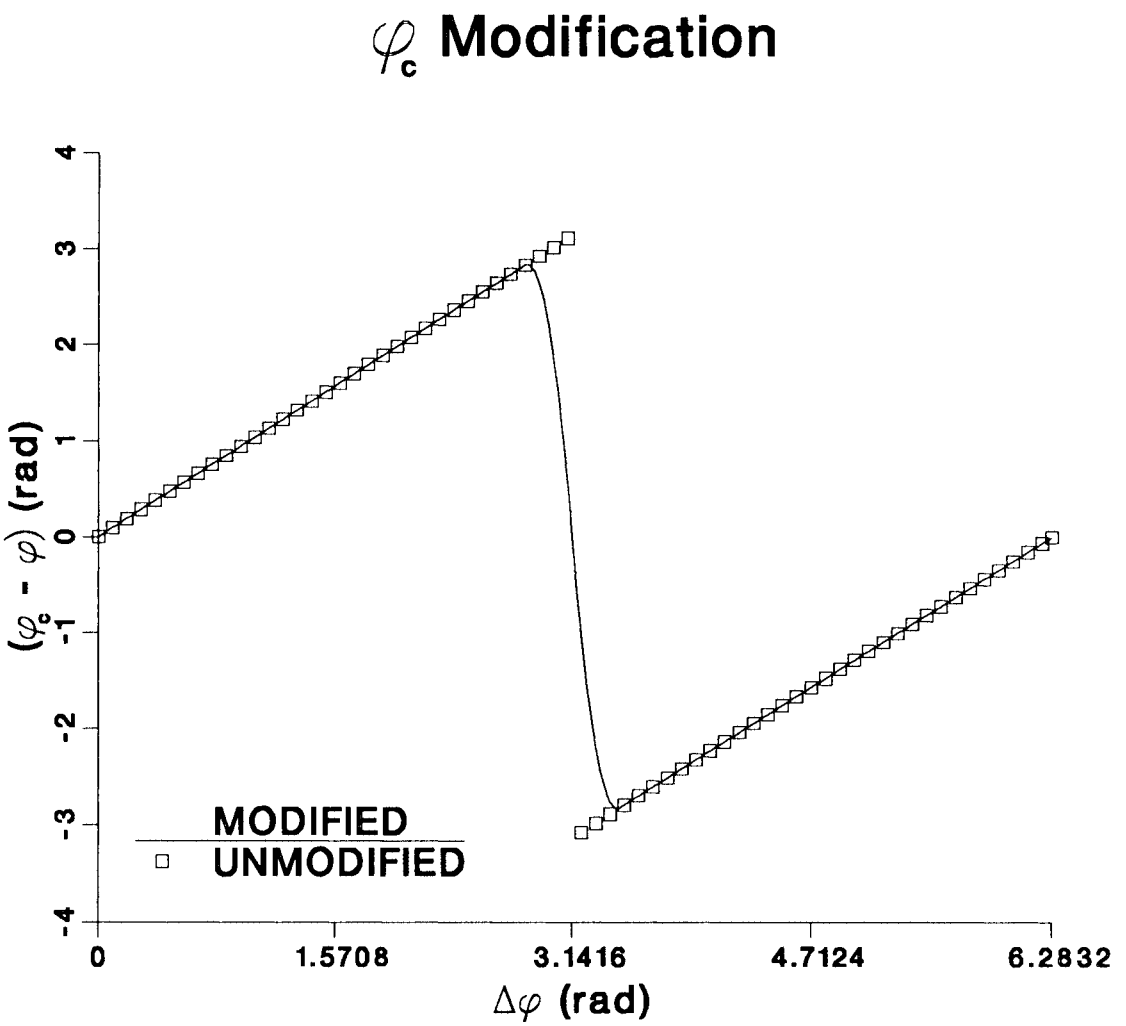

Figure 6: This plot illustrates how the angle-to-be-rolled $\phi_c - \phi$ is modified to reduce the unmodified $\Delta\phi$. The unmodified ϕ_c has a discontinuity at $\Delta\phi = \pi$ whereas the modified ϕ_c is continuous.

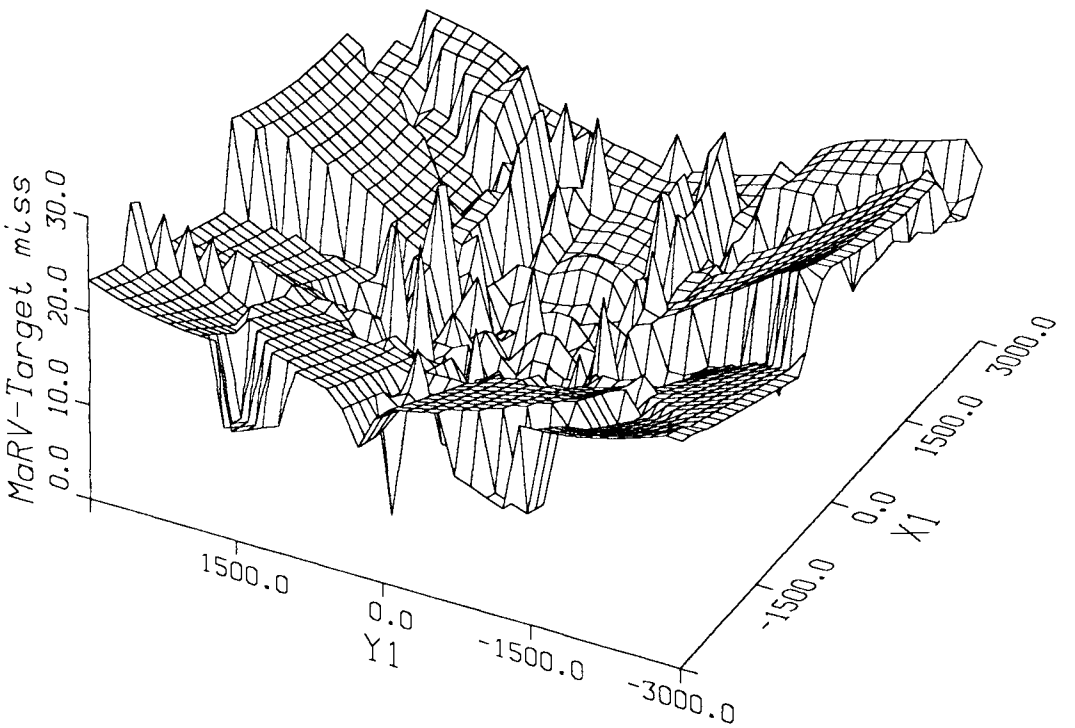

Figure 7: This plot illustrates the cusps produced by the guidance modification that reduces $\Delta\phi$ in the neighborhood of $\pm\pi$. For comparison with the unmodified version see figure 4.

Again ϕ_c and $\Delta\phi$ have the same meaning as in equation 2. This modification is retained in subsequent versions of the code.

3.3 LOS Rate Time Lag

This approach is based on the observation that if the LOS rate didn't change abruptly at the control switch, the MaRV would be able to keep up with the commanded roll angle, without $\Delta\phi$ getting even close to π. One way to slow the LOS rate is to apply a first-order time lag according to the equation

$$\dot{\bar{\omega}}_L = \frac{\bar{\omega}_s - \bar{\omega}_L}{\tau_\omega} \tag{10}$$

where $\bar{\omega}_s$ = actual LOS rate vector

$\bar{\omega}_L$ = lagged LOS rate vector

τ_ω = LOS rate time constant

($\tau_\omega = 0.1$)

This modification effectively prevents the LOS rate from reversing direction abruptly at control switches; however, it does not remove the discontinuity.

The flaw in this reasoning is the assumption that a large $|\Delta\phi|$ is the cause of the discontinuity. The real culprit is the reversal of the LOS rate that occurs when certain aim points are used. Whether the LOS rate switches direction abruptly as it does without the lag, or whether the switch is slowed with the lag is irrelevant. Neighboring trajectories may still have different roll directions.

However, with the lag, it takes a longer time, so differences in roll rate directions develop later in the flight.

3.4 Roll Direction Bias

The rule that determines which way the MaRV will roll says to roll in the direction that will align the lift vector with commanded acceleration quickest. This is not the only way to compute roll direction. An equally valid rule for computing roll direction might say to roll in a positive direction if the angle-to-be-rolled is less than $\frac{5\pi}{6}$, and to roll in a negative direction if the angle-to-be-rolled is greater than $\frac{-7\pi}{6}$. This example is implemented by the following equation with μ set to $\frac{\pi}{6}$.

$$\Delta\phi = \phi_c - \phi \qquad (\pi + \mu) \geq \Delta\phi > (-\pi - \mu) \tag{11}$$

where μ = roll direction bias

This equation biases the roll direction calculation to favor the direction with the opposite sign of μ.

The effect of the bias is to shift the location of the discontinuity from $\Delta\phi = \pm\pi$ to $\Delta\phi = \pi + \mu$ or $\Delta\phi = -\pi - \mu$. Moreover, there is a related shift in discontinuity aim points. In general, no advantage is gained by shifting the location of the discontinuity. However, for some simplified problems the shift can be useful. One

example involves using the parameter mapping of the next section for defining the second set of aim point coordinates, and fixing the third, final set of aim point coordinates to the true target. In this example, the range of values for $|\Delta\phi|$ at the second control switch is narrow. The value of $|\Delta\phi|$ is always near π thereby yeilding trajectories that are always near the discontinuity. For this particular problem, a shift in the location of the discontinuity can eliminate the discontinuity.

The reason for the narrow range of $\Delta\phi$ is related to the fixed aim point locations for this particular problem. The MaRV begins by flying straight toward the target. At 6 seconds the MaRV accelerates away from the original line to the target, towards the new aim point. At 8 seconds, the MaRV acceleration is reversed, causing the MaRV to turn back, towards the target. At this time, 8 seconds, guidance commands a $\Delta\phi$ of about $\pm\pi$. Thus it is advantageous to shift the location of the discontinuity away from $\Delta\phi = \pm\pi$. The bias, $\mu = \frac{\pi}{2}$, favors negative roll rates, thereby eliminating the use of positive roll rates. The critical time when this shift is required is just during and after the 8 second control switch. To minimize side effects of the bias, it is only implemented from 8.0 seconds to 8.5 seconds into the engagement.

4 Guidance Parameter Mapping

The discontinuity is removed with an aim point mapping that exploits the simplicity and predictability of the discontinuity. With the mapping, the optimization algorithm works in a new parameter space, in which the aim point location is described with r, p, q, rather than X, Y, Z. One side of the discontinuity in X, Y, Z maps to $q = +\infty$ while the other side of the discontinuity maps to $q = -\infty$. In doing so, the optimization algorithm is effectively prevented from crossing the discontinuity, and from taking finite difference derivatives across the discontinuity. The result is improved optimization convergence.

Two features of the discontinuity make this mapping possible. The first feature is the a priori knowledge of the location of the discontinuity in aim point coordinates; it is in the half plane described in Section 2. The second feature is the simple shape of the discontinuity that allows a transformation that maps discontinuities to infinity. If these characteristics are not met, such a mapping is not possible.

The simplest way of describing the location of the discontinuity is with a cylindrical coordinate system centered at the MaRV, with an axis of symmetry along the velocity vector. In this system, the discontinuity is described simply with a constant angle coordinate that is π from the lift vector angle coordinate. This coordinate system, shown in figure 9, is the basis of the mapping.

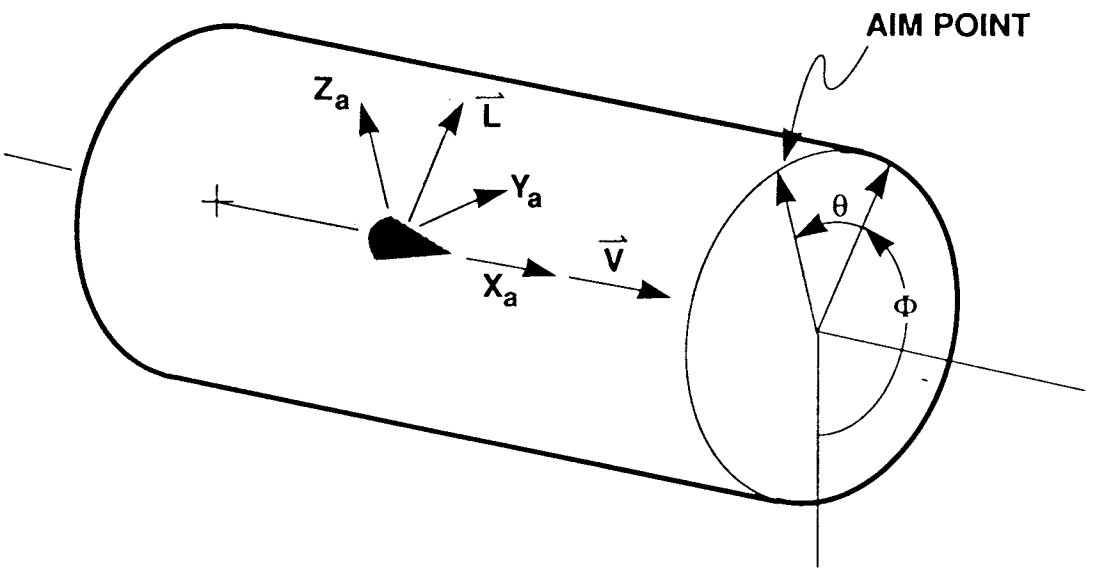

Figure 9: Cylindrical Coordinates referenced to "Aeroballistic Axes"

The angle to the target point measured from the lift vector about the X "aeroballistic axis" is given by variable name θ. (It turns out that θ is equivalent to the angle-to-be-rolled, $\Delta\phi$ of equation 2.) If θ is greater than π, the MaRV will roll in a positive direction. If θ is less than π, the MaRV will roll in a negative direction. The discontinuity exists at $\theta = \pi$. The angular coordinate, θ is stretched with an arctan function to get the optimizer parameter 'q'. Parameter q approaches positive infinity on one side of the discontinuity and negative infinity on the other side of the discontinuity. The relations that map optimization parameters, r, p, q, to aim point space, T_x, T_y, T_z, are given by the

following equations.

$$
\begin{aligned}
\theta &= 2\arctan(q) \\
T_{xa} &= p \\
T_{ya} &= 0 \\
T_{za} &= r^2 \\
T_x &= \cos(\gamma)\cos(\chi)T_{xa} \\
&\quad + [\sin(\theta+\phi)\sin(\gamma)\cos(\chi) - \cos(\theta+\phi)\sin(\chi)]T_{ya} \\
&\quad + [\cos(\theta+\phi)\sin(\gamma)\cos(\chi) + \cos(\theta+\phi)\sin(\chi)]T_{za} + X_{RV} \qquad (12) \\
T_y &= \cos(\gamma)\sin(\chi)T_{xa} \\
&\quad + [\sin(\theta+\phi)\sin(\gamma)\sin(\chi) + \cos(\theta+\phi)\cos(\chi)]T_{ya} \\
&\quad + [\cos(\theta+\phi)\sin(\gamma)\sin(\chi) - \sin(\theta+\phi)\cos(\chi)]T_{za} + Y_{RV} \\
T_z &= -\sin(\gamma)T_{xa} \\
&\quad + \sin(\theta+\phi)\cos(\gamma)T_{ya} \\
&\quad + \cos(\theta+\phi)\cos(\gamma)T_{za} + Z_{RV}
\end{aligned}
$$

where

T_x, T_y, T_z = aim point coordinates (earth fixed)

T_{xa}, T_{ya}, T_{za} = aim point coordinates ("aeroballistic axes")

p, q, r = optimization parameters

γ, θ, χ = MaRV Euler angles (see equation 1)

X_{RV}, Y_{RV}, Z_{RV} = MaRV coordinates (earth fixed)

ϕ = MaRV roll angle (zero when lift is down)

Notice that T_{za} is set to r^2, rather than to r. This prevents the discontinuity from being reached with a negative T_{za} and $q = 0$. Another point to remember is that allowing r and p to both be zero causes the aim point to be co-located at the MaRV, giving an undefined LOS rate.

The smooth surface of $\Delta\phi$ over optimization parameters r and q shown in figure 10 illustrates the improvement of the mapped parameter space over the original parameter space shown in figure 1.

This mapping is a reasonably simple, minor modification, requiring only a handful of additional lines of code to interface between the optimization algorithm and the trajectory simulation. It is of negligible extra computational expense as this sort of transformation between coordinate systems is routinely performed in guidance command updates. Yet as shown in Section 6, the mapping provides a tremendous improvement in convergence performance as discontinuities are removed.

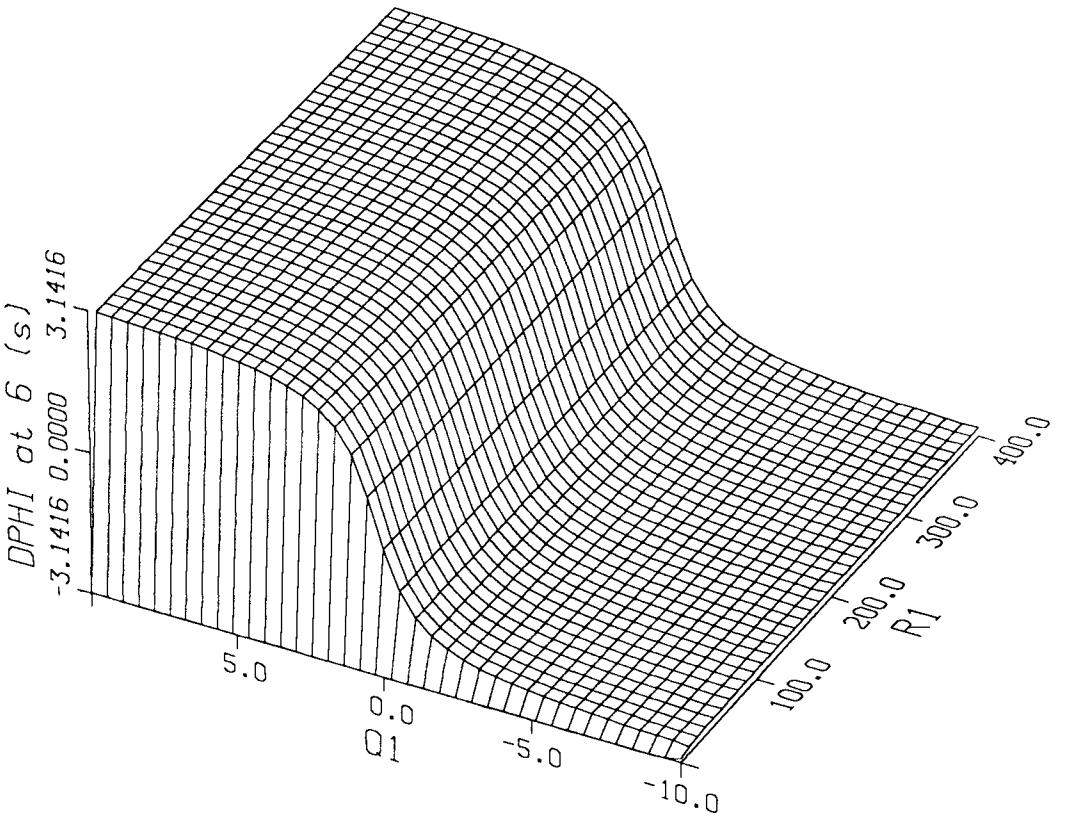

Figure 9: With the parameter mapping, the coordinate system is such that the parameters that cause $\Delta\phi = \pi$ are no longer adjacent to the parameters that cause $\Delta\phi = -\pi$. By contrast, in the unmapped version, figure 2, these regions are adjacent.

5 Other Modifications

For each parameter to be optimized, the user must initialize the optimization algorithm with a "best guess". The only restriction on this choice is that it must satisfy constraint functions, in other words it must be in the feasible set. In the original problem, the choice is easy. Usually, all that is required is to fix the final aim point's coordinates to the true target coordinates and switch to this aim point early enough for guidance to be effective. With the new 'mapped' problem, this is not a simple task, as the r, q, and r that correspond to the true target location depend on the position, velocity, and roll angle of the MaRV at the control switch. The selection could be completed with a pre-processor that computes the r, q, and p corresponding to the true target by integrating the MaRV trajectory up to the final control switch and using the computed MaRV state vector to define the reverse mapping. However such an algorithm was not developed due to time limitations on this study. Instead, a simpler to implement solution was adopted, one that involves a slight modification to the problem.

The solution adopted involves recognizing that as long as the initial position of the MaRV is not fixed, the trajectory can be shifted horizontally by shifting the pierce point horizontally, and by shifting the aim points horizontally. Thus, any trajectory that passes through the ground can be shifted so that miss distance to the target is zero. The ABM can then be flown against the shifted

MaRV trajectory, to compute the cost function. An optimum computed in such a manner can then be implemented by programming the MaRV with the shifted pierce point location and the shifted aim point locations.

For some sets of parameters, the MaRV doesn't even hit the ground. This possibility is accounted for with the constraint handling capabilities of the code by defining the MaRV-target miss distance to be the closest approach of the MaRV to the ground. In this manner, the MaRV-target miss distance constraint forces the trajectory to pass within 200 feet of the ground.

Another modification that was made to the mapped version of the problem was to reset parameter scale factors to be consistent with the range of values of the new parameters. With the old parameters, values in the 10,000's were typical. The new parameters range of values depends on the parameter. Typically, r varies from 0 to 100, q varies from -10 to 10, and p varies from -10,000 to 10,000. The scale factors used for the mapped version are listed in the input files of Appendix A.

6 Results

To indicate optimization algorithm performance, 12 sets of optimization runs were performed for each version of the problem, using different parameter initializations for each run. The versions that are based on the aim point parameters all use the same initial parameter settings. The version that uses mapped parameters uses its own set of initial values that are distributed over roughly the same aim point space. These runs are summarized in tables 1 through 7.

The tables have 8 columns with headings for run number, initial parameter values, final parameter values, miss distances, number of optimization cycles, and most importantly, whether or not the run ended on a discontinuity. This last measure of performance was determined from comparison of forward and backward finite difference derivative calculations, a large difference indicating an abrupt change in the cost function, generally caused by a discontinuity.

Results for the original problem, with no steps taken to smooth the discontinuity, indicate the presence of severe convergence problems, as 11 of the 12 runs hung-up on the discontinuity. Results for the reduced $\Delta\phi$ version described in Section 3.1 are no better. Again 11 of the 12 runs hung-up, this time on "cusps" or discontinuities in first derivatives of the cost function. The commanded acceleration reduction of Section 3.2 did provide significant improvement. For this problem, the optimizer hung-up only 4 of the 12 runs. This modification was

retained in subsequent versions. The LOS rate time lag of Section 3.3 provided a minor, statistically insignificant improvement, with only 3 of 12 runs hanging-up. The hybrid version of Section 3.4 that combines a parameter mapping for the 6 second control switch with a roll direction bias for the 8 second control switch, described in Section 3.5, produced a further minor improvement, hanging-up only 2 of 12 runs.

The problems with the optimizer hanging-up were almost completely eliminated with the mapped version of section 4, in which the optimizer converged successfully 11 of the 12 runs. However, a new type of discontinuity was observed on the unsuccessful run. Like the old discontinuity, the new discontinuity is caused by ambiguities in roll rate. The difference is that the new ambiguity occurs at times other than at control switches, and thus may not be removed with the parameter mapping. The circumstances where the new discontinuity is likely to appear is in trajectories where the LOS to the target is almost perfectly aligned with the velocity vector. A slight acceleration towards alignment causes the velocity vector to pass through the LOS, leaving the velocity vector pointing at the other side of the LOS. The LOS rate reverses thereby commanding a $|\Delta\phi|$ of π, giving rise to the discontinuity. This problem appears to occur less frequently than the roll rate ambiguity at the control switch, but it may be more difficult to remove.

The results for the mapped version of table 6 show significantly degraded evasive maneuvering performance for the optima. With the original version, the runs produce large ABM-MaRV miss distances, typically in the range of 500 to 1000 feet. By contrast, with the mapped version, ABM-MaRV miss distances are low, typically in the range of 20 to 50 feet. The reason for this difference is due to the settings for the final, fixed parameters. With the mapped version of table 6, the third and final setting for q was fixed to 0. (These settings are given in Appendix A.) This means that the MaRV got a final aim point that did not require a change in roll angle, thereby yielding a relatively straight, easy to intercept trajectory. To demonstrate this effect, another set of optimization runs was performed for the mapped version, this time, with the final value of q set to 5. No discontinuities were observed on any of these 12 runs, and in addition, the ABM-MaRV miss distances were large, typically in the range of 1800 to 2100 feet. These results are listed in table 7.

Since the ABM-MaRV miss distance performance is sensitive to the setting of the final, fixed aim point parameters, a direct comparison of evasive maneuvering performance with the original problem is not possible. In general the mapping is likely to slightly reduce evasive maneuvering performance as parameters on the discontinuity can no longer be selected. This restriction turns out to be an advantage as an optimum on the discontinuity cannot be reliably implemented

in any case. A slight perturbation on the trajectory may shift performance to the other side of the discontinuity where performance is poor.

The benefits of the parameter mapping extend to more complex problems. Four runs of a 6 parameter problem were performed, with none of the runs ending on a discontinuity. A brief description of these runs, along with the input files is given in Appendix B.

Table 1: Original Problem (Version 0)

Run No.	X^0	Y^0	$\hat{X}$	$\hat{Y}$	MaRV-target miss distance	ABM-MaRV miss distance	number of cycles	optimizer hangs-up
1	5000	5000	4844.2	15826.5	28.3	599.5	3	yes
2	-5000	5000	-94136.8	14712.0	27.0	591.7	3	no
3	5000	-5000	11950.3	-21650.6	11.8	479.3	7	yes
4	-5000	-5000	-11602.3	16227.1	38.1	673.2	2	yes
5	10000	10000	-2121.0	40069.8	43.8	647.8	4	yes
6	-10000	10000	-14663.8	17645.0	34.8	671.8	2	yes
7	10000	-10000	48444.0	1494.6	0.1	1042.0	2	yes
8	-10000	-10000	-24221.9	-7534.5	97.4	717.1	4	yes
9	10000	0	59099.6	7428.3	0.1	1051.9	2	yes
10	-10000	0	-29486.0	-290.3	61.6	701.8	3	yes
11	0	10000	-172.9	17014.8	36.6	614.0	2	yes
12	0	-10000	1761.6	-22814.6	41.5	462.0	2	yes

Optimization algorithm hangs-up on a discontinuity 11 of the 12 runs.

Table 2: Angle to be Rolled Reduction (Version 1)

Run No.	X^0	Y^0	$\hat{X}$	$\hat{Y}$	MaRV-target miss distance	ABM-MaRV miss distance	number of cycles	optimizer hangs-up
1	5000	5000	-7611.0	17437.2	40.1	649.2	3	yes
2	-5000	5000	-12759.1	8343.8	83.1	608.4	3	yes
3	5000	-5000	-7616.2	-17436.9	40.1	649.3	3	yes
4	-5000	-5000	-12759.0	-8343.9	83.1	608.4	3	yes
5	10000	10000	-296296.4	315582.8	131.4	573.0	4	yes
6	-10000	10000	-13845.5	16282.6	57.8	662.1	4	yes
7	10000	-10000	-296306.6	315594.3	131.5	573.1	4	yes
8	-10000	-10000	-13845.5	-16282.4	57.8	662.1	2	no
9	10000	0	9999.9	-1.1	107.5	1187.1	1	yes
10	-10000	0	10000.7	-402.4	40.0	293.1	2	yes
11	0	10000	218.4	9687.0	73.8	418.7	2	yes
12	0	-10000	-218.4	-9687.0	73.8	418.7	2	yes

Optimization algorithm hangs-up on a discontinuity 11 of the 12 runs.

Table 3: Commanded Lift Reduction (Version 2)

Run No.	X^0	Y^0	$\hat{X}$	$\hat{Y}$	MaRV-target miss distance	ABM-MaRV miss distance	number of cycles	optimizer hangs-up
1	5000	5000	5111.4	22324.3	17.5	485.6	2	no
2	-5000	5000	-15838.8	18814.7	25.1	515.6	2	no
3	5000	-5000	36474.1	-30918.2	0.2	723.5	1	no
4	-5000	-5000	-27068.4	8440.4	51.8	625.7	3	yes
5	10000	10000	-6616.4	29730.1	12.0	541.2	3	no
6	-10000	10000	-32152.9	121736.6	8.1	628.5	7	yes
7	10000	-10000	11865.5	-7161.0	31.3	255.8	2	no
8	-10000	-10000	-30650.0	-8900.1	51.7	628.8	3	yes
9	10000	0	53095.0	16040.4	0.4	1163.5	2	no
10	-10000	0	-34823.5	-5126.7	47.7	605.1	2	no
11	0	10000	1317.6	29541.0	8.5	606.5	2	no
12	0	-10000	-4491.0	-24735.8	20.4	478.7	2	yes

Optimization algorithm hangs-up on a discontinuity 4 of the 12 runs.

Table 4: LOS Rate Time Lag (Version 3)

Run No.	X^0	Y^0	$\hat{X}$	$\hat{Y}$	MaRV-target miss distance	ABM-MaRV miss distance	number of cycles	optimizer hangs-up
1	5000	5000	10775.0	16104.7	35.6	525.4	2	yes
2	-5000	5000	-15180.0	18857.6	36.9	587.1	2	no
3	5000	-5000	7586.8	-24004.2	16.8	468.9	2	no
4	-5000	-5000	-28901.0	6310.6	66.9	685.1	1	no
5	10000	10000	10031.1	10112.0	43.5	420.0	2	yes
6	-10000	10000	-41653.9	9483.7	71.8	664.7	10	no
7	10000	-10000	11510.0	-8274.1	44.2	335.6	2	no
8	-10000	-10000	-29202.5	-9540.7	69.8	696.9	2	yes
9	10000	0	48956.2	20140.5	0.3	1134.3	2	no
10	-10000	0	-34823.5	-5126.7	47.7	605.1	2	no
11	0	10000	4938.1	26513.9	13.8	668.3	4	no
12	0	-10000	-18405.3	-22339.5	40.3	591.1	2	no

Optimization algorithm hangs-up on a discontinuity 3 of the 12 runs.

Table 5: Roll Direction Bias (Version 4)

Run No.	X^0	Y^0	$\hat{X}$	$\hat{Y}$	MaRV-target miss distance	ABM-MaRV miss distance	number of cycles	optimizer hangs-up
1	5000	5000	8495.8	22511.2	16.2	719.4	2	no
2	-5000	5000	-17120.2	18840.3	38.2	590.2	2	no
3	5000	-5000	19231.0	-12489.5	61.2	643.7	2	no
4	-5000	-5000	-10436.7	34978.6	14.8	605.1	1	no
5	10000	10000	7514.0	23193.7	16.2	730.5	10	no
6	-10000	10000	32005.2	119948.6	12.6	673.4	10	no
7	10000	-10000	12408.8	331.1	35.4	470.3	4	yes
8	-10000	-10000	4420.3	27207.4	13.5	668.0	10	no
9	10000	0	11677.8	789.5	35.2	465.3	3	no
10	-10000	0	-29431.7	-9427.4	66.5	698.2	2	yes
11	0	10000	4449.5	26727.6	14.0	667.6	4	no
12	0	-10000	-12403.3	-21882.5	37.5	590.2	2	no

Optimization algorithm hangs-up on a discontinuity 2 of the 12 runs.

Table 6: Parameter Mapping (Version 5)

Run No.	r^0	q^0	$\hat{r}$	$\hat{q}$	MaRV-target miss distance	ABM-MaRV miss distance	number of cycles	optimizer hangs-up
1	84	2	368	-0.478	0.0	107.4	4	no
2	84	-2	367	0.480	0.0	107.3	4	NO
3	84	5	244	-0.452	0.0	81.3	10	yes
4	84	-5	610	564.3	0.0	24.3	3	no
5	119	2	116	1.164	0.0	33.8	3	no
6	119	-2	116	-1.165	0.0	27.2	2	no
7	119	5	62	2.510	0.0	24.3	2	no
8	119	-5	71	-2.510	0.0	24.3	2	no
9	100	0	131	0.000	0.0	23.1	2	no
10	100	3	84	1.838	0.0	25.2	2	no
11	100	-3	84	-1.838	0.0	25.2	2	no
12	100	10	65	9.939	0.0	23.7	1	no

Optimization algorithm hangs-up on a discontinuity 1 of the 12 runs.

Table 7: Parameter Mapping (Version 5)

Run No.	r^0	q^0	$\hat{r}$	$\hat{q}$	MaRV-target miss distance	ABM-MaRV miss distance	number of cycles	optimizer hangs-up
1	84	2	16228	64.9	0.0	2157.0	2	no
2	84	-2	84	-1.98	0.0	1839.2	2	no
3	84	5	35342	-54.1	0.0	1853.0	1	no
4	84	-5	94	-10.8	0.0	2101.9	1	no
5	119	2	53512	1347.7	0.0	2056.1	4	no
6	119	-2	7491	-9484.4	0.0	1971.8	1	no
7	119	5	117	14.7	0.0	2127.4	8	no
8	119	-5	119	-8.6	0.0	2127.5	7	no
9	100	0	112	10.2	0.0	2111.4	10	no
10	100	3	117	14.7	0.0	2127.4	15	no
11	100	-3	97	10.5	0.0	2055.3	1	no
12	100	10	81	7.85	0.0	2137.1	5	no

Optimization algorithm hangs-up on a discontinuity 0 of the 12 runs.

7 Conclusions

After a series of insightful, but unsuccessful attempts were made to reduce the harmful effects on optimization associated with the discontinuity, an innovative approach was found that effectively removed the discontinuity from the optimization problem. Through the parameter mapping, the optimizer can still select from the full set of aim points, except for those precisely at the discontinuity, as those map to infinity. The approach is simple to implement and is effective. The only drawback is that initial parameter selection in the feasible set is more difficult for the mapped version than it is for the unmapped version. This problem may be solved with a pre-processor algorithm that integrates the trajectory up to the final control switch, and does a reverse mapping to compute the r, q, and p that correspond to the true target location.

The other attempts to smooth the discontinuity with guidance modifications were generally unsuccessful as they tended either to change guidance to the point where the simulation no longer had any relevance to the real problem or, as was more often the case, the modifications simply didn't smooth the discontinuity.

References

[1] Dennis, J.E., and Moré, J.J., "Quasi-Newton Methods, Motivation and Theory," SIAM Review, Vol. 19, No. 1, January 1977, pp. 56-60.

[2] Cliff, E.M., Kelley, H.J., Lutze, F.H., Stalford, H.L. "Evasive Maneuvering Modeling and Program Description", Aerospace and Ocean Engineering Dept., Virginia Polytechnic Institute and State University, Blacksburg, VA, December 1986.

[3] Gill, S., "A Process for the Step-by-step Integration of Differential Equations in an Automatic Digital Computing Machine," Proc. Cambridge Phil. Soc. (47), 1951, pp. 96-108.

[4] Gill, P.E. and Murray, W., "The design and implementation of Software for Unconstrained Optimization," in Design and Implementation of Optimization Software, Sijthoff and Noordhoff, Alphen aan der Rijn, Netherlands, 1978.

[5] Kelley, H.J. and Speyer, J.L., "Accelerated Gradient Projection," Lecture Notes in Mathematics 132 Springer-Verlag, Berlin, 1970.

[6] Fletcher, R., "A New Approach to Variable Metric Algorithms," Computer Journal, July 1963.

[7] Kelley, H.J., Cliff, E.M., Lutze, F.H., "Optimization of Evasive Maneuvering", Aerospace and Ocean Engineering Dept., Virginia Polytechnic Institute and State University, Blacksburg, VA, January 1982.

[8] Cliff E.M., Kelley, H.J., Lutze, F.H., "Further Studies of Evasive Maneuvering", Aerospace and Ocean Engineering Dept., Virginia Polytechnic Institute and State University, Blacksburg, VA, September 1982.

A Two Parameter Problem Input Files

This section lists the trajectory simulation data sets and optimization data sets used for this study. Two data sets are used for input to the trajectory simulation through a namelist /TRJ/ read. The first listed is for the original problem and the other versions that use the unmapped aim point coordinates. The second set is used for the version with the parameter. The difference between these files is in the settings for the guidance parameters not selected by the optimizer, RVGDPM(6) through RVGDPM(9).

Input files used by the optimization algorithm, read with namelist /OPT/, are also listed.

```
&TRJ
  NUMINT=1, VALNH = 60000., 9*100000.,
  ILNCH = 4, 9*5,
  XLNCH = 0.0, 9*0.0,
  YLNCH = 0.0, 9*0.0,
  ITYPE =  10,   22,   23, 22*0,
  ITRIG =  20,   10,   10,  22*0,
  VAL   = 0.0,  6.0,  8.0,  22*-1.,
  NC=3,
  XNMAX = 200., XNMAXA= 100., 100.,
  NEXP=2,
  RVGDPM(4)  =    0.0             ,
  RVGDPM(5)  =    0.0             ,
  RVGDPM(6)  =    0.0             ,
  RVGDPM(7)  =    0.0             ,
  RVGDPM(8)  =    0.0             ,
  RVGDPM(9)  =    0.0             ,
  LPR = 3,
&END
&TRJ
  LPR=0,
&END
```

This is the data set read by the trajectory simulation for the unmapped versions of the problem.

```
&TRJ
  NUMINT=1, VALNH = 60000., 9*100000.,
  GAIN  = 4,
  ILNCH = 4, 9*5,
  XLNCH = 0.0, 9*0.0,
  YLNCH = 0.0, 9*0.0,
  ITYPE =  10, 22, 23, 22*0,
  ITRIG =  20,   10,   10,  22*0,
  VAL   = 0.0,  6.0,  8.0,  22*-1.,
  NC=3,
  XNMAX = 200., XNMAXA= 100., 100.,
  NEXP=2,
  RVGDPM(25) =  1.              ,
  RVGDPM(4) =  90.0             ,
  RVGDPM(5) =   0.0             ,
  RVGDPM(6) =  90000.           ,
  RVGDPM(7) =  90.00            ,
  RVGDPM(8) =   0.0             ,
  RVGDPM(9) =  90000.           ,
  LPR = 3,
&END
&TRJ
  LPR=0,
&END
```

This data set is read by the trajectory simulation for the mapped parameter version. The code is designed to treat the first 3 guidance parameters as aim point coordinates. The remaining guidance parameters are treated as mapped parameters, r, q, and p, and are reduced to aim point coordinates with equation set 12. RVGDPM(4), RVGDPM(5) and RVGDPM(6) define values for mapped parameters r, q, and p respectively, implemented at the first control switch. In a similar fashion, RVGDPM(7), RVGDPM(8) and RVGDPM(9) define values for mapped parameters r, q, and p at the second control switch. The units

corresponding to these parameters are feet for p and $\sqrt{\text{feet}}$ for r. The parameter q has no units, but corresponds to the angle-to-be-rolled. For example, $q = 0$ corresponds to the angle 0, $q = 1$ corresponds to the angle $\pi/2$, and $q = \infty$ corresponds to the angle π.

```
 &TRJ
   NUMINT=1, VALNH = 60000., 9*100000.,
   GAIN  = 4,
   ILNCH = 4, 9*5,
   XLNCH = 0.0, 9*0.0,
   YLNCH = 0.0, 9*0.0,
   ITYPE =  10, 22, 23, 22*0,
   ITRIG =  20,   10,   10,  22*0,
   VAL   = 0.0,  6.0,  8.0,  22*-1.,
   NC=3,
   XNMAX = 200., XNMAXA= 100., 100.,
   NEXP=2,
   RVGDPM(25) =  1.              ,
   RVGDPM(4) =  90.0             ,
   RVGDPM(5) =   0.0             ,
   RVGDPM(6) =  90000.           ,
   RVGDPM(7) =  90.00            ,
   RVGDPM(8) =   5.0             ,
   RVGDPM(9) =  90000.           ,
   LPR = 3,
 &END
 &TRJ
   LPR=0,
 &END
```

This data set is used for the runs in table 7. It is identical to the file used in table 6 except that RVGDPM(8) is set to 5.0 rather than 0.0. This variable specifies the setting for q for the final parameter setting. Setting this value to 0.0 causes no change in commanded roll angle at the control switch, thereby yielding poor evasive maneuvers of table 6. Setting this value to 5.0 causes a large change in commanded roll angle, thereby giving the large ABM-MaRV miss distances of table 7.

```
&OPT
  XIN =   5000.,  5000.,23*0.0,
  JPRV = 1, LC=20, IPPR = 4,
  PERT = 2*1.E-4,23*1.E-6,
  XSCALE = 2*10000., 23*1.,
  MEQ=0, MINEQ=1,
  ICONCD = 10, 24*-1,
  TOLMIS = 200.,
  N= 2,
  IVARCD =  104,   105,  23*-1,
&END
```

This is the basic data file used to create the runs for the unmapped versions presented in tables 1-5. The first two elements of array XIN are changed for each entry in the table.

```
 &OPT
   XIN =    100., 0., 23*0.0,
   JPRV = 1, LC=20, IPPR = 4,
   PERT = 3*1.E-6,22*1.E-6,
   XSCALE = 10., 1., 23*1.,
   MEQ=0, MINEQ=1,
   ICONCD = 10, 24*-1,
   TOLMIS = 200.,
   N = 2,
   IVARCD =  104, 105, 23*-1,
 &END
```

This is the basic data file used to create the runs for the mapped version presented in table 6. The first two elements of array XIN are changed for each entry in the table.

B Six Parameter Problem Input Files

To demonstrate that the parameter mapping solution is effective with more complex problems, a set of four, 6 parameter problems were run. These are basically the same problem as the reduced scope problem, except that all 6 of the parameters for the second and third aim points are optimized. The trajectory simulation input file for the 6 parameter problem is identical to that of the mapped 2 parameter problem. The input files for the optimization algorithm are different as they reflect the increased number of parameters. This appendix lists the 4 optimization input files and gives a brief description of each run. The initial values given to r, q, and p for the first control switch are specified with XIN(1), XIN(2), and XIN(3), respectively. The initial values given to r, q, and p for the second control switch are specified with XIN(4), XIN(5), and XIN(6), respectively. These initial values are changed for each run.

```
&OPT
  XIN    = 10.,-5., 90000., 10., 0., 90000.,19*0.0,
  JPRV   = 1, LC=10, IPPR = 4,
  PERT   = 3*1.E-6,22*1.E-6,
  XSCALE = 10., 1., 100000., 10., 1., 100000., 21*1.,
  MEQ    = 0, MINEQ=1,
  ICONCD = 10, 24*-1,
  TOLMIS = 200.,
  N      = 6,
  IVARCD = 104, 105, 106, 107, 108, 109,  19*-1,
&END
```

This run performed 3 optimization cycles to reduce the cost function by over 5 orders of magnitude. The procedure converged to an optimum with an ABM-MaRV miss distance of 2528.8 feet. Overall, 62 trajectory integrations were required.

```
&OPT
  XIN    = 20.,5., 90000., 10.,0., 90000.,19*0.0,
  JPRV   = 1, LC=10, IPPR = 4,
  PERT   = 3*1.E-6,22*1.E-6,
  XSCALE = 10., 1., 100000., 10., 1., 100000., 21*1.,
  MEQ    = 0, MINEQ=1,
  ICONCD = 10, 24*-1,
  TOLMIS = 200.,
  N      = 6,
  IVARCD = 104, 105, 106, 107, 108, 109,  19*-1,
&END
```

This run performed 3 optimization cycles to reduce the cost function by over 4 orders of magnitude. The procedure converged to an optimum with an ABM-MaRV miss distance of 1528.8 feet. Overall, 78 trajectory integrations were required.

```
&OPT
  XIN    = 40.,-2., 90000., 40.,2., 90000.,19*0.0,
  JPRV   = 1, LC=10, IPPR = 4,
  PERT   = 3*1.E-6,22*1.E-6,
  XSCALE = 10., 1., 100000., 10., 1., 100000., 21*1.,
  MEQ    = 0, MINEQ=1,
  ICONCD = 10, 24*-1,
  TOLMIS = 200.,
  N      = 6,
  IVARCD = 104, 105, 106, 107, 108, 109,  19*-1,
&END
```

This run performed 1 optimization cycle to reduce the cost function by over 5 orders of magnitude. The procedure converged to an optimum with an ABM-MaRV miss distance of 1919 feet. Overall, 62 trajectory integrations were required.

```
&OPT
  XIN    = 40.,0., 90000., 40., 0., 90000.,19*0.0,
  JPRV   = 1, LC=10, IPPR = 4,
  PERT   = 3*1.E-6,22*1.E-6,
  XSCALE = 10., 1., 100000., 10., 1., 100000., 21*1.,
  MEQ    = 0, MINEQ=1,
  ICONCD = 10, 24*-1,
  TOLMIS = 200.,
  N      = 6,
  IVARCD = 104, 105, 106, 107, 108, 109,  19*-1,
&END
```

This run performed 1 optimization cycle to reduce the cost function by over 5 orders of magnitude. The procedure converged to an optimum with an ABM-MaRV miss distance of 4030 feet. Overall, 39 trajectory integrations were required.

The vita has been removed from the scanned document