COMPUTER AIDED DESIGN TECHNIQUES APPLIED TO THE DEVELOPMENT OF AN MSAT GROUND STATION ANTENNA

By


Gordon Brent Neilson

A Thesis

Submitted to the Faculty of Graduate Studies

In Partial Fulfillment of the Requirements

for the Degree of Master of Science

University of Manitoba

Department of Electrical Engineering

Winnipeg, Manitoba

Canada

March, 1986

ISBN 0-315-33875-X

COMPUTER AIDED DESIGN TECHNIQUES APPLIED TO THE DEVELOPMENT OF AN MSAT GROUND STATION ANTENNA

BY

GORDON BRENT NEILSON

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ABSTRACT

Manual data entry techniques used during the design of antennas are tedious and error prone. An Interactive Graphics Antenna Design system was created as a tool to enable graphic model creation, automatic conversion to numerical input data, and visual analysis of the calculated electromagnetic characteristics of the modeled antenna.

Manual optimization of the antenna model, involving many design loop iterations, is again a tedious process. Numerical optimization techniques, specifically Rosenbrock's minimization algorithm, are described, and their use in the antenna design environment is investigated.

These two techniques, or tools, are then used to aid in the design of a planar phased array for the MSAT mobile communications environment. This array of crossed drooping dipoles provides good circular polarization and radiation in the region of $\theta = 40^\circ$ to 80°.

ACKNOWLEDGMENTS

I wish to express my sincere thanks and appreciation to the following people, for, without them, this thesis would never have been completed;

To Dr. L. Shafai, for his guidance and technical help throughout every stage of the project.

To Dr. M. Barakat, for his advice and motivation.

To Mike Borgford, a good friend, for the many hours of discussions, arguments, and most of all, inspiration.

To the Manitoba Research Council's Industrial Technology Centre, especially Kerry Thacher and Jeff Allston, for their help, support, and for the use of the Calma CAD/CAM/CAE system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Objective

The intent of this thesis is to develop a system which integrates interactive computer graphics data entry and display technology with numerical analysis and optimization techniques in order to improve the antenna design environment. The design of an MSAT ground station antenna was conducted as a test of the usefulness of this system.

In conjunction with the author, a similar but separate system was developed by M.I. Borgford [1], using a different numerical analysis routine.

1.2 Background

The study and practice of antenna design can be divided into three general approaches or categories according to the methods applied:

1) Manual Techniques

   Traditionally, prototypes are manually constructed, tested, and physically modified to optimize designs. This is a labor intensive, time consuming process, and rarely produces a truly optimum design.

2) Analytic Techniques

   Analytic methods more accurately demonstrate the relationships between antenna

geometry and performance, but are exceedingly complex and restricted to simple geometries. Many of the antenna designers even today use a combination of manual and analytic techniques.

3) Numerical Techniques

Recent advances in numerical analysis techniques have added another method to the two previously mentioned. Complex designs can be analyzed without the time and materials cost of actual construction. In addition, a variety of parameters can be calculated or derived from the calculated data. Numerical analysis routines exist for certain classes of antennas, including conducting wire or surface patch modeled antennas [2], rotationally symmetric conducting antennas [3], and single or dual reflector antennas [4].

Although numerical analysis techniques are a vast improvement over manual and analytic techniques, their use still suffers from tedious and time consuming data entry and analysis tasks. Antenna geometry must be entered as a file of formatted input records indicating the coordinates of key points. Output data is presented as a list of numbers which makes it very difficult to absorb anything more than the major indicators of the performance of the antenna. More recently, plotting routines have been used to display the calculated data more informatively, but these are usually separate and not very flexible.

Optimization of the antenna geometry in most cases is still being done by manually changing the model and re-entering the coordinate data. Many numerical optimization techniques exist, but again they are separate and inflexible and have not found wide acceptance in the antenna design field, with the notable exception given in Reference [5].

All this manual calculation and entry of data is susceptible to error, which can lead to increased design time and lower productivity.

1.3 Graphic Modeling and Display

Graphic modeling allows the user to create an object using actual geometric entities, such as lines, arcs, and splines, instead of numbers. More importantly, graphic display of the model throughout the creation process provides visual feedback for verification, or interaction (hense the term 'interactive graphics').

Interactive graphics provides a means of pictorial communication, "... a medium that is both natural and efficient to human beings, and yet is sufficiently precise for computer manipulation" [6]. Pictorial communication is much more informative than text, and can convey more complex messages easily.

Interactive graphic modeling techniques have become prevalent in many areas of engineering design. In electrical engineering, applications include printed circuit board design [7], integrated circuit design [8], and electromagnetic device (magnets, transformers) design [9]. Interactive graphics are now being recognized as valuable tools for anntenna design as well, and others are working in this area [10].

Interactive graphics can eliminate the errors associated with manual calculation and entry of antenna geometry. Graphic display techniques can increase the value of calculated data, communicating more information to the designer than simple numeric data.

1.4 Scope

There are three major topics involved in the development of the antenna design environment and discussed in this thesis; the IGAD system, the application of numerical optimization techniques, and an antenna that satisfies MSAT mobile ground station requirements.

1.4.1 The IGAD system

The first topic discussed is the development of an interactive graphics pre- and post-processor, tentatively labeled the IGAD system (Interactive Graphics Antenna Design).

This software system integrates the Calma [11] CAD/CAM/CAE software package with a moment method numerical antenna analysis routine. The Calma package is a turn-key three-dimensional interactive graphics modeling and display system which includes a programming environment suitable for the development of graphics application packages such as this IGAD system.

The pre-processor guides the user through visual construction of the antenna geometry, incorporating the intrinsic geometry construction restrictions imposed by the analysis routine. Furthermore, the user can set up a series of excitation, pattern calculation and/or frequency loops for the analysis routine to follow using menu picks and prompts, instead of the usual formatted file records or cards.

The post-processor stores the calculated data from the analysis routine with the antenna geometry in a system model that can be filed and recovered at any time. Futhermore, it can generate pattern plots in a wide variety of ways, and can produce professional quality drawings of these plots and the antenna geometry at any time. Examples of these drawings are included throughout this thesis, while all other drawings were produced using the general Calma software.

1.4.2 Application of Numerical Optimization Techniques

The second topic discussed is the application of numerical optimization techniques in the antenna design environment. Numerical techniques in general are discussed, along with the considerations for an objective function. One method, developed by Rosenbrock [12] uses rotation of search coordinates to perform multi-variant minimization and, along with constraint handling using the "Least P'th" or "Minimax" method [13], is applied to the design situation. Antenna design goals are applied as constraint penalty functions and are combined with geometric constraints to form the objective function.

1.4.3 Design of a MSAT Antenna

The third topic discussed is the detailed design of an antenna to meet the requirements of a mobile ground station for use with the MSAT communications satellite. The MSAT concept includes the provision of a satellite repeater/relay which makes national mobile communications possible. The design goals anticipated for the MSAT satellite dictate the type of antenna possible, and provide the constraints for optimization during the design procedure. The requirements of the Canadian climate and general cost constraints also dictate the types of antennas which can be considered.

1.4.4 Related Discussion

In addition to these main topics, the design procedure is summarized, from the use of the pre-processor to generate preliminary design types, to the application of the optimization routines to produce the best element under the given constraints, to the use of that element in various arrays. The usefulness of the techniques applied in the design procedure are examined and future enhancements discussed.

CHAPTER 2

INTERACTIVE GRAPHICS ANTENNA DESIGN SYSTEM

2.1 Introduction

The three stages associated with the use of numerical techniques in antenna design are:

1) The antenna must be modeled in such a way that the analysis routine can approximate its electromagnetic response to applied voltages, currents or fields.
2) The model must be submitted to the analysis routine, which calculates various specified parameters.
3) The output of the analysis routine must be examined to determine whether the antenna model is accurate and whether the design goals have been met.

The model is then refined, submitted, and analyzed, again and again, until the accuracy of the model is ensured and all design goals have been met. This process is called the "design loop", shown in Figure 2.1, and generally occurs in every type of design situation, not only with antennas.

2.1.1 The Modeling Stage

The modeling stage usually involves sketching the antenna geometry accurately enough to calculate or measure the coordinates of the key locations, and then typing these locations into a file for the analysis routine. This introduces two sources of error, namely,

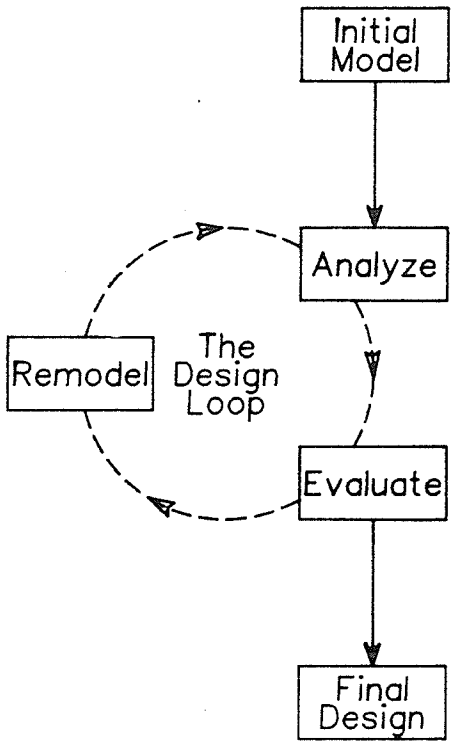


Figure 2.1: The Design Loop

inaccurate calculation of spatial coordinates, and mistakes in entering this numeric data. Since the designer is only dealing with numbers, these errors can go unnoticed for quite some time causing erroneous results and increasing the repetitions of the design loop. As well, manual calculation and input of data is a tedious and time consuming job; one which can easily be automated.

2.1.2 The Analysis Stage

After the modeling stage is completed and the analysis routine has finished its pattern calculation, the designer must then sift through the many pages of numeric output and try to determine if the antenna has been modeled correctly, and how it is reacting to its applied sources. Many errors can occur at this stage, most occurring from the fact that the designer cannot "see" the patterns, but must mentally visualize them from the lists of data. Furthermore, because this mental translation must take place, the designer is only looking for key points or major trends, and may not notice other, more subtle effects due to changes in the model. In many cases, plotting routines are now being used to aid in the analysis of calculated data, but these routines are typically separate from the analysis routine, and are not flexible. A versatile and integrated package of plotting and display routines will improve the speed and quality of the analysis stage of the design cycle.

2.1.3 Rationale

The rationale for developing the Interactive Graphics Antenna Design (IGAD) package is to provide the designer with visual feedback throughout the design loop in order to reduce the time per cycle and the number of cycles necessary to finalize the design, thereby increasing his productivity. Interactive graphics can aid in the modeling stage by eliminating most of the calculations, and by providing errorless translation from the model to the analysis routine input records. Furthermore, the designer can see sources of modeling error before running the analysis routine. Interactive graphics can aid in the analysis stage by plotting the

patterns in a wide variety of ways, performing subsequent calculations when necessary, eliminating the need for mental translation as outlined previously. One additional benefit to using interactive graphics is that more complex antennas can be modeled and analyzed with greater confidence than ever before.

2.1.4 IGAD Composition

The IGAD package breaks down naturally into two parts, a pre-processor for model creation, display and modification, and a post-processor for direct and derived pattern display, pattern plotting, and model dimensioning and plotting. This concept can be applied to any antenna analysis routine, as long as its particular modeling restrictions are incorporated. For the purposes of this thesis, and as an initial prototype system, the analysis routine chosen is based on the numerical solution of integral equations for the currents induced on a structure by applied sources or incident fields. The next section describes the mathematical basis for this routine and the implied modeling restrictions.

2.2 Numerical Antenna Analysis Routine

2.2.1 Mathematical Basis

The numerical antenna analysis routine (NAAR) used in the IGAD system applies a method of moments technique to solve the Electric Field Integral Equation (EFIE). The form of the EFIE used is given below for the electric field of a volume current distribution $\mathbf{J}(r)$,

$$\mathbf{E}(r) = \frac{-j\eta}{4\pi k} \int_V \mathbf{J}(r) \bullet \mathbf{G}(r,r')\, dv', \qquad (2.1)$$

where

$$\mathbf{G}(r,r') = (k^2\mathbf{I} + \nabla\nabla)\, g(r,r')$$

$$g(r,r') = \frac{e^{-jk\,|r-r'|}}{|r-r'|}$$

$$k = \omega\sqrt{\mu_0\varepsilon_0}$$

$$\eta = \sqrt{\mu_0/\varepsilon_0}$$

$\mathbf{I} = (\hat{x}\hat{x} + \hat{y}\hat{y} + \hat{z}\hat{z})$, the identity dyad.

This form can be greatly simplified by restricting the structure to a grid of thin wires of small or vanishing conductor volumes. Under these conditions, the following assumptions can be made:

1) Transverse currents can be neglected relative to axial currents on the wire.
2) The circumferential variation in the axial current can be neglected.
3) The current can be represented by a filament on the wire axis.
4) The boundary condition on the electric field need be enforced in the axial direction only.

Application of these assumptions reduces the above equation to the following scalar equation;

$$-\hat{s}\bullet\mathbf{E}^i(r) = \frac{-j\eta}{4\pi k}\int_L \mathbf{I}(s')\,(k^2\hat{s}\bullet\hat{s}' - \frac{\partial^2}{\partial s\partial s'})\, g(r,r')\, ds', \qquad (2.2)$$

where;

s = the distance parameter along the wire axis,

$I(s) = 2\pi J_S(r)$ = the filamentary current on the wire axis, and the integration is over the length of the wire.

The method of moments [14] is then applied to solve this integral equation for the currents on the wires due to arbitrary excitation. A complete description of the theory involved in this solution is found in Part 1 of "Numerical Electromagnetic Code (NEC) - Method of Moments" [2]. NAAR is a modified subset of NEC, and does not support the surface patch options because of their limited use and excessive restrictions.

2.2.2 Implied Modeling Restrictions

The EFIE method works best with structures whose dimensions are limited to several wavelengths. There is no theoretical size limit, but larger structures require higher order matrix equations, which increases solution time and introduces more and more computation error. Segment lengths should generally be less than 0.1 wavelengths, and the ratio of the segment length to the wire radius should be greater than 8 for errors less than 1% [1, Part III, pp 3-4]. Surfaces can be modeled using wire grids with reasonable accuracy in the far field. Figure 2.2 shows the segment restrictions and the wire segment modeling of surfaces. Figure 2.3 shows a dipole modeled with segments excited by a voltage source across its central segment.

2.2.3 Additional Features of NAAR

NAAR allows the user to apply voltage sources or loads on any segment in the model, or specify an incident field. Ground parameters can also be specified with varying degrees of solution accuracy. Infinite ground planes and free space situations are modeled accurately, but ground cliffs and variable conductivity mediums are not handled well and should be avoided. Wire segment ground planes should be used whenever possible in these

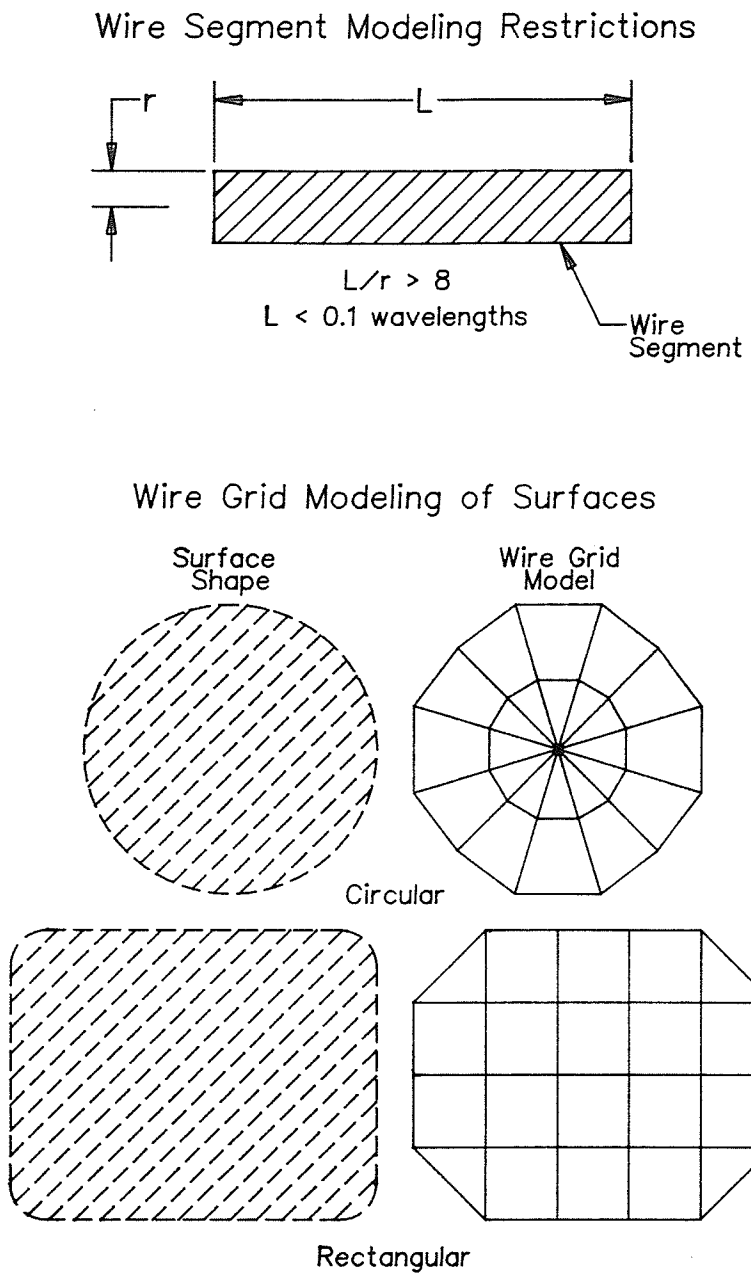


Figure 2.2: Wire Segment Modeling

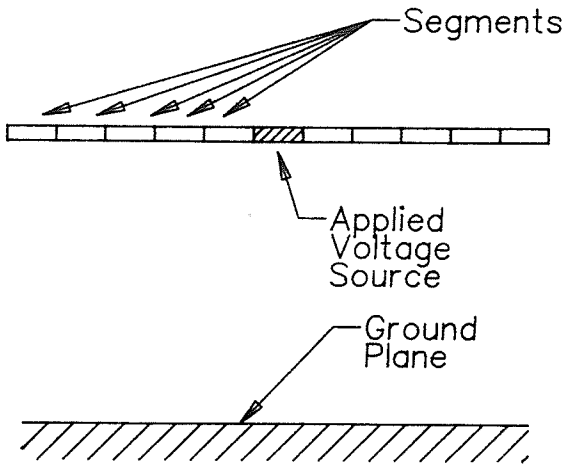


Figure 2.3: Segment Model of a Dipole

cases. Calculated data includes the current magnitude and phase along the structure, the near electric and magnetic fields, the far electric fields, polarization and gain.

2.2.4 Summary

The essential point of this description is that the antenna structure has two basic modeling restrictions. The first is that the antenna must be constructed of a series of short wire segments, whose volume is small enough that all but axial currents can be neglected. The second is that the complete structure must be electrically small, on the order of a maximum of several wavelengths. Large structures need large solution matrices, increasing solution time and reducing solution accuracy. These restrictions have been incorporated in the IGAD pre-processor described in the next section.

2.3 IGAD Pre-Processor

The pre-processing stage of the IGAD package provides an environment for the antenna designer to use interactive graphics to create and edit the antenna model, to specify applied sources and loads, and to indicate desired parameter calculations.

2.3.1 Calma CAD/CAM System

The Manitoba Research Council's Industrial Technology Centre's (ITC) facilities include a Calma CAD/CAM [11] system, consisting of a three-dimensional graphics modeling system, an interactive graphics command language, a FORTRAN-like graphics programming language, and four workstations. The inclusion of the Design Analysis Language (DAL) [15] for graphics programming makes this system particularly suited for the creation of applications packages such as IGAD. The Calma system runs on ITC's VAX 11/780 minicomputer.

2.3.1.1 The Calma Workstation

The Calma workstation, shown in Figure 2.4, consists of an alphanumeric monitor for text and command display, a high resolution monitor for graphics display, a keyboard, and a digitizing tablet for coordinate entry, command selection or item selection. This workstation is also linked to an HP7580 plotter, for producing hard copies of model views, and a printer.

2.3.1.2 Calma Modelling Environment

The Calma software uses a true three-dimensional model database that can be stored and recalled at anytime. This database contains graphic items stored as nodes which are processed through the graphic interface and displayed on the high resolution monitor. The tablet is used to digitize commands from the overlaid menu, or from an on-screen menu, as well as select items or points in three-dimensional space. The keyboard can also be used for command entry, and must be used for text entry. Geometry creation is accomplished by chaining commands and modifiers with locational references. For example, the command sequence to interactively create a line between two points digitized in three dimensional space is as follows,

_ ! _ LBP PDG <DIG> <DIG> C/C

In this sequence, LBP is the command, (**L**ine **B**etween **P**oints), PDG (**P**oint **D**i**G**itized) is a point modifier indicating that the temporary points used as endpoints will be digitized in three dimensional space on the tablet, <DIG> is the screen echo of the tablet digitize, and C/C (Command Complete) initiates the processing of the sequence.

The Calma system supports a wide variety of graphic items including points, lines, arcs, splines, cylinders, Bezier surfaces, notes, labels and dimensions. Non-graphic items are also included in the model database in order to store data arrays or documentation.

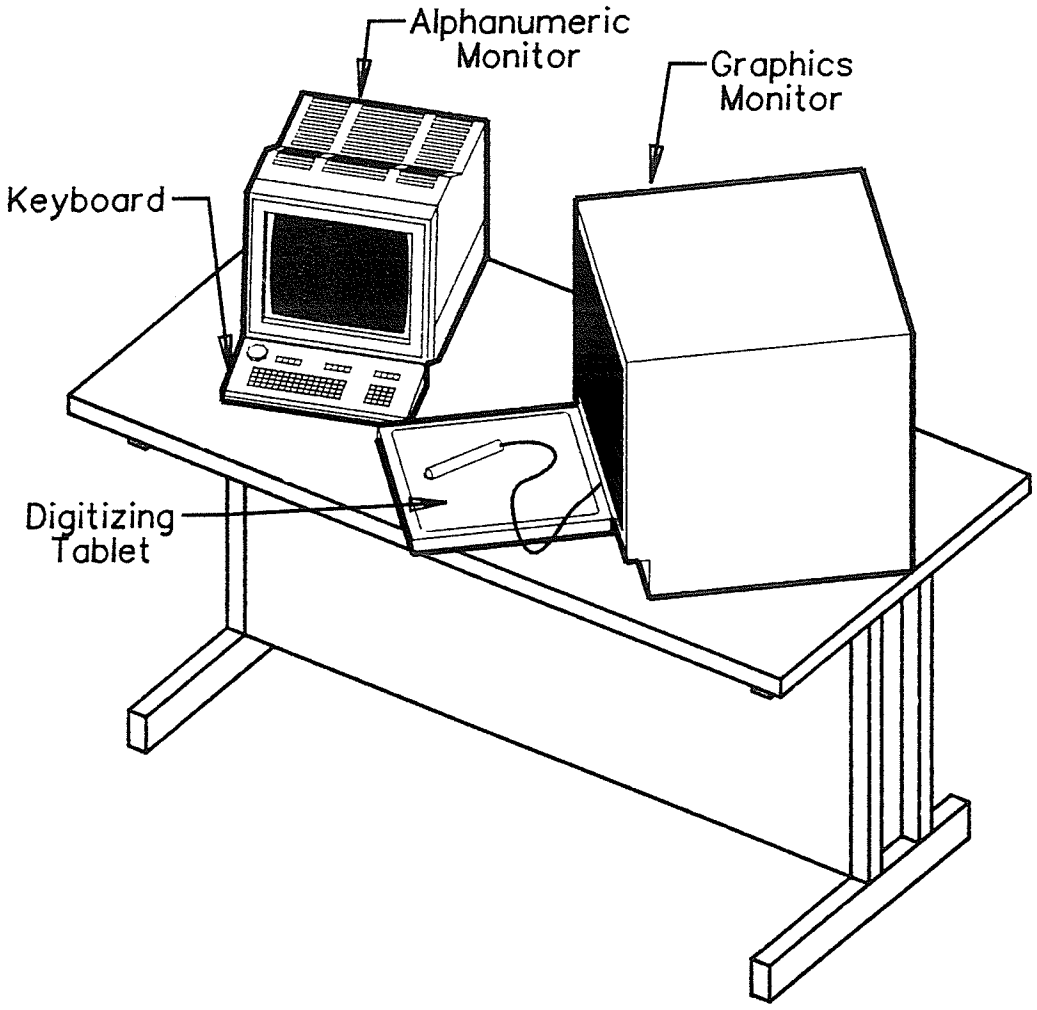


Figure 2.4: The Calma Workstation

2.3.1.3 Model Viewing

The graphics monitor can be partitioned into viewports, each containing a different view of the model. The number and orientation of these viewports can be modified at any time, as well any degree of magnification or panning.

2.3.1.4 DAL Programming Language

A geometric model is built interactively using subsequent creation or editing command sequences. The graphics programming language, DAL, allows pre-progamming of an ordered set of these command sequences to construct complex portions of a model. DAL also supports basic mathematical functions and a variety of user input methods, making it a complete programming language. Calma also supports a DAL-FORTRAN interface [16], allowing the use of faster FORTRAN routines for file I/O and number 'crunching'.

The only drawback of the Calma system is an inherent slowness due mainly to its size, and this can be circumvented to a degree by utilizing the DAL-FORTRAN interface whenever possible. This is particularly noticeable in the area of file I/O where FORTRAN routines run up to 1000 times faster than DAL routines.

2.3.2 IGAD Modeling Method

The analysis routine restricts the modeling of antenna structures to connecting sequences of short wire segments. The most obvious way of implementing this in the IGAD system is as a cylinder. Extensive testing has indicated that this method is slow, because the graphics processor must calculate the two dimensional image of the cylinder's surface, including ends, in each viewport on the screen, which is unnecessary. The only time a true three dimensional segment is necessary is when the designer wants to see the modeling error resulting from segment overlap. Consequently, the wire segments should be constructed out of simple lines with points displayed at each end. Another DAL routine to display the true geometry using cylinders if the designer wishes to examine the modeling error should also be

written. At this time, the cylinder model is still being used in the prototype IGAD system. Figure 2.5 shows the cylinder model, with segment overlap, along with the line and point model.

The restrictions on segment length, and length to radius ratio, are included as warning messages only, which can be by-passed at the user's discretion. In some cases, it is appropriate to exceed the recommended maximums without giving up solution accuracy. The size of the model is restricted to 500 segments, in which case NAAR takes approximately one hour of CPU time to run on the VAX 11/780.

Throughout the modeling process, an array of data is kept in a non-graphic Text-Data item. This array contains twelve items of data for each segment; its tag number, the X,Y,Z coordinates of the first endpoint, the X,Y,Z coordinates of the second endpoint, the wire radius, the real and imaginary parts of the applied voltage source, and the real and imaginary parts of the applied load. This array is also used to determine the connectivity of segments during certain editing procedures.

2.3.3 Interfaces Between NAAR and IGAD

Three basic interfaces exist between NAAR and the IGAD package and are shown in Figure 2.6:

1) The first is the creation of a graphic model from the analysis routine's formatted input records (a NAAR-to-IGAD translator).
2) The second is the creation of the analysis routine's formatted input records from an existing model (an IGAD-to-NAAR translator).
3) The third interface reads in the calculated data and stores it in non-graphic Text-Data items (a data to IGAD translator).

The creation of a graphic model from NAAR input commands is useful in many

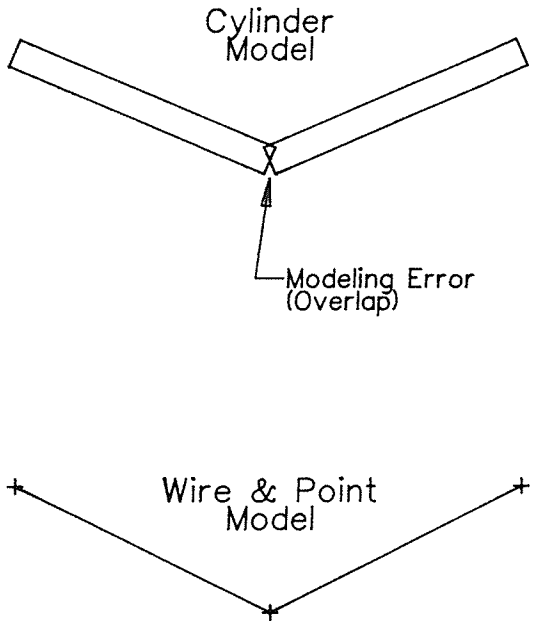


Figure 2.5: IGAD Modeling Methods

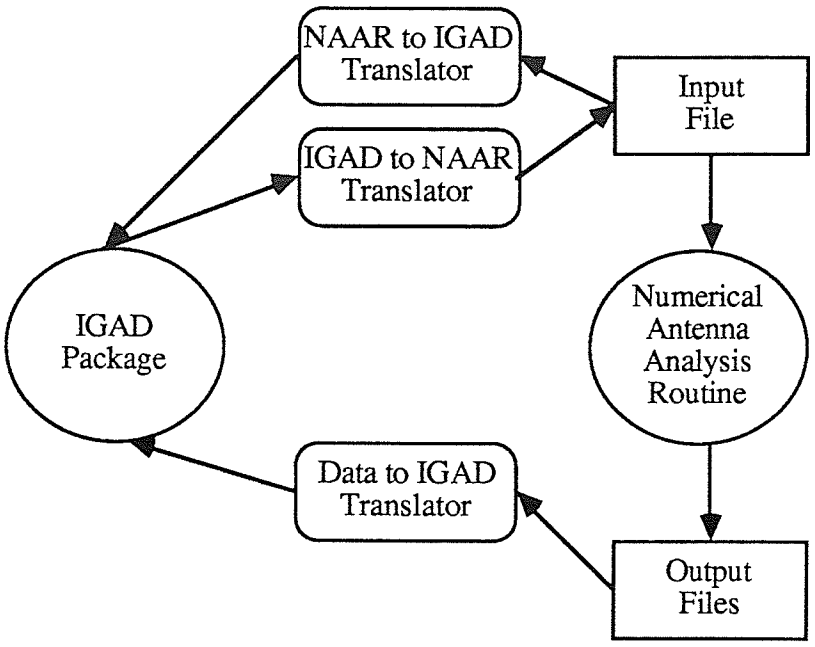


Figure 2.6: Interface Program Structure

cases when the designer wishes to make modifications in the actual input records, because it is easier or quicker than the same changes done graphically, and wants to display the model. This also allows designers to input existing geometry easily for display purposes.

2.3.4 Geometry Construction and Editing Features

The structure of the antenna is created with segments following a series of lines, arcs, and splines. In each case, a separate DAL program has been written which handles the input of construction data, creates the segment as a graphic item in the model database, stores the segment data in the segment array, and ensures that the designer is warned if modeling restrictions are exceeded. A directory of these DAL programs is included in Appendix A. The construction data can be input in a variety of ways:

1. Explicit data entry. The values are entered from the keyboard.
2. Graphically digitized data entry, using the graphics tablet.
   a) Digitize segment endpoints in three dimensional space on one of the viewports.
   b) Digitize one endpoint in three dimensional space, and digitize another existing endpoint to connect the constructed segment to the existing segment.
   c) Digitize endpoints of two existing segments to create a connection between them.
3. Combined digitized and explicit data entry. An endpoint is digitized and an explicit vectored offset is typed in.

The structure of the antenna can be edited in a variety of ways as well. DAL routines have been written to delete segments, and to move, mirror and rotate individual segment endpoints, segments, or groups of segments. Segment connectivity is important for proper

current distribution on the structure and can be maintained during editing if specified.

Other DAL routines exist to create, delete and list voltage sources and loads, to specify ground and frequency parameters and to specify the patterns to calculate. All data entered is stored with the model in non-graphic Text-Data items, which contain standard defaults on system start-up.

Display modification, model storage and model retrieval are all handled by separate DAL routines. The designer does not need to know any of the regular interactive commands, but the IGAD system allows their use as long as the integrity of the model is maintained.

2.3.5 The Design Environment

The designer is presented with an on-screen menu, shown in Figure 2.7, containing all the specialized commands developed for the IGAD pre-processor. With this menu in place, the keyboard need only be used for text entry, or to input explicit interactive commands available to the user who is familiar with Calma. Also included on this menu is the ability to switch between the pre-processor, or construction mode, and the post-processor, or plot mode.

After initializing a new empty model, the designer can create and edit the antenna structure from an input file, or with digitized commands. Once completed, the designer specifies a field source (either a voltage source on a segment, a current filament in three dimensional space, or an incident plane wave). The model can then be submitted to the analysis routine using the standard defaults for ground environment and pattern calculation, or, alternatively, specific ground parameters and pattern calculations can be specified. Additionally, segments can be loaded and frequency loops can be specified. The default frequency is approximately 300 mHz, (1 wavelength = 1 meter). All model dimensions are in meters, but can be scaled to be in wavelengths by using the default frequency, or by using the provided DAL scaling routine.

Once the model has been submitted to NAAR, and the geometry has been saved, the

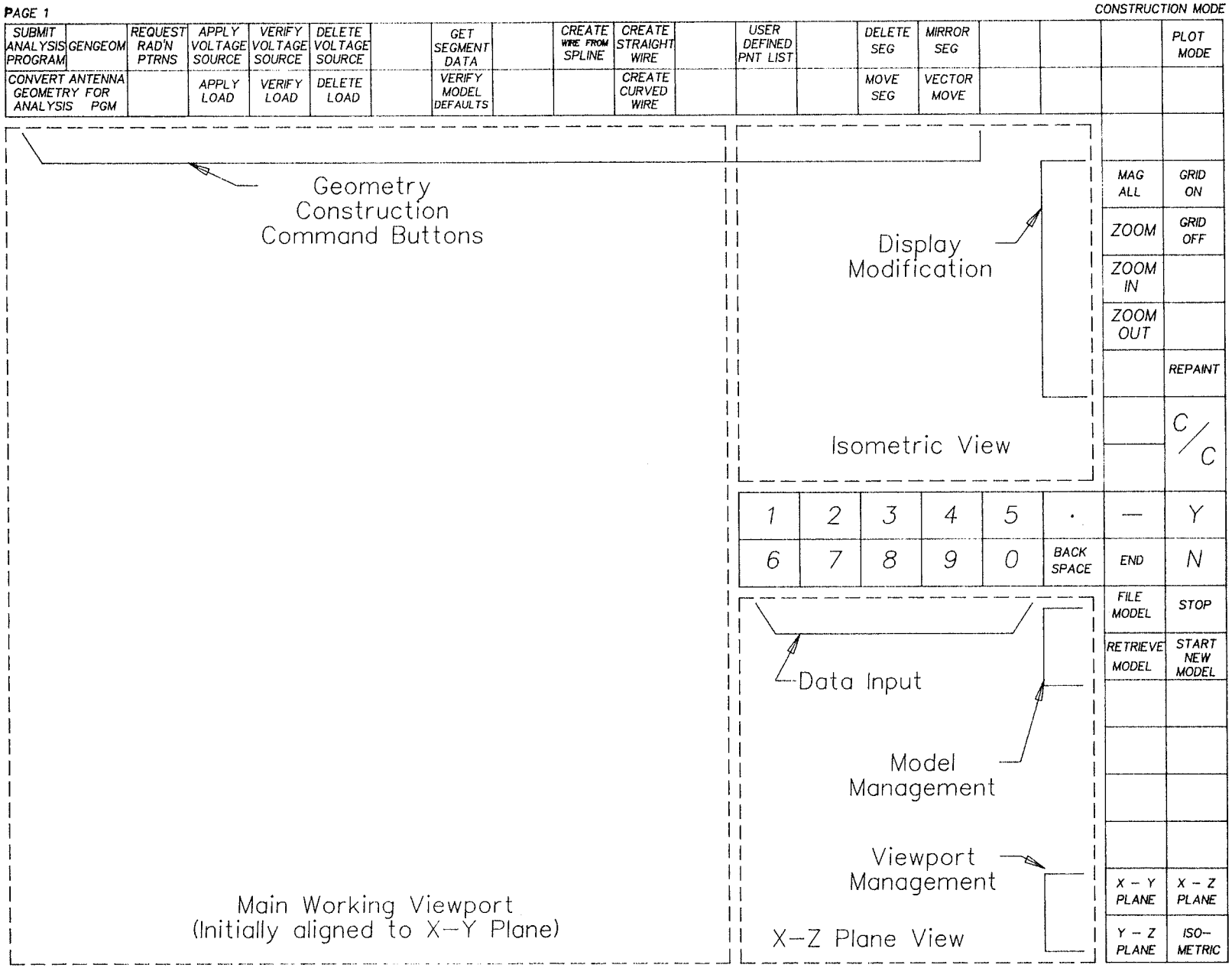


Figure 2.7: Pre–Processor On–Screen Menu

designer then pages the on-screen menu to the post-processor, or plot mode for data display. A completion response is displayed on the alpha-numeric monitor when NAAR has finished and the data can be read in. This design procedure is charted in Figure 2.8.

2.4 IGAD Post-Processor

The post-processing stage of IGAD can read in calculated data, create rectangular or radial plots of the direct or derived data, create displays of one or more plots along with title blocks and antenna geometry, and plot the displays on an HP7580 plotter.

The data calculated from NAAR is stored in formatted files according to their type. IGAD currently supports far-field data, including total gain, ellipticity, and the θ and ϕ components of the electric field magnitudes and phases. From this, the linear or circular polarized components can be derived. Also supported is near-field data and segment currents. The data is read into the model in FORTRAN, and stored in non-graphic Text-Data items.

The designer is again presented with an on-screen menu, shown in Figure 2.9, which contains all commands necessary to plot stored data, dimension the antenna geometry, create displays, and draw these displays on the plotter. All drawings and displays created are stored with the model when it is filed.

In order to create a display, the designer first creates the individual plots using the appropriate DAL commands. Plot programs have been set up for each of the far-field, near-field, and current arrays read into the model from NAAR. Options for each plot include rectangular or radial plots, linear or logarithmic scaling, and data normalization. The designer also has the option of specifying the plot ranges, to examine critical areas or standardize the plots, with the out-of-range data truncated. All plot geometry is stored as a group, and moved using this group name.

After each plot is created, it must be moved out of the plot area and onto one of the drawing areas, otherwise it is deleted when the next plot is created. The drawing areas

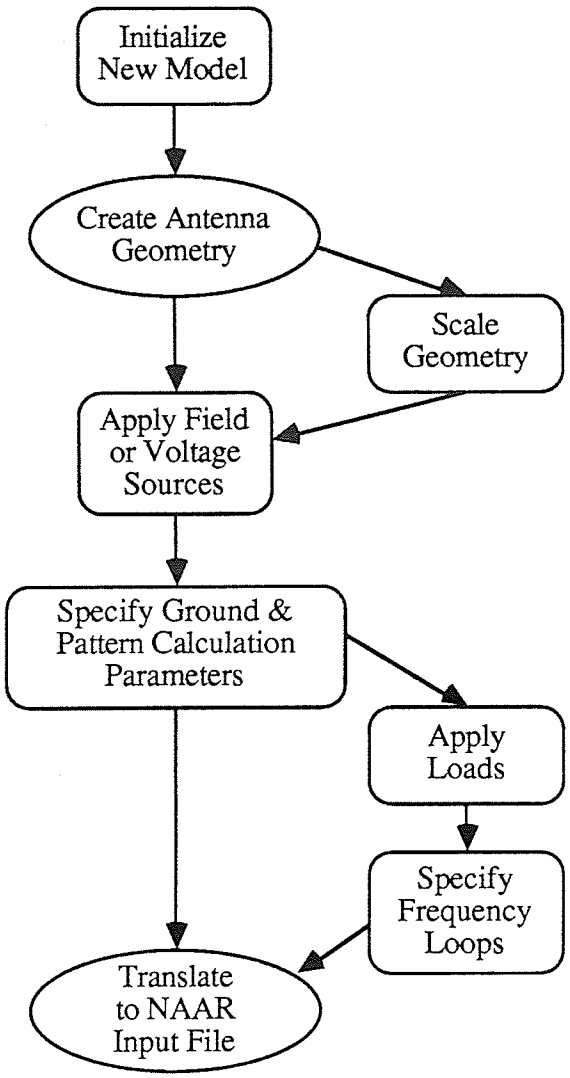


Figure 2.8: Model Creation Procedure

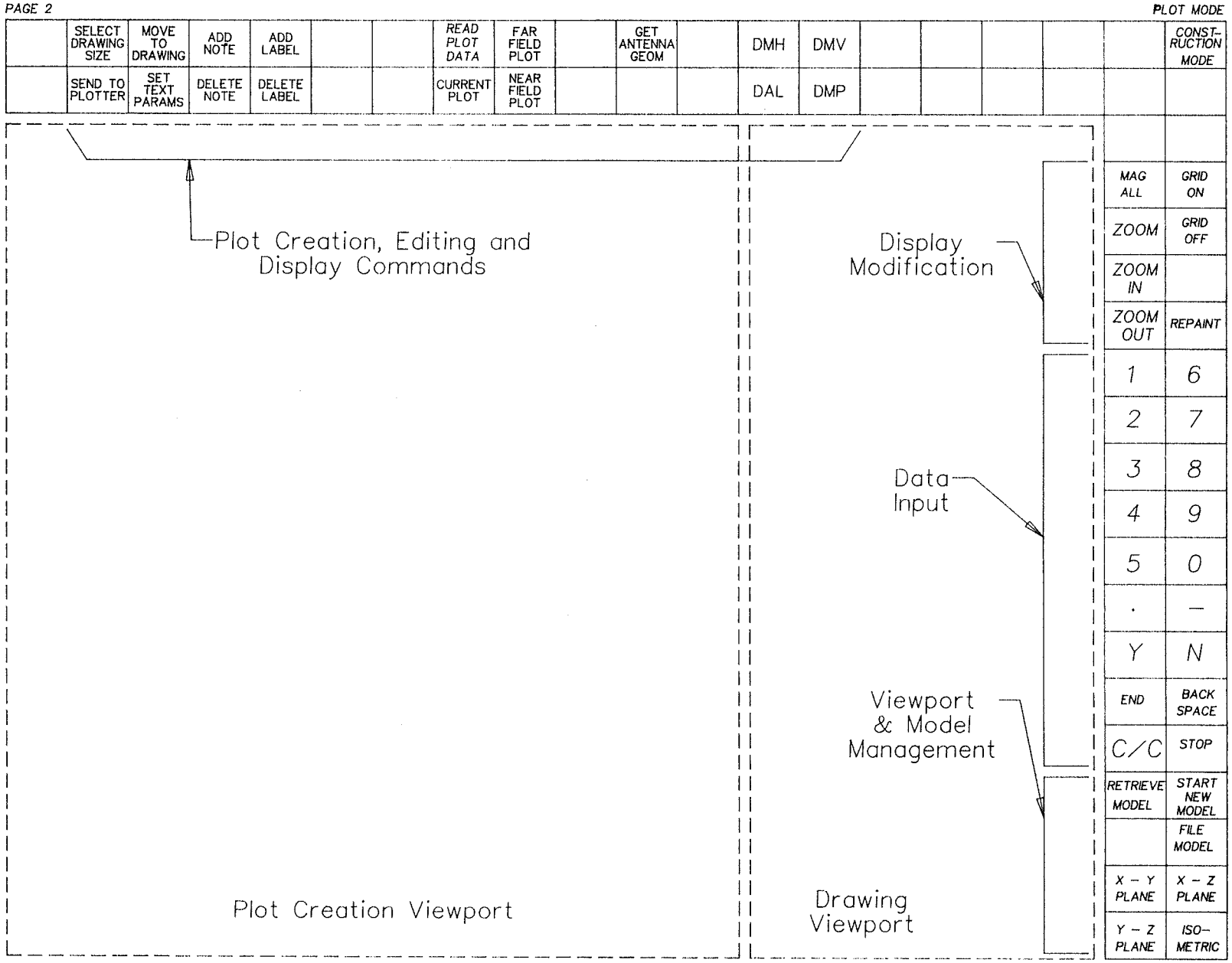


Figure 2.9: Post-Processor On-Screen Menu

correspond to plotter paper sizes and can be filled with as many plots as the designer wants.

The antenna model can be dimensioned and positioned on the drawings as well, in any orientation desired. There also is a title block which can be used for comments or other text display. Figure 2.10 shows a chart of the display creation process, and examples of the plots and displays that can be created are shown in Chapter 4.

2.5 System Evaluation and Future Developments

The IGAD package provides the designer with a good visual environment for antenna model creation and editing, and calculated pattern plotting. The package is very flexible in terms of the ways geometry and data can be displayed and manipulated. The concept itself is applicable to many design situations, and to other antenna analysis routines, while the actual coding and modeling restrictions are routine specific.

The major problem encountered with the prototype system was its inherent slowness, making manual changes to the input file records sometimes easier and faster than graphic model changes. Also the system, as it stands, will only run on an expensive turn-key graphics system, which may restrict its availability to the antenna design industry.

In the future, conversion of many of the routines to FORTRAN, and the elimination of the need for the Calma environment would greatly enhance the packages usefulness and possible marketability. Additional analysis routines could be incorporated, providing a modular package, and further work should be done on the analysis routines themselves, to increase accuracy and reduce the number of modeling restrictions.

The IGAD package could be implemented in a smaller work-station computer, if the analysis routine was segmented, and graphics routines were written to replace the Calma environment. These changes would make the package commercially viable, reducing the cost of the hardware requirements and eliminating external software costs.

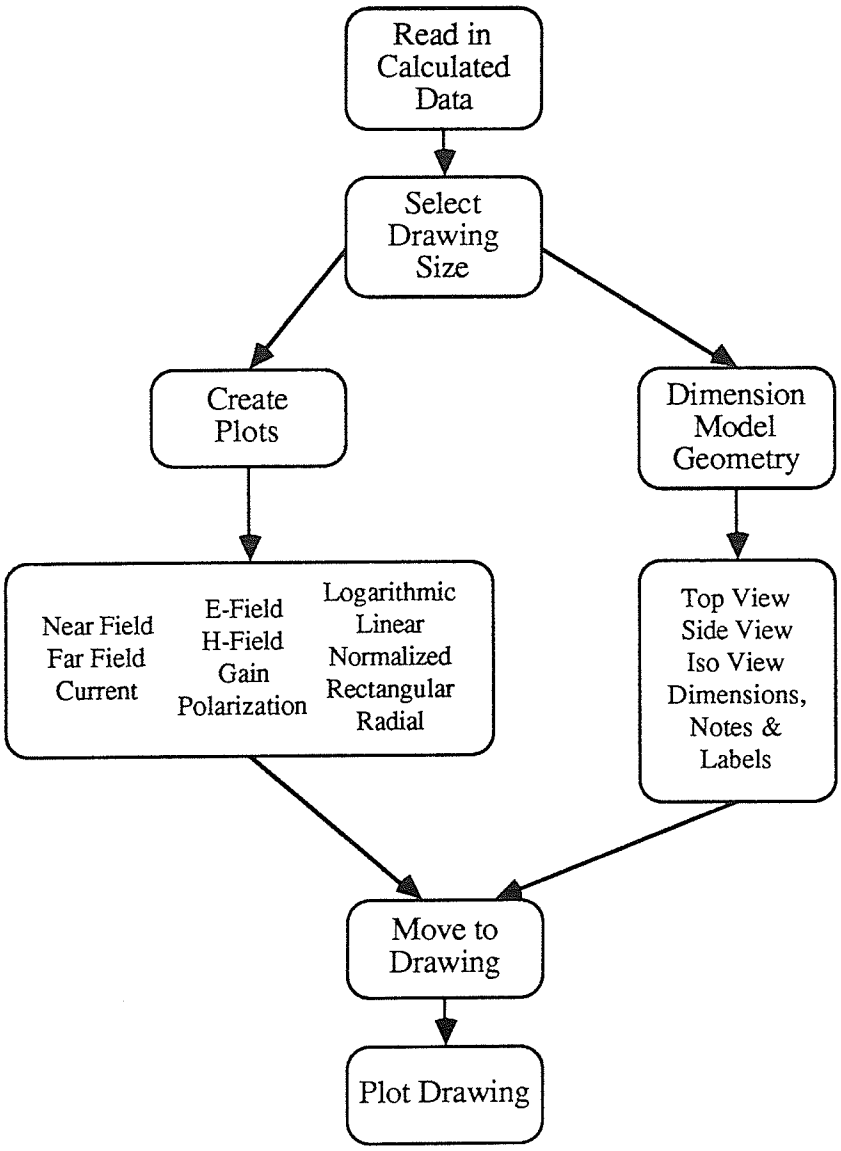


Figure 2.10: Plot Creation Procedure

CHAPTER 3

COMPUTER AIDED OPTIMIZATION

3.1 Introduction

Antenna optimization to satisfy design goals is the most difficult and time consuming phase of the design loop. Usually, if the designer is using a numerical analysis routine, this involves simple adjustment of the model and re-analysis or, if a build-and-test method is used, this involves adjustments or reconstruction of the prototype antenna and re-measurement. Both methods are time consuming and neither produces a true optimum design.

Numerical techniques can free the designer from most of the effort involved in antenna optimization by producing an optimum design without constant intervention, given the analysis routine accuracy and the design goals specified. The designer can also use numerical routines to better understand the operation of the antenna by placing more or less emphasis on certain design goals and optimizing to these 'weighted' goals.

This chapter introduces numerical optimization routines, and describes how one method can be applied to antenna design.

3.2 Numerical Optimization

The concept of numerical optimization refers to the process of determining the minimum value of a function of some design variables, called the objective function, by varying or manipulating those variables [17]. Throughout this discussion, optimization is

synonymous with function minimization since all functions can easily be converted to other functions which must be minimized to meet the design goals.

Included in the optimization process is a set of linear or nonlinear constraint functions which serve to define a feasible region within the variable space, within which, the objective function can be optimized. These constraint functions can be equality constraints, such as,

$$g_a(\Phi) = A_a ; \qquad a = 1,2, \ldots n, \tag{3.1}$$

or inequality constraints, such as,

$$g_b(\Phi) \leq B_b; \qquad b = 1,2, \ldots n, \tag{3.2}$$

Within the feasible region defined by these constraints, there can also be local and global minimums, valleys and saddle points, all defining the contours of the function within the variable space. In order to illustrate these ideas, Figure 3.1 shows the contours of an arbitrary function $\mathbf{U}(\phi_1,\phi_2)$ within the feasible region defined by a set of equality constraints.

3.2.1 Single Variable Techniques

In the case of a function of only one variable, there are many methods to minimize the function, with the speed of minimization generally increasing with complexity. The most simple technique is direct search, where the variable is stepped form one boundary to the next in order to find the function minimum. Another method is the Monte Carlo technique, where a fixed large number of points are chosen at random within the interval, and the function is measured at these points. The minimum measured value is assumed to be the function minimum. A more complex technique is the sequential search, where the function is

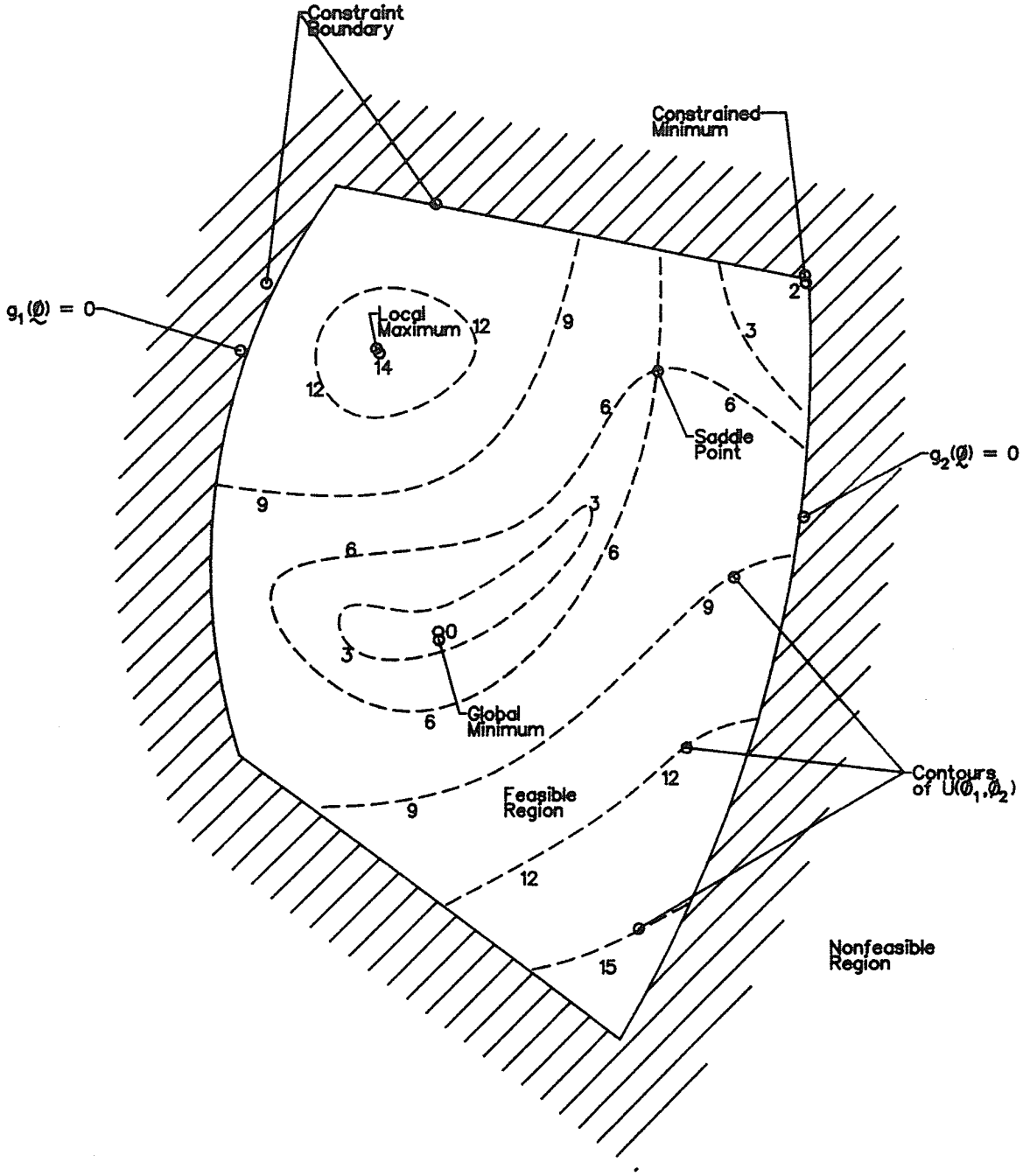


Figure 3.1: The Contours of an Abitrary Function

measured at two points, and a decision is made from these readings as to where to search next. Further points can be determined by moving a specific distance within the interval, measuring the function, and eliminating a portion of the interval. This technique is faster because the search interval decreases due to a decision on the search direction.

3.2.2 Multiple Variable Techniques

Similarly, there are many algorithms for function minimization when there is more than one variable, most of which fall into three categories; direct or vectored searches, simplex contraction searches, and differential or gradient searches. The speed of minimization in these cases depends not only on the complexity of the technique, but also on the suitability of the technique to the particular objective and constraint functions.

3.2.2.1 Direct Search Techniques

Direct techniques search incrementally along as many search vectors as there are variables, as shown in Figure 3.2. In this case, the search begins from some initial point, proceeds to the best minimum along one of the variables, and then goes on to the next variable. This process continues until no progress can be made along any of the search directions. This technique is easy to use, but can be slow if the minimum is not along one of the search vectors from the initial point. One advantage is that direct techniques usually converge to a minimum, which is useful when the contours of the function are unknown.

3.2.2.2 Simplex Search Techniques

Simplex or contraction techniques start off with at least enough initial points to form a simplex in the variable space (at least n + 1 points for n dimensions). The function is evaluated at each end point, and the maximum endpoint is mirrored through the centroid of the remaining points, forming a new, shifted simplex. If the mirrored point is still the maximum, then the simplex is contracted along the line through the centroid. In this way, as

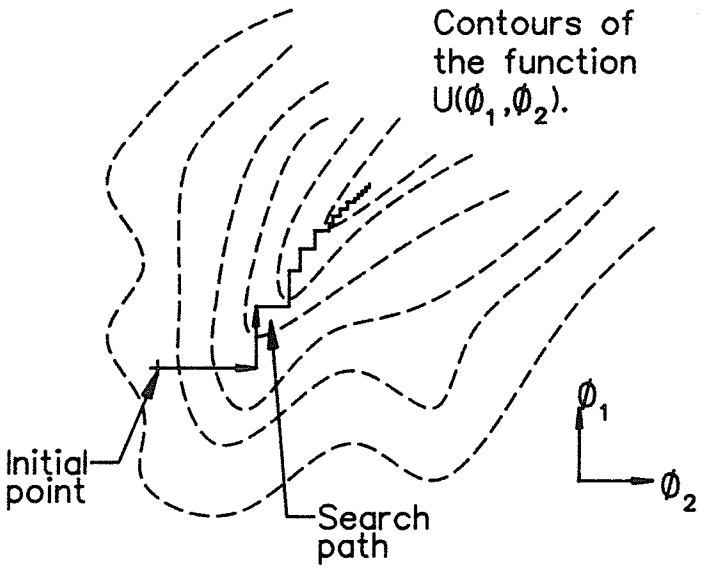


Figure 3.2: Direct Search Algorithm

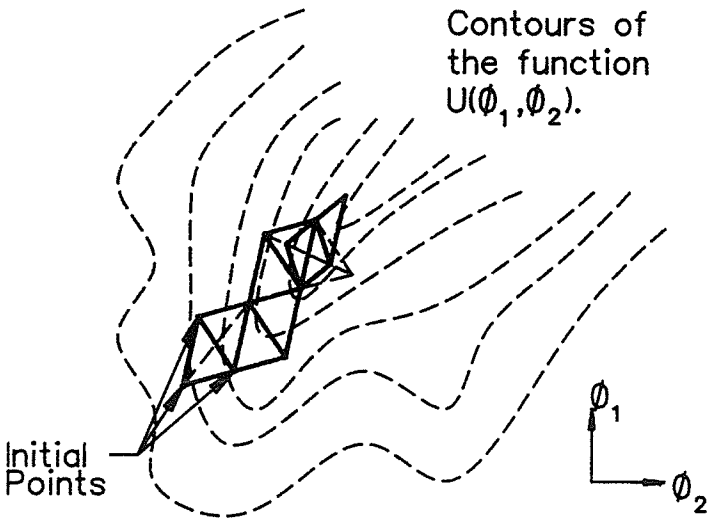


Figure 3.3: Simplex Search Algorithm

illustrated in Figure 3.3, the simplex moves to the minimum, and then contracts in on it.

This type of search is generally faster than the direct search methods, but has two serious problems. The first is that the simplex can contract so far as to eliminate one of the points. In the two dimensional case, the three points would become collinear, and the search would proceed along that line only. Choosing more points than the minimum required can reduce the possibility of this occurring. The second problem is that the simplex can easily oscillate around the minimum, never actually converging.

3.2.2.3 Derivative or Gradient Search Techniques

The gradient search uses the direction of steepest descent of the derivative of the objective function as its only search vector. This is followed to the best minimum, as illustrated in Figure 3.4, and then a new search vector is determined from that point, and so on to the function minimum. This is the fastest of the search techniques, but only if the derivative of the objective function is available. If the derivative must be determined numerically, as it would be in the case of antenna optimization, then the technique becomes much slower and more difficult to implement.

3.3 Rosenbrock's Minimization Algorithm

One special method, developed by Rosenbrock [12], is classified as a direct linear search technique and is based on orthonormal search directions, not necessarily the search variable coordinates, which can be adjusted to follow contour valleys in the objective function.

As illustrated in Figure 3.5, each vector is searched only once, and then a new vector is determined from the direction of overall progress. Gram-Schmidt orthogonalization [12] is then applied to derive the rest of the new set of search directions. This method is faster than simple direct techniques, because it tends to follow the valleys of the objective function. As well, it maintains the advantage of usual convergence.

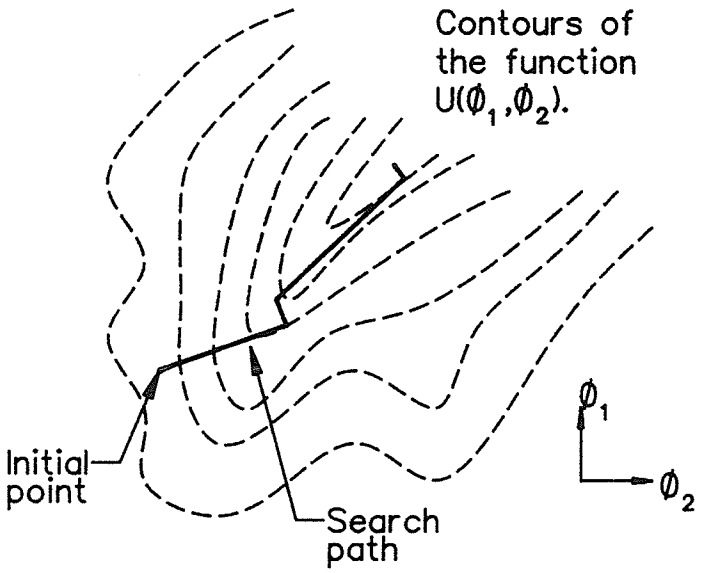


Figure 3.4: Gradient Search Algorithm

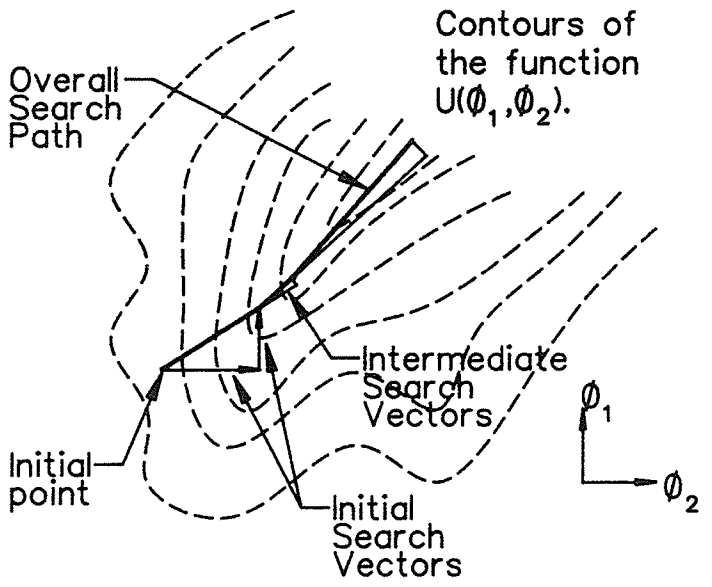


Figure 3.5: Rosenbrock's Search Algorithm

The algorithm was implemented in FORTRAN, and the source code is shown in Appendix B [18]. A more rigorous mathematical description of the algorithm is shown in [19].

3.4 The Objective Function

The preceding discussion uses a generic objective function in the description of the minimization algorithm, implying that this routine will work for any function. This is not generally true, but, if the function is restricted to be continuous, and its first derivative is restricted to be piece-wise continuous, and the function is relatively smooth and slowly changing, then there is a good chance that the algorithm will work.

This idea of 'a good chance of success' is the best that can be said of any minimization algorithm, because there always exists some function, or class of functions, that will cause the particular algorithm to fail. This is why there are many techniques available and care must be taken to ensure that the algorithm used is suitable to the particular objective function.

3.4.1 Constraint Handling

The description of Rosenbrock's algorithm also does not include any discussion of constraints on the objective function, or ensuring that the search remains within the feasible region.

Constraints are handled by incorporating them into the objective function as weighted penalties, which only add to the function value if the constraint is violated [13]. Only continuous penalty functions are used in order to maintain a continuous objective function. Typically, there are two types of penalty functions which are used; the single step penalty, and the two-step penalty.

3.4.1.1 The Single Step Penalty

The single step penalty function is used when you simply want to maximize a value or meet a goal. For example, if you want to maximize the gain of an element, then the single step penalty function would look like;

```
If (Actual Gain < Goal) then
    Gain Penalty = (Goal - Actual Gain) * Weight
End if
```

where 'Goal' is an optimistic estimate of the highest attainable gain, and 'Weight' is a scaling factor designed to adjust the emphasis of this penalty function when it is used in conjunction with other penalty functions

3.4.1.2 The Two-Step Penalty

The two-step penalty function is used when a certain parameter must attain a specified level, but ideally, could be optimized further. For example, if one of the design goals was to maximize the gain, and it had been determined that the gain had to be above a certain minimum value, then the penalty function would look like;

```
If (Gain < Goal) then
    If (Gain < Minimum) then
Step 1 :    Gain Penalty = (Goal - Minimum) * Weight1
Step 2:     Gain Penalty = Gain Penalty + (Minimum - Gain) *Weight2
    Else
        Gain Penalty = (Goal - Gain) * Weight1
    End if
End if
```

where 'Goal' is again an optimistic estimate of the highest attainable gain, 'Minimum' is the required minimum gain, and 'Weight1' and 'Weight2' are scaling factors. In addition, 'Weight2' is much greater than 'Weight1'. In optimization terms, this type of penalty causes more emphasis to be placed on attaining the feasible region (meeting the minimum gain), than on optimizing the gain.

Using these methods, any initial values for optimization variables can be given, not necessarily within the feasible region, and with proper weighting of the penalty functions, the search will minimize the objective function within the feasible region. This also includes the case where no feasible region exists for the given constraints, and the closest approximation is needed.

3.4.2 NAAR as an Objective Function

The numerical analysis routine can be used as an objective function for the optimization algorithm by creating a peripheral routine that converts a few simple scalar variables into the analysis routine input data, and interprets the output with respect to some design goals, returning a simple scalar score.

In order to do this, the designer must know the general form of the antenna to be optimized, and be able to describe it in terms of a small number of variables. Thus, the optimization technique can only be applied once the design is approximately correct, and can not be used as a general design tool to create antennas given the design goals.

One of the main considerations in using NAAR as an object function is the fact that the majority of the optimization time will be taken up in the analysis routine, and thus, the number of function evaluations must be kept to a minimum.

A more detailed discussion about using NAAR in the objective function is given in Section 4.3.2.

3.4.3 Generation of the Function Value

Generally, the function value returned to the optimization routine is a combination of the objective function value and the penalties. Since all the penalty functions must be minimized as well, the objective function is simply considered as another penalty function, and the function value is determined as follows:

$$\text{FVAL} = \text{Maximum Penalty} * \left[\sum_{n}^{i=1} \left(\frac{\text{Penalty(i)}}{\text{Maximum Penalty}} \right)^{p} \right]^{1/p} , \quad (3.3)$$

Where 'p' is ideally infinite, but practically large. This method is called the 'Least P^{th}', or 'Minimax' method and was developed by Bandler and Charlambous [13]. The net effect of using the 'Minimax' method is that the maximum violated constraint is always minimized. Optimization emphasis is implemented through the individual weights applied to each penalty function.

3.5 Summary

Numerical optimization routines generally minimize some objective function of design variables subject to a set of constraints on those variables. Rosenbrock's multivariant, rotation of coordinates algorithm is a direct linear search technique, and is applicable to a wide variety of objective functions. Constraints or design goals can be implemented as penalty functions, which can be incorporated into the objective function using the 'Minimax' technique, eliminating them from external consideration.

Numerical optimization routines, such as Rosenbrock's, can only be applied at the final stages of the design of an antenna, since the antenna must first be describable in terms of a few simple variables. Their use, however, can reduce the time necessary to finalize a design by producing the best possible antenna, given the weighted constraints applied, without constant intervention. As well, the designer can change the emphasis of certain

constraints, giving a better understanding of the performance of the antenna.

This technique is applied in the next chapter to optimize the design of an antenna element for use in the MSAT environment.

CHAPTER 4

DESIGN OF AN MSAT MOBILE ANTENNA

4.1 Introduction

The purpose of creating the IGAD package and describing how numerical optimization routines can be used is to create an improved environment for antenna design. The next logical step is to choose a realistic design situation and exercise this improved environment, since this is the best way to evaluate its usefulness, and to determine any improvements or enhancements which could be added.

The design of an MSAT mobile terminal antenna was chosen as a suitable design situation. This chapter deals with the design of the antenna, and the application of the optimization techniques, while evaluation of the IGAD package is described in the next chapter.

4.2 MSAT Mobile Antenna Requirements

The MSAT communications satellite will provide two-way telephone (point to point) and radio (point to coverage region) communications in the 821 to 870 MHz range, over most of Canada [20]. One application of this is a national mobile communications environment, which introduces the need for a highly specialized antenna. This antenna must be able to send and receive signals from a satellite in geostationary orbit over the equator at any vehicle location or orientation.

At the time this work was initiated, the MSAT satellite was in the development stage,

with only preliminary design goals specified. These specifications were being contested by various factions in the communications industry in Canada, and thus, were not finalized. In order to conduct a realistic antenna design, a fixed set of goals were used, corresponding to the most recent set published at the time. The desired characteristics for the mobile antenna are shown in Table 4.1 [21]:

Parameter	Value
Frequency Range	821 - 896 MHz
Polarization	LH or RH Circular (Switchable)
Scan Volume	$40^\circ \leq \theta \leq 80^\circ$ $0^\circ \leq \phi \leq 360^\circ$
Sidelobe Level	≤ -13 dB
Axial Ratio	≤ 2.5 dB
Peak Directivity ($\theta = 0^\circ$)	16 dBic
Scan Loss (Budgeted)	4 dB
Other Losses (Feed, Phase Shift, etc)	4 dB
Minimum Required Net Gain (At Receiver Terminals)	8 dBic

Table 4.1: Desired Characteristics for MSAT Mobile Terminal Antenna

One important note is that the polarization of the signal is either right or left-hand circular, implying that the antenna must be able to receive either signal, in a switchable configuration. The scan volume is omnidirectional in azimuth (ϕ), to provide for any orientation of the antenna, and is between 40° and 80° from the vertical (θ). The peak

directivity specified (16 dBic, $\theta = 0°$) is for antennas whose main beam is vertical so that enough gain will still be present when the beam is scanned to 60° off the boresite. The important value is the minimum required net gain of 8 dBic throughout the scan volume, after all antenna and feed losses are accounted for (ie. at the receiver terminals).

Other design characteristics which are not related to the performance of the antenna are also important. Obviously, the antenna must be reasonably inexpensive to manufacture in order to make the whole system economically viable. As well, it must be suitable for positioning on mobile vehicles such as cars, trucks, boats, and planes.

4.3 Antenna Element Design

In order to meet the gain specifications described in the previous section, either a mechanically steerable directional antenna or some type of electronically steerable array will be necessary for the final antenna configuration. Mechanical steering is undesirable due to the increased costs, slow response times, and reduced reliability in the harsh Canadian climate. Therefore, only electronically steerable arrays will be considered in this application. Before the array can be designed and analyzed, the individual elements which will be used in the array must be designed.

4.3.1 Survey of Possible Elements

There are many types of antenna elements which can provide the performance requirements of circular polarization and azimuth and elevation coverage. Woo [22], Butterworth and Matt [23], as well as many others, have investigated various elements which will be described next in order to provide some justification for the element type chosen.

4.3.1.1 Spiral Antenna

One classical circularly polarized antenna is the spiral shape. There are various types such as the circular or rectangular Archimedean spiral [24], the log-periodic spiral [25], and

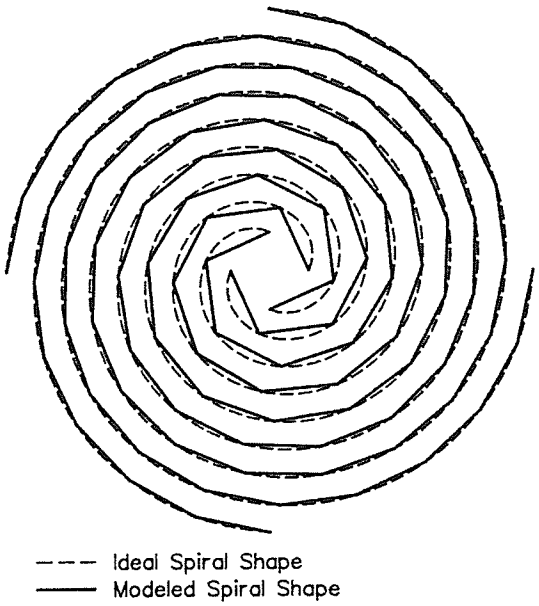


Figure 4.1: Four–Arm Circular Archimedean Spiral

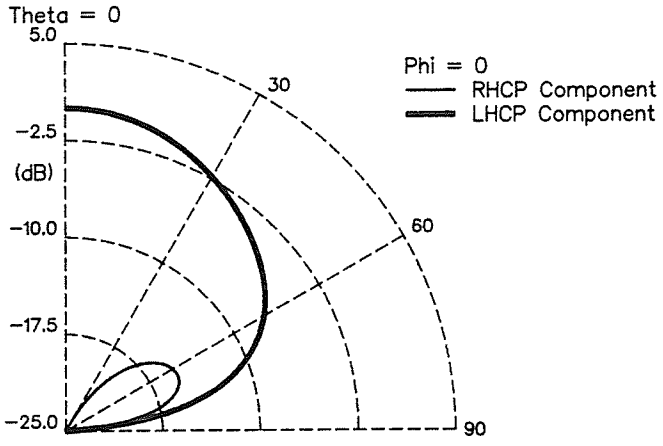


Figure 4.2: Far Field Radiation Characteristics of the Spiral Antenna

the conical log-periodic spiral [26,27]. A circularly polarized signal is radiated by the current propagating along the length of the spiral. Figure 4.1 shows an example of a four-arm circular Archimedean spiral and its radiation characteristics are shown in Figure 4.2.

This class of antennas can be formed from copper foil on a dielectric substrate, as with a microstrip patch antenna, and thus has major advantage in that, except for the conical spiral, they can be made conformal to the surface of the vehicle that they are mounted on.

Unfortunately, in order to switch the polarization from right to left circular, either the spiral must be fed from the end as opposed to the center, or spirals of opposite curvature must be placed in between or under the existing spirals. The first is undesirable because of the unequal impedance and radiation characteristics found when the spiral is fed oppositely, which could make beam steering difficult or impossible. The second is undesirable due to the increased number of components, increased complexity in the beam steering network, and the degradation of the radiation characteristics due to increased coupling or increased separation between elements.

4.3.1.2 Quadrifilar Helix Antenna

Four identical helices are wound equally spaced around a supporting cylinder in this type of antenna, forming a thick 'whip' which can be mounted without the need for a ground plane [28].

This type of antenna radiates an omnidirectional pattern in ϕ, and can be made to radiate at almost any angle θ, by adjusting the pitch, diameter, and number of turns of the helices. Unfortunately, it is very difficult to electronically scan the beam, since all the parameters which the beam's position is dependent on are physical. Thus, this antenna would make a very good stationary antenna, but, without some means of mechanical steering, it is not well suited for mobile applications.

4.3.1.3 Crossed Dipole Antennas

Crossed dipoles fed antiphase can create a circularly polarized signal which can be easily switched from right-sensed to left-sensed by shifting the phase of one of the signals by 180°. This class of antennas can be formed from wire, as in the crossed-drooping dipoles in Figure 4.3, or can be etched on a dielectric substrate again. Crossed-drooping dipoles can not conform to the surface of a vehicle, but give a much better radiated pattern in the region of interest than the etched crossed dipole.

Examples of the radiated patterns for a simple crossed half-wave dipole over an infinite ground plane, and for a crossed-drooping dipole over an infinite ground plane are shown in Figure 4.4. These patterns were calculated using NAAR and displayed using the IGAD post-processor routines.

4.3.1.4 Microstrip Patch Antennas

Much work has been done on creating patches which radiate circularly polarized signals in higher order modes, in order to cause the power to be concentrated in lower elevations as opposed to along the boresite [29,30]. This is accomplished by accurate placing of multiple feeds, with appropriate phase shifting of the applied signal.

Sense switching of the polarization is accomplished by phase shifting again, only in a more complicated manner due to the increased number of feeds. This increase in complexity is multiplied when the patch is used in an array configuration, increasing component costs and reducing overall system reliability. As well, by operating in higher order modes, the patch size becomes greater, increasing the overall size of the antenna.

Regardless of its disadvantages, this type of antenna promises to be the best in terms of radiation characteristics and ability to conform, and undoubtedly will be one of the types ultimately used.

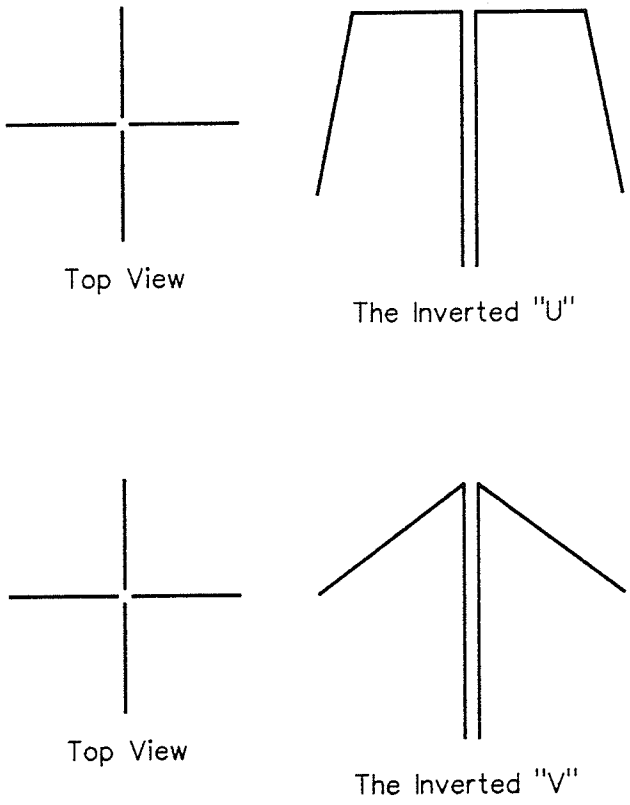


Figure 4.3: Crossed Drooping Dipole Antennas

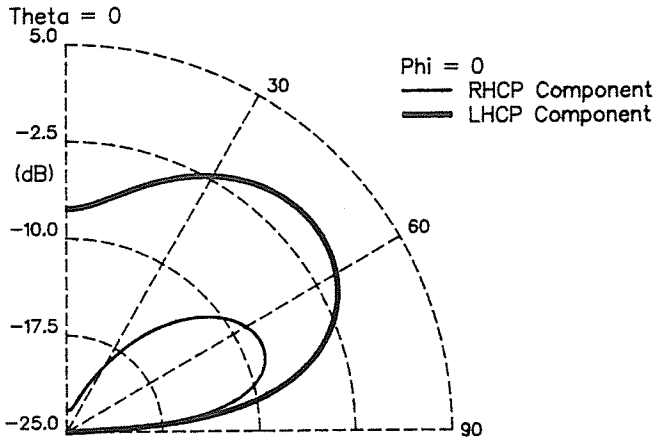


Figure 4.4a: Half-Wave Crossed Dipole

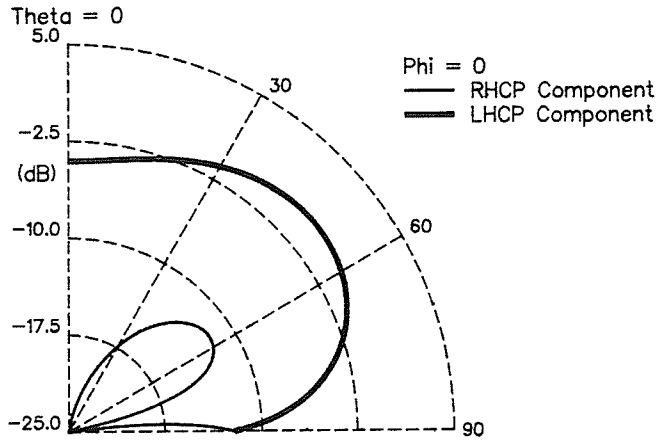


Figure 4.4b: Half-Wave Crossed Drooping Dipole

Figure 4.4: Comparison of Radiated Far Field Patterns of Crossed and Crossed Drooping Dipoles (0.45 Wavelengths above Infinite Ground)

4.3.2 The Crossed-Drooping Dipole

Although it has some disadvantages over other antenna elements, the crossed-drooping dipole (CDD) was chosen as the element for the MSAT antenna array. The CDD is easily modeled in terms of wire segments, and the feeds can be simulated as voltage sources across an intermediate wire. Figure 4.5 shows how the CDD can be modeled in terms of short straight wire segments, as well as the error resulting from segment overlap. This error is negligible as long as the angle of overlap is small, (<15°) and as long as the wire radius is small compared to the length (see Chapter 2).

4.3.2.1 Modeling for Optimization

In order to optimize the element, it first had to be modeled in terms of a few simple design variables. These variables then had to be constrained to make the model consistent with the restrictions imposed by the analysis routine. Two of the many ways to model the CDD are shown in Figure 4.6.

In the first case, each of the symmetric arms is separated into two sections, each section defined by its angle from horizontal, and its length. The height of the CDD above an infinite conducting ground plane is the fifth design variable. One of these variables can actually be eliminated, since, due to impedance considerations, the overall length of each dipole should be approximately one half-wavelength. This simple modeling method, using four design variables, was used during the numerical optimization. Note that this method encompasses both the "Inverted-U" and "Inverted-V" cases described in [22] and [23].

The second method shown would work equally as well, but is more difficult to implement. Each of the symmetric arms is defined by a quadratic polynomial, and an angle of rotation. The fourth variable is, again, the height of the CDD. This method can be enhanced by increasing the order of the polynomial, but each time this introduces another design variable. Many other methods exist, such as segmenting the arms into three or more sections, but these all use more variables, which would increase optimization time.

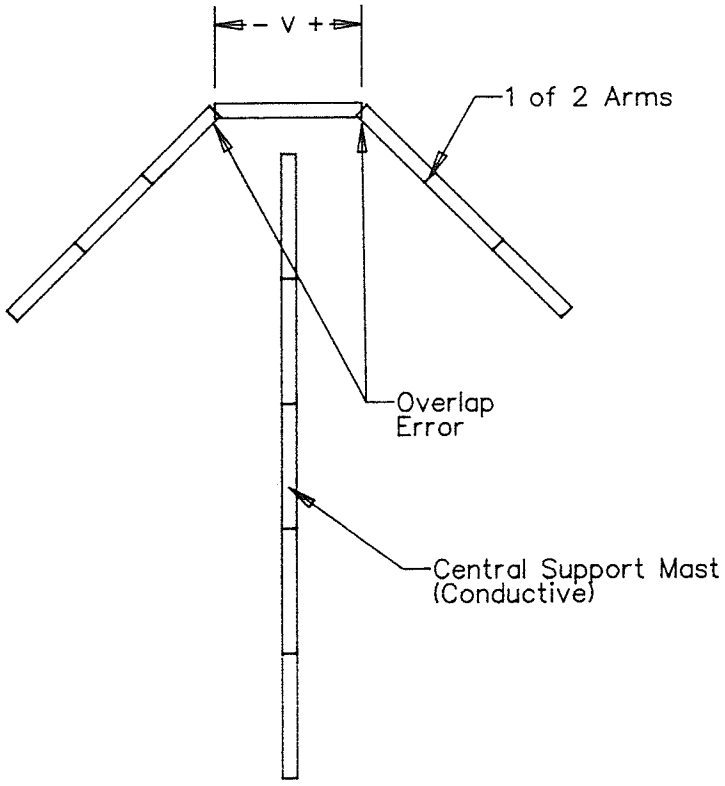
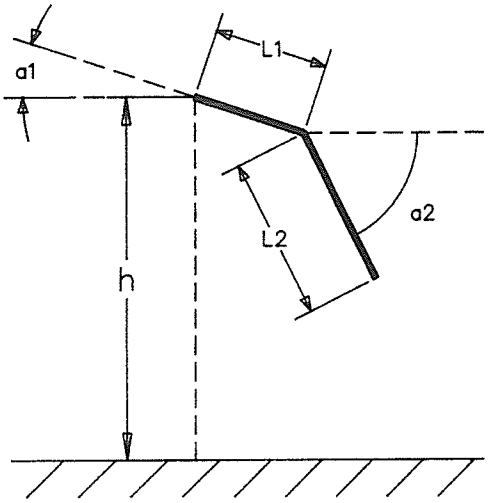
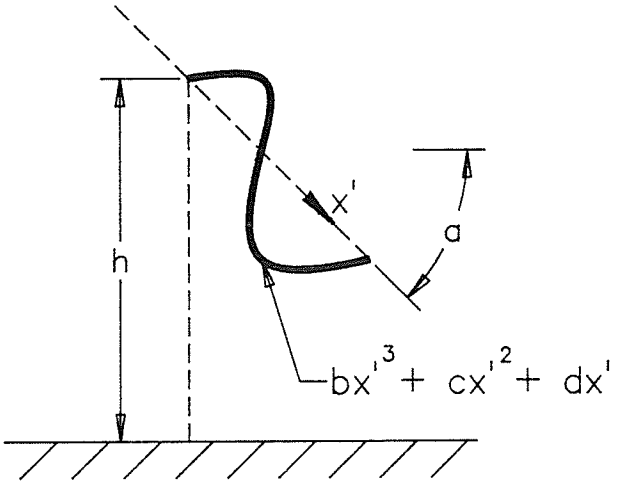


Figure 4.5: Wire Segment Modeling of the Crossed Drooping Dipole



Case 1: Discrete



Case 2: Polynomial

Figure 4.6: Two Ways to Model the CDD With Simple Design Variables

4.3.2.2 Ground Plane Considerations

An infinite conducting ground plane was chosen for the ground environment as a trade-off between calculated pattern accuracy and solution speed. Since the antenna would be situated on the roof of a vehicle, the actual ground plane would be finite. Attempts at creating a custom finite ground plane were successful, but required a much longer solution time for the analysis routine. Using an infinite ground plane is an acceptable substitute if the original finite ground plane is sufficiently large to reflect direct rays from the antenna to the far field region of interest. Error in the calculated far field pattern occurs in the regions closest to the horizon, since the conducting boundary condition forces the tangential (horizontal) components of the field to zero. This is reflected in the plots of the circularly polarized components of the far field by the fact that the left- and right-hand components are equal (ie. linear vertical polarization) at the horizon ($\theta = 90°$).

A routine was written which would generate the appropriate analysis routine input statements, given particular values of the design variables and is shown in entirety in Appendix C. The following sections describe the geometric and design goal constraints, and how they were implemented.

4.3.3 Geometric Constraints

As mentioned earlier, the design variables had to be constrained in order to make the model conform to the restrictions imposed by the analysis routine, as well as to eliminate situations which did not make sense. In this case, this meant that the final height could not be less than zero, (the antenna must be above the ground plane), the final X-coordinate could not be less than zero (the arms cannot overlap the central support mast), and finally, the length of either of the sections could not be zero.

The total vertical and horizontal components of the antenna arm are given by the expressions;

$$VC = L_1 \, Sin(a_1) + L_2 \, Sin(a_2), \tag{4.1}$$

$$HC = L_1 \, Cos(a_1) + L_2 \, Cos(a_2), \tag{4.2}$$

where a_1,a_2,L_1,L_2 are design variables as shown in Figure 4.6.

These constraint equations are used to form penalty functions in the method described in Chapter 3, and shown in Figure 4.7. If any of these geometric constraints are violated, the analysis routine is not executed, and an objective function value is generated using equation 3.7. The weights are adjusted in order to make the function value reasonable so that the search routine can decide which direction to follow in order to eliminate the violated constraints.

4.3.4 Design Constraints

The design goals for the element were limited to ensuring that the beam peak was in the center of the region of interest ($\theta = 60°$), that the half-power points were at the edge of the region ($\theta = 80°, 40°$) and that the cross-polarization was low throughout the radiation region. The gain was not constrained since the element would be placed in an array, and the radiation pattern is more important at this stage.

These design goals were used to form penalty functions using the method described in Chapter 2, and shown for one case in Figure 4.8. Notice that the polarization functions utilize the two-step penalty designed to place more emphasis on bringing the search into the feasible region, (meeting the minimum polarization requirements), than on eliminating the penalty all-together. The weights applied to the penalties were chosen as shown in Table 4.2, in order to place more emphasis on optimizing at the center of the region than at the edges. These penalties were used to generate an objective function value using equation 3.3 which was used by the search routine to optimize the CDD.

```
VC = alen1 * Sin(alpha1) + alen2 * Sin(alpha2)
If (VC.ge. Hite) Then
      Vpen = (VC - Hite) * Vweight
End if

HC = alen1 * Cos(alpha1) + alen2 * Cos(alpha2)
If (HC.le.0.01) Then
      Hpen = (0.01 - HC) * Hweight
End if
```

Figure 4.7: FORTRAN Code Fragment Showing Implementation of Geometric Constraints

```
If (theta(i).eq.60.) Then
      El60pen = 0.0
      If (Ellipticity(i).lt.0.75) El60pen = (0.75 - Ellipticity(i)) * Wel60
      If (Ellipticity(i).lt.1.0) El60pen = El60pen + (1.0 - Ellipticity(i))
                                          * Wel60/10.0
End if
```

Figure 4.8: FORTRAN Code Fragment Showing Implementation of Design Constraints

Parameter		Weight
Gain:	θ = 0°	30.0
	60°	20.0
	90°	20.0
Ellipticity:	θ = 44°	100.0
	48°	150.0
	52°	200.0
	56°	250.0
	60°	300.0
	64°	250.0
	68°	200.0
	72°	150.0
	76°	100.0

Table 4.2: Design Goal Constraint Weights

4.3.5 Optimized Crossed-Drooping Dipole

The optimization routine repeatedly converged to the antenna model shown in Figure 4.9. The peak gain of the element is 4.73 dB, and the calculated circularly polarized components of its far field radiation pattern are shown in Figure 4.10a ($\phi = 0°$, θ varies) and Figure 4.10b ($\theta = 60°$, ϕ varies).

When additional penalty functions were introduced to try to force the power radiated at the boresite ($\theta = 0°$) and at the horizon ($\theta = 90°$) to zero, the optimization routine converged to the antenna model shown in Figure 4.11. By adding these additional constraints, the maximum gain was increased marginally to 4.76 dB, and the gain at $\theta = 0°$ was reduced by 0.63 dB. The calculated circularly polarized components of the far field are shown in Figures 4.12a and 4.12b.

In both cases, many trials were made to ensure that the routine was not centering on a false optimum, using different initial values for the design variables.

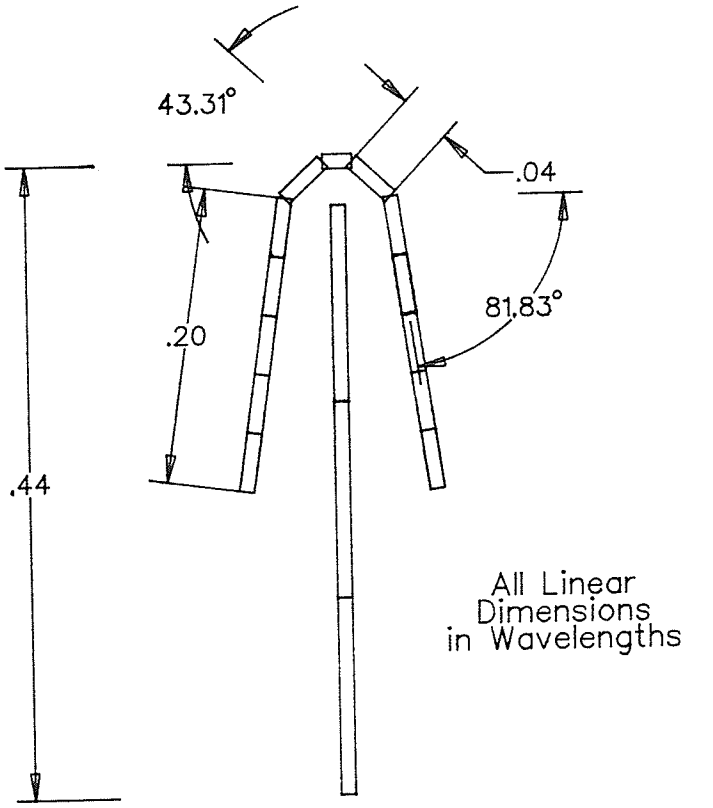


Figure 4.9: Optimized Crossed–Drooping Dipole (Case 1)

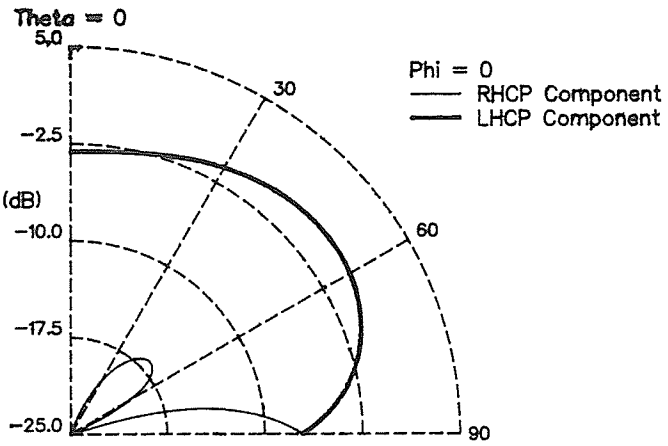


Figure 4.10a: Theta varies, Phi = 0

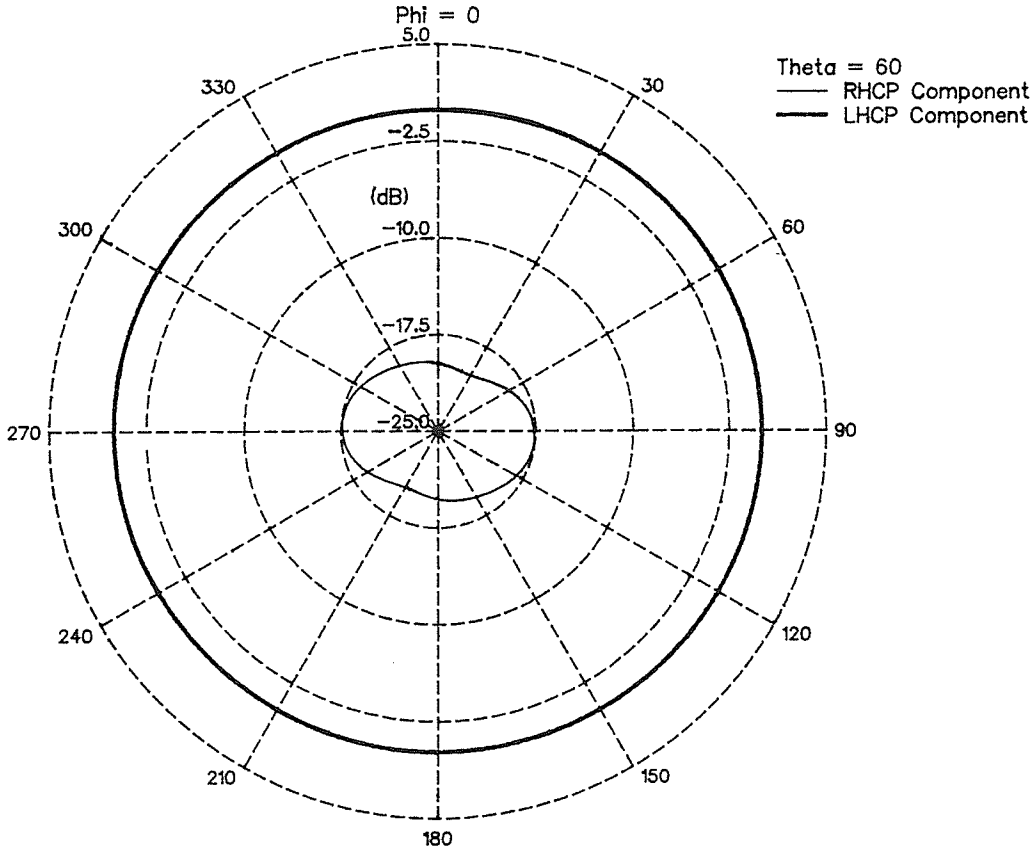


Figure 4.10b: Theta = 60, Phi varies

Figure 4.10: Radiated Field Components for the Optimized CDD (Case 1)

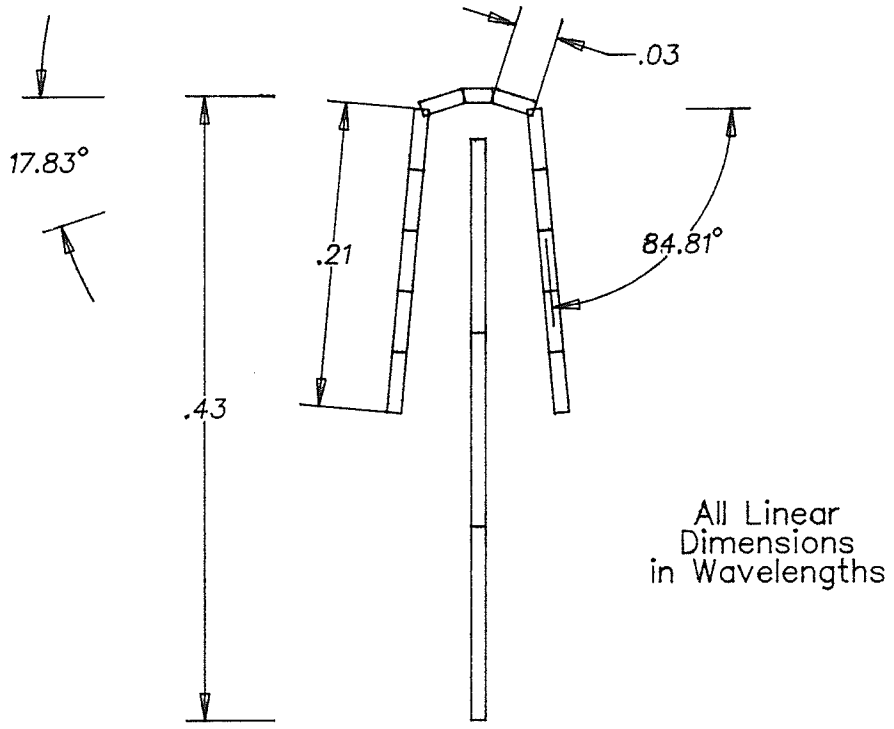


Figure 4.11: Optimized Crossed–Drooping Dipole (Case 2)

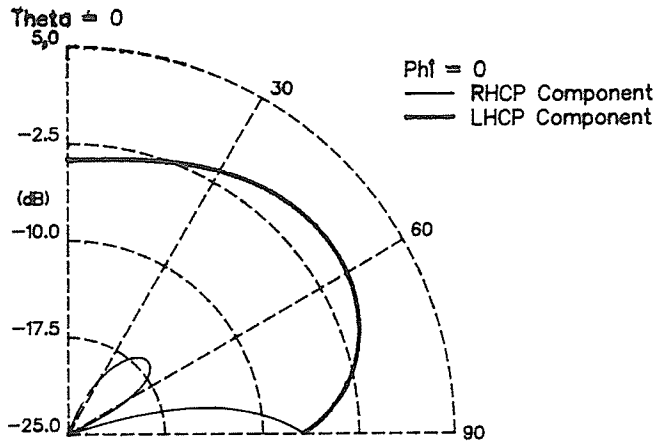


Figure 4.12a: Theta varies, Phi = 0

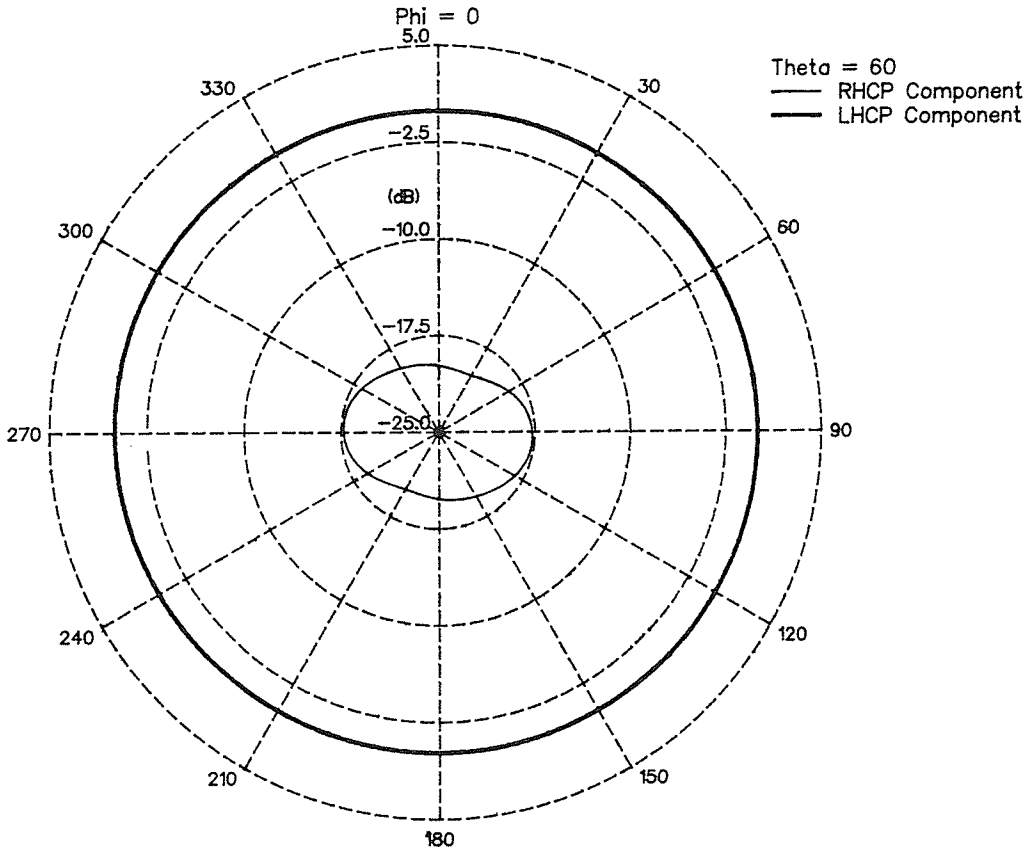


Figure 4.12b: Theta = 60, Phi varies

Figure 4.12: Radiated Field Components for the Optimized CDD (Case 2)

4.4 Antenna Array Design

The CDD can be used in both vertical and horizontal arrays depending on the requirements of the system. The following sections describe these arrays and their applicability to the MSAT design problem.

4.4.1 Vertical Array

A vertical array would eliminate the beam steering capability in the ϕ direction (as with the quadrifilar helix), but would sharpen the beam in the θ direction. The radiation pattern would be omnidirectional in ϕ, and could be steered to almost any angle in θ by adjusting the phase between vertically adjacent elements.

Using eight elements in this vertical array (see Figures 4.13 and 4.14), a gain of 9.53 dB has been calculated, which will not meet the requirements of the MSAT system, but which could be useful if the gain requirement was relaxed.

Note that only the lowest element is of the optimum shape, since one of the optimization variables was the height above the ground plane. As well, more error is introduced by assuming an infinite ground than in the single element case, since the actual finite ground plane will not be large enough to reflect direct rays from the upper elements to the region of interest. However, this is partially offset by the reduced effect of the ground on the contributions from higher elements which is due to the increased distance that reflected and diffracted rays would have to travel.

This type of array has the advantage of a more simple beam steering system, since the satellite does not have to be tracked due to the omnidirectional beam, and since fewer phase shifts between elements are required to obtain full coverage. Unfortunately, this array is not very practical, because it has a high wind resistance, (the eight element array shown in Figure 4.13 stands approximately 1.4 m high), and thus will be likely to deform under various driving conditions, interfering with the desired radiation pattern.

For this reason, and because it cannot meet the gain requirements, the vertical array

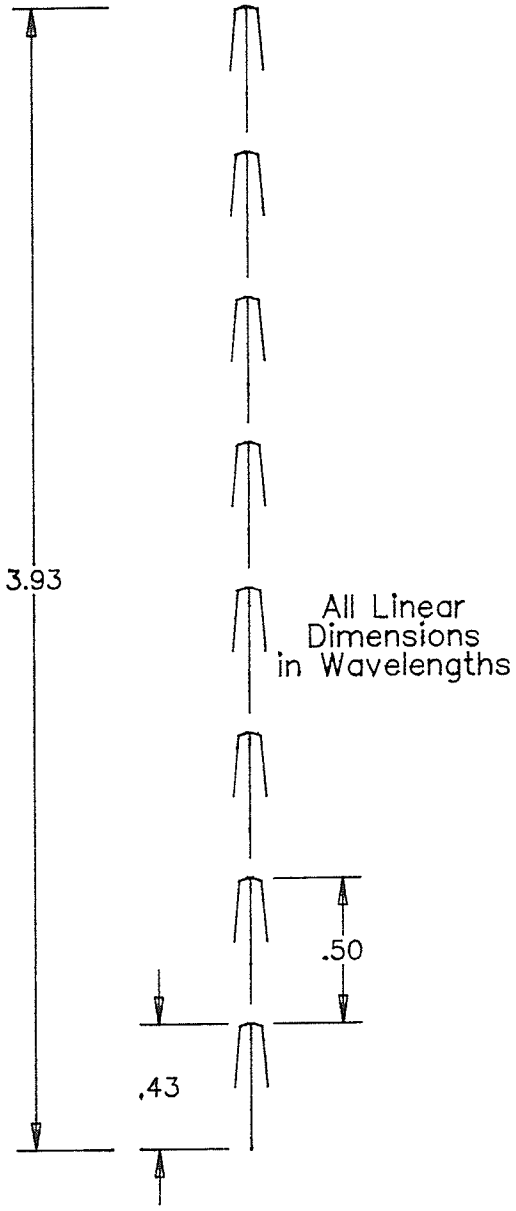


Figure 4.13: Vertical Array of
Crossed Drooping Dipoles

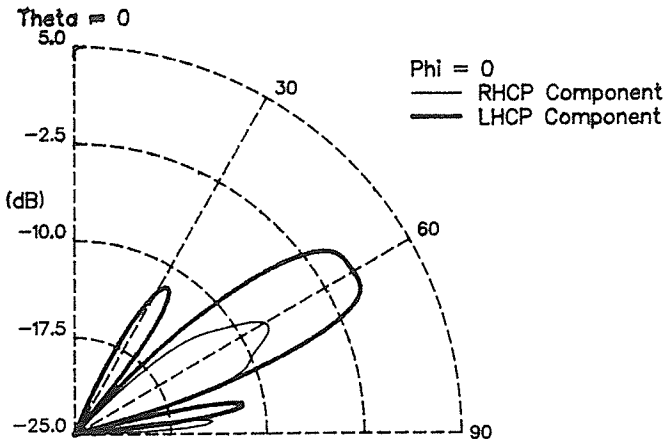


Figure 4.14a: Theta varies, Phi = 0

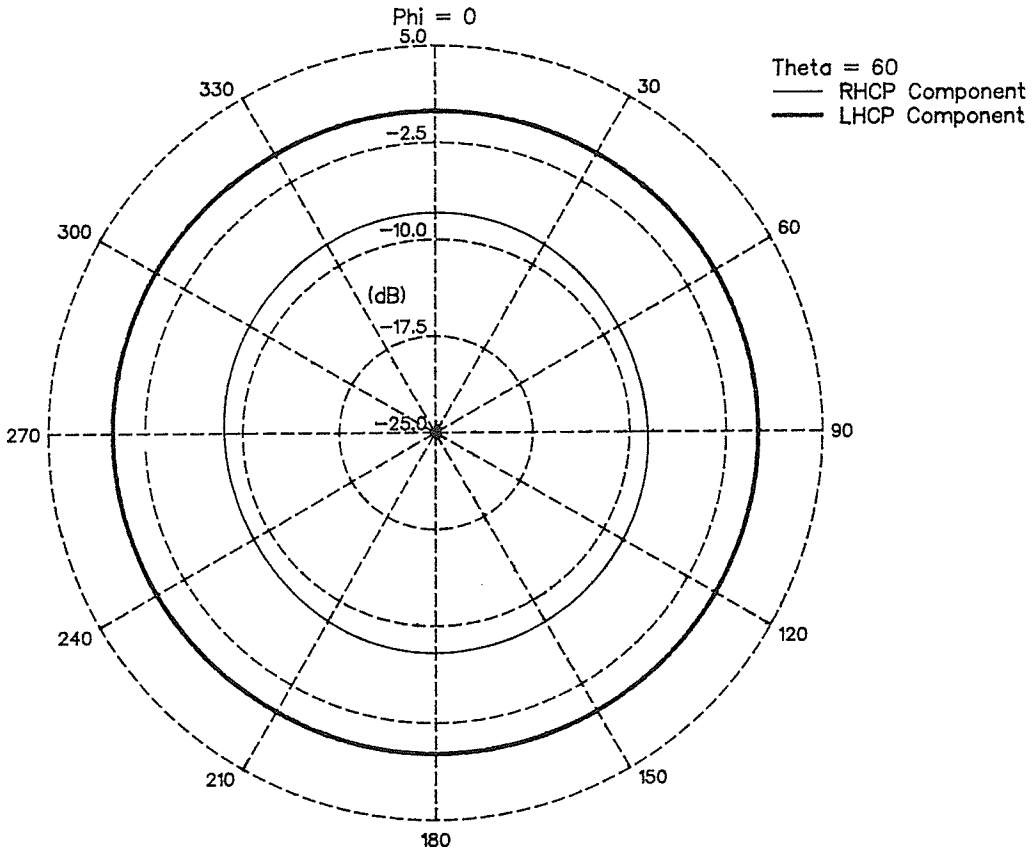


Figure 4.14b: Theta = 60, Phi varies

Figure 4.14: Vertical 8-Element Array Radiated Field Components

will not be considered as a candidate for the MSAT mobile ground station antenna.

4.4.2 Horizontal or Planar Array

There are many configurations for a planar array which can be used to meet the requirements of the MSAT system. The fact that the array must be able to cover a circular region in ϕ suggests that a circular array would be the best choice.

A single ring array, with and without a central element, was investigated. In one case, all elements in the outer ring were excited inphase, and in another case, each element was excited with the appropriate phase shift to steer the beam in a specified direction. A dual ring array was also investigated, with all elements excited inphase. The following sections describe these configurations in more detail.

4.4.2.1 Single Ring, Inphase

A six-element single-ring array was formed by placing elements on a circle at an angular separation of 60°, and is shown in Figure 4.15 (but without the central element). Note that the element orientation was rotated 45° in an attempt to reduce the effects of unequal coupling on the two dipoles of each element. This unequal coupling is caused by different separation distances between the arms of adjacent elements, and rotating the element orientation seems to give improved performance.

The radius of the circle was determined from the fact that, under cophasal excitation conditions, the radiated pattern in θ (ϕ = constant, θ varies from 0° to 90°) has the form of a $J_0(x)$ Bessel function [31]. The maximum beam power can be radiated at a specific angle, θ, if the condition;

$$x_n = ka\ \mathrm{Sin}(\theta), \qquad (4.3)$$

is satisfied, where 'x_n' is the n^{th} zero of the derivative of the $J_0(x)$ Bessel function, and 'a' is

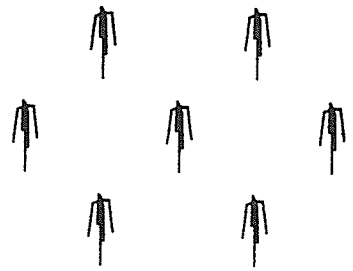


Figure 4.15a: Isometric View

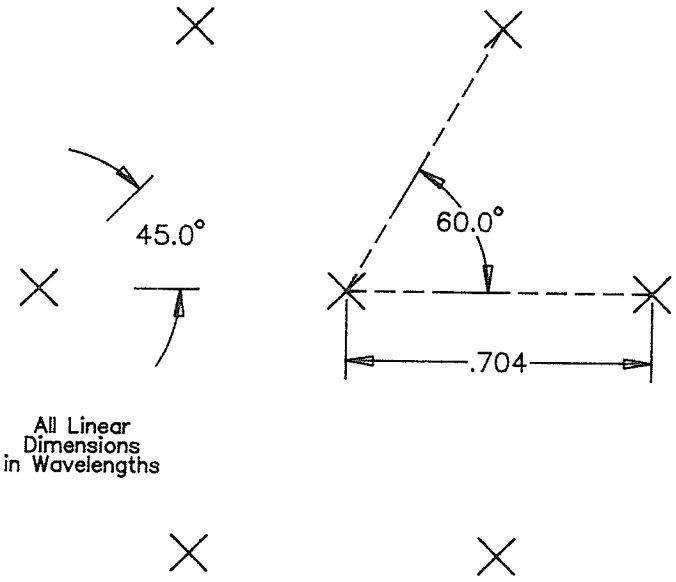


Figure 4.15b: Top View

Figure 4.15: Single Ring Array of CDD's With Central Element

the radius of the ring. The first three zeros of the derivative of the $J_0(x)$ Bessel function occur at x = 0, 3.832, and 7.016 [32]. Thus, in order to steer the main beam to $\theta = 60^\circ$, and using the second zero (3.832), the radius of the first ring was determined to be 0.704 λ. Unfortunately, $\theta = 0^\circ$ is also a solution to equation 4.3 for the first zero (x = 0), regardless of the radius of the ring, indicating that the dominant 1st order mode will also radiate along the boresite.

With all elements excited inphase, the resulting far field pattern is omnidirectional in ϕ. In order to scan the beam in θ, the radius of the array ring must be varied. This would suggest that inphase excitation is not desirable in a single ring array. However, the use of inphase excitation is desirable because it simplifies the feed system, since no inter-element phase shifting is required. Inphase excitation can be used in a multi-ring circular array, because the phase shift between rings of elements can be adjusted to steer the beam to any angle of θ desired.

Figure 4.16 shows the circularly polarized components of the far field of a single-ring six-element circular array of CDD's, all excited inphase. There is no central element in this case, and the ring radius is 0.704 λ. There is a strong lobe at $\theta = 60^\circ$, as expected, with very low cross-polarization, as well as the dominant 1st order mode along the boresite ($\theta = 0^\circ$).

The horizontal pattern ($\theta = 60^\circ$, ϕ varies) shows a 3.6 dB drop in gain and increased cross-polarization on inter-element radials. These are a result of the spacing between elements being too large, and can be rectified by increasing the number of elements in the ring. An eight-element ring provided a virtually omnidirectional pattern in ϕ.

Addition of the central element did not improve the far field radiation characteristics of the array, as had been expected, and actually degraded its performance. Figure 4.17 shows the circularly polarized components of the far field when the central element is excited 180° out of phase (antiphase) with respect to the elements on the outer ring. Figure 4.18 shows the results when the central element is fed inphase with the other elements.

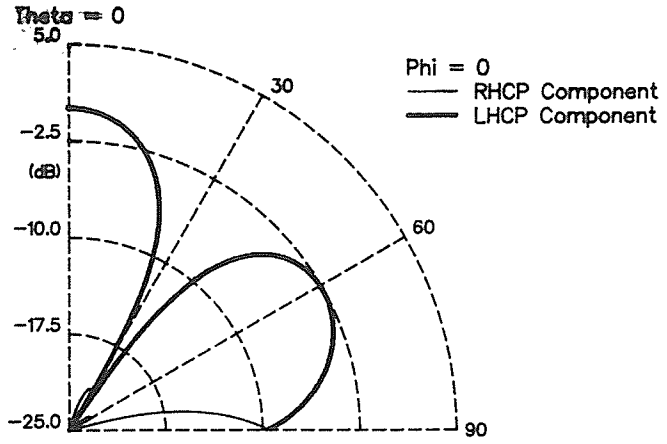


Figure 4.16a: Theta varies, Phi = 0

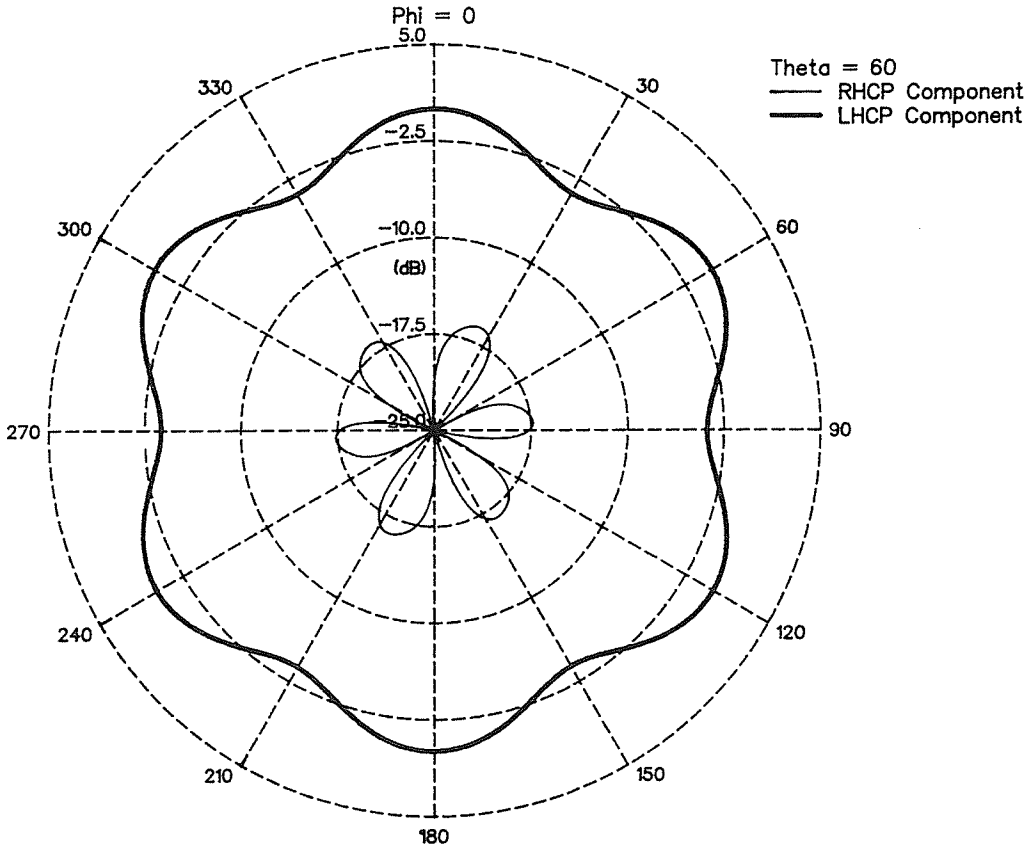


Figure 4.16b: Theta = 60, Phi varies

Figure 4.16: Planar 6-Element Array
Radiated Field Components
All Elements Excited In-Phase

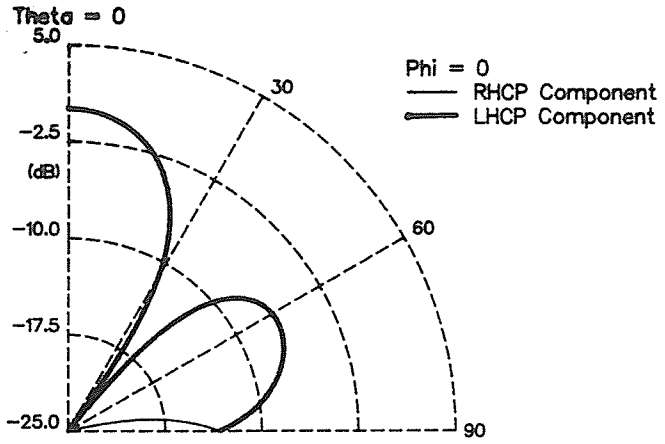


Figure 4.17a: Theta varies, Phi = 0

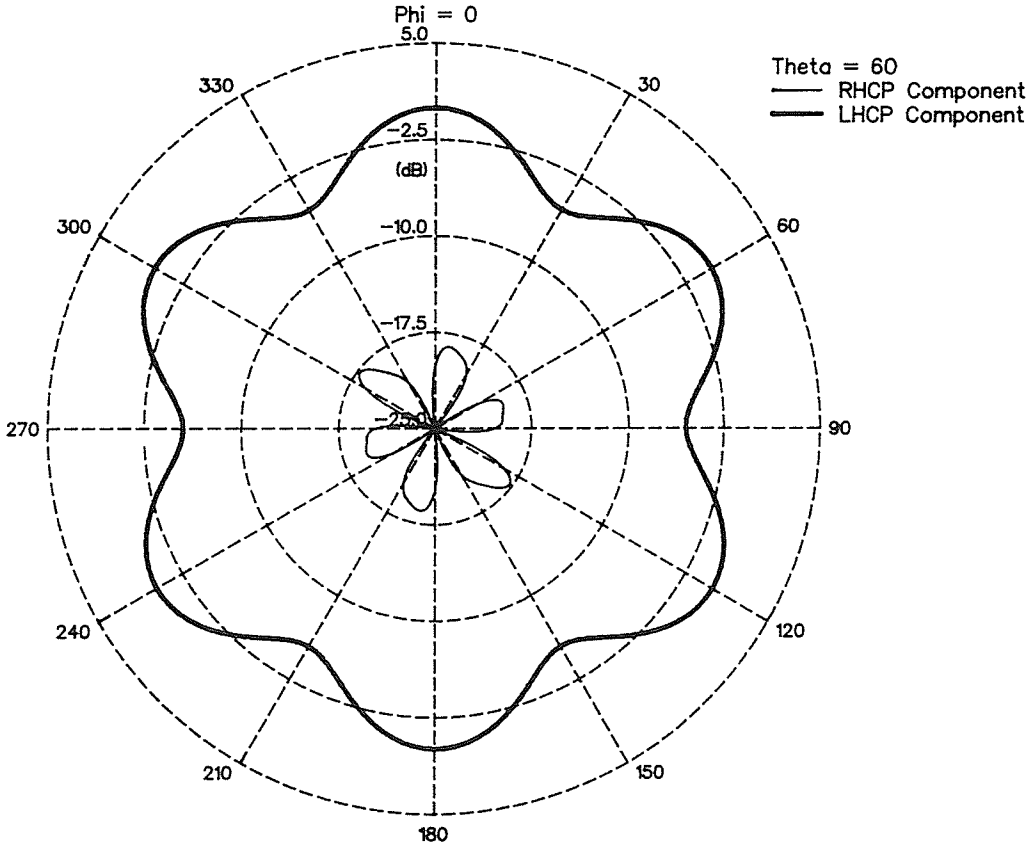


Figure 4.17b: Theta = 60, Phi varies

Figure 4.17: Planar 7-Element Array
Radiated Field Components
Central Element Excited Anti-Phase.

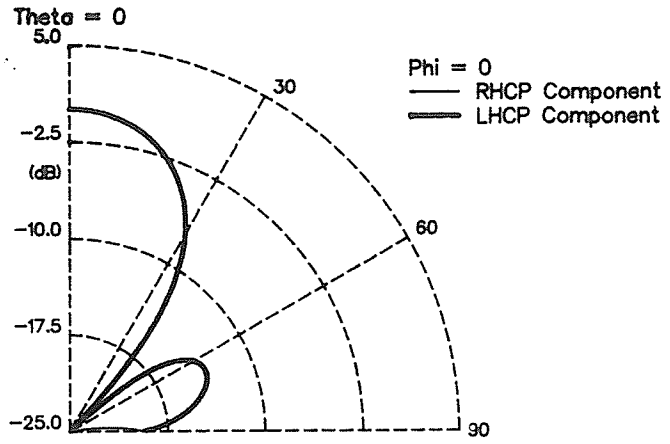


Figure 4.18a: Theta varies, Phi = 0

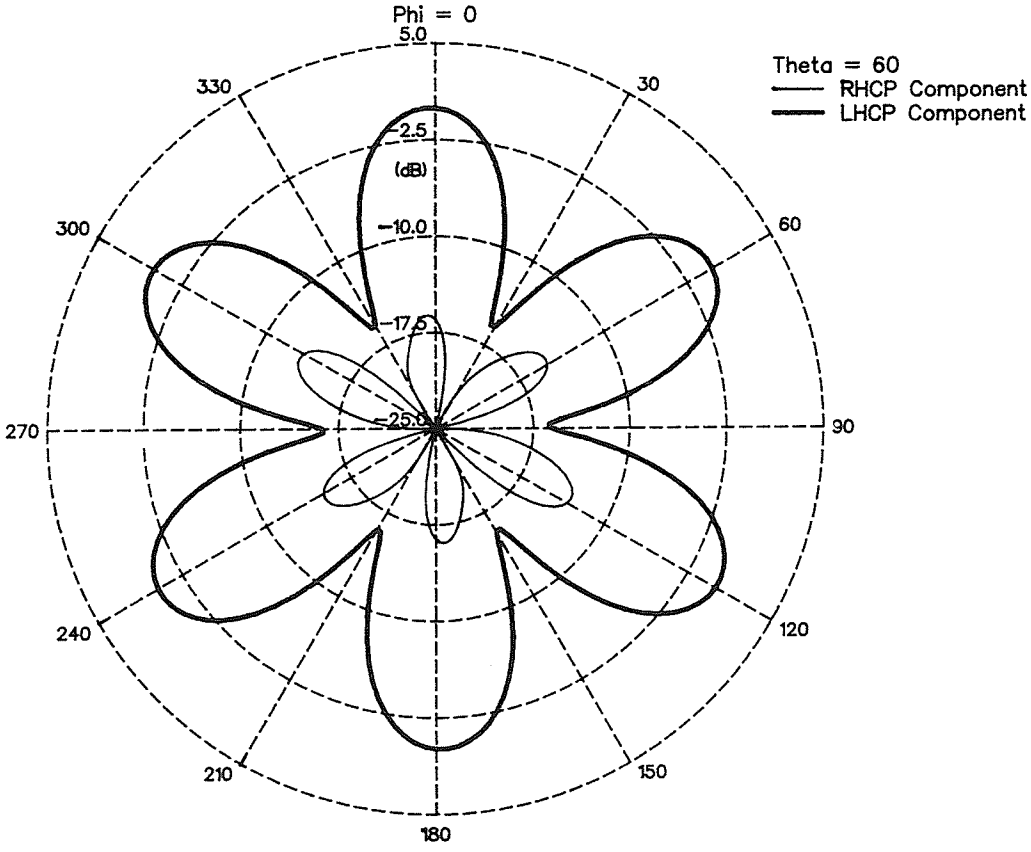


Figure 4.18b: Theta = 60, Phi varies

Figure 4.18: Planar 7-Element Array
Radiated Field Components
All Elements Excited In-Phase.

These results indicate that a single-ring circular array cannot provide the required performance for an MSAT ground station antenna, due to the inability to steer the beam, and to the undesirable 1st order mode radiation characteristics.

4.4.2.2 Dual Ring, Inphase

A second six-element ring of CDD's was placed at a radius of 1.289 λ, corresponding to the third zero of the derivative of the $J_0(x)$ Bessel function ($x = 7.016$) and calculated using equation 4.3 for an angle $\theta = 60^\circ$. Note that at this radius, $\theta = 28^\circ$ is also a solution to equation 4.3 with the second zero ($x = 3.832$), indicating another lobe will be radiated at this angle.

The elements were placed on the same 60° radials as the first ring in order to maintain the phase relationship between the rings. It was hoped that this second ring would improve the far field radiation performance and demonstrate the ability to steer the beam by adjusting the inter-ring phase shifts. The antenna geometry is shown in Figure 4.19 (but without the central element).

Each of the elements in each ring were excited inphase and initially, all elements in both rings were excited inphase as well. Figure 4.20 shows the calculated results for this configuration. The power in the boresite lobe is decreased in this case, but the gain at $\theta =$ 60° is not increased either. As well, the horizontal patterns are no longer omnidirectional, but show a large number of nulls.

Since the zeros of the derivative of the $J_0(x)$ Bessel function correspond to alternating maximum and minimums of $J_0(x)$, an inter-ring phase shift of 180° was tried. This should cause a re-enforcement of the power radiated at $\theta = 60^\circ$ and the calculated patterns are shown in Figure 4.21. Here we see the expected 2nd order mode lobe at $\theta =$ 28°, but unfortunately, while the horizontal pattern is somewhat improved, the vertical pattern shows no improvement.

These results indicate that the dual-ring array can also not be used for the MSAT

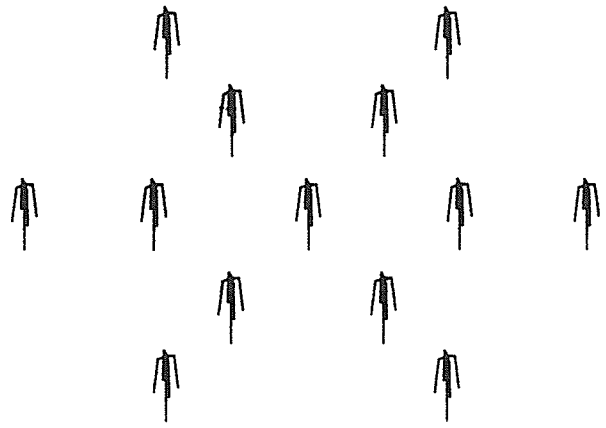


Figure 4.19a: Isometric View

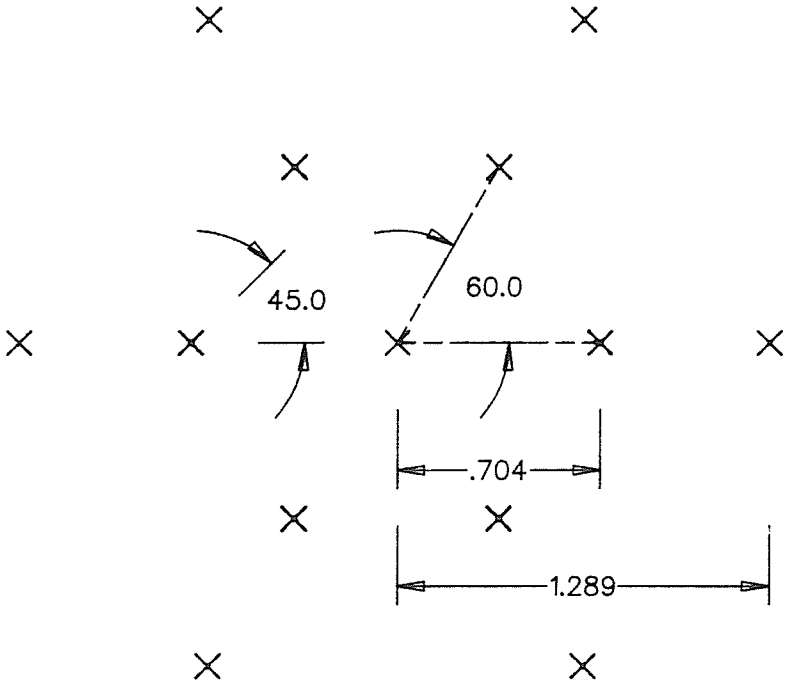


Figure 4.19b: Top View

Figure 4.19: Dual Ring Array of CDD's With Central Element

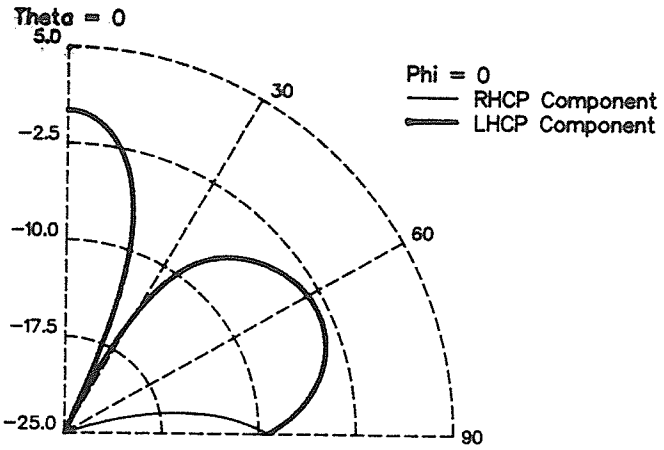


Figure 4.20a: Theta Varies, Phi = 0

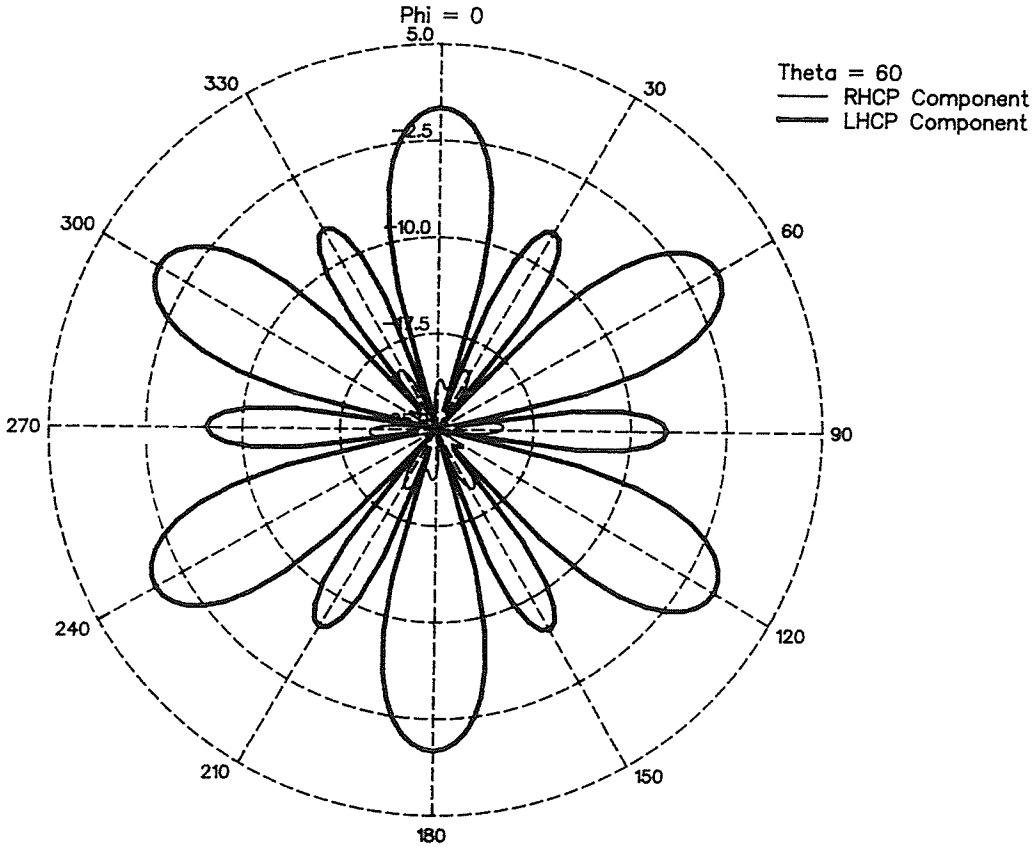


Figure 4.20b: Theta = 60, Phi varies

Figure 4.20: Planar 12-Element Array
Radiated Field Components
All Elements Excited In-Phase.

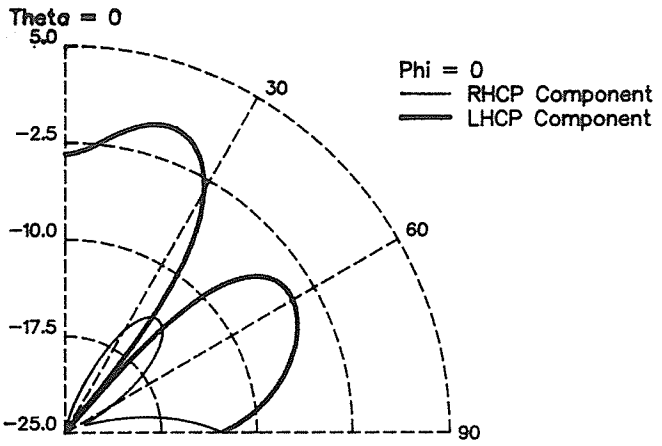


Figure 4.21a: Theta varies, Phi = 0

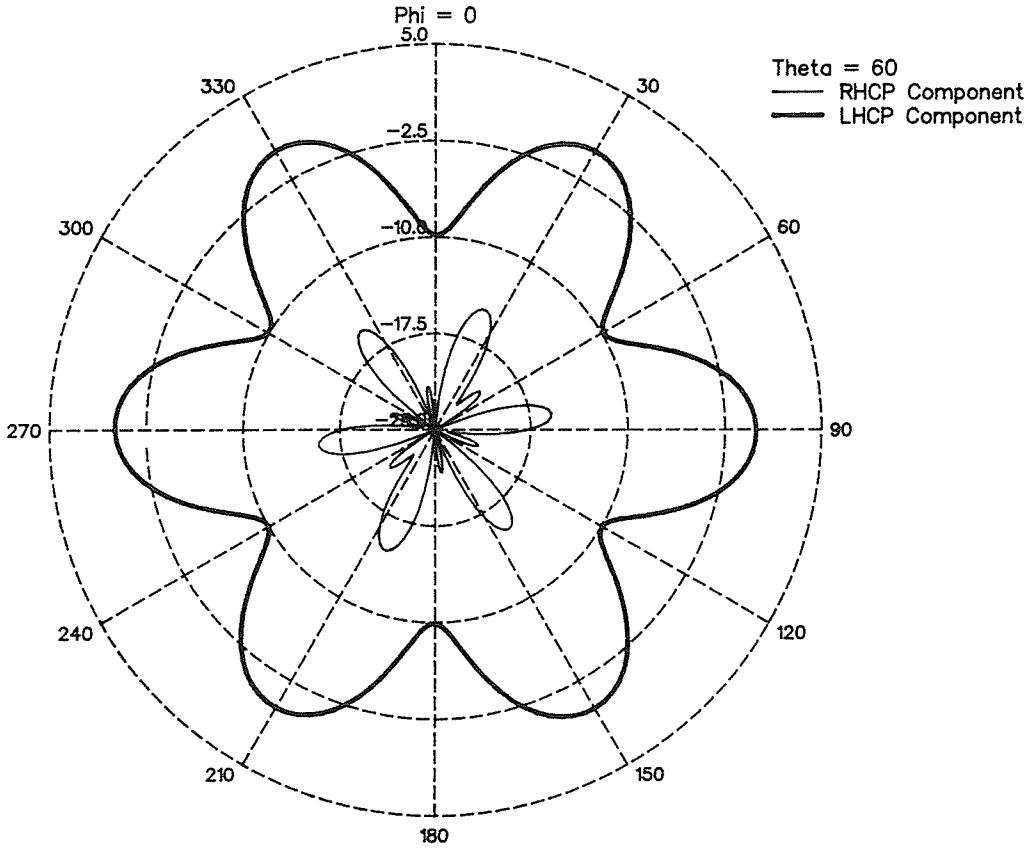


Figure 4.21b: Theta = 60, Phi varies

Figure 4.21: Planar 12-Element Array
Radiated Field Components
Second Ring Excited Anti-Phase.

ground station antenna when elements in each ring are excited inphase. The only other possible configuration which would provide an omnidirectional pattern in ϕ is the stacked-ring circular array, which will not be discussed herein since it would not be practical to place on the roof of an automobile.

4.4.2.3 Inter-Element Phase Shift Calculation

Using classical array theory, the beam from a circular array can be steered to any direction (θ,ϕ) by shifting the phase of the applied signal at each dipole (with respect to a central element) given by;

$$PS\ (^\circ) = 360\ a\ Cos(\phi_{ref} - \phi)\ Sin(\theta), \qquad (4.4)$$

where 'a' is the radius of the ring in which the element is situated, and ϕ_{ref} is the angle at which the particular element is situated around the ring.

For the six element ring shown in Figure 4.15, the angular spacing between elements is 60°. Table 4.3 shows calculated phase shifts required to steer the beam to $\phi = 0^\circ$ and $\phi = 30^\circ$ for array radii of 0.4, 0.45, 0.5, and 0.6 λ . The phase shifts for each element, and each arm of the element are shown, along with the corresponding voltage components to apply at the feed points in order to achieve these phase shifts. These shifts were calculated using the program listed in Appendix D.

4.4.2.4 Single Ring, Steered Beam

A six element, single ring, planar array was investigated using inter-element phase shifting to steer the beam. Various array radii were tried in order to determine the best radiated characteristics, as well as in an attempt to reduce the surface area needed to mount the antenna. This is an important consideration if the antenna is to be used on the roof of an automobile.

	Beam Angle (ϕ°)	Element Angle (ϕ°)	Arm 1 Phase Shift (°)	Arm 1 Applied Voltage Real	Arm 1 Applied Voltage Imaginary	Arm 2 Phase Shift (°)	Arm 2 Applied Voltage Real	Arm 2 Applied Voltage Imaginary
Array Radius = 0.4 λ	0	0	-125.0	-0.574	-0.819	-35.0	0.819	-0.574
		60	-62.5	0.462	-0.887	27.5	0.887	0.462
		120	62.5	0.462	0.887	152.5	-0.887	0.462
		180	125.0	-0.574	0.819	215.0	-0.819	-0.574
		240	62.5	0.462	0.887	152.5	-0.887	0.462
		300	-62.5	0.462	-0.887	27.5	0.887	0.462
	30	0	-108.3	-0.313	-0.950	-18.3	0.950	-0.313
		60	-108.3	-0.313	-0.950	-18.3	0.950	-0.313
		120	0.0	1.000	0.000	90.0	0.000	1.000
		180	108.3	-0.313	0.950	198.3	-0.950	-0.313
		240	108.3	-0.313	0.950	198.3	-0.950	-0.313
		300	62.5	1.000	0.000	90.0	0.000	1.000
Array Radius = 0.45 λ	0	0	-140.0	-0.766	-0.643	-50.0	0.643	-0.766
		60	-70.0	0.342	-0.940	20.0	0.940	0.342
		120	70.0	0.342	0.940	160.0	-0.940	0.342
		180	140.0	-0.766	0.643	230.0	-0.643	-0.766
		240	70.0	0.342	0.940	160.0	-0.940	0.342
		300	-70.0	0.342	-0.940	20.0	0.940	0.342
	30	0	-121.2	-0.519	-0.855	-31.2	0.855	-0.519
		60	-121.2	-0.519	-0.855	-31.2	0.855	-0.519
		120	0.0	1.000	0.000	90.0	0.000	1.000
		180	121.2	-0.519	0.855	211.2	-0.855	-0.519
		240	121.2	-0.519	0.855	211.2	-0.855	-0.519
		300	0.0	1.000	0.000	90.0	0.000	1.000
Array Radius = 0.5 λ	0	0	-156.0	-0.914	-0.407	-66.0	0.407	-0.914
		60	-78.0	0.208	-0.978	12.0	0.978	0.208
		120	78.0	0.208	0.978	168.0	-0.978	0.208
		180	156.0	-0.914	0.407	246.0	-0.407	-0.914
		240	78.0	0.208	0.978	168.0	-0.978	0.208
		300	-78.0	0.208	-0.978	12.0	0.978	0.208
	30	0	-135.1	-0.708	-0.706	-45.1	0.706	-0.708
		60	-135.1	-0.708	-0.706	-45.1	0.706	-0.708
		120	0.0	1.000	0.000	90.0	0.000	1.000
		180	135.1	-0.708	0.706	225.1	-0.706	-0.708
		240	135.1	-0.708	0.706	225.1	-0.706	-0.708
		300	0.0	1.000	0.000	90.0	0.000	1.000
Array Radius = 0.6 λ	0	0	-187.0	-0.993	0.122	-97.0	-0.122	-0.993
		60	-93.5	-0.061	-0.998	-3.5	0.998	-0.061
		120	93.5	-0.061	0.998	183.5	-0.988	-0.061
		180	187.0	-0.993	-0.122	277.0	0.122	-0.993
		240	93.5	-0.061	0.998	183.5	-0.998	-0.061
		300	-93.5	-0.061	-0.998	-3.5	0.998	-0.061
	30	0	-161.9	-0.951	-0.310	-71.9	0.310	-0.951
		60	-161.9	-0.951	-0.310	-71.9	0.310	-0.951
		120	0.0	1.000	0.000	90.0	0.000	1.000
		180	161.9	-0.951	0.310	251.9	-0.310	-0.951
		240	161.9	-0.951	0.310	251.9	-0.310	-0.951
		300	0.0	1.000	0.000	90.0	0.000	1.000

Table 4.3: Calculated Inter-Element Phase Shifts

The elements are situated on 60° radials, and the beam was steered in 30° intervals in ϕ. Because of the symmetry of this configuration, only two far field radiated patterns needed to be calculated, those for beams steered to $\phi = 0°$, and $\phi = 30°$. All other angles of beam steering will give the same pattern as one of these two.

Figures 4.22 and 4.23 show the circularly polarized components of the far field for an array radius of 0.4 λ, with the beam steered to $\phi = 0°$ and 30° respectively. The maximum gain in this case is 10.63 dB, with a 3 dB beamwidth in θ of 47°, and a 3 dB beamwidth in ϕ of 60°. The highest sidelobe is the backlobe, and is only -6 dB from the main beam. Notice on the vertical pattern,that there is only a single lobe, but that the gain at $\theta = 0°$ is not zero as desired.

Figures 4.24 and 4.25 are for an array radius of 0.5 λ. The gain in this case is 10.8 dB and the 3 dB beamwidths are virtually unchanged. The highest sidelobe level has been reduced to -7 dB from the main beam. Notice on the vertical pattern that a second lobe along the boresite is beginning to show.

Figures 4.26 and 4.27 are for an array radius of 0.6 λ. Notice the increase of the boresite lobe and the large backlobe on the 30° pattern. The gain in this case is increased, though, to 12.0 dB.

Figures 4.28 and 4.29 are for an array radius of 0.45 λ, which gives the best vertical pattern, ie. no boresite lobe. Unfortunately, the gain in this configuration is only 10.7 db, indicating that some gain enhancement will be necessary to meet the requirements for the MSAT system. Addition of a central element does not improve the peak gain, and actually degrades the ellipticity, probably due to increased coupling.

Note the high sidelobes and backlobe calculated with this configuration. These can be reduced by applying a symmetrical amplitude taper to the front half of the array, and low or zero excitation to the back half of the array [31].

Figures 4.30 and 4.31 are for an array radius of 0.45 λ, with a simple linear amplitude taper applied to the elements. In this case, the front elements are excited with

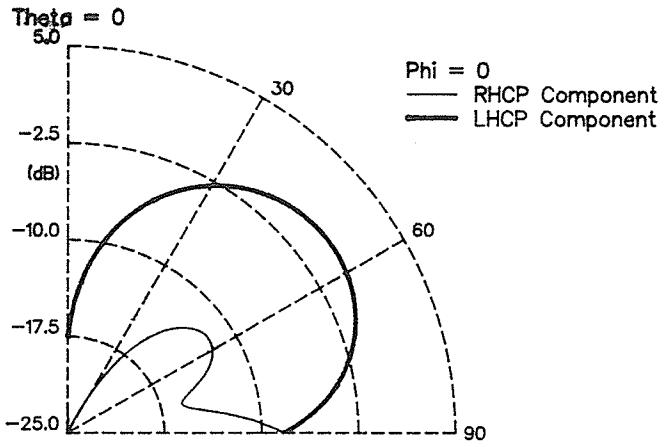


Figure 4.22a: Theta varies, Phi = 0

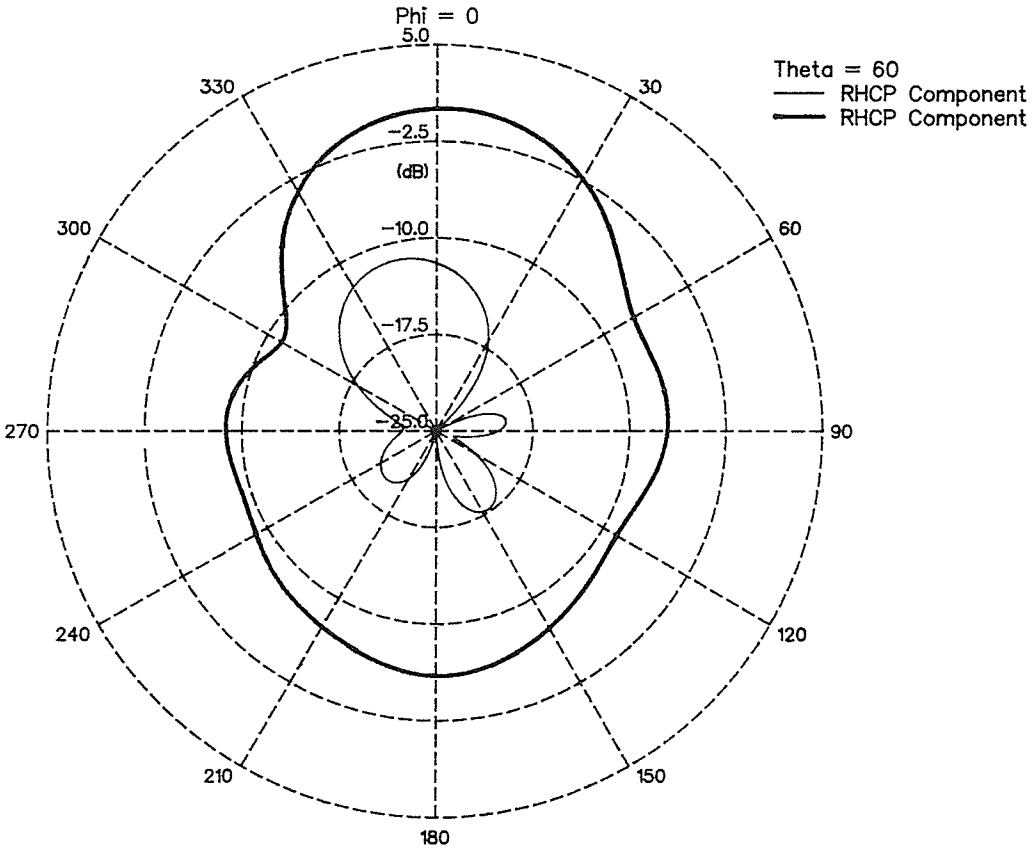


Figure 4.22b: Theta = 60, Phi varies

Figure 4.22: Planar 6-Element Array (r = 0.4)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 0

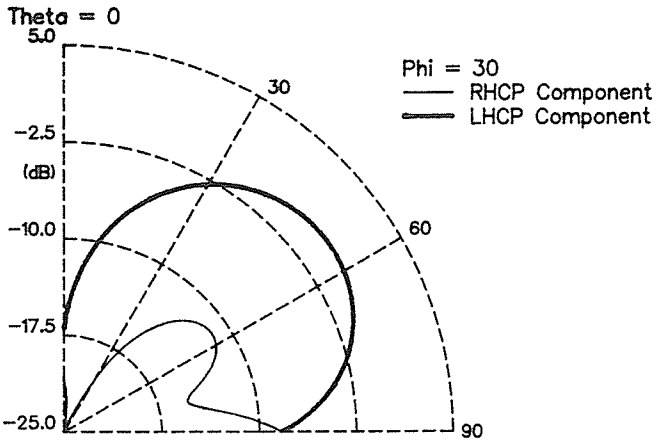


Figure 4.23a: Theta varies, Phi = 30

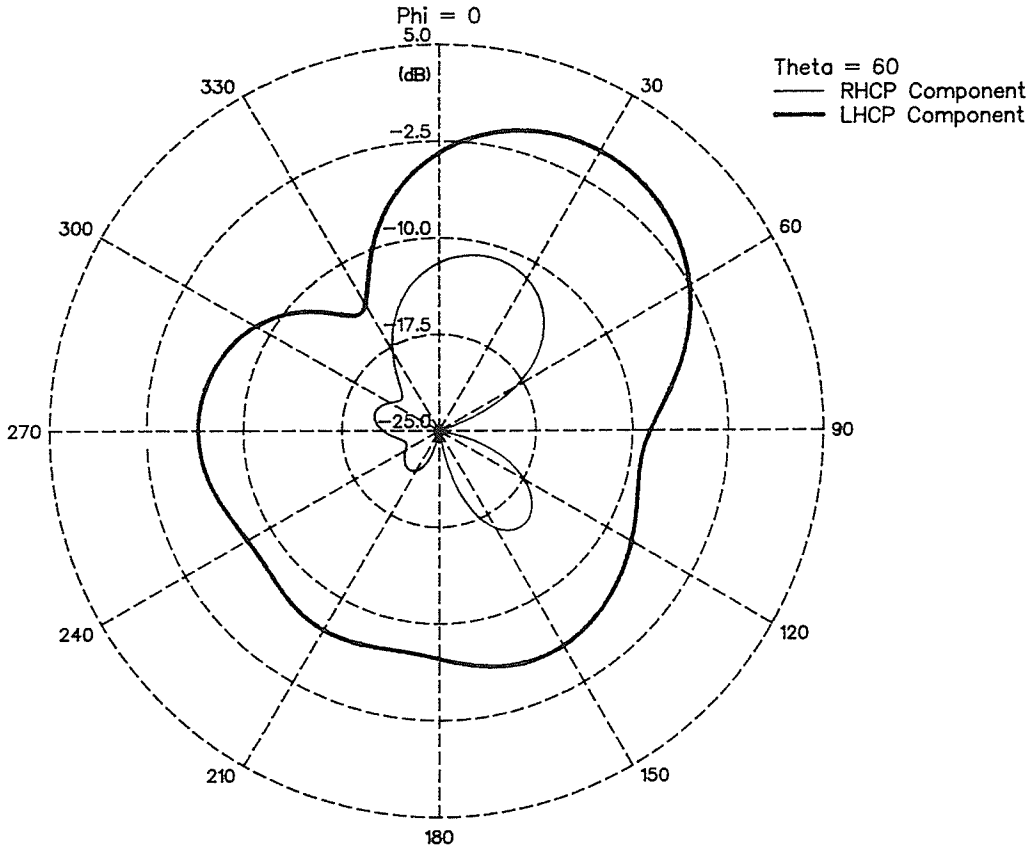


Figure 4.23b: Theta = 60, Phi varies

Figure 4.23: Planar 6-Element Array (r = 0.4)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 30

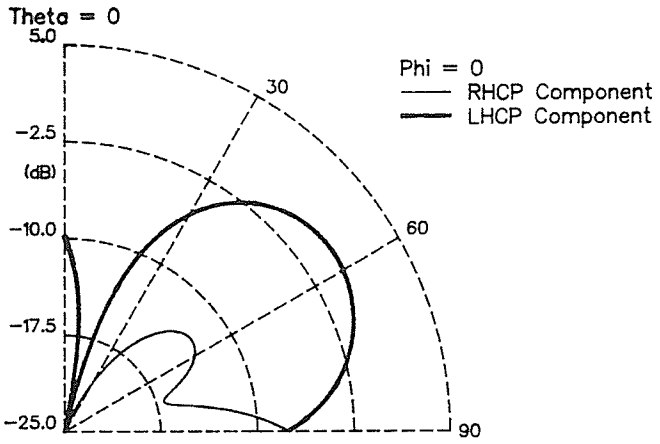


Figure 4.24a: Theta varies, Phi = 0

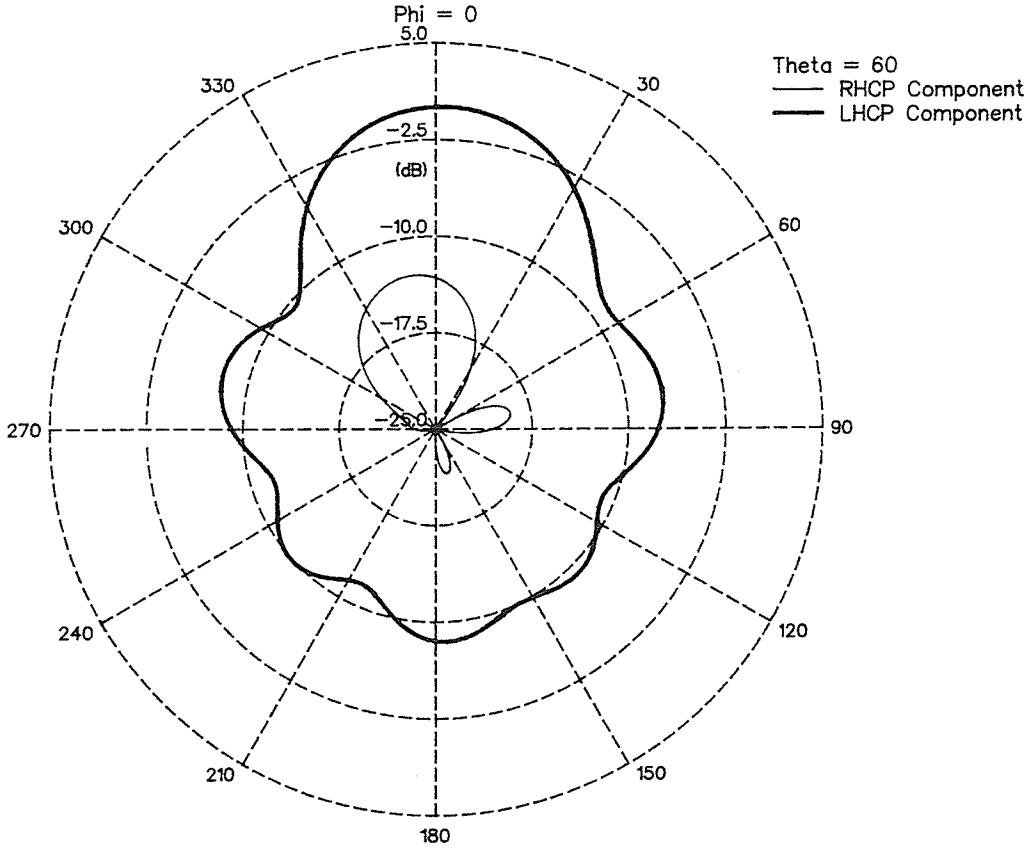


Figure 4.24b: Theta = 60, Phi varies

Figure 4.24: Planar 6-Element Array (r = 0.5)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 0

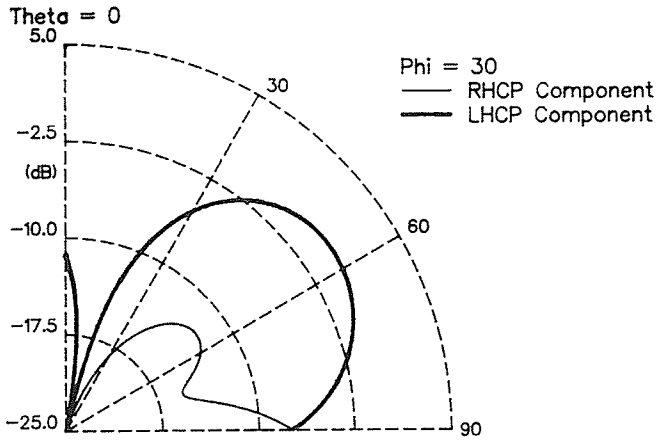


Figure 4.25a: Theta varies, Phi = 30

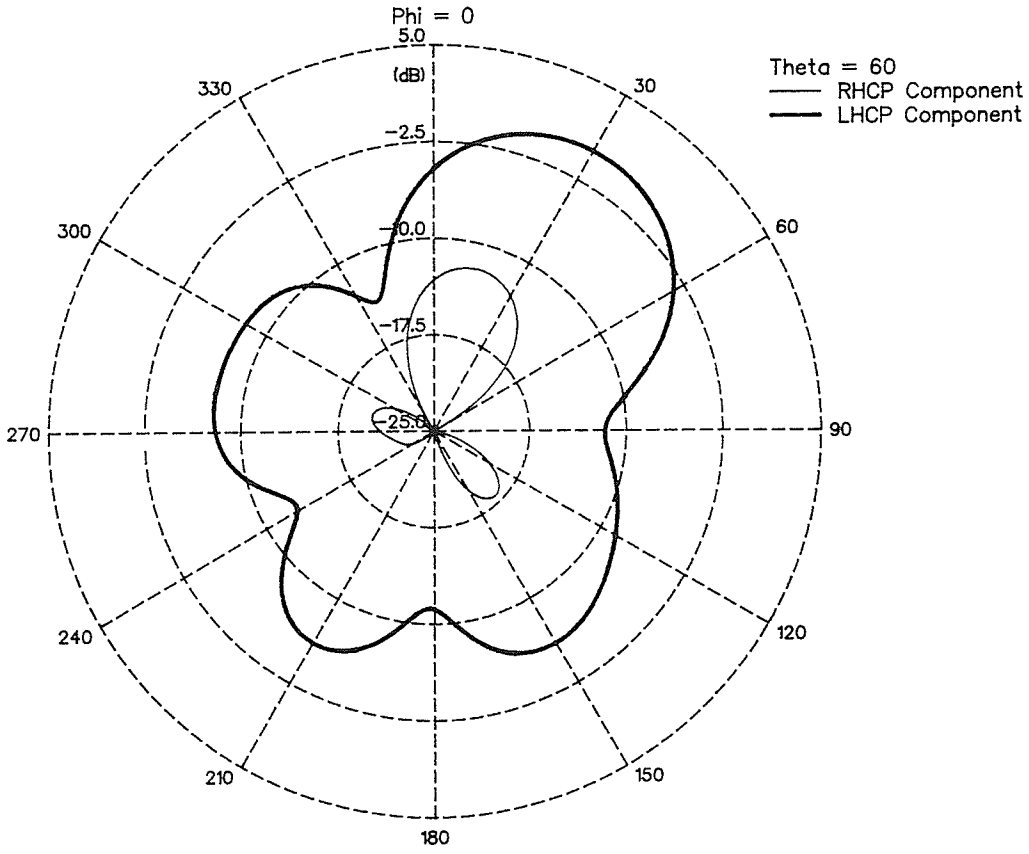


Figure 4.25b: Theta = 60, Phi varies

Figure 4.25: Planar 6-Element Array (r = 0.5)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 30

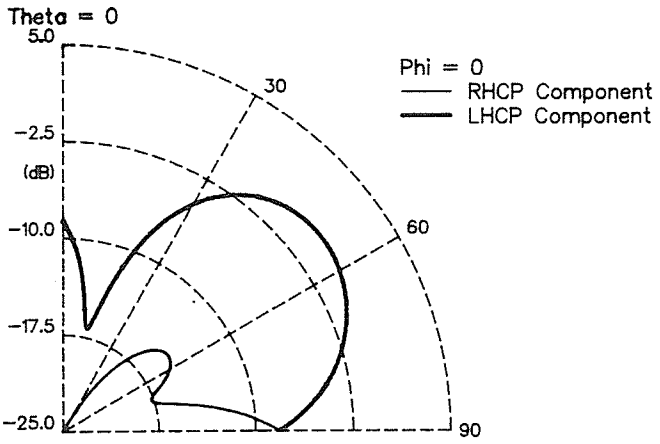


Figure 4.26a: Theta varies, Phi = 0

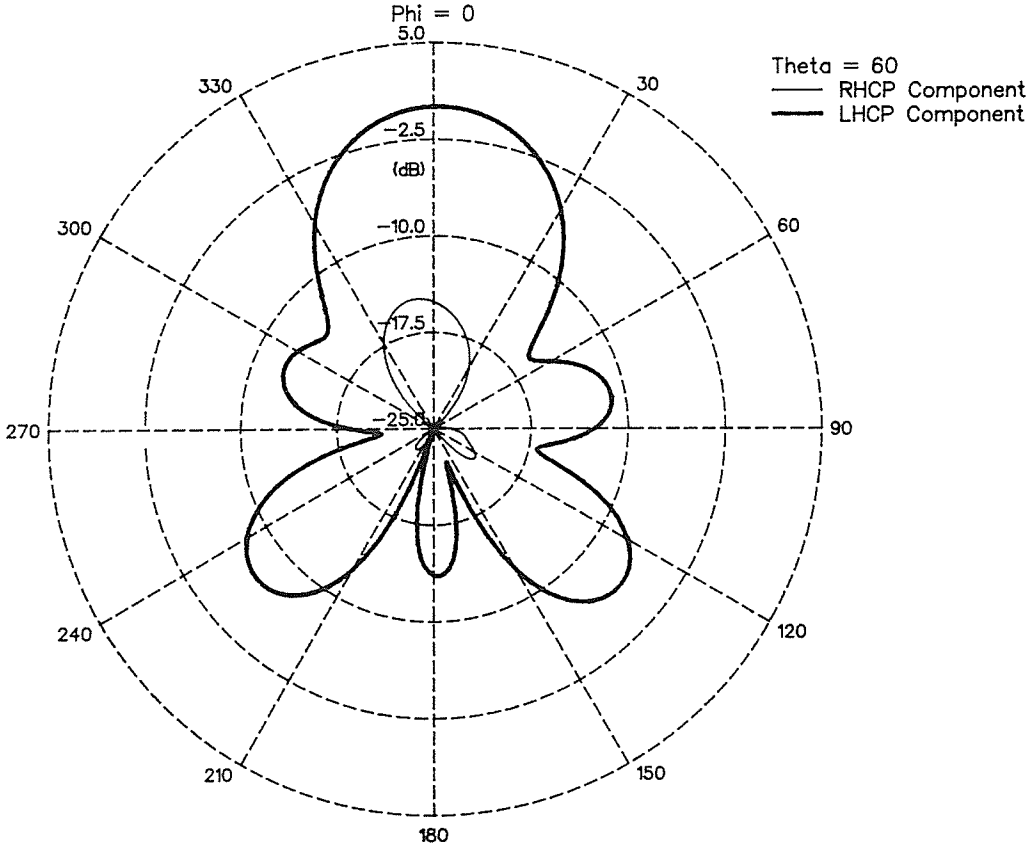


Figure 4.26b: Theta = 60, Phi varies

Figure 4.26: Planar 6-Element Array (r = 0.6)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 0

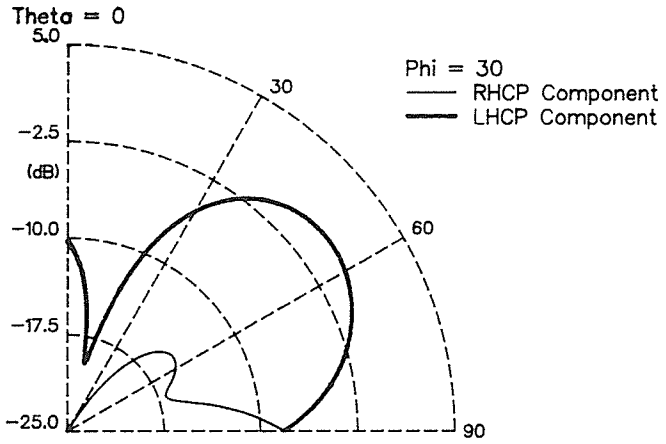


Figure 4.27a: Theta varies, Phi = 30

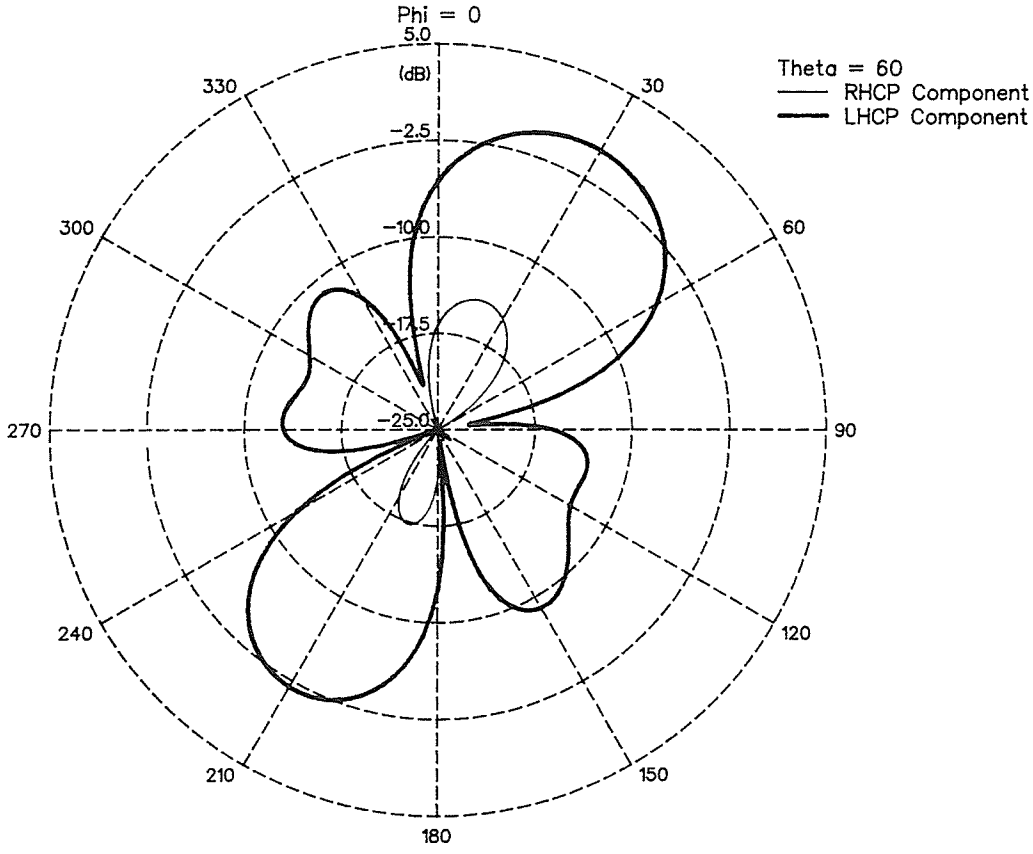


Figure 4.27b: Theta = 60, Phi varies

Figure 4.27: Planar 6-Element Array (r = 0.6)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 30

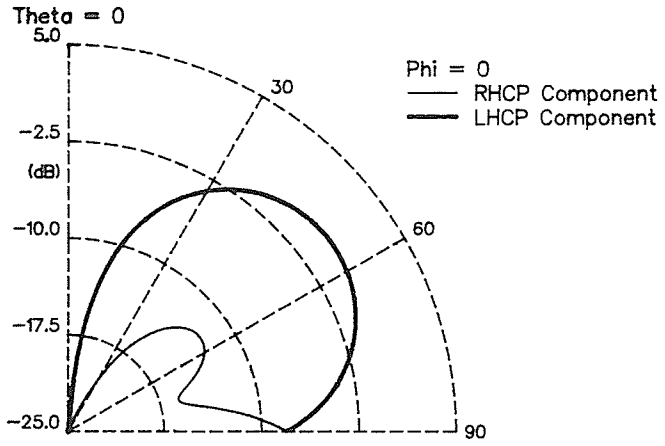


Figure 4.28a: Theta varies, Phi = 0

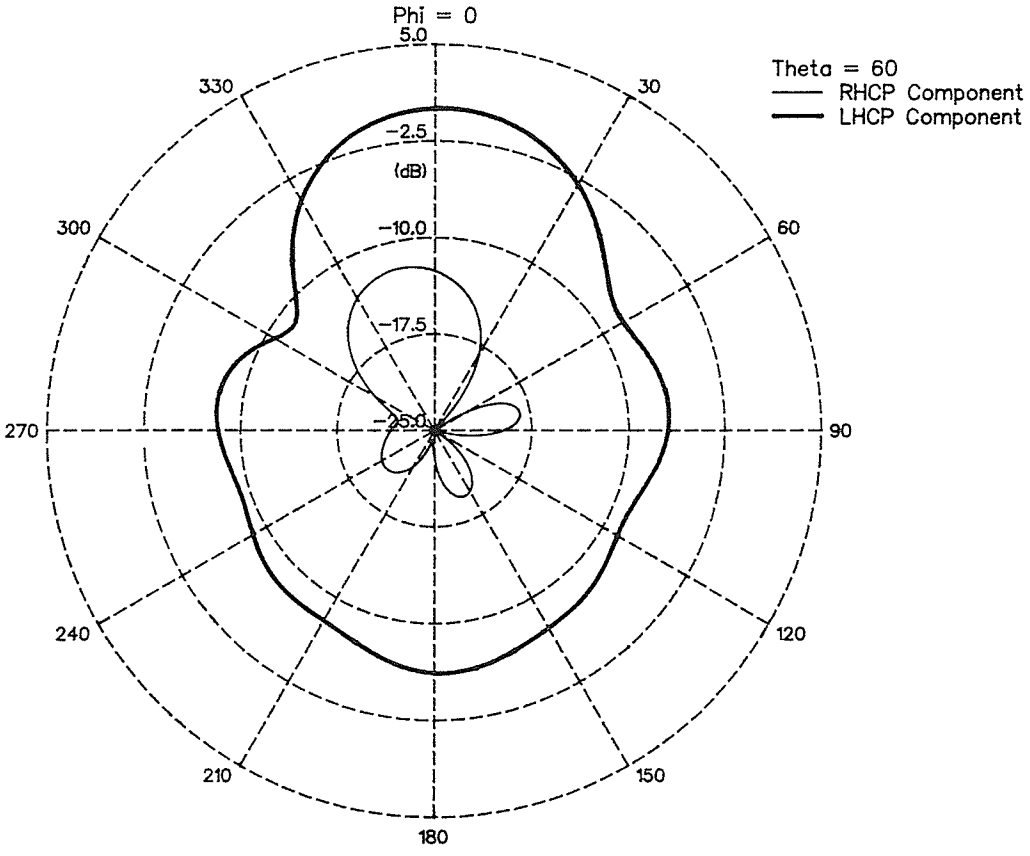


Figure 4.28b: Theta = 60, Phi varies

Figure 4.28: Planar 6-Element Array (r = 0.45)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 0

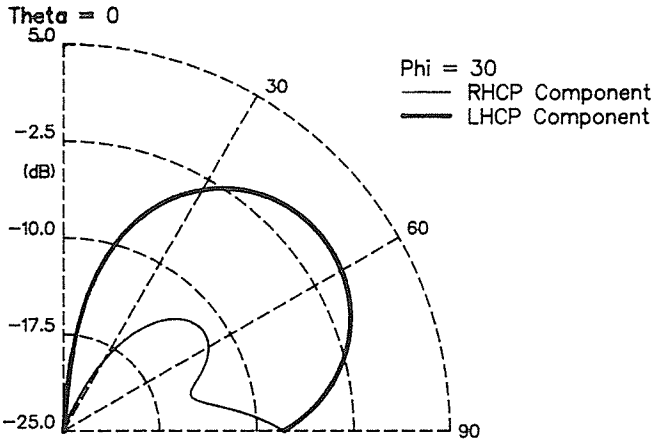


Figure 4.29a: Theta varies, Phi = 30

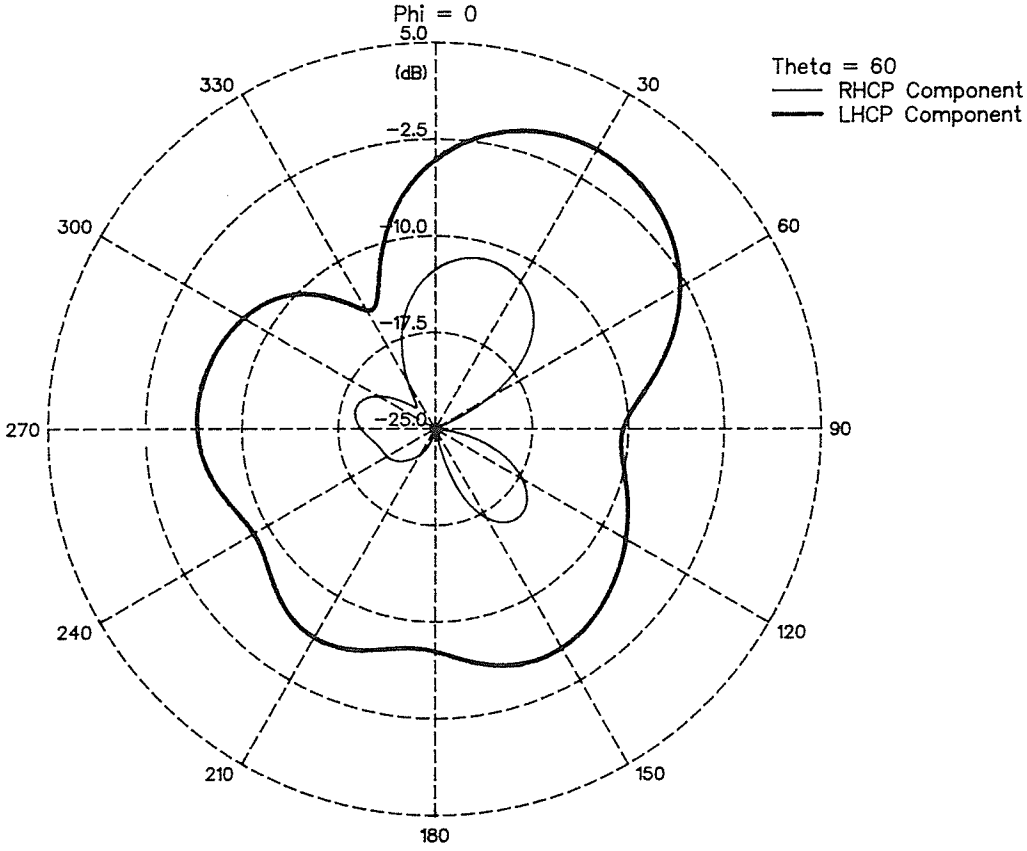


Figure 4.29b: Theta = 60, Phi varies

Figure 4.29: Planar 6-Element Array (r = 0.45)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 30

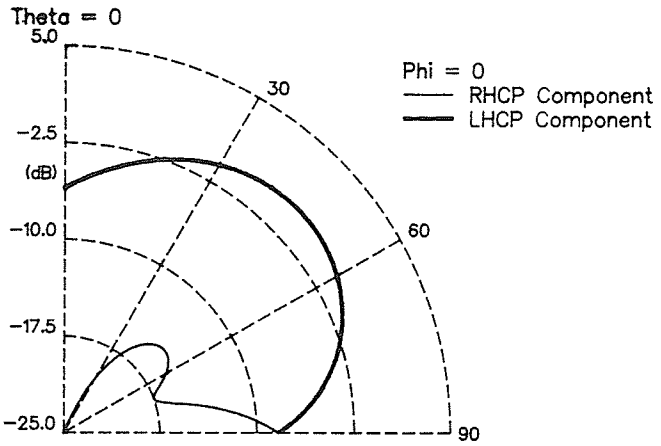


Figure 4.30a: Theta varies, Phi = 0

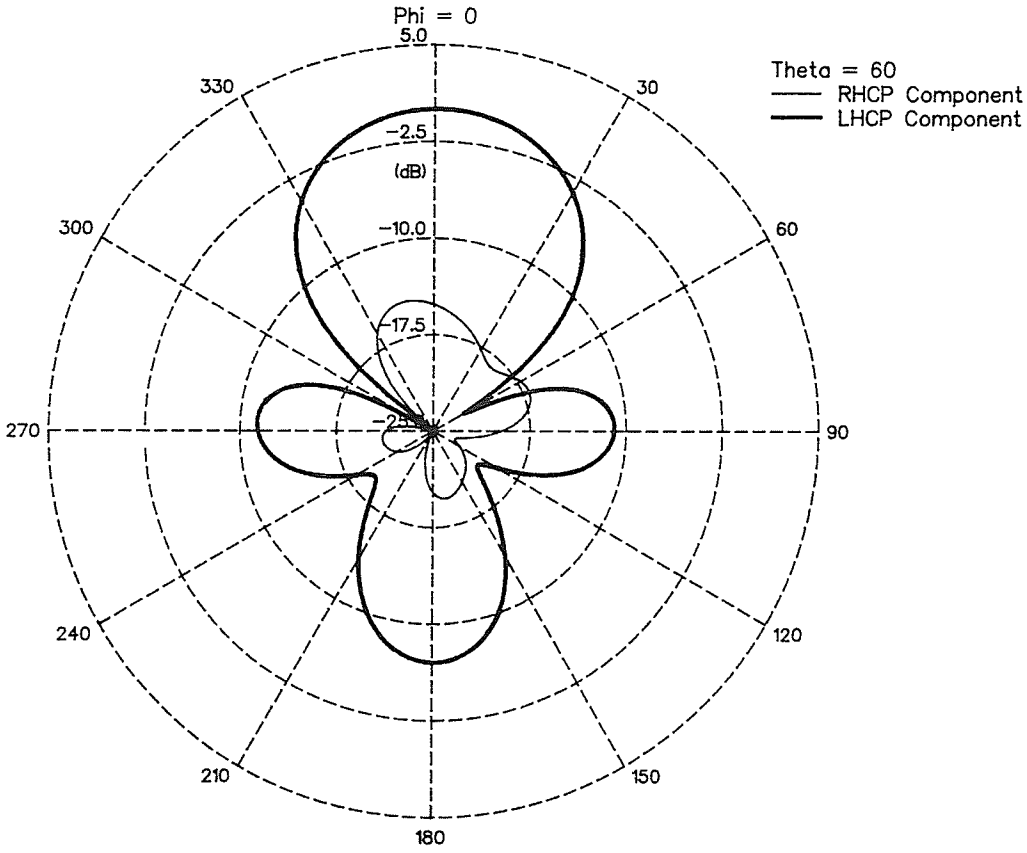


Figure 4.30b: Theta = 60, Phi varies

Figure 4.30: Planar 6-Element Array (r = 0.45)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 0
Tapered Amplitude Distribution

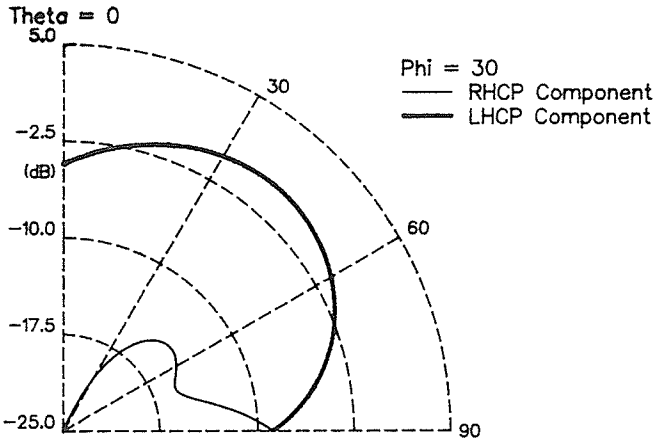


Figure 4.31a: Theta varies, Phi = 30

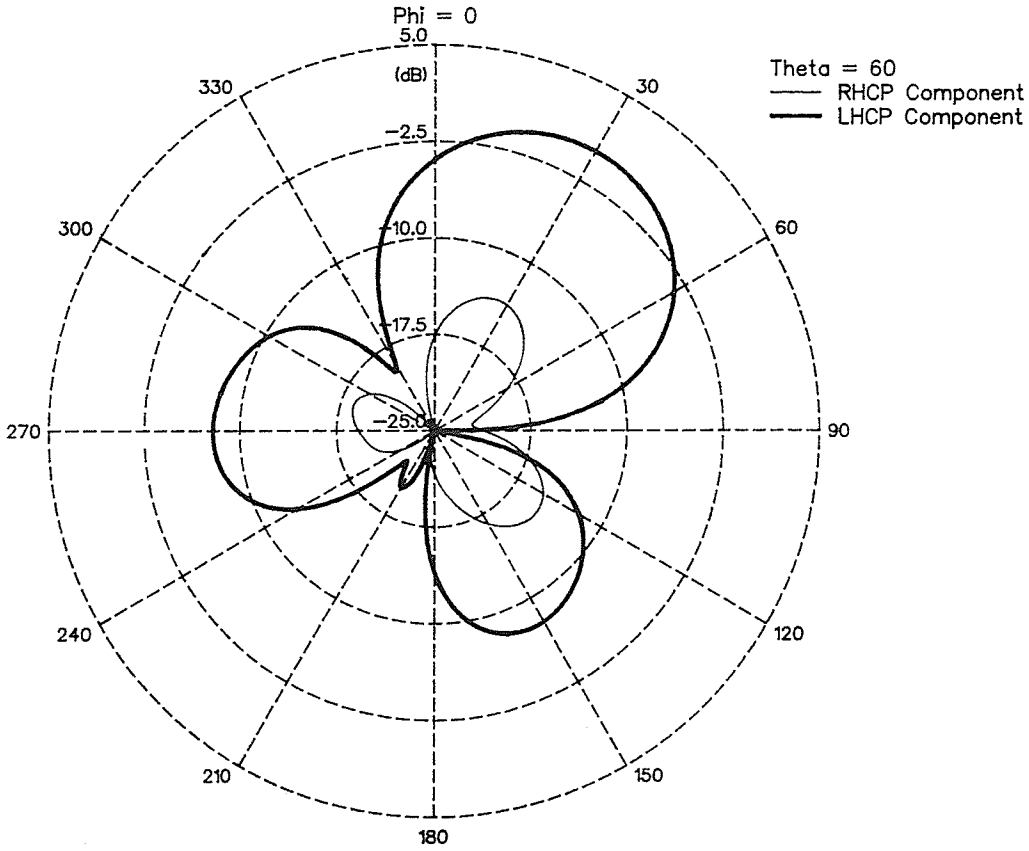


Figure 4.31b: Theta = 60, Phi varies

Figure 4.31: Planar 6-Element Array (r = 0.45)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 30
Tapered Amplitude Distribution

100% amplitude, and the back elements are not excited (0%). Notice the vast improvement in the horizontal pattern, with reduced sidelobes and backlobe. Unfortunately, the vertical pattern is severely degraded, actually approaching the single element pattern. The maximum gain has been increased to 11.3 dB.

Figures 4.32 and 4.33 are for a second tapered amplitude distribution, with the back elements excited with 50% amplitude. The horizontal pattern is not quite as good as the first taper, but the vertical pattern is much better. The gain in this case is 11.4 dB.

These results indicate that a planar phased array can meet the requirements for an MSAT ground station antenna. The six element, single ring, planar array gives the minimum gain required when a tapered amplitude distribution is used along with the appropriate inter-elemental phase shifts. Increasing the number of elements in the ring will increase the gain and should be investigated in the future.

4.5 Summary of Results

The CDD is a good choice for the element to be used in the MSAT mobile ground station antenna due to its good radiation characteristics in the region of interest, and due to its simplicity for feeding, steering and manufacturing. A major disadvantage of this element is the fact that it does not conform to the surface of the vehicle, and thus could not be used in some applications.

A six element circular array of CDDs can provide the required gain, polarization, and beam steering capabilities for the MSAT system, and thus would make a good final design. This circular array can be placed under a dome and situated on the roof of a mobile vehicle.

This discussion does not include an analysis of the impedance of the antenna, a problem that must be solved before the antenna can be used. Furthermore, the array has not been constructed to determine how the calculated fields compare to the actual radiated fields.

This design example shows how the IGAD system, in conjuction with the numerical optimization techniques, can be used in a practical situation.

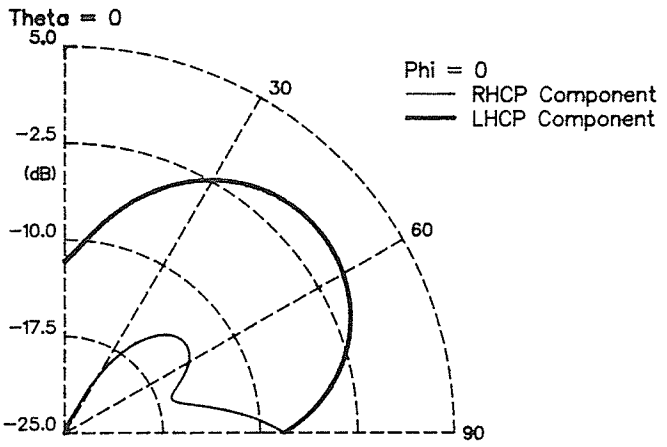


Figure 4.32a: Theta varies, Phi = 0

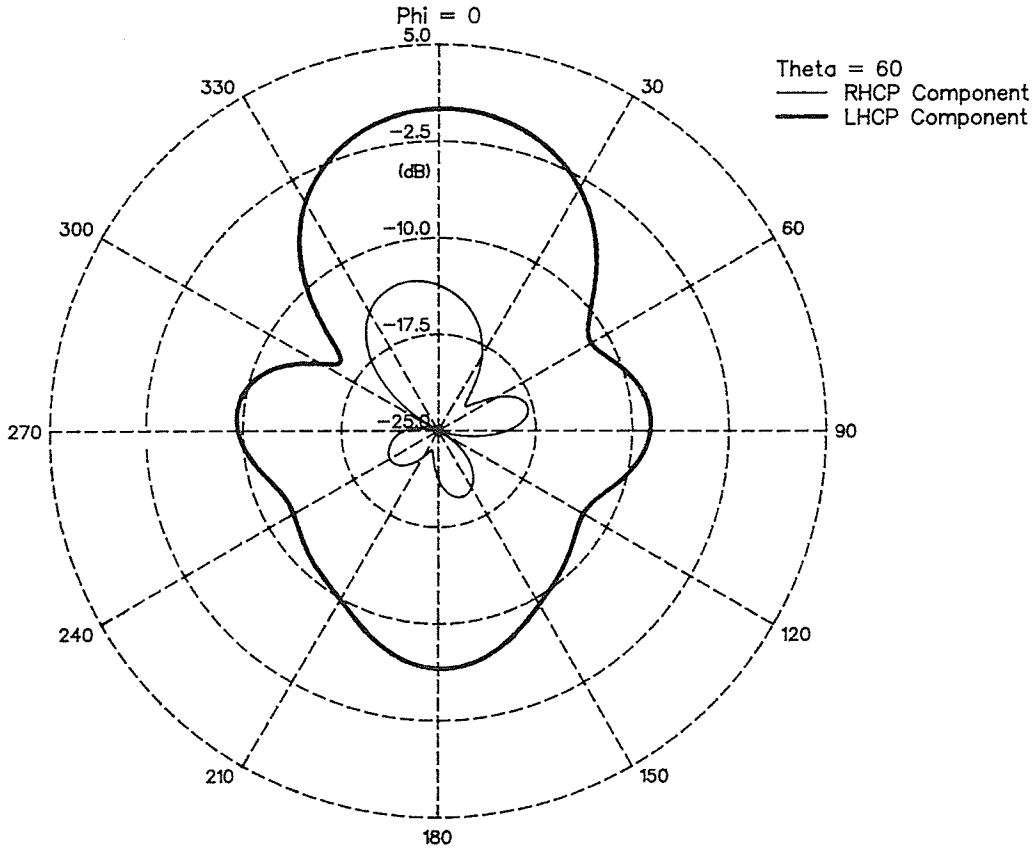


Figure 4.32b: Theta = 60, Phi varies

Figure 4.32: Planar 6-Element Array (r = 0.45)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 0
Tapered Amplitude Distribution (Case 2)

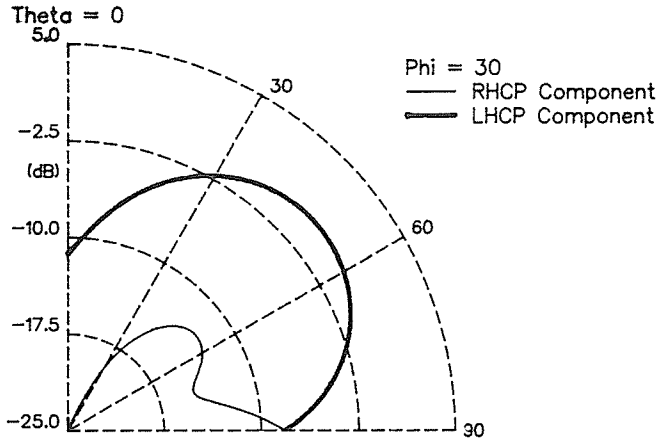


Figure 4.33a: Theta varies, Phi = 30

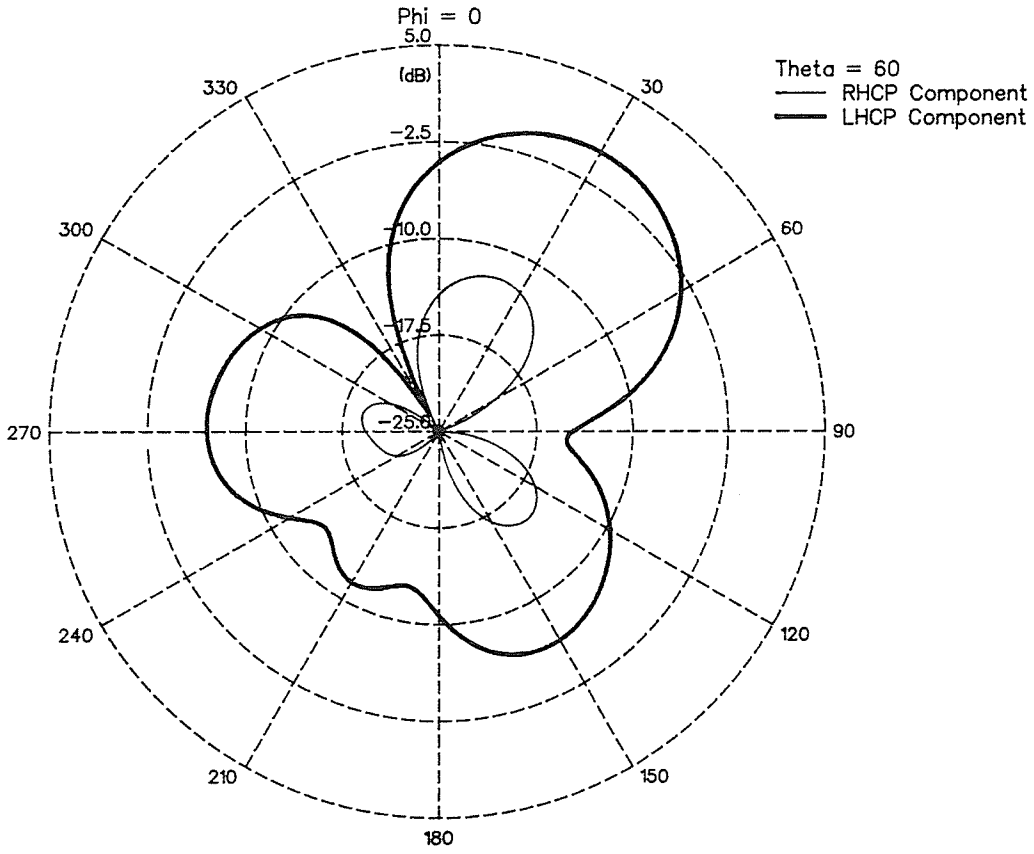


Figure 4.33b: Theta = 60, Phi varies

Figure 4.33: Planar 6-Element Array (r = 0.45)
Radiated Field Components
Beam Steered to Theta = 60, Phi = 30
Tapered Amplitude Distribution (Case 2)

CHAPTER 5

EVALUATION OF THE IGAD PACKAGE

5.1 Introduction

This chapter summarizes the usefulness of the IGAD package determined during the design of the MSAT mobile antenna. The purpose is to give a thorough critic of the system and identify areas of improvement.

5.2 The Pre-Processor

The pre-processor's main function is to facilitate the creation of antenna model geometry through interactive graphic input. In terms of emulating the analysis routine's standard input commands, the IGAD system works well. Unfortunately, this may not be the easiest and most efficient way to go about model creation.

During the initial model creation process, the functions developed specifically to create geometry according to the analysis routine's modeling restrictions were exercised. These routines were quite cumbersome to use, and sometimes it was difficult to achieve the desired results. This is because the routines were not developed fully enough to take into account all the special cases that exist, and thus could not be used in some ways. As well, since the routines were written in DAL, which is itself a second level language (all DAL routines were originally written in FORTRAN), they execute quite slowly. At times it was frustrating to invoke a command and then sit back for minutes at a time, waiting for it to execute.

The most useful of all the commands created was the routine to generate geometry from the analysis routine input records (the NAAR to IGAD Translator as described in Section 2.3.3). I found it more efficient and actually easier to simply modify the analysis routine input file using a text editor, and then use the IGAD routine to provide visual verification of the changes in the model. On many occasions, this visual verification helped eleminate errors in data entry before submitting the model to the analysis routine, which ultimately was the overall goal of the IGAD system.

There exists an alternative way to implement the modeling stage of the IGAD package, one that was not tried because it assumes that the user has a working knowledge of Calma's DDM interactive graphics system. In this case, the user would simply create the antenna model using all the standard DDM model items, such as lines, arcs, splines, etc. with all the standard creation, editing, and movement commands. The model could be constructed and edited much faster and more easily using these fully developed commands. When all modifications were complete, the user would then run a segmentation routine which would segment each item, according to the modeling restrictions, and create the analysis routine input file.

This last method is desirable for a number of reasons; model creation is easier and faster, the creation and editing commands are more versatile and more fully developed, much support exists for these standard commands in terms of help files and manuals, and finally, this style of model creation is more easily transportable between different analysis routines. For each new routine, only the segmentation routine (or its equivalent) must be created. The only reason this style of model creation was not tried initially was because the system was supposed to be available to a wide variety of users, without much prior CAD knowledge. Because of my experience using the IGAD system, I now believe that this method should at least be tried.

The patch options available in NEC were only implemented in a limited way, and were not developed because of the excess restrictions and difficulty of use. If standard DDM

commands were used to create Bezier or ruled surfaces, then these options could be used more easily and efficiently.

5.3 The Post-Processor

The post-processors main function is to facilitate the interpretation of the data calculated by the numerical analysis routine. This section of IGAD was used throughout every stage of the design of the MSAT antenna and during the analysis of alternative elements as well.

These routines have made the most noticeable improvement in the design cycle. The most useful option I found was the calculation of the circularly polarized components of the far field, which, when plotted on a radial plots, conveyed information as to the direction of the radiated power as well as the ellipticity or cross-polarization of the pattern.

Some consideration was given to three-dimensional plots, using contoured surfaces, but, since most of the antennas studied herein were omnidirectional in ϕ, it was not attempted. This type of plot is very useful and should be developed in the future.

There are many other parameters which could be derived from the data calculated by the numerical analysis routine, which have not been attempted here. These include, frequency bandwidth, beamwidth, energy density, and many others.

5.4 Model Storage

The IGAD system stores all model data and pattern calculations in a Calma model file. This file is quite large, and contains a lot of unnecessary data, and hence, takes a long time to reconstruct whenever it is recalled. As well, it is difficult to merge models if multiple pattern plots are desired for comparison purposes.

An alternative to this would be to create a custom model file, which would hold model and pattern data only. This could decrease the system start-up time, and facilitate multiple pattern plotting, as well as reducing the storage requirements.

CHAPTER 6

CONCLUSIONS AND FUTURE RECOMMENDATIONS

6.1 The IGAD Package

The IGAD package successfuly incorporates the use of interactive graphics with numerical analysis techniques, providing the user with a powerful antenna design tool.

The pre-processor contains a versatile set of geometry construction commands which incorporate all modeling restrictions imposed by the numerical analysis routine. All display modification, model management viewport management, geometry creation and data input functions are incorporated on an on-screen menu and can be invoked by digitizing on a tablet. The basic element of model construction is the straight wire segment, which is incorporated as a cylinder in the three dimensional model database. This choice was not the optimum, since it causes considerable and time consuming calculations by the display processor to generate each of the three dimentional views. A better method would have been to implement the segment as a simple line for construction purposes and this should be done in the future. In addition, a routine to display the true modeled geometry (as cylinders) should be developed in order to evaluate the modeling error due to segment overlap when required.

The geometry editing and movement commands are quite primitive, but incorporate all functions originally available in the analysis routine input command structure. Much work can be done to increase the ease and usefulness of these commands.

One alternative model creation method should be tried, where the designer would

model the antenna using any of the basic construction elements available in DDM (lines, arcs, splines, etc.), using all the well developed editing commands. Once the model is completed, the designer would simply call a segmentation routine which would segment each item of the model, maintaining the appropriate modelling restrictions. The only problem with this is that it requires that the user knows DDM before proceeding with the design of antennas.

The post-processor allows the user to plot the following parameters; current, gain, θ and ϕ components of the far field, circularly polarized components of the far field, X,Y,and Z components of the near field, and the ellipticity of the radiated signal, all as a function of position (θ,ϕ in the far field, X,Y,Z in the near field). Both radial and rectangular plots are available, with data normalization, conversion to logarithms, and user defined range specification. Three-dimensional contour plots have not yet been developed, and should be investigated in the future.

The IGAD package is presently tied to the Calma software, and thus is not readily available for use by other antenna designers. In the future, consideration should be given to moving the package to a smaller, more affordable workstation or personal computer.

6.2 Implementation of Numerical Optimization

Numerical optimization techniques can aid in the design process by taking an antenna model which is in its approximately final form (ie. can be described in terms of a few simple design variables), and generating the optimum design without user intervention.

Rosenbrock's minimization technique was used successfuly along with the 'Minimax' constraint handling technique to create an optimum crossed drooping dipole element for use in the MSAT mobile antenna array. Unfortunately, the programming and use of these techniques was all done manually, and separate from the IGAD system.

In future, incorporation of 'Variable Geometry' in the modeling stage will allow optimization techniques to be included as a pre-programmed option in IGAD, and should be investigated.

6.3 Design of the MSAT Mobile Antenna

An optimized crossed drooping dipole was chosen as the element for use in the MSAT mobile antenna array because of its inexpensive design, ease of construction, ease of modeling and analysis with NAAR, and most importantly, because of its performance; low cross-polarization and directed power characteristics in the region of interest.

This element was used in a six element planar array of dipoles, with inter-element phase shifting providing the necessary beam steering requirements. The six element array meets the minimum gain requirements, but more elements could be used to provide a greater margin for signal fading, and should be investigated more fully.

Tapered amplitude distributions were used to reduce the sidelobe and backlobe levels with limited success, since they caused a widening of the main beam as well. More work should be done in this area as well.

This work is strictly numerical, and is only concerned with the calculated far field radiation characteristics. No work has been done on the feed techniques, input impedence, phase shifters, or on the constructuction and testing of the antenna. This should be done in the future to ensure the calculated and measured data are in agreement, and to determine any other incidental problems which may occur with this design.

REFERENCES

Chapter 1

[1] Borgford, M.I., "The Design of a Simple Prime Focus Feed for Paraboloid Reflectors Using Graphic CAD Techniques," M.Sc. Thesis, University of Manitoba, 1985.

[2] Burke, G.J., and Pogio, A.J., "Numerical Electromagnetic Code (NEC) - Method of Moments," - A user-oriented computer code for analysis of the electromagnetic response of antennas and other metal structures, Naval Ocean Systems Centre, Technical Document 116, January 1981.

[3] "ROTM" - Numerical code for the solution of scattering and antenna problems with rotationally symmetric geometry, Department of Electrical Engineering, University of Manitoba, 1984.

[4] Jorgensen, Rolf, "Manual for GRASP," - Single and dual reflector antenna program, Ticra A/S, Engineering Consultants, Denmark, 1983.

[5] Popovic, B.D, Dragovic, M.B., and Djordjevic, A.R., "Analysis and Synthesis of Wire Antennas," Research Studies Press, John Wiley & Sons Limited, 1982

[6] Foley, J.D., and Van Dam, A., "Fundamentals of Interactive Computer Graphics," Addison-Wesley Publishing Company, 1983, pp xi.

[7] For example: "REDCAD," - including Redlog Circuit Capture Software and Redboard PCB Design Software, Racal-Redac, Inc., Westford, MA, 1985.

[8] For example: "IDEA Series," - Computer-aided engineering tools for electronics design engineers doing VLSI, gate array, PCB, and systems design, Mentor Graphics Corporation, Beaverton, Oregon, 1982.

[9] For Example: Biddlecombe, C.S., and Trowbridge, C.W., "CAE of Electromagnetic Devices," Computer-Aided Engineering Journal, Volume 1, Number 3, April 1984, pp 84-90.

[10] Miller, E.K., et al, "Computer Graphics Applications in Electromagnetic Computer Modeling", IEE Conference Publication Number 195, Antennas and Propagation Conference, 1981, pp 373 - 377.

[11] "Calma" is a trademark of Calma Company, Milpitas, California, 1985.

[12] Rosenbrock, H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function," Computer Journal, Volume 3, No. 3, October, 1960, pp 175 - 184.

[13] Bandler, J.W. and Charalambous, C., "Practical Least p^{th} Optimization of Networks," IEEE Transactions on Microwave Theory and Techniques, Volume MTT-20, No. 12, December 1972, pp 834 - 840.

Chapter 2

[14] Harrington, R.F., "Field Computation by Moment Methods," The Macmillan Company, New York, 1972.

[15] "DDM & DIMENSION III: DAL User's Manual," Calma Company, Milpitas, California, 1985.

[16] "DDM & DIMENSION III: Guide to DAL-FORTRAN," Calma Company, Milpitas, California, 1985.

Chapter 3

[17] Box, "Nonlinear Optimization Techniques," Oliver & Boyd, Edinburgh, 1969.

[18] "RSNBRK" - Software implementation of Rosenbrock's minimization algorithm, developed at the Department od Electrical Engineering, University of Manitoba, 1983.

[19] Wilde, D.J., and Beightler, C.S., "Foundations of Optimization," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967.

Chapter 4

[20] Barry, A.L. "The Canadian MSAT Program" Conference Proceedings, First Canadian Domestic and International Satellite Communications Conference, 1983, pp 7.5.1-2

[21] Azarbar, B., "Telesat Canada," - Private Communication, 1984

[22] Woo, K., "Low Gain and Steerable Vehicle Antennas for Communications with Land Mobile Satellite," , 1982, pp B1.6.1-5.

[23] Butterworth, J.S., and Matt, E.E., "The Development of a Vehicle Antenna for Satellite Communications in the 800 MHz Band," May, 1982.

[24] Kaiser, Julius A., "The Archimedean Two-Wire Spiral Antenna", IRE Transactions on Antennas and Propagation, Volume AP-8, May, 1960, pp 312-323.

[25] Dyson, John D., "The Equiangular Spiral Antenna," IRE Transactions on Antennas and Propagation, Volume AP-7, April, 1959, pp 181-187.

[26] Dyson, John D., "New Circularly-Polarized Frequency-Independent Antennas with Conical Beam or Omnidirectional Patterns," IRE Transactions on Antennas and Propagation, Volume AP-9, July, 1961, pp 334-342.

[27] Ramsdale, P.A., and Crampton, P.W., "Low Frequency Performance of Hemispherical Coverage Conical Log-Spiral Antennas," IEE Conference Publication Number 195, Antennas and Propagation Conference, 1981, pp 298-302.

[28] Kilgus, Charles C., "Shaped-Conical Radiation Pattern Performance of the Backfire Quadrifilar Helix", IEEE Transactions on Antennas and Propagation, Volume AP-23, May, 1975, pp 392-397.

[29] Parasnis, K., Kumar, G., and Shafai, L., "A New Microstrip Antenna for the Generation of Higher Order Modes," IEEE AP-S International Symposium Digest 1985, pp. 79-82.

[30] Kumar, G., and Shafai, L., "Microstrip Phased Array Antennas for Mobile Satellite Communication," IEEE AP-S International Symposium Digest 1985, pp. 719-722.

[31] Davies, D.E.N., "Circular Arrays: Their Properties and Potential Applications," IEE Conference Publication Number 195, Antennas and Propagation Conference, 1981, pp 1-10.

[32] Harrington, R.F., "Time Harmonic Electromagnetic Fields," McGraw Hill Book Company, 1961

APPENDIX A

Directory of DAL and DAL-FORTRAN programs in the IGAD package

DAL Programs

ANTCALC.DA	- converts model to analysis routine input file
APLOAD.DA	- apply load to an existing segment
APVOLT.DA	- apply voltage source to existing segment
ARCHIM.DA	- create Archimedean spiral antenna with options
CENTROID.DA	- calculates centroid of patch (used with patch programs)
CPVPLOT.DA	- plot circularly polarized components (Rev 1)
CRETDI.DA	- create text-data items
CURPLOT.DA	- plot current data (Rev 1)
CURVE01.DA	- create segmented curve
DELCON2.DA	- disconnect a deleted segment
DELLOAD.DA	- delete a load
DELSEG.DA	- delete a segment
DELVOLT.DA	- delete an applied voltage source
FARPLOT.DA	- plot far field data (Rev 2)
FILEMODEL.DA	- file model data
GAINPLOT.DA	- plot gain (Rev 1)
GENGEOM.DA	- create model geometry from analysis routine input file (Cylinders)
GENGEOM2.DA	- create model geometry from input file (line and point method)
GENWIRE.DA	- create straight line of segments
GETCOORDS.DA	- get 3-d coordinates of a point
GETDIGCORD.DA	- get coordinates of a digitize
GETPCHPT.DA	- get coordinates of a patch corner
GETSDATA.DA	- get stored data for a segment
GETSEGEND.DA	- get coordinates of a segment endpoint
HEADER.DA	- program header
IGAD.DA	- initialize modeling environment
INITIALIZE.DA	- set up defaults
INTERPRET.DA	- create geometry from input equation (not finished)
MIRROR.DA	- mirror geometry
MOVESEG.DA	- move segments
NEARPLOT.DA	- plot near field data (Rev 1)
NEWGPLOT.DA	- plot gain data (Rev 2)
PATCH.DA	- create surface patch (not fully implemented)

PCHARB.DA	- arbitrarily shaped patch
PCHQUA.DA	- quadrilateraly shaped patch
PCHREC.DA	- rectangularly shaped patch
PCHRSURF.DA	- cover a rectangular surface with patches
PCHTRI.DA	- triangularly shaped patch
PLOT3D.DA	- 3 d plot (not fully implemented)
POINTTEST.DA	- test point data
PTEST.DA	- test patch routines
PTRNREQ.DA	- request radiation pattern calculations
READPLT.DA	- read in calculated data (DAL version)
SEGDATA.DA	- set segment data
SEGTEST.DA	- test segment routine
SETDEF.DA	- set defaults
SPLINE01.DA	- create wire from spline (Rev 1)
SPLINE02.DA	- create wire from spline (Rev 2)
STMODEL.DA	- start new model
TEST.DA	- test routine
TESTAIM.DA	- test routine
TESTDETCON.DA	- establish connectivity of wires
TESTHIDE.DA	- test hidden line removal (not implemented)
TESTREAD.DA	- read in calculated data (DAL-FORTRAN version)
USREQN.DA	- get user's equation for interpret program (not imlemented)
VECTMOVE.DA	- move segments by vector
VERLOAD.DA	- verify applied loads
VERSEG.DA	- verify selected segment
VERVOLT.DA	- verify applied voltage sources
WIRE02.DA	- create wire (Cylinder model)
WIRE03.DA	- create wire (line and point model)
XACALC.DA	- execute NAAR

DAL-FORTRAN Programs

DRUS1.FOR	- subroutine call director
RECDECODE.FOR	- decode input record
RECTEST.FOR	- test for record
UCRTDI.FOR	- create text-data item
UDATADD.FOR	- add data to text-data item
UDDFIT.FOR	- adjust data to fit within plot extents
UDETCON.FOR	- detect segment connectivity
UDHPTLST.FOR	- create horizontal point list
UDLOGX.FOR	- convert data to logarithms
UDNORM.FOR	- normalize data
UDRPTLST.FOR	- create radial point list
UDSCALE.FOR	- scale data
UDTMAX.FOR	- truncate data greater than maximum
UDTMIN.FOR	- truncate data less than minimum
UGDATA.FOR	- get data from text-data item
URPLOTDAT.FOR	- read plot data (Rev 1)
URPTREAD.FOR	- read plot data (Rev 2)
UTDID.FOR	- ident text-data item
UTEST.FOR	- test routine

APPENDIX B

RSNBRK

FORTRAN Implementation of Rosenbrock's Minimization Algorithm

```
      SUBROUTINE RSNBRK(U,V,W,S,D,IS,IF,XO,X,N,NMX,STEP,ALPHA,BETA,
     & NFEMAX,EXIT,KEXIT)
C     ID   : DEBUG PARAMETER
C          0 NO PRINT-OUT WILL BE GIVEN,
C          1 LIMITED PRINT-OUT WILL BE GIVEN,
C          2 FULL PRINT-OUT WILL BE GIVEN.
C     N    : NUMBER OF VARIABLES.
C
C     ALPHA: EXPANSION FACTOR (3.0 IS RECOMENDED).
C     BETA : PENALTY FACTOR FOR FAILURE (0.5 IS RECOMENDED).
C     NFEMX: NUMBER OF MAXIMUM FUNCTION EVALUATION.
C     STEP : INITIAL STEP SIZE (0.1 IS RECOMENDED).
C     KEXIT: EXIT CRITERIA OPTION
C          1 TO EXIT WITH THE IMPROVEMENT OF THE FUNCTION VALUE.
C          2 TO EXIT ROOT MEAN SQUARE OF THE IMPROVEMENTS IN ALL
C            DIRECTIONS.
C     EXIT : EXIT CRITERIA VECTOR
C          1ST ELEMENT IS TO USE WITH K=1
C          2ND ELEMENT IS TO USE WITH K=2.

      COMMON /IOUT/IN,IO,ID
      DIMENSION IF(10),IS(10)
      REAL*8 U(10,10),V(10,10),W(10,10),S(10),D(10),
     &     XO(10),X(10),EXIT(2),ALPHA,BETA,STEP,
     &     F,FO,SUM,ST
C
      WRITE(IO,120) ID,N,ALPHA,BETA,NFEMAX,KEXIT,STEP,
     &              (EXIT(I),I=1,2)
      WRITE(IO,130) (XO(I),I=1,N)
C
      DO 20 J=1,N
      DO 10 I=1,N
10        U(J,I)=0.0
20        U(J,J)=1.0
      NFE=0
      CALL RSN_FUNCT(FO,XO,NFE)
30    F=FO
      DO 35 I=1,N
35        S(I)=STEP
      IF (ID.LT.2) GO TO 50
      DO 40 I=1,N
40        IF (ID.GE.2) WRITE(IO,140) I,(U(I,J),J=1,N)
50    IF (ID.GE.1) WRITE(IO,150) NFE,FO,(XO(I),I=1,N)
      CALL RSRCH(FO,XO,X,U,S,D,IF,IS,N,NMX,ALPHA,BETA,NFE)
      IF (NFE.GT.NFEMAX) ID=2
      IF (NFE.GT.(NFEMAX+20)) GO TO 90
      SUM=0.0
      DO 60 I=1,N
60        SUM=SUM+D(I)*D(I)
      ST=SQRT(SUM/N)
      IF (KEXIT.EQ.1) GO TO 70
      IF (ST.LT.EXIT(2)) GO TO 100
      GO TO 80
70    IF (ABS(FO-F).LT.EXIT(1)) GO TO 100
80    CALL ROTATE(U,V,W,D,N,NMX)
      IF (ID.GE.2) WRITE(IO,160) (D(J),J=1,N)
```

```
      IF (ID.GE.2) WRITE(IO,170) (S(J),J=1,N)
      GO TO 30
90    IF (ID.GE.0) WRITE(IO,180) FO,NFE
      GO TO 110
100   IF (ID.GE.0) WRITE(IO,190) FO,NFE
110   WRITE(IO,200) (XO(I),I=1,N)
      RETURN
120   FORMAT (1H1//1X,70('-')/2X,'ROSENBROCK METHOD OF ROTATING ',
     & 'CO-ORDINATES'/1X,70('-')///5X,'INPUT DATA:'/5X,11('-')//5X,
     & 'ID    = ',I3/5X,'N     = ',I5/5X,'ALPHA = ',F10.5/5X,
     & 'BETA  = ',F10.5/5X,'NFEMX = ',I5/5X,
     & 'KEXIT = ',I3/5X,'STEP  = ',F10.5/5X,
     & 'EXIT  = ',F12.7/11X,F12.7/)
130   FORMAT (1X,10F10.6)
140   FORMAT (2X,'U(',I2,')=',10F10.6)
150   FORMAT (/2X,'NFE=',I4,2X,'FO=',F13.7,2X,'XO=',10F10.5)
160   FORMAT (2X,'D(J)=',10F10.5)
170   FORMAT (2X,'S(J)=',10F10.6)
180   FORMAT (//1X,'PROGRAM STOPPED AT F=',F13.6,/2X,'BECAUSE OF EXCESS
     & FUNCTION EVALUATIONS. NFE=',I3//)
190   FORMAT (//1X,'MINIMUM FUNCTION VALUE OF',F13.7/2X,'IS FOUND IN ',
     & I4,' FUNCTION EVALUATIONS AT '/)
200   FORMAT (2X,'X=',10F10.6)
      END

      SUBROUTINE RSRCH(FO,XO,X,U,S,D,IF,IS,N,NMX,ALPHA,BETA,NFE)
      COMMON /IOUT/IN,IO,ID
      DIMENSION IF(10),IS(10)
      REAL*8 U(10,10),S(10),D(10),
     &       XO(10),X(10),ALPHA,BETA,FO,F
      DO 10 J=1,N
         IS(J)=0
         IF(J)=0
10       D(J)=0.0
20    DO 80 J=1,N
         DO 30 I=1,N
30          X(I)=XO(I)+S(J)*U(J,I)
         CALL RSN_FUNCT(F,X,NFE)
         IF (ID.GE.2) WRITE(IO,90) J,NFE,F,(X(I),I=1,N)
         IF (F.LE.FO) GO TO 40
         IF (IS(J).EQ.1) IF(J)=1
         S(J)=-BETA*S(J)
         GO TO 60
40       IS(J)=1
         D(J)=D(J)+S(J)
         S(J)=ALPHA*S(J)
         FO=F
         DO 50 I=1,N
50       XO(I)=X(I)
60       DO 70 JJ=1,N
70       IF (IF(JJ)*IS(JJ).EQ.0) GO TO 80
         RETURN
80       CONTINUE
      GO TO 20
90    FORMAT (/2X,'J=',I2,2X,'NFE=',I3,2X,'F=',F13.7/2X,'X=',10F10.5)
      END
```

```
      SUBROUTINE ROTATE(U,V,W,D,N,NMX)
      REAL*8 U(10,10),V(10,10),W(10,10),D(10),SQ,RSQ,SUM
      DO 10 I=1,N
10    V(N,I)=D(N)*U(N,I)
      DO 30 J=2,N
          K=N-J+1
          K1=K+1
          DO 20 I=1,N
20            V(K,I)=D(K)*U(K,I)+V(K1,I)
30        CONTINUE
      SQ=0.0
      DO 40 I=1,N
          W(1,I)=V(1,I)
40        SQ=SQ+W(1,I)**2
      RSQ=1.0/SQRT(SQ)
      DO 50 I=1,N
50        U(1,I)=W(1,I)*RSQ
      DO 110 J=2,N
          SQ=0.0
          DO 60 I=1,N
60            W(J,I)=V(J,I)
          K=J-1
          DO 80 L=1,K
              SUM=0.0
              DO 70 I=1,N
70                SUM=SUM+V(J,I)*U(L,I)
              DO 80 I=1,N
80                W(J,I)=W(J,I)-SUM*U(L,I)
          DO 90 I=1,N
90            SQ=SQ+W(J,I)**2
          IF (SQ.EQ.0.0) SQ = 1.0E-10
          RSQ=1.0/SQRT(SQ)
          DO 100 I=1,N
100           U(J,I)=W(J,I)*RSQ
110       CONTINUE
      RETURN
      END
```

APPENDIX C

RSN_FUNCT

FORTRAN Implementation of the Objective Function

for the Numerical Optimization Routine RSNBRK

```
      SUBROUTINE RSN_FUNCT(FUNC,VRBL,NFE)
      COMMON/IOUT/IN,IO,ID
      REAL TT(1600),PT(1600),GT(1600)
      REAL*8 VRBL(10),FUNC
      REAL TH(500),PH(500),ETHM(500),ETHA(500),EPHM(500),EPHA(500)
      REAL RV(500),LV(500),EL(500)
      CHARACTER  CURDATE*9,CURTIME*8,A*72
C
      NFE = NFE + 1
C
      ALPH1 = ABS(VRBL(1))
C
      ALEN1 = ABS(VRBL(2))
      IF (ALEN1.LT.0.02) THEN
          FUNC = 1/ALEN1*10
          RETURN
      ELSE IF (ALEN1.GT.0.23) THEN
          FUNC = (ALEN1-0.23) * 50000
          RETURN
      END IF
C
      ALPH2 = ABS(VRBL(3))
C
      ALEN2 = .24 - ALEN1
C
      HITE = ABS(VRBL(4))
C
      PMAX = 0.0
      VC = ALEN1*SIN(ALPH1)+ALEN2*SIN(ALPH2)
      IF (VC.GE.HITE) THEN
          VPEN = (VC-HITE)*50000.
          PMAX = VPEN
      END IF
      HC = ALEN1*COS(ALPH1) + ALEN2*COS(ALPH2)
      IF (HC.LE.0.0) THEN
          HPEN = (0.01-HC) * 50000.
          IF (HPEN.GT.PMAX) PMAX = HPEN
      END IF
      IF (PMAX.NE.0.) THEN
          FUNC = PMAX
          RETURN
      END IF
      OPEN(22,FILE='ANTCAL:ACALC.INP',STATUS='UNKNOWN')
      CALL DATE(CURDATE)
      CALL TIME(CURTIME)
      WRITE(IN,552)
552   FORMAT(//,10X,'***** RSN_FUNC FUNCTION EVALUATION *****')
      WRITE(IN,551)CURDATE,CURTIME
551   FORMAT(/,' DATE: ',A9,'  TIME: ',A8,/)
      WRITE(IN,*)' VARIABLES:   ALPH1 = ',ALPH1
      WRITE(IN,*)'              ALEN1 = ',ALEN1
      WRITE(IN,*)'              ALPH2 = ',ALPH2
      WRITE(IN,*)'              ALEN2 = ',ALEN2
      WRITE(IN,*)'              HITE  = ',HITE
      WRITE(IN,*)' EVALUATION NUMBER: ',NFE
      WRITE(22,111)'CM ',CURDATE,' ',CURTIME
```

```
111     FORMAT(A3,A9,A1,A8)
        AH = HITE
        CH = HITE-.01
        WRITE(22,110)'CE OPTIMIZATION TEST'
        WRITE(22,115)'GW1,1,-0.01,0.0,',AH,',0.01,0.0,',AH,',0.005'
        WRITE(22,115)'GW2,1,0.0,-0.01,',CH,',0.0,0.01,',CH,',0.005'
115     FORMAT(A16,F7.4,A10,F7.4,A6)
        XINIT = 0.01
        ZINIT = HITE
        C = ','
        ONE = '1'
        GW = 'GW'
        NSEG = 3
        NSEG1 = INT(ALEN1/.05) + 1
        NSEG2 = INT(ALEN2/.05) + 1
        WR = 0.005
        AX1 = ALEN1/(FLOAT(NSEG1))*COS(ALPH1)
        AZ1 = ALEN1/(FLOAT(NSEG1))*SIN(ALPH1)
        AX2 = ALEN2/(FLOAT(NSEG2))*COS(ALPH2)
        AZ2 = ALEN2/(FLOAT(NSEG2))*SIN(ALPH2)
        DO I = 1,NSEG1
            XFIN = XINIT + AX1
            ZFIN = ZINIT - AZ1
            XA = XINIT
            YA = 0.0
            ZA = ZINIT
            XB = XFIN
            YB = 0.0
            ZB = ZFIN
            WRITE(22,100)GW,NSEG,C,ONE,C,XA,C,YA,C,ZA,C,XB,C,YB,C,ZB,C,WR
            NSEG = NSEG + 1
            XA = (-XA)
            XB = (-XB)
            WRITE(22,100)GW,NSEG,C,ONE,C,XA,C,YA,C,ZA,C,XB,C,YB,C,ZB,C,WR
            NSEG = NSEG + 1
            XA = 0.0
            YA = XINIT
            ZA = ZINIT - .01
            XB = 0.0
            YB = XFIN
            ZB = ZFIN - .01
            WRITE(22,100)GW,NSEG,C,ONE,C,XA,C,YA,C,ZA,C,XB,C,YB,C,ZB,C,WR
            NSEG = NSEG + 1
            YA = (-YA)
            YB = (-YB)
            WRITE(22,100)GW,NSEG,C,ONE,C,XA,C,YA,C,ZA,C,XB,C,YB,C,ZB,C,WR
            NSEG = NSEG + 1
            XINIT = XFIN
            ZINIT = ZFIN
        END DO
        DO I = 1,NSEG2
            XFIN = XINIT + AX2
            ZFIN = ZINIT - AZ2
            XA = XINIT
            YA = 0.0
            ZA = ZINIT
            XB = XFIN
```

```
                YB = 0.0
                ZB = ZFIN
                WRITE(22,100)GW,NSEG,C,ONE,C,XA,C,YA,C,ZA,C,XB,C,YB,C,ZB,C,WR
                NSEG = NSEG + 1
                XA = (-XA)
                XB = (-XB)
                WRITE(22,100)GW,NSEG,C,ONE,C,XA,C,YA,C,ZA,C,XB,C,YB,C,ZB,C,WR
                NSEG = NSEG + 1
                XA = 0.0
                YA = XINIT
                ZA = ZINIT - .01
                XB = 0.0
                YB = XFIN
                ZB = ZFIN - .01
                WRITE(22,100)GW,NSEG,C,ONE,C,XA,C,YA,C,ZA,C,XB,C,YB,C,ZB,C,WR
                NSEG = NSEG + 1
                YA = (-YA)
                YB = (-YB)
                WRITE(22,100)GW,NSEG,C,ONE,C,XA,C,YA,C,ZA,C,XB,C,YB,C,ZB,C,WR
                NSEG = NSEG + 1
                XINIT = XFIN
                ZINIT = ZFIN
            END DO
            WRITE(22,555)'GW40,3,0.0,0.0,0.0,0.0,0.0,',HITE-0.03,',0.005'
555         FORMAT(A,F7.4,A)
            WRITE(22,110)'GE1'
            WRITE(22,110)'GN1'
            WRITE(22,110)'PT-2'
            WRITE(22,110)'EX0,1,1,0,1.0,0.0'
            WRITE(22,110)'EX0,2,1,0,0.0,1.0'
            WRITE(22,110)'RP0,91,1,1500,0.0,0.0,1.0,0.0'
            WRITE(22,110)'EN'
100         FORMAT(A2,I3,2A1,7(A1,F8.4))
110         FORMAT(A)
            CLOSE(22)
            CALL ACALCSUB  ! Run numerical analysis routine
C
C  Now get the calculated data and create an object function.
C
C  Read in gain data...
C
            OPEN(11,FILE='ANTCAL:ACALC.GAI',STATUS='UNKNOWN')
            READ(11,*) GAINMAX
            I = 1
520         READ(11,*,END=521)TT(I),PT(I),GT(I),TT(I+1),PT(I+1),
     &                        GT(I+1),TT(I+2),PT(I+2),GT(I+2)
            I = I + 3
            GOTO 520
521         CLOSE (11)
            NPTS = I - 3
            DO 522 I = 1,NPTS
                IF (GT(I).EQ.0.0) THATMAX = TT(I)
522         CONTINUE
            WRITE(IN,*)'GAINMAX = ',GAINMAX,' AT THETA = ',THATMAX
            gpen1 = 0.
            gpen2 = 0.
            gpen3 = 0.
```

```
      DO 523 I = 1,NPTS
          IF (TT(I).EQ.61.) THEN
              IF(GT(I).LT.0.) THEN
                  Write(in,*)'theta = ',tt(i)
                  write(in,*)'gain = ',gt(i)
                  GPEN1 = (-GT(I))*20.
              END IF
          END IF
c         IF (TT(I).EQ.1.) THEN
c             Write(in,*)'theta = ',tt(i)
c             write(in,*)'gain = ',gt(i)
c             if (gt(i).gt.(-10.)) gpen2 = 10. + gt(i)
c         END IF
c         IF (TT(I).EQ.89.) THEN
c             Write(in,*)'theta = ',tt(i)
c             write(in,*)'gain = ',gt(i)
c             if (gt(i).gt.(-10.)) gpen3 = 10. + gt(i)
c         END IF
523   CONTINUE
      WRITE(IN,*)'GAIN PENALTIES:',GPEN1,gpen2,gpen3
C
C Now read in data about RHCPV,LHCPV and ellipticity from ACALC.CPV
C
      OPEN(12,FILE='DUA2:[USER.NEILSON.ACALC]ACALC.CPV',
     &                    STATUS='UNKNOWN')
      I = 1
603   READ(12,*,END=604)TH(I),PH(I),RV(I),LV(I),EL(I)
      I=I+1
      GOTO 603
604   NPTS = I-1
      CLOSE(12)
      WAEL44 = 100.0
      WAEL48 = 150.0
      WAEL52 = 200.0
      WAEL56 = 250.0
      WAEL60 = 300.0
      WAEL64 = 250.0
      WAEL68 = 200.0
      WAEL72 = 250.0
      WAEL76 = 100.0
      FL44=0.
      FL48=0.
      FL52=0.
      FL56=0.
      FL60=0.
      FL64=0.
      FL68=0.
      FL72=0.
      FL76=0.
      DO 605 I=1,NPTS
          IF(FL44.EQ.0.)THEN
              IF(TH(I).EQ.44.) THEN
                  FL44=1.
                  AEL44=0.0
                  IF (EL(I).LT.0.75) AEL44 = (0.75-EL(I)) * WAEL44
                  IF (EL(I).LT.1.0) AEL44 = AEL44 + (1.0-EL(I))* WAEL44/10.
                  WRITE(IN,*)'AEL44 = ',AEL44,'  THETA = ',TH(I),'  ELIP = ',EL(I)
```

```
            GOTO 605
        END IF
    END IF
    IF(FL48.EQ.0.)THEN
        IF(TH(I).EQ.48.) THEN
            FL48=1.
            AEL48 = 0.0
            IF(EL(I).LT.0.75) AEL48 = (0.75-EL(I)) * WAEL48
            IF(EL(I).LT.1.0) AEL48 = AEL48 + (1.0-EL(I)) * WAEL48/10.
            WRITE(IN,*)'AEL48 = ',AEL48,' THETA = ',TH(I),' ELIP = ',EL(I)
            GOTO 605
        END IF
    END IF
    IF(FL52.EQ.0.)THEN
        IF(TH(I).EQ.52) THEN
            FL52=1.
            AEL52 = 0.0
            IF(EL(I).LT.0.75) AEL52 = ABS(0.75-EL(I)) * WAEL52
            IF(EL(I).LT.1.0) AEL52 = AEL52 + (1.0-EL(I)) * WAEL52/10.
            WRITE(IN,*)'AEL52 = ',AEL52,' THETA = ',TH(I),' ELIP = ',EL(I)
            GOTO 605
        END IF
    END IF
    IF(FL56.EQ.0.)THEN
        IF(TH(I).EQ.56.) THEN
            FL56=1.
            AEL56=0.0
            IF (EL(I).LT.0.75) AEL56 = (0.75-EL(I)) * WAEL56
            IF (EL(I).LT.1.0) AEL56 = AEL56 + (1.0-EL(I))* WAEL56/10.
            WRITE(IN,*)'AEL56 = ',AEL56,' THETA = ',TH(I),' ELIP = ',EL(I)
            GOTO 605
        END IF
    END IF
    IF(FL60.EQ.0.)THEN
        IF(TH(I).EQ.60.) THEN
            FL60=1.
            AEL60 = 0.0
            IF(EL(I).LT.0.75) AEL60 = (0.75-EL(I)) * WAEL60
            IF(EL(I).LT.1.0) AEL60 = AEL60 + (1.0-EL(I)) * WAEL60/10.
            WRITE(IN,*)'AEL60 = ',AEL60,' THETA = ',TH(I),' ELIP = ',EL(I)
            GOTO 605
        END IF
    END IF
    IF(FL64.EQ.0.)THEN
        IF(TH(I).EQ.64) THEN
            FL64=1.
            AEL64 = 0.0
            IF(EL(I).LT.0.75) AEL64 = ABS(0.75-EL(I)) * WAEL64
            IF(EL(I).LT.1.0) AEL64 = AEL64 + (1.0-EL(I)) * WAEL64/10.
            WRITE(IN,*)'AEL64 = ',AEL64,' THETA = ',TH(I),' ELIP = ',EL(I)
            GOTO 605
        END IF
    END IF
    IF(FL68.EQ.0.)THEN
        IF(TH(I).EQ.68.) THEN
            FL68=1.
            AEL68=0.0
```

```
                    IF (EL(I).LT.0.75) AEL68 = (0.75-EL(I)) * WAEL68
                    IF (EL(I).LT.1.0) AEL68 = AEL68 + (1.0-EL(I))* WAEL68/10.
                    WRITE(IN,*)'AEL68 = ',AEL68,' THETA = ',TH(I),' ELIP = ',EL(I)
                    GOTO 605
                END IF
            END IF
            IF(FL72.EQ.0.)THEN
                IF(TH(I).EQ.72.) THEN
                    FL72=1.
                    AEL72 = 0.0
                    IF(EL(I).LT.0.75) AEL72 = (0.75-EL(I)) * WAEL72
                    IF(EL(I).LT.1.0) AEL72 = AEL72 + (1.0-EL(I)) * WAEL72/10.
                    WRITE(IN,*)'AEL72 = ',AEL72,' THETA = ',TH(I),' ELIP = ',EL(I)
                    GOTO 605
                END IF
            END IF
            IF(FL76.EQ.0.)THEN
                IF(TH(I).EQ.76) THEN
                    FL76=1.
                    AEL76 = 0.0
                    IF(EL(I).LT.0.75) AEL76 = ABS(0.75-EL(I)) * WAEL76
                    IF(EL(I).LT.1.0) AEL76 = AEL76 + (1.0-EL(I)) * WAEL76/10.
                    WRITE(IN,*)'AEL76 = ',AEL76,' THETA = ',TH(I),' ELIP = ',EL(I)
                    GOTO 605
                END IF
            END IF
605     CONTINUE
        PMAX = 0.0
        IF(GPEN1.GT.PMAX) PMAX = GPEN1
        IF(GPEN2.GT.PMAX) PMAX = GPEN2
        IF(GPEN3.GT.PMAX) PMAX = GPEN3
        IF(AEL44.GT.PMAX) PMAX=AEL44
        IF(AEL48.GT.PMAX) PMAX=AEL48
        IF(AEL52.GT.PMAX) PMAX=AEL52
        IF(AEL56.GT.PMAX) PMAX=AEL56
        IF(AEL60.GT.PMAX) PMAX=AEL60
        IF(AEL64.GT.PMAX) PMAX=AEL64
        IF(AEL68.GT.PMAX) PMAX=AEL68
        IF(AEL72.GT.PMAX) PMAX=AEL72
        IF(AEL76.GT.PMAX) PMAX=AEL76
        IF (PMAX.GT.0.) THEN
            SUM = (GPEN1/PMAX)**10
            SUM = sum + (GPEN2/PMAX)**10
            SUM = sum + (GPEN3/PMAX)**10
            SUM = SUM + (AEL44/PMAX)**10
            SUM = SUM + (AEL48/PMAX)**10
            SUM = SUM + (AEL52/PMAX)**10
            SUM = SUM + (AEL56/PMAX)**10
            SUM = SUM + (AEL60/PMAX)**10
            SUM = SUM + (AEL64/PMAX)**10
            SUM = SUM + (AEL68/PMAX)**10
            SUM = SUM + (AEL72/PMAX)**10
            SUM = SUM + (AEL76/PMAX)**10
            FUNC = PMAX * SUM**0.1
        ELSE
            FUNC = 0.
        END IF
        WRITE(IN,*)'FUNC = ',FUNC
        RETURN
        END
```

APPENDIX D

PHASE

FORTRAN Implementation of Phase Shift Calculation Program

```
      Integer phi,phinot,i
      Real radarg,delta,aa,bb,cc,dd,ee,ff
      Write(1,11)
11    Format(//15x,'Calculated Phase Shift Table   (a = 0.45)',//
     &      ,5x,'Beam   Element  Phase    Real    Imaginary'/
     &      ,5x,'Angle   Angle   Shift    Part     Part'/)
c
      Do phi = 0,330,30          ! phi is the angle of main beam
c
         Do phinot = 0,300,60        ! phinot is angle of element
c
            radarg = (3.14159/180)*(phinot-phi)  ! convert to radians
            delta = (-140.) * cos(radarg)      ! val = -a*sin(theta)*360
c  -187 for a = 0.6
c  -156 for a = 0.5
c  -140 for a = 0.45
c  -125 for a = 0.4
            radarg = (3.14159/180)*delta       ! convert to radians again
            aa = cos(radarg)                ! real part of appl'd volt
            bb = sin(radarg)                ! imag. part for 1st dipole
            radarg = (3.14159/180)*(90.+delta)   ! add 90 deg shift for 2nd
            cc = cos(radarg)                ! real part of appl'd volt
            dd = sin(radarg)                ! imag part
            Write(1,12)phi,phinot,delta,aa,bb
            Write(1,13)90.+delta,cc,dd
         End do
      End do
12    Format(6x,i3,8x,i3,5x,f6.1,4x,f6.3,6x,f6.3)
13    Format(25x,f6.1,4x,f6.3,6x,f6.3/)
      Stop
      End
```