DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/99J-01

BISTATIC RADAR CROSS SECTION (RCS) CHARACTERIZATION OF COMPLEX OBJECTS

THESIS

Robert L Eigel, Jr., Captain, USAF

AFIT/GE/ENG/99J-01

Approved for public release, distribution unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government

AFIT/GE/ENG/99J-01

BISTATIC RADAR CROSS SECTION (RCS) CHARACTERIZATION OF COMPLEX OBJECTS

Robert L. Eigel, Jr., B.S.E.E.
Captain, USAF

Approved:

Dr. Andrew Terzuoli, Jr.
Chairman

4 JUN 99
date

Major Peter Collins, Ph. D.
Committee Member

4 JUN 99
date

Vittal P. Pyati
Dr. Vittal Pyati
Committee Member

4 JUN 99
date

AFIT/GE/ENG/99J-01

BISTATIC RADAR CROSS SECTION (RCS) CHARACTERIZATION OF COMPLEX OBJECTS

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Masters of Science in Electrical Engineering

Robert L. Eigel, Jr., B.S.E.E

Captain, USAF

June 1999

Approved for public release, distribution unlimited

ACKNOWLEDGMENTS

The purpose of this research was to evaluate Xpatch V2.4d's bistatic predictive capabilities and investigate Kell's and Crispin's monostatic-to-bistatic equivalence theorems' applicability to complex body scattering. The primary motivation stems from a desire to reduce data collection requirements for target recognition purposes, especially in the bistatic arena. Several simple targets have been studied in previous works, but this study concentrates on more complex, real world objects. If bistatic signatures in general are better understood, future analytical endeavors can prove more fruitful. Therefore, as a corollary, the work proposes several rules of thumb for predicting bistatic signatures during the analysis.

Numerous individuals expended a great deal of effort to accomplish this piece and I wish to express my gratitude to each of them. First I'd like to thank Dr. Andrew Terzuoli, my thesis advisor, whose experience and wisdom proved invaluable in all phases of this project. The committee members, Dr. Vital Pyati, Maj. Pete Collins, and Dr. Gary Thiele, and the research sponsors at ASC/ENAD, Mr. Enrique Mendez and Fred Tokarz all provided intelligent insight whenever it was needed. A special thanks to Jim Schmitz and the Demaco technical support crew who provided the Xpatch software and helped get it running on our computers. Thanks also goes to Lt Jeff Bartlett, Matt Zegler, and Tom Kauffman from AFRL/SNAD who developed and supplied the CAD and Xpatch models. Dr. John Nehrbass and the staff at ASC's Major Shared Resource Center (MSRC) provided countless hours of technical support for the Xpatch installation and maintenance on MSRC computers. I'd also like to thank Capt. Kelce Wilson and Maj. Bill Wood who took the time to lend a hand when it was needed. Special thanks also goes to Bill Kent, Byron Welsh, Lisa Wurst, Rob Layden, and especially Pete Bierely from Mission Research Corporation (MRC) who not only performed many of the test measurements, but also provided valuable insight into bistatic scattering concepts used throughout this work. Last but certainly not least my heartfelt thanks goes out to Dr. Guisppe Nesti and his extraordinary staff at the European Joint Research Center who was instrumental in providing the bistatic measurements upon which most of the analysis rests.

...And lest I forget, a warm and heartfelt thanks goes to my wife, Chrissy, whose patience and fortitude made all this possible...

Robert L. Eigel, Jr.

Table of Contents

List of Figures

List of Tables

Abstract

Radar development over the past fifty years has focused on two primary radar technologies: monostatic and bistatic. Traditionally, monostatic radars perform as the sensor backbone of modern offensive and defensive weapons systems and signature collection methods. While predominant, the monostatic scenario is less general than the bistatic and therefore is seen as more restrictive. This restriction prevents engineers from exploiting bistatic's numerous advantages, however, it does provide a welcome simplification to the vastly more complex bistatic situation. The shear volume of data generated during a bistatic engagement or measurement scenario imposed severe limitations on its utility. Recent technological advances, however, have made the immense data load more manageable, so the opportunity to exploit bistatic's inherent geometric flexibility is becoming more plausible. From an operational perspective, development of large sensor arrays which support numerous, distributed passive and active elements is more feasible. From a signature analysis perspective, an expanded feature set incorporating bistatic data may lead to more robust target identification models. The motivation for this project is in support of this later application.

This work focuses on the bistatic nature of complex objects. To fully exploit bistatic's advantages requires a better understanding of bistatic scattering mechanisms and the available tools which support bistatic signature prediction and analysis. Without such a proper understanding, exploitation of bistatic's key benefits may not be fully realized.

First this paper addresses some phenomenological aspects of bistatic scattering from several simple and complex objects with an emphasis on the delineation between specular and non-specular effects. It attempts to accomplish two goals: 1) to evaluate the bistatic prediction performance of Xpatch2.4d and 2) assess the accuracy of Kell's and Crispin's monostatic-to-bistatic equivalence theorems. In doing so, some phenomenological aspects of bistatic scattering analysis are reviewed and some rules of thumb for bistatic signature analysis proposed.

A flat plate (Object A) and two canonically-based complex geometries (Object B & C) are chosen for testing. One of the complex objects has a significant shadowing and cavity feature (Object B) which are not present on the other (Object C) The European Joint Research Center (JRC) and Mission Research Corporation (MRC) provide monostatic and bistatic measurement data between 7-15 GHz and Xpatch generates simulation data for comparison. The flat plate is illuminated broadside to produce strong specular and diffraction effects and the two complex objects are illuminated off-axis to produce specular, multi-bounce, and surface wave phenomena. Ray tracing, modal analysis, and data averaging are used to help describe the bistatic signatures. Original Matlab script and function files format, manipulate, and display the required information.

Conclusions can be summarized in two sections, one describing the Xpatch analysis, the other pertaining to Kell's and Crispin's equivalence theorems.

Upon comparison of the Xpatch and measured data for the flat plate, one immediately notices a large discontinuity in the Xpatch predictions at edge-on receiver look angles. This indicates a problem in the Xpatch prediction engine and is probably due to limitations in the technique used to implement the PO/PTD solution. Although Xpatch is designed to predict scattering from electrically large objects, Xpatch data for Object C yields surprising correlation to the measured data despite its apparent small size. Specular reflections dominate its bistatic signature and this observation becomes one of the key findings. When the other complex geometry is studied, one finds that its large

shadowing feature allows non-specular effects of similar magnitude to arise and help form the signature structure. Xpatch prediction correlation subsequently diminishes for this object because it does not predict non-specular effects. One also notices that Xpatch data tends to be shifted or skewed slightly toward larger bistatic angles. The reason for this remains uncertain, although

Kell's and Crispin's monostatic-to-bistatic equivalence theorems (MBETs) are reviewed in a similar fashion. MBET signatures are extracted from monostatic measured data and compared to measured bistatic data in their raw form, after a 5 degree sliding average window has been applied, and after a 9 degree sliding average window has been applied. The averaging is expected to improve the performance of the MBETs, but this is not always the case.

The geometric complexity of the object determines which scattering mechanisms dominate the scattered field, and it is this characteristic which dictates the appropriateness of the MBET prediction. Both MBETs predict purely specular activity from the flat plate (*simple* object) fairly well for bistatic angles less than 30-40 degrees, but the dual specular scattering of Object C (*minimally complex* object) decreases MBET performance to bistatic angles of 15-20 degrees, and the specular/non-specular interactions of Object B (*rigorously complex* object) make the MBETs useful for bistatic angles of only 5-10 degrees. MBET predictions at larger bistatic angles tend to be lower than measured data for the *minimally complex* object and higher than measured for the *rigorously complex* object. The discrepancies are primarily due to the changing nature of the scattering centers as a function of bistatic angle. Geometries which support wide lobewidth specular reflections exhibit less variation in the nature/existence of the scattering centers than those who support specular and non-specular effects, thus leading to better correlation between MBET and measured RCS. Averaging improves MBET correlation noticeably for the specular reflection of the flat plate, minimally for Object C's dual specular reflections, and not at all for Object B's specular/non-specular signature. Kell's MBET is slightly better at predicting the amplitude of diffraction components from a simple shape, but neither has the advantage when predicting scattering from the complex objects. Kell's formula also requires a larger monostatic data set than Crispin's to predict the same angular extent of bistatic RCS and suffers from a degradation in angular resolution near the transmitter illumination angle.

BISTATIC RADAR CROSS SECTION (RCS) CHARACTERIZATION

OF COMPLEX OBJECTS

I. Introduction

Radar development over the past fifty years has focused on two primary radar technologies: monostatic and bistatic. Monostatic refers to a radar configuration in which the transmit and receive antennas are co-located if not physically one in the while the bistatic situation places no restriction on the placement of either antenna in relation to each other or the target (Fig. 1). Traditionally, monostatic development has far surpassed that of bistatic. monostatic radars perform as the sensor backbone of modern offensive and defensive weapons systems and signature collection methods. While predominant, the monostatic scenario is less general than the bistatic and therefore is seen as more restrictive. This restriction prevents engineers from exploiting bistatic's inherent advantages such as, glint reduction, clutter tuning, radar cross section (RCS)

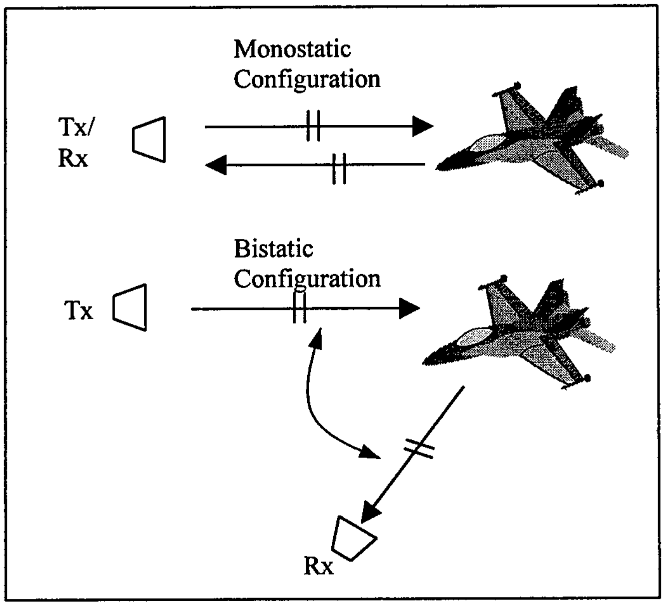


Figure 2: Monostatic vs. Bistatic Geometry

enhancement, improved semi-active seeker accuracy, covert operations, counter-ARM, and counter retro-directive jamming [10]. Yet the bistatic engagement or collection scenario's overwhelming complexity and immense data processing requirement have imposed severe limitations on its utility. Relegating analysis to the monostatic domain does provide a welcome simplification, however, and most radar development has occurred within this context since World War II. Recent technological advances, however, have made the immense data load more manageable, so the opportunity to exploit bistatic's inherent geometric flexibility is becoming more plausible. Dramatic increases in computer throughput, advanced algorithm development, software efficiency enhancements, and cost reductions have played a major role in making this technology a viable alternative. From an operational perspective, large sensor arrays which support numerous, distributed passive and active elements are more feasible. From a signature analysis perspective, an expanded feature set incorporating bistatic data may lead to more robust target identification models. The motivation for this project is primarily in support of this later application, but the data presented herein may find some utility in the former.

This work focuses on the bistatic signature nature of complex objects. To fully exploit bistatic's advantages requires a better understanding of bistatic scattering mechanisms and the available tools which support bistatic signature prediction and analysis. Without such a proper understanding, exploitation of bistatic's key benefits may not be fully realized. The scattering mechanisms which interact to form an object's far field signature pertain to both monostatic and bistatic situations. However, whereas the monostatic pattern is usually dominated by specular returns, the bistatic can be dominated by non-specular ones. The vast array of computational methods which predict these mechanisms can predict signatures from numerous perspectives, but some don't predict the non-specular mechanisms well, if at all, leading to a possible misapplication of the method in certain bistatic situations. Just as radar development has been accomplished under a primarily monostatic paradigm, so too has signature analysis. Any proper treatment of bistatics must begin with a good understanding of 1) the scattering mechanisms in the bistatic arena and 2) an understanding of the bistatic limitations within the chosen prediction code or method.

Most computational electromagnetic codes find their roots in theory applicable to both monostatic and bistatic situations, but as is typical not all codes are created equal. Some offer a more rigorous treatment to signature development at the cost of a high computational burden, but from a bistatic perspective this may be warranted, especially for complex objects. Usually a code's validity is in part, established by its performance compared against measured data. While monostatic data is plentiful for comparative purposes, true bistatic data is rare and evaluations of a code's bistatic predictive prowess are few and far between. Equally as rare are evaluations of the utility and appropriateness of several common monostatic-to-bistatic equivalence theorems (MBETs). Despite the plethora of advanced prediction methods, typically a simpler, faster solution is more desirable, especially if there is a minimum loss of information, accuracy, or precision. Both the MBETs and the US Air Force's high-frequency code, Xpatch 2.4d, offer this quicker, simpler prediction option, but their limitations need to be quantified.

Problem Statement: This paper addresses three basic issues: 1) it explores the phenomenological aspects of bistatic scattering from several simple and complex objects with an emphasis on the delineation between specular and non-specular effects, 2) it appraises the effectiveness and utility of the monostatic-to-bistatic equivalence theorems for predicting the signatures of complex objects, and 3) it evaluates the accuracy of the bistatic signature prediction for a popular commercial scattering code, Xpatch v2.4d. In conclusion some rules of thumb are proposed to help guide the reader in evaluating the bistatic RCS of complex shapes in general

To achieve this end three test objects (Fig. 2) are studied through computation and direct measurement of the RCS. The SLICY targets are derivatives of the SLICY model investigated in [11], and all targets are chosen for their ability to highlight certain scattering mechanisms as well as similarity to previously investigated and well documented objects. Direct monostatic and bistatic measurements of each object at several frequencies are compared to Xpatch predictions. Xpatch does an excellent job of predicting specular returns, even from complex geometries. However, one clearly sees through the analysis in Section IV that Xpatch 2.4d has difficulty in predicting edge diffraction and surface wave phenomena from a bistatic perspective. The bistatic

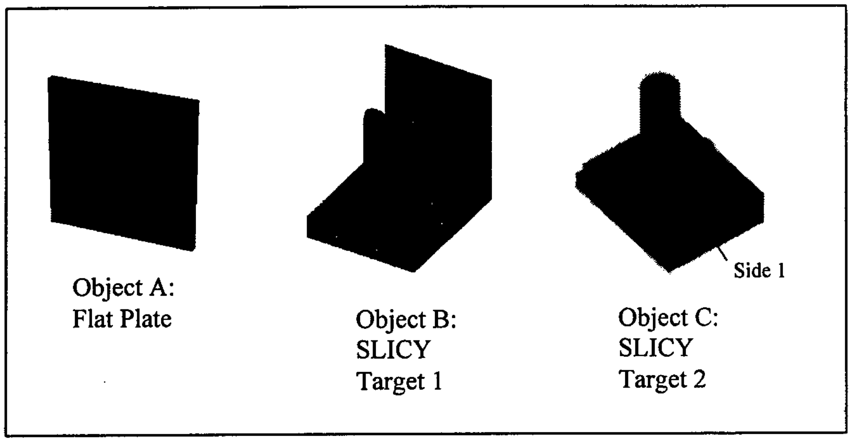


Figure 2: Test objects

signature may be dominated by these effects for certain observation angles and frequencies leading to a shortfalls in Xpatch's bistatic computational prowess. This problem (especially the surface wave difficulties) is somewhat mitigated by an understanding of Xpatch's limitation as a high frequency code and an appropriate utilization of the software within that context. However, the strong discontinuities apparent in a flat plate's bistatic signature pattern cannot be easily resolved, leading one to question not only theoretical basis of the software, but also its implementation within the code.

The bistatic signature as predicted by two common MBETs from monostatic measured data are also compared against the true measured data set. The geometric complexity of the object is related to the dominant scattering mechanisms and this characteristic dictates the suitability of the MBET. Both Kell's and Crispin's MBETs predict specular scattering from simple objects (e.g. flat plate) fairly well for bistatic angles of approximately 30-40 degrees. Kell's performs slightly better on average than Crispin's for regions in which diffraction scattering is noticeable (larger bistatic angles) by 1-3 dBsm for this simple target. For complex geometries neither MBET is particularly suitable for bistatic angles beyond 15-20 degrees. Again, targets whose monostatic signature is primarily dominated by direct specular reflections tend to produce bistatic signatures which correlate better with those predicted by either of the MBETs. MBET predictions at bistatic angles greater than 15° tend to be lower than measured data in this case. Once surface waves, multi-bounce reflections, and/or shadowing effects are introduced, the MBET correlation decreases substantially for even smaller bistatic angles (5-10 degrees). Predictions for larger bistatic angles tend to higher than measured in this instance. The discrepancies are primarily due to the changing nature of the scattering centers as a function of bistatic angle. Geometries which support mostly specular reflections exhibit less variation in the nature/existence of the scattering centers than those who support specular and non-specular effects, thus leading to better correlation between MBET RCS and measured RCS for the former. One should also note the practical limitation of Kell's formula. The frequency shift necessitates collection of data with roughly twice the angular resolution as that required by Crispin's to maintain the same average performance. Kell's prediction also incorporates an unavoidable degradation in angular resolution for small bistatic angles, which is not present in Crispin's. These limitations may themselves preclude use of Kell's formula for smaller data sets of courser angular resolution.

Providing an adequate explanation of the results demands familiarity with not only the data itself but the data collection methods and the theoretical basis upon which the methods are founded. The next section addresses these issues. Section II begins with a review of scattering physics, the computational methods for describing the physics, and implementation of the computational method within the Xpatch code. It concludes with a review of the measured data's collection process to provide an intuitive sense of the data's quality. A short description of the measurement facilities and a review of calibration procedures are included.

II. Background

Radar Cross Section:

For the radar engineer , it is convenient to describe the RF field energy scattered by an object by some normalized quantity, which is independent of the illuminating source's field strength and the distance between the object and this source. Typically this quantity describes the scattered energy in an isotropic, homogenous medium far from the field's originating source (*far field*) and is dubbed the *radar cross section* (RCS). Formally, RCS is defined as the equivalent aperture surface area which would radiate an equal amount of power being received at the receiver if it were to radiate isotropically. In general it's dependent on the incident radiation's frequency and polarization and the target's physical size, orientation with respect to the illuminating radar, and constituent materials. For the bistatic situation, the requirement for the target's orientation expands to include relativity to both the transmitting and receiving antennas. The RCS represents an equivalent aperture surface area of the target, which captures a certain amount of incident radiation, and which, if re-radiated isotropically, would produce an equivalent scattered field at the receiver. Mathematically, the RCS can be defined as:

$$\sigma = \lim_{R\to\infty} 4\pi R^2 \left|\frac{E_s}{E_i}\right|^2 \tag{1}$$

or,

$$\sigma = \lim_{R\to\infty} 4\pi R^2 \left|\frac{H_s}{H_i}\right|^2 \tag{2}$$

where: σ = RCS

R = distance from radiation source (typically observer's location) to object (typically the origin of the geometric coordinate system)

E_s = scattered electric field

E_i = incident electric field

H_s = scattered magnetic field

H_i = incident magnetic field

Immediatley one notices the unrealistic expectation in (1) and (2) that the distance between the radiation source and target should approach infinity. However for all practical purposes it only need be greater than the *far field* requirement. This requirment is derived from the necessity of approximating what is in reality a spherical incident field (indicated by the 4πR in the equations) by an assumed planar incident field. Although it can be arbitrarily chosen, in most cases this distance, R, is determined by the incident wavelenght, λ, and largest target dimension orthogonal to the incendent radiation direction, d, by:

$$R = \frac{2d^2}{\lambda} \tag{3}$$

This requirement establishes a one way phase error between scattering centers of π/8 attributable to the spherical nature of the incident field. It results in a maximum error of 1 dB in the RCS amplitude computation due soley to this field taper.

Expanding the scattered electric or magnetic field vector of (1) and (2) into its integral representaion yields (4) and (5) [6:16],

$$E_s = -\frac{1}{4\pi}\int_S \left\{ ik_o Z(n \times H)\frac{e^{i1k_oR}}{R} - (n \times E) \times \nabla\left(\frac{e^{ik_oR}}{R}\right) - (n \cdot E)\nabla\left(\frac{e^{ik_oR}}{R}\right) \right\} dS \tag{4}$$

$$H_s = -\frac{1}{4\pi}\int_S \left\{ ik_o Y(n \times E)\frac{e^{i1k_oR}}{R} - (n \times H) \times \nabla\left(\frac{e^{ik_oR}}{R}\right) - (n \cdot H)\nabla\left(\frac{e^{ik_oR}}{R}\right) \right\} dS \tag{5}$$

where,

$Y = \sqrt{\dfrac{\varepsilon}{\mu}}$

$Z = 1/Y$

ε= relative permativity

μ = relative permability

k_o = free space wavenumber, $2\pi/\lambda_o$

λ_o = free space wavelength

E = electric field at object surface

H = magnetic field at object surface

R = distance from target surface to observer

Notice (4) and (5) are analogous field expressions linked through the duality principle. Insertion of (4) into (1) and (5) into (2) is seen as an expanded form of the general RCS prediction equation and can apply equally to monostatic and bistatic situations. However solving either integral can prove impractical in many instances and therefore, approximations to the scattering solution are more useful. Those approximations which lend themselves to the same generality of geometry (i.e. applicable to monostatic and bistatic) are of particular concern here, but before embarking upon a discussion of computational methods, a review of scattering in general should be ccomplished to facilitate discussion of the analyis in Section IV.

Basic RF Scattering Regimes:

Due to the wave nature of radiation, metallic objects will interact more or less with incident radiation based on the physical dimensions of the object, L, in relation to the incident field wavelength, λ. This relationship, L/λ, is refered to as the object's "electrical size," and it dictates scattering behavior in each of three scattering regimes in which the scattering mechanims will differ: 1) the Rayleigh region, 2) the resonant region, and 3) the optical region. Each regime is defined by the incident field's phase continuity across the object's extent as follows. Although the scattering mechanisms which arrise in each of the regimes are normally explored from a monostatic perspective, most apply equally well to bistatic situations. [18].

Rayleigh region scattering occurs when the object's physical dimensions are much less than a wavelength, and arrises from the fact that the phase front of the impinging field remains relatively constant over the object's surface. Signature analysis in this region can be accomplished through static field methods, and, in general RCS amplitudes are inversely proportional to fourth power of the wavelength. However, because most objects of practial importance are much larger than the incident field wavelength, resonant and optical scattering are of greater concern [18].

Resonant region scattering occurs for $1 <= L/\lambda <= 10$ where the incident phase front across the object's surface begins to vary substantially. This leads to greater interaction between the object and the wavefront producing two classes of scattering mechanisms: optical and surface wave effects. Optical effects are characterized by reflections which occur in a direction equal to the angle of incidence as measured from an outward pointing surface normal at the point of reflection. Surface waves exist when EM energy stays attached to the object's surface and arrise when the incident E-field is contained within the plane of incidence (see Fig 3). The plane of incidence is defined as a plane containing the outward pointing surface normal and the direction vector of the incident radiation. Surface waves are classified in one of three ways: traveling waves, creeping waves, or edge traveling waves [18]. Traveling and creeping waves are essentially the same beast albeit the former exists in illunimated regions while the later only exists in shadowed regions. The illuminating wavefront tends to add contructively to the traveling wave as it traverses the object's surface, amplifying it as it goes, while the creeping wave dies off as it travels, having no source energy to recusitate it. Each of these waves are spawned from currents which travel along a surface radiating small amounts of energy as they go. When some surface discontinuiuty is encountered (edge, gap, small change in the surface radius of curvature, etc.) stronger pertubations cause more energy to be radiated as scattered fields. Surface wave scattering is independent of an object's size and tends to be proportional to λ^2. Small surface geometries don't contribute markedly to the overall RCS, but multiple reflections within an object's geometry are of particular concern because of the significance of the surface wave effect. Although they do exist at higher frequencies, their contribution to the overall RCS signature diminishes as one approaches the optical region [17,18].

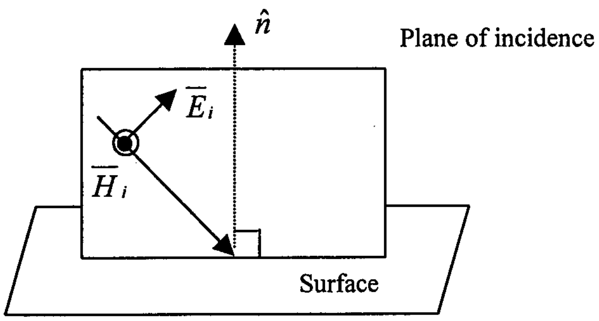


Figure 3: Plane of Incidence

Optical region scattering occurs for geometries greater than 10λ in extent. The totality of the scattered signature can now be thought of as having arrisen from the phasor interactions between a collection of independent scattering elements (scatteirng centers). The term *ray optical* is often used to describe the behavior of incident and scatterd fields as if they were individual rays of energy. While a more thorough discussion of scattering centers and ray optical behavior is covered later in this chapter, scattered energy in this regime can be classified as one of four types [18]:

Specular scattering: This type of scattering is the optical mechanism described before. It is characterized by a large reflection which occurs in a direction equal to the angle of incidence as measured from an outward pointing surface normal at the point of reflection. The lobe width of the reflection *spike* is related to the electrical size of the reflecting surface, but it tyically exists only over a narrow angular extent. The aformentioned geometry describes the general bistatic situation, but the more common monostatic exists if the angle of incidence is zero degrees. The amplitudes of the monostatic and bistatic specular spikes are roughly the same, yet, the lobe widths of specular spikes viewed from a bistatic perspective can be double that from the corresponding monostatic direction. Specular returns will be among the largest contributors to localized signature levels for many objects.

End region scattering: Scattering derived from end regions of fintie objects and is responsible for the sidelobe structure associated with specular reflections [18]. Again , amplitude levels between bistatic and monostatic are roughly the same, but the lobing/nulling structure can differ. The envelpe of the lobing amplitudes decreases away from specular directions for both situations.

Diffraction: End region scattering in the specular direction due to edge-induced currents along physical edges [18]. The orientaion of the E-field (polarization) in relation to the edge determines if the diffraction occurs from a leading or trailing edge. E-fields aligned parallel to an edge cause leading edge diffraction while E-fields aligned perpendicualr to an edge tend to cause trailing edge diffraction. Trailing edge diffraction originates from surface waves which, when encountering a edge, produce the diffraction. Edge diffraction is characterized by a fairly wide angular distribution similar to the scattering from a dipole antenna. As a result diffraction may be more prevalent in certain bistatic situations than in similarly configured monostatic ones because larger scattering source (e.g. specular) are more likely to dominate from a monostatic perspective.

Multi-bounce: Usually described as specular scattering which occurs between several surface elements before being re-radiated back to the receiver. This type of scattering can be the largest contributor to any monostatic signature level for certain geometries such as dihedrals and trihedrals. For cavities, monostatic multi-bounce RCS tends to have a lower overall amplitude, but a greater angular extent than a pure specular from a flat plate of similar size to the cavity opening. However, the specular return from these re-entrant geometries is centered about the illumination direction, and therefore become insignificant contibutors to the RCS at large bistatic angles. This phenomena can also arrise from surface wave reflections interacting between surface features, and although they are of a lower amplitude than the specular brethren, this type of multi-bounce may be more noticeable from a bistatic perspective.

Whenever present, specular and muti-bounce scattering dominate the monostatic RCS. Although highly dependent on object geometry and orientation, in general, whenever the monostatic is dominated by these mechanisms, the bistatic signature tends not to be (and vice-versa). This holds significant implications for low observable objects (e.g. ogive). Current low observable absorption technologies reduce surface wave phenomena to effectively minimize this RCS contribution from either perspective. But shaping techniques are primarily designed to reduce the monostatic signature. These same techniques may exaserbate the bistatic signature by deflecting reflected energy away from illumination soure (at some bistatic angle). This necessarily indicates a dependence of a bistatic signature on the bistatic angle itself while implying a certain relationship between monostatic and bistatic RCS. Because this thesis both of these, its convinent to group bistatic signature levels into regimes based on their relation to the bistatic angle. A brief desciption of three common bistatic angle regimes is described below. The extent of each will of course be interrelated with the others and be highly dependent on the target's electrical size.

Bistatic Regimes:

An object's electrical size plays a critcal role in establishing the extent of each bistatic regime. As previously mentioned, when L/λ is large (optical scattering), specular and multi-bounce spikes dominate the monostatic signature, and because of the small angular width of each spike, should dominate the bistatic return for small bistatic angles. Differences between the monostatic and bistatic RCS can arise from one of three sources: 1) phase differences among scattering centers, 2) radiative changes from scattering centers and/or 3) changes in the nature or existence of scattering centers themselves [14]. In the resonant region, scattering center analysis no longer applies, and there may be less association between the monostatic and bistatic RCS. Because optical and resonant region scattering are of primary concern, the following discussion pertains to objects of sufficient electrical size to fall within these regions.

Pseudomonostatic RCS Region: This regime is characterized by a high degree of correlation between the monostatic and bistatic signature levels for optical region scattering [31]. This correlation allows one to essentially compute the bistatic RCS from monostatic RCS through one of two common monostatic-to-bistatic equivalence theorems (MBET). A complete review of each MBET is accomplished in the *High-Frequency Computational Methods* section of this chapter. This region exists for decreasingly smaller bistatic angles with increasingly complex object geometry. It also exists to a lesser extent as the RCS becomes influenced by resonant region scattering. Quantifying the degree to which the MBETs remain accurate (and thus establishing an implicit measure of the existence of this region) is one goal of this work to be accomplished in Section IV.

Bistatic RCS Region: In this region the bistatic signature is independent of the monostatic for similar illumination angles, target orientations, polarizations, etc. When optical scattering dominates, the nature of scattering centers themselves causes the divergence between monostatic and bistatic as a result of one of the three the aforementioned sources. The first source provides fluctuations in the monostatic RCS as a function of aspect angle, but in this region, similar bistatic RCS fluctuations are caused by changes in the bistatic angle itself. The second source occurs for bistatic receiver positions just outside the main lobe spike from multi-bounce reflections. The third source usually arises when shadowing of surface features occurs as viewed from a bistatic receiver position. In general, the bistatic signature will be lower than the monostatic [31]. However, as mentioned before, exceptions exist for target geometries which produce large specular returns at certain bistatic angles and for low observable targets near the resonance region. Larger bistatic returns may also be prevalent in certain shadowing situations. The analytical focus of this thesis lies in the pseudomonostatic and bistatic regions for bistatic angles less than approximately 90-110 degrees. However, one more region can be identified for larger bistatic angles.

Forward Scatter Region: As the bistatic angle approaches 180 degrees a new region of enhanced scattering is encountered. Crispin and Seigel showed that in the optical region, the bistatic RCS amplitude approaches the physical optics (PO) approximation of a flat plate of similar size to the object's shadow region area projected in the forward direction [8,31]. Babinet's principle provides a reasonable explanation for this phenomena. Willis describes it this way:

"..Babinet's principle as used in optics states that 1) two diffraction screens are complimentary if the clear regions of the first are opaque (shadow) regions of the second and vice versa and 2) when the two complimentary screens are illuminated by a [plane wave] source, the fields produced on the other side of the screens add to give a field that would be produced with no screens..."

The screens must be planar and infintely thin, and the illumination source for the second screen must be the conjugate source (90 degrees out of phase) of the first. In an RF application, it essentially allows one to demonstrate that the radiation pattern of a dipole and a slot are identical [31]. Although further investigation of the forward scatter region is outside the scope of this report, this brief description is included for completeness.

The preceding discussion brings to light the concept of predicting RCS amplitudes through computational approximations (e.g. the PO solution as proposed by Crispin and Seigel). A more thorough treatment of common computational methods is now presented to familiarize the reader with the techniques utilized in the analysis section (IV).

High Frequency Computational Methods:

As an object's electrical size increases, solving for the scattered fields of (4) and (5) becomes more intractable. Solutions derived from eigenfunction and Rayleigh's methods exhibit poor convergence properties [19]. Even with the use of high speed computers, solving these equations with a method-of-moments approximation to the surface currents from which they arrise can be prohibitively time consuming for all but the simplest geometries. With this in mind asymptotic, high frequency methods have been developed. Asymptotic refers to a solution derived from an approximation in which some parameter approaches a limiting value. For scattering analysis, the value taken to a limit is usually the wavenumber, *k*, which translates to a very large (or infinite) frequency assumption. Fields described by an asymptotic expansion of the frequency characteristic are commonly referred to as *ray optic*. Although only approximations, many of these *high frequency* techniques are perfectly adequate for most situtations. Some of the more common methods germaine to this thesis are presented below.

Geometric Optics (GO):

One of the simplest and most commonly utilized scattering approximation methods is the Geometric Optics (GO) solution. It is derived from an application of Fermat's principle to establish the direction of field propogation and conservation of energy to determine the field's intensity. Figure 4 describes the geometry.

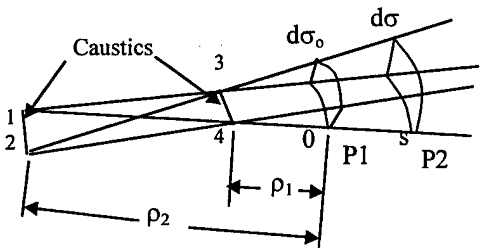


Figure 4: GO Field Propagation

Fermat's priciple states that light rays follows a path from point P1 to P2 which takes the least amount of time to traverse (thus allowing for the possibility that the path itself is not the geometrically shortest one) [12]. Mathmatically, the path length (OPL) must satisfy this equation:

$$OPL = \int_{P1}^{P2} n\,ds \tag{6}$$

where n = k/k$_o$, the index of refraction. Equation (6) can also be expressed in terms of the eikonal equation [7]:

$$|\nabla \psi|^2 = n^2 \tag{7}$$

which essentially determines the field's phase behavior (and thus direction).

For convienience, light rays can be bundled into astigmatic ray tubes as shown in Fig. 4 and treated as a single entity for the field amplitude computations. The energy flux of the field must remain the same within each ray tube throughout the course of propagation. Taking the field intensity at point P1 to be A_o^2 and P2 to be A^2 and the cross sectional area of the ray tube at P1 to be $d\sigma_0$ and that at P2 to be $d\sigma$, one finds in an isotropic, homogenous media that [19]:

$$A_o^2 d\sigma_o = A^2 d\sigma \tag{8}$$

The ratio of the cross sectional areas to the fields' radius of curvature is (R_{o1}, R_{o2} are principal radii of curvature at point P1; R_1, R_2 are principal radii of curvature at point P2:

$$A = A_o \sqrt{\frac{R_1 R_2}{R_{o1} R_{o2}}} \tag{9}$$

and, therefore:

$$\frac{d\sigma_o}{d\sigma} = \frac{R_1 R_2}{R_{o1} R_{o2}} \tag{10}$$

One can also show that the Luneburg-Kline expansion to Maxwell's equations [19]:

$$\nabla^2 \overline{E} + k^2 \overline{E} = 0 \tag{11a}$$

$$\nabla \cdot \overline{E} = 0 \tag{11b}$$

where $k^2 = \omega^2 \mu\varepsilon$

is:

$$\overline{E}(\overline{R},\omega) = e^{-jk_o\psi(\overline{R})} \sum_{m=0}^{\infty} \frac{\overline{E}_m(\overline{R})}{(j\omega)^m} \tag{12}$$

where:

ω = frequency

$k_o = \omega/c$, c = phase velocity in free space

ψ = Equipotential phase surface of the field wavefront

Equation (12) can also be written in the asymptotic limit of ω as:

$$\overline{E}(r) \cong \overline{E_o} e^{-jk_0\psi_o} \sqrt{\frac{R_1 R_2}{R_{o1} R_{o2}}} e^{-jks} \tag{13}$$

Equation (13) is the classical GO approximation. Taking note of the similarity between (13) and (9) one quickly sees the conservation of energy relation within the GO approximation, while inspection of (13) itself reveals that GO also describes the correct path upon which the field propogates. One can further refine (13) to describe the curvature of the field wavefront at P2 in terms of the reference at P1:

$$\overline{E}(r) \cong \overline{E_o} e^{-jk_0\psi_o} \sqrt{\frac{\rho_1 \rho_2}{(\rho_1 + s)(\rho_2 + s)}} e^{-jks} \tag{14}$$

For a smoothly curved, perfect electrically conducting (PEC) surface the $\overline{E_o} e^{-jk_0\psi_o}$ term on the right hand side of the equation (14) will relate the directions of the incident and reflected E field. The reflected field at the surface, $\overline{E_o{}^r} e^{-jk_0\psi}$, is equal to the inner product of a *dyadic reflection coefficient* and the incident field, $[\hat{n}\hat{n} - \hat{b}\hat{b}] \cdot \overline{E_o{}^i} e^{-jk_0\psi}$ (b is a unit vector tangent to the surface and defined as $\hat{b}(\hat{b} \cdot \overline{E}^i) = -\hat{n} \times (\hat{n} \times \overline{E}^i)$). The dyadic simplifies to a (–1) when the incident field is tangent to the surface, thus making the GO approximation very simple to compute [19]. Note, however, that the GO field amplitudes become infinite at s = -ρ₁ or -ρ₂ where the rays converge in either dimension at positions known as caustics. Caustics occur near shadow boundaries and for backscatter from flat surfaces. This translates into an inability of the GO method to predict reflections from flat surfaces or account for diffraction into shadow regions, and is seen as primary limitation to the technique. Because of this, alternative high frequency approximations seek to improve upon its accuracy.

Physical Optics (PO):

The GO fields of the previous section may be related to currents from which they arrise. These GO currents can in turn be used in an exact integral representation (similar to (4) or (5)) of the scattered field in an approximation known as the Physical Optics (PO) approach. Such a derivation must begin by relating the field to its respective source currents through (see Fig. 5 for geometry) [19]:

$$\overline{E_s} = -z_o \hat{R} \times \overline{H}_s \tag{15a}$$

$$\overline{H}_s = -\frac{jk}{4\pi}\left(\frac{e^{-jkR}}{R}\right)\overline{U} \tag{15b}$$

$$\overline{U} = \hat{R} \times \int_s \overline{J}(\overline{\rho}) e^{jk\overline{p}\cdot\hat{R}} dS \tag{15c}$$

$$\overline{J}(\overline{\rho}) = \hat{n} \times \overline{H}(\overline{\rho}) \tag{15d}$$

where:

$\overline{J}$ = surface current

$\overline{U}$ = vector far field amplitutde

$\overline{H}$ = total magnetic field

R = distance from field point to origin

$\hat{R}$ = unit vector in direcion of scattering

$\overline{\rho}$ = position vector from origin to point on surface S

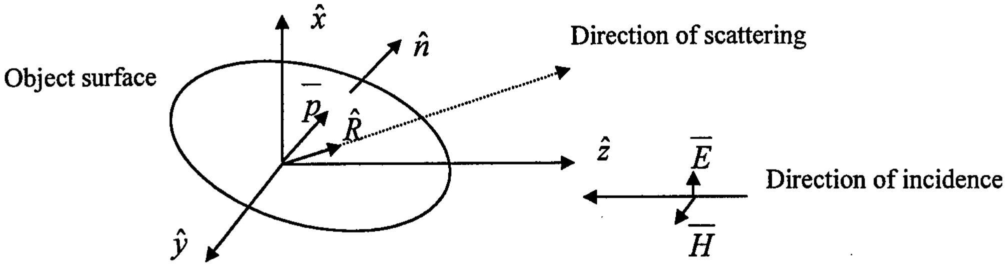


Figure 5: Geometry of PO Scattering Problem

If one assumes the surface to be an infinite, flat PEC the current in (15d) can be re-written as the GO current (Kirchoff's approximation) [7,19]:

$$\bar{J}(\bar{\rho}) = \begin{cases} 2\hat{n}\times\bar{H}^{i} & \text{in the illuminated region} \\ 0 & \text{in the shadow region} \end{cases} \qquad (16)$$

$\bar{H}^{i}$ = incident magnetic field

In this approximation, the limitation of GO in predicting backscatter from flat surfaces is overcome. However, the GO surface current near an edge or shadow boundary is still inaccurate and in general the PO approximation fails the reciprocity test. Consequently PO cannot adequately account for diffraction into shadow regions nor should it predict bistatic scattering away from specular reflecions very well. PO cannot capture surface and traveling wave phenomena, a primary limitation when dealing with any target whose bistatic scattered field can be dominated by such surface effects (i.e. an ogive) [16]. Despite its accuracy for speculars and ease of implementation, PO does not provide any physical insight into the nature of the scatterd fields themselves, leading to difficulty in assessing when it does and doesn't work correctly [7,19].

Physical Theory of Diffraction (PTD):

Ufimstev [18] attempted to improve upon the PO solution by accounting for diffraction effects through specialized coefficients in the scattering computation. Although no more intuitive than PO, physical theory of diffraction (PTD), as it has come to be called, does mitigate PO's inability to account for edge diffraction. Ufimstev subtracted the PO solution from an exact solution to the two-dimensional wedge problem and theorized that the remainder is the edge diffraction component. Capturing this diffraction effect within a specialized set of diffraction coefficients, he generalized the PTD solution for any geometry. The coefficents describe the diffraction behavior in one of three regions as a function of the illumination angle, ϕ_i, observation angle, ϕ_s, and wedge angle, $2\pi - \alpha$ (Fig 6). Essentially PTD formulation can be summarized as follows. The scattered E and H-fields of (15a) and (15b) can be written to include diffraction coefficients *f* and *g* [18]:

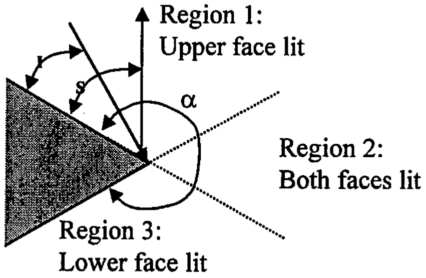


Figure 6: Regions of Illumination for Wedge Diffraction

$$\overline{E}^s = \overline{E}_o f \frac{e^{i(kr+\pi/4)}}{\sqrt{2\pi kr}} \tag{17}$$

$$\overline{H}^s = \overline{H}_o g \frac{e^{i(kr+\pi/4)}}{\sqrt{2\pi kr}} \tag{18}$$

$$f = \begin{cases} (X-Y)-(X_1-Y_1) & 0 \le \phi_i \le \alpha - \pi \\ (X-Y)-(X_1-Y_1)-(X_2-Y_2) & \alpha - \pi \le \phi_i \le \pi \\ (X-Y)-(X_2-Y_2) & \pi \le \phi_i \le \alpha \end{cases} \tag{19}$$

$$g = \begin{cases} (X+Y)-(X_1+Y_1) & 0 \le \phi_i \le \alpha - \pi \\ (X+Y)-(X_1+Y_1)-(X_2+Y_2) & \alpha - \pi \le \phi_i \le \pi \\ (X+Y)-(X_2+Y_2) & \pi \le \phi_i \le \alpha \end{cases} \tag{20}$$

$$X = \frac{(1/n)\sin(\pi/n)}{\cos(\pi/n)-\cos[(\phi_s-\phi_i)/n]} \tag{21}$$

$$Y = \frac{(1/n)\sin(\pi/n)}{\cos(\pi/n)-\cos[(\phi_s+\phi_i)/n]} \tag{22}$$

$$X_1 = -\frac{1}{2}\tan[(\phi_s-\phi_i)/2] \tag{23}$$

$$Y_1 = -\frac{1}{2}\tan[(\phi_s+\phi_i)/2] \tag{24}$$

$$X_2 = \frac{1}{2}\tan[(\phi_s-\phi_i)/2] \tag{25}$$

$$Y_2 = -\frac{1}{2}\tan[\alpha-(\phi_s+\phi_i)/2] \tag{26}$$

The diffraction coefficients allow the scattered field computation to remain finite, even in the transition regions near the shadow and reflection boundaries. However, because Ufimstev's derivation decends from a two dimensional geometry, it is only applicable for diffraction in the Keller cone direction [18].

Mitzner [18] extended PTD to arbitrary directions with an incremental edge diffraction coefficient method (ILDC). His implementation assumes that diffraction from an edge can be computed by integrating over the illuminated portion of that edge only and it utilizes a dyadic diffraction coefficient within the formal expression. The diffracted component of the scattered E-field in (17) now becomes:

$$\overline{E}_d = \overline{E}_o \frac{e^{i(kr-\pi/4)}}{\sqrt{2\pi r}} \overline{\overline{d}} \cdot \hat{p} dt \tag{27}$$

$\overline{\overline{d}}$ = diffraction dyadic

$\hat{p}$ =unit vector in direction of incident E-field

dt = incremental edge element

while the dyadic is expressed through a set of unit vectors:

$$\overline{\overline{d}} = \hat{d}_\perp \hat{d}_\perp \hat{e}^s_\perp \hat{e}^i_\perp + \hat{d}_{\perp\|} \hat{e}^s_\perp \hat{e}^i_\| + \hat{d}_{\|\perp} \hat{e}^s_\| \hat{e}^i_\perp + \hat{d}_{\|\|} \hat{e}^s_\| \hat{e}^i_\| \tag{28}$$

$$\hat{e}^i_\perp = \hat{t} \times \hat{i} / |\hat{t} \times \hat{i}| \tag{29}$$

$$\hat{e}^s_\perp = \hat{t} \times \hat{s} / |\hat{t} \times \hat{s}| \tag{30}$$

$$\hat{e}^i_\| = \hat{i} \times \hat{e}^i_\perp \tag{31}$$

$$\hat{e}^s_\| = \hat{s} \times \hat{e}^s_\perp \tag{32}$$

$\hat{t}$ = unit vector along the edge

$\hat{e}^i_\perp$ = incident field direction unit vector perpendicular to plane of incidence

$\hat{e}^i_\|$ = incident field direction unit vector parallel to plane of incidence

$\hat{e}^s_\perp$ = scattered field direction unit vector perpendicular to plane of incidence

$\hat{e}^s_\|$ = scattered field direction unit vector parallel to plane of incidence

Equations (27) –(32) can be re-written in PO terms and essentially the unit vectors will toggle on or off the appropriate PO coefficients depending on which wedge face is illuminated. Integrating along the edge length will produce the corresponding diffracted field component. Mitzer's ILDC diffraction coefficients simplify to Ufimstev's along the direction of the Keller cones and remain finite in and near the shadow and reflection boundaries [18]. Notice the generality of the formulation for monostaic and bistatic situations. When dealing with complex geometries, PO and diffraction effects would be computed from each illuminated surface and edge separately and then recombined to determine the total scatterd field. Keeping track of the various illumination directions and reflection points can be teadious, but another high frequncy technique does precisely by treating incident and reflected fields as discrete, *ray optic* bundles of energy.

Shooting and Bouncing Rays (SBR):

Ray optic fields may reflect and/or reverberate across several surfaces of a complex object before being re-radiated back to a receive antenna. The method for tracking the movement of such fields and eventually computing the total scatterd energy arising from the multi-bounce interactions is commonly refered to as the shooting and bouncing ray (SBR) technique [5]. A dense grid of rays is launched toward a target (from the transmitter's location) and the total scattered field arrising from each ray's bounce location is computed/summed with a PO approximation as a function of the target's orientation to the transmitter and receiver. The location and direction of each ray and bounce location is,again, dictated by the GO prediction. Snell's law more easily demonstrates the reflection direction as a function of the incident angle, θ_i, and material index of refraction, n_i, as related to the reflection/tranmission direction, θ_r, and material index of refraction, n_r:

$$n_i \sin(\theta_i) = n_i \sin(\theta_r) \tag{33}$$

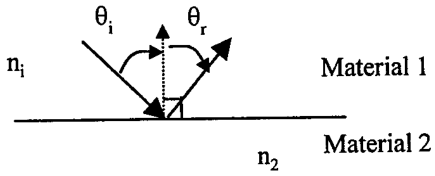


Figure 7: Geometry of Snell's Law

The incident and reflected angles are measured relative to the outward pointing surface normal of the material boundary at the point of reflection (see Fig. 7). Notice that when dealing with a reflection (e.g. from a PEC surface), the reflected ray angle is identical to the incident ray angle. Ray tracing can be computationally expensive, however, especially for a large grid of rays, but it does provide a good indication of where the rays interact with the target's surface(s) as they traverse through the target's geometry. It does not provide any more indication than the any other PO prediction of how they interact with each other or when higher order diffraction, surface, or traveling waves arrise [5].

Scattering Center Approach:

A much more intuitive approach to scattered field prediction is through analysis of radiation from scattering centers. Scattered field energy will contructively and destructively interfere as it is reflected from a target's surface dependent on the target geometry, transmitter/receiver orientaion, freqency,and polarization. The nature of the contructive/destructive field *zones* can be described through a statistical relationship of the relative amplitude and phase between any two of a collection of simple scattering components of which the object appears to be comprised [14]. From the receiver's perspective, the simple scattering component *zones* in which significant constructive interference occurs, appear to be the source locations from where the fields arrise, and are thus called *scattering centers*. Fields may or may not actually arrise from these points, and in fact may be spatially sepearted from the target surface altogether. Yet they povide a basis from which target identification charcteristics can be extrapolated quickly and efficiently. And, unlike the PO method, they capture the totality of the scattering contribution from all the scattering sources, including specular, diffraction, traveling waves, etc. This method is highly dependent on the target geometry in relation to transmitter and receiver, and in general, scattering centers exist only over a small angular extent. This implies that,although the approach may be valid for both monostatic and bistatic scattering, any monostatic scattering center model may not accurately represent a bistatic signature. However, several monostatic-to-bistatic equivalence theroems (MBET) have been proposed to establish just such a relationship, allowing one to convert monostatic information to an approximate bistatic data set.

Kell's Monostatic-to-Bistatic Equivalence Theorem:

Robert Kell [14] proposed a general scattering center-based equivalence theorem in 1965. It relates the bistatic RCS to the monostatic measured along the bisector at a frequency proportional to the cosecant of half the relevant bistatic angle. Equation 34 shows the relationship for the bistatic signature in a single plane (vertical), but this can be expanded to encompass the horizontal plane also.

$$\sigma_B(\theta = \beta, f) = \sigma_M\left(\theta = \frac{\beta}{2}, f \sec\left(\frac{\beta}{2}\right)\right) \tag{34}$$

σ_B = Bistatic RCS

σ_M = Monostatic RCS

θ = receiver orientation angle (azimuth)

f = frequency

As mentioned before, for this method, the RCS of complex objects is described through a statistical relationship of the relative phase between any two of a collection of simple scattering components or scattering centers. Kell derives a relationship between scattering center phase and the bisector of the bistatic angle through the following analysis [14].

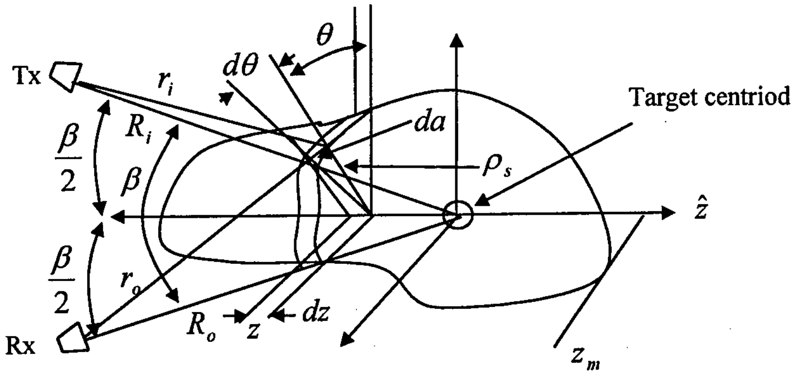


Figure 8: Kell's Bistatic Scattering Geometry

Referring to Fig. 8, one may re-write Eqn. (2) as:

$$\sigma = \frac{\pi}{\lambda^2}\left|\int I(z) e^{i2k_0 z\cos\beta/2} dz\right|^2 \tag{35}$$

$$I(z) = \int\left[(n \times h_s) \times r_0 e^{i\phi_H} + (n \cdot h_s) r_0 e^{i\phi_H} - (n \times e_s) e^{i\phi_E}\right]\rho(\theta, z) d\theta \tag{36}$$

where:

σ = Bistatic RCS

λ = wavelength

β = bistatic angle

h_s = normalized magnetic field vector on target surface at element da

ϕ_E = phase difference between surface and incident E-fields at da

ϕ_H = phase difference between surface and incident H-fields at da

r_o = distance from receiver to area da

$\rho(\theta,z)$ = conversion factor which relates da to dzdθ

Although I(z) provides an exact solution to (35), it can be difficult to compute (as has been mentioned before), so an approximation to the scattering solution is, again, more useful. Kell's approach yields an approximation contingent primarily upon phase interactions between scattering centers. The discrete scatter center interpretation of RCS (relative phase method) finds form in equation (37), [3]:

$$\sigma = \left| \sum_{m=1}^{M} \sqrt{\sigma_m} e^{i\phi_m} \right|^2 \tag{37}$$

σ = total RCS

M = number of discrete scattering centers

σ_M = RCS of mth scattering center

ϕ_M = phase of the mth scatterer's field relative to the first scattering center

The phase term is dependent on the distance, d_m, between any individual scattering center and a phase reference center, and thus requires a high degree of accuracy in determining λ/d_m. Should such information be available, one can relate the phase in a bistatic situation to the bistatic angle, β:

$$\phi_m = 2k_o z_m \cos\frac{\beta}{2} + \xi_m \tag{38}$$

$z_m(\alpha)$ = distance between mth and a reference phase center projected on the bistatic bisector axis

ξ_m = residual phase contribution of mth scattering center (i.e. path length phase contribution of creeping wave component)

Placing equation (38) into equation (37) leads to,

$$\sigma = \left| \sum_{m=1}^{M} \sqrt{\sigma_m} e^{i2k_o z_m \cos\frac{\beta}{2} + \xi_m} \right|^2 \tag{39}$$

which now relates the monostatic to the bistatic RCS. This formula essentially states that for a given target orientation α, if the RCS can be described by a scattering center summation and if the amplitudes, positions, and residual phases of the significant contributors to this summation are insensitive to the bistatic angle over the range of considered bistatic angles, then the bistatic RCS is equal to the monostatic measured along the bisector of the bistatic angle, β, at a frequency lower by cos β/2. In other words, to extrapolate the bistatic RCS data from monostatic data one needs to 1) measure the monostatic signature at an orientation angle, α=β/2, at a frequency sec β/2 higher than the desired equivalent bistatic frequency (no polarization shift) and 2) translate the data along the α-axis such that each data point at position α in the monostatic data set corresponds to a position 2α in the bistatic data set [14].

The nature of scattering centers gives some indication of the applicability of this approximation. As previously mentioned, differences between the monostatic and bistatic arise from one of three sources [14]. If changes in any of these areas should be small as a function of changing bistatic angle, the monostatic and bistatic returns should be similar. In essence, the MBET should prove more accurate for regions in which an equipotential phase distribution is common, namely where direct specular-type reflections dominate. In general, these will occur for smaller bistatic angles. At larger bistatic angles the frequency-dependent nature of individual scattering centers is of greater significance than the equivalent frequency shift provided by the cos β/2 term, and thus, Kell's MBET should begin to fail. Target features which produce shadowing effects, non-specular type effects (e.g. surface waves or diffraction), or multi-bounce situations (e.g. dihedral and trihedral-type reflectors) may provide dominant field scattering in the bistatic region. Here, too, the MBET is expected to break down. This may be especially prevalent in the forward scatter region, although such an analysis is outside the scope of this report.

Kell states that upon comparison of general theory of diffraction (GTD) calculated bistatic RCS data for a 24λ long, 4.8λ diameter cylinder, the RCS predicted by the MBET is within 3 dB of the measured value through 10 degrees bistatic angle (as measured from illumination at broadside and end-on). He also mentions that the MBET fails to accurately predict a creeping wave component to a scattered field from several different size spheres for bistatic angles greater than 1 degree [14]. MRC has shown that the MBET predicts significantly greater signature levels than is measured for dihedral and trihedral-type reflectors [20].

Crispin's Monostatic-to-Bistatic Equivalence Theorem:

Crispin and Siegel [8] also proposed an MBET, although it is solely based on PO considerations. The final derivation is similar to Kell's (34) except the frequency shift term is eliminated.

$$\sigma_B(\theta = \beta, f) = \sigma_M(\theta = \frac{\beta}{2}, f) \tag{40}$$

Crispin states the theorem in following fashion:

" In the limit of the vanishing wavelength, the bistatic cross section for transmitter direction $\hat{k}$ and receiver direction $\hat{n}_o$ is equal to the monostatic cross-section for the transmitter-receiver direction $\hat{k}+\hat{n}_o$ with $\hat{k} \neq \hat{n}_o$ for bodies that are sufficiently smooth. "

Albeit relatively ambiguous, *sufficiently smooth* usually refers to a surface whose surface features are at least two orders of magnitude smaller than the incident wavelength. Figure 9 depicts the geometry.

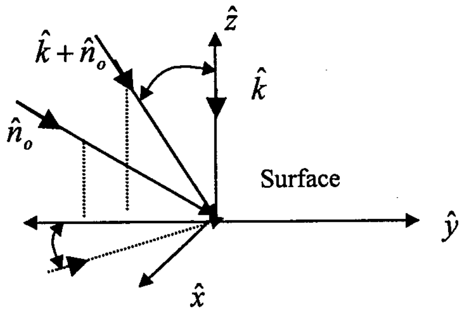


Figure 9: Crispin's Bistatic Scattering Geometry

In describing this derivation, the concept of a radiation vector, F, will be introduced into the PO formulation of (15a-15d). It is essentially the U vector of 15c combined with the coefficient of 15b and can be written:

$$\overline{\mathrm{F}} = \frac{jk}{2\pi}\left[(\hat{n}_o \cdot \hat{a})\mathrm{f} - (n_o \cdot \mathrm{f})\hat{a}\right] \qquad (41)$$

$$\overline{\mathrm{f}} = \int_S n e^{jkr' \cdot (\hat{n}_o + \hat{k})} ds \qquad (42)$$

where: $\hat{a}$ = unit vector in direction of incident magnetic field

S = illuminated portion of surface

$\hat{n}$ = outward pointing surface normal unit vector

r' = radius vector from some origin to point on surface

Although the transmitter and receiver directions can be arbitrary (e.g. $\hat{n}_o$ may be a function of ϕ and θ in Fig 9), for the remainder of this proof, the transmitter shall be restricted to align with the z-axis and the receiver shall be contained in the y-z plane (Fig. 9). As such the following monostatic (43-45) and bistatic (46-48) direction vectors can be written:

$$\hat{n}_o = \sin(\theta)\hat{y} - \cos(\theta)\hat{z} \tag{43}$$

$$\hat{k} = \sin(\theta)\hat{y} - \cos(\theta)\hat{z} \tag{44}$$

$$\hat{a} = \cos\phi\,\hat{x} + \sin\phi\cos\theta\,\hat{y} + \sin\phi\sin\theta\,\hat{z} \tag{45}$$

$$\hat{n}_o = \sin(2\theta)\hat{y} - \cos(2\theta)\hat{z} \tag{46}$$

$$\hat{k} = -\hat{z} \tag{47}$$

$$\hat{a} = \cos\phi\hat{x} + \sin\phi\hat{y} \tag{48}$$

As the wavelength becomes very small, f can be written:

$$\bar{\mathrm{f}} = \frac{-(\hat{n}_o + \hat{k})}{\left|\hat{n}_o + \hat{k}\right|} \int_S e^{jkr'\cdot(\hat{n}_o + \hat{k})} dS \tag{49}$$

which for the monostatic becomes:

$$\bar{\mathrm{f}} = \hat{p} \int_S e^{jkr'\cdot(2\hat{p})} dS \tag{50}$$

and for the bistatic:

$$\bar{\mathrm{f}} = -\hat{p} \int_S e^{jkr'\cdot(2\hat{p}\cos\theta)} dS \tag{51}$$

where:

$$\hat{p} = \sin\theta\,\hat{y} - \cos\theta\,\hat{z} \tag{52}$$

Evaluating the integrals via stationary phase, (50) and (51) become:

$$\bar{\mathrm{f}} = \left[A\sin\theta\,\hat{y} - A\cos\theta\,\hat{z}\right]e^{jkC} \qquad \text{(monostatic)} \qquad (53)$$

$$\bar{\mathrm{f}} = \left[A\tan\theta\,\hat{y} - A\,\hat{z}\right]e^{jkC\cos\theta} \qquad \text{(bistatic)} \qquad (54)$$

where A & C are constants. Plugging (53) and (54) back into (41) yields:

$$\overline{\mathrm{F_M}} = \frac{jk}{2\pi}\left[\left(Ae^{jkC}\right)\left(\cos\phi\,\hat{x} + \sin\phi\cos\theta\,\hat{y} + \sin\phi\sin\theta\,\hat{z}\right)\right] \qquad \text{(monostatic)} \quad (55)$$

$$\bar{\mathrm{F}}_B = \frac{jk}{2\pi}\left[\left(Ae^{jkC\cos\theta}\right)\left[\sin\phi\sin 2\theta\left(\tan\theta\,\hat{x} - \hat{z}\right) - \left(\tan\theta\sin 2\theta + \cos 2\theta\right)\left(\cos\phi\,\hat{x} + \sin\phi\,\hat{y}\right)\right]\right]$$

(bistatic) (56)

The magnitudes of (55) and (56) can be evaluated as:

$$\left|F_M\right|^2 = \left|\frac{kA}{2\pi}\right|^2 \qquad \text{(monostatic)} \quad (57)$$

$$\left|F_B\right|^2 = \left|\frac{kA}{2\pi}\right|^2\left[(\tan\theta\sin 2\theta + \cos 2\theta)^2\cos^2\phi + \sin^2\phi\cos^2 2\theta + \sin^2\phi\sin^2 2\theta\right] = \left|\frac{kA}{2\pi}\right|^2$$

(bistatic) (58)

Combining these results with (2) and (15b) reveals equivalent monostatic (59) and bistatic (60) RCS:

$$\sigma_M = \frac{(kA)^2}{\pi} \qquad (59)$$

$$\sigma_B = \frac{(kA)^2}{\pi} \qquad (60)$$

While the GO, PO, and PTD solutions provide a more rigorous explanation of the scattering solution than either Kell's or Crispin's MBET, all of these techniques have been incorporated into commercially available electromagnetic scattering prediction codes. Although the limitations of each method may be conceptually understood, the implementation of the method within a code is of greater importance when assessing the quality of a computed signature level. Ultimately, assessing the accuracy of any of these high frequency methods turns to a discussion of the accuracy of the software package which utilizes the method. Xpatch is the software suite under investigation here, and a brief introduction to its capabilities should prove beneficial.

Xpatch v2.4d:

Demaco, Inc (now SAIC) developed a suite of high frequency electromagnetic scattering prediction codes and supporting utilities for the US Air Force in the late 1980's and early 1990's which have been incorporated into the Xpatch 2.4d package [5,32]. This is the latest version for mass distribution to be released by the Air Force managing agency, AFRL/SNAS. It is comprised of six primary software modules: 1) a CAD/facet file manipulation/editing utility, CIFER, 2) a frequency-domain PO/PTD based signature prediction tool, Xpatchf, 3) its time-domain analog, Xpatcht, 4) a method of moments based signature prediction package, the Fast Illinois Solver Code (FISC) [32], 5) a CAD/facet file visualization tool, Xedge, and 6) an assortment of RCS data visualization/plotting tools (McRange, McImage, XYPlot, PSPlot).

Most simulation data within this thesis have been generated by the Xpatchf utility, so this module's bistatic prediction capability is evaluated. This PO-based code computes in-phase (I) and quadrature-phase (Q) field component data (monostatic or bistatic) for target geometries represented by certain IGES format object types, ACAD-format facet files, or BRL-CAD format CAD images. From this co, cross and circular polarization

RCS amplitude and phase data is generated to facilitate production of pattern cut, impulse response, and/or 2/3-D downrange images.

Essentially RCS data is produced by launching rays at the object under test, computing the PO response of the first reflection from the illuminated geometry, performing a ray trace (SBR) for each ray within the target geometry, and computing/adding to the first term a PO response of the last reflection point before a ray exits the target geometry. An option is available to compute and sum PO returns for all reflection points and to perform ("GTD" option) or not perform ("PTD" option) aperture blockage checks in the process. A PTD edge diffraction term (computed via Mitzner's method) can be incorporated if desired, but a separate edge geometry description file must first be built with the CIFER utility. For monostatic computations, the first reflection point contribution can be determined by either z-buffering or PO analysis, however, for bistatic predictions all computations are handled through the SBR method. The ray trace history can be captured and visualized along with the target geometry within Xedge to assist in scattering analysis.

Potential Sources of Error Within Xpatch:

Whenever dealing with approximations, error will be introduced into the computations. Moore, Burt, and Hunsberger [22] classified three primary error sources associated with any ray tracing analysis: facetization effects, current element shadowing, and estimating fields which arise from multiple reflections. Each is described below:

Facetization effects: Facetization effects arise from multi-bounce rays which can induce discontinuous current distributions upon a complex object's surface. The nature of the ray bounce and the ensuing current distribution is directly related to a facetized representation of the object's curved geometry. The currents are incorrectly integrated at the receiver in the PO formulation causing anomalous down-range signals. These signals cannot be mitigated by increasing the number of facets defining the geometry or the ray bundle density. Only curved surface representation with a high ray density will diminish this effect [22].

Current Shadowing: Errors can also arise when current patches are not directly visible to the receiver. Such current patches can arise from multi-bounce rays within a shadow region and can be integrated in a PO solution even though the rays emanating from them don't reflect back to the receiver. This leads to non-physical responses in a down-range image. Only current patches visible to a receiver should be integrated in the scattered field computation. This is of particular concern for bistatic analysis [22].

Integration of current patches: Xpatch by default integrates only the current patches induced at the first and last reflection points. However, all current patches induced by reflected rays will contribute a small amount to the scattered field. The amount a current patch will contribute the received scattered field is related to the angle between the average radar line of sight and the normal vector of the surface. The average radar line of sight is the average of the unit vector from the transmitter to the current patch and the unit vector from the receiver to the current patch [22].

With this in mind, gauging the validity of the Xpatch predictions (or any computationally derived data set) necessitates its comparison to a complimentary data set recognized to be of inherently greater precision, accuracy, or *real world* appeal; one needs to establish a basis so to speak. Measured scattering data from indoor or outdoor measurement ranges usually fulfils the requirement, so monostatic and bistatic measured data was collected at two different locations to support this research. Both are indoor facilities, and a few words about collection methods and the facilities themselves should equip the reader with a better sense of the measured data's quality.

Measurement Methods:

Facilities:

The measurements in this report are furnished by Mission Research Corporation (MRC), Dayton, OH, and the European Commission Joint Research Center (JRC), Ispra, Italy. Both facilities utilize indoor anechoic chambers, MRC's being a standard rectangular compact range design, while the JRC's utilizes a hemispherical chamber tailored for bistatic measurements. The typical compact range profile is shown in Figure 10 and is usually characterized by a large reflecting screen in front of a TX/RX antenna assembly. The reflector transforms the antenna's spherical wavefront into a plane wave,

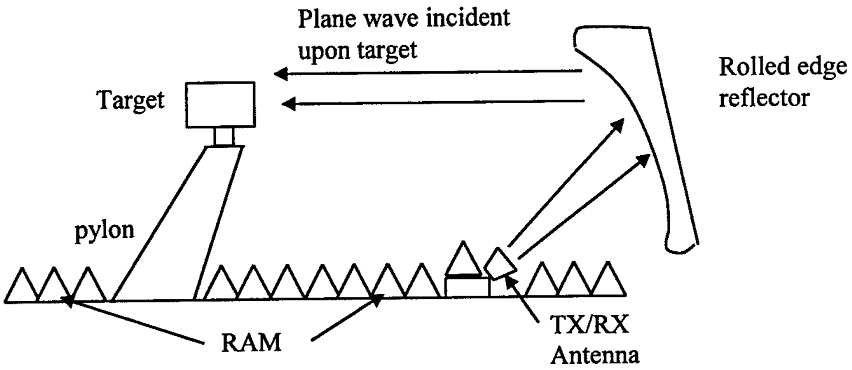


Figure 10: Profile of Compact Range Design

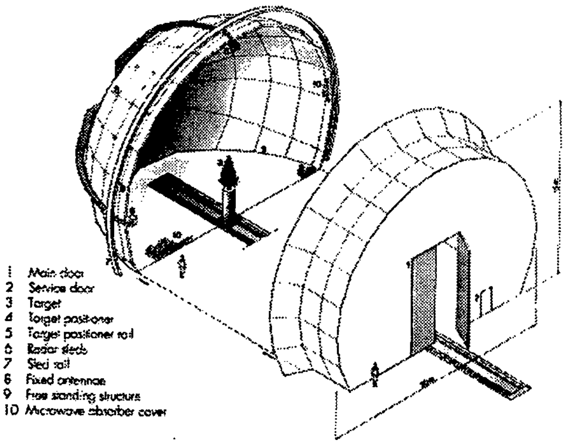


Figure 11: Exploded View of JRC Chamber

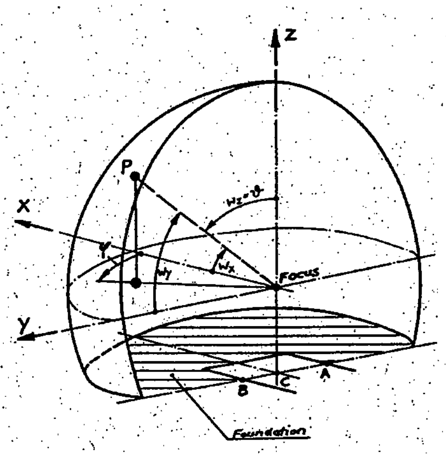


Figure 12: Reference Coordinates Within JRC Chamber

Table 1: MRC & JRC Range Characteristics

Range Characteristic	MRC	JRC
RF range (GHz)	2-18	1-40
Polarization	Full polarimetric	Full polarimetric
Sensitivity (dBsm)	?	-60
Dynamic range (dB)	?	100
Chamber dimensions – length (m)	20	20 (diameter)
Down/Cross range resolution		10 m – 1 cm
Target rotation/angular resolution - horizontal plane (deg)	0-360/?	0-360/0.05
Target rotation/ angular resolution - vertical plane (deg)	+/- 30/?	NA
Target positioning; resolution	Fixed position	+/- 2.5 meters (along x-axis); +/- 0.5 mm
Radar positioning/ angular resolution (deg)	Fixed position	-115/+115; +/- 0.005 (in x-z plane)

which becomes incident upon a test target several meters behind the antennas on a rotating pylon. MRC's pylon includes a target mount, which can rotate 360 degrees in the horizontal plane (primary measurement plane) and +/- 30 deg in the vertical. To minimize unwanted, spurious scattering, the pylon incorporates an ogival profile, the reflector a rolled edge design, and carbon-impregnated pyramid and wedge-shaped RF absorbing foam (RAM) covers the walls, ceiling, and floor. Although the dimensions of the "quite zone" (where the far field requirement is met) are based on frequency and bandwidth, they average several meters in any dimension, so fairly large targets can be accommodated [15]. An exploded view of JRC's range design is shown in Figure 11 and has been optimized for bistatic measurements. Within the primary bistatic measurement (vertical x-z plane, Fig 12) separate rail-mounted TX and RX antennas rotate about a target pylon. Smaller Rx-only antennas are arrayed along the semi-hemispherical chamber walls to take multi-static measurements simultaneously. The chamber supports pseudo-monostatic measurements in the horizontal plane (x-y plane) when the TX/RX antenna pair is locked at θ ~= 90 degs, but the angular separation between them is no smaller than 1.6 degrees [9]. A comparison of chamber attributes is noted in Table 1.

Calibration:

Calibration standards and methods vary widely primarily as a function of cost effectiveness, ease of implementation, and appropriateness for the measurement scenario. Jost and Fahlsing [13] succinctly describe the calibration process as an action taken to ensure that the measured data is traceable to some standard. The appropriateness factor is especially important in quantifying the degree of this tractability. The method of the calibration as well as the calibration target utilized play important roles in determining the quality of the final calibrated data set.

Different methods are more appropriate for monostatic and bistatic situations. Amplitude and phase stability should both be considered when selecting a calibration method. For monostatic measurements phase is of lesser concern because of the single look angle, and thus, methods which produce good amplitude measurements suffice (e.g. background subtraction). In bistatics phase becomes critical because of the generality of the signal collection orientation. Several general polarimetric calibration procedures have been proposed with equal applicability for monostatics and bistatics [28,29,30], and others address uniquely bistatic concerns [1,2,28].

Selection of a calibration target can make or break the accuracy of the final calibrated measurement. Metallic spheres prove to be excellent monostatic calibration sources because of their inherent symmetry, ease of manufacture, and readily available theoretical solution, but lately squat cylinders have become the preferred calibration source due to their larger backscatter return (a good calibration source should have an RCS similar in magnitude to that of the target under test). Bistatic calibration targets need stable signature returns over a wide bistatic angular extent. Spheres do not make good bistatic cal sources because of large fluctuations in the bistatic signature as a function of the sphere's electrical size. Dihedrals have nearly identical cross-pol components, thus providing a good full-polarimetric source in monostatic and bistatic situations. Circular metallic disks are also good candidates for bistatic calibrations because of signature stability over wide bistatic angles [13,15,26].

MRC utilizes the *background subtraction* calibration method. In-phase (I) and quadrature-phase (Q) field component data of the received scattered signature from four sources are combined with a theoretical solution through Equation 61 to produce the "calibrated" I and Q data. Sources of the measured data include the target being investigated, the target's environment, a calibration target, and the calibration target's environment while the theoretical solution is that of the calibration target itself. I and Q data is then converted to calibrated RCS amplitude and phase data through (62) and (63). MRC calibration targets include a 15" diameter squat cylinder, a 4" diameter sphere, and an 8" diameter sphere [4]. Each is sufficient for monostatic measurements and more than one can be used during a single measurement cycle to improve accuracy. The cylinder produces a larger monostatic return, and is therefore, the preferred calibration source. Full polarimetric calibrations are produced with separate, oppositely polarized antennas for transmit and receive.

$$\sigma_{Calibrated} = \frac{\sigma_{Target} - \sigma_{Target\ background}}{\sigma_{Cal\ target} - \sigma_{Cal\ target\ background}} \sigma_{Theoretical} \tag{61}$$

$$RCS_{Amplitude} = 20\log_{10}\left(\sqrt{I^2 + Q^2}\right) \tag{62}$$

$$RCS_{Phase} = \arctan\left(\frac{Q}{I}\right) \tag{63}$$

The JRC utilizes the single reference calibration (SRC) procedure outlined in [30] for full-polarimetric monostatic and bistatic measurements. The reference target is a 30 cm diameter flat metal disk. For bistatic calibrations the diehedral is replaced with a wire mesh [23,24]. Calibrations produced with this method are purported to provide amplitude accuracy to within 0.3 dB, phase accuracy within 3 degrees, and cross-polarization purity of greater than 50 dB. The cross-pol purity is especially remarkable as it is nearly 25 dB higher than that produced by other calibration techniques [30].

Both calibration techniques utilized by the JRC and MRC appear to produce similar RCS amplitude response curves. Figure 13 shows a direct comparison of Object B's monostatic signature for 180 degrees of azimuth at 8 and 14 GHz. The slight angular shift noted in the JRC data's lobing structure compared to the MRC data may result from the quasi-monostatic measurement limitation of that facility (Tx and Rx antennas separated by 1.6 degrees). The plots deviate from each other by less than 2 dBsm for most aspect angles, so for all practical purposes the MRC and JRC measurements are regarded as equal in accuracy, precision, and value.

The data presented in Fig. 13 is only part of the entire data set acquired for this

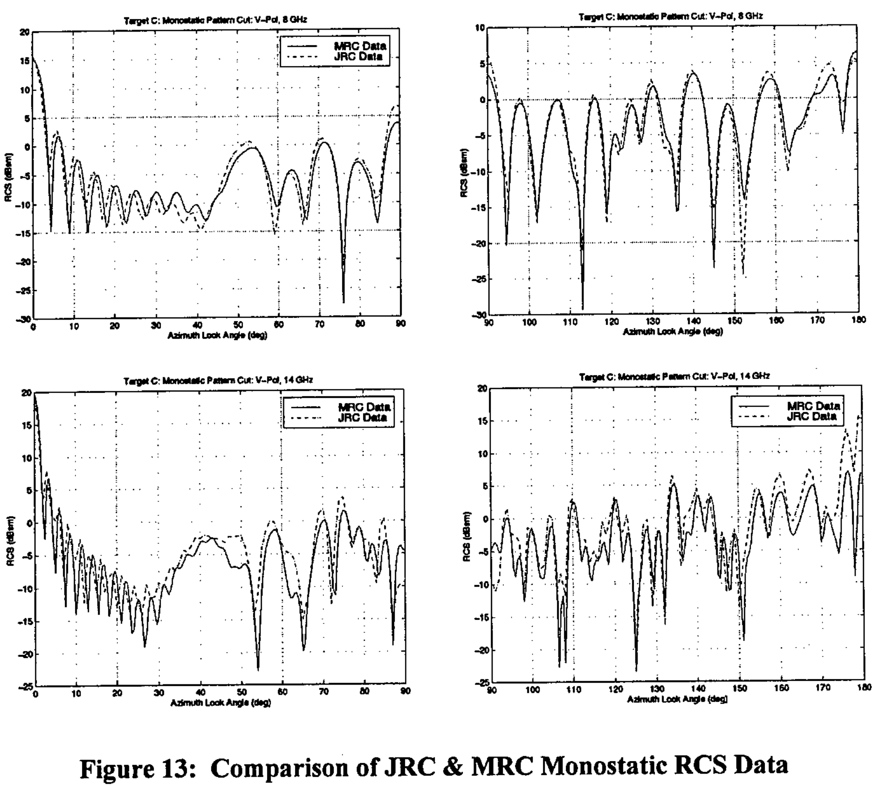


Figure 13: Comparison of JRC & MRC Monostatic RCS Data

research. Additional measurement and simulation runs complete the set and a host of original Matlab scripts and function files were created to help manipulate and format the data. The next sections describe the data and the Matlab scripts in greater detail in addition to explaining the target and measurement/simulation matrix selection.

III. Methodology

Overview:

The process by which data was gathered for this research can be found in Fig. 14. This flow chart reveals a three-pronged approach to gathering the raw data followed by a

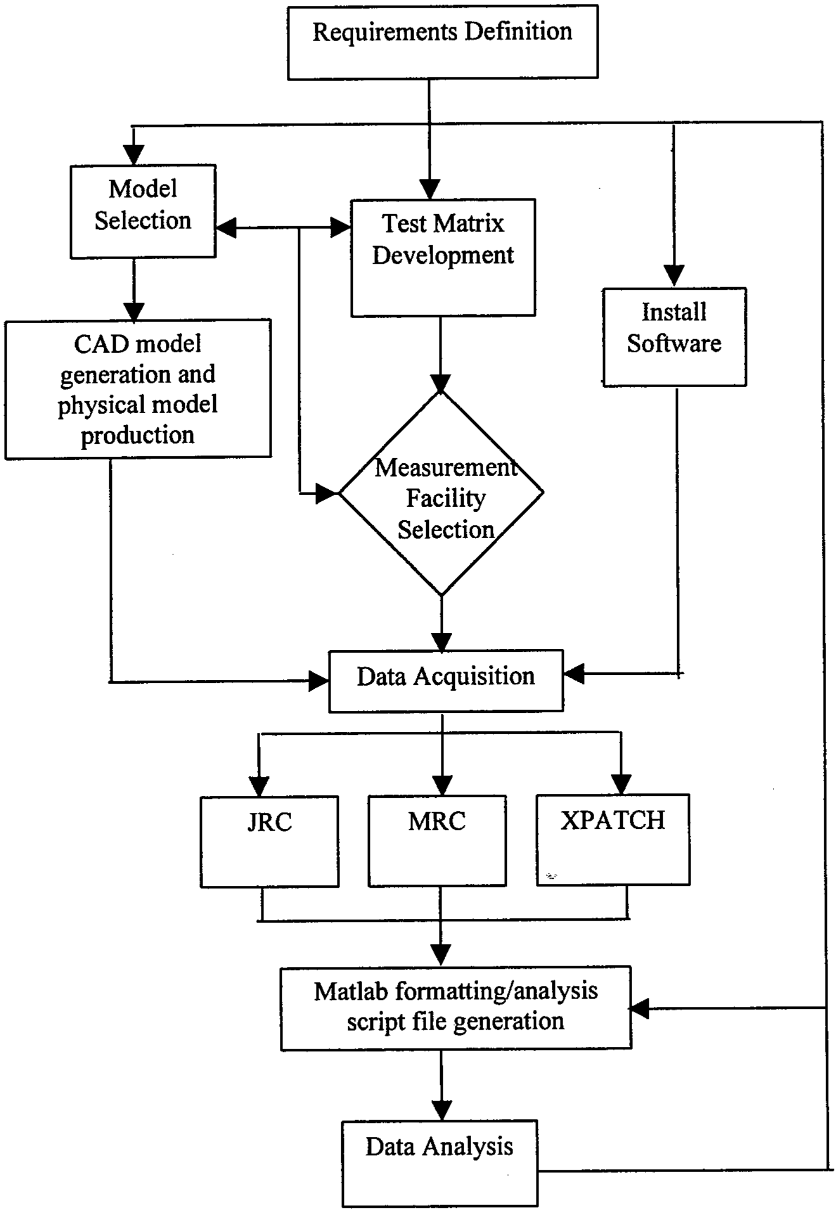

Requirements Definition
Model Selection
Test Matrix Development
Install Software
CAD model generation and physical model production
Measurement Facility Selection
Data Acquisition
JRC
MRC
XPATCH
Matlab formatting/analysis script file generation
Data Analysis

Figure 14: Research Progression Flow Diagram

sequence of code writing which would allow the data to be formatted, utilized, and manipulated. The analysis section followed and involved a great deal of feedback to the script writing phase as numerous bugs were eliminated and capabilities expanded as required.

Model Selection:

Because the computational methods under investigation are suspected of approximating only signatures from certain scattering mechanisms well, the test objects must produce distinctive scattering from an individual mechanism or some combination of several mechanisms. In particular, targets which support delineation between specular and non-specular phenomena are desired. Several other key considerations led to the selection of the targets in Fig 2a (repeated as Fig 15 for convenience). All objects were selected for their ease of manufacture, limitations of the measurement environment, similarity to objects which have already undergone rigorous monostatic appraisal, and their ability to produce certain scattering mechanisms. Testing the objects in the MRC or JRC chamber limited the objects' size to less than two meters in any dimension. The

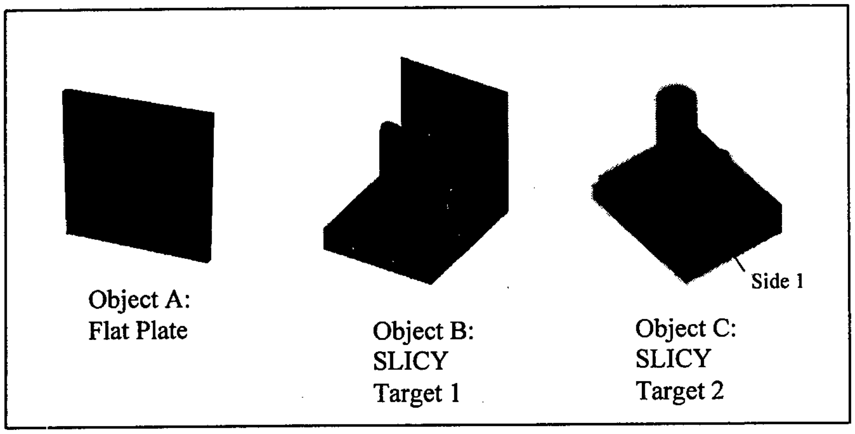


Figure 15: Research test objects

final dimensions (Fig. 16a-c, all dimensions in mm) were chosen to maximize each object's electrical size for the frequency bandwidth of interest (6-18 GHz) while acting within these constraints. The final products can be comfortably handled by one person and incorporate dimensions that fall within the resonance and optical regions depending on the frequency of interest. Anticipated scattering mechanisms helped solidify the choice of test objects. Object A provides a good specular return from broadside illumination, and when viewed bistatically from edge-on, should support good edge scattering effects for one polarization. Object B's geometry includes a large shadowing feature, which should block a fair amount of scattering from smaller bistatic look angles (Rx position angles) when the transmitter illumination angle is within 45 degrees of broadside to the flat plate. It includes a canted, hollow cylinder, which should provide some cavity coupling when illuminated. Object C provides an analog to Object B without a significant shadowing feature or large cavity coupler. Aluminum construction ensures that the reflecting surface closely approximates a perfect electrical conductor (PEC). The thin, flat metal parts are machined from 1.8 mm aluminum sheeting, the cylinders are manufactured from 3" diameter aluminum pipe, and end caps for each cylinder are cut from solid aluminum stock. A single base unit manufactured from a solid aluminum block is utilized for both Objects B & C, and the flat plate of Object A and Object C are one in the same. Each part is hand polished after being cut to the proper dimension by computer-controlled milling equipment with exacting tolerances of less than 1/1000th of an inch. Three sets were manufactured; one sent to the JRC, one sent to Ohio State University, and one

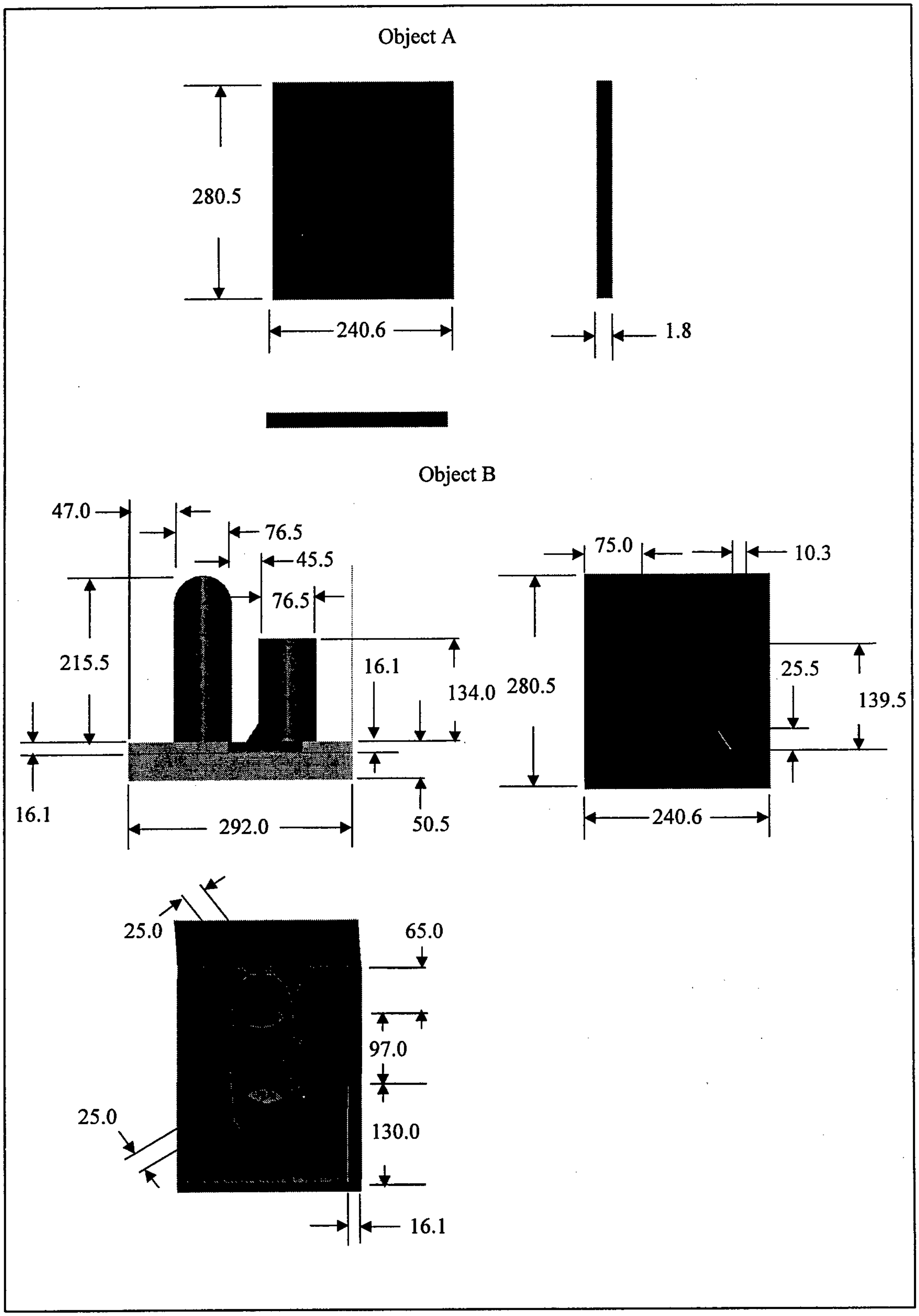


Figure 16 a) Object A Schematics, b) Object B Schematics

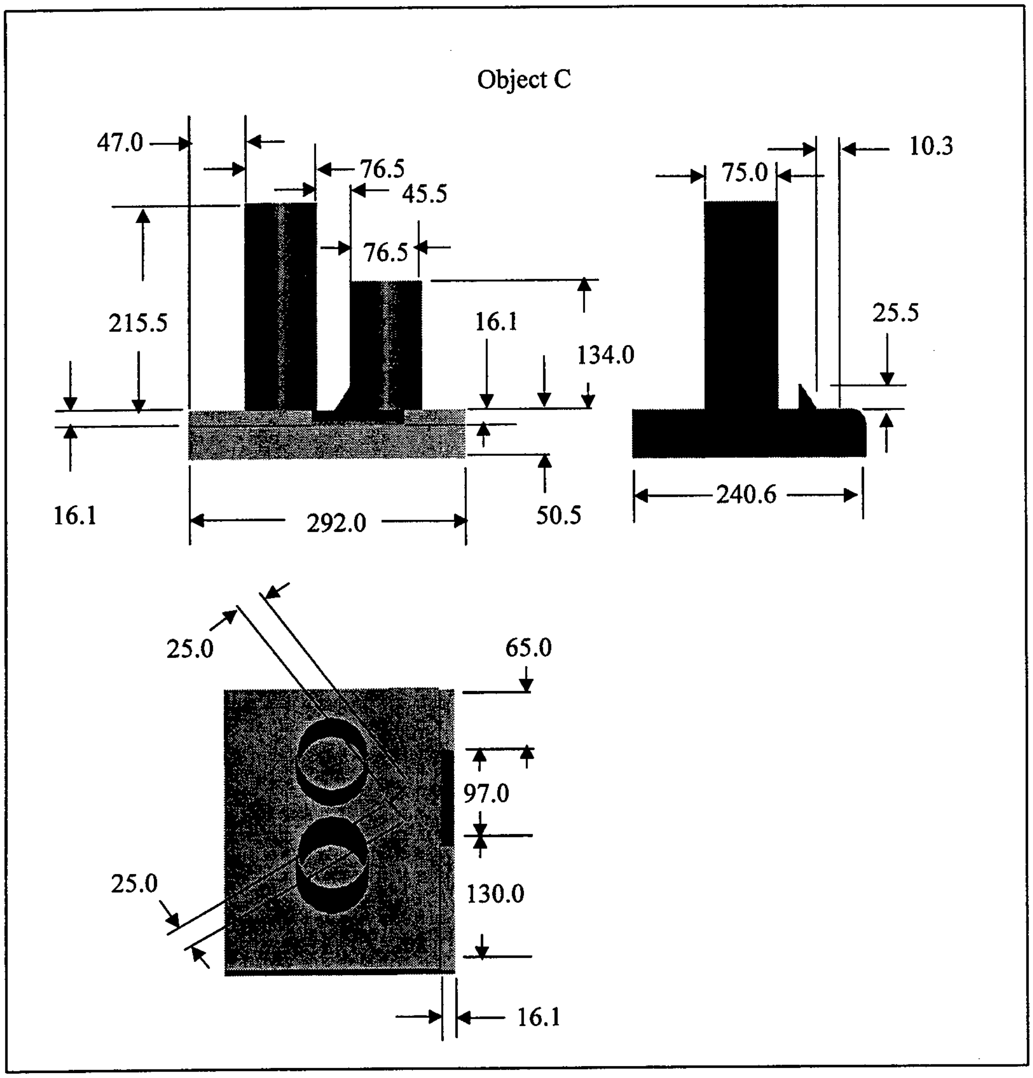


Figure 16 c) Object C Schematics

remaining at MRC and AFIT.

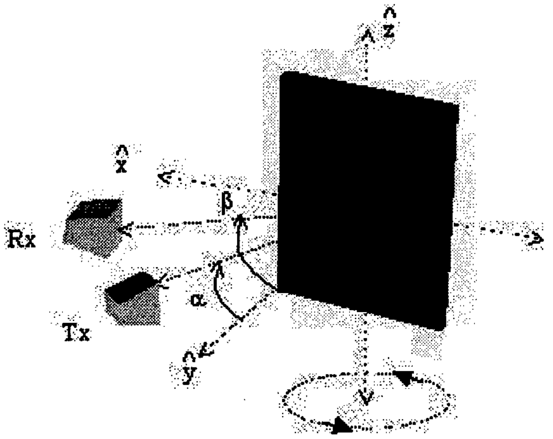
Monostatic and bistatic measured and simulated RCS amplitude and phase data was generated for each object for similar orientations and frequencies. As mentioned, the frequency band of interest is 6-18 GHz. This band is primarily dictated by the capabilities of the MRC and JRC measurement chambers. Noise begins to dominate measured reflections in both chambers above 18 GHz, and the electrical size of most target surface features becomes too small for frequencies below 6 GHz. A complete measurement test matrix and description can be found in the following section. For the most part, simulation data matrices mirror that of the measured data sets.

Measured Data Acquisition:

This section describes scattering data collected at both the MRC and JRC chambers. Polarization is defined relative to the ground plane for all analysis even though the actual collection sequence included several different measurement orientations. Objects should be thought of as having been placed on some sort of absorbing surface in the configurations shown in Figures 17-19 with the positive z-axis pointing in the positive vertical direction (up, away form the ground). VV-pol describes an E-field perpendicular to the ground plane (for Tx and Rx) and HH-pol has it parallel. (for Tx and Rx) All measurements occur in the azimuthal plane with either the object being rotated about the z-axis (stationary Tx./Rx antenna) for monostatic measurements or the receiver being rotated about a stationary object (and stationary Tx antenna) for bistatic measurements. This convention should help the reader conceptualize scattering from each object more easily. Appendix A describes the actual measurement orientation

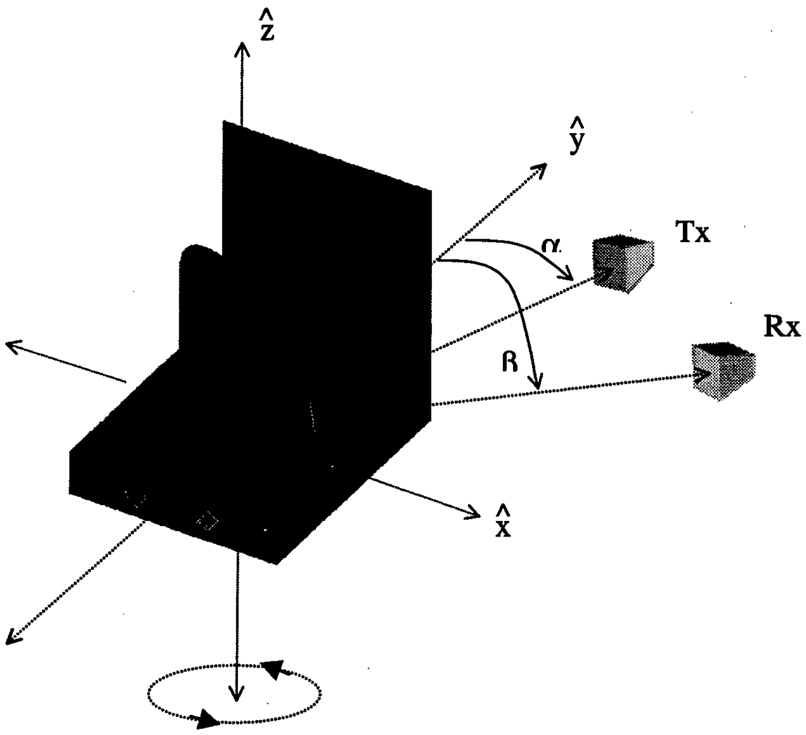
for all targets.

The complete target test matrix is shown in Tables 2a-c. Neither facility could perform all the desired measurements, so the matrix attempts to maximize the amount of data collected between the two. Only monostatic data at VV-pol is collected from MRC, but the JRC acquires full polarimetric monostatic and bistatic data. Data at 10 MHz increments between 6-18 GHz, for 180 degrees of azimuth (0.5 deg step size) is



Monostatic measurement
rotation direction

Figure 17: Object A Measurement Orientation



Monostatic measurement
rotation direction

Figure 18: Object B Measurement Orientation

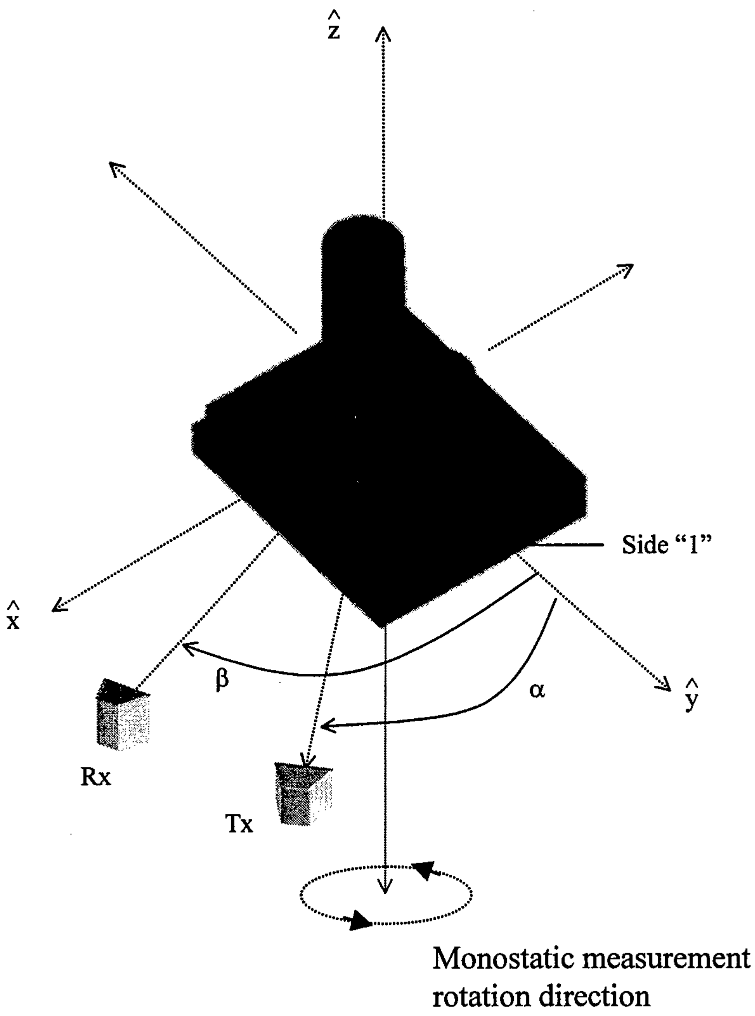


Figure 19: Object C Measurement Orientation

acquired at MRC but only 7-15 GHz (10 MHz intervals), 360 degrees of azimuth (1.0 deg. angular resolution) at JRC. A lower angular resolution and frequency bandwidth are necessitated by range time limitations at the JRC, and although MRC only provides VV-pol data (again, due to scheduling constraints), its higher resolution is deemed an adequate trade-off. Monostatic data are collected for Object B from both MRC and JRC locations for two reasons: 1) to compare data sets from both facilities and 2) to extract Kell's and Crispin's approximate bistatic RCS. JRC monostatic data are also available for Object A and MRC monostatic data for Object C. MRC monostatic data is used to

Table 2a: MRC Monostatic Measurement Matrix

Object	Measurement Type	Polarization	RF Bandwisth/ Step Size (GHz)	Azimuth Angle Range/Step Size (deg)
B	Monostatic in azimuth (x-y) plane	VV-pol	6-18/0.01	0-360/0.5
C	Monostatic in azimuth (x-y) plane	VV-pol	6-18/0.01	0-360/0.5

Table 2b: JRC Monostatic Measurement Matrix

Object	Measurement Type	Polarization	RF Bandwisth/ Step Size (GHz)	Azimuth Angle Range/Step Size (deg)
A	Monostatic in azimuth (x-y) plane	Full-polarimetric	7-15/0.01	0-360/1.0
B	Monostatic in azimuth (x-y) plane	Full-polarimetric	7-15/0.01	0-360/1.0

Table 2c: JRC Bistatic Measurement Matrix

Target	Measurement Type	RF Bandwisth/ Step Size (GHz)	Polarization	Tx Orientation α Angle (deg)	Rx Orientation β Angle (deg)/ Increment Step Size (deg)
A	Bistatic in azimuth (x-y) plane	7-15/0.01	Full-polarimetric	0	4 to 184/1.0 4 to 184/1.0
B	Bistatic in azimuth (x-y) plane	7-15/0.01	Full-polarimetric	45	-20 to +200/1.0 -20 to +200/1.0
C	Bistatic in azimuth (x-y) plane	7-15/0.01	Full-polarimetric	45	-20 to +200/1.0 -20 to +200/1.0

compute Kell's and Crispin's equivalent bistatic RCS for Object C. The bistatic data collected at the JRC exists for all test objects. Transmitter and Receiver position angles (α and β respectively) are positively measured in the x-y plane from the +y-axis toward the +x-axis. The bistatic angle equates to $\beta-\alpha$. The Tx illumination angle, α, for Object A is 0 degrees and the Rx look angle, β, is rotated from 4 to 184 degrees (through forward scatter). Objects B & C are illuminated from $\alpha = 45$ degrees, and the Rx rotates from $\beta = -20$ through +200 degrees azimuth. Again the selection of only a single transmitter illumination angle is dictated by time constraints and its position is justified because it provides good shadowing on Object B. A peak off Object A and B's flat plate broadside determines the 0 reference angle, while Object C's "side 1" (Fig. 19) serves as its 0 degree reference. No bistatic data exists within +/-4 degrees of the α angle because of measurement environment restrictions at the JRC.

Simulation Data Acquisition:

The Xpatchf module of the Xpatch 2.4d prediction suite generated most of the simulated data. Simulations corresponding to each of the measured data sets comprise the bulk of this set. Each simulation file utilizes target facet files created by AFRL/SNAD. They are based on dimensional measurements (mm units) from one of the original test models. The facet files are created as NURBS entities and then facetized all within the Windows-based Rhinoceros CAD package [25]. An AFRL post-processing script formats the facetized geometries into an Xpatch-readable format. No curvature files are associated with the facets files, but the resolution of the individual facets is deemed fine enough to prevent significant facetization effects. Edge files are extracted with the Cifer utility and incorporated into all simulations to allow computation of diffraction effects. A sequence of runs is also performed on each object without diffraction effects for comparison. Full polarimetric RCS amplitude data is stored in the *.rcs file for each simulation. Modified versions of these files (sans first 13 lines of textual header information) are used as input files for a series of Matlab scripts written to format/display the collected data.

Tables 3 & 4 list many of the simulation default settings for various simulations. Selection of these defaults proceeds from the Xpatch bistatic capabilities (limitations)

Table 3: Monostatic Xpatch Default Settings

Xpatchf Settings Enabled	**Object A**	**Object B**	**Object C**
PO solution for first bounce	Y	Y	Y
Edge diffraction	Y/N	Y/N	Y/N
Divergence Factor	N	N	N
Scattering contribution from all bounces, with blockage checks (GTD option)	N	Y	Y
Scattering contribution from all bounces, without blockage checks (GTD option)	Y	Y	Y

Table 4: Bistatic Xpatch Default Settings

Xpatchf Settings Enabled	Object A	Object B	Object C
SBR solution for all bounces	Y	Y	Y
Edge diffraction	Y/N	Y/N	Y/N
Divergence Factor	N	N	N
Scattering contribution from all bounces, with blockage checks (GTD option)	N	Y	Y
Scattering contribution from all bounces, without blockage checks (PTD option)	Y	Y	Y

being investigated. The first item for review is Xpatch's basic PO prediction algorithm. To determine if this is working correctly, specular amplitudes should correspond to measured and hand-computed values and specular sidelobe structure should be stable. The PTD implementation is investigated for its ability to accurately account for edge diffraction phenomena. Both of these can most readily be observed from Object A's signature. Object B and C's geometries provide an opportunity to inspect the contribution of intermediate bounce reflections (with and without blockage) and shadowing effects. They can also highlight edge diffraction effects for more complex geometries.

Matlab Scripts:

Matlab v5.3 is the application of choice for extracting, formatting, and manipulating the measured and simulated data. Raw measured data was presented in two different formats from MRC and JRC. Simulated data came in a third, Xpatch-specific ascii text file format. Therefore, a common simulation environment in which all data could be formatted and manipulated was desired. Matlab handily met the requirement.

Original script and function files were developed to handle several different tasks. Some extract and generate RCS amplitude and phase data from the measured data sets and write it to new ascii text tab-delimited files. Others cull amplitude data from either these new measured data files or the Xpatch data files based on user-specified options. Up to seven inputs are requested from the user before these types of scripts are executed. Although not common to every m-file, the required data are usually: 1) the desired object, 2) the RF of interest, 3) the polarization, 4) the Tx illumination angle, 5) the Rx look angle, 6) the figure number to which the data is plotted, and 7) the plot line type. After finding the requested information the script plots the data to a given figure number. Other scripts perform actions such as computing RCS averages over angular regions, computing/displaying the difference vector between two data sets, and printing the plots to .tiff and postscript files. Table 4 summarizes the function of the more important scripts and Appendix C contains a complete printout of each script file. These scripts were used to create all the plots contained in the next section, Results & Analysis.

Table 5: Matlab Script File Description

Script Filename	Script Function
kell.m	Extracts Kell's equivalent bistatic RCS from MRC monostatic data file and prints to screen.
kellj.m	Extracts Kell's equivalent bistatic RCS from JRC monostatic data file and prints to screen.
crisp.m	Extracts Crispin's equivalent bistatic RCS from MRC monostatic data file and prints to screen.
crispj.m	Extracts Crispin's equivalent bistatic RCS from JRC monostatic data file and prints to screen.
jplott.m	Plots desired JRC measured bistatic data to screen
splott.m	Plots desired Xpatch bistatic measured data to screen
plotmon.m	Plots desired MRC measured monostatic data to screen
jplotmon.m	Plots desired JRC measured monostatic data to screen
splotmon.m	Plots desired Xpatch monostatic data to screen
avg.m	Plots the sliding-window average of given RCS matrix based on input window size
fd.m	Computes/displays difference vector between measured bistatic data vector (loaded with jplott.m script) and input RCS matrix
printt.m	Prints given figure number to postscript and .tif-format (no compression) files

IV. Results & Analysis

Data analysis proceeds though two distinct phases: 1) a comparison of Xpatch predicted data to the measured data, and 2) a comparison of the MBET predictions to the measured data. The collected data spans a large frequency bandwidth and angular extent, but only discrete frequencies and limited angular regions can be reasonably analyzed. By its very nature, the research also focuses on only smaller angle bistatics, but a few larger bistatic angle measurements are referenced for clarification of some items. The following analysis investigates each bistatic target signature at only two frequencies (one *low*, one *high*) for a given angular region less than 110 degrees in width (defined by the Rx azimuth angle position, β, relative the aforementioned 0 degree reference for each object). Two RFs are chosen near either end of the collection spectrum to help differentiate any resonance region effects which may be notable at lower frequencies.

The presentation of the data follows the same pattern for all objects. Pattern cut plots,

Table 6: Primary Analysis Matrix

Test Object	Frequency Analyzed (GHz)	Investigation		Rx Look Angle Region Reviewed (deg)
		Xpatch	Kell's/ Crispin's MBET	
A	8, 14	X		0-110
	8, 14		X	0-110
B	8, 15	X		0-110, 110-160
	8, 12		X	0-110
C	8, 15	X		0-110, 110-160
	8, 15		X	0-110

difference plots, and ray trace diagrams are the main conveyors of information. For consistency's sake pattern cut plots typically show RCS amplitude data as a function of the Rx look angle, β, relative to the 0 reference position, not as a function of bistatic angle. The difference plots are only used to help visualize the MBET pattern cut data discrepancies. They show the subtraction of the measured data from the MBET prediction as a function of Rx look angle. (positive data indicates that the MBET predicts a higher amplitude than is measured). Ray trace diagrams help explain the scattering mechanisms reviewed in the Scattering Analysis and Xpatch Analysis sections.

The matrix of analyzed data is summarized in Table 5. The higher frequency values differ amongst targets to maximize the RF coverage and to provide sufficient data for investigation. This later point will become readily apparent when discussing Kell's equivalent bistatic RCS computations. Pattern cut plots of all measured monostatic and bistatic RCS data analyzed in the section are provided in Appendix B for review. The data's origin is also annotated.

Xpatch Analysis:

Xpatch has demonstrated ample capability for predicting monostatic signatures in the past, so a direct comparison between Xpatch predictions and the measured monostatic data sets is unnecessary. The discussion instead begins with an inspection of the bistatic data collected against the flat plate (Object A).

Object A:

As mentioned the Tx illumination angle, α, is broadside to the flat plate. It produces a distinct specular scattering signature characterized by a narrow main lobe at β=0 degrees, sidelobe activity through 60-70 degrees, and diffraction effects through edge-on incidence (β=90 degrees). Figures 20-21a-b show the patterns for 8 & 15 GHz (VV-pol & HH-pol) for the measured, Xpatch's PO, and Xpatch's PTD solutions. Immediately one notices the similarity in lobing structure, amplitude, and null placement for the first 30-40 degrees. This is expected. The PO solution begins to diverge, especially at HH-pol, from the measured data due to a lack of diffraction inclusion. The PTD solution correlates to the measured data to a greater extent, but on average predicts higher than measured through β = 90 degrees, and the lobes appears to be skewed toward higher bistatic angles. At β = 90 degrees a large discontinuity appears in all the predicted solutions. This may be explained by considering a limitation in the PO computation. Through β = 90 deg. the Rx sees two edges, at 90 deg. the far edge disappears, and beyond 90 deg. (in the shadow region) it remains invisible because no PO currents are computed there. One draws two conclusions based on these findings: 1) the PTD solution is either not being computed correctly within Xpatch2.4d or is insufficient for predicting the bistatic behavior of such a diffraction, and 2) no provision exists to account for edge diffraction effects from shadowed features.

A corollary observation should be noted here. The discrepancy in the null/lobe positions could indicate that either the measured data is faulty (e.g. the alignment of the object and Tx/Rx antennas is slightly off) or the Xpatch prediction is in error. More information regarding the strict measurement conditions is needed to conclusively state the more likely situation, but based on Xpatch's clear inability at β = 90 degrees, all measured data are regarded as more exact in this situation.

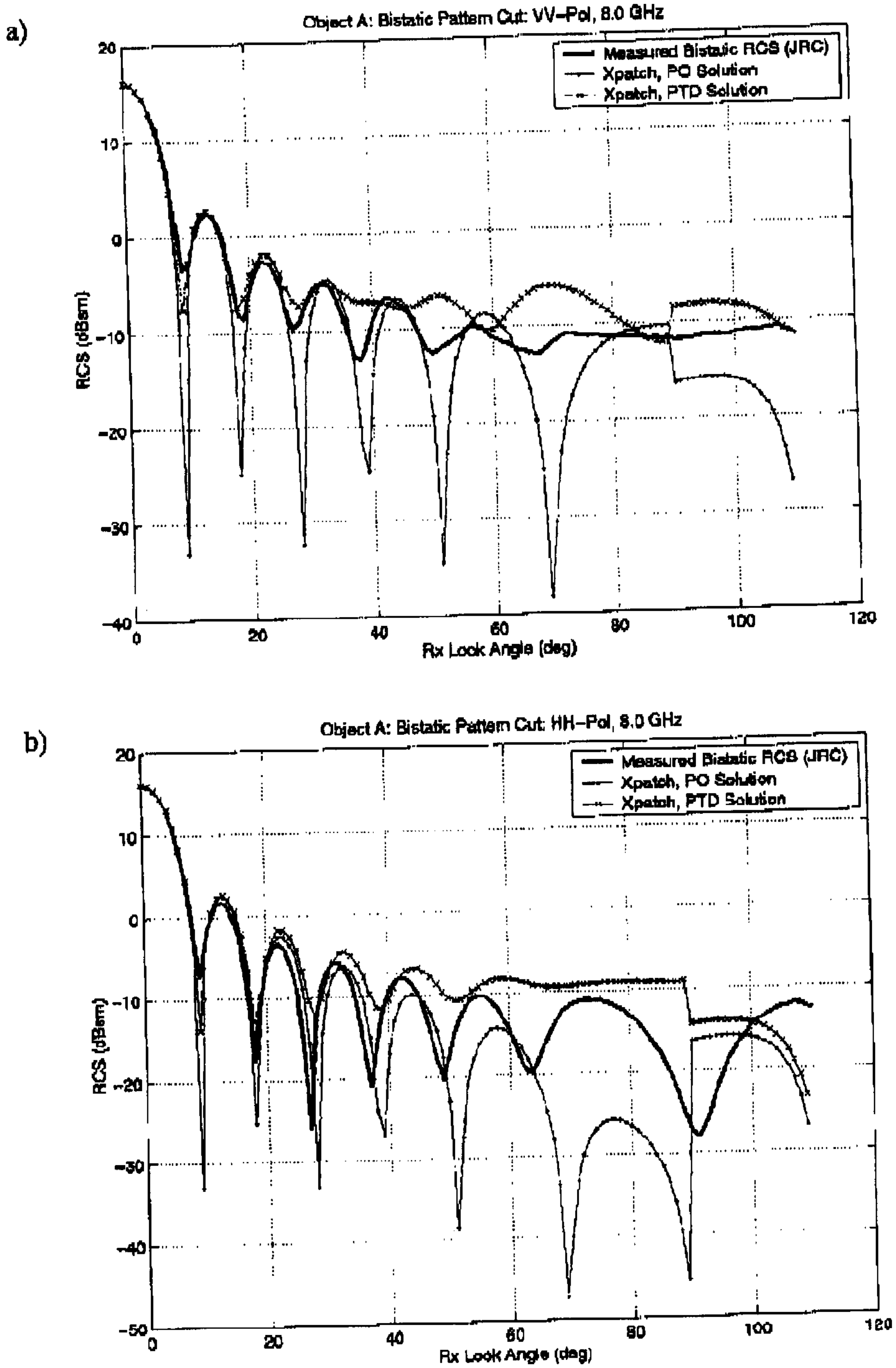


Figure 20a-b: Object A, Bistatic Xpatch RCS Predictions vs. Measured (JRC) data, 8 GHz, a) VV-pol, b) HH-pol

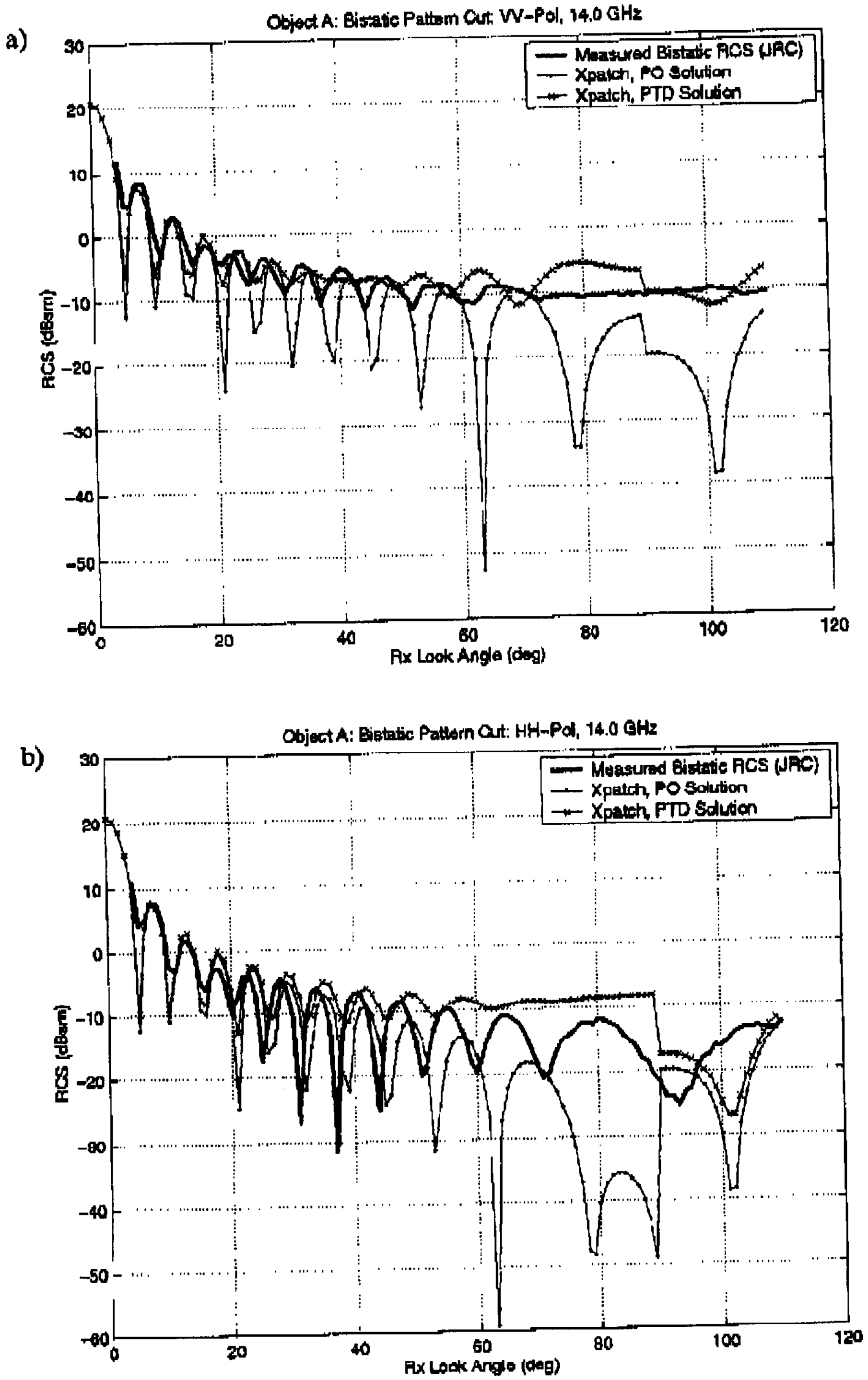


Figure 21a-b: Object A, Bistatic Xpatch RCS Predictions vs. Measured (JRC) data, 14 GHz, a) VV-pol, b) HH-pol

Objects B & C:

Our discussion continues by moving to the more complex geometries. The analysis for both targets is conducted simultaneously to highlight Xpatch's weaknesses, unanticipated strengths, and to discuss the various scattering mechanisms which form the overall scattered signature. It should be noted at this point that target features are between 0.67 and 14.03 lambda depending on the frequency of interest. Table 7 summarizes some of the target feature electrical sizes. Some experts define *electrically large* as anything greater than 3λ [17], but more commonly its referred to as feature lengths larger than 10λ. Due to the small nature of some features (i.e. the dihedral and cylinder diameters), Xpatch should not perform well at lower frequencies (i.e. 8 GHz).

Table 7: Electrical Size of Various Object Features

Object	Feature	Electrical Size (L/λ)			
		8 GHz	12 GHz	14 GHz	15 GHz
B/C	Dihedral length	0.67	1.00	1.17	1.25
B/C	Cylinder diameter	2.04	3.06	3.57	3.83
B	Flat plate height	7.48	11.22	13.09	14.03
B	Canted cylinder height (to low point on open end)	3.72	5.58	6.51	6.98
C	Large cylinder height	5.75	8.62	10.06	10.78
B/C	Small cylinder height	3.57	5.36	6.25	6.70

The following discussion only compares Xpatch's default PTD solution to the measured data. The default PTD solution computes the PO field from the first and last bounce points and adds any edge diffraction present. For these objects, edge diffraction contributions should be relatively small from almost any vantage point, and summing the PO component from all bounces shouldn't add appreciably to the overall signature. This supposition seems legitimate after a review of several PO and PTD computations as

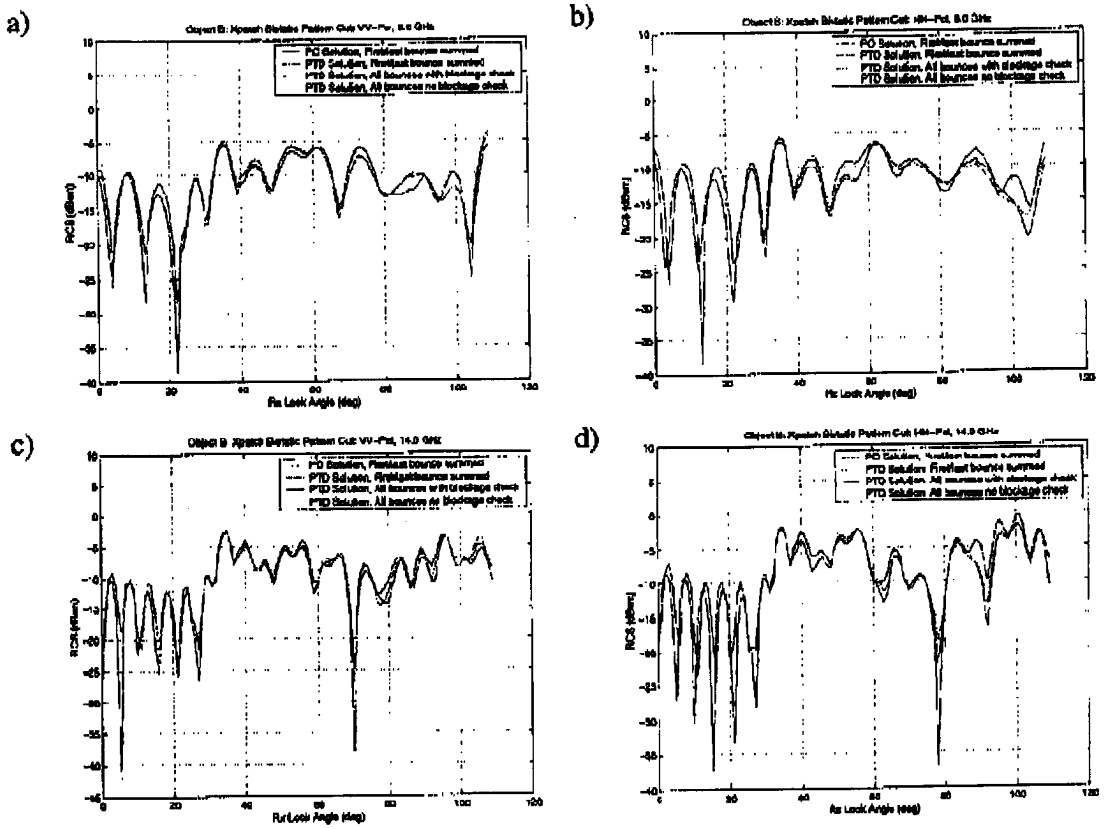


Figure 22a-d: Object B, Bistatic Xpatch RCS Predictions PO vs. PTD Solutions, a) VV-pol, 8 GHz, 0-110 deg β angle; b) HH-pol, 8 GHz, 0-110 deg β angle; c) VV-pol, 14 GHz, 0-110 deg β angle, d) HH-pol, 14 GHz, 0-110 deg β angle

shown in Figures 22-23. For all practical purposes the signatures from the four PO and PTD solutions are the same, so only comparing one against the measured data is necessary. The default Xpatch2.4d PTD computation is the chosen prediction. The Tx illumination angle, α, is also annotated on each pattern cut plot with a dashed vertical line for the reader's convenience. Remember that no measured data exists within +/-4 degrees of the α angle.

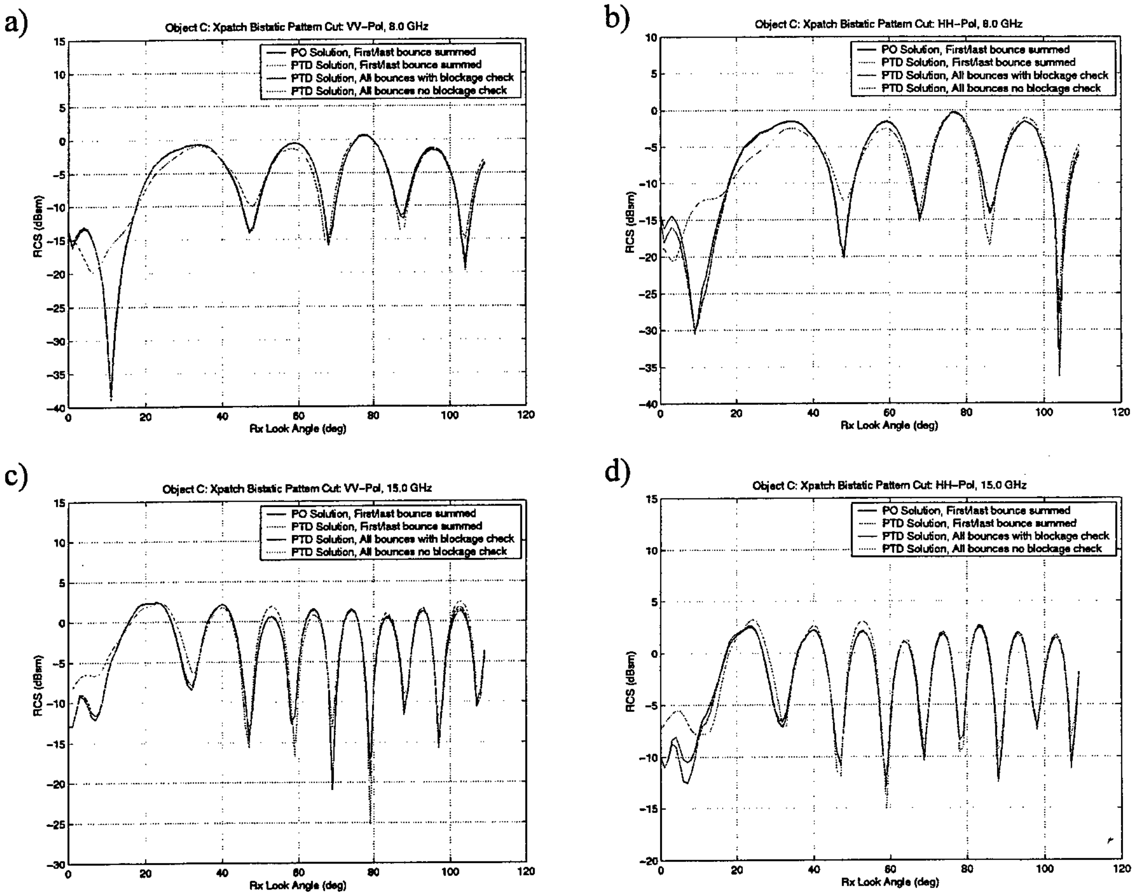


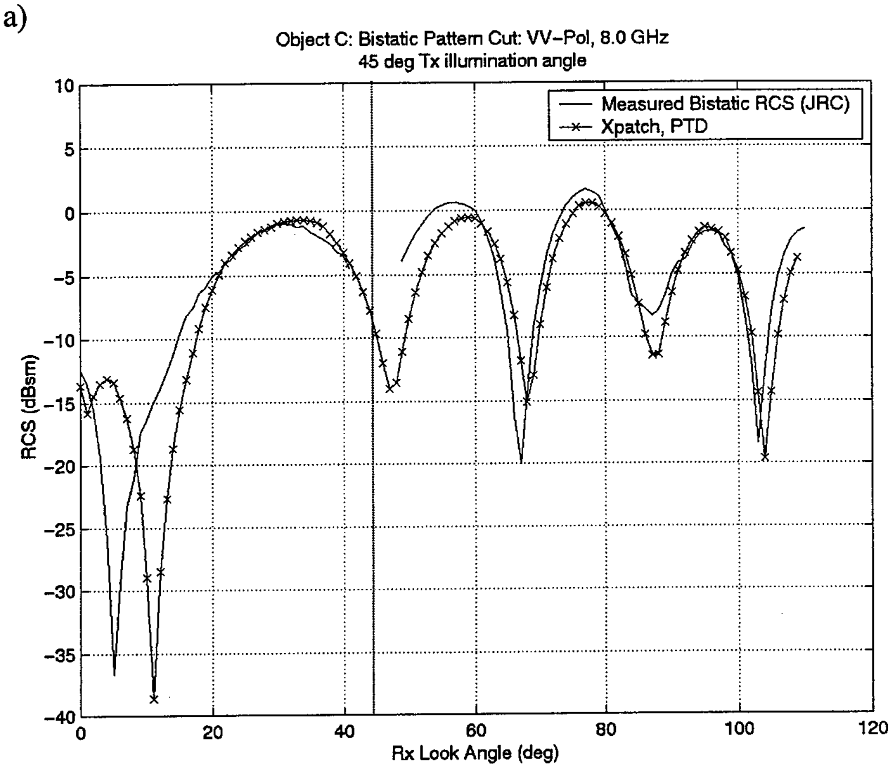
Figure 23a-d: Object C, Bistatic Xpatch RCS Predictions PO vs. PTD Solutions, a) VV-pol, 8 GHz, 0-110 deg β angle; b) HH-pol, 8 GHz, 0-110 deg β angle; c) VV-pol, 15 GHz, 0-110 deg β angle, d) HH-pol, 15 GHz, 0-110 deg β angle

The analysis begins with a review of the *simpler* target, Object C. This object is simpler in the sense that no large shadowing geometry is present nor is there a canted cylinder into which directly incident energy can couple and be scattered.

In evaluating the measured data's lobing structure in Figures 24 & 25, one notices what appears to be two main reflection sources beating in and out of phase over most of the covered region. They are most likely specular in nature and evolve directly from the cylinder bodies due to the similarity between polarizations. An inspection of Xpatch's ray trace history (Figs. 28 & 29) support this hypothesis. A small amount of multi-bounce interaction between cylinders is present but is probably small enough so as not to compete with the main cylinder speculars. The dihedral probably isn't a significant scatterer at these frequencies or a larger return would be present near the transmitter illumination angle. Xpatch predicts the signatures surprisingly well even at the lower frequency. The higher frequency lobes (Fig. 25) appear to be slightly skewed toward higher bistatic angles than the true data, but the amplitudes are for the most part correct. The discrepancies may be due to a slight misalignment of the target during measurement, which could produce small, second-order surface wave and diffraction effects.

At larger bistatic angles (Figs 26-27), the correlation isn't quite as good, but Xpatch predictions remain within 5 dB of the measured data for much of the region. The specular reflections from each cylinder being much less pronounced here are probably responsible for the Xpatch divergence (additional reflection sources are beginning to compete). Again, the simulated data also seems to be shifted toward larger β angles. The measured data may be undersampled or clipped near 135 degrees in Fig. 27. One may expect to see a large specular spike at this angle in the predicted data due to the facetized geometry. The curved surface of each cylinder is actually represented by flat sections, some of which are aligned such that they should produce a specular reflection in that direction (via Snell's law). The Xpatch data generally shows a null in that region similar to the measured data (although not as deep) leading one to conclude that there is some phasor dependency to the nature of the speculars. If this were true, simulations at higher frequencies should effectively minimize the effect of all phasor interactions and produce a strong specular at 135 deg. Figure 30 reveals such a spike.

a)



b)

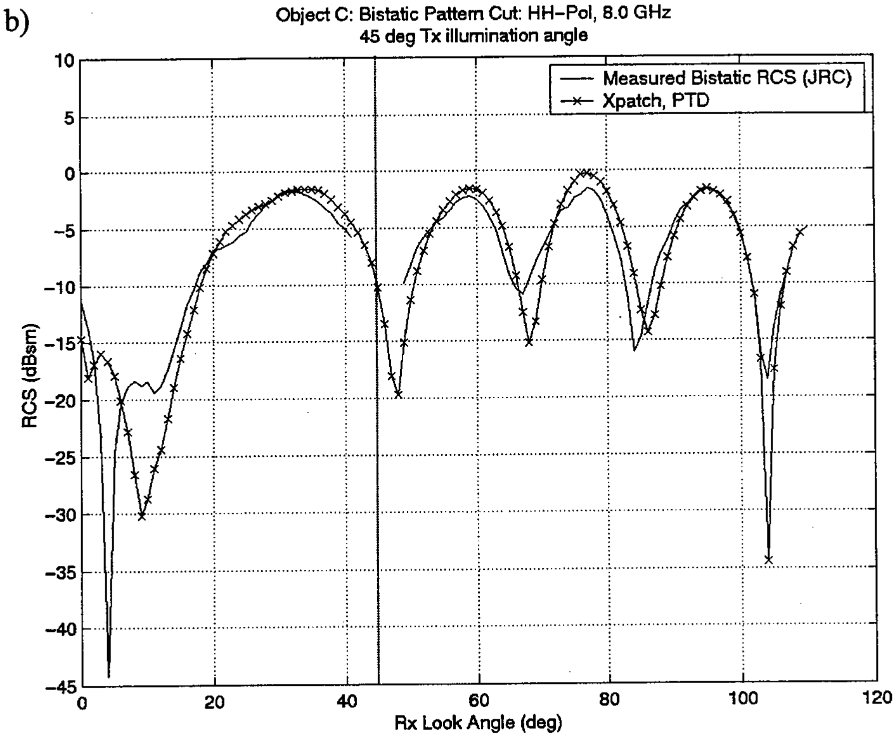


Figure 24a-b: Object C, Bistatic Xpatch Predictions (PTD) vs. Measured (JRC) Data, 8 GHz, 0-110 degrees Rx look angle
a) VV-pol, b) HH-pol

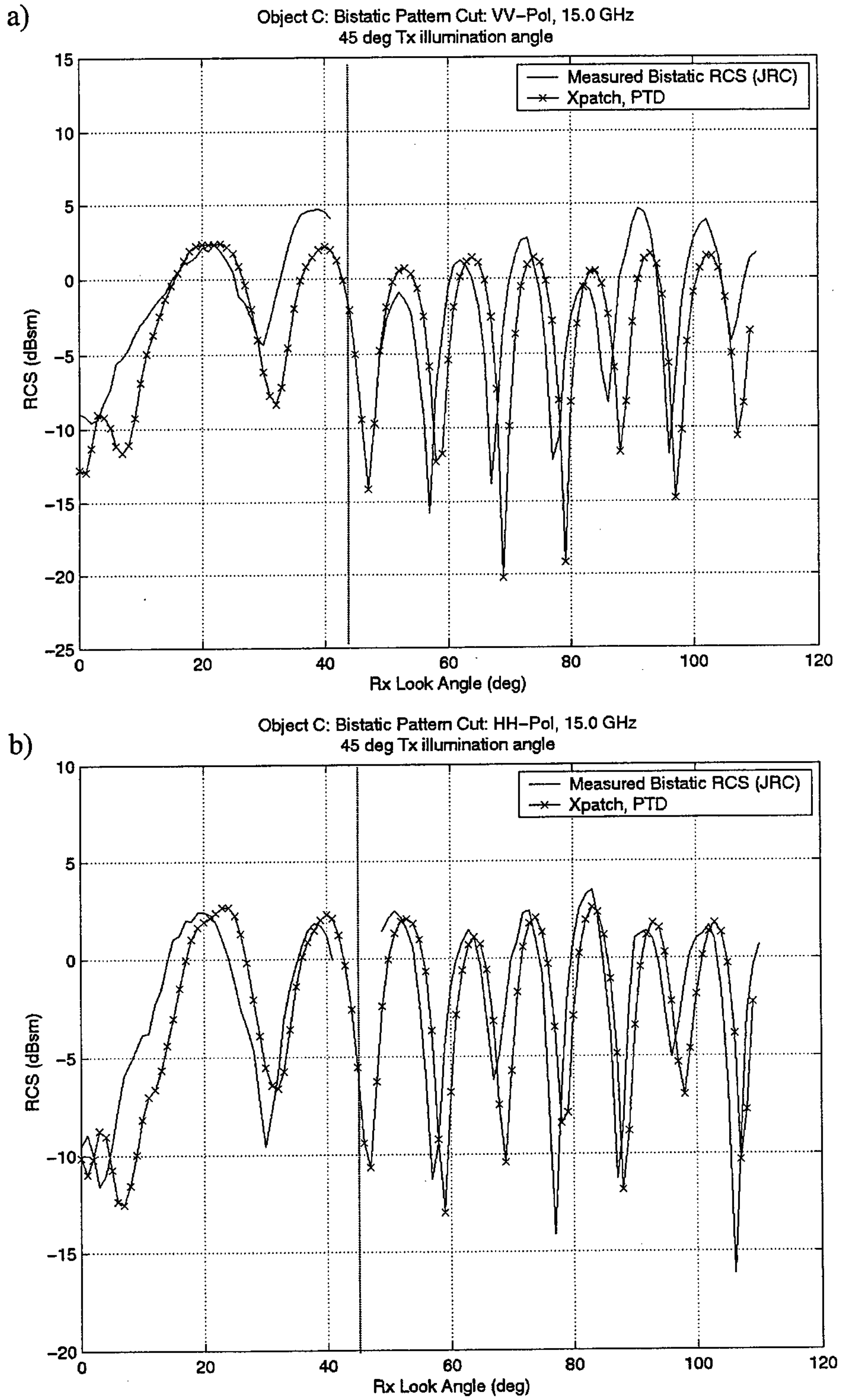
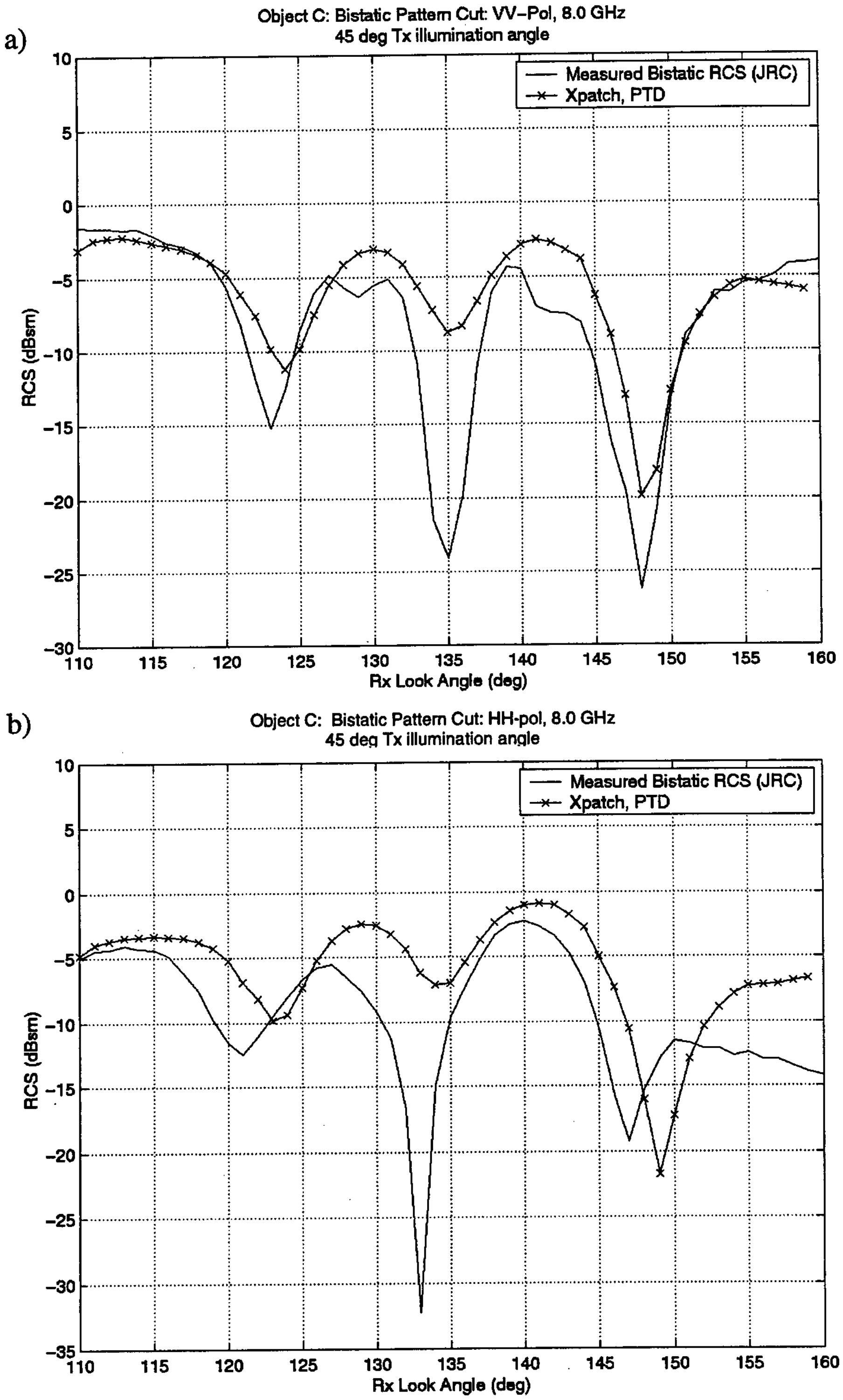


Figure 25a-b: Object C, Bistatic Xpatch Predictions (PTD) vs. Measured (JRC) Data, 15 GHz, 0-110 degrees Rx look angle, a) VV-pol, b) HH-pol


a)

Object C: Bistatic Pattern Cut: VV-Pol, 8.0 GHz
45 deg Tx illumination angle

b)

Object C: Bistatic Pattern Cut: HH-pol, 8.0 GHz
45 deg Tx illumination angle

Figure 26a-b: Object C, Bistatic Xpatch Predictions (PTD) vs. Measured (JRC) Data, 8 GHz, 110-160 degrees Rx look angle, a) VV-pol, b) HH-pol

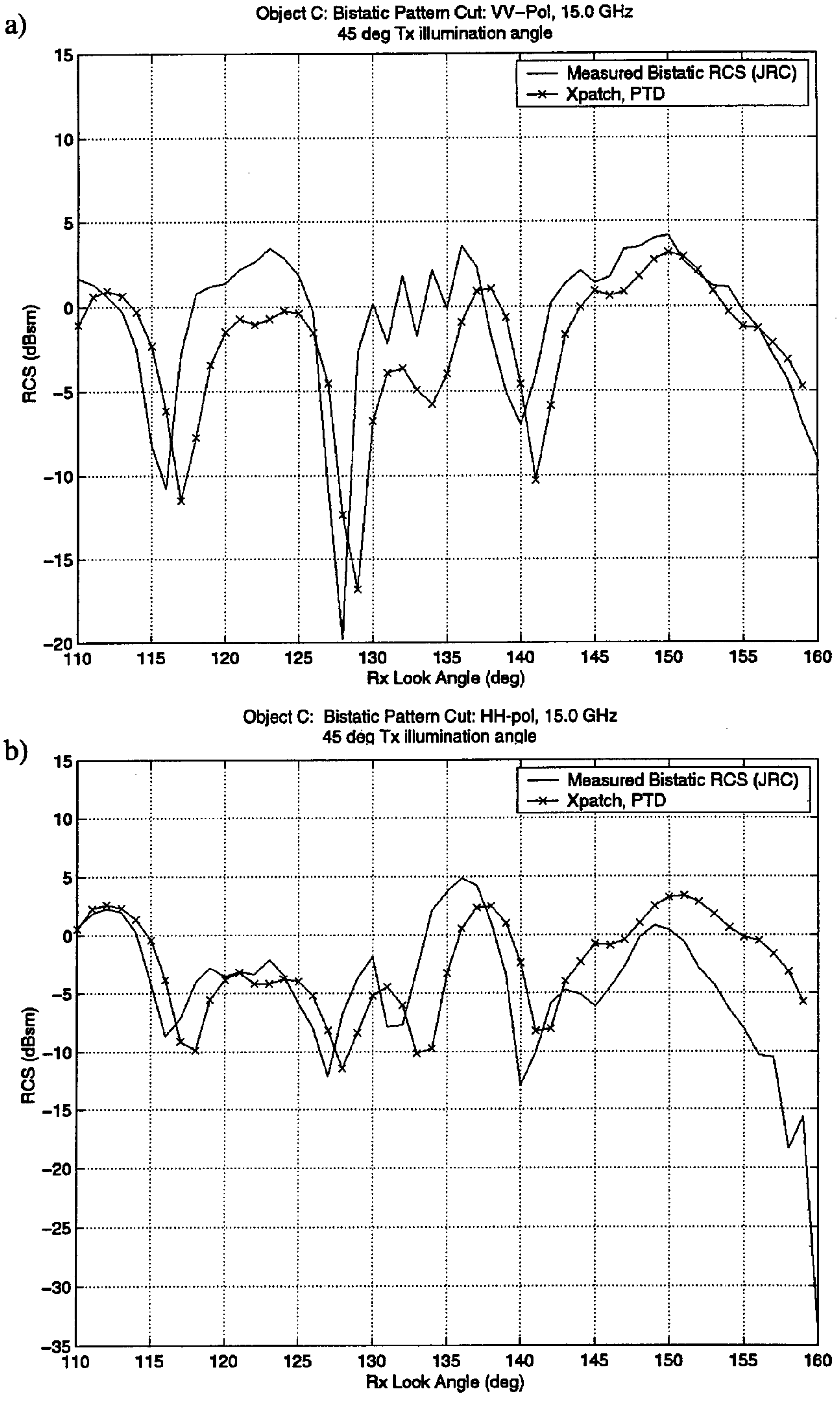


Figure 27a-b: Object C, Bistatic Xpatch Predictions (PTD) vs. Measured (JRC) Data, 15 GHz, 110-160 degrees Rx look angle, a) VV-pol, b) HH-pol

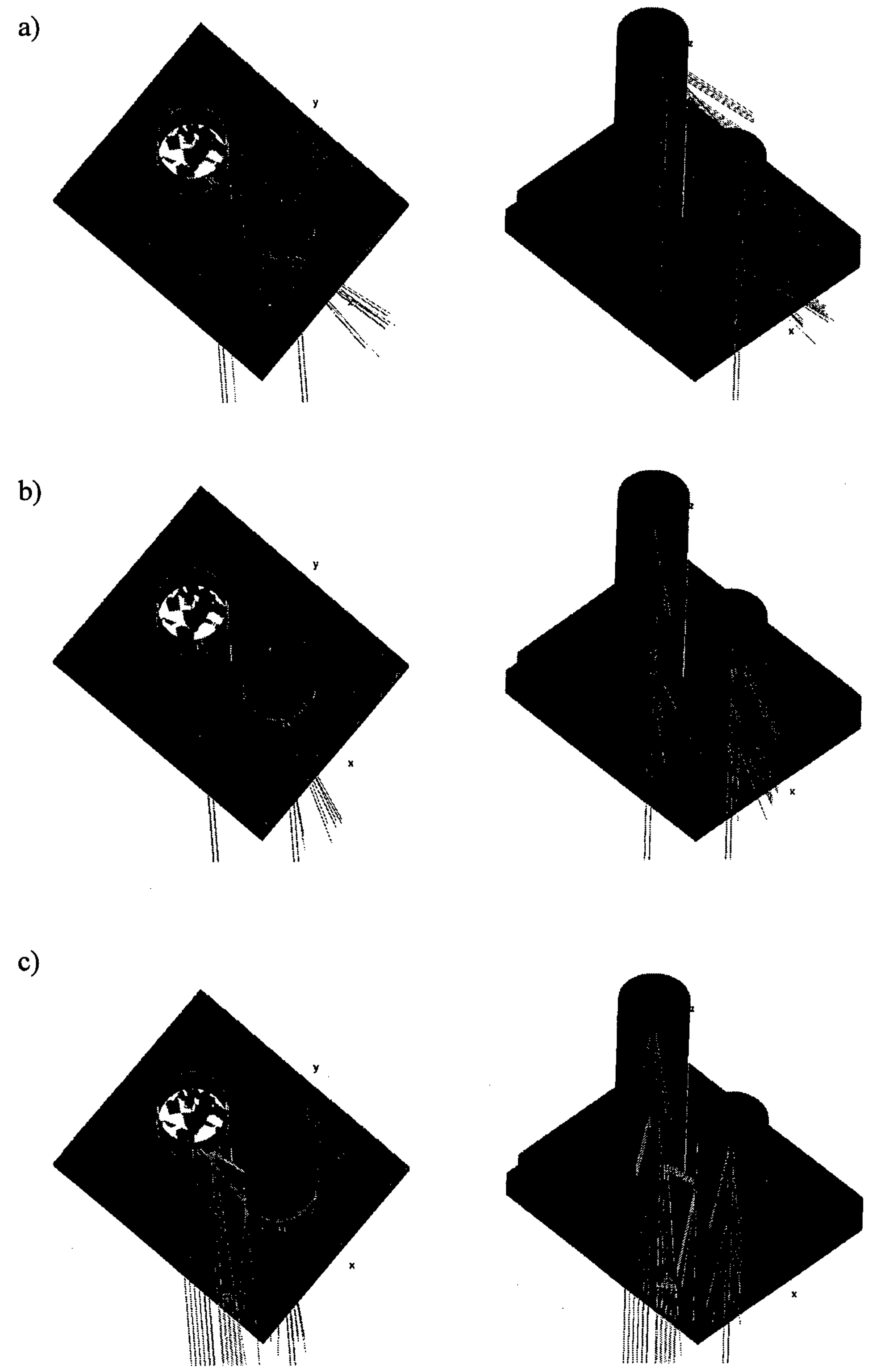


Figure 28a-c: Object C Ray Trace, 45 deg Tx illumination angle (α); Rx position in azimuth (β angle) at a) 0 deg, b) 30 deg, c) 45 degrees. Key: green = single bounce; blue = 2 bounces; orange = 3 or more bounces.

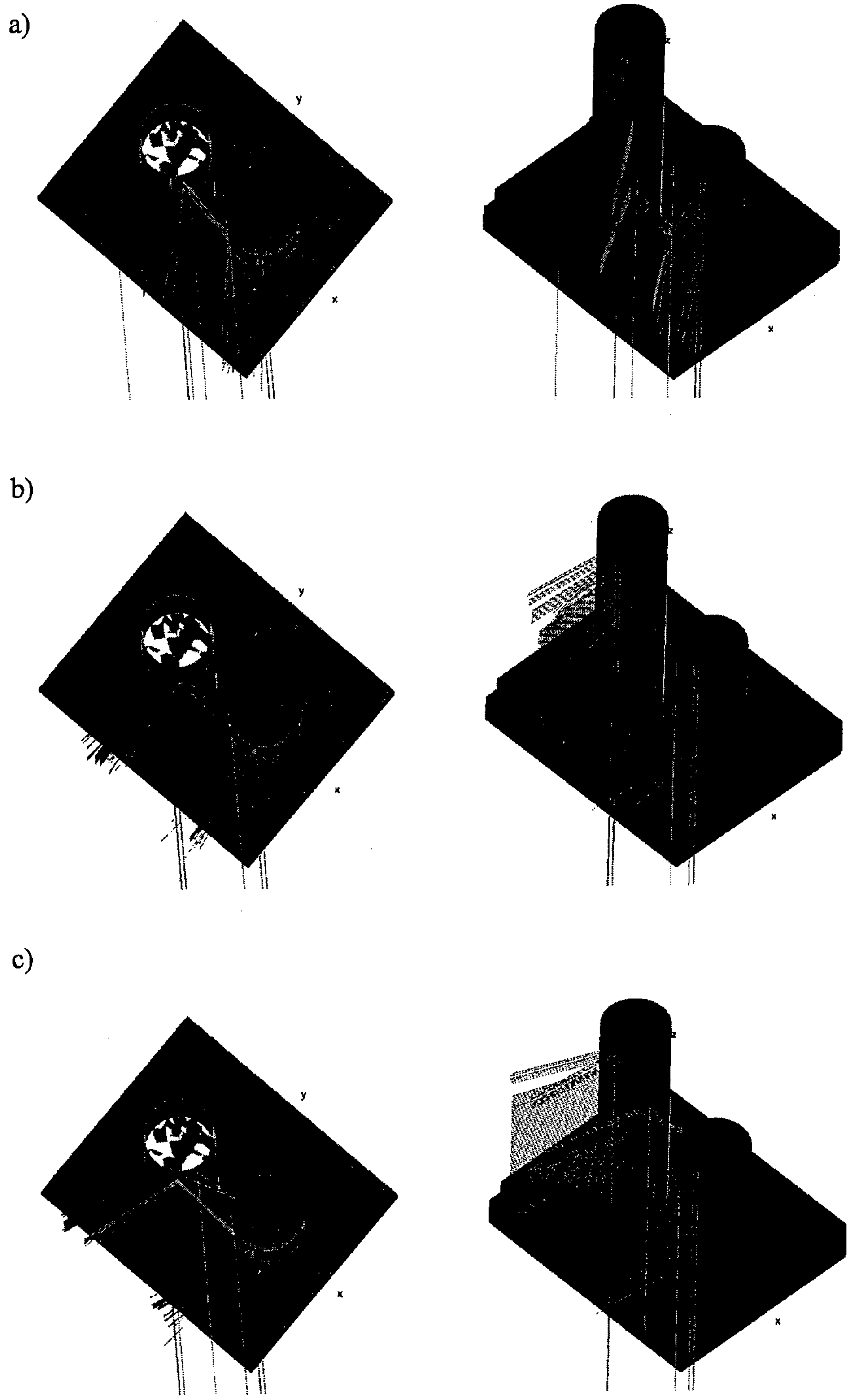


Figure 29a-c: Object C Ray Trace, 45 deg Tx illumination angle (α); Rx position in azimuth (β angle) at a) 60 deg, b) 90 deg, c) 105 degrees. Key: green = single bounce; blue = 2 bounces; orange = 3 or more bounces.

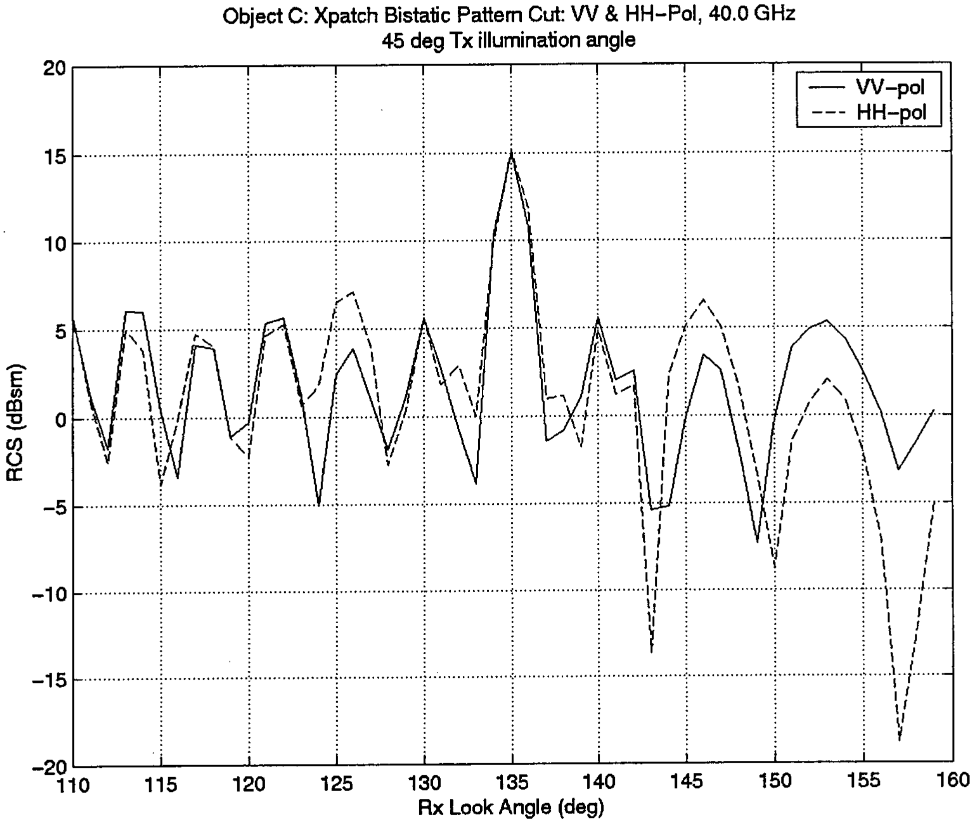


Figure 30: Object B Co-pol Bistatic Xpatch PTD Prediction at 40 GHz, β angle 110-160 degrees

The above discussion demonstrates that Xpatch 2.4d can predict reasonably accurate bistatic signatures from more complex geometries even if they don't meet the standard *electrically large* criteria. Such objects need to be comprised of surfaces which support several specular reflections with a wide angular distribution (i.e. cylinders). This analysis leads us to another conclusion regarding Xpatch. If one can infer through inspection that the dominant bistatic reflection mechanisms from a target are specular in nature, the Xpatch predictions should be fairly accurate (and conversely, if one cannot, then the Xpatch predictions will be less accurate). Lobing structure of the patterns may become skewed toward larger bistatic angles at higher frequencies, but lobe amplitude and null presence should be close to the real thing. One clearly distinctive characteristic of this type of object is a lack of significant shadowing features, which could mask specular effects.

To reinforce the conclusion, inspection of a target for which large specular reflections cannot be guaranteed from all α angles should be accomplished. Object B provides just such a geometry because of the large flat plate on one end. The first 110 degrees of Rx coverage are shown in Figures 31 and 32 (8 and 15 GHz). Clearly Xpatch does not produce nearly as adequate a prediction as before. Discrepancies are explained as follows.

The receiver is looking at the front of the flat plate from α = 0 to 45 degrees. In this region one expects to see mostly sidelobe activity associated with the plate's specular reflection which occurs near α = -45 deg (not shown). The lobe structure should be uniform with steadily increasing amplitude as one approaches smaller α angles, and this is exactly what is present. Xpatch predictions are consistent with measured data (albeit shifted, again) until close to the transmitter illumination angle, at which point they diverge. Between α = 30-40 deg. Xpatch is predicting higher than expected results for the lower RF. An exact cause is unknown, but inaccuracy of the PTD implementation is suspected. A similar situation existed for Object A's return near α = 70 deg. at lower RFs (see Figs 20-21).

Beyond α = 45 deg., the canted cylinder is illuminated completely but not the shorter cylinder as shown in the ray trace diagrams of Figs. 35-36. Therefore, only a single direct reflection analogous to one of Object C's speculars (namely from the canted cylinder) is now present. The nature of the remaining scattering pattern is changed accordingly. The lobing structure of Fig 31 suggests that once again two scatterers are beating against each other at VV-pol but probably not at HH-pol. This implies the presence of additional scattering mechanisms whose amplitudes are roughly the same as the speculars in this region. Some non-specular effects will be more noticeable at certain polarizations. The anticipated trend is that they will contribute to the measured signature more at lower frequencies, especially at HH-pol, and cause the Xpatch and measured data to be more dissimilar than for Object C.

A single specular reflection from the canted cylinder, diffraction from the plate edge, multi-bounce between cylinders, some specular or cavity effects from within the canted cylinder, and nominal surface wave effects are all suspected of competing throughout much of this region. At VV-pol the two dominant scatterers are most likely an edge diffraction from the flat plate and the specular from the canted cylinder. Because neither is as strong as the cylinder speculars noted on Object C, other non-specular effects may contribute more or less to the overall measured pattern and not be predicted by Xpatch's PO solution. The effect is witnessed in Xpatch's lower amplitude computations than the measured data even at larger bistatic angles (Figs 33-34). One contribution to the higher VV-pol measurement data which cannot be captured by Xpatch is the TM_{311} cavity mode supported by the cylinder at 8.0 GHz. The cavity does not completely support any other mode at 15 GHz, helping to explain the better correlation there. At HH-pol the plate edge diffraction does not exist, leaving other sources free to compete with the cylinder specular. Surface wave effects on the cylinders' would be most pronounced at this polarization, and because their effect diminishes with increasing frequency, one expects better correlation of the HH-pol data at 15 GHz. This is observed in the figures. Overall the presence of non-specular components whose amplitudes are on the same order as the one specular yield a greater dissociation between the measured and Xpatch data.

We can summarize the observation as another conclusion for Xpatch2.4d analysis as a whole. Namely, if upon inspection of an object one determines that a large shadowing feature or cavity exists, then bistatic Xpatch predictions may be lower than the true signature. If one determines that only a single direct specular reflection is likely to be present, especially those with low, broad angular distribution (i.e. from curved surfaces), additional non-specular or multi-bounce scattering mechanisms can arise and be of similar amplitude. They could compete with the specular in such a way as to make the Xpatch prediction even less accurate, especially for small electrically sized objects. At worst, Xpatch will be incapable of accurately predicting the bistatic RCS with any degree of certainty. Ray tracing can lend significant insight into determining the presence and origin of the specular to assist in the evaluation.

This conclusion isn't entirely unexpected. By it's very nature Xpatch should not perform well for smaller objects and lower frequencies. Low frequency techniques (e.g. method of moments) are better poised to handle predictions in this arena. However, because they can be computationally expensive and time consuming, some incorporate derivations of Kell's or Crispin's MBET as a speed enhancing option [32]. These two MBETs are to be investigated in the next section for their suitability for predicting the signatures of these same test objects.

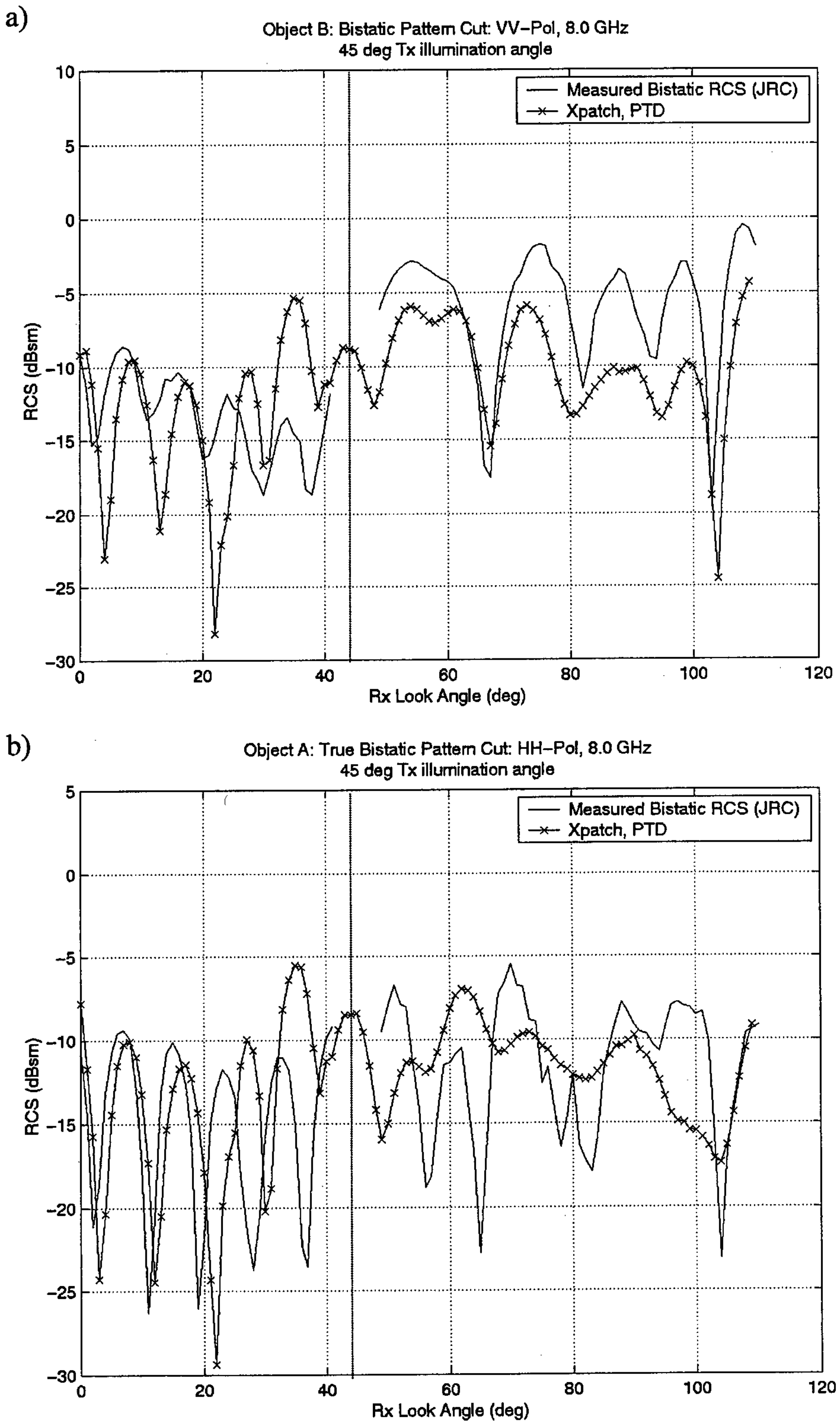


Figure 31a-b: Object B, Bistatic Xpatch Predictions (PTD) vs. Measured (JRC) Data, 8 GHz, 0-110 degrees Rx look angle
a) VV-pol, b) HH-pol

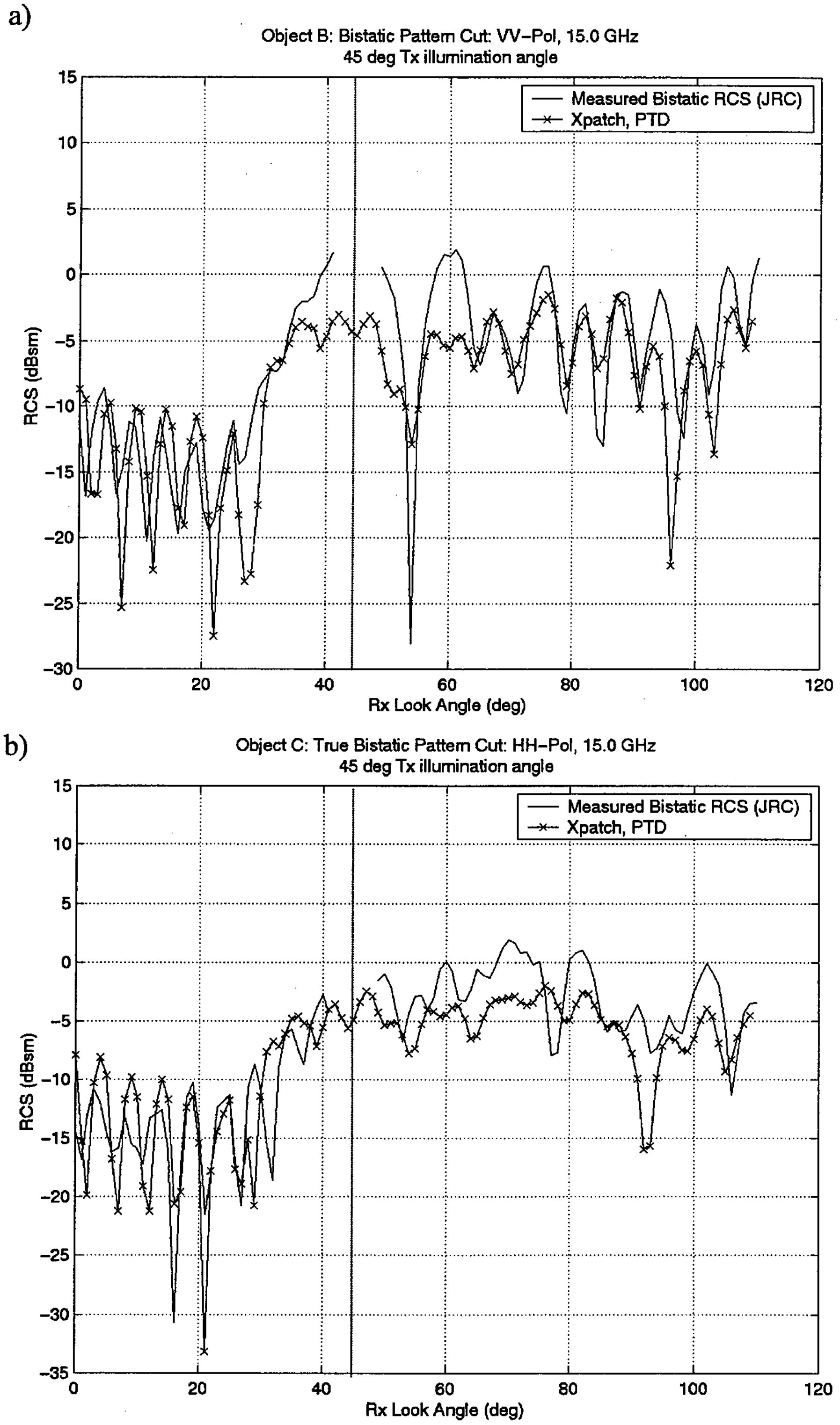


Figure 32a-b: Object B, Bistatic Xpatch Predictions (PTD) vs. Measured (JRC) Data, 15 GHz, 0-110 degrees Rx look angle a) VV-pol, b) HH-pol

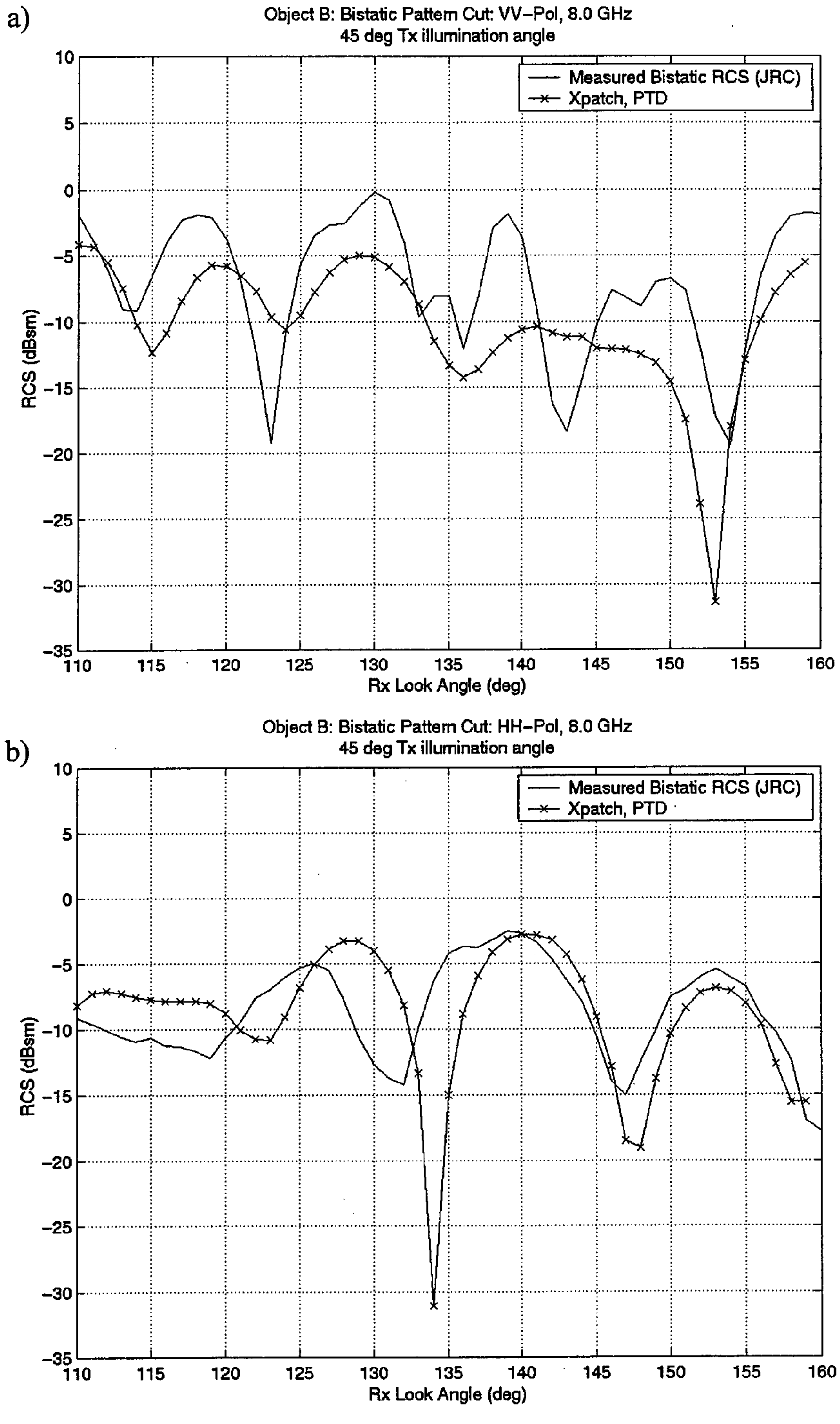


Figure 33a-b: Object B, Bistatic Xpatch Predictions (PTD) vs. Measured (JRC) Data, 8 GHz, 110-160 degrees Rx look angle a) VV-pol, b) HH-pol

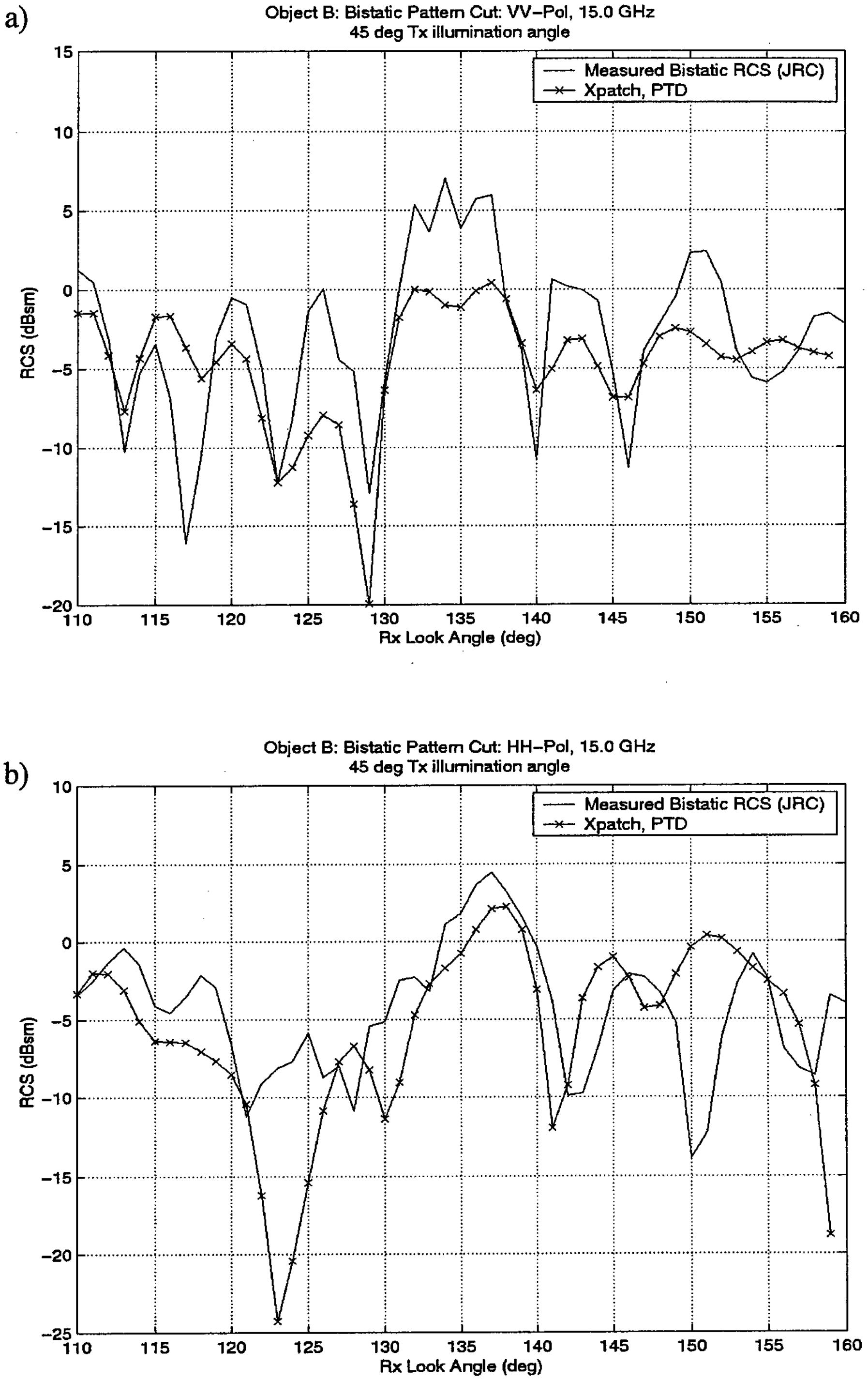


Figure 34a-b: Object B, Bistatic Xpatch Predictions (PTD) vs. Measured (JRC) Data, 15 GHz, 110-160 degrees Rx look angle a) VV-pol, b) HH-pol

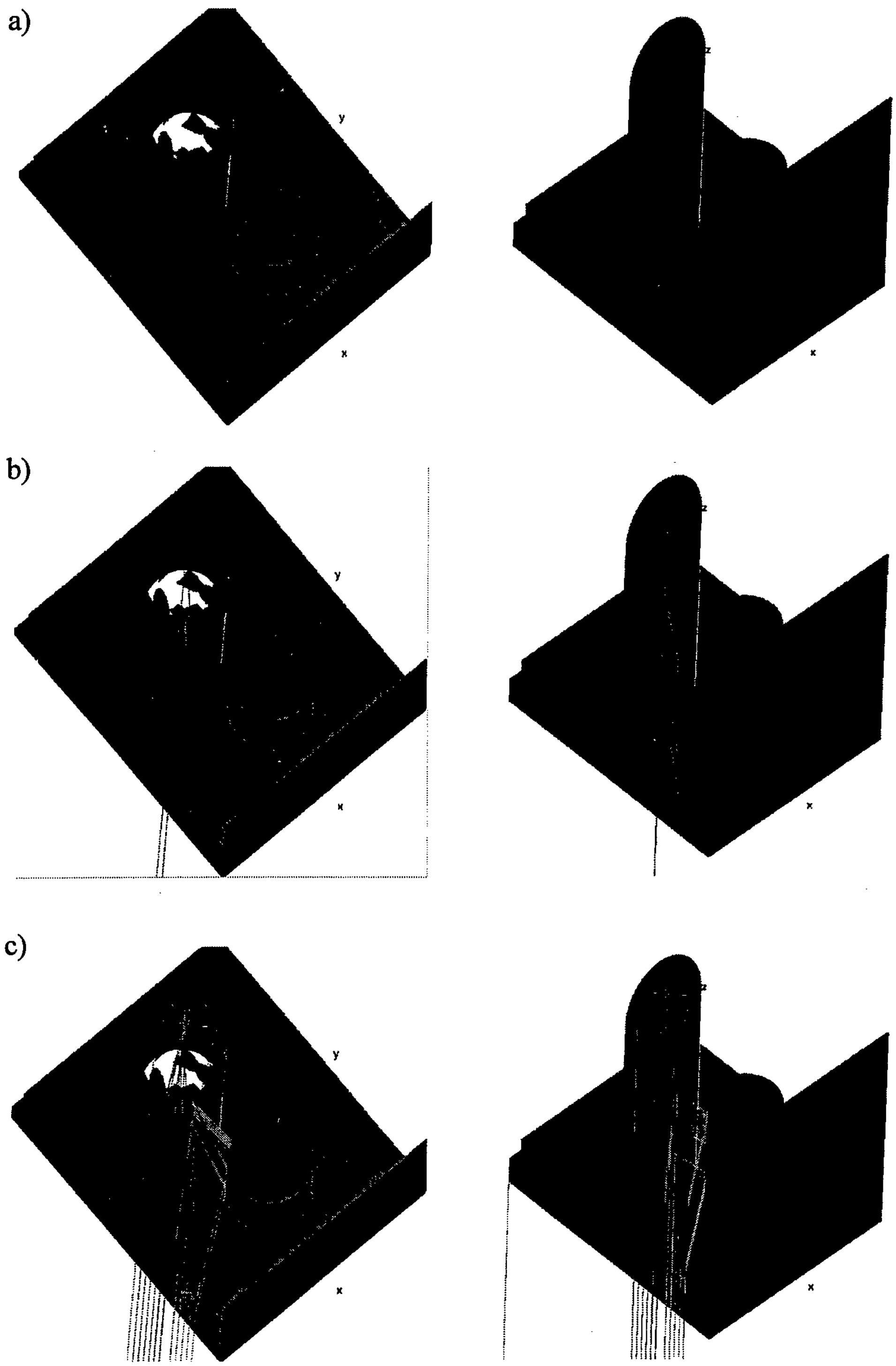


Figure 35a-c: Object B Ray Trace, 45 deg Tx illumination angle (α); Rx position in azimuth (β angle) at a) 15 deg, b) 30 deg, c) 45 degrees. Key: green = single bounce; blue = 2 bounces; orange = 3 or more bounces.

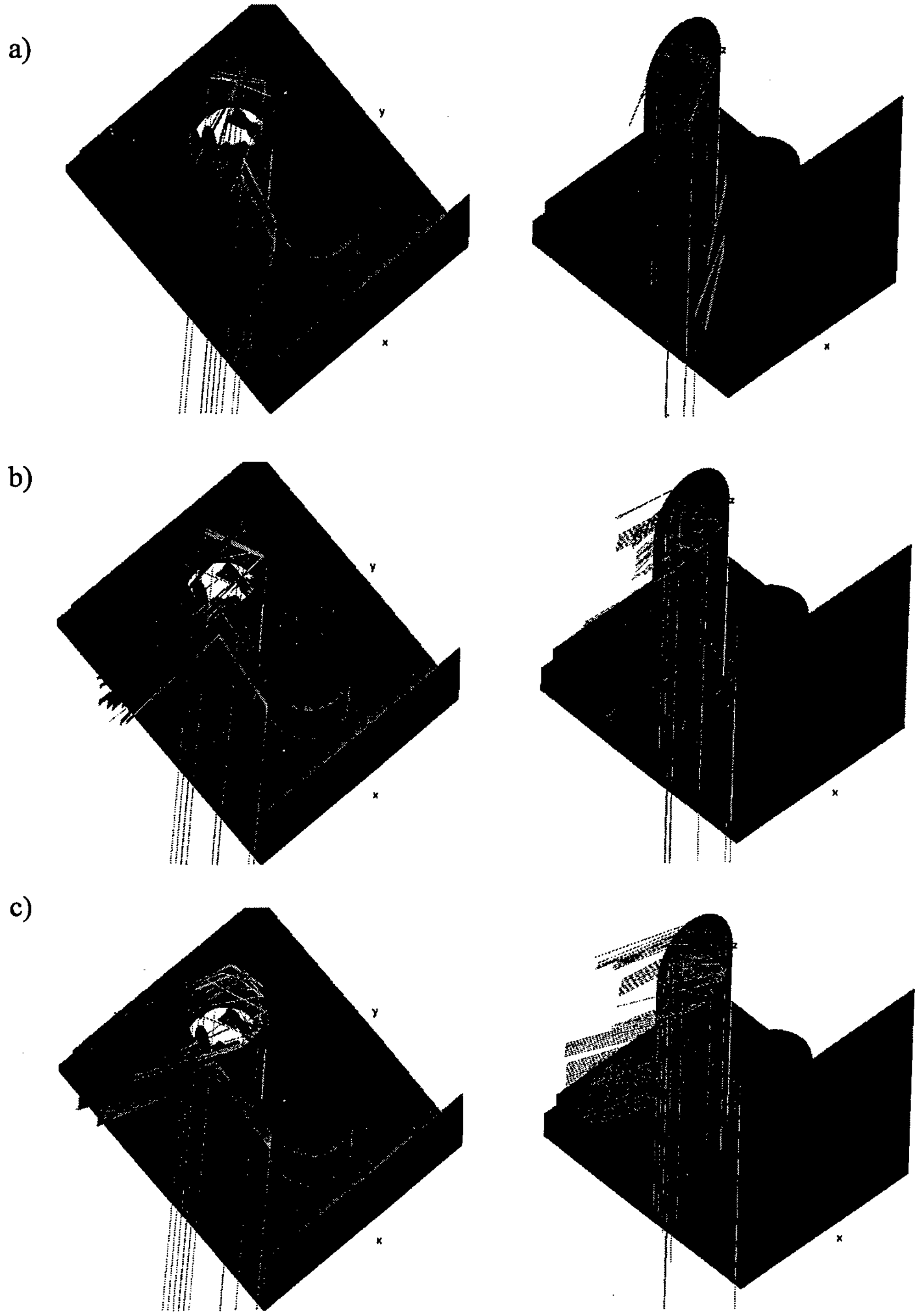


Figure 36a-c: Object C Ray Trace, 45 deg Tx illumination angle (α); Rx position in azimuth (β angle) at a) 60 deg, b) 90 deg, c) 105 degrees. Key: green = single bounce; blue = 2 bounces; orange = 3 or more bounces.

Kell's and Crispin's MBET:

As was mentioned in Chapter 2, one expects both monostatic-to-bistatic equivalence theorems to perform better for specular reflections, especially near the transmitter illumination angle. Like Xpatch, they are not expected to perform well in situations where shadowing effects, surface waves, or multi-bounce dominates, because the theory upon which each is based does not predict these phenomena well. They may predict diffraction effects fairly well due to their large angular distribution. However, the treatment of an object's signature as an average response over a limited angular or time extent may have been among the authors' original intentions [6,7]. This suggests another

Table 8: Mean and Standard Deviation of Difference Plot Amplitudes For Available MBET Data, $0 < \beta < 110$ degrees, Object A

MBET	Data (dBsm units)					
	No averaging					
	8 GHz			14 GHz		
	Polarization	Mean	Standard deviation	Polarization	Mean	Standard deviation
Kell's	VV-pol	-0.4520	2.1038	VV-pol	-1.2523	2.8487
	HH-pol	-0.7228	5.2145	HH-pol	-1.2550	5.2975
Crispin's	VV-pol	-0.7384	1.9067	VV-pol	-1.2810	2.0229
	HH-pol	-0.3987	4.2186	HH-pol	-1.1062	6.2325
	5 DEG WINDOW AVERAGE					
	8 GHz			14 GHz		
Kell's	VV-pol	-0.2887	1.3539	VV-pol	-0.4245	1.2473
	HH-pol	-0.2082	3.3084	HH-pol	0.3504	1.3983
Crispin's	VV-pol	-0.5713	1.4931	VV-pol	-1.0174	1.1400
	HH-pol	-0.084	3.1404	HH-pol	-0.2718	2.7822
	9 DEG WINDOW AVERAGE					
	8 GHz			14 GHz		
Kell's	VV-pol	-0.2590	0.8255	VV-pol	-0.1740	0.7957
	HH-pol	-0.2584	2.4867	HH-pol	0.6378	0.6378
Crispin's	VV-pol	-0.4452	1.2189	VV-pol	-0.8707	1.0487
	HH-pol	-0.0219	2.4815	HH-pol	-0.1355	2.1716

way of viewing the MBET data. Specifically, if the data were averaged in some way to better reflect an *average* signature, the MBET performance may improve. With this in mind, each signature is averaged with a 5 and 9 degree sliding window and the ensuing analysis accompanies that of the untreated returns.

Object A:

Object A is reviewed first for its dominant specular and distinguishable diffraction component. The measured and MBET bistatic signatures at 8 and 14 GHz are shown in Figs. 37a-c (8 GHz VV-pol), 38a-c (8 GHz HH-pol), 39a-c (14 GHz VV-pol), and 40a-c (14 GHz HH-pol). Figures 41-44 show the same data averaged with a 5 degree sliding window and in Figs. 45-49 apply a 9 degree sliding window is applied. Table 8 lists the mean and standard deviation for each difference plot to aid in the evaluation process. One immediately notes one of the primary limitations of Kell's derivation in the 14 GHz plots (Figs. 32,33,etc.); namely, that due to the frequency shift accompanying the increase in bistatic angle, the workable data set for use with Kell's approximation must be larger than that used for Crispin's. Kell's MBET for an equivalent bistatic frequency of 14 GHz has a limited angular extent of 41 degrees because the next higher angle's data corresponds to a measurement in the quasi-monostatic data set above 15 GHz.

Inspection of Object A's measured and MBET bistatic signatures reveals good correlation through approximately 30 degrees Rx look angle for both polarizations. The difference plots are somewhat misleading in their portrayal of the correlation for the first 30 degrees of azimuth. The specular peaks predicted by both MBETs at 0 deg is close to the PO prediction (16.1 and 21.0 dBsm) and the lobing structure through approximately 30 degrees azimuth is a close match, although the MBET predicted sidelobes seem to be slightly skewed toward the peak specular. The quasi-monostatic nature of the monostatic data set may account for some of the shift. Limitations of the MBETs probably become more significant at larger β angles. The net effect is captured and exaggerated in the difference plots as a wildly fluctuating amplitude for β angles between 40 and 70 degrees. But notice that the shifting seems to diminish with increasing frequency, and therefore, Figs. 39 & 40 difference plots are more well behaved. Beyond 30 degrees both approximation appears equally poor at predicting the position or amplitude of the sidelobes. As the Rx nears 90 degrees (edge-on), diffraction effects should be visible for VV-pol but not HH-pol. Both MBETs predict the edge diffraction at VV-pol lower than measured, with Crispin's being slightly worse. For HH-pol both MBETs predict amplitudes too high (by approximately 10 dB) at 90 degrees, missing the null entirely. This is understandable considering where the MBET data originates. The MBET data plotted in Figs. 36a-39a near 90 degrees is actually being generated from monostatic data near 45 degrees (monostatic data along the bisector). Inspection of Figure B-1 (Appendix B) drives the point home.

Table 8 suggests that the MBET predictions tend to be lower overall than actual levels and that averaging tends to increase correlation with increasing window size for the entire region of interest. Caution should be exercised when interpreting these results. The shifting of the MBET lobing structure and the averaging process itself could influence the eventual mean and standard deviation significantly even though a subjective evaluation of the data reveals an improved correlation. For this reason the absolute values in Table 8 are deemed unimportant, but the trend between analogous values for different averaging conditions is worthy of discussion. For the entire 110 deg. region (in which data exists), the mean difference and standard deviation decrease for most situations as more data is averaged (increasing window size). Overall this trend appears correct, but more insight may be gathered by looking at amplitude behavior within certain regions.

Averaging the data improves correlation for certain regions of the plot. The net effect of averaging this type of data is to reduce peak amplitudes of each lobe and fill-in the nulls. This increases the correlation for the first 30 degrees of azimuth, except for the large spike now present near the 4 degree position in some plots. This spike is an artifact of the averaging process. The averaging window slides over the data points in the data vector, averaging the number of points specified by the window width, centered on the widow's middle value. Because the averaging window doesn't slide over an entire window's width of data points for the first few elements in the data vector, there aren't as many data points to average. Consequently the first few *averaged* data points aren't computed from the same number of elements. Averaging does not appear to improve correlation near 90 degrees significantly for either frequency or polarization. Notice that as the data is averaged Kell's and Crispin's MBET solutions become more similar for similar polarizations and frequencies.

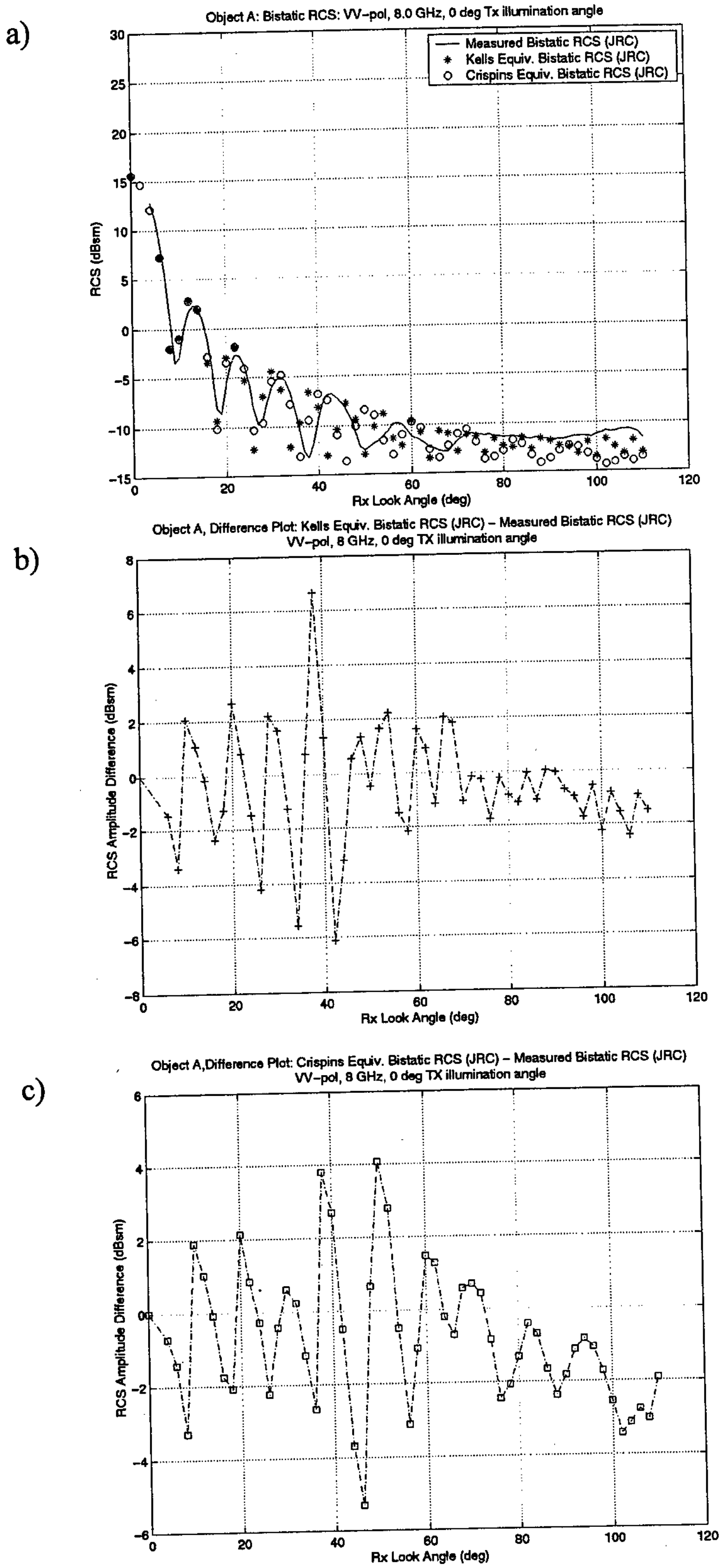


Figure 37a-c: Object A, VV-pol, 8 GHz

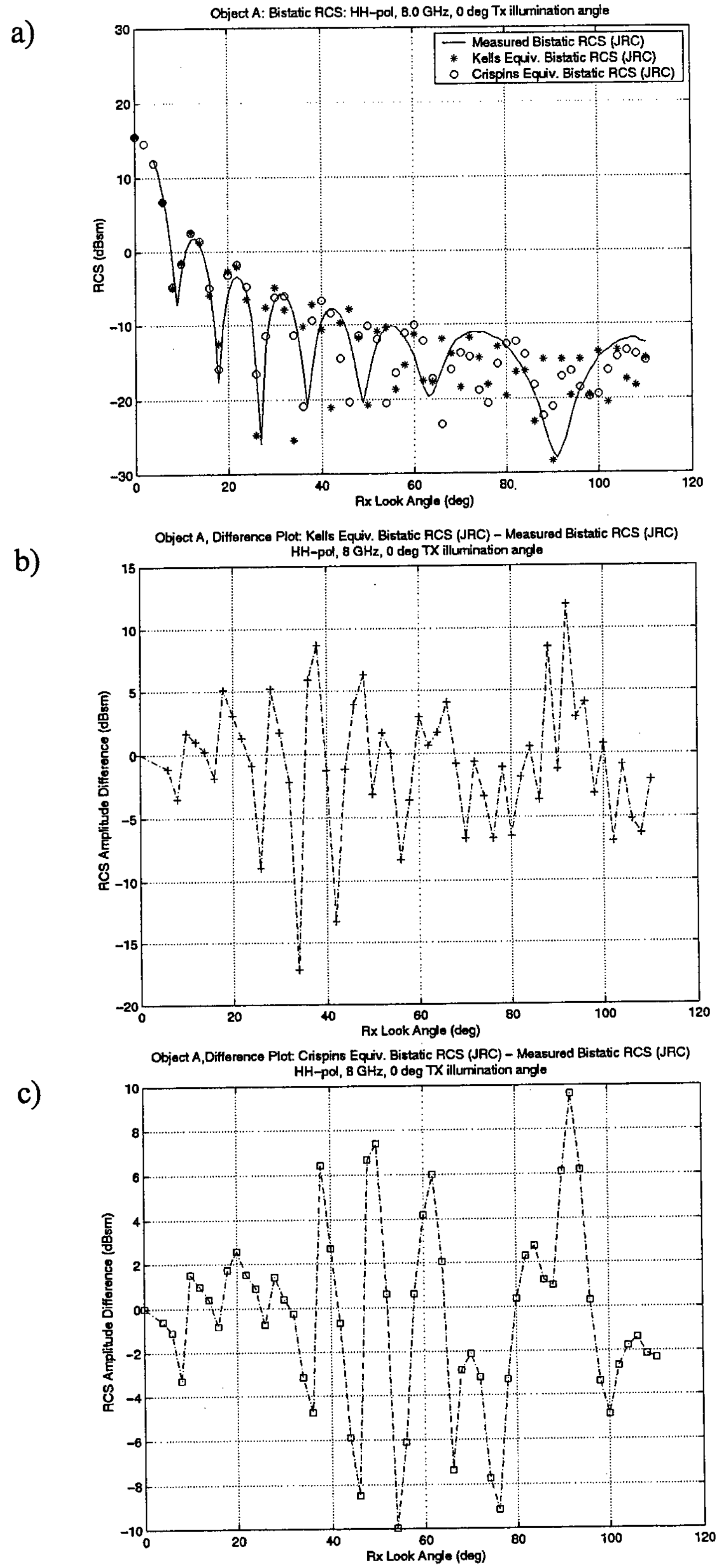


Figure 38a-c: Object A, HH-pol 8 GHz

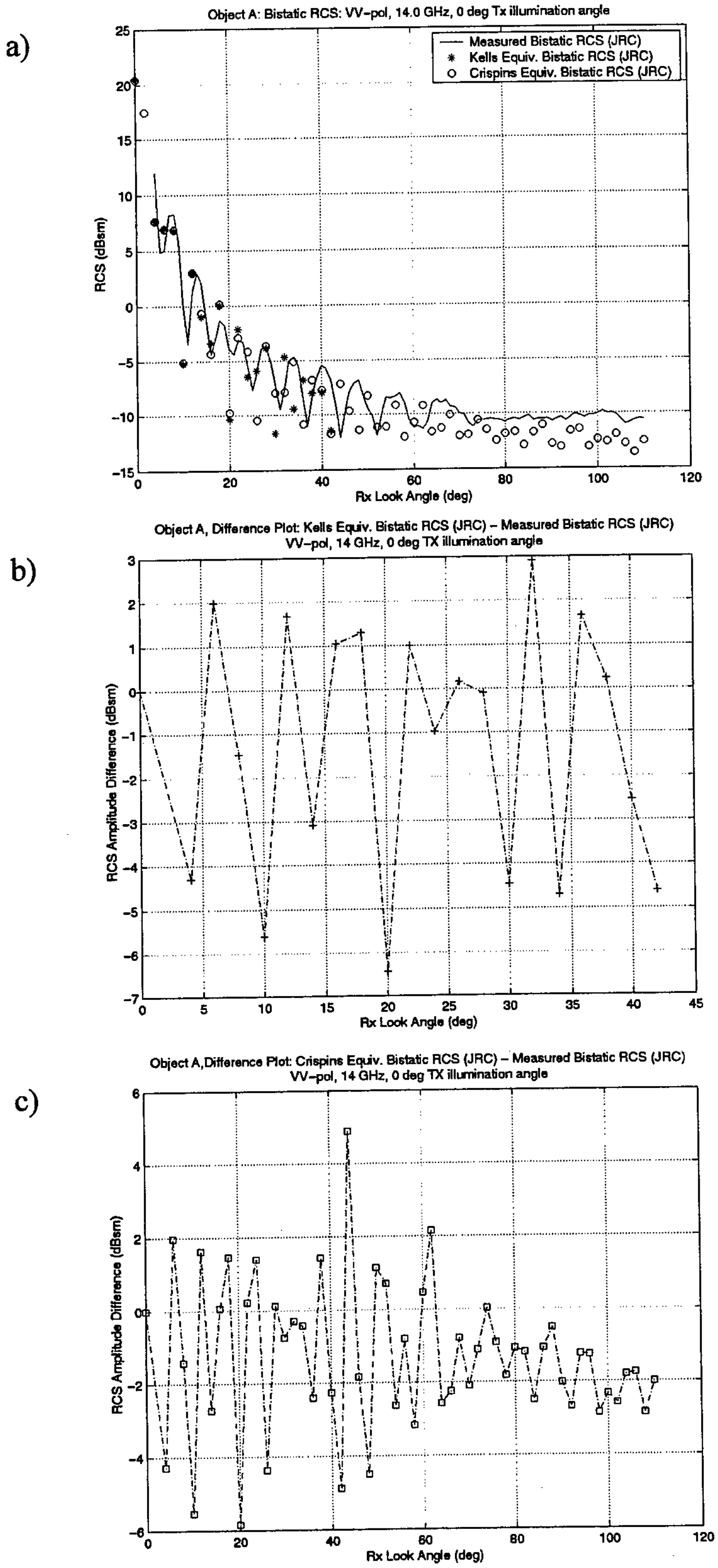


Figure 39a-c: Object A, VV-pol, 14 GHz

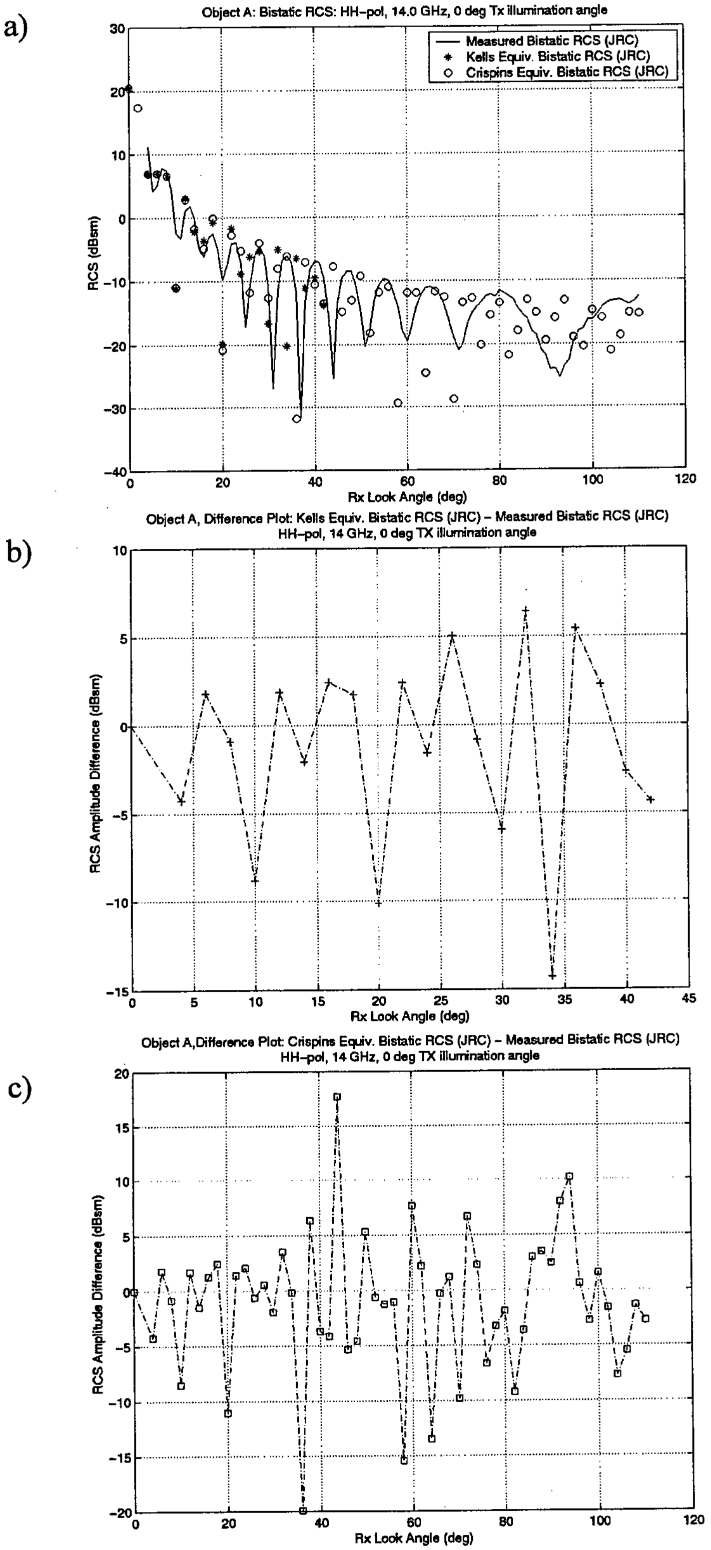


Figure 40a-c: Object A, HH-pol, 14 GHz

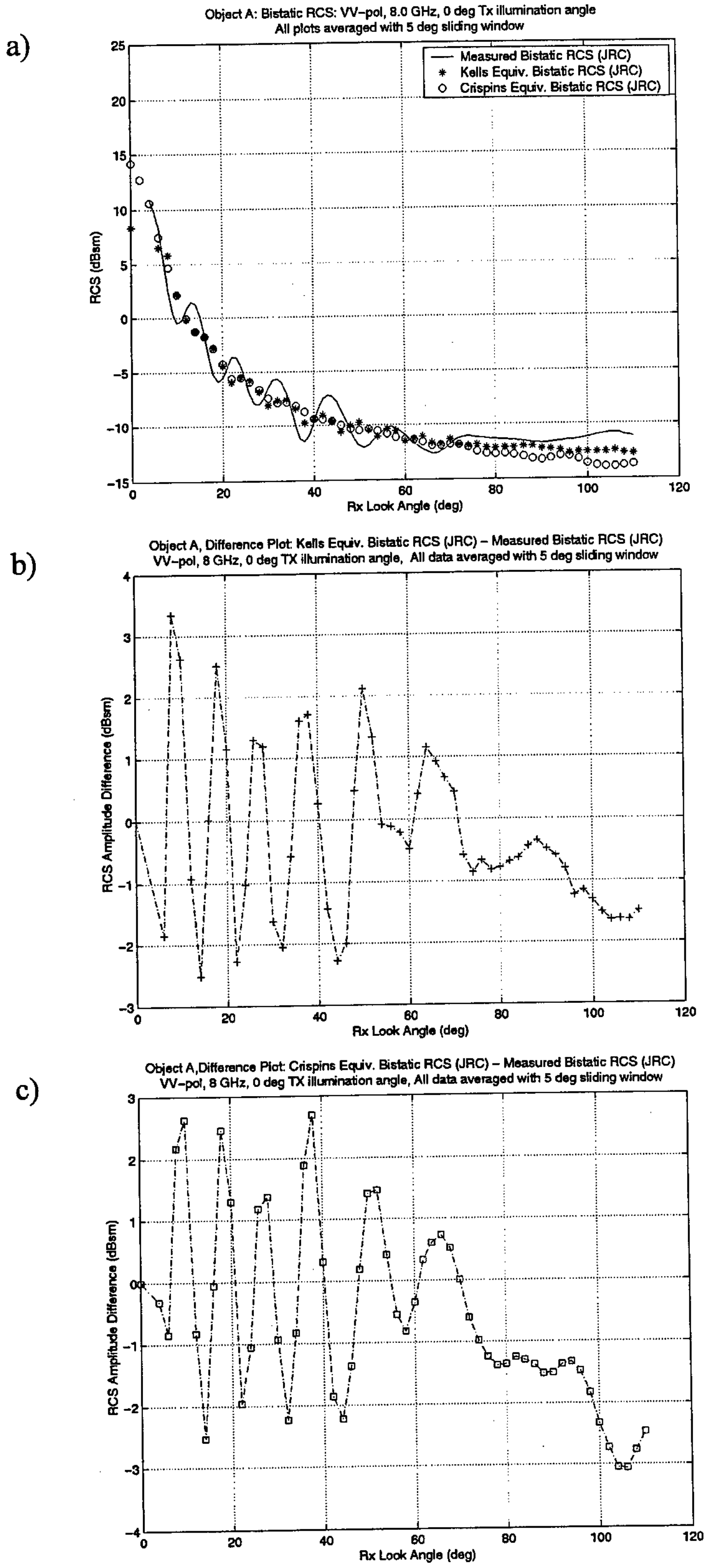


Figure 41a-c: Object A, VV-pol, 8 GHz, Data averaged with 5 degree sliding window

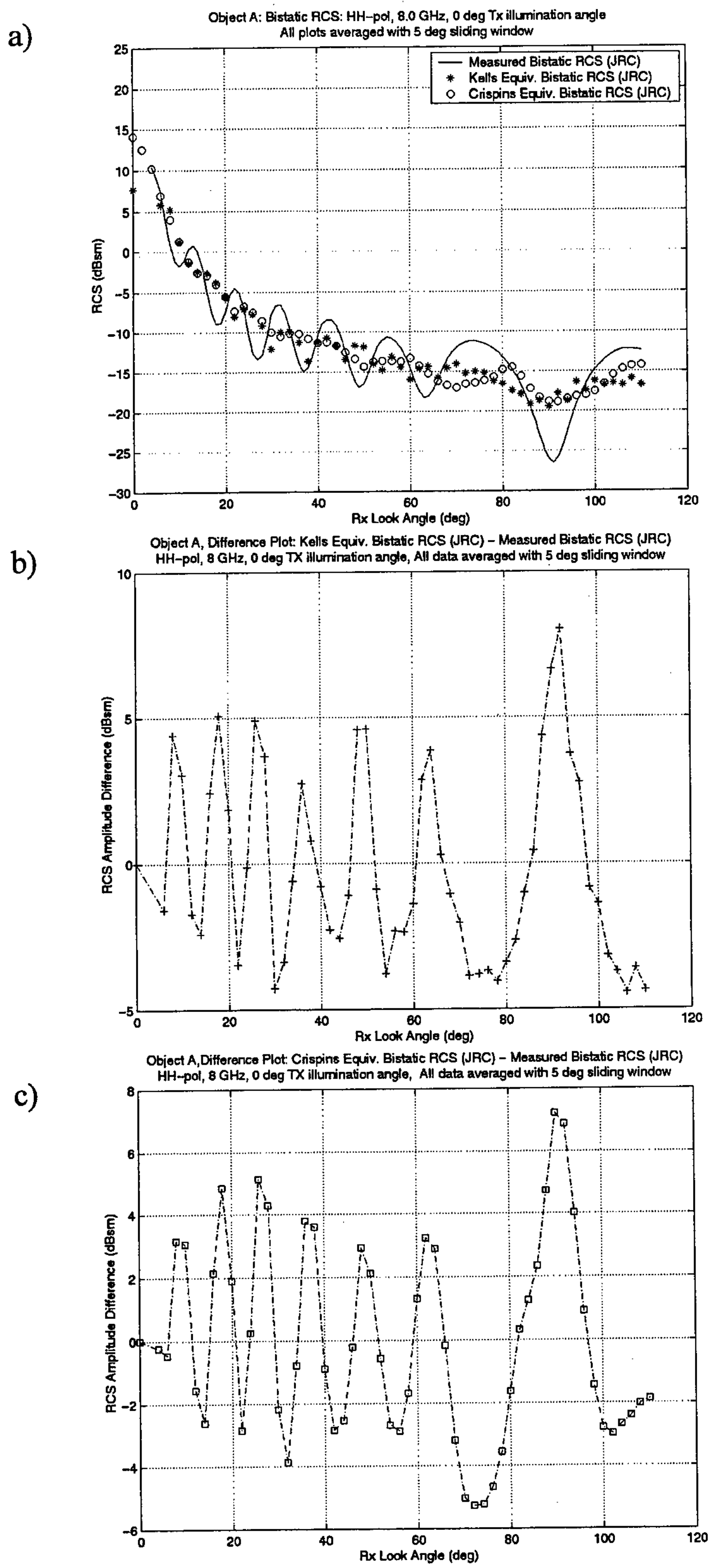


Figure 42a-c: Object A, HH-pol, 8 GHz, Data averaged with 5 degree sliding window

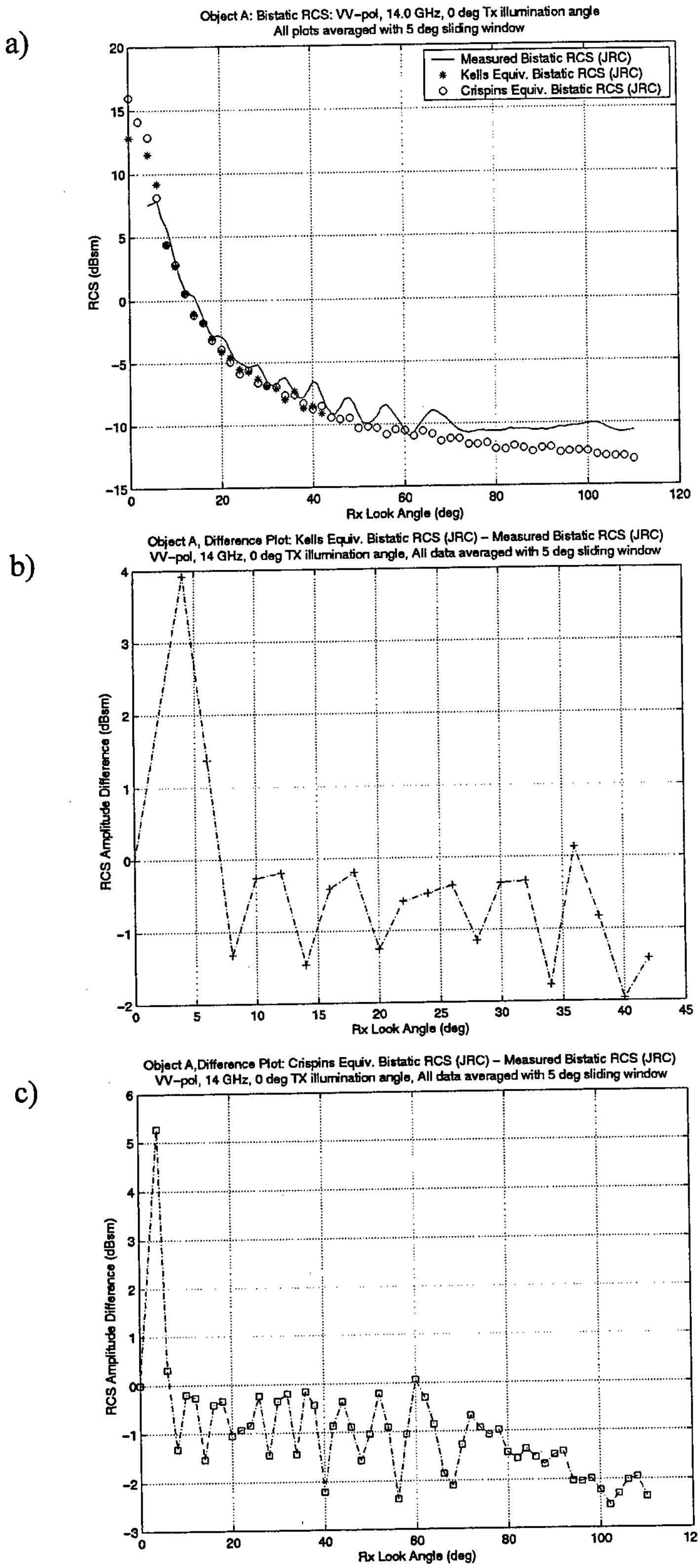


Figure 43a-c: Object A, VV-pol, 14 GHz, Data averaged with 5 degree sliding window

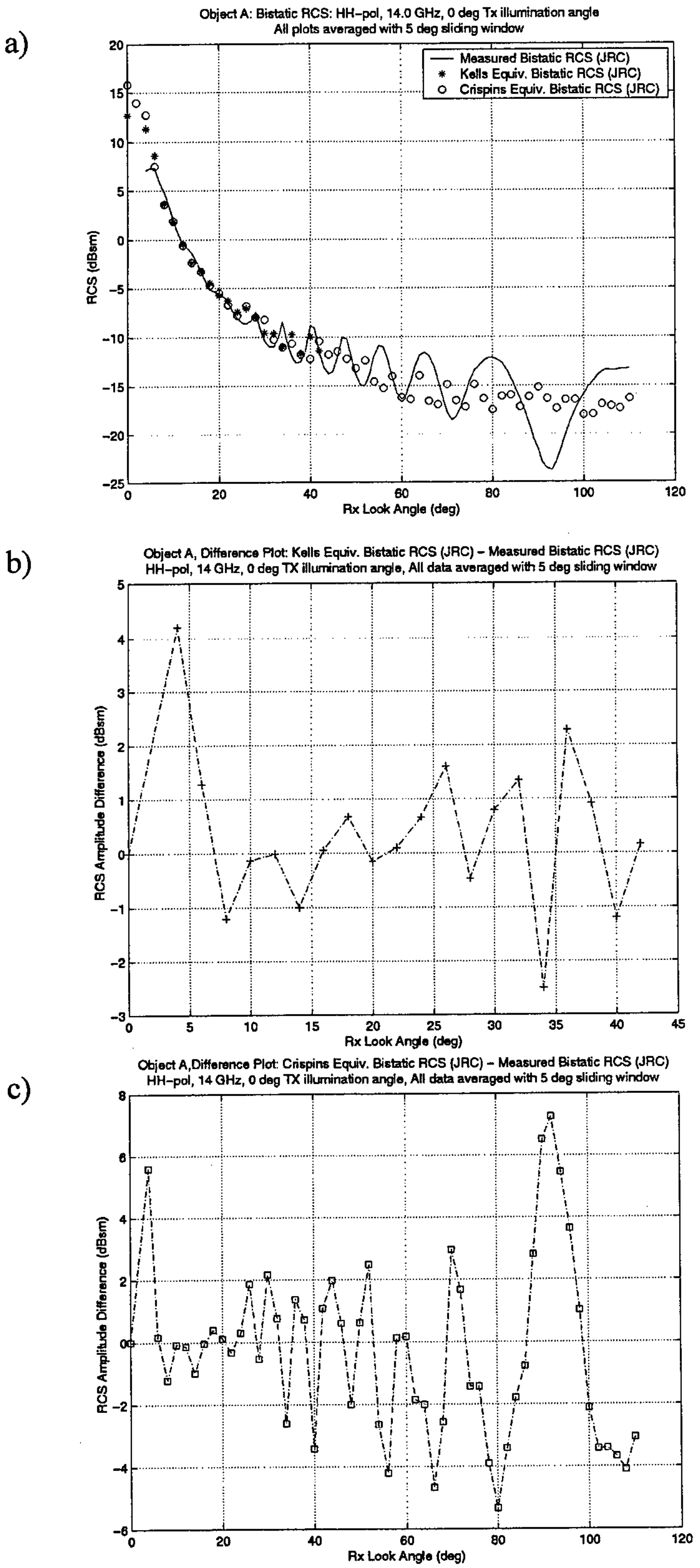


Figure 44a-c: Object A, HH-pol, 14 GHz, Data averaged with 5 degree sliding window

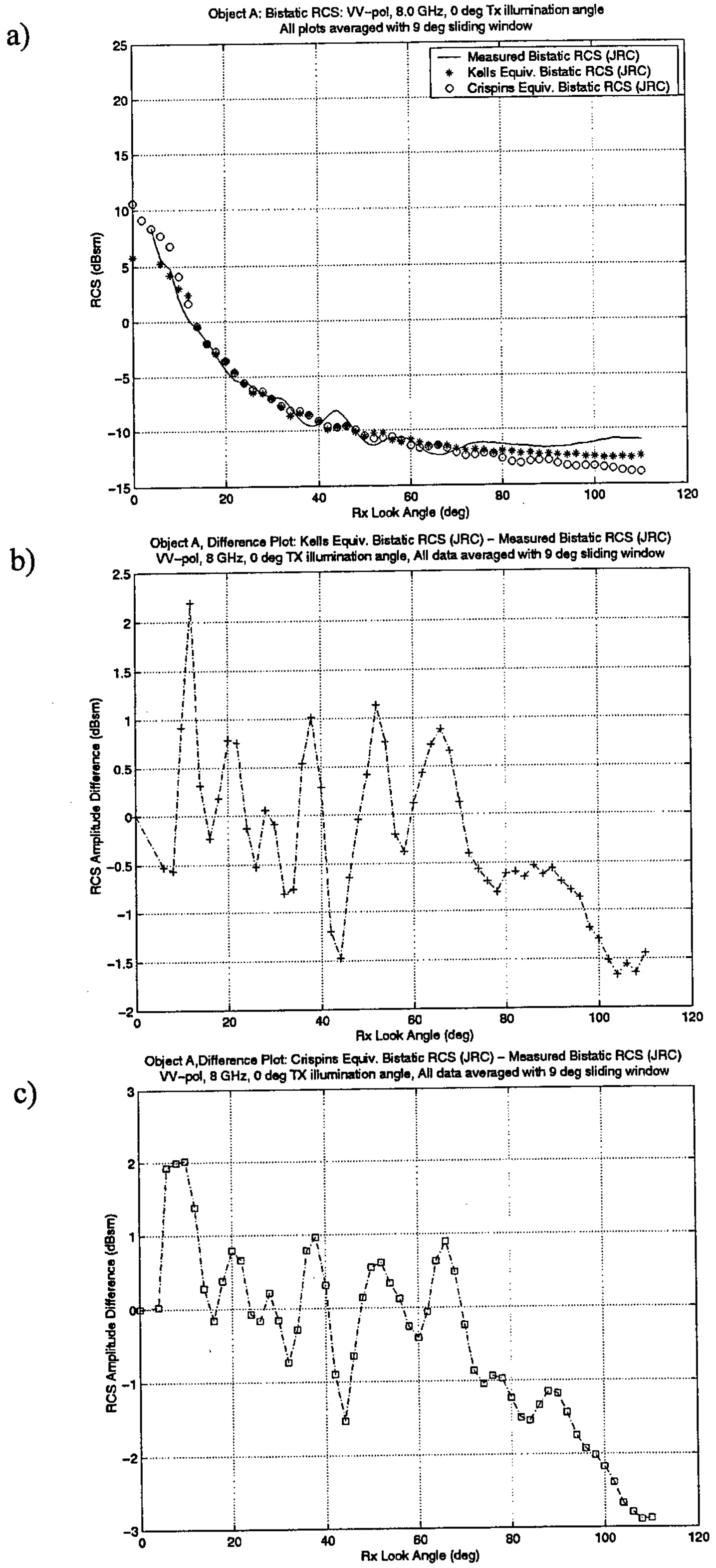


Figure 45a-c: Object A, VV-pol, 8 GHz, Data averaged with 9 degree sliding window

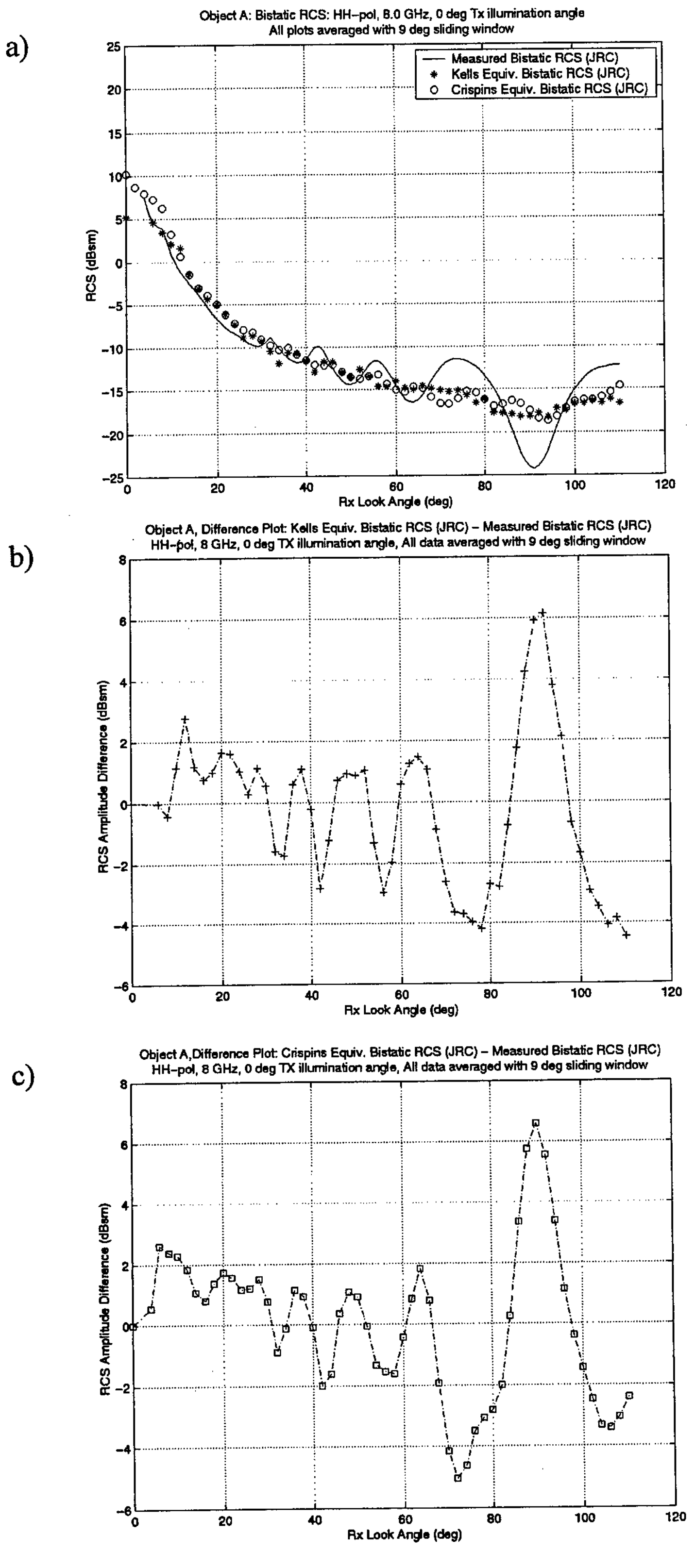


Figure 46a-c: Object A, HH-pol, 8 GHz, Data averaged with 9 degree sliding window

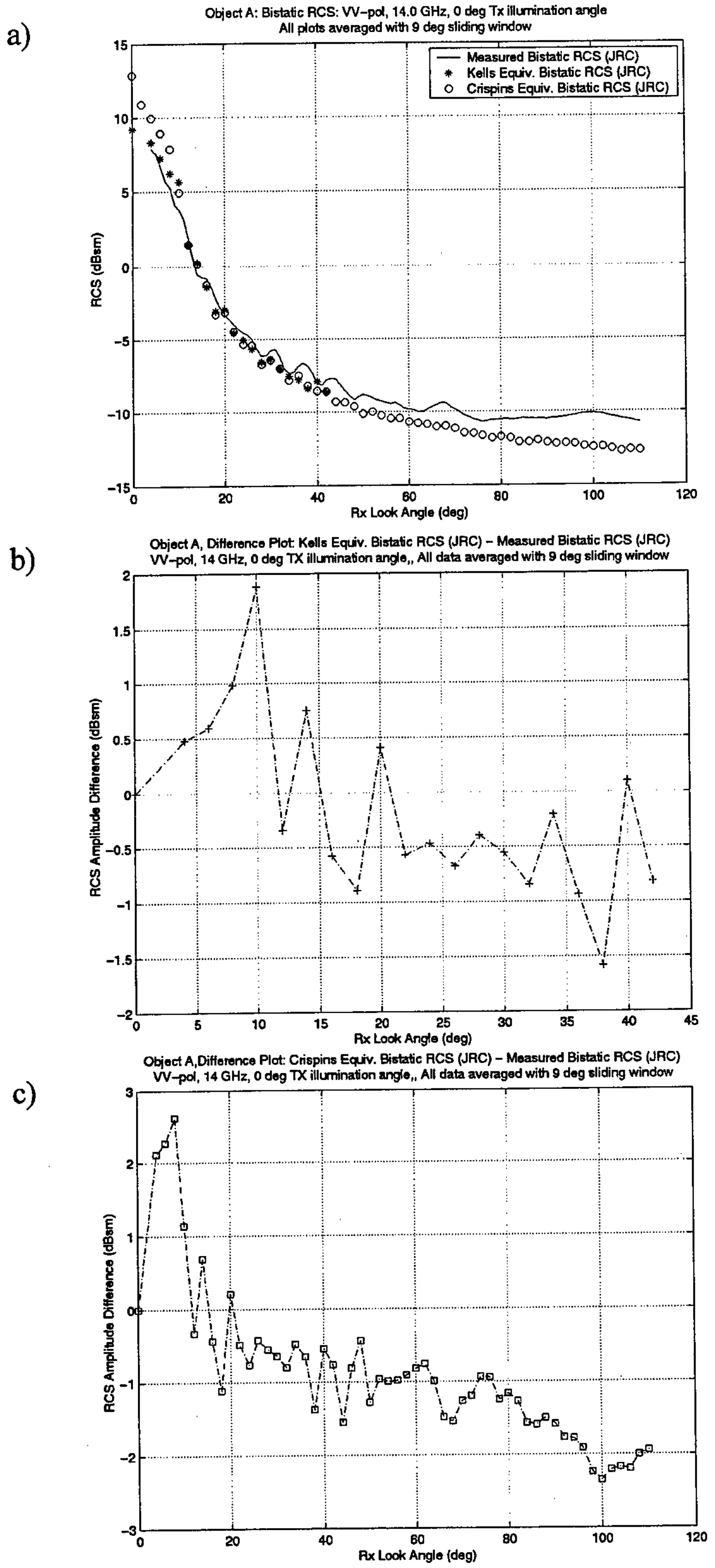


Figure 47a-c: Object A, VV-pol, 14 GHz, Data averaged with 9 degree sliding window

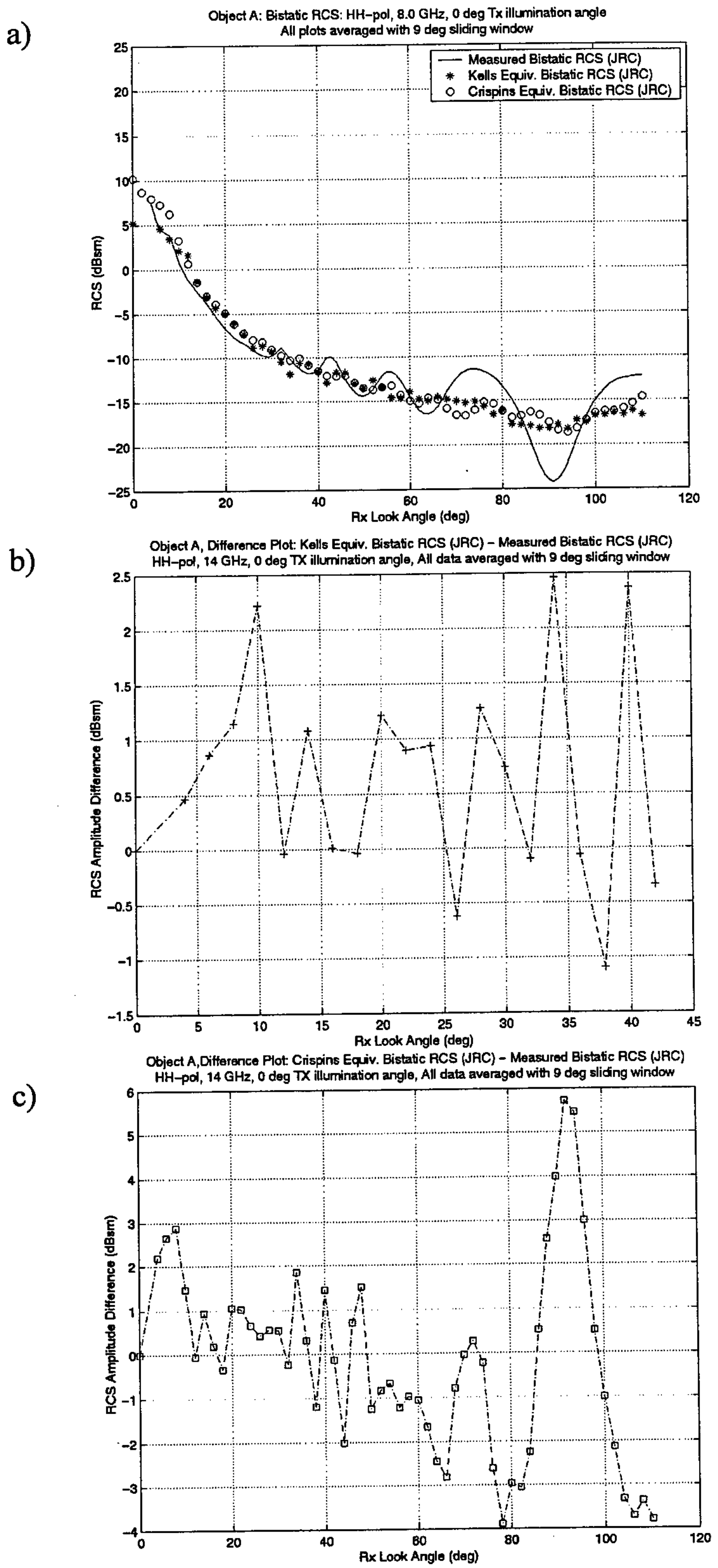


Figure 48a-c: Object A, HH-pol, 14 GHz, Data averaged with 9 degree sliding window

Object B & C:

Objects B & C are analyzed in tandem to differentiate between the specular and non-specular interactions dominating each scattered signature. The analysis often refers to the preceding Xpatch investigation to help explain and quantify MBET performance. The same scattering mechanisms discovered there apply here. For convenience, the transmitter illumination angle, α, is also marked with a dashed vertical line on the comparative pattern cut plots.

Figures 55-58 show Object B's MBET predicted patterns at 8 & 12 GHz for $0 < \beta < 110$ degrees, Figs. 59-62 are the same patterns with a 5 degree average applied, and Figs. 63-66 show the data with a 9 degree average applied. The higher RF of 12 GHz is chosen to allow Kell's MBET to predict a larger angular region than was accomplished for Object A. Analogous data for Object C are shown in Figures 49-54 at 8 & 15 GHz, but only for VV-pol. The MRC monostatic data set is the source of Object C MBET predictions, so a higher RF can be accommodated at the cost of the second polarization. Table 9 & 10 list the mean and standard deviation for each difference plot for Objects C and B respectively.

Because the MBETs are derived from principles similar to Xpatch's, there should be some similarity in the data analysis. Similarities should be confined to smaller bistatic angles, as the MBET formulas are only appropriate here. If the logic follows, the MBET predictions for this region will be slightly higher than measured and the correlation should improve if the data is averaged. One also expects that the correlation between MBET and measured data is greater for Object C than Object B, as was previously shown. Again, Object C is reviewed first.

Table 9: Mean and Standard Deviation of Difference Plot Amplitudes For Available MBET Data, 0 < β < 110 degrees, Object C

MBET	Data (dBsm units)					
	No averaging					
	8 GHz			15 GHz		
	Polarization	Mean	Standard deviation	Polarization	Mean	Standard deviation
Kell's	VV-pol	0.8874	9.2136	VV-pol	-1.5727	5.0074
Crispin's	VV-pol	-0.567	8.7628	VV-pol	-2.0441	5.8282
	5 DEG WINDOW AVERAGE					
	8 GHz			15 GHz		
Kell's	VV-pol	0.8183	7.9199	VV-pol	-1.5165	3.6837
Crispin's	VV-pol	-0.0359	7.4203	VV-pol	-2.1352	5.4003
	9 DEG WINDOW AVERAGE					
	8 GHz			12 GHz		
Kell's	VV-pol	0.8855	6.7135	VV-pol	-1.3567	2.5098
Crispin's	VV-pol	0.0604	6.3712	VV-pol	-1.7037	3.4829

In Figs 49-54, the receiver collects scattering from two primary specular sources, the cylinder bodies for 0 < β < 110 deg. region. The MBET data exhibits good correlation to the measured, especially near the Tx illumination angle as expected. Higher RF predictions are slightly low here (as was witnessed in the Xpatch predictions), and this lower trend characterizes both MBETs at larger β angles. In Table 9, again the absolute values of the mean and standard deviation are ignored, but the averaging trend is studied. Averaging the data seems to have less of an effect than it did for Object A. It seems that for complex objects whose bistatic signature is dominated by two wide-angle specular interactions, both MBETs can predict reasonably accurate RCS at lower frequencies where the specular lobe widths are wider. The extent of this capability is limited to bistatic angles of less than 15-20 degrees. As the frequency increases, the lobe widths become narrower and the MBETs begin to fail, tending to predict lower RCS than should

Table 10: Mean and Standard Deviation of Difference Plot Amplitudes For Available MBET Data, 0 < β < 110 degrees, Object B

MBET	Data (dBsm units)					
	No averaging					
	8 GHz			12 GHz		
	Polarization	Mean	Standard deviation	Polarization	Mean	Standard deviation
Kell's	VV-pol	1.8343	5.4581	VV-pol	0.3325	6.8510
	HH-pol	3.0669	5.4992	HH-pol	0.2441	5.6675
Crispin's	VV-pol	1.3639	6.4258	VV-pol	0.5953	6.2132
	HH-pol	1.9024	5.3804	HH-pol	2.3513	7.7804
	5 DEG WINDOW AVERAGE					
	8 GHz			12 GHz		
Kell's	VV-pol	1.7648	4.1543	VV-pol	0.6208	4.1443
	HH-pol	2.9753	4.0958	HH-pol	2.3818	3.7144
Crispin's	VV-pol	1.4370	4.2663	VV-pol	0.4485	5.5401
	HH-pol	1.9894	3.5521	HH-pol	1.9383	6.4762
	9 DEG WINDOW AVERAGE					
	8 GHz			12 GHz		
Kell's	VV-pol	1.6850	3.2285	VV-pol	0.7658	2.7382
	HH-pol	2.9874	2.6570	HH-pol	2.4854	3.0734
Crispin's	VV-pol	1.4909	2.6556	VV-pol	0.9368	3.0818
	HH-pol	1.9617	1.8444	HH-pol	2.5387	3.2832

be expected. The bistatic angles for which the approximations work is correspondingly narrowed. The nature of the scattering centers as perceived from a monostatic perspective also changes as a function of the bistatic angle. As the bistatic angle increases, the disassociation between monostatic and bistatic scattering centers becomes more apparent, leading to greater discontinuity between measured data and MBET predictions. Averaging the data provides minimal improvement to the correlation between the measured and MBET predicted data sets.

In Figs 55-58, the receiver is looking at the flat plate of Object B for the first 45 degrees and at the cylinder bodies for larger β angles. One expects correlation between the MBET predictions and the measured data to be less than that witnessed for Object C for many of the same reasons given in the Xpatch analysis. This is in fact the case. Discrepancies between MBET and measured data increase as the β angle (starting at 0) approaches the Tx illumination angle for VV-pol. Both MBET predicted lobing structures near 0 deg seem shifted, just as they did for Object A's sidelobe structure, which accounts for the rapid fluctuation in the difference plots. As β approaches α, the MBETs predict high, just as expected. The HH-pol patterns correlate much better than Xpatch predictions, however, for small bistatic angles. This is probably due to the fact that the MBET plots are pulled from measured monostatic data. Lower order effects, which could not be computed by Xpatch, are present in the measured data and are incorporated into the MBET bistatic prediction. Logically, the higher RF plots should demonstrate greater correlation as some of the non-specular effects become smaller. Figure. 57-58 show this to be true.

However, the useful range of both MBETs is limited to bistatic angles of approximately 10 degrees for VV-pol and 15 degrees at HH-pol for the higher RF. The large shadowing geometry produces even greater changes in the nature of each scattering center from monostatic and bistatic perspectives than is evident for Object C. In other words, the monostatic data (from which the MBETs are computed) arise from scattering centers with even less similarity to the true bistatic scattering centers, which are produced by the single specular and non-specular interactions. Table 10 again suggests that averaging the data does not improve the MBET correlation. In fact it may diminish the correlation in certain situations. One can conclude that whenever non-specular effects contribute markedly to the overall signature, averaging techniques should not be used to improve MBET performance.

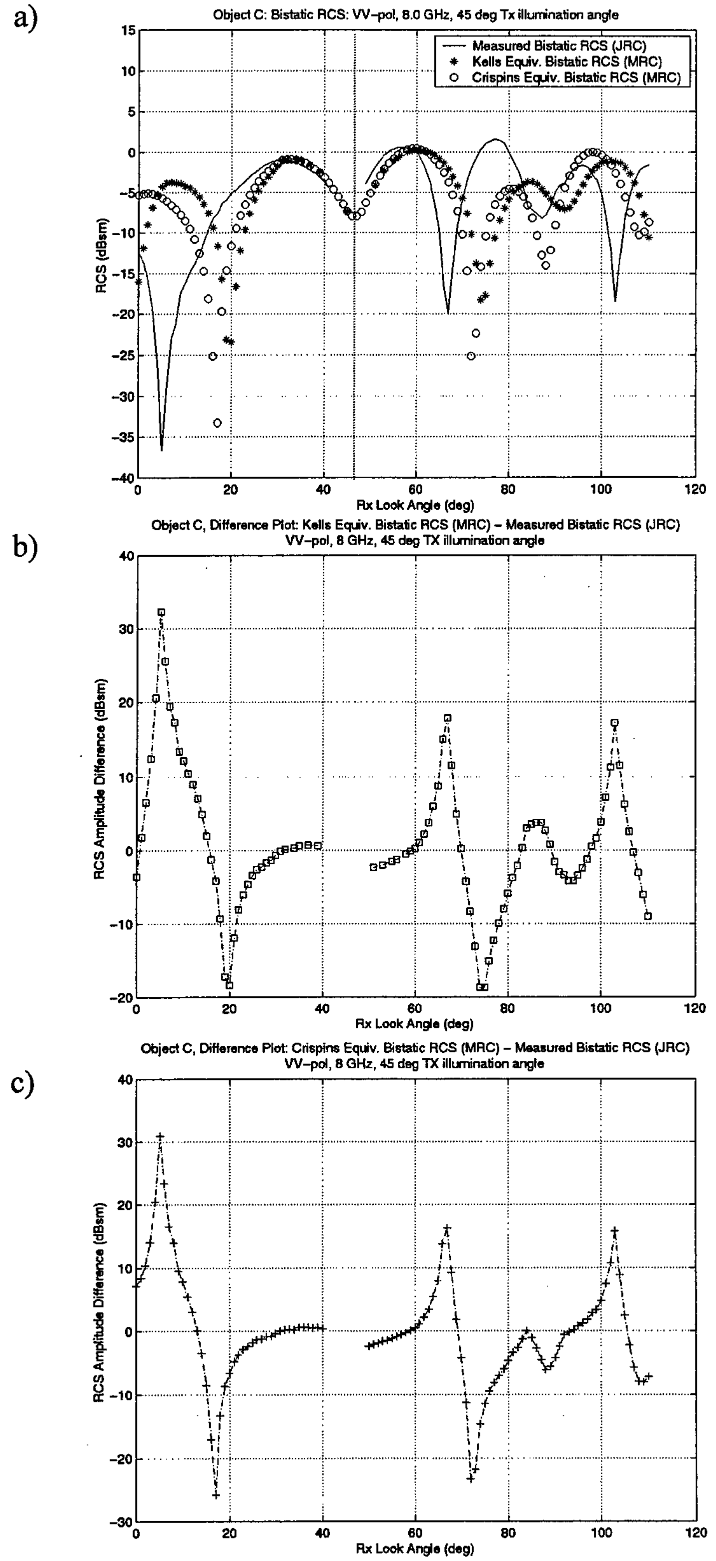


Figure 49a-c: Object C, VV-pol, 8 GHz

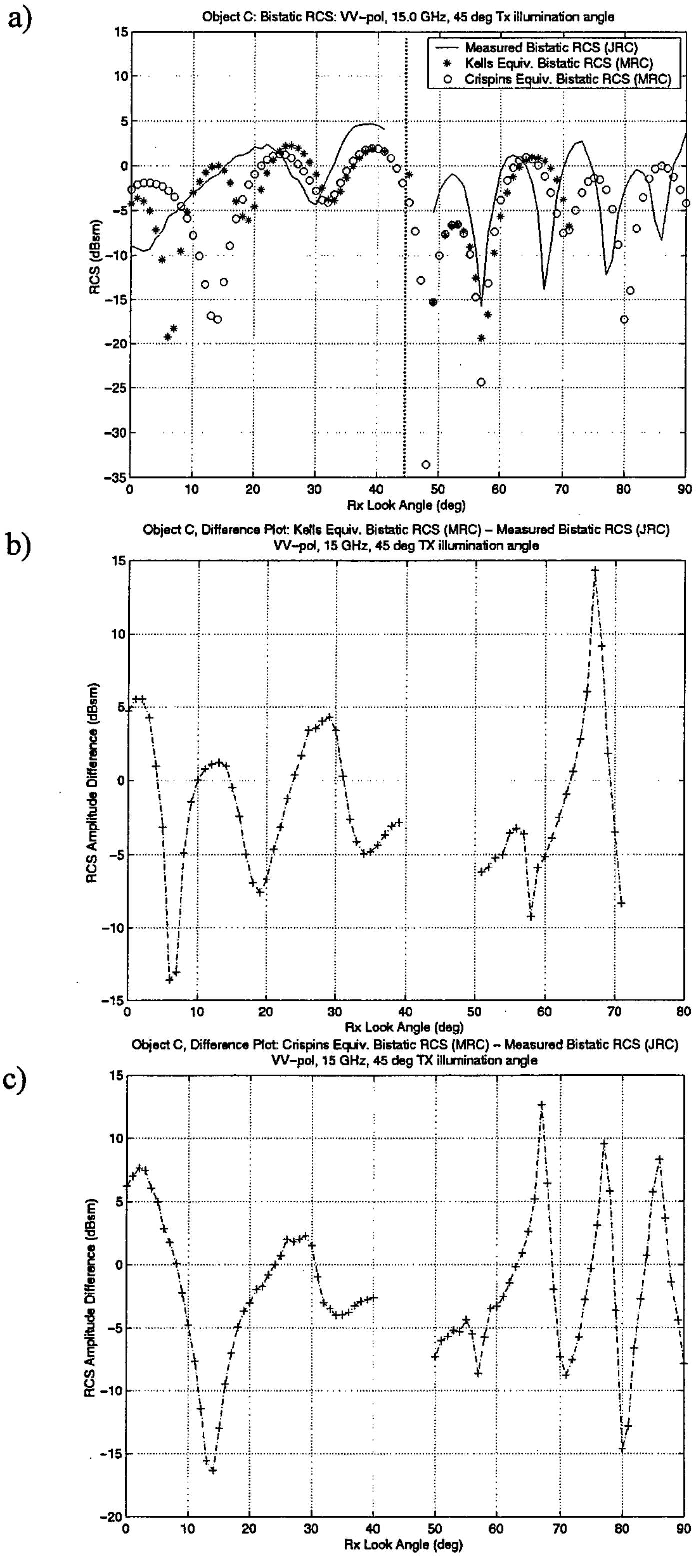


Figure 50a-c: Object C, VV-pol, 15 GHz

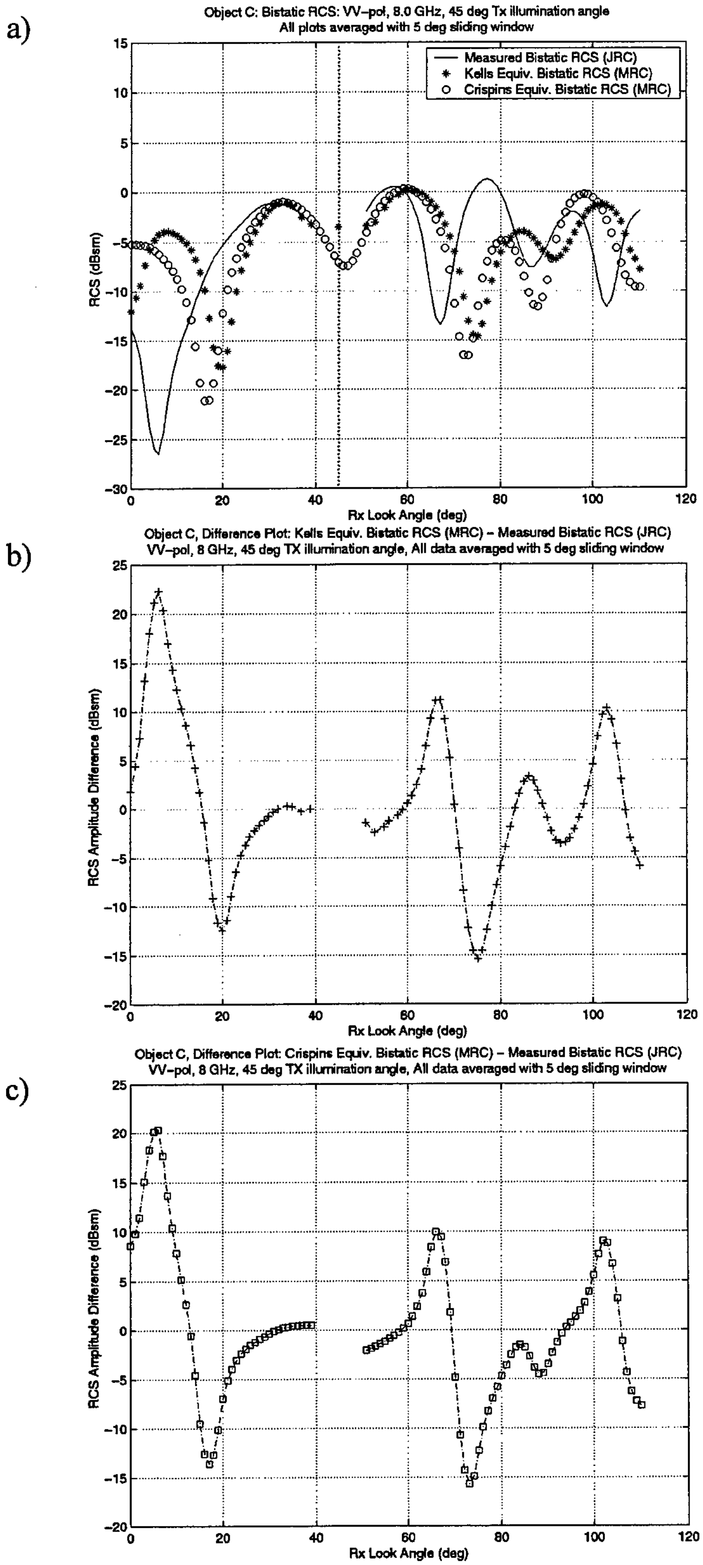


Figure 51a-c: Object C, VV-pol, 8 GHz, Data averaged with 5 degree sliding window

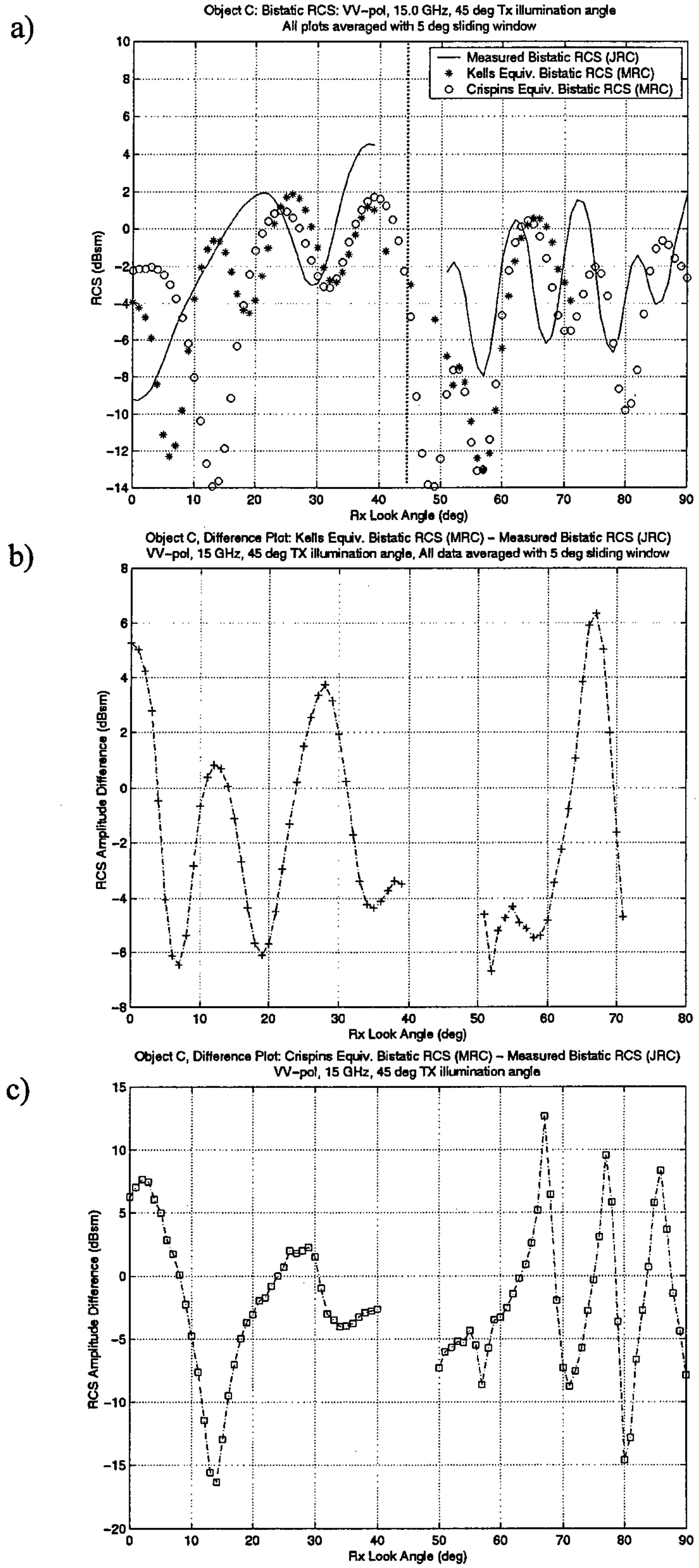


Figure 52a-c: Object C, VV-pol, 15 GHz, Data averaged with 5 degree sliding window

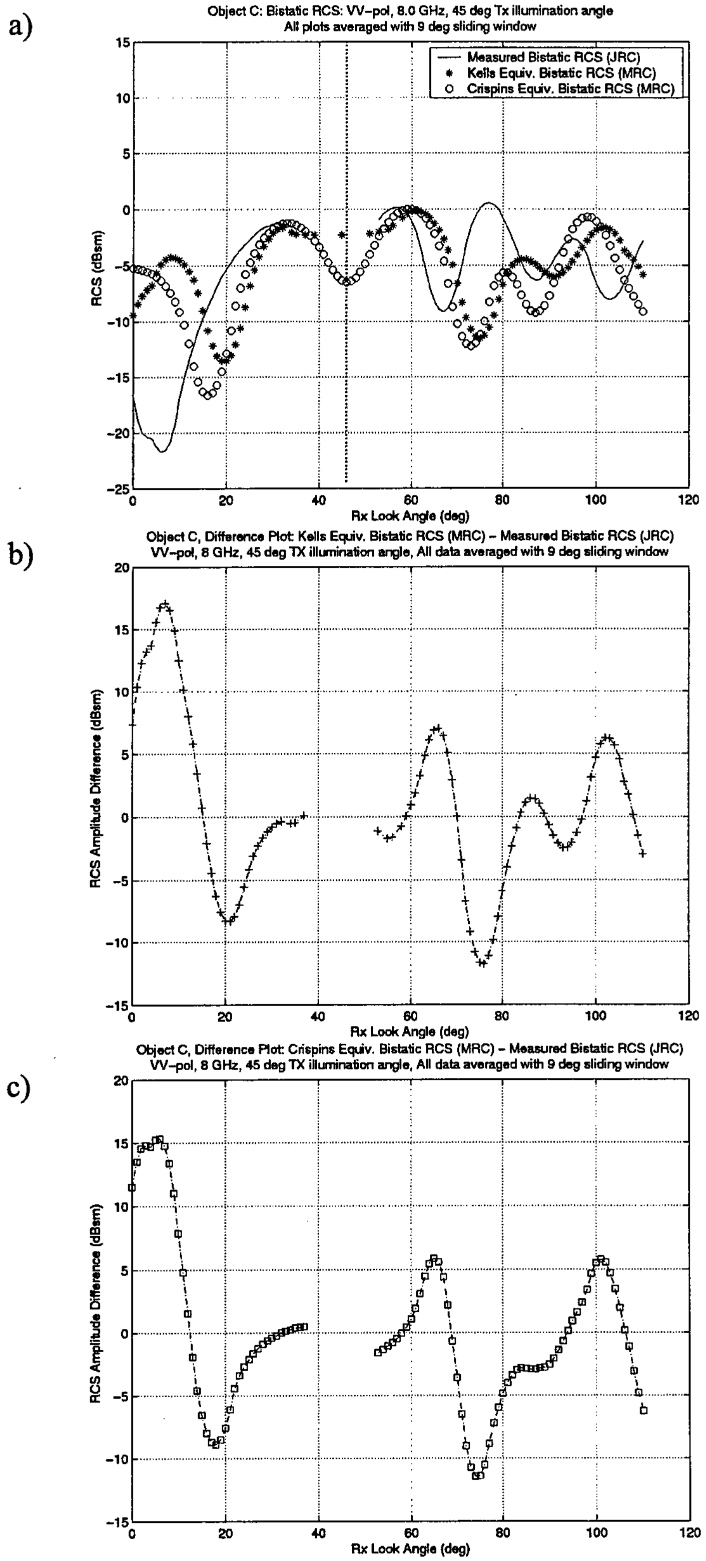


Figure 53a-c: Object C, VV-pol, 8 GHz, Data averaged with 9 degree sliding window

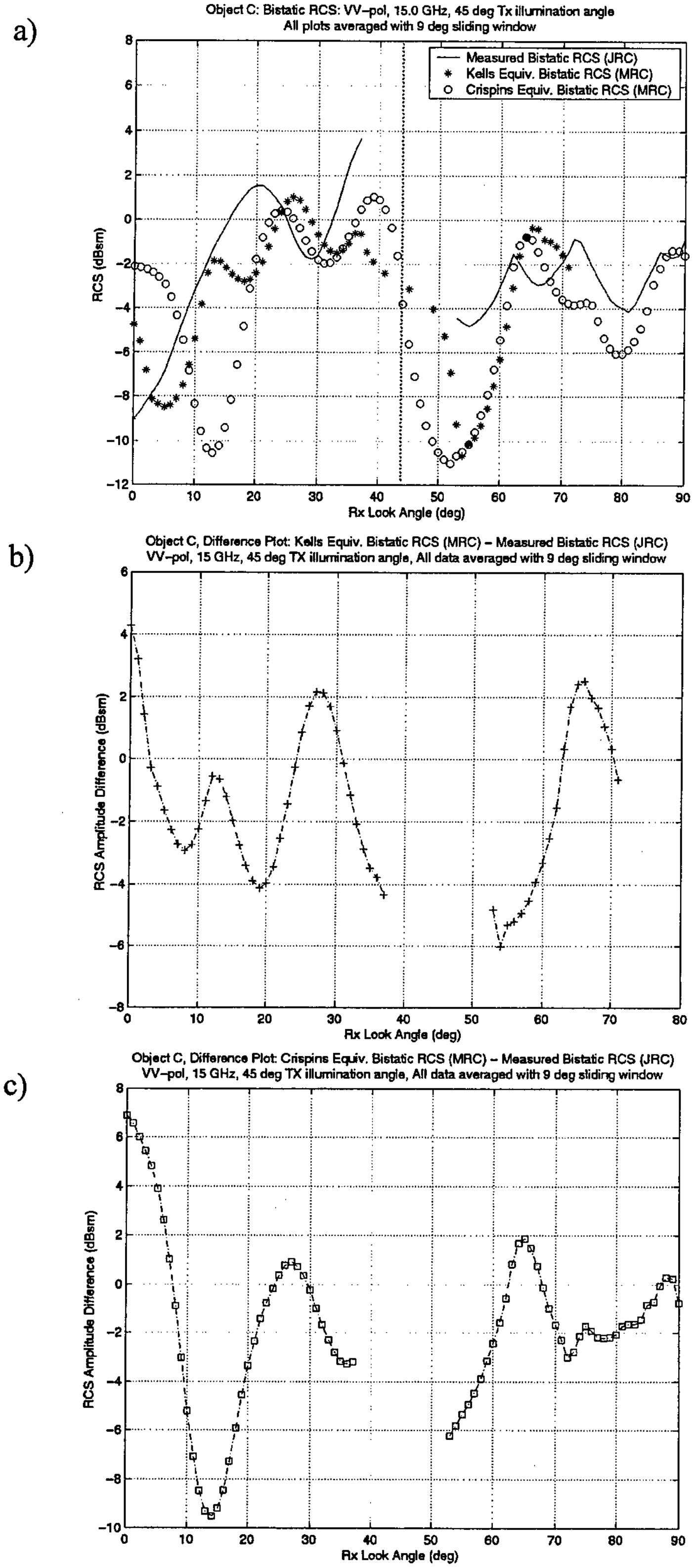


Figure 54a-c: Object C, VV-pol, 15 GHz, Data averaged with 9 degree sliding window

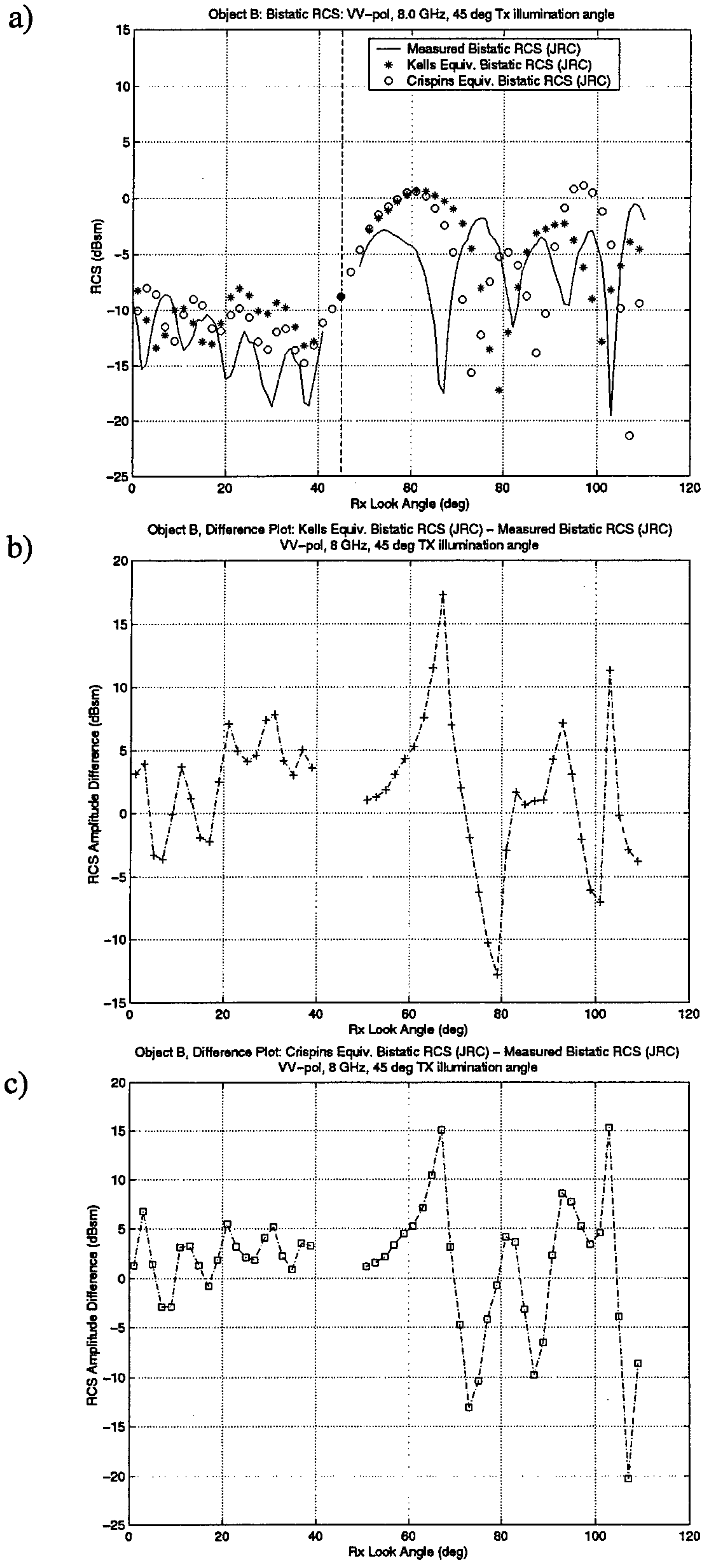
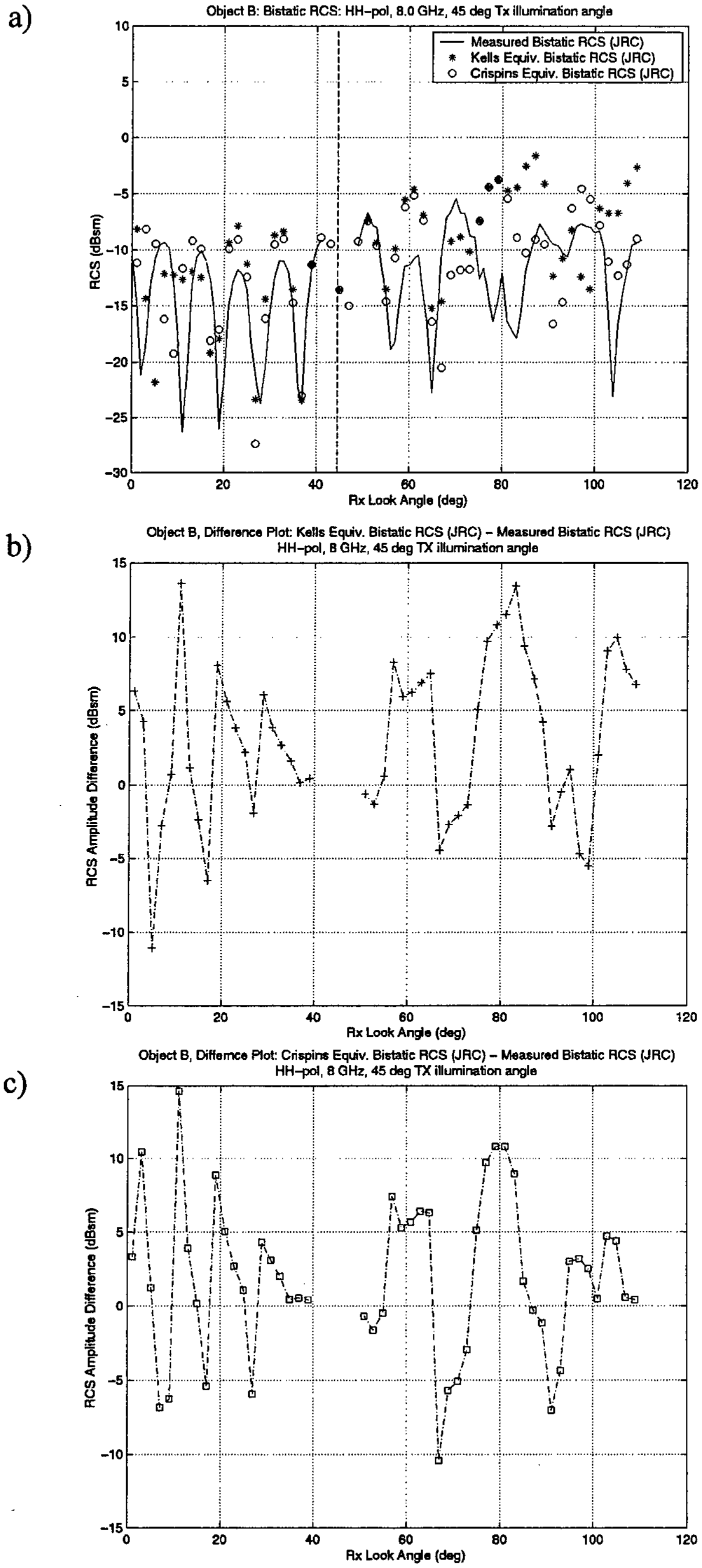


Figure 55a-c: Object B, VV-pol, 8 GHz



a) Object B: Bistatic RCS: HH-pol, 8.0 GHz, 45 deg Tx illumination angle

b) Object B, Difference Plot: Kells Equiv. Bistatic RCS (JRC) – Measured Bistatic RCS (JRC) HH-pol, 8 GHz, 45 deg TX illumination angle

c) Object B, Differnce Plot: Crispins Equiv. Bistatic RCS (JRC) – Measured Bistatic RCS (JRC) HH-pol, 8 GHz, 45 deg TX illumination angle

Figure 56a-c: Object B, HH-pol, 8 GHz

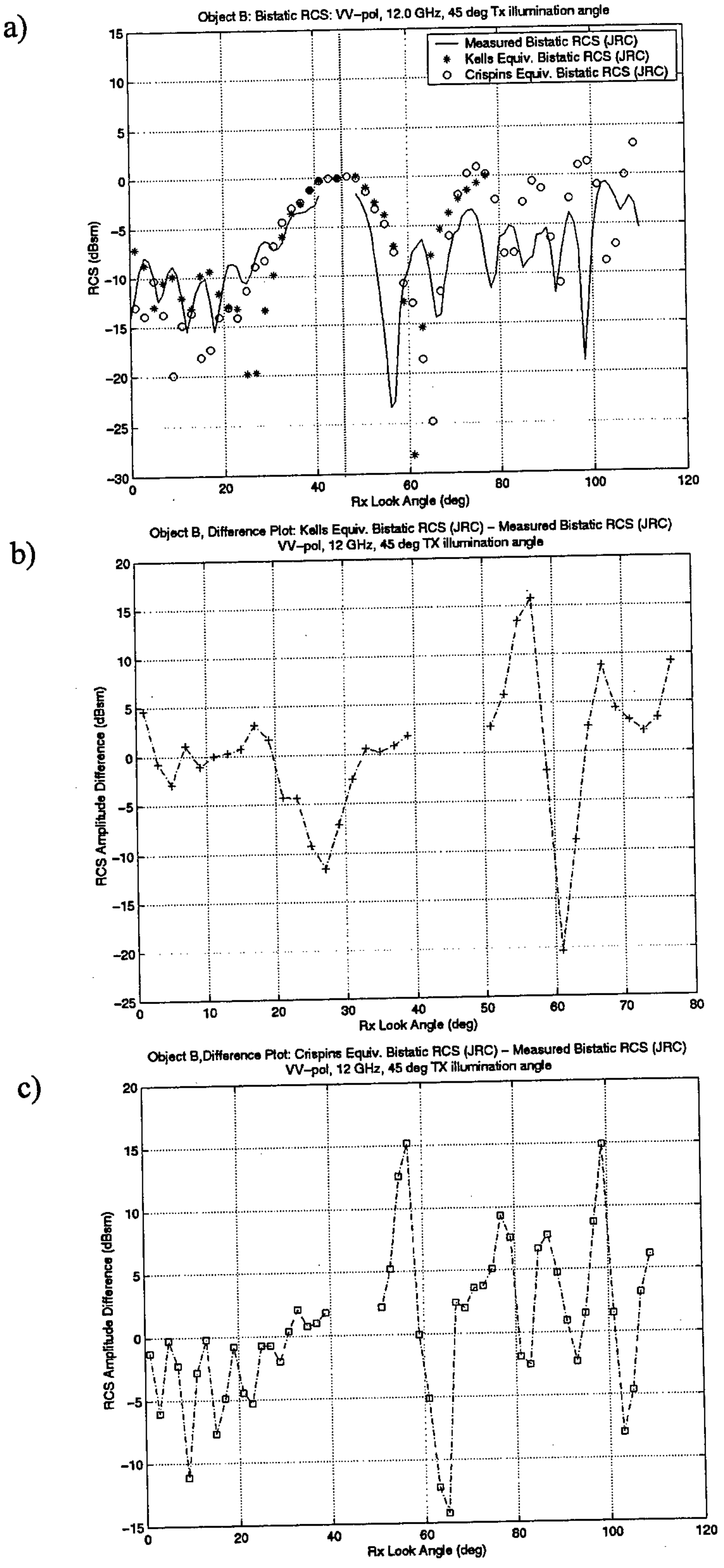


Figure 57a-c: Object B, VV-pol, 12 GHz

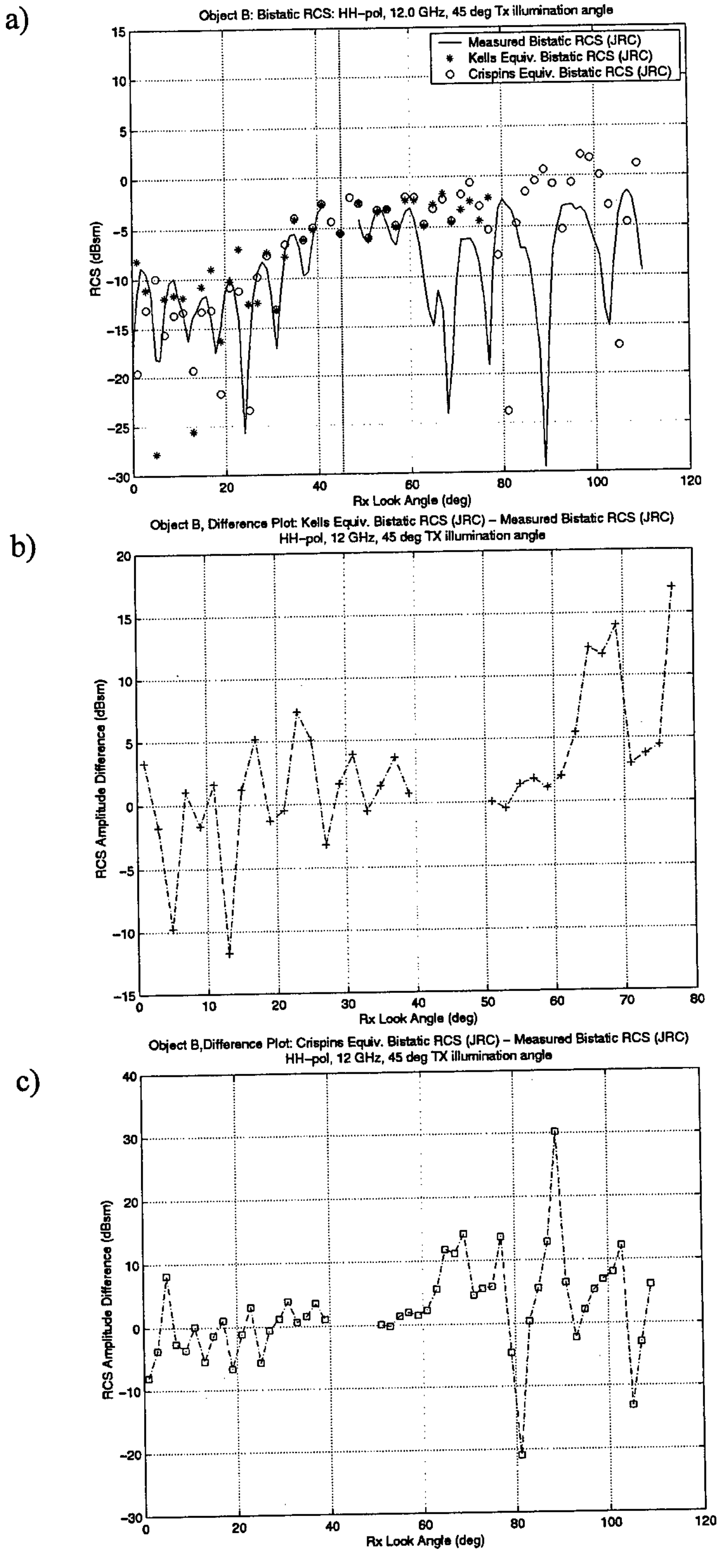


Figure 58a-c: Object B, HH-pol, 12 GHz

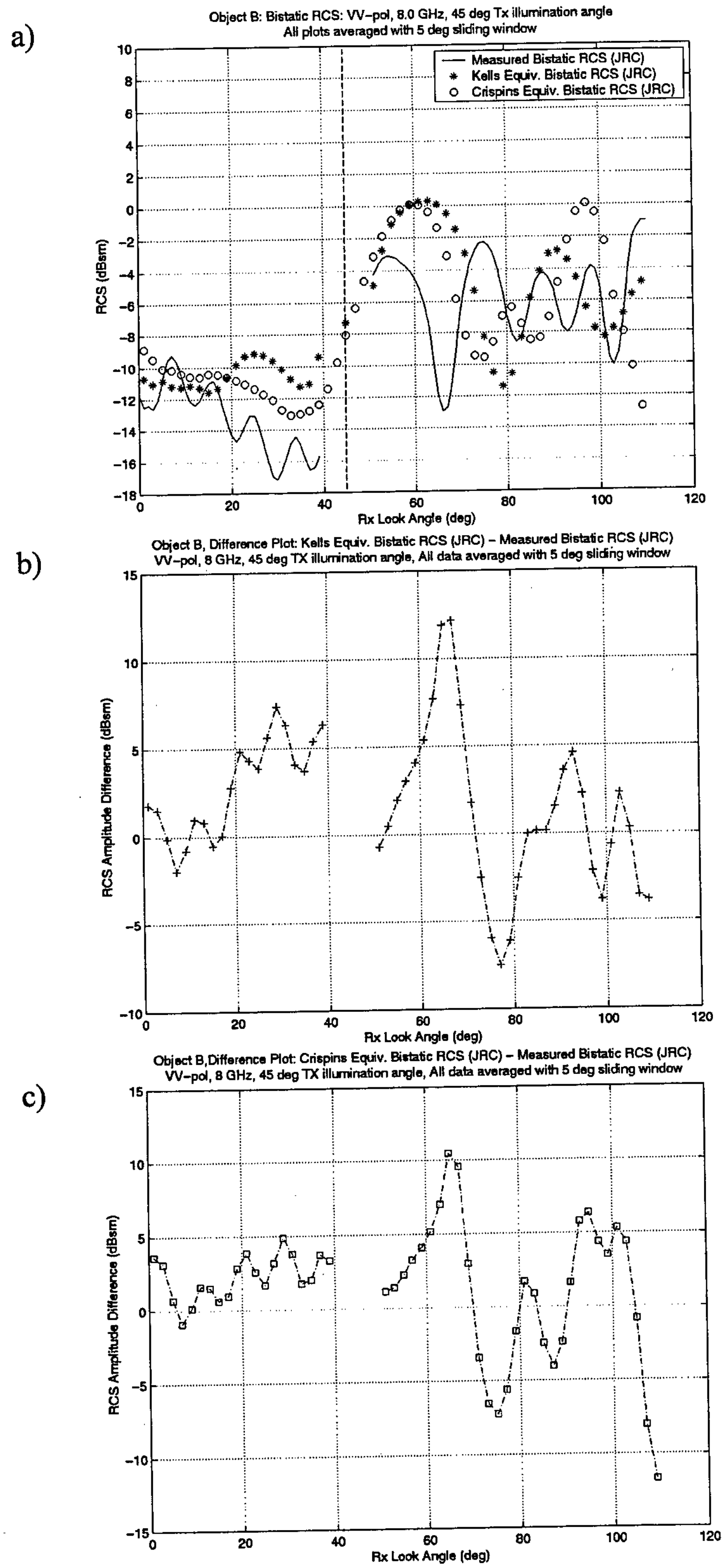


Figure 59a-c: Object B, VV-pol, 8 GHz, Data averaged with 5 degree sliding window

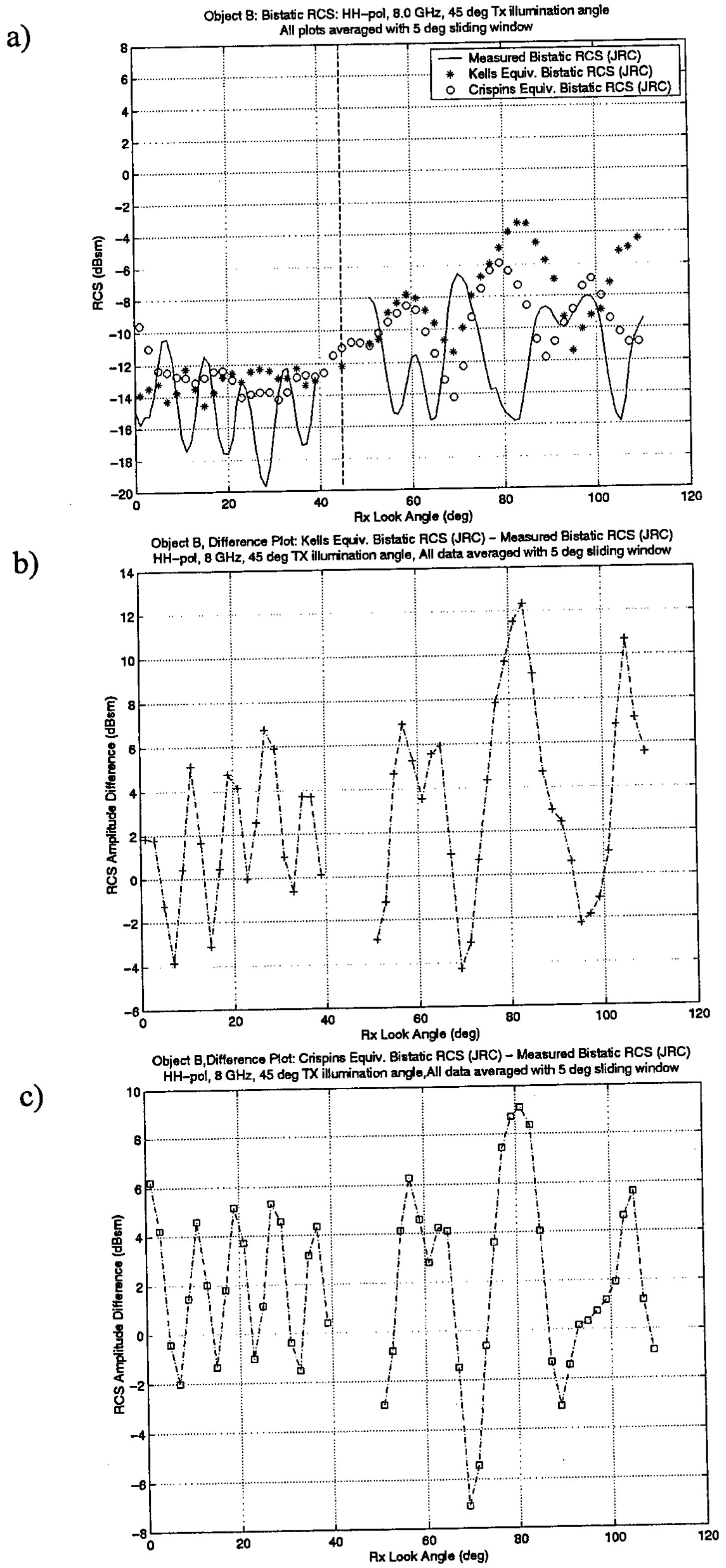


Figure 60a-c: Object B, HH-pol, 8 GHz, Data averaged with 5 degree sliding window

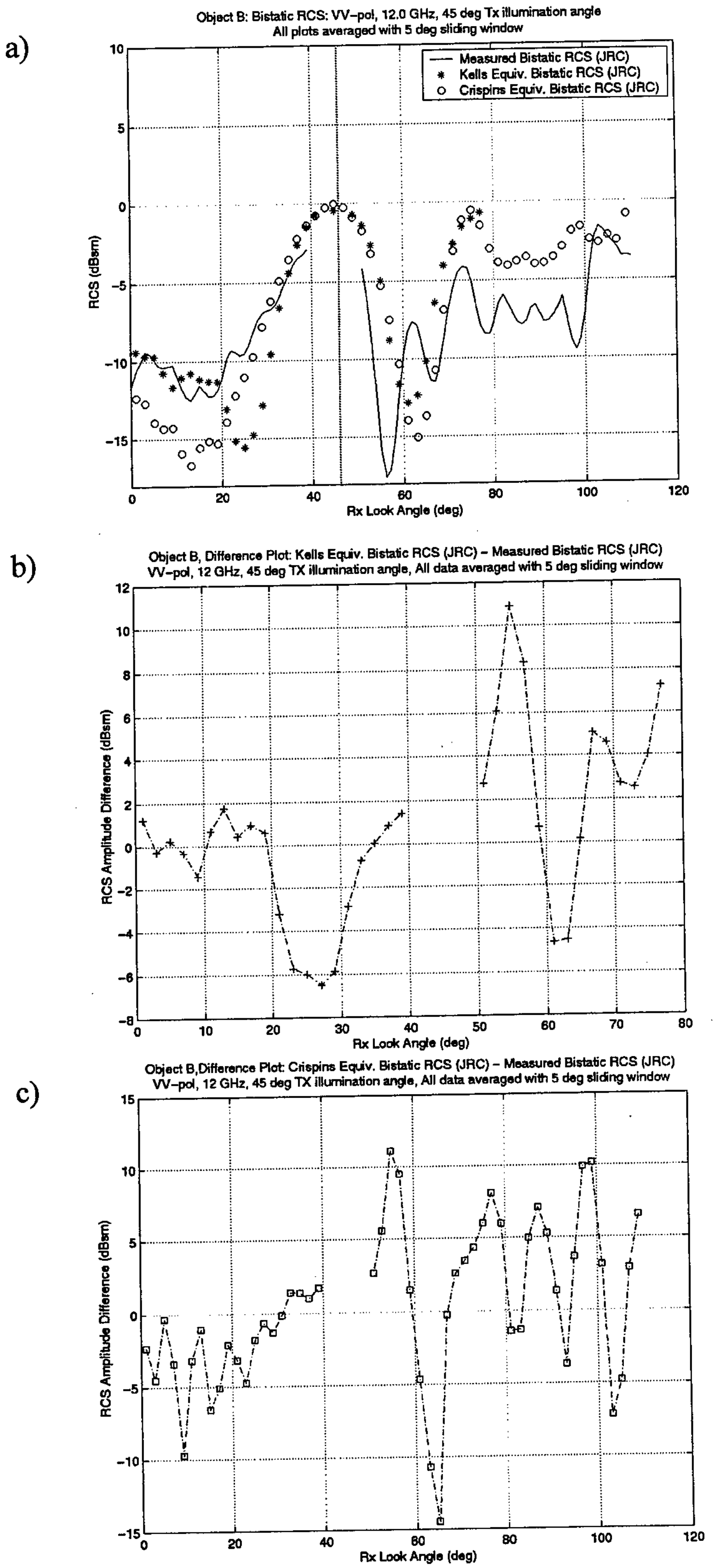


Figure 61a-c: Object B, VV-pol, 12 GHz, Data averaged with 5 degree sliding window

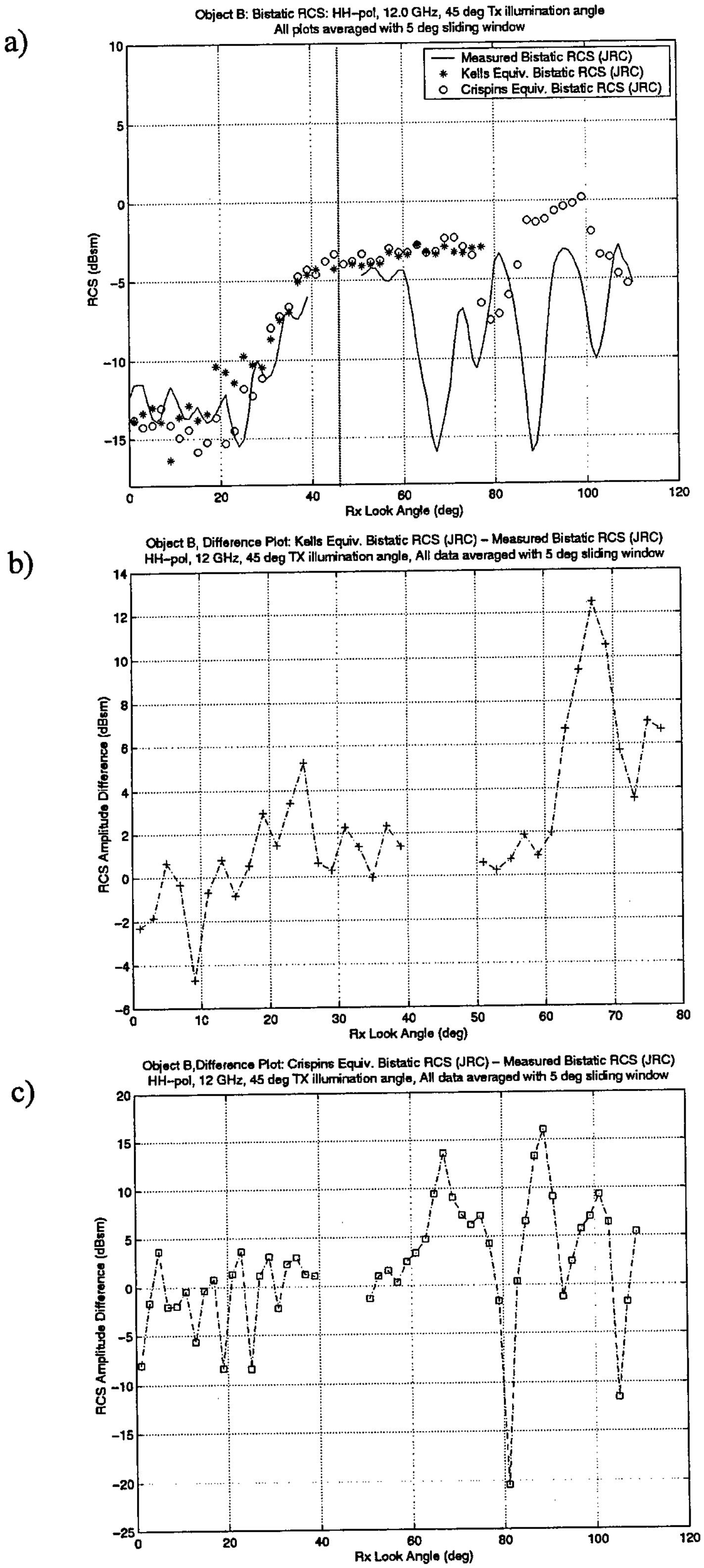


Figure 62a-c: Object B, HH-pol, 12 GHz, Data averaged with 5 degree sliding window

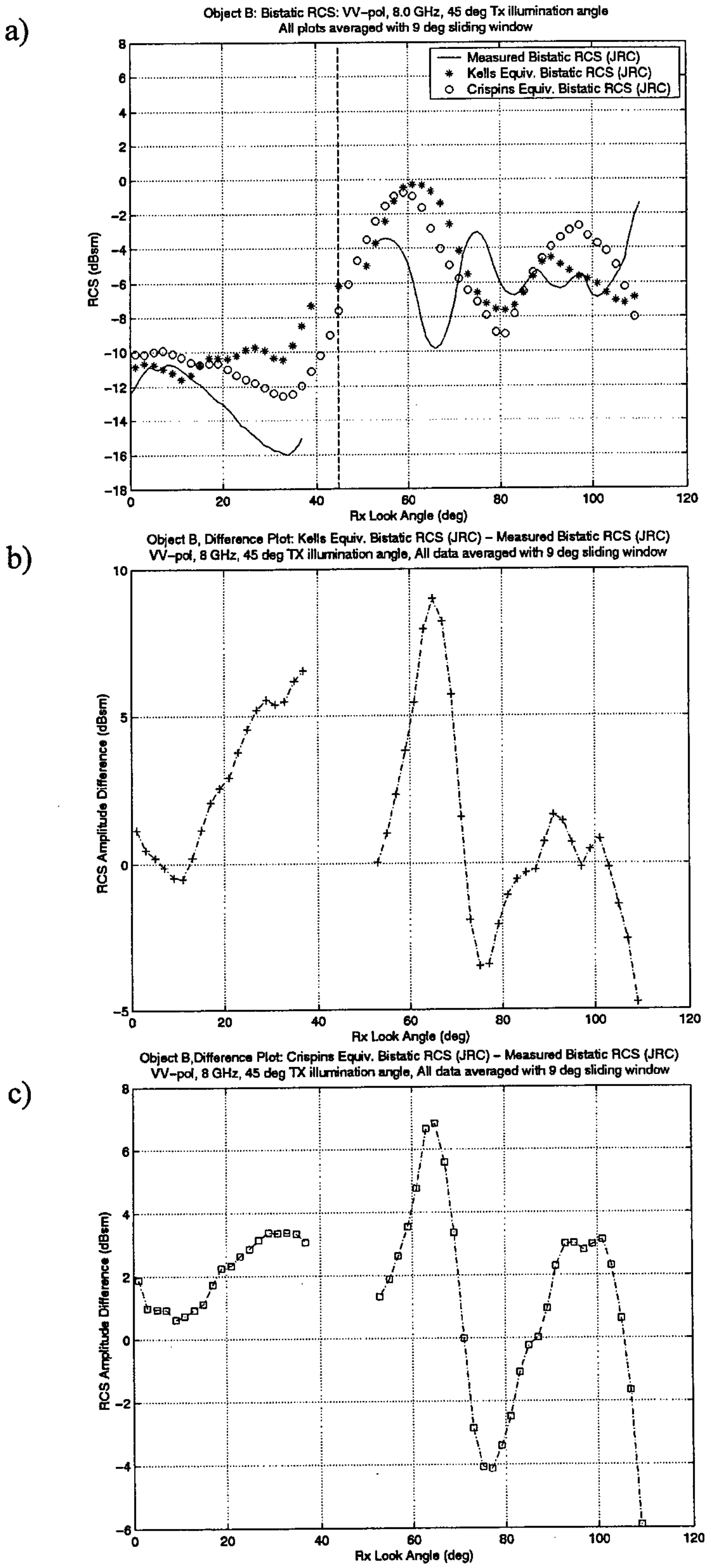


Figure 63a-c: Object B, VV-pol, 8 GHz, Data averaged with 9 degree sliding window

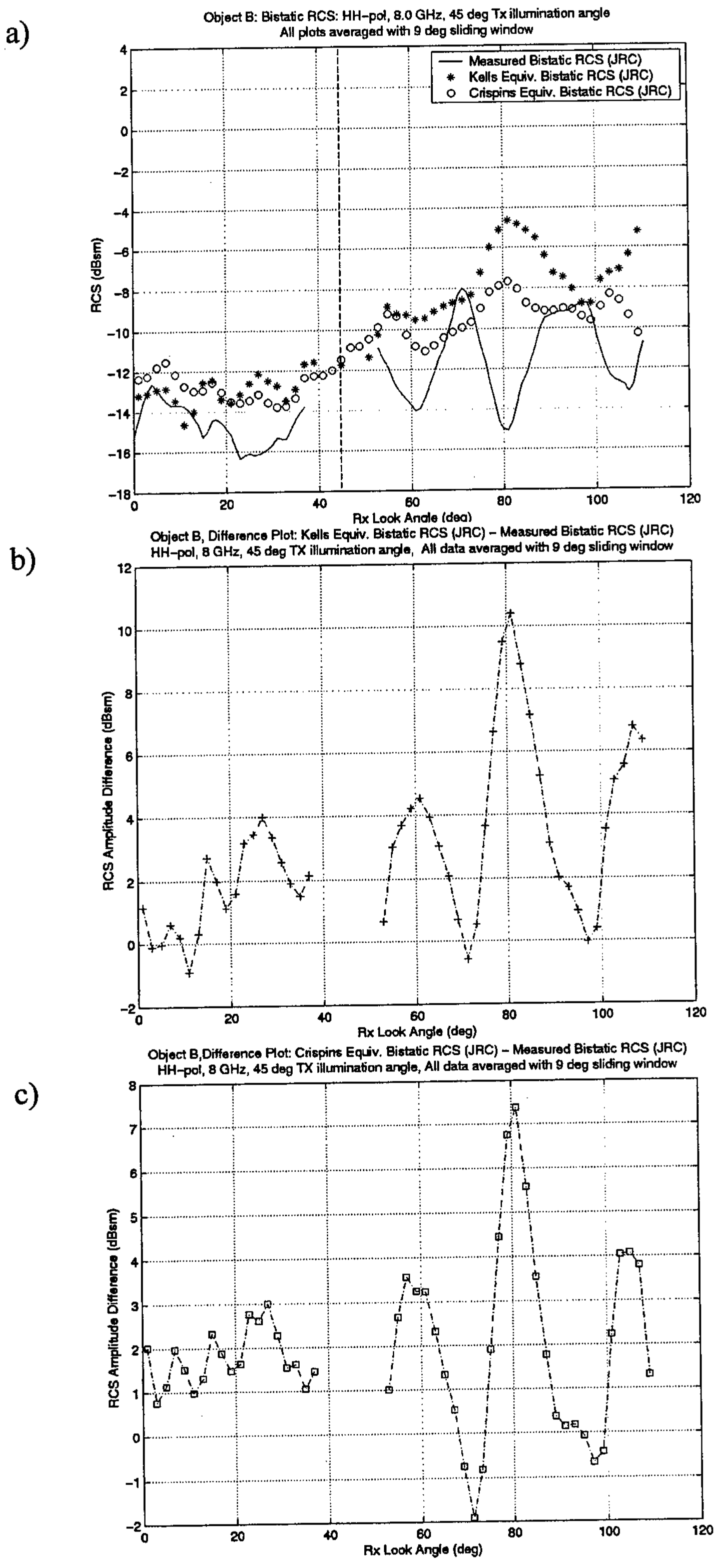
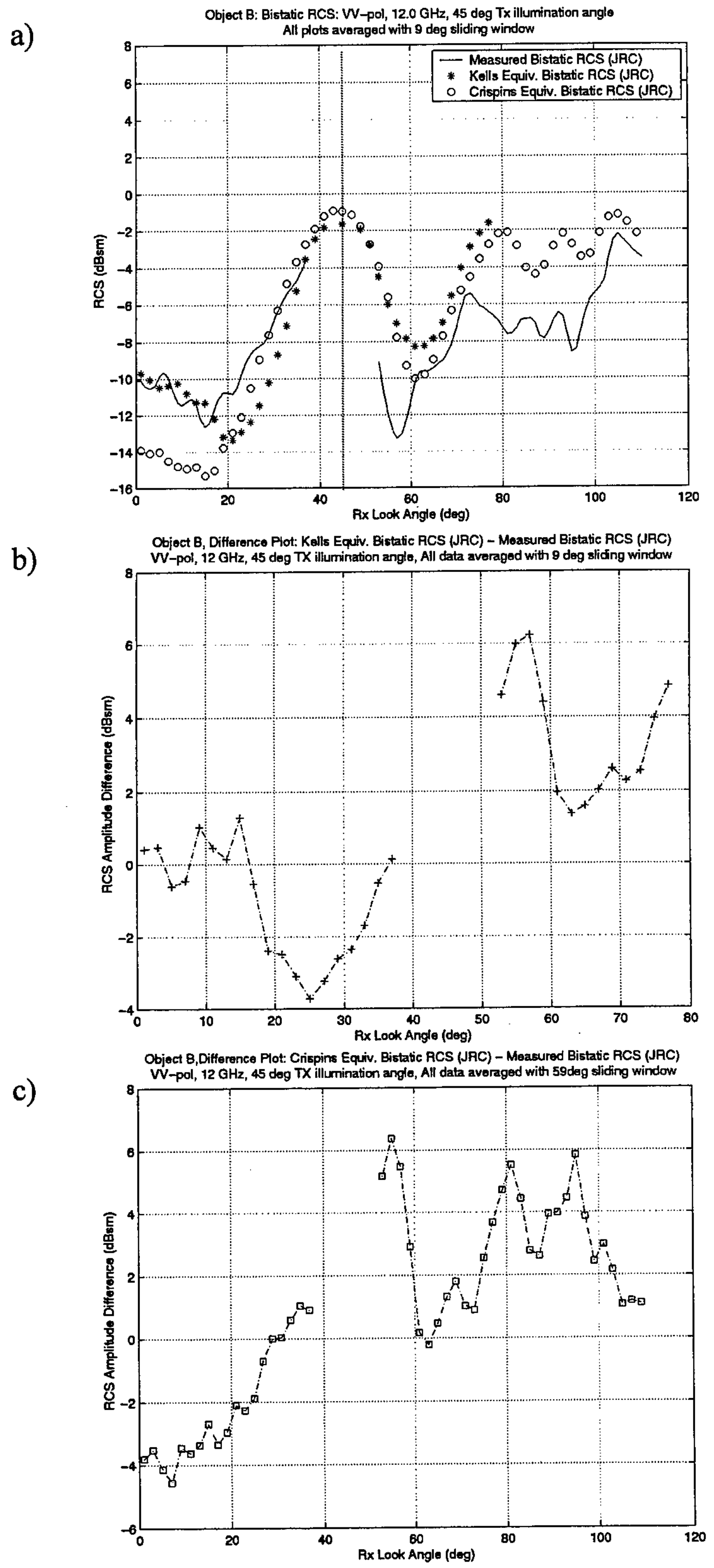


Figure 64a-c: Object B, HH-pol, 8 GHz, Data averaged with 9 degree sliding window



Object B: Bistatic RCS: VV-pol, 12.0 GHz, 45 deg Tx illumination angle
All plots averaged with 9 deg sliding window

Object B, Difference Plot: Kells Equiv. Bistatic RCS (JRC) – Measured Bistatic RCS (JRC)
VV-pol, 12 GHz, 45 deg TX illumination angle, All data averaged with 9 deg sliding window

Object B,Difference Plot: Crispins Equiv. Bistatic RCS (JRC) – Measured Bistatic RCS (JRC)
VV-pol, 12 GHz, 45 deg TX illumination angle, All data averaged with 59deg sliding window

Figure 65a-c: Object B, VV-pol, 12 GHz, Data averaged with 9 degree sliding window

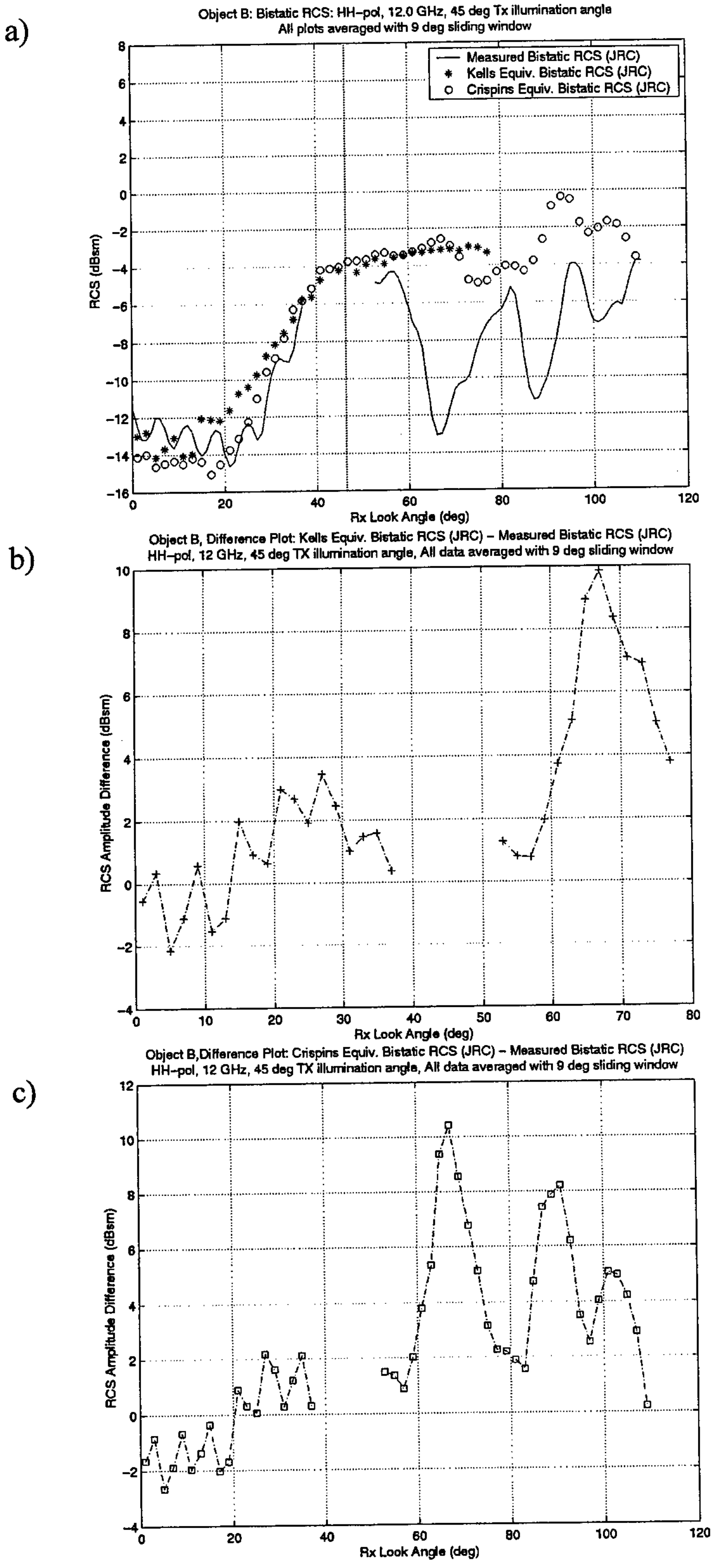


Figure 66a-c: Object B, HH-pol, 12 GHz, Data averaged with 9 degree sliding window

V. Conclusions and Recommendations

The following observations summarize the preceding section's analysis. The summary of the MBET performance is categorized as a function of the complexity of the object studied, because this characteristic is a significant predictor of MBET performance. Simple objects are those whose scattered signature is dominated by a single specular mechanism from any particular vantage point. Complex objects fall into one of two categories: 1) those whose RCS is dominated by a combination of specular interactions, and 2) those whose RCS is derived from specular and non-specular components of similar amplitude. The former is referred to as Minimally Complex Objects whose geometry generally incorporates canonical structures which support large specular reflections. The later is referred to as Rigorously Complex Objects. These are characterized by large shadowing features, cavities, or smoothly sloped surfaces, which may produce multi-bounce, diffraction, surface waves, etc. with amplitudes analogous to any speculars. Some recommendations for future study are included at the end.

Xpatch v2.4d Observations:

- Xpatch computing edge diffraction incorrectly near Rx edge-on incidence
- Xpatch predicts reasonably accurate bistatic RCS for targets dominated by wide angle speculars even if they don't meet electrically large criteria
- Xpatch predicts low bistatic RCS when shadowing features present

  - Second order scattering contributions more prevalent here and Xpatch doesn't predict them well
- Xpatch data skewed toward larger bistatic angles

  - Reason for this remains uncertain; could be misalignment of measured object, different electrical size of measured object and Xpatch facetized model or an Xpatch computation problem.

Kell's and Crispin's MBET Observations:

The overall performance of Kell's and Crispin's MBET are quite similar for all the tested objects, but there are some glaring limitations associated with using Kell's formula. Kell's MBET requires a much larger data set than Crispin's to predict an equivalent bistatic RCS matrix. Whereas a single monostatic pattern cut (i.e. single RF) can be used to predict the bistatic RCS through Crispin's formula, Kell's requires a large RF bandwidth monostatic measurement (at a very fine frequency resolution) to accomplish the same goal. The frequency shift also contributes to poor angular resolution near the transmitter illumination angle. And of course, extracting Kell's bistatic RCS from monostatic data sets proves to be much more computationally expensive than Crispin's.

Observations for both MBETs' performance as a function of the object's inherent complexity are provided below.

For Simple Objects:

- Kell's & Crispin's MBET work well for simple geometries for at least bistatic angles of 30 degrees (sidelobe structure & amplitudes); Crispin's has slight edge over Kell's in general, but Kell's has slight advantage when diffraction effects present
- Both predict edge diffraction effects lower on average than measured (2-3 dB)
- Averaging data improves correlation between MBET prediction and measured data

For Minimally Complex Objects:

- Both MBETs provide reasonable accuracy for bistatic angles of no more than 15-20°
- Both tend to predict bistatic RCS lower than measured for bistatic angles > 15° and at higher RFs. This is primarily due to a narrowing of monostatic specular spikes at higher RFs and the changing nature of the scattering centers as the bistatic angle increases.
- Averaging doesn't improve correlation with measured data sets nearly as much as it did for simple shapes and may even decrease correlation

For Rigorously Complex Objects:

- Both MBETs demonstrate reasonable accuracy for bistatic angles of no more than 5-10°
- Both tend to predict bistatic RCS higher than measured for bistatic angles > 10° and at higher RFs. Again the primary reason for poor correlation is the changing nature of the scattering centers. From these types of objects, the scattering centers change more rapidly as a function of bistatic angle because they are derived from roughly equivalent specular and non-specular components.
- Averaging data doesn't improve data correlation with measured data sets

Recommendations:

The process initiated for this research continues to collect data. More analysis can be accomplished on this new data as well as some of the existing data, which has yet to be investigated. A few of the recommendations are as follows:

- Investigate cross-polarization data in a similar fashion as was done for the co-pol data in this report.
- Investigate data from additional object (see Appendix A) not reviewed in this thesis
- Generate imaging plots of monostatic and bistatic data to highlight nature of scattering centers as a function of the bistatic angle
- Expand object test set to include more simple shapes to highlight certain non-specular effects and investigate existing objects from new transmitter and receiver look angles
  - Ogive investigation of surface wave effects or hollow cylinder for multi-bounce analysis
  - Look at forward scatter region for existing complex objects
- Derive more objective and mathematically justifiable criteria for defining the *complexity* of an object based on geometrical features and electrical size
- Perform more rigorous statistical analysis on the measured and MBET computed data to better define *correlation*

Currently, AFIT and the JRC enjoy a strong relationship, in part because of the work accomplished in this project. This relationship should be fostered and nurtured through additional joint ventures so bistatic scattering research can continue unabated. Additional joint projects between AFIT and several other DoD bistatic measurement facilities (at Point Magu and Hanscom) should also be pursued as more of a long term goal.

Appendix A. Measurement Orientations

Slightly different measurement configurations exist at both the MRC and JRC test chambers. This effects the polarization orientation as perceived by the reader. Because it is easier for a reader to envision a single reference orientation/polarization throughout the analysis, actual target/polarization orientations were changed from their original within the body of the text. This appendix is included for the reader's convenience to relate the true measurement geometries. Polarization is discussed in terms of E-field alignment with a particular coordinate axis.

The MRC chamber is configured just as has been previously discussed; no change in orientation exists. The horizontal plane is the main collection plane, and thus targets were placed on a large piece of RAM and spun around in azimuth to collect monostatic data. VV-pol for all objects derives from an E-field parallel with the z-axis (and vertical to ground plane), HH-pol is parallel to the x-y plane.

JRC uses two different target orientations depending on whether one is conducting monostatic or bistatic measurements. The monostatic measurements are taken from an orientation identical to MRC's. The bistatic data, however, is actually taken in the vertical plane. Targets are essentially laid on their side for experimentation. VV-pol is now parallel with the y-axis and HH-pol parallel to the x-z plane. Figures A-1 through A-3 show the actual monostatic and bistatic measurement orientations for Objects A, B and C. Additional bistatic data exists for a fourth target, a stand-alone canted hollow cylinder. The measurement orientation is given in Fig. A-4.

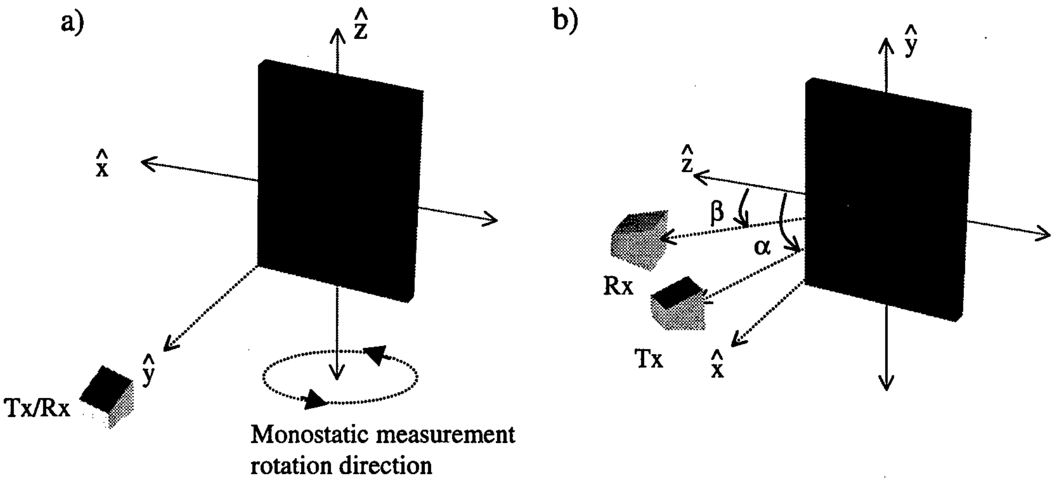


Figure A-1: a) Object A orientation for each JRC & MRC monostatic measurement, b) Object A orientation for JRC bistatic measurements

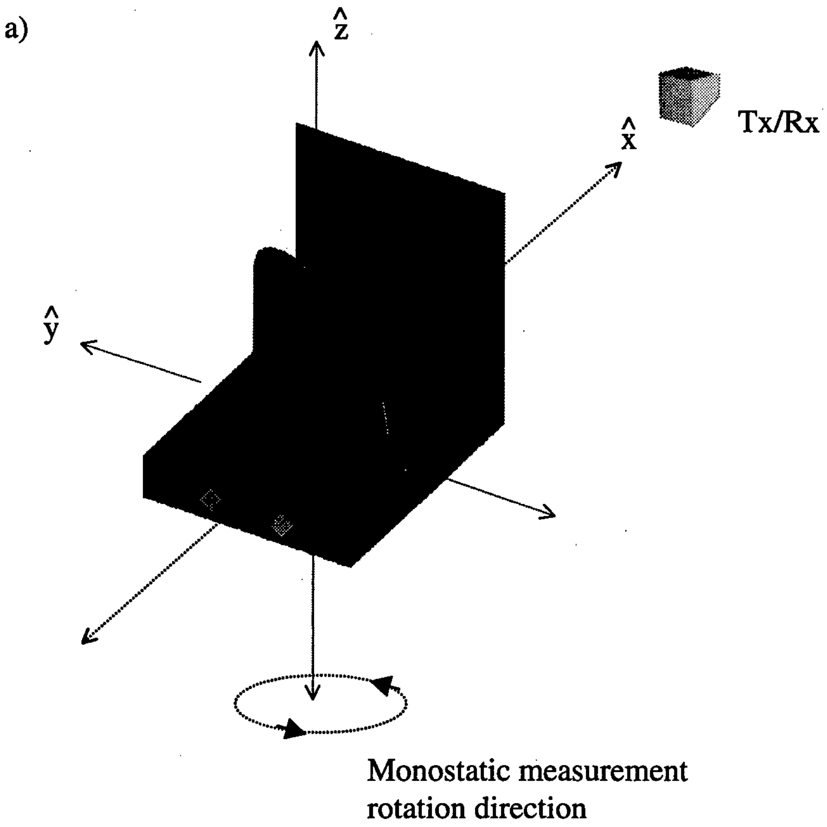


Figure A-2: a) Object B orientation for each JRC & MRC monostatic measurement

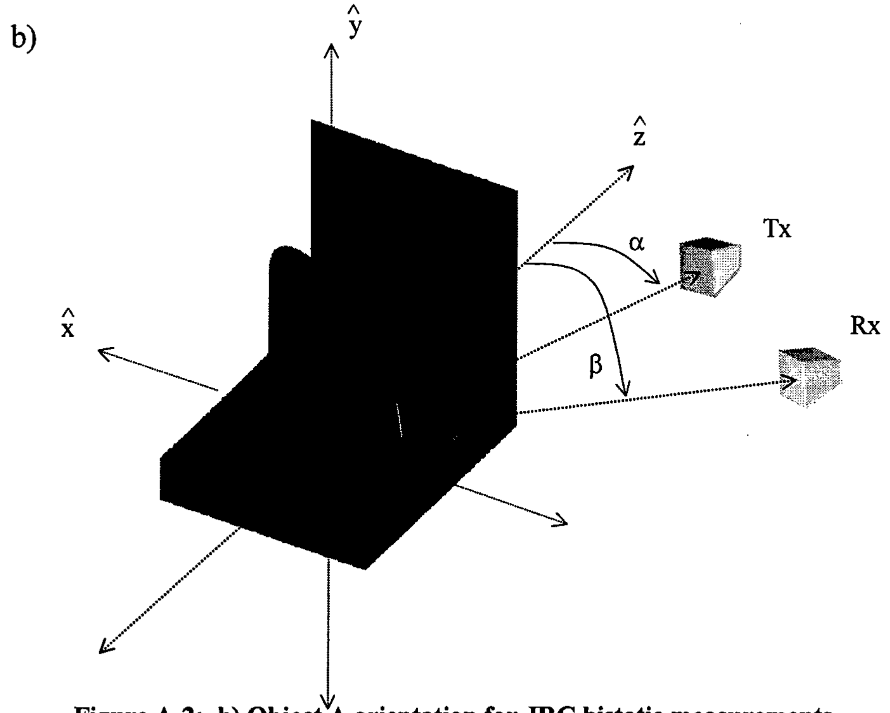


Figure A-2: b) Object A orientation for JRC bistatic measurements

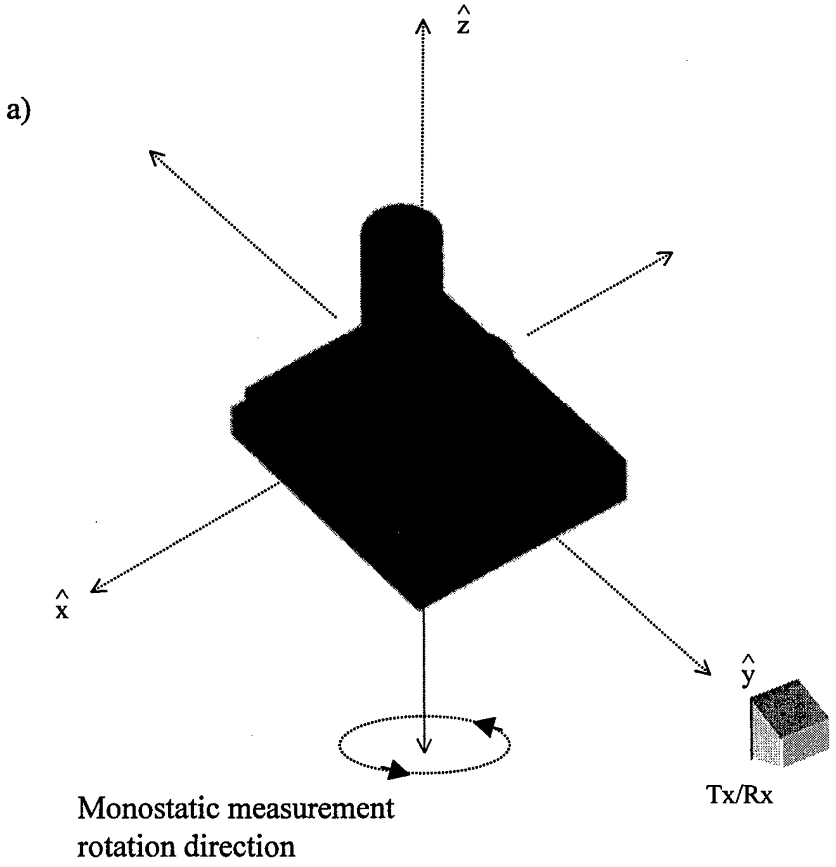


Figure A-3: a) Object C orientation for each JRC & MRC monostatic measurement

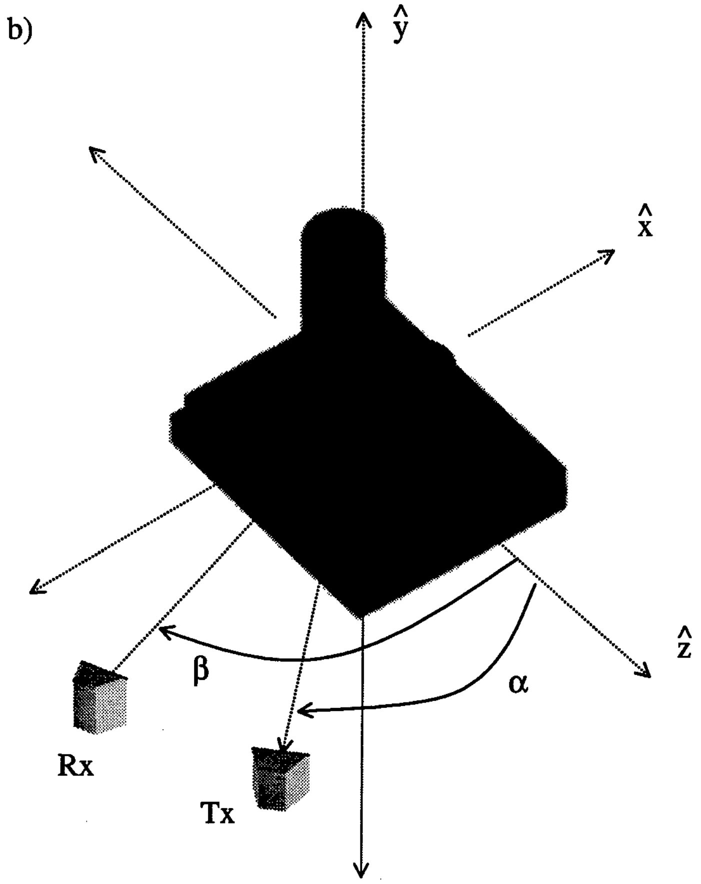


Figure A-3: b) Object A orientation for JRC bistatic measurements

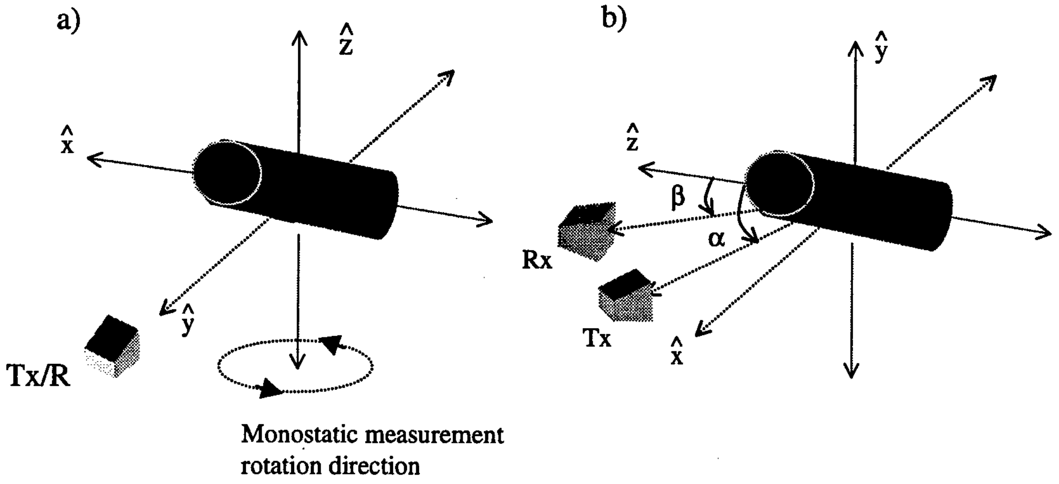


Figure A-4: a) Object D orientation for each JRC & MRC monostatic measurement, b) Object D orientation for JRC bistatic measurements

Appendix B: Waterline Pattern Cut Plots of Measured Data

Pattern cut plots of all measured test objects appear in Figs. B-1 through B-9. The entire collected data set spans the RF bandwidth and angular regions mentioned in Tables 2a-c, but these plots only depict data corresponding to the frequencies analyzed in Section IV. The fourth object (D) is included here for completeness, although it is not analyzed within this report.

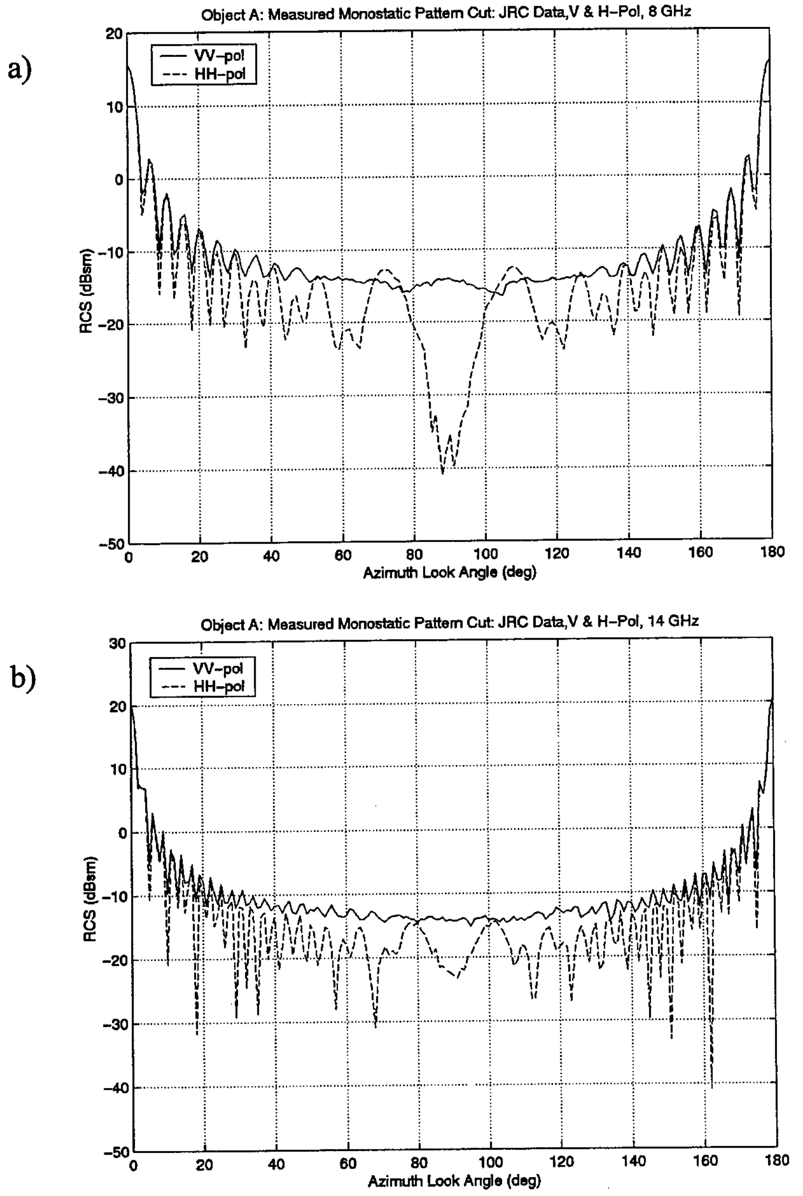


Figure B-1: JRC Measured Monostatic Data for Object A: a) 8 GHz, b) 14 GHz

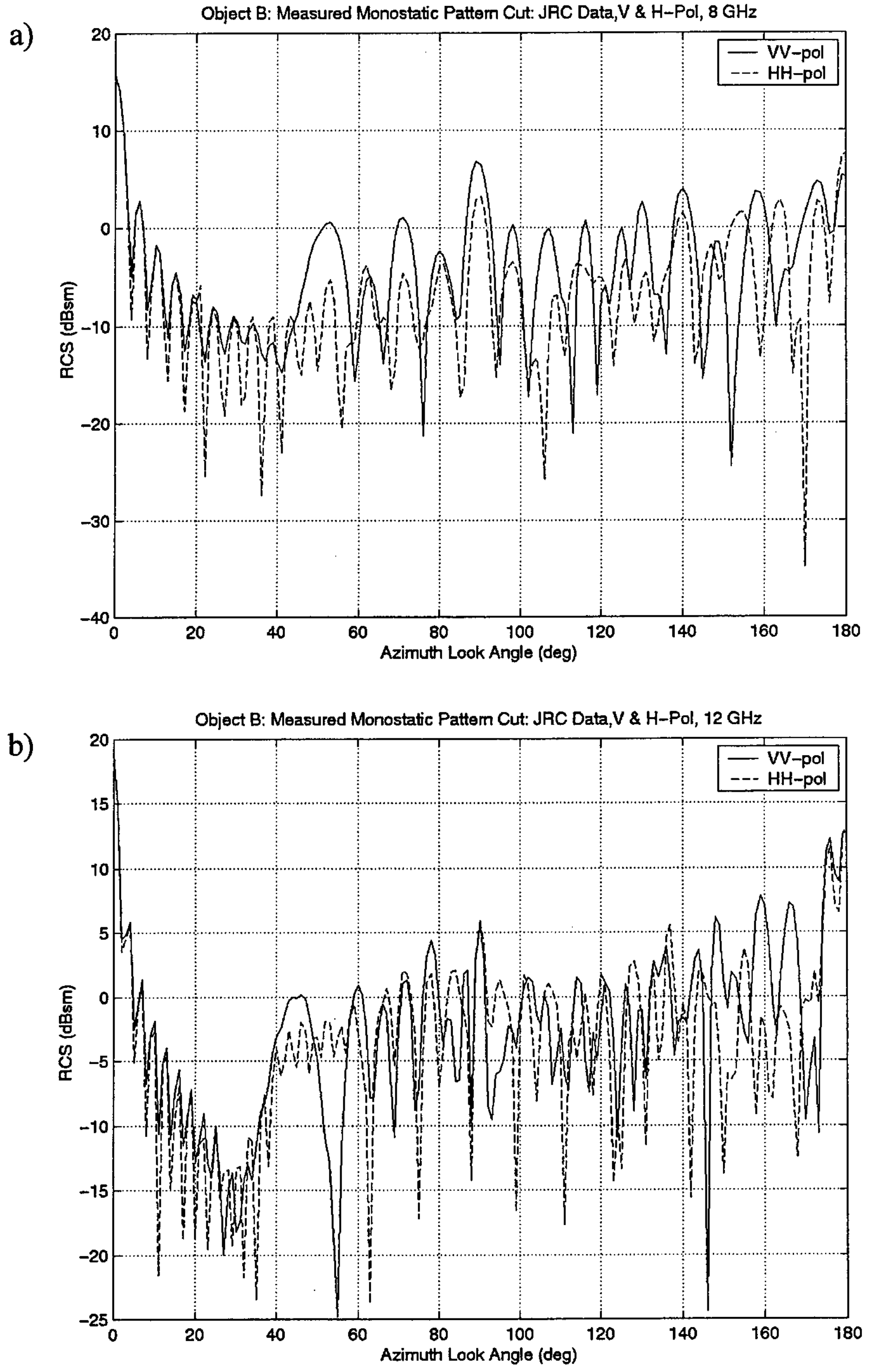


Figure B-2: JRC Measured Monostatic Data for Object B: a) 8 GHz, b) 12 GHz

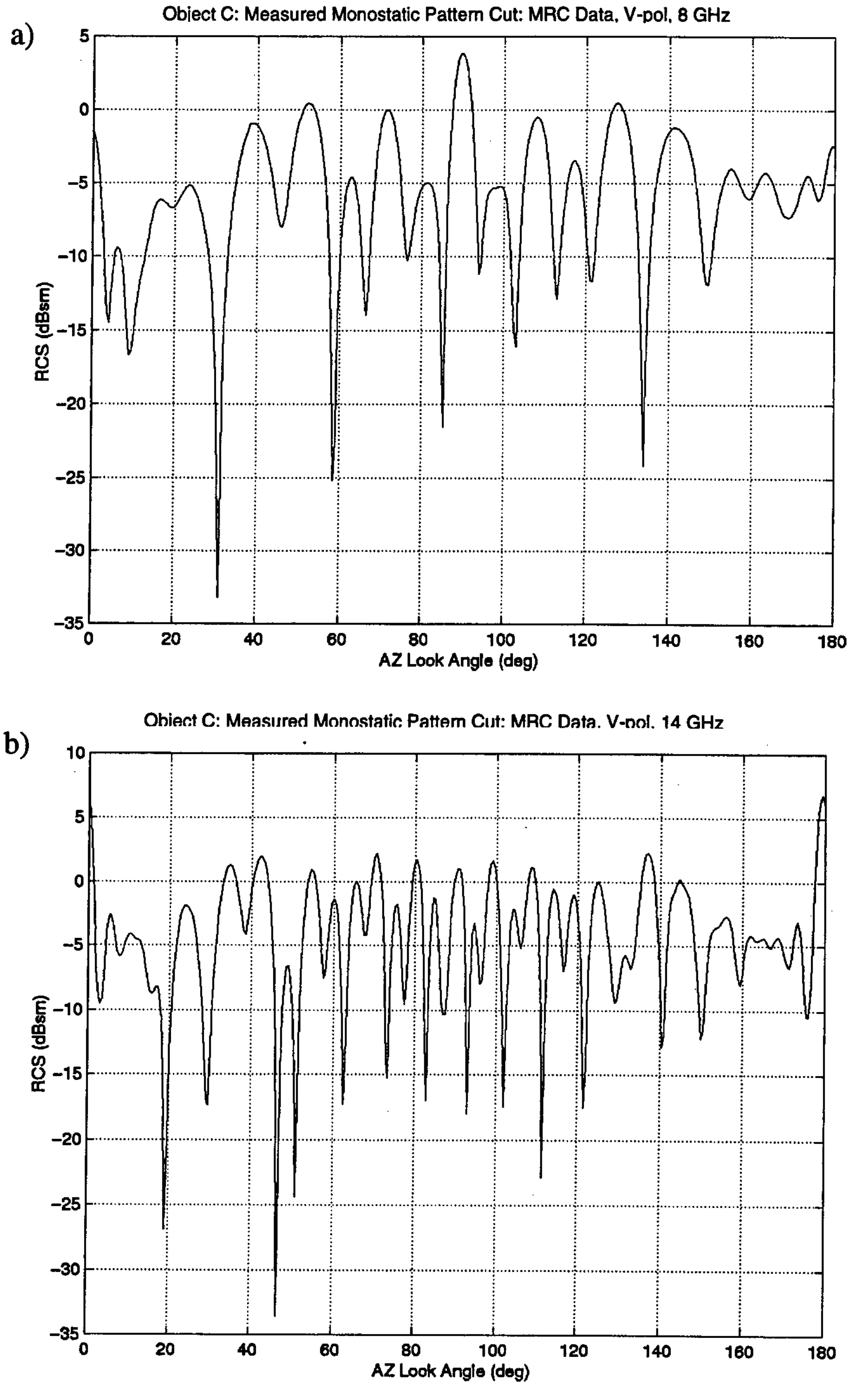


Figure B-3: MRC Measured Monostatic Data for Object C: a) 8 GHz, b) 14 GHz

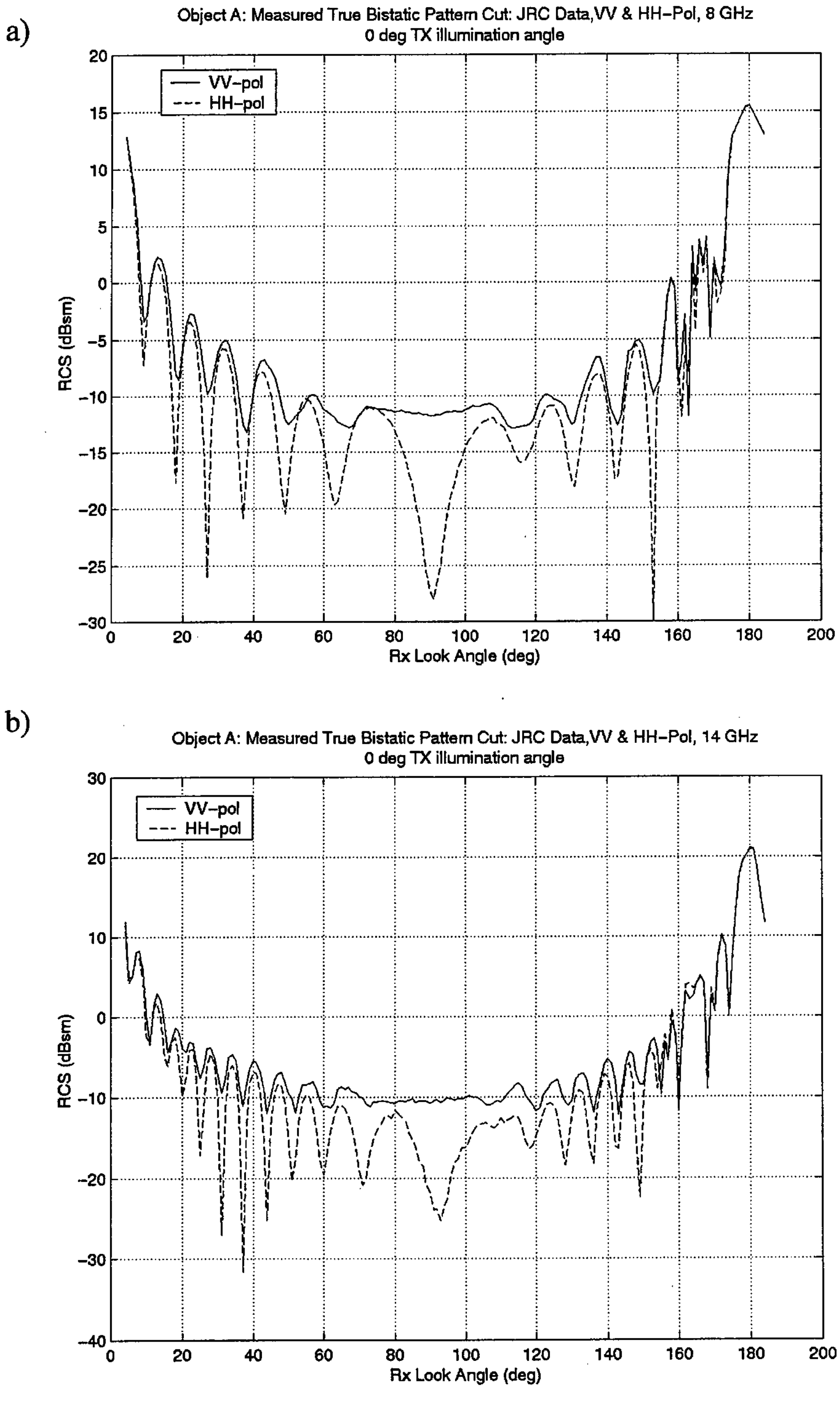


Figure B-4: JRC Measured Bistatic Data for Object A: a) 8 GHz, b) 14 GHz

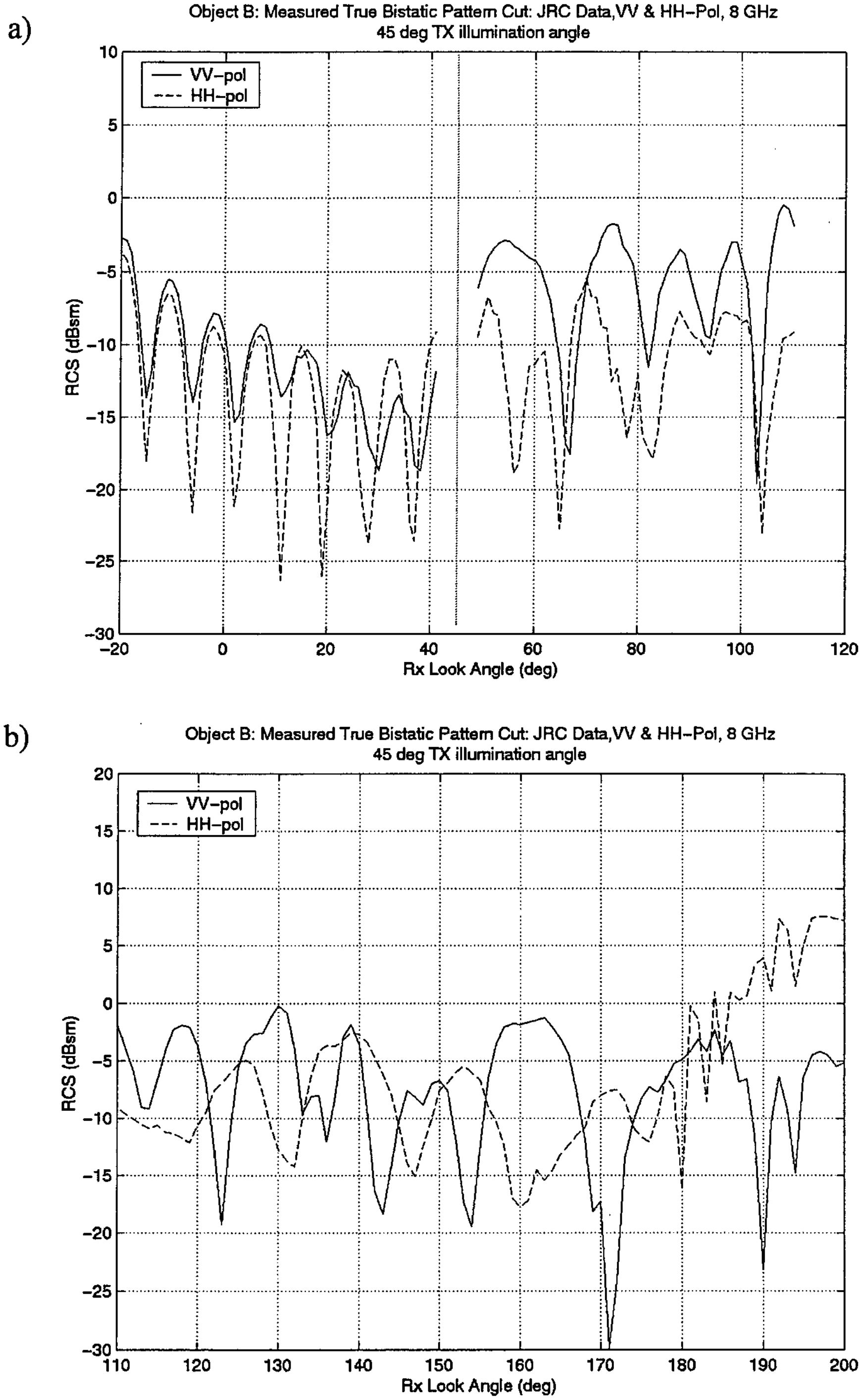


Figure B-5: JRC Measured Bistatic Data for Object B, 8 GHz: a) –20 to +90 degrees Rx look angle , b) +90 to +200 degrees Rx look angle

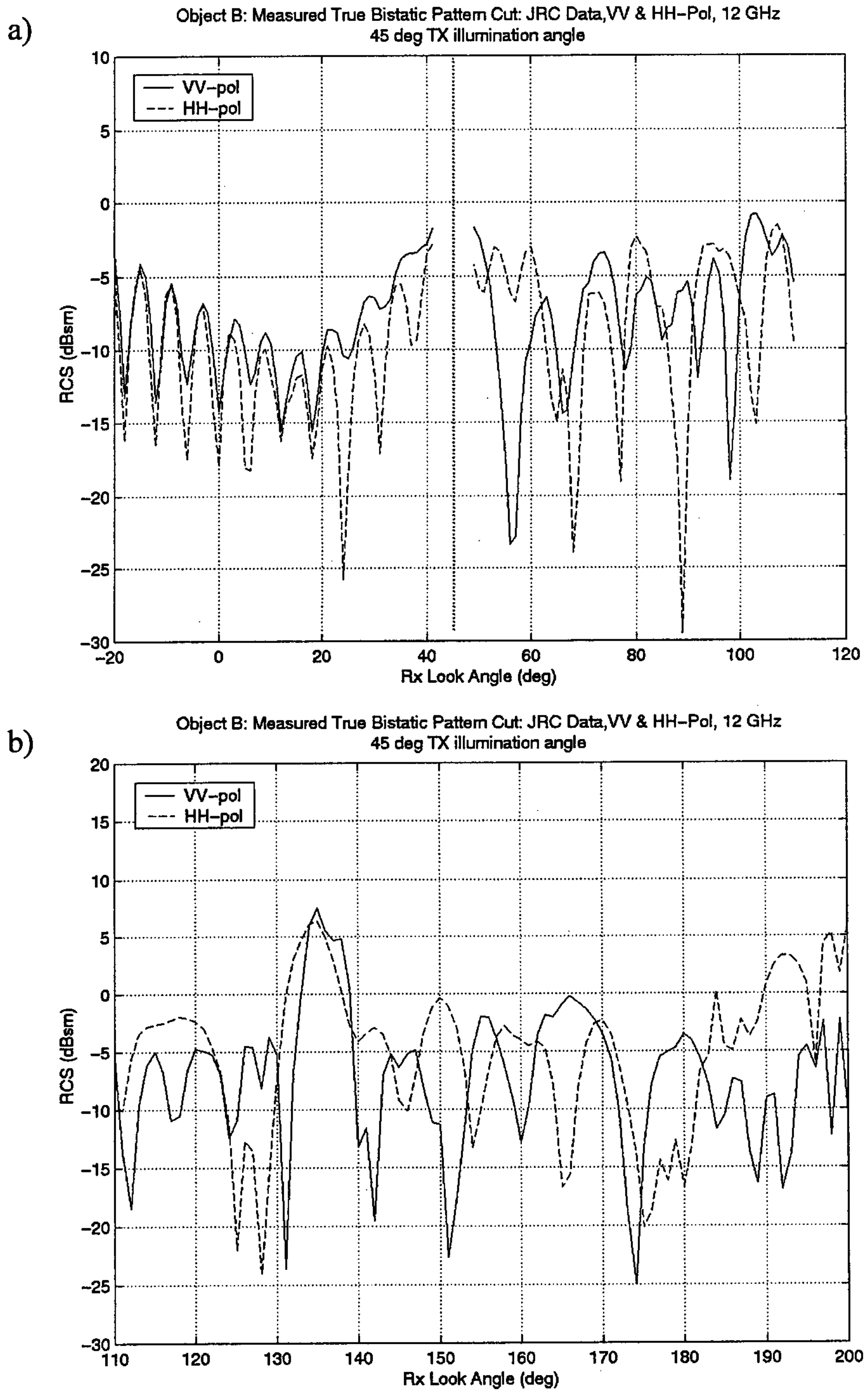


Figure B-6: JRC Measured Bistatic Data for Object B, 12 GHz: a) –20 to +90 degrees Rx look angle , b) +90 to +200 degrees Rx look angle

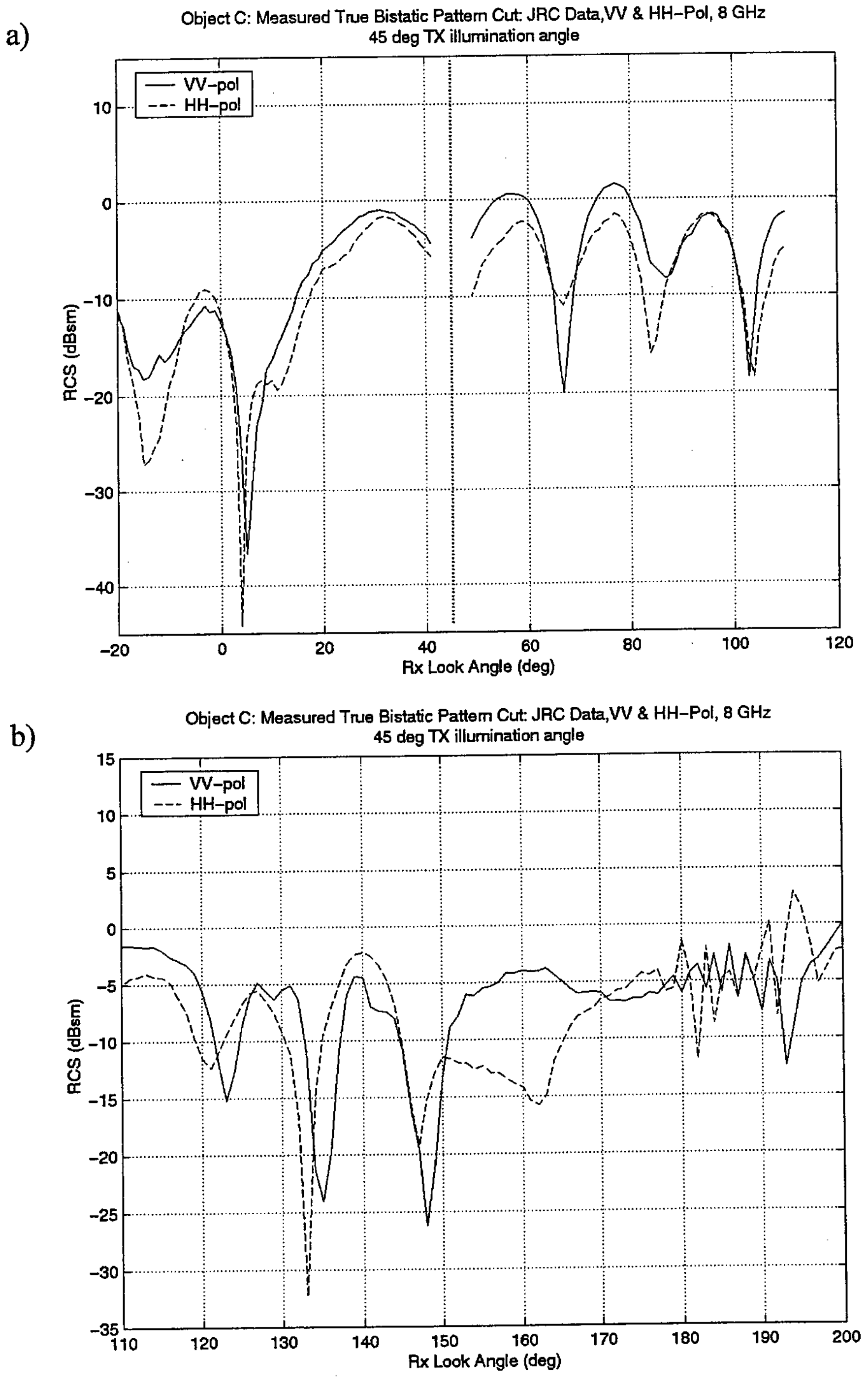


Figure B-7: JRC Measured Bistatic Data for Object C, 8 GHz: a) –20 to +90 degrees Rx look angle , b) +90 to +200 degrees Rx look angle

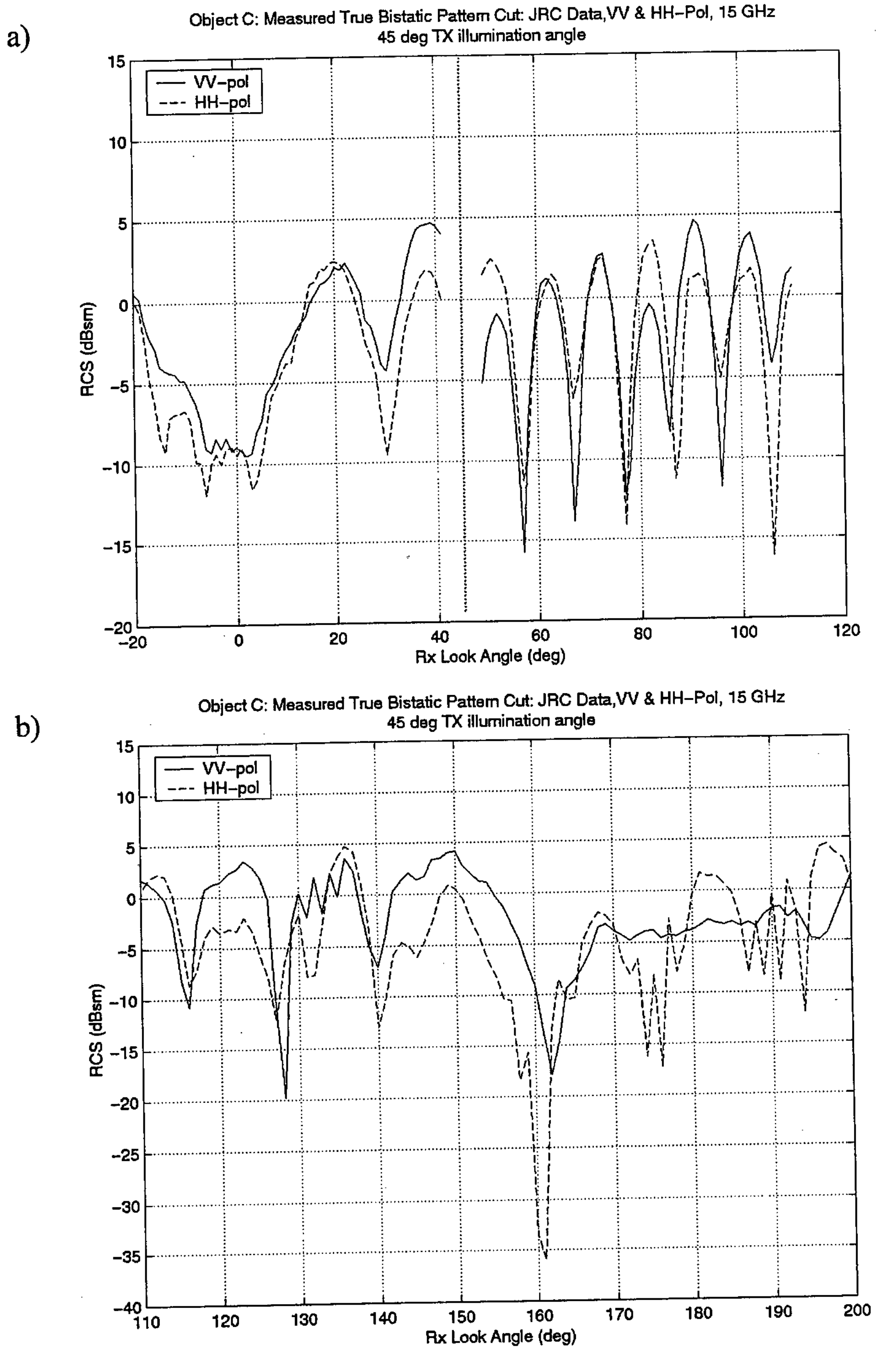


Figure B-8: JRC Measured Bistatic Data for Object C, 15 GHz: a) –20 to +90 degrees Rx look angle , b) +90 to +200 degrees Rx look angle

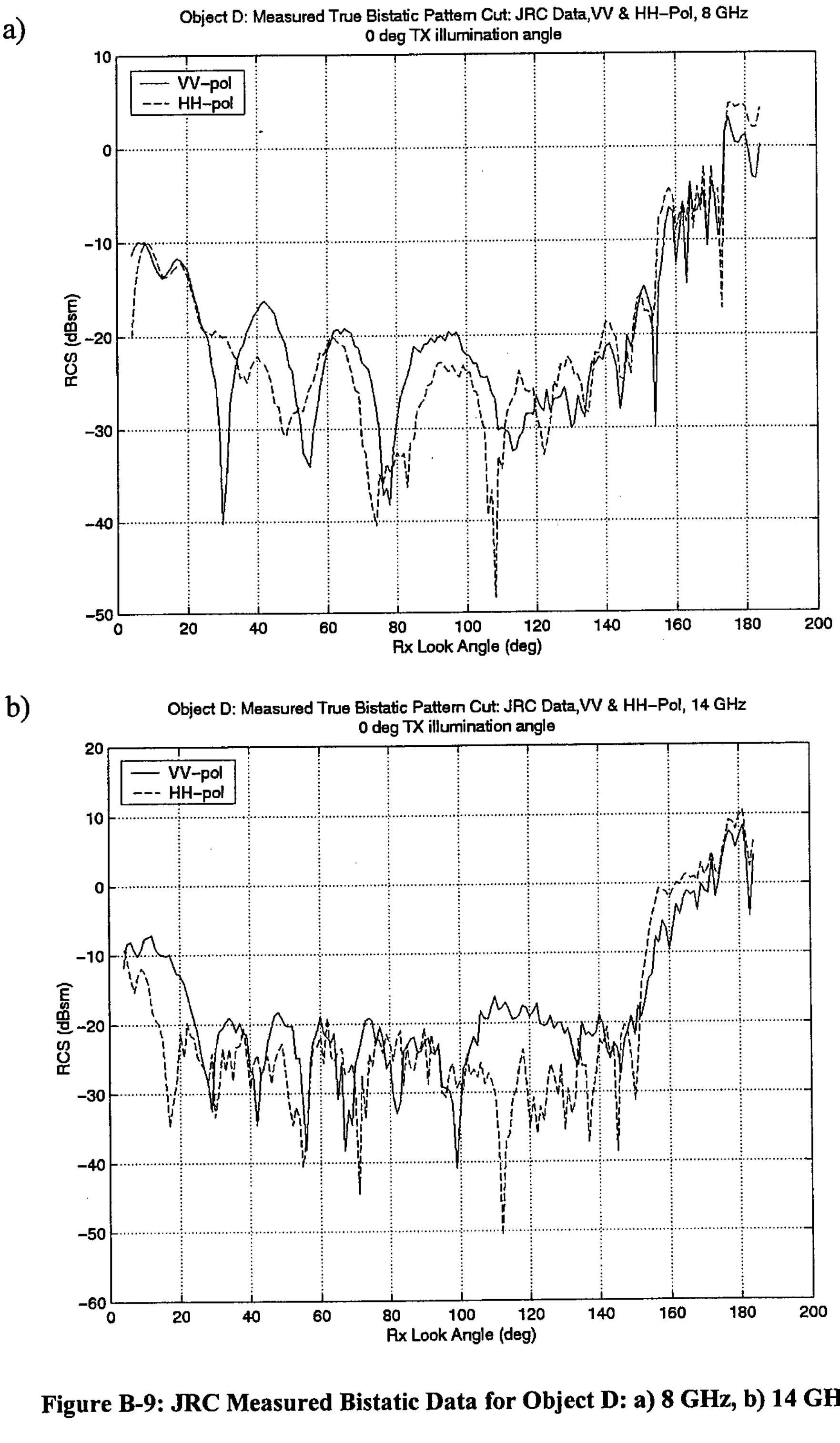


Figure B-9: JRC Measured Bistatic Data for Object D: a) 8 GHz, b) 14 GHz

Appendix C: Matlab Script File Listing

The contents of each Matlab script are shown below in alphabetical order. Documentation accompanies each script, so a complete explanation of a script's function, dependencies, and usage can be found in it's header block. Each script is separated by a lines of * characters.

****

```
% MATLAB FUNCTION:   avg.
% Usage:             avg
% Dependencies             matrix containing angle and RCS amplitude data to which an
%                      averaging window will be applied; various other input variables
%
% Function:  Produces averaged RCS plot using a sliding window technique
%             over an X degree azimuth window width (X is specified by user)
%
% Possible matrices to which a window can be applied; explanation of data
%   contained therein (M-file from which they are generated):
%
% whole_kell_rcs:       Kells equiv. bistatic RCS computed from MRC data, VV-pol only
data (kell.m)
% whole_jkell_rcs:      Kells equiv. bistatic RCS computed from JRC data (kellj.m)
% sim_rcs:              Kells equiv. bistatic RCS computed from Xpatch data (kell_sim.m)
*** Not implemented as of 23 Apr 99 ***
%
% crispin_rcs_matrix:   Crispins equiv. bistatic RCS computed from MRC data, VV-pol only
data (crisp.m)
% final_jrcs:           Crispins equiv. bistatic RCS computed from JRC data, (crispj.m)
% crisp_sim_rcs:        Crispins equiv. bistatic RCS computed from Xpatch data,
(crisp_sim.m) *** Not implemented as of 23 Apr 99 ***
%
% final_strcs:          True bistatic RCS computed with Xpatch (splott.m)
% final_jtrcs:          True measured bistatic RCS data from JRC (jplott.m)


clear bircs_matrix


current_obj = input('Object: ', 's');

% Obtain matrix to which averaging window will be applied

ok = 0;
while (ok ~= 1)

    bircs_matrix = input('Data matrix to be averaged: ','s');

    if (strcmp(bircs_matrix,'whole_kell_rcs') | strcmp(bircs_matrix,'whole_jkell_rcs') |
strcmp(bircs_matrix,'sim_rcs') | strcmp(bircs_matrix,'crispin_rcs_matrix') |
strcmp(bircs_matrix,'final_jrcs') | strcmp(bircs_matrix,'crisp_sim_rcs') |
strcmp(bircs_matrix,'final_strcs') | strcmp(bircs_matrix,'final_jtrcs'))

        ok = 1;

    else

        disp('Please enter one of the following valid matrix name: ');
        disp('whole_kell_rcs');
        disp('whole_jkell_rcs');
        disp('sim_rcs');
        disp('crispin_rcs_matrix');
        disp('final_jrcs');
        disp('crisp_sim_rcs');
        disp('final_strcs');
        disp('final_jtrcs');

    end
end


% Obtain user-specified window size and check to see if its an odd numeral

ok = 0;
while (ok ~= 1)

    win_size = input('Desired window size (in deg, must be odd number): ');

    if (rem(win_size,2) == 0)
       disp('Please enter a valid window size in degrees (must be an odd number): ');
    else
       ok = 1;
    end

end


if (strcmp(bircs_matrix,'whole_kell_rcs') | strcmp(bircs_matrix,'crispin_rcs_matrix'))
;
else
  plot_pol = input('Plot which polarization? (H, V, B=both): ', 's');
end

plot_num = input('Figure number: ');

cflag1 = input('Plot line 1 type/color: ', 's');
if(plot_pol == 'B')
  cflag2 = input('Plot line 2 type/color: ', 's');
end

% Check to see if polarization input is valid
ok = 0;
while(ok ~= 1)
  if (plot_pol == 'V' | plot_pol == 'H'| plot_pol == 'B')
      ok = 1;
  else
     disp('Please specify valid character for polarization you wish to plot.\n');
     plot_pol = input('Plot which polarization? (H, V, B=both): ','s');
  end
end


if (strcmp(bircs_matrix,'whole_kell_rcs'))

    bircs_matrix = whole_kell_rcs;
    bircs_matrix(:,7) = sqrt(10.^(bircs_matrix(:,6)./20));      % VV-pol RCS field
amplitude (straight units)
    bircs_matrix(:,8) = 0;    % initialize windowed data values

   % Compute windowed data

    indxa = floor(win_size/2);
    indxb = length(bircs_matrix);

   % Compute first and last several windowed field amplitudes for which a complete window
   %  cannot be applied

    for i = 1:indxa

        bircs_matrix(i,8) =
(sum(bircs_matrix(1:floor(win_size/2)+i,7)))/(floor(win_size/2)+i);
```

```
        bircs_matrix(indxb-i+1,8) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,7)))/(floor(win_size/2)+i);

    end


   % Compute remaining windowed field amplitudes for which a complete window can be
applied
   %  to the data

    for i = floor(win_size/2):indxb-floor(win_size/2)-1

        bircs_matrix(i+1,8) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),7)))/win_size;

    end


   % Compute windowed RCS data in dBsm from the new windowed field data

    bircs_matrix(:,9) = 20.*log10((bircs_matrix(:,8)).^2);  % VV-pol RCS


   % Plot generation

    if (any(plot_num))
       figure(plot_num)
    else
       figure(fig_no)
       fig_no=fig_no+1;
    end

    plot(bircs_matrix(:,2),bircs_matrix(:,9), cflag1);
    hold on;
    ylabel('RCS (dBsm)');
    xlabel('Rx Look Angle (deg)');
    grid on;
    title({['Object ',current_obj,': Measured Bistatic Pattern Cut: MRC Data,VV-Pol, ',
num2str(desired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX illumination
angle, ', num2str(win_size),' deg window average applied']});


elseif (strcmp(bircs_matrix,'whole_jkell_rcs'))

    bircs_matrix = whole_jkell_rcs;
    bircs_matrix(:,9) = sqrt(10.^(bircs_matrix(:,7)./20));   % HH-pol RCS field amplitude
(straight units)
    bircs_matrix(:,8) = sqrt(10.^(bircs_matrix(:,6)./20));   % VV-pol RCS field amplitude
(straight units)
    bircs_matrix(:,10:11) = 0;    % initialize windowed data values

   % Compute windowed data

    indxa = floor(win_size/2);
    indxb = length(bircs_matrix);

   % Compute first and last several windowed field amplitudes for which a complete window
   %  cannot be applied

    for i = 1:indxa

        bircs_matrix(i,11) =
(sum(bircs_matrix(1:floor(win_size/2)+i,9)))/(floor(win_size/2)+i);
        bircs_matrix(i,10) =
(sum(bircs_matrix(1:floor(win_size/2)+i,8)))/(floor(win_size/2)+i);
        bircs_matrix(indxb-i+1,11) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,9)))/(floor(win_size/2)+i);
        bircs_matrix(indxb-i+1,10) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,8)))/(floor(win_size/2)+i);

    end
   % Compute remaining windowed field amplitudes for which a complete window can be
applied
   %  to the data

    for i = floor(win_size/2):indxb-floor(win_size/2)-1

        bircs_matrix(i+1,11) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),9)))/win_size;
        bircs_matrix(i+1,10) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),8)))/win_size;

    end


   % Compute windowed RCS data in dBsm from the new windowed field data

    bircs_matrix(:,13) = 20.*log10((bircs_matrix(:,11)).^2);    % HH-pol RCS
    bircs_matrix(:,12) = 20.*log10((bircs_matrix(:,10)).^2);    % VV-pol RCS


   % Plot generation

    if (any(plot_num))
       figure(plot_num)
    else
       figure(fig_no)
       fig_no=fig_no+1;
    end

    switch plot_pol

      case 'V'
         plot(bircs_matrix(:,2),bircs_matrix(:,12), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',current_obj,': Kells Equiv. Bistatic Pattern Cut: JRC Data,VV-
Pol, ', num2str(desired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX
illumination angle, ', num2str(win_size),' deg window average applied']});
      case 'H'
         plot(bircs_matrix(:,2),bircs_matrix(:,13), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',current_obj,': Kells Equiv. Bistatic Pattern Cut: JRC Data,HH-
Pol, ', num2str(desired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX
illumination angle, ', num2str(win_size),' deg window average applied']});
      case 'B'
         plot(bircs_matrix(:,2),bircs_matrix(:,12), cflag1);
         hold on;
         plot(bircs_matrix(:,2),bircs_matrix(:,13), cflag2)
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',current_obj,': Kells Equiv. Bistatic Pattern Cut: JRC Data,VV
& HH-Pol, ', num2str(desired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX
illumination angle, ', num2str(win_size),' deg window average applied']});
         legend('VV-pol','HH-pol');
    end


elseif (strcmp(bircs_matrix,'crispin_rcs_matrix'))

    bircs_matrix = crispin_rcs_matrix;
    bircs_matrix(:,4) = sqrt(10.^(bircs_matrix(:,3)./20));     % VV-pol RCS field
amplitude (straight units)
    bircs_matrix(:,5) = 0;   % initialize windowed data values
```

```
  % Compute windowed data

   indxa = floor(win_size/2);
   indxb = length(bircs_matrix);

  % Compute first and last several windowed field amplitudes for which a complete window
  %  cannot be applied

   for i = 1:indxa

       bircs_matrix(i,5) =
(sum(bircs_matrix(1:floor(win_size/2)+i,4)))/(floor(win_size/2)+i);
       bircs_matrix(indxb-i+1,5) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,4)))/(floor(win_size/2)+i);

   end


  % Compute remaining windowed field amplitudes for which a complete window can be
applied
  %  to the data

   for i = floor(win_size/2):indxb-floor(win_size/2)-1

       bircs_matrix(i+1,5) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),4)))/win_size;

   end


  % Compute windowed RCS data in dBsm from the new windowed field data

   bircs_matrix(:,6) = 20.*log10((bircs_matrix(:,5)).^2);  % VV-pol RCS


  % Plot generation

   if (any(plot_num))
      figure(plot_num)
   else
      figure(fig_no)
      fig_no=fig_no+1;
   end

   plot(bircs_matrix(:,2),bircs_matrix(:,6), cflag1);
   hold on;
   ylabel('RCS (dBsm)');
   xlabel('Rx Look Angle (deg)');
   grid on;
   title({['Object ',current_obj,': Crispins Equiv. Bistatic Pattern Cut: MRC Data,VV-
Pol, ', num2str(desired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX
illumination angle, ', num2str(win_size),' deg window average applied']});


elseif (strcmp(bircs_matrix,'final_jrcs'))

   bircs_matrix = final_jrcs;
   bircs_matrix(:,8) = sqrt(10.^(bircs_matrix(:,6)./20));   % HH-pol RCS field amplitude
(straight units)
   bircs_matrix(:,7) = sqrt(10.^(bircs_matrix(:,5)./20));   % VV-pol RCS field amplitude
(straight units)
   bircs_matrix(:,9:10) = 0;   % initialize windowed data values

  % Compute windowed data

   indxa = floor(win_size/2);
   indxb = length(bircs_matrix);

  % Compute first and last several windowed field amplitudes for which a complete window
  %  cannot be applied
```

```
    for i = 1:indxa

        bircs_matrix(i,10) =
(sum(bircs_matrix(1:floor(win_size/2)+i,8)))/(floor(win_size/2)+1);
        bircs_matrix(i,9) =
(sum(bircs_matrix(1:floor(win_size/2)+i,7)))/(floor(win_size/2)+i);
        bircs_matrix(indxb-i+1,10) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,8)))/(floor(win_size/2)+i);
        bircs_matrix(indxb-i+1,9) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,7)))/(floor(win_size/2)+i);

    end

   % Compute remaining windowed field amplitudes for which a complete window can be
applied
   %  to the data

    for i = floor(win_size/2):indxb-floor(win_size/2)-1

        bircs_matrix(i+1,10) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),8)))/win_size;
        bircs_matrix(i+1,9) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),7)))/win_size;

    end


   % Compute windowed RCS data in dBsm from the new windowed field data

    bircs_matrix(:,12) = 20.*log10((bircs_matrix(:,10)).^2);    % HH-pol RCS
    bircs_matrix(:,11) = 20.*log10((bircs_matrix(:,9)).^2);     % VV-pol RCS


   % Plot generation

    if (any(plot_num))
       figure(plot_num)
    else
       figure(fig_no)
       fig_no=fig_no+1;
    end

    switch plot_pol

      case 'V'
         plot(bircs_matrix(:,4),bircs_matrix(:,11), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title([['Object ',current_obj,': Crispins Equiv. Bistatic Pattern Cut: JRC
Data,VV-Pol, ', num2str(jdesired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX
illumination angle, ', num2str(win_size),' deg window average applied']]);
      case 'H'
         plot(bircs_matrix(:,4),bircs_matrix(:,12), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title([['Object ',current_obj,': Crispins Equiv. Bistatic Pattern Cut: JRC
Data,HH-Pol, ', num2str(jdesired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX
illumination angle, ', num2str(win_size),' deg window average applied']]);
      case 'B'
         plot(bircs_matrix(:,4),bircs_matrix(:,11), cflag1);
         hold on;
         plot(bircs_matrix(:,4),bircs_matrix(:,12), cflag2)
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
        title({['Object ',current_obj,': Crispins Equiv. Bistatic Pattern Cut: JRC
Data,VV & HH-Pol, ', num2str(jdesired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg
TX illumination angle, ', num2str(win_size),' deg window average applied']});
        legend('VV-pol','HH-pol');
    end


elseif (strcmp(bircs_matrix,'final_strcs'))

    bircs_matrix = final_strcs;

    %bircs_matrix(:,6) = sqrt(10.^(bircs_matrix(:,3)./20));    % HH-pol RCS field
amplitude (straight units)
    %bircs_matrix(:,5) = sqrt(10.^(bircs_matrix(:,4)./20));    % VV-pol RCS field
amplitude (straight units)
    bircs_matrix(:,7:8) = 0;    % initialize windowed data values

   % Compute windowed data

    indxa = floor(win_size/2);
    indxb = length(bircs_matrix);

   % Compute first and last several windowed field amplitudes for which a complete window
   %  cannot be applied

    for i = 1:indxa

          bircs_matrix(i,8) =
(sum(bircs_matrix(1:floor(win_size/2)+i,6)))/(floor(win_size/2)+i);
          bircs_matrix(i,7) =
(sum(bircs_matrix(1:floor(win_size/2)+i,5)))/(floor(win_size/2)+i);
          bircs_matrix(indxb-i+1,8) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,6)))/(floor(win_size/2)+i);
          bircs_matrix(indxb-i+1,7) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,5)))/(floor(win_size/2)+i);

    end


   % Compute remaining windowed field amplitudes for which a complete window can be
applied
   %  to the data

    for i = floor(win_size/2):indxb-floor(win_size/2)-1

          bircs_matrix(i+1,8) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),6)))/win_size;
          bircs_matrix(i+1,7) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),5)))/win_size;

    end


   % Compute windowed RCS data in dBsm from the new windowed field data

    bircs_matrix(:,10) = 20.*log10((bircs_matrix(:,8)).^2);    % HH-pol RCS
    bircs_matrix(:,9) = 20.*log10((bircs_matrix(:,7)).^2);     % VV-pol RCS


   % Plot generation

    if (any(plot_num))
       figure(plot_num)
    else
       figure(fig_no)
       fig_no=fig_no+1;
    end

    switch plot_pol
```

```
      case 'V'
         plot(bircs_matrix(:,2),bircs_matrix(:,9), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',current_obj,': Xpatch Bistatic Pattern Cut: VV-Pol, ',
num2str(desired_strf),' GHz'],[num2str(tx_illum_angle),' deg TX illumination angle, ',
num2str(win_size),' deg window average applied']});
      case 'H'
         plot(bircs_matrix(:,2),bircs_matrix(:,10), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',current_obj,': Xpatch True Bistatic Pattern Cut: HH-Pol, ',
num2str(desired_strf),' GHz'],[num2str(tx_illum_angle),' deg TX illumination angle, ',
num2str(win_size),' deg window average applied']});
      case 'B'
         plot(bircs_matrix(:,2),bircs_matrix(:,9), cflag1);
         hold on;
         plot(bircs_matrix(:,2),bircs_matrix(:,10), cflag2)
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',current_obj,': Xpatch True Bistatic Pattern Cut: VV & HH-Pol,
', num2str(desired_strf),' GHz'],[num2str(tx_illum_angle),' deg TX illumination angle, ',
num2str(win_size),' deg window average applied']});
         legend('VV-pol','HH-pol');
   end


elseif (strcmp(bircs_matrix,'final_jtrcs'))

   bircs_matrix = final_jtrcs;
   bircs_matrix(:,5) = sqrt(10.^(bircs_matrix(:,3)./20));   % VV-pol RCS field amplitude
(straight units)
   bircs_matrix(:,6) = sqrt(10.^(bircs_matrix(:,4)./20));   % HH-pol RCS field amplitude
(straight units)
   bircs_matrix(:,7:8) = 0;   % initialize windowed data values

   % Compute windowed data

   indxa = floor(win_size/2);
   indxb = length(bircs_matrix);

   % Compute first and last several windowed field amplitudes for which a complete window
   %  cannot be applied

   for i = 1:indxa

       bircs_matrix(i,7) =
(sum(bircs_matrix(1:floor(win_size/2)+i,5)))/(floor(win_size/2)+i);
       bircs_matrix(i,8) =
(sum(bircs_matrix(1:floor(win_size/2)+i,6)))/(floor(win_size/2)+i);
       bircs_matrix(indxb-i+1,7) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,5)))/(floor(win_size/2)+i);
       bircs_matrix(indxb-i+1,8) = (sum(bircs_matrix(indxb-floor(win_size/2)-
i+1:indxb,6)))/(floor(win_size/2)+i);

   end


   % Compute remaining windowed field amplitudes for which a complete window can be
applied
   %  to the data

   for i = floor(win_size/2):indxb-floor(win_size/2)-1

       bircs_matrix(i+1,7) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),5)))/win_size;
```

```
        bircs_matrix(i+1,8) = (sum(bircs_matrix(i+1-
floor(win_size/2):i+1+floor(win_size/2),6)))/win_size;

   end

  % Compute windowed RCS data in dBsm from the new windowed field data

   bircs_matrix(:,9) = 20.*log10((bircs_matrix(:,7)).^2);   % VV-pol RCS
   bircs_matrix(:,10) = 20.*log10((bircs_matrix(:,8)).^2);  % HH-pol RCS


  % Plot generation

   if (any(plot_num))
      figure(plot_num)
   else
      figure(fig_no)
      fig_no=fig_no+1;
   end

   switch plot_pol

     case 'V'
        plot(bircs_matrix(:,2),bircs_matrix(:,9), cflag1);
        hold on;
        ylabel('RCS (dBsm)');
        xlabel('Rx Look Angle (deg)');
        grid on;
        title({['Object ',current_obj,': Measured Bistatic Pattern Cut: JRC Data,VV-Pol,
', num2str(desired_jtrf),' GHz'],['45 deg TX illumination angle. ', num2str(win_size),'
deg window average applied']});
     case 'H'
        plot(bircs_matrix(:,2),bircs_matrix(:,10), cflag1);
        hold on;
        ylabel('RCS (dBsm)');
        xlabel('Rx Look Angle (deg)');
        grid on;
        title({['Object ',current_obj,': Measured True Bistatic Pattern Cut: JRC
Data,HH-Pol, ', num2str(desired_jtrf),' GHz'],['45 deg TX illumination angle, ',
num2str(win_size),' deg window average applied']});
     case 'B'
        plot(bircs_matrix(:,2),bircs_matrix(:,9), cflag1);
        hold on;
        plot(bircs_matrix(:,2),bircs_matrix(:,10), cflag2)
        ylabel('RCS (dBsm)');
        xlabel('Rx Look Angle (deg)');
        grid on;
        title({['Object ',current_obj,': Measured True Bistatic Pattern Cut: JRC Data,VV
& HH-Pol, ', num2str(desired_jtrf),' GHz'],['45 deg TX illumination angle, ',
num2str(win_size),' deg window average applied']});
        legend('VV-pol','HH-pol');
   end

 else
 ;
 end


***********************************************************
```

```
% MATLAB SCRIPT:     crisp.m
% Usage:             crisp
% Dependencies:      load_c.m, load_d.m, or load_e.m
%
% Function: Extracts monostatic data from global matrix "mrcs" and converts it to
bistatic data at
%              at the given RF using Crispin's equivalence theorem ('mrcs' matrix must be
%              loaded first by having load_c.m, load_d.m, or load_e.m scripts available).
%              The data is plotted to the specified figure number.
%
% User inputs:
%  desired_target:  Object ID (C,D or E) for which Crispin's RCS will be generated
%
%  desired_bistatic_rf:  RF at which an equivalent bistatic RCS plot will be generated
from
%                          monostatic data.  Valid RF ranges are
6.0023<=desired_bistatic_rf<=17.9923.
%                          with 0.01 increments; always include 0.0023: Ex: 6.0123 or
9.1123.
%
%   tx_illum_angle = TX illumination angle
%
%   end_angle = last RX postion angle relative to 0 deg reference position at
%                which Crispin's RCS will be computed
%
%   plot_num:  figure number to which data will be plotted; if none given, standard
'fig_no' used
%
%   cflag:  flag indiating line type for plot

clear global crispin_rcs_matrix
clear crispin*
global mrcs crispin_rcs_matrix fig_no mmono_rcs_in_buffer


desired_target = input('Object: ','s');
desired_bistatic_rf = input('Bistatic RF:  ');
tx_illum_angle = input('TX illumunation angle: ');
disp('*** Valid receiver positon angles are: ***');
disp('  Object C: 0 to 180 degrees');
disp('  Object D & E: 0 to 110 deg for TX illumin angles <= 70 deg');
end_angle = input('Ending RX position angle: ');
plot_num = input('Figure number: ');
cflag = input('Plot line type:  ', 's');

mon_rfs = [6.0123:0.01000000000000:18.0023];


% Check to see if proper data set is loaded based on target id criteria; load v-pol data
if necessary

valid_id = 0;
while (valid_id ~= 1)
  if (desired_target == 'C' | desired_target == 'D' | desired_target == 'E')
     if ([size(mrcs,1) ~= 0] & [strcmp(mmono_rcs_in_buffer,desired_target)])
        break;
     elseif (desired_target == 'C')
        disp('Loading Object C monostatic date (MRC)...');
        mrcs = load_c;
        disp('Finished loading Object  data...');
        mmono_rcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'D')
        disp('Loading Object D monostatic data (MRC)...');
        mrcs = load_d;
        disp('Finished loading Object D data...');
        mmono_rcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'E')
        disp('Loading Object E monostatic data (MRC)...');
        mrcs = load_e;
        disp('Finished loading Object E data...');
        mmono_rcs_in_buffer = desired_target;
        valid_id = 1;
      end
   else
      sprintf('%s','Please specify valid Object ID: C, D or E.')
      desired_target = input('Object: ', 's');
   end
end


disp('Computing Crispins equiv. bistatic RCS from MRC data...');

% Establish angle vectors

crispin_angles = [0:1.0:end_angle];
crispin_start_rf_index = 100*(desired_bistatic_rf-mon_rfs(1,1))+1;
start_angle = tx_illum_angle + 1;
stop_angle = tx_illum_angle + end_angle + 1;



% Produce Crispins bistatic RCS vector
%
% This section computes Crispin's bistatic RCS relative to the
%  0 deg azimuth look angle (i.e. for Object C, the flat plate's surface
%  normal points outward at 0 deg azimuth) and the transmitter illumination
%  angle.  In other words, the bistatic RCS computed here, is analagous to
%  a measured bistatic RCS if the receiver were moved in azimuth from
%  0 through 'end_angle' degrees while leaving the
%  transmiter's position fixed at the given illumination angle.
%
% Also note that because MRC's RCS data exists for an angle resolution of 0.5 deg,
%  the TX illumination angle is identical to the position indicator for the
%  first data point to be used as part of Crispin's RCS calculation.  This vastly
%  simplifies the computational burden for extracting Crispin's equivalent
%  bistatic RCS.  For example, say the TX illumination angle is 45 deg.
%  The Crispins equiv. bistatic RCS at the 0 deg AZ position would be the
%  same as the monostatic RCS at half the TX illumination angle (22.5 deg).
%  This position (at 22.5 deg) is the bisector of the bistatic angle (45 deg),
%  and hence, is the angle from which the Crispin's equiv. RCS will be
%  extracted.  The position index (column number) within the "mrcs" matrix
%  of this RCS amplitude data point is 45.


 crispin_rcs_matrix = zeros(size(crispin_angles,2),3);
 crispin_rcs_matrix(:,1) = repmat(desired_bistatic_rf,1,size(crispin_angles,2))';
 crispin_rcs_matrix(:,2) = crispin_angles';
 crispin_rcs_matrix(:,3) = mrcs(crispin_start_rf_index,start_angle:stop_angle)';


 % Plot generation

 if (any(plot_num))
    figure(plot_num)
 else
    figure(fig_no)
    fig_no=fig_no+1;
 end

 plot(crispin_rcs_matrix(:,2),crispin_rcs_matrix(:,3), cflag);
 hold on;
 ylabel('RCS (dBsm)');
 xlabel('Rx Look Angle (deg)');
 grid on;
 title({['Object ',desired_target,', Cripsins Equiv. Bistatic Pattern Cut;
 ',num2str(tx_illum_angle),' deg TX illumination angle']});
 fig_no=fig_no+1;
```

```
%  MATLAB SCRIPT:      crispj.m
%  Usage:              crispj
%  Dependencies:       loadj_a.m or loadj_c.m
%
%  Function:  Extracts monostatic data from global matrix "jrcs" and converts it
%              to bistatic data at the given RF using Crispin's equivalence
%               theorem ("jrcs" matrix must be
%              loaded first by having loadj_a.m or loadj_c.m scripts available).
%              The data is plotted to the specified figure number.
%
%  User inputs:
%   jdesired_target:  Object ID (A or C) for which Crispin's RCS will be
%                     generated
%
%   plot_pol: Variable specifying which polarization to plot
%
%   jdesired_bistatic_rf:  RF at which an equivalent bistatic RCS plot will be
%                          generated from monostatic data.
%                          Valid RF ranges are 7.00<=desired_bistatic_rf<=15.00.
%                          with 0.01 increments.
%
%   start_angle:  Begining receiver azimuth look angle for which RCS mag will
%                  be computed
%
%   end_angle: Ending receiver azimuth look angle for which RCS mag will
%               be computed
%
%   tx_illum_angle:  Desired TX illumination angle
%
%   plot_num:  figure number to which data will be plotted; if none given,
%              standard 'fig_no' used
%
%   cflag:  flag indiating line type for plot

clear global final_jrcs
clear jmono_angles1 jmono_angles2 jmono_angles
clear jcrispin_angles1 jcrispin_angles2 jcrispin_angles
clear jcrispin_start_rf jchosen_rf final_jrf_vector
global jrcs final_jrcs fig_no jmono_rcs_in_buffer


% The following variables are defined above

jdesired_target = input('Object: ', 's');
jdesired_bistatic_rf = input('Bistatic RF:  ');
plot_pol = input('Plot which polarization? (H, V, B=both): ', 's');
tx_illum_angle = input('TX illumunation angle: ');
disp('*** Valid receiver positon angles are: ***');
disp('  Object A & C: 0 to 180 (deg)');
start_angle = input('Starting RX position angle:  ');
end_angle = input('Ending RX position angle:  ');

plot_num = input('Figure number: ');
cflag1 = input('Plot line 1 type/color:  ', 's');
if(plot_pol == 'B')
  cflag2 = input('Plot line 2 type/color: ', 's');
end



% Validate input variables:
%
% Check to see if desired target type is valid

ok = 0;
while(ok ~= 1)
 if (jdesired_target == 'A' | jdesired_target == 'C')
     ok = 1;
  else
     disp('Please specify valid Object ID.\n');
     jdesired_target = input('Plot data for which target? (A or C): ','s');
  end
end

% Check to see if polarization input is valid
ok = 0;
while(ok ~= 1)
  if (plot_pol == 'V' | plot_pol == 'H'| plot_pol == 'B')
      ok = 1;
  else
     disp('Please specify valid character for polarization you wish to plot.\n');
     plot_pol = input('Plot which polarization? (H, V, B=both): ','s');
  end
end

% Check to see if RF input is valid
ok = 0;
while(ok ~= 1)
  if (jdesired_bistatic_rf >= 7.0 | jdesired_bistatic_rf <= 15.0)
      ok = 1;
  else
     disp('Please specify valid RF.\n');
     jdesired_bistatic_rf = input('Plot which RF? (7.0 - 15.0 GHz): ','s');
  end
end

% Check to see if receiver position angles are valid

ok = 0;
while(ok ~= 1)
   if (start_angle >= 0 & end_angle <= 180.0)
         ok = 1;
   else
      disp('Please specify valid begining/ending receiver position angle.');
       start_angle = input('Start angle? (0 to +180 deg): ');
       end_angle = input('End angle? (0 to +180 deg): ');
   end
end



% Check to see if proper data set is loaded based on target id criteria; load VV & HH-pol
data if necessary

if ([size(jrcs,1) ~= 0] & [strcmp(jmono_rcs_in_buffer,jdesired_target)] )
        ;
elseif (jdesired_target == 'A')
        disp('Loading Object A monostatic data (JRC)...');
        jrcs = loadj_a;
        disp('Finished loading Object A data...');
        jmono_rcs_in_buffer = jdesired_target;
elseif (jdesired_target == 'C')
        disp('Loading Object C monostatic data (JRC)...');
        jrcs = loadj_c;
        disp('Finished loading Object C data...');
        jmono_rcs_in_buffer = jdesired_target;
end


% Compute Crispin's approximate RCS
%  Store Crispin's equivalent bistatic data taken from jrcs matrix
%   into "final_jrcs" matrix; the data therein is as follows:
%
%  Column:  Data Description:
%   1        Monostatic RFs corresponding to equivalent bistatic RF at
%             each azimuth look angle (these should all be the same value)
%   2        Monostatic azimuth look angles relative to 0 deg azimuth (extracted
%             from the "jrcs" matrix
%   3        Monostatic azimuth look angles relative to 0 deg azimuth (extracted
%             from the "jmon_angles" matrix
```

```
%
%   ** Note: cols 2 & 3 are redundant; they are provided as a check between
%             the angle data pulled from the "jrcs" matrix and the known correct
%             angle data contined in the "jmono_angles" vector (these cols should
%             have identical numbers)
%
%   4        Equivalent RX look angles to be used for bistatic plot generation
%   5        VV-pol RCS amplitude data
%   6        HH-pol RCS amplitude data
%   7        RF & RCS amplitude position indicator (within jrcs matrix; not
%             normally computed/stored, but useful for trouble shooting if needed)

disp('Computing Crispins equiv. bistatic RCS from JRC data...');
jmon_rfs = [7.00:0.01000000000000:15.00];


% Compute monostatic angles whose corresponding RCS amplitude data will be used
%  as the Crispin's equiv. bistatic RCS (this vector is dependent on the given
%  starting/ending RX position angles)

jmono_angles1 = [ceil((tx_illum_angle-start_angle)/2):1.0:tx_illum_angle-1];
jmono_angles2 = [tx_illum_angle:1.0:tx_illum_angle+floor((end_angle-tx_illum_angle)/2)];
jmono_angles = cat(2,jmono_angles1,jmono_angles2);


% Compute equivalent bistatic angles from the monostatic angles (these should occur
%  at 2 deg increments away from the TX illumination and be bounded by the given
%  starting/ending RX position angles

jcrispin_angles1 = (tx_illum_angle-2.*abs(jmono_angles1(1,:)-tx_illum_angle));
jcrispin_angles2 = (tx_illum_angle+2.*abs(jmono_angles2(1,:)-tx_illum_angle));
jcrispin_angles = cat(2,jcrispin_angles1,jcrispin_angles2);
jcrispin_start_rf = 100*(jdesired_bistatic_rf-jmon_rfs(1,1))+1;


% Find final rcs matrix

jchosen_rf = jdesired_bistatic_rf;
final_jrf_vector = repmat(jchosen_rf,1,length(jcrispin_angles));

e = 1;
index_d = length(final_jrf_vector);
index_e = length(jrcs);
counter = 0;
final_jrcs = zeros(index_d,6);
for d = 1:index_d
   counter = 0;
   while(counter ~= 1)
      if ([abs(jrcs(e,1)-final_jrf_vector(1,d)) <= 0.00001] & [abs(jrcs(e,2) -
jmono_angles(1,d)) <= 0.00001] )

          final_jrcs(d,1) = jrcs(e,1);            % RF data
          final_jrcs(d,2) = jrcs(e,2);            % monostatic AZ look angle from "jrcs"
matrix
          final_jrcs(d,3) = jmono_angles(1,d);    % monostatic AZ look angle from
"jmono_angles" vector
          final_jrcs(d,4) = jcrispin_angles(1,d); % RX look angles
          final_jrcs(d,5) = jrcs(e,5);            % VV-pol RCS data (dBsm)
          final_jrcs(d,6) = jrcs(e,3);            % HH-pol RCS data (dBsm)
          %final_jrcs(d,7) = e;
          counter = 1;
          e = e+1;

      else
          e = e+1;
      end
   end
end


% Plot generation
```

```
if (any(plot_num))
   figure(plot_num)
else
   figure(fig_no)
   fig_no=fig_no+1;
end


switch plot_pol

  case 'V'
         plot(final_jrcs(:,4),final_jrcs(:,5), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',jdesired_target,': Crispins Equiv. Bistatic Pattern Cut: JRC
Data,VV-Pol, ', num2str(jdesired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX
illumination angle']});
  case 'H'
         plot(final_jrcs(:,4),final_jrcs(:,6), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',jdesired_target,': Crispins Equiv. Bistatic Pattern Cut: JRC
Data,HH-Pol, ', num2str(jdesired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg TX
illumination angle']});
  case 'B'
         plot(final_jrcs(:,4),final_jrcs(:,5), cflag1);
         hold on;
         plot(final_jrcs(:,4),final_jrcs(:,6), cflag2)
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',jdesired_target,': Crispins Equiv. Bistatic Pattern Cut: JRC
Data,VV & HH-Pol, ', num2str(jdesired_bistatic_rf),' GHz'],[num2str(tx_illum_angle),' deg
TX illumination angle']});
         legend('VV-pol','HH-pol');
end

fig_no=fig_no+1;


************************************************************


%  MATLAB SCRIPT:   fd.m
%  Usage:           fd
%  Dependencies:    2 matrices containing angle and RCS amplitude data for the plots
%                    under consideration; one matrix must be the true bistatic
%                    measurement data collected by the JRC, the other should be
%                    the "test" matrix which will be compared to this "true" data
%
%  Function:  Extracts RCS amplitude and RX look angle info. from given matrices
%              and finds the mean and standard deviation of the "test" data
%              from the true bistatic signature data. Only dBsm units are computed,
%              but lines computing the difference in straight units are included
%              in the script for additional functionality.  They  happen
%              to be commented-out right now.
%
%
%  User inputs:
%
%   test_matrix:  Matrix containing simulated or otherwise computed bistatic data
%                  which will be compared to the measured data
%
%   plot_pol:  Polarization of difference vectors
%
```

```
%   plot_num:  figure number to which difference (mean) data will be plotted;
%               if none given, standard 'fig_no' used
%
%   cflag1 & 2:  flag indiating line type for plot
%
%
% ****  Before using this script, be sure to run jplott.m and    ****
% ****  whatever script will produce the desired bistatic RCS    ****
% ****  which will be compared to it.                            ****


clear diff_vector
global diff_vector


test_matrix = input('Test matrix: ','s');
if (length(test_matrix) ~= length('whole_kell_rcs'))
  plot_pol = input('Plot which polarization? (H, V, B=both): ', 's');
end
final_jtrcs2 = final_jtrcs;
plot_num = input('Figure number: ');
cflag1 = input('Line color: ', 's');
if(plot_pol == 'B')
  cflag2 = input('Line color: ', 's');
end


final_jtrcs2(:,5) = sqrt(10.^(final_jtrcs2(:,3)./20));    % VV-pol RCS field amplitude
(straight units)
final_jtrcs2(:,6) = sqrt(10.^(final_jtrcs2(:,4)./20));    % HH-pol RCS field amplitude
(straight units)

if (strcmp(test_matrix,'whole_kell_rcs'))

   size1 = length(final_jtrcs2);
   size2 = length(whole_kell_rcs);
   whole_kell_rcs(:,7) = sqrt(10.^(whole_kell_rcs(:,6)./20));


   if (size1 >= size2)

      e = 1;
      counter = 0;
      diff_vector = zeros(size2,4);
      for i = 1:size2
         counter = 0;
         while(counter ~= 1)

          if (final_jtrcs2(e,2)-whole_kell_rcs(i,2) > 0.0001)
                break;
          else

            if ([abs(final_jtrcs2(e,2)-whole_kell_rcs(i,2)) <= 0.00001] & [e <= 180])
               if([final_jtrcs2(e,2) >= 41] & [final_jtrcs2(e,2) <= 49] & [desired_target
== 'C' | desired_target == 'D'])
                     diff_vector(i,1) = final_jtrcs2(e,2);
                     diff_vector(i,2) = whole_kell_rcs(i,2);
                     diff_vector(i,3:5) = NaN;            % No measured RCS data between
41 & 49 deg; no amplitude difference data either
                     %diff_vector(i,4) = d;              % counter
                     %diff_vector(i,5) = e;              % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               else

                    diff_vector(i,1) = final_jtrcs2(e,2);
                    diff_vector(i,2) = whole_kell_rcs(i,2);
%                    diff_vector(i,5) = -(final_jtrcs2(e,5) - whole_kell_rcs(i,7));    %
VV-pol RCS amplitude difference value (straight units)
```

```
                    diff_vector(i,3) = final_jtrcs2(e,3);
                    diff_vector(i,4) = whole_kell_rcs(i,6);
                    diff_vector(i,5) = -(final_jtrcs2(e,3) - whole_kell_rcs(i,6));   %
VV-pol RCS amplitude difference value (dBsm units)

                    %diff_vector(i,6) = d;              % counter
                    %diff_vector(i,7) = e;              % diff_vector row position
indicator
                    counter = 1;
                    e = e+1;
               end

            elseif (e == 181)
               counter = 1;
            else
               e = e+1;
            end
          end
         end
      end


   else

      e = 1;
      counter = 0;
      diff_vector = zeros(size1,3);
      for i = 1:size1
         counter = 0;
         while(counter ~= 1)

          if (final_jtrcs2(e,2)-whole_kell_rcs(i,2) > 0.0001)
                break;
          else

            if (abs(final_jtrcs2(e,2)-whole_kell_rcs(i,2)) <= 0.00001)
               if([final_jtrcs2(e,2) >= 41] & [final_jtrcs2(e,2) <= 49] & [desired_target
== 'C' | desired_target == 'D'])
                     diff_vector(i,1) = final_jtrcs2(e,2);
                     diff_vector(i,2) = whole_kell_rcs(i,2);
                     diff_vector(i,3:5) = NaN;              % No measured RCS data between
41 & 49 deg; no amplitude difference data either
                     %diff_vector(i,4) = d;             % counter
                     %diff_vector(i,5) = e;             % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               else


                    diff_vector(i,1) = final_jtrcs2(e,2);
                    diff_vector(i,2) = whole_kell_rcs(i,2);
%                    diff_vector(i,5) = -(final_jtrcs2(e,5) - whole_kell_rcs(i,7));   %
VV-pol RCS amplitude difference value (straight units)
                    diff_vector(i,3) = final_jtrcs2(e,3);
                    diff_vector(i,4) = whole_kell_rcs(i,6);
                    diff_vector(i,5) = -(final_jtrcs2(e,3) - whole_kell_rcs(i,6));   %
VV-pol RCS amplitude difference value (dBsm units)

                    %diff_vector(i,6) = d;              % counter
                    %diff_vector(i,7) = e;              % diff_vector row position
indicator
                    counter = 1;
                    e = e+1;

               end
            else
               e = e+1;
            end
          end
         end
      end

   end

   figure(plot_num)
   plot(diff_vector(:,2),diff_vector(:,5), cflag1);
   hold on;
%   ylabel('RCS Amplitude Difference (straight units)');
   ylabel('RCS Amplitude Difference (dBsm)');

   xlabel('Rx Look Angle (deg)');
   grid on;
   title({['Object ',desired_target,', Difference Plot: Kells Equiv. Bistatic RCS (JRC) -
Measured Bistatic RCS (JRC)'],['VV-pol, ', num2str(desired_jtrf),' GHz, 45 deg TX
illumination angle']});


elseif (strcmp(test_matrix,'crispin_rcs_matrix'))

   size1 = length(final_jtrcs2);
   size2 = length(crispin_rcs_matrix);
   crispin_rcs_matrix(:,4) = sqrt(10.^(crispin_rcs_matrix(:,3)./20));    % VV-pol
Crispin's RCS field amplitude (straight units)

   if (size1 >= size2)

      e = 1;
      counter = 0;
      diff_vector = zeros(size2,4);
      for i = 1:size2
         counter = 0;
         while(counter ~= 1)

          if (final_jtrcs2(e,2)-crispin_rcs_matrix(i,2) > 0.0001)
                break;
          else

            if ([abs(final_jtrcs2(e,2)-crispin_rcs_matrix(i,2)) <= 0.00001] & [e <= 180])

               if([final_jtrcs2(e,2) >= 41] & [final_jtrcs2(e,2) <= 49] & [desired_target
== 'C' | desired_target == 'D'])
                     diff_vector(i,1) = final_jtrcs2(e,2);
                     diff_vector(i,2) = crispin_rcs_matrix(i,2);
                     diff_vector(i,3:5) = NaN;               % No measured RCS data
between 41 & 49 deg; no amplitude difference data either
                     %diff_vector(i,4) = d;               % counter
                     %diff_vector(i,5) = e;               % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               else

                     diff_vector(i,1) = final_jtrcs2(e,2);         % measured RCS angle
(should be same as column 2 below)
                     diff_vector(i,2) = crispin_rcs_matrix(i,2);   % Crispin's RCS angle
(should be same as column 1 above)
%                     diff_vector(i,5) = -(final_jtrcs2(e,5) - crispin_rcs_matrix(i,4));
% VV-pol RCS amplitude difference value (straight units)
                     diff_vector(i,3) = final_jtrcs2(e,3);
                     diff_vector(i,4) =  crispin_rcs_matrix(i,3);
                     diff_vector(i,5) = -(final_jtrcs2(e,3) - crispin_rcs_matrix(i,3));
% VV-pol RCS amplitude difference value (dBsm units)

                     %diff_vector(i,6) = d;               % counter
                     %diff_vector(i,7) = e;               % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               end

            elseif (e == 181)
```

```
               counter = 1;
            else
               e = e+1;
            end
          end
         end
      end

   else

      e = 1;
      counter = 0;
      diff_vector = zeros(size1,3);
      for i = 1:size1
         counter = 0;
         while(counter ~= 1)

          if (final_jtrcs2(e,2)-crispin_rcs_matrix(i,2) > 0.0001)
                break;
          else

            if (abs(final_jtrcs2(e,2)-crispin_rcs_matrix(i,2)) <= 0.00001)

               if([final_jtrcs2(e,2) >= 41] & [final_jtrcs2(e,2) <= 49] & [desired_target
== 'C' | desired_target == 'D'])
                     diff_vector(i,1) = final_jtrcs2(e,2);
                     diff_vector(i,2) = crispin_rcs_matrix(i,2);
                     diff_vector(i,3:5) = NaN;             % No measured RCS data between
41 & 49 deg; no amplitude difference data either
                     %diff_vector(i,4) = d;               % counter
                     %diff_vector(i,5) = e;               % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               else

                    diff_vector(i,1) = final_jtrcs2(e,2);         % measured RCS angle
(should be same as column 2 below)
                    diff_vector(i,2) = crispin_rcs_matrix(i,2);   % Crispin's RCS angle
(should be same as column 1 above)
%                    diff_vector(i,5) = -(final_jtrcs2(e,5) - crispin_rcs_matrix(i,4));
% VV-pol RCS amplitude difference value (straight units)
                     diff_vector(i,3) = final_jtrcs2(e,3);
                     diff_vector(i,4) =  crispin_rcs_matrix(i,3);
                     diff_vector(i,5) = -(final_jtrcs2(e,3) - crispin_rcs_matrix(i,3));
% VV-pol RCS amplitude difference value (dBsm units)

                    %diff_vector(i,6) = d;               % counter
                    %diff_vector(i,7) = e;               % diff_vector row position
indicator
                    counter = 1;
                    e = e+1;
                end

            else
               e = e+1;
            end
          end
         end
      end
   end

   figure(plot_num)
   plot(diff_vector(:,2),diff_vector(:,5), cflag1);
   hold on;
%    ylabel('RCS Amplitude Difference (straight units)');
   ylabel('RCS Amplitude Difference (dBsm)');
   xlabel('Rx Look Angle (deg)');
   grid on;
   title({['Object ',desired_target,', Difference Plot: Crispins Equiv. Bistatic RCS
(JRC) - Measured Bistatic RCS (JRC)'],['VV-pol, ', num2str(desired_jtrf),' GHz, 45 deg TX
illumination angle']});

elseif (strcmp(test_matrix,'whole_jkell_rcs'))

   size1 = length(final_jtrcs2);
   size2 = length(whole_jkell_rcs);
   whole_jkell_rcs(:,8) = sqrt(10.^(whole_jkell_rcs(:,7)./20)); % HH-pol Kell's RCS field
amplitude (straight units)
   whole_jkell_rcs(:,9) = sqrt(10.^(whole_jkell_rcs(:,6)./20)); % VV-pol Kell's RCS field
amplitude (straight units)



   if (size1 >= size2)

      e = 1;
      counter = 0;
      diff_vector = zeros(size2,4);
      for i = 1:size2
         counter = 0;
         while(counter ~= 1)

          if (final_jtrcs2(e,2)-whole_jkell_rcs(i,2) > 0.0001)
                break;
          else

            if ([abs(final_jtrcs2(e,2)-whole_jkell_rcs(i,2)) <= 0.00001] & [e <= 180])

               if([final_jtrcs2(e,2) >= 41] & [final_jtrcs2(e,2) <= 49] & [desired_target
== 'C' | desired_target == 'D'])
                     diff_vector(i,1) = final_jtrcs2(e,2);
                     diff_vector(i,2) = whole_jkell_rcs(i,2);
                     diff_vector(i,3:8) = NaN;            % No measured RCS data between
41 & 49 deg; no amplitude difference data either
                     %diff_vector(i,4) = d;              % counter
                     %diff_vector(i,5) = e;              % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               else

                     diff_vector(i,1) = final_jtrcs2(e,2);      % measured RCS angle
(should be same as column 2 below)
                     diff_vector(i,2) = whole_jkell_rcs(i,2);   % Kell's RCS angle
(should be same as column 1 above)
%                     diff_vector(i,3) = -(final_jtrcs2(e,5) - whole_jkell_rcs(i,9));   %
VV-pol RCS amplitude difference value (straight units)
%                     diff_vector(i,8) = -(final_jtrcs2(e,6) - whole_jkell_rcs(i,8));   %
HH-pol RCS amplitude difference value (straight units)
                     diff_vector(i,3) = final_jtrcs2(e,3);
                     diff_vector(i,4) = whole_jkell_rcs(i,6);
                     diff_vector(i,5) = -(final_jtrcs2(e,3) - whole_jkell_rcs(i,6));   %
VV-pol RCS amplitude difference value (dBsm units)
                     diff_vector(i,6) = final_jtrcs2(e,4);
                     diff_vector(i,7) = whole_jkell_rcs(i,7);
                     diff_vector(i,8) = -(final_jtrcs2(e,4) - whole_jkell_rcs(i,7));   %
HH-pol RCS amplitude difference value (dBsm units)
                     %diff_vector(i,9) = d;              % counter
                     %diff_vector(i,10) = e;              % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               end

            elseif (e == 181)
               counter = 1;
            else
               e = e+1;
            end
          end
         end
      end

   else

      e = 1;
      counter = 0;
      diff_vector = zeros(size1,3);
      for i = 1:size1
         counter = 0;
         while(counter ~= 1)

          if (final_jtrcs2(e,2)-whole_jkell_rcs(i,2) > 0.0001)
                break;
          else

            if (abs(final_jtrcs2(e,2)-whole_jkell_rcs(i,2)) <= 0.00001)

               if([final_jtrcs2(e,2) >= 41] & [final_jtrcs2(e,2) <= 49] & [desired_target
== 'C' | desired_target == 'D'])
                     diff_vector(i,1) = final_jtrcs2(e,2);
                     diff_vector(i,2) = whole_jkell_rcs(i,2);
                     diff_vector(i,3:8) = NaN;           % No measured RCS data between
41 & 49 deg; no amplitude difference data either
                     %diff_vector(i,4) = d;             % counter
                     %diff_vector(i,5) = e;             % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               else

                     diff_vector(i,1) = final_jtrcs2(e,2);      % measured RCS angle
(should be same as column 2 below)
                     diff_vector(i,2) = whole_jkell_rcs(i,2);   % Kell's RCS angle
(should be same as column 1 above)
%                     diff_vector(i,3) = -(final_jtrcs2(e,5) - whole_jkell_rcs(i,9));   %
VV-pol RCS amplitude difference value (straight units)
%                     diff_vector(i,8) = -(final_jtrcs2(e,6) - whole_jkell_rcs(i,8));   %
HH-pol RCS amplitude difference value (straight units)

                     diff_vector(i,3) = final_jtrcs2(e,3);
                     diff_vector(i,4) = whole_jkell_rcs(i,6);
                     diff_vector(i,5) = -(final_jtrcs2(e,3) - whole_jkell_rcs(i,6));   %
VV-pol RCS amplitude difference value (dBsm units)
                     diff_vector(i,6) = final_jtrcs2(e,4);
                     diff_vector(i,7) = whole_jkell_rcs(i,7);
                     diff_vector(i,8) = -(final_jtrcs2(e,4) - whole_jkell_rcs(i,7));   %
HH-pol RCS amplitude difference value (dBsm units)
                     %diff_vector(i,4) = d;             % counter
                     %diff_vector(i,5) = e;             % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               end

            else
               e = e+1;
            end
          end
         end
      end

   end

   figure(plot_num)

   switch plot_pol

   case 'V'
         plot(diff_vector(:,2),diff_vector(:,5), cflag1);
         hold on;
%         ylabel('RCS Amplitude Difference (straight units)');
         ylabel('RCS Amplitude Difference (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,', Difference Plot: Kells Equiv. Bistatic RCS
(JRC) - Measured Bistatic RCS (JRC)'],['VV-pol, ', num2str(desired_jtrf),' GHz, 45 deg TX
illumination angle']});
   case 'H'
         plot(diff_vector(:,2),diff_vector(:,8), cflag1);
         hold on;
%         ylabel('RCS Amplitude Difference (straight units)');
         ylabel('RCS Amplitude Difference (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,', Difference Plot: Kells Equiv. Bistatic RCS
(JRC) - Measured Bistatic RCS (JRC)'],['HH-pol, ', num2str(desired_jtrf),' GHz, 45 deg TX
illumination angle']});
   case 'B'
         plot(diff_vector(:,2),diff_vector(:,5), cflag1);
         hold on;
         plot(diff_vector(:,2),diff_vector(:,8), cflag2)
%         ylabel('RCS Amplitude Difference (straight units)');
         ylabel('RCS Amplitude Difference (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,', Difference Plot: Kells Equiv. Bistatic RCS
(JRC) - Measured Bistatic RCS (JRC)'],['VV-pol & HH-pol, ', num2str(desired_jtrf),'.0
GHz, 45 deg TX illumination angle']});
         legend('VV-pol','HH-pol');
   end


elseif (strcmp(test_matrix,'final_jrcs'))

   size1 = length(final_jtrcs2);
   size2 = length(final_jrcs);
   final_jrcs(:,7) = sqrt(10.^(final_jrcs(:,5)./20)); % VV-pol Crispin's RCS field
amplitude (straight units)
   final_jrcs(:,8) = sqrt(10.^(final_jrcs(:,6)./20)); % HH-pol Crispin's RCS field
amplitude (straight units)


   if (size1 >= size2)

      e = 1;
      counter = 0;
      diff_vector = zeros(size2,8);
      for i = 1:size2
         counter = 0;
         while(counter ~= 1)

          if (final_jtrcs2(e,2)-final_jrcs(i,4) > 0.0001)
                break;
          else

            if ([abs(final_jtrcs2(e,2)-final_jrcs(i,4)) <= 0.00001] & [e <= 180])

               if([final_jtrcs2(e,2) >= 41] & [final_jtrcs2(e,2) <= 49] & [desired_target
== 'C' | desired_target == 'D'])
                     diff_vector(i,1) = final_jtrcs2(e,2);
                     diff_vector(i,2) = final_jrcs(i,4);
                     diff_vector(i,3:8) = NaN;           % No measured RCS data between
41 & 49 deg; no amplitude difference data either
                     %diff_vector(i,4) = d;              % counter
                     %diff_vector(i,5) = e;              % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               else
```

```
                     diff_vector(i,1) = final_jtrcs2(e,2);      % measured RCS angle
(should be same as column 2 below)
                     diff_vector(i,2) = final_jrcs(i,4);        % Crispin's RCS angle
(should be same as column 1 above)
%                     diff_vector(i,5) = -(final_jtrcs2(e,5) - final_jrcs(i,7));   % VV-
pol RCS amplitude difference value (straight units)
%                     diff_vector(i,8) = -(final_jtrcs2(e,6) - final_jrcs(i,8));   % HH-
pol RCS amplitude difference value (straight units)
                     diff_vector(i,3) = final_jtrcs2(e,3);
                     diff_vector(i,4) = final_jrcs(i,5);
                     diff_vector(i,5) = -(final_jtrcs2(e,3) - final_jrcs(i,5));   % VV-
pol RCS amplitude difference value (dBsm units)
                     diff_vector(i,6) = final_jtrcs2(e,4);
                     diff_vector(i,7) = final_jrcs(i,6);
                     diff_vector(i,8) = -(final_jtrcs2(e,4) - final_jrcs(i,6));   % HH-
pol RCS amplitude difference value (dBsm units)
                     %diff_vector(i,9) = d;             % counter
                     %diff_vector(i,10) = e;            % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               end

            elseif (e == 181)
               counter = 1;
            else
               e = e+1;
            end
          end
         end
      end

   else

      e = 1;
      counter = 0;
      diff_vector = zeros(size1,8);
      for i = 1:size1
         counter = 0;
         while(counter ~= 1)

          if (final_jtrcs2(e,2)-final_jrcs(i,4) > 0.0001)
                break;
          else

            if (abs(final_jtrcs2(e,2)-final_jrcs(i,4)) <= 0.00001)

               if([final_jtrcs2(e,2) >= 41] & [final_jtrcs2(e,2) <= 49] & [desired_target
== 'C' | desired_target == 'D'])
                     diff_vector(i,1) = final_jtrcs2(e,2);
                     diff_vector(i,2) = final_jrcs(i,4);
                     diff_vector(i,3:8) = NaN;              % No measured RCS data
between 41 & 49 deg; no amplitude difference data either
                     %diff_vector(i,4) = d;             % counter
                     %diff_vector(i,5) = e;             % diff_vector row position
indicator
                     counter = 1;
                     e = e+1;
               else

                     diff_vector(i,1) = final_jtrcs2(e,2);      % measured RCS angle
(should be same as column 2 below)
                     diff_vector(i,2) = final_jrcs(i,4);        % Crispin's RCS angle
(should be same as column 1 above)
%                     diff_vector(i,5) = -(final_jtrcs2(e,5) - final_jrcs(i,7));   % VV-
pol RCS amplitude difference value (straight units)
%                     diff_vector(i,8) = -(final_jtrcs2(e,6) - final_jrcs(i,8));   % HH-
pol RCS amplitude difference value (straight units)
                     diff_vector(i,3) = final_jtrcs2(e,3);
                     diff_vector(i,4) = final_jrcs(i,5);
                     diff_vector(i,5) = -(final_jtrcs2(e,3) - final_jrcs(i,5));    % VV-
pol RCS amplitude difference value (dBsm units)
                     diff_vector(i,6) = final_jtrcs2(e,4);
                     diff_vector(i,7) = final_jrcs(i,6);
                     diff_vector(i,8) = -(final_jtrcs2(e,4) - final_jrcs(i,6));    % HH-
pol RCS amplitude difference value (dBsm units)
                     %diff_vector(i,9) = d;              % counter
                     %diff_vector(i,10) = e;             % diff_vector row position
indicator

                     counter = 1;
                     e = e+1;
               end

            else
               e = e+1;
            end
          end
         end
      end

   end

   figure(plot_num)

   switch plot_pol

   case 'V'
         plot(diff_vector(:,2),diff_vector(:,5), cflag1);
         hold on;
%         ylabel('RCS Amplitude Difference (straight units)');
         ylabel('RCS Amplitude Difference (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,',Difference Plot: Crispins Equiv. Bistatic RCS
(JRC) - Measured Bistatic RCS (JRC) '],['VV-pol, ', num2str(desired_jtrf),' GHz, 45 deg
TX illumination angle']});
   case 'H'
         plot(diff_vector(:,2),diff_vector(:,8), cflag1);
         hold on;
%         ylabel('RCS Amplitude Difference (straight units)');
         ylabel('RCS Amplitude Difference (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,',Difference Plot: Crispins Equiv. Bistatic RCS
(JRC) - Measured Bistatic RCS (JRC)'],['HH-pol, ', num2str(desired_jtrf),' GHz, 45 deg TX
illumination angle']});
   case 'B'
         plot(diff_vector(:,2),diff_vector(:,5), cflag1);
         hold on;
         plot(diff_vector(:,2),diff_vector(:,8), cflag2)
%         ylabel('RCS Amplitude Difference (straight units)');
         ylabel('RCS Amplitude Difference (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,',Difference Plot: Crispins Equiv. Bistatic RCS
(JRC) - Measured Bistatic (JRC)'],['VV-pol & HH-pol, ', num2str(desired_jtrf),'.0 GHz, 45
deg TX illumination angle']});
         legend('VV-pol','HH-pol');

   end
else
   ;

end


if ([strcmp(test_matrix,'whole_kell_rcs')] | [strcmp(test_matrix,'crispin_rcs_matrix')] )

    mean_v = nanmean(diff_vector(:,5))    % mean for VV-pol data
    standdev_v = nanstd(diff_vector(:,5))    % sigma for VV-pol data

else

    mean_v = nanmean(diff_vector(:,5))    % mean for VV-pol data
    mean_h = nanmean(diff_vector(:,8))    % mean for HH-pol data


    standdev_v = nanstd(diff_vector(:,5))    % sigma for VV-pol data
    standdev_h = nanstd(diff_vector(:,8))    % sigma for HH-pol data


end


***********************************************************


%  MATLAB FUNCTION:   plotmon.m
%  Usage:             plotmon
%  Dependencies              variaous input variables
%
%  Function:  Loads all Object A or C monostatic RCS & phase (HH & VV-pol), RF, and
azimuth
%             position angle data between 7-15 GHz and 0-180 deg azimuth angle into
matrix "jrcs"
%             Plots specified RCS data to given figure.
%
%  User inputs:
%   desired_target:  Object ID (A or C) for which the monostatic RCS will be generated
%
%   plot_pol:  polarization to plot (H = H-pol, V = V-pol, B = both pols)
%
%   jmon_rf_start:  RF to plot monostatic data for
%
%   start_angle:  Starting azimuth angle (relative to flat plate surface normal at 0 deg)
for plot
%
%   end_angle:  Ending azimuth angle for plot
%
%   plot_num:  figure number to which data will be plotted; if none given, standard
'fig_no' used
%
%   cflag1:  flag indiating line type for graph
%
%   cflag2:  flag indiating line type for second graph if both pols are plotted


clear global final_jrcs
global mrcs jrcs final_jrcs_mon fig_no jmono_rcs_in_buffer

desired_target = input('Object: ', 's');
plot_pol = input('Plot which polarization? (H, V, B=both): ', 's');
jmon_rf_start = input('Specify RF: ');
start_angle = input('Specify start angle (0-180 deg): ');
end_angle = input('Specify end angle (0-180 deg): ');
plot_num = input('Figure number: ');
cflag1 = input('Line 1 color: ', 's');
if(plot_pol == 'B')
  cflag2 = input('Line 2 color: ', 's');
end

% Check to see if input is valid

if (plot_pol == 'V' | plot_pol == 'H'| plot_pol == 'B')
    ;
else
     disp('Please specify valid character for polarization you wish to plot.\n');
     plot_pol = input('Plot which polarization? (H, V, B=both): ');
```

```
end


jmon_rfs = [7.00000:0.0100000:15.00000];
jmon_angles = [start_angle:1.0:end_angle];


% Check to see if proper data set is loaded based on target id criteria; load h&v-pol
data if necessary

valid_id = 0;
while (valid_id ~= 1)
  if (desired_target == 'A' | desired_target == 'C')
     if ([size(jrcs,1) ~= 0] & [jmono_rcs_in_buffer == desired_target])
        break;
     elseif (desired_target == 'A')
        disp('Loading Object A monostatic data (JRC)...');
        jrcs = loadj_a;
        disp('Finished loading Object A data...');
        jmono_rcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'C')
        disp('Loading Object C monostatic data (JRC)...');
        jrcs = loadj_c;
        disp('Finished loading Object C data...');
        jmono_rcs_in_buffer = desired_target;
        valid_id = 1;
     end
  else
     disp('Please specify valid Object ID: A or C.\n');
     desired_target = input('Object: ', 's');
  end
end


mesg1 = 'Finding JRC measured monostatic RCS at ';
mesg2 = ' GHz...';
mesg = cat(2,mesg1,num2str(jmon_rf_start),mesg2);
disp(mesg);


% Load appropriate data from jrcs into individual vectors

jrfs = jrcs(:,1);
j_angles = jrcs(:,2);
vpolrcs = jrcs(:,5);
hpolrcs = jrcs(:,3);


% Find final rcs matrix

[y,jrf_mono_startindex_vector1] = min(abs(jmon_rfs-jmon_rf_start));
jrf_mono_startindex_vector = 801*start_angle + jrf_mono_startindex_vector1;

index_d = size(jmon_angles,2);
counter = 0;
final_jrcs = zeros(index_d,4);
for d = 0:index_d-1
   jrcs_matrix(d+1,1) = jrcs(jrf_mono_startindex_vector+801*d,1);
   jrcs_matrix(d+1,2) = jrcs(jrf_mono_startindex_vector+801*d,2);
   jrcs_matrix(d+1,3) = jrcs(jrf_mono_startindex_vector+801*d,3);  % HH-pol RCS
   jrcs_matrix(d+1,4) = jrcs(jrf_mono_startindex_vector+801*d,5);  % VV-pol RCS
end

[y,jmon_start_angle_index] = min(abs(jrcs_matrix(:,2)-start_angle));
[y,jmon_end_angle_index] = min(abs(jrcs_matrix(:,2)-end_angle));
final_jrcs(:,4) = jrcs_matrix(jmon_start_angle_index:jmon_end_angle_index,1);
final_jrcs(:,3) = jrcs_matrix(jmon_start_angle_index:jmon_end_angle_index,2);
final_jrcs(:,1) = jrcs_matrix(jmon_start_angle_index:jmon_end_angle_index,3);  % HH=pol
RCS
```

```
final_jrcs(:,2) = jrcs_matrix(jmon_start_angle_index:jmon_end_angle_index,4);  % VV-pol
RCS


% Plot generation

if (any(plot_num))
   figure(plot_num)
else
   figure(fig_no)
   fig_no=fig_no+1;
end

switch plot_pol

  case 'V'
         plot(jmon_angles,final_jrcs(:,2), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Azimuth Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,': Measured Monostatic Pattern Cut: JRC Data,V-
Pol, ', num2str(jmon_rf_start),' GHz']});
  case 'H'
         plot(jmon_angles,final_jrcs(:,1), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Azimuth Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,': Measured Monostatic Pattern Cut: JRC Data,H-
Pol, ', num2str(jmon_rf_start),' GHz']});
  case 'B'
         plot(jmon_angles,final_jrcs(:,2), cflag1);
         hold on;
         plot(jmon_angles,final_jrcs(:,1), cflag2);
         ylabel('RCS (dBsm)');
         xlabel('Azimuth Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,': Measured Monostatic Pattern Cut: JRC Data,V
& H-Pol, ', num2str(jmon_rf_start),' GHz']});
         legend('VV-pol','HH-pol');
end

fig_no = fig_no+1;


**********************************************************


%  MATLAB FUNCTION:   jplott.m
%  Usage:             jplott
%  Dependencies              loadtbj_X.m, variaous input variables
%
%  Function:  Loads all Object A, B, C,or D true bistatic RCS, RF, and receiver
%              position angle data between 7-15 GHz,
%              -94 to +86 receiver position angle (deg) for Objects A & B
%              -20 to +41 & +49 to +200 receiver position angle (deg) for
%              Objects C & D.
%              This data is computed from measured I & Q data collected by
%              the JRC in March 1999.
%
%  User inputs:
%   desired_target:  Object ID (A,B,C, or D) for which the true bistatic RCS
%                    will be generated
%
%   desired_jtrf:  Desired RF (7-15 GHz, 0.01 GHz increments are valid)
%
%   plot_pol:  polarization to plot (H = HH-pol, V = VV-pol, B = both pols)
%
%   start_angle:  Starting receiver azimuth position angle for plot.
%                 This angle is relative to
%                   the 0 deg azimuth positon which is defined as either:
%                   1) the outward pointing surface normal of the flat plate for
%                      Objects A & C,
%                   2) the outward pointing surface normal of Object B's
%                      broadside such that one maximizes the viewed area of the
%                      inside of the canted cylinder opening if one is looking at
%                      the cylinder along this surface normal, or
%                   3) the outward pointing surface normal of Object D's base
%                      unit (short dimension) nearest to the short, solid
%                      cylinder
%
% *Note: The transmitter illumination angle for Object A & B is 0 deg, for
%        Object C & D its 45 deg.
% *Note: This angle must lie between 4 to +184 deg for Objects A or B
% *Note: This angle must lie between -20 to +41 or +49 to +200 deg for
%        Objects C or D
%
%   end_angle:  Ending receiver azimuth position angle for plot.
%
%   plot_num:  figure number to which data will be plotted; if none given,
%              standard 'fig_no' used
%
%   cflag1:  flag indiating line type for graph
%
%   cflag2:  flag indiating line type for second graph if both pols are plotted
%
%
%  The loadtbj_X.m files depend on the JRC data formatted into a set of files
%   with another set of matlab scripts.  A brief description is provided below:
%
%   JRC formatted data file (this file produced by executing the
%   "fixcd.m" file on one of the "jtbX_neg110topos110_hv.txt"
%   files where "X" is c or d). Both HH &
%   VV-pol RCS & phase data are contained in this file.
%   See the "fixcd.m" file for a complete column description
%   of the information contained therin. This file is loaded by
%   the "loadtbj_X.m" script ("X" is a,b,c,d) if needed.
%   "loadtbj_X.m" script also contains information on the
%   contents of the formatted JRC data file
%
%  ****  VERY IMPORTANT:  For the purposes of this thesis the polarization ****
%  ****                   orientation is reversed within this script file  ****
%  ****                   (i.e. VV-pol referenced in the "loadtbj_X.m" files **
%  ****                   becomes HH-pol & vice versa).  This is done to   ****
%  ****                   standardize the target measurement plane to the  ****
%  ****                   x-y plane for both monostatic and bistatic analysis *
%  ****                   Monostatic measurements were done in the x-y plane, *
%  ****                   but bistatic measurements were accomplished in   ****
%  ****                   the vertical (x-z) plane.                        ****
%  ****                   Essentially this new polarization orientation    ****
%  ****                   makes the E-field parallel (VV-pol) to the       ****
%  ****                   long edge of Object A, and the cylinders' long   ****
%  ****                   axis for Objects B, C & D.                       ****


clear final_jtrcs
global jtrcs fig_no final_jtrcs jtrcs_in_buffer


% The following variables are defined above

desired_target = input('Object: ', 's');
plot_pol = input('Plot which polarization? (H, V, B=both): ', 's');
desired_jtrf = input('Specify RF: ');

disp('*** Valid receiver positon angles are: ***');
disp('  Object A & B: 4 to 184 (deg)');
disp('  Object C & D: -20 to +41, +49 to +200 (deg)');
start_angle = input('Starting RX position angle:  ');
end_angle = input('Ending RX position angle:  ');
```

```
plot_num = input('Figure number: ');
cflag1 = input('Line 1 color: ', 's');
if(plot_pol == 'B')
  cflag2 = input('Line 2 color: ', 's');
end


% Validate input variables:
%
% Check to see if desired target type is valid
ok = 0;
while(ok ~= 1)
 if (desired_target == 'A' | desired_target == 'B'| desired_target == 'C' |
desired_target == 'D')
     ok = 1;
  else
     disp('Please specify valid Object ID.\n');
     desired_target = input('Plot data for which target? (A, B, C, D): ','s');
  end
end

% Check to see if polarization input is valid
ok = 0;
while(ok ~= 1)
  if (plot_pol == 'V' | plot_pol == 'H'| plot_pol == 'B')
      ok = 1;
  else
     disp('Please specify valid character for polarization you wish to plot.\n');
     plot_pol = input('Plot which polarization? (H, V, B=both): ','s');
  end
end

% Check to see if RF input is valid
ok = 0;
while(ok ~= 1)
  if (desired_jtrf >= 7.0 | desired_jtrf <= 15.0)
      ok = 1;
  else
     disp('Please specify valid RF.\n');
     desired_jtrf = input('Plot which RF? (7.0 - 15.0 GHz): ','s');
  end
end

% Check to see if receiver position angles are valid

if (desired_target == 'A' | desired_target == 'B')
   ok = 0;
   while(ok ~= 1)
     if (start_angle >= 4.0 & end_angle <= 184.0)
         ok = 1;
     else
        disp('Please specify valid begining/ending receiver position angle.');
        start_angle = input('Start angle? (+4 to +184 deg): ');
        end_angle = input('End angle? (+4 to +184 deg): ');
     end
   end
else
   ok = 0;
   while(ok ~= 1)
     if ([start_angle >= -20 & start_angle <= 41.0] | [start_angle >= 49.0 & start_angle
<= 200.0])
         ok = 1;
     else
        disp('Please specify valid begining receiver position angle.');
        start_angle = input('Start angle? (-20 to +41, +49 to 200): ');
     end
   end

   ok = 0;
   while(ok ~= 1)
```

```
     if ([end_angle >= -20.0 & end_angle <= 41.0] | [end_angle >= 49.0 & end_angle <=
200.0])
         ok = 1;
     else
        disp('Please specify valid begining receiver position angle.');
        end_angle = input('Start angle? (-20 to +41, +49 to 200): ');
     end
   end
end


if (desired_target == 'A' | desired_target == 'B')
   chosen_jt_angles = [start_angle:1.0:end_angle];
   final_jt_angles = [start_angle:1.0:end_angle];
else
   if (start_angle <= 41.0 & end_angle >= 49.0)
       chosen_jt_angles1 = [start_angle:1.0:41.0];
       chosen_jt_angles2 = [49.0:1.0:end_angle];
       chosen_jt_angles = cat(2,chosen_jt_angles1,chosen_jt_angles2);
       final_jt_angles = [start_angle:1.0:end_angle];
   else
       chosen_jt_angles1 = [];
       chosen_jt_angles = [start_angle:1.0:end_angle];
       final_jt_angles = [start_angle:1.0:end_angle];
   end
end



% Check to see if proper data set is loaded based on target id criteria; load h&v-pol
data if necessary

if ([size(jtrcs,1) ~= 0] & [strcmp(jtrcs_in_buffer,desired_target)])
        ;
elseif (desired_target == 'A')
        disp('Loading Object A bistatic data (JRC)...');
        jtrcs = loadtbj_a;
        disp('Finished loading Object A data...');
        jtrcs_in_buffer = desired_target;
elseif (desired_target == 'B')
        disp('Loading Object B bistatic data (JRC)...');
        jtrcs = loadtbj_b;
        disp('Finished loading Object B data...');
        jtrcs_in_buffer = desired_target;
elseif (desired_target == 'C')
        disp('Loading Object C bistatic data (JRC)...');
        jtrcs = loadtbj_c;
        disp('Finished loading Object C data...');
        jtrcs_in_buffer = desired_target;
elseif (desired_target == 'D')
        disp('Loading Object D bistatic data (JRC)...');
        jtrcs = loadtbj_d;
        disp('Finished loading Object D data...');
        jtrcs_in_buffer = desired_target;
end

mesg1 = 'Finding JRC measured bistatic RCS at ';
mesg2 = ' GHz...';
mesg = cat(2,mesg1,num2str(desired_jtrf),mesg2);
disp(mesg);

% Find final jtrcs matrix
%  Because data has been collected for different receiver positon angles for
%  each test object, the data loaded in the previous step must be manipulated
%  differently for each object

% Objects A & B

if (desired_target == 'A' | desired_target == 'B')
   e = 1;
   index_d = size(chosen_jt_angles,2);
   index_e = size(jtrcs,1);
   counter = 0;
   temp_jtrcs = zeros(index_d-1,4);
   for d = 1:index_d
      counter = 0;
      while(counter ~= 1)
         if ([abs(jtrcs(e,1)-desired_jtrf) <= 0.00001] & [abs(jtrcs(e,2)-
chosen_jt_angles(1,index_d+1-d)) <= 0.00001])

             temp_jtrcs(index_d+1-d,1) = jtrcs(e,1);    % RF value
             temp_jtrcs(index_d+1-d,2) = jtrcs(e,2);    % RX azimuth look angle (deg)
             temp_jtrcs(index_d+1-d,3) = jtrcs(e,3);    % VV-pol RCS
             temp_jtrcs(index_d+1-d,4) = jtrcs(e,5);    % HH-pol RCS
             %temp_jtrcs(index_d+1-d,5) = d;             % counter
             %temp_jtrcs(index_d+1-d,6) = e;             % jtrcs row position indicator
             counter = 1;
             e = e+1;

         else
             e = e+1;
         end
      end
   end

elseif (desired_target == 'C' | desired_target == 'D')

   e = 1;
   index_d = size(chosen_jt_angles,2);
   index_e = size(jtrcs,1);
   counter = 0;
   temp_jtrcs = zeros(index_d-1,4);
   for d = 1:index_d
      counter = 0;
      while(counter ~= 1)
         if ([abs(jtrcs(e,1)-desired_jtrf) <= 0.00001] & [abs(jtrcs(e,2)-
chosen_jt_angles(1,d)) <= 0.00001])

             temp_jtrcs(d,1) = jtrcs(e,1);     % RF value
             temp_jtrcs(d,2) = jtrcs(e,2);     % RX azimuth look angle (deg)
             temp_jtrcs(d,3) = jtrcs(e,3);     % VV-pol RCS
             temp_jtrcs(d,4) = jtrcs(e,5);     % HH-pol RCS
             %temp_jtrcs(d,5) = d;              % counter
             %temp_jtrcs(d,6) = e;              % jtrcs row position indicator
             counter = 1;
             e = e+1;

         else
             e = e+1;
         end
      end
   end

end


% Expand temp_jtrcs matrix to include +41 to +49 deg RX look angles if necessary
%  This will be used to help highlight this "dead" zone when plotted.

if (desired_target == 'A' | desired_target == 'B' | [desired_target == 'C' &
isempty(chosen_jt_angles1)] | [desired_target == 'D' & isempty(chosen_jt_angles1)])
   final_jtrcs = temp_jtrcs;
else
   if (size(final_jt_angles,2) ~= 0)
       [y,angle1] = min(abs(final_jt_angles - 41));
       [y,angle2] = min(abs(final_jt_angles - 49));
       final_jtrcs = repmat(NaN,size(final_jt_angles,2),4);
       final_jtrcs(1:size(chosen_jt_angles1,2),:) =
temp_jtrcs(1:size(chosen_jt_angles1,2),:);
```

```
       final_jtrcs(angle2:size(final_jtrcs,1),:) =
temp_jtrcs(angle1+1:size(temp_jtrcs,1),:);
       final_jtrcs(:,2) = final_jt_angles';
   else
       final_jtrcs = temp_jtrcs;
   end
end



% Plot generation

if (any(plot_num))
   figure(plot_num)
else
   figure(fig_no)
   fig_no=fig_no+1;
end

switch plot_pol

  case 'V'
         plot(final_jtrcs(:,2),final_jtrcs(:,3), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         if (desired_target == 'A' | desired_target == 'B')
            title({['Object ',desired_target,': Measured True Bistatic Pattern Cut: JRC
Data,VV-Pol, ', num2str(desired_jtrf),' GHz'],['0 deg TX illumination angle']});
         else
            title({['Object ',desired_target,': Measured True Bistatic Pattern Cut: JRC
Data,VV-Pol, ', num2str(desired_jtrf),' GHz'],['45 deg TX illumination angle']});
         end
  case 'H'
         plot(final_jtrcs(:,2),final_jtrcs(:,4), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         if (desired_target == 'A' | desired_target == 'B')
            title({['Object ',desired_target,': Measured True Bistatic Pattern Cut: JRC
Data,HH-Pol, ', num2str(desired_jtrf),' GHz'],['0 deg TX illumination angle']});
         else
            title({['Object ',desired_target,': Measured True Bistatic Pattern Cut: JRC
Data,HH-Pol, ', num2str(desired_jtrf),' GHz'],['45 deg TX illumination angle']});
         end
  case 'B'
         plot(final_jtrcs(:,2),final_jtrcs(:,3), cflag1);
         hold on;
         plot(final_jtrcs(:,2),final_jtrcs(:,4), cflag2)
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         if (desired_target == 'A' | desired_target == 'B')
            title({['Object ',desired_target,': Measured True Bistatic Pattern Cut: JRC
Data,VV & HH-Pol, ', num2str(desired_jtrf),' GHz'],['0 deg TX illumination angle']});
         else
            title({['Object ',desired_target,': Measured True Bistatic Pattern Cut: JRC
Data,VV & HH-Pol, ', num2str(desired_jtrf),' GHz'],['45 deg TX illumination angle']});
         end
         legend('VV-pol','HH-pol');
end

%keyboard;
fig_no = fig_no+1;


************************************************************
```

```
%  MATLAB SCRIPT:    kell.m
%  Usage:            kell
%  Dependencies:     variaous input variables and data files: load_c.m, load_d.m, or
load_e.m
%
%  Function:  Extracts MRC generated monostatic data from global matrix "mrcs" and
converts it to
%              bistatic data at the given RF using Kell's equivalence theorem ("mrcs"
matrix must be
%              loaded first by having load_c.m, load_d.m, or load_e.m scripts available).
%              The data is plotted to the specified figure number.
%
%             This m-file is for use only with MRC collected data (to plot JRC generated
monostatic data use
%              the "kellj.m" m-file; MRC & JRC data are stored in different formats and
thus require the
%              use of two different m-files for data manipulation.
%
%  User inputs:
%   desired_target:  Object ID (C,D or E) for which Kell's RCS will be generated
%
%   desired_bistatic_rf:  RF at which an equivalent bistatic RCS plot will be generated
from
%                         monostatic data.  Valid RF ranges are
6.0023<=desired_bistatic_rf<=17.9923.
%                         with 0.01 increments; always include 0.0023: Ex: 6.0123 or
9.1123.
%
%   tx_illum_angle:  Transmitter illumination angle (azimuth) relative to flat plate
surface normal (0 deg)
%
%   end_angle: RX ending azimuth position angle
%
%   plot_num:  figure number to which data will be plotted; if none given, standard
'fig_no' used
%
%   cflag:  flag indiating line type for plot
%
% The Kell's equivalent bistatic RCS being computed is relative to the
%  0 deg azimuth look angle (i.e. for Object C, the flat plate's surface
%  normal points outward at 0 deg azimuth) and the transmitter illumination
%  angle.  In other words, the bistatic RCS computed here, is analagous to
%  a measured bistatic RCS if the receiver were moved in azimuth from
%  0 through "end_angle" degrees while leaving the transmiter's position
%  fixed at the given illumination angle.
%
%
% Throughout this script references to RF and angle positions are quite common.  The
%  "positions" are correlated to the positions of corresponding RCS amplitudes stored in
%  the MRC data matrix, "mrcs".


clear global whole_kell_rcs
clear whole* bistatic* round* diff* true* temp* shifted*
clear angle_vector rf_index_vector
global mrcs fig_no whole_kell_rcs mrcs_in_buffer


desired_target = input('Object: ', 's');
desired_bistatic_rf = input('Bistatic RF: ');
tx_illum_angle = input('TX illumunation angle: ');
end_angle = input('Ending RX position angle:  ');
plot_num = input('Figure number: ');
cflag = input('Line color: ', 's');

mon_rfs = [6.0123:0.01000000000000:18.0023];
mon_angles = [0:0.5:end_angle];


% Check to see if proper data set is loaded based on target id criteria; load if
necessary
valid_id = 0;
while (valid_id ~= 1)
  if (desired_target == 'C' | desired_target == 'D' | desired_target == 'E')
     if ([size(mrcs,1) ~= 0] & [mrcs_in_buffer == desired_target])
        break;
     elseif (desired_target == 'C')
        disp('Loading Object C monostatic data (MRC)...');
        mrcs = load_c;
        disp('Finished loading Object C data...');
        mrcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'D')
        disp('Loading Object D monostatic data (MRC)...');
        mrcs = load_d;
        disp('Finished loading Object D data...');
        mrcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'E')
        disp('Loading Object E monostatic data (MRC)...');
        mrcs = load_e;
        disp('Finished loading Object E data...');
        mrcs_in_buffer = desired_target;
        valid_id = 1;
     end
  else
     sprintf('%s','Please specify valid Object ID: C, D, or E. ')
     desired_target = input('Object: ', 's');
  end
end


disp('Computing Kells equiv. bistatic RCS from MRC data...');

% Develop "true" bistatic angle vector dependent on 1) desired eqivalent bistatic RF
% and 2) available monostatic RFs for bistatic angles >= the TX illumination angle.

all_bistatic_angles = 2*180/pi.*acos(desired_bistatic_rf./mon_rfs);



% Limit bistatic angle vector elements to those less than given "end_angle"
%  relative to the TX illumin. angle

index_a = size(all_bistatic_angles,2);
p = 1;
for a = 1:index_a
    if (fliplr(all_bistatic_angles(:,a)) <= end_angle &
isreal(fliplr(all_bistatic_angles(:,a))))
       bistatic_angle1(:,p) = all_bistatic_angles(:,a);
       p = p+1;
    else
       ;
    end
end
bistatic_angle1 = nonzeros(bistatic_angle1);
if (bistatic_angle1(1,1) <= 0.001)
    bistatic_angle1(1,1) = 0;
    bistatic_angle1 = nonzeros(bistatic_angle1);
end
size_bi_angles = size(bistatic_angle1,1);
bistatic_angle = zeros(1,size_bi_angles+1)';
bistatic_angle(2:size_bi_angles+1,1) = bistatic_angle1(:,1);
bistatic_angle = bistatic_angle';

% Compute "true" monostatic angles corresponding to the "true" bistatic angles

computed_mono_angle = bistatic_angle./2;
```

```
% Develop vector containing monostatic angles rounded to the nearest 0.5 degree:
%  this is the "approximate" or "rounded" monostatic angle vector. This step is
%  necessitated by the fact that the MRC data is compiled with an angular
%  resolution of 0.5 degrees.  As one might expect there will be duplicate
%  "rounded" monostatic angles upon completion of this operation
%
% Store the difference between each "rounded" and "true" monostatic angle value as a
percent
%  and as a straight number within "diff_angle3" and "true_diff_angles" vectors
respectively.

index_b=size(computed_mono_angle,2);

diff_angle1 = computed_mono_angle - fix(computed_mono_angle);
diff_angle2 = diff_angle1 - 0.5;

for b = 1:index_b

  if (diff_angle2(1,b) <= 0)
     if (abs(diff_angle2(1,b)) >= 0.25)
       rounded_mono_angle(1,b) = computed_mono_angle(1,b) - diff_angle1(1,b);
       diff_angle3(1,b) = abs((computed_mono_angle(1,b) -
rounded_mono_angle(1,b))./computed_mono_angle(1,b));
       true_diff_angles(1,b) = (computed_mono_angle(1,b) - rounded_mono_angle(1,b));
     else
       rounded_mono_angle(1,b) = computed_mono_angle(1,b) - diff_angle1(1,b) + 0.5;
       diff_angle3(1,b) = abs((computed_mono_angle(1,b) -
rounded_mono_angle(1,b))./computed_mono_angle(1,b));
       true_diff_angles(1,b) = (computed_mono_angle(1,b) - rounded_mono_angle(1,b));
     end
  else
     if (diff_angle2(1,b) >= 0.25)
       rounded_mono_angle(1,b) = computed_mono_angle(1,b) - diff_angle1(1,b) + 1.0;
       diff_angle3(1,b) = abs((computed_mono_angle(1,b) -
rounded_mono_angle(1,b))./computed_mono_angle(1,b));
       true_diff_angles(1,b) = (computed_mono_angle(1,b) - rounded_mono_angle(1,b));
     else
       rounded_mono_angle(1,b) = computed_mono_angle(1,b) - diff_angle1(1,b) + 0.5;
       diff_angle3(1,b) = abs((computed_mono_angle(1,b) -
rounded_mono_angle(1,b))./computed_mono_angle(1,b));
       true_diff_angles(1,b) = (computed_mono_angle(1,b) - rounded_mono_angle(1,b));
     end
  end
end


% This section will establish a truly sequential rounded monostatic angle (and RF index
vector) by eliminating
%  duplicate monostatic angles (keep only the angles with the least deviation from the
true monostatic angle and
%  these will have a corresponding RF position index to be stored in "rf_index_vector").


n = index_b;                             % position indicator for the final rf, bistatic
bisector angle, and rcs vectors
m = 1;                                    % position indicator for the
"rounded_mono_angle" and "true_diff_angles" vectors
rf_index_vector = zeros(1,index_b);   % intialize final rf vector
angle_vector = zeros(1,index_b);      % intialize final bistatic bisector angle vector

while(m ~= index_b+1)

 temp_vector1 = zeros(1,15);     % initialize "temp_vectorX"'s; these vector will be used
later in
 temp_vector2 = zeros(1,15);     %  a while loop to store rf position, angle, and
correpsonding
 temp_vector3 = zeros(1,15);     %  "true_diff_angles" values for consecutively identical
"rounded_mono_angle"
 temp_vector4 = zeros(1,15);     %  values
 switch m

% Details:  In this section the program will evaluate the last value in the
"rounded_mono_angle" vector
%  The evaluation process is similar to that outlined in the "Details" section below and
will eventually
%  determine which of the RF and RCS values will become part of the final Kell's
equivalent bistatic
%  RCS matrix for positive bistatic angles ("temp_kell_rcs"):

   case {index_b}

        if (rounded_mono_angle(1,m) == rounded_mono_angle(1,m-1))
             if (abs(true_diff_angles(1,m)) <= abs(true_diff_angles(1,m-1)))
                   angle_vector(1,n+1) = rounded_mono_angle(1,m);
                   rf_index_vector(1,n+1) = m;
                   diff_angle_vector(1,n+1) = true_diff_angles(1,m);
                   break;
             else
                   break;
             end
        else
             angle_vector(1,n) = rounded_mono_angle(1,m);
             rf_index_vector(1,n) = m;
             diff_angle_vector(1,n) = true_diff_angles(1,m);
             %n = n-1;
             break;
        end

   otherwise,

% Details:  In this section, all sequentially identical "rounded_mono_angle" values and
corresponding
%  data are captured into separate vectors for evaluation.
%
% The captured data includes variable values: "m", "true_diff_angle",
"rounded_mono_angle", and
%  "true_diff_angle" which all correspond to a particular "rounded_mono_angle" value
cataloged by
%  the "m" position indicator. All values will be stored in "temp_vectorX" vectors which
are intialized to
%  15x1 "zeros" vectors but will eventually be culled down to vectors containing only
positive
%  values ("temp_vectorX_mod" vectors, temp_matrix).
%
% The evaluation process is as follows and will evetually determine which of the rf and
rcs values will become
%  part of the final Kell's equivalent bistatic RCS matrix ro positive bistatic angles
("temp_kell_rcs"):
%
% Choice of the minimum "true_diff_angle" data value corresponding to angles stored in
%  "rounded_mono_angle" vector among a group of duplicate "rounded" monostatic angles
will
%  determine which "rounded_mono_angle" (as identified by position "m") corresponds
%  to the closest match between the "true" monostatic angle and the "rounded" monostatic
angle.
%  When this minimum is determined the "m" position value will become the position index
for the
%  corresponding RF (stored in "rf_index_vector"), the "rounded_mono_angle" value will
become the bisector of the
%  final bistatic angle ("angle_vector"), and the "true_diff_angle" value will be stored
in a new "diff_angle_vector"
%  which could be used to produce statistical correlation of the true and rounded
monostatic angle values later.

        if (rounded_mono_angle(1,m) == rounded_mono_angle(1,m+1))          % check to see
if begining a section of identical "rounded_mono_angle"
             first_one = rounded_mono_angle(1,m);                          % values; if
so do the following:
             p = 1;
             while(rounded_mono_angle(1,m) == first_one & m ~= index_b)     % Write m,
diff_angle3, rounded_mono_angle, and true_diff_angle data
                  temp_vector1(1,p) = m;                                      %  to
"temp_vectorX" vectors
                  temp_vector2(1,p) = diff_angle3(1,m);
                  temp_vector3(1,p) = rounded_mono_angle(1,m);
                  temp_vector4(1,p) = true_diff_angles(1,m);
                  m = m+1;
                  p = p+1;
             end

             p = 1;
             temp_vector1_mod = nonzeros(temp_vector1);                  % save only nonzero
values of the "temp_vectorX" vectors
             temp_vector2_mod = nonzeros(temp_vector2);
             temp_vector3_mod = nonzeros(temp_vector3);
             temp_vector4_mod = nonzeros(temp_vector4);
             temp_matrix =
[temp_vector1_mod,temp_vector3_mod,temp_vector2_mod,temp_vector4_mod];

             [temp_diff,temp_index] = min(abs(temp_matrix(:,4)));        % choose minimum
"true_diff_angles" value
             angle_vector(1,n) = temp_matrix(temp_index,2);              % store angle
information based on above choice
             rf_index_vector(1,n) = temp_matrix(temp_index,1);           % store rf position
indicator based on above choice
             diff_angle_vector(1,n) = temp_matrix(temp_index,4);         % store difference
between chosen "rounded" monostatic
             n = n-1;                                                    %  angle value and
"true" monostatic angle value

        else
             angle_vector(1,n) = rounded_mono_angle(1,m);                     % if
consecutive values of "rounded_mono_angle" vector
             rf_index_vector(1,n) = m;                                        %  are NOT
identical (i.e. you don't need to determine
             diff_angle_vector(1,n) = true_diff_angles(1,m);                  %  which among
identical "rounded_mono_angle" values
             m = m+1;                                                         %  to use),
just write the "m", "rounded_mono_angle",
             n = n-1;                                                         %  and
difference values to appropriate vectors.
        end
   end
end

angle_vector = nonzeros(fliplr(angle_vector));
rf_index_vector = nonzeros(fliplr(rf_index_vector));
diff_angle_vector = nonzeros(fliplr(diff_angle_vector));


% Add "0" angle (equal to the TX illumination angle) back into angle_vector &
diff_angle_vector

index_c=size(rf_index_vector,1);
temp_angle_vector = zeros(index_c,1);
temp_diff_angle_vector = zeros(index_c,1);

temp_angle_vector(2:index_c,1) = angle_vector(:,1);
temp_diff_angle_vector(2:index_c,1) = diff_angle_vector(:,1);


shifted_angle_vector_pos = temp_angle_vector+tx_illum_angle;
shifted_angle_vector_neg = repmat(tx_illum_angle,size(temp_angle_vector,1)-1,1) -
angle_vector;
temp_rf_index_vector = rf_index_vector+(100*(desired_bistatic_rf-mon_rfs(1,1)));


% Find monostatic angle position index vector
```

```
index_d = size(shifted_angle_vector_pos);
for d=1:index_d
    [y,shifted_angle_index_vector(d,1)] = min(abs(mon_angles-
shifted_angle_vector_pos(d,1)));
end


% Produce temporary vector containing Kells bistatic RCS values for the positive
% bistatic angles

temp_rcs = mrcs(temp_rf_index_vector,shifted_angle_index_vector);
kell_rcs_temp = diag(temp_rcs);


% Store RF, azimuth angle (relative to TX illumination angle) and RCS data in a single
matrix
%
% If one wanted to plot the Kell's equivalent RCS relative to the TX illumination angle
%  (i.e 0 deg azimuth corresponding to the TX illumin. angle), then plot the 2nd and
%  3rd columns from the "temp_kell_rcs" matrix.  Otherwise, continue computations
%  to find bistatic RCS relative to the normal 0 deg azimuth angle (i.e for Object C, the
flat
%  plate's surface normal points outward at 0 deg azimuth). Column description of data
stored in
%  "temp_kell_rcs" matrix:
%
%  Column:  Data Description:
%   1        Monostatic RFs corresponding to equivalent bistatic RF at each azimuth look
angle
%   2        Bistatic RX azimuth look angles relative to 0 deg azimuth
%   3        Monostatic azimuth look angles relative to TX illumin. angle
%   4        RF position indicator relative to "mon_rfs" vector
%   5        Monostatic azimuth look angle postions relative to "mon_angles"
%   6        RCS amplitude corresponding to monostatic RF and angle

indx = 6;
temp_kell_rcs = zeros(size(temp_angle_vector,1),indx);

temp_kell_rcs(:,1) = mon_rfs(temp_rf_index_vector)';
temp_kell_rcs(:,2) = 2*temp_angle_vector+tx_illum_angle;
temp_kell_rcs(:,3) = temp_angle_vector;
temp_kell_rcs(:,4) = temp_rf_index_vector;
temp_kell_rcs(:,5) = shifted_angle_index_vector;
temp_kell_rcs(:,6) = kell_rcs_temp;



if (shifted_angle_vector_neg(1,1) <=0 )

    whole_kell_rcs = temp_kell_rcs;

else

% For TX illumination angles greater than 0, divide all relavent vectors into
%  2 sections: one for "positive" bistatic angles (those larger than the
%  TX illumination angle and one for "negative" bistatic angles (those
%  leass than TX illumination angle).
%
% We've already computed all the proper RF's and the angle/RF position values
%  for the "positive" bistatic angles (in fact we have too many for
%  TX illumin. angles > 0). Now we need similar data for what will
%  become the negative bistatic angles in the Kell's equivalent RCS vector
%
% Find the angles in the shifted_angle_vector_neg vector which can be used as
%  the "negative" bistatic angles (these will have corresponding RF and RCS data which
%  can be used as the negative bistatic angle RF and RCS values; these must
%  be >= 0 but <= TX illumin. angle).
%
% The "negative" angle positions are actually the mirrored positions
%  (image positions) of the "positive" angle positions centered about the TX
%  illumination angle. The RF values associated with the "positive" angles
%  must also be "imaged" to the "negative" angles.  This relationship is captured
%  below such that the first column of "temp_kell_rcs2" matrix contains the
%  "imaged" RF values of the "positive" bistatic angles.
%
% Based on this RF and angle relationship one can compute the necessary RCS position
%  values within the "mrcs" matrix.  The actual angles are stored in the first column
%  and the position values are stored in the second column of "neg_angles" matrix.

num_rcs_data_points = size(kell_rcs_temp,1);
last_neg_angle = tx_illum_angle/2;

counter = 1;
flag = 0;
while(flag == 0)
     if (shifted_angle_vector_neg(counter,1) >= last_neg_angle)
        neg_angles(1,counter) = shifted_angle_vector_neg(counter,1);      % negative
bistatic angle vector
        [y, neg_angles(2,counter)] = min(abs(mon_angles - neg_angles(1,counter)));
        [y, neg_angles(3,counter)] = min(abs(shifted_angle_vector_neg -
neg_angles(1,counter)));
        neg_angles(4,counter) = -angle_vector(counter,1);
        flag = 0;
        counter = counter +1;
     else
        flag = 1;
     end
end

neg_angles(3,:) =  neg_angles(3,:) + 1;


% Image the RF's contained in the "positive" bistatic angle RCS matrix,"temp_kell_rcs"
into the "negative" bistatic angle
%  RCS matrix, "temp_kell_rcs2", 1st column.  Second, 3rd, 4th, 5th col data described
below:
%
%  Column:  Data Description:
%   1        Monostatic RFs corresponding to equivalent bistatic RF at each azimuth look
angle
%   2        Bistatic RX azimuth look angles relative to 0 deg azimuth
%   3        Monostatic azimuth look angles relative to TX illumin. angle
%   4        RF position indicator relative to "mon_rfs" vector
%   5        Monostatic azimuth look angle positions relative to "mon_angles"
%   6        VV-pol RCS amplitude corresponding to monostatic RF and angle


    temp_kell_rcs2 = zeros(size(neg_angles,2),6);

    for index2 = 1:size(neg_angles,2)

       temp_kell_rcs2(index2,1) = temp_kell_rcs(neg_angles(3,index2),1);
       temp_kell_rcs2(index2,2) = tx_illum_angle - 2*abs(neg_angles(4,index2));
       temp_kell_rcs2(index2,3) = neg_angles(4,index2);
       [y,rcs_position_neg] = min(abs(mon_rfs(1,:) - temp_kell_rcs2(index2,1)));
       temp_kell_rcs2(index2,4) = rcs_position_neg;
       temp_kell_rcs2(index2,5) = neg_angles(2,index2);
       temp_kell_rcs2(index2,6) = mrcs(rcs_position_neg,neg_angles(2,index2));

    end

    temp_kell_rcs2 = flipud(temp_kell_rcs2);


% Establish a whole Kell's equivalent RCS matrix encompassing both "positive" and
"negative" bistatic
%  angle data. There will be some rows of data excluded from the "temp_kell_rcs" matrix
for
%  TX illumination angles > 0 in order to retain only bistatic angles <= "end_angle" deg.
```

```
    whole_kell_rcs1 = zeros(size(temp_kell_rcs,1),6);

    size_all = size(temp_kell_rcs,1);
    size_neg = size(temp_kell_rcs2,1);
    size_pos = size(temp_kell_rcs,1) - size_neg;

    whole_kell_rcs1(1:size_neg,:) = temp_kell_rcs2(:,:);
    whole_kell_rcs1(size_neg+1:size_all,:) = temp_kell_rcs(1:size_pos,:);

    [y,last_bi_angle_position] = min(abs(whole_kell_rcs1(:,2)-end_angle));

    whole_kell_rcs = zeros(last_bi_angle_position,6);
    whole_kell_rcs(1:last_bi_angle_position,:) =
whole_kell_rcs1(1:last_bi_angle_position,:);
    whole_kell_rcs(:,7) = sqrt(10.^(whole_kell_rcs(:,6)./20));

end


% Find the mean & standard deviation between the "true" and "approximate" monostatic
angle vectors
% Note: these statistics are only valid for 0 deg TX illumination angle

%mean_angles = mean(final_diff_angle_vector)
%standdev_angles = std(final_diff_angle_vector)


% Plot generation

if (any(plot_num))
   figure(plot_num)
else
   figure(fig_no)
   fig_no=fig_no+1;
end

plot(whole_kell_rcs(:,2),whole_kell_rcs(:,6), cflag);
hold on;
ylabel('RCS (dBsm)');
xlabel('Rx Look Angle (deg)');
grid on;
title({['Object ',desired_target,': Bistatic Pattern Cut; ',num2str(tx_illum_angle),' deg
TX illumination angle']});


************************************************************
```

```
% MATLAB SCRIPT:      plotmon.m
% Usage:              plotmon
% Dependencies               variaous input variables and data files: load_c.m, load_d.m,
load_e.m
%
% Function:  Loads all MRC -generated Object "X" monostatic RCS (VV-pol only), RF, and
azimuth
%            position angle data with given input criteria (RF must be between 6.0023-
18.0023 GHz
%            and  AZ angles between 0-180 deg) into matrix "mrcs" (loading "mrcs"
matrix requires
%            that load_c.m, load_d.m, or load_e.m scripts and the data files they
reference are
%            available).
%
%           This m-file is for use only with MRC collected data (to plot JRC generated
monostatic data use
%            the "jplotmon.m" m-file; MRC & JRC data are stored in different formats
and thus require the
%            use of two different m-files for data manipulation.
%
% User inputs:
%  desired_target:  Object ID (C, D or E) for which the monostatic RCS will be generated
%
%  m_data_file:  Name od data file from whihc monostatic data will be extracted
%
%  plot_pol:  polarization to plot (H = H-pol, V = V-pol, B = both pols)
%
%  chosen_mon_rf:  RF to plot monostatic data for
%
%  start_angle:  Starting azimuth angle (relative to flat plate surface normal at 0 deg)
for plot
%
%  end_angle:  Ending azimuth angle for plot
%
%  plot_num:  figure number to which data will be plotted; if none given, standard
'fig_no' used
%
%  cflag:  flag indiating line type for graph

clear global final_mon_rcs
global mrcs final_mon_rcs fig_no mmono_rcs_in_buffer

desired_target = input('Object: ', 's');
chosen_mon_rf = input('Specify RF: ');
start_angle = input('Specify start angle (0-180 deg): ');
end_angle = input('Specify end angle (0-180 deg): ');
plot_num = input('Figure number: ');
cflag = input('Line color: ', 's');

mon_rfs = [6.0023:0.01:18.0023];
mon_angles = [0.00000:0.50000:180.00000];

chosen_angles = [start_angle:0.500000:end_angle];


% Check to see if proper data set is loaded based on target id criteria; load if
necessary

valid_id = 0;
while (valid_id ~= 1)
  if (desired_target == 'C' | desired_target == 'D'| desired_target == 'E')
     if ([size(mrcs,1) ~= 0] & [mmono_rcs_in_buffer == desired_target])
        break;
     elseif (desired_target == 'C')
        disp('Loading Object C monostatic data (MRC)...');
        mrcs = load_c;
        disp('Finished loading Object C data...');
        mmono_rcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'D')
```

```
        disp('Loading Object D monostatic data (MRC)...');
        mrcs = load_d;
        disp('Finished loading Object D data...');
        mmono_rcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'E')
        disp('Loading Object E monostatic data (MRC)...');
        mrcs = load_e;
        disp('Finished loading Object E data...');
        mmono_rcs_in_buffer = desired_target;
        valid_id = 1;
     end
  else
     disp('Please specify valid Object ID: C,D or E.\n');
     desired_target = input('Object: ', 's');
  end
end


mesg1 = 'Finding MRC measured bistatic RCS at ';
mesg2 = ' GHz...';
mesg = cat(2,mesg1,num2str(chosen_mon_rf),mesg2);
disp(mesg);


% Find final rcs matrix

[y,mon_rf_index] = min(abs(mon_rfs-chosen_mon_rf));
[y,mon_start_angle_index] = min(abs(mon_angles-start_angle));
[y,mon_end_angle_index] = min(abs(mon_angles-end_angle));

mon_rf_index = mon_rf_index-1;

mon_rcs = mrcs(mon_rf_index,:);
final_mon_rcs(:,1) = repmat(chosen_mon_rf,1,size(chosen_angles,2))';
final_mon_rcs(:,2) = chosen_angles';
final_mon_rcs(:,3) = mon_rcs(1,mon_start_angle_index:mon_end_angle_index)';


% Plot generation

if (any(plot_num))
   figure(plot_num)
else
   figure(fig_no)
   fig_no=fig_no+1;
end

plot(final_mon_rcs(:,2),final_mon_rcs(:,3), cflag);
hold on;
ylabel('RCS (dBsm)');
xlabel('AZ Look Angle (deg)');
grid on;
title({['Object ',desired_target,': MRC Monostatic Pattern Cut: V-Pol']});

fig_no = fig_no+1;
```

```
************************************************************
```

```
function [] = printt(figno)
% MATLAB FUNCTION:      printt.m
% Usage:                printt
% Dependencies:         figure number which will be printed specified by
%                        "figno"; output filename sans extension
%
% Function: This function produces postscript and tiff format images of the
%           the figure referenced by the "figno" number.

filename = input('Filename for images: ','s');

figure(figno);
eval(['print -dtiffnocompression /workspace/eigelr/ALL_DATA/PLOT_DATA/', filename,
'.tif;']);
eval(['print -dpsc /workspace/eigelr/ALL_DATA/PLOT_DATA/', filename, '.ps;']);
```

```
***************************************************************
```

```
% MATLAB SCRIPT:     splotmon.m
% Usage:             splotmon
% Dependencies              variaous input variables
%
% Function:  Loads all Object A monostatic RCS & phase (HH & VV-pol), RF, and azimuth
%             position angle data between 6.0023-18.0023 GHz and 0-180 deg azimuth angle
into matrix "srcs"
%
% User inputs:
%  desired_target:  Object ID (C, D or E) for which the monostatic RCS will be generated
%
%  plot_pol:  polarization to plot (H = H-pol, V = V-pol, B = both pols)
%
%  mmon_rf_start:  RF to plot monostatic data for
%
%  start_angle:  Starting azimuth angle (relative to flat plate surface normal at 0 deg)
for plot
%
%  end_angle:  Ending azimuth angle for plot
%
%  plot_num:  figure number to which data will be plotted; if none given, standard
'fig_no' used
%
%  cflag1:  flag indiating line type for graph
%
%  cflag2:  flag indiating line type for second graph if both pols are plotted

clear
global fig_no srcs smon_rcs smon_rfs smon_angles schosen_mon_rf srcs_in_buffer
smon_datafile_in_buffer sim_data_file

desired_target = input('Object: ', 's');
sim_data_file = input('Sim data file: ','s');
chosen_smon_rf = input('Specify RF: ');
start_angle = input('Specify start angle (0-180 deg): ');
end_angle = input('Specify end angle (0-180 deg): ');
plot_num = input('Figure number: ');
cflag = input('Line color: ', 's');

smon_rfs = [6.002:0.01:18.002];
smon_angles = [0.00000:0.50000:180.00000];

chosen_angles = [start_angle:0.500000:end_angle];


% Check to see if proper data set is loaded based on target id criteria; load h&v-pol
data if necessary
```

```
valid_id = 0;
while (valid_id ~= 1)
  if (desired_target == 'A' | desired_target == 'C' | desired_target == 'D'|
desired_target == 'E')
     if ([size(srcs,1) ~= 0] & [srcs_in_buffer == desired_target] &
[smon_datafile_in_buffer == sim_data_file])
        break;
     elseif (desired_target == 'A')
        srcs = loads_a;
        smon_datafile_in_buffer = sim_data_file;
        srcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'C')
        srcs = loads_c;
        smon_datafile_in_buffer = sim_data_file;
        srcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'D')
        srcs = loads_d;
        smon_datafile_in_buffer = sim_data_file;
        srcs_in_buffer = desired_target;
        valid_id = 1;
     elseif (desired_target == 'E')
        srcs = loads_e;
        smon_datafile_in_buffer = sim_data_file;
        srcs_in_buffer = desired_target;
        valid_id = 1;
     end
  else
     disp('Please specify valid Object ID: C,D or E.\n');
     desired_target = input('Object: ', 's');
  end
end


% Find final rcs matrix

e = 1;
index_d = size(chosen_angles,2);
index_e = size(srcs,1);
counter = 0;
final_srcs = zeros(index_d-1,4);
for d = 1:index_d
   counter = 0;
   while(counter ~= 1)
      if ([abs(srcs(e,1)-chosen_smon_rf) <= 0.00001] & [abs(srcs(e,3)-chosen_angles(1,d))
<= 0.00001])

          final_srcs(d,1) = srcs(e,1);
          final_srcs(d,2) = srcs(e,3);
          final_srcs(d,3) = srcs(e,4);
          final_srcs(d,4) = srcs(e,7);
          counter = 1;
          e = e+1;

      else
          e = e+1;
      end
   end
end


% Plot generation

if (any(plot_num))
   figure(plot_num)
else
   figure(fig_no)
   fig_no=fig_no+1;
```

```
end

plot(chosen_angles,final_srcs(:,3), cflag);
hold on;
ylabel('RCS (dBsm)');
xlabel('AZ Look Angle (deg)');
grid on;
title({['Object ',desired_target,': Monostatic Waterline (AZ) Cut,VV-Pol']});

fig_no = fig_no+1;
```

****

```
%  MATLAB FUNCTION:   splott.m
%  Usage:             splott
%  Dependencies                variaous input variables
%
%  Function:  Loads all Object A, C, D, or E true bistatic RCS, RF, and azimuth
%              position angle data between 6-25,35 or 45 GHz and 0-180 deg azimuth angle
into matrix "strcs"
%
%  User inputs:
%   desired_target:  Object ID (A,C,D, or E) for which the true bistatic RCS will be
generated
%
%   st_data_file:  should be of the form "tbX_YY_mod.rcs" where X=a,c,d,e &
YY=00,15,30,45,60 which contains
%                  pertinent data (6-25,30,35,45 GHz, 0-180 deg azimuth data, All-pol
%
%   desired_strf:  Desired rf (8-25,30,35,45)
%
%   plot_pol:  polarization to plot (H = H-pol, V = V-pol, B = both pols)
%
%   desired_strf:  RF to plot bistatic data for
%
%   start_angle:  Starting azimuth angle (relative to flat plate surface normal at 0 deg)
for plot
%
%   end_angle:  Ending azimuth angle for plot
%
%   plot_num:  figure number to which data will be plotted; if none given, standard
'fig_no' used
%
%   cflag1:  flag indiating line type for graph
%
%   cflag2:  flag indiating line type for second graph if both pols are plotted

clear global final_strcs
global strcs fig_no final_strcs strcs_in_buffer st_data_file


% The following variables are defined above

desired_target = input('Object: ', 's');
st_data_file = input('Xpatch data file: ', 's');
desired_strf = input('Specify RF: ');
plot_pol = input('Plot which polarization? (H, V, B=both): ', 's');
tx_illum_angle = input('TX illumunation angle: ');
start_angle = input('Starting RX position angle: ');
end_angle = input('Ending RX position angle: ');
plot_num = input('Figure number: ');
cflag1 = input('Plot line 1 type/color: ', 's');
if(plot_pol == 'B')
  cflag2 = input('Plot line 2 type/color:: ', 's');
end
```

```
% Check to see if polarization input is valid

if (plot_pol == 'V' | plot_pol == 'H'| plot_pol == 'B')
    ;
else
     disp('Please specify valid character for polarization you wish to plot.\n');
     plot_pol = input('Plot which polarization? (H, V, B=both): ','s');
end


chosen_st_angles = [start_angle:1.0:end_angle];


% Check to see if proper data set is loaded based on target id criteria; load h&v-pol
data if necessary

valid_id = 0;
while (valid_id ~= 1)
  if (desired_target == 'A' | desired_target == 'C' | desired_target == 'D' |
desired_target == 'E')
     if ([size(strcs,1) ~= 0] & strcmp(strcs_in_buffer,desired_target) &
strcmp(st_data_file,stfile_in_buffer))
        break;
     else
        mesg1 = 'Loading Object ';
        mesg2 = ' bistatic data (Xpatch)...';
        mesg3 = 'Finished loading Object ';
        mesg4 = ' data...';
        mesgA = cat(2,mesg1,desired_target,mesg2);
        disp(mesgA);
        strcs = loadst;
        mesgB = cat(2,mesg3,desired_target,mesg4);
        disp(mesgB);
        strcs_in_buffer = desired_target;
        stfile_in_buffer = st_data_file;
        valid_id = 1;
     end
  else
     disp('Please specify valid Object ID: A,C,D,or E.\n');
     desired_target = input('Object: ', 's');
  end
end


mesg1 = 'Finding Xpatch computed bistatic RCS at ';
mesg2 = ' GHz, ';
mesg3 = ' TX illumin. angle';
mesg = cat(2,mesg1,num2str(desired_strf),mesg2,num2str(tx_illum_angle),mesg3);
disp(mesg);

% Find final rcs matrix

e = 1;
index_d = size(chosen_st_angles,2);
index_e = size(strcs,1);
counter = 0;
final_strcs = zeros(index_d-1,4);
for d = 1:index_d-1
   counter = 0;
   while(counter ~= 1)
      if ([abs(strcs(e,1)-desired_strf) <= 0.00001] & [abs(strcs(e,3)-
chosen_st_angles(1,d)) <= 0.00001])

          final_strcs(d,1) = strcs(e,1);
          final_strcs(d,2) = strcs(e,3);
          final_strcs(d,3) = strcs(e,4);
          final_strcs(d,4) = strcs(e,7);
          final_strcs(d,5) = sqrt(10.^(final_strcs(d,3)./20));   % V-pol RCS field
amplitude (straight units)
```

```
          final_strcs(d,6) = sqrt(10.^(final_strcs(d,4)./20));   % H-pol RCS field
amplitude (straight units)
          counter = 1;
          e = e+1;

      else
          e = e+1;
      end
   end
end


% Plot generation

if (any(plot_num))
   figure(plot_num)
else
   figure(fig_no)
   fig_no=fig_no+1;
end

switch plot_pol

  case 'V'
         plot(final_strcs(:,2),final_strcs(:,3), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,': Bistatic Pattern Cut: VV-Pol, ',
num2str(desired_strf),'.0 GHz']});
  case 'H'
         plot(final_strcs(:,2),final_strcs(:,4), cflag1);
         hold on;
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,': True Bistatic Pattern Cut: HH-Pol, ',
num2str(desired_strf),'.0 GHz']});
  case 'B'
         plot(final_strcs(:,2),final_strcs(:,3), cflag1);
         hold on;
         plot(final_strcs(:,2),final_strcs(:,4), cflag2)
         ylabel('RCS (dBsm)');
         xlabel('Rx Look Angle (deg)');
         grid on;
         title({['Object ',desired_target,': True Bistatic Pattern Cut: VV & HH-Pol, ',
num2str(desired_strf),'.0 GHz']});
end

fig_no = fig_no+1;
```

Bibliography

1. Alexander, N. T., Currie, N. C., Tuley, M. T. "Calibration of Bistatic RCS Measurements." Proceedings of Antenna and Propagation Techniques Association (AMTA) 1995 Symposium. Columbus, OH: 13-17, November 1995.

2. Alexander, N. T., Currie, N. C., Tuley, M. T. "Unique Calibration Issues for Bistatic Radar Reflectivity Measurements." Proceedings of IEEE National Radar Conference 1996. Ann Arbor, MI: 142-147, May 1996.

3. Bachman, Christian. Radar Targets. Lexington MA: Lexington Books, 1982.

4. Beirly, Peter and Welsh Byron. Radar Cross Section Engineers, Mission Research Corp., Dayton, OH. Personal interview. 12 May 1999.

5. Bhalla, R.,H. Ling, J. Moore, D.J. Andersh, S.W. Lee, and J Hughes. "3D Scattering Center Representation of Complex Targets using the Shooting and Bouncing Ray Technique: A Review," IEEE Antenna and Propagation Magazine, Vol 40 No 5: 30-39. Oct 1998.

6. Bowman, J. J. and others. Electromagnetic and Acoustic Scattering By Simple Shapes. Amsterdam: North-Holland Publishing Co., 1969.

7. Collins, Peter J. Class notes, EENG630, Radar Cross Section and Reduction Techniques. School of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH. Fall 1998.

8. Crispin, J. W. and Siegel, K. M. Methods of Radar Cross Section Analysis. New York: Acedemic Press, 1968.

9. European Microwave Signature Lab (EMSL), Institute for Remote Sensing Applications, Informational Packet as supplied by G. Nesti to Dr. Andrew Terzuoli (AFIT/ENG), Ispra: Italy, Joint Research Center (JRC), 16 November 1998.

10. Glaser, Jerome I. "Some Results in the Bistatic Radar Cross Section (RCS) of Complex Objects," Proceedings of the IEEE, Vol 77, No 5: 639-647 (May 1989).

11. Hastriter, M. L.. Range Dependent Scattering in the Intermediate Zone. MS thesis, AFIT/GE/ENG/97D-14, School of Electrical Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB, OH, December 1997.

12. Hecht, Eugene. Optics. Reading, MA: Addison-Wesley Publishing Co. 1987.

13. Jost, Randy J. and Richard F. Fahlsing. "Bistatic Cross-Polarization Calibration," Proceedings of the 19th Antenna Measurements And Techniques Association (AMTA), 313-318. 1997.

14. Kell, Robert E., "On the Derivation of the Bistatic RCS from Monostatic Measurements," Proceedings of the IEEE, Vol XX No Y: 983-988, Aug 1965.

15. Kent, Brian and William D. Wood, Maj. "The Squat Cylinder and Modified Bicone Primary Static RCS Range Calibration Standards," Proceedings of the 19th Antenna Measurements And Techniques Association (AMTA), 319-324. 1997.

16. Kent, William and Wurst, Lisa. Radar Cross Section Engineers, Mission Research Corp., Dayton, OH. Personal interview. 15 April 1999.

17. Knott, Eugene F. and Senior, Thomas B. Non-Specular Radar Cross Section Study. Technical Report: AFAL-TR-73-2. Ann Arbor: University of Michigan, February 1973 (AD-908-720/XAG).

18. Knott, Eugene F., Shaeffer, John F., Tuly, Michael T. Radar Cross Section, 2nd Edition. Boston: Artech House. 200-217, 1993.

19. Kouyoumijan, Robert G. "Asymptotic High-Frequency Methods," Proceedings of the IEEE, Vol. 52 No Y: 983-988, Aug 1965.

20. Layden, R. G., Wurst, L. A., Blume, R. S., Kent, W. J. Bistatic RCS Predictions, Measurements, and Bistatic Theorems. MRC-R-DN-07-002. Mission Research Corp., Dayton, OH. 61-62, April 1997.

21. MacLennan, J. M. Verification of the Limitations of the Monostatic-Bistatic Radar Cross Section Derived By Kell. MS thesis, AFIT/GE/ENG/88D-24, School of Electrical Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB, OH, December 1988.

22. Moore, Thomas G., Earl C. Burt, and Forrest P Hunsberger. "On The Use of Ray Tracing for Complex Targets", Conference Proceedings of The 10th Annual Review of Progress in Applied Computational Electromagnetics Vol II. 328-334. 1994.

23. Nesti, G. and Mortensen, Henrik B. "Analysis of Polarimetric Scattering From a Wire Mesh." Submitted to: 3emes Journees Internationales de la Polarimetrie Radar, Nantes, France, 21-23 March 1995.

24. Nesti, G. Research Scientist at European Microwave Signature Lab (EMSL), Joint Research Center (JRC), Ispra, Italy. Electronic mail dated 11 May 1999.

25. Rhinoceros: NURBS Modeling for Windows, Version 1.0, Software, Robert McNeel & Associates, Copyright 1993-1998.

26. Trott, K. D. The Disk: A Comparison of Electromagnetic Scattering Solutions and Its Use as a Calibration Standard for Bistatic RCS Measurements. RADC-TR-88-16. Rome Air Development Center: Griffiss AFB, NY. February 1988. (AD-A200 327)

27. Umari, M. H., Ghodgaonkar, D. K., Varadan, V. V., and Varadan, V. K. "A Free-Space Bistatic Calibration Technique for the Measurement of parallel and Perpendicular Reflection Coefficients of Planar Samples." IEEE Transactions on Instrumentation and Measurement, Vol 40, No 1: 19-24, February 1990.

28. Whitt, M. W. and Ulaby, F. T. "A polarimetric radar calibration technique with insensitivity to target orientation." Radio Science, Vol 25, No 6: 1137-1143, Nov-Dec 1990.

29. Whitt, M. W., Ulaby, F. T., Polatin, P., and Liepa, V. V. "A General Polarimetric Radar Calibration Technique." IEEE Transactions on Antennas and Propagation, Vol 39, No 1: 62-67, January 1991.

30. Wiesbeck, W. and Kahny, D. "Single Reference, Three Target Calibration and Error Correction for Monostatic Polarimetric Free Space Measurements," Proceedings of the IEEE, Vol 79, No 10: 1551-1558 (October 1991).

31. Willis, Nicholas J. Bistatic Radar. Boston: Artech House. 145-155, 1991.

32. Xpatch 2.4/FISC 1.0 Users Manuals. Software documentation, Demaco, Inc., Champaign, IL, 1991.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 1999	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE
RADAR CROSS SECTION (RCS) CHARACTERIZATOIN OF COMPLEX OBJECTS

5. FUNDING NUMBERS

6. AUTHOR(S)
Robert L. Eigel, Jr., Capt., USAF
Andrew J. Terzuoli, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2950 P Street
WPAFB, OH
45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/99M

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Enrique Mendez	Fred Beaman
ASC/ENAD	NAIC/TAER
2530 Loop Road West	WPAFB OH
WPAFB, OH	45433
45433-7101	

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Dr. Andrew J. Terzuoli, 937-255-3636 ext 4717/andrew.terzuoli@afit.af.mil

12a. DISTRIBUTION AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This reserach addresses some phenomenological aspects of bistatic scattering from a complex object with an emphasis on specular, shadowing, dihedral, and cavity effects. Five targets are investigated for their simplicity and ability to highlight certain scattering phenomena. Direct measurements of scattered electromagnetic energy and simulation data is gathered for a frequency bandwidth of 6-18 GHz. Both ray tracing and scattering center approaches are used to describe the bistatic mechanisms. An appraisal of the effectiveness and utility of the monostatic-to-bistatic equivalence theorems (Kell's and Crispin's) and several commercial scattering prediction codes is also accomplished. Simulation data is generated from two different electromagnetic scattering prediction codes, Xpatch and FISC. Xpatch is a physical optics (PO) based code while FISC is a more rigorous method of momnet (MOM) bsaed tool. This data is utilized to achieve three objectives: 1) study Kell's and Crispin's monostatic-to-bistatic equivalence theorems (MBET) for a complex object through theoretical derivations and comparison of measured and simulated data sets, 2) evaluate the performance of Xpatch and FISC through direct comparisons to measured data, and 3) gain insight into the bistatic scattering phenomenology while extracting appropriate rules-of-thumb for bistatic scattering predictions. These rules of thumb are proposed to help guide the reader in evaluating the bistatic RCS of complex shapes in general.

14. SUBJECT TERMS
Bistatic, Radar Cross Section, Computational Electromagnetics, Physical Optics, Method of Moments

15. NUMBER OF PAGES
207

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL