Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5310--12-9346

An Improved Empirical Model for Radar Sea Clutter Reflectivity

Vilhelm Gregers-Hansen

AEGIS Coordinator
Radar Division

Rashmi Mital

Radar Analysis Branch
Radar Division

April 27, 2012

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 27-04-2012	**2. REPORT TYPE** Memorandum Report	**3. DATES COVERED** *(From - To)* June 2009 – November 2010

4. TITLE AND SUBTITLE

An Improved Empirical Model for Radar Sea Clutter Reflectivity

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62271N

6. AUTHOR(S)

Vilhelm Gregers-Hansen and Rashmi Mital

5d. PROJECT NUMBER
53-6018-0-1

5e. TASK NUMBER

5f. WORK UNIT NUMBER
53-6018

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/5310--12-9346

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

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The most fundamental characteristic of sea clutter, as used in radar performance evaluation, is its apparent reflectivity, used here as a reminder that any measurement of sea clutter reflectivity includes the effects of propagation and shadowing close to the sea surface. Sea clutter reflectivity depends on many factors including sea state, wind velocity, grazing angle, polarization and radar frequency. A comprehensive tabulation of measurements from approximately 60 sources is included in the 1991 edition of Nathanson's book and probably represents the most complete data base of sea clutter reflectivity available to the radar systems engineer. A widely used sea clutter model proposed by Horst et al. (1978) (GIT model) has found widespread acceptance in the radar community. However, the model does not always agree with the tabulated measured sea clutter data, especially at low sea states. The discrepancies are at times quite large and may lead to overly optimistic radar performance predictions. In this report, a new empirical model for sea clutter reflectivity is presented based entirely on the experimental results presented by Nathanson. This model is defined as a function of frequency, polarization, sea state and grazing angle.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Rashmi Mital
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	Unclassified Unlimited	45	**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified			(202) 767-2584

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

ABSTRACT

The most fundamental characteristic of sea clutter, as used in radar performance evaluation, is its apparent reflectivity defined as $\sigma^0(\frac{m^2}{m^2})$. The qualifier apparent is used here as a reminder that any measurement of sea clutter reflectivity inevitably includes the effects of propagation and shadowing close to the sea surface. Sea cluster reflectivity depends on many factors including sea state, wind velocity, grazing angle, polarization, and radar frequency. A comprehensive tabulation of measurements from approximately 60 sources were included in the 1991 edition of Nathanson's book [1]and this probably represents the most complete database of sea clutter reflectivity available to the radar systems engineer. Also included in the book by Nathanson was a detailed description of an empirical sea clutter model proposed by Horst et. al. [2], the so-called Georgia Institute of Technology (GIT) model. This model has found widespread acceptance in the radar community although its experimental basis appears to be somewhat vague.

As pointed out by Nathanson, his tabulated measured sea clutter data does not always agree with the GIT model, in particular at low Sea States. While this difference qualitatively can be explained by measurement inaccuracies, unknown propagation conditions (such as ducting), and uncertainties in defining the underlying sea state, these discrepancies are at times quite large and may lead to overly optimistic radar performance predictions if the GIT sea clutter model is used. Also, it is not clear that the GIT model was based on better data than that presented by Nathanson.

In this report, a new empirical model for sea clutter reflectivity is presented based entirely on the experimental results presented by Nathanson. The model is defined as a function of radar frequency, polarization, sea state, and grazing angle. The functional form of this empirical equation was chosen such that the average absolute deviation in dB, between the model and the experimental data tabulated by Nathanson, is minimized for grazing angles up to 30degrees. Subsequently, we shall refer to this model as the NRL Sea Clutter Model.

1 INTRODUCTION

Ideally, an accurate model of sea clutter must include both its temporal and spatial characteristics and may ultimately require a sophisticated probabilistic description [3]. The most fundamental characteristic of sea clutter, however, is its average reflectivity defined in the dimensionless unit of square meters of radar cross section per square-meter of locally horizontal surface area illuminated by the radar, often denoted by σ^0. Any other more elaborate probabilistic model must be constrained to match the average level of sea clutter determined by the radar parameters andσ^0. The numeric value of the sea clutter reflectivity is a function of many parameters such as sea state, grazing angle, polarization, radar frequency, propagation condition, and wind direction relative to the radar look angle. Since the early days of radar hundreds of measurement campaigns have been conducted with numerous papers published and summarized in books.

For the average sea clutter reflectivity, a critical review of a large body of data was included in the 1991 edition of the book by Nathanson [1], the form of seven tables showing reflectivity versus sea state,

Manuscript approved June 6, 2011.

frequency, polarization, and grazing angles up to 30degrees. These tables were first published in Nathanson's original book [4] but were extensively updated for the 2nd edition. For the radar systems engineer, these tables provide invaluable reference data often used as the basis for radar specifications and performance prediction. Often, however, a particular radar application involves parameters sets that are not easily obtained from these tables by interpolating between adjacent values. In addition, some amount of smoothing of unavoidable experimental errors would be desirable.

This leads to the desire for an empirical model, which while validated by the experimental data, allows computations to be performed over a continuum of parameter values. One such model, developed at the Georgia Institute of Technology around 1978 [2] has received widespread acceptance and was described in detail in Nathanson's book. Other sea clutter reflectivity models have been proposed as will be discussed in Section 3.

2 THE GIT MODEL AND NATHANSON'S TABLES

A direct comparison between the GIT model and Nathanson's tables does not appear to have been addressed anywhere, at least in the open literature. One problem encountered when attempting to compare the GIT model with the tables in [1] is that the GIT model does not use sea state as an input although it is almost universally used in radar system specification and analysis. The GIT model instead uses as inputs the average wave height and the average wind velocity. For a fully risen sea, these quantities are stated to be related by:

$$v_w = 8.67 \times h_{av}^{0.4} \qquad (1)$$

The GIT model specifically allows for separately specified inputs of average wave height and wind velocity in order to account for the effect of sea clutter reflectivity of rising or falling seas, but the experimental basis for this generalization was not included in the paper.

Wave height is more commonly described in terms of the significant wave height defined as the average peak-to-trough height of the 1/3 highest waves. The relation to average wave height is usually taken as:

$$h_{1/3} = 1.6 \times h_{av} \qquad (2)$$

From this relationship between average wind speed significant wave height is:

$$v_w = 7.18 \times h_{\frac{1}{3}}^{0.4} \qquad (3)$$

or inversely:

$$h_{1/3} = 0.00724 \times v_w^{0.25} \qquad (4)$$

The relationship between significant wave height and sea state assumed by Nathanson was given by rows 5 and 6 in his Figure 7.1 [1]. Although Sea State 0 is not explicitly labeled in this figure, Nathanson states in the notes accompanying the tabulated results (pg. 280) the he "arbitrarily" assumed that Sea State 0 corresponds to winds less than 4knots and significant wave heights less than 0.25ft. On this basis, the relationship between significant wave heights and sea state is as shown in Figure 1. The two black points with a significant wave-height of $h_{1/3} = 0.09m$ were obtained by extrapolating the values given in Figure 7.1 [1] towards lower values.

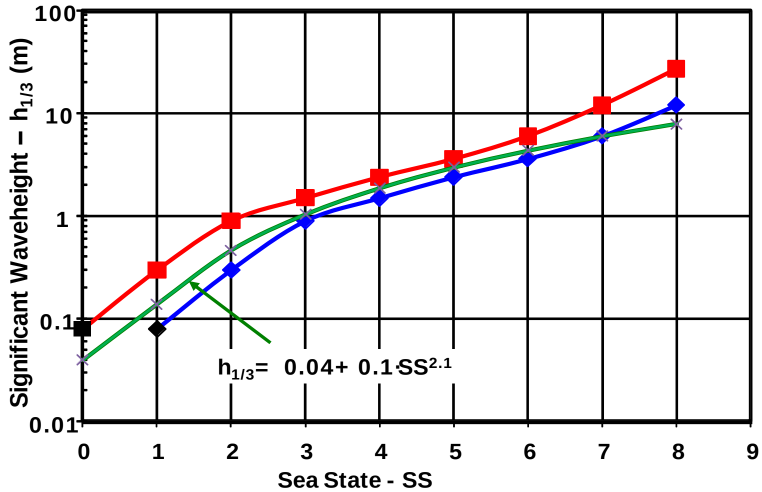

Figure 1: Relationship between Sea State and significant wave height as assumed by Nathanson [1]

The middle (green) curve is an empirical fit to the data using the equation:

$$h_{1/3} = 0.04 + 0.1 \times SS^{2.1} \qquad (5)$$

This equation will be used to compare the GIT empirical model with the Nathanson tables.

A slightly different relationship between sea state and significant wave height ($h_{1/3}$) is given in the World Meteorological Organization Universal Sea State code. A good fit to the WMO code is provided by the expression:

$$h_{1/3} = 1.6 \times h_{av} = 0.049 \times SS^{2.6} \qquad (6)$$

The most significant difference in the WMO code is that Sea State 0 corresponds to zero wave height (glassy surface) and thus would have no radar reflectivity at all. However, other authors agree with Nathanson's approach mapping Sea State 0 reflectivity to non-zero reflectivity, see for example the discussion by Briggs [5] and Barton [6].

For the comparison between the Nathanson tables and the GIT model, as presented below, the expressions given by equations (1) and (5) were assumed, since we believe that this is closest to Nathanson's assumptions when mapping the measured clutter reflectivity data to sea state.

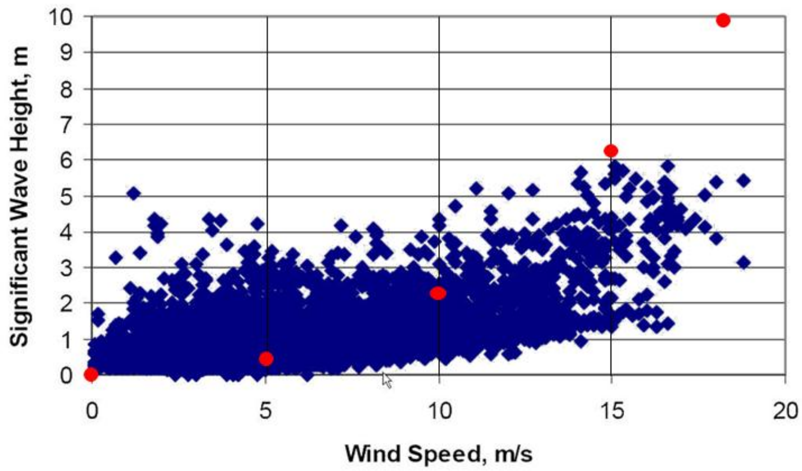

Figure 2: Waveheight vs. wind speed measured at Buoy 44013 off Hull, Massachusetts during 2006. Points shown in red were calculated from equation (4)

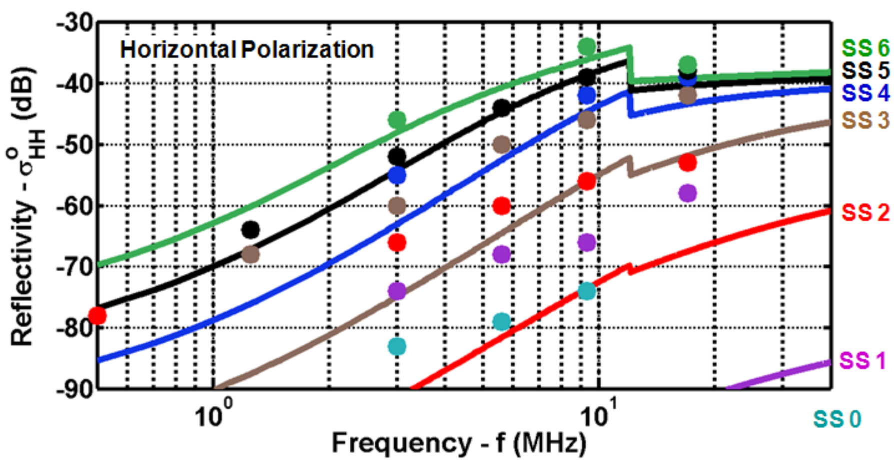

Figure 3: Grazing angle 0.3degrees. Points from Nathanson, curves GIT model. Horizontal Polarization

It should also be noted that a fully risen sea, as assumed in equation (4), rarely will be encountered in the real world. An example of data collected during the year 2006 at Buoy 44013 off the coast at Hull, Massachusetts, is shown in Figure 2 [7]. Each point shown was based on a 15-minute average value of both wind speed and wave-height. For comparison, the red circles were calculated using equation (4). This example illustrates the difficulty of relating wind speed and wave height based on experimental observations.

We also note that Nathanson states that the values in his tables are "averages of the decibel values of upwind, downwind and crosswind where available". Thus, when comparing the results with the GIT model, we shall use the crosswind case, where the wind is perpendicular to the radar look direction.

While describing the GIT model, Nathanson pointed out that is sometimes predicts a much lower reflectivity than the actual measured data. He speculated that unknown propagation conditions might account for some of these differences, but did not provide quantitative insight.

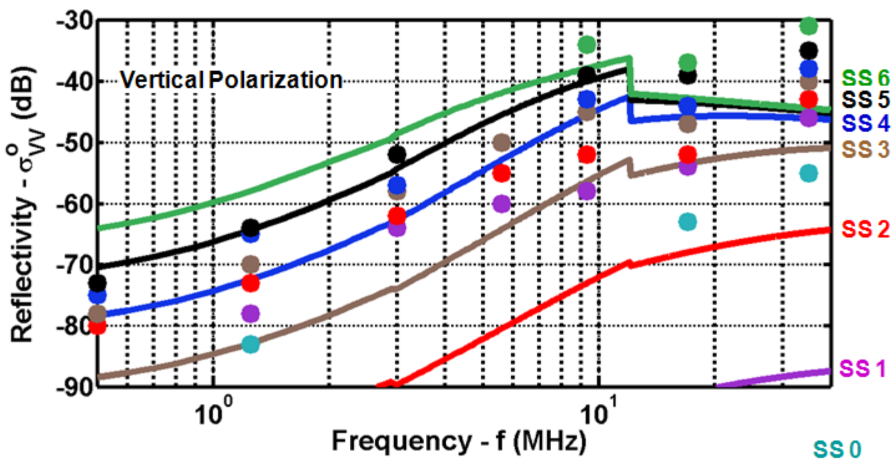

Figure 4: Grazing angle 0.3degrees. Points from Nathanson, curves from GIT model. Vertical Polarization

Using equations (1) through (5), we can now compare Nathanson's tables with the GIT empirical equations as shown in Figure 3 and Figure 4 for a grazing angle of 0.3degrees and horizontal and vertical polarization respectively[1]. While the agreement at Sea State 4 and above is reasonably good, large discrepancies are noted for Sea States 3 and below. For example, for Sea State 3, the difference across L- and X-band is around 10dB. For Sea State 2, the difference is even larger.

Note, that the curve calculated from the GIT model for Sea State 0 is below the scale used in these graphs. These large differences obviously give rise to the concern that if the GIT model is used in predictions of radar performance at moderate Sea States, much too optimistic results may be obtained. Also, note that the GIT model claims to be valid for grazing angles up to 15 degrees.

In the next section, a new empirical model is introduced, which provides much better agreement with the experimental data points in Nathanson. This model is an update of preliminary results presented at the 2009 IEEE Radar Conference in Pasadena, CA [8].

3 AN IMPROVED EMPIRICAL SEA CLUTTER MODEL

The point of departure for this paper is that any empirical model of sea clutter reflectivity, or any other sea clutter characteristic for that matter, must agree reasonably well with available experimental data. Consequently, a parameterized expression was sought, which could be used as a basis for such a new empirical sea clutter model using the Nathanson tables as the points of reference. Some authors have

[1] In the original paper by Horst et al, the discontinuity in the equations was located at 10GHz. It was later recommended to move the break point to 12GHz (as shown here) to have a single valid expression for all X-band frequencies.

raised doubts about the methodology used by Nathanson to obtain these tables. However, it must be noted that this data base evolved over a period of more than 20years and had inputs from many researchers in the field, so that until a new and better database becomes available, these results represent the de-facto standard for sea clutter reflectivity as a function of frequency, grazing angles, sea state, and polarization. Our proposed expression has the form:

$$\sigma_{H,V} = c_1 + c_2 \cdot \log_{10} \sin\alpha + \frac{(27.5+c_3\cdot\alpha)\cdot\log_{10} f}{(1+0.95\cdot\alpha)} + c_4 \cdot (1+SS)^{\frac{1}{2+0.085\cdot\alpha+0.033\cdot SS}} + c_5 \cdot \alpha^2 (dB) \qquad (7)$$

where α is the grazing angle in degrees, SS is the sea state, and f is the radar frequency in GHz. This empirical expression has five free parameters to match the equation to the experimental points for horizontal and vertical polarization respectively.

The five parameters, $c_1, c_2, \cdots c_5$, were adjusted to minimize the average absolute deviation between the empirical equation and the 267 data points for horizontal polarization and the 286 data points for the vertical polarization. Different sets of these five parameters are used for Horizontal and Vertical polarization. Appendix B provides the code to plot the figures (vs. frequency or grazing angles) shown in this report as well as to calculate the mean deviation from Nathanson data for both the NRL empirical model as well as the GIT model. The mean deviation numbers are provided in Table 2.

Table 1: Constants Used in Empirical Sea Clutter Model

	POLARIZATION	
CONSTANTS	HORIZONTAL	VERTICAL
c_1	-73.00	-50.79
c_2	20.78	25.93
c_3	7.351	0.7093
c_4	25.65	21.58
c_5	0.00540	0.00211

The results of the optimization of this set of five parameters using all data in Nathanson's tables for the grazing angles of 0.1, 0.3, 1.0, 3.0, 10.0, 30.0, and 60.0 degrees and all RF frequencies ranging from 0.5GHz to 35GHz is shown in Table 1. The optimization was carried out using the SOLVER function in Microsoft EXCEL.

In Figure 5 and Figure 6, the new NRL empirical equation and the data points from Nathanson are compared. In addition to the much better agreement with the measured data, no conditional expressions are used in the empirical equation. In Appendix A, a complete comparison of the NRL empirical model is shown for all of the grazing angles included in Nathanson's tables.

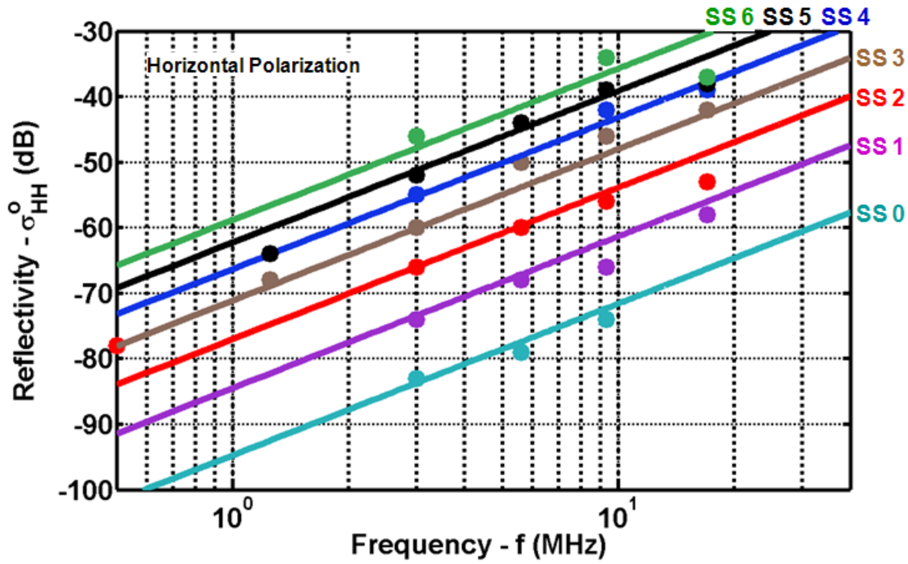

Figure 5: Grazing angle 0.3 degrees. Points from Nathanson, curves using NRL empirical model. Horizontal Polarization

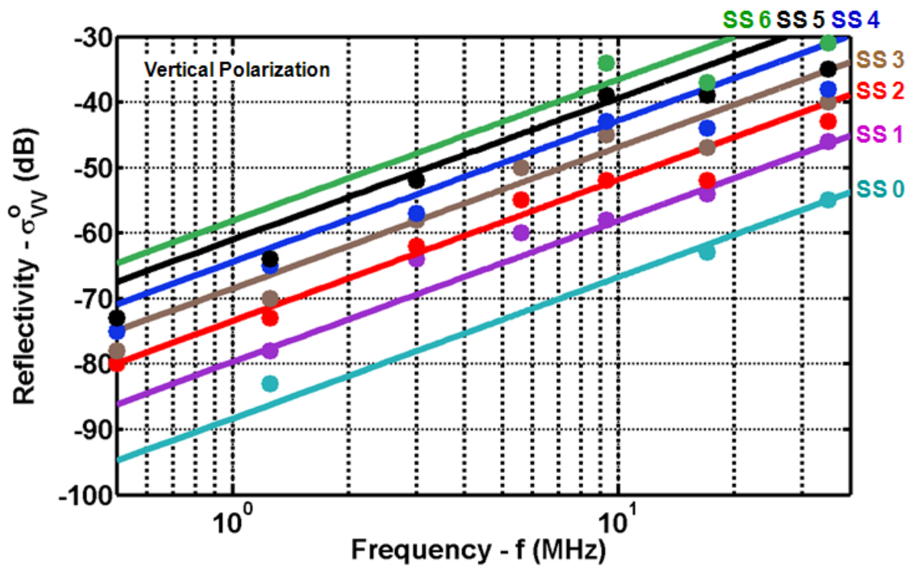

Figure 6: Grazing angle 0.3degrees. Points from Nathanson, curves using NRL empirical model. Vertical Polarization

For grazing angles up to 10degrees, the average absolute deviation of the NRL empirical equation for horizontal polarization is 2.3dB while for vertical polarization it is 2.0dB. In the following section, the average absolve deviation is computed for the GIT as well as several other empirical models proposed in the past.

4 OTHER EMPIRICAL SEA CLUTTER MODELS

A complete comparison of the GIT model with the Nathanson tables was used to obtain the average absolute deviation shown in the first row of Table 2. Several other empirical models have been proposed since the original GIT model was first published. One of these, the so-called Hybrid model [9], was introduced in an attempt to account for the effects of the evaporation duct. This model combines elements of the GIT model with new empirical equations. Again, the experimental data justifying this extension to the GIT model do not appear to be readily available. A comparison of this Hybrid model with the Nathanson tables shows some improvement over the GIT model for vertical polarization but the average deviation is still large (see Table 2). Note however, that the Hybrid mode is not valid for Sea State 0 so the much larger deviations for this case were excluded. Finally, a model developed by the Technology Service Corporation (TSC) was included in a commercial radar performance evaluation software package. This model has been summarized in a report published by the Australian Defense Science and Technology Organization (DSTO) [10][2]. The average absolute deviation between the TSC model and the Nathanson data is also included in Table 2 and provides only slight improvement. Again, Sea State 0 was excluded from this deviation computation. The final row shows the deviation for the proposed NRL model, which is noted to be well below any of the other models and which does include Sea State 0 case along with all other conditions.

Table 2: Average absolute deviation comparison to 10degrees grazing angle

MODEL	POLARIZATION	
	HORIZONTAL	VERTICAL
GIT Model[3]	13.4 dB	12.0 dB
HYBRID Model[3]	14.5 dB	8.7 dB
TSC Model[3]	7.9 dB	10.1 dB
NRL Model	2.3 dB	2.2 dB

Table 3: Average absolute deviation comparison to 60degrees grazing angle

MODEL	POLARIZATION	
	HORIZONTAL	VERTICAL
GIT Model[3]	12.8 dB	10.2 dB
HYBRID Model[3]	14.2 dB	9.3 dB
TSC Model[3]	8.6 dB	13.7 dB
NRL Model	2.6 dB	2.6 dB

[2] http://www.dsto.defence.gov.au/publications/2188/DSTO-TR-0679.pdf

[3] Sea State 0 excluded

In Table 3, a similar comparison is shown but for all grazing angles (from 0.1degrees to 60.0degrees) that are included in Nathanson's tables.

While none of the other models discussed claim validity up to 60degrees grazing angles, the deviation of the NRL model, which was optimized for this case, shows only a small increase in the average absolute deviation.

In addition to the above, several other workers have proposed sea clutter models for use in radar performance assessments. In the books by Briggs [5] a model is derived for S- and X-band in his Chapter 11, based on the Nathanson tables, using a methodology similar to that used in the present paper. His empirical equation uses only three (3) free parameters but these parameters must be chosen independently for each radar frequency. Briggs also puts forth arguments why significant sea clutter returns may be encountered at Sea States as low as 0 and 1.

In Barton's book [6], a much simplified model is described (Chapter 3) based on the so-called constant gamma model for surface reflectivity, where gamma is determined from an empirical equation, which is a function of sea state and radar wavelength. The sea reflectivity is then further adjusted by a two-way propagation factor based on wave height.

5 REFLECTIVITY AS A FUNCTION OF GRAZING ANGLE

Sea reflectivity is often graphed as a function of grazing angle with either sea state or radar frequency as a parameter. Such plots show the typical behavior of a rapid increate at very low grazing angles, followed by the plateau region at intermediate angles, and finally a rapid rise as grazing angles approach 90 degrees. To illustrate this behavior for the new proposed empirical mode, some of the results, already shown in Appendix A, are repeated here but as a function of grazing angle. While examining these results, it must be kept in mind that the empirical model is based on five tables at grazing angles at or below 10degrees and only two tables at 30 and 60 degrees. Hence, a better match is to be expected in the regimen of lower grazing angles.

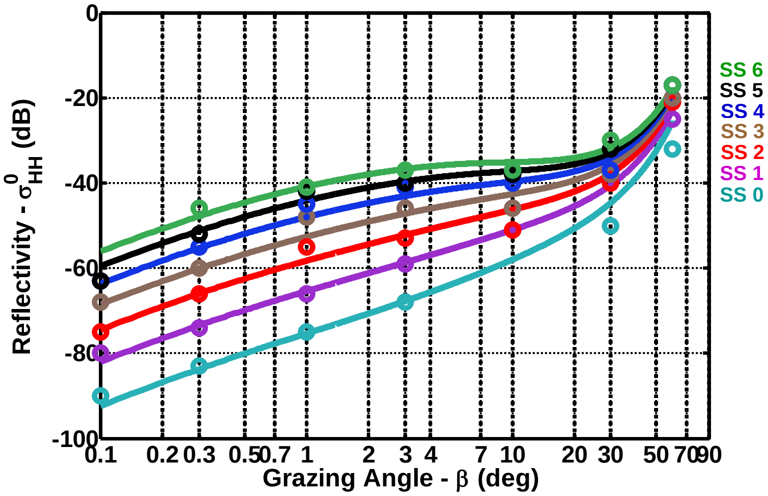

Figure 7: Frequency of 3GHz. Points from Nathanson, curves using NRL empirical model. Horizontal Polarization

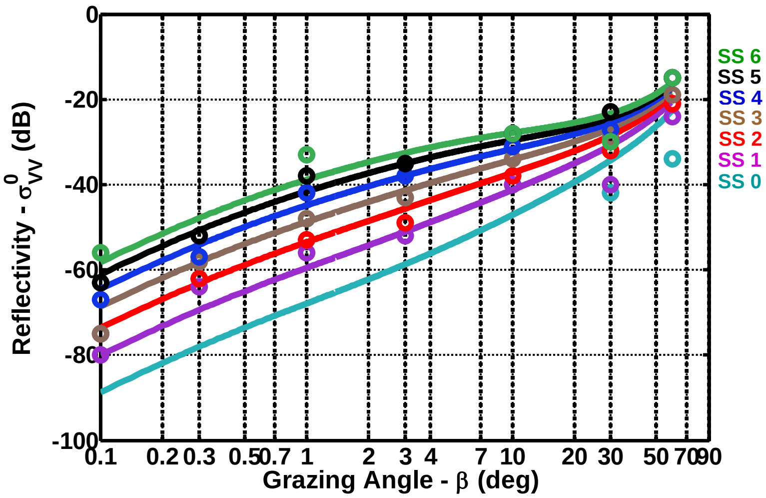

Figure 8: Frequency of 3GHz. Points from Nathanson, curves using NRL empirical model. Vertical Polarization

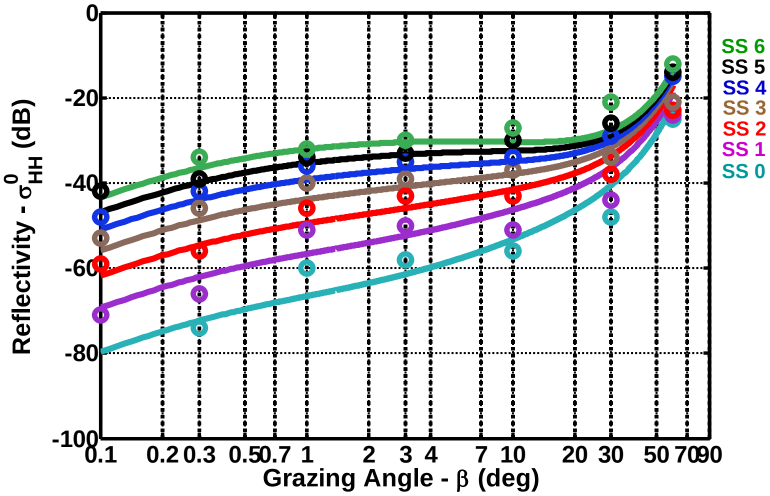

Figure 9: Frequency of 9.3GHz. Points from Nathanson, curves using NRL empirical model. Horizontal Polarization

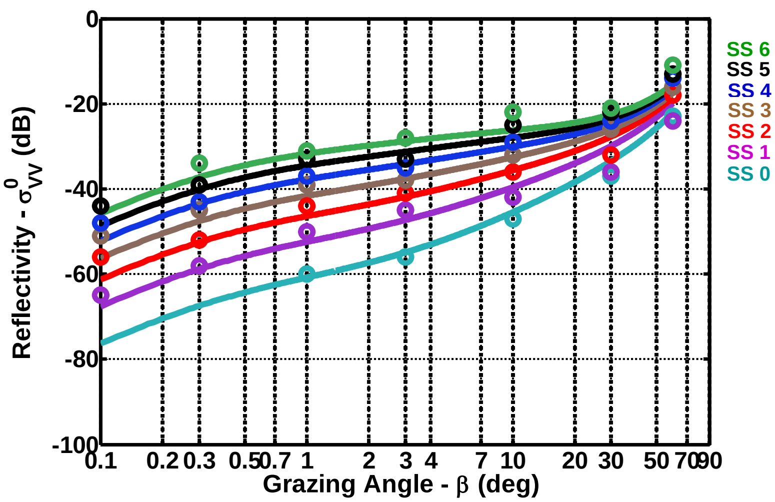

Figure 10: Frequency of 9.3GHz. Points from Nathanson, curves using NRL empirical model. Vertical Polarization

6 CONCLUSIONS

When compared with the Nathanson tables of measured sea clutter reflectivity [1], the GIT sea clutter model, proposed in 1978, predicts significantly lower values at sea states up to three. In this paper, a new empirical model has been proposed, which provides a much better match with the Nathanson data and is indexed to sea state as commonly assumed in radar performance assessments. This NRL empirical model matches the experimental results with an average absolute deviation of about 2.2 to 2.3dB for grazing angles from 0.1deg to 10deg (and 2.6dB for grazing angles up to 60degrees), and frequencies from 500 MHz to 35GHz. The deviation from the GIT model over this same range of grazing angles and frequencies is 10 to 13dB. The functional form for this empirical model is the same for horizontal and vertical polarization but uses different sets of the five free parameters. For completeness, appendix A compares the model to all data presented by Nathanson for grazing angles from 0.1 to 10degrees for both horizontal and vertical polarizations. Appendix B lists the MATLAB code that can be used to determine the sea clutter reflectivity given the sea state, radar frequency, grazing angles and polarization.

In developing this new empirical sea clutter model, no attempt was made to explicitly include any pre-assumed behavior as a function of either frequency or grazing angle. The only criterion used was to achieve a reasonable match with measured data, using a relatively simple mathematical expression with a modest number of free parameters. As shown, the empirical expression is fit to 267 data points for horizontal polarization and the 286 data points for vertical polarization using just two set of five parameters each. We believe that this new NRL model for the average reflectivity of sea clutter will be of value for radar performance analysis over a wide range of frequencies, grazing angles, and sea states. For high fidelity performance evaluation of target detection in a sea clutter environment, additional statistical descriptions of sea clutter temporal and spatial characteristics would have to be added. This area is still a subject of considerable research and discussion.

7 REFERENCES

[1] F. E. Nathanson, J. P. Reilly, and M. Cohen, *Radar Design Principles*, 2 ed. New York: McGraw-Hill, 1991.

[2] M. M. Horst, F.B.Dyer, and M.Tuley, "Radar sea clutter model," College Park, MD: 1978, pp. 6-10.

[3] K. D. Ward, R.J.A.Tough, and S.Watts, *Sea Clutter : Scattering, the K Distribution, and Radar Performance* Institution of Engineering and Technology, 2006.

[4] F. E. Nathanson, *Radar Design Principles*. New York: McGraw-Hill, 1969.

[5] J. N. Briggs, *Target Detection by Marine Radar*. London: The Institution of Electrical Engineers, 2004.

[6] D. K. Barton, *Radar System Analysis and Modeling*. Boston: Artech House, 2005.

[7] J. F. Manwell, J.F.MacGowan, C.N.Elkinton, and J.A.McLeod, "Status Report on the Hull Offshore Wind Project," American Wind Energy Association, Los Angeles, CA, 2007: 2007.

[8] V. Gregers-Hansen and R.Mital, "An empirical sea clutter model for low grazing angles," Pasadena, CA: 2009, pp. 1-5.

[9] J. P. Reilly and G.D.Dockery, "Influence of evaporation ducts on radar sea return," *IEE Proceedings Pt. F*, vol. 137, no. 2, pp. 80-88, Apr.1990.

[10] I. Antipov, "Simulation of Sea Clutter Returns," Defence Science and Technology Organization`,DSTO-TR-0679, June1999.

APPENDIX A -- COMPLETE DATA COMPARISON

This appendix includes a set of graphs, Figure 11 through Figure 16, comparing all of the experimental data tabulated by Nathanson [1], for grazing angles of 0.1 to 60degrees (except for 0.3degrees shown previously), with the NRL empirical reflectivity model described in this report. As pointed out previously, the average absolute deviation in dB, across all of these graphs, between the experimental data points and the corresponding value predicted by the NRL empirical equation, is 2.6dB. The frequency range covered by these results extends from 500 MHz to 35GHz. Figures 11 to 16 plot sea reflectivity vs. frequency while Figure 17 to 21 plot sea reflectivity vs. grazing angles for sea states 0 to 6.

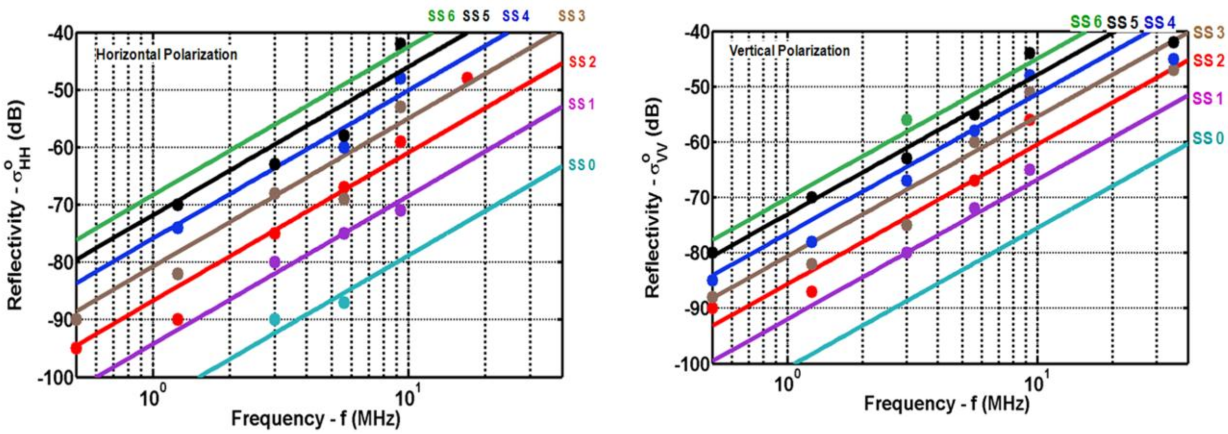

Figure 11 - Grazing angle 0.1degrees. Points from Nathanson, curves using NRL empirical model.

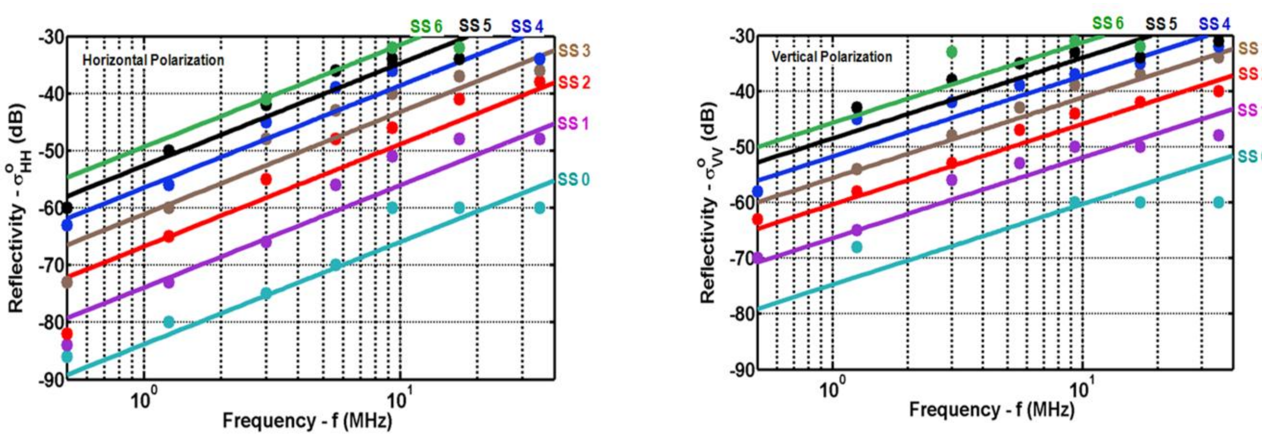

Figure 12 - Grazing angle 1.0degrees. Points from Nathanson, curves using NRL empirical model.

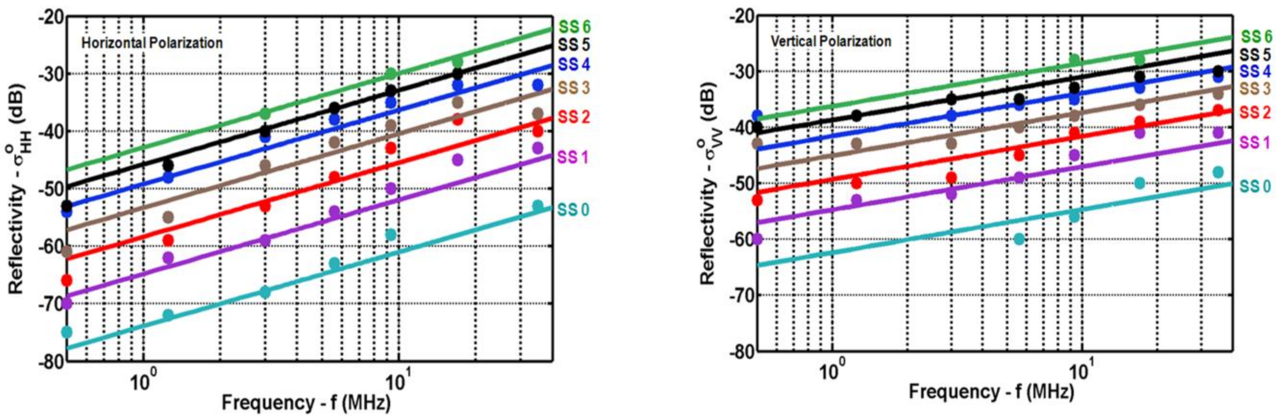

Figure 13 - Grazing angle 3degrees. Points from Nathanson, curves using NRL empirical model.

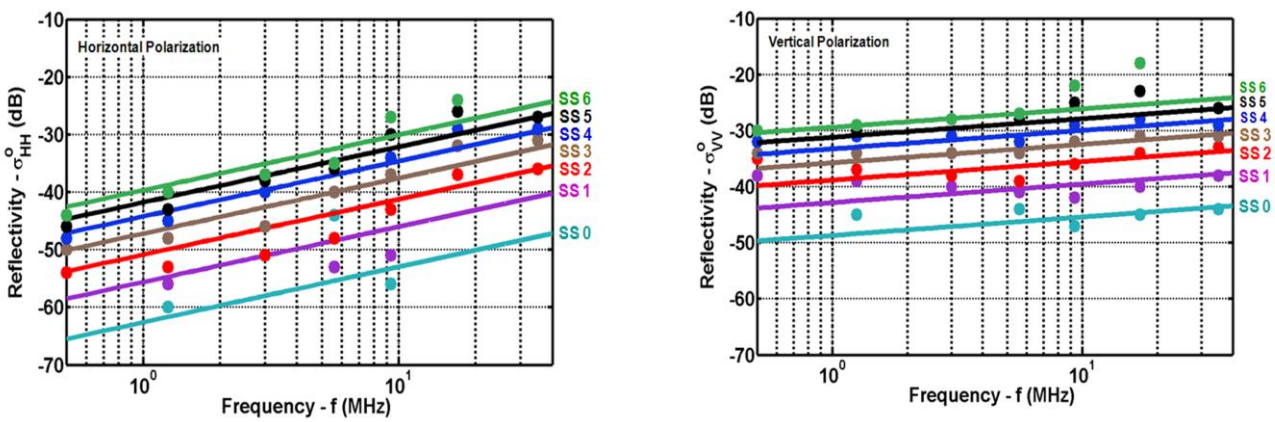

Figure 14 - Grazing angle 10degrees. Points from Nathanson, curves using NRL empirical model.

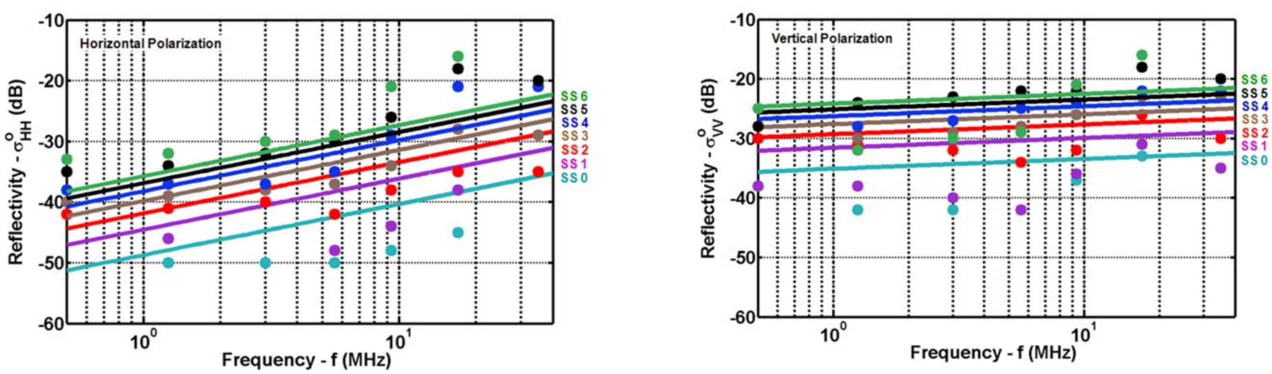

Figure 15 - Grazing angle 30degrees. Points from Nathanson, curves using NRL empirical model.

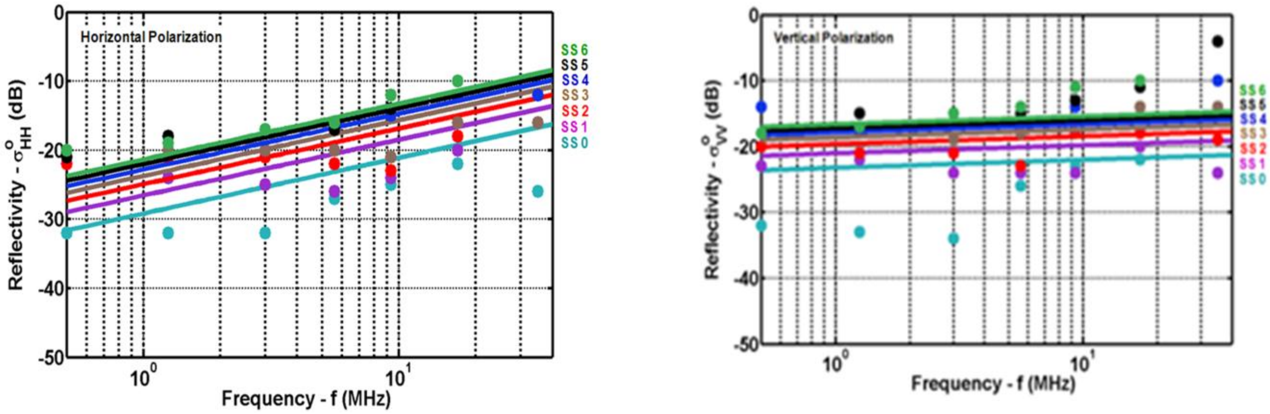

Figure 16 - Grazing angle 60degrees. Points from Nathanson, curves using NRL empirical model.

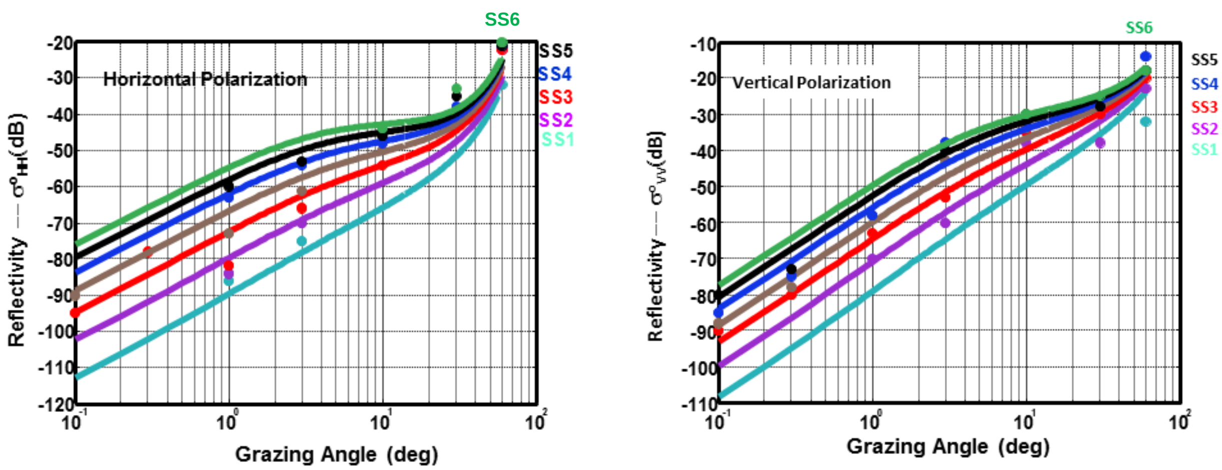

Figure 17 – Frequency 0.5GHz. Points from Nathanson, curves using NRL empirical model.

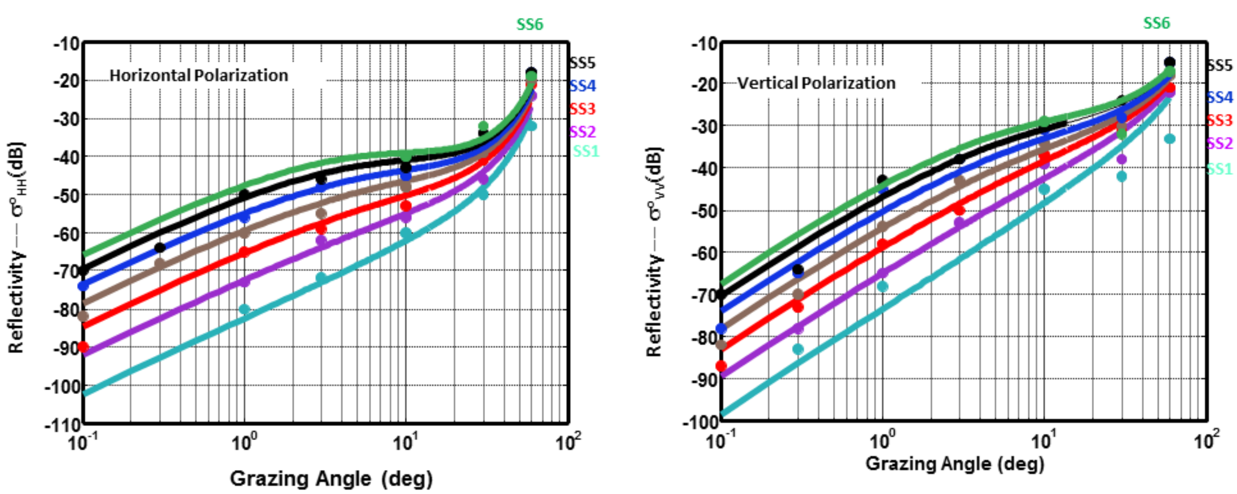

Figure 18—Frequency 1.25GHz. Points from Nathanson, curves using NRL empirical model

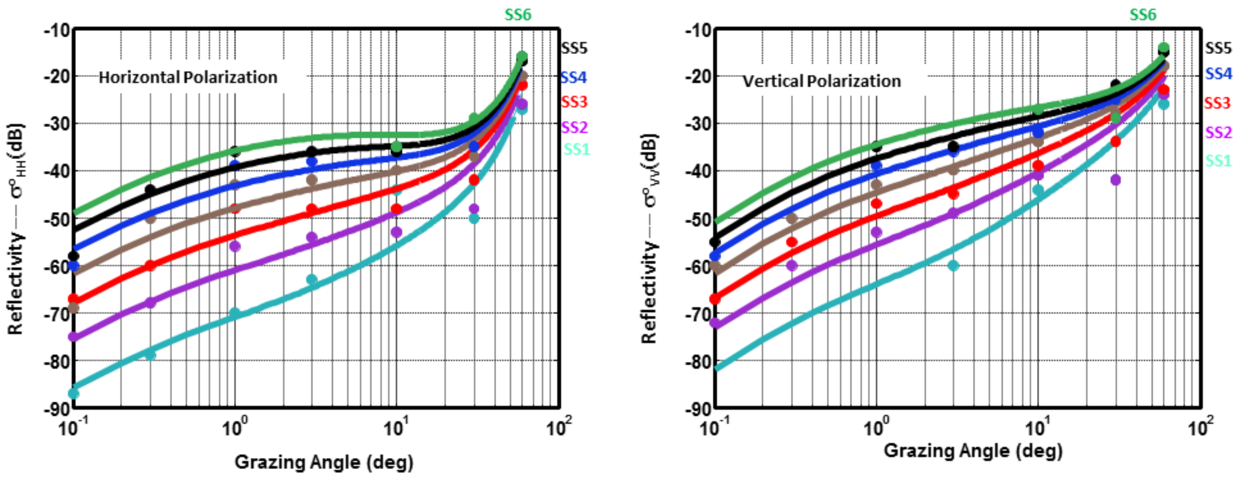

Figure 19—Frequency 5.6GHz. Points from Nathanson, curves using NRL empirical model

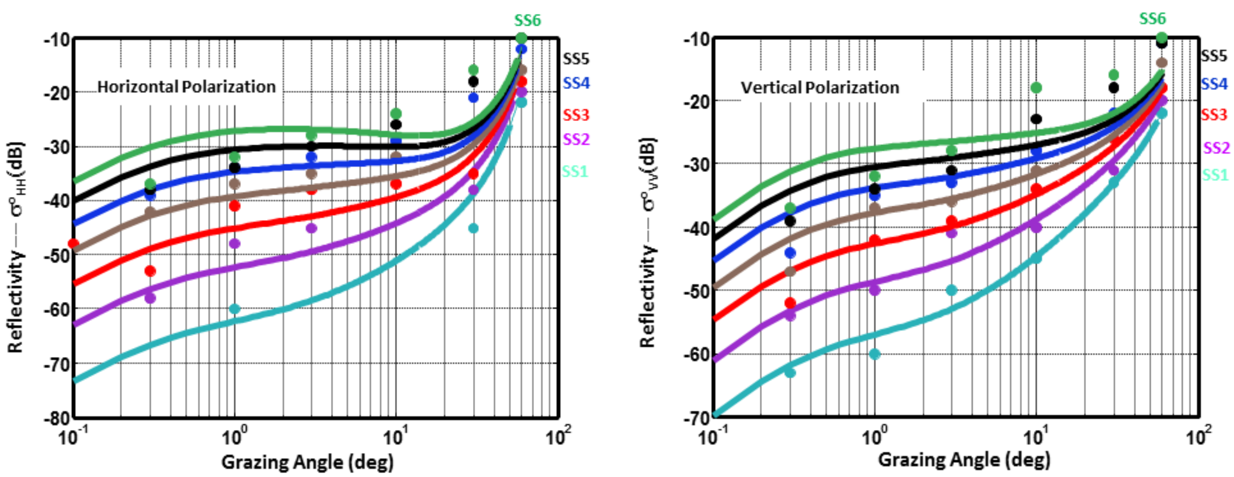

Figure 20—Frequency 17GHz. Points from Nathanson, curves using NRL empirical model

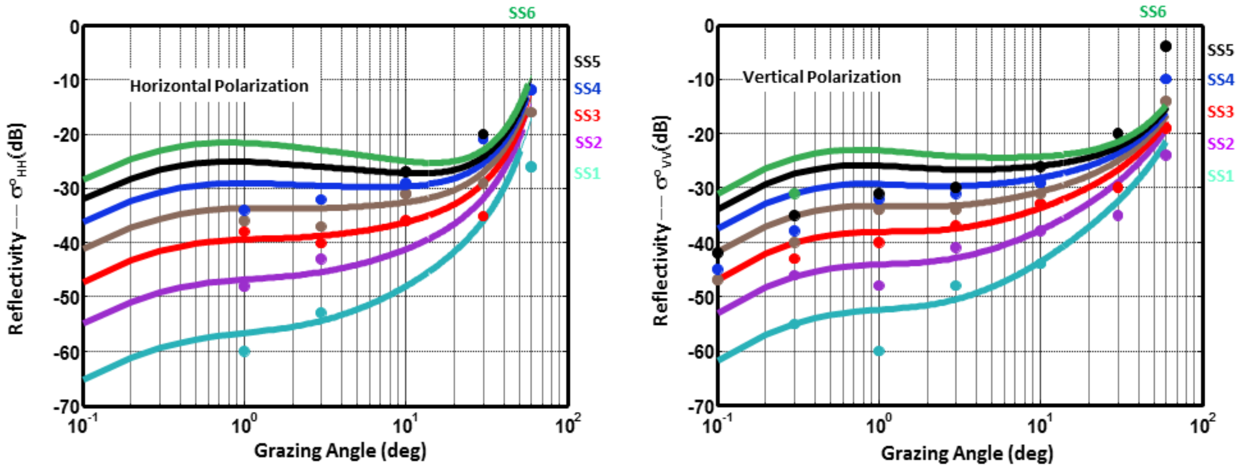

Figure 21—Frequency 35GHz. Points from Nathanson, curves using NRL empirical model

APPENDIX B -- MATLAB CODE

In this appendix, the MATLAB® listing of six programs is given. What follows is a short description of each of these programs.

- NRL_SigmaSea.m – This frunction calculates the reflectivity coefficient based on NRL empirical model (see Equation(7)) . The input to this code is the radar frequency, sea state, polarization and grazing angle while the output is reflectivity coefficient in dB

```
function SigZ = NRL_SigmaSea(fGHz,SS,Pol,Psi,ThWind)
% Vilhelm Gregers-Hansen, Naval Research Laboratory
% 5 May 2010
% SigmaSea computes reflectivity coefficient for sea clutter in dB
% fGHz is radar frequency in GHz
% SS is sea state (0-7)
% Pol is polarization - Pol=V or Pol=H
% Psi is grazing angle in deg
% ThWind is look direction relative to wind - NOT USED by VGHSigmaSea

% Convert grazing angle to radians
Psi_rad = Psi*pi/180;

if(Pol=='H')
   % These coefficients were optimized for 0 to 60 deg grazing angle

   CC1= -73.0;    CC2= 20.781; CC3= 7.351;    CC4=  25.65;  CC5=  0.0054;

elseif(Pol=='V')
   % These coefficients were optimized for 0 to 60 deg grazing angle

   CC1= -50.796; CC2=  25.93; CC3=  0.7093; CC4=  21.588; CC5= 0.00211;

end

   SigZ = CC1 + CC2*log10(sin(Psi_rad)) + (27.5+CC3*Psi)*log10(fGHz)./ ...
          (1.+0.95*Psi) + CC4*(SS+1).^(1.0 ./(2+0.085*Psi+0.033*SS))+ ...
          CC5*Psi.^2;
```

Figure 22 – MATLAB® listing of NRL_SigmaSea.m

- GTI_SigmaSea.m – This function implements the sea clutter reflectivity model described in Ref[2]. The inputs to this function include radar frequency, sea state, polarization, grazing angle and relative wind direction. This model is valid for sea states 1 through 6. If sea state 0 is chosen, then the function returns -200dB for the reflectivity coefficient.

```
function SigZ=GTI_SigmaSea(fGHz,SS,Pol,Psi,ThWind)

% This function implements the sea clutter reflectivity model
%    described in Horst, Dyer, and Tuley, Radar sea clutter model.

% Relative wind direction at 0 is worst case
%    At S-band average is 2 dB lower
%    At X-band average is 3.85 dB lower

% Model not valid for Sea State 0 - will return SigZ = -200 dB for this
%     case

% SigmaSea computes reflectivity coefficient for sea clutter in dB
% fGHz is radar frequency in GHz
% SS is sea state (0-7)
% Pol is polarization - Pol=V or Pol=H
% Psi is grazing angle in deg

% VALUES DEFINED FOR STAND-ALONE TEST OF FUNCTION
% fGHz = (0.5:0.1:40)
% SS = 1
% Pol = 'H'
% Psi = .1
% ThWind = 0

% Constants and conversions
c = 0.2997924562; % Speed of Light in Giga meters per second
DtoR = pi/180;
GrazAng = Psi*DtoR;
phi_rad = ThWind*DtoR;
Vw = 2.15*((SeaState+0)^1.04);
hav = 4.52*(10.0^(-3.0))*(Vw^(2.5))

lambda = c/fGHz;

if SeaState>0

   % Determine SigZ using GTI clutter model
   if  fGHz <= 12

      %         hav = 4.52*(10.0^(-3.0))*(Vw^(2.5));
      sigma_phi = (14.4 * lambda + 5.5).*GrazAng.*hav./(lambda+0.015);

      Ai = (sigma_phi.^4)./(1 + sigma_phi.^4);
      Au = exp((0.2.*abs(cos(phi_rad)).*(1 - 2.8.*(GrazAng)))./(lambda +
0.015).^0.4);

      qw = 1.1./(lambda + 0.015)^0.4;
      %         Vw = 8.67*hav^0.4;
```

```
      qw = 1.1./(lambda + 0.015)^0.4;
      %         Vw = 8.67*hav^0.4;

      Aw = ((1.94*Vw)./(1+Vw./15.4)).^qw;

      sigmaHH = 10.*log10(3.9e-6 .* lambda .* GrazAng.^0.4 .* Ai .* Au .*
Aw);
      if Pol == 'H'

         SigZ = sigmaHH;
      elseif Pol == 'V'
         if fGHz >= 3
            sigmaVV = sigmaHH - 1.05.*log(hav + 0.015) + 1.09.*log(lambda) +
1.27.*log((GrazAng) + 0.0001) + 9.70;
         elseif fGHz < 3
            sigmaVV = sigmaHH - 1.73.*log(hav + 0.015) + 3.76.*log(lambda) +
2.46.*log((GrazAng) + 0.0001) + 22.2;
         end;
         SigZ = sigmaVV;
      else
         disp('No valid polarization in SigmaSeaGTI')
      end

   elseif fGHz > 12

      %         hav = 4.52*(10.0^(-3.0))*(Vw^(2.5));
      sigma_phi = (14.4*lambda + 5.5).*GrazAng.*hav./(lambda+0.015);

      Ai = (sigma_phi.^4)./(1 + sigma_phi.^4);
      Au = exp(0.25.*abs(cos(phi_rad)).*(1-2.8.*sin(GrazAng).*lambda.^(-
0.33)));

      qw = 1.93*lambda^(-0.04);
      %         Vw = 8.67.*hav^(0.4);

      Aw = ((1.94*Vw)./(1+Vw./15.4)).^qw;
      sigmaHH = 10.*log10(5.78e-6 .* GrazAng.^0.547 .* Ai .* Au .* Aw);
      sigmaVV = sigmaHH - 1.38.*log(hav) + 3.43.*log(lambda) +
1.31.*log(GrazAng+eps) + 18.55;

      if Pol == 'H'
         SigZ = sigmaHH;
      elseif Pol == 'V'
         SigZ = sigmaVV;
      else
         disp('No valid polarization in SigmaSeaGTI')
      end

   end;
else
   SigZ = -200*ones(size(fGHz));
end
```

Figure 23: MATLAB® listing of GTI_SigmaSea.m

- HYB_SigmaSea.m -- This function implements the sea clutter reflectivity model described in Ref[9]. The inputs to this function include radar frequency, sea state, polarization, grazing angle and wind direction. This model is valid for sea states 1 through 6. If sea state 0 is chosen, then the function returns -200dB for the reflectivity coefficient.

```
function SigZ = HYB_SigmaSea(fGHz,SS,Pol,Psi,ThWind)

% This function implements the Hybrid sea clutter reflectivity model
%    described in Reilly and Dockery : Influence of evaporation duct
%    on radar sea return, IEE Proc. Pt. F, April 1990.

% Wind direction is relative to radar look direction - upwind is 0 deg
%    ThWind=0 deg is worst case (highest reflectivity)
%    At S-band average is 2 dB lower
%    At X-band average is 3.85 dB lower

% Model is not valid for Sea State 0
if SS<1
   SigZ = -200*ones(size(fGHz));
   return
end


% SigmaSea computes reflectivity coefficient for sea clutter in dB
% fGHz is radar frequency in GHz
% SS is sea state (0-7)
% Pol is polarization - Pol=V or Pol=H
% Psi is grazing angle in deg

% fGHz = 3
% SS = 3
% Pol = 'H'
% Psi = 1

% Conversions

DtoR = pi()/180;
GrazAng = Psi*DtoR;

SeaState = SS;

freq = fGHz;

Lambda = 0.3/freq;

if freq > 12.5
   SigZRef = 3.25*log10(freq) - 42.0;
else
   SigZRef = 24.4*log10(freq)-65.2;
end

SigHeight = 0.031*SeaState^2;
hWave = 0.08*SeaState^2;
```

```
% Grazing Angle Adjustment
% ========================
GrazAngRef = 0.1*DtoR;
GrazAngTrans = asin(0.66*Lambda/SigHeight);
%GrazAngTrans = asin(0.0632*Lambda/SigHeight);    % DSTO Report

if GrazAngTrans<GrazAngRef
   if GrazAng > GrazAngRef
      KGraz = 10*log10(GrazAng/GrazAngRef);
   else
      KGraz=0;
   end
else       % Transitional >= Reference
   if GrazAng<GrazAngRef
      KGraz=0;
   elseif (GrazAng >= GrazAngRef) && (GrazAng<=GrazAngTrans)
      KGraz = 20*log10(GrazAng/GrazAngRef);
   else
      KGraz =
10*log10(GrazAng/GrazAngTrans)+20*log10(GrazAngTrans/GrazAngRef);
   end
end

% Sea State Adjustment
% ====================
KSea = 5*(SeaState - 5);

% Polarization Adjustment
% =======================
if Pol=='H'
   if freq<3
      KPol = 1.7*log(hWave+0.015) - 3.8*log(Lambda) - 2.5*log(GrazAng/57.3 +
0.0001)-22.2;
   elseif (freq>=3) && (freq<=10)
      KPol = 1.1*log(hWave+0.015) - 1.1*log(Lambda) - 1.3*log(GrazAng/57.3 +
0.0001)-9.7;
   else
      KPol = 1.4*log(hWave) - 3.4*log(Lambda) - 1.3*log(GrazAng/57.3)-18.6;
   end
elseif Pol == 'V'
   KPol = 0;
else
   disp('ERROR - Incorrect Polarization Specified')
   exit
end

% Wind Direction Adjustment
% =========================
KDir = (2+1.7*log(0.1/Lambda))*(cos(ThWind)-1);

% Resultant Sigma Zero
% ====================
SigZ = SigZRef + KGraz +KSea + KPol + KDir;
```

Figure 24: MATLAB® listing of HYB_SigmaSea.m

- TSC_SigmaSea.m -- This function implements the sea clutter reflectivity model described in Ref[10]. The inputs to this function include radar frequency, sea state, polarization, grazing angle and wind direction. This model is valid for sea states 1 through 6. If sea state 0 is chosen, then the function returns -200dB for the reflectivity coefficient.

```
function SigZ = TSC_SigmaSea(fGHz,SS,Pol,Psi,ThWind)

% This function implements the TSC model implemented in Antipov,
% "Simulation of Sea Clutter Returns", DSTO, 1999

% Wind direction is relative to radar look direction - upwind is 0 deg
%    ThWind=0 deg is worst case (highest reflectivity)
%    At S-band average is 2 dB lower
%    At X-band average is 3.85 dB lower

% Vilhelm Gregers-Hansen 08 FEB 2009

% Model is not valid for Sea State 0 - set result to -200 dB
if SS<1
    SigZ = -200*ones(size(fGHz));
    return
end

 % SigmaSea computes reflectivity coefficient for sea clutter in dB
% fGHz is radar frequency in GHz
% SS is sea state (0-7)
% Pol is polarization - Pol=V or Pol=H
% Psi is grazing angle in deg

% fGHz = 3
% SS = 3
% Pol = 'H'
% Psi = 1

% Conversions

DtoR = pi()/180;
GrazAng = Psi*DtoR;

SeaState = SS;

freq = fGHz;

Lambda = 0.3/freq;

ThWindRad = ThWind*DtoR;

SigHeight = 0.115*SeaState^1.95;
SigAlpha = 14.9*GrazAng*(SigHeight+0.25)/Lambda;
GA = SigAlpha^1.5/(1.0+SigAlpha^1.5);

% Wind Velocity
Vw = 6.2*SS^0.8;

 % Wind Speed Factor Gw
% ========================
```

```
Q = GrazAng^0.6;

A1 = (1+(Lambda/0.03)^3)^0.1;
A2 = (1+(Lambda/0.1)^3)^0.1;
A3 = (1+(Lambda/0.3)^3)^(Q/3);
A4 = 1+0.35*Q;
A = 2.63*A1/(A2*A3*A4);
Gw = ((Vw+4.0)/15)^A;

GrazAngRef = 0.1*DtoR;
GrazAngTrans = asin(0.66*Lambda/SigHeight);
%GrazAngTrans = asin(0.0632*Lambda/SigHeight);    % DSTO Report

% Aspect Factor Gu - Special case of 45 degrees ignored
% ====================
Gu = exp(0.3*cos(ThWindRad)*(exp(-GrazAng/0.17)/(Lambda^2+0.005)^0.2));

% Compute Reflectivity coefficient
% =========================

SigZH = 10*log10(1.7E-5*GrazAng^0.5*Gu*Gw*GA/(Lambda+0.05^1.8));
if Pol=='H'
    SigZ = SigZH;
else
    if fGHz<2
        SigZ = SigZH-10*log10(1.73*log(2.507*SigHeight+0.05)+3.76*log(Lambda)
...
            +2.46*log(sin(GrazAng+0.0001))+19.8);
    else
        SigZ = SigZH-10*log10(1.05*log(2.507*SigHeight+0.05)+1.09*log(Lambda)
...
            +1.27*log(sin(GrazAng+0.0001))+9.65);
    end
end
```

Figure 25: MATLAB® listing of TSC_SigmaSea.m

- SigmaSea_vs_Freq.m – This code is used to generate Figures 11 through 16 where sea reflectivity (Horizontal and Vertical pol) is plotted vs. frequency. Depending on the model that is used for comparison (choose between NRL, GTI, HYB or TSC), the function also calculates the mean deviation between the model and the tabulated data in Nathanson. These results were used to generate Tables 2 and 3.

```
% Program plots the NEW 5-PARAMETER VGH empirical sea clutter model
% Also plots points from Nathanson 2nd ed.
% Now optimized to 60 deg grazing angle
% Average absolute error to 60 deg:   2.64 Hor, 2.54 V

%Author: Rashmi Mital
%Date: 11/30/2007
clc;
clear all;
close all;

%DEFINE CONSTANTS
c = 2.997924562e8; %Speed of Light
dtor = pi/180;

% Define INPUTS
%Grazing angle (data available for 0.1 0.3 1.0 3.0 10.0 30.0 60.0)
%GrAng =[0.1 0.3 1.0 3.0 10.0 30.0 60.0];
GrAng =[0.1 0.3 1.0 3.0 10.0];
%GrAng =[0.3];
NGrAng = length(GrAng);

phi = 0; %Angle betwen boresight and upwind (deg), only affects GTRI model

% Just a dummy in this program
ThWind = 0;

%Choose Model to Compare with Nathanson Data
Model = 'NRL'; %Option: 'NRL', 'GTI','HYB','TSC'

% Frequency axis for plotting empirical curves
freq = (0.5:0.1:40); %Frequency in GHz

fMin=freq(1);
fMax=max(freq);
fMax=freq(end);

% Variables for computing average abs deviation
SumDevH = 0.0;
NValH = 0;
SumDevV = 0.0;
NValV = 0;

phi_rad = phi .* dtor;

% GRAPHICS DEFINITIONS
ha=[];hl=[];hp=[];ht=[];htx=[];hty=[];
set(0,'Units','pixels')
```

```
% Set SCreen Size
scnsize=get(0,'ScreenSize')
pos1  = [5+scnsize(1),0.02*scnsize(4),.99*scnsize(3),.75*scnsize(4)];

FigFont = 28;
CircSize = 14;
LineSize = 7;
AxWidth = 4;

% Use these for saved jpg figures
% FigFont = 21;
% CircSize = 7;
% LineSize = 4;
% AxWidth = 3;

% Initialize figure number
nFig = 0;
% Defines units for values returned by ScreenSize

% Define colors for curves and points
Cols = 'bgrcmyk'

ColsMat = [ 0.161 0.698 0.725;    % magenta
            0.608 0.18  0.8;      % violet
            1.0   0.0   0.0;      % red
            0.541 0.416 0.361;    % brown
            0.067 0.204 0.898;    % dark blue
            0.0   0.0   0.0;      % black
            0.224 0.678 0.333];   % dark green

%########################################################################
% ALL GRAZING ANGLES
for iGrAng = 1:NGrAng
   alpha = GrAng(iGrAng);
   alpha_rad = alpha .* dtor;
   nFig = nFig + 2;

   % Default plot limits
   SigmaMin=-100;
   SigmaMax=0;

   % Read Nathan Table for current grazing angle and both polarizations
   Freq_Nath = [0.5 1.25 3 5.6 9.3 17 35];

   % Initialize Tables
   Measured_SigmaHH=zeros(7,7);
   Measured_SigmaVV=zeros(7,7);

   if abs(alpha-0.1)< 0.01
      SigmaMin=-100;
      SigmaMax=-40;
      Measured_SigmaHH = ...
         [ 0    0 -90 -87   0   0    0;
         0    0 -80 -75 -71   0    0;
         -95  -90 -75 -67 -59 -48    0;
         -90  -82 -68 -69 -53   0    0;
         0  -74 -63 -60 -48   0    0;
```

```
      0  -70 -63 -58 -42   0    0;
      0    0   0   0   0   0    0];

   Measured_SigmaVV = ...
      [ 0    0   0   0   0   0    0;
      0    0 -80 -72 -65   0    0;
      -90  -87 -75 -67 -56   0    0;
      -88  -82 -75 -60 -51   0  -47;
      -85  -78 -67 -58 -48   0  -45;
      -80  -70 -63 -55 -44   0  -42;
      0    0 -56   0   0   0    0];

elseif abs(alpha-0.3)< 0.01
   SigmaMin=-90;
   SigmaMax=-30;
   Measured_SigmaHH = ...
      [  0    0 -83 -79 -74   0   0;
      0    0 -74 -68 -66 -58   0;
      -78    0 -66 -60 -56 -53   0;
      0  -68 -60 -50 -46 -42   0;
      0    0 -55   0 -42 -39   0;
      0  -64 -52 -44 -39 -38   0;
      0    0 -46   0 -34 -37   0];

   Measured_SigmaVV = ...
      [  0  -83   0   0   0 -63 -55;
      0  -78 -64 -60 -58 -54 -46;
      -80  -73 -62 -55 -52 -52 -43;
      -78  -70 -58 -50 -45 -47 -40;
      -75  -65 -57   0 -43 -44 -38;
      -73  -64 -52   0 -39 -39 -35;
      0    0   0   0 -34 -37 -31];

elseif abs(alpha-1.0)< 0.01
   SigmaMin=-90;
   SigmaMax=-30;
   Measured_SigmaHH = ...
      [-86 -80 -75 -70 -60 -60 -60;
      -84 -73 -66 -56 -51 -48 -48;
      -82 -65 -55 -48 -46 -41 -38;
      -73 -60 -48 -43 -40 -37 -36;
      -63 -56 -45 -39 -36 -34 -34;
      -60 -50 -42 -36 -34 -34   0;
      0   0 -41   0 -32 -32   0];

   Measured_SigmaVV = ...
      [     0 -68   0   0 -60 -60 -60;
      -70 -65 -56 -53 -50 -50 -48;
      -63 -58 -53 -47 -44 -42 -40;
      -58 -54 -48 -43 -39 -37 -34;
      -58 -45 -42 -39 -37 -35 -32;
      0 -43 -38 -35 -33 -34 -31;
      0   0 -33   0 -31 -32   0];

elseif abs(alpha-3.0)< 0.01
   SigmaMin=-80;
   SigmaMax=-20;
```

```
    Measured_SigmaHH = ...
       [-75 -72 -68 -63 -58   0 -53;
       -70 -62 -59 -54 -50 -45 -43;
       -66 -59 -53 -48 -43 -38 -40;
       -61 -55 -46 -42 -39 -35 -37;
       -54 -48 -41 -38 -35 -32 -32;
       -53 -46 -40 -36 -33 -30   0;
       0   0 -37   0 -30 -28   0];

    Measured_SigmaVV = ...
       [  0   0   0 -60 -56 -50 -48;
       -60 -53 -52 -49 -45 -41 -41;
       -53 -50 -49 -45 -41 -39 -37;
       -43 -43 -43 -40 -38 -36 -34;
       -38 -38 -38 -36 -35 -33 -31;
       -40 -38 -35 -35 -33 -31 -30;
       0   0   0   0 -28 -28   0];

 elseif abs(alpha-10.0)< 0.01
    SigmaMin=-70;
    SigmaMax=-10;
    Measured_SigmaHH = ...
       [   0 -60   0 -44 -56   0   0;
       0 -56   0 -53 -51   0   0;
       -54 -53 -51 -48 -43 -37 -36;
       -50 -48 -46 -40 -37 -32 -31;
       -48 -45 -40 -36 -34 -29 -29;
       -46 -43 -38 -36 -30 -26 -27;
       -44 -40 -37 -35 -27 -24   0];

    Measured_SigmaVV = ...
       [   0 -45   0 -44 -47 -45 -44;
       -38 -39 -40 -41 -42 -40 -38;
       -35 -37 -38 -39 -36 -34 -33;
       -34 -34 -34 -34 -32 -31 -31;
       -32 -31 -31 -32 -29 -28 -29;
       -30 -30 -28 -28 -25 -23 -26;
       -30 -29 -28 -27 -22 -18   0];

 elseif abs(alpha-30)< 0.01
    SigmaMin=-70;
    SigmaMax=-10;
    Measured_SigmaHH = ...
       [  0 -50 -50 -50 -48 -45   0;
       0 -46   0 -48 -44 -38   0;
       -42 -41 -40 -42 -38 -35 -35;
       -40 -39 -38 -37 -34 -28 -29;
       -38 -37 -37 -35 -29 -21 -21;
       -35 -34 -32 -30 -26 -18 -20;
       -33 -32 -30 -29 -21 -16   0];

    Measured_SigmaVV = ...
       [   0 -42 -42 -42 -37 -33   0;
       -38 -38 -40 -42 -36 -31 -35;
       -30 -31 -32 -34 -32 -26 -30;
       -28 -30 -29 -28 -26 -23 -23;
       -28 -28 -27 -25 -24 -22 -22;
```

```
         -28 -24 -23 -22 -22 -18 -20;
         -25 -32 -30 -29 -21 -16   0];

   elseif abs(alpha-60.0)< 0.01
      SigmaMin=-60;
      SigmaMax=0;
      Measured_SigmaHH = ...
         [  -32 -32 -32 -27 -25 -22 -26;
         -22 -24 -25 -26 -24 -20   0;
         -22 -21 -21 -22 -23 -18   0;
         -21 -20 -20 -20 -21 -16 -16;
         -21 -18 -17 -16 -15 -12 -12;
         -21 -18 -17 -17 -14 -10   0;
         -20 -19 -17 -16 -12 -10   0];

      Measured_SigmaVV = ...
         [-32 -33 -34 -26 -23 -22   0;
         -23 -22 -24 -24 -24 -20 -24;
         -20 -21 -21 -23 -18 -18 -19;
         -18 -18 -19 -18 -16 -14 -14;
         -14 -15 -15 -15 -14 -11 -10;
         -18 -15 -15 -15 -13 -11  -4;
         -18 -17 -15 -14 -11 -10   0];

   end;

   % Define Plot Axes
   minyaxis=SigmaMin;
   maxyaxis=SigmaMax;
   minxaxis=fMin;
   maxxaxis=fMax;


   hFig=figure(nFig-1);   % Plot Hor Pol

   set(hFig,'Position',pos1);

   ht(end+1)=gca;
   ha(end+1)=gca;
   set (ha(end),'PlotBoxAspectRatio',[1 .2 .5])

   % Do all sea states for HH
   for SeaSt=0:6
      SeaState=SeaSt;
      SS_Plus1 = SeaState + 1;

      Pol = 'H';

      switch upper(Model)
          case 'NRL'
              SigZHor = NRL_SigmaSea(freq,SeaState,Pol,alpha,ThWind);
          case 'GTI'
              for ifreq = 1 : length(freq)
                  SigZHor(ifreq) =
GTI_SigmaSea(freq(ifreq),SeaState,Pol,alpha,ThWind);
              end;
          case 'HYB'
```

```
            for ifreq = 1 : length(freq)
                SigZHor(ifreq) =
HYB_SigmaSea(freq(ifreq),SeaState,Pol,alpha,ThWind);
            end;
        case 'TSC'
            for ifreq = 1 : length(freq)
                SigZHor(ifreq) =
TSC_SigmaSea(freq(ifreq),SeaState,Pol,alpha,ThWind);
            end;
    end;


    SigmaHH = SigZHor;

    % Plot empirical curve first
    NOWColorH=ColsMat(SeaSt+1,:)

    %figure(nFig-1)
    %       hl(end+1) = semilogx(freq,SigmaHH,'LineWidth',LineSize);

    hl(end+1) = semilogx(freq,SigmaHH);
    hold on
    set(hl(end),'Color',ColsMat(SeaSt+1,:));
 end

 for SeaSt=0:6
    SeaState=SeaSt;
    SS_Plus1 = SeaState + 1;
    % Plot measured values from Nathanson
    for ii=1:length(Freq_Nath)
       if Measured_SigmaHH(SS_Plus1,ii) ~= 0
          hp(end+1) =
semilogx(Freq_Nath(ii),Measured_SigmaHH(SS_Plus1,ii),'o');
          set(hp(end),'Color',ColsMat(SeaSt+1,:)); % Use as example for
custom colors
          hold on
          fGHz = Freq_Nath(ii);
          Pol = 'H';

           switch upper(Model)
               case 'NRL'
                   SigZErr = abs(Measured_SigmaHH(SS_Plus1,ii)-
NRL_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
               case 'GTI'
                   SigZErr = abs(Measured_SigmaHH(SS_Plus1,ii) -
GTI_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
               case 'HYB'
                   SigZErr = abs(Measured_SigmaHH(SS_Plus1,ii) -
HYB_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
               case 'TSC'
                   SigZErr = abs(Measured_SigmaHH(SS_Plus1,ii) -
TSC_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
           end;

%             SigZErr = abs(Measured_SigmaHH(SS_Plus1,ii)-
VGHSigmaSeaNew(fGHz,SeaState,Pol,alpha,ThWind));
          SumDevH = SumDevH + SigZErr;
```

```
            NValH = NValH + 1;
         end;
      end
   end     % END Sea State Loop

   grid on;



   set(gca,'YLim',[minyaxis maxyaxis],'XLim',[minxaxis maxxaxis]);
   htx(end+1) = xlabel('Frequency - f (MHz) ');
   hty(end+1) = ylabel('Reflectivity - \sigma_{HH}^o (dB)');

   %set(ha,'YLim',[minyaxis maxyaxis],'XLim',[minxaxis maxxaxis]);
   set(ha,'FontWeight','Bold');
   set(ha,'FontSize',FigFont-1);
   set(htx,'FontSize',FigFont);
   set(htx,'FontWeight','Bold');
   set(hty,'FontSize',FigFont);
   set(hty,'FontWeight','Bold');
   set(hl,'LineWidth',LineSize);
   set(hp,'LineWidth',CircSize);
   set(ha,'LineWidth',AxWidth);
   set(ha,'xtick',[0.5,.7,1.0,2.0,3.0,5.0,7.0,10.0,20.0,...
          30.0,40.0]);
   set(ha,'XLim',[0.5 40])

   whitebg(gcf,[1 1 1])

   set (gca,'PlotBoxAspectRatio',[1 .5 1])
   FilFig = ['HH_VGHNew' num2str(round(10*alpha))];
   saveas(gcf, [FilFig '.jpg'])
   %pause

   hFig=figure(nFig);   % Plot Vert Pol First

   ht(end+1)=gca;
   ha(end+1)=gca;
   set(gcf,'Position',pos1);
   set (ha(end),'PlotBoxAspectRatio',[1 .5 1])

   % Do all sea states for VV
   for SeaSt=0:6
      SeaState=SeaSt;
      SS_Plus1 = SeaState + 1;
      Pol = 'V';

       switch upper(Model)
           case 'NRL'
               SigZVer = NRL_SigmaSea(freq,SeaState,Pol,alpha,ThWind);
           case 'GTI'
               for ifreq = 1 : length(freq)
                   SigZVer(ifreq) =
GTI_SigmaSea(freq(ifreq),SeaState,Pol,alpha,ThWind);
               end;
           case 'HYB'
               for ifreq = 1 : length(freq)
```

```
                   SigZVer(ifreq) =
HYB_SigmaSea(freq(ifreq),SeaState,Pol,alpha,ThWind);
               end;
           case 'TSC'
               for ifreq = 1 : length(freq)
                   SigZVer(ifreq) =
TSC_SigmaSea(freq(ifreq),SeaState,Pol,alpha,ThWind);
               end;
       end;

%       SigZVer = VGHSigmaSeaNew(freq,SeaState,Pol,alpha,ThWind);


      SigmaVV = SigZVer;

      NOWColorV=ColsMat(SeaSt+1,:)

      hl(end+1) = semilogx(freq,SigmaVV,'LineWidth',7);
      set(hl(end),'Color',ColsMat(SeaSt+1,:));
      hold on
   end

   for SeaSt=0:6
      SeaState=SeaSt;
      SS_Plus1 = SeaState + 1;
      Pol = 'V';
      % Plot measured values from Nathanson
      for ii=1:length(Freq_Nath)
         if Measured_SigmaVV(SS_Plus1,ii) ~= 0
            hp(end+1) =
semilogx(Freq_Nath(ii),Measured_SigmaVV(SS_Plus1,ii),'o','LineWidth',CircSize
);
            %htemp=semilogx(Freq_Nath(ii),Measured_SigmaVV(SS_Plus1,ii),'-
','LineWidth',1);
            set(hp(end),'Color',ColsMat(SeaSt+1,:));
            hold on
            fGHz = Freq_Nath(ii);
            Pol = 'V';

            switch upper(Model)
                case 'NRL'
                    SigZErr = abs(Measured_SigmaVV(SS_Plus1,ii)-
NRL_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
                case 'GTI'
                    SigZErr = abs(Measured_SigmaVV(SS_Plus1,ii)-
GTI_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
                case 'HYB'
                    SigZErr = abs(Measured_SigmaVV(SS_Plus1,ii)-
HYB_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
                case 'TSC'
                    SigZErr = abs(Measured_SigmaVV(SS_Plus1,ii)-
TSC_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
            end;

%             SigZErr = abs(Measured_SigmaVV(SS_Plus1,ii)-
NRL_SigmaSea(fGHz,SeaState,Pol,alpha,ThWind));
            SumDevV = SumDevV + SigZErr;
```

```
            NValV = NValV + 1;
         end;
      end

      hold on;
   end     % END Sea State Loop

   grid on;

   set(gca,'YLim',[minyaxis maxyaxis],'XLim',[minxaxis maxxaxis]);
   htx(end+1) = xlabel('Frequency - f (MHz) ');
   hty(end+1) = ylabel('Reflectivity - \sigma_{VV}^o (dB)');
   %title('DOPPLER ESTIMATION','FontSize',20,'FontWeight','bold');


   grid on;

   %set(ha,'YLim',[minyaxis maxyaxis],'XLim',[minxaxis maxxaxis]);
   set(ha,'FontWeight','Bold');
   set(ha,'FontSize',FigFont-1);
   set(htx,'FontSize',FigFont);
   set(htx,'FontWeight','Bold');
   set(hty,'FontSize',FigFont);
   set(hty,'FontWeight','Bold');
   set(hl,'LineWidth',LineSize);
   set(hp,'LineWidth',CircSize);
   set(ha,'LineWidth',AxWidth);
   set(ha,'xtick',[0.5,.7,1.0,2.0,3.0,5.0,7.0,10.0,20.0,...
        30.0,40.0]);
   set(ha,'XLim',[0.5 40])

   whitebg(gcf,[1 1 1])

   set (gca,'PlotBoxAspectRatio',[1 .5 1])
   FilFig = ['VV_VGHNew' num2str(round(10*alpha))];
   saveas(gcf, [FilFig '.jpg'])

end     % END Grazing Angle Loop


% Compute Average Absolute Difference
MeanAbsDev_Hor = SumDevH/NValH
MeanAbsDev_Vert = SumDevV/NValV
```

Figure 26: MATLAB® listing of SigmaSea_vs_Freq.m

- SigmaSea_vs_GrazAng.m – This code is used to generate Figures 17 through 21 where sea reflectivity (Horizontal and Vertical pol) is plotted vs. grazing angle for the following frequencies: 0.5GHz, 1.25GHz, 5.6GHz, 17GHz and 35GHz. The "circles" represent the tabulated data from Nathanson, while the solid lines is the data from any of the four models: NRL, GTI, TSC and HYB. For the five figures, the solid lines represent the NRL model data.

```
% Program plots the NEW 5-PARAMETER VGH empirical sea clutter model
% e:\work\matldat\seacl\VGH_SeaClutterModelNewGrazAng.m
% Also plots points from Nathanson 2nd ed.
% Now optimized to 60 deg grazing angle
% Average absolute error to 60 deg:   2.64 Hor, 2.54 V

%Author: Rashmi Mital
%Date: 11/30/2007
clc;
clear all;
close all;

%DEFINE CONSTANTS
c = 2.997924562e8; %Speed of Light
dtor = pi/180;

% Define INPUTS
%Grazing angle (data available for 0.1 0.3 1.0 3.0 10.0 30.0 60.0)
GrAng =[0.1 0.3 1.0 3.0 10.0 30.0 60.0];
%GrAng =[0.1 0.3 1.0 3.0 10.0];
%GrAng =[0.3];
NGrAng = length(GrAng);

% Nathanson Frequencies
Freq_Nath = [0.5 1.25 3 5.6 9.3 17 35];
NFreq = length(Freq_Nath);

phi = 0; %Angle betwen boresight and upwind (deg), only affects GTRI model
phi_rad = phi .* dtor;

% Just a dummy in this program
ThWind = 0;

% Grazing Angle axis for plotting empirical curves
GrAngPlot = (0.1:0.01:60); %Frequency in GHz

GrAngMin=GrAng(1);
GrAngMax=max(GrAng);
%GrAngMax=freq(end);



%Choose Model to Compare with Nathanson Data
Model = 'NRL'; %Option: 'NRL', 'GTI','HYB','TSC'

% GRAPHICS DEFINITIONS
```

```
ha=[];hl=[];hp=[];ht=[];htx=[];hty=[];
set(0,'Units','pixels')

% Set SCreen Size
scnsize=get(0,'ScreenSize')
pos1  = [5+scnsize(1),0.02*scnsize(4),.99*scnsize(3),.75*scnsize(4)];

FigFont = 28;
CircSize = 14;
LineSize = 7;
AxWidth = 4;

% Use these for saved jpg figures
% FigFont = 21;
% CircSize = 7;
% LineSize = 4;
% AxWidth = 3;

% Initialize figure number
nFig = 0;
% Defines units for values returned by ScreenSize

% Define colors for curves and points
Cols = 'bgrcmyk'

ColsMat = [ 0.161 0.698 0.725;    % magenta
            0.608 0.18  0.8;      % violet
            1.0   0.0   0.0;      % red
            0.541 0.416 0.361;    % brown
            0.067 0.204 0.898;    % dark blue
            0.0   0.0   0.0;      % black
            0.224 0.678 0.333];   % dark green

%#########################################################################
% READ ALL NATHANSON TABLES

   % Initialize Tables
   Measured_SigmaHH=zeros(7,7,7);   % SSPlus1,Freq,GrazAng
   Measured_SigmaVV=zeros(7,7,7);

      Measured_SigmaHH(:,:,1) = ...
         [ 0    0 -90 -87   0   0    0;
         0    0 -80 -75 -71   0    0;
         -95  -90 -75 -67 -59 -48    0;
         -90  -82 -68 -69 -53   0    0;
         0  -74 -63 -60 -48   0    0;
         0  -70 -63 -58 -42   0    0;
         0    0   0   0   0   0    0];

      Measured_SigmaVV(:,:,1) = ...
         [ 0    0   0   0   0   0    0;
         0    0 -80 -72 -65   0    0;
         -90  -87 -75 -67 -56   0    0;
```

```
   -88  -82 -75 -60 -51   0  -47;
   -85  -78 -67 -58 -48   0  -45;
   -80  -70 -63 -55 -44   0  -42;
   0    0 -56   0   0   0    0];

Measured_SigmaHH(:,:,2) = ...
   [  0    0 -83 -79 -74   0   0;
   0    0 -74 -68 -66 -58   0;
   -78    0 -66 -60 -56 -53   0;
   0  -68 -60 -50 -46 -42   0;
   0    0 -55   0 -42 -39   0;
   0  -64 -52 -44 -39 -38   0;
   0    0 -46   0 -34 -37   0];

Measured_SigmaVV(:,:,2) = ...
   [  0  -83   0   0   0 -63 -55;
   0  -78 -64 -60 -58 -54 -46;
   -80  -73 -62 -55 -52 -52 -43;
   -78  -70 -58 -50 -45 -47 -40;
   -75  -65 -57   0 -43 -44 -38;
   -73  -64 -52   0 -39 -39 -35;
   0    0   0   0 -34 -37 -31];

Measured_SigmaHH(:,:,3) = ...
   [-86 -80 -75 -70 -60 -60 -60;
   -84 -73 -66 -56 -51 -48 -48;
   -82 -65 -55 -48 -46 -41 -38;
   -73 -60 -48 -43 -40 -37 -36;
   -63 -56 -45 -39 -36 -34 -34;
   -60 -50 -42 -36 -34 -34   0;
   0   0 -41   0 -32 -32   0];

Measured_SigmaVV(:,:,3) = ...
   [     0 -68   0   0 -60 -60 -60;
   -70 -65 -56 -53 -50 -50 -48;
   -63 -58 -53 -47 -44 -42 -40;
   -58 -54 -48 -43 -39 -37 -34;
   -58 -45 -42 -39 -37 -35 -32;
   0 -43 -38 -35 -33 -34 -31;
   0   0 -33   0 -31 -32   0];

Measured_SigmaHH(:,:,4) = ...
   [-75 -72 -68 -63 -58   0 -53;
   -70 -62 -59 -54 -50 -45 -43;
   -66 -59 -53 -48 -43 -38 -40;
   -61 -55 -46 -42 -39 -35 -37;
   -54 -48 -41 -38 -35 -32 -32;
   -53 -46 -40 -36 -33 -30   0;
   0   0 -37   0 -30 -28   0];

Measured_SigmaVV(:,:,4) = ...
   [  0   0   0 -60 -56 -50 -48;
   -60 -53 -52 -49 -45 -41 -41;
   -53 -50 -49 -45 -41 -39 -37;
```

```
   -43 -43 -43 -40 -38 -36 -34;
   -38 -38 -38 -36 -35 -33 -31;
   -40 -38 -35 -35 -33 -31 -30;
   0   0   0   0 -28 -28   0];

Measured_SigmaHH(:,:,5) = ...
   [   0 -60   0 -44 -56   0   0;
   0 -56   0 -53 -51   0   0;
   -54 -53 -51 -48 -43 -37 -36;
   -50 -48 -46 -40 -37 -32 -31;
   -48 -45 -40 -36 -34 -29 -29;
   -46 -43 -38 -36 -30 -26 -27;
   -44 -40 -37 -35 -27 -24   0];

Measured_SigmaVV(:,:,5) = ...
   [   0 -45   0 -44 -47 -45 -44;
   -38 -39 -40 -41 -42 -40 -38;
   -35 -37 -38 -39 -36 -34 -33;
   -34 -34 -34 -34 -32 -31 -31;
   -32 -31 -31 -32 -29 -28 -29;
   -30 -30 -28 -28 -25 -23 -26;
   -30 -29 -28 -27 -22 -18   0];

Measured_SigmaHH(:,:,6) = ...
   [  0 -50 -50 -50 -48 -45   0;
   0 -46   0 -48 -44 -38   0;
   -42 -41 -40 -42 -38 -35 -35;
   -40 -39 -38 -37 -34 -28 -29;
   -38 -37 -37 -35 -29 -21 -21;
   -35 -34 -32 -30 -26 -18 -20;
   -33 -32 -30 -29 -21 -16   0];

Measured_SigmaVV(:,:,6) = ...
   [   0 -42 -42 -42 -37 -33   0;
   -38 -38 -40 -42 -36 -31 -35;
   -30 -31 -32 -34 -32 -26 -30;
   -28 -30 -29 -28 -26 -23 -23;
   -28 -28 -27 -25 -24 -22 -22;
   -28 -24 -23 -22 -22 -18 -20;
   -25 -32 -30 -29 -21 -16   0];

Measured_SigmaHH(:,:,7) = ...
   [  -32 -32 -32 -27 -25 -22 -26;
   -22 -24 -25 -26 -24 -20   0;
   -22 -21 -21 -22 -23 -18   0;
   -21 -20 -20 -20 -21 -16 -16;
   -21 -18 -17 -16 -15 -12 -12;
   -21 -18 -17 -17 -14 -10   0;
   -20 -19 -17 -16 -12 -10   0];

Measured_SigmaVV(:,:,7) = ...
   [-32 -33 -34 -26 -23 -22   0;
   -23 -22 -24 -24 -24 -20 -24;
   -20 -21 -21 -23 -18 -18 -19;
```

```
        -18 -18 -19 -18 -16 -14 -14;
        -14 -15 -15 -15 -14 -11 -10;
        -18 -15 -15 -15 -13 -11  -4;
        -18 -17 -15 -14 -11 -10   0];



   for iFreq = 1:NFreq
   FNow = Freq_Nath(iFreq);
   nFig = nFig + 2;

   % Default plot limits
   SigmaMin=-100;
   SigmaMax=0;

   % Define Plot Axes
   minyaxis=SigmaMin;
   maxyaxis=SigmaMax;
   minxaxis=GrAngMin;
   maxxaxis=GrAngMax;


   hFig=figure(nFig-1);   % Plot Hor Pol

   set(hFig,'Position',pos1);

   ht(end+1)=gca;
   ha(end+1)=gca;
   set (ha(end),'PlotBoxAspectRatio',[1 .2 .5])

   % Do all sea states for HH
   for SeaSt=0:6
      SeaState=SeaSt;
      SS_Plus1 = SeaState + 1;

      Pol = 'H';

      switch upper(Model)
          case 'NRL'
              SigZHor =
NRL_SigmaSea(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot,ThWind);
          case 'GTI'
              for iAng = 1 : length(GrAngPlot)
                  SigZHor(iAng) =
GTI_SigmaSea(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot(iAng),ThWind);
              end;
          case 'HYB'
              for iAng = 1 : length(GrAngPlot)
                  SigZHor(iAng) =
HYB_SigmaSea(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot(iAng),ThWind);
              end;
          case 'TSC'
```

```
              for iAng = 1: length(GrAngPlot)
                  SigZHor(iAng) =
TSC_SigmaSea(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot(iAng),ThWind);
              end;
      end;

%       SigZHor =
VGHSigmaSeaNew(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot,ThWind);

      SigmaHH = SigZHor;

      % Plot empirical curve first
      NOWColorH=ColsMat(SeaSt+1,:)
      %figure(nFig-1)
      %       hl(end+1) = semilogx(freq,SigmaHH,'LineWidth',LineSize);
      hl(end+1) = semilogx(GrAngPlot,SigmaHH);
      hold on
      set(hl(end),'Color',ColsMat(SeaSt+1,:));
   end

   for SeaSt=0:6
      SeaState=SeaSt;
      SS_Plus1 = SeaState + 1;
      % Plot measured values from Nathanson
      for ii=1:NGrAng
         if Measured_SigmaHH(SS_Plus1,iFreq,ii) ~= 0
            hp(end+1) =
semilogx(GrAng(ii),Measured_SigmaHH(SS_Plus1,iFreq,ii),'o');
            set(hp(end),'Color',ColsMat(SeaSt+1,:)); % Use as example for
custom colors
            hold on

         end;
      end
   end     % END Sea State Loop

   grid on;



   set(gca,'YLim',[minyaxis maxyaxis],'XLim',[minxaxis maxxaxis]);
   htx(end+1) = xlabel('Grazing Angle - \beta (deg) ');
   hty(end+1) = ylabel('Reflectivity - \sigma_{HH}^0 (dB)');

   %set(ha,'YLim',[minyaxis maxyaxis],'XLim',[minxaxis maxxaxis]);
   set(ha,'FontWeight','Bold');
   set(ha,'FontSize',FigFont-2);
   set(ha,'FontName','Arial')
   set(htx,'FontSize',FigFont);
   set(htx,'FontName','Arial')
   set(htx,'FontWeight','Bold');
   set(hty,'FontSize',FigFont);
   set(hty,'FontName','Arial')
```

```
   set(hty,'FontWeight','Bold');
   set(hl,'LineWidth',LineSize);
   set(hp,'LineWidth',CircSize);
   set(ha,'LineWidth',AxWidth);
   set(ha,'xtick',[0.1,.2,0.3,0.5,0.7,1.0,2.0,3.0,4.0,7.0,10.0,20.0,...
        30.0,50.0,70.,90.]);
   set(ha,'XLim',[0.1 90])

   whitebg(gcf,[1 1 1])

   set (gca,'PlotBoxAspectRatio',[1 .7 1])
   FilFig = ['HH_VGHNew' num2str(round(10*FNow))];
   saveas(gcf, [FilFig '.jpg'])
   %pause

   hFig=figure(nFig);   % Plot Vert Pol First

   ht(end+1)=gca;
   ha(end+1)=gca;
   set(gcf,'Position',pos1);
   %set (ha(end),'PlotBoxAspectRatio',[1 .7 1])

   % Do all sea states for VV
   for SeaSt=0:6
      SeaState=SeaSt;
      SS_Plus1 = SeaState + 1;
      Pol = 'V';

       switch upper(Model)
           case 'NRL'
               SigZVer =
NRL_SigmaSea(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot,ThWind);
           case 'GTI'
               for iAng = 1: length(GrAngPlot)
                   SigZVer(iAng) =
GTI_SigmaSea(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot(iAng),ThWind);
               end;
           case 'HYB'
               for iAng = 1: length(GrAngPlot)
                   SigZVer(iAng) =
HYB_SigmaSea(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot(iAng),ThWind);
               end;
           case 'TSC'
               for iAng = 1:length(GrAngPlot)
                   SigZVer(iAng) =
TSC_SigmaSea(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot(iAng),ThWind);
               end;
       end;
%       SigZVer =
VGHSigmaSeaNew(Freq_Nath(iFreq),SeaState,Pol,GrAngPlot,ThWind);
      SigmaVV = SigZVer;

      NOWColorV=ColsMat(SeaSt+1,:)
```

```
      hl(end+1) = semilogx(GrAngPlot,SigmaVV,'LineWidth',7);
      set(hl(end),'Color',ColsMat(SeaSt+1,:));
      hold on
   end

   for SeaSt=0:6
      SeaState=SeaSt;
      SS_Plus1 = SeaState + 1;
      Pol = 'V';
      % Plot measured values from Nathanson




            for ii=1:NGrAng
         if Measured_SigmaVV(SS_Plus1,iFreq,ii) ~= 0

            hp(end+1) =
semilogx(GrAng(ii),Measured_SigmaVV(SS_Plus1,iFreq,ii),'o','LineWidth',CircSi
ze);
            %htemp=semilogx(Freq_Nath(ii),Measured_SigmaVV(SS_Plus1,ii),'-
','LineWidth',1);
            set(hp(end),'Color',ColsMat(SeaSt+1,:));
            hold on

         end;
      end

      hold on;
   end     % END Sea State Loop

   grid on;

   set(gca,'YLim',[minyaxis maxyaxis],'XLim',[minxaxis maxxaxis]);
   htx(end+1) = xlabel('Grazing Angle - \beta (deg) ');
   hty(end+1) = ylabel('Reflectivity - \sigma_{VV}^0 (dB)');
   %title('DOPPLER ESTIMATION','FontSize',20,'FontWeight','bold');



   grid on;

   %set(ha,'YLim',[minyaxis maxyaxis],'XLim',[minxaxis maxxaxis]);
   set(ha,'FontWeight','Bold');
   set(ha,'FontSize',FigFont-2);
   set(ha,'FontName','Arial')
   set(htx,'FontSize',FigFont);
   set(htx,'FontName','Arial')
   set(htx,'FontWeight','Bold');
   set(hty,'FontSize',FigFont);
   set(hty,'FontName','Arial')
   set(hty,'FontWeight','Bold');
   set(hl,'LineWidth',LineSize);
```

```
    set(hp,'LineWidth',CircSize);
    set(ha,'LineWidth',AxWidth);
    set(ha,'xtick',[0.1,.2,0.3,0.5,0.7,1.0,2.0,3.0,4.0,7.0,10.0,20.0,...
         30.0,50.0,70.,90.0]);
    set(ha,'XLim',[0.1 90])

    whitebg(gcf,[1 1 1])

    set (gca,'PlotBoxAspectRatio',[1 .7 1])
    FilFig = ['VV_VGHNew' num2str(round(10*FNow))];
    saveas(gcf, [FilFig '.jpg'])

end     % END Frequency Loop
```

Figure 27: MATLAB® listing of SigmaSea_vs_GrazAng.m