RAIN RATE RETRIEVAL ALGORITHM FOR AQUARIUS/SAC-D MICROWAVE RADIOMETER

by

ROSA ANA MENZEROTOLO
B.S. University of Central Florida, 2005

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, FL

Fall Term
2010

ABSTRACT

Microwave radiometers are used to measure blackbody microwave emissions emitted by natural targets. Radiative transfer theory provides a well founded physical relationship between the atmosphere and surface geophysical parameters and the brightness temperature measured by these radiometers. The atmospheric brightness temperature is proportional to the integral of the microwave absorption of water vapor, oxygen, and liquid water between the top of the atmosphere and the surface. Inverse radiative transfer models use to retrieve the water vapor, cloud liquid and oxygen content in the atmosphere are very well known; however, the retrieval of rain rate in the atmosphere is still a challenge.

This project presents a theoretical basis for the rain rate retrieval algorithm, which will be implemented in the Aquarius/SAC-D Microwave Radiometer (MWR). This algorithm was developed based on the radiative transfer model theory for a single layer atmosphere using four WindSat channels. Transmissivity due to liquid water (rain and cloud liquid water) is retrieved from the four channel brightness temperatures, and a statistical regression is performed to relate the rain rate, rain physical temperature and rain height to the liquid water transmissivities at 24 GHz and 37 GHz. Empirical validation results are presented using the WindSat radiometer observations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

Aquarius/SAC-D is a joint mission of discovery between National Aeronautics and Space Administration (NASA) and Argentina's National Commission of Space Activities (Comisión Nacional de Actividades Espaciales - CONAE).

Aquarius/SAC-D's main objective is to provide accurate global salinity measurements using only satellite observations. These salinity measurements will allow the study and better understanding of our oceanic activities, which includes the water cycle in planet Earth and the effect of the Oceans on weather and climate. [8]

The Aquarius/SAC-D satellite has six instruments two of which are microwave sensors: Aquarius and Microwave Radiometer (MWR). Figure 1-1 shows the Aquarius/SAC-D satellite with NASA's Aquarius instrument, a L-Band, 3-beam, pushbroom radiometer/scatterometer, which will provide global sea surface salinity measurements every 7 days. The MWR, CONAE's microwave radiometer shown in Fig. 1-2, is an 8 beam pushbroom, with a three channel Dicke radiometer that operates at 23.8 GHz (horizontal polarization) and 36.5 GHZ (vertical and horizontal polarization). The MWR measurements of ocean wind speed, rain rate and sea ice concentration are valuable ancillary measurements that aid the Aquarius salinity retrieval.

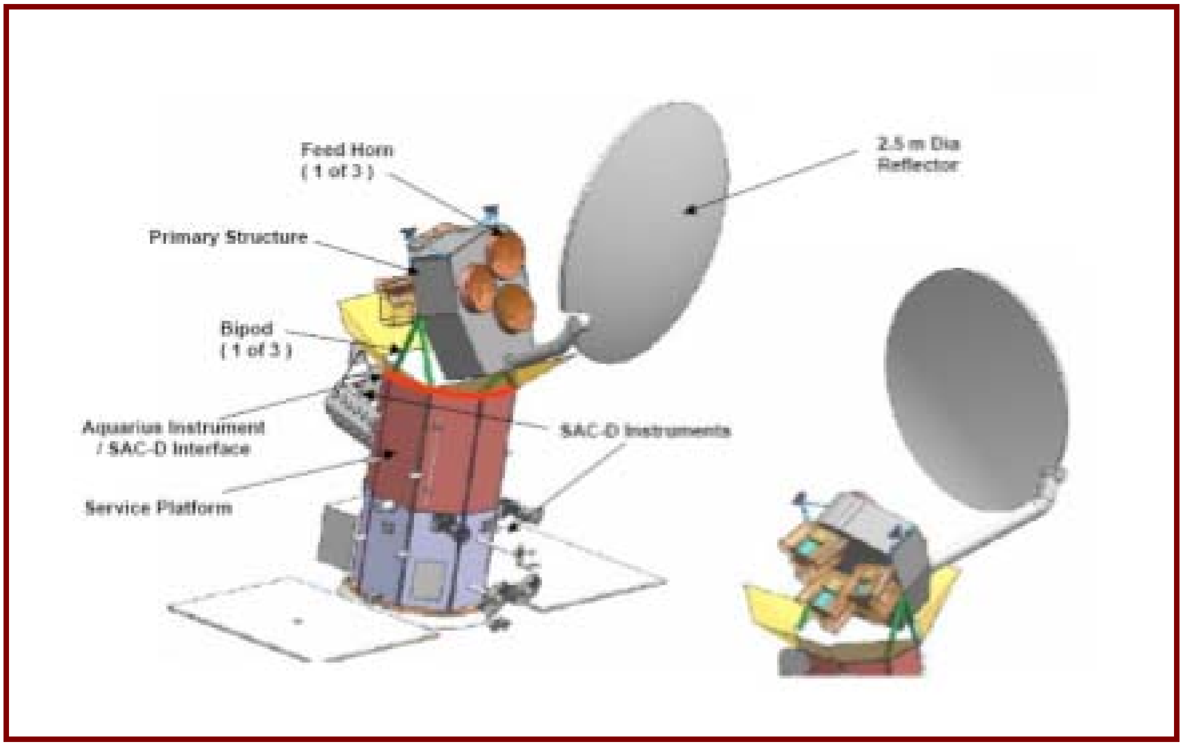


Figure 1-1 Aquarius/SAC-D Satellite – Aquarius instrument

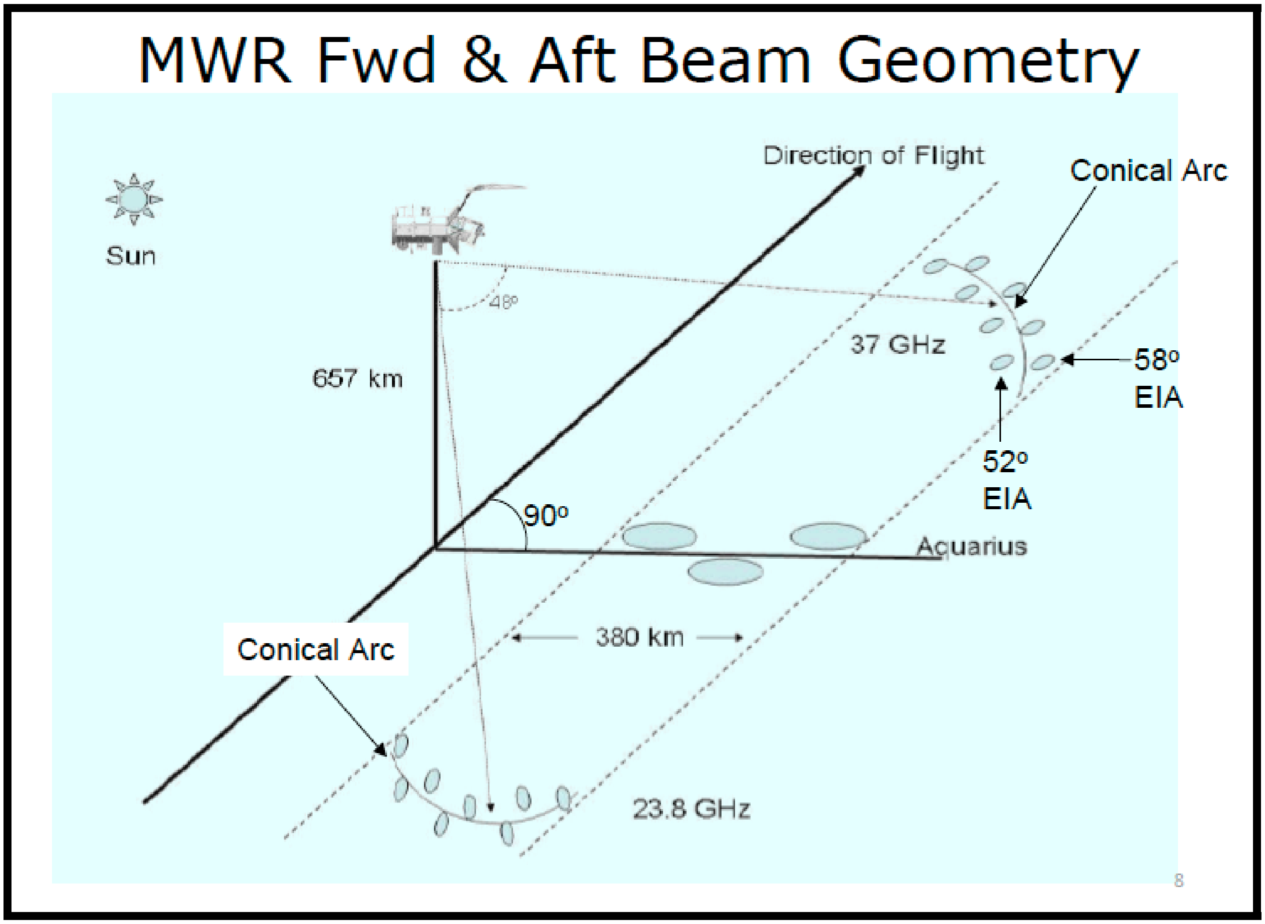


Figure 1-2: MWR Fwd and AFT Beam Geometry

The importance of MWR to the Aquarius mission can be understood by examining the Aquarius salinity measurement error budget. The Aquarius brightness temperature error allocation for the salinity measurement is 0.38K, which corresponds to a root mean square (rms) ocean surface salinity error of 0.2 psu. Aquarius salinity retrievals need to be unbiased and independent from environmental parameters. Based on results from an error analysis comparing all geophysical parameters and their effects on the salinity accuracy measurements, it was shown that surface roughness generates the greatest error, rain is the second greatest, and sea ice concentration produces a lesser error [8]. A major objective for MWR is to improve Aquarius salinity retrievals by performing simultaneous collocated geophysical measurements and by developing brightness temperature corrections due to surface roughness, rain, and sea ice concentration within the Aquarius radiometer footprint.

This thesis deals with only one of the above, namely, the development of a MWR rain rate retrieval algorithm. Because MWR has only 3 radiometer channels, this research presents a major challenge. Typically rain rate retrieval algorithms involve a complex relationship with environmental parameters that requires many microwave radiometer channels (dual polarized), where frequencies range from 10 GHz to 87 GHz [8]. Thus, the MWR rain rate retrieval will be highly constrained, with the need to provide many geophysical parameters as a priori information. Further, the algorithm will be forced to rely on statistical correlation between rain rate and the brightness temperature of the 3 radiometer channels. As a result, there is a strong desire to use real brightness measurements of these channels in the algorithm development. Since Aquarius/SAC-D has not yet launched, we rely on other satellite radiometers that are currently operating to simulate MWR measurements.

Fortunately, the WindSat radiometer on the Navy’s Coriolis satellite flies in a similar orbit as Aquarius/SAC-D, and also the MWR channels are a subset of the WindSat channels. As a result the WindSat environmental data records (EDR) and brightness temperatures (IDR) values are available for this project to generate pre-launch testing and simulation data.

The main objective of this thesis was to develop a rain rate retrieval algorithm for the Aquarius/SAC-D microwave radiometer MWR. This approach is based upon both radiative transfer theory and statistical regressions. The algorithm is tuned using empirical brightness temperature data from WindSat, and it is validated using WindSat environmental data records (EDR). Finally a secondary objective of this thesis is to contribute to the development of an algorithm theoretical basis document (ATBD) to give CONAE the specification and theoretical approach for developing processing software to be used post-launch.

CHAPTER 2. MICROWAVE RADIATIVE TRANSFER THEORY

Microwave brightness temperature measured by a satellite radiometer, shown in Fig 2-1, is the sum of three temperature components: the upwelling atmospheric brightness temperature, the downwelling atmospheric brightness temperature reflected at the ocean surface, and the ocean surface brightness temperature. This process is known as radiative transfer theory.

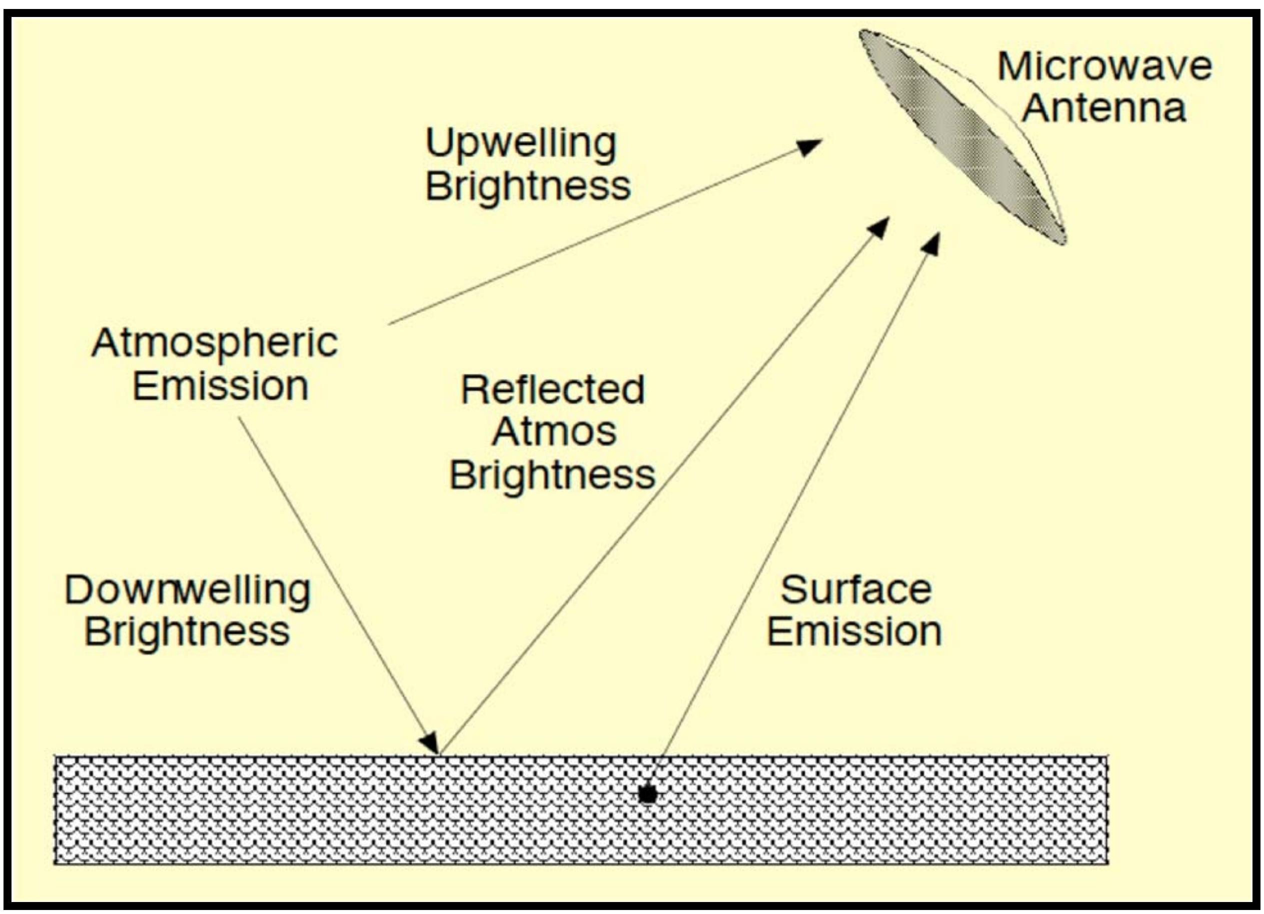


Figure 2-1 Simplified Radiative Transfer Model

The upwelling brightness temperature depends on the (absorption) emission of certain atmospheric gases (oxygen, water vapor) and liquid water contained in the Earth's atmosphere. The downwelling brightness temperature depends on the emissivity of the gases in the atmosphere for microwaves traveling from the atmosphere towards the ocean, being reflected by the surface, and traveling back through the atmosphere toward the radiometer. Surface brightness temperature depends upon the product of the sea surface temperature (SST) and the emissivity of the surface of the ocean for microwave emission into the atmosphere along the line of sight to the radiometer antenna.

In the RTM, there are two methods for calculating atmospheric emissivity; 1.) the radiative transfer model the atmosphere can appear as a single layer or 2.) be divided into many spherical shells, which are integrated (summed). In this retrieval algorithm, the atmosphere was modeled as a single layer with "equivalent" single-layer parameters.

The apparent brightness temperatures equations for a single layer atmosphere approximation are shown below.

$$T_{ap} = T_{BU} + \tau_{atmos} * (T_b + T_{scat}) \qquad (2\text{-}1)$$

T_b is the surface brightness temperature, which is defined as the emissivity of the ocean surface times the physical temperature of the ocean. T_{ap} is the brightness temperature measured by the radiometer. T_{scat} is the scattered atmospheric temperature, which is defined as the temperature of the sky times the ocean surface reflectivity.

$$T_{scat} = (1 - \varepsilon) * (T_{BD} + \tau_{atmos} * T_{BC}) \qquad (2\text{-}2)$$

T_{scat} is defined as the temperature of the sky times the ocean surface microwave power reflectivity, which is 1 minus the emissivity of the ocean surface. τ_{atmos} represents the transmissivity of the atmosphere. T_{BC} is the space temperature, which has a value of 2.73K. T_{BD} is the downwelling brightness temperature.

$$T_{BU} = (1 - \tau_{atmos}) * T_U \qquad (2\text{-}3)$$

T_U is the effective air upwelling brightness temperature. T_{BU} is the upwelling brightness temperatures.

$$T_{BD} = (1 - \tau_{atmos}) * T_D \qquad (2\text{-}4)$$

T_D is the effective air downwelling brightness temperature.

The equations for atmospheric absorption and transmissivity of the atmosphere are shown in equation 2-5, through 2-8.

$$\alpha_{atmos} = \int_{0}^{TOA} K \ dZ \qquad (2.5)$$

where α_{atmos} is the atmospheric absorption.

$$\alpha_{atmos} = \alpha_O + \alpha_V + \alpha_L \qquad (2.6)$$

α_O is the oxygen absorption; α_V is the water vapor absorption; α_L is the liquid water (cloud liquid water and rain) absorption coefficient.

$$\tau_{atmos} = \exp[-\sec\theta * (\alpha_O + \alpha_V + \alpha_L)] \qquad (2.7)$$

τ_{atmos} is the transmissivity of the atmosphere and θ represents the incidence angle.

$$\tau_{atmos} = \tau_O + \tau_V + \tau_L \qquad (2.8)$$

τ_O is the transmissivity due to oxygen, τ_V is the transmissivity due to water vapor, and τ_L is the transmissivity due to liquid water.

The theoretical basis for the rain retrieval algorithm comes from radiative transfer theory as defined in three journal papers by Wentz et al. [1, 2, 3]. These papers describe ocean and atmospheric environmental parameter retrievals for the SSMI and AMSR satellite radiometers. We start with considering the four WindSat channels 24 GHz and 37 GHz for V- and H-pols, where the forward radiative model equation relates the environmental parameters to the apparent brightness temperature measured by the radiometer as:

$$\boxed{F_{freq}(W,V,\tau_L) = T_{BU} + \tau_{atmos} \times [\varepsilon \times SST + (1-\varepsilon) \times (T_{BD} + \tau_{atmos} \times T_{BC})]} \qquad (2.9)$$

The relationship between upwelling brightness and the atmosphere effective air temperature for upwelling is given in equation 2-3, and a similar relationship for the downwelling brightness temperature and effective air temperature for downwelling is shown in equation 2-4. According to Wentz [3], the effective air temperatures can be parameterized as functions of sea surface temperature (SST), atmospheric water vapor, and radiometer channel frequency.

The ocean surface emissivity is the combination of smooth ocean surface emissivity plus a smaller additive excess emissivity due to wind. The smooth ocean surface emissivity is depends upon the dielectric constant of seawater, the radiometer incidence angle and the electromagnetic wave polarization. It should be noted that the dielectric constant of sea water depends on SST, salinity, and frequency. As a result, the surface emissivity ends up being a function of wind speed, SST, salinity, incidence angle, polarization and frequency.

By substituting the upwelling brightness temperature, downwelling brightness temperature, surface emissivity, and atmospheric transmissivity into equation 2-9, the apparent brightness temperature measured by the radiometer ends up being a function of three unknown environmental parameters: water vapor, wind speed, and transmissivity of liquid water. This equation is discussed in greater detail in the next chapter.

CHAPTER 3. ALGORITHM DEVELOPMENT APPROACH

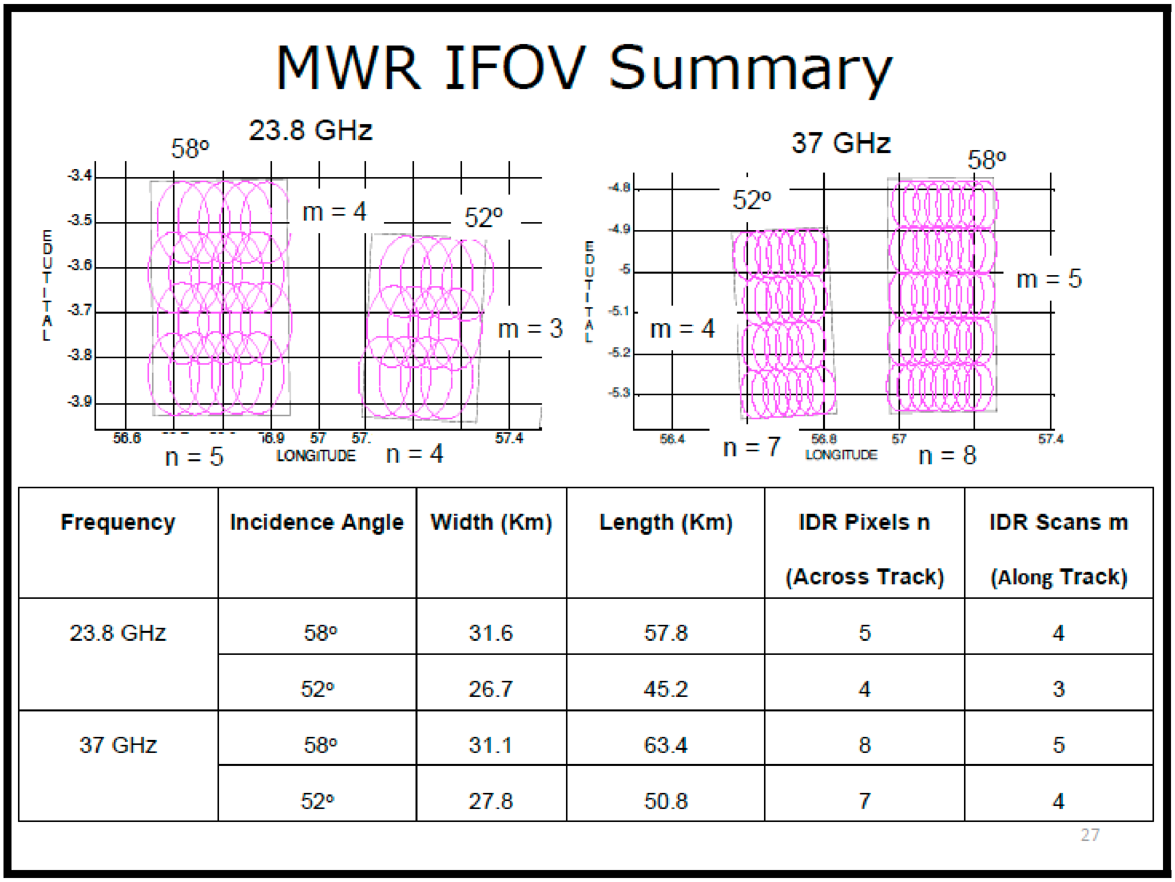
3.1. WindSat Selection

It is highly desirable to use actual satellite radiometer data to provide pre-launch testing of the MWR rain retrieval algorithm. Fortunately, the WindSat Polarimetric Radiometer and MWR are both in similar polar sun synchronous orbits. This allows the collocation of WindSat swath into MWR swath in order to perform pre launch tests and simulations. Further, both radiometers have many similarities as described below.

The WindSat radiometer contains 22 channels: 6.8 GHz, 10.7 GHz, 18.7 GHz, 23.8 GHz, 36.5 GHz. Channels 10.7 GHz, 18.7 GHz and 36.5 GHz are polarimetric, and channels 6.8 GHz and 23.8 GHz are vertically and horizontally polarized. WindSat has a large conically scanning antenna with a fixed incident angle of 53 degrees for the 23.8 and 36.5 GHz channels. This results in brightness temperatures measurements over approximately a 1,000 km swath. The spatial resolutions (instantaneous field of view, IFOV) for these channels are approximately 12 x 20 km and 8 x 13 km respectively.

MWR channels are a subset of WindSat channels: 23.8GHz horizontal polarization and 36.5GHz vertical and horizontal polarization. The MWR antenna has 16 fixed antenna beams in a pushbroom configuration as shown in Fig. 1-2 (8 @ 36.5 GHz looking forward and 8 @23.8 GHz looking aft). The feeds are staggered which results in MWR incident angles of 52 degrees and 58 degrees with IFOV's of approx 30 x 50 km. For this thesis, we selected only the 52 degree incidence angle, which is comparable to the WindSat 53 degree value.

A study conducted by Khan [4] indicates that MWR IFOV's can be formed by compositing WindSat Intermediate Data Record (IDR) measured brightness temperature in all its channels as shown in Fig. 3-1. This process expedites the simulation of MWR data by using WindSat IDR's. Also available are WindSat Environmental Data Records (EDR), which contain surface wind speed, integrated water vapor, cloud liquid water and rain retrievals. These two data sets provide the necessary information to implement the radiative transfer theory model required to develop and test the rain retrieval algorithm.



Frequency	Incidence Angle	Width (Km)	Length (Km)	IDR Pixels n (Across Track)	IDR Scans m (Along Track)
23.8 GHz	58°	31.6	57.8	5	4
	52°	26.7	45.2	4	3
37 GHz	58°	31.1	63.4	8	5
	52°	27.8	50.8	7	4

Figure 3-1: MWR IFOV Summary from Khan [4].

3.2. WindSat SDR, WindSat EDR and GDAS Collocation

As previously mentioned, WindSat is a sun-synchronized satellite that completes about 14 orbits per day. Because of the conically scanning antenna and orbit geometry, each orbit contains a forward looking swath of approximately 1000 km where the brightness temperatures (Tb) Intermediate Data Record (IDR) and the Environmental Data Record

(EDR) may be collocated. The IDR and EDR are provided in separate data files that have been sampled at fixed azimuth steps and stored per antenna revolution also known as a scan. Both data records contain the WindSat data and surface location (latitude/longitude) for each pixel. Further each file has meta-data that provides the date, start time, finish time, orbit revolution number of the satellite, and other identifiers (e.g. software version).

For retrievals, simulations and test purposes, WindSat SDR and EDR were collocated in two ways. The first collocation was generated at the highest spatial resolution for developing and testing the rain rate algorithm. For this purpose, WindSat IDR and EDR were earth gridded in 1/8 x 1/8 degree latitude/longitude boxes (global lat/lng grid: 1440 x 2880 pixels) on a single orbit basis.

The second collocation was generated to develop (and test) the geophysical retrieval algorithm for water vapor and wind speed. For this purpose, the IDR and EDR values were averaged in 1 degree latitude by 1 degree longitude boxes (global earth grid: 720 x 1440 pixels) per satellite revolution. Next, numeric weather model data from the National Oceanic and Atmospheric Administration (NOAA) Global Data Assimilation System (GDAS) were merged with these data to form a triple match-up data set. The GDAS data was used to provide independent validation for the WindSat EDR water vapor and wind speed data used to develop the retrieval algorithm.

These collocations were only generated on selected days of the year, which will be explained in the next chapters.

3.3. WindSat Brightness Temperature

Because the IDR contains ocean Tb's for all of the WindSat 22 channels; the first task was to extract the desired 23.8 GHz TbV & TbH and 36.5 GHz TbV & TbH data and save them as separate files of matching format (i.e., only pixels where all four Tb's were valid were kept). Each pixel contained a vector of parameters including the latitude, longitude, data quality flags, etc. Bad Tb's are represented as "-9999". In order to filter these bogus pixels, all the data entries containing -9999 were identified and assigned to be "not a number" (NaN). Then, the brightness temperatures were earth gridded in 1/8 x 1/8 degree boxes ignoring the NaN entries. Where more than one pixel occupied the same box, their values were averaged. This procedure generated four 2-D matrices where each matrix contained the polarized brightness temperature for the given channel with its corresponding latitude and longitude.

3.4. WindSat Environmental Data Record (EDR)

The IDR brightness temperature data were binned into the EDR locations. Where either data type was missing, these pixel's were deleted.

Sometimes EDR pixels also contain bad data, which were represented with -9999 (same as bogus brightness temperatures). These bad data were identified and assigned as NaN. Then each EDR value (rain, water vapor, cloud liquid water, sea surface temperature, and wind speed) was earth gridded into 1/8 latitude by 1/8 longitude boxes, where pixels occupying the same grid box were averaged ignoring the NaN values. This procedure generated five 2-D matrices where each entry represents an EDR value with its corresponding longitude and latitude. After all the EDRs where allocated, the earth gridded EDR and IDR values were saved under the scan/revolution number given in the original file name.

3.5. GDAS

The environmental parameter data, obtained from the NOAA Global Data Assimilation System (GDAS) internet archive, provided global numerical weather model outputs four times daily at 0000, 0600, 1200 and 1800 GMT and on a 100 km grid (1 deg x 1 deg lat/lng grid). These data included atmospheric profiles of pressure, temperature, and water vapor at 21 pressure levels as well as columnar cloud liquid water, sea surface temperature, and ocean wind speed at 10 m height. The vertical profiles of water vapor were integrated to provide the columnar water vapor that corresponds to the radiometer retrieved water vapor. GDAS wind speeds were compared directly to the EDR wind speeds averaged over the 1 degree boxes.

3.6. Rain Rate Retrieval Algorithm

Given the approximation of a single layer atmosphere and the SST as a priori information, the radiative transfer forward model for the four WindSat channels reduces to four equations with four unknowns: water vapor, wind speed, atmospheric transmissivity at 24 GHz and 37 GHz due to liquid water. Normally the retrieval is the inversion of the forward RTM; however given the complexity of these four equations with the 4th orders of the unknown variables, the approach of simultaneous solutions are not an acceptable option due to intensive computer processing requirements. For the algorithm to be a viable solution for MWR, it must be simple and run faster than real time.

Given this issue, we elected to use statistical regression to solve for a single parameter in four sequential steps as shown in Fig. 3-2. The most robust statistical regression is for water vapor, so this was selected to be the first step. Next came the wind speed regression, which was followed by the 24 GHz and 37 GHz transmissivities due to liquid water, and finally rain rate. . These are described below.

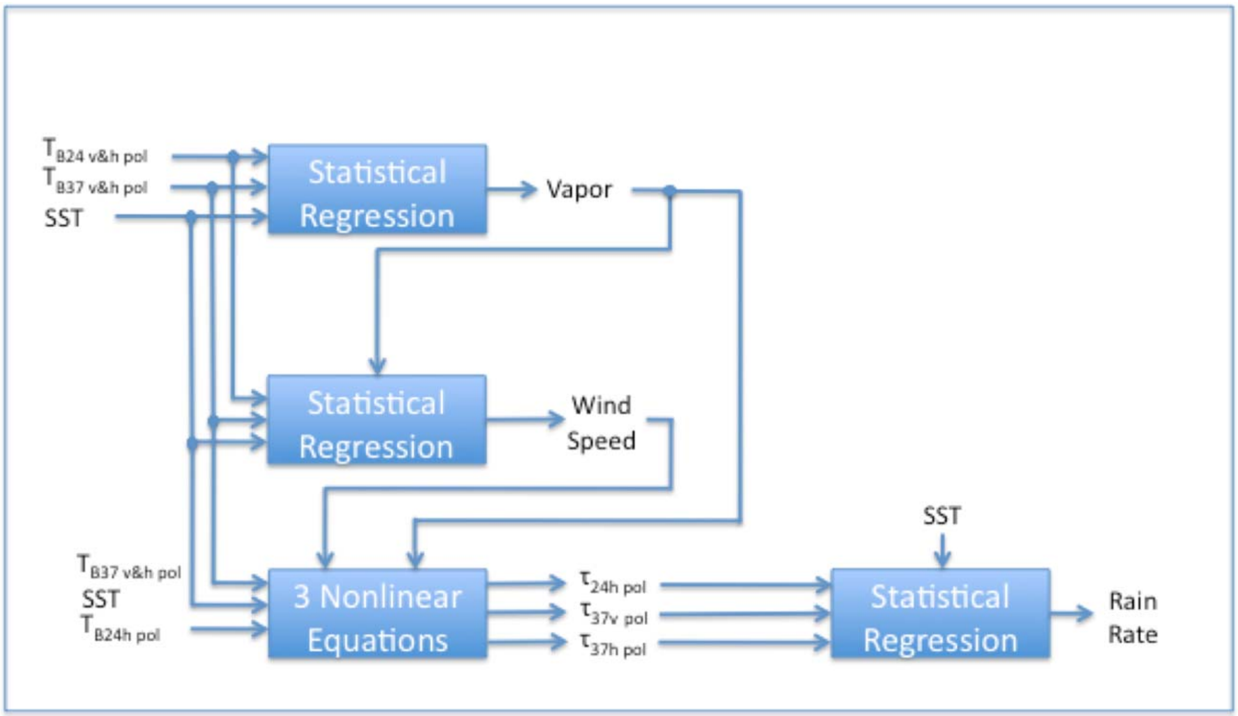


Figure 3-2: Rain Rate Algorithm Block Diagram

3.6.1. Water Vapor Statistical Regression

To develop the water vapor retrieval, a statistical regression was performed using a training data set of WindSat four channels Tb's and EDR water vapor and SST. To generate this data set, 48 days of data over a one year period were extracted from the collocated data set previously explained. The EDRs and IDRs were binned and averaged into 1 deg latitude by 1 deg longitude boxes on a per rev basis. All orbits (14) in one day sampled every two weeks were selected for the entire year (less two months without measurements). Boxes were edited to remove rain and heavy clouds (CLW > 0.5 mm). Then these data were divided into two sets: one set of data was used to perform the regression analysis to retrieve water vapor, and the other set was used to test the regression (results presented in Fig. 3-3).

To calculate water vapor content in the atmosphere, a second order statistical regression was performed, where the EDR water vapor value was the independent variable. The brightness temperatures for channels 24GHz and 37GHz both horizontal and vertical polarization, and sea surface temperature were the dependent variables in the statistical regression.

The regressed polynomial has the form shown below and the coefficient values are given in table 3-1.

$$
\begin{aligned}
&a_1 \times T_{B24v}^2 + a_2 \times (T_{B24v} + T_{B24h}) \\
&+a_3(T_{B24v} + T_{B37v}) + a_4(T_{B24v} + T_{B37h}) \\
&+a_5(T_{B24v} + SST) + a_6 \times T_{B24v} + a_7 \times T_{B24h}^2 \\
&+a_8(T_{B24h} + T_{B37v}) + a_9(T_{B24h} + T_{B37h}) \\
&+a_{10}(T_{B24h} + SST) + a_{11} \times T_{B24h} + a_{12} T_{B37v}^2 \\
&+a_{13}(T_{B37v} + T_{B37h}) + a_{14}(T_{B37v} + SST) \\
&+a_{15} \times T_{B37v} + a_{16} T_{B37h}^2 + a_{17}(T_{B37h} + SST) \\
&+a_{18} \times T_{B37h} + a_{19} \times SST^2 + a_{20} \times SST + a_{21}
\end{aligned}
\tag{3-1}
$$

Table 3-1 Water Vapor Regression Coefficient values

Coeff.	Value
a_1	8.4614E-03
a_2	1.4700E-03
a_3	-2.7612E-02
a_4	5.6412E-03
a_5	-1.3190E-03
a_6	1.9263E+00
a_7	6.1136E-05
a_8	9.8280E-03
a_9	-4.4122E-03
a_10	-4.1883E-03
a_11	-2.0565E-01
a_12	2.3767E-02
a_13	-2.4081E-03
a_14	-6.3752E-03
a_15	-3.4204E+00
a_16	1.6942E-04
a_17	3.3885E-03
a_18	-1.3714E+00
a_19	-2.2503E-04
a_20	1.7947E+00
a_21	4.5415E-02

Figure 3-3 shows a scatter plot of the calculated water vapor using the regression equation and the withheld Tb's compared with the corresponding EDR water vapor values. It can be seen that the regression equation water vapor reproduces the WindSat EDR water vapors extremely well.

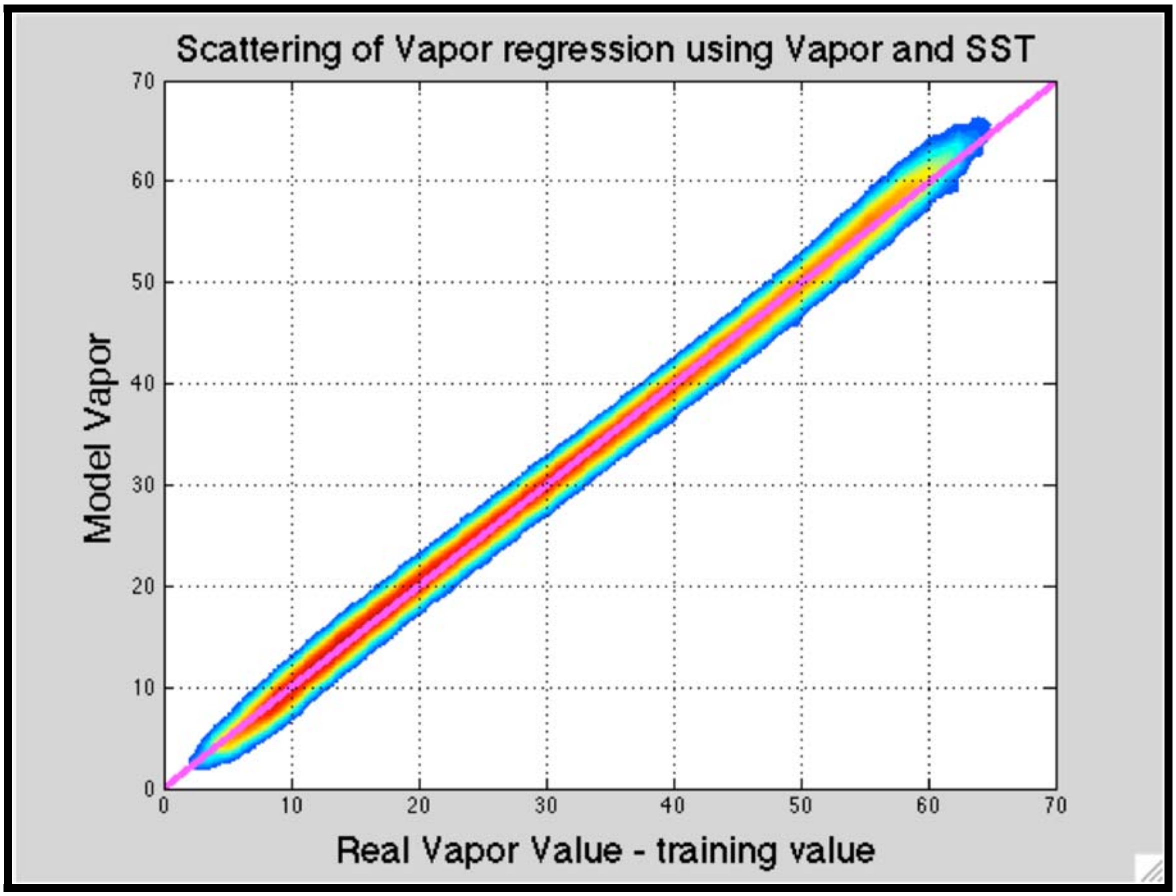


Figure 3-3: Water Vapor Comparison

3.6.2. Wind Speed Statistical Regression

In a similar manner, a statistical regression was performed to calculate the wind speed values. In this regression, the EDR wind speed value was the independent variable, and the brightness temperatures for channels 24GHz and 37GHz both horizontal and vertical polarization, sea surface temperature, and retrieved water vapor values were the dependent variables. The regression polynomial for wind speed has the form shown below, and the coefficient values are shown in table 3-2.

$$\begin{aligned}
& a_1 \times T_{B24v}^2 + a_2(T_{B24v} + T_{B24h}) + a_3(T_{B24v} + T_{B37v}) \\
& + a_4(T_{B24v} + T_{B37h}) + a_5(T_{B24v} + SST) + \\
& a_6(T_{B24v} + V) + a_7 \times T_{B24v} + a_8 \times T_{B24h}^2 \\
& + a_9(T_{B24h} + T_{B37v}) + a_{10}(T_{B24h} + T_{B37h}) + \\
& a_{11}(T_{B24h} + SST) + a_{12}(T_{B24v} + V) + a_{13} \times T_{B24h} \\
& + a_{14} \times T_{37v}^2 + a_{15}(T_{B37v} + T_{B37h}) + a_{16}(T_{B37v} + SST) \\
& + a_{17}(T_{B37v} + V) + a_{18} \times T_{B37v} + a_{19} \times T_{B37h}^2 \\
& + a_{20}(T_{B37h} + SST) + a_{21}(T_{B37h} + V) + a_{22} \times T_{B37h} \\
& + a_{23} \times SST^2 + a_{24}(SST + V) + a_{25} * SST \\
& + a_{26} \times V^2 + a_{27} \times V + a_{28}
\end{aligned} \qquad (3\text{-}2)$$

Table 3-2: Water Vapor Regression Coefficient values

Coeff.	Value
a_1	6.000E-02
a_2	-8.076E-02
a_3	-1.959E-01
a_4	7.492E-02
a_5	4.821E-02
a_6	-5.125E-03
a_7	1.836E+01
a_8	3.133E-02
a_9	1.355E-01
a_10	-6.038E-02
a_11	-3.793E-02
a_12	-5.688E-03
a_13	-9.406E+00
a_14	1.029E-01
a_15	-1.199E-01
a_16	-4.299E-02
a_17	2.400E-02
a_18	-1.233E+01
a_19	3.337E-02
a_20	3.523E-02
a_21	3.075E-03
a_22	7.659E+00
a_23	1.346E-02
a_24	4.770E-03
a_25	-3.931E+00
a_26	-1.429E-03
a_27	-2.870E+00
a_28	-1.135E-01

Figure 3-4 shows a scatter plot of the calculated wind speed using the regression equation, for the withheld Tb's, are compared with the corresponding EDR wind speed values. Again the regression equation for wind speed reproduces the WindSat EDR wind speed well.

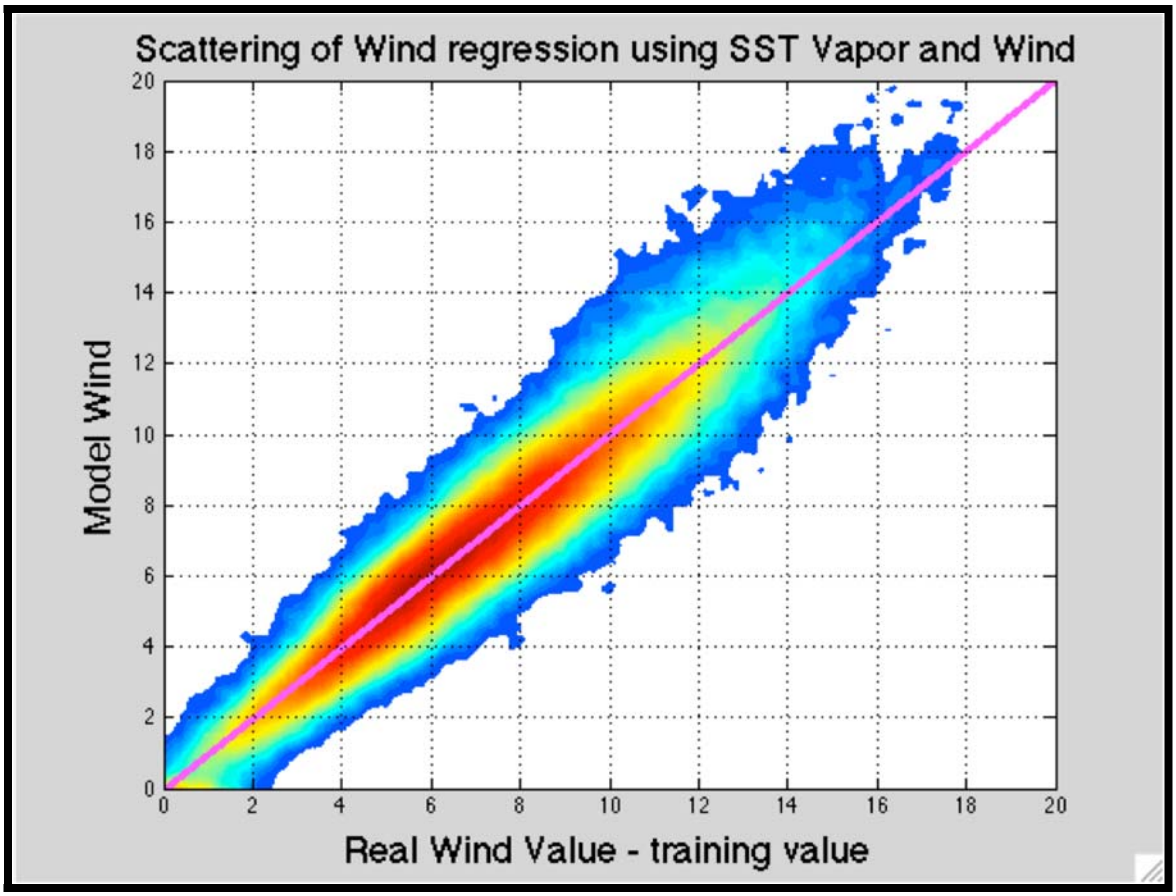


Figure 3-4: Wind speed values comparison

3.6.3. Three non linear equations – Radiative transfer model forward equations

Wentz et al. [2] defines the relationship between ocean brightness temperature measured by the radiometer through a single layer atmosphere can be approximated by equation 9 given as:

$$\boxed{F_{freq}(W,V,\tau_L) = T_{BU} + \tau_{atmos} \times [\varepsilon \times SST + (1-\varepsilon) \times (T_{BD} + \tau_{atmos} \times T_{ex})]} \qquad (3\text{-}3)$$

Also in Wentz' papers [2, 3], he provides parametric equations that relate atmospheric transmissivity and effective air temperature to the SST, and atmospheric integrated water vapor, cloud liquid water, and rain rate for different radiometer channel frequencies, including the MWR channels (23.8GHz and 36.5GHz).

The atmospheric transmissivity or atmospheric optical depth is defined as the product of the transmissivities due to water vapor, oxygen, and liquid water (rain and cloud liquid). Using Wentz statistical correlations [3], the transmissivity due to water vapor and oxygen can be obtained by only knowing the integrated water vapor in the atmosphere and the SST.

The equations for cloud liquid water transmissivity, and transmissivity due to rain are very well known for low rain rate values (less than 2mm) [1] [2]. These relations are shown in equations 3-4 and 3-5.

$$\alpha_L = a_{L1}[1 - a_{L2}(T_L - 283)]L \quad (3\text{-}4)$$

α_L is the attenuation due to cloud liquid water, L is the amount of cloud liquid water and T_L is the temperature of liquid water.

$$\alpha_R = a_{L3}[1 - a_{L4}(T_L - 283)]H \times R^{a_{L5}} \quad (3\text{-}5)$$

α_R is the attenuation due to rain when rain rate is less than 2mm, and R represents rain rate.

When rain rate increases, equations 3.4 and 3.5 do not apply and the relationship is more complicated. In order to simplify the retrieval, the transmissivity due to cloud liquid water and rain were combined into one unknown variable [3]. The transmissivity due to liquid water values at 24GHz and 37GHz was found using Wentz non linear equations shown below:

$$F_{24h} = A_1 + B_1 \times \tau_{L24h} + C_1 \times \tau_{L24h}^2 \quad (3\text{-}6)$$

$$\boxed{F_{37v} = A_2 + B_2 \times \tau_{L37v} + C_2 \times \tau^2_{L37v}} \tag{3-7}$$

$$\boxed{F_{37h} = A_3 + B_3 \times \tau_{L37h} + C_3 \times \tau^2_{L37h}} \tag{3-8}$$

In both of his papers, Wentz set limited values to the sea surface temperature typical for water vapor (T_V), the relationship between sea surface temperature and T_V, and to the height of the columnar rain. He finds that by setting these limitations, he is able to obtain more accurate results and a better relationship between the brightness temperatures measured by the radiometers and calculated brightness temperature (one to one linear relationship)[3].

To obtain faster results for use in real time data calculations, these limitations were translated into 13 different cases. The criteria for these cases include the following:

- Case 1:
  - Water vapor greater than 48mm,
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) greater than 20,

- the difference between sea surface temperature and the sea surface temperature typical for water vapor provides a negative value, and
- sea surface temperature is less than 301K.

- Case 2:
  - Water vapor greater than 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) greater than 20,
  - the difference between sea surface temperature and the sea surface temperature typical for water vapor provides a positive value;
  - sea surface temperature is less than 301K.

- Case 3:
  - Water vapor greater than 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) less than 20;
  - sea surface temperature is less than 301K.

- Case 4:
  - Water vapor less or equal to 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) greater than 20,

- the difference between sea surface temperature and the sea surface temperature typical for water vapor provides a negative value;
- sea surface temperature is less than 301K.

- Case 5:
  - Water vapor less or equal to 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) greater than 20,
  - the difference between sea surface temperature and the sea surface temperature typical for water vapor provides a positive value;
  - sea surface temperature is less than 301K.

- Case 6:
  - Water vapor less or equal to 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) less than 20;
  - sea surface temperature is less than 301K.

- Case 7:
  - Water vapor greater than 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) greater than 20,

- the difference between sea surface temperature and the sea surface temperature typical for water vapor provides a negative value;
- sea surface temperature is greater than 301K.

- Case 8:
  - Water vapor greater than 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) greater than 20,
  - the difference between sea surface temperature and the sea surface temperature typical for water vapor provides a positive value;
  - sea surface temperature is greater than 301K.

- Case 9:
  - Water vapor greater than 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) less than 20;
  - sea surface temperature is greater than 301K.

- Case 10:
  - Water vapor less or equal to 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) greater than 20

- the difference between sea surface temperature and the sea surface temperature typical for water vapor provides a negative value;
- sea surface temperature is greater than 301K.

- Case 11:
  - Water vapor less or equal to 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) greater than 20
  - the difference between sea surface temperature and the sea surface temperature typical for water vapor provides a positive value;
  - sea surface temperature is greater than 301K.

- Case 12:
  - Water vapor less or equal to 48mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) less than 20;
  - sea surface temperature is greater than 301K.

- Case 13:
  - Water vapor greater than 58mm;
  - absolute value for the difference between sea surface temperature and the sea surface temperature typical for water vapor (T_V) less than 20;

- sea surface temperature is greater than 301K.

By introducing these cases, for loops for pixel-by-pixel operations were eliminated from the coding program, and vector operation was introduced. The time frame was changed from 2 hours per every satellite scan (using the for loop) to less than a minute per revolution (using the different cases).

The combined solution for all these cases provides the calculated brightness temperatures using the radiative transfer model theory.

3.6.4. Tuning of calculated brightness temperatures to WindSat brightness temperature measurements

Water vapor and wind speed values were calculated using the statistical regression equations (equation 3-1 and equation 3-2). Liquid water transmissivity equation is well known when cloud liquid water values are low, and rain rate is less than 2mm/h. Equations used to calculate liquid water transmissivity are equation 3-4 and equation 3-5. These four retrieved parameters (V, W, τ_{L24}, τ_{L37},) were input into the three non linear equations (eq. 3-6, eq. 3-7, and eq. 3-8) to calculate the three WindSat brightness temperatures (Tb at 24GHz h-pol, Tb at 37GHz h and v pol).

These calculated brightness temperatures were then compared to the real brightness temperatures measured by the WindSat radiometer. The comparisons for these brightness temperatures are shown in Figure 3-5, figure 3-6, figure 3-7.

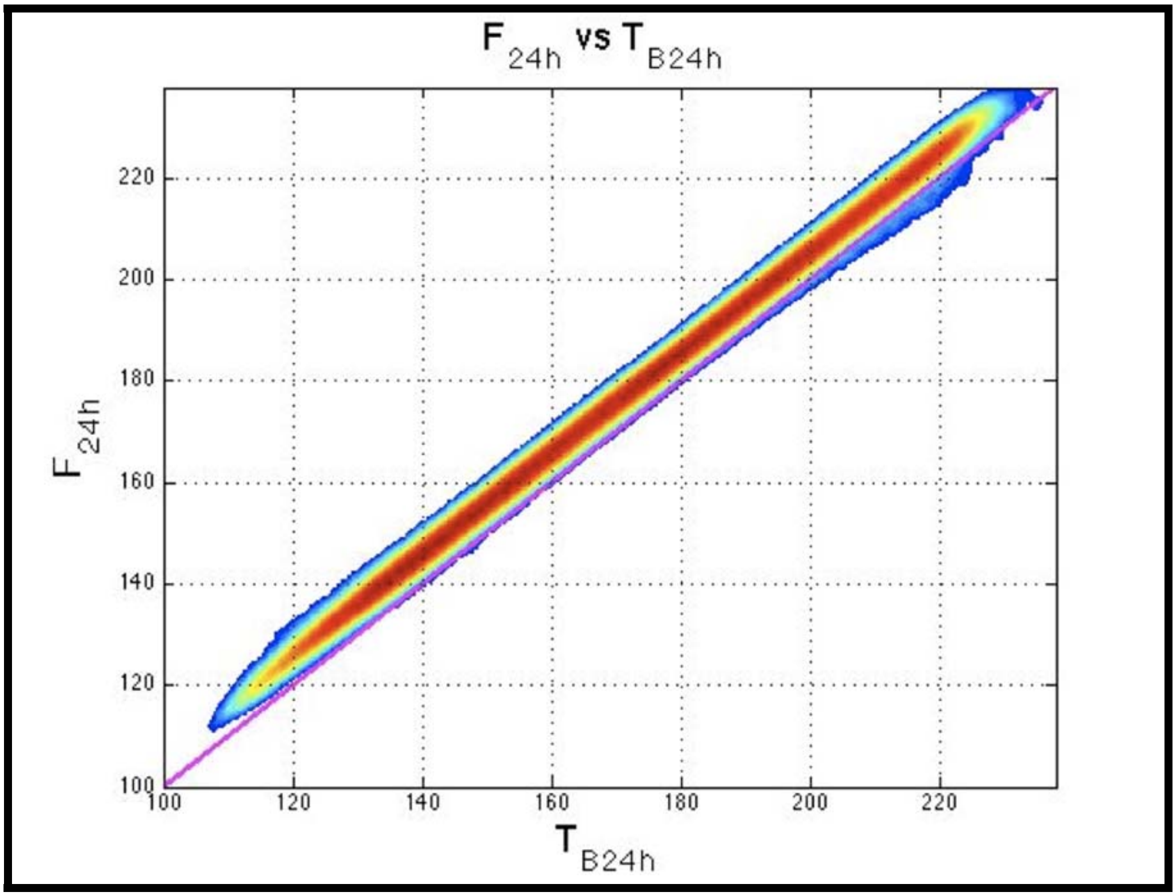


Figure 3-5: Brightness Temperature comparison at 24GHz h-pol

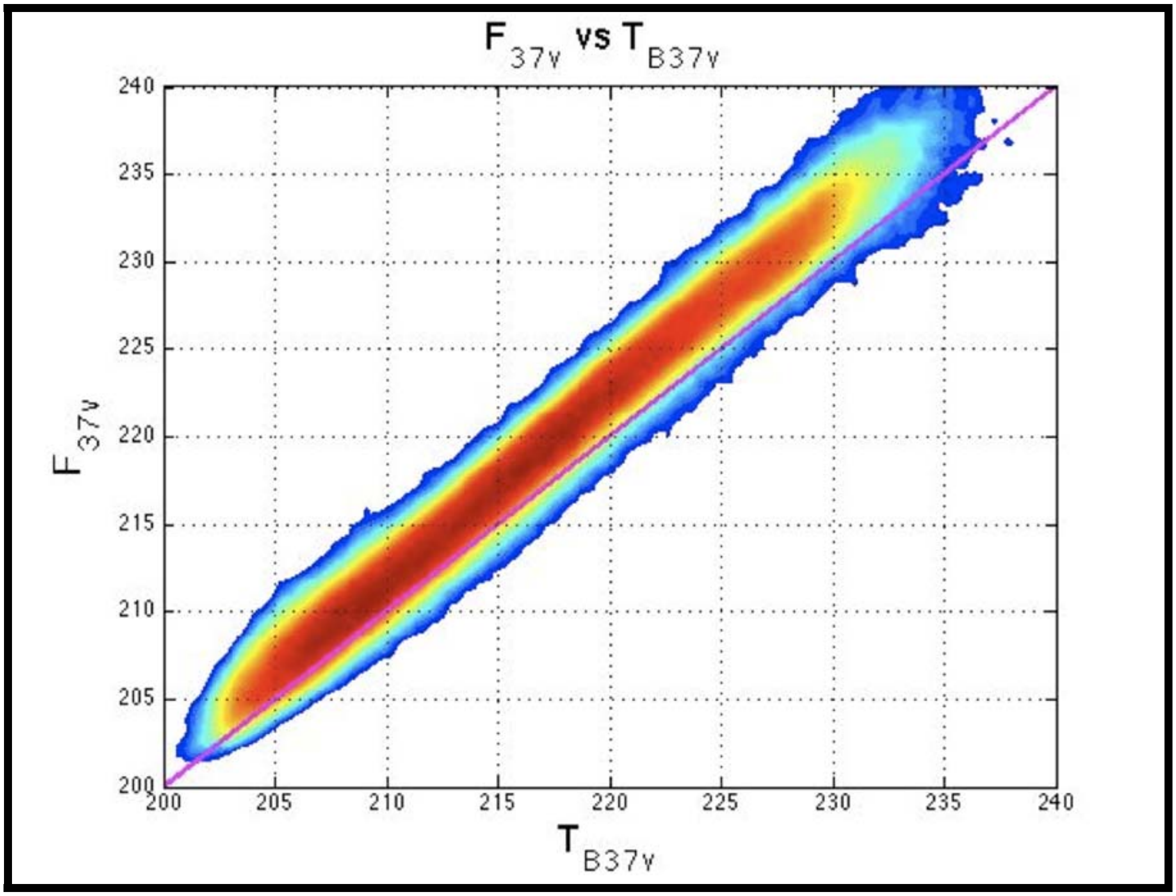


Figure 3-6: Brightness Temperature comparison at 37GHz v-pol

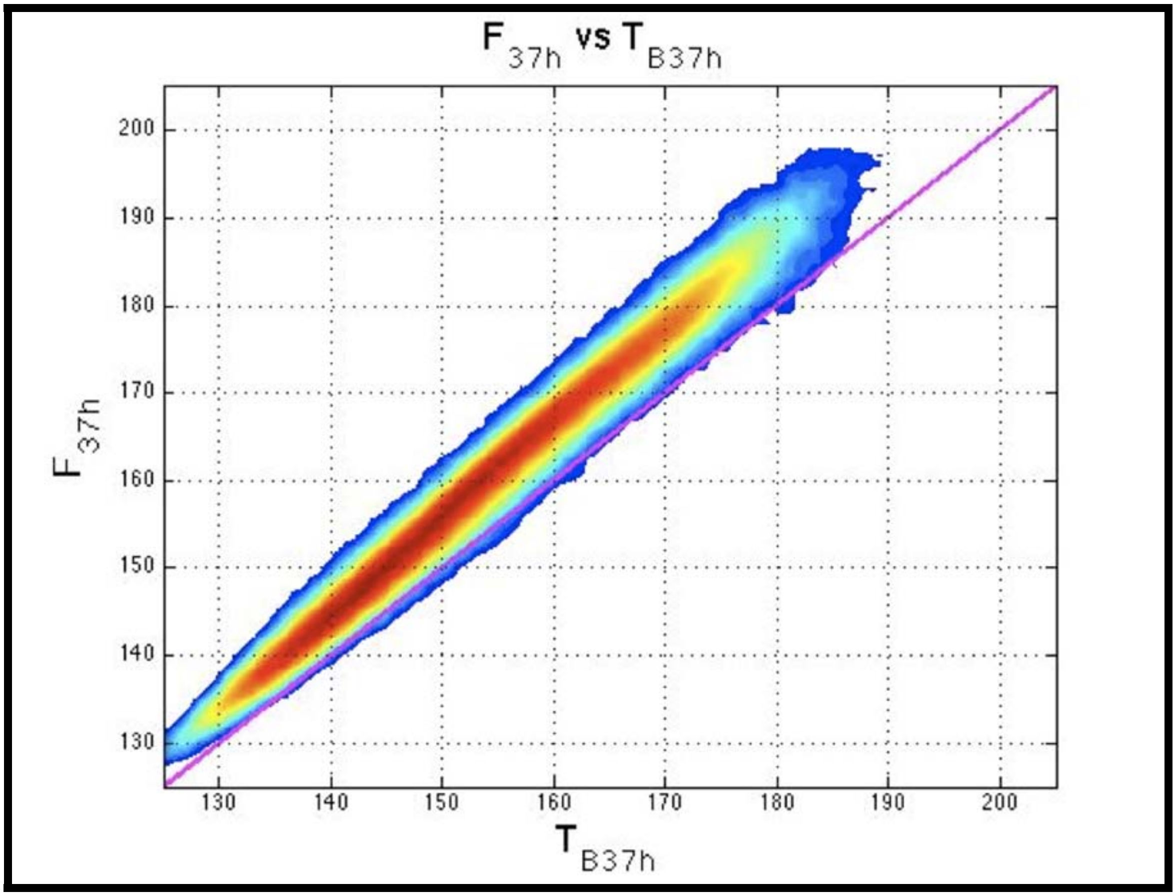


Figure 3-7: Brightness Temperature comparison at 37GHz h-pol

Because the three non linear equation coefficients were defined and calculated by Wentz for AMSR, the calculated brightness temperature contained a bias. This bias can be observed on shown the three previous figures. In order to tune these calculated brightness temperatures, delta values were introduced to the forward model equations. These delta values are shown in table 3-3. Only 4 of the 13 cases had significant numbers of observations to perform the bias analysis.

Table 3-3: Delta Values

Case 3	$\Delta_{24h} = (1-0.86188)X - 29.706$ $\Delta_{37v} = (1-0.72493)X - 59.993$ $\Delta_{37h} = (1-0.80094)X - 27.508$
Case 6	$\Delta_{24h} = (1-1.0006)X + 5.7077$ $\Delta_{37v} = (1-0.9003)X - 19.483$ $\Delta_{37h} = (1-0.88289)X - 12.389$
Case 9	$\Delta_{24h} = (1-0.88467)X - 20.451$ $\Delta_{37v} = (1-0.72396)X - 60.733$ $\Delta_{37h} = (1-0.79)X - 30.135$
Case 12	$\Delta_{24h} = (1-0.99523)X + 4.4295$ $\Delta_{37v} = (1-0.83687)X - 33.695$ $\Delta_{37h} = (1-0.86035)X - 15.912$

After the delta values were added to the equations and the program was modified, the brightness temperatures (Tb at 24GHz h-pol and Tb 37GHz v and h-pol) were compared again. These new comparisons are shown in figure 3-8, figure 3-9 and figure 3-10. It is shown on these figures that the brightness temperatures have been calibrated when cloud liquid water contains low values and rain is not present.

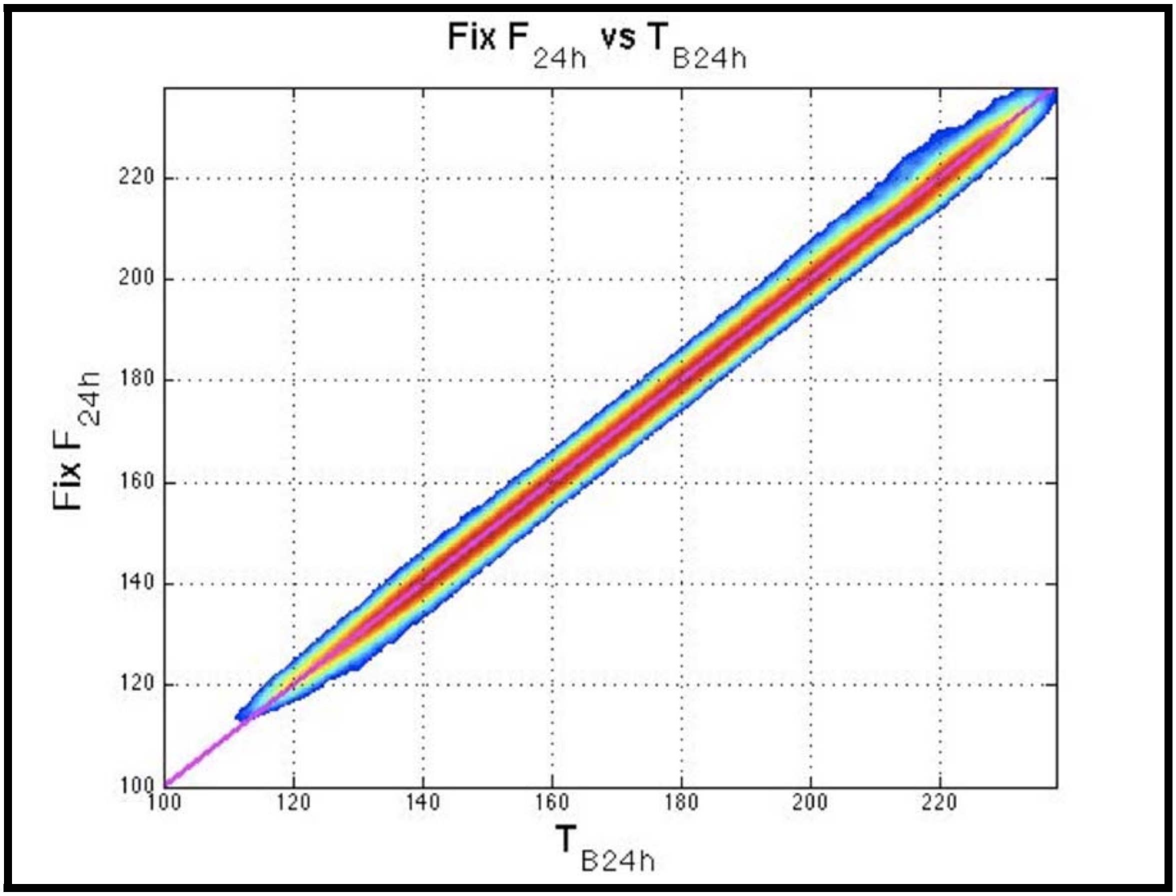


Figure 3-8: Brightness Temperature comparison at 24GHz h-pol

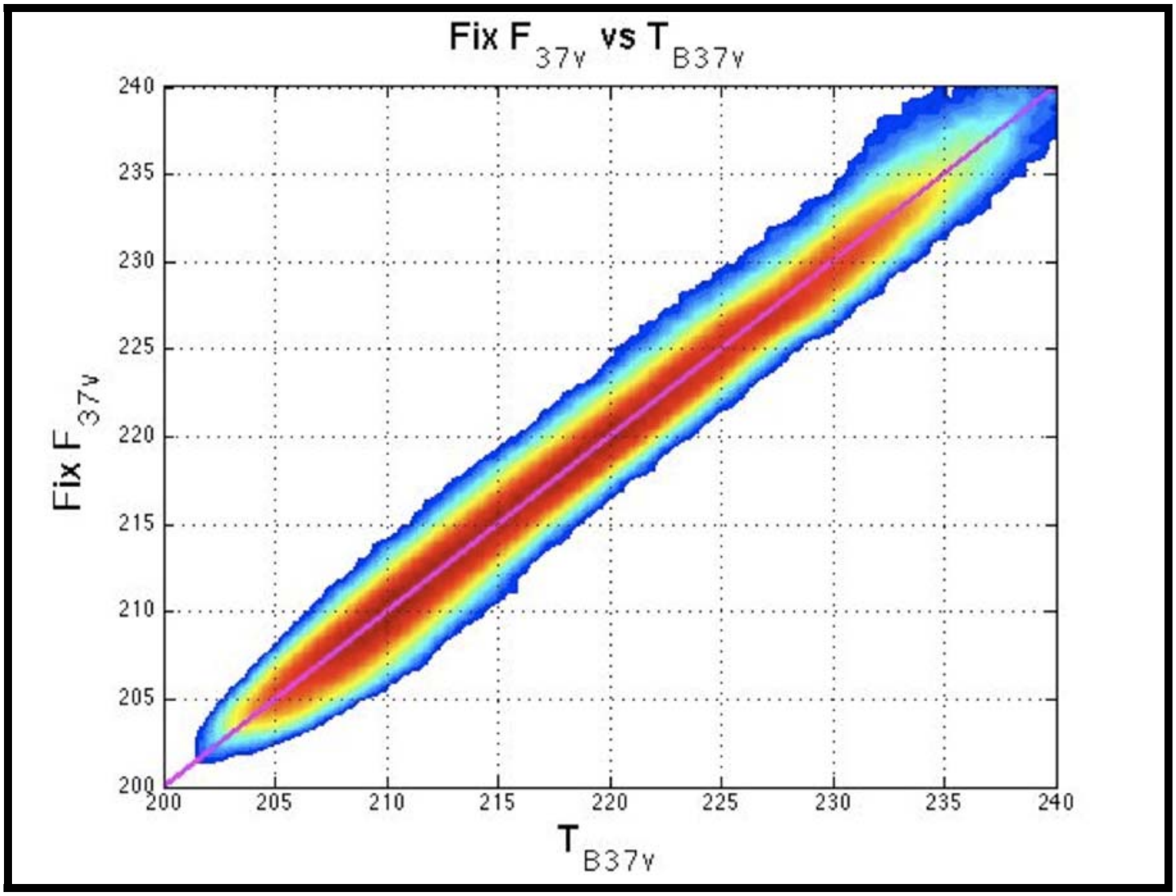


Figure 3-9: Brightness Temperature comparison at 37GHz v-pol

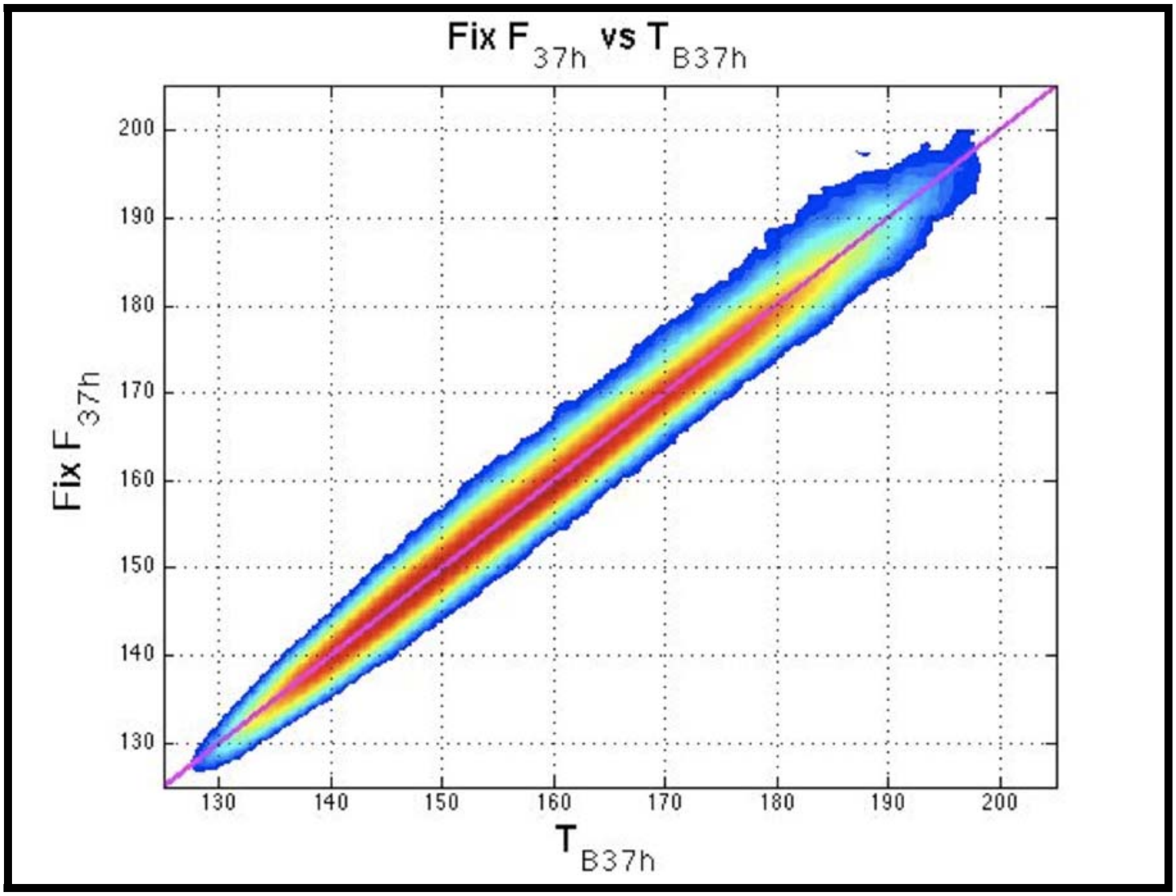


Figure 3-10: Brightness Temperature comparison at 37GHz h-pol

3.6.5. Transmissivity due to liquid water retrieval and validation

When brightness temperature, sea surface temperature, water vapor, emissivity, and wind speed are incorporated into equations 3-6, 3-7, 3-8, the solution for the transmissivities de to liquid water is a quadratic equation. These equations are shown in equations 3-9, 3-10, and 3-11.

$$\tau_{L24h} = \frac{-B_2 \pm \sqrt{B_2{}^2 - 4 \times C_2 \times (A_2 - T_{B24h})}}{2 \times C_2}$$
$$A_1 = A_2 = T_{U24}$$
$$A_3 = A_4 = T_{U37} \qquad (3\text{-}9)$$

$$\tau_{L37v} = \frac{-B_3 \pm \sqrt{B_3{}^2 - 4 \times C_3 \times (A_3 - T_{B37v})}}{2 \times C_3}$$
$$A_1 = A_2 = T_{U24}$$
$$A_3 = A_4 = T_{U37} \qquad (3\text{-}10)$$

$$\tau_{L37h} = \frac{-B_4 \pm \sqrt{B_4{}^2 - 4 \times C_4 \times (A_4 - T_{B37h})}}{2 \times C_4}$$
$$A_1 = A_2 = T_{U24}$$
$$A_3 = A_4 = T_{U37} \qquad (3\text{-}11)$$

Because the data used for the water vapor and wind speed retrievals contained cloud liquid water levels up 0.5mm and rain was not present, the transmissivity due to liquid water was calculated. The equations used to solve for these transmissivity values are previously shown in equation 3-4.

To check the solutions from equation 3-9, equation 3-10, and equation 3-11, a scatter plot was generated. The real transmissivity values due to liquid water, calculated using equation 3-4, were compared against the retrieved transmissivity values due to liquid water obtained from those three previous equations.

These comparisons are shown in figure 3-11 and figure 3-12. It is clear in these two figures that the obtained transmissivities using equations 3-9 through 3-11 provide a very good approximation for the real transmissivity values.

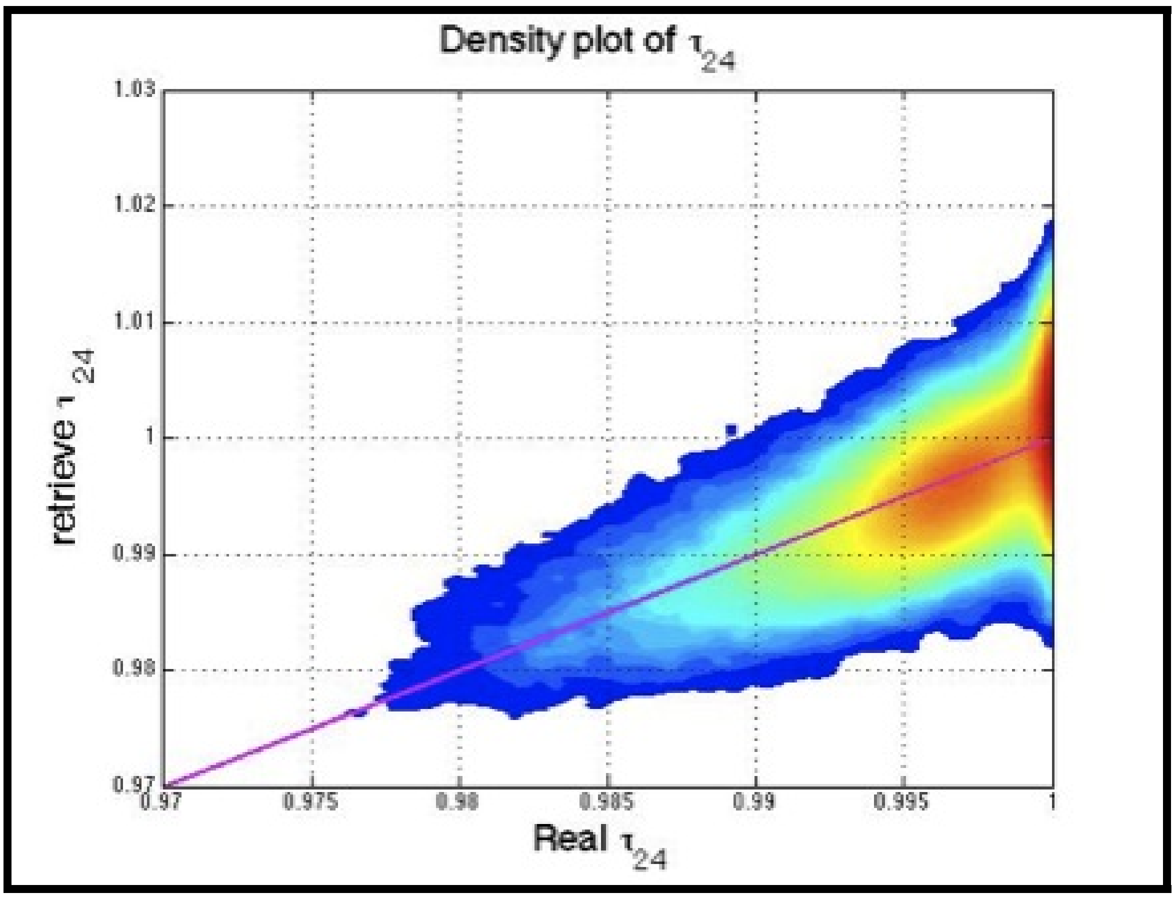


Figure 3-11: Scatter Density Plot of transmissivity due to liquid water at 24GHz

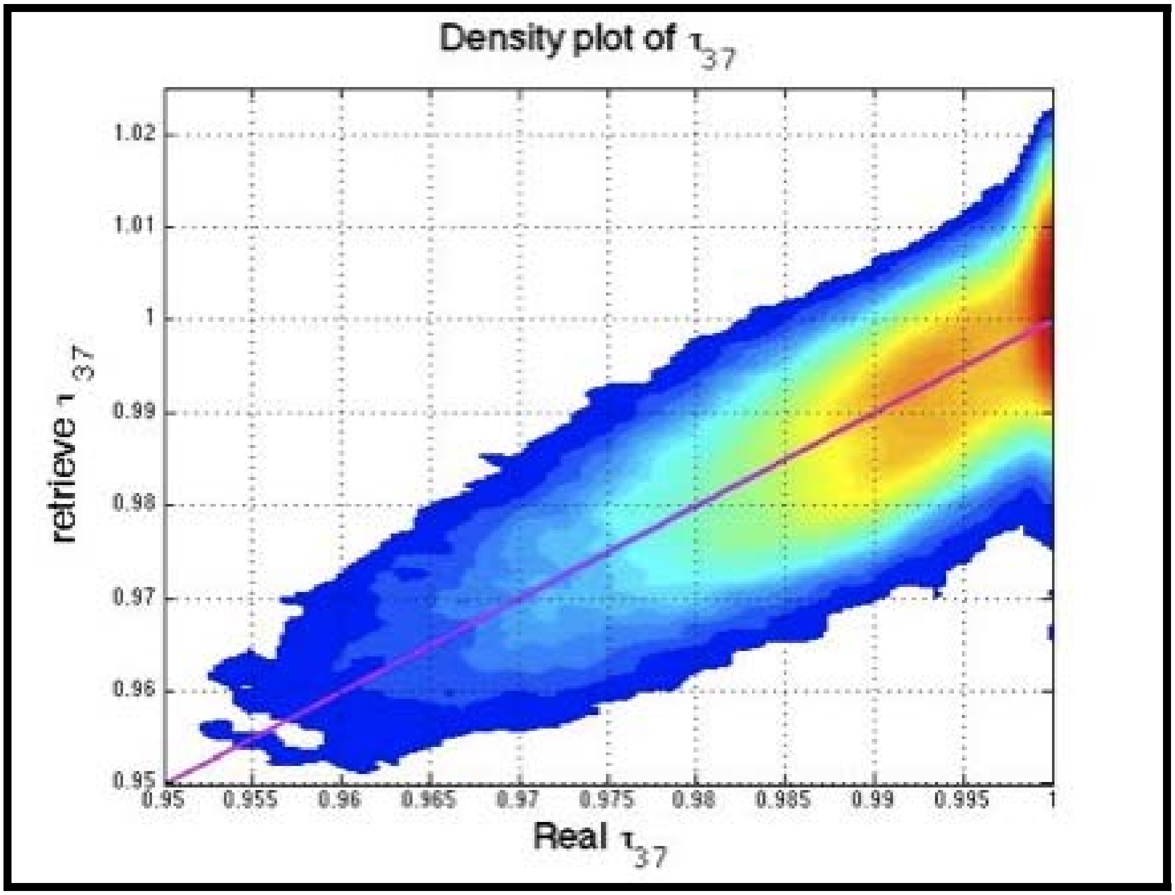


Figure 3-12: Scatter Density Plot of transmissivity due to liquid water at 37GHz

3.6.6. Rain Rate Retrieval

To calculate rain rate, a statistical regression was executed. This statistical regression had five dependent variables, and one independent variable. The dependent variables were: transmissivity due to liquid water at 24GHz h pol, transmissivity due to liquid water at 37GHz v pol, transmissivity due to liquid water at 37 GHz h-pol, height of the rain (sea surface temperature dependent), and temperature of the rain (sea surface temperature dependent and very well correlated to height of the rain). The independent variable for this regression was the EDR Rain Rate collected from WindSat data.

The data collected for this retrieval contained the IDR and EDR values for WindSat one day every other week for an entire year. To eliminate scattering, the temperature of the rain and rain rate data per day was binned and average. Temperature of rain was divided into 15 groups: from 272K to less than 288K on increments of 1K (total of 14 bins), and from 288K and above. For every temperature bin, there were 30 bins for different rain rates and liquid water transmissivity values. Rain rate and liquid water transmissivity was averaged for each rain rate range. This rain rate range bin was distributed as follows: starting from 0 mm/h to 15 mm/h in increments of 1mm/h with an overlap of 0.5mm/h (29 bins), and the last range was from 14.5mm/h and up. The maximum rain rate in this data was 16 mm/h.

Per day, there was around 100,000 points for rain rate greater than 0.1mm without averaging the data. When the daily data was averaged, this number went down to 450 points. The total number of averaged points for the regression was 11,700 data points.

When the regression was done, different orders of regressions were calculated and compared to the EDR rain rate. The regression that provided the most accurate values was the 3rd order statistical regression. Equation 3-12 shows the polynomial for this 3rd

degree statistical regression, and table 3-4 shows the coefficient values for this polynomial.

$$
\begin{aligned}
& a_1 \times T_{rain}^3 + a_2(T_{rain}^2 + \tau_{L24h}) + a_3(T_{rain}^2 + \tau_{L37v}) \\
& + a_4(T_{rain}^2 + \tau_{L37h}) + a_5 \times T_{rain}^2 + a_6(T_{rain} + \tau_{L24h}^2) \\
& + a_7(T_{rain} + \tau_{L24h} + \tau_{L37v}) + a_8(T_{rain} + \tau_{L24h} + \tau_{L37h}) \\
& + a_9(T_{rain} + \tau_{L24h}) + a_{10}(T_{rain} + \tau_{L37v}^2) \\
& + a_{11}(T_{rain} + \tau_{L37v} + \tau_{L37h}) + a_{12}(T_{rain} + \tau_{L37v}) \\
& + a_{13}(T_{rain} + \tau_{L37h}^2) + a_{14}(T_{rain} + \tau_{L37h}) + a_{15} \times T_{rain} \\
& + a_{16} \times \tau_{L24h}^3 + a_{17}(\tau_{L24h}^2 + \tau_{L37v}) + a_{18}(\tau_{L24h}^2 + \tau_{L37h}) \\
& + a_{19} \times \tau_{L24h}^2 + a_{20}(\tau_{L24h} + \tau_{L37v}^2) \\
& + a_{21}(\tau_{L24h} + \tau_{L37v} + \tau_{L37h}) + a_{22}(\tau_{L24h} + \tau_{L37v}) \\
& + a_{23}(\tau_{L24h} + \tau_{L37h}^2) + a_{24}(\tau_{L24h} + \tau_{L37h}) \\
& + a_{25} \times \tau_{L24h} + a_{26} \times \tau_{L37v}^3 + a_{27}(\tau_{L37v}^2 + \tau_{L37h}) \\
& + a_{28} \times \tau_{L37v}^2 + a_{29}(\tau_{L37v} + \tau_{L37h}^2) + a_{30}(\tau_{L37v} \\
& + \tau_{L37h}) + a_{31} \times \tau_{L37v} + a_{32} \times \tau_{L37h}^3 + a_{33} \times \tau_{L37h}^2 \\
& + a_{34} \times \tau_{L37h} + a_{35}
\end{aligned}
\qquad (3\text{-}12)
$$

Table 3-4: Rain Rate 3rd order statistical regression polynomial coefficients

a_1	4.0311E-03
a_2	-4.7426E-01
a_3	-7.4355E-01
a_4	4.9928E-01
a_5	-2.7015E+00
a_6	8.4195E+00
a_7	-1.3940E+00
a_8	-1.9472E+01
a_9	2.6534E+02
a_10	2.7010E+00
a_11	3.0738E+01
a_12	3.8837E+02
a_13	-1.8357E+01
a_14	-2.5651E+02
a_15	5.6720E+02
a_16	1.6274E+02
a_17	-1.0576E+02
a_18	-4.8746E+02
a_19	-2.5495E+03
a_20	-7.4173E+01
a_21	9.7442E+02
a_22	-1.3758E+02
a_23	-2.8297E+02
a_24	6.4742E+03
a_25	-3.6979E+04
a_26	1.7320E+00
a_27	4.6081E+01
a_28	-7.4221E+02
a_29	-1.0282E+03
a_30	-8.0027E+03
a_31	-5.0363E+04
a_32	5.6020E+02
a_33	4.8866E+03
a_34	3.1930E+04
a_35	-3.5416E+04

Figure 3-13 shows the density scatter plot where the retrieved rain rate is compared against the EDR rain from WindSat. This figure shows that the regression provides good rain rate estimation.

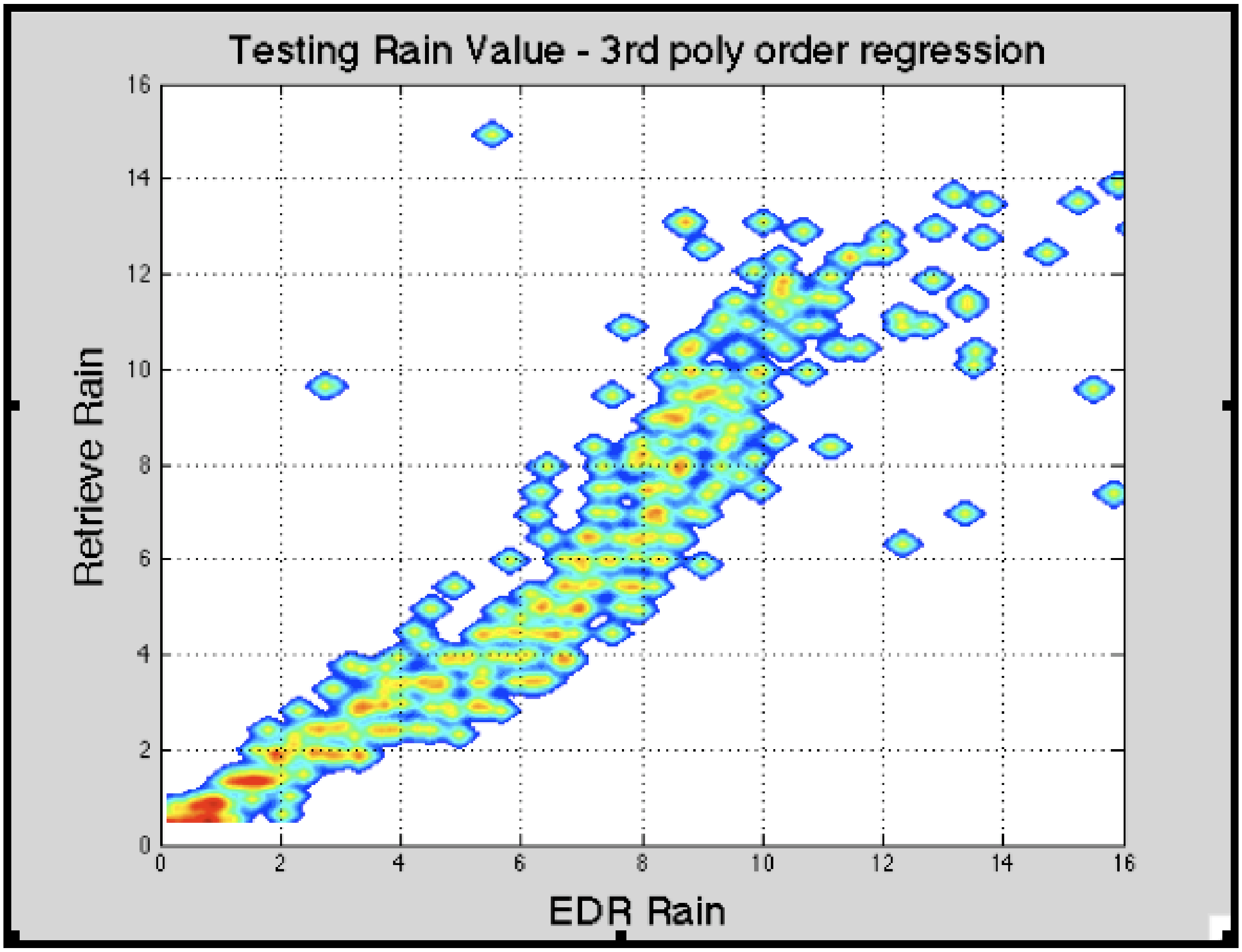


Figure 3-13: Density scatter plot for retrieve rain rate vs. EDR rain rate

CHAPTER 4. RESULTS

This chapter contains the results obtained from the statistical regressions performed on this study: Water vapor, wind speed, and rain rate. Rain rate retrieval will also be compared against WindSat EDR rain rate in order to generate conclusions.

4.1. WindSat Evaluation Data Set

Two different types of WindSat EDR data were used to evaluate the retrievals. When EDR data was collected to perform the statistical regressions for water vapor and wind speed values, cloud liquid water levels were low and rain was absent. As a result, the data was collocated and averaged within 1 degree latitude by 1 degree longitude boxes. Additionally, these data were located in the Pacific Ocean only as shown in Figure 4-1.

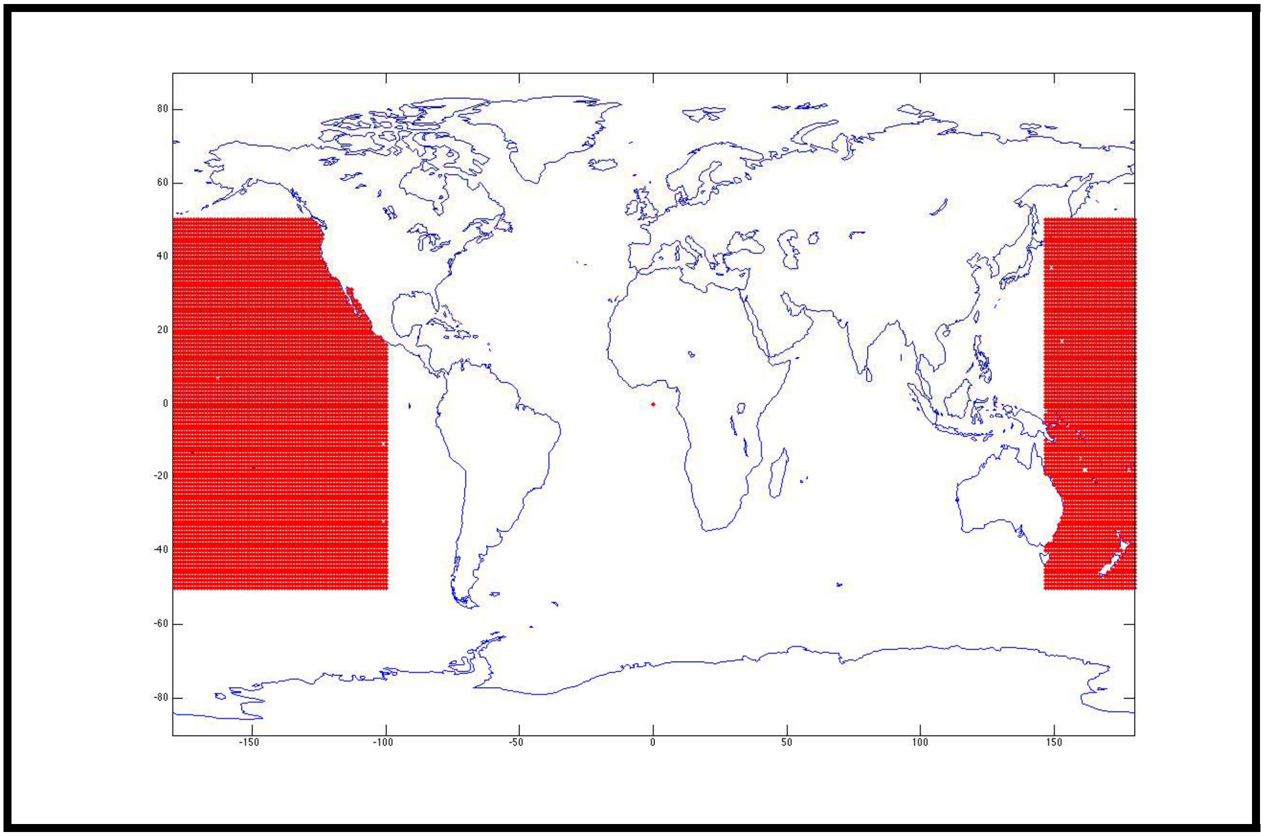


Figure 4-1: Geographical locations for data values used for the Water Vapor and Wind Speed Regression

As previously mentioned, the data for the water vapor and wind speed regressions contained one day every other week for an entire year. These data were saved according to the month it was collected; for example, if there were 3 days of data collected in March, those 3 days were saved under a file indicating the month, March. When all data for each month was located in one file, half of the data every month was extracted to perform the regression. The other half of the data was used to test the vapor and wind regression. Each data set (the set to generate the regression and the set to test the regression) contained 148,254 data points.

Figure 4-2 and figure 4-3 show the comparison of retrieved water vapor and wind speed values against WindSat EDR water vapor and wind speed values. These retrieved values were obtained using the second order polynomial equation given in the previous chapter.

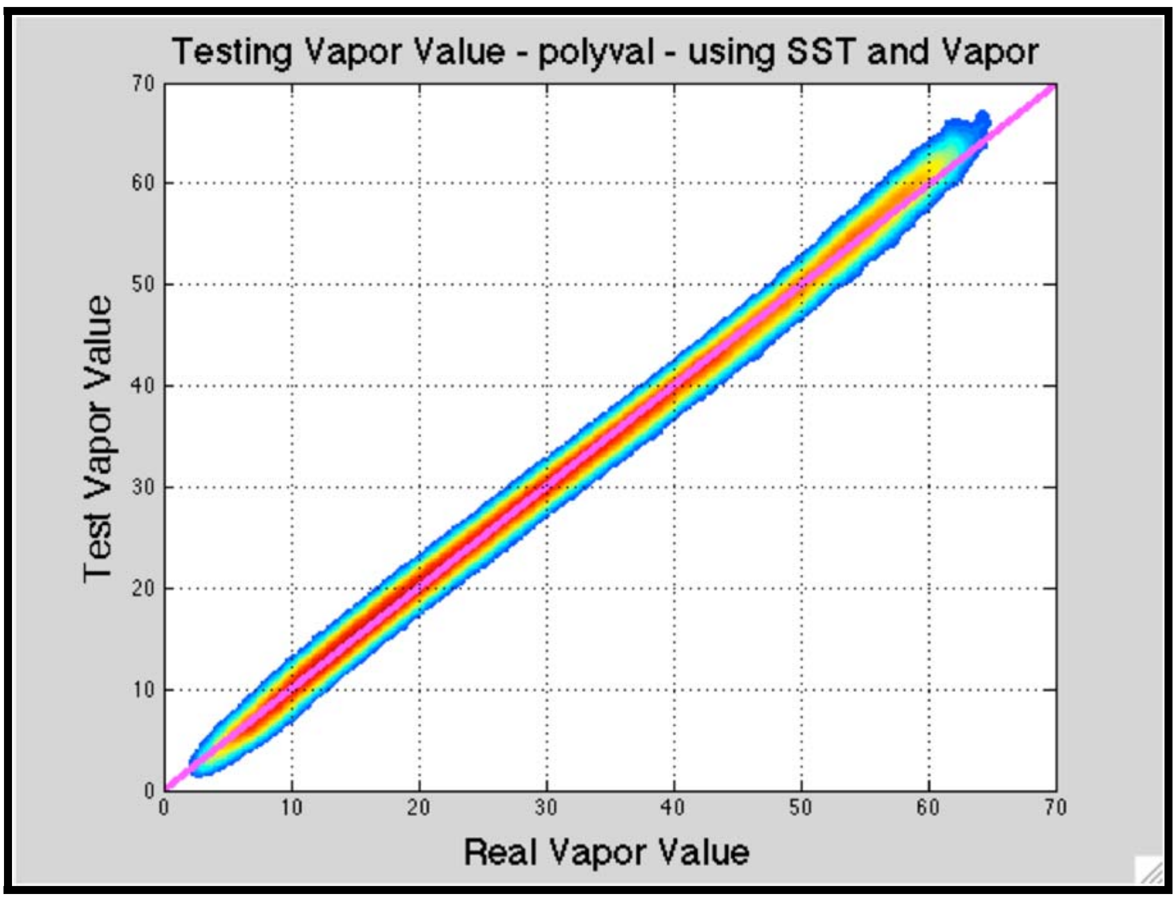


Figure 4-2: Retrieved water vapor vs. WindSat EDR water vapor

Figure 4-2 shows a high correlation between water vapor and brightness temperatures (23.8GHz and 37GHz vertical and horizontal polarization) and sea surface temperature. This high correlation only existed when values of clouds liquid water and rain were low.

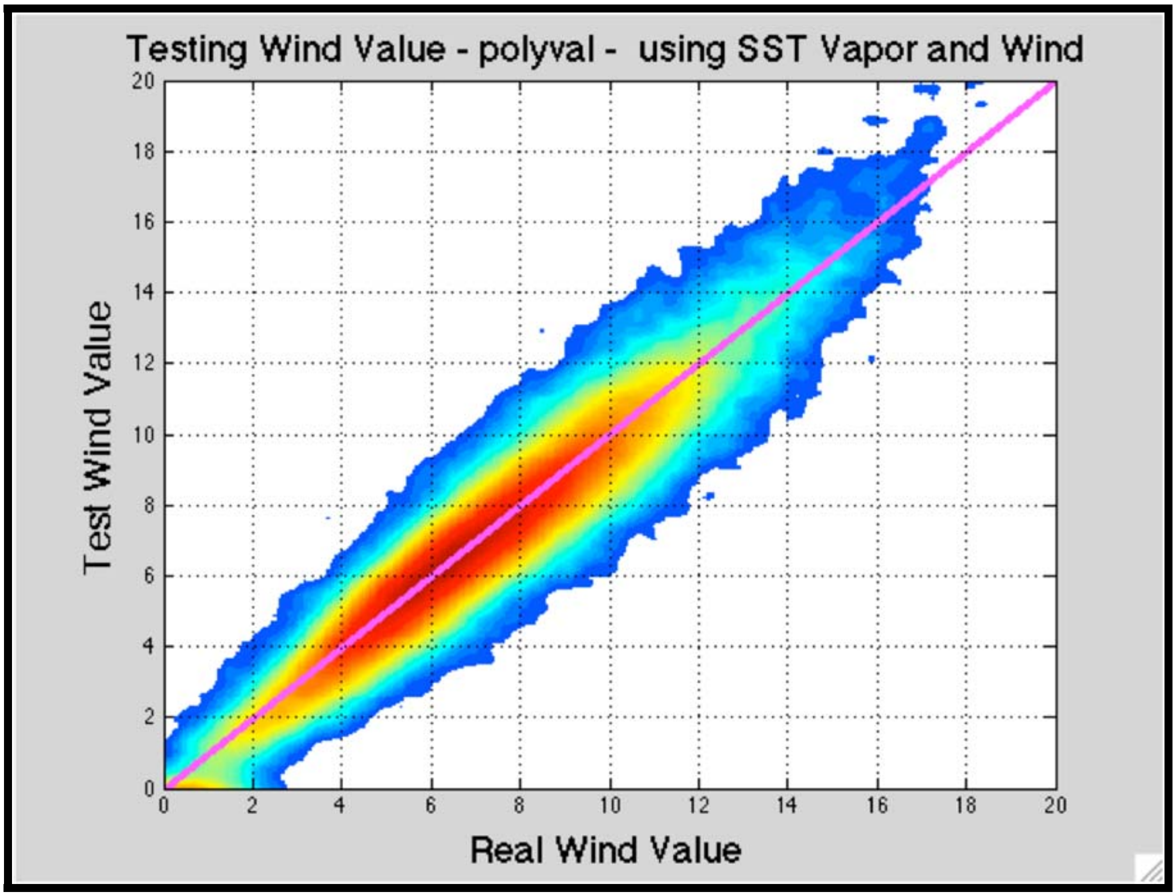


Figure 4-3: Retrieved wind speed vs. WindSat EDR wind speed

Figure 4-3 also shows a good approximation for wind speed retrievals, although not as well as the water vapor regression.

When rain was present, the majority of retrievals for water vapor were still satisfactory even though some negative water vapor values were obtained from the retrieval. On the other hand, the retrieved values for wind speed were not satisfactory as it was expected. Typically negative wind speed values were obtained using the statistical regression polynomial. When the retrieved values for wind speed and water vapor were negative, water vapor was set to a value of 0mm, and wind speed was set to a value of 6.5 m/s.

The retrieved water vapor and WindSat EDR water vapor are imaged in figures 4-4 and figure 4-5, and the results are very similar.

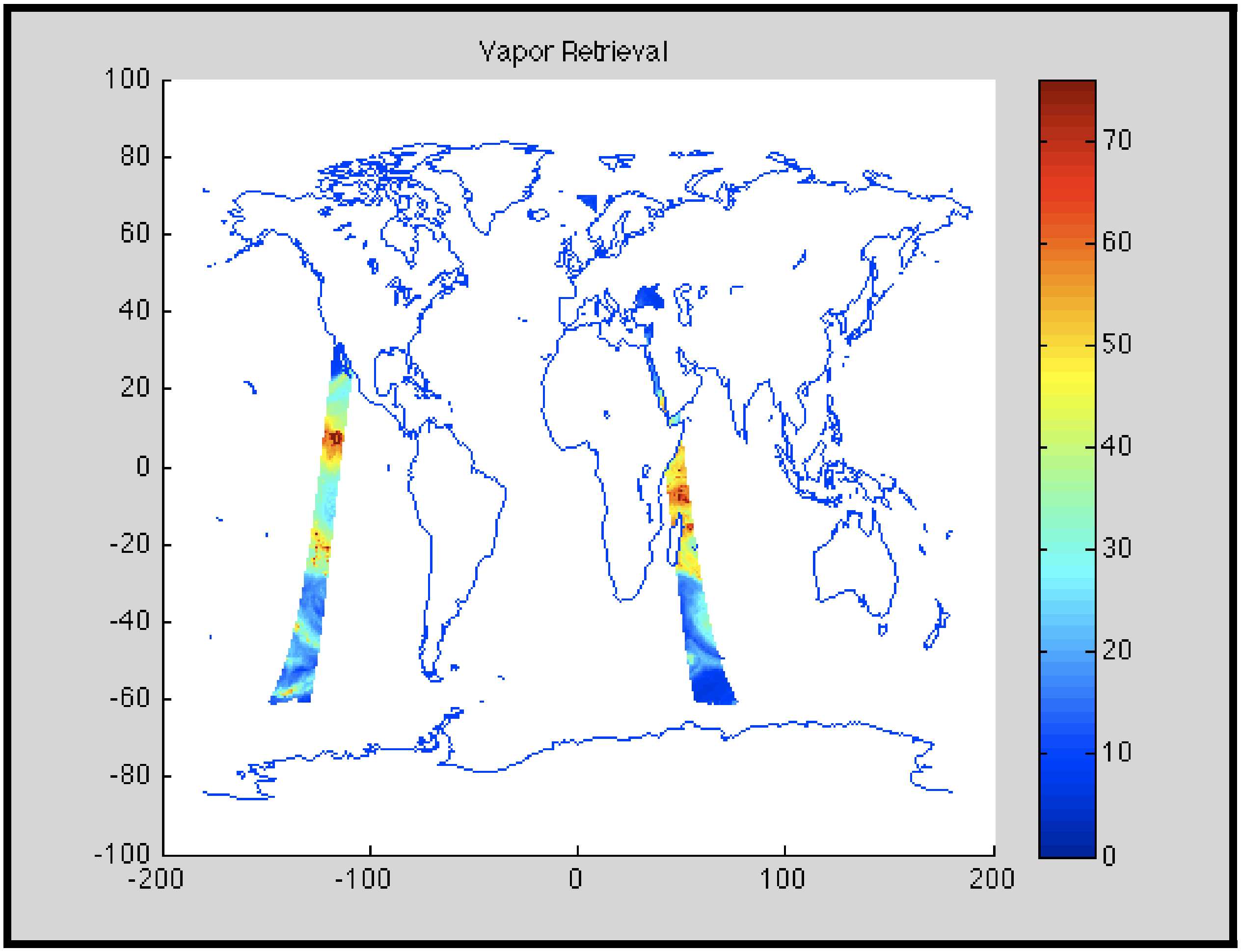


Figure 4-4: Retrieved vapor - One revolution from December 6th 2006

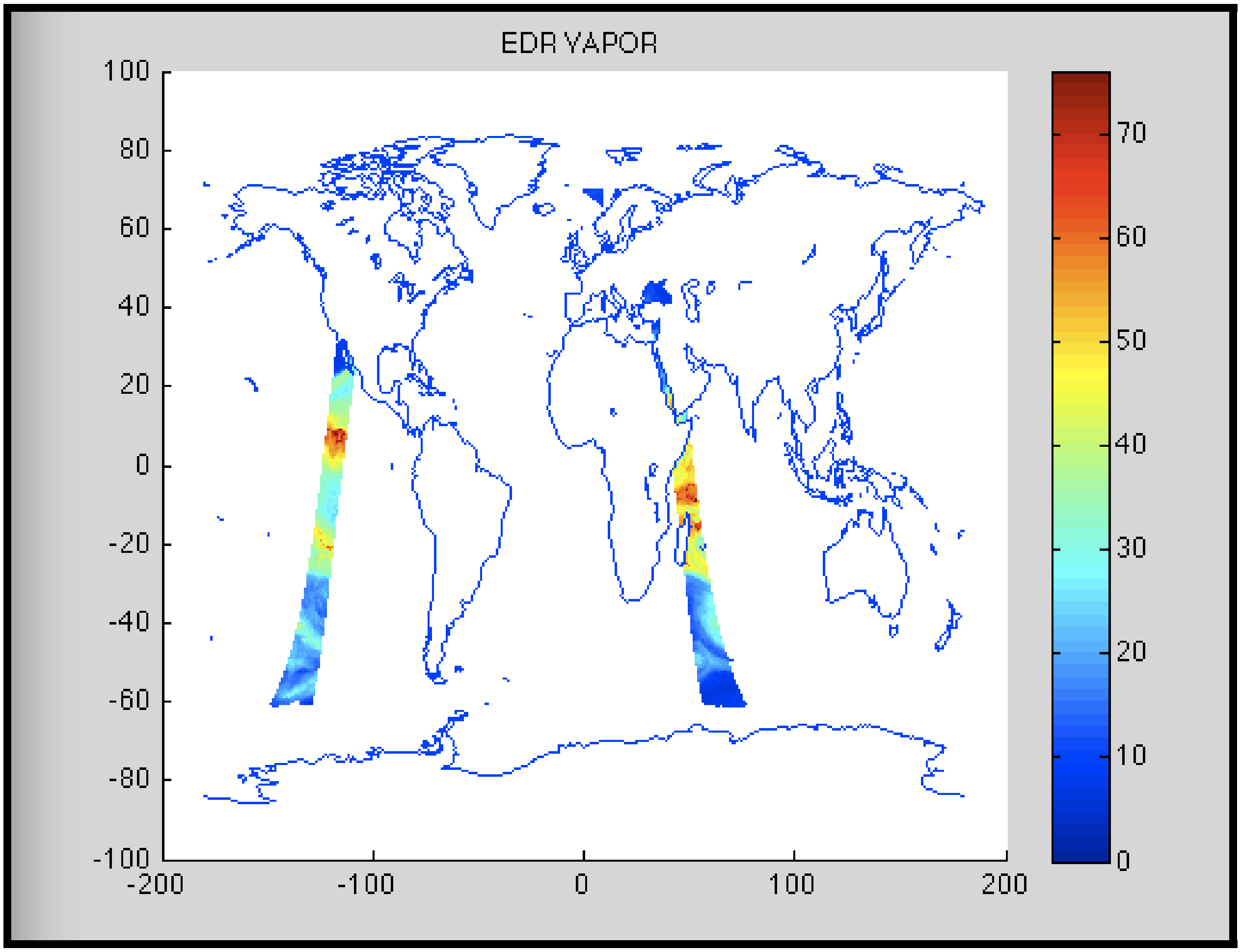


Figure 4-5: EDR vapor - One revolution from December 6th 2006

However a closer look for these water vapor images and EDR rain image is provided in Figure 4-6 through Figure 4-8. It can be seen on these images that when rain is present, the retrieved water vapor provides a higher value compared to the EDR water vapor from WindSat.

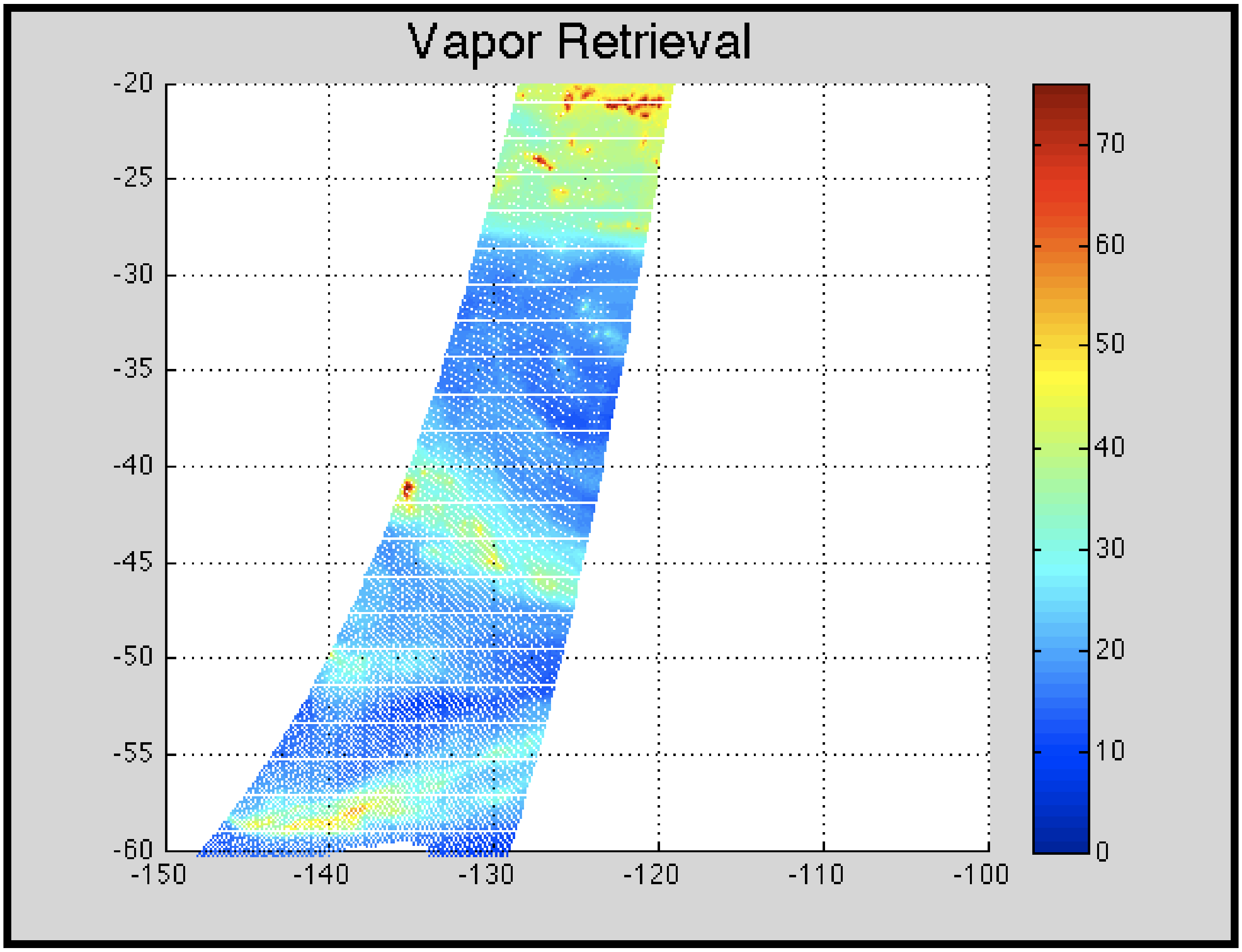


Figure 4-6: Retrieved vapor - One revolution from December 6th 2006

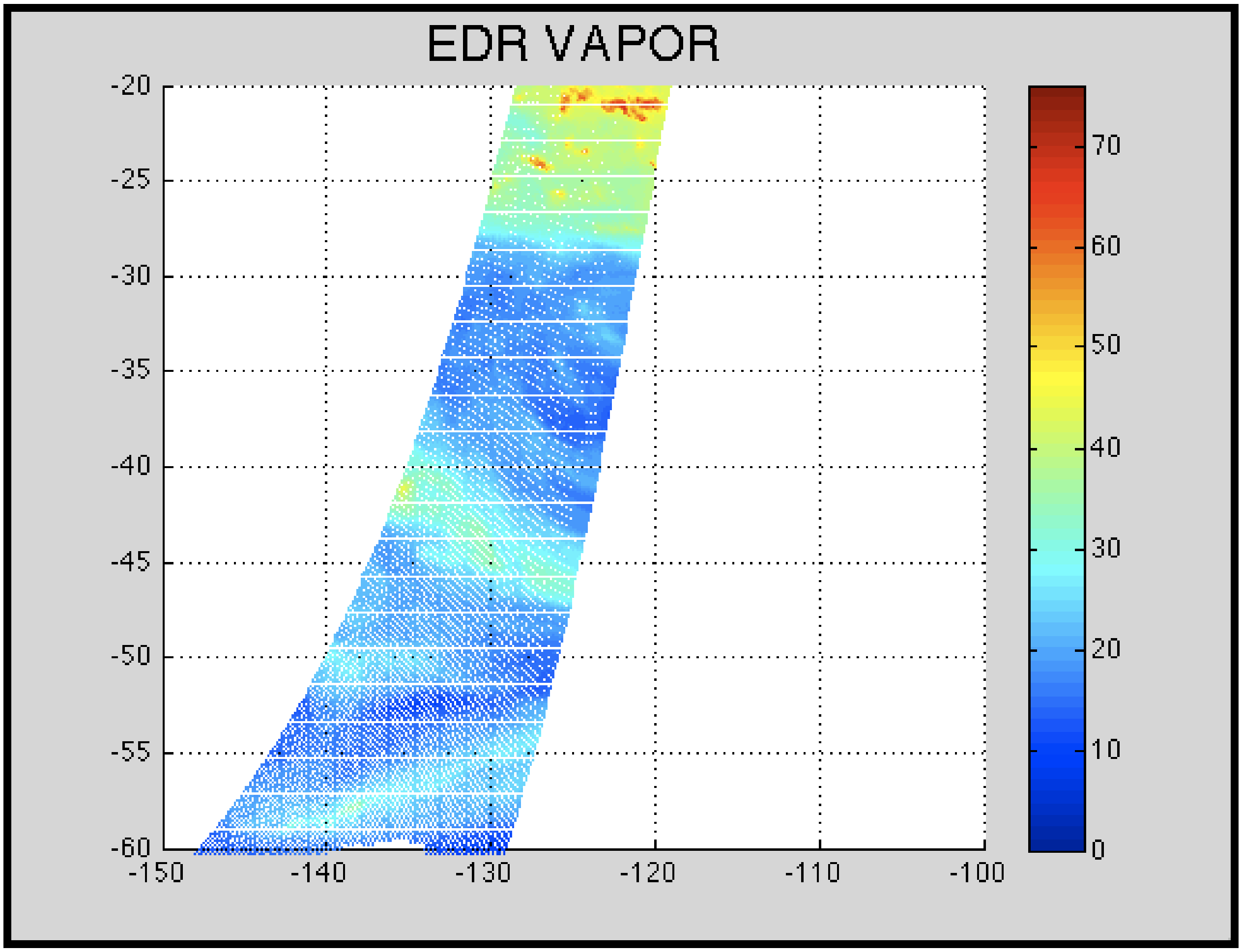


Figure 4-7: EDR vapor - One revolution from December 6th 2006

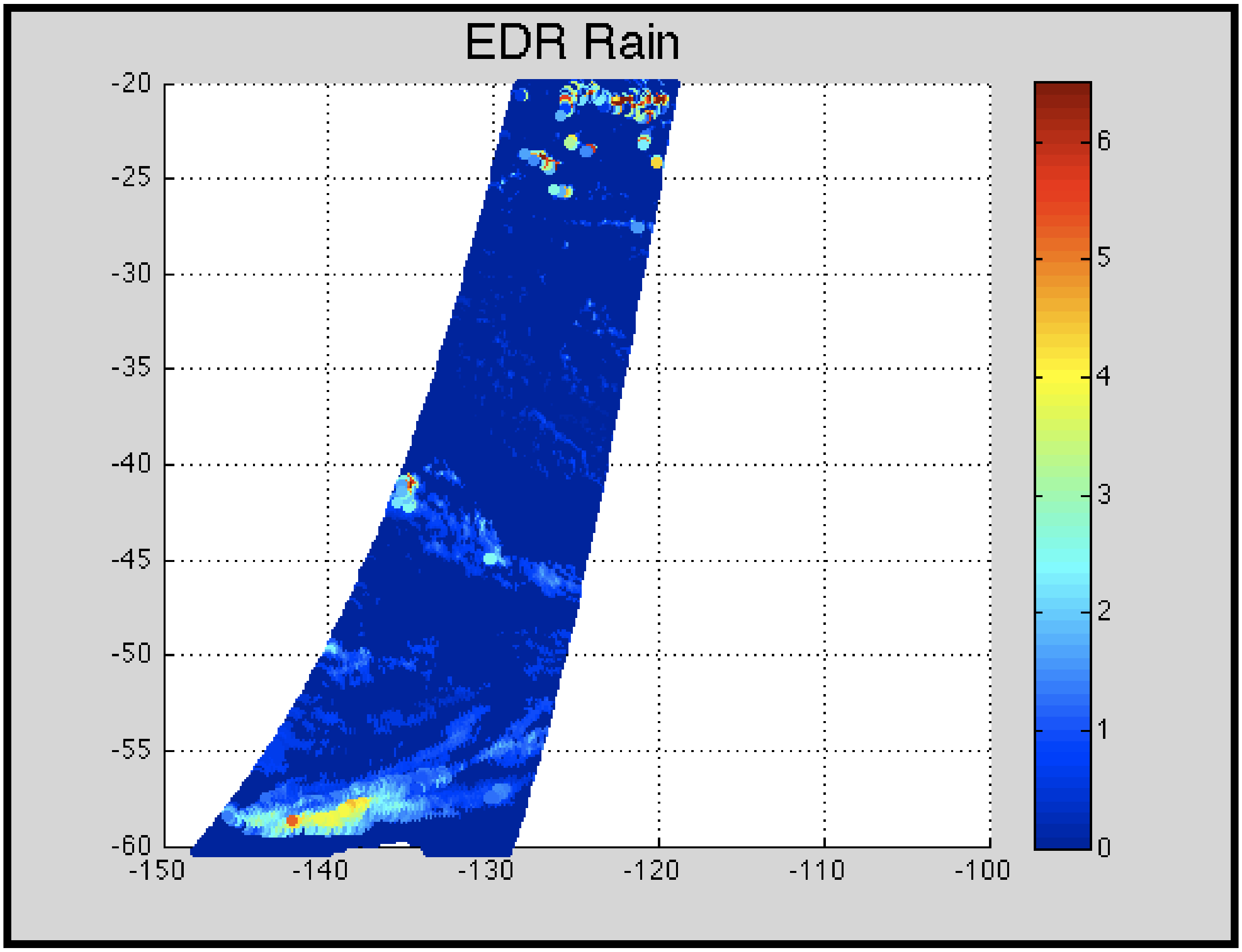


Figure 4-8: EDR rain - One revolution from December 6th 2006

The effects of rain in the wind speed retrieval and the comparison with EDR wind are shown in Figure 4-9 and figure 4-10. The accuracy of the wind speed retrieval values is severely degraded as rain rate increases.

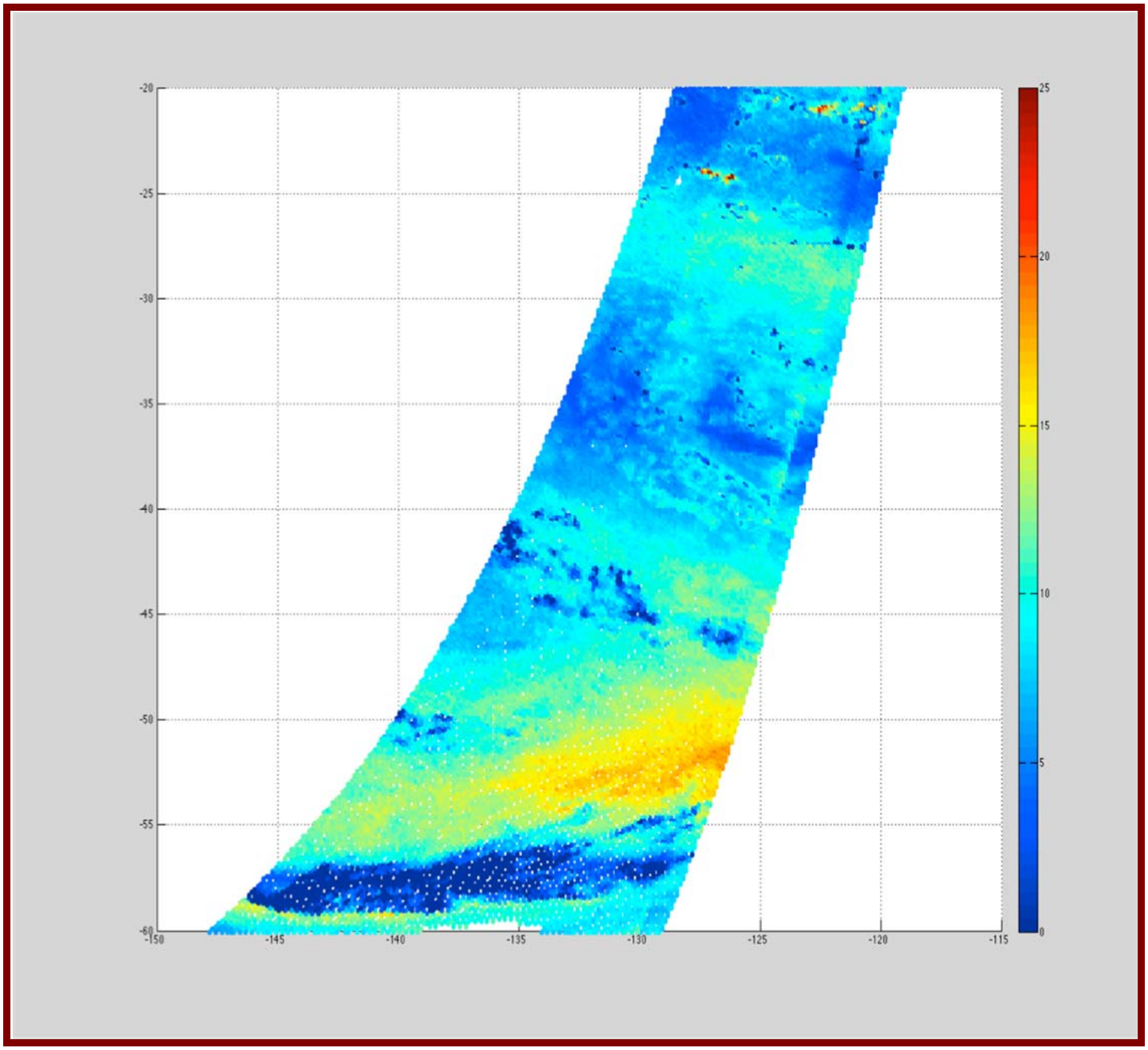


Figure 4-9: Retrieved Wind - One revolution from December 6th 2006

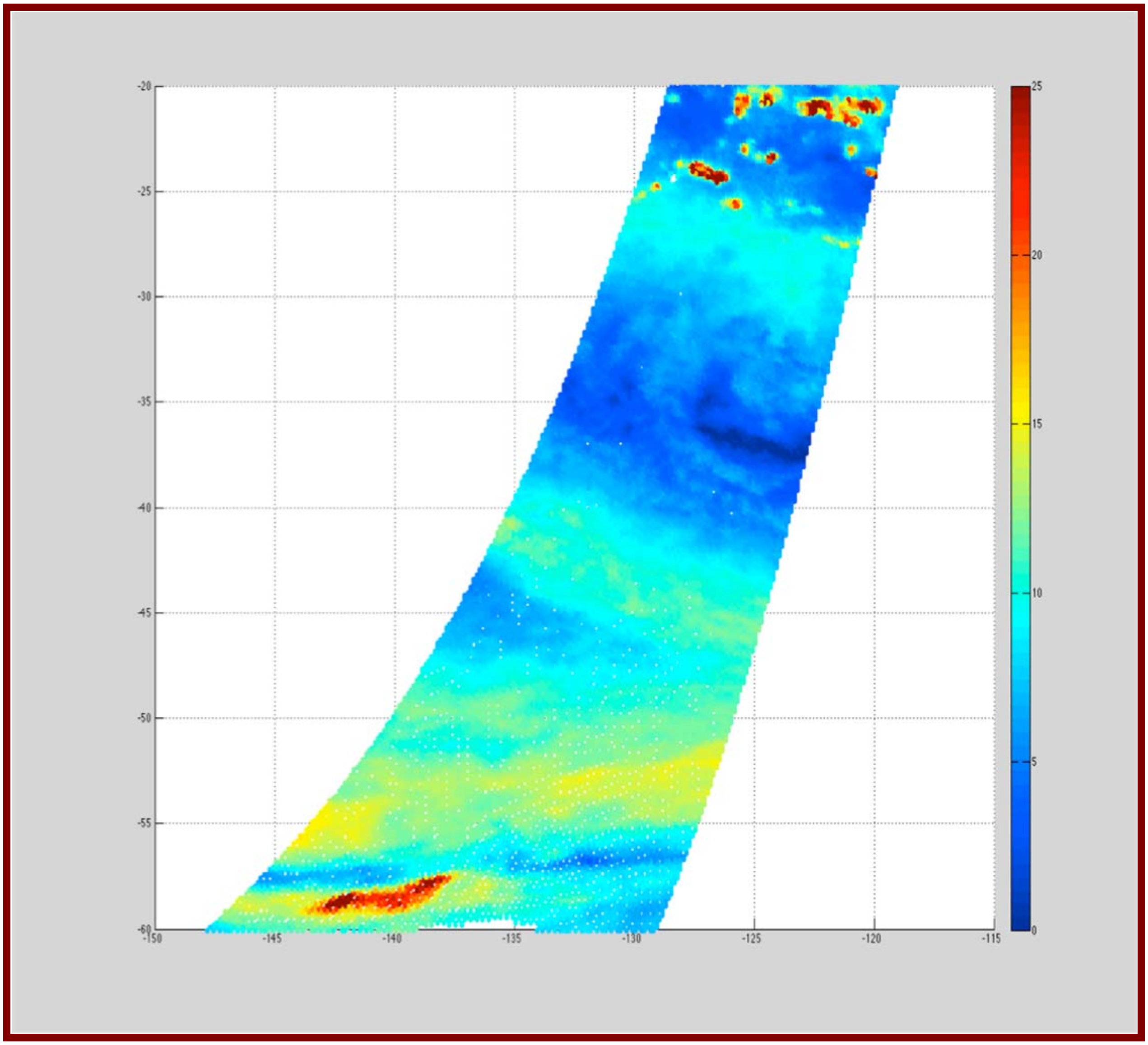


Figure 4-10: EDR wind speed - One revolution from December 6th 2006

As mentioned in chapter 3, the data set with rain present was collected one day every other week for an entire year. A higher resolution was required for this procedure, so each point was collocated and averaged within an 1/8 degree x 1/8 degree box. Due to the data high resolution, each data file contained over 100,000 points.

Data to train the statistical regression needed to reflect an entire year. To minimize the amount of data per file, data containing rain less than 0.1mm were filtered out before the regression was performed. This entire data set was used to execute the rain rate statistical regression. Another data set from WindSat, different revolutions than those used for the regression, was collected to test the rain rate retrieval. As an example, data collected on October 5th and October 13th 2006 is shown in Figure 4-11.

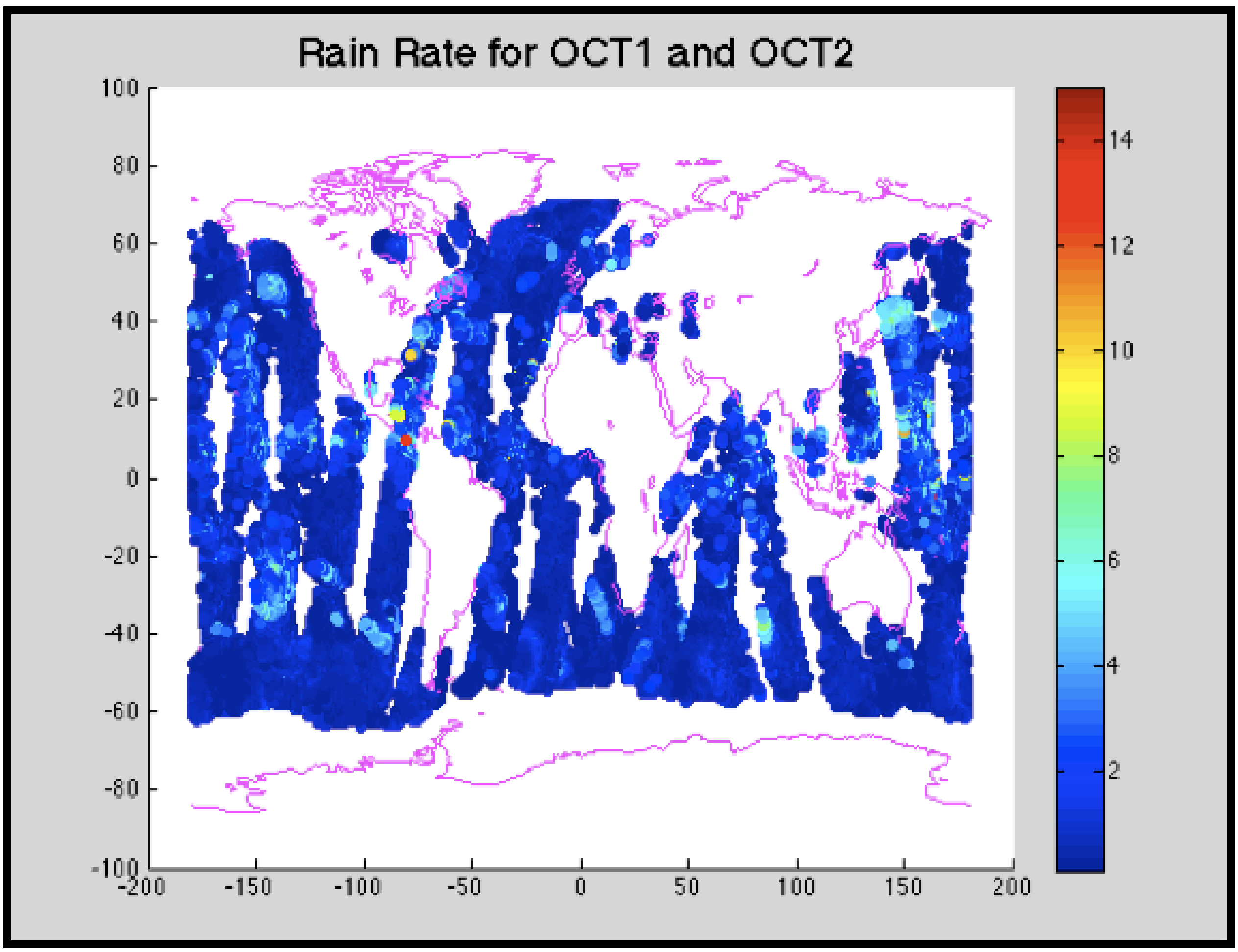


Figure 4-11: Geographical location for data points in the month of October

4.2. GDAS vs. EDR values

To confirm and test data value accuracy for these retrievals, the EDR water vapor values, wind speed values, upwelling brightness temperatures at 24GHz and 37GHz, downwelling brightness temperatures at 24GHz and 37GHz, and transmissivity of the atmosphere at 24GHz and 37GHz were compared to the GDAS model. The comparison between WindSat water vapor and GDAS simulated water vapor is presented in figure 4-12 with the color scale being proportional to the density of points. It can be seem on this plot that these two data sets are well correlated (correlation coefficient of 0.73), and a small bias is present in the EDR water vapor data set.

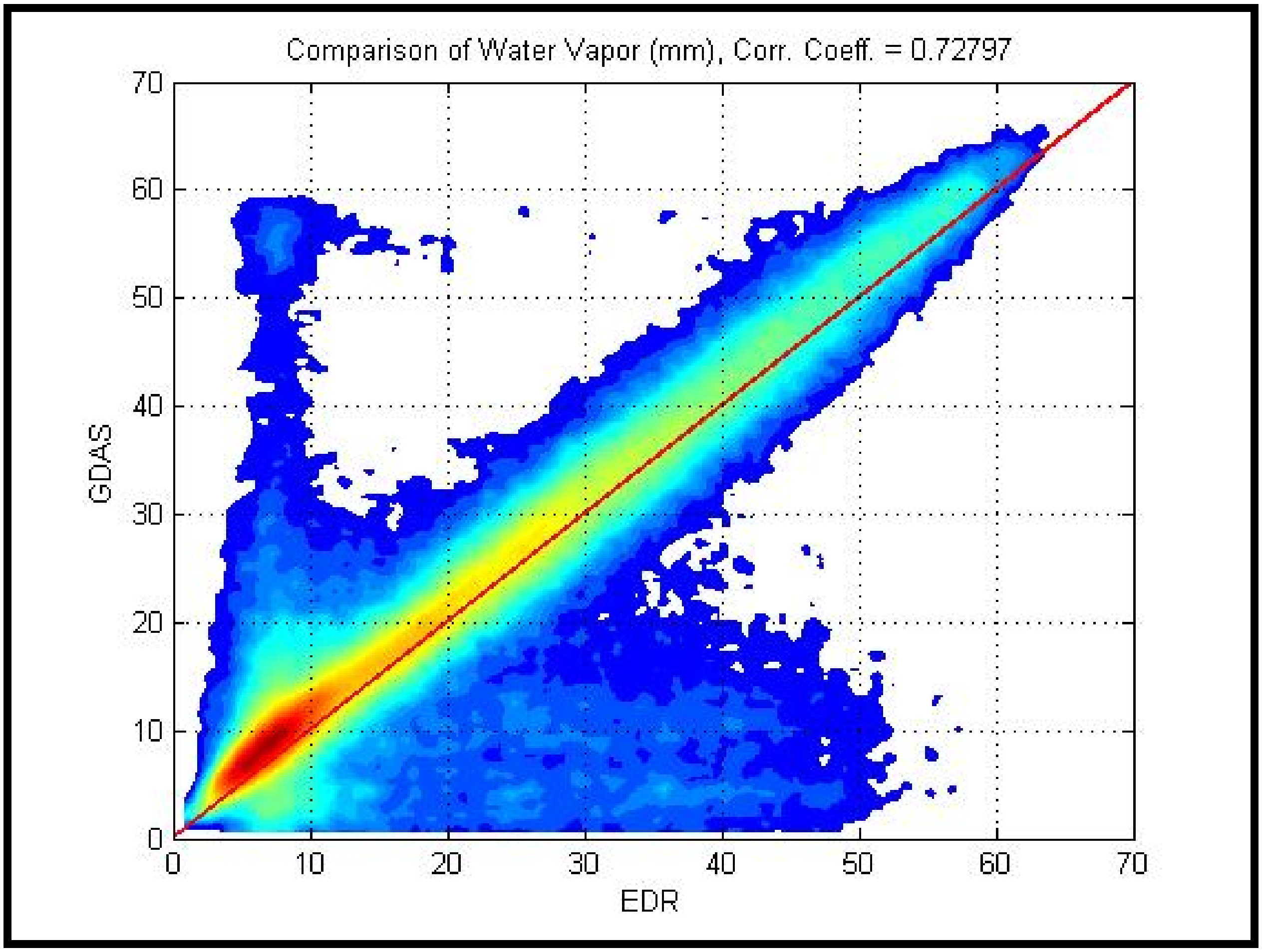


Figure 4-12: Water vapor comparison

WindSat and GDAS wind speed comparisons shown in figure 4-13 exhibit more scatter (correlation coefficient of 0.6), compared to the water vapor results.

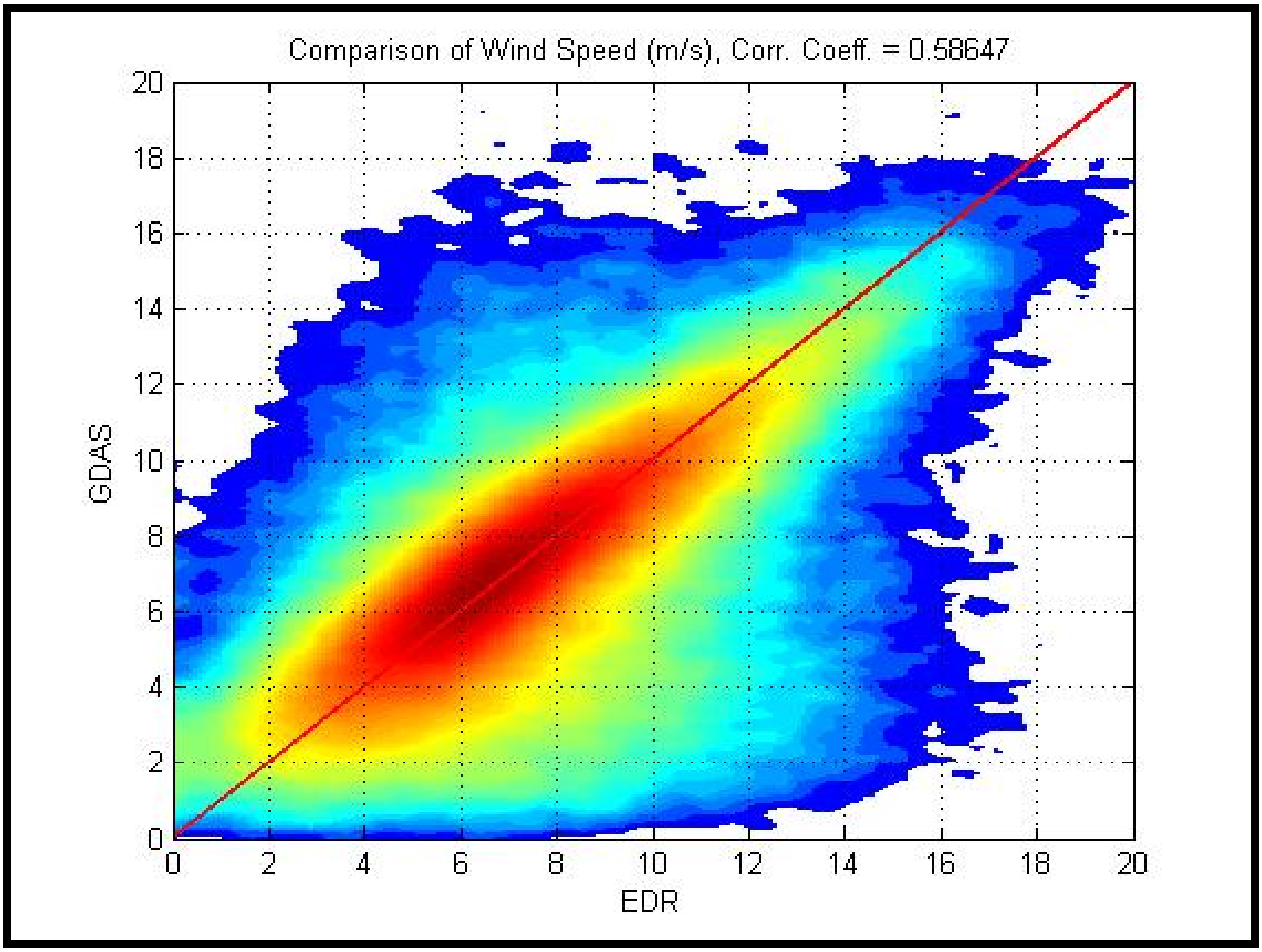


Figure 4-13: Wind speed comparison

The comparison between WindSat upwelling brightness temperature and GDAS simulated upwelling brightness temperature is presented in figures 4-14 and 4-15. It can be seem on these plots that these two data sets are well correlated except for small off-set biases, which are expected.

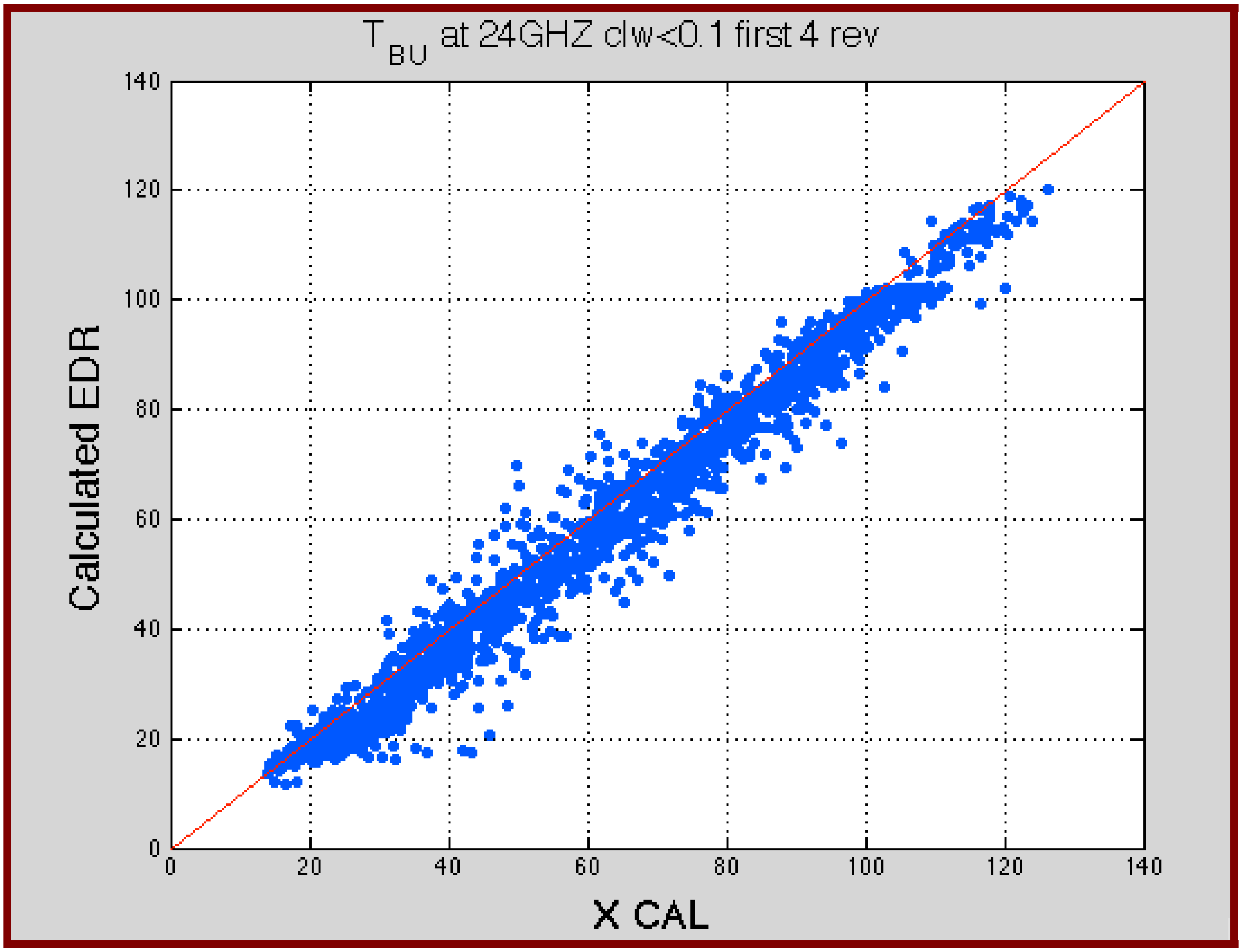


Figure 4-14: Tbu comparison at 24GHz

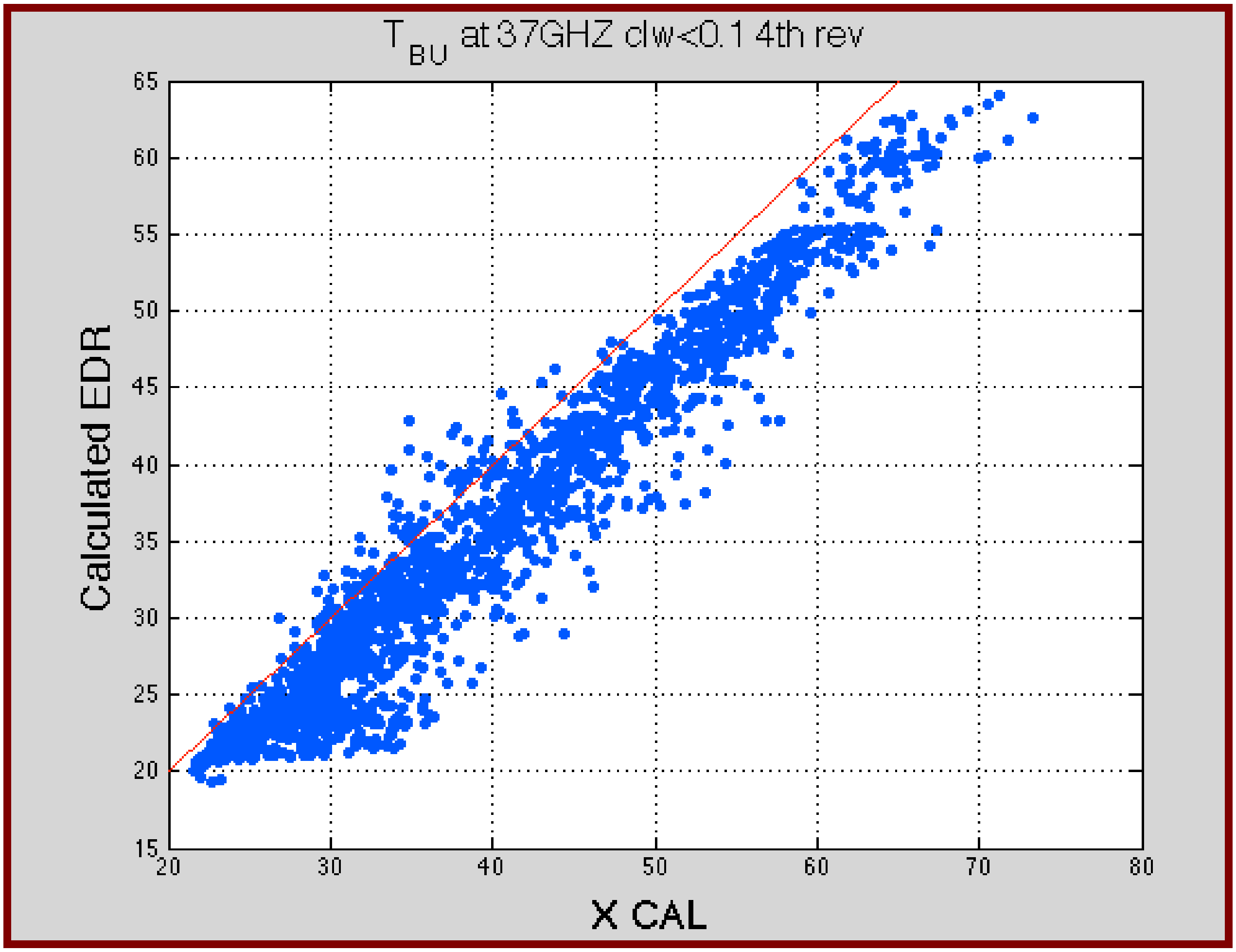


Figure 4-15: Tbu comparison at 37GHz

It can be seen on figures 4-16 and 4-17 the comparison between WindSat downwelling brightness temperature and GDAS simulated downwelling brightness temperatures are also similar.

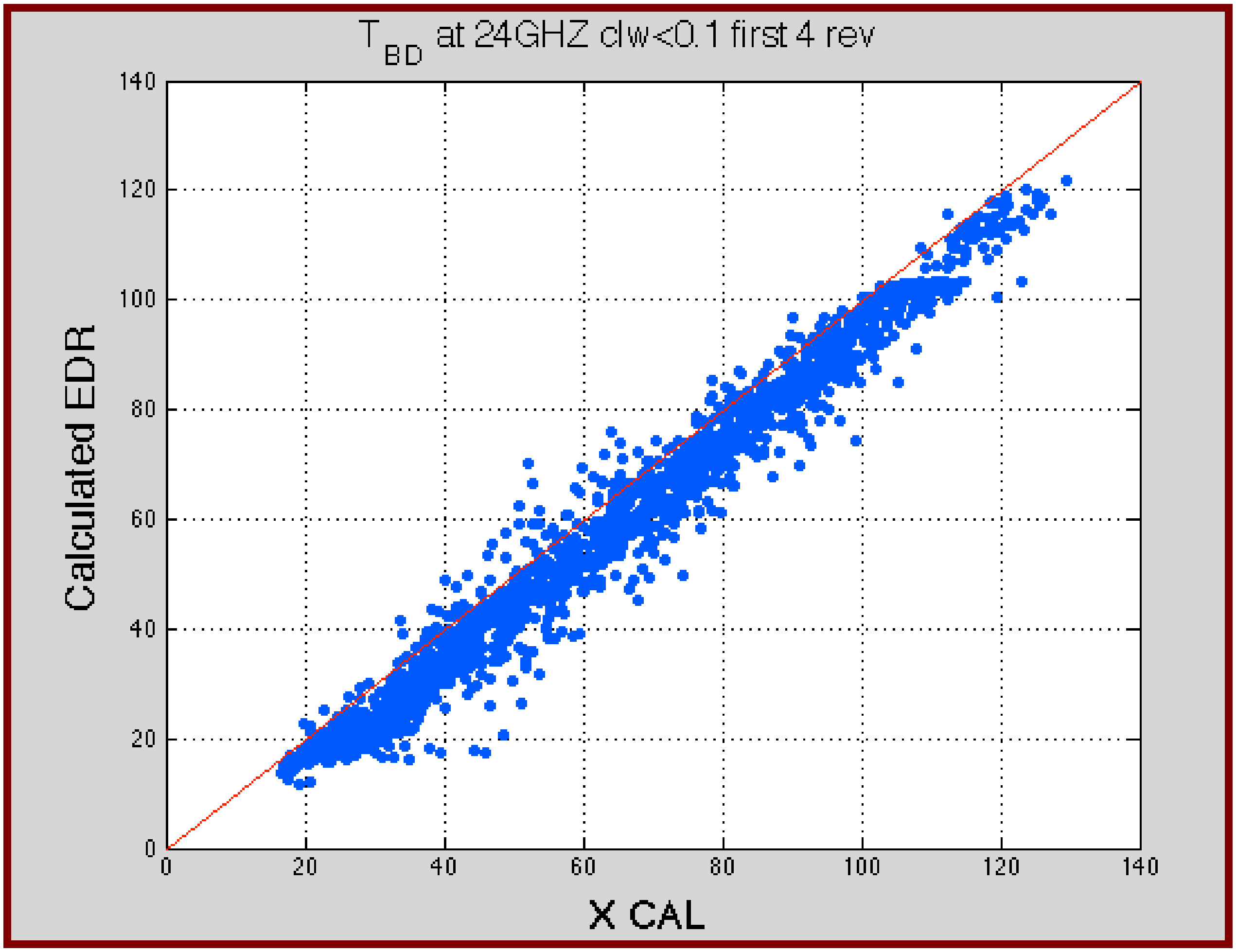


Figure 4-16: Tbd comparison at 24GHz

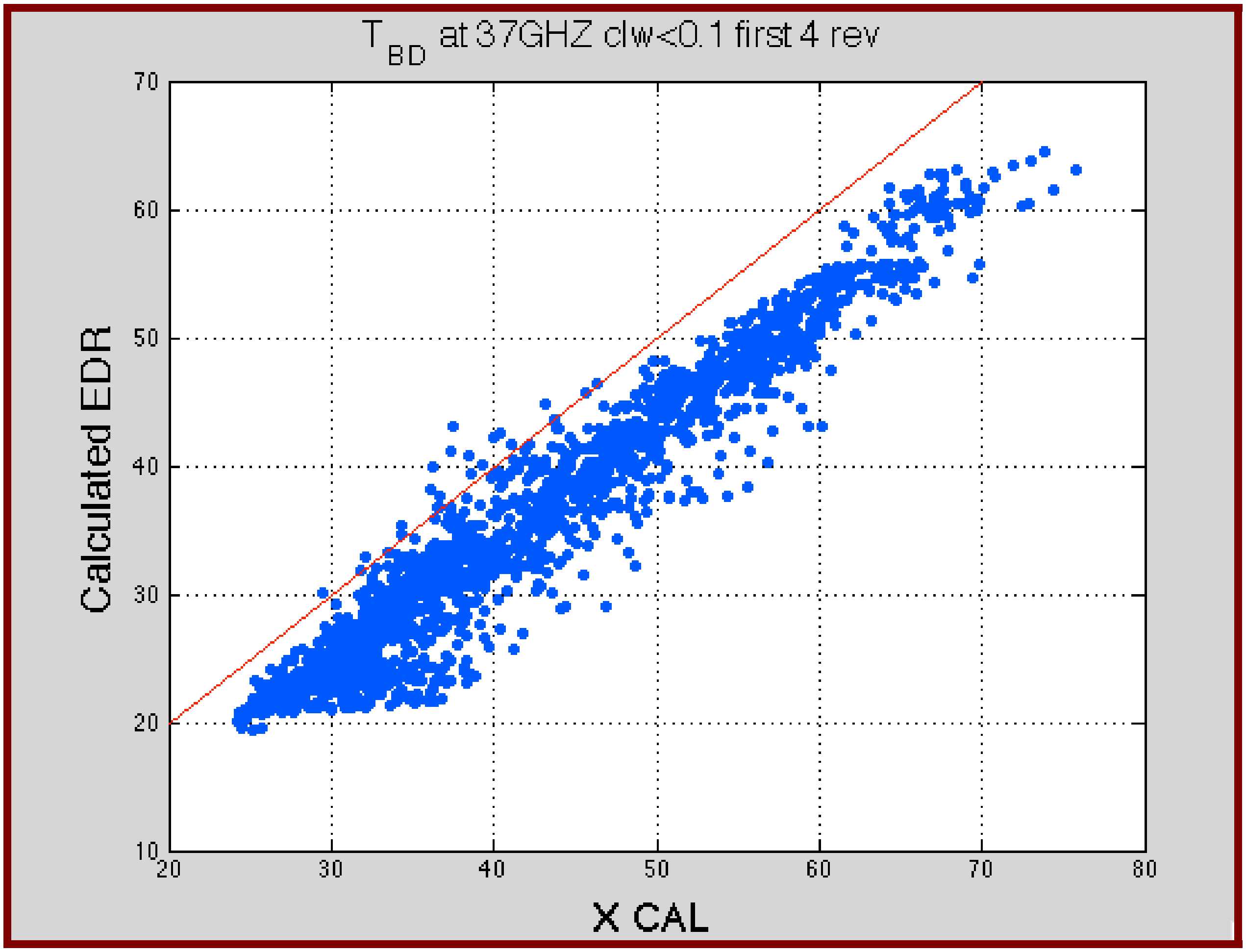


Figure 4-17: Tbd comparison at 37GHz

The comparison between WindSat atmospheric transmissivity and GDAS simulated atmospheric transmissivity is presented in figure 4-18 and 4-19. It can be seem on these plots that these two data sets are well correlated,

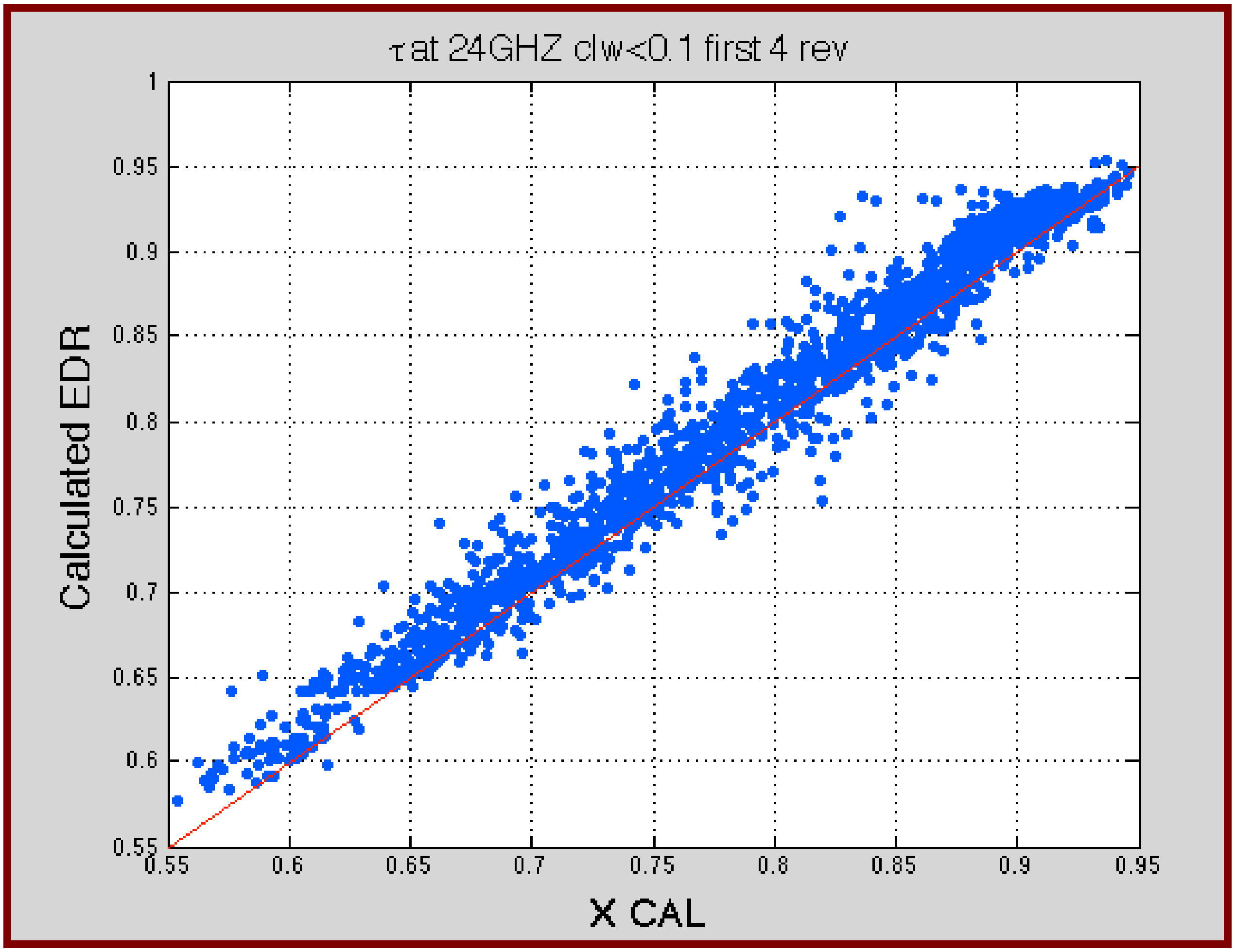


Figure 4-18: Atmospheric transmissivity comparison at 24GHZ

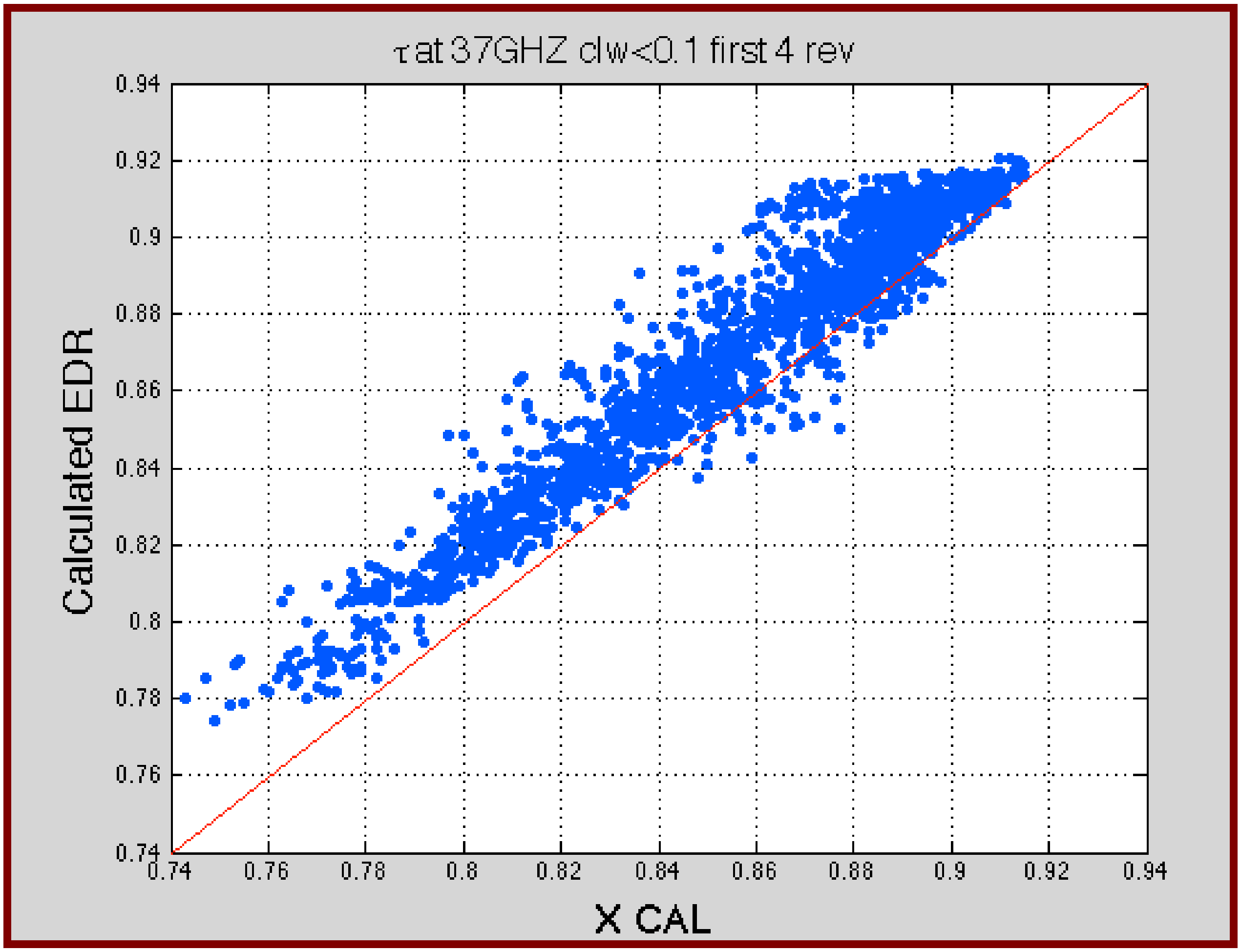


Figure 4-19: Atmospheric transmissivity comparison at 37GHz

4.3. Rain Rate Retrieval

Figure 3-13 in chapter 3 shows the scattered density plot where the rain rate retrieval was compared to WindSat EDR rain using a 3rd degree statistical regression. For a better understanding of this regression and retrieved rain rate values, figure 4-20 and figure 4-21 show the images of one revolution in September 1st 2006 (This day was not included in the data set used for calculating the regression).

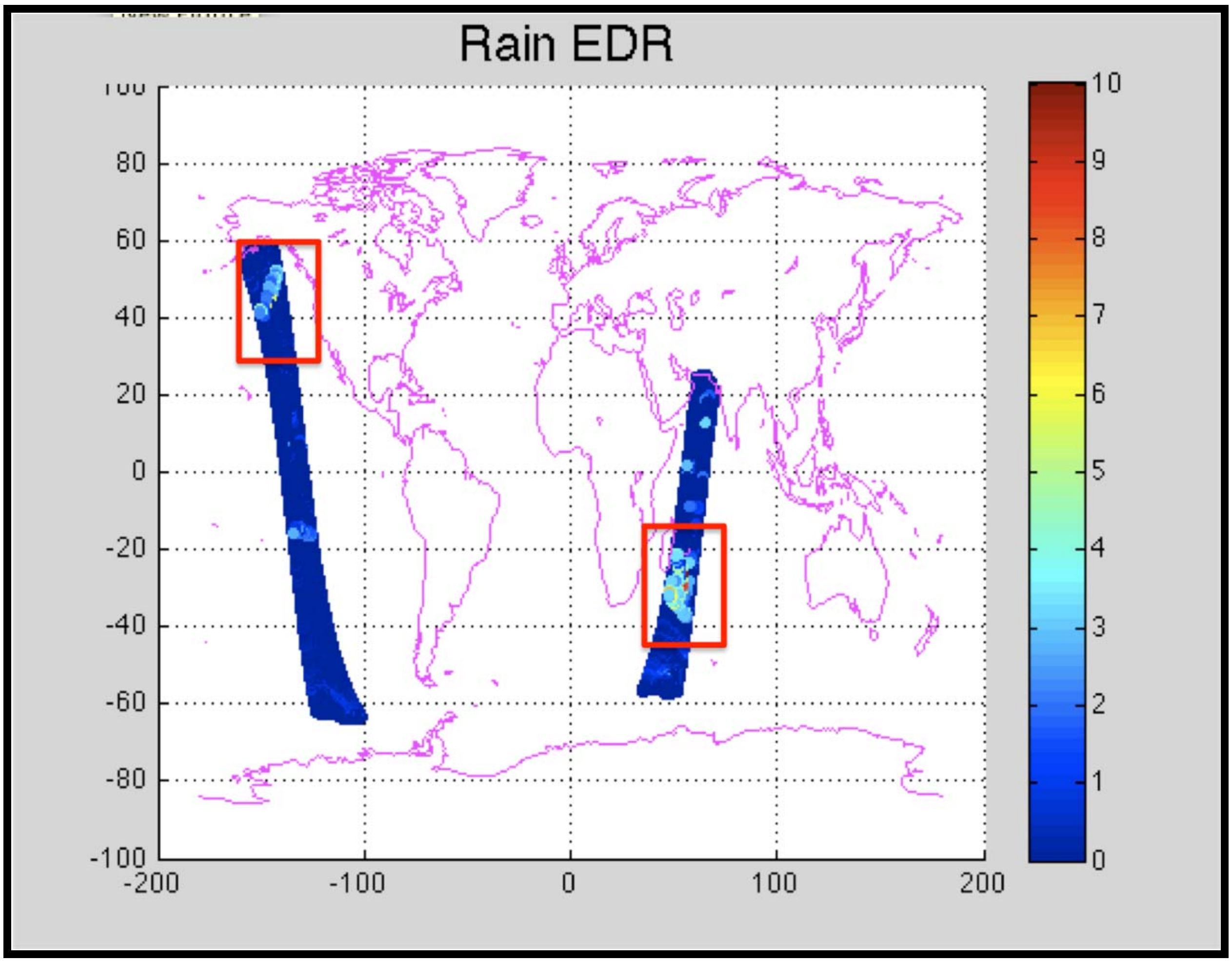


Figure 4-20: EDR Rain Image for one revolution in September 1st 2006

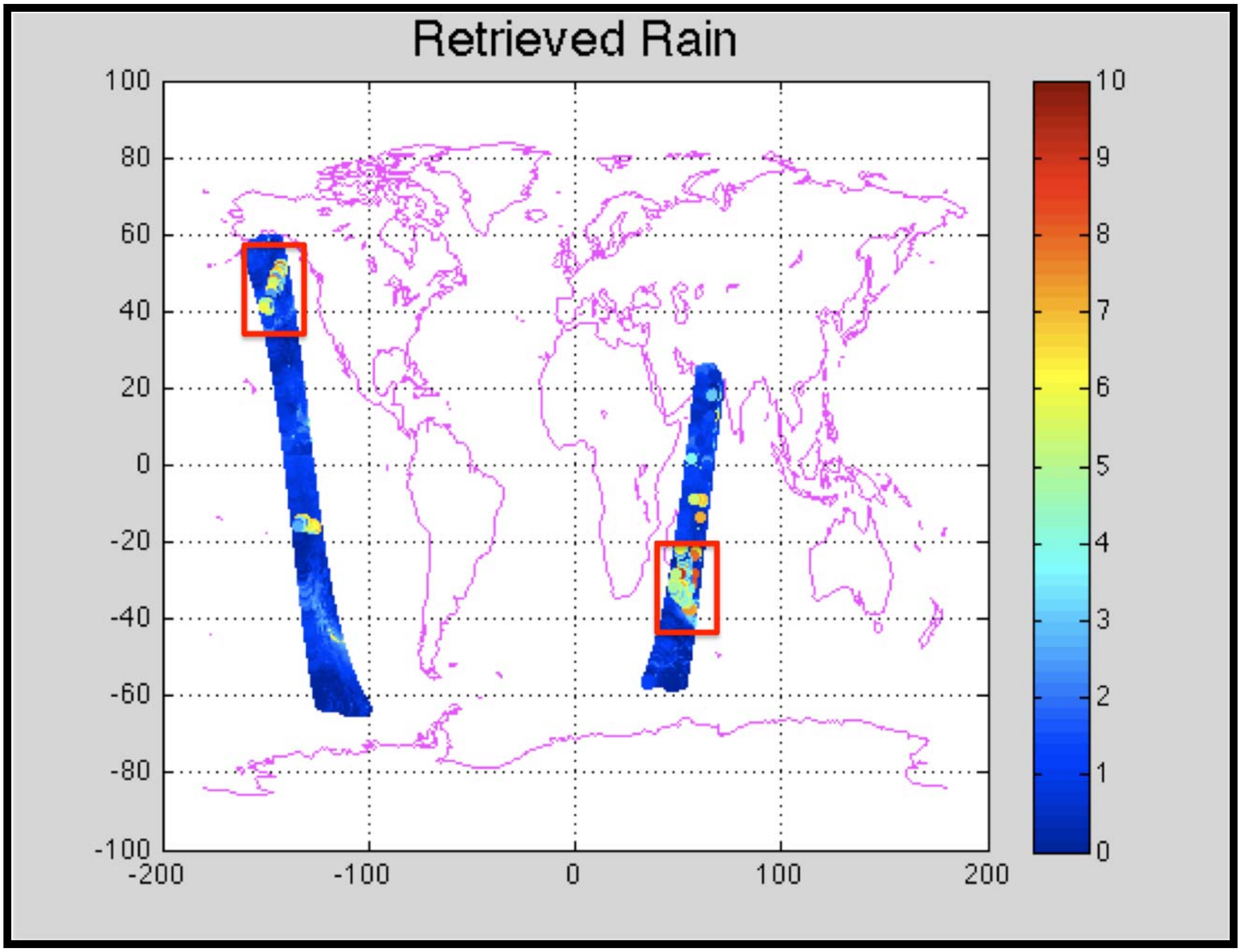


Figure 4-21: Retrieved rain image for one revolution in September 1st 2006

If the sections marked on the images are magnified, a more detailed comparison can be observed.

Figures 4-22 and 4-23 are images of EDR rain and retrieved rain for the north highlighted section. These two rain patterns are highly correlated is shown in figure 4-22, but it appears that the retrieved rain rates are biased high by a factor of ~ 2 compared to EDR rain rate.

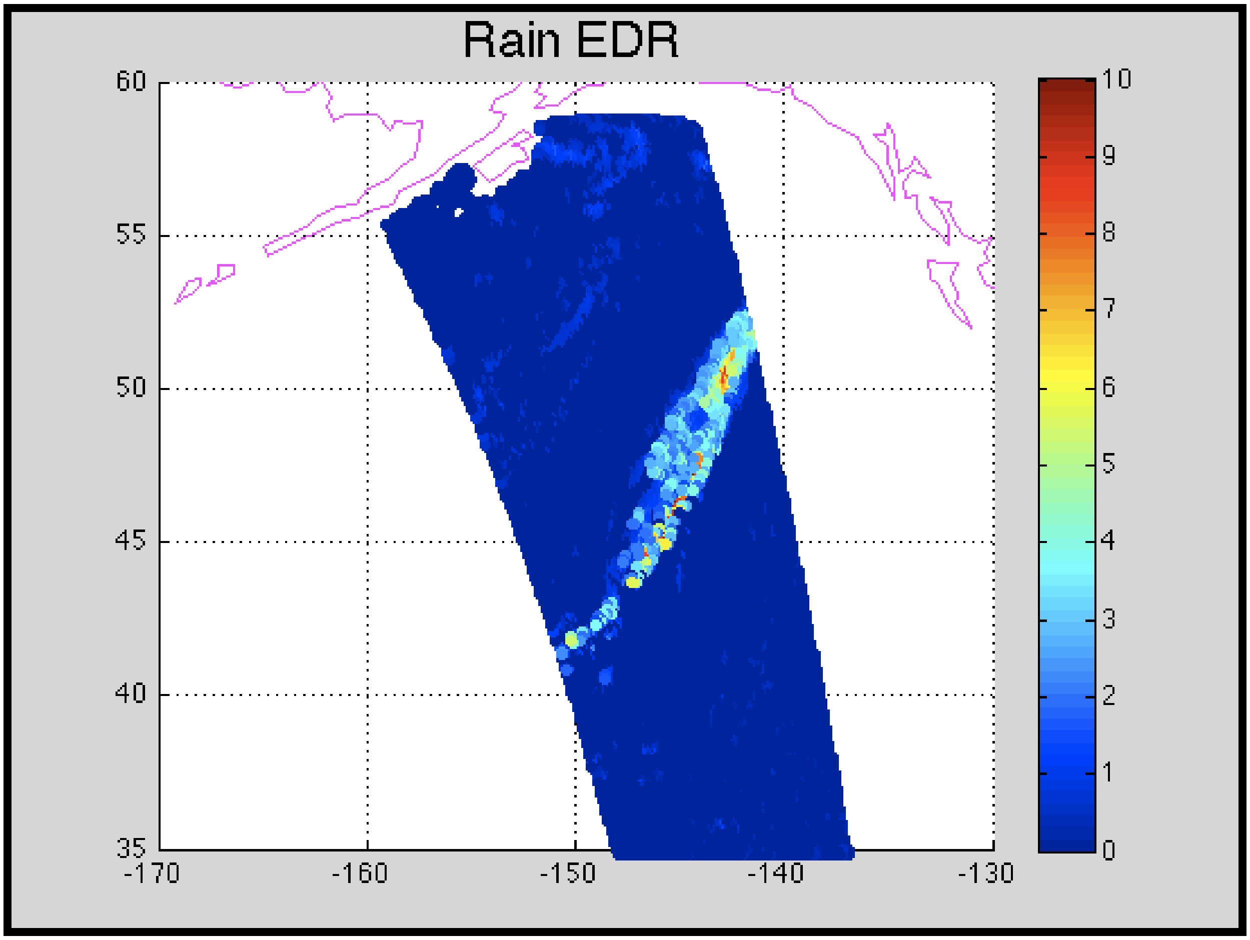


Figure 4-22: EDR rain - September 1st 2006

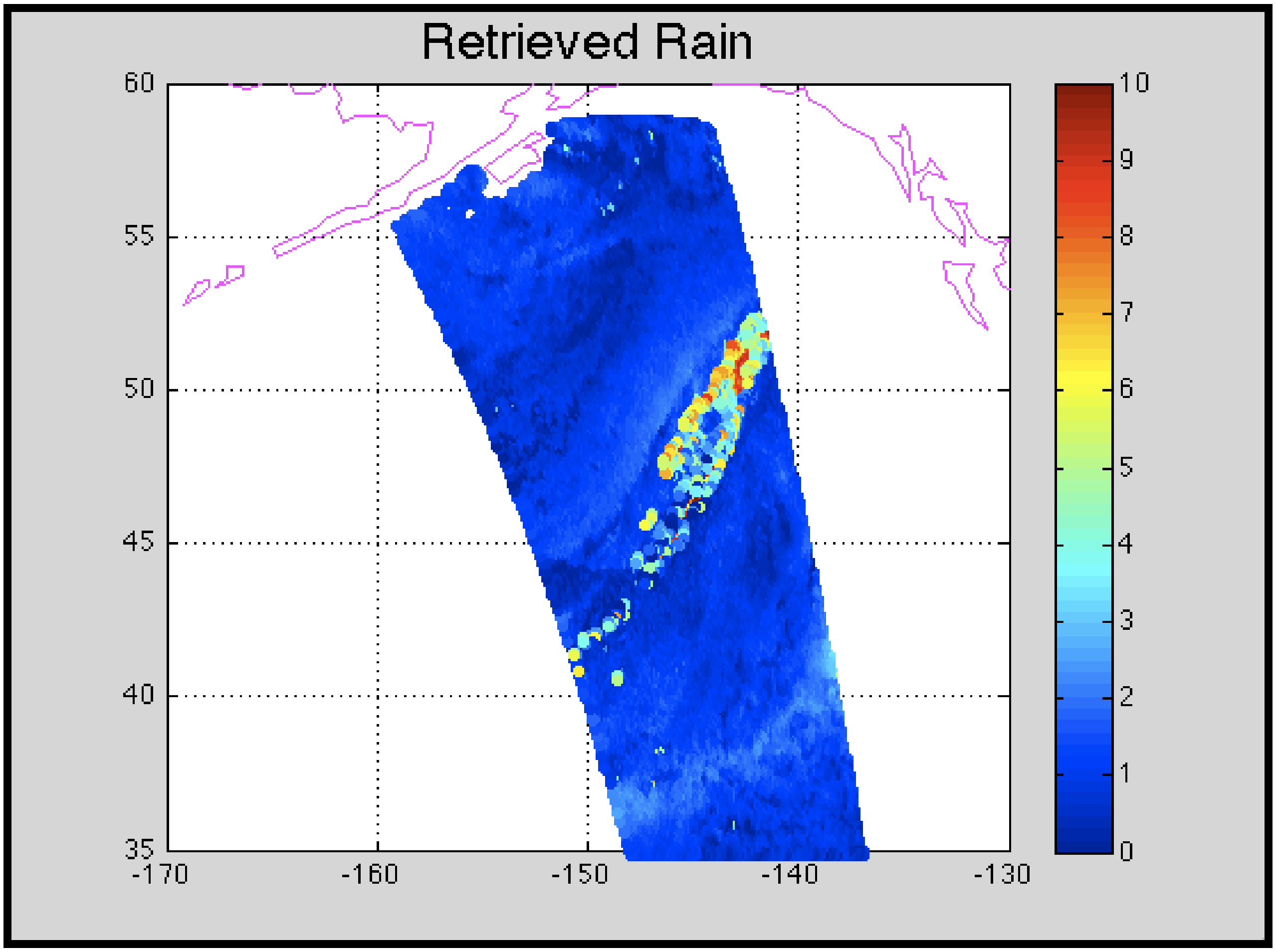


Figure 4-23: Retrieved rain - September 1st 2006

Figures 4-24 and 4-25 are images of EDR rain and retrieved rain for the south highlighted section. In comparing these two images, rain patterns are also highly correlated (same as in the previous image). However, there are regions of light retrieved rain values that are false alarms. A false alarm is defined as a region (seen in figure 4-23 and figure 4-25) where the retrieved values show light rain patterns, whereas the EDR image shows the corresponding areas as rain-free. Further investigation is required to determine why these false alarms are being produced.

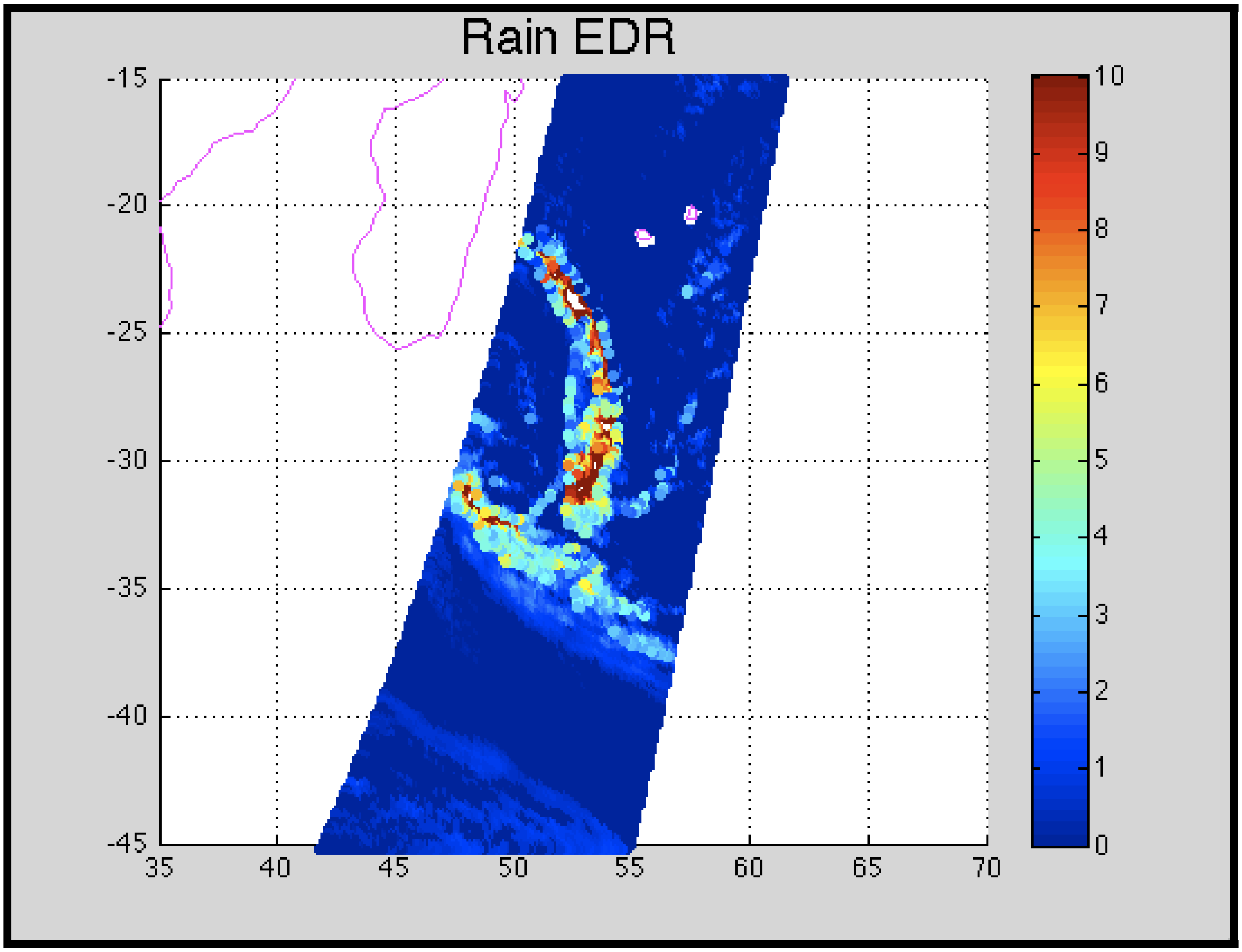


Figure 4-24: EDR rain - September 1st 2006

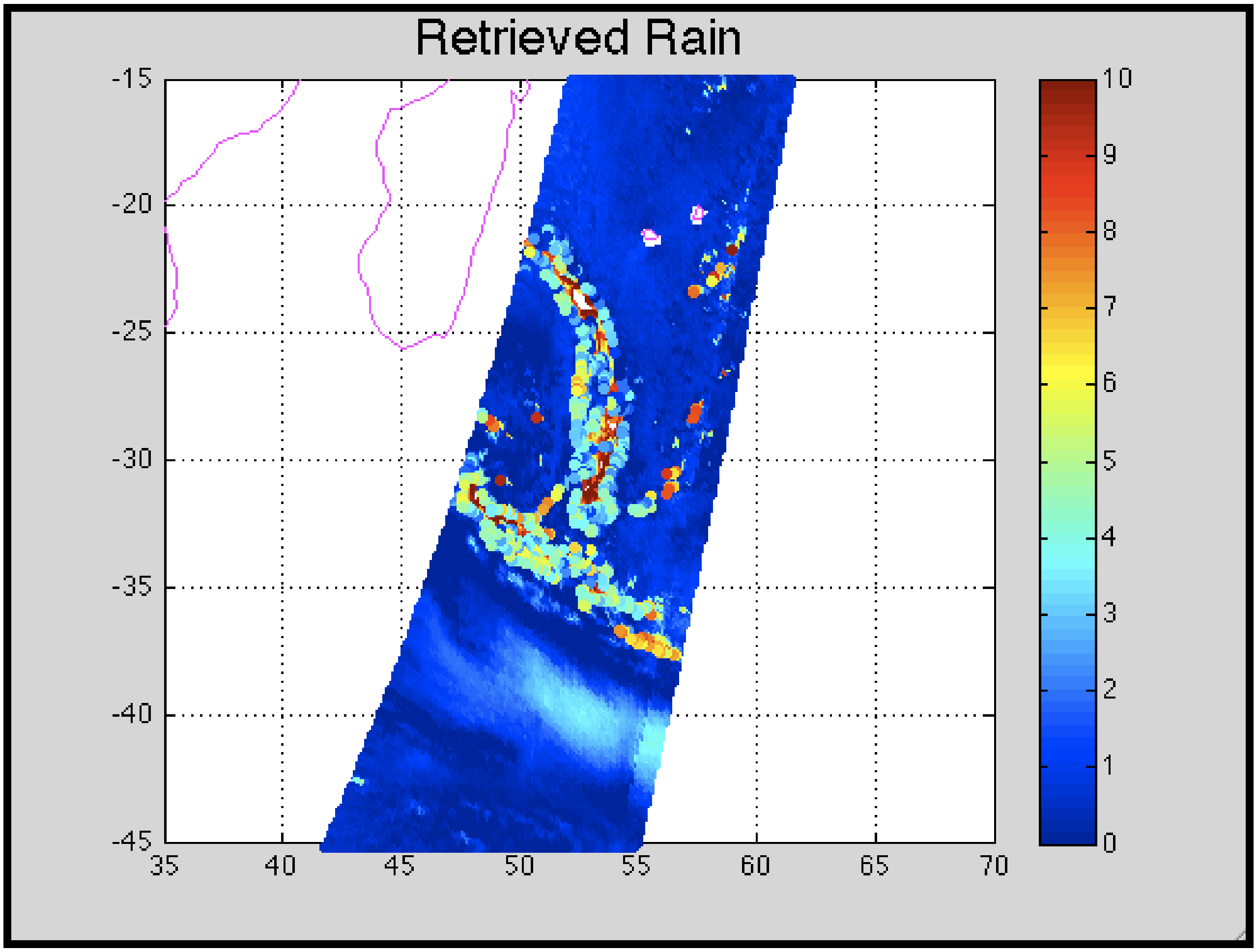


Figure 4-25: Retrieved rain - September 1st 2006

A comparison between Retrieved rain and EDR rain for the month of September and October is shown in figure 4-26. EDR rain rate less than 0.1mm was excluded from these retrievals due to the amount of data points. Also to reduce false alarms, retrieved rain rates less than 0.5mm were excluded from the scatter density plot. This was only partially successful, as some false alarms shown on the previous images can be seen on this comparison as well.

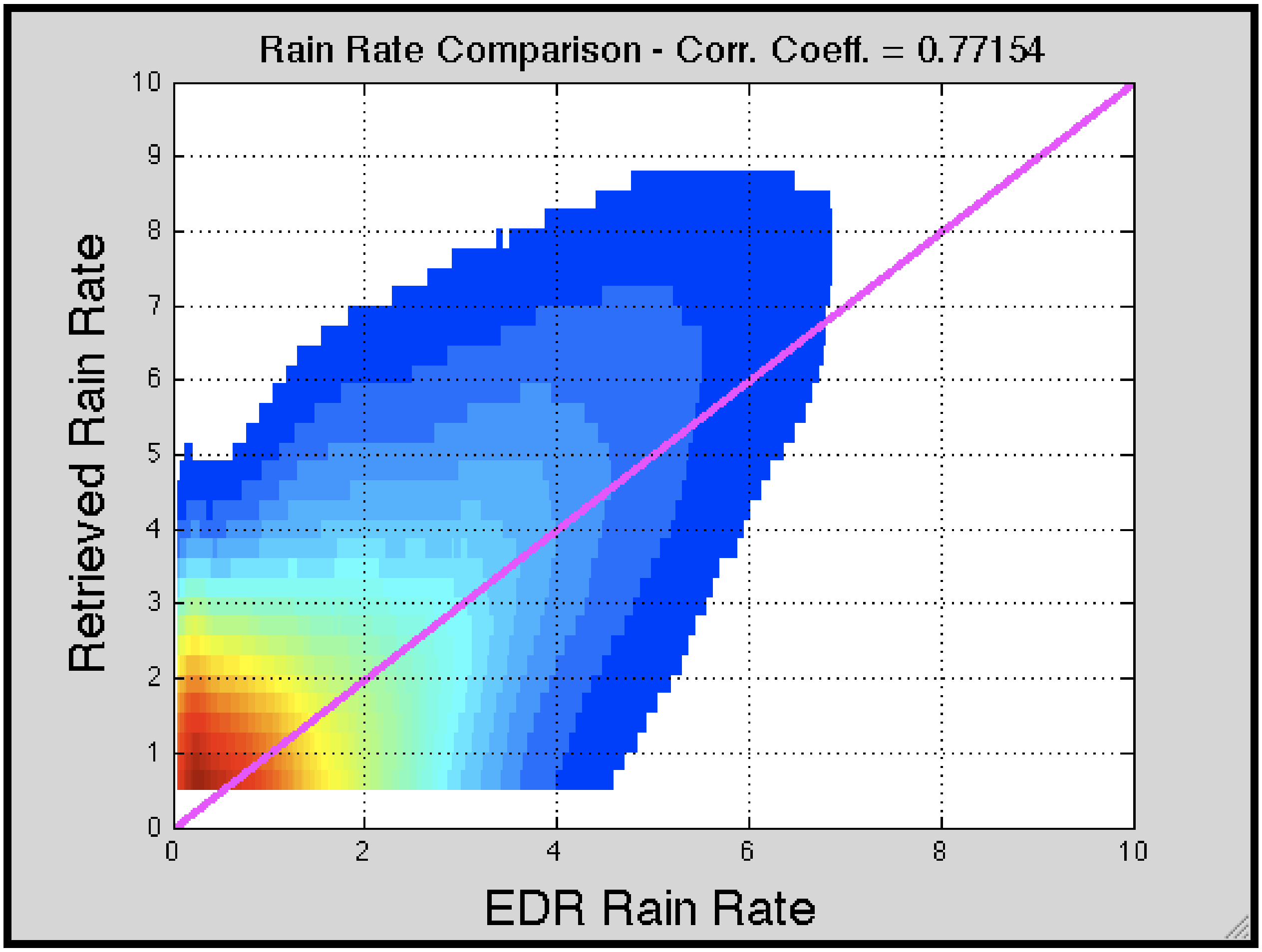


Figure 4-26: Rain Rate comparison - September

CHAPTER 5. CONCLUSION

This thesis presented a new rain rate retrieval algorithm for WindSat using only three channels: 23.8 GHz horizontal polarization, 36.5 GHz vertical polarization, and 36.5 GHz horizontal polarization. Water vapor and wind speed retrievals have been included in this algorithm as well; however, for these retrievals four WindSat channels were used: 23.8GHz vertical and horizontal polarization, and 36.5GHz vertical and horizontal polarization. Cloud liquid water retrieval was excluded from this algorithm, which probably contributes to false alarm light rain retrievals.

Wind speed retrievals only provided accurate values in the absence of rain. On the other hand, the retrieved water vapor values were accurate in scenarios with no rain, and in the presence of light rain. When the EDR rain rate was greater than 4mm/hr, the retrieved water vapor exhibited similar patterns as the EDR vapor, but the retrieved values were slightly higher. A possible reason for this discrepancy could be the absence of rain in the statistical regression’s training data set.

Rain increases the brightness temperature measured by the radiometer. Because the water vapor regression excluded rain, the excess in brightness temperature due to rain is not expressed in the regressed polynomial. As a result, when rain data are included in the water vapor retrieval, the excess in brightness temperature due to rain appears in the retrieval and is perceived as an increase in water vapor content.

The wind speed retrieval does not provide satisfactory values in the presence of rain. In the previous chapter, multiple figures showed comparisons for EDR rain, retrieved water vapor and retrieved wind. It appears that the retrieval exhibits increased water vapor in the presence of rain. Further, this increase in the retrieved water vapor content seems to produce very low wind speed (or negative) values in the retrieval. As a result, the relationship given by the regress polynomial cannot be used to retrieve wind speed when rain is present. This is a usual effect that most atmospheric and surface geophysical retrievals are severely degraded in the presence of rain.

It is known that rain rate has an exponential statistical distribution. The majority of rain rate values are located between 0 to 0.2 mm/h. Wentz mentions in his paper [1] that rain rates greater than 0.2mm/hr represent less than 10% of values collected over a year, and oceanic average rainfall is around 3 mm per day to 4 mm per day depending on the distance from the Equator. These percentages are illustrated in the density plot for the month of September and October shown in chapter 4. The majority of the data points in this figure lie between 0.1mm/hr to 2mm/hr.

When the initial allocation for WindSat EDR and IDR values was done, data points with -9999 wind speed values were excluded. Later it was discovered that some of those -9999 wind speed values represented an error on the WindSat EDR wind speed retrieval due to high rain rates. A few of the images in chapter 4 show these missing data for high rain rate. By excluding those data points, the collected EDR rain rate was set to a maximum value around 17 mm/h. Therefore, this rain rate algorithm will only be accurate for rain rates between 0 mm/h to 15 mm/h. Further research and testing must be performed to increment this range.

The correlation between retrieved rain rate and EDR rain rate was 0.77. This value provides confidence on the accuracy of this algorithm; however, the scattered comparison in the images provided a significant number of false alarms. Further investigations should be performed to find the cause for these false alarms and reduce the number of occurrences.

To translate this algorithm for MWR, a new statistical regression will need to be executed where only three WindSat channels will be used to retrieve water vapor and wind speed. Another alternative is to use the GDAS model to retrieve these two values. If this alternative is selected, further testing and validation must be performed when rain is present in the data.

APPENDIX A: QUADRATIC EQUATION VARIABLES

$$A_1 = T_{U24}$$
$$B_1 = (-T_{U24} \times \tau_{V24} \times \tau_{O24}) + (\tau_{O24} \times \tau_{V24} \times \varepsilon_{24v} \times SST) + (\tau_{O24} \times \tau_{V24} \times (1-\varepsilon_{24v}) \times T_{D24}$$
$$C_1 = (-(1-\varepsilon_{24v}) \times T_{D24} \times \tau_{V24}^2 \times \tau_{O24}^2) + (2.7 \times (1-\varepsilon_{24v}) \times \tau_{V24}^2 \times \tau_{O24}^2)$$

$$A_2 = T_{U24}$$
$$B_2 = (-T_{U24} \times \tau_{V24} \times \tau_{O24}) + (\tau_{O24} \times \tau_{V24} \times \varepsilon_{24h} \times SST) + (\tau_{O24} \times \tau_{V24} \times (1-\varepsilon_{24h}) \times T_{D24}$$
$$C_2 = (-(1-\varepsilon_{24h}) \times T_{D24} \times \tau_{V24}^2 \times \tau_{O24}^2) + (2.7 \times (1-\varepsilon_{24h}) \times \tau_{V24}^2 \times \tau_{O24}^2)$$

$$A_3 = T_{U37}$$
$$B_3 = (-T_{U37} \times \tau_{V37} \times \tau_{O37}) + (\tau_{O37} \times \tau_{V37} \times \varepsilon_{37v} \times SST) + (\tau_{O37} \times \tau_{V37} \times (1-\varepsilon_{37v}) \times T_{D37}$$
$$C_3 = (-(1-\varepsilon_{37v}) \times T_{D37} \times \tau_{V37}^2 \times \tau_{O37}^2) + (2.7 \times (1-\varepsilon_{37v}) \times \tau_{V37}^2 \times \tau_{O37}^2)$$

$$A_4 = T_{U37}$$
$$B_4 = (-T_{U37} \times \tau_{V37} \times \tau_{O37}) + (\tau_{O37} \times \tau_{V37} \times \varepsilon_{37h} \times SST) + (\tau_{O37} \times \tau_{V37} \times (1-\varepsilon_{37h}) \times T_{D37}$$
$$C_4 = (-(1-\varepsilon_{37h}) \times T_{D37} \times \tau_{V37}^2 \times \tau_{O37}^2) + (2.7 \times (1-\varepsilon_{37h}) \times \tau_{V37}^2 \times \tau_{O37}^2)$$

APPENDIX B: MATLAB SCRIPTS

```
function [DATA_case1] = forward_model1_case1(DATA_48)

%Here V>48, abs(SST-TV)>20, (SST-TV)<0,SST<301




% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_48];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
```

```
CLW = DATA(:,8);
V = DATA(:,5);
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%     T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%    T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);
```

```
[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);

[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);


% % id_e24v = find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));

% % E_24v = E_24v(id_e24v);

% % E_24h = E_24h(id_e24v);

% % E_37v = E_37v(id_e24v);

% % E_37h = E_37h(id_e24v);

% % W = W(id_e24v);

% % V = V(id_e24v);

% % SST = SST(id_e24v);

% % T_B24v = T_B24v(id_e24v);

% % T_B24h = T_B24h(id_e24v);

% % T_B37v = T_B37v(id_e24v);

% % T_B37h = T_B37h(id_e24v);




%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..

... first row is b_0, second row is b_1, and so on until b_7
```

```
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];




%           if (V>48)
             T_v = 301.16;
%            else
%            T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%            end

%            if (abs(SST-T_v)>20)
fid_sv20 = find((abs(SST-T_v))>20);
if fid_sv20>0;
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
```

```
T_B24h = T_B24h(fid_sv20);

T_B37v = T_B37v(fid_sv20);

T_B37h = T_B37h(fid_sv20);

V = V(fid_sv20);

W = W(fid_sv20);

V_edr = V_edr(fid_sv20);

W_edr = W_edr(fid_sv20);

SST = SST(fid_sv20);

CLW = CLW(fid_sv20);

E_24v = E_24v(fid_sv20);

E_24h = E_24h(fid_sv20);

E_37v = E_37v(fid_sv20);

E_37h = E_37h(fid_sv20);

R = R(fid_sv20);

%          if ((SST-T_v)<0)

fid_neg = find((SST-T_v)<0);

latitude = latitude(fid_neg);

longitude = longitude(fid_neg);

T_B24v = T_B24v(fid_neg);

T_B24h = T_B24h(fid_neg);

T_B37v = T_B37v(fid_neg);

T_B37h = T_B37h(fid_neg);
```

```
V = V(fid_neg);
W = W(fid_neg);
V_edr = V_edr(fid_neg);
W_edr = W_edr(fid_neg);
SST = SST(fid_neg);
CLW = CLW(fid_neg);
E_24v = E_24v(fid_neg);
E_24h = E_24h(fid_neg);
E_37v = E_37v(fid_neg);
E_37h = E_37h(fid_neg);
R = R(fid_neg);

% if (SST<301)
fid_300 = find(SST<301);
latitude = latitude(fid_300);
longitude = longitude(fid_300);
T_B24v = T_B24v(fid_300);
T_B24h = T_B24h(fid_300);
T_B37v = T_B37v(fid_300);
T_B37h = T_B37h(fid_300);
V = V(fid_300);
W = W(fid_300);
```

```
V_edr = V_edr(fid_300);

W_edr = W_edr(fid_300);

SST = SST(fid_300);

CLW = CLW(fid_300);

E_24v = E_24v(fid_300);

E_24h = E_24h(fid_300);

E_37v = E_37v(fid_300);

E_37h = E_37h(fid_300);

R = R(fid_300);

                    gui = -14;

%                   else

%                       gui = 14;

%                   end

%               else

%                   gui = 1.05.*(SST - T_v).*(1 - ((SST-T_v).^2)./1200);

%               end


% for I = 1:2;

T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 + b(5,1).*V.^4 + ...

    b(6,1).*gui;

T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 + b(5,2).*V.^4 + ...

    b(6,2).*gui;
```

```
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3 + b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);
```

```
%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));

%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));


% tau_atm24 = tau_v24.*tau_o24.*tau_l24;

% tau_atm37 = tau_v37.*tau_o37.*tau_l37;


%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));

%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));

%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));

%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);

% end


%Finding the height of rain


% if (SST<301)


   H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;

% else

%     H_l = 2.96;

% end
```

```
%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;

T_cl = (SST - (6.5).*H_l);


L = CLW;
A_l24test =0.0891.*(1-0.0281.*(T_cl-283)).*L;
tau_l24test = exp(-secd(theta_i24).*A_l24test);
%tau_l24 = 1;
A_l37test =0.2027.*(1-0.0261.*(T_cl-283)).*L;
tau_l37test = exp(-secd(theta_i37).*A_l37test);
%tau_l37 = 1;


A_1 = T_U24;
B_1 = (-T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
   (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
   (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
```

```
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

A_2 = T_U24;
B_2 = (-T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

A_3 = T_U37;
B_3 = (-T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37;
B_4 = (-T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
```

```
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-T_B24v)).*(C_1)))))./(2.*C_1);

sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-T_B24v)).*(C_1)))))./(2.*C_1);

sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-T_B24h)).*(C_2)))))./(2.*C_2);

sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-T_B24h)).*(C_2)))))./(2.*C_2);



tau_l24 = (sol2_24+sol4_24)./2;


%tau_l24(tau_l24>1)=1;


dif1_24 = abs(sol1_24-sol3_24);

dif2_24 = abs(sol1_24-sol4_24);

dif3_24 = abs(sol2_24-sol3_24);

dif4_24 = abs(sol2_24-sol4_24);


%

% sol1_37 = (-c_9+sqrt(((c_9).^2) -(4*(c_10).*(c_11)))))./(2.*c_10);

% sol2_37 = (-c_9-sqrt(((c_9).^2) -(4*(c_10).*(c_11)))))./(2.*c_10);

% sol3_37 = (-d_9+sqrt(((d_9).^2) -(4*(d_10).*(d_11)))))./(2.*d_10);

% sol4_37 = (-d_9-sqrt(((d_9).^2) -(4*(d_10).*(d_11)))))./(2.*d_10);


sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-T_B37v)).*(C_3)))))./(2.*C_3);
```

```
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-T_B37v)).*(C_3)))))./(2.*C_3);

sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-T_B37h)).*(C_4)))))./(2.*C_4);

sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-T_B37h)).*(C_4)))))./(2.*C_4);


tau_l37 = (sol2_37+sol4_37)./2;

%tau_l37(tau_l37>1)=1;


dif1_37 = abs(sol1_37-sol3_37);

dif2_37 = abs(sol1_37-sol4_37);

dif3_37 = abs(sol2_37-sol3_37);

dif4_37 = abs(sol2_37-sol4_37);



[DATA_case1] = [T_B24v T_B24h T_B37v T_B37h V W ...

   SST CLW latitude longitude...

   sol2_24 sol4_24 sol2_37 sol4_37...

   R V_edr W_edr T_l H_l];

%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];




else
```

```
    display('case 1 no values for abs(SST-Tv) greater than 20');
    DATA_case1 = [];
end


function [DATA_case2] = forward_model1_case2(DATA_48)

%Here V>48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_48];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
```

```
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%     T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%    T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);
```

```
[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);

% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);



%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
```

```
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
                T_v = 301.16;
%              else
%              T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
fid_sv20 = find((abs(SST-T_v))>20);
if fid_sv20>0;
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
```

```
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
R = R(fid_sv20);
%                  if ((SST-T_v)<0)
fid_pos = find((SST-T_v)>0);
latitude = latitude(fid_pos);
longitude = longitude(fid_pos);
T_B24v = T_B24v(fid_pos);
T_B24h = T_B24h(fid_pos);
T_B37v = T_B37v(fid_pos);
T_B37h = T_B37h(fid_pos);
V = V(fid_pos);
W = W(fid_pos);
V_edr = V_edr(fid_pos);
W_edr = W_edr(fid_pos);
SST = SST(fid_pos);
CLW = CLW(fid_pos);
E_24v = E_24v(fid_pos);
E_24h = E_24h(fid_pos);
E_37v = E_37v(fid_pos);
E_37h = E_37h(fid_pos);
R = R(fid_pos);
```

```
% if (SST<301)
fid_300 = find(SST<301);
latitude = latitude(fid_300);
longitude = longitude(fid_300);
T_B24v = T_B24v(fid_300);
T_B24h = T_B24h(fid_300);
T_B37v = T_B37v(fid_300);
T_B37h = T_B37h(fid_300);
V = V(fid_300);
W = W(fid_300);
V_edr = V_edr(fid_300);
W_edr = W_edr(fid_300);
SST = SST(fid_300);
CLW = CLW(fid_300);
E_24v = E_24v(fid_300);
E_24h = E_24h(fid_300);
E_37v = E_37v(fid_300);
E_37h = E_37h(fid_300);
R = R(fid_300);
%                     gui = -14;
%                  else
                      gui = 14;
%                  end
%              else
```

```
%                  gui = 1.05.*(SST - T_v).*(1 - ((SST-
T_v).^2)./1200);
%              end

% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;
```

```
%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
```

```
%     H_l = 2.96;
% end


%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;

T_cl = (SST - (6.5).*H_l);


L = CLW;
A_l24test =0.0891.*(1-0.0281.*(T_cl-283)).*L;
tau_l24test = exp(-secd(theta_i24).*A_l24test);
%tau_l24 = 1;
A_l37test =0.2027.*(1-0.0261.*(T_cl-283)).*L;
tau_l37test = exp(-secd(theta_i37).*A_l37test);
%tau_l37 = 1;


A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
```

```
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
```

```
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
```

```
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11)))./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case2] = [T_B24v T_B24h T_B37v T_B37h V W ...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];
```

```
else
    display('case 2 no values for abs(SST-Tv) greater than 20');
    DATA_case2 = [];
end


function [DATA_case3] = forward_model1_case3(DATA_48)

%Here V>48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_48];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
```

```
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
V_edr = DATA(:,12);
R = DATA(:,11);

% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%      T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%    T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;
```

```
% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);

% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);


%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
```

```
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
                T_v = 301.16;
%              else
%              T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
%finding values abs(SST-tv)<=20
fid_sv20 = find((abs(SST-T_v))<=20);
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
```

```
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
R = R(fid_sv20);

%                  if ((SST-T_v)<0)
% fid_pos = find((SST-T_v)>0);
% latitude = latitude(fid_pos);
% longitude = longitude(fid_pos);
% T_B24v = T_B24v(fid_pos);
% T_B24h = T_B24h(fid_pos);
% T_B37v = T_B37v(fid_pos);
% T_B37h = T_B37h(fid_pos);
% V = V(fid_pos);
% W = W(fid_pos);
% SST = SST(fid_pos);
% CLW = CLW(fid_pos);
% E_24v = E_24v(fid_pos);
```

```
% E_24h = E_24h(fid_pos);
% E_37v = E_37v(fid_pos);
% E_37h = E_37h(fid_pos);

% if (SST<301)
fid_300 = find(SST<301);
latitude = latitude(fid_300);
longitude = longitude(fid_300);
T_B24v = T_B24v(fid_300);
T_B24h = T_B24h(fid_300);
T_B37v = T_B37v(fid_300);
T_B37h = T_B37h(fid_300);
V = V(fid_300);
W = W(fid_300);
V_edr = V_edr(fid_300);
W_edr = W_edr(fid_300);
SST = SST(fid_300);
CLW = CLW(fid_300);
E_24v = E_24v(fid_300);
E_24h = E_24h(fid_300);
E_37v = E_37v(fid_300);
E_37h = E_37h(fid_300);
R = R(fid_300);

%find vapor less than 58
fid_58 = find(V<58);
```

```
latitude = latitude(fid_58);
longitude = longitude(fid_58);
T_B24v = T_B24v(fid_58);
T_B24h = T_B24h(fid_58);
T_B37v = T_B37v(fid_58);
T_B37h = T_B37h(fid_58);
V = V(fid_58);
W = W(fid_58);
V_edr = V_edr(fid_58);
W_edr = W_edr(fid_58);
SST = SST(fid_58);
CLW = CLW(fid_58);
E_24v = E_24v(fid_58);
E_24h = E_24h(fid_58);
E_37v = E_37v(fid_58);
E_37h = E_37h(fid_58);
R = R(fid_58);

%                     gui = -14;
%                  else
%                      gui = 14;
%                  end
%              else
gui = 1.05.*(SST - T_v).*(1 - ((SST-T_v).^2)./1200);
%              end
```

```
% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);
```

```
tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
%     H_l = 2.96;
% end
```

```
%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;

T_cl = (SST - (6.5).*H_l);


L = CLW;
A_l24test =0.0891.*(1-0.0281.*(T_cl-283)).*L;
tau_l24test = exp(-secd(theta_i24).*A_l24test);
%tau_l24 = 1;
A_l37test =0.2027.*(1-0.0261.*(T_cl-283)).*L;
tau_l37test = exp(-secd(theta_i37).*A_l37test);
%tau_l37 = 1;






delta_24v = (1-0.87158).*T_B24v - 29.706;
delta_24h = (1-0.86188).*T_B24h - 24.592;
delta_37v = (1-0.72493).*T_B37v - 59.993;
delta_37h = (1-0.80094).*T_B37h - 27.508;
```

```
% a_1 = T_U24;
% a_2 = E_24v.*SST;
% a_3 = 1-E_24v;
% a_4 = T_D24;
% a_5 = 2.7;
% a_6 = a_2+(a_3.*a_4);
% a_7 = (a_3.*a_5)-(a_3.*a_4);
% a_8 = a_6-a_1;
% a_11 = a_1+(T_B24v);
% a_10 = a_7.*((tau_v24).^2).*((tau_o24).^2);
% a_9 = a_8.*tau_v24.*tau_o24;
%
% b_1 = T_U24;
% b_2 = E_24h.*SST;
% b_3 = 1-E_24h;
% b_4 = T_D24;
% b_5 = 2.7;
% b_6 = b_2+(b_3.*b_4);
% b_7 = (b_3.*b_5)-(b_3.*b_4);
% b_8 = b_6-b_1;
% b_11 = b_1+(T_B24h);
% b_10 = b_7.*((tau_v24).^2).*((tau_o24).^2);
% b_9 = b_8.*tau_v24.*tau_o24;
%
% c_1 = T_U37;
% c_2 = E_37v.*SST;
```

```
% c_3 = 1-E_37v;
% c_4 = T_D37;
% c_5 = 2.7;
% c_6 = c_2+(c_3.*c_4);
% c_7 = (c_3.*c_5)-(c_3.*c_4);
% c_8 = c_6-c_1;
% c_11 = c_1+(T_B37v);
% c_10 = c_7.*((tau_v37).^2).*((tau_o37).^2);
% c_9 = c_8.*tau_v37.*tau_o37;
%
% d_1 = T_U37;
% d_2 = E_37h.*SST;
% d_3 = 1-E_37h;
% d_4 = T_D37;
% d_5 = 2.7;
% d_6 = d_2+(d_3.*d_4);
% d_7 = (d_3.*d_5)-(d_3.*d_4);
% d_8 = d_6-d_1;
% d_11 = d_1+(T_B37h);
% d_10 = d_7.*((tau_v37).^2).*((tau_o37).^2);
% d_9 = d_8.*tau_v37.*tau_o37;
%
% sol1_24 = (-a_9+sqrt(((a_9).^2) -
(4*(a_10).*(a_11)))./(2.*a_10);
% sol2_24 = (-a_9-sqrt(((a_9).^2) -
(4*(a_10).*(a_11)))./(2.*a_10);
```

```
% sol3_24 = (-b_9+sqrt(((b_9).^2) -
(4*(b_10).*(b_11)))))./(2.*b_10);
% sol4_24 = (-b_9-sqrt(((b_9).^2) -
(4*(b_10).*(b_11)))))./(2.*b_10);
%A_1 = T_U24+delta_24v;
A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

%A_2 = T_U24+delta_24h;
A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

%A_3 = T_U37+delta_37v;
A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
```

```
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

%A_4 = T_U37+delta_37h;
A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-(T_B24v-
delta_24v))).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-(T_B24v-
delta_24v))).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-(T_B24h-
delta_24h))).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-(T_B24h-
delta_24h))).*(C_2))))./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;
```

```
dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-(T_B37v-
delta_37v))).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-(T_B37v-
delta_37v))).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-(T_B37h-
delta_37h))).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-(T_B37h-
delta_37h))).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
```

```
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case3] = [T_B24v T_B24h T_B37v T_B37h V W ...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];


function [DATA_case4] = forward_model1_case4(DATA_47)

%Here V<=48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);
```

```
% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_47];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%      T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%     T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;
```

```
salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);

% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);
```

```
%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
%                 T_v = 301.16;
%              else
             T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
fid_sv20 = find((abs(SST-T_v))>20);
if fid_sv20>0;
```

```
T_v = T_v(fid_sv20);
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
R = R(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
%                  if ((SST-T_v)<0)
fid_neg = find((SST-T_v)<0);
test = isempty(fid_neg);
if test==0;
latitude = latitude(fid_neg);
longitude = longitude(fid_neg);
T_B24v = T_B24v(fid_neg);
T_B24h = T_B24h(fid_neg);
```

```
T_B37v = T_B37v(fid_neg);
T_B37h = T_B37h(fid_neg);
V = V(fid_neg);
W = W(fid_neg);
V_edr = V_edr(fid_neg);
W_edr = W_edr(fid_neg);
SST = SST(fid_neg);
CLW = CLW(fid_neg);
E_24v = E_24v(fid_neg);
E_24h = E_24h(fid_neg);
E_37v = E_37v(fid_neg);
E_37h = E_37h(fid_neg);
R = R(fid_neg);

% if (SST<301)
fid_300 = find(SST<301);
latitude = latitude(fid_300);
longitude = longitude(fid_300);
T_B24v = T_B24v(fid_300);
T_B24h = T_B24h(fid_300);
T_B37v = T_B37v(fid_300);
T_B37h = T_B37h(fid_300);
V = V(fid_300);
W = W(fid_300);
V_edr = V_edr(fid_300);
W_edr = W_edr(fid_300);
```

```
SST = SST(fid_300);

CLW = CLW(fid_300);

E_24v = E_24v(fid_300);

E_24h = E_24h(fid_300);

E_37v = E_37v(fid_300);

E_37h = E_37h(fid_300);

R = R(fid_300);

                    gui = -14;

%                  else

%                      gui = 14;

%                  end

%              else

%                  gui = 1.05.*(SST - T_v).*(1 - ((SST-
T_v).^2)./1200);

%              end


% for l = 1:2;

T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...

    b(6,1).*gui;

T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...

    b(6,2).*gui;

%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...

%         b(6,l).*gui;
```

```
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;
```

```
%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
%     H_l = 2.96;
% end



%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;

T_cl = (SST - (6.5).*H_l);



L = CLW;
A_l24test =0.0891.*(1-0.0281.*(T_cl-283)).*L;
```

```
tau_l24test = exp(-secd(theta_i24).*A_l24test);
%tau_l24 = 1;
A_l37test =0.2027.*(1-0.0261.*(T_cl-283)).*L;
tau_l37test = exp(-secd(theta_i37).*A_l37test);


A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
```

```
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1)))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1)))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2)))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2)))))./(2.*C_2);



tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
```

```
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;
```

```
dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case4] = [T_B24v T_B24h T_B37v T_B37h V W ...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];



else
    display('case 4 no negative values for (SST-Tv)');
    DATA_case4 = [];
end

else
    display('case 4 no values for abs(SST-Tv) greater than 20');
    DATA_case4 = [];
end
```

```
function [DATA_case5] = forward_model1_case5(DATA_47)

%Here V>48, abs(SST-TV)>20, (SST-TV)>0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_47];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
```

```
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%     T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%    T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);
```

```
% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);


%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
```

```
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
%                 T_v = 301.16;
%              else
T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
fid_sv20 = find((abs(SST-T_v))>20);
if fid_sv20>0;
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
```

```
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
T_v = T_v(fid_sv20);
R = R(fid_sv20);
%                  if ((SST-T_v)<0)
fid_pos = find((SST-T_v)>0);
latitude = latitude(fid_pos);
longitude = longitude(fid_pos);
T_B24v = T_B24v(fid_pos);
T_B24h = T_B24h(fid_pos);
T_B37v = T_B37v(fid_pos);
T_B37h = T_B37h(fid_pos);
V = V(fid_pos);
W = W(fid_pos);
V_edr = V_edr(fid_pos);
W_edr = W_edr(fid_pos);
SST = SST(fid_pos);
CLW = CLW(fid_pos);
E_24v = E_24v(fid_pos);
E_24h = E_24h(fid_pos);
E_37v = E_37v(fid_pos);
E_37h = E_37h(fid_pos);
T_v = T_v(fid_pos);
R = R(fid_pos);
```

```
% if (SST<301)
fid_300 = find(SST<301);
latitude = latitude(fid_300);
longitude = longitude(fid_300);
T_B24v = T_B24v(fid_300);
T_B24h = T_B24h(fid_300);
T_B37v = T_B37v(fid_300);
T_B37h = T_B37h(fid_300);
V = V(fid_300);
W = W(fid_300);
V_edr = V_edr(fid_300);
W_edr = W_edr(fid_300);
SST = SST(fid_300);
CLW = CLW(fid_300);
E_24v = E_24v(fid_300);
E_24h = E_24h(fid_300);
E_37v = E_37v(fid_300);
E_37h = E_37h(fid_300);
R = R(fid_300);
%                    gui = -14;
%                 else
                     gui = 14;
%                 end
%             else
%                 gui = 1.05.*(SST - T_v).*(1 - ((SST-
T_v).^2)./1200);
```

```
%               end

% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;
```

```
tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
%     H_l = 2.96;
% end
```

```
%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;

T_cl = (SST - (6.5).*H_l);


L = CLW;
A_l24test =0.0891.*(1-0.0281.*(T_cl-283)).*L;
tau_l24test = exp(-secd(theta_i24).*A_l24test);
%tau_l24 = 1;
A_l37test =0.2027.*(1-0.0261.*(T_cl-283)).*L;
tau_l37test = exp(-secd(theta_i37).*A_l37test);
%tau_l37 = 1;


A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));
```

```
A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
```

```
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
```

```
sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);



[DATA_case5] = [T_B24v T_B24h T_B37v T_B37h V W ...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];
```

```
else
    display('case 5 no values for abs(SST-Tv) greater than 20');
    DATA_case5 = [];
end


function [DATA_case6] = forward_model1_case6(DATA_47)

%Here V>48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_47];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
```

```
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
R = DATA(:,11);
V = DATA(:,5);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%     T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%    T_B37v T_B37h V SST]);

W = DATA(:,6);
V_edr = DATA(:,12);
W_edr = DATA(:,13);
clear('DATA','DATA_47');

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;



% size_w = size(W,1);
% size_v = size(V,1);
```

```
[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);

% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);


%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
```

```
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];




%             if (V>48)
%                 T_v = 301.16;
%              else
T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
%finding values abs(SST-tv)<=20
fid_sv20 = find((abs(SST-T_v))<=20);
T_v = T_v(fid_sv20);
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
```

```
V = V(fid_sv20);
V_edr = V_edr(fid_sv20);
W = W(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
R = R(fid_sv20);
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
clear('fid_sv20');
%                 if ((SST-T_v)<0)
% fid_pos = find((SST-T_v)>0);
% latitude = latitude(fid_pos);
% longitude = longitude(fid_pos);
% T_B24v = T_B24v(fid_pos);
% T_B24h = T_B24h(fid_pos);
% T_B37v = T_B37v(fid_pos);
% T_B37h = T_B37h(fid_pos);
% V = V(fid_pos);
% W = W(fid_pos);
% SST = SST(fid_pos);
% CLW = CLW(fid_pos);
% E_24v = E_24v(fid_pos);
% E_24h = E_24h(fid_pos);
```

```
% E_37v = E_37v(fid_pos);
% E_37h = E_37h(fid_pos);

% if (SST<301)
fid_300 = find(SST<301);
T_v = T_v(fid_300);
latitude = latitude(fid_300);
longitude = longitude(fid_300);
T_B24v = T_B24v(fid_300);
T_B24h = T_B24h(fid_300);
T_B37v = T_B37v(fid_300);
T_B37h = T_B37h(fid_300);
V = V(fid_300);
V_edr = V_edr(fid_300);
W = W(fid_300);
W_edr = W_edr(fid_300);
SST = SST(fid_300);
CLW = CLW(fid_300);
R = R(fid_300);
E_24v = E_24v(fid_300);
E_24h = E_24h(fid_300);
E_37v = E_37v(fid_300);
E_37h = E_37h(fid_300);
clear('fid_300');
%                    gui = -14;
%                 else
```

```
%                     gui = 14;
%                 end
%             else
gui = 1.05.*(SST - T_v).*(1 - ((SST-T_v).^2)./1200);
%             end

% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
```

```
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)
```

```
    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
%     H_l = 2.96;
% end


%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;

T_cl = (SST - (6.5).*H_l);


L = CLW;

A_r24test = 0.0188.*(1+0.0020.*(T_l-283)).*H_l.*((R).^(1.0220));
%L = 0.18.*(1+sqrt((H_l.*R)));
A_l24test =0.0891.*(1-0.0281.*(T_cl-283)).*L;
%tau_l24test = exp(-secd(theta_i24).*(A_l24test+A_r24test));
tau_l24test = exp(-secd(theta_i24).*(A_l24test));
%tau_l24 = 1;
A_l37test =0.2027.*(1-0.0261.*(T_cl-283)).*L;
A_r37test = 0.0425.*(1-0.0020.*(T_l-283)).*H_l.*((R).^(0.9546));
%tau_l37test = exp(-secd(theta_i37).*(A_l37test+A_r37test));
tau_l37test = exp(-secd(theta_i37).*(A_l37test));
```

```
%tau_l37 = 1;



T_BC = 2.7;


delta_24v = (1.0017-1).*T_B24v - 2.3431;

delta_24h = (1.0006-1).*T_B24h - 5.7077;

delta_37v = (1-0.90003).*T_B37v - 19.483;

delta_37h = (1-0.88289).*T_B37h - 12.389;


% % a_1 = T_U24;

% % a_2 = E_24v.*SST;

% % a_3 = 1-E_24v;

% % a_4 = T_D24;

% % a_5 = 2.7;

% % a_6 = a_2+(a_3.*a_4);

% % a_7 = (a_3.*a_5)-(a_3.*a_4);

% % a_8 = a_6-a_1;

% % a_11 = a_1+(T_B24v);

% % a_10 = a_7.*((tau_v24).^2).*((tau_o24).^2);

% % a_9 = a_8.*tau_v24.*tau_o24;

% %

% % b_1 = T_U24;

% % b_2 = E_24h.*SST;

% % b_3 = 1-E_24h;

% % b_4 = T_D24;
```

```
% % b_5 = 2.7;
% % b_6 = b_2+(b_3.*b_4);
% % b_7 = (b_3.*b_5)-(b_3.*b_4);
% % b_8 = b_6-b_1;
% % b_11 = b_1+(T_B24h);
% % b_10 = b_7.*((tau_v24).^2).*((tau_o24).^2);
% % b_9 = b_8.*tau_v24.*tau_o24;
% %
% % c_1 = T_U37;
% % c_2 = E_37v.*SST;
% % c_3 = 1-E_37v;
% % c_4 = T_D37;
% % c_5 = 2.7;
% % c_6 = c_2+(c_3.*c_4);
% % c_7 = (c_3.*c_5)-(c_3.*c_4);
% % c_8 = c_6-c_1;
% % c_11 = c_1+(T_B37v);
% % c_10 = c_7.*((tau_v37).^2).*((tau_o37).^2);
% % c_9 = c_8.*tau_v37.*tau_o37;
% %
% % d_1 = T_U37;
% % d_2 = E_37h.*SST;
% % d_3 = 1-E_37h;
% % d_4 = T_D37;
% % d_5 = 2.7;
% % d_6 = d_2+(d_3.*d_4);
```

```
% % d_7 = (d_3.*d_5)-(d_3.*d_4);
% % d_8 = d_6-d_1;
% % d_11 = d_1+(T_B37h);
% % d_10 = d_7.*((tau_v37).^2).*((tau_o37).^2);
% % d_9 = d_8.*tau_v37.*tau_o37;

%A_1 = T_U24-delta_24v;
A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

%A_2 = T_U24-delta_24h;
A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

%A_3 = T_U37+delta_37v;
A_3 = T_U37;
```

```
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

%A_4 = T_U37+delta_37h;
A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
(T_B24v+delta_24v))).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
(T_B24v+delta_24v))).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
(T_B24h+delta_24h))).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
(T_B24h+delta_24h))).*(C_2))))./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;
```

```
dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);


sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-(T_B37v-
delta_37v))).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-(T_B37v-
delta_37v))).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-(T_B37h-
delta_37h))).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-(T_B37h-
delta_37h))).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol2_37)./2;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case6] = [T_B24v T_B24h T_B37v T_B37h V W ...
```

```
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];




function [DATA_case7] = forward_model1_case7(DATA_48)

%Here V>48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_48];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
```

```
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%      T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%     T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);
```

```
[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);

% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);


%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
```

```
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
                T_v = 301.16;
%              else
%              T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
fid_sv20 = find((abs(SST-T_v))>20);
if fid_sv20>0;
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
```

```
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
R = R(fid_sv20);
%                  if ((SST-T_v)<0)
fid_neg = find((SST-T_v)<0);
latitude = latitude(fid_neg);
longitude = longitude(fid_neg);
T_B24v = T_B24v(fid_neg);
T_B24h = T_B24h(fid_neg);
T_B37v = T_B37v(fid_neg);
T_B37h = T_B37h(fid_neg);
V = V(fid_neg);
W = W(fid_neg);
V_edr = V_edr(fid_neg);
W_edr = W_edr(fid_neg);
SST = SST(fid_neg);
CLW = CLW(fid_neg);
E_24v = E_24v(fid_neg);
E_24h = E_24h(fid_neg);
```

```
E_37v = E_37v(fid_neg);
E_37h = E_37h(fid_neg);
R = R(fid_neg);

% if (SST>301)
fid_301 = find(SST>301);
latitude = latitude(fid_301);
longitude = longitude(fid_301);
T_B24v = T_B24v(fid_301);
T_B24h = T_B24h(fid_301);
T_B37v = T_B37v(fid_301);
T_B37h = T_B37h(fid_301);
V = V(fid_301);
W = W(fid_301);
V_edr = V_edr(fid_301);
W_edr = W_edr(fid_301);
SST = SST(fid_301);
CLW = CLW(fid_301);
E_24v = E_24v(fid_301);
E_24h = E_24h(fid_301);
E_37v = E_37v(fid_301);
E_37h = E_37h(fid_301);
R = R(fid_301);
                    gui = -14;
%                 else
%                     gui = 14;
```

```
%                  end
%              else
%                  gui = 1.05.*(SST - T_v).*(1 - ((SST-
T_v).^2)./1200);
%              end

% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%      T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%          b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%      T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%      A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
```

```
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)
```

```
%    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
H_l = 2.96.*ones(size(SST,1));
% end


%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;


%tau_l37 = 1;


A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
```

```
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);
```

```
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2)))) ./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11)))) ./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11)))) ./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11)))) ./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11)))) ./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3)))) ./(2.*C_3);
```

```
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case7] = [T_B24v T_B24h T_B37v T_B37h V W...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];

else
    display('case 7 no values for abs(SST-Tv) greater than 20');
    DATA_case7 = [];
end
```

```
function [DATA_case8] = forward_model1_case8(DATA_48)

%Here V>48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_48];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
```

```
V = DATA(:,5);
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%     T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%    T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);
```

```
% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);



%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
```

```
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
                T_v = 301.16;
%              else
%              T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
fid_sv20 = find((abs(SST-T_v))>20);
if fid_sv20>0;
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
```

```
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
R = R(fid_sv20);
%                  if ((SST-T_v)<0)
fid_pos = find((SST-T_v)>0);
latitude = latitude(fid_pos);
longitude = longitude(fid_pos);
T_B24v = T_B24v(fid_pos);
T_B24h = T_B24h(fid_pos);
T_B37v = T_B37v(fid_pos);
T_B37h = T_B37h(fid_pos);
V = V(fid_pos);
W = W(fid_pos);
V_edr = V_edr(fid_pos);
W_edr = W_edr(fid_pos);
SST = SST(fid_pos);
CLW = CLW(fid_pos);
E_24v = E_24v(fid_pos);
E_24h = E_24h(fid_pos);
E_37v = E_37v(fid_pos);
E_37h = E_37h(fid_pos);
R = R(fid_pos);

% if (SST<301)
fid_301 = find(SST>=301);
```

```
latitude = latitude(fid_301);

longitude = longitude(fid_301);

T_B24v = T_B24v(fid_301);

T_B24h = T_B24h(fid_301);

T_B37v = T_B37v(fid_301);

T_B37h = T_B37h(fid_301);

V = V(fid_301);

W = W(fid_301);

V_edr = V_edr(fid_301);

W_edr = W_edr(fid_301);

SST = SST(fid_301);

CLW = CLW(fid_301);

E_24v = E_24v(fid_301);

E_24h = E_24h(fid_301);

E_37v = E_37v(fid_301);

E_37h = E_37h(fid_301);

R = R(fid_301);

%                     gui = -14;

%                  else

                      gui = 14;

%                  end

%              else

%                  gui = 1.05.*(SST - T_v).*(1 - ((SST-
T_v).^2)./1200);

%              end
```

```
% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);
```

```
tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

%    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
H_l = 2.96.*ones(size(SST,1));
% end
```

```
%finding Temperature of rain
T_l = (SST+273)./2;




A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
```

```
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
```

```
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11)))) ./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11)))) ./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11)))) ./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11)))) ./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3)))) ./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3)))) ./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4)))) ./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4)))) ./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;
```

```
dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case8] = [T_B24v T_B24h T_B37v T_B37h V W ...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];

else
    display('case 8 no values for abs(SST-Tv) greater than 20');
    DATA_case8 = [];
end


function [DATA_case9] = forward_model1_case9(DATA_48)

%Here V>48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
```

```
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_48];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%      T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%     T_B37v T_B37h V SST]);

W = DATA(:,6);
```

```
W_edr = DATA(:,13);

clear('DATA','DATA_48');
%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);

% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
```

```
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);



%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
                T_v = 301.16;
%              else
%              T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end
```

```
%               if (abs(SST-T_v)>20)
%finding values abs(SST-tv)<=20
fid_sv20 = find((abs(SST-T_v))<=20);
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
R = R(fid_sv20);
clear('fid_sv20');
%                   if ((SST-T_v)<0)
% fid_pos = find((SST-T_v)>0);
% latitude = latitude(fid_pos);
% longitude = longitude(fid_pos);
```

```
% T_B24v = T_B24v(fid_pos);
% T_B24h = T_B24h(fid_pos);
% T_B37v = T_B37v(fid_pos);
% T_B37h = T_B37h(fid_pos);
% V = V(fid_pos);
% W = W(fid_pos);
% SST = SST(fid_pos);
% CLW = CLW(fid_pos);
% E_24v = E_24v(fid_pos);
% E_24h = E_24h(fid_pos);
% E_37v = E_37v(fid_pos);
% E_37h = E_37h(fid_pos);

% if (SST>301)
fid_301 = find(SST>=301);
latitude = latitude(fid_301);
longitude = longitude(fid_301);
T_B24v = T_B24v(fid_301);
T_B24h = T_B24h(fid_301);
T_B37v = T_B37v(fid_301);
T_B37h = T_B37h(fid_301);
V = V(fid_301);
W = W(fid_301);
V_edr = V_edr(fid_301);
W_edr = W_edr(fid_301);
SST = SST(fid_301);
```

```
CLW = CLW(fid_301);
E_24v = E_24v(fid_301);
E_24h = E_24h(fid_301);
E_37v = E_37v(fid_301);
E_37h = E_37h(fid_301);
R = R(fid_301);
clear('fid_301');
%                     gui = -14;
%                  else
%                      gui = 14;
%                  end
%              else
gui = 1.05.*(SST - T_v).*(1 - ((SST-T_v).^2)./1200);
%              end

% % % % % sz_td = size(R,1);
% % % % % mat_1 = ones(sz_td,1);
% % % % % T_D24 = 288.*mat_1;
% % % % % T_D37 = 288.*mat_1;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
```

```
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;

% % % % % T_U24 = T_D24 -2;
% % % % % T_U37 = T_D37 -2;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
```

```
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

%    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
H_l = 2.96.*ones(size(SST,1));
% end



%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;
```

```
T_BC = 2.7;

% delta_24v = 0;
% delta_24h = 0;
% delta_37v = 0;
% delta_37h = 0;

delta_24v = (1-0.90746).*T_B24v - 21.092;
delta_24h = (1-0.88467).*T_B24h - 20.451;
delta_37v = (1-0.72396).*T_B37v - 60.733;
delta_37h = (1-0.79).*T_B37h - 30.135;

% a_1 = T_U24;
% a_2 = -tau_v24.*tau_o24.*T_U24;
% a_3 = tau_v24.*tau_o24;
% a_4 = E_24v.*SST;
% a_5 = 1-E_24v;
% a_6 = T_D24;
% a_7 = -tau_v24.*tau_o24.*T_D24;
% a_8 = 2.73.*(tau_v24.*tau_o24);
% a_9 = a_5.*a_6;
% a_10 = a_5.*a_7;
% a_11 = a_5.*a_8;
% a_12 = a_4+a_9;
% a_13 = a_10+a_11;
```

```
% a_14 = a_2 + a_3.*a_12;
% a_15 = a_1 - (T_B24v+delta_24v);
% a_16 = a_3.*a_13;
%
% b_1 = T_U24;
% b_2 = -tau_v24.*tau_o24.*T_U24;
% b_3 = tau_v24.*tau_o24;
% b_4 = E_24h.*SST;
% b_5 = 1-E_24h;
% b_6 = T_D24;
% b_7 = -tau_v24.*tau_o24.*T_D24;
% b_8 = 2.73.*(tau_v24.*tau_o24);
% b_9 = b_5.*b_6;
% b_10 = b_5.*b_7;
% b_11 = b_5.*b_8;
% b_12 = b_4+b_9;
% b_13 = b_10+b_11;
% b_14 = b_2 + b_3.*b_12;
% b_15 = b_1 - (T_B24h+delta_24h);
% b_16 = b_3.*b_13;
%
% c_1 = T_U37;
% c_2 = -tau_v37.*tau_o37.*T_U37;
% c_3 = tau_v37.*tau_o37;
% c_4 = E_37v.*SST;
% c_5 = 1-E_37v;
```

```
% c_6 = T_D37;
% c_7 = -tau_v37.*tau_o37.*T_D37;
% c_8 = 2.73.*(tau_v37.*tau_o37);
% c_9 = c_5.*c_6;
% c_10 = c_5.*c_7;
% c_11 = c_5.*c_8;
% c_12 = c_4+c_9;
% c_13 = c_10+c_11;
% c_14 = c_2 + c_3.*c_12;
% c_15 = c_1 - (T_B37v+delta_37v);
% c_16 = c_3.*c_13;
%
%
% d_1 = T_U37;
% d_2 = -tau_v37.*tau_o37.*T_U37;
% d_3 = tau_v37.*tau_o37;
% d_4 = E_37h.*SST;
% d_5 = 1-E_37h;
% d_6 = T_D37;
% d_7 = -tau_v37.*tau_o37.*T_D37;
% d_8 = 2.73.*(tau_v37.*tau_o37);
% d_9 = d_5.*d_6;
% d_10 = d_5.*d_7;
% d_11 = d_5.*d_8;
% d_12 = d_4+d_9;
% d_13 = d_10+d_11;
```

```
% d_14 = d_2 + d_3.*d_12;
% d_15 = d_1 - (T_B37h+delta_37h);
% d_16 = d_3.*d_13;
%
% sol1_24 = (-a_14+sqrt((a_14).^2 -4*(a_16).*(a_15)))./(2.*a_15);
% % sol2_24 = (-a_14-sqrt((a_14).^2 -
4*(a_16).*(a_15)))./(2.*a_15);
% sol3_24 = (-b_14+sqrt((b_14).^2 -4*(b_16).*(b_15)))./(2.*b_15);
% % sol4_24 = (-b_14-sqrt((b_14).^2 -
4*(b_16).*(b_15)))./(2.*b_15);
%
% tau_l24 = (sol1_24+sol3_24)./2;


% dif1_24 = abs(sol1_24-sol3_24);
% dif2_24 = abs(sol1_24-sol4_24);
% dif3_24 = abs(sol2_24-sol3_24);
% dif4_24 = abs(sol2_24-sol4_24);


%
% sol1_37 = (-c_14+sqrt((c_14).^2 -4*(c_16).*(c_15)))./(2.*c_15);
% % sol2_37 = (-c_14-sqrt((c_14).^2 -
4*(c_16).*(c_15)))./(2.*c_15);
% sol3_37 = (-d_14+sqrt((d_14).^2 -4*(d_16).*(d_15)))./(2.*d_15);
% % sol4_37 = (-d_14-sqrt((d_14).^2 -
4*(d_16).*(d_15)))./(2.*d_15);
```

```
%
% tau_l37 = (sol1_37+sol3_37)./2;

% dif1_37 = abs(sol1_37-sol3_37);
% dif2_37 = abs(sol1_37-sol4_37);
% dif3_37 = abs(sol2_37-sol3_37);
% dif4_37 = abs(sol2_37-sol4_37);


% [DATA_case9] = [T_B24v T_B24h T_B37v T_B37h V W ...
%     SST CLW latitude longitude...
%     tau_l24 tau_l37 tau_l24test tau_l37test];


%A_1 = T_U24+delta_24v;
A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

%A_2 = T_U24+delta_24h;
A_2 = T_U24;
```

```
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

%A_3 = T_U37+delta_37v;
A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

%A_4 = T_U37+delta_37h;
A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-(T_B24v-
delta_24v))).*(C_1)))))./(2.*C_1);
```

```
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-(T_B24v-
delta_24v))).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-(T_B24h-
delta_24h))).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-(T_B24h-
delta_24h))).*(C_2))))./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
```

```
sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-(T_B37v-
delta_37v))).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-(T_B37v-
delta_37v))).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-(T_B37h-
delta_37h))).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-(T_B37h-
delta_37h))).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case9] = [T_B24v T_B24h T_B37v T_B37h V W...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];
```

```
function [DATA_case10] = forward_model1_case10(DATA_47)

%Here V<=48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_47];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
```

```
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%      T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%     T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);
```

```
% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);



%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
```

```
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
%                 T_v = 301.16;
%              else
             T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
fid_sv20 = find((abs(SST-T_v))>20);
if fid_sv20>0;
T_v = T_v(fid_sv20);
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
```

```
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
R = R(fid_sv20);
%                  if ((SST-T_v)<0)
fid_neg = find((SST-T_v)<0);
test = isempty(fid_neg);
if test==0;
T_v = T_v(fid_neg);
latitude = latitude(fid_neg);
longitude = longitude(fid_neg);
T_B24v = T_B24v(fid_neg);
T_B24h = T_B24h(fid_neg);
T_B37v = T_B37v(fid_neg);
T_B37h = T_B37h(fid_neg);
V = V(fid_neg);
W = W(fid_neg);
V_edr = V_edr(fid_neg);
W_edr = W_edr(fid_neg);
SST = SST(fid_neg);
CLW = CLW(fid_neg);
E_24v = E_24v(fid_neg);
E_24h = E_24h(fid_neg);
E_37v = E_37v(fid_neg);
E_37h = E_37h(fid_neg);
```

```
R = R(fid_neg);

% if (SST<301)
fid_301 = find(SST>=301);
latitude = latitude(fid_301);
longitude = longitude(fid_301);
T_B24v = T_B24v(fid_301);
T_B24h = T_B24h(fid_301);
T_B37v = T_B37v(fid_301);
T_B37h = T_B37h(fid_301);
V = V(fid_301);
W = W(fid_301);
V_edr = V_edr(fid_301);
W_edr = W_edr(fid_301);
SST = SST(fid_301);
CLW = CLW(fid_301);
E_24v = E_24v(fid_301);
E_24h = E_24h(fid_301);
E_37v = E_37v(fid_301);
E_37h = E_37h(fid_301);
R = R(fid_301);
                    gui = -14;
%                 else
%                     gui = 14;
%                 end
%             else
```

```
%                  gui = 1.05.*(SST - T_v).*(1 - ((SST-
T_v).^2)./1200);
%              end

% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;
```

```
%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

%    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
```

```
H_l = 2.96.*ones(size(SST,1));
% end


%finding Temperature of rain
T_l = (SST+273)./2;



A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

A_3 = T_U37;
```

```
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;
```

```
%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);
```

```
tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case10] = [T_B24v T_B24h T_B37v T_B37h V W ...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];

else
    display('case 10 no negative values for (SST-Tv)');
    DATA_case10 = [];
end

else
    display('case 10 no values for abs(SST-Tv) greater than 20');
    DATA_case10 = [];
end
```

```
function [DATA_case11] = forward_model1_case11(DATA_47)

%Here V>48, abs(SST-TV)>20, (SST-TV)>0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_47];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
```

```
V_edr = DATA(:,12);
R = DATA(:,11);
% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%     T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%    T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);
```

```
% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);



%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
```

```
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
%                 T_v = 301.16;
%              else
T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
fid_sv20 = find((abs(SST-T_v))>20);
if fid_sv20>0;
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
```

```
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
T_v = T_v(fid_sv20);
R = R(fid_sv20);
%                  if ((SST-T_v)<0)
fid_pos = find((SST-T_v)>0);
latitude = latitude(fid_pos);
longitude = longitude(fid_pos);
T_B24v = T_B24v(fid_pos);
T_B24h = T_B24h(fid_pos);
T_B37v = T_B37v(fid_pos);
T_B37h = T_B37h(fid_pos);
V = V(fid_pos);
W = W(fid_pos);
V_edr = V_edr(fid_pos);
W_edr = W_edr(fid_pos);
SST = SST(fid_pos);
CLW = CLW(fid_pos);
E_24v = E_24v(fid_pos);
E_24h = E_24h(fid_pos);
E_37v = E_37v(fid_pos);
E_37h = E_37h(fid_pos);
T_v = T_v(fid_pos);
R = R(fid_pos);
```

```
% if (SST<301)
fid_301 = find(SST>=301);
test = isempty(fid_301);
if test==0;
latitude = latitude(fid_301);
longitude = longitude(fid_301);
T_B24v = T_B24v(fid_301);
T_B24h = T_B24h(fid_301);
T_B37v = T_B37v(fid_301);
T_B37h = T_B37h(fid_301);
V = V(fid_301);
W = W(fid_301);
V_edr = V_edr(fid_301);
W_edr = W_edr(fid_301);
SST = SST(fid_301);
CLW = CLW(fid_301);
E_24v = E_24v(fid_301);
E_24h = E_24h(fid_301);
E_37v = E_37v(fid_301);
E_37h = E_37h(fid_301);
R = R(fid_301);
%                   gui = -14;
%                else
                    gui = 14;
%                end
%            else
```

```
%                  gui = 1.05.*(SST - T_v).*(1 - ((SST-
T_v).^2)./1200);
%              end

% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;
```

```
%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

%    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
```

```
     H_l = 2.96.*ones(size(SST,1));
% end


%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;



A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));
```

```
A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);
```

```
tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);
```

```
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4)))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case11] = [T_B24v T_B24h T_B37v T_B37h V W ...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];

else
    display('case 11 no values of SST greater or equal to 301K
when abs(SST-Tv) greater than 20');
    DATA_case11 = [];
end
else
    display('case 11 no values for abs(SST-Tv) greater than 20');
    DATA_case11 = [];
```

```
end


function [DATA_case12] = forward_model1_case12(DATA_47)


[DATA] = [DATA_47];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
V = DATA(:,5);
W = DATA(:,6);
V_edr = DATA(:,12);
W_edr = DATA(:,13);
R = DATA(:,11);


salinity = 34;
```

```
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);

% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);
```

```
%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
%                 T_v = 301.16;
%              else
T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
%finding values abs(SST-tv)<=20
fid_sv20 = find((abs(SST-T_v))<=20);
T_v = T_v(fid_sv20);
latitude = latitude(fid_sv20);
```

```
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
R = R(fid_sv20);
%                  if ((SST-T_v)<0)
% fid_pos = find((SST-T_v)>0);
% latitude = latitude(fid_pos);
% longitude = longitude(fid_pos);
% T_B24v = T_B24v(fid_pos);
% T_B24h = T_B24h(fid_pos);
% T_B37v = T_B37v(fid_pos);
% T_B37h = T_B37h(fid_pos);
% V = V(fid_pos);
% W = W(fid_pos);
```

```
% SST = SST(fid_pos);
% CLW = CLW(fid_pos);
% E_24v = E_24v(fid_pos);
% E_24h = E_24h(fid_pos);
% E_37v = E_37v(fid_pos);
% E_37h = E_37h(fid_pos);

% if (SST<301)
fid_301 = find(SST>=301);
T_v = T_v(fid_301);
latitude = latitude(fid_301);
longitude = longitude(fid_301);
T_B24v = T_B24v(fid_301);
T_B24h = T_B24h(fid_301);
T_B37v = T_B37v(fid_301);
T_B37h = T_B37h(fid_301);
V = V(fid_301);
W = W(fid_301);
V_edr = V_edr(fid_301);
W_edr = W_edr(fid_301);
SST = SST(fid_301);
CLW = CLW(fid_301);
E_24v = E_24v(fid_301);
E_24h = E_24h(fid_301);
E_37v = E_37v(fid_301);
E_37h = E_37h(fid_301);
```

```
R = R(fid_301);
%                     gui = -14;
%                  else
%                      gui = 14;
%                  end
%              else
gui = 1.05.*(SST - T_v).*(1 - ((SST-T_v).^2)./1200);
%              end

% for l = 1:2;
T_D24 = b(1,1) + b(2,1).*V + b(3,1).*V.^2 + b(4,1).*V.^3 +
b(5,1).*V.^4 + ...
    b(6,1).*gui;
T_D37 = b(1,2) + b(2,2).*V + b(3,2).*V.^2 + b(4,2).*V.^3 +
b(5,2).*V.^4 + ...
    b(6,2).*gui;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
%         b(6,l).*gui;
T_U24 = T_D24 + b(7,1) + b(8,1).*V;
T_U37 = T_D37 + b(7,2) + b(8,2).*V;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);
```

```
%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;

%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain
```

```
% if (SST<301)

%    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
H_l = 2.96.*ones(size(SST,1));
% end


%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;

%tau_l37 = 1;

% % % % % %defining the waiting function for the forward model
% % % % % x = (A_l(1,2)-0.04)/0.11;
% % % % % if (x<0)
% % % % %     alpha = 0;
% % % % % elseif (x>1)
% % % % %     alpha = 1;
% % % % % else
% % % % %     alpha = (3*(x)^2)-(2*(x)^3);
% % % % % end
%W_o = 6.5;
```

```
%defining the forward model for each frequency and polarization
%F will be a 4 by 1 matrix. T_BU is a 1 by 2 matrix. E is a
%2 x 2 matrix, where the first row is h-pol, and first column
%is 23.8GHz

T_BC = 2.7;

% % % for l=1:2;
% % %     %f = [23.8 37];
% % %     f = [23.8 36.5]; %this has to be in GHz
% % %     if (f(1,l)<37)
% % %         delta_S_sq(1,l) = 5.22.*(10.^-3) *(1 -
0.00748.*(37-f(1,l)).^(1.3)).*W;
% % %     else
% % %         delta_S_sq(1,l) = 5.22.*(10.^-3).*W;
% % %     end
% % %     omega_v(1,l) = (2.5+0.018.*(37-
f(1,l))).*(delta_S_sq(1,l) -...
% % %         70.*(delta_S_sq(1,l)).^3)*tau_atm(1,l).^3.4;
% % %     omega_h(1,l) = (6.2-0.001.*(37-
f(1,l)).^2).*(delta_S_sq(1,l) -...
% % %         70.*(delta_S_sq(1,l)).^3).*tau_atm(1,l).^2;
% % % end
% for l=1:2;

%tau_l24 = 0.1:0.05:1;
```

```
% % % % T_BU24 = T_U24.*(1-(tau_v24.*tau_o24.*tau_l24));
% % % % T_BU37 = T_U37.*(1-(tau_v37.*tau_o37.*tau_l37));
% % % %     T_BU(1,l) = T_U(1,l).*(1-tau_atm(1,l));
%tau_v*tau_o
% % % tau_atm24 = (tau_v24.*tau_o24.*tau_l24);
% % % tau_atm37 = (tau_v37.*tau_o37.*tau_l37);

% % % T_BD24 = T_D24.*(1-(tau_v24.*tau_o24.*tau_l24));
% % % T_BD37 = T_D37.*(1-(tau_v37.*tau_o37.*tau_l37));
%     T_BD(1,l) = T_D(1,l).*(1-tau_atm(1,l));




%F_24h = A+B.*tau_l24.*(C_1+D_1.*(G+H.*tau_l24));
%F_24v = A+B.*tau_l24.*(C_2+D_2.*(G+H.*tau_l24));
% end

% tau_l24 = -((C+D*G)/(D*H))+sqrt(((C+D*G)/(D*H))^2 -((4*A+T_b24v
%
% % % % F_24v = T_BU24 + tau_atm24.*((E_24v.*SST+((1-E_24v).*(...
% % % %     T_BD24 + tau_atm24.*T_BC))));
% % % % F_24h = T_BU24 + tau_atm24.*((E_24h.*SST+((1-E_24h).*(...
% % % %     T_BD24 + tau_atm24.*T_BC))));
% % % % F_37v = T_BU37 + tau_atm37.*((E_37v.*SST+((1-E_37v).*(...
% % % %     T_BD37 + tau_atm37.*T_BC))));
```

```
% % % % F_37h = T_BU37 + tau_atm37.*((E_37h.*SST+((1-E_37h).*(...
% % % %     T_BD37 + tau_atm37.*T_BC))));
% % % %
% % % % delta_24v = (1-0.99182).*T_B24v + 0.23622;
% % % % delta_24h = (1-0.99648).*T_B24h + 4.7464;
% % % % delta_37v = (1-0.838).*T_B37v - 33.48;
% % % % delta_37h = (1-0.86246).*T_B37h - 15.6;
% % % %
% % % % F_f24v = F_24v + delta_24v;
% % % % F_f24h = F_24h + delta_24h;
% % % % F_f37v = F_37v + delta_37v;
% % % % F_f37h = F_37h + delta_37h;

% F1_24v48 = F_24v;
% F1_24h48 = F_24h;
% F1_37v48 = F_37v;
% F1_37h48 = F_37h;
delta_24v = (1-0.99047).*T_B24v - 0.12293;
delta_24h = (1-0.99523).*T_B24h + 4.4295;
delta_37v = (1-0.83687).*T_B37v - 33.695;
delta_37h = (1-0.86035).*T_B37h - 15.912;

% a_1 = T_U24;
% a_2 = -tau_v24.*tau_o24.*T_U24;
% a_3 = tau_v24.*tau_o24;
% a_4 = E_24v.*SST;
```

```
% a_5 = 1-E_24v;
% a_6 = T_D24;
% a_7 = -tau_v24.*tau_o24.*T_D24;
% a_8 = 2.73.*(tau_v24.*tau_o24);
% a_9 = a_5.*a_6;
% a_10 = a_5.*a_7;
% a_11 = a_5.*a_8;
% a_12 = a_4+a_9;
% a_13 = a_10+a_11;
% a_14 = a_2 + a_3.*a_12;
% a_15 = a_1 - (T_B24v+delta_24v);
% a_16 = a_3.*a_13;
%
% b_1 = T_U24;
% b_2 = -tau_v24.*tau_o24.*T_U24;
% b_3 = tau_v24.*tau_o24;
% b_4 = E_24h.*SST;
% b_5 = 1-E_24h;
% b_6 = T_D24;
% b_7 = -tau_v24.*tau_o24.*T_D24;
% b_8 = 2.73.*(tau_v24.*tau_o24);
% b_9 = b_5.*b_6;
% b_10 = b_5.*b_7;
% b_11 = b_5.*b_8;
% b_12 = b_4+b_9;
% b_13 = b_10+b_11;
```

```
% b_14 = b_2 + b_3.*b_12;
% b_15 = b_1 - (T_B24h+delta_24h);
% b_16 = b_3.*b_13;
%
% c_1 = T_U37;
% c_2 = -tau_v37.*tau_o37.*T_U37;
% c_3 = tau_v37.*tau_o37;
% c_4 = E_37v.*SST;
% c_5 = 1-E_37v;
% c_6 = T_D37;
% c_7 = -tau_v37.*tau_o37.*T_D37;
% c_8 = 2.73.*(tau_v37.*tau_o37);
% c_9 = c_5.*c_6;
% c_10 = c_5.*c_7;
% c_11 = c_5.*c_8;
% c_12 = c_4+c_9;
% c_13 = c_10+c_11;
% c_14 = c_2 + c_3.*c_12;
% c_15 = c_1 - (T_B37v+delta_37v);
% c_16 = c_3.*c_13;
%
%
% d_1 = T_U37;
% d_2 = -tau_v37.*tau_o37.*T_U37;
% d_3 = tau_v37.*tau_o37;
% d_4 = E_37h.*SST;
```

```
% d_5 = 1-E_37h;
% d_6 = T_D37;
% d_7 = -tau_v37.*tau_o37.*T_D37;
% d_8 = 2.73.*(tau_v37.*tau_o37);
% d_9 = d_5.*d_6;
% d_10 = d_5.*d_7;
% d_11 = d_5.*d_8;
% d_12 = d_4+d_9;
% d_13 = d_10+d_11;
% d_14 = d_2 + d_3.*d_12;
% d_15 = d_1 - (T_B37h+delta_37h);
% d_16 = d_3.*d_13;
%
% sol1_24 = (-a_14+sqrt((a_14).^2 -4*(a_16).*(a_15)))./(2.*a_15);
% % sol2_24 = (-a_14-sqrt((a_14).^2 -
4*(a_16).*(a_15)))./(2.*a_15);
% sol3_24 = (-b_14+sqrt((b_14).^2 -4*(b_16).*(b_15)))./(2.*b_15);
% % sol4_24 = (-b_14-sqrt((b_14).^2 -
4*(b_16).*(b_15)))./(2.*b_15);
%
% tau_l24 = (sol1_24+sol3_24)./2;
%
%
% % dif1_24 = abs(sol1_24-sol3_24);
% % dif2_24 = abs(sol1_24-sol4_24);
% % dif3_24 = abs(sol2_24-sol3_24);
```

```
% % dif4_24 = abs(sol2_24-sol4_24);
%
%
% sol1_37 = (-c_14+sqrt((c_14).^2 -4*(c_16).*(c_15)))./(2.*c_15);
% % sol2_37 = (-c_14-sqrt((c_14).^2 -
4*(c_16).*(c_15)))./(2.*c_15);
% sol3_37 = (-d_14+sqrt((d_14).^2 -4*(d_16).*(d_15)))./(2.*d_15);
% % sol4_37 = (-d_14-sqrt((d_14).^2 -
4*(d_16).*(d_15)))./(2.*d_15);
%
% tau_l37 = (sol1_37+sol3_37)./2;
%
% % dif1_37 = abs(sol1_37-sol3_37);
% % dif2_37 = abs(sol1_37-sol4_37);
% % dif3_37 = abs(sol2_37-sol3_37);
% % dif4_37 = abs(sol2_37-sol4_37);
%
%
% [DATA_case12] = [T_B24v T_B24h T_B37v T_B37h V W ...
%     SST CLW latitude longitude...
%     tau_l24 tau_l37 tau_l24test tau_l37test];

A_1 = T_U24+delta_24v;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
```

```
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

A_2 = T_U24+delta_24h;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

A_3 = T_U37+delta_37v;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

A_4 = T_U37+delta_37h;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
```

```
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-
T_B24v)).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-
T_B24h)).*(C_2))))./(2.*C_2);


tau_l24 = (sol2_24+sol4_24)./2;

%tau_l24(tau_l24>1)=1;

dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);

%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
```

```
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11)))./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11)))./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-
T_B37v)).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-
T_B37h)).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case12] = [T_B24v T_B24h T_B37v T_B37h V W ...
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
```

```
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];

function [DATA_case13] = forward_model1_case13(DATA_48)

%Here V>48, abs(SST-TV)>20, (SST-TV)<0,SST<301



% data_name = 'DATA_NO_R&CLW.mat';
% poly_values = 'data_poly.mat';
% load(data_name);
% load(poly_values);

% TF = find(DATA(:,1)~=0);
[DATA] = [DATA_48];

latitude = DATA(:,9);
longitude = DATA(:,10);
T_B24v = DATA(:,1);
T_B24h = DATA(:,2);
T_B37v = DATA(:,3);
T_B37h = DATA(:,4);
SST = DATA(:,7);
sali = 34;
CLW = DATA(:,8);
```

```
V = DATA(:,5);
V_edr = DATA(:,12);
R = DATA(:,11);

% V = polyvaln(polymodel_v2,[T_B24v T_B24h...
%      T_B37v T_B37h SST]);

%W = polyvaln(polymodel_w3,[T_B24v T_B24h...
%     T_B37v T_B37h V SST]);

W = DATA(:,6);
W_edr = DATA(:,13);

%W(W<0) = 0;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


% size_w = size(W,1);
% size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);
```

```
% % id_e24v =
find(real(E_24v)&real(E_24h)&real(E_37v)&real(E_37h));
% % E_24v = E_24v(id_e24v);
% % E_24h = E_24h(id_e24v);
% % E_37v = E_37v(id_e24v);
% % E_37h = E_37h(id_e24v);
% % W = W(id_e24v);
% % V = V(id_e24v);
% % SST = SST(id_e24v);
% % T_B24v = T_B24v(id_e24v);
% % T_B24h = T_B24h(id_e24v);
% % T_B37v = T_B37v(id_e24v);
% % T_B37h = T_B37h(id_e24v);


%b coefficients to calculate T_D and T_U - 23.8GHz and 36.5GHz..
... first row is b_0, second row is b_1, and so on until b_7
b = double([241.69 239.45;310.32E-2 254.41E-2;-814.29E-4 -
512.84E-4;...
    998.93E-6 452.02E-6;-48.37E-7 -14.36E-7;0.2 0.58;-0.2 -
0.57;...
    -5.21E-2 -2.38E-2]);
%oxigen coeffecient 23.8GHz, 36.5GHz
a_o = [15.75E-3,40.06E-3;-0.87E-4,-2.00E-4];
```

```
%water vapor coeff 23.8GHz, 36.5GHz
a_v = [5.14E-3 1.88E-3;0.19E-5 0.09E-5];



%             if (V>48)
                T_v = 301.16;
%              else
%              T_v = 273.16 + 0.8337.*V - 3.029E-5.*V.^(3.33);
%              end

%              if (abs(SST-T_v)>20)
%finding values abs(SST-tv)<=20
fid_sv20 = find((abs(SST-T_v))<=20);
latitude = latitude(fid_sv20);
longitude = longitude(fid_sv20);
T_B24v = T_B24v(fid_sv20);
T_B24h = T_B24h(fid_sv20);
T_B37v = T_B37v(fid_sv20);
T_B37h = T_B37h(fid_sv20);
V = V(fid_sv20);
W = W(fid_sv20);
V_edr = V_edr(fid_sv20);
W_edr = W_edr(fid_sv20);
SST = SST(fid_sv20);
CLW = CLW(fid_sv20);
```

```
E_24v = E_24v(fid_sv20);
E_24h = E_24h(fid_sv20);
E_37v = E_37v(fid_sv20);
E_37h = E_37h(fid_sv20);
R = R(fid_sv20);

%                 if ((SST-T_v)<0)
% fid_pos = find((SST-T_v)>0);
% latitude = latitude(fid_pos);
% longitude = longitude(fid_pos);
% T_B24v = T_B24v(fid_pos);
% T_B24h = T_B24h(fid_pos);
% T_B37v = T_B37v(fid_pos);
% T_B37h = T_B37h(fid_pos);
% V = V(fid_pos);
% W = W(fid_pos);
% SST = SST(fid_pos);
% CLW = CLW(fid_pos);
% E_24v = E_24v(fid_pos);
% E_24h = E_24h(fid_pos);
% E_37v = E_37v(fid_pos);
% E_37h = E_37h(fid_pos);

% if (SST<301)
fid_300 = find(SST<301);
latitude = latitude(fid_300);
```

```
longitude = longitude(fid_300);
T_B24v = T_B24v(fid_300);
T_B24h = T_B24h(fid_300);
T_B37v = T_B37v(fid_300);
T_B37h = T_B37h(fid_300);
V = V(fid_300);
W = W(fid_300);
V_edr = V_edr(fid_300);
W_edr = W_edr(fid_300);
SST = SST(fid_300);
CLW = CLW(fid_300);
E_24v = E_24v(fid_300);
E_24h = E_24h(fid_300);
E_37v = E_37v(fid_300);
E_37h = E_37h(fid_300);
R = R(fid_300);

%find vapor less than 58
fid_58 = find(V>58);
latitude = latitude(fid_58);
longitude = longitude(fid_58);
T_B24v = T_B24v(fid_58);
T_B24h = T_B24h(fid_58);
T_B37v = T_B37v(fid_58);
T_B37h = T_B37h(fid_58);
V = V(fid_58);
```

```
W = W(fid_58);
V_edr = V_edr(fid_58);
W_edr = W_edr(fid_58);
SST = SST(fid_58);
CLW = CLW(fid_58);
E_24v = E_24v(fid_58);
E_24h = E_24h(fid_58);
E_37v = E_37v(fid_58);
E_37h = E_37h(fid_58);
R = R(fid_58);

%                    gui = -14;
%                 else
%                     gui = 14;
%                 end
%             else
gui = 1.05.*(SST - T_v).*(1 - ((SST-T_v).^2)./1200);
%             end

% for l = 1:2;
sz_td = size(R,1);
mat_1 = ones(sz_td,1);
T_D24 = 288.*mat_1;
T_D37 = 288.*mat_1;
%     T_D(1,l) = b(1,l) + b(2,l).*V + b(3,l).*V.^2 + b(4,l).*V.^3
+ b(5,l).*V.^4 + ...
```

```
%         b(6,l).*gui;
T_U24 = T_D24 -2;
T_U37 = T_D37 -2;
%     T_U(1,l) = T_D(1,l) + b(7,l) + b(8,l).*V;

A_o24 = a_o(1,1) + a_o(2,1).*(T_D24-270);
A_o37 = a_o(1,2) + a_o(2,2).*(T_D37-270);

%     A_o(1,l) = a_o(1,l) + a_o(2,l).*(T_D(1,l)-270);

A_v24 = a_v(1,1).*V + a_v(2,1).*V.^2;
A_v37 = a_v(1,2).*V + a_v(2,2).*V.^2;

%     A_v(1,l) = a_v(1,l).*V + a_v(2,l).*V.^2;

tau_o24 = exp(-secd(theta_i24).*A_o24);
tau_o37 = exp(-secd(theta_i37).*A_o37);

tau_v24 = exp(-secd(theta_i24).*A_v24);
tau_v37 = exp(-secd(theta_i37).*A_v37);

%tau_l24 = exp(-secd(theta_i24).*A_l24(idl24));
%tau_l37 = exp(-secd(theta_i37).*A_l37(idl37));

% tau_atm24 = tau_v24.*tau_o24.*tau_l24;
% tau_atm37 = tau_v37.*tau_o37.*tau_l37;
```

```
%     tau_o(1,l) = exp(-secd(theta_i(1,l)).*A_o(1,l));
%     tau_v(1,l) = exp(-secd(theta_i(1,l)).*A_v(1,l));
%     tau_l(1,l) = exp(-secd(theta_i(1,l)).*A_l(1,l));
%     tau_atm(1,l) = tau_v(1,l).*tau_o(1,l).*tau_l(1,l);
% end

%Finding the height of rain

% if (SST<301)

    H_l = 1+0.14.*(SST-273)-0.0025.*(SST-273).^2;
% else
%     H_l = 2.96;
% end


%finding Temperature of rain
T_l = (SST+273)./2;

%z = 2.96;

T_cl = (SST - (6.5).*H_l);


L = CLW;
```

```
A_l24test =0.0891.*(1-0.0281.*(T_cl-283)).*L;
tau_l24test = exp(-secd(theta_i24).*A_l24test);
%tau_l24 = 1;
A_l37test =0.2027.*(1-0.0261.*(T_cl-283)).*L;
tau_l37test = exp(-secd(theta_i37).*A_l37test);
%tau_l37 = 1;
```

```
delta_24v = (1-0.87158).*T_B24v - 29.706;
delta_24h = (1-0.86188).*T_B24h - 24.592;
delta_37v = (1-0.72493).*T_B37v - 59.993;
delta_37h = (1-0.80094).*T_B37h - 27.508;

% a_1 = T_U24;
% a_2 = E_24v.*SST;
% a_3 = 1-E_24v;
% a_4 = T_D24;
% a_5 = 2.7;
% a_6 = a_2+(a_3.*a_4);
% a_7 = (a_3.*a_5)-(a_3.*a_4);
% a_8 = a_6-a_1;
% a_11 = a_1+(T_B24v);
```

```
% a_10 = a_7.*((tau_v24).^2).*((tau_o24).^2);
% a_9 = a_8.*tau_v24.*tau_o24;
%
% b_1 = T_U24;
% b_2 = E_24h.*SST;
% b_3 = 1-E_24h;
% b_4 = T_D24;
% b_5 = 2.7;
% b_6 = b_2+(b_3.*b_4);
% b_7 = (b_3.*b_5)-(b_3.*b_4);
% b_8 = b_6-b_1;
% b_11 = b_1+(T_B24h);
% b_10 = b_7.*((tau_v24).^2).*((tau_o24).^2);
% b_9 = b_8.*tau_v24.*tau_o24;
%
% c_1 = T_U37;
% c_2 = E_37v.*SST;
% c_3 = 1-E_37v;
% c_4 = T_D37;
% c_5 = 2.7;
% c_6 = c_2+(c_3.*c_4);
% c_7 = (c_3.*c_5)-(c_3.*c_4);
% c_8 = c_6-c_1;
% c_11 = c_1+(T_B37v);
% c_10 = c_7.*((tau_v37).^2).*((tau_o37).^2);
% c_9 = c_8.*tau_v37.*tau_o37;
```

```
%
% d_1 = T_U37;
% d_2 = E_37h.*SST;
% d_3 = 1-E_37h;
% d_4 = T_D37;
% d_5 = 2.7;
% d_6 = d_2+(d_3.*d_4);
% d_7 = (d_3.*d_5)-(d_3.*d_4);
% d_8 = d_6-d_1;
% d_11 = d_1+(T_B37h);
% d_10 = d_7.*((tau_v37).^2).*((tau_o37).^2);
% d_9 = d_8.*tau_v37.*tau_o37;
%
% sol1_24 = (-a_9+sqrt(((a_9).^2) -
(4*(a_10).*(a_11))))./(2.*a_10);
% sol2_24 = (-a_9-sqrt(((a_9).^2) -
(4*(a_10).*(a_11))))./(2.*a_10);
% sol3_24 = (-b_9+sqrt(((b_9).^2) -
(4*(b_10).*(b_11))))./(2.*b_10);
% sol4_24 = (-b_9-sqrt(((b_9).^2) -
(4*(b_10).*(b_11))))./(2.*b_10);
%A_1 = T_U24+delta_24v;
A_1 = T_U24;
B_1 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24v.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24v).*T_D24);
```

```
C_1 = (-(1-E_24v).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24v).*((tau_v24).^2).*((tau_o24).^2));
%F_test24v = A_1+(B_1.*tau_l24)+(C_1.*((tau_l24).^2));

%A_2 = T_U24+delta_24h;
A_2 = T_U24;
B_2 = (-
T_U24.*tau_v24.*tau_o24)+(tau_o24.*tau_v24.*E_24h.*SST)+...
    (tau_o24.*tau_v24.*(1-E_24h).*T_D24);
C_2 = (-(1-E_24h).*T_D24.*((tau_v24).^2).*((tau_o24).^2))+...
    (2.7.*(1-E_24h).*((tau_v24).^2).*((tau_o24).^2));
%F_test24h = A_2+(B_2.*tau_l24)+(C_2.*((tau_l24).^2));

%A_3 = T_U37+delta_37v;
A_3 = T_U37;
B_3 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37v.*SST)+...
    (tau_o37.*tau_v37.*(1-E_37v).*T_D37);
C_3 = (-(1-E_37v).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37v).*((tau_v37).^2).*((tau_o37).^2));
%F_test37v = A_3+(B_3.*tau_l37)+(C_3.*((tau_l37).^2));

%A_4 = T_U37+delta_37h;
A_4 = T_U37;
B_4 = (-
T_U37.*tau_v37.*tau_o37)+(tau_o37.*tau_v37.*E_37h.*SST)+...
```

```
    (tau_o37.*tau_v37.*(1-E_37h).*T_D37);
C_4 = (-(1-E_37h).*T_D37.*((tau_v37).^2).*((tau_o37).^2))+...
    (2.7.*(1-E_37h).*((tau_v37).^2).*((tau_o37).^2));
%F_test37h = A_4+(B_4.*tau_l37)+(C_4.*((tau_l37).^2));
sol1_24 = (-B_1+sqrt(((B_1).^2) -(4*((A_1-(T_B24v-
delta_24v))).*(C_1))))./(2.*C_1);
sol2_24 = (-B_1-sqrt(((B_1).^2) -(4*((A_1-(T_B24v-
delta_24v))).*(C_1))))./(2.*C_1);
sol3_24 = (-B_2+sqrt(((B_2).^2) -(4*((A_2-(T_B24h-
delta_24h))).*(C_2))))./(2.*C_2);
sol4_24 = (-B_2-sqrt(((B_2).^2) -(4*((A_2-(T_B24h-
delta_24h))).*(C_2))))./(2.*C_2);



tau_l24 = (sol2_24+sol4_24)./2;


%tau_l24(tau_l24>1)=1;


dif1_24 = abs(sol1_24-sol3_24);
dif2_24 = abs(sol1_24-sol4_24);
dif3_24 = abs(sol2_24-sol3_24);
dif4_24 = abs(sol2_24-sol4_24);


%
% sol1_37 = (-c_9+sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
```

```
% sol2_37 = (-c_9-sqrt(((c_9).^2) -
(4*(c_10).*(c_11))))./(2.*c_10);
% sol3_37 = (-d_9+sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);
% sol4_37 = (-d_9-sqrt(((d_9).^2) -
(4*(d_10).*(d_11))))./(2.*d_10);

sol1_37 = (-B_3+sqrt(((B_3).^2) -(4*((A_3-(T_B37v-
delta_37v))).*(C_3))))./(2.*C_3);
sol2_37 = (-B_3-sqrt(((B_3).^2) -(4*((A_3-(T_B37v-
delta_37v))).*(C_3))))./(2.*C_3);
sol3_37 = (-B_4+sqrt(((B_4).^2) -(4*((A_4-(T_B37h-
delta_37h))).*(C_4))))./(2.*C_4);
sol4_37 = (-B_4-sqrt(((B_4).^2) -(4*((A_4-(T_B37h-
delta_37h))).*(C_4))))./(2.*C_4);

tau_l37 = (sol2_37+sol4_37)./2;
%tau_l37(tau_l37>1)=1;

dif1_37 = abs(sol1_37-sol3_37);
dif2_37 = abs(sol1_37-sol4_37);
dif3_37 = abs(sol2_37-sol3_37);
dif4_37 = abs(sol2_37-sol4_37);


[DATA_case13] = [T_B24v T_B24h T_B37v T_B37h V W ...
```

```
    SST CLW latitude longitude...
    sol2_24 sol4_24 sol2_37 sol4_37...
    R V_edr W_edr T_l H_l];
%    tau_l24 tau_l37 tau_l24test tau_l37test R V_edr W_edr];


function [data_rain] = forward_model_function(data)
%This function finds tau_L for 24v 24h 37v 37h
%T_L and H_L.

% clear
% close
% clc
%
% load(path_file name '*.mat')
%data_name = 'DATA_NO_R&CLW.mat';
%load('/Volumes/disk3/IDR_SDR_DATA/RosaData/d20061207_s132201_e15
1505_r20296.mat');
poly_values = 'data_poly.mat';
% load(data_name);
load(poly_values);
first_row = 1;
last_row = 0;

DATA = data;
```

```
latitude = DATA(:,10);
%latitude = real(latitude);
longitude = DATA(:,11);
%longitude = real(longitude);
T_B24v = (DATA(:,6));
%T_B24v = real(T_B24v);
T_B24h = (DATA(:,7));
%T_B24h = real(T_B24h);
T_B37v = (DATA(:,8));
%T_B37v = real(T_B37v);
T_B37h = (DATA(:,9));
%T_B37h = real(T_B37h);
SST = (DATA(:,2));
%SST = real(SST);
sali = 34;
CLW = (DATA(:,4));
%CLW = real(CLW);
R = (DATA(:,1));
%R = real(R);

V_edr = DATA(:,3);
V_edr = real(V_edr);

V = polyvaln(polymodel_v2,[T_B24v T_B24h...
```

```
   T_B37v T_B37h SST]);
% V = V(TF);

W = polyvaln(polymodel_w3,[T_B24v T_B24h...
  T_B37v T_B37h V SST]);
%
W_edr = DATA(:,5);
W_edr = real(W_edr);
% W = W(TF);
clear('DATA');

W(W<0) = 6.5;
V(V<0) = 0;
V(V>75)=75;

salinity = 34;
theta_i24 = 53.5;
theta_i37 = 53.5;


size_w = size(W,1);
size_v = size(V,1);


[E_24v E_24h] = emiss_oe(23.8,SST,W,theta_i24,salinity);
[E_37v E_37h] = emiss_oe(36.5,SST,W,theta_i37,salinity);
```

```
%%%finding vapor greater than 48 and savin it under DATA_48
fid_v48 = find(V>48);

latitude_v48 = latitude(fid_v48);
longitude_v48 = longitude(fid_v48);
T_B24v_v48 = T_B24v(fid_v48);
T_B24h_v48 = T_B24h(fid_v48);
T_B37v_v48 = T_B37v(fid_v48);
T_B37h_v48 = T_B37h(fid_v48);
V_48 = V(fid_v48);
W_48 = W(fid_v48);
SST_48 = SST(fid_v48);
CLW_48 = CLW(fid_v48);
R_48 = R(fid_v48);
V_edr48 = V_edr(fid_v48);
W_edr48 = W_edr(fid_v48);
[DATA_48] = [T_B24v_v48 T_B24h_v48 T_B37v_v48 T_B37h_v48 V_48
W_48...
    SST_48 CLW_48 latitude_v48 longitude_v48 R_48 V_edr48
W_edr48];

clear('T_B24v_v48','T_B24h_v48','T_B37v_v48','T_B37h_v48','V_48
W_48',...
```

```
'SST_48','CLW_48','latitude_v48','longitude_v48','R_48','V_edr48'
,...
    'W_edr48');

[DATA_case1] = forward_model1_case1([DATA_48]);
[DATA_case2] = forward_model1_case2([DATA_48]);
[DATA_case3] = forward_model1_case3([DATA_48]);
[DATA_case7] = forward_model1_case7([DATA_48]);
[DATA_case8] = forward_model1_case8([DATA_48]);
[DATA_case9] = forward_model1_case9([DATA_48]);
[DATA_case13] = forward_model1_case13([DATA_48]);

clear('DATA_48');
%%%finding vapor less or equal t0 48 and saving it under DATA_47
fid_v47 = find(V<=48);

latitude_v47 = latitude(fid_v47);
longitude_v47 = longitude(fid_v47);
T_B24v_v47 = T_B24v(fid_v47);
T_B24h_v47 = T_B24h(fid_v47);
T_B37v_v47 = T_B37v(fid_v47);
T_B37h_v47 = T_B37h(fid_v47);
V_47 = V(fid_v47);
W_47 = W(fid_v47);
SST_47 = SST(fid_v47);
```

```
CLW_47 = CLW(fid_v47);
R_47 = R(fid_v47);
V_edr47 = V_edr(fid_v47);
W_edr47 = W_edr(fid_v47);
[DATA_47] = [T_B24v_v47 T_B24h_v47 T_B37v_v47 T_B37h_v47 V_47
W_47...
    SST_47 CLW_47 latitude_v47 longitude_v47 R_47 V_edr47
W_edr47];


clear('T_B24v_v47','T_B24h_v47','T_B37v_v47','T_B37h_v47','V_47
W_47',...

'SST_47','CLW_47','latitude_v47','longitude_v47','R_47','V_edr47'
,...
    'W_edr47');

[DATA_case4] = forward_model1_case4([DATA_47]);
[DATA_case5] = forward_model1_case5([DATA_47]);
[DATA_case6] = forward_model1_case6([DATA_47]);
[DATA_case10] = forward_model1_case10([DATA_47]);
[DATA_case11] = forward_model1_case11([DATA_47]);
[DATA_case12] = forward_model1_case12([DATA_47]);

clear('DATA_47');
```

```
sz1 = size(DATA_case1,1);
sz2 = size(DATA_case2,1);
sz3 = size(DATA_case3,1);
sz4 = size(DATA_case4,1);
sz5 = size(DATA_case5,1);
sz6 = size(DATA_case6,1);
sz7 = size(DATA_case7,1);
sz8 = size(DATA_case8,1);
sz9 = size(DATA_case9,1);
sz10 = size(DATA_case10,1);
sz11 = size(DATA_case11,1);
sz12 = size(DATA_case12,1);

for l = 1:13;
names = ['DATA_case01';'DATA_case02';'DATA_case03';...
    'DATA_case04';'DATA_case05';'DATA_case06';...
    'DATA_case07';'DATA_case08';'DATA_case09';...
    'DATA_case10';'DATA_case11';'DATA_case12';'DATA_case13'];

    case_num = names(l,10:11);
    c_01 = strcmp('01',case_num);
    c_02 = strcmp('02',case_num);
    c_03 = strcmp('03',case_num);
    c_04 = strcmp('04',case_num);
    c_05 = strcmp('05',case_num);
    c_06 = strcmp('06',case_num);
```

```
c_07 = strcmp('07',case_num);
c_08 = strcmp('08',case_num);
c_09 = strcmp('09',case_num);
c_10 = strcmp('10',case_num);
c_11 = strcmp('11',case_num);
c_12 = strcmp('12',case_num);
c_13 = strcmp('13',case_num);

if c_01 ==1;
    data_case = DATA_case1;
elseif c_02 ==1;
    data_case = DATA_case2;
elseif c_03 ==1;
    data_case = DATA_case3;
elseif c_04 ==1;
    data_case = DATA_case4;
elseif c_05 ==1;
    data_case = DATA_case5;
elseif c_06 ==1;
    data_case = DATA_case6;
elseif c_07 ==1;
    data_case = DATA_case7;
elseif c_08 ==1;
    data_case = DATA_case8;
elseif c_09 ==1;
    data_case = DATA_case9;
```

```
    elseif c_10 ==1;
        data_case = DATA_case10;
    elseif c_11 ==1;
        data_case = DATA_case11;
    elseif c_12 ==1;
        data_case = DATA_case12;
    elseif c_13 ==1;
    data_case = DATA_case13;
    end

    last_row = last_row + (size(data_case,1));
%     test_frow = 1+(size(data_case,1));
    test_lrow = size(data_case,1);
    if test_lrow>0;
    DATA_FINAL(first_row:last_row,1) =
data_case(1:size(data_case,1),1);
    DATA_FINAL(first_row:last_row,2) =
data_case(1:size(data_case,1),2);
    DATA_FINAL(first_row:last_row,3) =
data_case(1:size(data_case,1),3);
    DATA_FINAL(first_row:last_row,4) =
data_case(1:size(data_case,1),4);
    DATA_FINAL(first_row:last_row,5) =
data_case(1:size(data_case,1),5);
    DATA_FINAL(first_row:last_row,6) =
data_case(1:size(data_case,1),6);
```

```
    DATA_FINAL(first_row:last_row,7) =
data_case(1:size(data_case,1),7);
    DATA_FINAL(first_row:last_row,8) =
data_case(1:size(data_case,1),8);
    DATA_FINAL(first_row:last_row,9) =
data_case(1:size(data_case,1),9);
    DATA_FINAL(first_row:last_row,10) =
data_case(1:size(data_case,1),10);
    DATA_FINAL(first_row:last_row,11) =
data_case(1:size(data_case,1),11);
    DATA_FINAL(first_row:last_row,12) =
data_case(1:size(data_case,1),12);
    DATA_FINAL(first_row:last_row,13) =
data_case(1:size(data_case,1),13);
    DATA_FINAL(first_row:last_row,14) =
data_case(1:size(data_case,1),14);
    DATA_FINAL(first_row:last_row,15) =
data_case(1:size(data_case,1),15);
    DATA_FINAL(first_row:last_row,16) =
data_case(1:size(data_case,1),16);
    DATA_FINAL(first_row:last_row,17) =
data_case(1:size(data_case,1),17);
    DATA_FINAL(first_row:last_row,18) =
data_case(1:size(data_case,1),18);
    DATA_FINAL(first_row:last_row,19) =
data_case(1:size(data_case,1),19);
```

```
    first_row = first_row+(size(data_case,1));

    else
    end

end

data_rain = DATA_FINAL;
```

LIST OF REFERENCES

[1] Frank J. Wentz, “A well-calibrated ocean algorithm for special sensor microwave / imager”, *Journal of Geophysical Research 102:C4*, 8703-8718,1997

[2] W. Roy Spencer and Frank J. Wentz, “SSM/I Rain Retrievals within a Unified All-Weather Ocean Algorithm”, *Journal of the Atmospheric Sciences 55:9*, 1613-1627, 1998

[3] F. J. Wentz and T. Meissner, “Algorithm Theoretical Basis Document (ATBD), version 2: AMSR Ocean Algorithm”, *RSS Tech. Proposal 121599A-1. Remote Sensing Systems, Santa Rosa, CA,* 66 pp, 2000

[4] Salman Khan, “Simulation of Brightness Temperature for the Microwave Radiometer on the Aquarius/SAC-D Mission”, *In Electrical Engineering and Computer Science*. Vol. PhD Orlando: UCF, 2009

[5] Kaushik Gopalan, Linwood Jones, Sayak Biswas, Steve Bilanow, Thomas Wilheit and Takis Kasparis, “A Time-Varying Radiometric Bias Correction for the TRMM Microwave Imager, *accepted for publication IEEE Trans. GeoSci & Rem Sens,* 2009

[6] Salem El-Nimri, “Development of an Improved Microwave Ocean Surface Emissivity Radiative Transfer Model”, *In Electrical Engineering and Computer Science*. Vol. PhD Orlando: UCF, 2010

[7] Spencer Farrar, Linwood Jones, Sergio Masualli and Juan-Cruz Gallio, “Microwave Radiometer (MWR) Oceanic Integrated Rain Rate Algorithm for Aquarius/SAC-D”, *In Central Florida Remote Sensing Laboratory (CFRSL)*. Orlando: UCF

[8] http://aquarius.gsfc.nasa.gov/