Remote Sensing Solutions, Inc.

PROJECT: Development of Operational SFMR Validation and Processing Tools

NOAA Award Number: NA07OAR4590426

First Year - Interim Report 01

Prepared By: Dr. James Carswell

Date: 10/31/2007

Table of Contents

List of Figures

1 Introduction

1.1 Scope

This document serves as the first interim report for the Joint Hurricane Testbed (JHT) project entitled, "Development of Operational SFMR Validation and Processing Tools". It provides a summary of the work performed to date by Remote Sensing Solutions and Hurricane Research Division for this effort.

1.2 Applicable Documents

The following is a list of references for citations from within this report:

1. JHT Proposal: "Development of Operational SFMR Validation and Processing Tools"

1.3 Document Breakdown

This document contains three sections. Section 1 contains the introduction. Section 2 reviews the work performed and results obtained to date by Remote Sensing Solutions (RSS) for this effort. Section 3 summarizes the work performed and results obtained to date by the Hurricane Research Division. Section 4 lists errors and problems discovered in recon data files while building the real-time processor.

2 RSS - Work Performed & Results Obtained

During the first interim period, Remote Sensing Solutions focused on developing tools and analyses techniques required to validate the SFMR retrievals transmitted from the Air Force and NOAA aircraft and to provide advanced products for display within the forecaster's environment. To achieve these objectives, Remote Sensing Solutions designed and developed: 1) a real-time processor that automatically fetches reconnaissance observations and center fix data, parses these data into storm files, implements storm relative processing and collocates the dropsonde and SFMR data; and 2) an analysis tool that can be implemented within the SFMR system to automatically validate the calibration of the SFMR measurements without the need for in situ surface wind and rain measurements. Below these efforts and the results obtained are presented.

2.1 Real-time Processor – Stage One

Remote Sensing Solutions designed and implemented the first stage of the real-time processor. The primary objective of this processor is to collect and process the reconnaissance data so that their information can be conveyed to the end users when, where and how they need it. Specifically, the processor will provide forecasters with the SFMR retrievals and derived products, as well as a means to validate these observations in a storm relative coordinate system that can be displayed within the NAWIPS environment and also through other real-time display applications. This processor is designed to run unattended, 24 hours a day – 7 days a week (24/7). Since it is Python-based, it can run on Linux, UNIX or Windows based computers. The only requirements are that the computer it resides on has Internet access to communicate with the servers receiving the posted reconnaissance and center fix data files, Python 2.4 or later, and the proper NetCDF libraries to handle archiving of processed data. It is envisioned that this real-time processor will run at NHC and HRD. It could also be deployed at other WFOs.

Figure 1 presents the flow chart for the first stage of the real-time processor (and post flight processor). The green boxes indicate python applications, the red boxes are data files in their original format and the yellow boxes are NetCDF files containing the quality controlled processed data. The real-time processor consists of three main layers:

Layer 1: Initial acquisition

Layer 2: Quality control, parsing and location processing

Layer 3: Collocation processing

To minimize code changes and software maintenance from year to year associated with format changes in the reconnaissance data files and other data files, the processor has been developed such that only the parsing code in Layer 2 will need to be updated. The output of this second layer is a series of standard NetCDF files that all following applications read. Thus the second layer buffers all following layers and stages from format changes in the reconnaissance and other data files.

2.1.1 Layer 1 - Initial Acquisition

Layer 1 acquires and locally archives reconnaissance data files and other data files required for storm relative processing and validation of the SFMR retrievals. This initial acquisition layer consists of two python applications: fetchWeatherMessages.py and fetchCenterFixes.py. Their functions are described below.

2.1.1.1 Weather Messages

The application, fetchWeatherMessages.py, automatically detects and retrieves reconnaissance data files as they are posted to the NOAA web site:

http://ratfish.nhc.noaa.gov /archive/recon

and stores them in a local archive. The file types include, but are not limited to, HDOB (URNT15), REPNT2 and REPNT3. An XML configuration file governs the operation of this application by specifying the data types it should fetch and their relative location within the archive. It also allows the user to specify other parameters, such as the year. Normally, this application fetches data from the current year (i.e. assumes real-time operation), but it can also retrieve files from years past. This allows users to automatically build full archives of the reconnaissance data when desired. The primary data server from which it retrieves these data files is 'ratfish'. NHC personnel instructed us to use this server as it would provide reliable, 24/7 access.

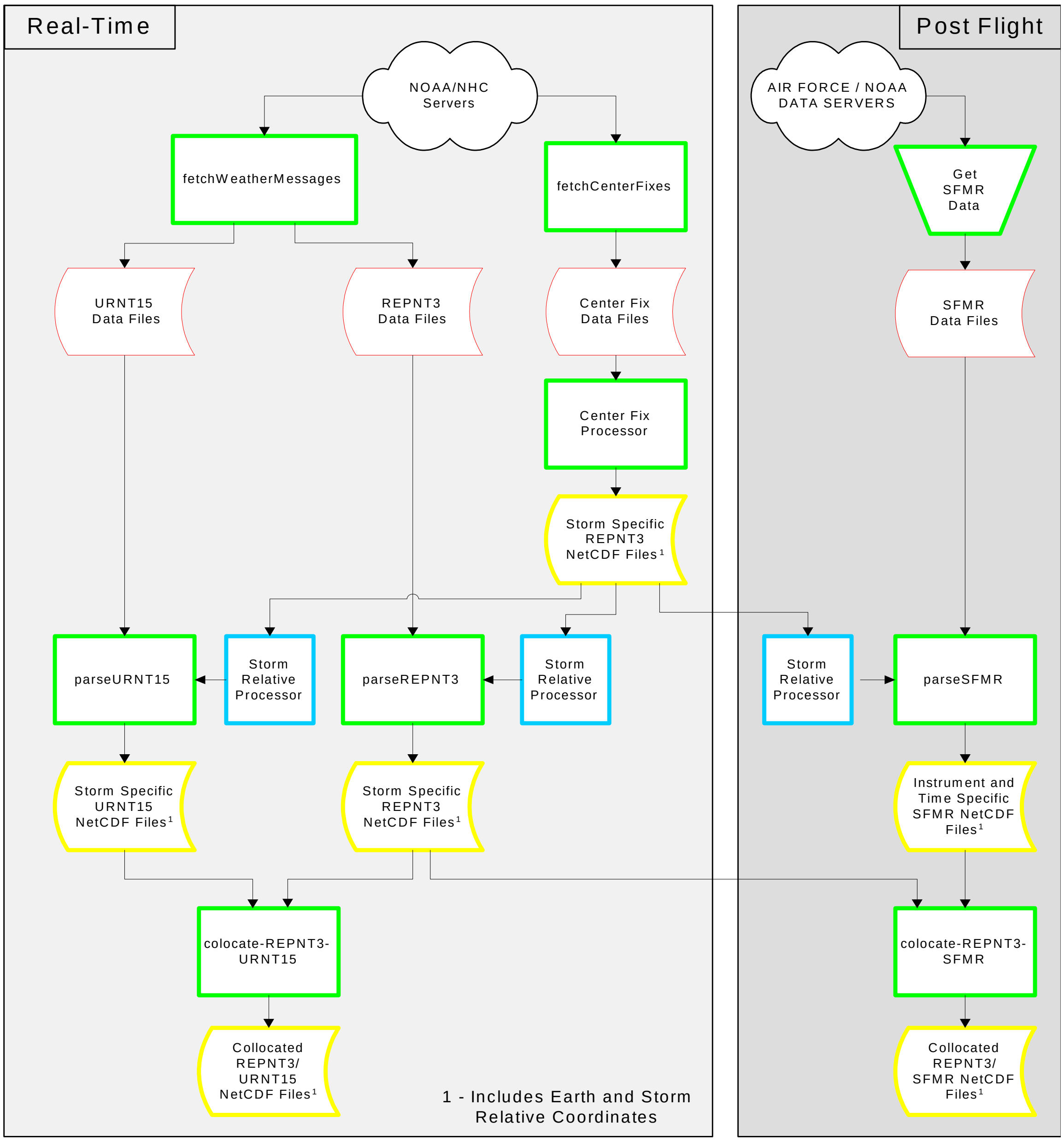

Figure 1: Flow chart of real-time and post flight processing.

In its normal running mode, the application queries the website every 30 seconds and auto-discovers any new files that have been posted. Once new files are detected, the application retrieves them and writes them to the local archive. The application can be run unattended, 24/7. For testing purposes, Remote Sensing Solutions has run this application on several Linux-based servers throughout the 2007 hurricane season without fault.

2.1.1.2 Center Fixes

The application, fetchCenterFixes.py, retrieves center fix data files posted at:

ftp://ftp.nhc.noaa.gov/atcf/fix

This application is also configurable by an XML configuration file. It detects any new files that are posted for the specified year and downloads them to its local archive. Normally it runs in a loop back mode checking for new data every two minutes. It is designed to run unattended, 24/7. Since being built in September, it has run full-time without fault.

2.1.2 Layer 2 – QOC, Parsing & Location Processing

Layer 2 serves three primary purposes. First, it implements a buffer zone to handle changes in reconnaissance and other data formats. Each year recommendations are implemented that potentially modify the structure and/or format of the reconnaissance data files, storm files and data servers. To prevent these changes from propagating through the entire real-time processor, and to increase the IO efficiency of the third layer and later processing stages, layer 2 parses the files retrieved by layer 1, which are in the format specified by the National Hurricane Operations Plan (NHOP) document, and stores the parsed information in standard formats within NetCDF files. In this manner, layer 3 applications and later stages are unaffected by format changes in the initial files (e.g. those regulated by NHOP). With NetCDF files, later applications can be written to automatically configure themselves and self generate read procedures to access data within the NetCDF files. The applications can efficiently access subsets of these files (i.e. individual variables) with block reads rather than complicated pointer manipulation and single reads. This significantly reduces software development and maintenance time and improves run time efficiency.

Each parser also follows a QOC rule set to detect, and correct or remove, any errors in the original files so that later applications are not impacted. It has been our experience that the reconnaissance files contain several errors and that significant coding is required to catch and correct these errors. Catching them in layer 2 significantly reduces the amount of QOC monitoring and handling in layer 3 applications and beyond, thus improving their performance and simplifying their development.

Currently, layer 2 consists of three main applications: center fix processor, URNT15 parser and the REPNT3 parser. Future expansion of this layer may include parsers for the other reconnaissance data or any other data types that are needed for processing stages that follow. The center fix processor, URNT15 parser and REPNT3 parser are described below.

2.1.2.1 Center Fix Processor

The center fix processor monitors the center fix file archive created by the level 1 application, fetchCenterFixes.py. When new data arrives, it passes the data through the quality control procedure, extracts the new center fix data (i.e. time, latitude, longitude, etc), interpolates to 30 second intervals between fixes and writes the data to NetCDF files which are organized in terms of the tropical depression number (and storm name when given). This application also extrapolates the center fix data 30 minutes beyond the last fix by using the last known storm motion. Values that are extrapolated are flagged and when new center fixes arrive, the extrapolated data are replaced by the new information. Extrapolation is needed to ensure that the storm relative processor for the reconnaissance data processors obtains the information it needs.

2.1.2.2 Storm Relative Processor

The storm relative processor was built as a python module to enable its use by all parsers to map their data into a storm relative coordinate system. At each run, this module updates its center fix locations, stored by TD number and time, in the center fix files created by layer 1. For each data point, the storm relative processor looks up the center fix location using an efficient index method. It calculates the radial distance and angle, from the center fix location at the time the data point was acquired, to the data point location (i.e. storm relative coordinate system). The processor passes this storm relative position back to the parser to be stored along with the data in the NetCDF file. In this manner, the parsed data can be mapped to a center fix location at any point in time by mapping the storm relative coordinate system to that center fix location.

During operation, the storm relative processor continues to check for updated center fix data. It then updates coordinates that were based on extrapolated values as soon as the actual values are available in the center fix file created in layer 1. Recall that the layer 1 application extrapolates data 30 minutes beyond the last center fix to ensure that a center fix location is always available. When it does extrapolate, a flag is set to let the storm relative processor know that the value is extrapolated, not measured.

Following the discussion of the URNT15 parser, an example of the storm relative processing will be shown (Figure **3**).

2.1.2.3 URNT15 Parser

The URNT15 parser application monitors the URNT15 data files in the local archive detecting the presence of new files. When a new file appears, the parser reads the file and determines which TD number / storm it belongs to. Following a specific rule set, it quality controls the data within the file, extracts each parameter into its NetCDF variable, calls the storm relative processing function to determine the storm coordinate system for each point and writes the variables and storm relative coordinate system to output a NetCDF file associated with the TD number. Note that the NetCDF files were originally organized by the storm name. For example all URNT15 data that were collected from missions through Dean are stored in the URNT15-DEAN.nc file. We are now modifying this approach to store the data according to the TD number. So for the given example, the data will now be stored in URNT15-TD04-DEAN.nc.

As with previously built applications, this module runs continuously in an unattended mode (24/7). It has a variable timing loop that by default is set to 30 seconds (the minimal reporting interval). Remote Sensing Solutions has run this application throughout the 2007 hurricane season to ensure its performance and ability to trap / handle errors in the original files. It has been running without fault since early September 2007.

Figure 2 plots the SFMR data contained within the URNT15 parsed file for Hurricane Dean. Each SFMR data point is plotted versus its latitude and longitude position and color coded according the SFMR wind speed estimate. Figure 3 plots the same data but uses the storm relative position calculated in real-time by the storm relative processor. This example is a bit extreme in that it shows data from multiple successive days, in this case, August 16th through 19th, 2007. In comparison, a user may choose to focus only on a twelve hour or shorter period. Nevertheless, with the storm motion removed, information from the SFMR on wind radii and storm structure is more easily viewed. The coordinate system in this figure can be placed at any center fix location just by added the fix for that time.

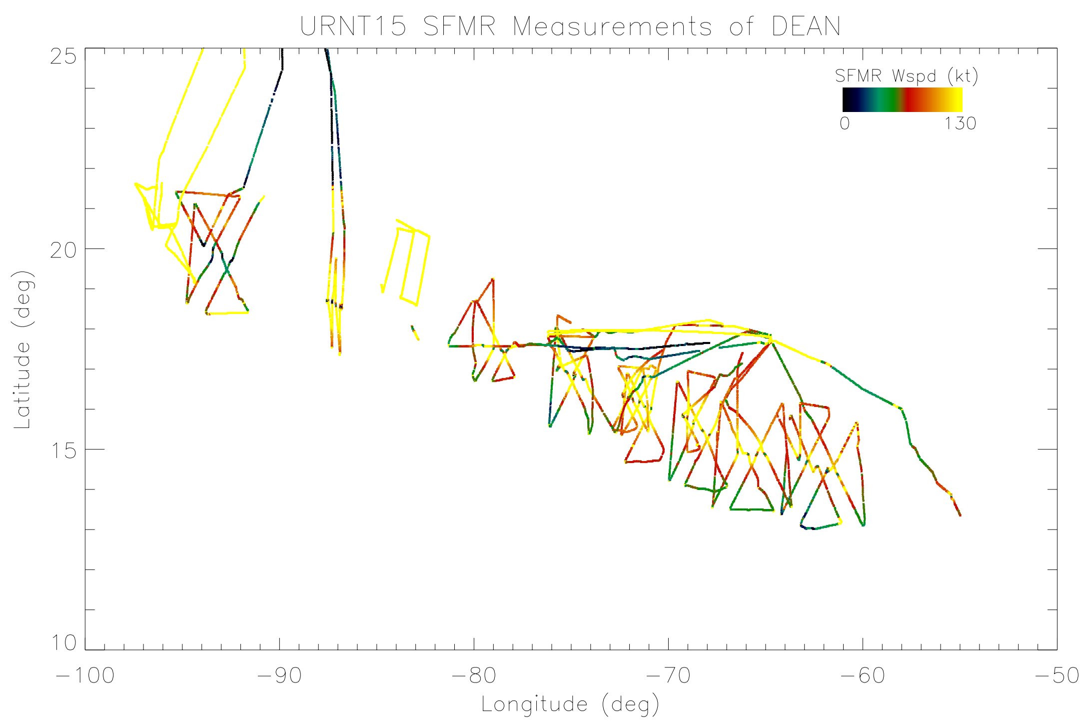

Figure 2: Flight tracks locations for URNT15 data are plotted for missions through Hurricane Dean. The points are color coded according the SFMR wind speed reported in the URNT15 files. Note that these data were extracted from the output of the URNT15 parser discussed in the next section.

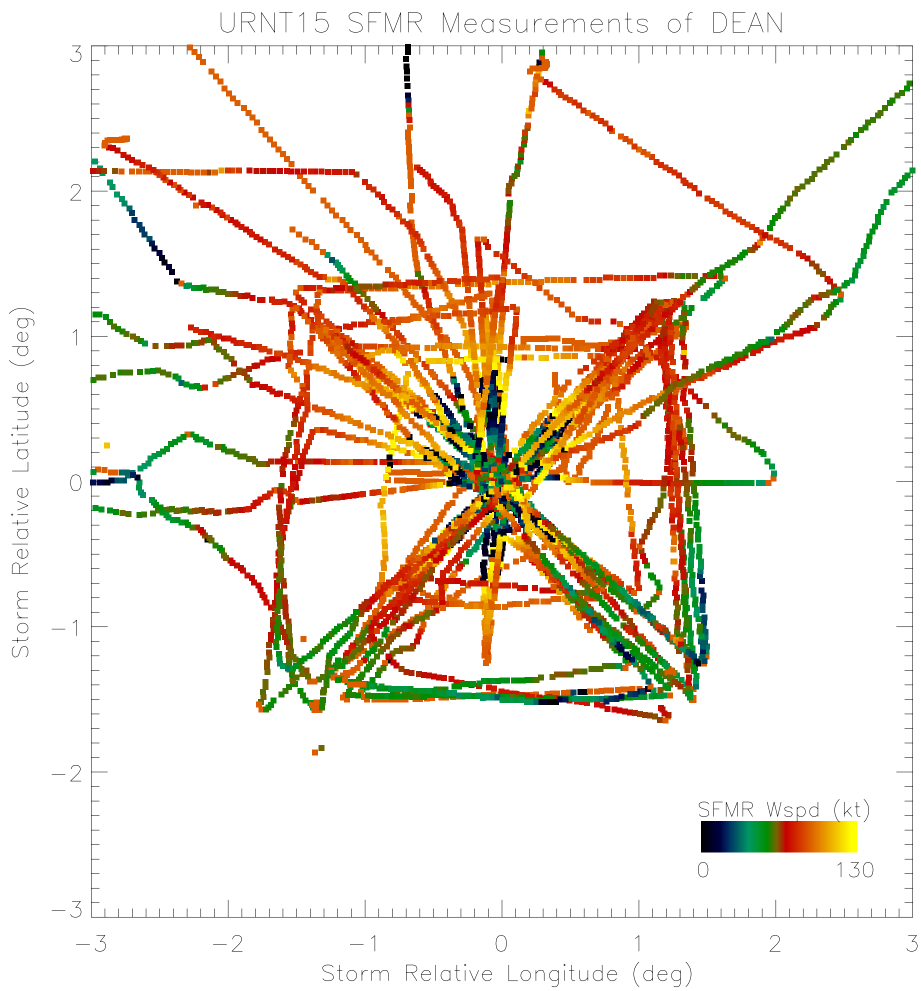

Figure 3: Same data as in Figure 2 but plotted versus storm relative coordinate system.

2.1.2.4 REPNT3 Parser

The REPNT3 parser application is built and runs similarly to the URNT15 parser application except that it parses the GPS dropsonde data rather than the SFMR and flight level observations. Currently this parser only extracts Part A - the standard levels. If needed, it could be modified to parse Part B of these files as well. Like the previous parser, it also calculates the storm relative coordinate system for each data point and stores the parsed data in NetCDF files organized by TD number. It is designed to run unattended, 24/7, and by default, checks for new files every 30 seconds. Remote Sensing Solutions has run this application during the 2007 hurricane season to ensure it handles all errors in the original files. It has been running without fault since September.

2.1.3 Layer 3 – Collocation Processing

To provide real-time validation of the SFMR retrievals and a data archive that can be used for more in depth analysis following the hurricane season, a collocation processor was designed and built. Like applications in layers 1 and 2, this processor is designed to run unattended, 24/7. Its primary objective is to discover all URNT15 and REPNT3 data that are within a specified distance and time window of one another. Note that the main criterion is distance since the aircraft moves quickly from the dropsonde and the wind field changes quickly in the radial direction (direction most often flown by the aircraft). The time filter serves as means to prevent later flight legs that may over fly the splash point from being used. The splash location and splash time of the REPNT3 are used for this collocation since the SFMR measures the surface wind.

Monitoring the parsed URNT15 and REPNT3 files, this application detects when new data are present. It then determines if the new data are collocated within the specified distance and time window. For each data point that meets this criterion, the URNT15 and REPNT variables for that point are written to a collocation NetCDF file. These collocation files are organized by TD number. Along with the data, the distance and time separation between the URNT15 and REPNT3 data is stored so that later applications can use a stricter window if needed.

Remote Sensing Solutions has run this application during the 2007 hurricane season, and it continues to run without fault. Figure 4 plots the flight track for a mission through Hurricane DEAN on 19 August, 2007. The location of the SFMR retrievals is shown by the black and green dots. The green dots indicate those points that met the collocation criteria of 15 km and 1 hour within the splash location (red dots) and time of a GPS dropsonde measurement. The black circles show the 15 km radius circle around the splash location. For reference, time stamps are given at the blue triangles and the observation number for each dropsonde is given. As this figure shows, the collocation process (and all the other processes discussed above) are performing as intended.

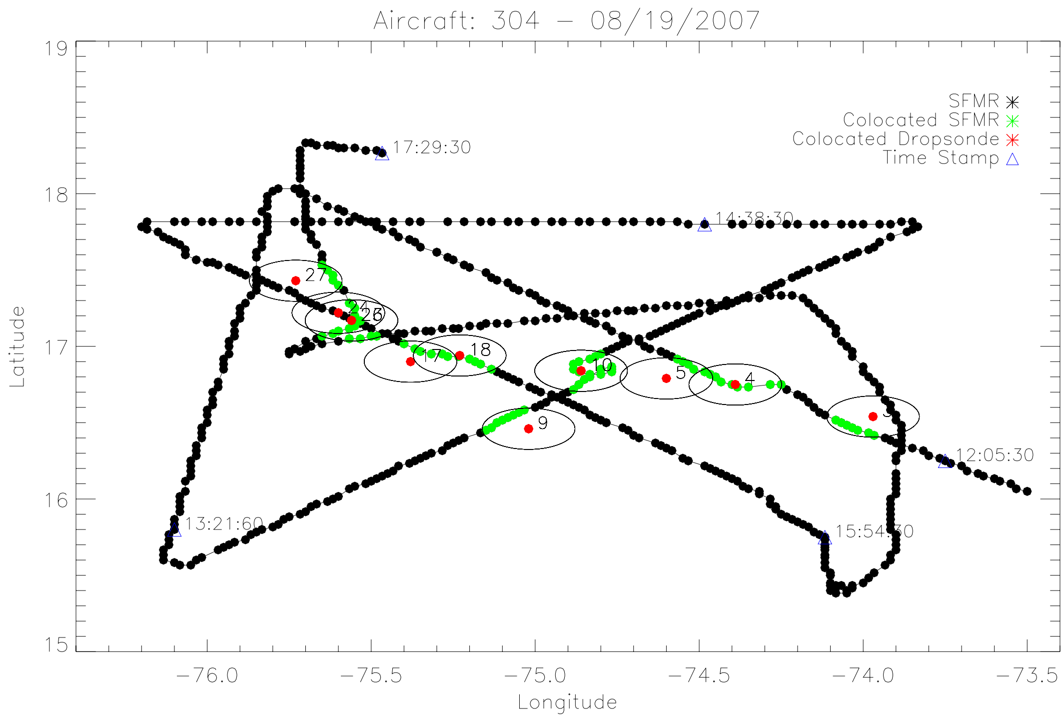

Figure 4: Flight track on 19 August 2007 through Hurricane Dean is plotted. The black and green dots show the SFMR measurement locations with the green indicating those found to be within 15 km and 1 hour of a GPS dropsonde splash location (red dots). A 15 km radius circle around each splash location is drawn and the observation number given. Time stamps a long the flight track are marked by the blue triangles as a reference.

2.2 Real-time Analysis Tools

As part of this project, a key capability that must be developed and implemented is the ability to validate and detect errors in the SFMR calibration and stability. As detailed in our previous JHT project, “Operational SFMR-NAWIPS Airborne Processing and Data Distribution Products”, which focused on the NOAA SFMR, small calibration errors in the SFMR translate to significant errors in the SFMR wind speed estimates. For the Air Force SFMR units, ProSensing, Inc, the manufacturer, performs a laboratory calibration. The instrument is then installed and a calibration flight in low to moderate wind conditions is flown. GPS dropsonde surface wind observations and buoy-based wind observations are compared against the SFMR wind estimates. If the SFMR wind observations are found to disagree from the buoy and dropsonde measurements, the SFMR calibration offset parameter is tuned to eliminate this error. Although this approach will remove significantly large errors, it still does not provide the necessary accuracy to ensure an acceptable maximum level of uncertainty in the SFMR retrievals. Part of the problem lies in the sole criteria being applied is that of the wind speed retrievals being compared to the in situ wind speed measurements. For a small range of wind and rain conditions, the calibration bias for the SFMR can be tuned to produce reasonable wind comparisons, but residual error in the calibration may and probably will still exist. That is, some of the channels may produce higher brightness temperature measurements and other channels lower brightness temperature measurements compared to the model function. For the limited set of wind / rain conditions sampled, these errors can offset each other thereby erroneously producing wind retrievals that agree with the in situ measurements. However, under different wind and rain conditions, these errors can produce significant errors in the SFMR wind speed retrievals. To ensure this scenario is eliminated, an approach to validate the calibration of the SFMR that accounts for both the wind and rain contributions is required.

Remote Sensing Solutions has developed such an analysis approach. The novel part of this approach is that it does not require in situ wind or rain estimates and uses a parameter already calculated within the retrieval process. With a minor modification to the real-time processor, this calibration-validation approach could be implemented on the operational SFMR systems. The premise of the approach is: If the instrument calibration is properly tuned to the model function, then the measurements should agree in the mean with the predicted brightness temperatures that are based on the retrievals. In fact, the retrieval process itself tries to accomplish this objective. It adjusts its wind speed and rain rate estimates to minimize the error between the six frequency brightness temperature measurements (per channel) and the predicted brightness temperature values derived using the SFMR model function.

To illustrate, RSS' SFMR simulator was used to produce a set of SFMR measurements for wind and rain conditions ranging from 0 to 80 m/s and 0 to 80 mm/hr. The standard deviation of the simulated brightness temperature measurements was set to 0.5 K. and 200 realizations at each wind speed and rain rate level were produced. The simulated measurements were passed through the SFMR retrieval process and wind speed and rain rate estimates derived. These estimates showed a zero mean bias since a calibration bias was not introduced. The wind and rain retrievals were then passed back through the SFMR brightness temperature model function to produce a set of predicted brightness temperature measurements. For each channel the difference between the measured and predicted brightness temperature measurement were calculated. **Figure 5** plots a histogram of this difference in terms of percent of occurrence as a function of the error for each channel. As expected, each channel has a zero mean bias indicating that there is no calibration error.

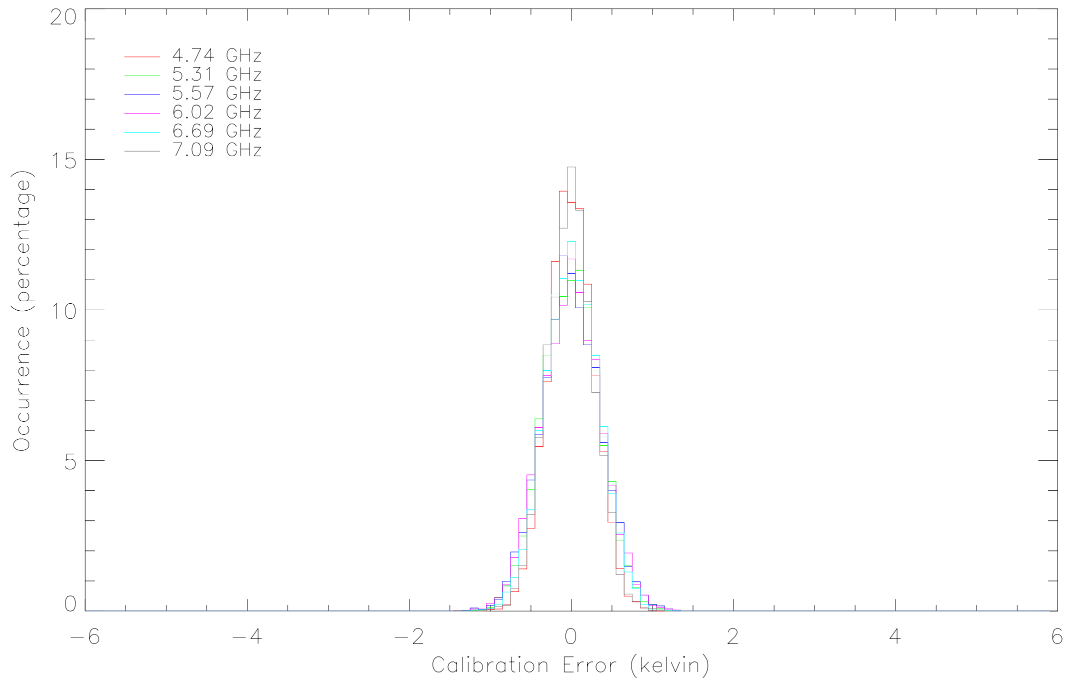

Figure 5: Histogram of the difference between the measured Tb and the predicted Tb for each channel is shown. The calibration error for all channels is set to zero.

Following the same procedure, a calibration error was introduced into the second highest frequency channel (-1 K error). The retrieval process was run and the predicted brightness temperatures derived from the retrievals. Figure 6 presents the error histogram. In this case, channel 5 clearly shows a negative bias. Because the retrieval process believes the measurements to be true and attempts to minimize the error between the measured and predicted brightness temperatures, its wind speed retrievals for this case are slightly high and the rain rate retrievals slightly low. The error in channel 5 spreads into adjacent channels with channel 6 having a slightly high bias. In any event, this approach clearly detects calibration errors. Further it does not require any in situ wind measurements nor does it require specific wind or rain conditions. In fact, it can be run continuously during all missions to monitor the calibration and health of the instrument. Additionally, as mentioned previously, any caluculated difference within the retrieval process could be made available as a quality control and calibration validation tool.

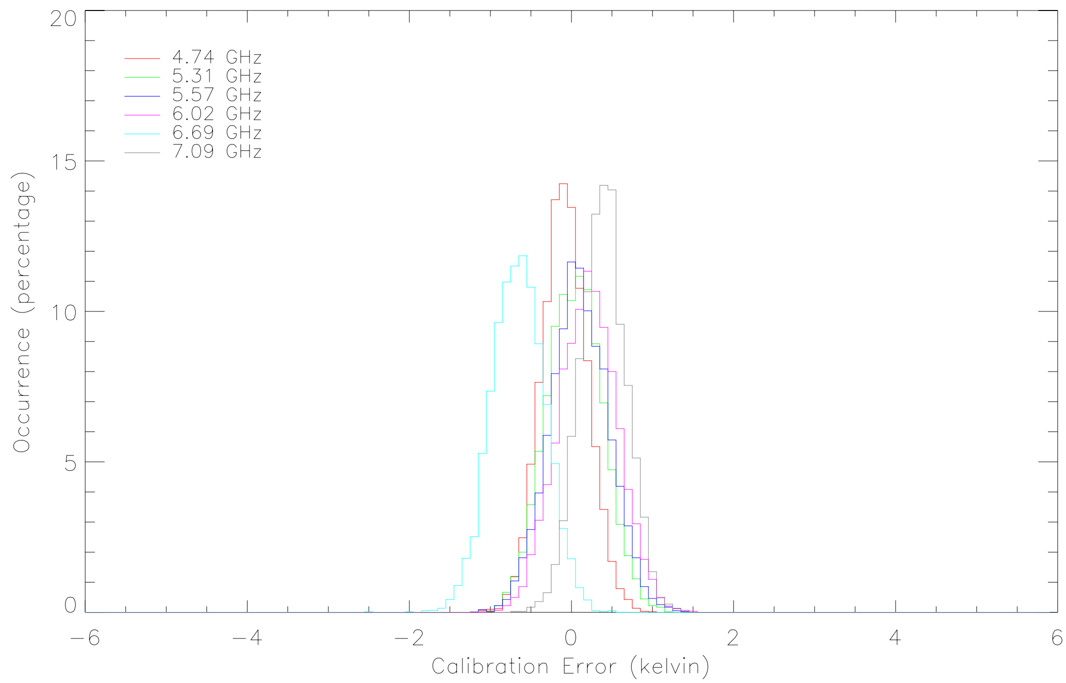

Figure 6: Same as Figure 5 except a -1K calibration error was introduced into channel 5.

Applying this analysis to SFMR observations collected from missions through Hurricane Dean, Figure 7 through Figure 9 plot the SFMR calibration errors for each channel derived from measurements on the 16^{th}, 17^{th} and 18^{th} of August, 2007. These plots show that channel 5 is biased low and channel 6 is biased high, which is exactly as shown in the above simulation. Also present are small errors in the lower channels. As mentioned previously, these calibration errors will result in errors in the retrieved wind speed. To show the stability of this approach, the errors calculated on the 16^{th} were removed from the measurements on the 17^{th} and 18^{th} and the error histograms recalculated. The results are shown in Figure 10 and Figure 11. As these figures show, the same errors are seen on all three flights.

Using both this new analysis approach to provide guidance along with the existing calibration tuning procedures, it should be possible to remove the calibration errors of the SFMR. Remote Sensing Solutions is currently working on a few methods to automate this process.

Another advantage of this approach is its ability to be run continuously in order to provide feedback on the performance of each SFMR. In the event that an instrument's measurements begin to drift or one of its channels begins to fail, this analysis will immediately detect the problem. In fact, by setting a simple threshold (such as an absolute bias less then 0.2K), a simple indicator for each channel can notify the operator that the threshold has been crossed. If the error is found to originate from a single channel, that channel can be disabled. Therefore, the detection of this error state during a flight will allow the operator to immediately address this issue after the flight. Note that all information to implement this approach resides within the SFMR instrument itself, meaning the operator and/or end user is not required to possess a detailed understanding of the SFMR operations and the remote sensing theory behind the process in order to effectively use this information.

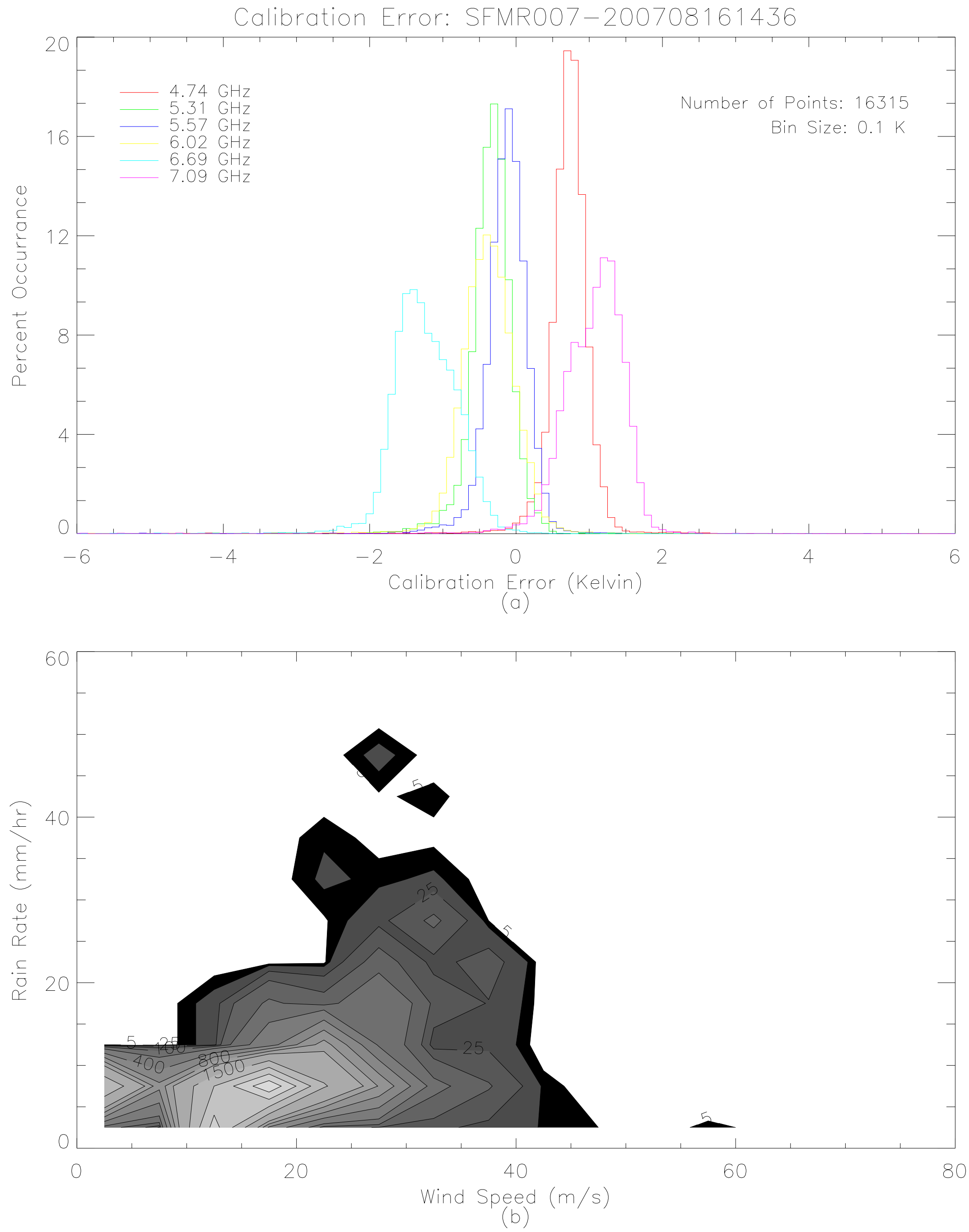

Figure 7: Panel (a) plots a histogram of the calibration error for SFMR007 during a mission through Dean on 16 August 2007. Panel (b) presents a 2-D histogram of the retrieved wind speeds and rain rates for this flight to provide an estimate of the conditions sampled. Note that because a calibration error exists, the wind and rain observations will also be in error.

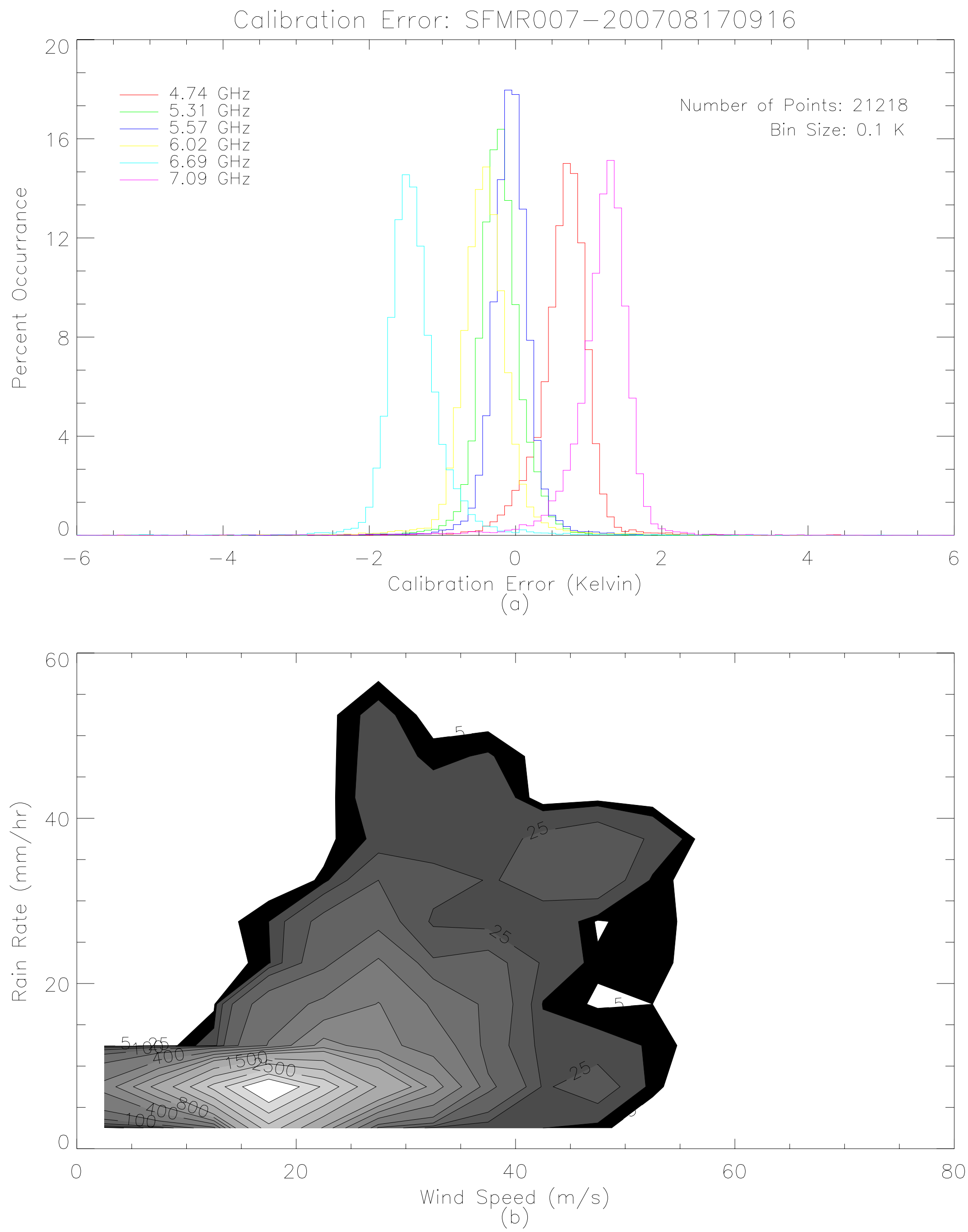

Figure 8 Panel (a) plots a histogram of the calibration error for SFMR007 during a mission through Dean on 17 August 2007. Panel (b) presents a 2-D histogram of the retrieved wind speeds and rain rates for this flight to provide an estimate of the conditions sampled. Note that because a calibration error exists, the wind and rain observations will also be in error.

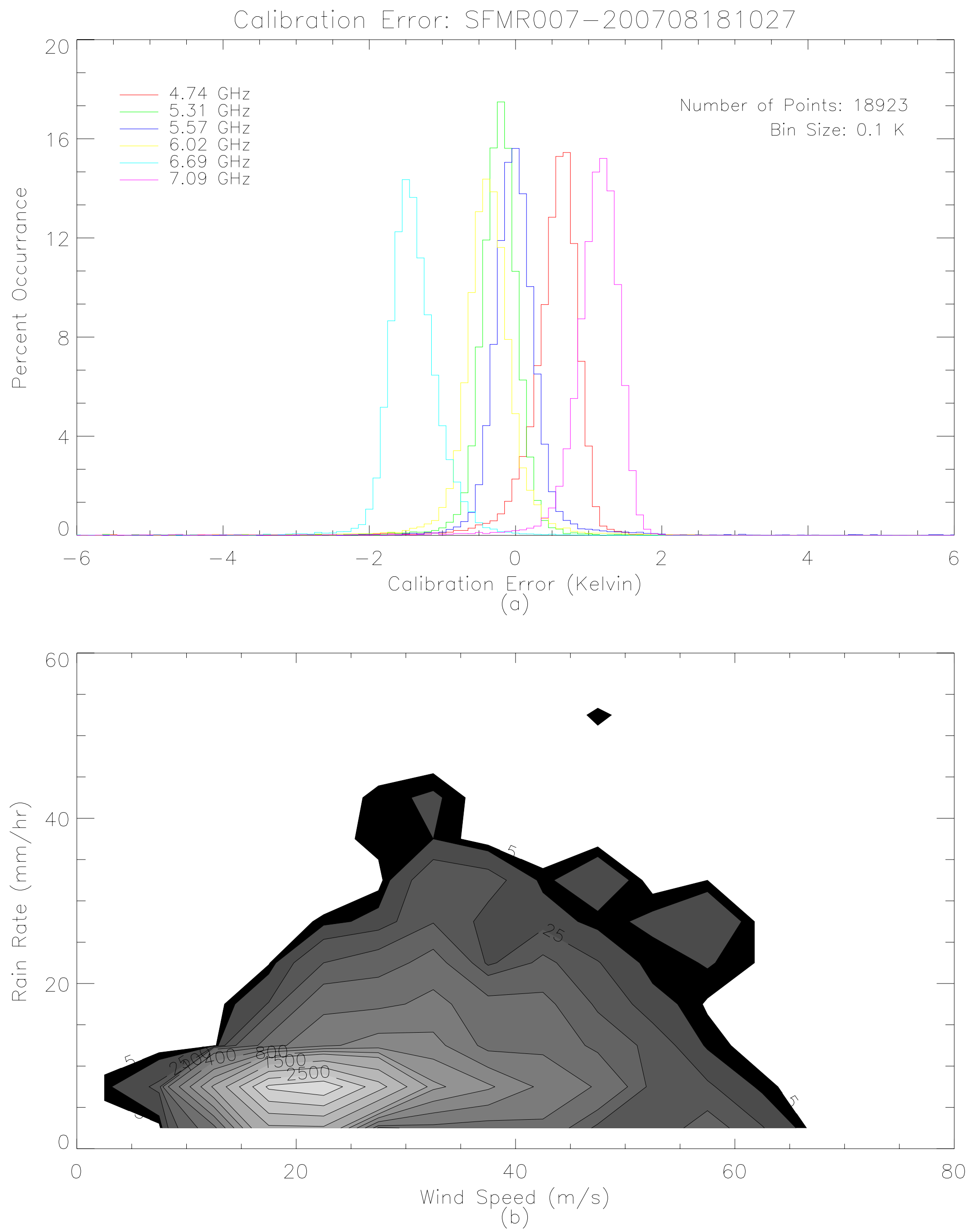

Figure 9 Panel (a) plots a histogram of the calibration error for SFMR007 during a mission through Dean on 18 August 2007. Panel (b) presents a 2-D histogram of the retrieved wind speeds and rain rates for this flight to provide an estimate of the conditions sampled. Note that because a calibration error exists, the wind and rain observations will also be in error

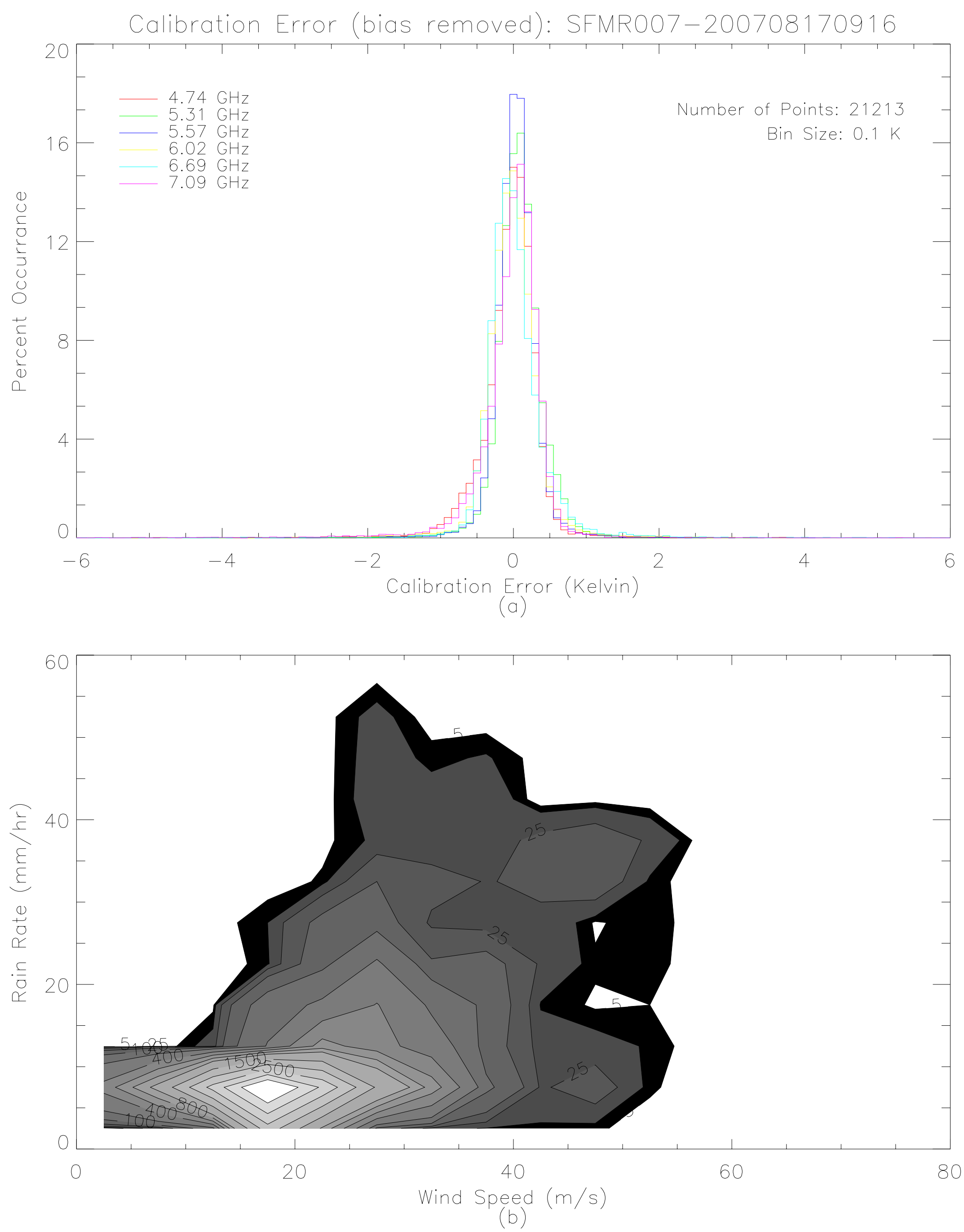

Figure 10: Same as Figure 8 except the mean errors calculated from the 16 August 2007 flight are used to correct the measurements. The errors now have a zero mean for all channels.

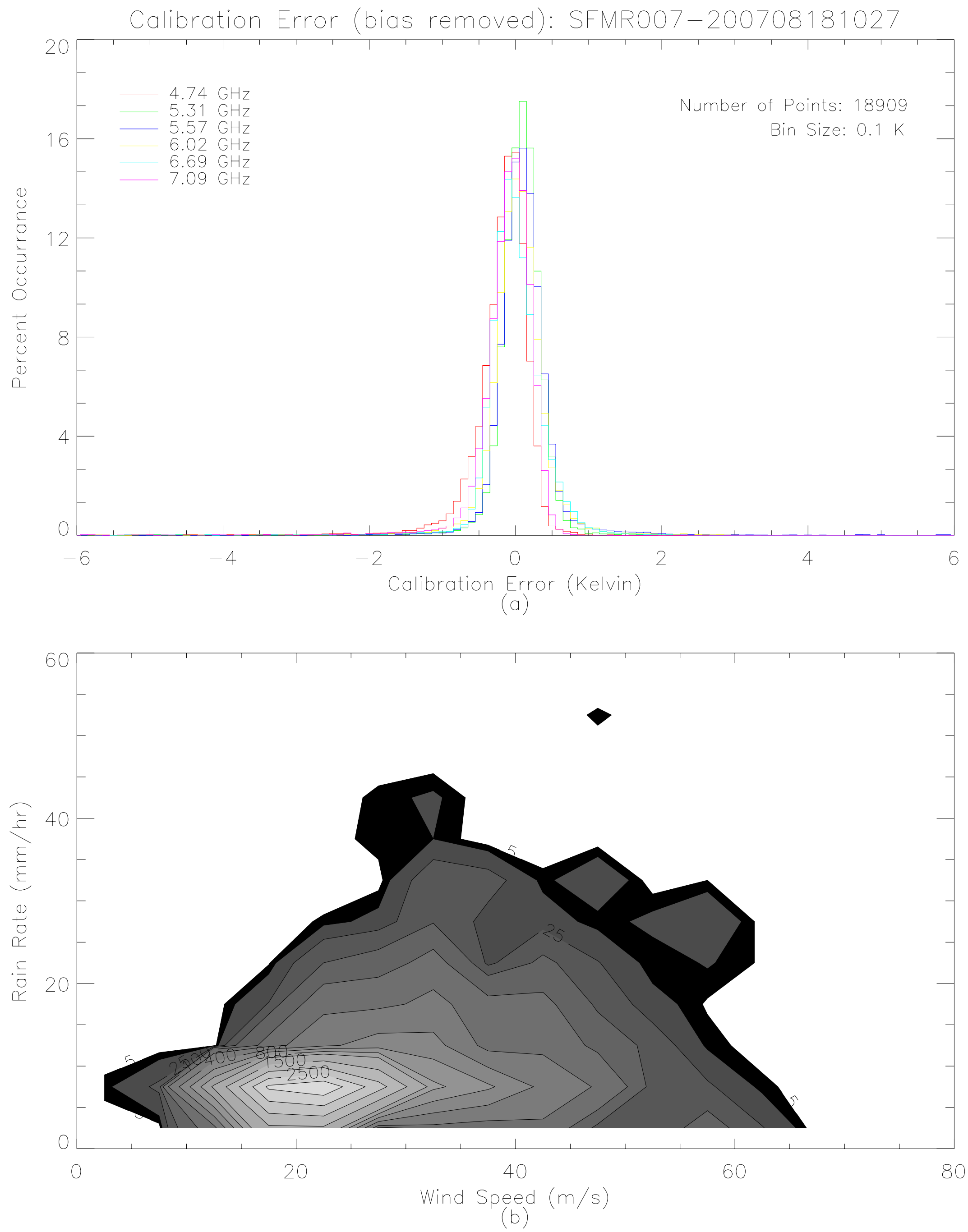

Figure 11 Same as Figure 9except the mean errors calculated from the 16 August 2007 flight are used to correct the measurements. The errors now have a zero mean for all channels.

3 HRD – Work Performed & Results Obtained

The following work and results were obtained by HRD as part of this effort:

- HRD performed initial evaluations of SFMR winds from Air Force C-130 aircraft.

- Working with ProSensing, HRD identified real-time code error in which first-guess solutions were improperly initialized. This problem caused the retrieval algorithm to fail in extreme winds in Hurricane Dean. This error has since been fixed.

- Analyzed surface wind retrievals in Hurricane Humberto (09/13). Real-time max. surface winds were estimated to be ~100 kts. However, a check of the brightness temperature data indicated that RFI contamination of one channel caused this spurious measurement. Post-processing of data indicated a max. wind of ~85 kts. It is apparent that the real-time QC procedure (which rejects bad TB data) used on the NOAA P-3 aircraft has not been implemented on the Air Force aircraft.

- Analyzed extreme wind measurements from NOAA P-3 SFMR data in Hurricane Felix. A max wind estimate of ~165 kts appears to be of good quality. Several measurements were in fact rejected in real time due to the very high TBs. This was due to the crude landmasking which attributed TB>270 K as being from land. Post-processing of this data with a higher threshold yielded clean retrievals through the eyewall.

4 Summary of Errors in Recon Data Files

Below is a summary of errors and notes that Remote Sensing Solution compiled as the real-time processing code was developed.

4.1 REPNT3

- Data files for flights during Felix have incorrect month. Month is given as “08” in file names. Felix took place in September. There is no date field within the data and therefore no way to automatically check the dates of the files without knowing when specific storms occurred. Similar discrepancies in the day of month value could easily go undetected.
- At least one data file for Noel have an incorrect month. Month is given as “11” when it should have been “10”. In combination with a day value of “31” this causes the additional problem of causing date calculation errors since November 31 is an invalid date.
- A change in file naming conventions causes files to be listed and read in alphabetical rather than chronological order. This requires the data to be sorted by date after all files are parsed. Prior to 8/1/2007 file were named REPNT3.YYYYMMDDHHMM.txt. Following 8/1/2007 files were named REPNT3-K???. YYYYMMDDHHMM.txt causing files to be listed and read in the order of the values contained in “???”.
- The octant of the globe field is consistently invalid. This value is needed for determining the sign values of latitude and longitude since they are not otherwise given. Since the octant of the globe value contained in the file is incorrect, the sign values were hard-coded so that longitude is always negative and latitude is always positive. The octant of the globe value contained in the files is consistently given as 7 when the value for these storms should usually be 0 or 1.
- The data for the Nationally Developed Codes (tag value 62626) contains unpredictable lines breaks. The data for this tag is spread over several lines in the data files. It appears that the lines break at a certain length (approximately 65 characters), without regard for the data. This causes data values and data tags to be split across multiple lines. This precludes the ability to search for a specific token or assume that a data value that follows a tag is complete. The number of lines taken by the data also varies. Therefore, the parsing code must look ahead across several lines to find the next data tag. Then it must strip off the line feeds and concatenate all of the lines that make up the data in order to ensure that all of the fields in the data are complete.
- Data files for training flights contain unpredictable patterns of incorrect, invalid, or incomplete data. There appear to be no rules regarding what can be inserted into training data files. As a result, no attempt is made to parse known training flight data files. These files are excluded by

searching for storm names “TRAIN”, “TR” or “WXWXA”. The limitation on this logic is that any storm name can be inserted into these files. If future training files contain storm names other than those listed, it is likely that the script will encounter exceptions when attempting to process the file.

- Files contain missing, invalid, or inconsistent data. In some cases data lines do not contain all of the fields expected. In some cases entire sections of data are missing. There are sometimes random invalid characters, such as equal signs contained within the data.
- Some files for Noel contain a storm name of “NOEL” and other have a storm name of “NOEL1”.
- Some files contain duplicated data. Generally, in these cases all of the data has been written to the file twice. The second instance of the data is ignored.
- Documentation of Nationally Developed Codes (62626) shows MBL WIND data field as “dddff” when it should be “ddfff”.

4.2 SFMR Data Files

- Data is not reported consistently at 1 Hz. Each record is time stamped with 1 second granularity. However, the time stamps do not ramp consistently in 1 second increments. There are often duplicates and/or time gaps. Many times these occur in unison. When duplicate timestamps are encountered, the parser must determine if there was a time gap before the duplicates or if one follows. If there is a time gap, followed by duplicate time stamps, the first of the duplicates is assumed to be the data that belongs in the preceding time gap. If the duplicates were not preceded by a time gap, but are followed by one, the second duplicate time stamp data is assumed to belong in the time gap. When duplicate time stamps occur without any preceding or trailing time gaps, the second set of data values overwrite the first.
- There is no identification of the storm contained in the file or the file name. This slows the collocation process with Dropsonde data because all files that match on date must be opened and searched for collocated data. This also means that there can be several SFMR files for any given storm, unlike the REPNT3 and URNT15 data which are consolidated into output files based on the storm and given file name that reflect the name of the storm, making it easier to find storm specific data.
- Some files contain large gaps of several seconds in the data without any duplicate data to extrapolate into the output file.
- The SFMR serial number provided in the Retrieved Values of Wind Speed and Rain Rate (R-record) is invalid at the beginning of the file. It takes several seconds before this value becomes valid which forces to parser to read into the file until finding a valid value.
- Some data files contain retrieval data (R-records) without matching Aux Info (A-records). Since the A-records contain information about aircraft

location, the R-record information is not useful without the matching A-records.

- Data collected for Felix contains bursts of data followed by large time gaps throughout the file. Also, there are no K-records (which contain the brightness temperature information) for the entire storm.

4.3 Center Fix Data Files

- Some of the entries contain no latitude or longitude. In some cases the only valid data appears to be the timestamp.
- At least one file contains a TD number of 90 which appears to be invalid.
- Some files begin with data lines that contain invalid dates. For example, dates from 2006 were observed in files containing 2007 storm data.