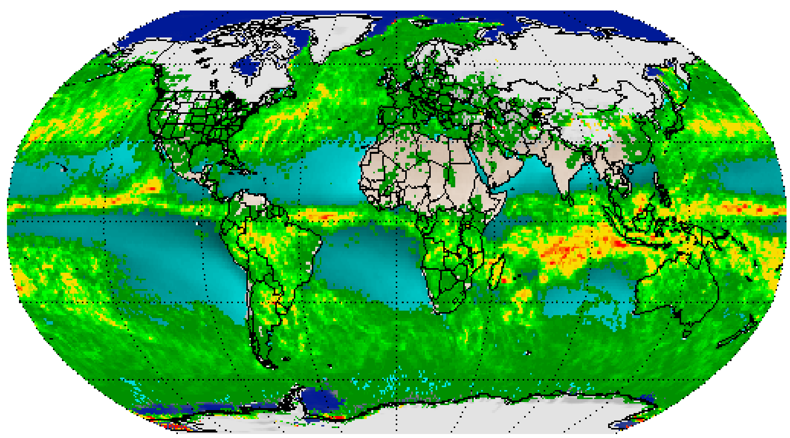
Microwave Integrated Retrieval System (MIRS)

System Description Document

September 2014



Prepared for:
U.S. Department of Commerce
National Oceanic and Atmospheric Administration (NOAA)
National Environmental Satellite, Data, and Information Service (NESDIS)
Center for Satellite Applications and Research (STAR)

Prepared by:
Xiwu Zhang, *NOAA/NESDIS/STAR*,
Christopher Grassotti, *AER, Inc.*,
Kevin Garrett, *Riverside Technology Inc.*,
Tanvir Islam, *CIRA, Colorado State University*

Version 1.11

Microwave Integrated Retrieval System (MIRS)

System Description Document (SDD)

September 2014

Prepared for:

U.S. Department of Commerce
National Oceanic and Atmospheric Administration (NOAA)
National Environmental Satellite, Data, and Information Service (NESDIS)

Approval Page

Document Numbers: 1 NOAA/NESDIS/STAR MIRS-SDD-01	Approved: September 2014

Document Title Block:

Microwave Integrated Retrieval System (MIRS)

System Description Document (SDD)

PROGRAM: MIRS	**DOCUMENT RELEASE DATE**: September 30, 2014
APPROVALS	
GROUP: NOAANESDIS/STAR Date NAME: Xiwu Zhang, Science Lead	GROUP: NOAA/NESDIS/STAR Date NAME: Sid Boukabara, Oversight Panel Chair
GROUP: NOAA/NESDIS/OSPO Date NAME: Limin Zhao, Oversight Panel member	GROUP: NOAA/NESDIS/STAR Date NAME: Ralph Ferraro, Oversight Panel member
CM RELEASE APPROVAL: Date NAME: Xiwu Zhang, Configuration Manager	

Version Description Record

DOCUMENT TITLE:
Microwave Integrated Retrieval System (MIRS) System Description Document (SDD)

DOCUMENT CHANGE HISTORY			
DCN No.	Revision	Date	Description of Change
	1.0	January 31st, 2007	Initial Release of MIRS System Description Document
	1.1	April 26th, 2007	Kleespies Initial Review
	1.2	June 15th, 2007	Second Release of MIRS System Description Document
	1.3	December 15th, 2007	Third release of MIRS System Description Document
	1.4	June 15th, 2008	Fourth release of MIRS System Description Document
	1.5	December 15th, 2008	Fifth release of MIRS System Description Document
	1.6	June 15th, 2009	Sixth release of MIRS System Description Document
	1.7	June 15th, 2010	Seventh release of MIRS System Description Document
	1.8	November 9th, 2012	Eighth release of MIRS System Description Document
	1.9	January 8th, 2013	Ninth release of MIRS System Description Document
	1.10	March 31, 2014	Tenth release of MIRS System Description Document
	1.11	September, 2014	Eleventh release of MIRS System Description Document

Preface

This document comprises the National Oceanic and Atmospheric Administration (NOAA) National Environmental Satellite, Data, and Information Service (NESDIS) publication of the Microwave Integrated Retrieval System (MIRS) *System Description Document (SDD)*.

This System Description Document (SDD) describes the key components of the Microwave Integrated Retrieval System (MIRS) at a relatively high level but provides a level of detail necessary to perform system administrative and reactive maintenance functions. Contents of the SDD includes subsystem functions and capabilities, key interfaces, system inputs and outputs, procedures for the scheduling of operational jobs, and monitoring and maintenance procedures.

The document is controlled under the configuration management tool in NOAA/NESDIS/STAR and will be updated as required.

Table of Contents

Appendices

List of Figures

List of Tables

Section 1.0 Introduction

1.1 Document Overview

This document is organized as follows:

Section 1.0 – provides a summary of MIRS scientific concepts.

Section 2.0 – provides an overview of the MIRS system. The overview includes description of MIRS applications and products and component processes.

Section 3.0 – provides details on MIRS scientific and system design characteristics. It also gives program and execution details for each individual application, the script that controls their execution sequence and the configuration file associated with it. It also describes the naming convention of the output figures.

Section 4.0 – provides a brief description of the inputs and outputs.

Section 5.0 – provides a description of the operational procedures for running MIRS.

Section 6.0 – Gives information on hardware and software requirements. It also provides an estimate of the timing efficiency of the system.

Section 7.0 – Describes the procedures to perform scientific maintenance and updates. In particular, it describes what is required to extend MIRS to additional sensors. It also describes the quality assurance and performance monitoring system associated with MIRS. Imbedded in this section is a number of ways on how the user can interact with MIRS.

Appendix A – defines each acronym, as necessary.

Appendix B – lists the references.

Appendix C – lists the main programs and their functions.

Appendix D – lists the main library modules.

1.2 Scientific Background

The Microwave Integrated Retrieval System (MIRS) was developed by the NOAA/NESDIS Center for Satellite Application and Research (STAR) as a major upgrade to the existing suite of microwave retrieval algorithms called the Microwave Surface and Precipitation Product System (MSPPS). MSPPS lacks profiling capability and is specific to a single instrument, the AMSU. Another objective for developing MIRS was to provide retrievals in all-weather and over all-surface conditions with the immediate benefits of extending the spatial coverage to critical areas such as actives regions and using non-exploited measurements such as those made by surface-sensitive channels for temperature sounding

MIRS is applicable to the existing and future microwave sensors. It is currently being applied operationally to the NOAA-18, NOAA-19, METOP-A and METOP-B AMSU/MHS suite and the DMSP-F16 and F18 SSMI/S sensors, to NPOESS Prepatory Project (NPP) Advanced Technology Microwave Sounder (ATMS) data and (MT) SAPHIR measurements, as well as to TRMM TMI, in research mode. It could also be a system for Infrared (IR) sensors onboard JPSS and GOES-R. Having one platform for a multitude of sensors is scientifically sound because the radiative transfer physics involved is by and large the same and the mathematical basis for the inverse problem is identical. The practical advantages of having one single system for a multitude of sensors are numerous. They include among others, the time and cost savings related to generating a retrieval algorithm for a new sensor, the optimal use of the information content and the consistent treatment of time series of satellite data for long-term trend monitoring and climate studies. MIRS is coupled with the Joint Center for Satellite Data Assimilation (JCSDA) Community Radiative Transfer Model (CRTM) which is valid in both microwave and infrared spectral regions, in clear, cloudy and precipitating conditions and over all surface types.

The MIRS is an iterative, physically-based retrieval system based on the one Dimensional Variational Retrieval (1DVAR). It optimally extracts the information content present in the measurements. Retrievals are performed in a consistent fashion, with the end result being a set of consistent geophysical parameters, or Environmental Data Records (EDRs) that are computed simultaneously and, when used as inputs to the forward model, should nominally fit the measured radiances to within the noise level. The retrieval is performed in a reduced space by using Empirical Orthogonal Function (EOF) decomposition to allow a more stable inversion, a faster retrieval and to avoid the null space. The number of selected principal components is tuned for each instrument of interest.

The 1DVAR physical principle is to minimize a two-source cost function, composed of the departure of the simulated radiances from the actual measurements and the departure of the retrieved parameters from their respective backgrounds. In the retrieval scheme used by MIRS, the departure from the measured radiances is normalized by the noise level (NEDT) impacting the measurements and the uncertainty in the forward modeling, making it possible to use the signal of a particular channel when the geophysical signature (through the derivative) is stronger than the noise (leading to a useful signal-to-noise level), and some other times dismiss the same channel when the signal in question is within the uncertainty/noise level. The departure from the background is also scaled by the uncertainty placed on the background. The source of these backgrounds could vary from simple climatology (loose background errors) to Numerical Weather Prediction (NWP) forecast fields (tight errors in the temperature background). Figure 1 summarizes the concept of MIRS.

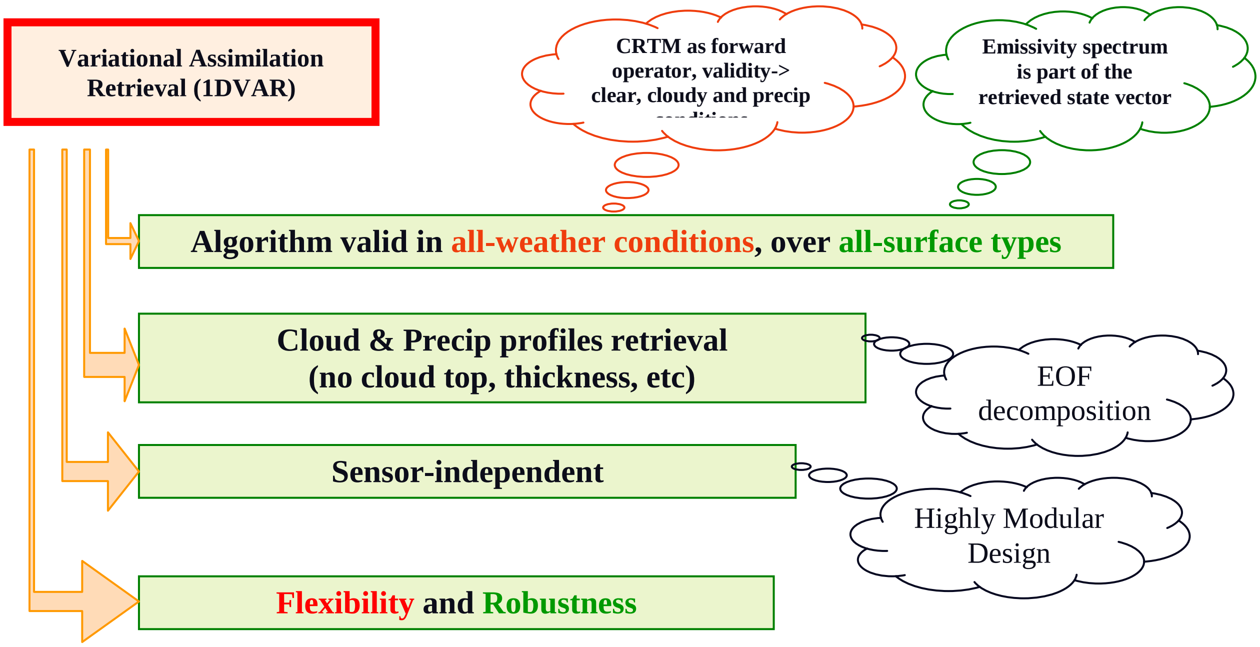


Figure 1. Concept characteristics of MIRS

Section 2.0 System Overview

This section provides an overview of the MIRS system. First, a summary of MIRS products is provided, followed by a description of MIRS component processes and applications.

2.1 MIRS Products

The list of products generated by MIRS is summarized in Table 1. Also included is the list of heritage products, e.g., generated by MSPPS. “Standard” products are labeled those that have been routinely retrieved and for which extensive validation and testing has been done. “New” products are not routinely retrieved and extensively tested. Listed are the MIRS 1DVAR and derived products. MIRS 1DVAR products include the parameters that are part of the retrieval state vector. MIRS derived products are those generated using MIRS 1DVAR parameters as inputs and a post-processing procedure, e.g., a simple vertical integration of retrieved water vapor profile for computing Total Precipitable Water (TPW), or a new algorithm, e.g., for the estimation of Snow Water Equivalent (SWE) from the retrieved surface emissivities. Note in the Table MIRS profiling capability, which is lacking in MSPPS. Note also the list of MIRS new products (1DVAR and derived) which would require new validation efforts.

MIRS Heritage Products	MIRS 1DVAR Products	MIRS derived products
Standard Products		
Total Precipitable Water (TPW)	Atmospheric temperature profile (T)	Q-based TPW
Cloud Liquid Water (CLW)	Atmospheric humidity profile (Q)	NPCP-based CLW
Land Surface Temperature (LST)	Land Surface Temperature (LST)	IGP-based IWP
Emissivity at certain window channels	Emissivity vector (Em)	IGP-based Rain Water Path (RWP)
Rain Rate (RR)		Em-based SIC
Ice Water Path (IWP)		Em-based SCE
Snow Water Equivalent (SWE)		Em-Based Surface Type
Sea Ice Concentration (SIC)		RP&T-based RR
Snow Cover Extent (SCE)		Em-based SWE
New Products		
Snow Fall Rate (SFR)	Ice Surface Temperature (IST)	Em-based Multi-Year SIC
	Snow Surface Temperature (SST)	Em-based First-Year SIC
	Non-precipitating Cloud Profile (NPCP)	Em-based Snow Effective Grain Size
	Rain Profile (RP)	RP&T-based SFR
	Ice/Graupel Profile (IGP)	

Table 1. List of the heritage, advanced and derived MIRS products

Because the same algorithm is used consistently across the platforms (different channels spectra, polar and geostationary orbits), the time series of these retrievals will thus be self consistent, making the resulting climate data records free of jumps due to changes in retrieval algorithms.

2.2 MIRS Processing Components

This section provides a description of MIRS processing components. *Top-to-bottom* descriptions are provided: first, the high-level blocks are described, followed by individual and more detailed descriptions of each block. Figure 2 presents the basic conceptual diagram of the high-level MIRS blocks. The radiance processing block interfaces with the inputs to MIRS (the sensor data files or the raw data) and the inversion processing block. It generates ready-to-invert radiances or brightness temperatures used as inputs to the inversion processing. The inversion processing generates main MIRS retrieval outputs.

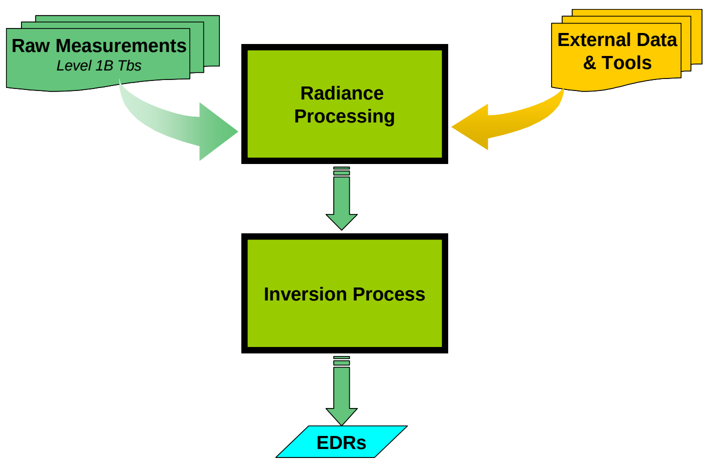


Figure 2. Overall conceptual diagram of the MIRS retrieval concept.

2.2.1 **Radiance Processing**

The radiance processing is described in detail in Figure 3. The figure shows the chain of applications for converting raw sensor data, e.g. AMSU and MHS from the decoded raw sensor data to ready-to invert radiances. First, raw sensor data are converted into MIRS internal format (`rdr2tdr`). Next, MIRS internal format sensor data are antenna-pattern corrected (`tdr2sdr`), footprint-matched (`fm`) and bias-corrected. This block also generates the noise files (NEDT) used for computing instrument noise matrix E (only for AMSU-MHS and ATMS). The noise values are assessed as part of the performance monitoring. This block collocates gridded NPW fields with satellite measurements. These collocated files are used as inputs to the forward model to simulate brightness temperatures and compare them with measurements for assessment of instrumental bias. This assessed bias is channel and scan position dependent. Along with the bias file, the radiative transfer model (RTM) uncertainty is also assessed based on the standard deviation between the measured and simulated radiances.

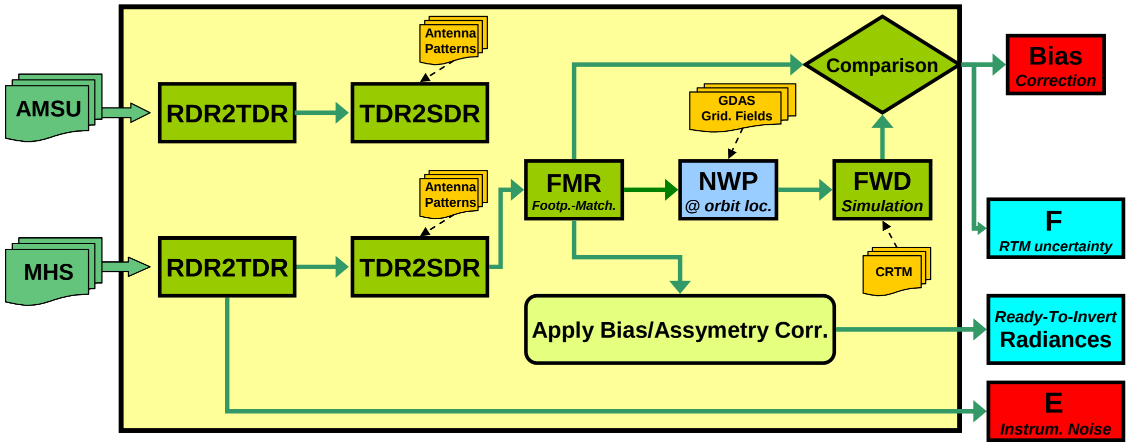


Figure 3. Conceptual diagram of the radiance processing block and its component processes

2.2.1.1 Antenna Pattern Correction

Currently, this application is a simple placeholder. No antenna pattern is applied at this time. There is therefore no difference between the TDR and the SDR files in MIRS. The first order effect of not accounting for the antenna pattern is a bias in the obtained field of view measurements. This first order effect is removed through the bias removal procedure described below. A second order effect is due to the scene interaction with the antenna pattern which increases the standard deviation of the difference between the true brightness temperature and the actual one impacted by the side lobes of the antenna. This effect is currently not removed in MIRS.

2.2.1.2 Footprint Matching

Footprint matching is the procedure that ensures that all channels for the retrieval view the same location on the Earth. The footprint matching is a sensor-specific application because every sensor has its own characteristics and viewing geometry.

For the NOAA-18, NOAA-19, and Metop-A satellites, footprint matching is performed between the AMSU and MHS, each containing a different footprint size for their suite of channels. In this application, a simple 3x3 footprint averaging technique is applied to each MHS frequency channel to match the footprint size of the AMSU-A channels, thus producing a set of 20 channels (AMSU and MHS together) with the same (AMSU-A) footprint. This technique is the nominal mode of operation. In an alternative mode (high-resolution mode), the AMSU measurements are assumed valid across the 3x3 MHS footprints, in which case a 20-channel measurement at the MHS resolution is obtained. This is the default operational mode for Metop-B processing.

For the F16, F17, and F18 satellites, sets of SSMI/S channels have four different resolutions. The lowest resolution is the Upper Atmospheric Sounding (UAS), followed by the Lower Atmospheric Sounding (LAS), environmental (ENV), and the highest resolution, imaging (IMG). For example, footprint matching at the lowest SSMI/S resolution is the average of UAS channels across 6 scan lines (1x6) to attain a more circular footprint geometry, with more complicated averaging/subsampling done for footpring matching at higher resolution ENV and IMG resolutions. For the NPP satellite, the ATMS instrument produces 96 footprints per scan line and all channels are bore-sighted but with different beam widths.

For ATMS, the footprint averaging/resampling algorithm from the EUMETSAT ATOVS and AVHRR Preprocessing Package (AAPP) is used. As currently implemented, the lower frequency channels 1 and 2 (23 and 31 GHz) which have an original measurement beam width of 5.2 degrees are resampled to an equivalent beam width of 3.3 degrees. Channels 3 through 16 (50 through 88 GHz) which have a beam width of 2.2 degrees, and channels 17 through 22 (165 through 183 GHz) which have a beam width of 1.1 degrees, are not averaged or resampled.

For Megha-Tropiques/SAPHIR, the footprint matching step can be run at three different resolutions and two different sampling modes. The spatial resolutions for SAPHIR footprint matching are high resolution, low resolution, and coarse resolution, referred to as HR, LR, and CR, respectively. At HR or full resolution, no modification of the original resolution is done and the data are processed at a resolution of 10 km (nadir) using all measurements along a scan line (130 per scan). In LR, only half the number of measurements are used (65 per scan), and in CR, one quarter of the measurements are used (23 per scan). The footprint matching sampling modes are either thinning, or averaging. For LR and CR in thinning mode (the default), the measurements are sub-sampled along the scan to obtain the reduced number of samples, and the original spatial resolution of each measurement is retained. In averaging mode, the individual FOVs are averaged along the scan to obtain the reduced number of samples, thereby reducing the effective spatial resolution of each measurement. Note that in HR footprint matching, the sampling mode has no effect since all the data are used at the full resolution.

2.2.1.3 Bias Removal

The bias removal is a procedure that applies a pre-computed bias offset (generated in the radiance processing block) to the radiance measurements that generates bias-free radiances used as inputs to inversion processing. Bias removal is a generic term to define the removal of systematic differences between the forward operator and the actual sensor measurements. The MIRS accommodates several techniques to compute (and apply) the bias, which is as follows:

- Offset bias removal. A mean value of the differences between the simulations and the measurements is computed for each scan position for every channel and stored. The application of the bias is also simple. It is done by applying an additive term to the measurements.
- Slope/Intercept Correction. Instead of computing the mean bias, the slope and intercept are computed from linear regression of the measurements against the simulations. These slope/intercept pairs are then applied to the brightness temperatures.
- Histogram Adjustment. This methodology removes bias by adjusting the histogram of the brightness temperature difference between simulated and actual measurements to make it centered around zero. This process reduces the sensitivity of measurements to clouds, precipitation and coastal contamination.

The bias removal method used by default in MIRS is the histogram adjustment.

2.2.2 **Inversion Processing**

Figure 4 describes the conceptual organization of the inversion processing. As shown, ready-to-invert radiances generated from the radiance processing are used as inputs to the heritage algorithms and to the advanced algorithm (1DVAR). Heritage algorithms consist of physical non-iterative and empirical microwave algorithms. The products generated by the heritage algorithms could optionally be used as first guesses for 1DVAR initialization. Note that the use of these first guesses does not violate the mathematical requirement that the background errors and the instrumental/RTM errors be uncorrelated. This required condition is not violated because the heritage algorithms are not used as background constraints.

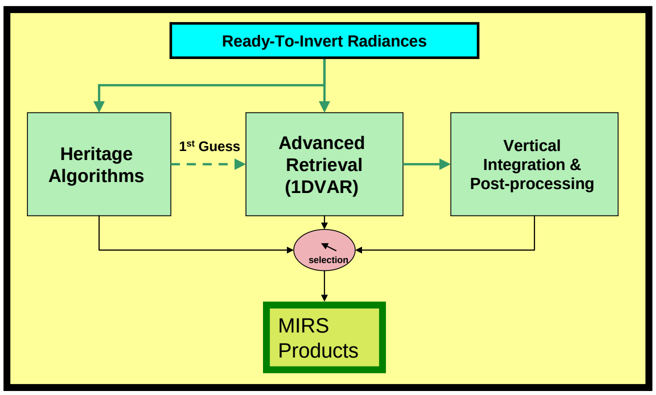


Figure 4. High-level conceptual diagram of the inversion processing block showing the MIRS concept of merging heritage, advanced and derived products

The outputs from 1DVAR are a set of advanced products, consisting of temperature, humidity, non-precipitating cloud and hydrometeors profiles and the surface temperature and emissivities. These main products are post-processed (new algorithms or vertical integration of MIRS retrieved parameters) to produce MIRS derived products. So, there are three types of MIRS products: (1) heritage products, (2) advanced products and (3) derived products. A mechanism is then implemented to select which one of these products is selected to be part of the final MIRS products (see Figure 4). This selection is based mainly on the degree of confidence we have (and the extent of the validation performed) in each of the products. This is why in the first phases of MIRS, the heritage algorithm (MSPPS) will continue to be producing the main MIRS products, and the advanced algorithm will add more products and replace some others, in an incremental manner.

2.2.2.1 Heritage Algorithms

The heritage algorithms (Figure 5) include the existing regression- and physically-based non-iterative algorithms such as MSPPS algorithms based on AMSU-MHS, or other sensor-specific algorithms such as those based on DMSP SSMI/S and WINDSAT. Main purpose of these algorithms is their application in the MIRS 1DVAR processing as first guesses. Additionally, new regression algorithms, referred to as "locally developed algorithms", are developed for the retrieval of a wider range of EDRs that can be integrated more efficiently and utilized in MIRS 1DVAR processing as first guesses. These local algorithms have been developed off-line from brightness temperature data collocated with geophysical parameters.

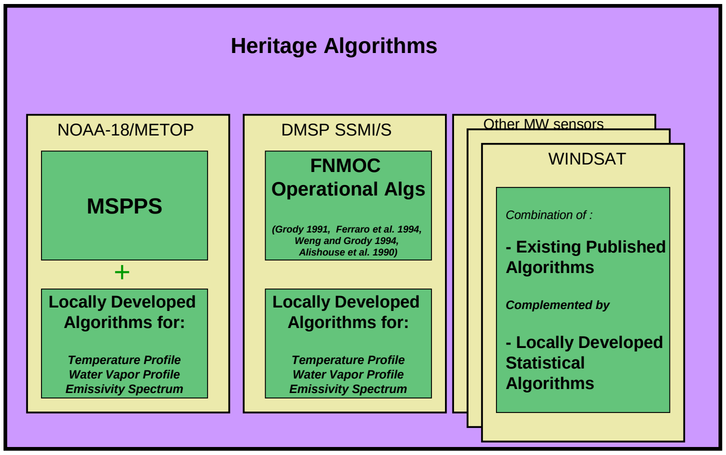


Figure 5. MIRS heritage algorithms. For NOAA-18 and METOP AMSU-MHS, the heritage algorithms are obtained from MSPPS.

2.2.2.2 Advanced Algorithm (*1DVAR*)

The 1DVAR approach to retrievals is referred to as “advanced” as being more optimal compared to heritage algorithms and incorporating a sophisticated forward operator, which the heritage algorithms do not have, that fully assimilates sensor radiance measurements. It is schematically represented in Figure 6. The forward operator is based on the Community Radiative Transfer Model (CRTM) developed by the Joint Center for Satellite Data Assimilation (JCSDA). The CRTM produces the simulated radiances Y as well as the Jacobians K. The iterative loop is ended when convergence is reached. The following unconstrained cost function is used as a metric for deciding if convergence has been reached:

$$\chi^2 = [Y^m - Y(X)]^T E^{-1} [Y^m - Y(X)]$$

Convergence is reached when $\chi^2 <= 1.0$. The iterative loop is also ended if the convergence criterion is not met within seven iterations.

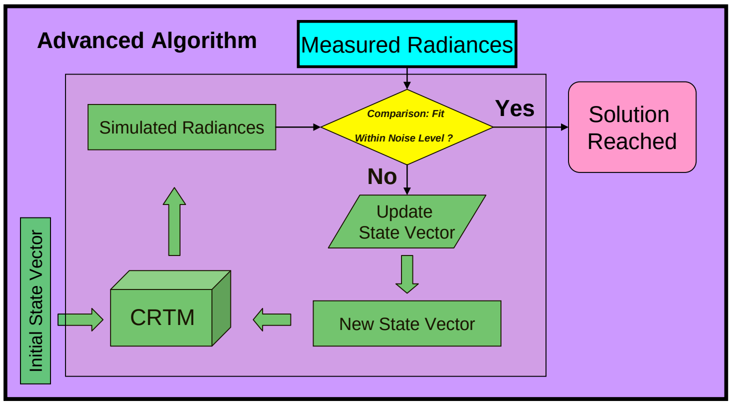


Figure 6. General description of the 1DVAR retrieval iterative system. The Initial state vector (or first guess) starts the iterations, the update of the solution takes place at each iteration depending on the local derivatives, simulated brightness temperatures, etc. The solution is reached when the final simulations fit the measurements within the noise level. CRTM is used to generate the simulated measurements.

2.2.2.3 Vertical Integration and Post-Processing

The products generated by 1DVAR are utilized in a post processing stage further processed to generate derived products, as shown in Figure 7. This post-processing can take the form of a simple vertical integration e.g. to derive TPW by vertically integrating the water vapor profile Q, or an algorithm, e.g., to derive the surface properties of snow cover and sea ice based on the 1DVAR retrieved parameters of surface emissivities and skin temperature.

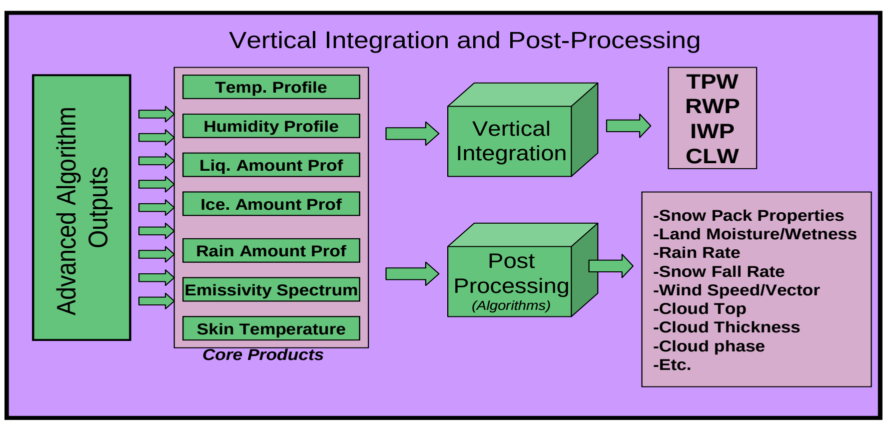


Figure 7. Schematic diagram of the MIRS main and derived products.

To summarize, the derived products generated by vertically integrating the corresponding MIRS core retrieved products are: total precipitable water (TPW, from the retrieved water vapor mixing ratio profile), rain water path (RWP, from the retrieved rain water profile), ice water path (IWP, from the retrieved ice water profile), and cloud liquid water (CLW, from the retrieved cloud water profile).

Also related to the hydrometeor retrievals described above is the retrieval of surface rainfall rate (RR) which is derived from a post-processing algorithm which operates on the vertically integrated products CLW, RWP, and IWP. The algorithm to derive rain rate takes advantage of the physical relationship found between atmospheric hydrometeor amounts and surface rain rate. As shown in the equations below, the MIRS rain rate algorithm uses a multi-linear regression approach that requires integrated CLW, IWP, and RWP (in mm), and a set of regression coefficients corresponding to each hydrometeor in order to retrieve the instantaneous rain rate in mm/hr over ocean and land.

$$RR_{ocean} = a_0 + a_1CLW + a_2RWP + a_3IWP$$

$$RR_{land} = a_0 + a_2RWP + a_3IWP$$

Where RR is the estimated surface rain rate, given in mm/hr, and a_{0i}, a_{1i}, a_{2i} and a_{3i} are the regression coefficients. The regression coefficients are static components in the algorithm that have been determined based on an off-line training using collocated sets of rainfall rate and hydrometeor products from both the Penn State University and the National Center for Atmospheric Research Mesoscale Model (MM5) data for the ocean case and from the Operational Microwave Surface and Precipitation System (MSPPS) for the land case.

Post-processing of the retrieved emissivity spectrum relies on the development of an offline-computed catalog of emissivity spectra (indicated schematically in Figure 8 below), for a multitude of values of the parameters to be derived. The post-processing stage is then a simple look-up-table procedure that searches for the catalog pre-computed value that corresponds to a spectrum that matches closely with the retrieved one.

The MIRS derived products generated by post-processing the core retrieved emissivity spectrum and skin temperature are: snow water equivalent (SWE), and total sea ice concentration (SIC). The SWE product uses the retrieved surface emissivities as inputs and a catalog of surface emissivities and snow pack properties derived off-line from a one-layer Dense Media Radiative Transfer snow emissivity model. The retrieved MIRS emissivity spectra are compared with those from the catalog to find the closest match. A bi-product of the SWE generation is also an estimation of the effective snow grain size. The SIC product uses MIRS retrieved surface emissivities and skin temperature as inputs and a catalog of surface emissivities and ice fractions derived off-line from emissivity spectra of pure water and ice surface types. The retrieved MIRS emissivity spectra are compared with those from the catalog to find the closest match and compute SIC. A bi-product of the SWE product generation is also sea ice type, which can be either first year ice, or multiyear ice.

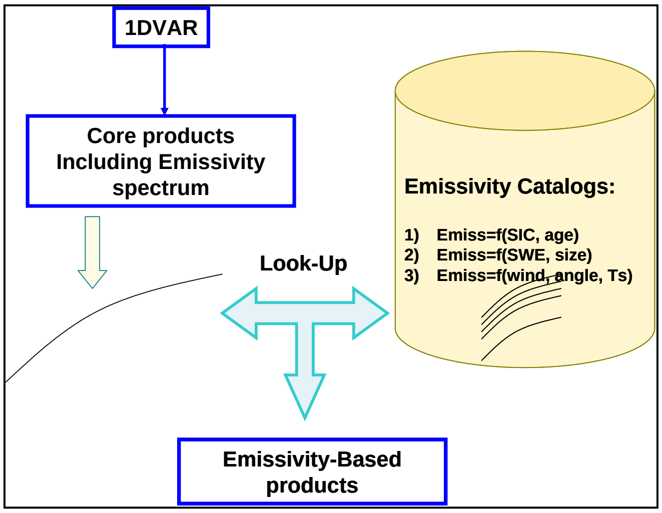


Figure 8. Schematic representation of emissivity spectrum post-processing

2.2.3 **Conversion to User-Driven Formats**

Work in progress

2.2.4 **Area-Of-Interest (AOI) Segmentation**

MIRS provides capability for Area-Of-Interest (AOI) segmentation, where processing is restricted to a user-defined geographic region. This is done by specifying the minimum and maximum latitude and longitude for the desired AOI, as well as setting the “geoLimit” flag to 1 in the MIRS Paths and Configuration File (PCF), which is described in Section 3.3. MIRS radiance processing will proceed normally until the footprint matching step. The outputs from the footprint matching step are then only the Field-of-Views (FOVs) for the specified AOI. The option is available regardless of the selected sensor resolution (e.g. low resolution or high resolution).

2.2.5 **Level-III Compositing**

Work in progress

Section 3.0 System Description

3.1 Design Characteristics

3.1.1 General Characteristics

The MIRS system is designed to be a flexible retrieval and assimilation tool and is suited for applications in the microwave as well as the infrared, although it has been applied only to microwave frequencies thus far. It allows the user to select which channels to use for a particular retrieval and which parameters to be retrieved. It performs the retrieval in a consistent manner where the state vector is retrieved to fit the measured radiances. To allow a stable retrieval, the retrieval space is projected into a space consisting of a limited number of most relevant eigenvalues. These features (reduced space retrieval, integrated approach and flexibility), coupled with the advanced radiative transfer model (CRTM) used as the forward operator, allow MIRS to be a cutting edge system, readily applicable to current and future sensors, both sounders and imagers or any combination thereof.

3.1.2 Scientific Characteristics

Table 2 is a succinct summary of the major scientific characteristics of the MIRS algorithm.

	Comments
Hooked to CRTM	- The RTM component in MIRS is a separate module. In this way, the CRTM or alternatively another model can be used if deemed necessary. This allows MIRS to benefit from advances in RTM science at no or low cost.
Multiple scattering	- CRTM operates in clear sky and scattering mode depending on the inputs, which saves time. This allows the retrieval (sounding and imaging) to take place in cloud and rainy conditions. This is a major advance compared to the existing retrieval algorithms.
Robust external information/1st Guess	- The first guess can come from microwave regression models, from climatology a-priori statistics or from a combination of both.
Retrieval-Quality outputs	- The convergence metrics is a first-degree quality parameter. In addition, the following are produced by MIRS: - Uncertainty matrix, $S = S_a - S_a.K^t(KS_aK^t+S_y)^{-1}KS_a$ - Average Kernel. This gives an indication if the retrieval relied more heavily on the background (if close to 0) or on the radiance itself (if close to 1). This could be a useful feature for the data assimilation impact studies. - Contribution functions. These functions are indicators of the ill-conditioning of the retrieval problem. The bigger they are, the more noise amplification is happening.
All-surface retrievals	- Retrieval of spectral emissivities (with constraints built offline) over all surface types. This makes the retrieval independent of the surface background and allows spectral variability. The emissivity->wind vector over ocean, or emissivity->vegetation/snow cover, etc… can be handled externally (post-processing); using MIRS itself is a possibility.
Retrieval in reduced space	- The retrieval is generally ill-conditioned. Having external constraints is not enough to well-condition the problem. Reducing the number of parameters to retrieve is achieved by projecting the retrieval space into eigenvector space.
Non-linearities	- Generally, iterative retrieval methods rely on the assumption that the problem is locally linear. In the linear case, the variational solution gives the exact solution in only one iteration. In the non-linear case, the model moves one step at a time, each time assuming some sort of linearity. This is sometimes complicated because either the convergence could be very slow or if we overshoot, the solution could oscillate between states (non-convergence).
Convergence	- The convergence is controlled by the ratio of the radiance residuals to the noise level of the channels.
Noise	- Noise values are used in the estimation of the optimal solution and in the convergence criteria.

Table 2. Summary of the scientific characteristics of MIRS.

3.1.3 **System characteristics**

Table 3 presents the main characteristics of the MIRS system implementation.

Characteristic	**Comments**
System design and architecture	Modular design
Algorithm coding standards	- Strict adherence to Fortran 95 Standard *(no extension)* - BASH for scripts - All constants, coefficients, filenames, are passed via modules and USE statements - Software design based on encapsulation - Dynamic memory allocation used across MIRS *(allocatable arrays mainly and Pointers only when necessary)* - Interface via structures if possible *(more stable interface)* - Error-prone features banned *(GOTO, COMMON BLOCK, implicit array passing, implicit declaration, etc)*
Software configuration management standards	- Configuration management using Subversion - Pyramid-type makefile web used for compilation and to build all executables - Centralization of common utilities (across MIRS) - Instructions/Readme document provided - Standard naming convention adopted for outputs - Delivery Algorithm Package (DAP) compatible with NDE standard - One single tar file, accessible via anonymous. ftp
Error handling	- OSDPD standard (through opus_msg)
Software architecture	-Pyramidal. All libraries consolidated by applicability and encapsulated.
Directory system	-Highly flexible: controlled through Paths & Configuration Files (PCF)
Interaction and process control	a) High level, GUI-based tool called the MIRS Control Panel (MCP) b) Mid-level, where the user uses the Sequence Control Scripts (SCS) for an individual sensor and the associated Paths & Configuration File (PCF) to reconfigure directory system and control information flow c) the low level, in which case the user could go directly to the individual application (i.e. 1dvar, footprint matching, etc) and the associated control/input file (a namelist file for F95 applications and a similar controlling-parameters list file for IDL applications

Table 3. Summary of the design characteristics of MIRS

3.2 Description of MIRS Applications

A MIRS individual application represents the lowest level of interaction with the user. Higher-level interactions such as the Sequence Control Scripts (SCS) or the Graphical User Interface (GUI)-based MIRS Control Panel (MCP) are external value-added tools imbedded in MIRS that allow the user to reconfigure, control the sequence of and execute MIRS applications. These higher level interactions are described in greater detail in section 7.1.

An overview of the MIRS component processes and their linkages was presented in Section 2.0. This section provides further details on individual applications. A MIRS application consists of the program files and a control file. The program files reside in the "/src" top level directory. They consist of source code files (written in Fortran 95 and IDL), compiled files and the makefiles. The control files reside in the "/Data/ControlData" directory. The control file contains information on input parameters such as the location of input/output files needed to run the application. The control files can be constructed directly, or they can be constructed via the

SCS that resides in the "`/scripts`" top level directory. Note that the SCS can be generated automatically using the GUI Interface.

Table 4 provides a summary description of MIRS applications for the NOAA-18 AMSU-MHS sensor. The applications provided below are also applicable to other microwave sensors being processed such as NOAA-19, Metop-A and Metop-B, the DMSP F16 and F18 SSMI/S, NPP ATMS, and Megha-Tropiques SAPHIR.

MIRS Application	Program Files	Control File Name	Control File Content
RDR to TDR conversion *the first step in the generation of radiances from raw sensor data (level 1-b) to antenna temperature data (level 1-b internal format)*	Local directory: *"src/testbed/rdr2tdr/n18_amsua_mhs/amsua"* *"src/testbed/rdr2tdr/n18_amsua_mhs/mhs"* Source code file: *rdr2tdr_amsua.f90* *rdr2tdr_mhs.f90*	*n18*_amsua_rdr2tdr_*yyy_mm_dd*.in *n18*_mhs_rdr2tdr_*yyy_mm_dd*.in	- Name/directory of the input file that contains the list of rdr input files, - Directory of output (TDR) file - Name/directory of NEDT file - Name/directory of instrument configuration file, - Name/ldirectory of Warm Target file, - Number of orbits to process – set to a very high value - Name/Directory of log file
Merge NEDTs *Performs merging of instrumental noise files, e.g. for AMSU-A and MHS*	Local directory: *src/testbed/mergeNEDTofDiffInstr'* Source code file: *mergeNEDT.f90*	*n18_mergeNEDT_yyy_mm_dd.in*	- Name/directory of AMSU-A NEDT file - Name/directory of MHS NEDT file - Name/Directory of the merged NEDT file - Name/Directory of the log file
TDR to SDR conversion *Generation of brightness temperature data after antenna patterns have been applied (sdr)*	Local directory: *"src/testbed/tdr2sdr"* Source code file: *tdr2sdr.f90*	*n18*_amsua_tdr2sdr_*yyy_mm_dd*.in *n18*_mhs_tdr2sdr_*yyy_mm_dd*.in	- Name/directory of the file that contains the list of tdr input files, - Format of TDR file - Name/directory of antenna patterns file - Directory of output file - Name/Directory of log file - Number of orbits to process
Footprint matching *Generation of brightness temperature data having same spatial resolution across frequencies (fmsdr)*	Local directory: *"src/testbed/fm/n18_amsua_mhs"* Source code file: *Fm_n18.f90*	n18_amsua_mhs_fm_*yyy_mm_dd*.in_	- Name/directory of the file that contains amsua sdr files - Name/directory of the file that contains mhs sdr files - Directory of output (fmsdr) file - Name/Directory of log file - Name/Directory of QC file - Number of orbits to process, - FM type, number of scan , lines to skip, time collocation mode
Collocation of NWP with footprint matched radiance data *Spatial and temporal matching of geophysical NWP data (gdas or ecmw) with brightness temperature measurements*	Local directory: "*src/testbed/nwp"* Source code file: *colocNWPwRad.f90*	n18_amsua_mhs_colocNWPwRAD_yyy_mm_dd.in_gdas(ecmw)	- Name/Directory of input file containing the fmsdr files - Name/Directory of input file containing NWP files of atmospheric EDRs -Name/Directory of input file containing NWP files of surface EDRs - Name/Directory of output files - Nume/Directory of topography file - Name/Directory of geophysical covariance background file - Name/Directory of log file - Number of orbits to process - Sensor id (1 – for N18, 2 – for MetopA, and 3 – for F16) - NWP source (1 - for gdas and 2 - for ecmw) - Name/Directory of CRTM coefficients file - Name/Directory of CRTM cloud optical properties file - Name/Directory of bias file - Name/Directory of tuning file

Forward simulation on NWP data *Simulation of brightness temperatures from NWP geophysical data (GFS, GDAS or ECMWF) usingCRTM*	Local directory: "src/fwd" Source code file: fwd.f90	n18_cntrl_fwd_*yyy_mm_dd*.in_*_*gdas(ecmw)* ** integer value that denote orbit number for daily processing (usually from 0 to 15)*	- Name/Directory of collocated NWP files - Name/Directory of CRTM coefficients files - Name/Directory of Instr. Config. file - Nume/Directory of simulated radiance output files - Name/Directory of CRTM cloud optical properties file - Channel selection & print monitor mode - Nume/Directory of the noise file - Number of profiles to process - Name/Directory of log file
Bias computation *Comparison of the observed brightness temperatures with simulated brightness temperatures (from NWP analysis data) and the computation of brightness temperature biases*	Local directory: src/testbed/biasGenerAndMonit Source code file: Calib_generic_rad.pro	*n18_Inputs4BiasComputation_yyy_mm_dd*.in_gdas(ecmw)	- Name/Directory of the file listing the collocated NWP files - Name/Directory of the file listing the simulated radiance files from NWP data - Name/Directory of the file listing the observed radiance files (fmsdr) - Name/Directory of the computed bias correction file - Bias correction method - Name/Directory of the computed model error file - Name/Directory of the bias generation file in postscript format - Number of orbits to process
Chopping of fmsdr files *Orbital files are chopped into granules for faster processing*	Local directory *"src/testbed/chopp"* Source code file: chopp.f90	n18_Chopp_*yyy_mm_dd*.in	- Name/Directory of file containing list of FM-SDR files - Directory of chopped files - Name/Directory of log file - Number of chopped files
Application of regression algorithms *Applies the regression coefficients* on observed radiance data (fmsdr) to generate *EDRs (for 1st guess and/or background 1dvar processing)*	Local directory: *"src/testbed/retrRegress'* Source code file: *ApplyRegress.f90*	n18_ApplyRegress_*yyy_mm_dd*.in	- Number of files containing regression coefficients (maximum of 24) - The name/directory of the regression files - List of radiances files to be used - Directory/name of generated EDR files - Name/directory of the topography file - Names/directories of atmospheric and surface covariance files - Name/directory of the Log file - Name/Directory of the Bias File - Sensor ID - Number of orbits to process - Name/Directory of the tuning file - Algorithm serial number
1DVAR *Application of 1DVAR inversion algorithm for generating main product EDRs*	Local directory: *"src/1dvar*" Source code file: *1dvar.f90*	n18_CntrlConfig_1dvar_*yyy_mm_dd*.in_* ** integer value that denotes orbit number for daily processing (usually from 0 to 15)*	- Algorithm serial number - Number of orbits to process - Flag indicating option for monitoring iterations - Number of retrieval attempts (maximum of two) - Flag indicating option to use external data as first guesses (0 – for no use and 1 – for use - Flag indicating on-screen print monitoring option (0 for no on-screen printing and 1 – for on-screen printing) - Flag indicating the pass mode - Geographical limits (in latitude and longitude) - Name/Directory of; fmsdr measurement file - Name/Directory of the bias file - Name/Directory of the tuning files (maximum of two) - Name/Directory of the output File (edr) - Name/Directory of the covariance background files for the atmospheric parameters (maximum of two) - Name/Directory of the covariance background files for the surface parameters (maximum of two) - Name/Directory of the model error files (maximum of two) - Name/Directory of the external data file - Name/Directory of the noise file - Name/Directory of the monitoring file - Name/Directory of the topography - Names/Directories of the CRTM files -Name/Directory of the log file

Merge EDR *Process of merging bits of EDRs into a full orbital file*	Local directory: *“src/testbed/mergeEDR”* Source code file: *mergeEDR.f90*	n18_mergeEDR_*yyy_mm_dd*.in	- Name/Directory of file containing list of to-be-merged mini EDRs - Directory of output (merged EDRs) files - Name/Directory of log file
Bias verification and minitoring *Generation of bias monitoring figures*	Local directory: *“src/testbed /biasGenerAndMonit”* Source code file: *biasMonitor.pro*	n18_Inputs4BiasVerification_*yyy_mm_dd*.in	- Name/Directory of the bias file - Name/Directory of the bias monitoring figures - Satellite id (1 – for NOAA-18, 2 – for Metop-A, 3 – for F16 SSMIS
Generation of gridded parameter files *Generatation of gridded files for a specific retrieved or monitoring parameter for easy mapping and plotting)*	Local directory: *“src/testbed/grid”* Source code files: *gridDep.f90* *gridRad.f90* *gridEdr.f90* *gridBias.f90* *p2pEdr.f90* *p2pDep.f90* *p2pRad.f90*	n18_Grid_2008-06-06.in_edr n18_Grid_2008-06-06.in_dep n18_NWPGrid_2008-06-06.in_gdas n18_NWPGrid_2008-06-06.in_ecmw n18_BiasGrid_2008-06-06.in_gdas(ecmw)	-sensor id -Directory location of binary grid output files - Directory location of png image output filesoutput image files - date in yyy-mm-dd - flag (0 or 1) to indicate if output file is MIRS (1) or otherwise, e.g., GDAS (0)
Data quality monitoring *Generates daily file with NEDT computed values* *Generates daily file with convergence and QC flag percentages*	Local directory: “src/testbed/nedtMonitoring” Source code file: *MonitorNEDT_n18.pro* *MonitorQC_mirs.pro*	Calling Sequence: MonitorNEDT,nedtList=arg1,psFilename=arg2 MonitorQC_mirs, namelist=arg1	Called directly from IDL with arguments: - List of NEDT individual files - File name that will contain the plots of NEDT monitoring MonitorQC_mirs called directly from IDL with args: - Namelist path and filename contain sensor_id, path of output figures, directory path with orbital convergence and QC data files
Vertical Integration and Post Processing (VIPP) *Generates derived product (DEP) from MIRS (1DVAR) retrieved EDRs*	Local directory: “src/testbed/vipp Source code files: *vipp.f90*	n18_Vipp_ *yyy_mm_dd.in*	- Name/Directory location of file with lists of EDR files - Directory location of output files - Name/Directory of Sea Ice Emissivity Catalog - Name/Directory of Snow Emissivity Catalog - Nazme/Directyry of output file - Directory of the log file - Maximum number of orbits to process - The maximum number of profiles to process - Sensor ID
Figure Generation *Generates maps and plots of the retrieved and monitoring parameters*	Local directory: “src/testbed/grid Source code files: gridMirs.pro gridRad.pro gridNwp.pro gridNwpAsym.pro gridNwpBias.pro p2p_mirs_mspps.pro p2p_mirs_nwp.pro	n18_Inputs4FigsGener_yyy_mm_dd.in n18_Inputs4FigsGener_yyy_mm_dd.in_gdas(ecmw)(mspps)	-Satellite id (1 – for NOAA-18, 2 – for METOP-A, and 3 – for F16 SSMIS - Grid factor, e.g., 4 for $1/4^{th}$ degree grid cell) - Directory of gridded binary files - Process mode (0 – for orbit, 1 – for daily) - Version number - Geographical limits in latitude and longitude

Table 4. Description of MIRS applications, including program files and the content of the control files. Examples are for NOAA-18 AMSU-MHS, but similar files exist for other satellites e.g. METOP-A.

3.3 Paths and Configuration Files (*PCF*)

The MIRS PCF files allow the user the option to reconstruct the MIRS directory system: the set-up of paths to data, processes and applications and to reconfigure its implementation. These files reside in the MIRS "/setup" top level directory. They consist of two general utility files and a sensor-specific file, e.g., for AMSU-MHS, SSMIS, etc. The general set-up files are designated as "paths" and "paths_idl.pro". They define paths to MIRS sensor-independent libraries and subdirectories such as CRTM, IDL code libraries and execution subdirectories. The other set-up file is a sensor-specific bash script that defines paths to sensor-specific and sensor-independent processes, data and applications. The file name is denoted as "*sensor*_pcf.bash" . where "*sensor*" denotes satellite sensor name, e.g., "n18_amsua_mhs", "metopA_amsua_mhs", "metopB_amsua_mhs", "f16_ssmis", and "f18_ssmis" for NOAA-18 AMSU-MHS, Metop-A AMSU-MHS, Metop-B AMSU-MHS, F16-SSMI/S, and F18-SSMI/S, respectively.
An example of the content of this PCF file for the NOAA-18 AMSU-MHS is provided below in Figure 9. Note that the PCF can also be generated automatically from the GUI-based MIRS Control Panel (MCP).

```
#--------------------------------------------------------------------------------
#
# S E C T I O N  O F  D A T A  A N D   P A T H S
#--------------------------------------------------------------------------------
#   Major root paths
#--------------------------------------------------------------------------------
rootPath='/net/orbit006l/home/sidb/mirs'
dataPath=${rootPath}'/data'
#--------------------------------------------------------------------------------
#      External data & Paths
#--------------------------------------------------------------------------------
externalDataPath=${dataPath}/ExternalData
rdrSensor1Path=${externalDataPath /rdr/n18_amsua_mhs
rdrSensor2Path=${externalDataPath}/rdr/n18_amsua_mhs
rdrOrbitPath=${externalDataPath}/rdr/OrbitalMode
nwpGdasGridPath=/net/orbit138l/disk2/pub/wchen/gdas
nwpEcmwfGridPath=/net/orbit138l/disk2/pub/wchen/ecmwf
#--------------------------------------------------------------------------------
#      Static data & Paths
#--------------------------------------------------------------------------------
staticDataPath=${dataPath}'/StaticData'
instrumentPath=${staticDataPath}/InstrConfigInfotopographyFile=${staticDataPath}/Topogra
phy/topography.bin_sgi
#------------------------------------------------------------------------
# S E C T I O N  O F  S W I T C H E S(W H I C H  A P P L I C A T I O N  T O  R U N)
#------------------------------------------------------------------------
step_rdr2tdrSensor1=1   #RDR->TDR (Sensor1)
step_rdr2tdrSensor2=1   #RDR->TDR (Sensor2)
step_mergeNedt=1        #MERGE NEDTs (Sensor1 and Sensor2)
#------------------------------------------------------------------------
# S E C T I O N OF C O N T R O L L I N G F L A G S
#--------------------------------------------------------------------
processMode=1   #0:Orbit processing  1:Daily processing
sensorId=1      #Sensor:1:N18,2:MetopA,3:F16, 4:Windsat
outFMAccuracy=0 #Flag to output of the FM accuracy metric (DeltaTB @89)
prefixFMAccuracy=QCcheck               #Prefix of file(s) w FM-acuracy metric (only if
outFMaccur=1)
nAttempts=1     #Number of retrieval attempts in case of non-convergence
```

Figure 9. A sample of the PCF file

3.4 Sequence Control Scripts (*SCS*)

The SCS file is a sensor-dependent bash script that controls and executes the sequence of processes described in Table 4 that are required for end-to-end retrievals, monitoring and generation of image files. The SCS script uses the PCF from which it extracts information on system configuration about paths to data and MIRS applications. The SCS also uses another bash script denoted as "`script_functions.bash`" that contains functions used to run testbed applications. Note that the SCS could also be generated automatically from the GUI-based MIRS Control Panel (MCP) for different sensors. The user could also use the SCS provided below as template and modify the sequence of processes to suit its own needs. Below we provide the content of a SCS file for a testbed application generated from the GUI.

Directory: "`scripts`"

File name: "`n18_scs_daily.bash`"

Purpose: Main system script for NOAA-18 AMSU-MHS system. It controls the processes and information flow for end-to-end MIRS retrievals

```
#!/bin/bash

#####################################################################################################
#
# Description:
#       This is the GUI generated bash script used to run the MIRS testbed.
#
# Record of revisions:
#          Date        Ver.        Programmer              Description of change
#      ============   =====   ======================   ===========================================
#       09/03/2005     v0      Sid-Ahmed Boukabara       Original script created
#                              (NOAA/NESDIS/ORA/IMSG)
#
#       02/20/2006     v1      Ninghai Sun               Modify script for operational testbed
#                              (NOAA/NESDIS/ORA/IMSG)
#
#       03/20/2006     v2      Ninghai Sun               Modify script to compromise to SSM/IS
#                              (NOAA/NESDIS/ORA/IMSG)    Change the way to find GDAS data to standard
#
#       03/31/2006     v3      Sid Ahmed Boukabara       Changes related to :
#                                                        (1) new footprint-matching,
#                                                        (2) addition of new covariance matrix
#                                                        (3) sensor ID (to distinguish sensor-dependent classifiers),
#                                                        (4) flexible handling of scanlines shift (mhs vs amsu)
#                                                        (5) bias correction method (bias removal or slope/intercept)
#                                                        (6) Added threshold checking of relative humidity
#       05/31/2006     v4      Sid Ahmed Boukabara       Changes related to directory structure more in line with oper.
#
#
#       12/19/2006     v5      Sid Ahmed Boukabara       Added the capability to run the scripts in
#                              (IMSG@NOAA/NESDIS/STAR)   orbital mode in addition to daily mode.
#
#       12/22/2006     v6      Sid Ahmed Boukabara       Major changes to make the script as simple as possible to
#                                                        maintain and ultimately to be generated through the JAVA GUI.
#
#       12/28/2006     v7      Sid Ahmed Boukabara       Extensive revision to make all functions general.
#                              (IMSG@NOAA/NESDIS/STAR)   Scripts to be generated automatically through a Java GUI.
#                                                        This removes the need to maintain as many scripts as sensors.
#       03/10/2010     v8      Wanchun Chen              Many functions previously in script are moved into library.
```

```
#                       (QSS/PSGS/DELL)
#
#
##############################################################################################################

#------Store arguments & set starting date
script=$0
oldargs=("$@")
nargs=$#
sdate=`date`
MIRS_ROOT=`grep ^MIRS_ROOT ../setup/paths | cut -f2 -d '=' | sed -e 's/^[   ]*//'`
HDF4LIB=`grep HDF4LIB ../setup/paths | cut -f2 -d '=' | sed -e 's/^[              ]*//'`
HDF5LIB=`grep HDF5LIB ../setup/paths | cut -f2 -d '=' | sed -e 's/^[              ]*//'`
HDFEOSLIB=`grep HDFEOSLIB ../setup/paths | cut -f2 -d '=' | sed -e 's/^[      ]*//'`
SZIPLIB=`grep SZIPLIB ../setup/paths | cut -f2 -d '=' | sed -e 's/^[   ]*//'`
ZLIBLIB=`grep ZLIBLIB ../setup/paths | cut -f2 -d '=' | sed -e 's/^[  ]*//'`
NETCDF4LIB=`grep NETCDF4LIB ../setup/paths | cut -f2 -d '=' | sed -e 's/^[  ]*//'`


#-----Include libraries and setup Info
. ../scripts/script_functions.bash
. ../setup/n18_pcf.bash
# Add more colon-separated shared libraries here:
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HDF4LIB:$HDFEOSLIB:$SZIPLIB:$ZLIBLIB:$HDF5LIB:$NETCDF4LIB
# set stack size to unlimited
ulimit -s unlimited

displayVerif ${script} ${logFile} ${processMode} ${sensorId}\
        ${outFMAccuracy} ${prefixFMAccuracy} ${nProfs2Retr} ${nProfs2Fwd}\
        ${addDeviceNoise} ${monitorIterative} ${nAttempts}\
        ${externalDataAvailable} ${monitorRetrieval} ${monitorFwd} ${geoLimit}\
        ${minLat} ${maxLat} ${minLon} ${maxLon} ${maxDaysArchived} ${nDaysBack}\
        ${tdrFormat} ${cend} ${dayUsed4Bias} ${dayUsed4Alg} ${nOrbits2Process}\
        ${gifDensity} ${externalDataSrc} ${fmType} ${biasComputeMethod} \
        ${nChoppedFilesPerOrbit} ${retrOnOrbitOrSubOrbit} ${retrOnWhichSDR}\
        ${fwdMatrix2Use} ${makeOrNot} ${useCPU}

extResol=`determineSpatialResol ${satId} ${fmType}`

version='9999'
if [[ -s ${rootPath}/version.txt ]] ; then
   version=`cat ${rootPath}/version.txt`
fi

os=`uname -s`
#Linux g95 and gfortran is the same as AIX xlf90/95(stream,unformatted)
#Linux ifort is (sequential,binary)
accessStr='sequential'
formStr='binary'
if [[ ${os} == 'Linux' ]] ; then
  accessStr='sequential'
  formStr='binary'
elif [[ ${os} == 'AIX' ]] ; then
  accessStr='stream'
  formStr='unformatted'
fi

gdasData=1
ecmwfData=2
gfsData=3

#----Construct date extension(s) for building directories
if [[ ${nargs} -eq 0 ]] ; then
   extensions=`DetermineYesterdExtAndAlanysExt ${nDaysBack}`
   date=$(echo ${extensions}|cut -c1-10)
   rdrSensor1Dir=${rdrSensor1Path}/${date}
   rdrSensor2Dir=${rdrSensor2Path}/${date}
elif [[ ${nargs} -eq 1 ]] ; then
```

```
   #----Check 3 different paths (absolute/relative/a nude dir name)
   indx_slash=`echo ${oldargs} | awk -v slash="/" '{printf index($1,slash)}'`
   if [[ ${indx_slash} -eq 1 ]] ; then # =1: absolute path
     inputPath=${oldargs}
   elif [[ ${indx_slash} -gt 1 ]] ; then # >1: relative path
     inputPath=$(readlink -f ${oldargs}) # to get absolute full path
   else # <1: a nude dir name with no slash in it
     inputPath=${rdrSensor1Path}/${oldargs}
   fi
   date=`getDateFromFile ${satId} ${inputPath}`
   if [[ ${date} == 'xxxx-xx-xx' ]] ; then # something wrong
     op_msg "Error: date $date is incorrect"
     exit 1
   fi
   rdrSensor1Dir=${inputPath}
   rdrSensor2Dir=${inputPath}
   extensions=`DetermineExtAndAlanysExtFromArgument ${date}`
else
   ErrMessDue2UsageInDailyMode
fi
rdirExt=$(echo ${extensions}|cut -c1-10)
rdirAnalysExt=$(echo ${extensions}|cut -c12-21)
rdirNextAnalysExt=$(echo ${extensions}|cut -c23-32)
#----set the generic extension
fileExt=`determineFileExt ${rdirExt}`
fileAnalysExt=`determineFileExt ${rdirAnalysExt}`
orbitInfo=Dummy

#----Names of directories for both orbital and Daily processing
tdrSensor1Dir=${tdrSensor1Path}/${rdirExt}
tdrSensor2Dir=${tdrSensor2Path}/${rdirExt}
sdrSensor1Dir=${sdrSensor1Path}/${rdirExt}
sdrSensor2Dir=${sdrSensor2Path}/${rdirExt}
fmsdrDir=${fmsdrPath}/${rdirExt}
fmsdrChoppDir=${choppPath}/${rdirExt}
edrDir=${edrPath}/${rdirExt}
depDir=${depPath}/${rdirExt}
gridDir=${gridPath}/${rdirExt}
ncDir=${ncPath}/${rdirExt}
figsDir=${figsPath}/${rdirExt}
regressRetrDir=${regressRetrPath}/${rdirExt}
orbitMonPath=${perfsMonitorPath}/orbitmon/
#----control file and input file identifiers depending on processing mode
if [[ ${processMode} -eq 0 ]] ; then
  identifier=${orbitInfo}
else
  identifier=${rdirExt}
fi
#----Control file
rdr2tdrSensor1ControlFile=${rdr2tdrSensor1ControlFile}_${identifier}.in
rdr2tdrSensor2ControlFile=${rdr2tdrSensor2ControlFile}_${identifier}.in
mergeNedtControlFile=${mergeNedtControlFile}_${identifier}.in
tdr2sdrSensor1ControlFile=${tdr2sdrSensor1ControlFile}_${identifier}.in
tdr2sdrSensor2ControlFile=${tdr2sdrSensor2ControlFile}_${identifier}.in
fmControlFile=${fmControlFile}_${identifier}.in
modifyNedtControlFile=${modifyNedtControlFile}_${identifier}.in
fmsdr2edrControlFile=${fmsdr2edrControlFile}_${identifier}.in
grid2nwpControlFile=${grid2nwpControlFile}_${identifier}.in
fwdControlFile=${fwdControlFile}_${identifier}.in
regressControlFile=${regressControlFile}_${identifier}.in
choppControlFile=${choppControlFile}_${identifier}.in
mergeEdrControlFile=${mergeEdrControlFile}_${identifier}.in
vippControlFile=${vippControlFile}_${identifier}.in
gridControlFile=${gridControlFile}_${identifier}.in
nwpGridControlFile=${nwpGridControlFile}_${identifier}.in
fwdGridControlFile=${fwdGridControlFile}_${identifier}.in
biasGridControlFile=${biasGridControlFile}_${identifier}.in
biasCompuControlFile=${biasCompuControlFile}_${identifier}.in
```

```
biasVerifControlFile=${biasVerifControlFile}_${identifier}.in
regressGenControlFile=${regressGenControlFile}_${identifier}.in
figsGenControlFile=${figsGenControlFile}_${identifier}.in
#---- Input file list
rdrSensor1List=${rdrSensor1List}_${identifier}.list
rdrSensor2List=${rdrSensor2List}_${identifier}.list
tdrSensor1List=${tdrSensor1List}_${identifier}.list
tdrSensor2List=${tdrSensor2List}_${identifier}.list
sdrSensor1List=${sdrSensor1List}_${identifier}.list
sdrSensor2List=${sdrSensor2List}_${identifier}.list
fmsdrList=${fmsdrList}_${identifier}.list
fmsdr4BiasList=${fmsdr4BiasList}_${identifier}.list
fmsdr4ChoppList=${fmsdr4ChoppList}_${identifier}.list
fmsdr4NwpList=${fmsdr4NwpList}_${identifier}.list
fmsdr4BiasList=${fmsdr4BiasList}_${identifier}.list
fmsdr4RegressList=${fmsdr4RegressList}_${identifier}.list
fmsdr4ApplyRegressList=${fmsdr4ApplyRegressList}_${identifier}.list
edrList=${edrList}_${identifier}.list
edr4BiasList=${edr4BiasList}_${identifier}.list
dep4BiasList=${dep4BiasList}_${identifier}.list
edr4MergeList=${edr4MergeList}_${identifier}.list
depList=${depList}_${identifier}.list
nedtList=${nedtList}_${identifier}.list
nedtSensor1List=${nedtSensor1List}_${identifier}.list
nedtSensor2List=${nedtSensor2List}_${identifier}.list
gridSfcNwpAnalysList=${gridSfcNwpAnalysList}_${identifier}.list
gridAtmNwpAnalysList=${gridAtmNwpAnalysList}_${identifier}.list
nwpAnalysList=${nwpAnalysList}_${identifier}.list
nwpAnalysRetrList=${nwpAnalysRetrList}_${identifier}.list
nwpAnalys4BiasList=${nwpAnalys4BiasList}_${identifier}.list
nwpAnalys4RegressList=${nwpAnalys4RegressList}_${identifier}.list
fwdAnalys4BiasList=${fwdAnalys4BiasList}_${identifier}.list

#----Names of directories exclusively for Daily processing
figs4BiasDir=${perfsMonitorPath}/${rdirAnalysExt}
qcCheckDir=${perfsMonitorPath}/${rdirExt}
figs4RegressDir=${perfsMonitorPath}/${rdirAnalysExt}
nwpAnalysDir=${nwpAnalysPath}/${rdirAnalysExt}
fwdAnalysDir=${fwdAnalysPath}/${rdirAnalysExt}
fwd2hdf5Dir=${externalDataPath}/rdr/npp_atms/${rdirAnalysExt}
fmsdr4BiasDir=${fmsdrPath}/${rdirAnalysExt}
edr4BiasDir=${edrPath}/${rdirAnalysExt}
dep4BiasDir=${depPath}/${rdirAnalysExt}
grid4BiasDir=${gridPath}/${rdirAnalysExt}
#----Names of the files dynamically generated
sensor1Nedt=${nedtSensor1Path}/${satId}_${sensor1}_nedt_${fileExt}.dat
sensor2Nedt=${nedtSensor2Path}/${satId}_${sensor2}_nedt_${fileExt}.dat
nedtExt=${fileExt}
if [[ ${satId} == "f16" || ${satId} == "aqua" || ${satId} == "f18" || ${satId} == "fy3ri" || ${satId} == "trmm" || ${satId} == "gpm" || ${satId} ==
"mtma" || ${satId} == "mtsa" ]] ; then
  nedtBefFMFile=${nedtSensor1Path}/${satId}_${sensor1}_nedt_${fileExt}_befFM.dat
  nedtAftFMFile=${nedtSensor1Path}/${satId}_${sensor1}_nedt_${fileExt}_aftFM.dat
  nedtDir=${nedtSensor1Path}/${rdirExt}
elif [[ ${satId} == "npp" ]]; then
  nedtExt=`determineNedtExt ${satId} ${rdrSensor1Dir} ${rdrType}`
  nedtBefFMFile=${nedtSensor1Path}/${satId}_${sensor1}_nedt_${nedtExt}_befFM.dat
  nedtAftFMFile=${nedtSensor1Path}/${satId}_${sensor1}_nedt_${nedtExt}_aftFM.dat
  nedtDir=${nedtSensor1Path}/${rdirExt}
else
  nedtBefFMFile=${nedtSensor1Sensor2Path}/${satId}_${sensor1}_${sensor2}_nedt_${fileExt}_befFM.dat
  nedtAftFMFile=${nedtSensor1Sensor2Path}/${satId}_${sensor1}_${sensor2}_nedt_${fileExt}_aftFM.dat
  nedtDir=${nedtSensor1Sensor2Path}/${rdirExt}
fi
sensor1Wt=${nedtSensor1Path}/${satId}_${sensor1}_wt_${fileExt}.dat
sensor2Wt=${nedtSensor2Path}/${satId}_${sensor2}_wt_${fileExt}.dat
sensor1Sensor2Wt=${nedtSensor1Sensor2Path}/${satId}_${sensor1}_${sensor2}_wt_${fileExt}.dat
modelErrFile=${biasPath}/ModelErrFile_${satId}_${fileExt}.dat        #Model err file that will be generated
if [[ ${satId} == "npp" ]] ; then
```

```
  modelErrFile=${biasPath}/ModelErrFile_${satId}_${nedtExt}.dat
fi
logFile=${logFile}_${fileExt}.dat
#----Names of files dynamically generated only for daily processing
biasFile=${biasPath}/biasCorrec_${satId}_${fileAnalysExt}.dat        #bias file that will be generated
if [[ ${satId} == "npp" ]] ; then
  biasFile=${biasPath}/biasCorrec_${satId}_${nedtExt}.dat
fi
biasCheckFile=${biasPath}/biasAfterCorr_${satId}_${fileAnalysExt}.dat #bias residual file
#----EDR/DEP/NWP grid/p2p products list
edrGridStr="angle,chisq,em,nattempt,niter,psfc,qc,scanday,scanpos,sfc,tbf,tbu,tbc,tbl,temp,tskin,wv,clwp,rainp,graupelp,pressure"
edrP2PStr="em,scanpos,sfc,tbu,tskin,wspd,temp,wv"
depGridStr="clw,gs,iwp,lwp,rr,rwp,sfc2,sice,sicefy,sicemy,snow,swe,tpw"
depP2PStr="clw,iwp,lat,lwp,rr,rwp,sfc2,swe,tpw"
nwpGridStr="angle,chisq,clw,em,iwp,nattempt,niter,psfc,rr,scanday,scanpos,sfc,swe,tbf,tbu,tbc,tbl,temp,tpw,tskin,wv"
nwpP2PStr="clw,em,iwp,lwp,rwp,sfc,swe,tbu,tpw,tskin,temp,wv"
if [[ ${satId} == "f16" || ${satId} == "f18" || ${satId} == "trmm" || ${satId} == "gpm" || ${satId} == "mtma" || ${satId} == "mtsa" ]] ; then
  depGridStr="clw,gs,iwp,lwp,rr,rwp,sfc2,sice,sicefy,sicemy,snow,swe,tpw,wspd"
  depP2PStr="clw,iwp,lat,lwp,rr,rwp,sfc2,swe,tpw,wspd"
  nwpGridStr="angle,chisq,clw,em,iwp,nattempt,niter,psfc,rr,scanday,scanpos,sfc,swe,tbf,tbu,tbc,tbl,temp,tpw,tskin,wspd,wv"
  nwpP2PStr="clw,em,iwp,lwp,rwp,sfc,swe,tbu,tpw,tskin,temp,wspd,wv"
fi

#---- fwd model error matrix
modelErrFile1ToUse=${biasPath}/ModelErrFile_${satId}.dat
modelErrFile2ToUse=${biasPath}/ModelErrFile_${satId}.dat

#---Check existence of the directories. If negative, make them
DirGen ${tdrSensor1Dir} "TDR-${sensor1}"
if [[ $sensor2 != "dummy" ]] ; then
  DirGen ${tdrSensor2Dir} "TDR-${sensor2}"
fi
DirGen ${sdrSensor1Dir} "SDR-${sensor1}"
if [[ $sensor2 != "dummy" ]] ; then
  DirGen ${sdrSensor2Dir} "SDR-${sensor2}"
fi
if [[ ${satId} == "npp" ]] ; then
  DirGen ${nedtDir} "NEDT"
fi
DirGen ${fmsdrDir} "FM-SDR"
DirGen ${fmsdrChoppDir} "CHOPPED FMSDRs"
DirGen ${edrDir} "EDR"
DirGen ${depDir} "DEP"
DirGen ${gridDir} "GRID"
DirGen ${ncDir} "NETCDF4"
DirGen ${figsDir} "FIGS"
DirGen ${figs4BiasDir} "FIGS"
DirGen ${grid4BiasDir} "GRID-BIAS"
DirGen ${qcCheckDir} "QCCheck"
DirGen ${figs4RegressDir} "regr-FIGS"
DirGen ${nwpAnalysDir} "NWP-ANALYSIS"
DirGen ${fwdAnalysDir} "FWD SIMUL on Analyses"
DirGen ${orbitMonPath} "QC Monitor"
DirGen ${regressRetrDir} "Regression-Based Retrievals"

#-------------------------------------------------------------------------------
#     NPP/ATMS special case: to generate NEDT from SDR files
#-------------------------------------------------------------------------------
if [[ ${satId} == "npp" && ${rdrType} -eq 1 ]] ; then
  rdrType2=3
  nfile_gatmo=`ls -1 ${rdrSensor1Dir}/GATMO_npp* 2> /dev/null | wc -l`
  nfile_satms=`ls -1 ${rdrSensor1Dir}/SATMS_npp* 2> /dev/null | wc -l`
  #---- only do this when SDR files exist and have equal numer of GATMO files
  if [[ ${nfile_satms} -ge 1 && ${nfile_gatmo} -eq ${nfile_satms} ]] ; then
    rdr2tdr ${rdrSensor1Dir} ${rdrSensor1List} ${sensor1} ${tdrSensor1Dir} ${sensor1Nedt}\
    ${instrumentSensor1File} ${sensor1Wt} ${nOrbits2Process} ${logFile}\
    ${rdr2tdrSensor1ControlFile} ${rdr2tdrSensor1Src} ${makeOrNot} ${binPath}\
    ${processMode} ${orbitInfo} ${satId} "${rdirExt}" ${controlDataPath}\
```

```
    ${calibBiasFitFile} ${calibDTRlutFile} ${accessStr} ${formStr} ${rdrType2}
    #---- clean up ( only need remove TDR to avoid conflict )
    rm -f ${tdrSensor1Dir}/TDR*
  fi
fi

#-------------------------------------------------------------------------------
#     step: RDR to TDR conversion (sensor1)
#-------------------------------------------------------------------------------
if [[ "${step_rdr2tdrSensor1}" -eq 1 ]] ; then
  rdr2tdr ${rdrSensor1Dir} ${rdrSensor1List} ${sensor1} ${tdrSensor1Dir} ${sensor1Nedt}\
  ${instrumentSensor1File} ${sensor1Wt} ${nOrbits2Process} ${logFile}\
  ${rdr2tdrSensor1ControlFile} ${rdr2tdrSensor1Src} ${makeOrNot} ${binPath}\
  ${processMode} ${orbitInfo} ${satId} "${rdirExt}" ${controlDataPath}\
  ${calibBiasFitFile} ${calibDTRlutFile} ${accessStr} ${formStr} ${rdrType}
fi

#-------------------------------------------------------------------------------
#     step: RDR to TDR conversion (sensor2)
#-------------------------------------------------------------------------------
if [[ "${step_rdr2tdrSensor2}" -eq 1 ]] ; then
  rdr2tdr ${rdrSensor2Dir} ${rdrSensor2List} ${sensor2} ${tdrSensor2Dir} ${sensor2Nedt}\
  ${instrumentSensor2File} ${sensor2Wt} ${nOrbits2Process} ${logFile}\
  ${rdr2tdrSensor2ControlFile} ${rdr2tdrSensor2Src} ${makeOrNot} ${binPath}\
  ${processMode} ${orbitInfo} ${satId} "${rdirExt}" ${controlDataPath}\
  ${calibBiasFitFile} ${calibDTRlutFile} ${accessStr} ${formStr} ${rdrType}
fi

#-------------------------------------------------------------------------------
#     step: Merge Different Sensor NEDT files into a single file
#-------------------------------------------------------------------------------
if [[ "${step_mergeNedt}" -eq 1 ]] ; then
  mergeNedt ${sensor1Nedt} ${sensor2Nedt} ${nedtBefFMFile} ${logFile}\
  ${mergeNedtControlFile} ${mergeNedtSrc} ${makeOrNot} ${binPath} ${satId}\
  "${nedtNominalFile}"  "${nedtDir}"
fi

#-------------------------------------------------------------------------------
#     step: Footprint Matching (FM) sensor1/sensor2
#-------------------------------------------------------------------------------
if [[ "${step_fm}" -eq 1 ]] ; then
  fm ${sdrSensor1Dir} ${sdrSensor1List} ${sdrSensor2Dir} ${sdrSensor2List}\
  ${sensor1} ${sensor2} ${fmsdrDir} ${outFMAccuracy} ${perfsMonitorPath}\
  ${prefixFMAccuracy} ${fmType} ${nScanLineSensor1Skip} ${nScanLineSensor2Skip}\
  ${scanLineIndexSensor2TimeColloc} ${nOrbits2Process} ${logFile} ${fmControlFile}\
  ${fmSrc} ${makeOrNot} ${binPath} ${processMode} ${orbitInfo} ${satId}\
  ${nedtBefFMFile} ${nedtAftFMFile} ${modifyNedtControlFile} ${figs4BiasDir}\
  ${identifier} ${outFMAccuracy} ${inputDataPath} ${controlDataPath}\
  ${geoLimit} ${minLat} ${maxLat} ${minLon} ${maxLon}
fi

#-------------------------------------------------------------------------------
#     step: Apply regression-based algorithms on FM-SDR radiances
#-------------------------------------------------------------------------------
if [[ "${step_externalDataFromRegress}" -eq 1 ]] ; then
  applyRegress ${fmsdrDir} "${extResol}" ${fmsdr4ApplyRegressList} ${regressRetrDir}\
  ${topographyFile} ${covBkgAtm1File} ${covBkgSfc1File} ${logFile} ${regressControlFile}\
  ${applyRegressAlgSrc} ${makeOrNot} ${binPath} ${processMode} ${orbitInfo}\
  ${regressCoeffOceanClwFile}   ${regressCoeffSeaIceClwFile}   ${regressCoeffLandClwFile}   ${regressCoeffSnowClwFile}\
  ${regressCoeffOceanTskinFile} ${regressCoeffSeaIceTskinFile} ${regressCoeffLandTskinFile} ${regressCoeffSnowTskinFile}\
  ${regressCoeffOceanTpwFile}   ${regressCoeffSeaIceTpwFile}   ${regressCoeffLandTpwFile}   ${regressCoeffSnowTpwFile}\
  ${regressCoeffOceanEmFile}    ${regressCoeffSeaIceEmFile}    ${regressCoeffLandEmFile}    ${regressCoeffSnowEmFile}\
  ${regressCoeffOceanWvFile}    ${regressCoeffSeaIceWvFile}    ${regressCoeffLandWvFile}    ${regressCoeffSnowWvFile}\
  ${regressCoeffOceanTempFile}  ${regressCoeffSeaIceTempFile}  ${regressCoeffLandTempFile}  ${regressCoeffSnowTempFile}\
  ${regressCoeffOceanGwpFile}   ${regressCoeffSeaIceGwpFile}   ${regressCoeffLandGwpFile}   ${regressCoeffSnowGwpFile}\
  ${retrOnOrbitOrSubOrbit} ${fmsdrChoppDir} ${sensorId}\
  ${nOrbits2Process} ${biasFileToUse} ${tune1File} ${version}
fi
```

```
#-------------------------------------------------------------------------------
#      step: EDRs retrieval from the FM-SDRs
#-------------------------------------------------------------------------------
if [[ "${step_fmsdr2edr}" -eq 1 ]] ; then
  fmsdr2edr ${fmsdr2edrSrc} ${makeOrNot} ${retrOnWhichSDR} ${fmsdrDir}\
  ${fwdAnalysDir} ${retrOnOrbitOrSubOrbit} ${fmsdrChoppDir} ${fmsdrList}\
  ${nOrbits2Process} ${externalDataAvailable} ${externalDataSrc} ${nwpAnalysDir}\
  ${nwpAnalysRetrList} ${regressRetrDir} ${edrDir} ${nProfs2Retr}\
  ${monitorIterative} ${nAttempts} ${monitorRetrieval} ${geoLimit}\
  ${minLat} ${maxLat} ${minLon} ${maxLon} ${cend} ${sensorId}\
  ${tune1File} ${tune2File} ${covBkgAtm1File} ${covBkgAtm2File}\
  ${covBkgSfc1File} ${covBkgSfc2File} ${modelErrFile1ToUse} ${modelErrFile2ToUse} \
  ${nedtAftFMFile} ${monitorFile} ${topographyFile}\
  ${spcCoeffFile} ${tauCoeffFile} ${cldOptPropFile} ${logFile}\
  ${fmsdr2edrControlFile} ${useCPU} ${processMode} ${orbitInfo} ${binPath}\
  ${biasFileToUse} ${version} "${extResol}" "${rdirExt}"
fi

#-------------------------------------------------------------------------------
#      step: To generate vertical integrated products
#-------------------------------------------------------------------------------
if [[ "${step_vipp}" -eq 1 ]] ; then
  vipp ${edrList} ${edrDir} ${depDir}  ${logFile} ${nOrbits2Process} ${nProfs2Retr}\
  ${sensorId} ${processMode} ${orbitInfo} "${extResol}" ${vippControlFile}\
  ${vippSrc} ${binPath} ${siceEmissCatalogFile} ${snowEmissCatalogFile}
fi

#-------------------------------------------------------------------------------
#      step: EDRs Figures Generation
#-------------------------------------------------------------------------------
if [[ "${step_figsGen}" -eq 1 ]] ; then
  figsGen ${satId} ${gridFactor} ${gridDir} ${gridSrc} ${figsDir} ${IDL}\
  ${identifier} ${figsGenControlFile} ${processMode} ${controlDataPath} ${version}\
  ${minLat} ${maxLat} ${minLon} ${maxLon}
fi

#-------------------------------------------------------------------------------
#      step: Data monitoring (plots)
#-------------------------------------------------------------------------------
if [[ "${step_dataMonitor}" -eq 1 ]] ; then
  qcRetrieval ${depDir} ${depList} ${orbitMonPath} ${IDL} ${gridSrc}\
  ${controlDataPath}/qcRetrieval_abnormal_${satId}_${fileExt}\
  ${controlDataPath}/qcRetrieval_namelist_${satId}_${fileExt}\
  "${email}" "${website}" ${controlDataPath}

  dataQualityMonitor ${nedtSensor1Sensor2Path} ${nedtList} ${nedtMonitorSrc} ${IDL}\
  ${orbitMonPath} ${figsDir} ${processMode} ${fileExt} ${satId} ${controlDataPath} ${perfsMonitorPath}
fi
```

Figure 10.This sequence control script is generated automatically throught the MIRS Control Panel (MCP) and submitted to cronjob or other schedulers to run operationally.

3.5 Design of Image Files

The image files consist of output files in PNG format of MIRS retrieved (EDR) and performance parameters. The user can directly access these files as generated from a particular retrieval attempt. For more refined analysis, monitoring, validation, and quality control purposes, the user is advised to access these image products through the MIRS website tools.

The image files are located in the "data/TestbedData/Outputs/Figs" local directory. The main programs that generate these files are written in IDL and are located in the "/src/testbed/grid" local directory.

The file naming convention of the image files is as follows:

mirs_alg_*satellite*_sensor_*region*_yyymmdd_*parameter_sfctype_node*.png

Where:

alg - refers to the algorithm used, e.g., "*her*" denotes the heritage algorithm and "*adv*" denotes the advanced algorithm.

Satellite – refers to the satellite name, e.g, "poes_n18", "poes_metopA", "dmsp_f16", and "dmsp_f18" denote the POES NOAA-18, METOP-A, DMSP F16, and DMSP F18 satellites, respectively.

Sensor - refers to sensor name, e.g, "amsuamhs" and "ssmis" denote the AMSU-MHS and SSMIS sensors, respectively.

region –refers to the coverage area of the particular figure:. For instance, the "*glb*" notation means that the coverage is global.

yyymmdd – refers to time in year, month and day of month format, e.g., 20071125 denotes November 25, 2007.
parameter -refers to the retrieved or monitoring parameter

sfctype– refers to surface type, e.g., "lnd" over land, "sea" over water and "all" over both land and water

node– refers to the sensor node, e.g. "as" for ascending and "ds" for descending.

Table 5 below provides a list of parameter notations and a short description.

Parameter notation	**Description**
clw	Denotes the cloud liquid water
chisq	Denotes the QC converge parameter; A value less than 5 indicates converge of acceptable quality
em	Denotes emissivity (0 to 1) at a given frequency, e.g., "em23h" denotes emissivity at 23 GHz channel, vertical polarization
gs	Denotes snow effective grain size in mm
iwp	Denotes the ice water path
lwp	Denotes the liquid water path
nattempt	Denotes the number of retrieval attempts
qc	Denotes the value of quality control flags
rr	Denotes rain rate (mm/hr)
niter	Denotes the number of iterations

psfc	Denotes the surface pressure (hPa)
lwp	Denotes liquid water path
rwp	Denotes the rain water path
sfcTyp2	Denotes the emissivity-based derived surface type, 0 - for ocean, 1 – for ice , 2- for land, and 3- for snow cover
sfcTyp	Denotes the TB-based surface type
Sicefy	Denotes first-year sea ice concentration (in %)
sicemy	Denotes multi-year sea ice concentration (in %)
sice	Denotes total sea ice concentration (I %)
snow	Denotes snow cover (1 – for snow cover and 0- for snow-free land
swe	Denotes Snow Water Equivalent (in cm)
tb	Denotes the brightness temperatures (in kelvin)
temp	Denotes layer temperature in K, e.g, “temp_800mb” denotes temperature at 800 mbar layer
tpw	Denotes the total precipitable water parameter (mm)
tskin	Denotes the skin temperature (K)
wspd	Denotes surface wind speed (m/s)
wv	Denotes the layer water vapor mixing ratio (g/kg), e.g, “wv_800mb” denotes water vapor at 800 mbar layer

Table 5. Description of parameter notation in the naming of the image files

Section 4.0 MIRS Inputs and Outputs

This section provides a summary description of MIRS input and output data files. The input files consist of the set-up files, the external data files and the static data files. The output files consist of the MIRS retrieval files and other (level II or III) output files. Provided below are brief descriptions of these files. Content and format details are described in the Interface Control Document (ICD).

4.1 Input Datasets

4.1.1 Set-Up Files

The set-up files consist of the PCFs and are used for defining paths to MIRS data and applications. The PCFs were described in section 3.3 above.

4.1.2 External Data Files

The external data files are input files coming from sources outside of MIRS. They consist of the level 1-b sensor data files, e.g., AMSU-A and MHS, SSMI/S, ATMS, AMSR-E, etc, and the NWP analysis (GDAS/ECMWF) or forecast (GFS) gridded files. These files reside in the "/data/ExternalData" sub-directory.

4.1.2.1 Sensor Data Files

For the case of NOAA-18, these will be level 1b data files. For DMSP sensors, these will be the NRL TDR files. For ATMS, these are currently the TDR files.

4.1.2.2 Gridded NWP Analysis Files

These are *optional* files and are used only for calibration and monitoring purposes. Calibration involves using surface and atmospheric state vectors contained in these data to run the CRTM forward simulator. This process generates simulated brightness temperatures, which are compared to measured radiances for computing bias. Monitoring involves comparisons with retrieved EDRs. The analysis data can be used however in the 1DVAR as fist guesses at user discretion if it is deemed that they help convergence of certain parameters.

4.1.2.3 Gridded NWP Forecast Files

Similar to analysis files, these forecast files are also *optional* and serve the same purpose.

4.1.3 Static Data Files

The static datasets consist of various nominal instrument, system and processes input parameters. These are permanent files that remain unchanged during the course of a retrieval process. They consist of the CRTM files, NEDT files, Geophysical Covariance matrices, topography tables, tuning files and ice and snow emissivity catalogs. These data reside in the "/data/StaticData/" sub-directory.

4.1.3.1 Forward Operator Look-Up Tables (*CRTM files*)

These are ASCII files containing coefficient parameters as input to the CRTM module. They come as part of the CRTM package and are not generated by MIRS.

4.1.3.2 Nominal Instrumental Error Covariance Matrices (*NEDT files*)

These are ASCII files containing instrument antenna temperature error covariance matrices needed as input to 1DVAR. These nominal NEDT values are used for those sensors where it is not possible to compute noise values "on the fly", e.g., SSMIS. For the AMSU-MHS, the NEDTs are dynamically generated following the methodology as described in Mo (2002). .

4.1.3.3 Geophysical Covariance Matrices

These data are ASCII files containing nominal surface and atmospheric parameters covariance matrices needed as input to 1DVAR. They also contain the mean background information derived from climatological studies.

4.1.3.4 Topography Tables

The topography tables are binary gridded file containing surface type and elevation data. The source of these tables is the USGS.

4.1.3.5 Tuning Files

These are ASCII files that contain tuning parameters needed to customize retrieval attempts. They contain such parameters as what EDRs to retrieve, channels selected, the number of iterations, etc. Note that up to two tuning files could be used, the second would be used in the case when there are two retrieval attempts. This means that a retrieval is performed using the parameter set contained in the first tuning file. If the first attempt retrieval did not converge and if the user has specified two attempts a second retrieval will be attempted using the second tuning file. The control of the number of attempts and the names of the tuning files are set in the paths and configuration file.

4.1.3.6 Emissivity Catalog Data Files

The emissivity catalogue files are ASCII files that contain surface emissivity spectra and geophysical parameters computed off-line from physical models. These files are used as input to the MIRS Vertical Integration and Post-processing (VIPP) module to compute emissivity-based EDRs.

There are two emissivity catalogue files for each sensor: the sea ice and snow cover emissivity catalogue files. For NOAA-18 AMSU-MHS, these files are denoted as "SeaIceEmissCatalog_n18_amsua_mhs.dat" and "SnowEmissCatalog_n18_amsua_mhs.dat", respectively. The Sea ice emissivity catalogue is used for computing sea ice parameters and the snow emissivity catalogue is used for computing snow cover parameters. These files reside in the "data/StaticData/EmissCatalog" sub-directory.

4.2 MIRS Outputs

4.2.1 MIRS Retrieval Files

MIRS retrieval files are level II swath binary files and the gridded binary and image files. The level II swath binary files consist of the MIRS main product file and the derived product file. The main product file contains MIRS retrieved EDRs, and the derived product file contains parameters that are derived from retrieved EDRs contained in the main product file through vertical integration and post-processing algorithms, e.g., the TPW parameter in the derived product file is derived from retrieved profiles of water vapor, CLW is derived from retrieved profiles of cloud liquid water, etc.

MIRS grid product files consist of binary and PNG image files of MIRS retrieved EDRs and other ancillary data. MIRS main product files reside in the "/data/TestbedData/Outputs/edr" subdirectory, the derived product files reside in the "/data/TestbedData/Outputs/dep" subdirectory. Figures and grid files reside in the "/data/TestbedData/Outputs/figs/" and "/data/TestbedData/Outputs/grids/" sub-directories, respectively. Finally, if specified in the PCF, the MIRS products can also be written to netCDF4 format files which would reside in the "/data/TestbedData/Outputs/nc/" sub-directory.

4.2.1.1 Main Product Swath Binary Files *(level II-a, EDR)*

The MIRS swath binary product file is generated by MIRS Input/Output modules. Both Fortran-95 and IDL readers are available to manipulate this file. File specifications are given in the ICD. The code for reading/writing swath product data also is provided in Appendix D of the ICD.

4.2.1.2 Derived Product Swath Binary Files *(level II-a, EDR)*

The MIRS Derived Product (DEP) swath binary file is generated by MIRS Input/Output modules written in both Fortran-95 and IDL. Both Fortran-95 and IDL readers are available to manipulate this file. File specifications are given in the ICD. The code for reading/writing swath product data is also provided in Appendix D of the ICD.

4.2.1.3 Image Files

The image files consist of PNG files of a specific retrieved or monitoring parameter. These files are generated by modules written in IDL. The image files are also used for providing the product monitoring figures.

4.2.2 HDF-EOS Swath Files *(level II-a, EDR)*

For the operational data distribution and archiving, MIRS output data will be reformatted as HDF-EOS files. Each MIRS output EDR binary data file will be rewritten as 2 HDF-EOS files. One is the profile HDF-EOS file which contains the retrieved atmospheric profile products for each layer and the other one is the image product HDF-EOS file which contains the retrieved image products.

The MIRS reformatting process will follow each orbit product generating process. First the products are read out at each retrieval spot from the binary orbit output file, the position indexes are checked and the output is written into an intermediate file. Then the output data are regrouped: each product of the whole orbit is put into a swath array based on scan number and the FOV index. Finally the HDF-EOS swath structures are defined and the product arrays are written into the HDF-EOS swath.

4.2.3 netCDF Swath Files *(level II-a, EDR)*

For operational data distribution and archiving of NPP ATMS products, MIRS output data will be reformatted as netCDF4 files. Each MIRS output EDR and DEP binary data file will be rewritten as two netCDF4 files. One is the profile netCDF4 file (filename beginning with "`SND_`") which contains the retrieved atmospheric profile products for each layer and the other one is the image product netCDF4 file (filename beginning with "`IMG_`") which contains the retrieved image products. This is an *optional* step, and the files are saved under the "`/data/TestBedData/Outputs/nc`" subdirectory.

4.2.4 Product Monitoring Figures

Extensive use of advanced product monitoring tools and techniques is a hallmark of MIRS. These tools provide important leverage and opportunities for users to monitor MIRS retrieved parameters.

4.2.5 Product Quality Assurance / Quality Control

Along with the MIRS products, several quality parameters (QPs) are provided. These are related only to the advanced algorithm products, generated by the 1DVAR algorithm:

- Convergence metrics (χ^2 & Y^{fwd})
- Quality Control (QC)
- Uncertainty matrix (***S***)
- Contribution functions (***D***)
- Average kernel (***A***)

Currently, only the convergence metrics and the Quality Control structure are determined and provided as part of the output file.

4.2.5.1 Interpretation and Usage of the Convergence Metrics

Along with the χ^2, the last-simulated brightness temperatures using the forward operator Y^{fwd} are also output. These can be used to examine the individual (channel by channel) residuals between the measurements and the brightness temperatures that were simulated using the retrieved state vector.

The convergence metric χ^2 is the rms of all residuals between the last-simulated brightness temperatures (using the forward operator) and the measurements. Only those channels effectively used in the retrieval are used to compute this metric. It is also sometimes called the non-constrained cost function. There is one single value of χ^2 for every profile and it is of type real. Nominally, the convergence is deemed reached when:

$$\chi^2 <= 1$$

Users can however relax this criterion to allow the use of profiles that reach a value of 5. Above this value of 5, the retrievals are deemed unreliable.

4.2.5.2 Interpretation and Usage of the Quality Control

In addition to the convergence metric, a detailed quality control structure is implemented and its content is also part of the output file. It contains four words, each one of sixteen bits. A general description of each word is as follows:

- The first word is defined as the general validity control flag. It ranks the quality of all MIRS products at a specific location as good (value set to 0), some problem (value set to 1), and bad (value set to 2). This general ranking is made based on the flags contained in the second, third and fourth word of the quality flag structure.
- The second word provides information about the convergence and the presence and the type of precipitation (light, medium or heavy). Also, a set of out-of-band flags are implemented. Those flags are turned on if one of the MIRS products is out of the predefined bound values, typically a non-physical value.
- The third word provides information about the detection of temperature lapse rate (including super-adiabatic lapse rate), temperature inversion, super saturation, humidity inversion and cloud detection
- The fourth word provides information about the quality of the measurements, It also provides information on surface type.

Finally, note that both the swath EDR and DEP binary files contain a QC variable, with the same structure described above. The QC variable in the DEP file will largely contain the same information as that from the EDR file, but with several bits possibly changed due to additional information obtained in the post-processing generation of the derived products (e.g. the presence of rainfall or hydrometeors).

4.2.5.3 Interpretation and Usage of the Uncertainty Matrix

Work in progress

4.2.5.4 Interpretation and Usage of the Contribution Function

Work in progress

4.2.5.5 Interpretation and Usage of the Average Kernel

Work in progress

Section 5.0 Operational Scenario

This section describes the operational scheduling of jobs and the execution mechanism.

5.1 Scheduling of Jobs

The operational testbed implementation of MIRS is controlled by the SCSs as shown in Figure 10. All the sequencing and synchronization of MIRS applications for the generation and monitoring of MIRS products is contained in the SCS. Currently, the daily processing mode is implemented with automatic calls to SCSs for NOAA-18, NOAA-19, METOP-A AMSU-MHS and METOP-B AMSU-MHS, as well as DMSP F16 SSMI/S, NPP ATMS, and the DMSP F18 SSMI/S satellites, respectively, once daily at predefined local times.

5.2 Job Initiation and Execution Mechanism

Job initiation is done automatically at the time specified above through submittals of SCSs to a cronjob.

Section 6.0 Resource Requirements

This section describes MIRS system storage requirements, computer resources and communication needs for a normal real-time application.

6.1 Storage Requirements

The MIRS package consists of source codes (Fortran-95, C/C++, IDL , JAVA and BASH scripts), logs files, binary files, and data in different formats. The directory structure of MIRS has been designed to be streamlined and compatible with operational needs and requirements. Figure 11 is the top level directory system of MIRS, which is briefly described as follows:

- The "/src" directory" contains all source files coded in Fortran-95, C/C++, IDL, BASH and JAVA.
- The "/scripts" directory contains all scripts (called sequence control scripts or SCS). These scripts are generated automatically by the MCP. They could be used as templates by users to schedule operational running.
- The "/setup" directory" contains the setup control files. Some of these set-up files could be controlled by the MCP (the Paths and Configuration Files or PCFs) and others need to be updated manually.
- The "/bin" directory contains all executable files. Sensor-specific processes and applications are distinguished by prefix. For example, the "fm_n18_amsua_mhs" executable runs the process of foot-print matching, represented by the "fm" prefix, for NOAA 18 AMSU-MHS.
- The "/doc" directory contains all MIRS documents including User's Manual, Interface Control Document, System Description Document, MIRS Directory Structure, etc.
- The "/gui" directory contains the GUI-based MIRS Control Panel (MCP) that will make and execute any part of or the entire MIRS system
- The "/data" directory contains the data files, including control and inputs files for running MIRS processes, external data files such as sensor and GDAS files, static and semi-static data files, and testbed data files.
- The "/logs" directory contains all log files generated by MIRS.

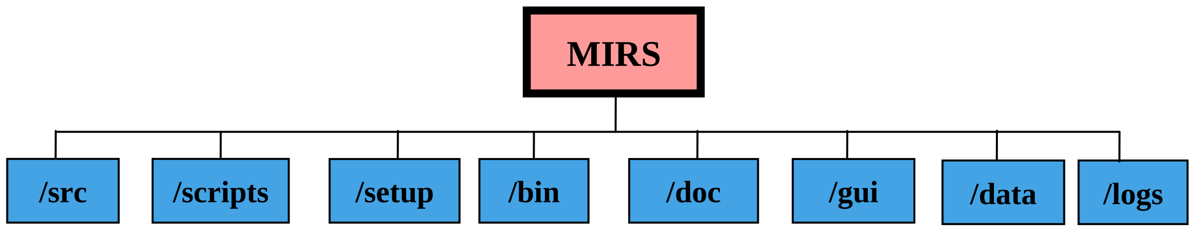


Figure 11. MIRS top level directory structure

6.1.1 Code Volumes

Presented here are the sizes of all MIRS codes for the NOAA 18 system. Sizes for other systems could be smaller or larger depending on the sensor.
As shown in Figure 12, the source code directory hosts 10 subdirectories, which are briefly described as follows:

- The "/1dvar" directory contains 1DVAR code for generating EDRs from radiance data.
- The "/fwd" directory contains the forward operator code for reproducing radiances from EDRs.
- The "/crtm" directory contains the Community Radiative Transfer Model code.
- The "/lib" directory contains the library modules written in Fortran-95 for data typing, reading/writing and the inverse processing.
- The "/lib_idl" directory contains the IDL libraries for data typing, input/output reading and writing, data collocation, background covariance computations, etc.
- The "/lib_java" directory contains Java code general utility functions.
- The "/lib_cpp" directory contains C++ code general utility functions.
- The "/qcDelivery' directory contains the IDL code for generating EDRs for comparison with benchmark files.
- The "/testbed" directory contains codes for MIRS real time applications, e.g., for footprint matching, bias correction and monitoring, figure generation, etc.

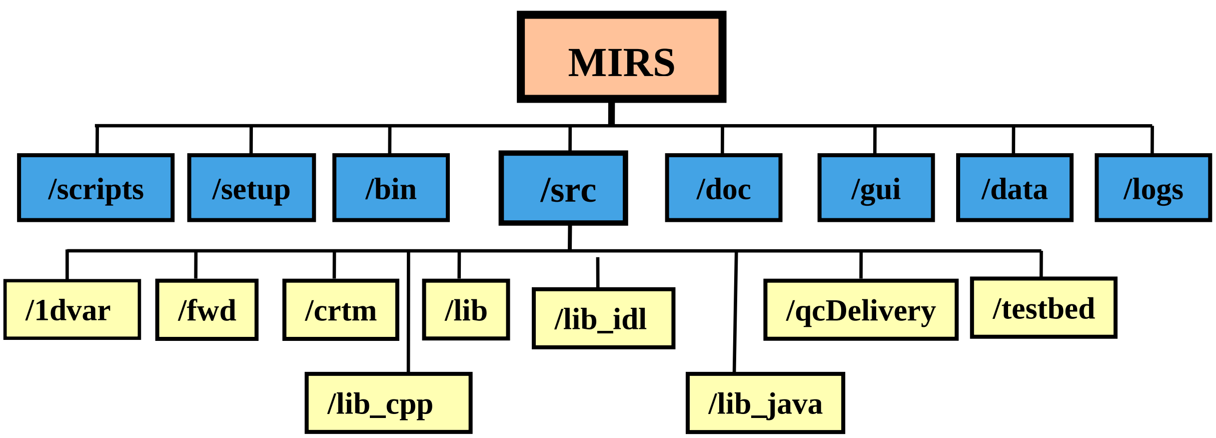


Figure 12. MIRS source code directory structure

The current volumes of MIRS source codes, including makefiles and BASH scripts are summarized in Table 6. Main programs and libraries are described in detail in the appendix.

MIRS	Files or Directory		Size (KB)	Total (KB)
setup		paths	2	22
		n18_pcf.bash	19	
		paths_idl.pro	1	
script		n18_scs.bash	22	18
src	**1dvar**	1dvar.f90	42	47
		makefile	5	
	fwd	fwd.f90	10	15
		makefile	5	
		//CRTMp_twostream_1024	1768	1768
	crtm	makefile	0.2	
	lib	/FwdOperProcess	32	646
		/io	236	
		/math	16	
		/mspps	36	
		/PrePostProcess	20	
		/utilities	72	
		/InversProcess	96	
		/misc	76	
		/noise	8	
		/qc	44	
		Makefile	10	
	lib_idl	algors.pro	42	656
		Export_IMG.pro	24	
		io_coloc.pro	18	
		io_covBkg.pro	12	
		io_dropsondes.pro	12	
		io_Mapping.pro	63	
		io_measur.pro	31	
		io_misc.pro	24	
		io_monitor.pro	18	
		io_regressAlgors.pro	11	
		io_scene.pro	45	
		io_dep.pro	8	
		meteorFcts_sub.pro	7	
		misc.pro	277	
		stats_sub.pro	14	
		utilities.pro	50	
Total MIRS code volume (without CRTM)				1404 KB
Total MIRS code volume (with CRTM)				3172 KB

Table 6. Sizes of MIRS source codes

6.1.2 **Data Volumes**

The data used by the MIRS system are stored in the "`/data`" top level directory. Figure 13 presents the expanded data directory structure. The MIRS generated dynamic data, NEDTs, outputs, and performance monitoring data are in the "data/TestbedData" sub-directory. The "data/SemiStatic" data sub-directory contains data files related to bias correction and regression algorithms. The "data/InputsData" sub-directory contains files that have input file name and location information. The "data/StaticData" sub-directory contains files that are not or are rarely modified. The files used for control of the individual applications are in "data/ControlData" sub-directory. Input data coming from external sources are in the "data/ExternalData" sub-directory. The "data/BenchmarkData" sub-directory contains benchmark files that come with the MIRS package.

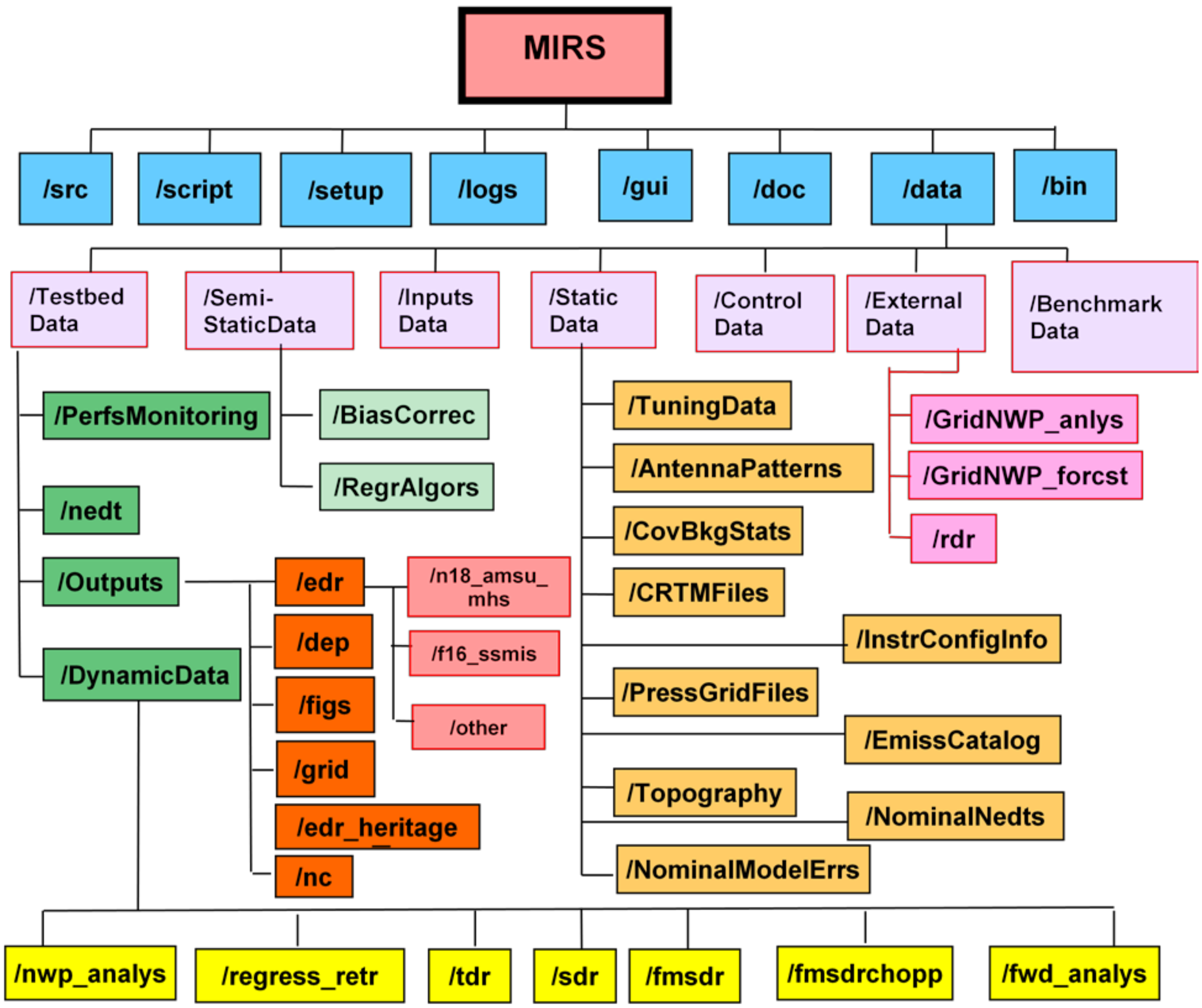


Figure 13. MIRS top level directory with expanded data directory structure and flow

Table 7 presents the sizes of MIRS stored data files in the data directories: /InputsData, /ExternalData, /TestbedData, /SemiStaticData, /StaticData, and /ControlData for NOAA-18 AMSU-MHS for 1 orbit on 2006-02-01. In the /TestbedData directory, only the outputs and the NEDTs (that are stored) are listed. The dynamic intermediate files are not stored and thus are not presented.

Data	Input File Names	Size(KB)	Total (MB)
InputsData (List of files used for different applications)	n18_atmNWPanalys.list	4	0.108
	n18_nedtDirs_amsua.list	4	
	n18_ecfmsdrFiles_4ApplyRegress.list	4	
	n18_nedtDirs_amsuam.list	4	
	n18_ecfmsdrFiles_4Bias.list	4	
	n18_nedtDirs_mhs.list	4	
	n18_ecfmsdrFiles_4Chopping.list	4	
	n18_NWPanalysFiles_4Bias.list	4	
	n18_ecfmsdrFiles_4nwp.list	4	
	n18_NWPanalysFiles_4Regress.list	4	
	n18_ecfmsdrFiles_4regress.list	4	
	n18_NWPanalysFiles.list	4	
	n18_ecfmsdrFiles.list	4	
	n18_rdrFiles_amsua.list	4	
	n18_edrFiles_4Bias.list	4	
	n18_rdrFiles_mhs.list	4	
	n18_edrFiles_4png.list	4	
	n18_sdrFiles_amsua.list	4	
	n18_edrFiles.list	4	
	n18_sdrFiles_mhs.list	4	
	n18_fmsdrFiles_4Bias.list	4	
	n18_sfcNWPanalys.list	4	
	n18_fmsdrFiles.list	4	
	n18_tdrFiles_amsua.list	4	
	n18_FullOrbitEDR_4Merging.list	4	
	n18_tdrFiles_mhs.list	4	
	n18_FWDanalysSimulFiles_4Bias.list	4	
SemiStaticData	/biasCorrec	52	0.340
	/regressAlgors	288	
StaticData (Files that are not or are rarely changed)	/AntennaPatterns	0	75.932
	/CRTMFiles	216	
	/NominalModelErrs	0	
	/PressGridFiles	8	
	/TuningData	4	
	/CovBkgStats	57440	
	/InstrConfigInfo	4	
	/NominalNedts	0	
	/Topography	18264	
ExternalData (All data generated by external sources)	/gridNWP_analys	88708	225.428
	/gridNWP_forcst	0	
	/rdr	136720	

ControlData (files used on control for the different applications)	n18_colocNWPwRAD_amsuam.in	4	0.304
	n18_Inputs4BiasComputation.in	4	
	n18_rdr2tdr_amsua.in	4	
	n18_rdr2tdr_mhs.in	4	
	n18_mergeNEDT.in	4	
	n18_tdr2sdr_amsua.in	4	
	n18_tdr2sdr_mhs.in	4	
	n18_fm_amsua_mhs.in	4	
	n18_cntrl_fwd.in_0	4	
	n18_cntrl_fwd.in_1	4	
	n18_cntrl_fwd.in_2	4	
	n18_cntrl_fwd.in_3	4	
	n18_cntrl_fwd.in_4	4	
	n18_cntrl_fwd.in_5	4	
	n18_cntrl_fwd.in_6	4	
	n18_cntrl_fwd.in_7	4	
	n18_cntrl_fwd.in_8	4	
	7n18_cntrl_fwd.in_9	4	
	n18_cntrl_fwd.in_10	4	
	n18_cntrl_fwd.in_11	4	
	n18_cntrl_fwd.in_12	4	
	n18_cntrl_fwd.in_13	4	
	n18_cntrl_fwd.in_14	4	
	n18_cntrl_fwd.in_15	4	
	n18_fmsdr2ecfmsdr.in	4	
	n18_Chopp.in	4	
	n18_Inputs4regressGen.in	4	
	n18_ApplyRegress.in	4	
	n18_CntrlConfig_1dvar.in_0	8	
	n18_CntrlConfig_1dvar.in_1	8	
	n18_CntrlConfig_1dvar.in_2	8	
	n18_CntrlConfig_1dvar.in_3	8	
	n18_CntrlConfig_1dvar.in_4	8	
	n18_CntrlConfig_1dvar.in_5	8	
	n18_CntrlConfig_1dvar.in_6	8	
	n18_CntrlConfig_1dvar.in_7	8	
	n18_CntrlConfig_1dvar.in_8	8	
	n18_CntrlConfig_1dvar.in_9	8	
	n18_CntrlConfig_1dvar.in_10	8	
	n18_CntrlConfig_1dvar.in_11	8	
	n18_CntrlConfig_1dvar.in_12	8	
	n18_CntrlConfig_1dvar.in_13	8	
	n18_CntrlConfig_1dvar.in_14	8	
	n18_CntrlConfig_1dvar.in_15	8	
	n18_Inputs4BiasVerification.in	4	
	n18_Inputs4pngFigsGener.in	4	
TestbedData	/nedt	4	
	/Outputs	100,048	100.052
Total data volume with external data (for NOAA-18 AMSU-A/MHS per 1 orbit)			402.164 MB
Total data volume without external data			176.736 MB

Table 7. Sizes of MIRS data files for NOAA-18 AMSU-/MHS for one day processing

6.2 Computer Resource Requirements

Presented below are the necessary current computer resource requirements for MIRS. Figure 14 summarizes the current MIRS STAR IT Architecture. Due to the large number of satellite/sensor data being handled, MIRS processing is run on several different multi-CPU Linux servers. Generally speaking, the servers have on the order of 12 to 24 CPUs, with a single CPU processing speed at or above 2.0 GHz, with multi-TB of available disk space; it is a multi-user environment working with a single gate keeper to control code updates to the repository.
The MIRS package is version-controlled by the Subversion configuration management tool, and the repository is backed up daily through the STAR backup procedure (under "/net/backup"). The performance monitoring is done on a machine where products are visualized. The same online monitoring tool and machine is expected to be used.

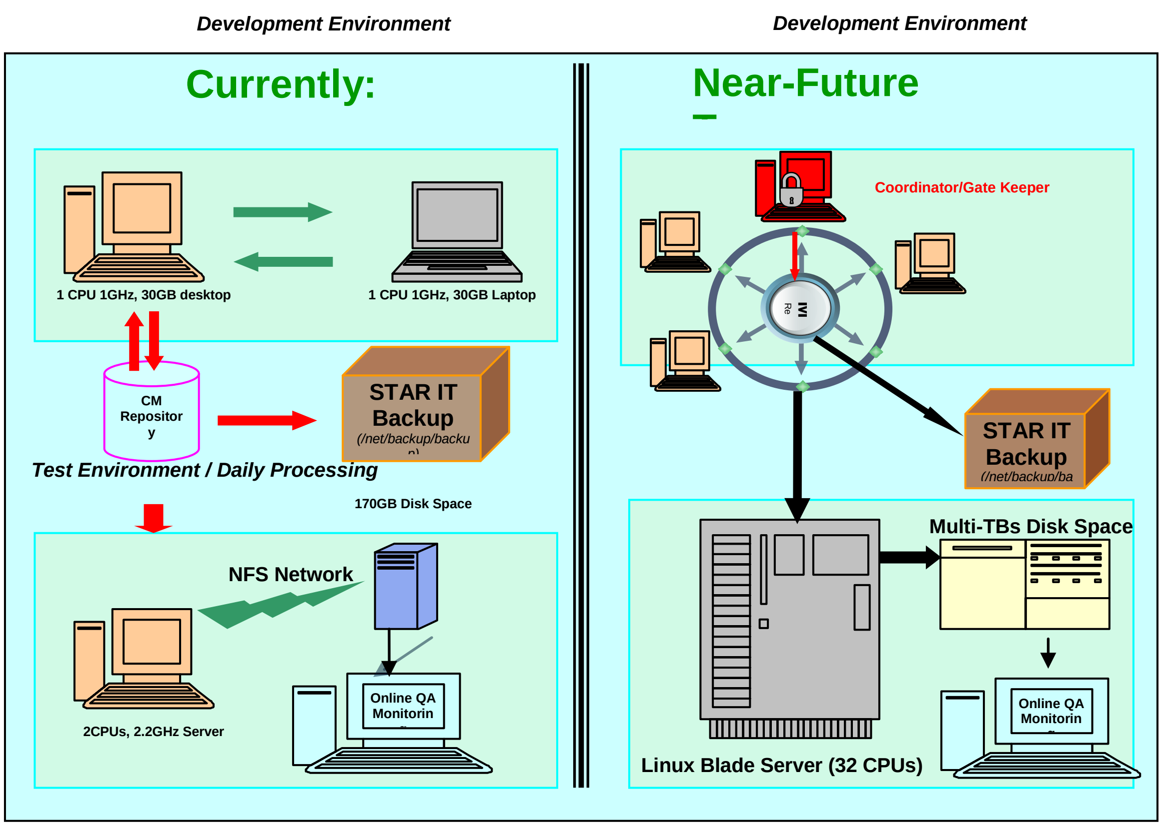


Figure 14. MIRS STAR IT architecture (current and near-future)

6.2.1 Hardware Requirements

Currently, 1CPU, 1GHz Memory, and 5 GB available hard drive space is the minimum required for the purpose of the environmental and daily processing test of MIRS. In the near future, CPUs will be expanded to 32 for the Linux blade server in STAR, and the disk space will be replaced by multi-TBs storage. The current hardware requirements (CPU, RAM, and Hard Disk space) in MIRS package with NOAA-18 for one day processing are described in Table 8.

MIRS Requirements	Minimum# of CPUs	Minimum RAM	Min. Hard Disk Space	Platform Type
	1	1 GB	5 GB	Linux/Unix

Table 8. Hardware requirements in MIRS package for NOAA-18 AMSU-MHS for one day processing

1.1.1 Software Requirements

The MIRS core retrieval system, the testbed and the external tools have been implemented with a standard Fortran-95 (for the algorithm part), BASH (for the testbed script) and IDL (for the external tools). It was successfully tested on several platforms including Linux and IBM/AIX. Note that for the latter, a compiler bug was discovered. As a result, an older version of XLF on IBM/AIX was used for which a patch exists. The MIRS uses dynamic memory allocation, which makes it sensor- and parameter-independent. It is modular, so its maintenance and updates are easy to implement. Table 9 summarizes the software requirements.

MIRS Requirements	Operating System	F95 Compiler	Commercial Software(s)	Freeware(s)
	Linux/AIX	Standard	IDL v5 and Up	BASH, JAVA v1.6 and up

Table 9. Software requirements

6.3 Performance Statistics

Presented here is the performance statistics for the NOAA-18 AMSU-MHS system. The testbed automatically generates mapped products in PNG format for monitoring and visualization. The channels noise NEDT is monitored daily as part of the testbed since it is an input to the 1DVAR processing. The date 2006-02-01 is chosen as the sample input files for N18 AMSU/MHS. The performance steps and times are listed in Table 10 for 1 orbit. In this case, the total number of profiles submitted and processed is 25800.

Step	Performance	Time (s)	Description
1	RDR to TDR conversion (AMSUA)	1	TDR is the antenna temperatures file
2	RDR to TDR conversion (MHS)	1	
3	Merge AMSUA & MHS NEDT files into a single file	2	Merge AMSUA & MHS NEDT files
4	TDR to SDR conversion (AMSUA)	1	SDR is Brightness temperature file where antenna patterns corrections have been applied
5	TDR to SDR conversion (MHS)	1	
6	Footprint Matching (FM) AMSUA/MHS	8	FMSDR contain Footprint matched brightness temperature file
7(*)	Collocation of NWP analyses with	33	File contains NWP information collocated

	radiances (for bias)		
8(*)	Application of forward operator on NWP analysis data	460	CRTM is used as a forward operator
9(*)	Determination of the bias (by comparing FWD simulated and measured)	49	Bias determination
10 (**)	Chopping the FM-SDR file into pieces for faster processing	2	Efficient processing
11 (*)	Apply regression-based algorithms on FM-SDR radiances	339	Regression processing
12		21	
13	EDRs retrieval from the EC-FM-SDRs	2312	1dvar algorithm is applied to retrieve the EDRs
14 (**)	Merge the mini EDR files into full orbit file		Merging EDR files into orbit file.
15 (***)	Verification of Bias efficiency (radiometric and geophysical)	206	Verification of Bias efficiency
16 (***)	EDRs Figures generation(maps)	403	Image PS/PNG files
17 (***)	EDRs Figures generation (maps) in PNG format	180	Image PNG file
18 (***)	NEDT monitoring (plots)	3	NEDT monitoring figures
19 (***)	Cleaning Up	1	Disk cleaning/purging
Total time		4025	
Average time for 1 profile (all steps)		0.156	
Average time for 1 profile (operational steps only)		0.091	

(*) : These applications produce the semi-static files. These files are not run daily, but only when an update is needed.
(**): These steps are optional. They aim at taking advantage of multiple CPUs capability if available.
(***): These steps are optional because they are not needed for the operational processing.

Caution: These estimates are for information only. They do include CPU, IO, and system times. They have been obtained using a Linux server. (Speed 2.8 GHz). They do not assume that MIRS was the only application running on the machine.

Table 10. Performance steps and times: 1 orbit from 2006-02-01 for N18 AMSU-MHS

The example of the retrieval time breakdown for 1 orbit running 1DVAR is shown in Table 11. This breakdown is generated automatically by the code itself. The total number of profiles submitted and processed is 25800.

Procedures	Time (s)	Percentage (%)
Uploading/Initialization	1.78000	0.09449
Reading radiance/aux data	1.08961	0.05784

Pre-classifying/Preparing matrices	356.98218	18.94973
Setting up scene/Merging cov, EOF proj	208.74754	11.08097
Forward Operator	1252.07214	66.46391
Inversion & Pre-Post Processing	63.10627	3.34988
Error Analysis & characterization	0.05978	0.00317
Total	2233.22998	100.00000%

Table 11. Examples of the retrieval time breakdown in the advanced retrieval (1dvar). The small running time difference with respect to 1DVAR as reported in Table 10 is due to the 1DVAR being run separately and in synchronization with other processes (through SCS).

Section 7.0 Maintenance and Monitoring

This section provides information for MIRS maintenance and monitoring purposes. First, it briefly describes data flow, software organization and user interaction as prerequisite information for maintenance and monitoring. Next, routine maintenance requirements such as calibration updates and new satellite implementation are described, followed by descriptions and presentations of monitoring tools.

7.1 Environment

Before any maintenance is performed, a pre-requisite is to understand the MIRS environment in terms of data structure and software organization, so that a user interested in performing any maintenance work would be able to easily locate the software or the database to be maintained. This section will present the MIRS environment.

7.1.1 Data Flow

A generic description of how the MIRS package is organized in terms of data structure was given in Figure 12. This data directory can by modified by the user through the Paths & Configuration File (PCF). If the directory structure shown in Figure 12 is not appropriate for the particular application the user is interested in, it is possible to redo the whole directory structure by modifying only the PCF file. For instance, it is possible to consolidate temporary files (currently separated by types, sensors, etc) into a single directory called "*temp*" then refer all directories in the PCF file related to the temporary directories to consolidate to the full path of this "*temp*" directory.

The data flow in MIRS is controlled by the user through the SCSs. SCSs can be modified manually or automatically via the GUI interface. An example real time application was provided in Figure 8.

7.1.2 Software Organization

The software is organized in the "`/src`" top level directory. For every directory under "`/src`", there is a makefile that can be used independently to compile and run (if it is a main program) the particular application under that directory. There is also a general makefile in "`/src`" which executes every sub-directory's makefile.

The chart presented in Figure 15 depicts the expanded "testbed" source code directory structure. It contains codes for MIRS real time applications. Short descriptions of MIRS main source code sub-directories were provided in section 6.1.1. MIRS applications (source code and directory) were described in Table 4.

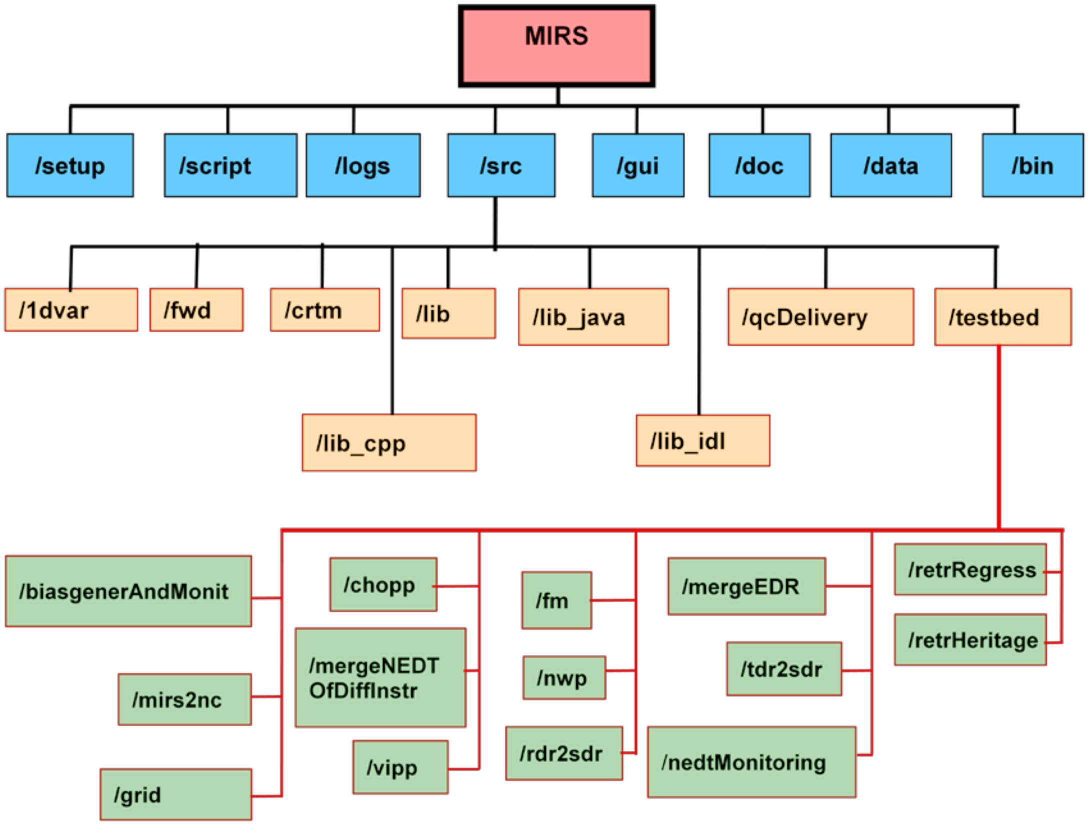


Figure 15. Source code directory tree of MIRS package with the expanded testbed sub-directory structure. Testbed sub-directories contain code for specific MIRS applications.

7.1.3 **User Interaction and the MIRS Control Panel (*MCP*)**

The MIRS allows different levels of interaction which depend mainly on the need and level of expertise of the user. Figure 16 presents a diagram that shows the top-to-bottom interactions that the user can have with MIRS with increasing level of expertise.

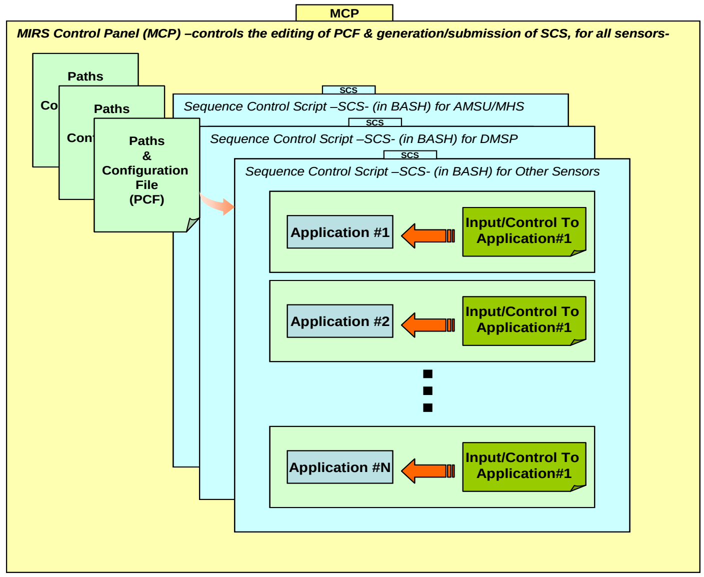


Figure 16. Description of the three (nested) levels of user interaction. The innermost small blocks represent the low-level interaction consisting of individual applications and the control files as input (indicated by red arrows) to their executables. The mid-level interaction is represented by the (outer) panels with the PCFs and SCSs for a particular satellite that control the sequencing of the individual applications. The outermost panel represents the highest level of interaction through the GUI-based MCP.

The highest level would only require working with a GUI-based tool called the MIRS Control Panel (MCP). At the mid-level interaction, the user would directly edit the Sequence Control Scripts (SCS) for an individual sensor, and the associated Paths & Configuration File (PCF) to control which process to run, what options to turn ON/OFF, etc. At the lowest level of interaction, the user would directly execute the individual application (i.e. 1dvar, footprint matching, etc) and edit the associated control/input file (a namelist file for F95 applications and a similar controlling-parameters list file for IDL applications). Further details are given below

7.1.3.1 High-Level Interaction

At the highest level, the user would make use of the MIRS Control Panel (MCP) to run and control MIRS processes and applications. The MCP is a Graphical User Interface (GUI) Java-based tool aimed at making the use of MIRS as user-friendly as possible. It also makes the maintenance of the MIRS system as seamless as possible, since all tasks are centralized in one place. The MCP allows the user to interactively adapt the data directory structure to the user's particular needs, modify compilation and running using different scientific options, run each or a collection of MIRS processes, dynamically generate BASH scripts in order to later submit them through cronjobs and/or other schedulers, switch between the daily and orbital mode processing and select parameters to be retrieved in 1DVAR.A snapshot of the MCP display is presented in Figure 17.

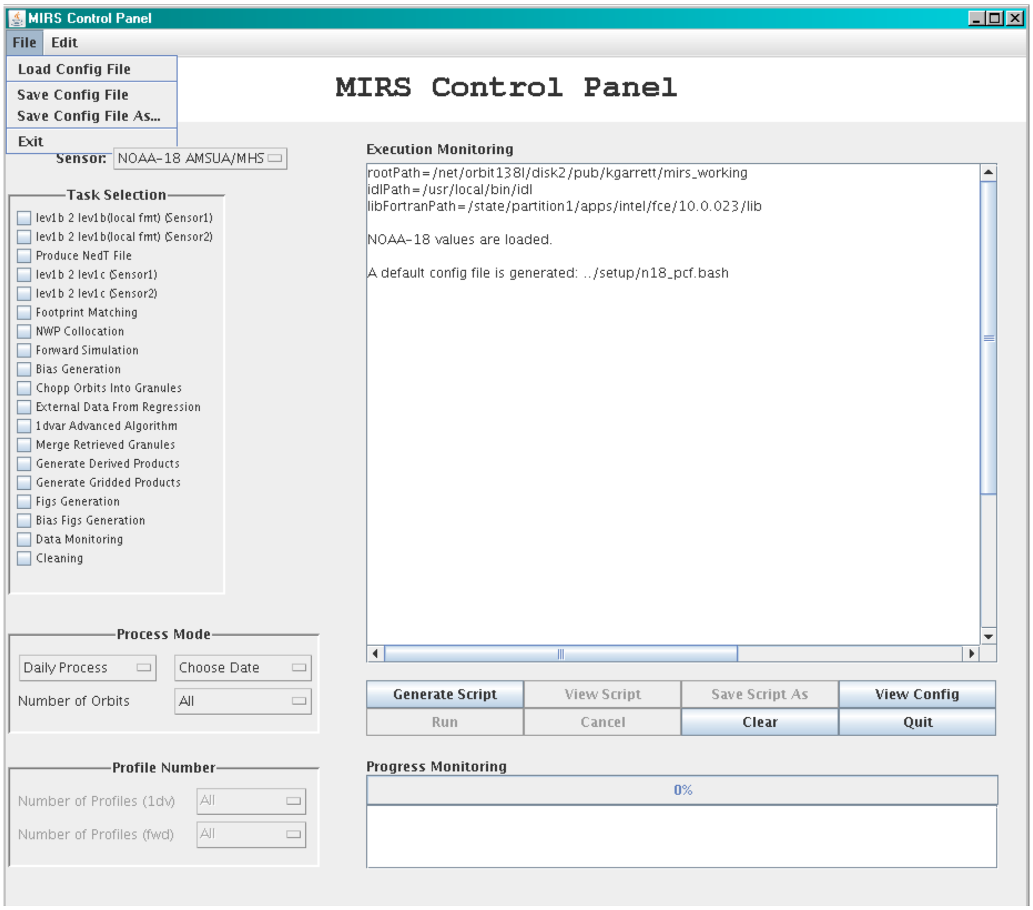


Figure 17. Snapshot of the MIRS Control Panel that shows some of the options available in the main window.

It is important to note that the maintenance of the MCP would require modifying the main program `mcp.java` under the "`gui`" top level directory as well as the following library files: `PathDialog.java` and `PreferenceDialog.java`, and the `lib_java` `scripts_functions.bash` under the "`/src/lib_bash`" sub-directory.

To activate the MCP, the following command lines are executed in the "`gui`" top level directory:

`make` - for compiling and,

`make run` - for running the MCP. The MCP panel will pop up.

7.1.3.2 Mid-Level Interaction

As part of the MIRS package, a number of Sequence Control Scripts (SCS) written in BASH are distributed as benchmark files. Corresponding to these scripts are the Paths and Configuration Files (PCFs) from which the SCS extracts the information on configuration and paths. The PCFs and SCSs can be interactively modified and generated via the MCP. The user can also manually generate his own SCSs and PCFs by changing the content of the provided SCSs or the PCFs. An example of the PCF content was presented above (Figure 9).

7.1.3.3 Low-Level Interaction

Low-level interaction requires the user to control each individual application, e.g. 1dvar, forward operator, footprint matching, by manually editing the control file (namelist) and modifying the inputs required for the particular application. The control files are located in the "/data/ControlData" sub-directory and are distinguished by process and sensor represented by prefix in their names. The input files named in the control file are located in the "/data/InputsData" sub-directory. The executables for all applications are consolidated in the "/bin" top level directory. An example of manually running an application through this low-level interaction is the following command line which executes the footprint matching process for NOAA-18 AMSU-MHS:

~user/mirs/bin/fm < ~user/mirs/data/ControlData/n18_fm.in

An example of the content of the control file is presented in Figure 18.

```
&ContrlRDR2TDR
rdrfileList_mhs='/net/orbit006L/home/sidb/mirs/data/InputsData/n18_rdrFiles_mhs.list'
pathTDR_mhs='/net/orbit006L/home/sidb/mirs/data/TestbedData/DynamicData/tdr/mhs_tdr_data/2007-01-28/'
NEDTfile='/net/orbit006L/home/sidb/mirs/data/TestbedData/nedt/mhs_nedt/n18_mhs_nedt_2007_01_28.dat'
InstrConfigFile='/net/orbit006L/home/sidb/mirs/data/StaticData/InstrConfigInfo/mhs/InstrConfig_mhs_n18.dat'
WarmTargetFile='/net/orbit006L/home/sidb/mirs/data/TestbedData/nedt/mhs_nedt/n18_mhs_wt_2007_01_28.dat'
norbits2process=50
LogFile='/net/orbit006L/home/sidb/mirs/logs/n18_logFile_2007_01_28.dat'
/
```

Figure 18. Sample control file for the RDR2TDR application. Content includes list of raw data (AMSU and MHS), path, where to put the TDR files, the NEDT and warm target files that will be generated on the fly, the instrumental configuration file, the number of orbits to process and the log file name. This control file is generated automatically by the SCS (bash script) which itself is generated automatically by the MIRS control Panel (MCP), a GUI-based Java script.

7.2 Science Maintenance

MIRS is designed in a highly flexible way that makes it easy to maintain and easy to keep up to date with cutting edge science. The entire radiative transfer modeling is implemented in MIRS by the Community Radiative Transfer Model (CRTM). A large part of the maintenance of the MIRS science is to keep up to date with the latest version of CRTM. CRTM is funded, maintained, expanded and validated independently from MIRS within the Joint Center for Satellite Data Assimilation (JCSDA).

The plan is to update the CRTM forward operator within MIRS every year. This would become an easier task once the latest version is implemented, as the any changes to the interface would be minor.

Other science maintenance require updating library functions, I/O modules, as well as adding new capabilities.

7.2.1 New Satellite Implementation

One characteristic of MIRS is that it is designed to be sensor-independent from a software point of view. There are certain items that need to be generated to expand the capability of MIRS to new sensors. These will be covered in this section.
A new satellite/sensor implementation requires the following:

- Generation of a new CRTM coefficient files that apply to the particular sensor. This is usually a trivial step as CRTM comes already with a set of pre-computed files for many existing and future instruments.
- Generation of a new covariance matrix for the surface parameters. This is due to the fact that the surface covariance matrix contains the emissivity which is sensor dependent. An external tool associated with MIRS is used to generate this covariance matrix. It also requires generating new emissivity datasets over land, sea ice and snow.
- Generation of a new transformation matrix (based on eigenvalue decomposition) associated with the above covariance matrix file.
- Generation of a new surface pre-classification algorithm to determine the surface background type so as to choose the surface covariance matrices.
- Generation of bias correction values to measured brightness temperatures. Satellites with new or even heritage sensors contain their own scan-dependent biases relative to the forward model CRTM.
- Development of new sensor-specific application. An example of a sensor-specific application is the footprint matching: every sensor has its own geometry and measurement characteristics and therefore requires its own footprint matching application.
- Expansion of the MCP to generate sensor-specific SCSs and PCFs.

7.2.2 Adding a New Sub-System

The first step is to develop the new application independent of MIRS. The recommended way of passing the inputs is through namelists for Fortran applications. For IDL applications a namelist type of file should also be put together and read from the IDL subroutine. An example of the definition/call of the IDL routine is

```
PRO Calib_generic_rad,Namelist=NamelistInputs,sensor_id=sensor_id
```

Once the application has been developed and tested, the next step is to make it part of the sequence control script. This is done through the MCP, which has a user selectable sequence for every sensor.
The regular user will find this pre-set sequence already available through MCP.

7.2.3 Modifying an Existing Sub-System

In modifying an existing subsystem, there are two possible cases:

- The change requires no modification to the interface: this means that the inputs required by the subsystem are still the same and the output items are also identical to the existing ones. An example might be a change in the integration of the TPW within a particular application. In this case, the only change required is within the application itself. No changes will be made to the MCP and therefore no changes will be impacting the automatically generated SCS and PCF.

- The change requires a modification in the way the subsystem interfaces. In other words, the changes require that the inputs and/or outputs be modified (reduced number of arguments, augmented, changes in the type of arguments, etc.). This requires making the necessary changes to the application itself but also to the MCP. The change in the MCP would mean that the generated SCS and associated PCF will carry a few extra (or few less) arguments when calling the application.

7.3 Library Maintenance

Library maintenance is made easier due to MIRS high modularity. All source codes and library modules are centralized in the "`/src`" top level directory. Library module sub-directories are classified by language, e.g., the "`/lib`" sub-directory contains Fortran, the "`/lib_idl`" contains IDL, and "`/lib_java`" contains Java libraries. Each library sub-directory contains domain-specific modules/sub-directories including input/output, forward operator, inversion processing, mathematical functions, noise subroutines, pre and post-processing, QC-related modules, utilities as well as miscellaneous functions.

7.4 Image Display and Monitoring

The retrieved MIRS products are converted from the internal binary format into several user-driven formats. Along with the products dissemination, a tool has been designed to allow the monitoring of global and regional maps of the MIRS products. This MIRS Online Monitoring Tool (MOMT) can be accessed from the following URL address.
"http://mirs.nesdis.noaa.gov/".
From this MIRS web page, MIRS products are presented with figures through a set of visualization and analysis tools. Images on the same page are displayed by choosing the options such as number of panels (1 – 4), sensor (AMSU/MHS, SSMI/S), product, region (global, North America, US), node (ascending, descending), surface (land, sea, all), algorithm (heritage, advanced), date, and animation (N-increment) in the control panel. Therefore, time series, cross-sensor, cross-talk, and geophysical bias can be monitored. Figure 19 shows the MIRS on-line menus.

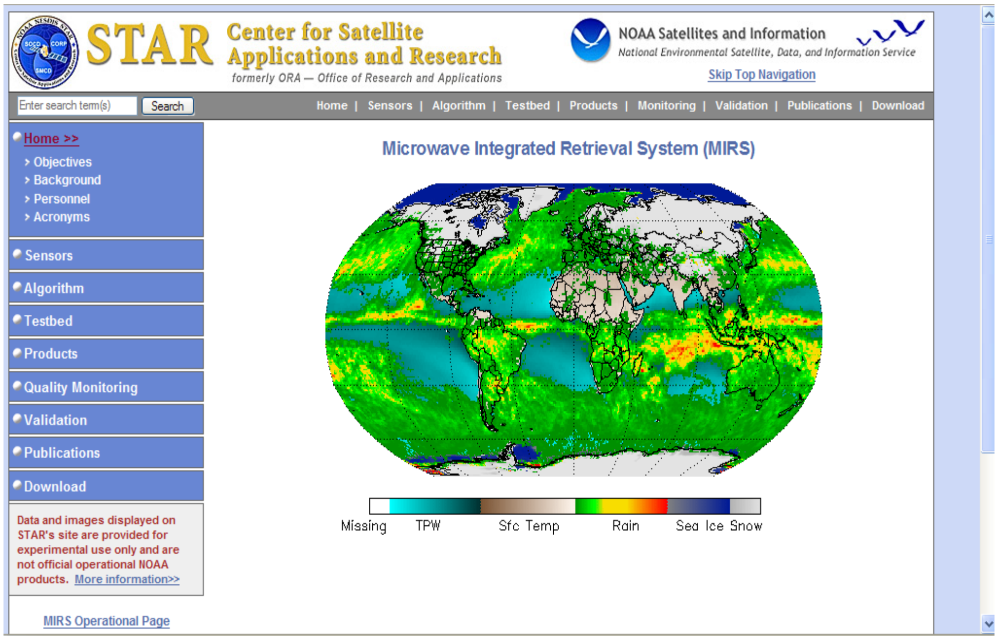


Figure 19. Snapshot of the MIRS online menus

A snapshot of the MOMT display features is presented in Figure 20. It is important to note that this tool is not tied to any particular sensor. It can display different products from different sensors corresponding to different dates and different regions.

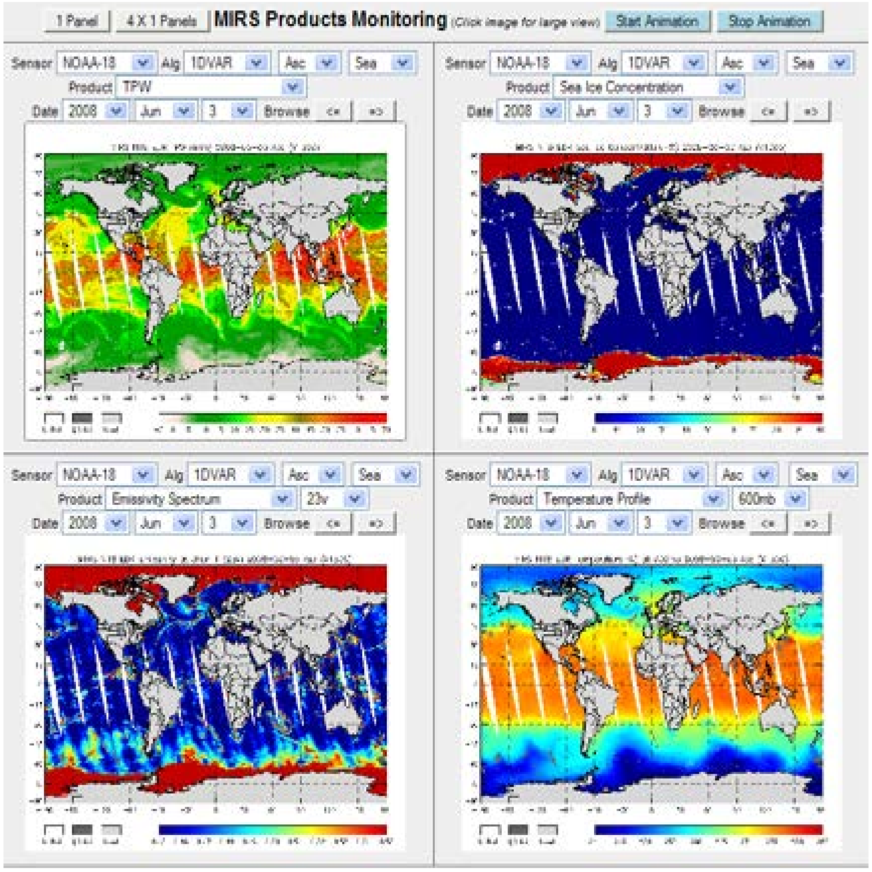


Figure 20. Snapshot of the MIRS Online Monitoring Tool (MOMT). Mirs Products Monitoring panl is shopwn that includes fours ub- panels.

7.5 Data Quality Monitoring

7.5.1 Radiometric Noise Monitoring

The knowledge of noise level (NEDT) is very important as it is used in both the estimation of the optimal solution (through the estimation of the measurements error covariance) as well as in the convergence criterion. It is optimal to correctly estimate the NEDT values. An overestimation of this NEDT level could result in loss of useful information content and lead to a smooth retrieval. An extreme case is when radiances are so noisy that almost any atmospheric state would fit them. The retrieval in this case would have no tendency to move from its background. The retrieval product would therefore be very smooth. With an underestimation of NEDT however, there is the risk of over fitting the noise signal, resulting in a noisy retrieval. These two risks associated with the knowledge of the true noise level impacting the radiances, are illustrated in Figure 21.

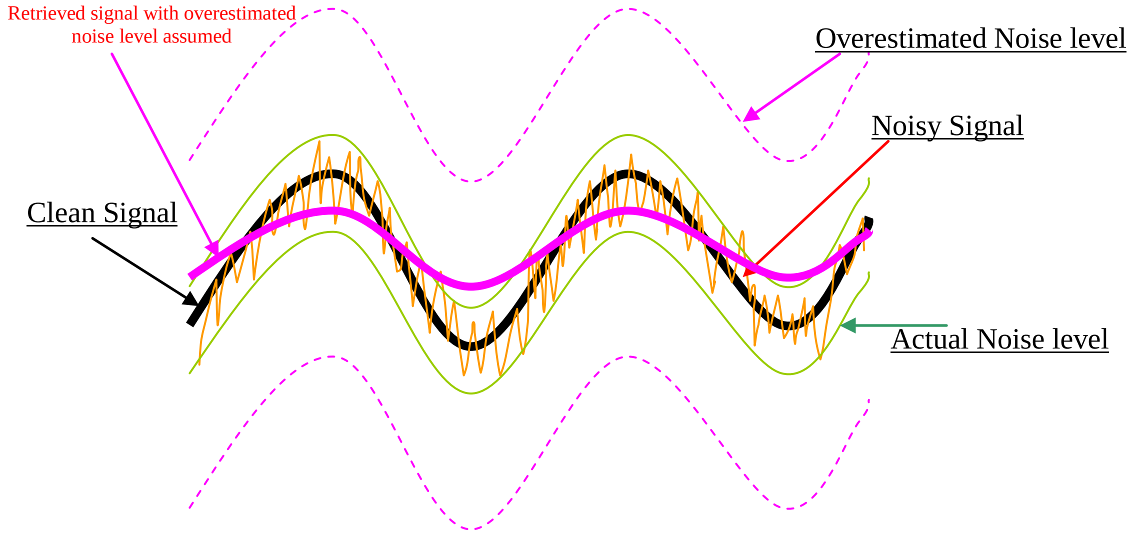


Figure 21..Importance of the knowledge of the noise level impacting the measurements. Shown is an example for the overestimated noise level. The black solid line represents the true signal (clean of noise). The orange solid line represents the noisy signal. The green envelope represents the true noise level while the purple dashed line represents the overestimated noise envelope. In case of the overestimation of the noise level, the risk is to end up with a smooth retrieved signal (solid purple line), therefore loosing on information content. In the case of underestimation (not shown), the risk is to overfit the radiances and resulting in either a noisy retrieval or non-convergence.

As part of the MIRS testbed, channel noise NEDT is monitored daily as it is an input to the MIRS retrieval system (it enters into the radiance uncertainty matrix and in the convergence metric computation). The NEDT monitoring figures are generated as well. They are cumulative; meaning that at day D, all data will be displayed from MIRS' inception to day D. The MIRS NEDT monitoring panel is shown in Figure 22 with four sub-panels showing examples of NEDT monitoring of four AMSU-MHS channels.

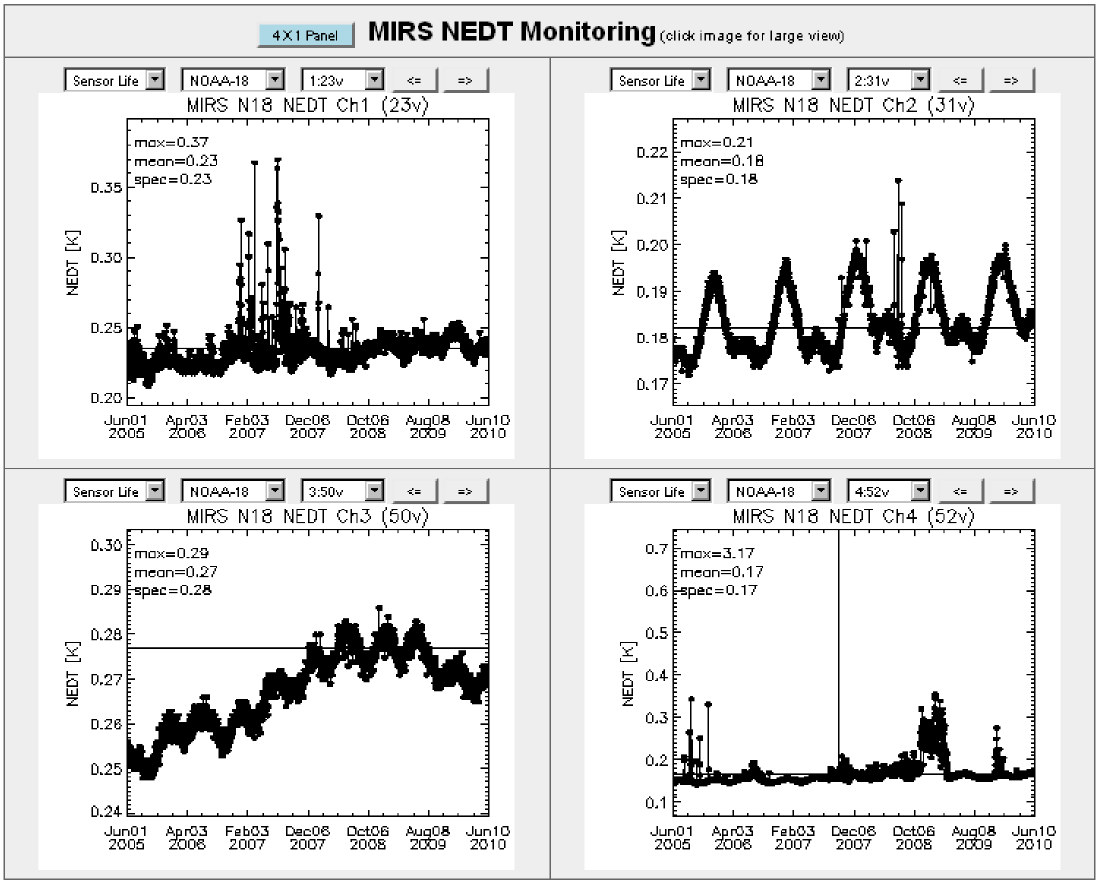


Figure 22. NEDT monitoring for NOAA18 AMSU-MHS testbed application.

7.5.2 **Convergence and Quality Control Monitoring**

Stability of the MIRS retrievals can be assessed from monitoring the rate at which convergence is reached (chi-squared metric), and also by the quality control (QC) flags denoting a good retrieval, a retrieval to use with caution, or a bad retrieval (QC=0, 1, and 2 respectively).

During the extraction of post-processed (DEP) fields-of-view point-by-point with the p2pDep.f90 program, critical information about the quality of the data is extracted and stored on an orbit by orbit basis. These files, by default, are stored in the 'data/TestbedData/PerfsMonitoring/sensor_id/orbitmon' directory. The file format is ascii and contains the year, ordinal date plus the fraction of the day, the convergence rate, and percentages of QC equal to 0, 1, and 2.

In the data monitoring step, all files within the 'orbitmon'directory are read in the MonitorQC_mirs.pro IDL program, and time series of convergence rates and percentages of QC flags are generated for monitoring purposes. Figure 23 shows and example of the plots generated.

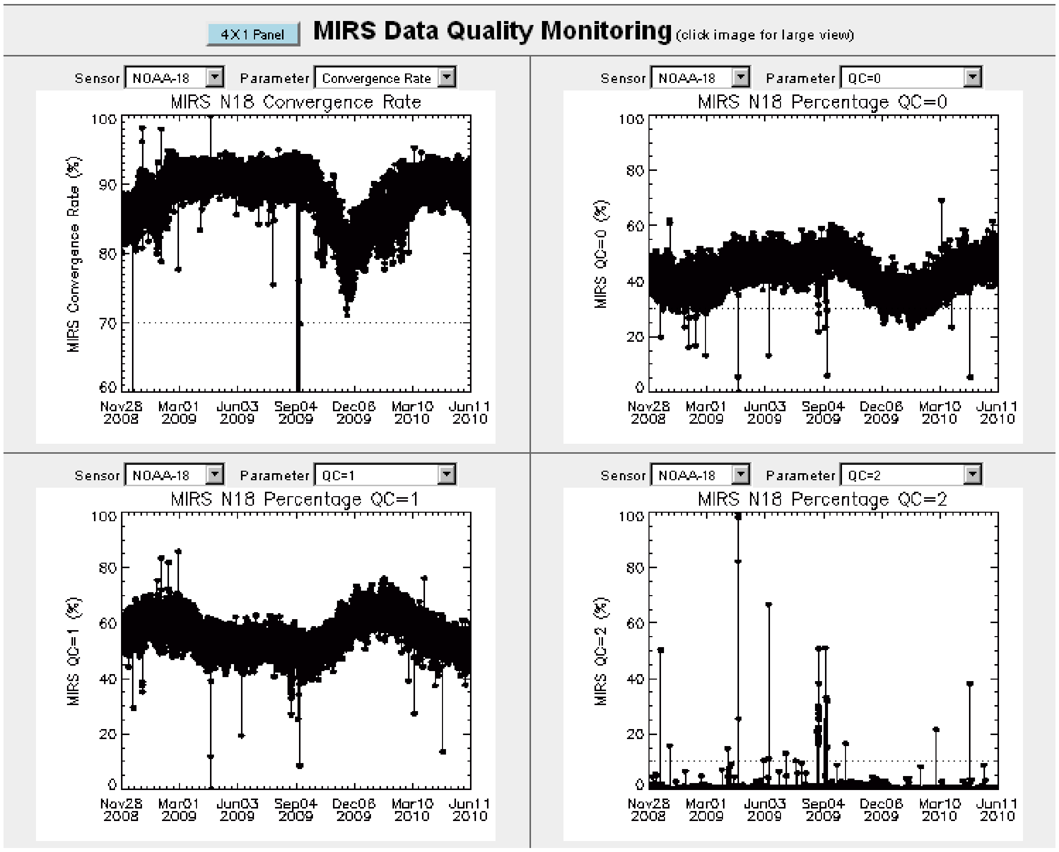


Figure 23. QC monitoring for NOAA-18NOAA18 AMSU-MHS testbed application.

7.6 Radiometric Performance and Bias Monitoring

For the purpose of the radiometric performance and bias monitoring through the MOMT, the brightness temperature bias can be assessed using simulations from the forward operator applied to a geophysical file of NWP data collocated with measurements. Figure 24 shows an example of the on-line displays of measured and simulated brightness temperatures (from GDAS data) at NOAA-18 AMSU-MHS 23 GHz channel and the differences in the brightness temperatures as well as the brightness temperature differences as a function of scan angle on October 25, 2008. Figure 25 shows the evolution of the brightness temperature mean bias for each scan position at the NOAA-18 AMSU-MHS channel 1.

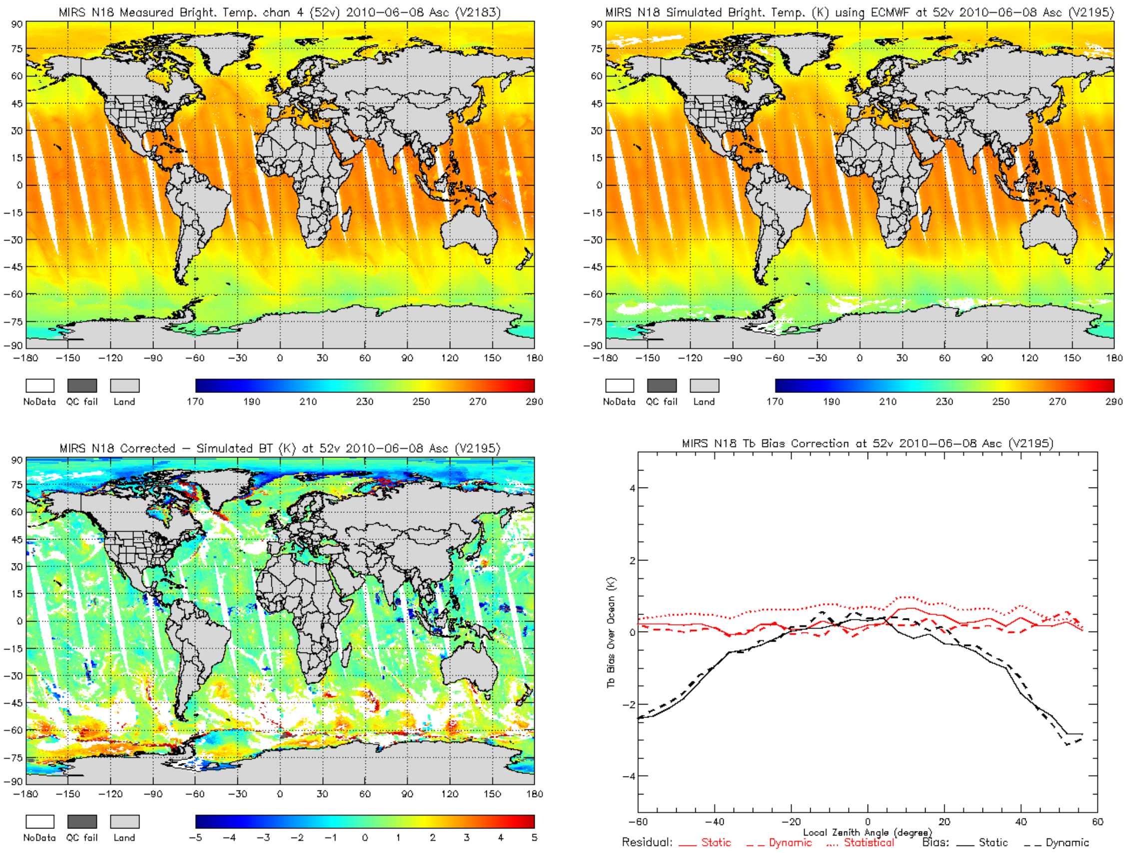


Figure 24. Brightness temperature measurements and simulation differences at NOAA-18 AMSU-MHS 52 GHz channel on June 8, 2010

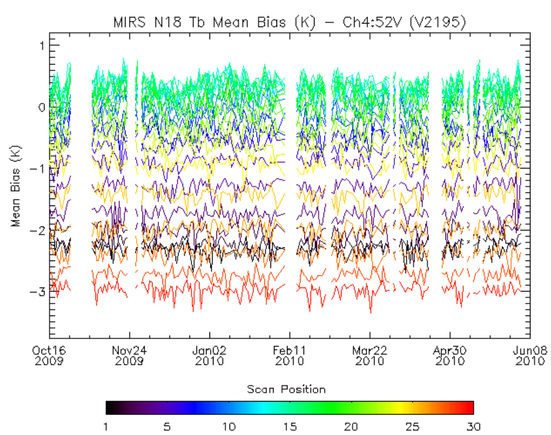


Figure 25. On-line monitoring of the brightness temperature mean bias for NOAA-18 AMSU-MHS channel 4.

7.6.1 **Geophysical Performance and Bias Monitoring**

Geophysical and bias monitoring is figures are generated by comparing the retrieved parameters with those from collocated NWP data (GDAS and ECMW). Figure 26 shows an example of the geophysical performance and bias monitoring figures generated for the TPW parameter over ocean.

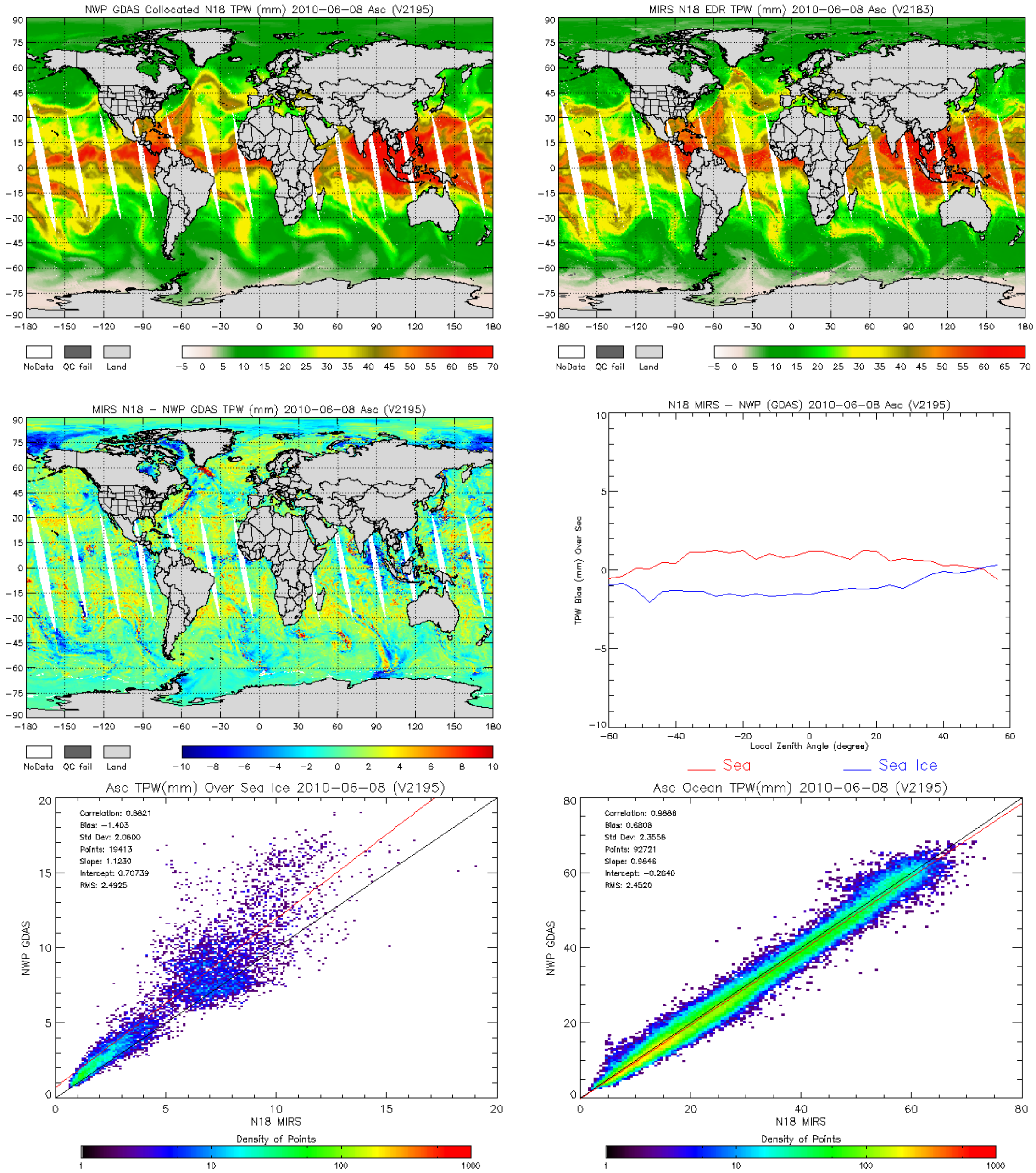


Figure 26. An on-line example of MIRS retrieved versus GDAS TPW. Top panel shows the retrieval maps, the middle panels show the differences (biases) globally and as a function of scan angle, and the bottom panels show scatter plots and performance statistics

7.6.2 Drift Monitoring

The MOMT system has an *Animation* option that allows the user to visualize the variability of any parameter with time. The starting and ending times and increment are all user-specified. This allows the user to detect any time trends in the products, in the biases or any drift that might be occurring as a result of a change in the sensor performances.

7.7 Products Monitoring

The MIRS products (radiometric and geophysical) and biases are available online. The same tool (MOMT) is used to monitor both radiometric and geophysical products as well as their performances (biases and scan dependence). Because MOMT allows the user to choose independent panels with different instrument/coverage/algorithm/product configurations, the following capabilities are available in the monitoring application:

7.7.1 Time Series Monitoring

The performance monitoring time scales can be categorized as near real-time, daily and long term. After pipeline processing or orbital processing, product images can be monitored in near real-time. Brightness temperature images and performance plots can be monitored daily. Climate can be monitored by N-Increment/N-time-scale static or dynamic time series. For the time series, animation can be run by clicking the buttons in the control panel. Figure 27 shows the MIRS time series monitoring tool, and Figure 28 shows an example of time series monitoring through the MIRS product monitoring tool.

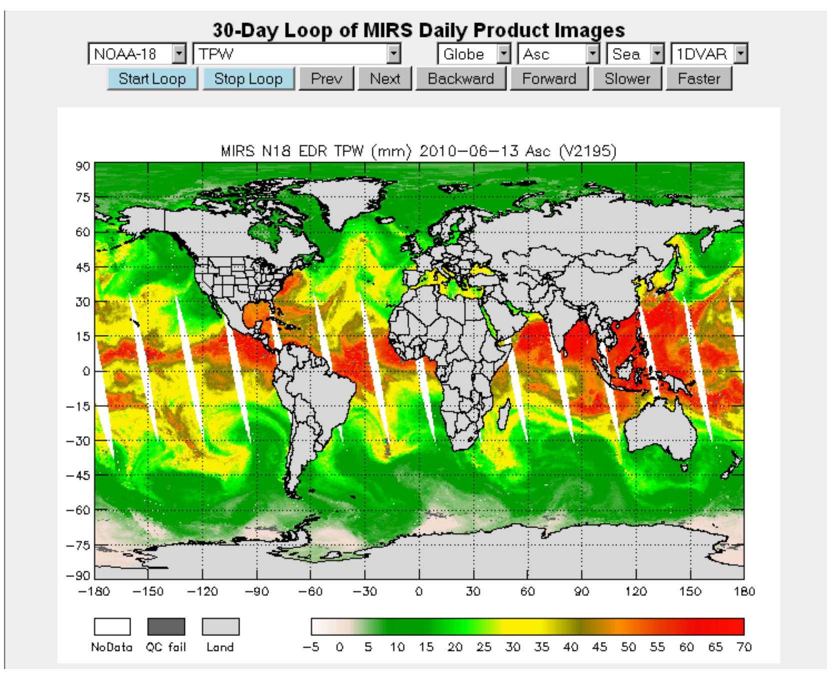


Figure 27. MIRS Time Series Monitoring Tool.

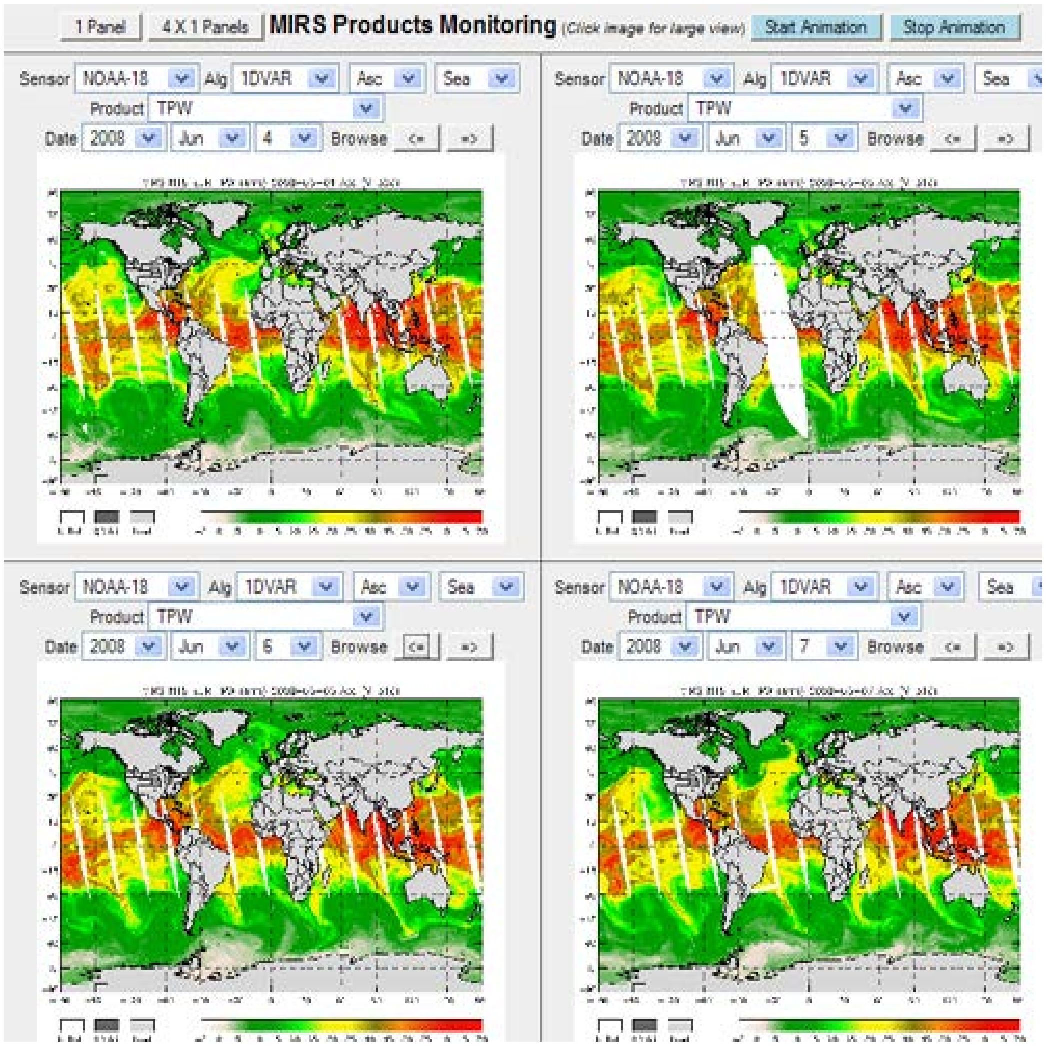


Figure 28. Example of time series monitoring (up to four days) using the Products monitoring four-panel display feature.

7.7.2 **Cross-Sensor Monitoring**

A cross-sensor intercomparison is also available in the products performance monitoring. Figure 29 shows an example of inter comparing NOAA-18 AMSU/MHS and SSMI/S retrievals. In this case, two panels are selected, total precipitable water is selected as the product, AMSU/MHS and SSMI/S are chosen as sensors from the control panel.

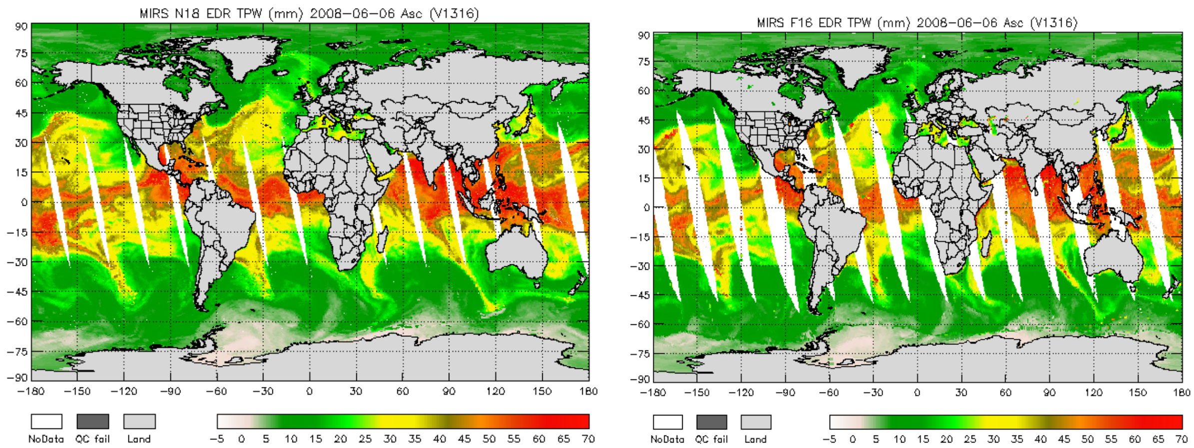


Figure 29. Comparisons of advanced products (retrieved TPW images) from AMSU-MHS and SSMI/S.

7.7.3 **Cross-Talk Monitoring**

Cross-talk is the dependence among the geophysical parameters (EDRs) errors. For example, the total precipitable water error could be correlated with the cloud liquid water amount. This type of monitoring is possible through MOMT. An example is displayed in Figure 30 where the retrieved EDRs are shown globally.

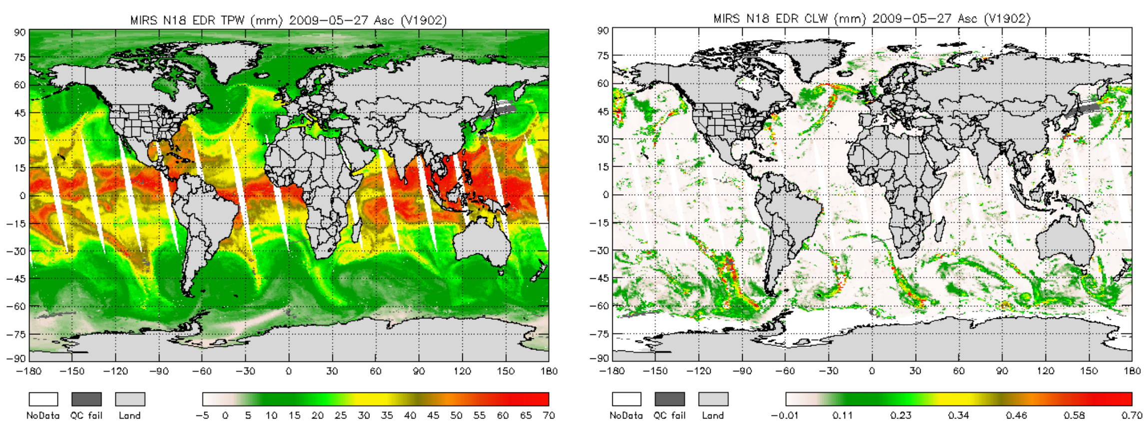


Figure 30. MIRS retrieved TPW (left) and CLW (right) on May 27, 2009

7.7.4 **Vertical Cross-Section Monitoring**

Vertical cross sections of temperature, water vapor, and hydrometeor profiles are also available. The user can monitor the profile data at any latitude for the global scale, the Gulf of Mexico**,** and Southeast Asia. Figure 31 gives a snapshot of this tool.

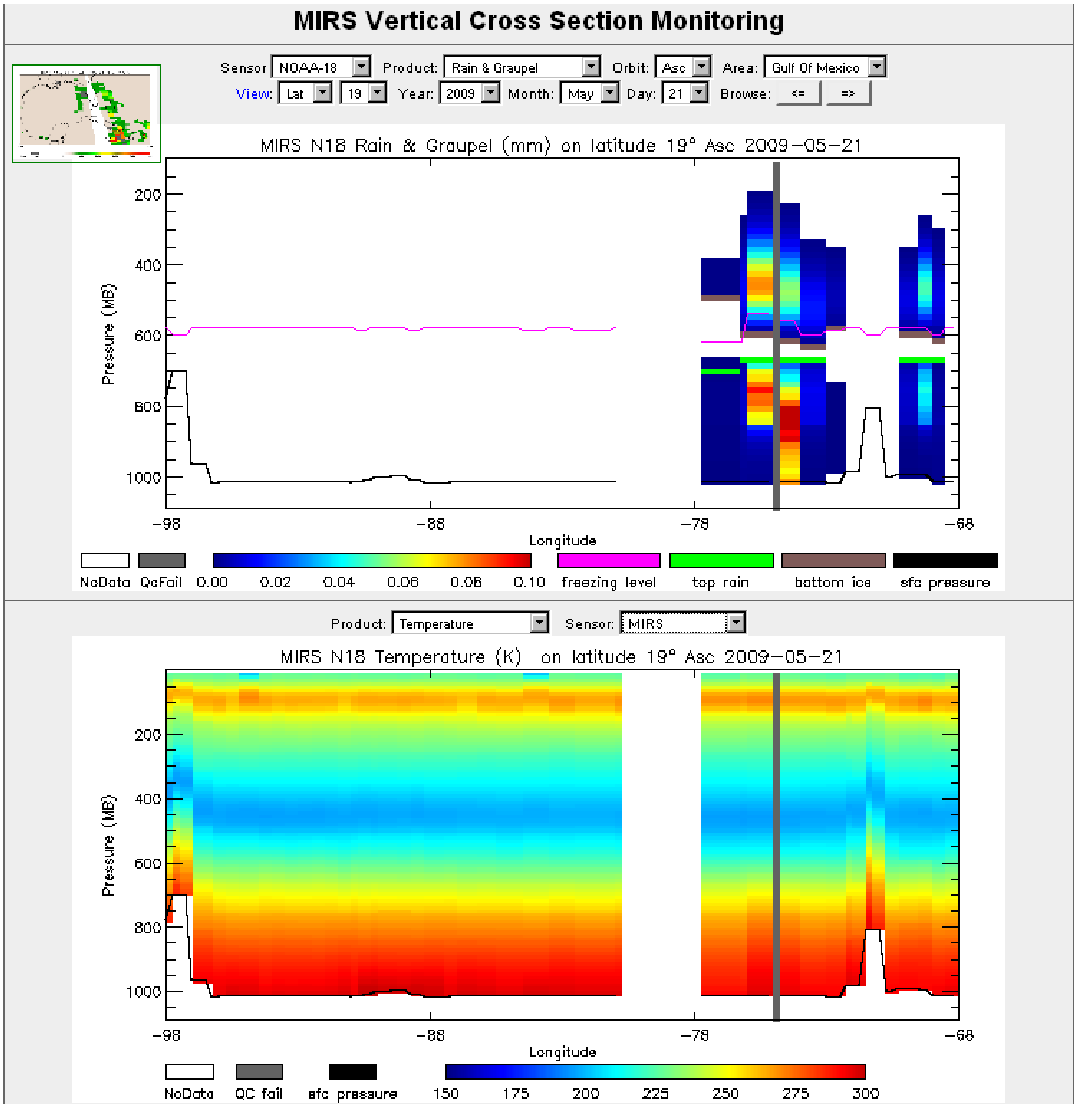


Figure 31. Snapshot of the Vertical Cross Section monitoring tool over Gulf of Mexico along Latitude 19º on May 21, 2009. The upper panel shows rain and ice profiles, freezing level, lowest layer with ice and highest layer with rain. The lower panel shows the corresponding vertical cross section of the temperature profile. The white area is the orbit gap.

7.7.5 High Resolution Monitoring

High resolution product monitoring is also available (currently only the Gulf of Mexico). Sensors processed at high resolution currently are: N18 AMSUA/MHS (global), NPP ATMS (global), TRMM TMI (Gulf of Mexico), F16 and F18 SSMIS (South China Sea). This is helpful in assessing MIRS precipitation monitoring and for monitoring/researching case studies as well. Figure 32 is a snapshot of the high resolution available for MIRS.

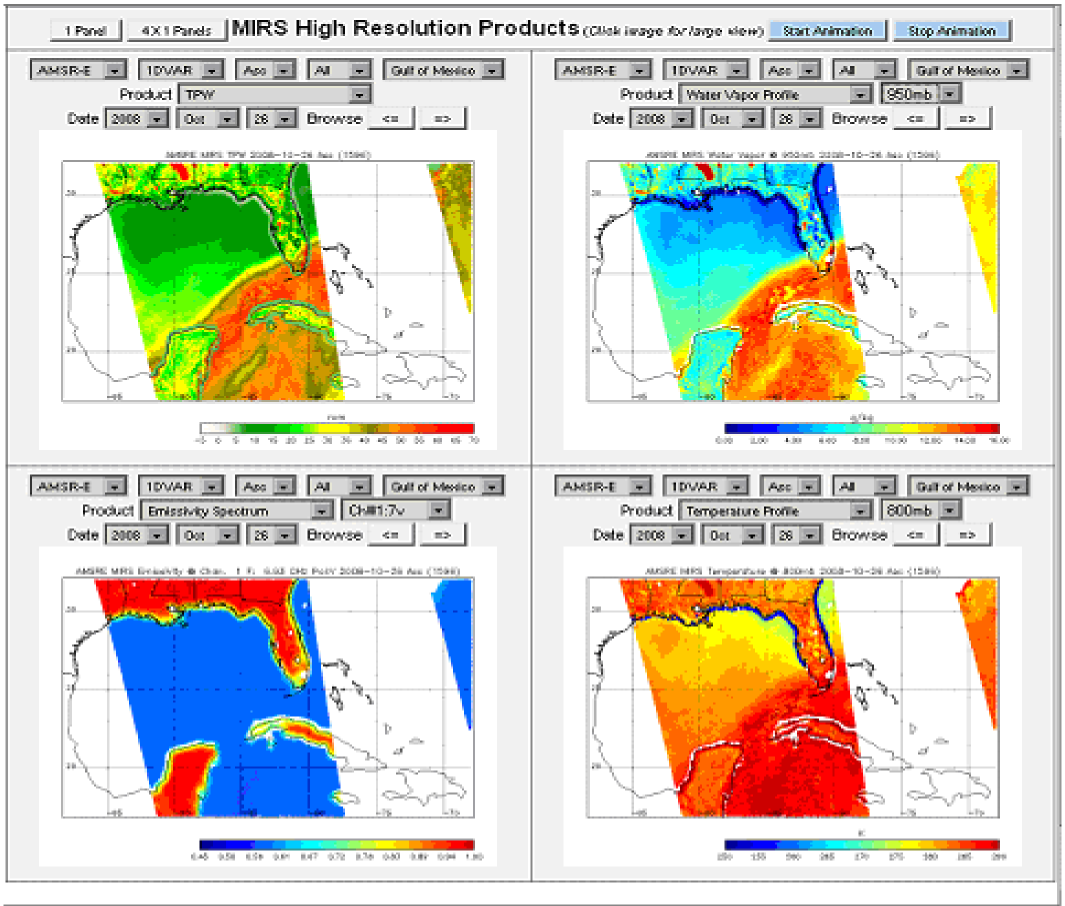


Figure 32. Snapshot of the high resolution available for MIRS

7.7.6 Climate Monitoring

Monitoring of geophysical parameters which are important for climate studies on multiple timescales can provided added information about the performance of the retrieval algorithm. The MIRS produces daily, pentad, and monthly composites for total precipitable water, rain rate, cloud liquid water, ice water path, and sea ice concentration. An example is given in Figure 33

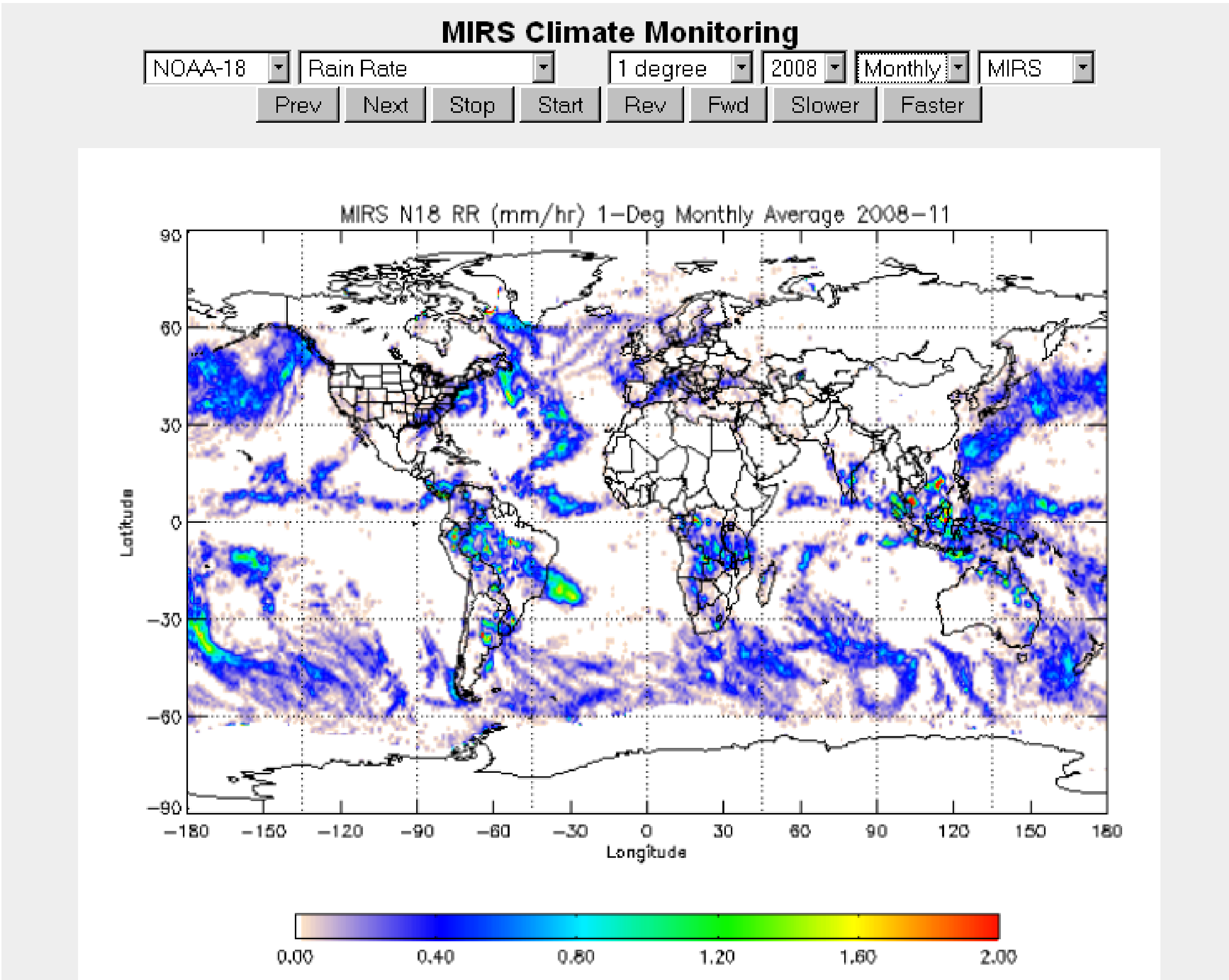


Figure 33. Snapshot of MIRS climate monitoring. Here the November monthly rain rate retrieval from NOAA-18 is shown.

Appendix A. Acronyms and Abbreviations

Acronym	Definition
1DVAR	One dimensional variational assimilation
AD	Adjoint
ADA	Advanced Double-Adding
AMSU	Advanced Microwave Sounding Unit
AOI	Area-Of-Interest
ASCII	American Standard Code for Information Interchange
ATBD	Algorithm Theoretical Basis Document
ATMS	Advanced Technology Microwave Sounder
CLW	Non-Precipitating Liquid Cloud Amount
CRTM	Community Radiative Transfer Model
DoD	Department of Defense
DAP	Delivery Algorithm Package
GUI	Graphical User Interface
GDAS	Global Data Assimilation System
ECMWF	European Center for Medium-range Weather Forecast
EDR	Environmental Data Records (Geophysical parameters)
EOF	Empirically Orthogonal Function
EOS	Earth Observing System
FM	Footprint-Matched
HRC	Hurricane Research Center
ICD	Interface Control Document
IST	Ice Surface Temperature
IWP	Ice Water Path
SFR	Snow Fall Rate
JCSDA	Joint Center for Satellite Data Assimilation
LST	Land Surface Temperature
NASA	National Aeronautics and Space Administration
NCDC	National Climatic Data Center
NEDT	Noise Equivalent Delta Temperature
NESDIS	National Environmental Satellite, Data, and Information Service
NOAA	National Oceanic and Atmospheric Administration
NPOESS	National Polar-orbiting Operational Environmental Satellite System
NPP	National Polar-orbiting Operational Environmental Satellite System (NPOESS) Preparatory Project
MVS	Minimum variance solution
MCP	MIRS Control Panel
METOP	Meteorological Operational Polar Satellite

Acronym	Definition
MHS	Microwave Humidity Sounder
MIRS	Microwave Integrated Retrieval System
MOMT	MIRS Online Monitoring Tool
MPS	Maximum probability solution
MSPPS	Microwave Surface & Precipitation Products System
OPTRAN	Optical Path TRANsmittance
OSDPD	Office of Satellite Data Processing and Distribution
PCF	Paths & Configuration File
QC	Quality Control
OM	MIRS Operations Manual
QP	Quality parameters
RWP	Rain Water Path
RR	Rain Rate
TDR	Temperature Data Record
TL	Tangent Linear
TPW	Total Precipitable Water
SCS	Sequence Control Scripts
SDD	System Description Document
SDR	Sensor Data Record
SSM/I	Special Sensor Microwave Imager
SST	Sea Surface Temperature
STAR	Center for Satellite Applications and Research
UM	User’s Manual
VIPP	Vertical Integration & Post-Processing

Appendix B. Bibliography

Boukabara, S, K. Garrett, C. Grassotti, F. Iturbide-Sanchez, W. Chen, Z. Jiang, S. A. Clough, X. Zhan, P. Liang, Q. Liu, T. Islam, V. Zubko, and A. Mims. A physical approach for a simultaneous retrieval of sounding , surface, hydrometeor, and cryospheric parameters from SNPP/ATMS. *J Geophys. Res. Atmos.*, *118*, DOI: 10.1002/2013JD020448.2158438, 2013.

Boukabara, S, K. Garrett, W. Chen, F. Iturbide-Sanchez, C. Grassotti, C. Kongoli, R. Chen, Q. Liu, B. Yan, F. Weng, R. Ferraro, T. Kleespies, and H. Meng. MiRS: An all-weather satellite data assimilation and retrieval system. *IEEE Trans. Geosc. Rem. Sens.*, DOI: 10.1109/TGRS.2011.2158438, 2011.

Iturbide-Sanchez, F., S. Boukabara, R. Chen, K. Garrett, C. Grassotti, W. Chen, and F. Weng. Assessment of a variational inversion system for rainfall rate over land and water surfaces. *IEEE Trans. Geosc. Rem. Sens.*, DOI: 10.1109/TGRS.2011.2119375, 2011.

Kongoli, C., S. Boukabara, B. Yan, F. Weng, R. Ferraro. A new sea-ice concentration algorithm based on microwave suface emissivities – Application to AMSU measurements. *IEEE Trans. Geosc. Rem. Sens.*, DOI: 10.1109/TGRS.2011.2052812, 2011.

T. Mo, *"Calibration of the Advanced Microwave Sounding Unit-A radiometers for NOAA-N and NOAA-N''"*, NOAA technical report; NESDIS 106, 2002.

S. Boukabara, F. Weng and Q. Liu. Passive Microwave Remote Sensing of Extreme Weather Events Using NOAA-18 AMSU and MHS, submitted, *IEEE Trans. Geosc. Rem. Sens.*, DOI: 10.1109/TGRS.2011.2052812, 2007.

S-A. Boukabara , F. Weng , R. Ferraro , L. Zhao , Q. Liu , B. Yan , A. Li , W. Chen , N. Sun , H. Meng , T. Kleespies , C. Kongoli , Y. Han , P. Van Delst , J. Zhao and C. Dean . *"Introducing NOAA's Microwave Integrated Retrieval System (MIRS)"*. Presentation at the ITSC-15 conference in Maratea, Italy, October 2006.

Sid-Ahmed Boukabara, Fuzhong Weng and Quanhua Liu, *"Microwave Variational Retrieval of Surface and Atmospheric Parameters: Application to AMSU/MHS and SSMI/S sensors."* Poster at the American Meterological Society Annual meeting., Atlanta, Georgia 2006.

S. Boukabara and F. Weng. *"Microwave Emissivity Retrieval Over Ocean in All-Weather Conditions. Validation Using satellite sensors and airborne GPS-Dropsondes"*. Presentation at the 1st Workshop on Modeling and Remote Sensing of Surface Properties, Paris, France, June 2006.

S. Boukabara and F. Weng . *"Passive Microwave Remote Sensing of Extreme Weather Events*

Using NOAA-18 AMSU and MHS". Presentation at the 9th Specialist meeting on Microwave Radiometry and Applications, San Juan, Puerto-Rico, February 2006.

S. Boukabara and F. Weng. "SSMIS Ocean EDR Algorithms". Presentation at the NRL/JCSDA mini-workshop on preparation for SSMIS radiance assimilation, Monterey, CA, October 2005

Q. Liu and F. Weng, *"One-Dimensional Variational Retrieval Algorithm of Temperature, Water Vapor and Cloud Water Profiles From Advanced Microwave Sounding Unit (AMSU)"*, IEEE Trans. on Geosc. and Remote Sensing, vol. 43, No 5, 2005

Moncet et al, 2001, CMIS ATBD.

Appendix C. List of Main Programs and Their Functions

Advanced retrieval code (1dvar) in /1dvar

- Location : `/mirs/src/1dvar`
- Language: Fortran-95

Name	1dvar	
Main Function	Retrieval of a set of geophysical parameters (or EDRs)	
Inputs	Foot Print Matched Radiances, RTM Uncertainty Matrix, Covariance Matrix, Channel Information, Tuning Parameters, Bias correction file,	
Outputs	Geophysical parameters (or EDRs)	Simulated radiances, Temperature profile, Moisture profile, Hydrometeor profile, Cloud water profile, Surface temperature, Surface emissivity Convergence metric

Forward operator code (fwd) in /fwd

- Location : `/mirs/src/fwd`
- Language: Fortran-95

Name	Fwd
Main Function	forward simulation for all sensors
Inputs	- Atmosphere information from the Scene file (Pressure, Temperature, Water vapor, CLW, Rain, Snow, Graupel, Ice) - Surface Information from the Scene file (Surface temperature, Surface Type, Emissivity, Reflectivity) - Geometry Information - Channel Information
Outputs	Radiances (or Brightness temperature)

Main programs in /testbed

- Location : `/mirs/src/testbed`

Name	Main Function	Inputs	Outputs	Language
biasGenerAndMonit	Calibration of microwave radiances by comparing them to simulations based on analyses (GDAS)	simulations based on analyses (GDAS), Measurements	bias residuals and geophysical biases	IDL
fm	footprint Matching (FM) AMSUA/MHS	SDR	FMSDR	Fortran 95
nwp	collocates the NWP gridded data into an orbit-based space using radiance measurements	Radiance, measurements, Gridded analyses	Collocated data of NWP at the measurement location	Fortran 95
regressAlgors	generate Regress Algorithms from AMSUA/MHS radiances	analyses/measurements collocation	Regression	IDL
retrRegress	apply regression-based algorithms on FM-SDR radiances	FM-SDR radiances	external data to be used with MIRS	Fortran 95
rdr2tdr	RDR to TDR conversion (AMSUA/MHS)	RDR	TDR	Fortran 95
tdr2sdr	TDR to SDR conversion (AMSUA/MHS)	TDR	SDR	Fortran 95
chopp	chopping the ECFMSDR file into pieces for faster processing	EC-FM-SDR	Chopped EC-FM-SDR	Fortran 95
mergeEDR	merge EDRs	granulized EDRs	a full orbital file	Fortran 95
mergeNEDTofDiffInstr	merge NEDT	NEDT from the number of sensors	Combined NEDT file	Fortran 95
psFigsGener	quick look at scenes content(s) in the satellite coordinates.	Scenes data, EDRs	Figures (maps)	IDL
pngMapsGener	figures generation (maps) in PNG format	EDRs	Figures (maps) in PNG format	IDL
nedtMonitoring	monitor the NEDT used by MIRS, day after day.	NEDT	Monitoring plots	IDL
Fwd	application of forward operator on GDAS analyses	GDAS analyses	forward simulation for sensors	Fortran 95
biasGener	determination of the bias (by comparing FWD simul. and meas.)	FWD simulation and measurements	Bias	IDL

Main programs in /lib_idl

- Location : /mirs/src/testbed
- Language: IDL

Name	Main Function
algors	Summary of all subroutines related to regression algorithms
io_covBkg	Summary of all subroutines related to I/O processes for the different covariance/background files
io_monitor	Summary of all subroutines related to I/O processes for the monitoring file (iteration by iteration)
misc	Summary of miscellaneous subroutines needed across the applications
Export_IMG	export grided data into png files using z-buffer
io_dropsondes	Summary of all subroutines related to I/O processes for the different dropsondes files
io_regressAlgors	Summary of all subroutines related to I/O processes for the different types algorithm coefficients files, etc
stats_sub	Summary of subroutines related to computing statistical data and performing EOF decomposition
io_Mapping	Summary of all subroutines related to mapping subroutines
io_scene	Summary of all subroutines related to I/O processes for the different scene files (geophysical data)
utilities	Summary of subroutines related to performing utility functions (colorbar, etc)
io_coloc	Summary of all subroutines related to I/O processes for the colocation files (geophysical and/or radiometric data)
io_measur	Summary of all subroutines related to I/O processes for the different radiometric files
io_misc	Summary of all subroutines related to I/O processes for the different miscellaneous files
meteorFcts_sub	Summary of all subroutines related to meteorological functions and how to integrate the vertical distributions of water vapor and other hydrometeors.

Appendix D. List of Library Modules

Location : /mirs/src/lib
- Language: Fortran-95

Directory	Name	Main Function
FwdOperProcess	FwdOper	all subroutines needed for the interface with the forward operator
io	IO_AuxData	reading auxillary data accompanying measurements data
	IO_InstrConfig	reading instrument configuration information
	IO_Monitor	handling monitoring data (iteration/iteration)
	IO_Scene	I/O of the scene data
	IO_Colocate	I/O of the collocated datasets
	IO_MeasurData	reading measurements data (radiances, brightness temperatures)
	IO_Noise	handling of the noise I/O (except for the loading of Noise structures, still under the Noise module
	IO_ExternData	I/O of the External data
	IO_Misc	I/O of several different files
	IO_Regress	regression algorithms
misc	CntrlParams	controlling paramaters used in the 1dVAR/retrieval. Extended to also handle controlling parameters of the fwd operator
	Consts	many constants needed across applications
	misc	several routines of different functions and several applications
	TunParams	tuning paramaters used in the 1dVAR/retrieval
noise	Noise	handling of the noise
qc	ErrorHandling	error handling from fortran, especially op_msg which is a mimic function to the OSDPD function by the same name. Its use is in lieu of linking to the OSDPD-specific subroutine
	QCchecking	quality control and other related processing
InversProcess	GeophCovBkg	loading the geophysical covariance matrix/Bkg/Transf Matrx. This cov matrx is dimensioned np x np x nTyp. It is assumed that we could have several types of covariances (depending on the sfc/atm classes, etc)
	SeFeErrCov	loading the measurements error covariance matrix. This matrix is dimensioned nChanSe x nChanSe x nSeTyp. It is assumed that we could have several types of measurement error covariances. Extended to include other subroutines related to building the covariance matrix of the instrumental/forward errors
	VarInversion	performing the different operations related to the variational retrieval
math	mathFcts	mathematical fcts/procedures that are not standard in Fortran
PrePostProcess	PreClassif	pre-classification process. Note that this module could be sensor-specific, as every sensor would have its own pre-classification algorithm.
utilities	utilities	various utility subroutines