VAISALA

USER'S MANUAL

IRIS™ and RDA Dual Polarization

M211452EN-C

PUBLISHED BY

Vaisala Oyj
P.O. Box 26
FI-00421 Helsinki
Finland

Phone (int.): +358 9 8949 1
Fax: +358 9 8949 2227

Visit our Internet pages at www.vaisala.com

Table of Contents

CHAPTER 1
GENERAL INFORMATION 5
1.1 About This Manual 5
1.1.1 Contents of This Manual 5
1.2 Version Information 6
1.3 Related Manuals 6
1.1 Documentation Conventions 6
1.2 Trademarks 7
1.3 License Agreement 7
1.4 Warranty 7

CHAPTER 2
INTRODUCTION TO DUAL POLARIZATION 9
2.1 Overview of Dual Polarization 9
2.2 Z_H, Z_V, and Z_{HV}: Horizontal, Vertical, and Enhanced Reflectivities 10
2.3 Radar System Considerations 12
2.3.1 Transmit Modes 12
2.3.2 Receive Modes 12
2.3.3 Summary of Radar System Characteristics 13
2.4 Overview of Processing Algorithms 14
2.4.1 Input Receiver Sample Notation 15
2.5 Case 1: Fixed Transmit: Dual-Channel Receiver 18
2.5.1 Input Receiver Samples 18
2.5.2 Calculation of the Polarization Measurands 19
2.6 Case 2: Simultaneous Dual Transmit and Receive (STAR) 20
2.6.1 Input Receiver Samples 20
2.6.2 Calculation of the Polarization Measurands 20
2.7 Case 3: Alternating H/V Transmit: Single Receiver 21
2.7.1 Input Receiver Samples 21
2.7.2 Calculation of the Polarization Measurands 22
2.8 Case 4: Alternating H/V Transmit: Dual Receiver 23
2.8.1 Input Receiver Samples 23
2.8.2 Calculation of the Polarization Measurands 23
2.9 Standard Moment Calculations (T, Z, V, W) 24
2.9.1 Overview 24
2.9.2 Model for Standard Moment Autocorrelations 26
2.9.3 Calibration Parameters 26
2.10 Calibration Considerations 35

CHAPTER 3
ATTENUATION CORRECTION OF Z 39
3.1 Overview 39
3.2 PHIDP Conditioning 41
3.2.1 PHIDP Unfolding and Conditioning 41
3.3 Dual Polarimetric Attenuation Correction 43
3.3.1 Weather Classification Algorithm 45
3.3.2 Iterative PhiDP Correction 46
3.4 Adaptive K_{dp} Moment Estimation 48
3.5 User Instructions 49
3.5.1 Install, Licensing, and Activation in Setup 49
3.5.2 Dual Pol Configuration File 50

CHAPTER 4
HYDROCLASS 57
4.1 Overview 57
4.2 Description of Algorithms 58
4.2.1 Methodology 58
4.2.2 Fuzzy Logic 59
4.2.3 HydroClass: A Synthesis of Public Methods 60
4.2.4 Classification data 61
4.3 Classifiers 63
4.3.1 Preclassifier 63
4.3.2 MeteoClassifiers 66
4.3.3 PrecipClassifier 77
4.3.4 CellClassifier 77
4.4 User Instructions 78
4.4.1 Installing, Licensing, and Activating in Setup 78
4.4.2 Running HydroClass 78
4.4.3 Configuration 80
4.4.4 Miscellaneous 102
4.5 APPENDICES 105
4.5.1 Default Configuration for S-band Radars 105
4.5.2 Parametrized Fuzzy Parameters 123

CHAPTER 5
WIDE DYNAMIC RANGE 127
5.1 Overview 127
5.2 Introduction 127
5.3 Dynamic Range in General 127
5.4 Factors of Current RVP Dynamic Range 130
5.5 Dual-Polarization RVP Dynamic Range Expansion 131
5.5.1 IFD ADC Terminal Definitions 132
5.6 Architecture 133
5.7 Algorithms 134
5.7.1 ADC DC/Zero Drift Detection 134
5.7.2 Relative Gain Detection 135
5.7.3 Relative Delay Detection 136
5.7.4 Saturation and Overlap Detection 141
5.7.5 Fractional Delay FIR Optimization 144

5.7.6 Error Tolerance and FDF Update 151
5.8 Use of DSPX Utility or Manual User Configuration and Status Monitoring . . . 153
5.8.1 Dspx -Mb 153
5.8.2 Dspx -Mc 153
5.8.3 Dspx -Mp 154
5.8.4 Dspx -Pr 156
5.9 STATUS MONITORING 157
5.10 TESTING 158
5.10.1 Dspx Configuration 159
5.10.2 Dspx Plot 160
5.10.3 RVP9 - ShowRxMode Status Monitoring 161
5.10.4 Zauto: Noise Level and Dynamic Range 164
5.10.5 Ascope: Noise Level and Dynamic Range 167
5.10.6 Kumpula Radar Operational Test (HEL Lab) 172
5.11 SUMMARY 173

CHAPTER 6
CALIBRATION 175
6.1 Calibration Considerations 175
6.1.1 dBZo Calibration for dBZ 175
6.1.2 GDR Calibration for Z_{dr} 176
6.1.3 Offset Calibration for LDR 177
6.2 Z_{dr} 178
6.2.1 Invoking Zcal 178
6.2.2 Zcal Commands and Prompts 179
6.2.3 Changing LOG Receiver Calibration Numbers 180

CHAPTER 7
QUALITY CONTROL 183
7.1 Thresholding of Polarization Parameters 183

CHAPTER 8
REFERENCES 185

APPENDIX A
GLOSSARY 189

CHAPTER 1

GENERAL INFORMATION

1.1 About This Manual

This manual provides information on the IRIS and RDA dual polarization.

1.1.1 Contents of This Manual

This manual consists of the following chapters:

- Chapter 1, General Information: This chapter provides general notes for the manual and the product.
- Chapter 2, Introduction to Dual Polarization: This chapter describes the different hardware implementations to achieve dual polarization and the availability of different data types with each type of implementation.
- Chapter 3, Attenuation Correction of Z: This chapter describes the methodology to make correction to reflectivity and differential reflectivity caused by caused by precipitation.
- Chapter 4, HydroClass: This chapter describes the algorithm for estimating the bulk particle types within the sample volume.
- Chapter 5, Wide Dynamic Range: This chapter describes Wide Dynamic Range (WDR).
- Chapter 6, Calibration: This chapter describes calibration techniques in a dual polarization radar for reflectivity, differential reflectivity, and linear depolarization ratio.
- Chapter 7, Quality Control: This chapter describes the metrics used to determine data quality and how they are applied to the dual polarimetric data.
- Chapter 8, References: This chapter provides material references.
- Appendix A, Glossary: This appendix contains the glossary.

1.2 Version Information

Manual Code	Description
M211452EN-C	This manual. Third version. November 2013
M211452EN-B	Previous manual. Second version. March 2013
M211452EN-A	Previous manual. First version. August 2012

1.3 Related Manuals

1.Manual Code	1.Manual Name
M211315EN	Software Installation Manual
M211316EN	IRIS and RDA Utilities Manual
M211317EN	IRIS Radar Manual
M211318EN	IRIS Programmer's Manual
M211319EN	IRIS Product and Display Manual
M211320EN	RCP8 User's Manual
M211321EN	RVP8 User's Manual
M211322EN	RVP900 User's Manual

You can download the latest versions of the manuals from Vaisala product website, http://www.vaisala.com. They can be read online using by Adobe® Reader®, which is installed with IRIS.

Vaisala encourages you to send your comments and/or corrections to:

Vaisala Inc.
7A Lyberty Way
Westford, MA 01886
email: helpdesk@vaisala.com

1.1 Documentation Conventions

Throughout the manual, important safety considerations are highlighted as follows:

WARNING	Warning alerts you to a serious hazard. If you do not read and follow instructions very carefully at this point, there is a risk of injury or even death.

CAUTION	Caution warns you of a potential hazard. If you do not read and follow instructions carefully at this point, the product could be damaged or important data could be lost.

NOTE	Note highlights important information on using the product.

`prompt`—Some features of the RVP900 operate by displaying questions and waiting for you to type an answer. The text of prompts is displayed in bold, monospaced type.

1.2 Trademarks

Vaisala and the Vaisala logo are registered trademarks of Vaisala Oyj in the United States and/or other countries.

All other company, product names, and brands used herein may be the trademarks or registered trademarks of their respective companies.

1.3 License Agreement

All rights to any software are held by Vaisala or third parties. The customer is allowed to use the software only to the extent that is provided by the applicable supply contract or Software License Agreement.

1.4 Warranty

Visit our Internet pages for more information and for our standard warranty terms and conditions: www.vaisala.com/warranty.

Any such warranty may not be valid in case of damage due to normal wear and tear, exceptional operating conditions, negligent handling or installation, or unauthorized modifications. See the applicable supply contract or Conditions of Sale for details of the warranty for each product.

CHAPTER 2

INTRODUCTION TO DUAL POLARIZATION

2.1 Overview of Dual Polarization

Conventional weather radars transmit and receive only a single polarization (usually horizontal). State-of-the-art dual polarization radars transmit and receive both horizontal and vertical polarization. Vaisala Hydrometeor Classification (HydroClass™) software makes optimal use of these dual-channel measurements to deduce the types of scatterers present in the atmosphere, such as rain, hail, snow, graupel, and even non-meteorological targets such as insects, chaff, and sea clutter. In addition to the improvements in precipitation estimation that are achieved with dual-polarization radar, the ability to deduce and map the types of scatterers greatly enhances the power of dual-polarization radar for applications such as:

- Hail detection
- Lightning hazard potential forecasting
- Detection of convection and stratiform rain
- Highway snow removal
- Airport terminal operation
- Rain/snow line demarcation
- Melting height detection
- Weather modification for hail mitigation
- Insurance industry claims verification
- Military detection of chaff
- Data quality improvement by elimination of non-meteorological targets
- Improved precipitation forecasting
- Hydrological modeling

2.2 Z_H, Z_V, and Z_{HV}: Horizontal, Vertical, and Enhanced Reflectivities

In the case of amplitude (power) measurements, the power returned to the radar can be related either to the particles diameter in the horizontal plane or vertical plane. When expressing this power through the radar equation, these become horizontal reflectivity (Z_H), which is the traditional single polarization reflectivity, and vertical reflectivity (Z_V).

Z_H and Z_V are typically calculated using an autocorrelation power estimator, or by performing a cross-correlation of the signal with itself. Autocorrelation is a technique to find periodic signals buried within noise. A special case is to perform the cross-correlation of the horizontal power to the vertical power. This becomes enhanced reflectivity (Z_{HV}) and can be shown to have greater ability to weak signals in the presence of noise, compared to Z_H or Z_V alone.

All power estimators are sensitive to intervening attenuation from precipitation, requiring a data correction for quantitative use with C- or X-band radars.

- **Differential Reflectivity (Z_{dr})**

  In the case of amplitude (power) measurements, the larger horizontal axis of drops causes the power measured at horizontal polarization (of the electric field) to be larger than the power measured at vertical polarization. The ratio of the reflectivity factors Z_H/Z_V, expressed in dB, is called differential reflectivity (Z_{dr}).

  Zdr is generally positive in rain (that is, >1) and is usually less than 5 dB. When the rainfall rate is large, there are typically more large drops, so Z_{dr} is larger. Low Z_{dr} and high dBZ indicates the presence of hail, which may be tumbling with no preferred orientation. Z_{dr}, because it is a ratio of powers, is not sensitive to the radar calibration, as long as the overall gain of the H and V channels is the same (or calibrated). For information about calibrating the overall gains of the H and V channels, see Chapter 5, Wide Dynamic Range, on page 127.

- **Differential Phase (PhiDP) and Specific Differential Phase (K_{dp})**

  In the case of phase measurement, the speed of propagation is also affected by the asymmetry of the larger drops. Because of the longer dimension of the horizontal axis of drops, the medium is effectively more dense for horizontal than for vertical polarization. This causes the horizontal wavelength to be slightly compressed (more phase cycles per unit distance) in comparison with the vertical wavelength, which leads to a phase difference between horizontal and vertical. This difference is called the differential phase shift, PhiDP.

PhiDP increases with range, since the phase shifts faster (more frequency cycles per unit distance) for the compressed horizontal microwaves compared to the faster vertical microwaves. The range derivative of PhiDP, that is, the change of phase per unit distance, is called specific differential phase (K_{dp}). K_{dp} is almost directly proportional to the rainfall rate, so that it has the potential for improving precipitation rate measurements, as compared to traditional Z–R relationship measurements, which can be highly inaccurate. For information on techniques for calculating KDP, see Chapter 3, Attenuation Correction of Z, on page 39.

- **Linear Depolarization Ratio (LDR)**

  Some advanced polarization radars can transmit at one polarization and receive simultaneously in two channels, usually the co–polarized and cross–polarized components. For example, when transmitting horizontal, both horizontal (co-polarized) and vertical (cross-polarized) polarized energy are received by two separate channels. In the case of vertical or horizontal, the ratio of the power Zcross or "Zco" is called the linear depolarization ratio (LDR). The amount of incident radiation that is depolarized by a particle, depends on the particle shape and orientation (for example, canting angle with respect to horizontal). Perfectly spherical particles do not depolarize either horizontal or vertical polarization, so the LDR is zero. Particles that are wet, tumbling and irregularly shaped will give larger LDR values. Therefore, LDR values in rain tend to be small, for example, less than -25dB. Larger values of LDR can occur in the bright band or in the presence of hail.

  A radar and antenna system must be optimized to measure LDR by assuring that the antenna, feed and supporting struts, and radome are not themselves depolarizing the transmitted and received radiation. This is called "cross-pol isolation". The integrated cross-pol isolation of the antenna pattern must be more than 35 dB for LDR measurement since -20 dB is a large LDR. If the Radar is not able to achieve this level of cross-pol isolation will be limited to how accurate the rainfall estimates can be from the dual polarization data.

- **[RhoHV, PhiDP] [RhoH, PhiH] [RhoV, PhiV]: Correlation Variables**

  There are several correlation functions that can be calculated, depending on the capabilities of the radar. These are generally complex, having both an amplitude and phase. These are all normalized so that a perfect correlation magnitude is 1 and perfectly decorrelated is 0.

  RhoH and PhiDP are the magnitude and phase of the correlation between the horizontal and vertical co-polarized channels. These are available on H/V switching systems or on systems that transmit

simultaneous H and V. As discussed in Differential Phase (PhiDP) and Specific Differential Phase (K_{dp}) on page 10, PhiDP can be used to infer precipitation rate. RhoHV, in rain, is typically very close to 1 (0.98). RhoHV values can be reduced in the case of irregularly shaped, randomly oriented, wet tumbling particles; thus, RhoHV provides information on the particle type.

RhoH and PhiH are the magnitude and phase of the correlation between the co-polarized and cross-polar channels for H transmission, and simultaneous H and V reception. RhoV and PhiV denote the cross–channel correlation magnitude and phase for vertical transmission. These are available on a dual-channel receiver with transmit, either fixed or alternating. The information content of the cross-pol correlations is the topic of current research.

2.3 Radar System Considerations

A polarization radar is characterized by how it transmits and receives. For simplicity, we will assume that the radar uses horizontal and/or vertical polarization. However, other polarization pairs could be used (for example, right and left circular polarization).

2.3.1 Transmit Modes

- **Fixed** (Horizontal or Vertical)—Controlled by a switch or the radar can be fixed to transmit a single polarization. If a switch is used, it can be a simple, slow, waveguide switch rather than a fast switch (pulse-to-pulse).
- **Alternating** (Horizontal and Vertical)—Radar alternates pulse-to-pulse between horizontal and vertical. A high-power, fast switch is used to switch the polarization between the two channels.
- **Simultaneous** (Horizontal and Vertical)—Horizontal and vertical are transmitted simultaneously.

2.3.2 Receive Modes

- **Single-Channel Receiver**–Used only for alternating transmission. The receiver typically receives the co–polarized radiation (transmit H and receive H, then transmit V and receive V).
- **Dual-Channel Receiver**–Receives two channels (H and V) simultaneously.

Table 1. Transmitter Types on page 13 summarizes the various transmit and receive cases, and the polarization variables that are available for each.

NOTE Standard parameters are available for all cases (dBT, dBZ, V, and W). The RVP900 supports all of these cases.

Table 1 Transmitter Types

	Transmitter Type			
Receiver Type	**Fixed H**	**Fixed V**	**Alternating H&V**	**Simultaneous H+V**
Single-Channel	Conventional Radar	Conventional Radar	Z_V Z_{HV} Z_{dr} RhoHV PhiDP and K_{dp}	Not applicable
Dual-Channel	LDRH RhoH PhiH	LDRV RhoV PhiV	Z_V Z_{HV} LDRH LDRV RhoH RhoV PhiH PhiV Z_{dr} RhoHV PhiDP and K_{dp}	Z_V Z_{HV} Z_{dr} RhoHV PhiDP and K_{dp} *(STAR mode)*

The fixed, single-channel cases are conventional radars rather than polarization radars. The case of simultaneous H+V transmission and a single-channel receiver does not make physical sense. The other cases provide various polarization measurements. The fixed, dual-channel cases allow the optional Dual Polarization cross-polarization LDR and the co-pol/cross-pol correlation amplitude and phase to be measured (for example, RhoH and PhiH). The simultaneous H+V transmission and dual-channel reception is sometimes called the STAR mode (simultaneous transmit and receive). This allows the co–pol measurements to be made (Z_{dr}, RhoHV, PhiDP, and K_{dp}). The alternating transmission, dual-channel receiver allows both the co–pol and the cross-pol measurements to be made; it is the most complete.

2.3.3 Summary of Radar System Characteristics

The RVP900 supports all of these transmit and receive modes; most polarization radar systems do not. The measurement of cross-pol parameters such as LDR (fixed or alternating transmission and dual-channel reception) requires a radar system that has been optimized for cross–pol isolation, for example, an offset feed antenna and no radome. By removing the feed, support struts, and radome from the path of the radiation, the cross–pol isolation can be improved.

The single-channel, alternating method has been used in several polarization radars for Z_{dr} measurement. The advantage of this approach, is that it is relatively easy to modify a conventional radar by adding a dual-port feed and a high-power, fast switch above the antenna rotary joints. The disadvantage is that the switch is costly and will eventually fail.

For these reasons, the STAR mode has come into recent use. No switch is required and the components are fairly reliable. The disadvantage of the approach (as it is usually implemented), is that a dual-rotary joint and dual waveguides are required, to duct both the H and the V through the antenna pedestal up to the antenna feed. In spite of this, the STAR mode offers the best approach for upgrading an existing radar, or for factory installation on a new radar of conventional design.

2.4 Overview of Processing Algorithms

The RVP900 supports four polarization modes summarized in the table below. For each case, the standard moments (T, Z, V and W) are calculated as well. The notation for the outputs used here is similar to that in standard usage (for example, Doviak and Zrnic). However, for LDR we use the notation LDRH to indicate that this is the LDR for horizontal transmission. The notation RhoH and PhiH is used to indicate the magnitude and phase of the covariance between the co– and cross–polarized channels for H transmit.

Table 2 Supported Polarization Modes and Outputs

Case	Transmit	Receive	Processing Mode	Polarization Outputs
1	Fixed Horizontal or Fixed Vertical	Dual-Channel	PPP only	LDRH RhoH PhiH or LDRV RhoV PhiV
2	Simultaneous H+V (STAR Mode)	Dual-Channel	PPP or Z_{dr} for FFT, Random Phase and DPRT1&2	Z_V Z_{HV} Z_{dr} PhiDP K_{dp} RhoHV
3	Alternating H/V	Single-Channel	PPP only	Z_V Z_{HV} Z_{dr} PhiDP K_{dp} RhoHV
4	Alternating H/V	Dual-Channel	PPP only	Z_V Z_{HV} Z_{dr} LDRH RhoH PhiH LDRV RhoV PhiV PhiDP K_{dp} RhoHV

2.4.1 Input Receiver Sample Notation

For the discussion of polarization, we will adopt the notation used by Doviak and Zrnic. The received signal for pulse n from a single range bin shall be denoted as:

s^n_{hh}	Receive h: Transmit h	Horizontal co-polar signal
s^n_{vh}	Receive v: Transmit h	Horizontal cross–polar signal
s^n_{vv}	Receive v: Transmit v	Vertical co–polar signal
s^n_{hv}	Receive h: Transmit v	Vertical cross-polar signal

The pulse index is now indicated by the superscript as opposed to the subscript. The first subscript indicates the received polarization, while the second subscript indicates the transmit polarization. If the transmit is the same as the received polarization, this is the co-polarized signal. If the transmit and receive are different, this is the cross-polarized signal.

These variables are complex and are the same as the "s_n" notation used in Chapter 5 of the *RVP900 User's Manual - Digital Receiver and Signal Processor*. For example, we can write:

$$S^n_{hh} = I^n_{hh} + jQ^n_{hh}$$

to show the relationship to the received *I* and *Q* values. Either filtered and unfiltered versions of the samples can be selected for processing.We drop the s' notation for filtered samples.

2.4.1.1 Notation and Model for Correlations

The pulse pair processing mode is used for all of the polarization calculations, except that Z_{dr}-only processing for the STAR case can be done in either FFT or random phase as well as pulse pair. As with the standard moments, the autocorrelations form the basis for the processing of the polarization variables.

An autocorrelation is the cross-correlation of a signal with itself. Informally, it is the similarity between observations as a function of the time separation between them. It is a mathematical tool for finding repeating patterns, such as the presence of a periodic signal, which has been buried under noise. However, it is also possible to perform a correlation of the S_{hh} and S_{vv} signal, which is used to create Z_{HV}.

The autocorrelations are computed similar to the standard moments, for example, in pulse pair mode the autocorrelations for the horizontal transmit co-polar channel are:

$$T_{ohh} = \frac{1}{M}\sum_{n=1}^{M} S_{hh}^{n} * S_{hh}^{n}$$

$$R_{ohh} = \frac{1}{M}\sum_{n=1}^{M} s'^{n}_{hh} * s'^{n}_{hh}$$

$$R_{1hh} = \frac{1}{M-1}\sum_{n=1}^{M-1} s'^{n}_{hh} * s'^{n+1}_{hh}$$

$$R_{2hh} = \frac{1}{M-2}\sum_{n=1}^{M-2} s'^{n}_{hh} * s'^{n+2}_{hh}$$

What is different is that for dual polarization systems, these correlations can be applied up to four different ways (R_{hh}, R_{vv}, R_{hv}, and R_{vh}), where R_{hv} and R_{vh} are equivalent. The physical model for the channel powers is identical to the model used for the standard moment cases:

Co-Channel Power

$$R_0^{hh} = \frac{1}{M}\sum_{n=1}^{M-1} s'^{n}_{hh} * s'^{n}_{hh} = g_h^r g_h^t S'_{hh} + N_h$$

$$R_0^{vv} = \frac{1}{M}\sum_{n=1}^{M-1} s'^{n}_{vv} * s'^{n}_{vv} = g_v^r g_v^t S'_{vv} + N_v$$

Diagonal Channel Power

$$R_0^{hv} = \frac{1}{M}\sum_{n=1}^{M-1} s'^{n}_{hh} * s'^{n}_{vv} = \frac{g_h^r g_h^t S'_{hh} + g_v^r g_v^t S'_{vv}}{2}$$

Here *S* denotes the actual backscatter average power to the radar. When multiplied by the appropriate transmitter and receiver gains, *S* yields the actual measured power. Sometimes in comparing powers in two channels (for example, Z_{dr} and LDR), we need to know the relative gains of the two channels. However, in many calculations, the relative gains cancel out, and in these cases the algorithms are implemented assuming all the gains are equal to 1.

In the R_{hv} term, the noise variable is not present. This is because the noise between the horizontal receiver and vertical receiver is random, having a normalized coherency of 0 with an infinite number of samples. A finite number of samples needs to be used, typically between 30 and 60, in weather radar signal processing. However, due to the noise coherency going to 0, the noise variance also becomes smaller, allowing us to lower the detection thresholds, while having same false alarm rates as the traditional R_{hh} term. Lowering the detection threshold increases the apparent sensitivity when inserting the R_{hv} term into the radar equation for reflectivity.

In the algorithm descriptions in the following sections, we use the notation common in the literature, for example:

$$R_0^{hh} = \frac{1}{M}\sum_{n=1}^{M-1} s'^{n}_{hh} * s'^{n}_{hh} = \left\langle |s'_{hh}|^2 \right\rangle = S_{hh}$$

$$R_0^{vv} = \frac{1}{M}\sum_{n=1}^{M-1} s'^{n}_{vv} * s'^{n}_{vv} = \left\langle |s'_{vv}|^2 \right\rangle = S_{vv}$$

$$R_0^{hv} = \frac{1}{M}\sum_{n=1}^{M-1} s'^{n}_{hh} * s'^{n}_{vv} = \left\langle |s'_{hv}|^2 \right\rangle = S_{hv}$$

2.4.1.1.1 Noise Bias in Channel Powers and Correlations and Optional Correction

The average noise powers N_v and N_h are assumed to be receiver noise only. These bias the autocorrelations at lag zero, that is, the channel power measurements, and must be subtracted. Autocorrelations at lags 1 and 2 are not biased by noise, while cross-channel correlations are biased by finite noise. Cross-channel phases are smeared by finite noise while unbiased, assuming that the noises in the two channels are independent (a good assumption).

The channel noise values are measured directly by the RVP900 during noise sampling. The use of these measurements for correcting the effects of finite noise is configured in the TTY setups. The choices are made with the mp non-volatile setup settings "Polarimetric Power Params – NoiseCorrected:YES/NO" and "Polarimetric Correlations – NoiseCorrected:YES/NO"

If "Polarimetric Power Params – NoiseCorrected" is enabled, the Z_{dr} and LDR are computed from channel powers, subtracted for noise levels. The sensitivity is enhanced to observe weather effects in weak echoes (SNR << 10 dB). At the SNR low limit, Z_{dr} approaches 0 dB as soon as noise levels are properly set. With no noise correction, Z_{dr} values in weak signal regions are biased, and approach the ratio of channel noise levels. A finite noise level may set a limit for LDR observations.

If "Polarimetric Power Correlations – NoiseCorrected" is left disabled, RhoHV, RhoH, and RhoV approach 0 dB in weak echoes, while setting "YES" makes them approach unity. Phase measurements: PhiDP, K_{dp}, PhiH, and PhiV are not affected by these settings.

2.4.1.1.2 Clutter Filtering

The polarization variables are computed from data filtered for clutter. The settings are the same, and are user configurable for standard moments. The filter is applied identically for each polarimetric channel.

2.5 Case 1: Fixed Transmit: Dual-Channel Receiver

2.5.1 Input Receiver Samples

In fixed mode, the radar is configured (either permanently or by means of a switch) to transmit either vertical or horizontal polarization with dual-channel reception of both the co- and cross-channel polarizations. For example, the radar can transmit horizontal and receive both horizontal (co) and vertical (cross) polarizations.

The received samples in the two transmit cases are as follows:

Transmit Horizontal

$$\left[S_{hh}^{1}:S_{vh}^{1}\right]\left[S_{hh}^{2}:S_{vh}^{2}\right]\left[S_{hh}^{3}:S_{vh}^{3}\right]\ldots\left[S_{hh}^{M}:S_{vh}^{M}\right]$$

or

Transmit Vertical

$$\left[S_{vv}^{1}:S_{hv}^{1}\right]\left[S_{vv}^{2}:S_{hv}^{2}\right]\left[S_{vv}^{3}:S_{hv}^{3}\right]\ldots\left[S_{vv}^{M}:S_{hv}^{M}\right]$$

2.5.2 Calculation of the Polarization Measurands

The processing in this mode is done by pulse pair algorithm. The user may select a clutter filter.

The polarization measurands for the two transmit cases are as follows:

Transmit Horizontal	**or**	**Transmit Vertical**
$LDRH = 10LOG\left[\frac{S_{vh}}{S_{hh}}\right]$	or	$LDRV = 10LOG\left[\frac{S_{hv}}{S_{vv}}\right]$
$= 10LOG\left[\frac{<\lvert S_{vh}\rvert^2>-N_v}{<\lvert S_{hh}\rvert^2>-N_h}\right]-XDR$	or	$= 10LOG\left[\frac{<\lvert S_{hv}\rvert^2>-N_h}{<\lvert S_{vv}\rvert^2>-N_v}\right]+XDR$
$RHOH = \lvert\rho_h\rvert$	or	$RHOV = \lvert\rho_v\rvert$
$PHIH = \arg\lvert\rho_h\rvert$	or	$PHIV = \arg\lvert\rho_v\rvert$

The H and V average channel powers are computed with optional noise correction as follows:

Co- $g_h^r g_h^t S_{hh} = <\lvert S_{hh}\rvert^2> - N_h$ or $g_v^r g_v^t S_{vv} = <\lvert S_{vv}\rvert^2> - N_v$

Cross- $g_v^r g_h^t S_{vh} = <\lvert S_{vh}\rvert^2> - N_h$ or $g_h^r g_v^t S_{hv} = <\lvert S_{hv}\rvert^2> - N_h$

The complex covariance ρ (used above) is as follows:

for H transmit $\rho_h = \frac{<S_{vh}S_{hh}^*>}{\sqrt{S_{vh}S_{hh}}}$ or for V transmit $\rho_h = \frac{<S_{hv}S_{vv}^*>}{\sqrt{S_{hv}S_{vv}}}$

Fortunately, the algorithms do not require us to know all of the individual gain terms. They cancel in the calculation of *r*, so they are taken as =1 in the implementation. However, the differential receiver gain LDR Offset must be known from calibration to calculate LDR:

dB value is $XDR = 10LOG\,xdr$ where the linear value is $xdr \frac{g_v^r}{g_h^r}$

2.6 Case 2: Simultaneous Dual Transmit and Receive (STAR)

2.6.1 Input Receiver Samples

In this mode, there is simultaneous transmit and receive of both vertical and horizontal polarization. For each pulse, there is a measurement of the amplitude in each channel, for example:

$$\left[S_{hh}^1 : S_{vv}^1\right]\left[S_{hh}^2 : S_{vv}^2\right]\left[S_{hh}^3 : S_{vv}^3\right]\ldots\left[S_{hh}^M : S_{vv}^M\right]$$

We assume that the *M* samples are collected for processing.

NOTE	Even though there is cross-polarized radiation received in each channel, this cross-polar contribution can be neglected, since the co-polarized received signal is much stronger.

2.6.2 Calculation of the Polarization Measurands

The processing in this case is done by pulse pair mode. However, both FFT and random phase processing can be performed, if only Z_{dr} and standard moments are requested for output. In any mode, the user may select a clutter filter.

The RVP900 calculates the following polarization parameters as follows:

$$ZDR = 10LOG\left[\frac{S_{hh}}{S_{vv}}\right]$$

$$ZDR = 10LOG\left[\frac{<|S_{hh}|-N_h>}{<|S_{vv}|-N_v>}\right]+GDR$$

$$RHOHV = |\rho_{hv}(0)|$$

$$PHIDP = \arg[\rho_{hv}(0)]$$

where the following definitions are used:

$$g_h^r g_h^t S_{hh} = <|S_{hh}|^2> - N_h \qquad g_v^r g_v^t S_{vv} = <|S_{vv}|^2> - N_v$$

The noise powers within the two channels are denoted as N_h and N_v. The noise corrections to S_{hh} and S_{vv} are optionally configured in the TTY setups. The Z_{dr} *Offset* is the total (transmit and receive) differential channel gain. It must be calibrated for the system as follows:

dB value is *ZDR Offset* = 10*LOG offset*

where the linear value is $offset = \frac{g_v^r g_v^t}{g_h^r g_h^t}$

The correlation function is computed as follows:

$$\rho_{hv}(0) = \frac{<S_{vv}S_{hh}^*>}{\sqrt{S_{hh}S_{vv}}}$$

The gain terms cancel in the calculation of ρ, so in the implementation they are simply assumed to be =1.

2.7 Case 3: Alternating H/V Transmit: Single Receiver

2.7.1 Input Receiver Samples

This is the traditional Z_{dr} radar with a high-power, fast switch that alternates between horizontal and vertical on each pulse. The switch is made just prior to the transmit pulse, so that the transmitter radiates, and then receives at a single polarization for each pulse. Thus, the samples are as follows:

$$S_{hh}^{1} \quad S_{vv}^{2} \quad S_{hh}^{3} \quad \ldots \quad S_{vv}^{M+1}$$

For the discussion below, we assume that there are M+1 total samples with M/2 horizontal pulses indexed by (1, 2, 3...M–1) and M/2+1 vertical pulses indexed at (2, 4, 6...M).

NOTE	The processor does not assume that the first pulse in a sequence is horizontal.

2.7.2 Calculation of the Polarization Measurands

The processing is done in pulse pair with optional clutter filter.

The RVP900 calculates the following:

$$ZDR = 10LOG\left[\frac{S_{hh}}{S_{vv}}\right]$$

$$ZDR = 10LOG\left[\frac{<|S_{hh}|^2> - N_h}{<|S_{vv}|^2> - N_v}\right] + GDR$$

$$PHIDP = \frac{1}{2}\arg\,[R_a R_b^*]$$

$$RHOHV = \frac{|\rho_{hv}(T_s)|}{[\rho_{hv} 2T_s]^{0.25}}$$

where the following definitions are used:

$$|\rho_{hv}(T_s)| = \frac{|R_a| + |R_b|}{2\sqrt{S_{hh} S_{vv}}}$$

$$\rho(2T_s) = \frac{\sum_{n=1}^{M/2-1} (s_{hh}^*[2n-1] s_{hh}[2n+1] + s_{vv}^*[2n] s_{vv}[2n+2])}{(M/2-1)(s_{hh} + s_{vv})}$$

$$R_a = \frac{1}{M/2}\sum_{n=1}^{M/2} s_{hh}^{2n-1*}\ s_{vv}^{2n}$$

and $$R_b = \frac{1}{M/2}\sum_{n=1}^{M/2} s_{vv}^{2n*}\ s_{hh}^{2n+1}$$

The calculation of the channel powers ($< |s_{hh}|^2 >$ and $< |s_{vv}|^2 >$) is done using alternating pulses.

NOTE	In the calculation of R_b, the RVP900 uses the extra M+1 sample. The gain terms cancel in the calculation of r, so in the implementation they are assumed to be =1.

2.8 Case 4: Alternating H/V Transmit: Dual Receiver

2.8.1 Input Receiver Samples

This is the most comprehensive case of polarization operation, since it permits calculation of all of the polarization measurands. In this case, the transmitter alternates pulse-to-pulse between horizontal and vertical polarization, and the dual-channel receiver provides measurement of both the co- and the cross-polarized return as follows:

$$\left[S_{hh}^1 : S_{vh}^1\right]\left[S_{vv}^2 : S_{hv}^2\right]\left[S_{hh}^3 : S_{vh}^3\right]\left[S_{hh}^3 : S_{hv}^3\right]\ldots\left[S_{vv}^{M+1} : S_{hv}^{M+1}\right]$$

We assume that M+1 samples are collected for processing (an extra sample is required for the calculation Rb per Section 2.7 Case 3: Alternating H/V Transmit: Single Receiver on page 21).

2.8.2 Calculation of the Polarization Measurands

The RVP900 calculates the following:

Co-polar channel measurements

Z_{dr}, PhiDP, RhoHV	Identical to alternating case, see Section 2.7 Case 3: Alternating H/V Transmit: Single Receiver on page 21.
Cross-polar channel measurements	
LDRH, RhoH, PhiH LDRV, RhoV, PhiV	Identical to fixed case, see Section 2.5 Case 1: Fixed Transmit: Dual-Channel Receiver on page 18.

The co-polar channel measurements are exactly as they are for the alternating single-receiver case. The cross–polar measurements are calculated using fixed case algorithms, except they are calculated for *both* H and V polarizations.

2.9 Standard Moment Calculations (T, Z, V, W)

2.9.1 Overview

Standard moments are available for all four of the polarization cases. Since there can be up to four different channels of time series input, there are several choices for computing the standard moments. For example, in the STAR mode (see Section 2.6 Case 2: Simultaneous Dual Transmit and Receive (STAR) on page 20), the standard moments are computed from:

- S_{hh} samples
- S_{vv} samples
- Average of the results from the S_{hh} and S_{vv} samples

NOTE The reflectivity moment can be computed from the S_{hh}, S_{vv}, and S_{hv} samples simultaneously in all three choices.

In Section 2.7 Case 3: Alternating H/V Transmit: Single Receiver on page 21, the case is handled by averaging the individual channel correlations, and then using the average correlations in the standard moment processing. The averaging must take into account the differential gain of the channels.

The method to use is selected in the setup. There are four questions in the mp section:

```
T/Z/V/W computed from: H-Xmt:YES V-Xmt:YES

T/Z/V/W computed from: Co-Rcv:YES Cx-Rcv:NO
```

The first two questions are used to specify that, *given a choice* between vertical and horizontal transmit, which transmit polarization should be used.

When choosing:

- *H-Xmt:YES V-Xmt:NO* logic, *V* and *W* are computed from the H-channel co-receiver
- *H-Xmt:NO V-Xmt:YES* logic, *V* and *W* are computed from the V-channel co-receiver
- *H-Xmt:YES V-Xmt:YES* logic, *V* and *W* are computed from the averaged correlations from both receivers

NOTE The RVP900 is always able to produce Z_H, Z_V, and Z_{HV} with any logical selection.

Thus, for the fixed H or V case, where there is only one transmit polarization, this question does not apply. The processor uses samples for the polarization that is transmitted.

The second two questions are used to specify that, *given a choice* between using the co- or cross-polar receivers, which one should be used. This question applies only to systems that can measure LDR, (fixed or alternating transmit, dual-channel receiver systems).

Table 3. Case 1H: Fixed Horizontal Transmit, Dual Channel Receive–(HH, VH) on page 27, Table 4 on page 29, Table 5 on page 31, and Table 6 on page 32 summarize the standard moment calculations for each of the four modes and how to configure the four TTY setup responses.

NOTE These are the only supported modes. Some combinations of responses are unsupported. For example, it is not supported to answer both *Co-Rcv: NO* and *Cx-Rcv: NO*.

Each table identifies the transmitter/receiver case and what samples are available. The notation HH signifies that the s_{hh} samples are available. The tables use "—" to indicate that either a YES or NO response causes the same result; the RVP900 does not care what response is made. In cases where averaging is performed, the type of weighting used is indicated.

2.9.2 Model for Standard Moment Autocorrelations

The model for the moment autocorrelation calculations is as follows (using R_0 as an example):

$$R_0^{hh} = g_h^r g_h^t S_{hh} + N_h$$

$$R_0^{vv} = g_v^r g_v^t S_{vv} + N_v$$

$$R_0^{hv} = \frac{g_h^r g_h^t S_{hh} + g_v^r g_v^t S_{vv}}{2}$$

where:

$R_0{}^{hh}$, $R_0{}^{vh}$, $R_0{}^{vv}$, $R_0{}^{hv}$	Are the autocorrelations if the samples at lag zero.
S_{hh}, S_{vh}, S_{vv}, S_{hv}	The average power returned from the scatterers.
$g^r{}_h$, $g^r{}_v$	Receiver gains for horizontal and vertical receive.
$g^t{}_h$, $g^t{}_v$	Transmitter gains for horizontal and vertical transmit.
N_h, N_v	Measured noise power of the samples.

The power that is measured in a channel has two components:

- Backscattered power from the targets that is effected by the transmitter and receiver channel gains
- Receiver noise, which is measured by the RVP900 during noise sampling.

In R1 and R2 autocorrelations, the model is similar, except there is no noise bias.

2.9.3 Calibration Parameters

For dBZ calculations, a calibration constant is required; the dBZ_o value found in the *RVP900 User's Manual - Digital Receiver and Signal Processor.* Depending on the polarization case and the technique selected

for standard moment calculation, it may also be required to have Z_{dr} Offset and LDR Offset:

- Z_{dr} Offset—The ratio of the total gains (transmit/receive) of the two co–receive channels.
- LDR Offset—The ratio of the receiver gains in a dual receiver system. This is not required for Section 2.6 Case 2: Simultaneous Dual Transmit and Receive (STAR) on page 20 or Section 2.7 Case 3: Alternating H/V Transmit: Single Receiver on page 21.

The RVP900 supports a single calibration reflectivity dBZ_o. In all cases, it is assumed that the $dBZ_o$is for the horizontal co–receive (HH) channel. The only exception is for fixed vertical polarization, in which the algorithm assumes that the calibration is for the vertical co-receive (VV) channel. LDR Offset and Z_{dr} Offset are also downloaded and used to adjust the dBZ_o, as required, depending on the user's selection for the standard moments. For example, in STAR mode, if the user selects dBZ to be computed from the VV channel, the dBZ_o for the HH and a Z_{dr} Offset adjustment are used to calculate the dBZ in the VV channel. For more information on how to calibrate LDR Offset and Z_{dr} Offset, see Section 2.10 Calibration Considerations on page 35.

Table 3 Case 1H: Fixed Horizontal Transmit, Dual Channel Receive– (HH, VH)

dBZ_o from HH Channel	**TTY Setup Question Responses**			
Calculate T, Z, V, W from:	**HXmt**	**VXmt**	**CoRcv**	**CxRcv**
HH (co) (Recommended)	—	—	YES	NO
VH (LDR Offset^{-1}weighting)	—	—	NO	YES
HH+VH (xdr^{-1}weighting)	—	—	YES	YES

2.9.3.1 HH Channel (Co-Pol)

HH channel (co-pol) is the recommended channel for the case of linear polarization. The reason is that for linear polarization, the co–polar channel has the strongest signal. Processing is identical to a conventional radar.

2.9.3.2 VH Channel (Cross-Pol)

VV channel (cross-pol) is used for circular or elliptic transmit polarization. Since the algorithm assumes that dBZ_o is from the co–polar channel, xdr is used to adjust the autocorrelations as follows:

$$T_0 = xdr^{-1} T_0^{vh}$$

$$R_{(0)} = xdr^{-1} R_0^{vh}$$

$$R_1 = xdr^{-1} R_1^{vh}$$

$$R_2 = xdr^{-1} R_2^{vh}$$

$$N = xdr^{-1} N_v$$

These adjusted autocorrelations are then used as input to the standard moment processing for a conventional radar. To illustrate this, consider the example of reflectivity processing. The radar equation can be written as follows (refer to the *RVP900 User's Manual - Digital Receiver and Signal Processor*):

$$Z^{vh} = C S_{vh} r^2 = \left[\frac{C r_0^2 N_v}{g_v^r g_h^t}\right]\left[\frac{r^2}{r_0^2}\right]\left[\frac{T_0^{vh} - N_v}{N_v}\right], \text{ where } T_0^{vh} = g_v^r g_h^t S_{vh} - N_v$$

$$= \left[\frac{C r_0^2 N_h}{g_h^r g_h^t}\right]\left[\frac{r^2}{r_0^2}\right]\left[\frac{g_h^r}{g_v^r}\right]\left[\frac{T_0^{vh} - N_v}{N_h}\right]$$

The third term is 1/XDR and is written as follows:

$$Z^{vh} = \left[\frac{C r_0^2 N_h}{g_h^r g_h^t}\right]\left[\frac{r^2}{r_0^2}\right]\left[\frac{xdr^{-1} T_0^{vh} - xdr^{-1} N_v}{N_h}\right]$$

The first term is the dBZ_o for the HH channel. We can use the dBZ_o for the HH channel to calibrate the cross-channel, if we first adjust the cross-channel noise and power by 1/xdr, and then normalize by N_h. The reflectivity calculation assumes that the calibrated xdr value compensates for any differences in the radar constant between the two channels; we do not need to have separate radar constants for the two channels.

2.9.3.3 HH+VH Channels

HH+VH channels are used for elliptic transmit polarizations that give comparable return signal in both the co- and cross-channels. The approach is to obtain average autocorrelation functions as follows:

$$T_0 = \frac{T_0^{hh} + xdr^{-1} T_0^{vh}}{2}$$

$$R_0 = \frac{R_0^{hh} + xdr^{-1} R_0^{vh}}{2}$$

$$R_1 = \frac{R_1^{hh} + xdr^{-1} R_1^{vh}}{2}$$

$$R_2 = \frac{R_2^{hh} + xdr^{-1} R_2^{vh}}{2}$$

$$N = \frac{N_h + xdr^{-1} N_v}{2}$$

These adjusted autocorrelations are then used as input to the standard moment processing for calibration with respect to the HH channel.

Table 4 **Case 1V: Fixed Vertical Transmit and Dual Channel Receive– (VV, HV)**

dBZo from VV Channel	**TTY Setup Question Responses**			
Calculate T, Z, V, W from:	HXmt	VXmt	CoRcv	CxRcv
VV (co)	—	—	YES	NO
HV (xdr weighting)	—	—	NO	YES
VV+HV (xdr weighting)	—	—	YES	YES

This is the only case for which the calibration constant dBZ_o for the VV channel should be downloaded to the signal processor.

2.9.3.4 VV Channel (Co-Pol)

VV channel (co-pol) is the recommended channel for the case of linear polarization. The reason is that for linear polarization, the co-polar channel has the strongest signal. Processing is identical to a conventional radar.

2.9.3.5 HV Channel (Cross-Pol)

HV channel (cross-pol) is used for circular or elliptic transmit polarization when most of the return is in the cross-pol channel. Since the algorithm assumes that dBZ_o is from the co-polar channel, xdr is used to adjust the autocorrelations as follows:

$$T_0 = xdr\ T_0^{hv}$$

$$R_0 = xdr\ R_0^{hv}$$

$$R_1 = xdr\ R_1^{hv}$$

$$R_2 = xdr\ R_2^{hv}$$

$$N = xdr\ N_h$$

These adjusted autocorrelations are then used as input to the standard moment processing with dBZ_o calibrated with respect to the VV channel.

2.9.3.6 VV+HV Channels

VV+HV channels are used for elliptic transmit polarizations that give comparable return signal in both the co- and cross-channels. The approach is to obtain average autocorrelation functions as follows:

$$T_0 = \frac{T_0^{vv} + xdr\ T_0^{hv}}{2}$$

$$R_0 = \frac{R_0^{vv} + xdr\ R_0^{hv}}{2}$$

$$R_1 = \frac{R_1^{vv} + xdr\ R_1^{hv}}{2}$$

$$R_2 = \frac{R_2^{vv} + xdr\ R_2^{hv}}{2}$$

$$N = \frac{N_v + xdr\ N_h}{2}$$

These adjusted autocorrelations are then used as input to the standard moment processing algorithms with dBZ_ocalibrated with respect to the VV channel.

Table 5 Case 2: Simultaneous Transmit and Receive– STAR (HH, VV)
Case 3: Alternating Transmit Single–Channel Receive (HH, VV)

dBZo from HH Channel	**TTY Setup Question Responses**			
Calculate T, Z, V, W from:	H–Xmt	V–Xmt	Co–Rcv	Cx–Rcv
HH	YES	NO	—	—
VV (Z_{dr} Offset^{-1}weighting)	NO	YES	—	—
HH+VV (gdr^{-1}weighting)	YES	YES	—	—

A fundamental difference between these two cases is that for all standard moment processing choices, the STAR case has double the number of samples as compared to the single-channel alternating case. However, the processing is otherwise identical.

2.9.3.7 HH Channel

Since the HH channel is directly calibrated, this is the recommended choice. Processing is identical to a conventional radar.

2.9.3.8 VV Channel

In VV channel, GDR is used to adjust the autocorrelations as follows:

$$T_0 = gdr^{-1} T_0^{vv}$$

$$R_0 = gdr^{-1} R_0^{vv}$$

$$R_1 = gdr^{-1} R_1^{vv}$$

$$R_2 = gdr^{-1} R_2^{vv}$$

$$N = gdr^{-1} N_v$$

These adjusted autocorrelations are then used as input to the standard moment processing algorithms with dBZ_o calibrated with respect to the HH channel.

2.9.3.9 HH+VV Channels

HH+VV channels approach gives the benefit of doubling the number of samples used for the reflectivity calculation as follow:

$$T_0 = \frac{T_0^{hh} + gdr^{-1} T_0^{vv}}{2}$$

$$R_0 = \frac{R_0^{hh} + gdr^{-1} R_0^{vv}}{2}$$

$$R_1 = \frac{R_1^{hh} + gdr^{-1} R_1^{vv}}{2}$$

$$R_2 = \frac{R_2^{hh} + gdr^{-1} R_2^{vv}}{2}$$

$$N = \frac{N_h + gdr^{-1} N_v}{2}$$

These adjusted autocorrelations are then used as input to the standard moment processing algorithms with dBZ_o calibrated with respect to the HH channel.

Table 6 Case 4: Alternating Dual–Channel (HH, VH, VV, HV)

dBZo from HH Channel	**TTY Setup Question Responses**			
Calculate T, Z, V, W from:	HXmt	VXmt	CoRcv	CxRcv
HH	YES	NO	YES	NO
VH (xdr^{1}weighting)	YES	NO	NO	YES
VV (gdr^{-1}weighting)	NO	YES	YES	NO
HV (xdr/gdr weighting)	NO	YES	NO	YES
HH+VV (gdr^{-1}weighting)	YES	YES	YES	NO
HV+VH (xdr & gdr weighting)	YES	YES	NO	YES

2.9.3.10 HH Channel

Since the HH channel is directly calibrated, this is the recommended choice. Processing is identical to a conventional radar.

2.9.3.11 VH Channel

Processing is identical to Case 1H (see Table 3. Case 1H: Fixed Horizontal Transmit, Dual Channel Receive– (HH, VH) on page 27).

2.9.3.12 VV Channel

Processing is identical to Case 2 (see Section 2.6 Case 2: Simultaneous Dual Transmit and Receive (STAR) on page 20) and Case 3 (see Section 2.7 Case 3: Alternating H/V Transmit: Single Receiver on page 21).

2.9.3.13 HV Channel

The weighting in HV channel uses both xdr and gdr as follows:

$$T_0 = \frac{xdr}{gdr} T_0^{hv}$$

$$R_0 = \frac{xdr}{gdr} R_0^{hv}$$

$$R_1 = \frac{xdr}{gdr} R_1^{hv}$$

$$R_2 = \frac{xdr}{gdr} R_2^{hv}$$

$$N = \frac{xdr}{gdr} N_h$$

These adjusted autocorrelations are then used as input to the standard moment processing algorithms with dBZ_o calibrated with respect to the HH channel.

2.9.3.14 HH + VV Channels

Processing is identical to Case 2 (see Section 2.6 Case 2: Simultaneous Dual Transmit and Receive (STAR) on page 20) and Case 3 (see Section 2.7 Case 3: Alternating H/V Transmit: Single Receiver on page 21).

2.9.3.15 HV + VH Channels

The weighting in HV + VH processing has to correct for both transmitter and receiver effects in order to use the HH channel dBZ_o as follows:

$$T_0 = \frac{\frac{xdr}{gdr} T_0^{hv} + xdr^{-1} T_0^{vh}}{2}$$

$$R_0 = \frac{\frac{xdr}{gdr} R_0^{hv} + xdr^{-1} R_0^{vh}}{2}$$

$$R_1 = \frac{\frac{xdr}{gdr} R_1^{hv} + xdr^{-1} R_1^{vh}}{2}$$

$$R_2 = \frac{\frac{xdr}{gdr} R_2^{hv} + xdr^{-1} R_2^{vh}}{2}$$

$$N = \frac{\frac{xdr}{gdr} N_h + xdr^{-1} N_v}{2}$$

These adjusted autocorrelations are then used as input to the standard moment processing algorithms with dBZ_o calibrated with respect to the HH channel.

Suppose that we want to compute the average of the reflectivities for the VH and HV channels. An example of how this weighted averaging works is as follows:

$$Z^{hv+vh} = Cr^2 \frac{S_{hv} + S_{vh}}{2}$$

$$= Cr^2 \frac{\frac{T_0^{hv} - N_h}{g_h^r g_h^t} + \frac{T_0^{vh} - N_v}{g_v^r g_h^t}}{2} = \frac{Cr^2}{g_h^r g_h^t} \frac{\left(T_0^{hv} - N_h\right)\frac{g_h^t}{g_v^t} + \left(T_0^{vh} - N_v\right)\frac{g_h^r}{g_v^r}}{2}$$

but since $xdr = \frac{g_v^r}{g_v^h}$ and $gdr = \frac{g_v^r g_v^t}{g_h^r g_h^t}$

$$Z^{vh+hv} = \frac{Cr^2}{g_h^r g_h^t}\left[\frac{\frac{xdr}{gdr} T_0^{hv} + xdr^{-1} T_0^{vh}}{2} - \frac{\frac{xdr}{gdr} N_h + xdr^{-1} N_v}{2}\right]$$

$$Z^{vh+hv} = \frac{Cr^2}{g_h^r g_h^t}[T_0 - N] = \left[\frac{Cr^2 N_h}{g_h^r g_h^t}\right]\left[\frac{r^2}{r_0^2}\right]\left[\frac{T_0 - N}{N_h}\right]$$

The first term in brackets is precisely dBZ_o for the HH channel. If we average the correlations using the appropriate gdr and xdr weighting, the average reflectivity is obtained by using conventional processing with the HH channel dBZ_o.

2.10 Calibration Considerations

Polarization systems require additional calibration as compared to conventional systems. There are three aspects to the calibration:

- dBZ_o measurement in both channels for dBZ and dBT calibration.
- gdr measurement for Z_{dr} calibration.
- xdr measurement for LDR calibration.

These are discussed below.

dBZ_oCalibration for dBZ

The RVP900 supports separate calibration of both polarization channels. Measurement of dBZ_o for each channel of a dual polarization system is identical to the conventional radar case described in the *RVP900 User's Manual - Digital Receiver and Signal Processor*. Note that for a single-channel switching system, the only difference between the horizontal and vertical signal paths occurs after the high power switch, that is, differential

insertion loss of the switch itself and any differential insertion loss of the waveguides and feed after the switch. This means that for single-channel switching systems it may be sufficient to calibrate at one polarization and then adjust the calibration of the other channel by the differential gain GDR (see below).

GDR Calibration for Z_{dr}

The Z_{dr} Offset is the dB value of the relative gain between the co-polarized channels including both transmitter and receiver gain, that is,

$$ZDROffset = 10LOG\ \frac{g_v^r g_v^t}{g_h^r g_h^t} \quad \text{and} \quad gdr = \frac{g_v^r g_v^t}{g_h^r g_h^t}$$

GDR is input into the processor as a dB value. However, for analyses in this chapter, the linear gdr value is sometimes more convenient.

In principle, if dBZ_o could be calibrated perfectly in both channels, measurement of GDR would not be required. In practice, this is not possible because dBZ_o cannot be calibrated to an absolute accuracy sufficient for Z_{dr}, that is, to 1/16th of a dB. Therefore, the RVP900 uses the GDR approach.

Since GDR includes both transmitter and receiver differential gains, accurate calibration requires that an actual target be observed. One way to do this is as follows:

- Set the GDR to be 0 dB using your application software (for example, for Vaisala IRIS systems in the setup utility RVP section). Disable clutter filtering for Z_{dr} in either your application software (by selecting filter 0) or explicitly in the RVP900 TTY setups mp section.
- Place the antenna at 90 degrees elevation (vertical incidence) during moderate to heavy rain. The melting layer should be at a height that is well above the recovery zone of the T/R and in the antenna "far zone". A melting layer higher than 2 km is suggested, but the specific characteristics of the radar should be considered.
- Collect Z_{dr} data at vertical incidence while the antenna is rotating in azimuth.
- Use a separate application program to average the Z_{dr} values around a full 360 degrees at each range bin (height). Generate a plot of 360-average Z_{dr} vs height.
- You should observe that the average Z_{dr} values in regions of strong signal (>20 dB SNR) below the bright band are approximately

constant with height. This is the value that should be used in your application software for GDR.

- Enter the value and repeat the calibration to verify that the average Z_{dr} is now 0 dB.

The rationale for this approach is as follows. When viewed at vertical incidence, rain should have a Z_{dr} of 0 dB since the drops will all appear circular. The reason for averaging over 360 degrees is to cancel-out effects from sidelobe contamination from nearby ground targets and other artifacts of the antenna/feed/radome system. For example the radome may have an obstruction light on the top. Some of these artifacts can be minimized by assuring the weather targets are strong, that is, heavy rain is preferred for this calibration.

Offset Calibration for LDR

XDR is the dB value of the relative gain between the co- and cross-receiver channels for LDR measurements. Analogous to GDR, it is defined as the dB value of the ratio of the vertical to horizontal receiver gains, for example,

$$XDR = 10LOG\frac{g_v^r}{g_h^r} \quad \text{and} \quad xdr = \frac{g_v^r}{g_h^r}$$

Three techniques for calibration of XDR are discussed. It is recommended for the transmitter to be off for all of these methods.

- **Solar Method**

  Use the sun to measure LDR. The measured value of LDR is then the XDR offset. LDR should be measured in fixed mode for both LDRH and LDRV. The values should be reciprocal (for example, +1 dB and -1 dB). Use the average of the absolute value if they are not precisely reciprocal (for example, for +1.4 and -1.2 use 1.3). Finally after inputting the XDR value, retest to verify that the sun has been properly corrected to have zero LDR.

- **Signal Generator Method with Connection to Waveguide**

  Connect a signal generator with a splitter to both channels and measure XDR directly. This does not account for any effects that are before the coupler (for example, waveguide, feed, radome, antenna gain).

- **Linear Feed Horn Remote Radiator Method**

  Use a calibrated linear feed horn with an RF source located several hundred meters from the radar. Maximize the H channel return and

measure the response using the RVP900 pr command "Filtered" power in the "Primary Channel". Now rotate the feed horn to vertical and maximize the power in the "Secondary Channel". The difference in dB is XDR. Note that signal multi–path effects could bias the results from this technique.

In all cases it is recommended that for the calibration, XDR be set to 0 dB in the application user software and that the RVP900 TTY setups be configured as follows:

- Noise correction enabled for LDR and noise sample taken prior to the measurements (with care not to sample with a test signal turned–on or while looking at the sun).
- Clutter correction disabled for LDR.

CHAPTER 3

ATTENUATION CORRECTION OF Z

3.1 Overview

Within dual polarization radars, the measurements still consists of measuring the amplitude and phase of the received waveforms in the two separate channels. PhiDP is the phase difference between the two polarization channels at each range bin.

The PhiDP quantity primarily comes from the propagational effects, as the RF energy passes through some medium to the measurement point and back again to the transmitter/receiver. The PhiDP measurement tends to be a noisy, unstable quantity, due to the slightly different backscattering phase shifts from the scatterers at the range point and statistical measurement errors. Furthermore, the unambiguous range of PhiDP is 360 degrees in the simultaneous mode of operation. However, phase changes more than 360 degrees are common, especially with smaller wavelength radars, and will cause PhiDP to be phase wrapped.

PhiDP is directly related to the amount of liquid water content the transmitted energy passed through, to get to the scatterers at a specific range bin. Therefore, PhiDP has excellent advantages, when used to estimate attenuation caused by liquid water. The range derivative, or specific differential phase (K_{dp}), also becomes a key data type to indicate rainfall rates with dual polarization radars.

However, prior to using PhiDP for any of these applications, it must be conditioned and unfolded. Here, conditioning means to smooth out the inherent noisy properties of the data, yet be able to capture and maintain the range and amplitude resolution common within intense rain cells. Section 3.2 PHIDP Conditioning on page 41 describes two methodologies available to condition and unfold PhiDP. This conditioned PhiDP is fed into a dual polarization-based attenuation correction algorithm described in Section 3.3 Dual Polarimetric Attenuation Correction on page 43, and

an adaptive K_{dp} algorithm described in Section 3.4 Adaptive K_{dp} Moment Estimation on page 48. Section 3.5 User Instructions on page 49 describes the configuration parameters and default settings for user controllable parameters.

After all these are applied, the IRIS and RDA software makes optimal use of these dual-channel measurements, creating data types for the following applications:

- Hydrometeor particle identification
- Lightning hazard potential forecasting
- Detection of convection and stratiform rain
- Highway snow removal
- Airport terminal operation
- Rain/snow line demarcation
- Melting height detection
- Weather modification for hail mitigation
- Insurance industry claims verification
- Military detection of chaff
- Data quality improvement by elimination of non-meteorological targets
- Improved precipitation forecasting
- Hydrological modeling

Dual polarimetric attenuation correction is a collection of algorithms, which may be performed in real-time, within the RVP900 signal processors, or within the IRIS application software. The RDA software has the advantage of applying immediate corrections, with the signal processor being available on real-time data output. This becomes important in mission critical now-casting within the 0 to 1 hour time frame, where the latest observed information is needed for precise decision-making. The IRIS processing has an advantage since the original Z and Z_{dr} moments are uncorrected for attenuation; IRIS makes new data types. This allows users to compare the results of the correction.

- **RVP900 DP Attenuation Correction Processing**—Ray data are processed in the RDA, and the bin level moments are output in real-time for each range bin. The advantage of RDA is that all moments are available in every bin, whereas when performing the function in IRIS, certain moments within the bins will have been thresholded. This approach is well suited to applications where IRIS software is not active, since the particle type can be displayed directly by the customer's display software.

- **IRIS DP Attenuation Correction Processing**–Dual polarization data comes from the RVP900, or from a third party processor, and are passed to an IRIS/Radar or an IRIS/Analysis system that is enabled with the dual polarization feature license. The bin level algorithms produce ingest data, which can be formatted as RAW products for archival, or for rerouting in the network. The correction algorithm can also be applied to historical data.

3.2 PHIDP Conditioning

3.2.1 PHIDP Unfolding and Conditioning

When computed for each range bin, PhiDP is inherently noisy, especially in volume samples not having meteorological scatterers present. Additionally, PhiDP can typically have values greater than 360 degrees of phase difference, causing folding within the valid data ranges used within IRIS/RDA. Therefore, the first processing steps are to smooth and unfold the PhiDP data.

There are two methodologies to unfold and condition the PhiDP data. The first methodology is called the least squares fit (LSQ).[21] The second methodology is called the cubic spline fit.[5] The cubic spline fit is the recommended process. The methodology to be executed is a user-selectable option in IRIS and RDA processing.

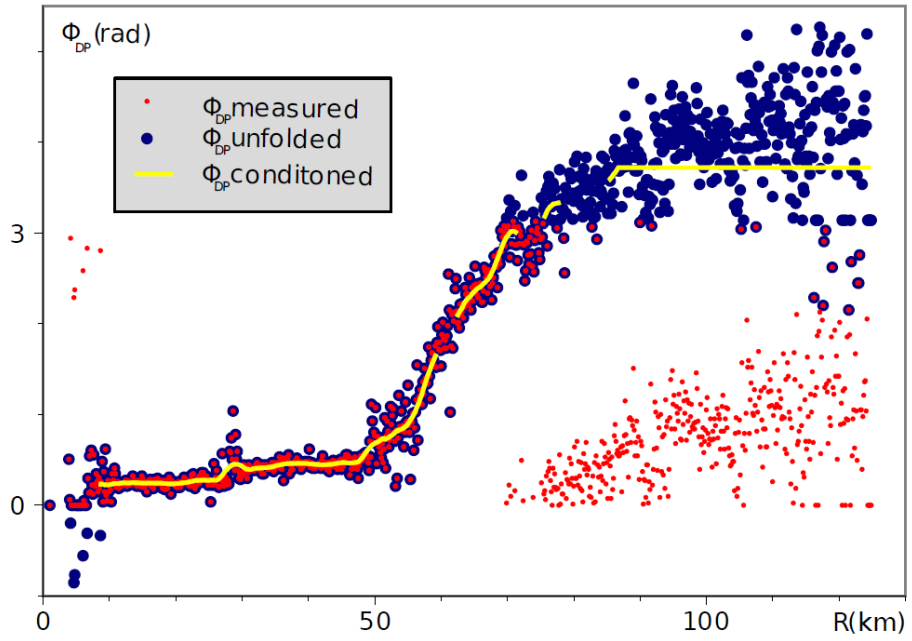


Figure 1 Unfolding and Conditioning of Bin to Bin PhiDP

3.2.1.1 Least Squares Fit

In the LSQ model, PhiDP unfolding procedure is combined with the conditioning/filtering of PhiDP. The measured PhiDP are stored as 32-bit unsigned integers. The PhiDP ambiguity interval is mapped linearly into the full range of unsigned integers. At each bin, an LSQ, weighted with FIR filter coefficients, is performed to compute "conditioned" PhiDP. This LSQ is performed thee times at each bin, each having a unique phase shift being tracked. One phase shift, φ, is constantly adjusted to keep the current PhiDP in the middle of the ambiguity interval. For the other two LSQs, the phases are shifted by the ambiguity interval, φ ± ⅔. The LSQ with the lowest signal-to-noise ratio is assumed to be the unfolded quantity. Figure 1 on page 42 shows the input data and outcomes of this LSQ technique.

3.2.1.2 Cubic Spline Fit

LSQ is a practical phase unfolding approach, however; it creates a test condition to check for phase unfolding when at times the test may fail.

The cubic spline fit approach transforms the PhiDP between the two receive channels to the complex domain, creating an Angular PhiDP vector. This is a similar concept used when creating I,Q for signal processing. The advantage in using Angular PhiDP is that the data remains continuous in the complex domain, where as the traditional PhiDP

moments in the real domain phase wraps with abrupt jumps. To compute the Angular PhiDP moment:

$\vartheta(r) = e^{j\psi_{dp}(r)}$ for STAR mode dual polarization

$\vartheta(r) = e^{j2\psi_{dp}(r)}$ for alternating mode of dual polarization

Where $\vartheta(r)$ is the Angular PhiDP at some range and ψ_{dp} is the total differential phase. Conditioning and smoothing of Angular PhiDP is described in Section 3.4 Adaptive K_{dp} Moment Estimation on page 48.

3.3 Dual Polarimetric Attenuation Correction

Dual polarimetric attenuation correction is a collection of algorithms, using the unfolded and conditioned PhiDP. These algorithms may be performed in real-time within the RVP900 signal processors, or within the IRIS application software. The RDA software has the advantage of applying immediate corrections with the signal processor being available on real-time data output. This becomes important in mission critical now-casting within the 0 to1 hr time frame, where the latest observed information is needed for precise decision making. The IRIS processing has an advantage since the original Z and Z_{dr} moments are uncorrected for attenuation; IRIS makes new data types. The IRIS processing also allows for creating the corrected moments with archived data. This allows users to compare the results of the correction.

- **RVP900 DP Attenuation Correction Processing**—Ray data are processed in the RDA, and the bin level moments are output in real-time for each range bin. The advantage of RDA is that all moments are available in every bin, whereas when performing the function in IRIS, certain moments within the bins have been thresholded. This approach is well suited for applications where IRIS software is not active, since the particle type can be displayed directly by the customer's display software.
- **IRIS DP Attenuation Correction Processing**—Dual polarization data comes from the RVP900, or from a third party processor, and are passed to an IRIS/Radar or an IRIS/Analysis system that is enabled with the dual polarization feature license. The bin level algorithms produce ingest data, which are formatted as RAW products for archival, or for rerouting in the network. The correction algorithm can also be applied to historical data.

Disadvantages to using IRIS are the time delay to have the corrected moments and that data is likely already been thresholded causing some range bins to have missing data.

Signal attenuation in precipitation is a well known problem for weather radars using smaller wavelengths, like 3 to 5 cm in the X- and C-band spectrums. The attenuation causes a decrease in the Z and Z_{dr} measurements, making their use in quantitative analysis limited. The attenuation effects also impact the outcomes of the 'fuzzy logic' set of algorithms like HydroClass. Therefore, it is important to estimate the amount of attenuation, and make corrections to the Z and Z_{dr} data types prior to processing algorithms. These corrections ultimately improve the ability to accurately estimate rainfall, differentiate hydrometeors, and assess data quality metrics.

In case of the dual polarization radar, the differential propagation phase moment PhiDP, provides an excellent method to estimate attenuation of Z and Z_{dr}, as the change in PhiDP is directly related to the liquid water content along the propagation path. PhiDP is not affected by attenuation or calibration errors, but allows large, accurate corrections.

The traditional single polarization attenuation correction is based on the following relationship:

$$A(r) = a[Z(r)]^b$$

Figure 2 Equation 1

where $A(r)$ is the specific attenuation and $Z(r)$ is the intrinsic reflectivity at any particular range gate. This particular equation is implicit, where an unknown value $A(r)$, is expressed through another unknown value $Z(r)$. The expression to solve for total two-way attenuation is:

$$dBZc = dBZ + 2C\Delta r\sum Z^E$$

Figure 3 Equation 2

where $2C\Delta r\sum Z^E$ term estimates the total two-way attenuation to the range bin, based on the accumulated reflectivity measurement. C and E are constants in this estimation. This equation becomes unstable very quickly and must be highly constrained in order that the corrected reflectivity does not end up having less value than the original.

With dual polarization radar an independent estimate of the specific attenuation is possible:

$$A(r) = \alpha K_{dp}(r)$$

Figure 4

In practice, the PhiDP(r) measurements to product K_{dp} have much lower accuracy then $Z(r)$, and deviate significantly from the pure "propagation" component. However, the estimate of total cumulative attenuation along the radial is quite accurate and can be used as a constraint for the range specific attenuation corrections.

The IRIS/RDA implementation of dual polarimetric attenuation correction is illustrated in Figure 5 on page 45.

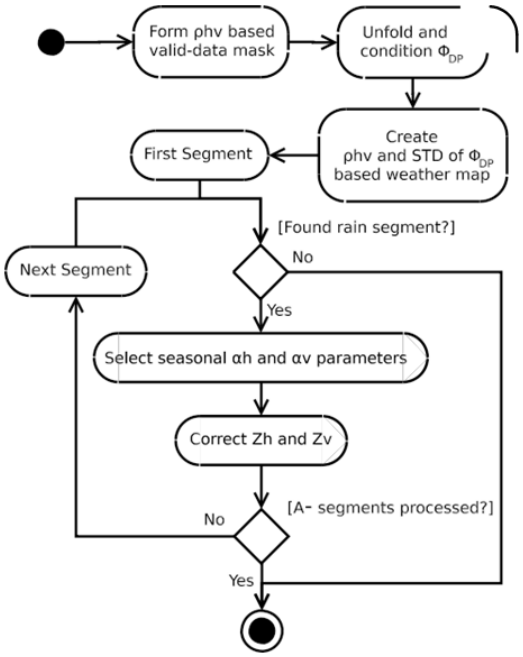


Figure 5 Dual Polarimetric Attenuation Correction Processing Stages

3.3.1 Weather Classification Algorithm

The first processing step is to identify segments along the radial where attenuation occurs. The radial is broken into one or more segments where each segment is either flagged as "liquid rain" or "no weather". The liquid

rain portions of the segment are used to calculate the constrained PhiDP. This step is a hydrometeor classifier.

The simple weather classification ensures that changes in PhiDP from non-meteorological targets like ground clutter, point targets, refractivity gradients, and insects do not introduce errors in the attenuation estimate. Reflectivity and differential reflectivity values outside of the segments, identified as rain, are not corrected. Also, small ice particles add only a minute amount to the overall one-way path attenuation.Therefore, once the beam propagation along the radial becomes higher than the freezing level, additional attenuation is no longer calculated.

The weather classification algorithm relies on three known principles or rules about hydrometeor particles:

- Hydrometeors particles tend to have a high value of the co-polar correlation coefficient, RhoHV. A mean RhoHV is computed over a defined length. If the mean value becomes lower than the threshold, range bins are flagged to not use the data. The default value used within this classifier is 0.85.
- Quantitative measurements of PhiDP tend to be highly correlated, from range bin to range bin, when hydrometeor targets are present. This gives a "smooth" looking data field versus more "noisy" fields that are likely to have non-meteorological particles. Therefore, the texture over some area is used as a discriminator. The standard deviation of PhiDP is calculated over the same range as the mean RhoHV. This is used to represent texture along the radial.
- Convective and stratiform meteorological events have some size to them; typically more than a few kilometers. This rule is used to decide when to enter a rain segment and how long this segment must be before exiting to a no weather segment.

Each range bin is uniquely evaluated as being rain or no weather using Equation 1 and Equation 2; however, to have a rain segment.

3.3.2 Iterative PhiDP Correction

For the segments identified as rain, a first guess of intrinsic reflectivity is created. The approach is to use the same expression seen in Equation 2, but substituting the change of PhiDP from the beginning to the end of the segment for Z. This appears as:

$$dBZc = dBZc + 2C\Delta r \sum (\theta_{r\,\min} - \theta_{r\,\max})^{E}$$

Figure 6 Equation 3

For the first rain segment nearest to the radar, the system PhiDP values are initially set by a configuration setting in the RVP900 dspx menus. However, this system PhiDP value is constantly being updated with the average PhiDP values seen in the first bins of the rain segments.

The initial guess of the intrinsic reflectivity is then adjusted using an iterative process, which is constrained by the total attenuation along the entire radial. The total attenuation is now determined by:

$$A = ½\ \alpha_h\ \Delta\ \Phi_{DP}$$

Figure 7 Equation 4

where the α coefficient is a constant, slightly dependent on temperature. The adjustment is performed by using the ratio of the total attenuation and a summation of measured reflectivity values to the particular range bin to scale the correction.

$$dBZc = dBZ(r) + \frac{A}{\sum_{r=\min}^{r=\max} 10^{0.1b(dBZ(r-1))}} \sum_{r=\min}^{r} 10^{0.1b(dBZ(r-1))}$$

Figure 8 Equation 5

In this manner, the sum of the corrections applied to each range bin are equivalent to the total attenuation, as determined by $\Delta\Phi_{DP}$.

This first guess of intrinsic reflectivity, sometimes called the DP method, may be output from IRIS or the RVP900 software for research purposes. For operational radars, the full IDPC methodology should be used and is the default selection.

3.4 Adaptive K_{dp} Moment Estimation

The specific differential phase, K_{dp}, is the range derivative of the Angular PhiDP, described in Section 3.2.1.2 Cubic Spline Fit on page 42 is used. This is written as:

$$K_{dp}(r) = -j\left\{\frac{\vartheta'(r)}{\vartheta(r)}\right\} + e(r)$$

Figure 9

Recall that Angular PhiDP has no folding, but is still noisy due to backscattering effects and measurement errors. When performing differentiation of noisy data in the more traditional LSQ method of calculating K_{dp}, the errors are magnified within the K_{dp} output. With the continuous complex form of Angular PhiDP data, a cubic spline methodology is chosen, because it has the greatest smoothness over all functions evaluating derivatives. Any smoothing function can be overly aggressive causing loss in data fidelity; the over smoothed situation. Therefore, an adaptive process is created to adjust the smoothing function.

The weather classification algorithm described in Section 3.3.1 Weather Classification Algorithm on page 45 is re-used for calculating K_{dp}. The weather classification algorithms defines segments having liquid rain versus other segments. In areas where there is no liquid rain, K_{dp} should be zero. This assumption provides a convenient end condition to determine all the coefficients needed in the cubic spline equation.

The cubic spline processing[34] of the Angular PhiDP is performed in two steps. The first pass uses a standard fixed smoothing function, which evaluates the mean and dispersion of Angular PhiDP. The dispersion results from the first pass is then used as a weighting function to adapt or scale the smoothing function to match the properties of the Angular PhiDP data. This adaptive function ensures that the trade-off between smoothness and data fidelity is optimized.

3.5 User Instructions

3.5.1 Install, Licensing, and Activation in Setup

A valid dual polarization license code is needed to use the dual polarization attenuation correction function. The code is activated and encrypted in the setup utility GUI field "License". Two use cases are possible:

- **Real-time correction in RVP900**, which creates the original Z and ZDR moments already corrected. To activate the feature, ensure that:
  - Signal Processor is licensed for dual polarization
  - Signal Processor fundamental operating mode in dspx settings is configured for dual polarization

#	BW	DynR	Filt	Pol	IFD	Description
0	**Full**	**Norm**	**Norm**	**1**	**1**	**Standard single channel**
2	**Full**	**Norm**	**Norm**	**2**	**2**	**Dual Pol on separate IFDs**
3	**Half**	**Norm**	**Norm**	**2**	**1**	**Dual Pol on single IFD**
4	**Half**	**Wide**	**Norm**	**1**	**1**	**Extra wide dynamic range**
5	**Half**	**Norm**	**Long**	**1**	**1**	**Extra long/fast FIR filters**

Fundamental RVP900 Operating Mode: 3

  - Dual Polarization attenuation filter is enabled by selection in the mp TTY setups:

Polarimetric Attenuation Correction:"USER" or "ALWAYS"

- **IRIS dual polarization attenuation correction** functions during the ingest/re-ingest process. To activate the correction:

  In IRIS task configuration menu within the processor configuration, enable DP Attn Cor Z ZDR and remember to set in the dspx utility:

Polarimetric Attenuation Correction:"USER"

  In IRIS/Setup Utility within the ingest tab, enable the **DP Atten correction in ReINGEST**, then restart the IRIS system.

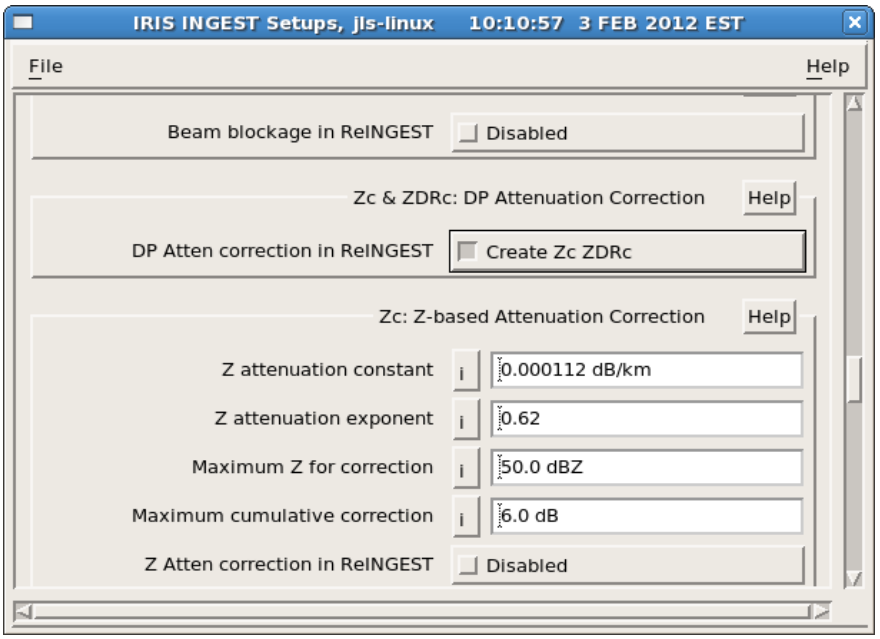


Figure 10 IRIS/Setup Utility - Ingest Tab

The installed and activated Dual Polarimetric Attenuation Correction runs promptly on appropriate input data and computes results using the current configuration. The configuration in both cases is specified within the file / usr/sigmet/config/dualpol.conf. This file is explained in the Section 3.5.2 Dual Pol Configuration File on page 50.

3.5.2 Dual Pol Configuration File

All distinct parameter lines are listed and described below, with their default settings for C-band radars.

The first section of the configuration file sets the parameters used to calculate the actual attenuation at each range bin.

```
#================================================#
#Attenuation correction
# Algorithm selection. Available options:
# DP - propagational depolarization based attenuation
#      (dBZ_real(r) = dBZ_measured(r) + alpha PhiDP(r)).
# IDPC - iterative DP constrained attenuation correction.
#        Single iteration with DP corrected dBZ(r) as
#        an initial dBZ(r).
algorithm = IDPC
```

This setting selects of the dual polarimetric attenuation corrections is based on the iterative constrained process or only the first guess field. The recommended setting for all operational systems is IDPC.

```
# DP attenuation constant for horizontal polarization
alpha_h = 0.105
```

This setting sets the value of the constant used within equation 4) when computing the total horizontal reflectivity attenuation along the radial. The 0.08 value is the recommended global setting.

This value is slightly variable with temperature so we provide reference to values, which may be appropriate in other climates. Jameson 1991 provides estimated `alpha_h` correction coefficients for several different wavelengths and mean temperatures. These are listed below:

Temperature (°C)	S-band (2.80 GHz)	C-band (5.48 GHz)	X-band (9.34 GHz)
0	**0.026**	**0.097**	**0.248**
10	**0.019**	**0.081**	**0.241**
20	**0.015**	**0.066**	**0.230**
30	**0.012**	**0.053**	**0.213**
40	**0.009**	**0.043**	**0.195**

```
# DP attenuation constant for vertical polarization
alpha_v = 0.065
```

This setting sets the value of the constant used within equation 4) when computing the total vertical reflectivity attenuation along the radial. The 0.105 value is the recommended setting. The value is slightly variable with temperature.

```
# The melting layer (aka bright band) above sea level.
# No attenuation is added in the part of the ray above ML
melting_layer = 1300.0
```

This setting sets the height of the 0 degree C level above the surface reference height in IRIS. This sets a limit in the correction algorithm to not add accumulate additional attenuation when the beam is above this height. This value is only used within the IRIS/Reingest process. The RVP900 uses the melting level values specified in setup memory location.

```
# The above value for melting layer is used only if melting
# layer altitude is not set for processed data set or if
# the following flag is set to 'yes'
force_conf_melting_layer = no
```

When re-ingesting data into IRIS and performing dual pol attenuation correction, setting this question to "yes" overwrites the melting level height originally stored with the data. This allows users to correct wrong heights or add a height value to archived data which may not have had the variable stored.

```
# The b parameter from the reflectivity based attenuation
# equation: A(r) = a Z(r)^b
b_exp = 0.78
```

This sets the value of the exponent within the summation terms seen in equation 5. The recommended value for c-band radars is 0.78 and 0.46 for x-band radars.[7]

The PhiDP section of the configuration file sets the parameters used to condition and unfold PhiDP for input to the attenuation correction, either the LSQ or Cubic Spline techniques.

```
# PHIDP processing parameters

# PHIDP processing algorithm type (in IRIS reingest, use
# dspx for RVP):
# LSQ - Weighted least square fit to a linear function
#       FIR filter coefficients are used as weights. Use
#       fir_width to control width of the filter.
# SPLINES - Smoothing and numerical derivatives calculation
#           using cubic splines algorithm[5]
PhiDP_alg = SPLINES
```

This parameter allows you to choose SPLINES or LSQ method during IRIS/Reingest processing. Within RVP900 processing a similar question is found in dspx mp menu.

```
# Smoothing factor for the first, non-adaptive, stage of
# cubic splines processing (in IRIS reingest, use
# dspx for RVP)
standard_smoothing_factor = 0.1
```

For IRIS/Reingest, this parameter sets the smoothing factor during the first pass through the Angular PhiDP data when performing cubic spline processing. Within RVP900 processing a similar question is found in dspx mp menu.

```
# Smoothing factor for the second, adaptive, part of
# cubic splines processing (in IRIS reingest, use
# dspx for RVP)
adaptive_smoothing_factor = 1.1
```

For IRIS/Reingest, this parameter sets the adaptive smoothing factor during the second pass through the Angular PhiDP data when performing cubic spline processing. This factor will be adjusted with a weighting function. Within RVP900 processing a similar question is found in dspx mp menu.

```
# The width, in km, of FIR filter used as weights in
# LSQ processing. (in IRIS reingest, use
# dspx for RVP)
fir_width = 3.0
```

For IRIS/Reingest attenuation correction, the impulse response of the FIR filter is configured in kilometer widths. Within the RVP900 processing these parameters are set through the dspx mp menu interface. This variable also sets the range window for calculating K_{dp}.

The data thresholding parameters are used to reduce the amount of data being considered during the attenuation correction process. This is performed to reduce the CPU load on the computer.

```
# ============================================================
# Data thresholding parameters (for processing mask)

# The shortest distance, in km, at which polarimetric
# measurements, such as PHIDP, RHOHV, ZDR, etc, are
meaningful.
recovery_range = 1.0
```

Close to the radar site the dual polarimetric data values may not be representative of the intrinsic values due to near-field antenna issues. This setting specifies an initial range limit to ignore during the attenuation correction processing.

```
# Lowest allowed/meaningful SNR
snr_lim = 3.0
```

With very weak signals the PhiDP measurements become very noise. A signal to noise ratio threshold may be set. This flags range bins, having SNR below this value, to not be included into the calculations.

```
# Lowest allowed/meaningful RHOHV
rhohv_lim = 0.8
```

Hydrometeor particles typically have RhoHV values greater than 0.85. The setting creates a threshold to not allow data from range bins having RhoHV values below the criteria to enter the calculations.

The weather classification algorithm parameters are used in creating the rain versus no weather classifications discussed in Section 3.2.1.2 Cubic Spline Fit on page 42.

```
# ============================================================
# The weather classification algorithm parameters. (for the
# build in weather mask)

# Number of sequentially "good" bins needed to transition
# from "no weather" to "rain".
# Note: It is also the width of the gate used to compute
# STD of PHIDP, and mean value of RHOHV. [bins]
long_gate = 15
```

This question defines the width of a sliding window to calculate the average RhoHV and standard deviation of PhiDP at specific range bins. The window width is defined in range bin units.

```
# Number of sequentially "bad" bins needed to transition
# to "no weather". [bins]
short_gate = 10
```

When the range bin is currently in a rain segment, but the average RhoHV and PhiDP standard deviation are below the thresholds, this setting states how many consecutive range bins must be below the criteria to exit the rain segment.

```
# Lowest allowed mean RHOHV at the transition to "rain"
rhohv_enter_r = 0.85
# Highest standard deviation of PhiDP at the transition
# to "rain" [deg]
PhiDP_std_enter_r = 7
```

These two criteria are the definitions when the range bin is considered to be the start or a rain segment. Both criteria must be met to define the start of a rain segment.

```
# Low mean RHOHV that triggers exit from "rain"
rhohv_exit_r = 0.70

# High standard deviation of PhiDP that triggers exit
# from "rain" [deg]
PhiDP_std_exit_r = 15.0
```

This sets the thresholds when the range bin is thought to be bad and should be in the No Weather category. When either of the actual values are below the average RhoHV or above the standard deviation of PhiDP, the bin is flagged as bad. When a consecutive number of [short gate] bins are flagged as bad the rain segment transitions to No Weather.

```
# Lowest allowed SNR in "hail" region
# Note: Should be higher then snr_lim
# snr_exit_h = 8.0
# Lowest allowed RHOHV in "hail" region.
# Note: Should be lower then rhohv_lim_r.
# rhohv_exit_h = 0.6

# Standard deviation of PhiDP that triggers exit from "hail"
# Note: Should be higher then PhiDP_std_exit_r
# PhiDP_std_exit_h = 20.0
```

As hail is uniquely different than the presumed rain and No Weather classification, there is a different amount of attenuation than with liquid hydrometeors. In the future, the attenuation correction algorithm will include a different computation to calculate the amount of attenuation

within the hail segment. For now, the parameters to distinguish a hail segment are place holders for future development.

The following section allows users to specify parameter quantities per site. This is especially useful in the case of IRIS/Reingest receiving data files from multiple radar sites. To use simply state `[SITENAME]` as a header line and repeat any of the name-value pairs from within this configuration file which need to be unique to that site.

```
# ========================================================
# per-site sections can be used to add per-site variation
# of the parameters set in the general section.
[wes-grad]
alpha_h = 0.1
```

CHAPTER 4
HYDROCLASS

4.1 Overview

HydroClass is a collection of echo identification algorithms in the Interactive Radar Information System (IRIS) and the Digital Signal Processor (RVP900).

Conventional weather radars transmit and receive only a single polarization (usually horizontal). State-of-the-art dual polarization radars transmit and receive both horizontal and vertical polarization. The HydroClass software makes optimal use of these dual-channel measurements to deduce the types of scatterers present in the atmosphere, such as rain, hail, snow, graupel, and even non-meteorological targets such as insects, chaff, and sea clutter. In addition to the improvements in precipitation estimation that are achieved with dual-polarization radar, the ability to deduce and map the types of scatterers greatly enhances the power of dual-polarization radar for applications such as:

- Hail detection
- Lightning hazard potential forecasting
- Detection of convection and stratiform rain
- Highway snow removal
- Airport terminal operation
- Rain/snow line demarcation
- Melting height detection
- Weather modification for hail mitigation
- Insurance industry claims verification
- Military detection of chaff
- Data quality improvement by elimination of non-meteorological targets

- Improved precipitation forecasting
- Hydrological modeling

The HydroClass is a collection of algorithms which output an HCLASS data type of echo identification. It is formally analogous to other data types, such as DBZ, V, Z_{dr}, and so on. For example, the HCLASS data type may be used as input to IRIS products such as PPI, CAPPI, XSECT, and WARN. The HydroClass algorithms can be activated in various steps of the radar data processing, depending on the customer use case:

- **RVP900 HydroClass Processing—**Ray data are processed in the RVP900 and the bin level class assignments are output in real-time for each range bin. This approach is well-suited to applications where IRIS software is not active since the particle type can be displayed directly by the customer's display software.
- **RVP900 and IRIS/Radar Processing—**The identification of convection and stratiform rain is based on vertical radar echo structures, which show up in volumes of sweep data. These data are available in the processes of IRIS/Radar. The HydroClass classification of convective and stratiform rain is implemented as part of the IRIS/Radar real-time process.
- **IRIS HydroClass Processing—**Dual polarization data come from the RVP900 or from a third party processor, and are passed to an IRIS/Radar or an IRIS/Analysis system that is enabled with the HydroClass feature. The bin level algorithms produce HCLASS ingest data, which can be formatted as RAW products for archival or for rerouting in the network. HCLASS products are color-coded maps of precipitation classification categories, which can be output to and displayed on other IRIS workstations and IRIS/Web clients. Output of the GIF and other standard image formats is also supported. Archived RAW data can also be reprocessed into classes.

4.2 Description of Algorithms

4.2.1 Methodology

4.2.1.1 Gate contents and precipitation patterns

HydroClass echo identification and classification methods belong to two categories (in regards to their consideration of data radar echo):

- **Local (Bin-to-Bin) Classification**—These algorithms analyze the echo features in each volume element of observation (gate bin) to

estimate the *bulk contents of the volume element*. Typically, each gate bin is declared to consist of scatterers of a given type (a class of hydrometeors or of other scatterers). The algorithm may use features of data in the next surroundings of the gate; even broader information can be used as a specific constraint. In the first order, each bin assignment is deduced from the data features of that particular bin.

NOTE	Most of the HydroClass classification methods belong to the local bin-to-bin category.

- **Spatially Distributed Objects (Weather Pattern)**—The type precipitation is sometimes best to be described as a weather event (pattern), characterized as an extended object in the atmosphere. A weather event derives its features from large scale spatial (and temporal) distributions. Bin-to-bin consideration ignores these aspects of data. Precipitation patterns are evident to a trained radar meteorologist, and deterministic algorithms exist to identify specified event types. A classical example is the division into stratiform and convective precipitation. HydroClass supports these type of methodology, too.

4.2.2 Fuzzy Logic

As scrutiny, HydroClass methods use the fuzzy logic approach to polarimetric radar echo identification. Fuzzy logic extends from the classical reasoning of deterministic decision. In other words, fuzzy logic allows algorithms to cope with varied levels of consistency, instead of exclusive statements of "yes" and "no". The fuzzy logic outputs can be thought of as degrees of consistencies. In most published literature and in the Vaisala documentation, these output values are known as the rule strengths (RS).

The rule strengths should not be confused with probabilities because the fuzzy memberships and rules typically represent empirically defined sets, rather than statistical likelihoods of events—the latter relates to the frequentist approach to probability. The fuzzy logic is well suited for the purpose of identifying radar echoes, and subsequently hydrometeor types, because the radar moment signatures of different radar targets and hydrometeors are not mutually exclusive or unique.

The input variables to a fuzzy logic algorithm are first passed into membership functions (MBF). Membership functions quantify the consistency of the particular input variable within a specified output set. A value of 0 means the input variable is not included for a given set and a value of 1 describes a fully included variable. Values between 0 and 1

characterize partial or fuzzy membership. Membership functions can be one-dimensional or multi-dimensional. For example, consider the use of reflectivity (Z_H) for characterizing hail. It is known empirically, that for lower reflectivity values below ~45 dBZ, the presence of significant hail is unlikely. Using this knowledge, a one-dimensional membership function can be designed for this simple case, depicted in Figure 11.

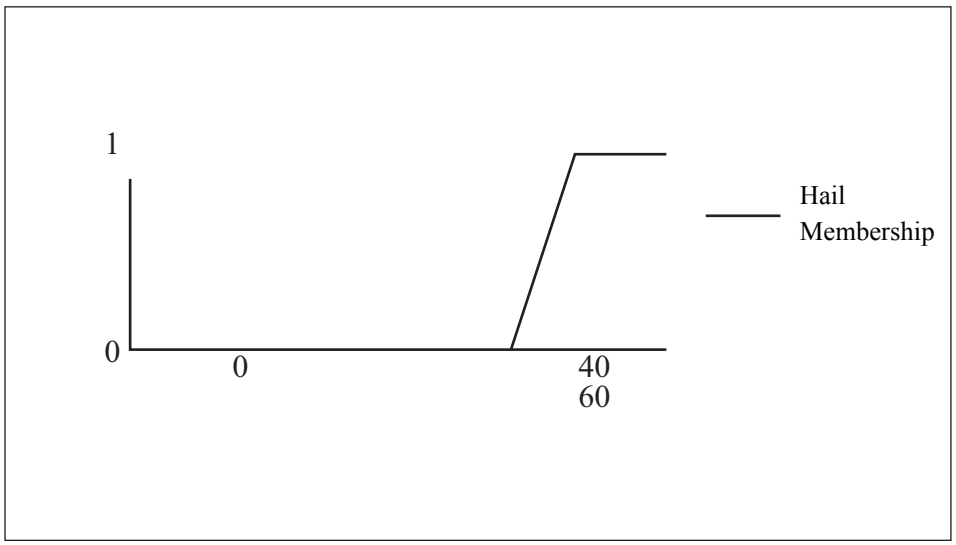


Figure 11 Membership Function in a Generic Fuzzy Logic Algorithm Schematic View

Using this membership function alone, a reflectivity at range bins of lower values implies a rule strength of 0 for hail, in other words, the observation is inconsistent with hail. With multiple radar moments available as input, the rule strength for hail is an appropriate functional combination of all the membership functions using the inputs. Multiple outputs can be evaluated by constructing rule strengths to each outcome hypothesis. A deterministic outcome is obtained by comparing the rule strengths, at each range bin.

4.2.3 HydroClass: A Synthesis of Public Methods

The HydroClass library is a collection of public echo identification methods resulting from major distinct developments in the weather radar community. Three main sources of origin can be named:

- **Meteo**—HydroClass results of fuzzy classification for the S- and C-band polarimetric weather radar echoes, evolved and field tested by the Colorado State University (CSU), Fort Collins, USA, and by their research partners.[3,18,19]

- **Precip**—HydroClass results of echo and hydrometeor classifiers for the polarimetric version of the S-band WSR-88D, developed by the National Severe Storms Laboratory (NSSL), and evaluated for operational use in the Joint Polarimetric Experiment (JPOLE) and subsequent transition of the NEXRAD radar network into polarimetry.[22,28,30]
- **Cell**—HydroClass results of analysis of the vertical structure precipitation echo, historical developments for single polarization radars[33]; originally the method intended for detecting hail events, here applied to detecting general cases of convective precipitation.

The implementation HydroClass is essentially that of a synthesis, which respects the integrity of each algorithm. Their best aspects are materialized through their sequential arrangement in which:

- **Preclassifier** (adapted from JPOLE)—An echo classifier algorithm serves as a prior quality control to:
  - **MeteoClassifiers** (of CSU origin) and **PrecipClassifier** (of NSSL origin)—Alternative methods of gate-to-gate hydrometeor classification
  - **CellClassifier**—A weather pattern classifier of convection and stratiform rain used as a further attribute to the previous classifications

Full tunability of the named algorithms is a further key feature of HydroClass, this is, the software distribution includes a comprehensive set of parameters, which specify each algorithm. The default settings reflect the original algorithms, as applicable ariants of parameter settings are provided for polarimetric radars operating at C-band and S-band. If desired, users may customize the settings, based on their local knowledge, experiences, and application needs.

The IRIS/RDA implementation of these distinct classifiers is described in Section 4.3 Classifiers on page 63. The usage information, that is, the typical first usages of the algorithms, the derived products, and the configurability for advanced users are described in Section 4.4 User Instructions on page 78.

4.2.4 Classification data

HydroClass echo identification results are reported as the HCLASS data type, which is managed analogously to other IRIS/RDA data types (for example, binary formatted RDA outputs and the IRIS archive data RAW).

The HCLASS data type is special in regards the property of enumeration—the data values do not represent a continuum of measurements. For

example, their arithmetic operations have no meaning, which calls for special approaches for the operations of interpolation and averaging which are often needed when the gate data are projected into IRIS products.

Furthermore, the HCLASS data consist of several bit segments of enumerated data, filled by the multiple HydroClass classification methods. A set of algorithms can be configured to run simultaneously, and provide complementary echo identification information, each bin. Digesting and presenting such richness of information are challenges of selection, while they also open opportunities of logical merging of the classification results for best overall interpretation of data. IRIS product software are equipped with extensions that allow computing IRIS products of HCLASS, as conveniently as with other data types.

4.2.4.1 The Nearest Neighbor in Interpolations

Interpolation of data is needed in IRIS products of cross-section (XSECT). The HCLASS data are interpolated with a rule of the nearest neighbor.

In its simplest form, the HCLASS value in any point X along a straight line between the points *A* and *B*, in which the HCLASS data values are *HCLASS(A)* and *HCLASS(B)*, is obtained in the parametrization:

$$\vec{X}(t) = \overrightarrow{Ax}(t) + \vec{B}x(1-t), \text{when } t \in [0,1]$$

$$HCLASS(\vec{X}) = \begin{cases} HCLASS(\vec{A}), & \text{when } t \le 0.5 \\ HCLASS(\vec{B}), & \text{when } t > 0.5 \end{cases}$$

This approach generalizes to interpolating in the N-dimensional data grid:

$$HCLASS(\vec{X}) = HCLASS(\overrightarrow{A_i})$$

where $\vec{A}_i$ has the shortest distance to the data point $\vec{X}$

$$r_i = \left|\vec{X} - \vec{A}_i\right| = min\{r_j = \left|\vec{X} - \vec{A}_j\right|, \text{ when } A_j \in \textit{proximate data}\}$$

4.2.4.2 The Majority Rule in Averaging

Averaging of data is needed when projecting high resolution data into IRIS products. The HCLASS data are averaged with the rule of majority consensus, which selects the HCLASS value that appears most frequently in the set of data being averaged. A value if selected randomly among equally frequent values.

4.3 Classifiers

4.3.1 Preclassifier

The HydroClass "Preclassifier" is an implementation of the JPOLE algorithm, following closely the reference.[28] The JPOLE algorithm uses the fuzzy logic approach and is the first processing step for HydroClass. It uses polarimetric variables to classify the data in the range bins as:

- GC/AP (Ground Clutter/Anomalous Propagation)
- BIO or biological targets
- METEO or hydrometeor scatterers

Five radar variables are used as input to the JPOLE algorithm. These five variables are horizontal reflectivity Z_H, differential reflectivity Z_{dr}, cross-correlation coefficient RhoHV, a texture parameter of the Z_H field TX(Z_H), and a texture parameter of differential phase TX(PhiDP). To obtain TX(Z_H), the Z_H data is averaged over a running average window and then the smoothed estimates of Z_H are subtracted from the original values, in other words the standard difference, effectively. A similar procedure is used for computing TX(PhiDP). The length of the running window is configurable, the default being 5 bins for TX(Z_H) and 10 bins for TX(PhiDP).

Trapezoidal membership functions are used for the JPOLE algorithm (see Figure 12 on page 64).

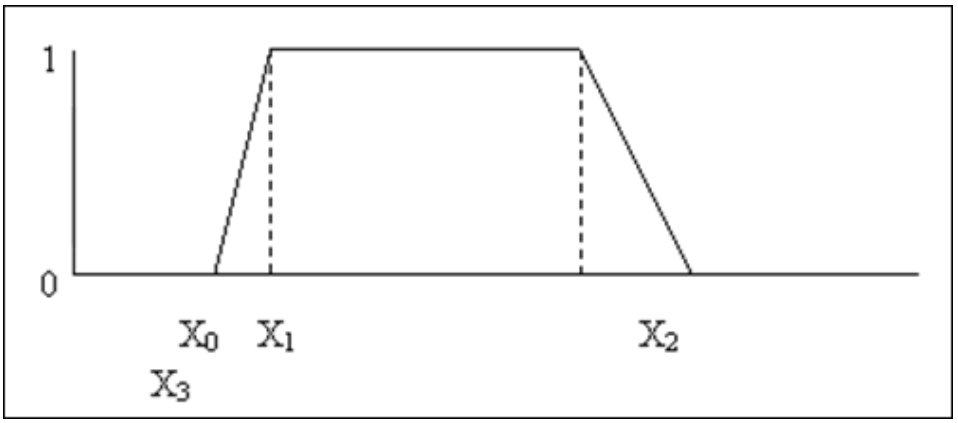


Figure 12 Example of Trapetzoid Function used as a Parametrization of the Membership Function in the JPOLE Algorithm

The displays of the default JPOLE membership functions are in Figure 13 on page 64 through Figure 17 on page 66. The computation of additive rule strengths and class identification with maximum aggregation are in the literature reference.

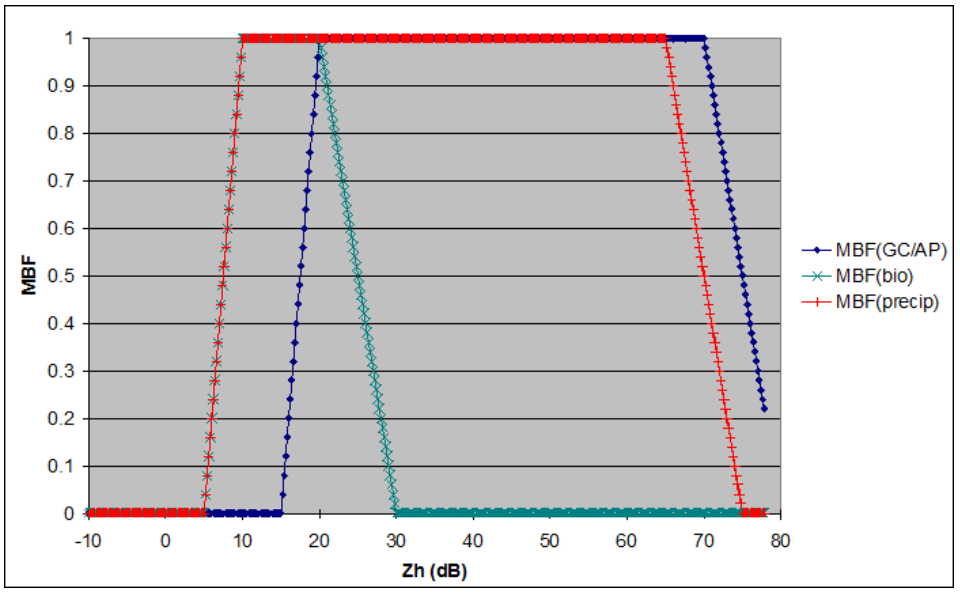


Figure 13 Default Settings of the JPOLE Membership Functions for Z_H

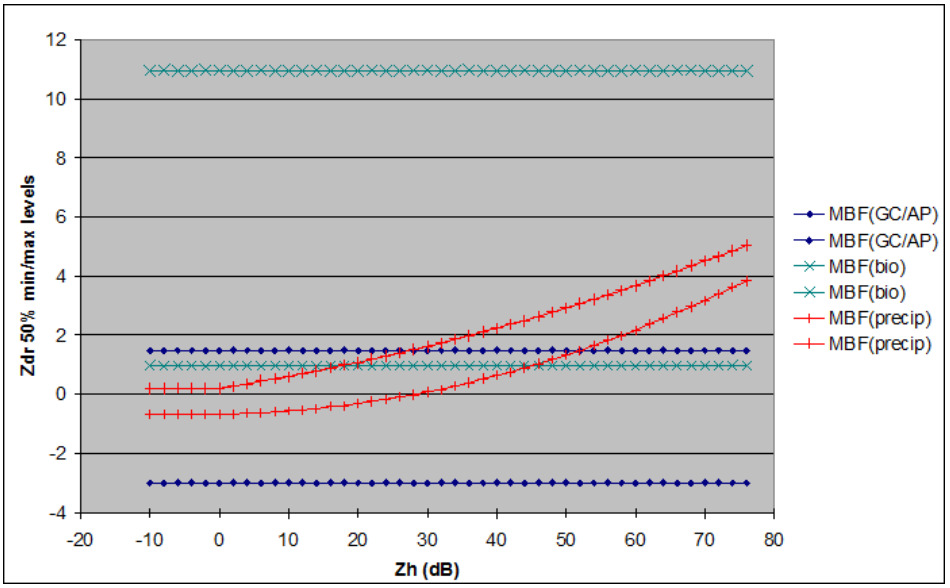


Figure 14 Default Settings of the JPOLE 2D Membership Functions for Z_{dr}

The two-dimensional (2D) membership functions are expressed as 0.5 contour levels as function of reflectivity. Class compatibilities are greater than 0.5 when Z_{dr} values fall between the regions depicted by the contours.

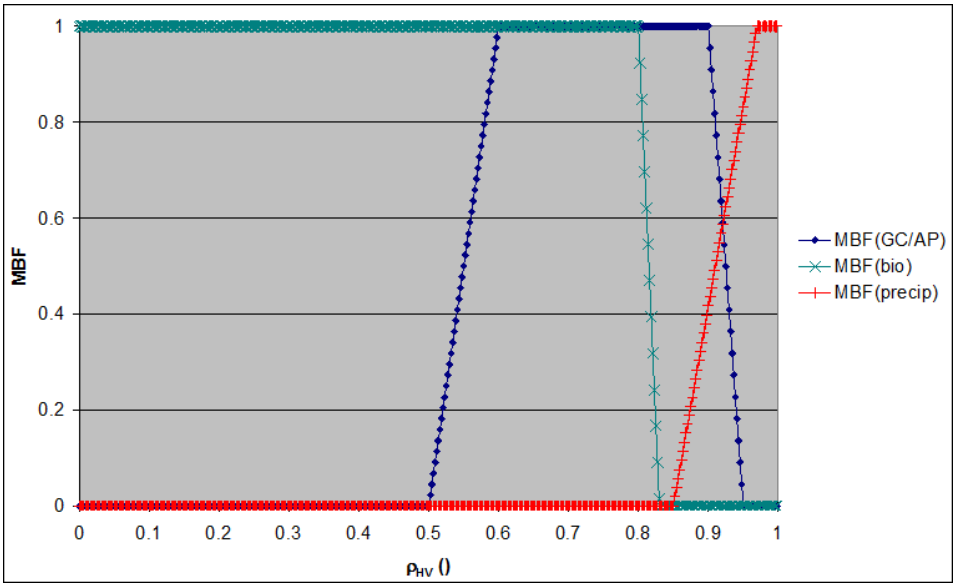


Figure 15 Default Settings of the JPOLE Membership Functions for RhoHV

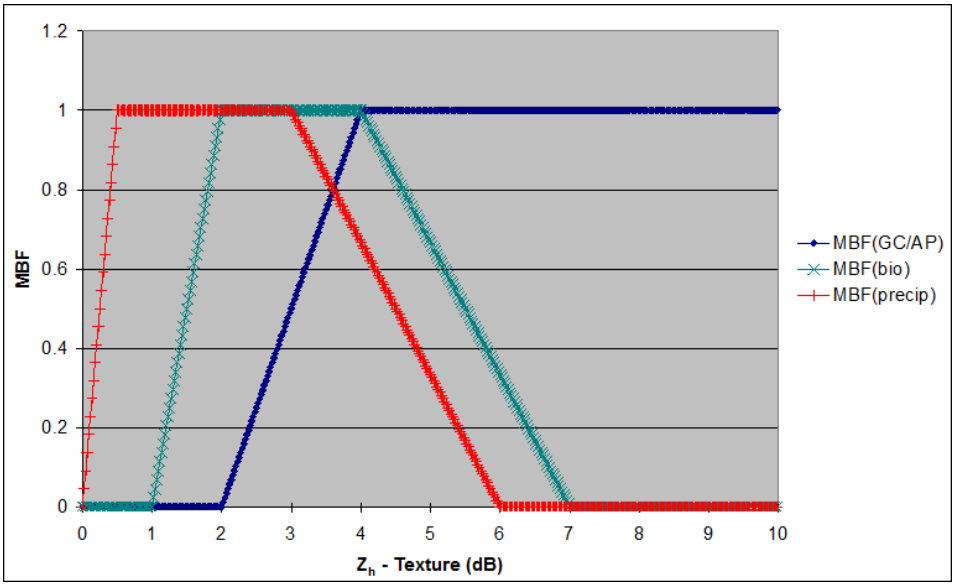


Figure 16 Default Settings of the JPOLE Membership Functions for TX(Z_H)

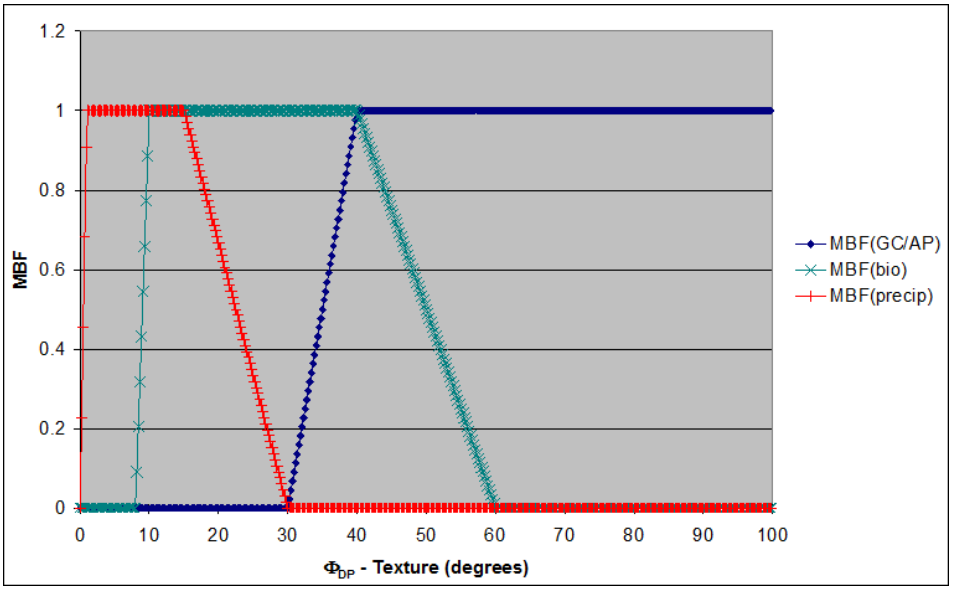


Figure 17 Default Settings of the JPOLE Membership Functions for TX(PhiDP)

4.3.2 MeteoClassifiers

The default classification system of warm and cold season hydrometeors is based on the fuzzy method approach.[19] The implementation follows closely the updated version[18], which is the recommended general

reference. The main aspects are briefly explained and are followed by the features specific to the present implementation.

The method described in the main reference[18] was developed using radar measurements at the CSU-CHILL Facility (for technical description, see http://lab.chill.colostate.edu/chill-technical.html). CHU-CHILL facility is a S-band, Doppler radar with full polarization agility and diversity at Colorado State University, U.S. The referenced classification system represents state-of-the-art knowledge and it has been verified by comparing the CSU-CHILL measurements with the in-situ airborne observations made with instruments such as 2D cloud particle measurement probe, high volume particle sampler (HVPS) and hail spectrometer.[18]

In the fuzzy method approach, the signatures of specified hydrometeor classes are quantified as a set of membership functions (MBF), which take the measured dual-polarization parameters obtained at each bin as input. The strength of each hydrometeor class is then expressed as the outcome (rule strength) of an inference function which takes the MBF values as input. The membership functions and the inference rule strength function formalize the meteorological interpretation encoded in the classification method.

The membership functions (MBF) of the observable x are parametrized in the CSU method as beta functions:

$$MBF(x;m,a,b) = \frac{1}{1 + (\frac{x-m}{a})^{2b}}$$

in which x stands for reflectivity (Z_H), differential reflectivity (Z_{dr}), specific differential phase (K_{dp}), cross-correlation coefficient (HV), or observation altitude (h). The expression specifies the one dimensional case (1D), while evolutions of MBF(Z_{dr}) and MBF(K_{dp}) at varied reflectivities are detailed further by parametrizing y=y(Z_H) where y=m, a, or b. Such parameterization allow extending the 1D MBFs into 2D.

The membership function of the radar observation volume altitude has dependence on the melting layer height (ML), which is thus also used as input. The membership function of altitude can also be formulated as a 2D MBF, in which parameters m and a are dependent on ML.

In IRIS/RDA, the melting layer height is defined to coincide with the 0 oC isotherm, and related expressions in literature algorithms are converted to match the IRIS/RDA definition. ML can be fed in from an external source, or it is acquired from the archived RAW file headers, or it can specified explicitly in the HydroClass own configuration.

The classification results are presented by labeling each bin with the hydrometeor class that is most compatible with the observations, i.e. by choosing the class of highest rule strength. The rule strengths are computed using the equations of the literature reference using the quoted four radar variables and the altitude as input. Threshold parameters associated to each of the rule strengths are used to specify bins for which the class is ambiguous, for example, non-meteorological targets.

4.3.2.1 Specific Features of IRIS/RDA

4.3.2.1.1 Hydrometeor Classes

The reference method characterizes membership functions for eleven classes, while the results were interpreted in the scheme of nine final output classes. Simplicity was considered advantageous from view of typical applications, and five final classes have been defined as the basis for classification in the present implementation:

- Rain
- Wet Snow
- Dry Snow
- Graupel
- Hail

In order to characterize the precipitation phenomena in realistic fashion, a commonly emerging mixture 'rain and hail' has been modelled with specific MBFs, in addition. Outcomes in this intermediate class are merged into the final output class 'hail'.

Membership functions (MBFs) and rule strengths (RS) have been re-optimized using the new class basis in a dedicated effort, see Section 4.3.2.1.4 Customization to Simultaneous Transmission, Simultaneous Receive Mode on page 69.

4.3.2.1.2 Prior Quality Considerations

HydroClass uses standard IRIS/RDA measurands as input, and their quality thresholds propagate into HydroClass quality thresholds.

In addition, the Preclassifier of non-meteorological versus precipitation echoes (JPOLE algorithm) is used as a prior quality factor to the hydrometeor classifier (CSU algorithm) by passing in the bins classified as meteorological echoes by JPOLE, only.

4.3.2.1.3 Variants for Warm and Cold Seasons

The melting layer height (ML) is one of the inputs for the set of MBFs and RSs. The main mode of the classification system is dedicated to the summer like situation with a positive value of ML. The organization of MBF and RS parameters follow this case of the main reference, and their default values are described in Section 4.2.3 HydroClass: A Synthesis of Public Methods on page 60.

In cold season, any large scale melting layer is absent. The problem of classifying precipitation is better to be started from a different a priori expectation. The task is still nontrivial, as liquid forms of precipitation are not excluded (warm fronts, freezing rain). Significant convection can occur in cold season, too, producing heavy solid hydrometeors.

In order to describe these circumstances, a variant of the method has been constructed for the case ML less than zero, signalling precipitation in cold season. In such climatology, modified MBFs are computed for altitude, and the ML does not enter in MBF calculation in this case. Explicitly, liquid form of precipitation (rain) is allowed up to a fixed altitude of 5 km (MSL), wet snow is allowed to a characteristic altitude of winter storms, assumed constant of 2.5 km, at present. Other types of precipitation are indifferent with respect to altitude.

Other MBFs are unchanged from their warm season settings. Subsequently, modified weights in rule strengths calculation are used, see Table [RULE STRENGTHS].

4.3.2.1.4 Customization to Simultaneous Transmission, Simultaneous Receive Mode

The main literature reference method parameters have been tuned, using data obtained in the alternating horizontal/vertical polarization mode, in which both the co-polar terms are observables, that is, H received from H transmitted (HH) and V received from V transmitted (VV). The cross-polar terms, that is, V received from H transmitted (VH) and the (HV) signals are observables as well. As a consequence, the linear depolarization ratio (LDR) can be promptly used as an input observable, paying special attention to cases of LDR data at low signal to noise ratio.

Operation in the mode of simultaneous transmission and simultaneous receive, known as the 'hybrid' mode, excludes availability of LDR at pulse-to-pulse basis. Adding to this, characteristics of other polarimetric observables depend on the processing mode, in presence of significant propagation effects in particular.

These two general reasons suggest for method customizations specific to polarimetric processing modes, and has in part motivated the procedure.

4.3.2.1.5 Fuzzy Parameters Optimized for C-Band

The new features imply reprocessing of the algorithm parameters in the new class basis, in a variety of climates, and in the hybrid polarimetric processing mode. Adding these customizations, the understanding of data and phenomena at S-band was extended to C-band.

The existing CSU parameter optimization procedure was applied to multi-season data samples obtained at the Vaisala polarimetric weather radar prototype at University of Helsinki.[25] The data samples represented a variety of weather cases (see Figure 18 on page 71 through Figure 20 on page 73) as follows:

- Summer convection
- Winter storm
- Stratiform frontal precipitation

As result, new membership functions and rule strength coefficients were obtained that are used as default settings for C-band weather radars operating in the 'hybrid' mode. The distributions of the membership functions are visualized in Figure 21 on page 74 through Figure 25 on page 76. The numerical values and more details of the membership function parameters are listed in Table 7 on page 123 through Table 10 on page 126.

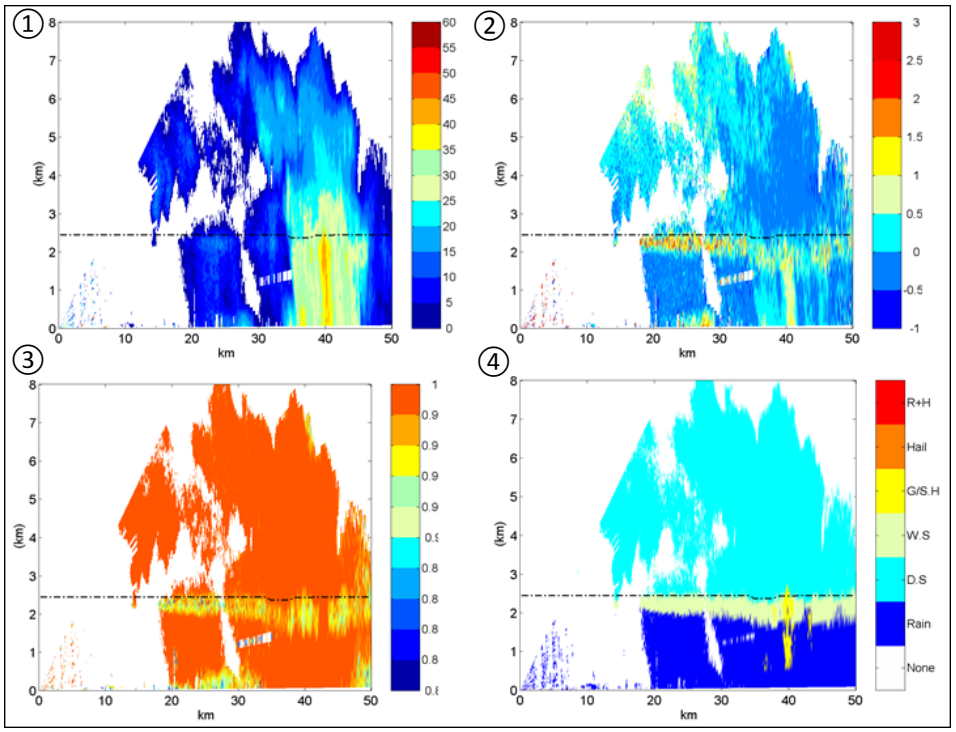


Figure 18 Selected Observables in the RHI Data Sample of a Summer Convection used for Optimizing the Fuzzy Parameters

1 = Reflectivity Z (dB)
2 = Differential reflectivity Z_{dr} (dB)
3 = Cross-correlation coefficient RhoHV(0)
4 = Classification result ('D.S': dry snow, 'W.S': wet snow, 'G/S.H' graupel, small hail)

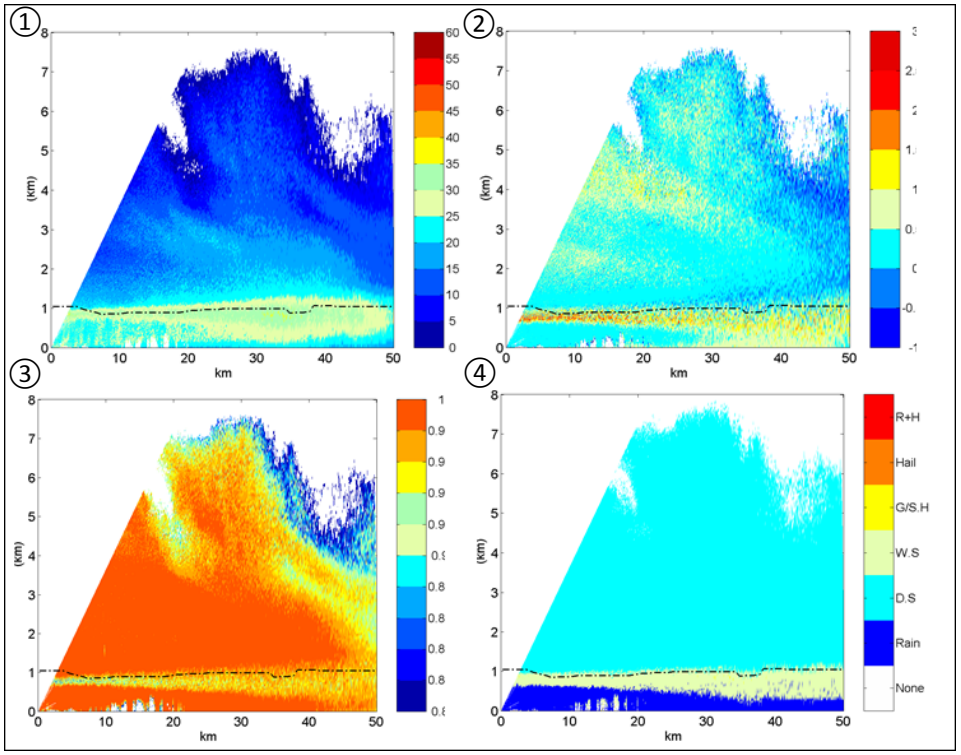


Figure 19 **Selected Observables in the RHI Data Sample of a Stratiform Precipitation (Warm Front) used for Optimizing the Fuzzy Parameters**

1 = Reflectivity Z (dB)
2 = Differential reflectivity Z_{dr} (dB)
3 = Cross-correlation coefficient RhoHV(0)
4 = Classification result ('D.S': dry snow, 'W.S': wet snow)

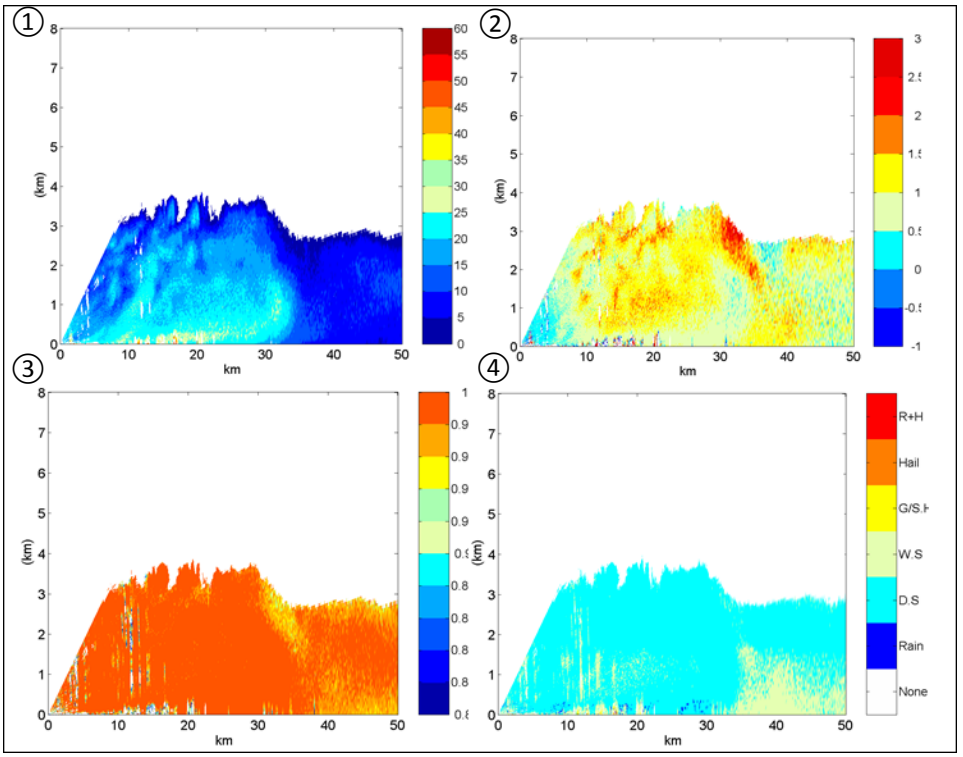


Figure 20 Selected Observables in the RHI Data Sample of a Winter Precipitation Event used for Optimizing the Fuzzy Parameters

1 = Reflectivity Z_H (dB)
2 = Differential reflectivity Z_{dr} (dB)
3 = Cross-correlation coefficient RhoHV(0)
4 = Classification result ('D.S': dry snow, 'W.S': wet snow)

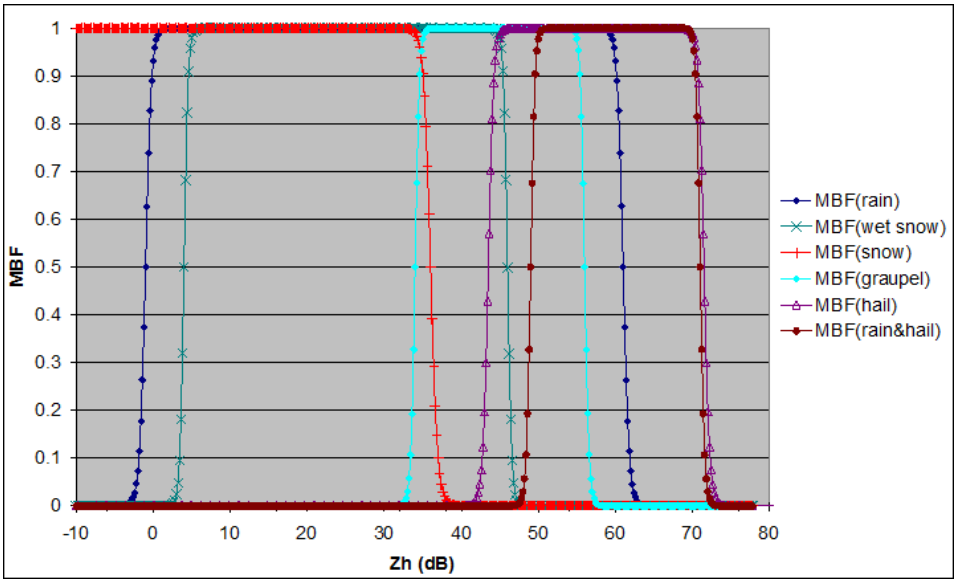


Figure 21 **C-band Default Settings of the Membership Functions for Reflectivity (Z_H)**

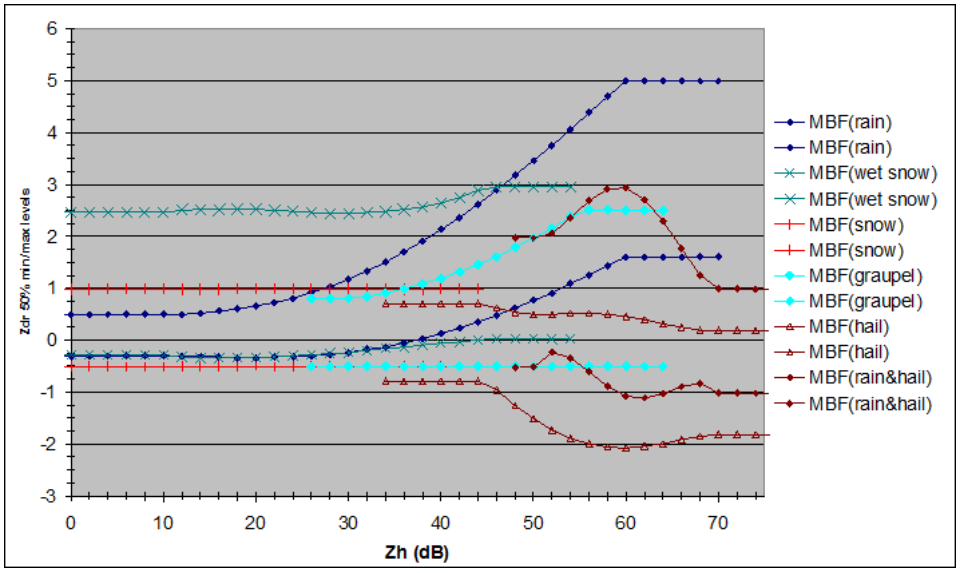


Figure 22 **2D Membership Functions for Differential Reflectivity (Z_{dr}) (Expressed as 0.5 Compatibility Contours as Function of Reflectivity)**

NOTE	Class compatibilities are greater than 0.5 when Z_{dr} values fall between the regions depicted by the contours.

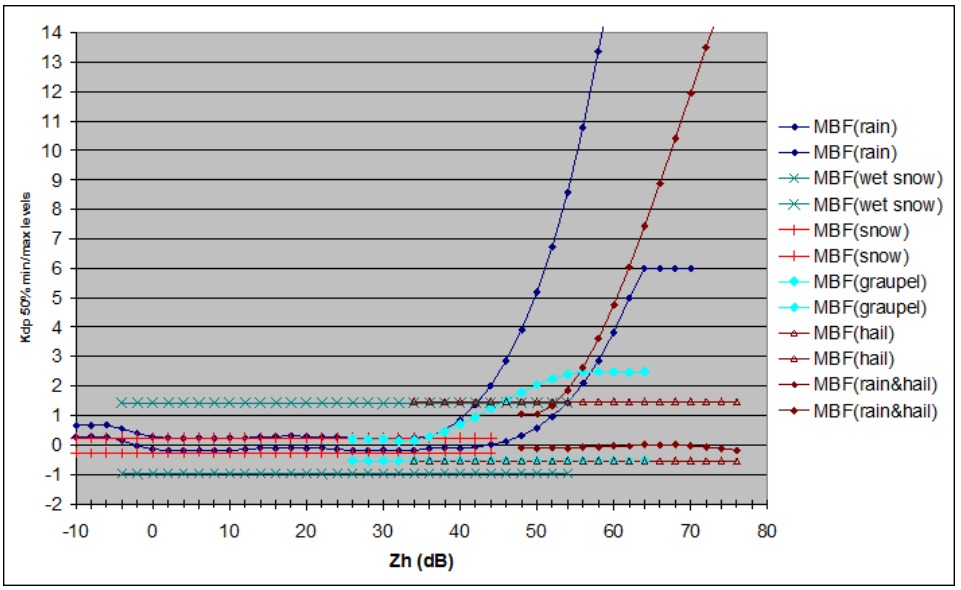


Figure 23 **2D Membership Functions for Specific Differential Phase (K_{dp}) (Expressed as 0.5 Compatibility Contours as Function of Reflectivity)**

NOTE	Class compatibilities are greater than 0.5 when K_{dp} values fall between the regions depicted by the contours.

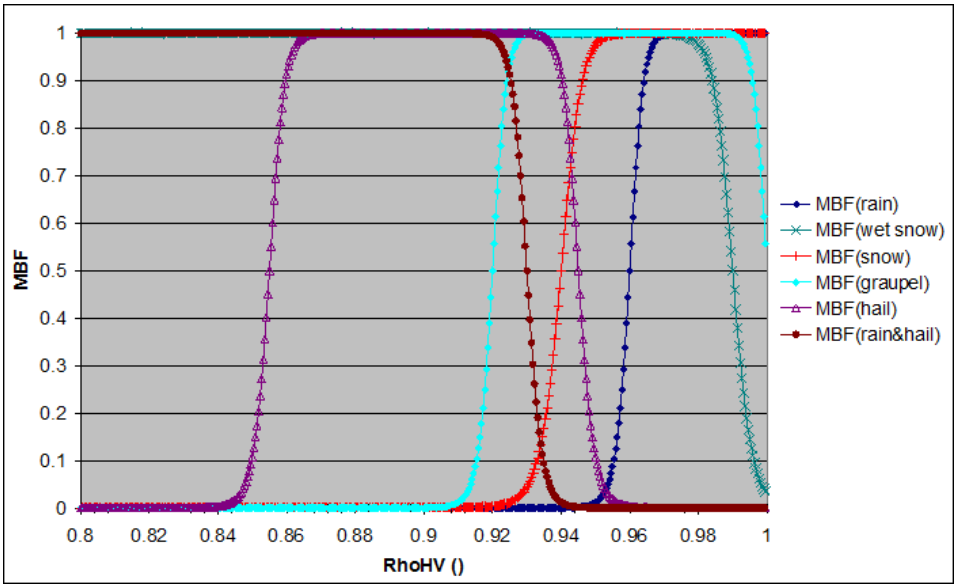


Figure 24 **Membership Functions of the Cross-Correlation Coefficient (RhoHV)**

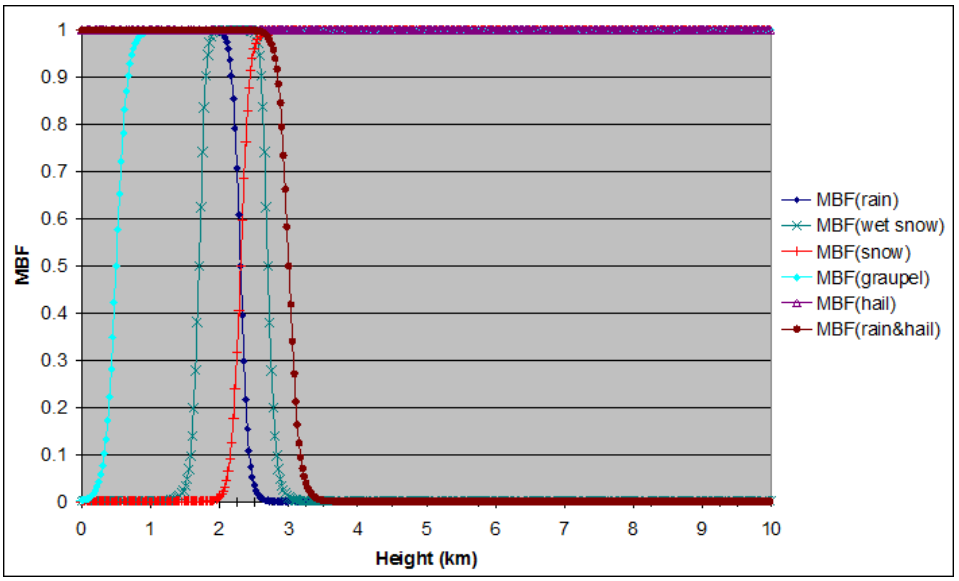


Figure 25 Membership Functions for Altitude (h) (Depicted for Melting Layer Height at 2.5 km)

NOTE	The contours shift linearly as function of the melting layer height. All heights are expressed with respect to mean sea level.

The optimized expressions for rule strengths (RS) are analogous to the literature reference with the following new default coefficients:

$$RS_i\,(\mathrm{ML}>0) = \mathrm{MBF}_i\,(Z_H) \times \mathrm{MBF}_i\,(\mathrm{h}) \times [\mathrm{MBF}_i\,(Z_{dr}) + 0.5 \times \mathrm{MBF}_i\,(K_{dp}) + 0.5 \times \mathrm{MBF}_i\,(\mathrm{RhoHV})]/2$$

$$RS_i\,(\mathrm{ML}<0) = \mathrm{MBF}_i\,(Z_H) \times [0.7 \times \mathrm{MBF}_i\,(Z_{dr}) + 0.3 \times \mathrm{MBF}_i\,(K_{dp}) + \mathrm{MBF}_i\,(\mathrm{h}) + \mathrm{MBF}_i\,(\mathrm{RhoHV})]/3$$

In these expressions $RS_i(ML>0)$ and $RS_i(ML<0)$ refer to rule strengths of a class i for warm and cold season cases, respectively. $MBF_i(x)$ refers to membership functions of variable x for the class *i*.

4.3.2.1.6 Features Not Implemented

The melting layer height is one of the variables used in the hydrometeor classification fuzzy method. The main literature reference introduces a radar based estimate for a variable melting layer heights in convective storms. In essence, the estimate is based on vertical gradient profiles of differential reflectivity. Other approaches have been proposed for stratiform cases and they also use vertical profiles and/or gradients in reflectivity, differential reflectivity, and/or polarization cross-correlation coefficient, as prime signatures of melting layer.

Observing vertical profiles and/or gradients, unambiguously, requires dedicated scan strategies such as RHI scans, or alternatively several horizontal sweeps at varied elevations in a well designed volume scan. Such requirements contradict a HydroClass design goal of independence on scan strategy. Subsequently, the melting layer height is left as an open method in HydroClass, and the information is fed in as an external input variable. Methods are available for feeding in the values at momentary basis, see Section 4.4.2 Running HydroClass on page 78.

4.3.3 PrecipClassifier

The PrecipClassifier implementation closely follows the fuzzy method of Version 2: Warm season[28], which includes the default fuzzy parameter values. For the classes of rain, the default values of the membership functions for textures of reflectivity and differential phase take the common values of precipitation of the preclassifier.

This classifier has the same default settings for S-band and C-band radars. The optimal parameters of membership functions for differential reflectivity are somewhat different for C-band radars.

4.3.4 CellClassifier

The PrecipClassifier implementation is based on the historical signature of hail and donvection[33]: high level of reflectivity at a specified height above the melting layer can be used as a signature of hail (and convection in general).

In this implementation, the general signature of convection is fuzzified as a rule strength, which is a product of two membership functions (trapezoids):

- Minimal reflectivity required for convection
- Minimum altitude, with respect to the 0°C isotherm

For simplicity of configuration, the shapes of the trapezoids are fixed. The full rule strength is reached when the reflectivity is 5 dB more than the minimal threshold, and when the height of those reflectivity levels reaches 1 km above the minimum altitude.

4.4 User Instructions

4.4.1 Installing, Licensing, and Activating in Setup

HydroClass is an integral part of IRIS and RDA releases, and no specific actions are necessary in standard IRIS and RDA installs.

A valid HydroClass license code is needed to use HydroClass. The code is activated and encrypted in the setup utility graphical user interface (GUI) field "License". Two license codes are provided:

- HydroClass in Reingest–Allows activating the classifier methods of Preclassifier, MeteoClassifiers and PrecipClassifier in IRIS/Analysis, i.e. in post processing. This license code also enables processing real-time data with the CellClassifier in IRIS/Radar
- RVP900 HydroClass Generation, which allows activating the classification methods of Preclassifier, MeteoClassifiers and PrecipClassifier in RDA real-time ray processing.

HydroClass in enabled IRIS/reingest using the menu "HClass: Hydrometeor Classification in ReINGEST" in the setup block "Ingest". The installed and activated HydroClass runs promptly on appropriate input data and computes results using the current configuration, as soon as appropriate input data arrive or a task with HClass data type is running.

Similarly, one enables HydroClass in RDA using the menu "Optional Data Parameters" in the setup block "RVP Radar Video Processor". HydroClass algorithms process data if the HCLASS data type is configured to be reported.

4.4.2 Running HydroClass

HydroClass is implemented in the major mode of PPP in RVP900. Alternatively, HydroClass is able to process any appropriate polarimetric radar data converted into IRIS RAW format. By construction, HydroClass sets minimal requirements on task parameters, while driving certain task settings to extremes may compromise meteorological performance.

Minimally, HydroClass requirements for input data (effectively, the RVP900 run-time parameters, IRIS task parameters, or input RAW characteristics) are:

- Polarization mode: H+V

- Major mode PPP
- Active data types: DBZ, Z_{dr}, K_{dp}, PhiDP and RhoHV
- in case HydroClass is configured to use the external melting layer height (ML), a valid ML value in RVP900, or in the RAW header structure

For uniform meteorological performance, it is recommended:

- To constrain the observation distances at the scale of 100 km; far beyond 100 km, the radar observation volume is expected to contain a lot of structures, in vertical direction in particular, and unambiguous classification of hydrometeors is not well justified
- To prefer reasonable samplings (32 or more) and modest antenna speeds
- To pay balanced attention to all aspects of radar data quality (pedestal alignment, calibration of reflectivity, H- to V-channel calibration balance, noise samplings, radome quality)

4.4.2.1 Operating RDA/HydroClass with IRIS/Radar

IRIS/Radar is a typical method of operating radar equipped with RVP900 signal processing. The most convenient mode of using HydroClass at these radars is to activate it at RVP900 and run it within standard IRIS tasks configured to 'H+V' mode of polarization and to the RVP900 major mode of pulse-pair-processing (PPP).

In a given IRIS task, HydroClass is switched on by enabling the HClass data type in the data pop-up in the 'Processor Configuration' block. In addition, one must enable the required data types. Given a valid HydroClass configuration file, and valid melting layer height input (if specified that way in HydroClass configuration) no other action is necessary, and HydroClass will generate the HClass data type in the ingest data stream, analogously to other data types.

4.4.2.2 General Method of Operating HydroClass in RVP900

HydroClass is built in as an output data type in RVP900 processing, and subsequently it is available to the use case in which the RVP900 is operated with the user radar software. In this case, the feature is available through activation of the polarimetric HClass data type, assuming a standard HydroClass configuration.

4.4.2.3 HydroClass in IRIS

In case HydroClass is enabled in IRIS/reingest, it considers each RAW file routed to reingest and appends the HClass data type in the ingest collection whenever the RAW data input is suitable to HydroClass.

The input radar moments must be recorded in 16-bit data format. The algorithm will not create output from 8-bit RAW data, directly. However, it is relatively straightforward to convert such data sets into 16-bit presentation in IRIS (intermediate RAW products) and use them for obtaining the HClass data type.

4.4.3 Configuration

HydroClass application is equipped with a configuration interface that is common to HydroClass in RVP900 and in IRIS. The configuration data are in the configuration files:

```
$IRIS_ROOT/config/dpolapp_C-band.conf
```

```
$IRIS_ROOT/config/dpolapp_S-band.conf
```

for C-band radars and for S-band radars, respectively. The files are equivalent in structure, but with default parameter sets that match with these widely used radar types.

It is important to note, that the HydroClass functionality is ready to use after the software install, radar calibration and start of radar operations. The installed configuration meets the first needs of evaluation and general use. HydroClass automatically selects the correct configuration file to match with the radar frequency.

Site specific customizations are usually needed in the advanced use cases i.e. for optimal performance, only. The optimization is best made with feed back and needs of the end-user applications. Substantial amount of observational data and theor careful consideration are needed for consistent customization of HydroClass.

The configurable features are:

- "Choices of activation, and relations of uses of the distinct classification algorithms "PreClassifier", "MeteoClassifiers", "PrecipClassifier" and "CellClassifier"
- Complete sets of fuzzy parameters of these classifier
- Miscellaneous settings, such as origin of melting layer height, data corrections, and quality considerations of the results

HydroClass polls for possible updates of the configuration file and reloads upon file change. The update takes place for each RAW file in IRIS/ reingest - restart of IRIS is not required. Consecutive RAW files are allowed to originate from radars that operate at different frequencies. HydroClass configuration is checked for each period of PPP major mode in RVP900 run. It is noteworthy at RVP900 radars operated by IRIS that a new task does not trigger a HydroClass configuration update, but a switch into another major mode, for example, FFT, and back to PPP are required. The updates initially trigger an IRIS/RDA message, which are lated muted off, in order not to spam log files.

4.4.3.1 HydroClass Configuration File

A parameter line of the HydroClass configuration file has one of the following format:

```
"dpolapp[.SubSpecifier].[ParameterGroup].ParameterItem =
1234"
```

where:

- `SubSpecifier` parts are optional, typically referring to distinct classifier methods "Preclassifier", "MeteoClassifiers" "PrecipClassifier" and "CellClassifier"
- `ParameterGroup` refers to (a vector of) parameter group, such as membership functions `MBFinputs`
- `ParameterItem` refers to (a vector or a matrix of) parameters

Each `ParameterGroup` is typically preceded by comment lines, starting with a letter "#". The number of subsequent parameter lines and the dimensions of the parameter vectors and matricer reflect structure of the algorithm. The structure of the algorithm is briefly explained in the paragraph of comments at start of each classifier method, indicated by a comment line:

"#==="

4.4.3.1.1 Main Structure of the Configuration File

Current configuration files consist of the main blocks listed below. In typical first usages, the parameter settings in the block 2) need to be considered, with a few additional key in the group 4):

- As soon as a consistent source of melting layer height is determined at the radar, it is recommended to change the setting:

  ```
  dpolapp.Quality.Aux[0] =  1
  ```

- to

  ```
  dpolapp.Quality.Aux[0] =  0
  ```

Modifying parameters in the subsequent groups 5) to 8) should be seen as advanced usage.

1. # === RDA/IRIS HydroClass configuration file (C-band) ==
   title block and the configuration nick name
2. # ==
   # Main algorithm selection menu
   selectors for activating distinct algorithms (blocks)
3. #===
   # RDA/IRIS major version
   current IRIS/RDA version and HydroClass revision
4. #===
   # Detailed data quality configurations:
   miscellaneous quality settings in the block *Quality*
5. #===
   # Detailed configuration of the Preclassifier
   output settings and fuzzy parameters of the *Preclassifier*
6. #===
   # Detailed configuration of the MeteoClassifiers (warm and cold season)
   output settings and fuzzy parameters of the *MeteoClassifiers*
7. #===
   # Detailed configuration of the PrecipClassifier
   output settings and fuzzy parameters of the *PrecipClassifier*
8. #===
   # Detailed configuration of the CellClassifier

   fuzzy parameters of the *CellClassifier*

4.4.3.1.2 Detailed Contents of the Configuration File

All distinct parameter lines are listed and described with their default settings for C-band radars.

Title and Configuration Nick Name

The title comment line is machine written:

```
dpolapp.sConfigurationName ="dpolapp_C-band"
```

A descriptive name for the current configuration settings.

This parameter is the only one that is communicated as meta data in the header structures of IRIS/RDA data.The amount of HydroClass configurable parameters makes it impractical to communicate them all with the actual measurement data. The purpose of the nickname is to provide minimal means of indicating possible evolutions and modifications made of the HydroClass configuration.

The default string `"dpolapp_C-band"` suggests for using the default settings. It is recommended to change to a descriptive name as soon as algorithm parameters are modified.

Main Algorithm Selection Menu

```
dpolapp.bRunQuality = 1
```

activates the quality considerations in the main block "Quality".

```
dpolapp.usClassificationMethods = 222
```

activates a combination of the distinct echo classifier methods. Possible settings are:

- "1": Preclassifier, only (the JPOLE algorithm for meteo versus bio versus GC/AP)
- "2": Meteoclassifiers, with the Preclassifier as a quality mask
- "21": PrecipClassifier: the JPOLE warm algorithm, with the Preclassifier mask
- "22": PrecipClassifier and MeteoClassifiers, with the Preclassifier mask
- "201": CellClassifier for stratiform and convective rain, with the Preclassifier mask
- "202": Meteoclassifiers and CellClassifier, with the Preclassifier mask
- "221": Cellclassifier and PrecipClassifier, with the Preclassifier mask
- "222:" Cellclassifier, Precip classifier, Meteoclassifiers and CellClassifier, with the Preclassifier mask

Software Version Information

```
dpolapp.sVersion = "8.13"
```

machine written major IRIS/RDA version. No need to modify this.

```
dpolapp.sRevision = "8.13"
```

machine written minor IRIS/RDA version. No need to modify this.

```
dpolapp.sDpolappRevision = "15"
```

machine written HydroClass running version. No need to modify this.

Miscellaneous Quality Settings

```
dpolapp.Quality.Aux[0] =  1
```

Source of the melting layer height input:

- "1": HydroClass uses the melting layer height value suggested in the configuration file.
- "0": instructs HydroClass in IRIS to use the ML value in the input RAW file, and HydroClass in RDA to use the RVP900 process value of ML. The RVP900 value can be updated during operation (dynamically).
- In case dpolapp.Quality.Aux[0] =0 but the requested values (in RAW or RVP900) appear undefined, HydroClass reports about the missing input. HydroClass in IRIS moves over to using the value specified in the configuration file, while HydroClass in RDA switches off until a well defined value is provided.

```
dpolapp.Quality.Aux[1] =2300
```

An estimate of the current ML in meters above mean sea level (MSL). This estimate is used when dpolapp.Quality.Aux[0] =1, see previous field. It is recommended to move to using IRIS/RDA current values in continued use.

```
dpolapp.Quality.OtherInputs[0].dParams[0] =0.0
```

A posteriori HydroClass specific off-set to H-/V-channel balance (Z_{dr}). The off-set is intrinsic to HydroClass and does not affect the Z_{dr} output data type.

Use of non-zero value is discouraged in real-time applications (HydroClass in RVP900). The recommended IRIS/RDA approach is to use the tools described in *RVP900 User's Manual - Digital Receiver and Signal Processor, Appendix B.12 "Calibration Considerations"*.

General Settings of PreClassifier

```
dpolapp.Preclassifier.iClassifierID =-1
```

Are the PreClassifier results formatted as in the field of PRECIP_CLASSIFIER in the bits 3,4,5 from LSB=0 of the DB_HCLASS data type?

- "-1": NO, the legacy HydroClass format
- "1": YES, JPOLE classes of PRECIP_CLASSIFIER used.

```
dpolapp.Preclassifier.uiMaxNonMeteo = 1
```

This parameter is irrelevant in all cases, except:
`dpolapp.usClassificationMethods =1`

This parameter is a recipe of what to do with the bins that fail acceptance to any class of PreClassifier. It allows simplifying the class settings further through regrouping. One can re-group Preclassifier (un)identified results:

- "1": unidentified are 'uiNonMeteoID
- "2": in addition, 'GC/AP' are 'uiNonMeteoID'
- "3": in addition, 'BIO' are 'uiNonMeteoID'

```
dpolapp.Preclassifier.uiNonMeteoID =1s
```

What to do with the regrouped (un)identified results:

- "0": set as thresholded in IRIS/RDA
- "1": report as GC/AP
- "2": report as BIO
- "3": report as PRECIP

```
dpolapp.Preclassifier.dMinRS[0] =0
```

Minimum rule strength for accepting GC/AP.

```
dpolapp.Preclassifier.dMinRS[1] =0
```

Minimum rule strength for accepting BIO.

```
dpolapp.Preclassifier.dMinRS[2] =0
```

Minimum rule strength for accepting PRECIP.

```
dpolapp.Preclassifier.iAux[3] =2
```

The preferred class of the Polarimetric Meteo Index (PMI):

- "2": PMI gets high values when the PRECIP class is preferred
- "1": PMI gets high values when the BIO class is preferred
- "0": PMI gets high values when the GC/AP class is preferred

Settings for Polarimetric Meteo Index

```
dpolapp.Preclassifier.iAux[4] =1
```

PMI mathematical formula.

- "1": the 'score'

$$PMI = 1 - \frac{1}{1 + \left(\frac{RULESTRENGTH_{PREFERRED}}{\max \langle RULESTRENGTH \rangle_{NONPREFERRED}} \right)}$$

- "0": the likelihood ratio

$$PMI = \frac{RULESTRENGTH_{PREFERRED}}{\sum RULESTRENGTH}$$

Set "Calibration Considerations".

Detailed Settings of PreClassifier

`dpolapp.Preclassifier.OtherInputs[0].iParams[0] = 5`

The number of consecutive bins used to compute texture of reflectivity.

`dpolapp.Preclassifier.OtherInputs[1].iParams[0] = 10`

The number of consecutive bins used to compute texture of differential phase.

```
# MBFinputs and their use:
# Reflectivity in preclassifier:
# Input data: Zh.
# MBF(Zh) is computed as JPOLE trapetzoid; an additive factor
in rule strength
# MBF: default initialization with parameter settings of
Table 1 Ref.1
# Zh MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[0].dMBF[0][0] = 15
dpolapp.Preclassifier.MBFinputs[0].dMBF[1][0] = 20
dpolapp.Preclassifier.MBFinputs[0].dMBF[2][0] = 75
dpolapp.Preclassifier.MBFinputs[0].dMBF[3][0] = 85
# Zh MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[0].dMBF[0][1] = 5
dpolapp.Preclassifier.MBFinputs[0].dMBF[1][1] = 10
dpolapp.Preclassifier.MBFinputs[0].dMBF[2][1] = 20
dpolapp.Preclassifier.MBFinputs[0].dMBF[3][1] = 30
# Zh MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[0].dMBF[0][2] = 5
dpolapp.Preclassifier.MBFinputs[0].dMBF[1][2] = 10
dpolapp.Preclassifier.MBFinputs[0].dMBF[2][2] = 75
dpolapp.Preclassifier.MBFinputs[0].dMBF[3][2] = 80
# Differential reflectivity in preclassifier:
# Input data: Zdr (In reingest: adjusted with Quality offset,
internally), and Zh.
```

```
# MBF(Zdr;Zh) is computed as JPOLE 2D-trapetzoid; additive
in rule strength
# Zdr MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[1].dMBF[0][0] = -4
dpolapp.Preclassifier.MBFinputs[1].dMBF[1][0] = -2
dpolapp.Preclassifier.MBFinputs[1].dMBF[2][0] = 1
dpolapp.Preclassifier.MBFinputs[1].dMBF[3][0] = 2
# Zdr MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[1].dMBF[0][1] = 0
dpolapp.Preclassifier.MBFinputs[1].dMBF[1][1] = 2
dpolapp.Preclassifier.MBFinputs[1].dMBF[2][1] = 10
dpolapp.Preclassifier.MBFinputs[1].dMBF[3][1] = 12
# Zdr MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[1].dMBF[0][2] = -0.3
dpolapp.Preclassifier.MBFinputs[1].dMBF[1][2] = 0
dpolapp.Preclassifier.MBFinputs[1].dMBF[2][2] = 0
dpolapp.Preclassifier.MBFinputs[1].dMBF[3][2] = 0.3
# A polynomial dependence of the Zdr precipitation trapezoid
on Zh
# The Zh range in which the polynomial is applied:
dpolapp.Preclassifier.MBFinputs[1].dMBF[4][2] = 0
dpolapp.Preclassifier.MBFinputs[1].dMBF[5][2] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.Preclassifier.MBFinputs[1].dP[2][0][0] = -0.5
dpolapp.Preclassifier.MBFinputs[1].dP[2][0][1] = 0.0025
dpolapp.Preclassifier.MBFinputs[1].dP[2][0][2] = 0.00075
dpolapp.Preclassifier.MBFinputs[1].dP[2][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.Preclassifier.MBFinputs[1].dP[2][2][0] = 0.08
dpolapp.Preclassifier.MBFinputs[1].dP[2][2][1] = 0.0364
dpolapp.Preclassifier.MBFinputs[1].dP[2][2][2] = 0.000357
dpolapp.Preclassifier.MBFinputs[1].dP[2][2][3] = 0
# Cross correlation coefficient in preclassifier:
# Data type used as input: RhoHV.
# MBF(RhoHV) is computed as JPOLE trapetzoid;  an additive
factor in rule strength
# MBF: default initialization with parameter settings of
Table 1 Ref.1
# RHOhv MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[2].dMBF[0][0] = 0.5
dpolapp.Preclassifier.MBFinputs[2].dMBF[1][0] = 0.6
dpolapp.Preclassifier.MBFinputs[2].dMBF[2][0] = 0.9
dpolapp.Preclassifier.MBFinputs[2].dMBF[3][0] = 0.95
# RHOhv MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[2].dMBF[0][1] = 0
dpolapp.Preclassifier.MBFinputs[2].dMBF[1][1] = 0
dpolapp.Preclassifier.MBFinputs[2].dMBF[2][1] = 0.8
dpolapp.Preclassifier.MBFinputs[2].dMBF[3][1] = 0.83
# RHOhv MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[2].dMBF[0][2] = 0.85
dpolapp.Preclassifier.MBFinputs[2].dMBF[1][2] = 0.97
```

```
dpolapp.Preclassifier.MBFinputs[2].dMBF[2][2] = 1
dpolapp.Preclassifier.MBFinputs[2].dMBF[3][2] = 1.01
# Differential phase texture in preclassifier:
# The input data type is computed internally.
#--- MBF(Texture-1) is computed as JPOLE trapetzoid;  an
additive factor in rule strength
# MBF: default initialization with parameter settings of
Table 1 Ref.1
# PHIDP texture MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[3].dMBF[0][0] = 30
dpolapp.Preclassifier.MBFinputs[3].dMBF[1][0] = 40
dpolapp.Preclassifier.MBFinputs[3].dMBF[2][0] = 10800
dpolapp.Preclassifier.MBFinputs[3].dMBF[3][0] = 10800
# PHIDP texture MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[3].dMBF[0][1] = 8
dpolapp.Preclassifier.MBFinputs[3].dMBF[1][1] = 10
dpolapp.Preclassifier.MBFinputs[3].dMBF[2][1] = 40
dpolapp.Preclassifier.MBFinputs[3].dMBF[3][1] = 60
# PHIDP texture MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[3].dMBF[0][2] = 0
dpolapp.Preclassifier.MBFinputs[3].dMBF[1][2] = 1
dpolapp.Preclassifier.MBFinputs[3].dMBF[2][2] = 15
dpolapp.Preclassifier.MBFinputs[3].dMBF[3][2] = 30
# Reflectivity texture in preclassifier:
# Data type is computed internally.
# MBF(Texture-2) is computed as JPOLE trapetzoid; an additive
factor in the rule strength
# MBF: default initialization with parameter settings of
Table 1 Ref.1
# Zh texture MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[4].dMBF[0][0] = 2
dpolapp.Preclassifier.MBFinputs[4].dMBF[1][0] = 4
dpolapp.Preclassifier.MBFinputs[4].dMBF[2][0] = 10000
dpolapp.Preclassifier.MBFinputs[4].dMBF[3][0] = 10000
# Zh texture MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[4].dMBF[0][1] = 1
dpolapp.Preclassifier.MBFinputs[4].dMBF[1][1] = 2
dpolapp.Preclassifier.MBFinputs[4].dMBF[2][1] = 4
dpolapp.Preclassifier.MBFinputs[4].dMBF[3][1] = 7
# Zh texture MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[4].dMBF[0][2] = 0
dpolapp.Preclassifier.MBFinputs[4].dMBF[1][2] = 0.5
dpolapp.Preclassifier.MBFinputs[4].dMBF[2][2] = 3
dpolapp.Preclassifier.MBFinputs[4].dMBF[3][2] = 6
# Other flags: none.
```

General Settings of MeteoClassifiers

```
dpolapp.MeteoClassifiers.uiNonMeteoID = 1
```

This parameter is a recipy what to do with the bins that fail the preceding quality consideration (PreClassifier), or fail the acceptance to any class of MeteoClassifiers.

- "0": MeteoClassifiers will redirect the unaccepted and unclassified bins as IRIS/RDA 'CLASS_THRESHOLD'
- "1": MeteoClassifiers will redirect the unaccepted and unclassified bins as IRIS/RDA "CLASS_NON_MET"

```
dpolapp.MeteoClassifiers.dMinRS[0] = 0
```

Minimum rule strength for accepting rain

```
dpolapp.MeteoClassifiers.dMinRS[1] = 0
```

Minimum rule strength for accepting wet snow

```
dpolapp.MeteoClassifiers.dMinRS[2] = 0
```

Minimum rule strength for accepting snow

```
dpolapp.MeteoClassifiers.dMinRS[3] = 0
```

Minimum rule strength for accepting graupel

```
dpolapp.MeteoClassifiers.dMinRS[4] = 0
```

Minimum rule strength for accepting hail

```
dpolapp.MeteoClassifiers.dMinRS[5] = 0
```

Minimum rule strength for accepting rain/hail mixture (reported as hail)

Detailed Settings of MeteoClassifiers

```
# MBFinputs and their use:
# Reflectivity in MeteoClassifiers:
# Data type used as input: Zh.
# MBF(dBZ) is computed as the CSU beta function.
# MBF: C-band, default settings:
# Zh beta MBF for rain: central value, width, slope
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][0] = 30
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][0] = 31
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][0] = 40
# Zh beta MBF for wet snow.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][1] = 25
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][1] = 21
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][1] = 40
```

```
# Zh beta MBF for snow.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][2] = 36
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][2] = 40
# Zh beta MBF for graupel/small hail.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][3] = 45
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][3] = 11
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][3] = 20
# Zh beta MBF for large hail.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][4] = 57.5
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][4] = 14
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][4] = 20
# Zh beta MBF for rain+hail mixture.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][5] = 60
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][5] = 11
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][5] = 20
# Altitude in MeteoClassifiers:
# The IRIS/RDA estimate of the melting level height will be
used.
# MBF(altitude;melting level) is computed as the CSU beta(ML)
function.
# MBF: C-band, default settings:
# Altitude beta MBF for rain: central value, width, slope
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][0] = -0.2
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][0] = 20
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][0] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][0] = 5
# Altitude beta MBF for wet snow.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][1] = -0.3
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][1] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][1] = 5
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][1] = 1
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][1] = 5
# Altitude beta MBF for snow.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][2] = 10
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][2] = 10.2
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][2] = 60
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][2] = 0
# Altitude beta MBF for graupel/small hail.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][3] = 10
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][3] = 10
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][3] = 12
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][3] = 60
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][3] = 0
# Altitude beta MBF for hail.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][4] = 10
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][4] = 15
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][4] = 20
```

```
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][4] = 0
# Altitude beta MBF for rain+hail mixture.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][5] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][5] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][5] = 20
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][5] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][5] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][5] = 0
# Differential reflectivity in MeteoClassifiers:
# Input data: Zdr (adjusted with Quality offset, internally
in reingest), and Zh
# MBF(Zdr) is computed as the CSU 2D-beta function of Zh.
# MBF: C-band, default settings:
# Zdr 2D beta MBF for rain at Zh=<Zh> dB (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][0] = 0.48155
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][0] = 0.70578
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][0] = 10.958
# Polynomial dependence of rain MBF(Zdr) on Zh-<Zh>
#  The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][0] = 30
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][0] = 30
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraints:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][0] =
0.047498
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][1] =
0.0017624
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][2] = 6.993E-
06
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][3] = -
4.4975E-07
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][4] = 0.1
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][5] = 5.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][0] =
0.025917
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][1] =
0.00043621
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][2] = -
4.8951E-06
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][3] = -
5.6218E-08
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][4] = 0.4
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][5] = 8
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][0] = 0.22873
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][1] =
0.0054945
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][2] = -
5.8275E-05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][4] = 8
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][5] = 25
```

```
# Zdr 2D beta MBF for wet snow at Zh=<Zh> (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][1] = 1.1031
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][1] = 1.3594
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][1] = 15
# Polynomial dependence of wet snow MBF(Zdr) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][1] = 27.5
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][1] = 17.5
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][0] =
0.004623
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][1] =
0.00064286
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][2] =
2.2222E-05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][4] = 1.1
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][5] = 2.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][0] = -
0.0091144
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][1] =
0.00021429
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][2] =
3.8384E-05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][4] = 1
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][5] = 2.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][4] = 5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][5] = 20
# Zdr 2D beta MBF for snow at Zh=<Zh> (central value, width,
slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][2] = 0.25
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][2] = 0.75
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][2] = 8
# Zdr 2D beta MBF for graupel at Zh=<Zh> (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][3] = 0.42969
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][3] = 0.92969
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][3] = 12.5
# Polynomial dependence of graupel MBF(Zdr) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][3] = 42.5
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][3] = 12.5
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th orde;min/max constraints:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][0] =
0.033714
```

```
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][1] =
0.00096429
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][4] = 0.155
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][5] = 1.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][0] =
0.033714
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][1] =
0.00096429
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][4] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][5] = 2
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][0] = 0.25714
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][4] = 9
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][5] = 17
# Zdr weight and 2D beta MBF for hail at Zh=<Zh> (central
value, width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][4] = -0.75928
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][4] = 1.2778
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][4] = 15.859
# Polynomial dependence of hail MBF(Zdr) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][4] = 57.5
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][4] = 12.5
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order;min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][0] = -
0.017336
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][1] =
0.0015104
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][2] = -
8.3333E-05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][3] =
4.1667E-06
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][4] = -2
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][5] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][0] =
0.0057788
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][1] = -
0.0045313
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][2] =
2.7778E-05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][3] = 1.25E-
05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][4] = 0.1
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][5] = 1.3
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][1] = -
0.14167
```

```
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][3] =
0.00066667
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][4] = 7
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][5] = 17
# Zdr weight and 2D beta MBF for rain+hail mixture at Zh=<Zh>
(central value, width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][5] = 0.93571
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][5] = 2
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][5] = 15
# Polynomial dependence of rain+hail mixture MBF(Zdr) on Zh-
<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][5] = 60
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][5] = 10
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][0] = -
0.054167
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][1] = -
0.0057143
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][2] =
0.00016667
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][4] = -0.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][5] = 1.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][0] =
0.0041667
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][1] = -
0.02375
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][2] = -
0.00016667
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][3] = 0.00015
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][4] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][5] = 3
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][0] = -
5.9212E-17
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][1] =
0.016667
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][2] =
5.9212E-19
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][3] = -
0.00066667
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][4] = 5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][5] = 20
# Specific differential phase in MeteoClassifiers:
# Input data: Kdp, and Zh.
# MBF(Kdp) is computed as the CSU 2D-beta function of Zh.
# MBF: C-band, default settings:
# Kdp 2D beta MBF for rain at Zh=<Zh> dB (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][0] =
0.0079391
```

```
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][0] = -
0.032435
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][0] = 11.154
# Polynomial dependence of rain MBF(Kdp) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[3][0] = 30
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[4][0] = 35
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][0] = -
0.0049584
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][1] =
0.0020558
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][2] =
0.00019114
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][3] =
3.9888E-06
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][4] = 0.05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][5] = 14
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][0] =
0.011085
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][1] =
0.0029526
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][2] =
0.00010385
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][3] =
6.1086E-07
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][4] = 0.2
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][5] = 8
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][0] = 0.22873
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][1] =
0.0054945
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][2] = -
5.8275E-05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][4] = 9
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][5] = 16
# Kdp 2D beta MBF for wet snow at Zh=25 dB: central value,
width, slope.
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][1] = 0.25
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][1] = 1.2
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][1] = 10
# Kdp 2D beta MBF for snow at Zh=17.5 dB (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][2] = 0.25
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][2] = 10
# Kdp 2D beta MBF for graupel at Zh=42.5 dB (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][3] = 0.26875
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][3] = 0.76875
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][3] = 12.5
# Polynomial dependence of graupel MBF(Kdp) on Zh-<Zh>
```

```
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[3][3] = 42.5
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[4][3] = 12.5
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][0] =
0.068704
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][1] = 0.001
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][2] = -
0.00014815
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][4] = -0.25
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][5] = 1.5
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][0] =
0.068704
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][1] = 0.001
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][2] = -
0.00014815
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][4] = 0.3
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][5] = 2
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][0] = 0.25714
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][4] = 9
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][5] = 17
# Kdp weight and 2D beta MBF for hail at Zh=57.5 dB (central
value, width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][4] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][4] = 1
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][4] = 10
# Kdp weight and 2D beta MBF for rain+hail mixture at Zh=60
dB (central value, width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][5] = 2.3714
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][5] = 2.3971
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][5] = 15
# Polynomial dependence of rain+hail mixture MBF(Kdp) on Zh-
<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[3][5] = 60
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[4][5] = 20
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][0] = 0.30833
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][1] =
0.0085714
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][2] = -
0.00033333
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][4] = 0
polapp.MeteoClassifiers.MBFinputs[3].dP[5][0][5] = 10
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][0] = 0.29667
```

```
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][1] =
0.0088571
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][2] = -
0.00026667
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][4] = -1
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][5] = 10
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][0] = -
0.16667
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][1] =
2.4371E-17
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][2] =
0.0066667
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][4] = 8
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][5] = 25
# Cross correlation coefficient in MeteoClassifiers:
#--- Data type used as input: RhoHV.
#--- MBF(Rhohv) is computed as the CSU beta function.
# # MBF:  C-band, default settings:
# RHOHV beta MBF for rain: central value, width, slope
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][0] = 1
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][0] = 0.04
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][0] = 10
# RHOHV beta MBF for wet snow.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][1] = 0.88
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][1] = 0.11
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][1] = 20
# RHOHV beta MBF for snow.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][2] = 1
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][2] = 0.06
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][2] = 10
# RHOHV beta MBF for graupel.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][3] = 0.96
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][3] = 0.04
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][3] = 10
# RHOHV beta MBF for hail.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][4] = 0.9
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][4] = 0.045
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][4] = 10
# RHOHV beta MBF for rain+hail mixture.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][5] = 0.8
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][5] = 0.13
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][5] = 30
```

General Settings of PrecipClassifier

```
dpolapp.PrecipClassifier.uiNonMeteoID = 1
```

This parameter is a recipe of what to do with the bins that fail the preceding quality consideration (PreClassifier), or fail the acceptance to any class of PrecipClassifier.

- "0": PrecipClassifier will redirect the unaccepted and unclassified bins as IRIS/RDA 'CLASS_THRESHOLD'
- "1": PrecipClassifier will report the unaccepted PreClassifier classifications unchanged, and report the unclassified bins as IRIS/RDA "CLASS_NON_MET"

```
dpolapp.PrecipClassifier.dMinRS[0] = 0
```

Minimum rule strength for accepting light rain.

```
dpolapp.PrecipClassifier.dMinRS[1] = 0
```

Minimum rule strength for accepting moderate rain.

```
dpolapp.PrecipClassifier.dMinRS[2] = 0
```

Minimum rule strength for accepting heavy rain.

```
dpolapp.PrecipClassifier.dMinRS[3] = 0
```

Minimum rule strength for accepting large drops.

Detailed Settings of PrecipClassifier

```
# Precip classifier uses the textures of reflectivity and
differential phase.
# MBFinputs and their use:
# Reflectivity in rain classifier:
# Input data: Zh.
# MBF(Zh) is computed as JPOLE trapetzoid; an additive factor
in rule strength
# MBF: default initialization with parameter settings of
Table 2 Ref.2
# Zh MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[0][0] = 5
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[1][0] = 10
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[2][0] = 35
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[3][0] = 40
# Zh MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[0][1] = 30
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[1][1] = 35
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[2][1] = 45
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[3][1] = 50
# Zh MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[0][2] = 40
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[1][2] = 45
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[2][2] = 75
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[3][2] = 80
# Zh MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[0][3] = 15
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[1][3] = 20
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[2][3] = 45
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[3][3] = 50
```

```
# Differential reflectivity in rain classifier:
# Input data: Zdr (In reingest: adjusted with Quality offset,
internally), and Zh.
# MBF(Zdr;Zh) is computed as JPOLE 2D-trapetzoid; additive
in rule strength
# Zdr MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[0][0] = -0.3
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[1][0] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[2][0] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[3][0] = 0.3
# The polynomial dependence of the Zdr light precipitation
trapezoid on Zh
# The Zh range in which the polynomial is applied:
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[4][0] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[5][0] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][0][0] = -0.5
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][0][1] = 0.0025
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][0][2] = 0.00075
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][2][0] = 0.08
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][2][1] = 0.0364
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][2][2] =
0.000357
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][2][3] = 0
# Zdr MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[0][1] = -0.3
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[1][1] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[2][1] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[3][1] = 0.3
# The polynomial dependence of the Zdr moderate precipitation
trapezoid on Zh
# The Zh range in which the polynomial is applied:
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[4][1] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[5][1] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][0][0] = -0.5
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][0][1] = 0.0025
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][0][2] = 0.00075
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][2][0] = 0.08
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][2][1] = 0.0364
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][2][2] =
0.000357
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][2][3] = 0
```

```
# Zdr MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[0][2] = -0.3
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[1][2] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[2][2] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[3][2] = 0.3
# The polynomial dependence of the Zdr heavy precipitation
trapezoid on Zh
# The Zh range in which the polynomial is applied:
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[4][2] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[5][2] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][0][0] = -0.5
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][0][1] = 0.0025
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][0][2] = 0.00075
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][2][0] = 0.08
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][2][1] = 0.0364
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][2][2] =
0.000357
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][2][3] = 0
# Zdr MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[0][3] = -0.3
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[1][3] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[2][3] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[3][3] = 0.3
# The polynomial dependence of the Zdr large drops trapezoid
on Zh
# The Zh range in which the polynomial is applied:
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[4][3] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[5][3] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][0][0] = -0.5
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][0][1] = 0.0025
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][0][2] = 0.00075
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][2][0] = 0.08
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][2][1] = 0.0364
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][2][2] =
0.000357
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][2][3] = 0
# Cross correlation coefficient in rain classifier:
# Data type used as input: RhoHV.
# MBF(RhoHV) is computed as JPOLE trapetzoid;  an additive
factor in rule strength
```

```
# MBF: default initialization with parameter settings of
Table 2 Ref.1
# RHOhv MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[0][0] = 0.85
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[1][0] = 0.97
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[2][0] = 1
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[3][0] = 1.01
# RHOhv MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[0][1] = 0.85
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[1][1] = 0.97
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[2][1] = 1
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[3][1] = 1.01
# RHOhv MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[0][2] = 0.85
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[1][2] = 0.97
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[2][2] = 1
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[3][2] = 1.01
# RHOhv MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[1][3] = 0.97
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[0][3] = 0.85
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[2][3] = 1
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[3][3] = 1.01
# Differential phase texture in rain classifier:
# The input data type is computed internally.
#--- MBF(Texture-1) is computed as JPOLE trapetzoid;  an
additive factor in rule strength
# MBF: default initialization with parameter settings of
Table 2 Ref.1
# PHIDP texture MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[0][0] = 0
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[1][0] = 1
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[2][0] = 15
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[3][0] = 30
# PHIDP texture MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[0][1] = 0
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[1][1] = 1
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[2][1] = 15
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[3][1] = 30
# PHIDP texture MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[0][2] = 0
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[1][2] = 1
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[2][2] = 15
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[3][2] = 30
# PHIDP texture MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[0][3] = 0
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[1][3] = 1
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[2][3] = 15
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[3][3] = 30
# Reflectivity texture in rain classifier:
# Data type is computed internally.
# MBF(Texture-2) is computed as JPOLE trapetzoid; an additive
factor in the rule strength
# MBF: default initialization with parameter settings of
Table 2 Ref.1
# Zh texture MBF trapetzoid for light precipitation.
```

```
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[0][0] = 0
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[1][0] = 0.5
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[2][0] = 3
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[3][0] = 6
# Zh texture MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[0][1] = 0
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[1][1] = 0.5
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[2][1] = 3
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[3][1] = 6
# Zh texture MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[0][2] = 0
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[1][2] = 0.5
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[2][2] = 3
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[3][2] = 6
# Zh texture MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[0][3] = 0
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[1][3] = 0.5
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[2][3] = 3
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[3][3] = 6
```

Settings of CellClassifier

```
dpolapp.CellClassifier.dMinRS[0] = 0.5
```

Minimum rule strength for accepting convective rain.

```
# MBFinputs and their use:
# Reflectivity&Height in cell classifier:
# Input data: Zh and height difference w.r.t. the current 0oC
isotherm.
# MBF(Zh,height) is computed as 2D product of trapetzoids;
summed up to rule strength.
# Min Zh required for convection (full fuzzy strength will
be +5 dBZ from min).
dpolapp.CellClassifier.MBFinputs[0].dMBF[0][0] = 25
# Min height w.r.t. 0oC isotherm (full fuzzy strength will
be +1. km higher ).
dpolapp.CellClassifier.MBFinputs[0].dMBF[4][0] = 0
```

4.4.4 Miscellaneous

The following list is a compilation of issues encountered in first use cases of HydroClass in field use. The topics as listed in format of Frequently Asked Questions (FAQ).

Q: How/when does HydroClass check for runtime configuration updates?

A:

Q: How HydroClass treats typical failure cases related to dpolapp.conf, such as:

- Configuration file missing?
  A: switches off until restart of IRIS/RDA.
- Syntax errors in dpolapp.conf parameter lines?
  A: switches off until file fixed.

Q: How to feed in the melting level (ML) value as an external input to HydroClass?

A: When operating:

- RVP900 with IRIS/Radar `"setup_change -load"` tool, see SRI Product documentation
- RVP900 without IRIS, contact Vaisala
- IRIS/reingest, the ML stored in RAW is used

NOTE

In order to use these momentary values of ML, HydroClass configuration dpolapp.conf needs to be set up with:

```
"dpolapp.Quality.iAux[0] = 0"
```

Alternatively, one can edit and update (either manually or with customized script tools) `"dpolapp.Quality.iAux[0] = 1"`,

and

```
"dpolapp.Quality.iAux[1] = 3500"
```

(where 3500 represents an example ML value, meters above mean sea level)

Q: How to monitor the ML value used/in use?

A: When operating in IRIS/Radar, the utility command `"setup_change -list | grep ifalls"` is available; when operating plain RVP900, use `"dspx > Mp"`

Q: When running HydroClass in distributed PC architectures such as RVPO900/AMR:

- Where is the HydroClass configuration in use?
  A: at PC running IRIS/Radar
- Where to update the melting level with `setup_change`?
  A: at IRIS/Radar

Q: How do HClass data look when HydroClass is tested using noise source input?

A: In commissioning phase and anytime, a viable method of testing functionality of HydroClass is to couple a coherent noise source to RVP900 IF inputs (for example, AMR built-in noise). HydroClass is able to process such a data stream, by configuring RVP900 appropriately. Observing HClass data stream with ascope, for example, see the data in Figure 26 on page 104.

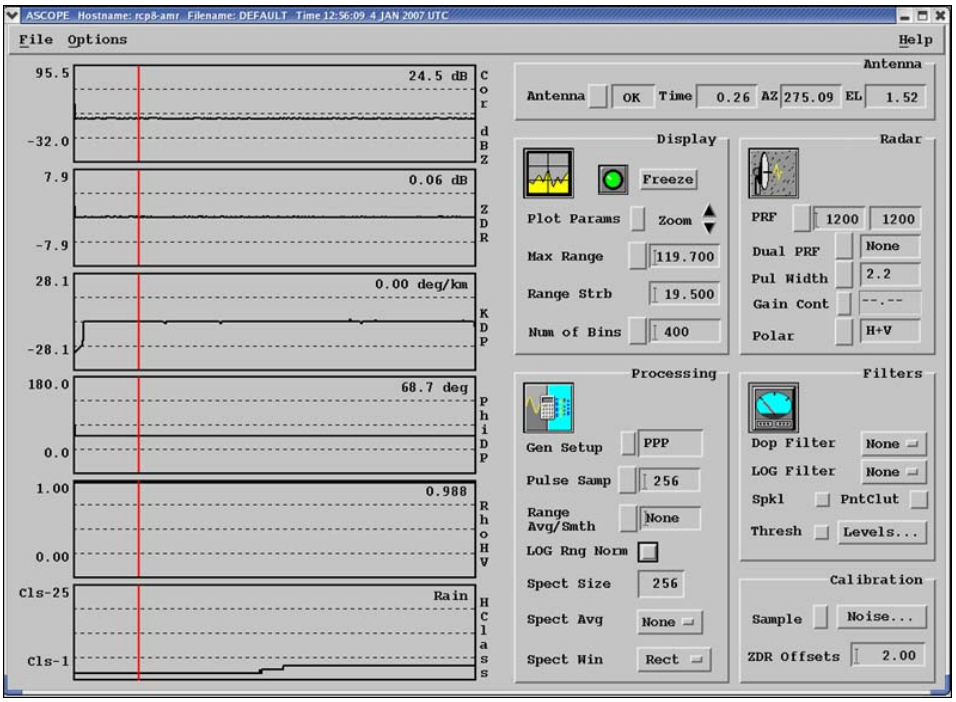


Figure 26 Polarimetric Measurements with HClass in an Ascope View When RVP900 is Processing Artificial Noise Inputs (Correlated IF Signal in Both Receiver Ports H and V)

NOTE	HydroClass is in default configuration and uses the melting layer height value given in the configuration file (2300 m).

4.4.4.1 Additional notes

- There are limitations in ML height settings when running ascope; it is possible to use the default value of ML in RVP900, or a specified value in dpolapp.conf (dynamic update does not work).
- There are limitations in ML settings when running time series data in play-back; RVP900 uses the ML value stored on TS data.

- The HydroClass ML is defined as the 0°C isotherm, see Section 4.3.4 CellClassifier on page 77. Consequently, the melting layer (or the radar bright band) is typically located at altitudes below ML.

4.5 APPENDICES

4.5.1 Default Configuration for S-band Radars

The following is the default configuration file for S-band radars:

```
# ==== RDA/IRIS HydroClass configuration file (S-band) ====

# HydroClass configuration name:
# use a descriptive name max 15 characters, saved in ingest
header when IRIS
dpolapp.sConfigurationName = "dpolapp_S-band"

#
============================================================
====================
# Main algorithm menu. Activate algorithms by selecting
(1:yes, 0:no):
# Activate quality actions (see block Quality):
dpolapp.bRunQuality = 1
# Activate the identification algorithms:
# 1: Preclassifier, only (the JPOLE algorithm for meteo vs.
bio vs. GC/AP)
# 2: Meteoclassifiers, with the preclassifier mask
# 21: Precip classifier (the JPOLE warm algorithm, with
preclassifier mask.
# 22: Meteoclassifiers and Precip classifier, with
preclassifier mask.
# 201: Classifier for stratiform and convective rain (volume
level analysis), with preclassifier mask.
# 202: Meteoclassifiers and stratifrom/convective
classifier, with preclassifier mask.
# 221: Precip classifier and stratifrom/convective
classifier, with preclassifier mask.
# 222: Precip classifier, Meteoclassifiers and stratifrom/
convective classifier, with preclassifier mask.
dpolapp.usClassificationMethods = 222

#
============================================================
====================
# RDA/IRIS major version
dpolapp.sVersion = "8.13"
# RDA/IRIS revision
dpolapp.sRevision = ".0"
```

```
# dpolapp running version
dpolapp.sDpolappRevision = "15"

#
============================================================
====================
# Detailed data quality configurations:
# Input source of the melting layer height:
# 1: retrieve the value in this configuration file
# 0: use the value in RAW reingested (reingest), or the value
in INU record (rda):
dpolapp.Quality.iAux[0] = 1
# The explicit melting layer height. Signed integer [m]:
dpolapp.Quality.iAux[1] = 2300
#
# Mode of input data quality considerations
# 0: select the input data passing quality thresholds
(recommended).
# 1: select all the computed input data (possible in RDA,
only).
dpolapp.Quality.iAux[2] = 0
#
# Extra ZDR balance offset to be applied [dB] (applicable in
reingest, only):
dpolapp.Quality.OtherInputs[0].dParams[0] = 0

#
============================================================
====================
# Detailed configuration of the preclassifier.
# Fixed features:
# Preclassifier has 5 input observables: Zh, Zdr, RhoHV, Zh-
texture, Phidp-texture.
# There are 3 output classes: GC/AP, BIO, PRECIP.
# The rule strength function RS is sum of MBF(i) and the class
assignment is max RS.
#
# Are the results to be formatted as PRECIP_CLASSIFIER in the
bits 3,4,5 from LSB=0 of DB_HCLASS(2)?
# default: -1 (NO, the legacy HydroClass format),
1:PRECIP_CLASSIFIER
dpolapp.Preclassifier.iClassifierID = -1
# IF (usClassificationMethods>1, Preclassifier outcomes are
interpreted as 'NoMet' or 'Precipitation')
# .. see details in blocks of MeteoClassifiers,
PrecipClassifier and CellClassifier.

# ELSE IF (usClassificationMethods=1, i.e. preclassifier
alone):'
# .. one can re-group Preclassifier results into
'uiNonMeteoID':
# .. 1: unidentified are 'uiNonMeteoID'; 2:'GC/AP' are
'uiNonMeteoID', too; 3:'BIO' are 'uiNonMeteoID', too.
dpolapp.Preclassifier.uiMaxNonMeteo = 1
# .. 'uiNonMeteoID' group is redirected:
```

```
# .. 0:IRIS/RDA threshold, 1:'GC/AP' 2:'BIO' 3:'PRECIP'
dpolapp.Preclassifier.uiNonMeteoID = 0
# Class specific thresholds on rule strengths for a class to
be accepted (recommend:0.)
dpolapp.Preclassifier.dMinRS[0] = 0
dpolapp.Preclassifier.dMinRS[1] = 0
dpolapp.Preclassifier.dMinRS[2] = 0
# Polarimetric quality index, given the set of 'PRECIP',
'BIO' and 'GC/AP'
# The preferred class. Default: 2 'PRECIP'. Alternatives: 1
'BIO', 0 'GC/AP'
dpolapp.Preclassifier.iAux[3] = 2
# PMI formula. Default: 1: 'score' (1-(1/(1+RSpref/
max(RScomplement)))) 0: likelihood ratio
dpolapp.Preclassifier.iAux[4] = 1

# Preclassifier computes the textures of reflectivity and
differential phase.
# Texture-1:
# Input data: Zh.
# Range interval for averaging [Nr of gates]:
dpolapp.Preclassifier.OtherInputs[0].iParams[0] = 5
# Texture-2:
# Data type used as input: Phidp.
# Range interval for averaging [Nr of gates]:
dpolapp.Preclassifier.OtherInputs[1].iParams[0] = 10

# MBFinputs and their use:
# Reflectivity in preclassifier:
# Input data: Zh.
# MBF(Zh) is computed as JPOLE trapetzoid; an additive factor
in rule strength
# MBF: default initialization with parameter settings of
Table 1 Ref.1
# Zh MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[0].dMBF[0][0] = 15
dpolapp.Preclassifier.MBFinputs[0].dMBF[1][0] = 20
dpolapp.Preclassifier.MBFinputs[0].dMBF[2][0] = 75
dpolapp.Preclassifier.MBFinputs[0].dMBF[3][0] = 85
# Zh MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[0].dMBF[0][1] = 5
dpolapp.Preclassifier.MBFinputs[0].dMBF[1][1] = 10
dpolapp.Preclassifier.MBFinputs[0].dMBF[2][1] = 20
dpolapp.Preclassifier.MBFinputs[0].dMBF[3][1] = 30
# Zh MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[0].dMBF[0][2] = 5
dpolapp.Preclassifier.MBFinputs[0].dMBF[1][2] = 10
dpolapp.Preclassifier.MBFinputs[0].dMBF[2][2] = 75
dpolapp.Preclassifier.MBFinputs[0].dMBF[3][2] = 80
# Differential reflectivity in preclassifier:
# Input data: Zdr (In reingest: adjusted with Quality offset,
internally), and Zh.
# MBF(Zdr;Zh) is computed as JPOLE 2D-trapetzoid; additive
in rule strength
# Zdr MBF trapetzoid for GC/AP.
```

```
dpolapp.Preclassifier.MBFinputs[1].dMBF[0][0] = -4
dpolapp.Preclassifier.MBFinputs[1].dMBF[1][0] = -2
dpolapp.Preclassifier.MBFinputs[1].dMBF[2][0] = 1
dpolapp.Preclassifier.MBFinputs[1].dMBF[3][0] = 2
# Zdr MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[1].dMBF[0][1] = 0
dpolapp.Preclassifier.MBFinputs[1].dMBF[1][1] = 2
dpolapp.Preclassifier.MBFinputs[1].dMBF[2][1] = 10
dpolapp.Preclassifier.MBFinputs[1].dMBF[3][1] = 12
# Zdr MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[1].dMBF[0][2] = -0.3
dpolapp.Preclassifier.MBFinputs[1].dMBF[1][2] = 0
dpolapp.Preclassifier.MBFinputs[1].dMBF[2][2] = 0
dpolapp.Preclassifier.MBFinputs[1].dMBF[3][2] = 0.3
# A polynomial dependence of the Zdr precipitation trapezoid
on Zh
# The Zh range in which the polynomial is applied:
dpolapp.Preclassifier.MBFinputs[1].dMBF[4][2] = 0
dpolapp.Preclassifier.MBFinputs[1].dMBF[5][2] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.Preclassifier.MBFinputs[1].dP[2][0][0] = -0.5
dpolapp.Preclassifier.MBFinputs[1].dP[2][0][1] = 0.0025
dpolapp.Preclassifier.MBFinputs[1].dP[2][0][2] = 0.00075
dpolapp.Preclassifier.MBFinputs[1].dP[2][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.Preclassifier.MBFinputs[1].dP[2][2][0] = 0.08
dpolapp.Preclassifier.MBFinputs[1].dP[2][2][1] = 0.0364
dpolapp.Preclassifier.MBFinputs[1].dP[2][2][2] = 0.000357
dpolapp.Preclassifier.MBFinputs[1].dP[2][2][3] = 0
# Cross correlation coefficient in preclassifier:
# Data type used as input: RhoHV.
# MBF(RhoHV) is computed as JPOLE trapetzoid;  an additive
factor in rule strength
# MBF: default initialization with parameter settings of
Table 1 Ref.1
# RHOhv MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[2].dMBF[0][0] = 0.5
dpolapp.Preclassifier.MBFinputs[2].dMBF[1][0] = 0.6
dpolapp.Preclassifier.MBFinputs[2].dMBF[2][0] = 0.9
dpolapp.Preclassifier.MBFinputs[2].dMBF[3][0] = 0.95
# RHOhv MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[2].dMBF[0][1] = 0
dpolapp.Preclassifier.MBFinputs[2].dMBF[1][1] = 0
dpolapp.Preclassifier.MBFinputs[2].dMBF[2][1] = 0.8
dpolapp.Preclassifier.MBFinputs[2].dMBF[3][1] = 0.83
# RHOhv MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[2].dMBF[0][2] = 0.85
dpolapp.Preclassifier.MBFinputs[2].dMBF[1][2] = 0.97
dpolapp.Preclassifier.MBFinputs[2].dMBF[2][2] = 1
dpolapp.Preclassifier.MBFinputs[2].dMBF[3][2] = 1.01
# Differential phase texture in preclassifier:
```

```
# The input data type is computed internally.
#--- MBF(Texture-1) is computed as JPOLE trapetzoid;  an
additive factor in rule strength
# MBF: default initialization with parameter settings of
Table 1 Ref.1
# PHIDP texture MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[3].dMBF[0][0] = 30
dpolapp.Preclassifier.MBFinputs[3].dMBF[1][0] = 40
dpolapp.Preclassifier.MBFinputs[3].dMBF[2][0] = 10800
dpolapp.Preclassifier.MBFinputs[3].dMBF[3][0] = 10800
# PHIDP texture MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[3].dMBF[0][1] = 8
dpolapp.Preclassifier.MBFinputs[3].dMBF[1][1] = 10
dpolapp.Preclassifier.MBFinputs[3].dMBF[2][1] = 40
dpolapp.Preclassifier.MBFinputs[3].dMBF[3][1] = 60
# PHIDP texture MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[3].dMBF[0][2] = 0
dpolapp.Preclassifier.MBFinputs[3].dMBF[1][2] = 1
dpolapp.Preclassifier.MBFinputs[3].dMBF[2][2] = 15
dpolapp.Preclassifier.MBFinputs[3].dMBF[3][2] = 30
# Reflectivity texture in preclassifier:
# Data type is computed internally.
# MBF(Texture-2) is computed as JPOLE trapetzoid; an additive
factor in the rule strength
# MBF: default initialization with parameter settings of
Table 1 Ref.1
# Zh texture MBF trapetzoid for GC/AP.
dpolapp.Preclassifier.MBFinputs[4].dMBF[0][0] = 2
dpolapp.Preclassifier.MBFinputs[4].dMBF[1][0] = 4
dpolapp.Preclassifier.MBFinputs[4].dMBF[2][0] = 10000
dpolapp.Preclassifier.MBFinputs[4].dMBF[3][0] = 10000
# Zh texture MBF trapetzoid for bio scatter.
dpolapp.Preclassifier.MBFinputs[4].dMBF[0][1] = 1
dpolapp.Preclassifier.MBFinputs[4].dMBF[1][1] = 2
dpolapp.Preclassifier.MBFinputs[4].dMBF[2][1] = 4
dpolapp.Preclassifier.MBFinputs[4].dMBF[3][1] = 7
# Zh texture MBF trapetzoid for precipitation.
dpolapp.Preclassifier.MBFinputs[4].dMBF[0][2] = 0
dpolapp.Preclassifier.MBFinputs[4].dMBF[1][2] = 0.5
dpolapp.Preclassifier.MBFinputs[4].dMBF[2][2] = 3
dpolapp.Preclassifier.MBFinputs[4].dMBF[3][2] = 6
# Other flags: none.

#
============================================================
=========
# Detailed configuration of the CSU hydrometeor classifier.

# Fixed features:
# MeteoClassifiers has 5 inputs (Zh, ML, Zdr, Kdp, RhoHV).
# There are five application classes: RAIN, WET SNOW, SNOW,
GRAUPEL/SMALL HAIL, HAIL.
# MeteoClassifiers considers hail and rain mixtures, and
assings likely mixtures as HAIL.# MeteoClassifiers use the
CSU sum/product hybrid as the rule strength (RS), the class
```

```
result is MAX(RS).
# RS class weights: Zh:1, ML:1(summer i.e. ML>0)/
0.333(winter, ML<0)
#  Zdr: 0.5(ML>0)/0.2333(ML<0), Kdp: 0.25(ML>0)/0.1(ML<0),
Rhv 0.25(ML>0)/0.333(ML<0).
# The results are formatted as METEO_CLASSIFIER in the bits
0,1,2 starting from LSB=0 of DB_HCLASS(2).

# MeteoClassifiers considers the Preclassifier(JPOLE)
'meteo' bins, only. Other input bins are 'NoMet'
# MeteoClassifiers will redirect the 'NoMet' and
unclassified bins as:
# 0: thresholded or 1: 'NoMet' or any other class of
MeteoClassifiers.
dpolapp.MeteoClassifiers.uiNonMeteoID = 1
# Class specific thresholds on rule strengths
# Recommend 0.0 i.e. all JPOLE 'meteo' bins are assigned to
a CSU class
dpolapp.MeteoClassifiers.dMinRS[0] = 0
dpolapp.MeteoClassifiers.dMinRS[1] = 0
dpolapp.MeteoClassifiers.dMinRS[2] = 0
dpolapp.MeteoClassifiers.dMinRS[3] = 0
dpolapp.MeteoClassifiers.dMinRS[4] = 0
dpolapp.MeteoClassifiers.dMinRS[5] = 0
# MBFinputs and their use:
# Reflectivity in MeteoClassifiers:
# Data type used as input: Zh.
# MBF(dBZ) is computed as the CSU beta function.
# MBF: S-band, default settings:
# Zh beta MBF for rain: central value, width, slope
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][0] = 30
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][0] = 31
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][0] = 70
# Zh beta MBF for wet snow.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][1] = 30
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][1] = 16
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][1] = 35.6
# Zh beta MBF for snow.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][2] = 36
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][2] = 81.6
# Zh beta MBF for graupel/small hail.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][3] = 41.5
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][3] = 11.5
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][3] = 36
# Zh beta MBF for large hail.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][4] = 57.5
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][4] = 13.5
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][4] = 30
# Zh beta MBF for rain+hail mixture.
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[0][5] = 57.5
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[1][5] = 13.5
dpolapp.MeteoClassifiers.MBFinputs[0].dMBF[2][5] = 30
# Altitude in MeteoClassifiers:
# The IRIS/RDA estimate of the melting level height will be
```

```
used.
# MBF(altitude;melting level) is computed as the CSU beta(ML)
function.
# MBF: S-band, default settings:
# Altitude beta MBF for rain: central value, width, slope
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][0] = -0.2
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][0] = 20
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][0] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][0] = 5
# Altitude beta MBF for wet snow.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][1] = -0.3
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][1] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][1] = 5
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][1] = 1
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][1] = 5
# Altitude beta MBF for snow.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][2] = 10
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][2] = 10.2
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][2] = 60
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][2] = 0
# Altitude beta MBF for graupel/small hail.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][3] = 10
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][3] = 12
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][3] = 60
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][3] = 0
# Altitude beta MBF for hail.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][4] = 10
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][4] = 15
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][4] = 20
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][4] = 0
# Altitude beta MBF for rain+hail mixture.
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[0][5] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[1][5] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[2][5] = 20
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[3][5] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[4][5] = 0
dpolapp.MeteoClassifiers.MBFinputs[1].dMBF[5][5] = 0
# Differential reflectivity in MeteoClassifiers:
# Input data: Zdr (adjusted with Quality offset, internally
in reingest), and Zh
# MBF(Zdr) is computed as the CSU 2D-beta function of Zh.
# MBF: S-band, default settings:
# Zdr 2D beta MBF for rain at Zh=<Zh> dB (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][0] = 1.5334
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][0] = 1.6292
```

```
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][0] = 17.5905
# Polynomial dependence of rain MBF(Zdr) on Zh-<Zh>
#  The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][0] = 30
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][0] = 31
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraints:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][0] = 0.0766
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][1] =
0.00013578
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][2] = -
1.4943E-05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][3] =
2.6655E-07
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][0][5] = 5.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][0] = 0.0438
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][1] = -0.0011
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][2] = -
8.6469E-06
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][3] =
1.1002E-06
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][4] = 0.4
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][1][5] = 8
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][0] = 0.4983
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][1] = -0.0128
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][2] = -
9.0093E-05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][3] =
1.2834E-05
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][4] = 8
dpolapp.MeteoClassifiers.MBFinputs[2].dP[0][2][5] = 30
# Zdr 2D beta MBF for wet snow at Zh=<Zh> (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][1] = 1.5
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][1] = 1.7
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][1] = 37.9
# Polynomial dependence of wet snow MBF(Zdr) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][1] = 30
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][1] = 16
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][4] = 1.1
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][0][5] = 2.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][3] = 0
```

```
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][4] = 1
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][1][5] = 2.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][4] = 5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[1][2][5] = 20
# Zdr 2D beta MBF for snow at Zh=<Zh> (central value, width,
slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][2] = 0.25
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][2] = 0.95
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][2] = 20.7
# Zdr 2D beta MBF for graupel at Zh=<Zh> (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][3] = 0.2983
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][3] = 0.9283
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][3] = 9.7482
# Polynomial dependence of graupel MBF(Zdr) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][3] = 41.5
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][3] = 13
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th orde;min/max constraints:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][0] = 0.035
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][1] = 0.0015
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][0][5] = 1
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][0] = 0.035
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][1] = 0.0015
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][4] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][1][5] = 2
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][0] = 0.3
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][1] = 0.008
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][4] = 5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[3][2][5] = 15
# Zdr weight and 2D beta MBF for hail at Zh=<Zh> (central
value, width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][4] = -0.75
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][4] = 1.45
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][4] = 32.2
# Polynomial dependence of hail MBF(Zdr) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][4] = 57.5
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][4] = 13.5
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order;min/max constraint:
```

```
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][4] = -2
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][0][5] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][4] = 0.1
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][1][5] = 1.3
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][0] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][1] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][4] = 7
dpolapp.MeteoClassifiers.MBFinputs[2].dP[4][2][5] = 17
# Zdr weight and 2D beta MBF for rain+hail mixture at Zh=<Zh>
(central value, width, slope):
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[0][5] = 0.5045
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[1][5] = 1.3545
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[2][5] = 30.0125
# Polynomial dependence of rain+hail mixture MBF(Zdr) on Zh-
<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[3][5] = 57.5
dpolapp.MeteoClassifiers.MBFinputs[2].dMBF[4][5] = 15.5
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][0] = -0.0461
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][1] = -
0.00068749
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][2] =
0.00018148
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][3] = -5E-06
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][4] = -0.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][0][5] = 1.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][0] = -0.0261
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][1] = -
0.00068749
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][2] =
0.00018148
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][3] = -5E-06
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][4] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][1][5] = 2
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][0] = -0.6002
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][1] = -0.0173
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][2] = 0.0042
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][3] = -
0.00010667
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][4] = 20
dpolapp.MeteoClassifiers.MBFinputs[2].dP[5][2][5] = 35
# Specific differential phase in MeteoClassifiers:
```

```
# Input data: Kdp, and Zh.
# MBF(Kdp) is computed as the CSU 2D-beta function of Zh.
# MBF: S-band, default settings:
# Kdp 2D beta MBF for rain at Zh=<Zh> dB (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][0] = 0.7916
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][0] = 0.8538
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][0] = 17.2646
# Polynomial dependence of rain MBF(Kdp) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[3][0] = 30
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[4][0] = 31
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][0] = 0.0011
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][1] = -
0.00071936
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][2] =
0.00015317
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][3] =
5.0804E-06
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][0][5] = 14
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][0] = 0.0132
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][1] = -
0.00043912
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][2] =
6.8089E-05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][3] =
2.6246E-06
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][4] = 0.2
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][1][5] = 8
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][0] = -0.0093
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][1] = -0.033
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][2] =
0.00090559
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][3] =
5.3507E-05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][4] = 4
dpolapp.MeteoClassifiers.MBFinputs[3].dP[0][2][5] = 55
# Kdp 2D beta MBF for wet snow at Zh=25 dB: central value,
width, slope.
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][1] = 0.5
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][1] = 0.6
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][1] = 12.6
# Kdp 2D beta MBF for snow at Zh=17.5 dB (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][2] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][2] = 0.6
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][2] = 12.6
# Kdp 2D beta MBF for graupel at Zh=37.5 dB (central value,
width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][3] = 0.2123
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][3] = 0.4873
```

```
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][3] = 9.9993
# Polynomial dependence of graupel MBF(Kdp) on Zh-<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[3][3] = 37.5
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[4][3] = 8.5
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][0] = 0.0045
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][1] =
0.00010064
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][2] =
8.6663E-05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][3] = -
1.201E-05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][4] = -0.25
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][0][5] = 1.5
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][0] = 0.0353
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][1] = -
0.00022246
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][2] = -
4.6667E-05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][3] =
4.1159E-05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][4] = 0.3
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][1][5] = 2
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][0] = 0.7825
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][1] = -0.0029
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][2] = 0.0028
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][3] =
0.00047804
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][4] = 4
dpolapp.MeteoClassifiers.MBFinputs[3].dP[3][2][5] = 20
# Kdp weight and 2D beta MBF for hail at Zh=57.5 dB (central
value, width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][4] = 0.25
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][4] = 0.85
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][4] = 18.4
# Kdp weight and 2D beta MBF for rain+hail mixture at Zh=57.5
dB (central value, width, slope):
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[0][5] = 1.8241
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[1][5] = 1.9253
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[2][5] = 26.8047
# Polynomial dependence of rain+hail mixture MBF(Kdp) on Zh-
<Zh>
# The center  <Zh> and the Zh range in which the polynomial
is applied:
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[3][5] = 57.5
dpolapp.MeteoClassifiers.MBFinputs[3].dMBF[4][5] = 15.5
# Polynomial coefficients; the linear, the 2nd, the 3rd, and
the 4th order; min/max constraint:
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][0] = 0.3156
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][1] = 0.0124
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][2] = -
0.00066667
```

```
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][3] = -3.8E-
05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][0][5] = 10
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][0] = 0.3138
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][1] = 0.0122
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][2] = -
0.00062963
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][3] = -
3.4667E-05
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][4] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][1][5] = 10
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][0] = 2.6738
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][1] = 0.0805
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][2] = -2
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][3] = 0
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][4] = 4
dpolapp.MeteoClassifiers.MBFinputs[3].dP[5][2][5] = 70
# Cross correlation coefficient in MeteoClassifiers:
#--- Data type used as input: RhoHV.
#--- MBF(Rhohv) is computed as the CSU beta function.
# # MBF:  S-band, default settings:
# RHOHV beta MBF for rain: central value, width, slope
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][0] = 1
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][0] = 0.04
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][0] = 10
# RHOHV beta MBF for wet snow.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][1] = 0.88
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][1] = 0.11
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][1] = 20
# RHOHV beta MBF for snow.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][2] = 1
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][2] = 0.06
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][2] = 10
# RHOHV beta MBF for graupel.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][3] = 0.96
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][3] = 0.04
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][3] = 10
# RHOHV beta MBF for hail.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][4] = 0.9
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][4] = 0.045
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][4] = 10
# RHOHV beta MBF for rain+hail mixture.
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[0][5] = 0.8
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[1][5] = 0.13
dpolapp.MeteoClassifiers.MBFinputs[4].dMBF[2][5] = 30

#
=========================================================
=========
# Detailed configuration of the JPOLE rain classifier.

# Fixed features:
# Precip classifier has 5 input observables: Zh, Zdr, RhoHV,
Zh-texture, Phidp-texture.
```

```
# There are 4 output classes: LIGHT, MODERATE and HEAVY
PRECIPITATION, and LARGE DROPS
# The rule strength function RS is sum of MBF(i) and the class
assignment is max RS.
# The results are formatted as PRECIP_CLASSIFIER in the bits
3,4,5 starting from LSB=0.

# Precip classifier considers the Preclassifier(JPOLE)
'meteo' bins, only.
# Precip classifier will leave other than 'meteo' bins
unchanged, and redirect new unclassified outputs as:
# 0: thresholded or 1: 'GC/AP' or any class of precipClasses.
dpolapp.PrecipClassifier.uiNonMeteoID = 0
# Class specific thresholds on rule strengths for a class to
be accepted.
dpolapp.PrecipClassifier.dMinRS[0] = 0
dpolapp.PrecipClassifier.dMinRS[1] = 0
dpolapp.PrecipClassifier.dMinRS[2] = 0
dpolapp.PrecipClassifier.dMinRS[3] = 0

# Precip classifier uses the textures of reflectivity and
differential phase.

# MBFinputs and their use:
# Reflectivity in rain classifier:
# Input data: Zh.
# MBF(Zh) is computed as JPOLE trapetzoid; an additive factor
in rule strength
# MBF: default initialization with parameter settings of
Table 2 Ref.2
# Zh MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[0][0] = 5
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[1][0] = 10
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[2][0] = 35
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[3][0] = 40
# Zh MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[0][1] = 30
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[1][1] = 35
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[2][1] = 45
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[3][1] = 50
# Zh MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[0][2] = 40
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[1][2] = 45
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[2][2] = 75
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[3][2] = 80
# Zh MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[0][3] = 15
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[1][3] = 20
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[2][3] = 45
dpolapp.PrecipClassifier.MBFinputs[0].dMBF[3][3] = 50
# Differential reflectivity in rain classifier:
# Input data: Zdr (In reingest: adjusted with Quality offset,
internally), and Zh.
# MBF(Zdr;Zh) is computed as JPOLE 2D-trapetzoid; additive
in rule strength
```

```
# Zdr MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[0][0] = -0.3
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[1][0] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[2][0] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[3][0] = 0.3
# The polynomial dependence of the Zdr light precipitation
trapezoid on Zh
# The Zh range in which the polynomial is applied:
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[4][0] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[5][0] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][0][0] = -0.5
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][0][1] = 0.0025
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][0][2] = 0.00075
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][2][0] = 0.08
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][2][1] = 0.0364
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][2][2] =
0.000357
dpolapp.PrecipClassifier.MBFinputs[1].dP[0][2][3] = 0
# Zdr MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[0][1] = -0.3
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[1][1] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[2][1] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[3][1] = 0.3
# The polynomial dependence of the Zdr moderate precipitation
trapezoid on Zh
# The Zh range in which the polynomial is applied:
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[4][1] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[5][1] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][0][0] = -0.5
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][0][1] = 0.0025
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][0][2] = 0.00075
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][2][0] = 0.08
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][2][1] = 0.0364
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][2][2] =
0.000357
dpolapp.PrecipClassifier.MBFinputs[1].dP[1][2][3] = 0
# Zdr MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[0][2] = -0.3
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[1][2] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[2][2] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[3][2] = 0.3
```

```
# The polynomial dependence of the Zdr heavy precipitation
trapezoid on Zh
# The Zh range in which the polynomial is applied:
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[4][2] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[5][2] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][0][0] = -0.5
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][0][1] = 0.0025
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][0][2] = 0.00075
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][2][0] = 0.08
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][2][1] = 0.0364
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][2][2] =
0.000357
dpolapp.PrecipClassifier.MBFinputs[1].dP[2][2][3] = 0
# Zdr MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[0][3] = -0.3
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[1][3] = 0

dpolapp.PrecipClassifier.MBFinputs[1].dMBF[2][3] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[3][3] = 0.3
# The polynomial dependence of the Zdr large drops trapezoid
on Zh
# The Zh range in which the polynomial is applied:
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[4][3] = 0
dpolapp.PrecipClassifier.MBFinputs[1].dMBF[5][3] = 80
# Dependency on Zh: left tail constant term P0, linear P1,
2nd order P2, and 3rd order P3.
# The left shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][0][0] = -0.5
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][0][1] = 0.0025
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][0][2] = 0.00075
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][0][3] = 0
# The right tail constant term P0, linear P1, 2nd order P2
and 3rd order P3
# The right shoulder is the same.
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][2][0] = 0.08
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][2][1] = 0.0364
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][2][2] =
0.000357
dpolapp.PrecipClassifier.MBFinputs[1].dP[3][2][3] = 0
# Cross correlation coefficient in rain classifier:
# Data type used as input: RhoHV.
# MBF(RhoHV) is computed as JPOLE trapetzoid;  an additive
factor in rule strength
# MBF: default initialization with parameter settings of
Table 2 Ref.1
# RHOhv MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[0][0] = 0.85
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[1][0] = 0.97
```

```
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[2][0] = 1
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[3][0] = 1.01
# RHOhv MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[0][1] = 0.85
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[1][1] = 0.97
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[2][1] = 1
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[3][1] = 1.01
# RHOhv MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[0][2] = 0.85
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[1][2] = 0.97
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[2][2] = 1
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[3][2] = 1.01
# RHOhv MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[0][3] = 0.85
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[1][3] = 0.97
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[2][3] = 1
dpolapp.PrecipClassifier.MBFinputs[2].dMBF[3][3] = 1.01
# Differential phase texture in rain classifier:
# The input data type is computed internally.
#--- MBF(Texture-1) is computed as JPOLE trapetzoid;  an
additive factor in rule strength
# MBF: default initialization with parameter settings of
Table 2 Ref.1
# PHIDP texture MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[0][0] = 0
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[1][0] = 1
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[2][0] = 15
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[3][0] = 30
# PHIDP texture MBF trapetzoid for moderate precipitation.
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[0][1] = 0
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[1][1] = 1
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[2][1] = 15
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[3][1] = 30
# PHIDP texture MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[0][2] = 0
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[1][2] = 1
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[2][2] = 15
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[3][2] = 30
# PHIDP texture MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[0][3] = 0
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[1][3] = 1
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[2][3] = 15
dpolapp.PrecipClassifier.MBFinputs[3].dMBF[3][3] = 30
# Reflectivity texture in rain classifier:
# Data type is computed internally.
# MBF(Texture-2) is computed as JPOLE trapetzoid; an additive
factor in the rule strength
# MBF: default initialization with parameter settings of
Table 2 Ref.1
# Zh texture MBF trapetzoid for light precipitation.
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[0][0] = 0
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[1][0] = 0.5
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[2][0] = 3
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[3][0] = 6
# Zh texture MBF trapetzoid for moderate precipitation.
```

```
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[0][1] = 0
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[1][1] = 0.5
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[2][1] = 3
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[3][1] = 6
# Zh texture MBF trapetzoid for heavy precipitation.
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[0][2] = 0
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[1][2] = 0.5
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[2][2] = 3
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[3][2] = 6
# Zh texture MBF trapetzoid for large drops.
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[0][3] = 0
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[1][3] = 0.5
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[2][3] = 3
dpolapp.PrecipClassifier.MBFinputs[4].dMBF[3][3] = 6
# Other flags: none.

#
============================================================
=========
# Detailed configuration of the classifier of stratiform vs.
convective precipitation.

# Fixed features:
# Convective check considers the pair of Zh and height w.r.t
0oC isotherm.
# The rule strength function RS is a column sum of MBFs
# The results are formatted as CELL_CLASSIFIER in the bits
6-7 starting from LSB=0.

# Cell classifier considers the Preclassifier(JPOLE) 'meteo'
bins, only. Other bins are set thresholded.
# Rule strength threshold on the CONVECTIVE class.
dpolapp.CellClassifier.dMinRS[0] = 0.5

# Cell classifier considers columns in the volume data of
multiple sweeps.

# MBFinputs and their use:
# Reflectivity&Height in cell classifier:
# Input data: Zh and height difference w.r.t. the current 0oC
isotherm.
# MBF(Zh,height) is computed as 2D product of trapetzoids;
summed up to rule strength.
# Min Zh required for convection (full fuzzy strength will
be +5 dBZ from min).
dpolapp.CellClassifier.MBFinputs[0].dMBF[0][0] = 35
# Min height w.r.t. 0oC isotherm (full fuzzy strength will
be +1. km higher ).
dpolapp.CellClassifier.MBFinputs[0].dMBF[4][0] = 0
```

4.5.2 Parametrized Fuzzy Parameters

The hydrometeor classification membership functions are parametrized either as 1D or 2D beta functions. Table 7. Parameters for 1D and 2D Membership Functions at Specified Reference Planes of the Second Function Variable on page 123 contains the beta function parameters of all the 1D membership functions. Table 7. Parameters for 1D and 2D Membership Functions at Specified Reference Planes of the Second Function Variable on page 123 also gives insight into the 2D membership functions, expressing their projections at the reference levels of the auxiliary input variables. For example, the parameter values of 2D membership functions of MBF(Z_{dr}, Z_H) visualize the shape of the MBF at Z_H = 0 dBZ.

The evolution of the 2D membership beta function parameters y=y(Z_H) where y=m, a, or b were optimized using the CSU approach. The parameter dependencies are expressed as 5th order polynomials, with coefficients that are listed in Table 8. Parametrized Evolution of the Parameter m of the 2D Membership Functions on page 125 through Table 10. Parametrized Evolution of the Parameter b of the 2D Membership Functions on page 126. Parametrizing 2D membership function with polynomials simplifies the implemented software, as the approach is common to CSU and the JPOLE algorithms.

Table 7 Parameters for 1D and 2D Membership Functions at Specified Reference Planes of the Second Function Variable

MBF	Hydrometeor Class ID	m	a	b
MBF(Z_H [dB]) 1D				
	1) rain	30.0	31.0	40.0
	2) wet snow	25.0	21.0	40.0
	3) snow	0.0	36.0	40.0
	4) graupel/small hail	45.0	11.0	20.0
	5) hail	57.5	14.0	20.0
	6) hail, rain mixture	60.0	11.0	20.0
MBF(height [km]) 2D @ melting level height (MSL) = 2.5 km (warm season)				
	1) rain	0.0	2.3	5.0
	2) wet snow	2.2	0.5	5.0
	3) snow	10.0	7.7	60.0
	4) graupel/small hail	10.0	9.5	60.0
	5) hail	0.0	15.0	20.0
	6) hail, rain mixture	0.0	3.0	20.0
MBF(height [km]) 1D @ melting level height < 0 km (cold season, storm height =5 km)				
	1) rain	0.0	0.5	5.0

Table 7 Parameters for 1D and 2D Membership Functions at Specified Reference Planes of the Second Function Variable

MBF	Hydrometeor Class ID	m	a	b
	2) wet snow	0.0	2.5	5.0
	3) snow	10.0	15.0	20.0
	4) graupel/small hail	10.0	15.0	20.0
	5) hail	10.0	15.0	20.0
	6) hail, rain mixture	10.0	15.0	20.0
MBF(Z_{dr} [dB]) 2D				
@ Z_H = 30 dBZ	1) rain	0.5	0.7	10.0
@ Z_H = 27.5 dBZ	2) wet snow	1.1	1.4	15.0
@ Z_H = 0 dBZ	3) snow	0.5	0.75	8.0
@ Z_H = 42.5 dBZ	4) graupel/small hail	0.43	0.9.	12.5
@ Z_H = 7.5 dBZ	5) hail	-0.76	1.28	15.9
@ Z_H = 60.0 dBZ	6) hail, rain mixture	0.94	2.0	15.0
MBF(K_{dp} [dgr/km]) 2D				
@ Z_H = 30 dBZ	1) rain	0.05	0.2	11.1
@ Z_H = 25 dBZ	2) wet snow	0.25	1.2	10.0
@ Z_H = 7.5 dBZ	3) snow	0.0	0.25	10.0
@ Z_H = 42.5 dBZ	4) graupel/small hail	0.27	0.77	12.5
@ Z_H = 57.5 dBZ	5) hail	0.5	1.0	10.0
@ Z_H = 60.0 dBZ	6) hail, rain mixture	2.37	2.40	15.0
MBF (Rho_{HV} []) 1D				
	1) rain	1.00	0.04	10.0
	2) wet snow	0.88	0.11	20.0
	3) snow	1.0	0.06	10.0
	4) graupel/small hail	0.96	0.04	10.0
	5) hail	0.90	0.045	10.0
	6) hail, rain mixture	0.80	0.13	30.0

NOTE

In Table 8. Parametrized Evolution of the Parameter m of the 2D Membership Functions on page 125 through Table 10. Parametrized Evolution of the Parameter b of the 2D Membership Functions on page 126: parametrization is expressed with 5th order polynomials of the difference with respect to the reference planeo of the quoted auxiliary observable. The polynomial coefficients P0 to P4 are in increasing order, P0 values are in Table 7. Parameters for 1D and 2D Membership Functions at Specified Reference Planes of the Second Function Variable on page 123 (constant terms at the reference plane). The **Max/Min** column expresses the boundaries that the parametrized polynomial values must fall within.

Table 8 Parametrized Evolution of the Parameter m of the 2D Membership Functions

m	ID	P1	P2	P3	P4	Max/Min
MBF(height [km]) (warm season)						
-	1)	0.0	0	0	0	-
-	2)	0.0	0	0	0	-
ML-2.5	3)	1.0	0	0	0	-
-	4)	0.0	0	0	0	-
-	5)	0.0	0	0	0	-
-	6)	0.0	0	0	0	-
MBF(Z_{dr} [dB])						
(Z_H-30) ±30 dBZ	1)	4.75E-02	1.76E-03	6.99E-06-	4.50E-07	5.5/0.1
(Z_H-27.5) ±17.5 dBZ	2)	4.62E-03	6.42E-04	2.22E-05	0.0000E+0	02.5/1.1
(Z_H-0) ±327 dBZ	3)	0	0	0	0	0.5/-0.5
(Z_H-42.5) ±12.5 dBZ	4)	3.37E-02	9.64E-04	0	0	1.5/0.155
(Z_H-57.5) ±12.5 dBZ	5)	-1.73E-02	1.51E-03	-8.33E-05	4.17E-06	5.5/0.1
(Z_H- 60) ±10 dBZ	6)	-5.41E-02	-5.71E-03	1.67E-04	0	5.5/0.1
MBF(K_{dp} [dgr/km])						
(Z_H-30) ±35 dBZ	1)	-4.96E-03	2.06E-03	1.91E-04	3.99E-06	14/0.05
(Z_H-27.5) ±17.5 dBZ	2)	0	0	0	0	0.25/0.25
(Z_H-0) ±327 dBZ	3)	0	0	0	0	0/0
(Z_H-42.5) ±12.5 dBZ	4)	6.87E-02	1.00E-03	-1.48E-040	0	2.0/0.3
(Z_H-57.5) ±12.5 dBZ	5)	0	0	0	0	0.5/0.5
(Z_H- 60) ±10 dBZ	6)	3.08E-01	8.57E-03	3.33E-04	0	10./0.0

Table 9 Parametrized Evolution of the Parameter a of the 2D Membership Functions

a	ID	P1	P2	P3	P4	Max/Min
MBF(height [km]) (warm season)						
-	1)	1.0	0	0	0	-
-	2)	-1.0	0	0	0	-
ML-2.5	3)	0.0	0	0	0	-
-	4)	-1.0	0	0	0	-
-	5)	0.0	0	0	0	-
-	6)	1.0	0	0	0	-
MBF(Z_{dr} [dB])						
(Z_H-30) ±30 dBZ	1)	2.59E-02	4.36E-04	-4.89E-06	-5.62E-08	8/0.4
(Z_H-27.5) ±17.5 dBZ	2)	-9.11E-03	2.1429E-04	3.84E-05	0.0000E+00	2.5/1.1
(Z_H-0) ±327 dBZ	3)	0	0	0	0	0.5/-0.5
(Z_H-42.5) ±12.5 dBZ	4)	3.37E-02	9.64E-04	0	0	2.0/0.5
(Z_H-57.5) ±12.5 dBZ	5)	5.78E-03	4.53E-03	2.78E-05	1.25E-05	0/-2
(Z_H- 60) ±10 dBZ	6)	4.17E-03	-2.38E-02	-1.67E-04	1.50E-04	3.0/0.5

Table 9 Parametrized Evolution of the Parameter a of the 2D Membership Functions

a	ID	P1	P2	P3	P4	Max/Min
MBF(K_{dp} [dgr/km])						
(Z_H-30) ±35 dBZ	1)	1.11E-02	2.95E-03	1.04E-04	6.11E-07	8/0.2
(Z_H-27.5) ±17.5 dBZ	2)	0	0	0	0	0.4/0.4
(Z_H-0) ±327 dBZ	3)	0	0	0	0	0.25/0.25
(Z_H-42.5) ±12.5 dBZ	4)	6.87E-02	1.00E-03	-1.48E-04	0	2.0/0.3
(Z_H-57.5) ±12.5 dBZ	5)	0	0	0	0	1/1
(Z_H- 60) ±10 dBZ	6)	2.97E-01	8.86E-03	2.67E-04	0	10./0.1

Table 10 Parametrized Evolution of the Parameter b of the 2D Membership Functions

b	ID	P1	P2	P3	P4	Max/Min
MBF(height [km]) (warm season)						
-	1)	0	0	0	0	-
-	2)	0	0	0	0	-
ML-2.5	3)	0	0	0	0	-
-	4)	0	0	0	0	-
-	5)	0	0	0	0	-
-	6)	0	0	0	0	-
MBF(Z_{dr} [dB])						
(Z_H-30) ±30 dBZ	1)	2.29E-01	5.49E-03	-5.83E-05	0	25/8
(Z_H-27.5) ±17.5 dBZ	2)	0	0	0	0	20/5
(Z_H-0) ±327 dBZ	3)	0	0	0	0	8/8
(Z_H-42.5) ±12.5 dBZ	4)	0	0	0	0	12.5/12.5
(Z_H-57.5) ±12.5 dBZ	5)	2.103E-18-	1.42E-01	2.47E-19	6.67E-04	17/7
(Z_H- 60) ±10 dBZ	6)	-5.92E-17	1.67E-02	5.92E-19	-6.67E-04	20/5
MBF(K_{dp} [dgr/km])						
(Z_H-30) ±35 dBZ	1)	2.29E-01	5.49E-03	-5.83E-05	0	16./9.
(Z_H-27.5) ±17.5 dBZ	2)	0	0	0	0	10/10
(Z_H-0) ±327 dBZ	3)	0	0	0	0	10/10
(Z_H-42.5) ±12.5 dBZ	4)	2.57E-01	0	0	0	17.0/9
(Z_H-57.5) ±12.5 dBZ	5)	0	0	0	0	10/10
(Z_H- 60) ±10 dBZ	6)	-1.67E-01	2.44E-17	6.67E-03	0	25/8.0

CHAPTER 5

WIDE DYNAMIC RANGE

5.1 Overview

This chapter describes the operational theory (architecture and algorithms) of the RVP900 wide dynamic range (WDR) and demonstrates its performance in a lab test bench environment.

5.2 Introduction

A new design has been integrated into the RVP900 to extend its dynamic range. Compared to the standard receiver mode, WDR for dual-polarization and single-polarization has extended the RVP900 dynamic range so that it is similar to the existing extended mode.

NOTE	Extended mode is for single-polarization only and is implemented in the Linux PC by processing received raw pulse through Ethernet.

5.3 Dynamic Range in General

In sensing and signal processing, the need for a WDR is driven by the ratio of strong signals to weak signals that correspond to the dynamic range of the input signal itself. The actual achievable dynamic range is often limited by some bottleneck parts of the processing chain, such as the CCD used in digital imaging. To remove or alleviate the problem, the bottleneck part either can be time-split shared or duplicated with a specific gain-gain separation, and then synthesized as used in image processing[36,38], shown in Figure 27 on page 128.

(a) Low Gain (Short Exposure) (b) High Gain (Long Exposure) (c) Synthesized

Figure 27 Illustration of Dynamic Range Extension in General Digital Processing

The system can be viewed as having two (virtual) channels. The actual dynamic range of this gain separation scheme, or automatic gain control (AGC)[10,11], depends on the dynamic range of the bottleneck itself, the separation gain, and the overlapped region (shown in Figure 28 on page 129). Within a certain dynamic range of the bottleneck component, increasing the ***gain separation*** increases the overall dynamic range, while decreasing the ***overlap region*** between the two channels. Decreasing the ***gain separation*** increases the ***overlap region***, while decreasing the overall dynamic range. The minimum gain separation is the difference between the overall dynamic range of the system and that of a single channel, at which the overlap region is maximized. The maximum gain separation is the dynamic range of a single channel, at which the overlap is minimized as 0. One important factor, which needs to be considered, is that a considerably wide overlap region is needed to get useful synthesis channel characteristics.

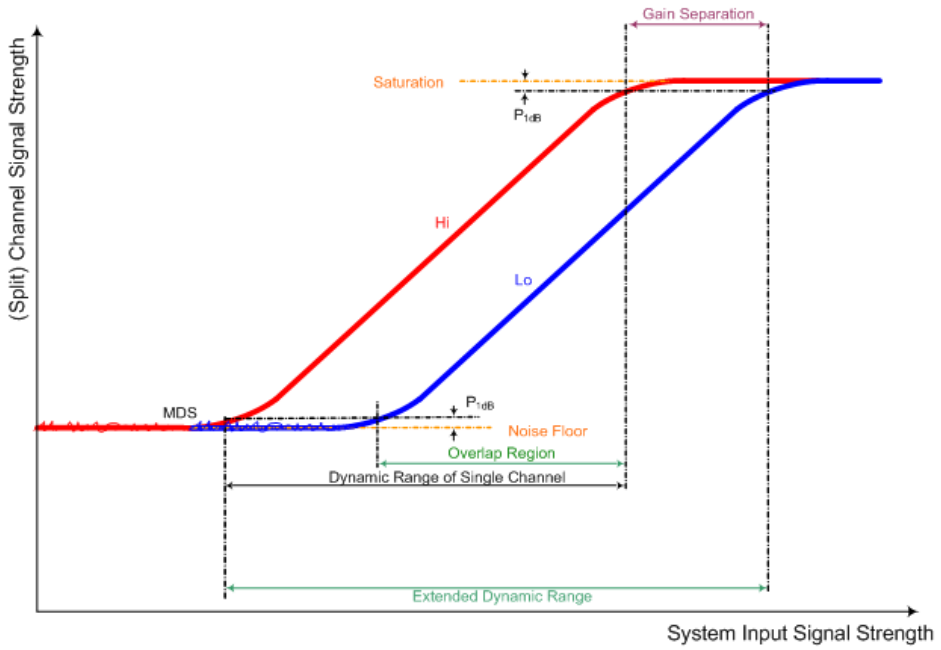


Figure 28 Illustration of Dynamic Range, Channel Overlap and Separation

NOTE	In the current RVP documentation, 1dB compression is used for both saturation and noise floor to calculate dynamic range. Normalized gain as unity is assumed.

More than just the gain separation needs to be considered in RF/IF dynamic range extension. The most significant one is the changing delays between channels due to factors, such as temperature (shown in Figure 29 on page 130), aging, skin effect[37], and RF components. In a lab environment, a 20 dB attenuator (1 inch in length) can cause a delay of about 0.3 nS, with higher ADC sampling frequency of 100 MHZ. It is equivalent to a 0.03 ADC sample delay, or about 6 degree phase shift of a 60 MHZ IF signal. The RF components may cause phase delays anywhere between −180 to +180 degrees. This delay causes strong noise (residue power) in the overall system (synthesized) output, if it not monitored and compensated properly.

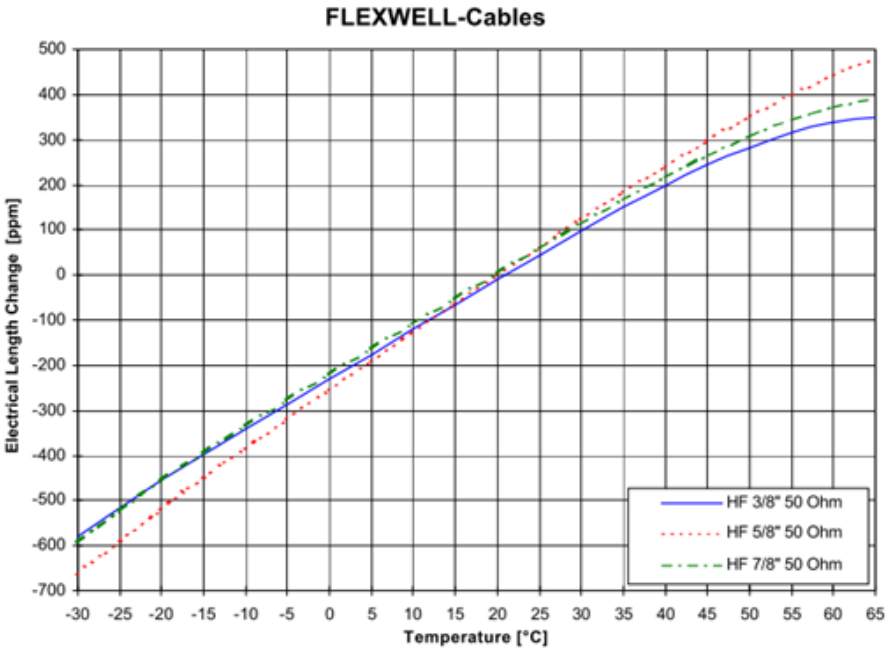


Figure 29 Example of RF Cable Electrical Length and Temperature

5.4 Factors of Current RVP Dynamic Range

For radar to detect weak signals in the presence of strong signal or interference (such as clutters), it must have both high SNR levels and wide dynamic range.

A receiver signal processing chain generally consists of a front-end, low-noise-amplifier (LNA) and IF digitizer (IFD). The dynamic range of the IFD includes the wideband SNR of the ADC and processing gain (oversampling):

$$DR_{IFD} = 20 \cdot log_{10} 2^N + 20 \cdot log_{10} \sqrt{\tfrac{3}{2}} + 10 \cdot log_{10} \frac{f_{ADC}/2}{BW}$$

Figure 30 Equation 1

where N is the number of ADC bits, $fADC$ is the sample frequency, and BW the bandwidth of the FIR filter. In practice, the 16-bit ADC chip used in RVP900 has only 79 dB SNR (refer to the *RVP900 User's Manual - Digital Receiver and Signal Processor*). Assuming 72 MHZ sampling frequency and 1 MHZ bandwidth, it offers only 95 dB dynamic range. Although some techniques, such as statistical linearization (enhancement) of ADC signals that exceed saturation level, extract more dynamic range

of coherent signal below 0 dB SNR, each add 4 dB to the standard RVP receiver mode. A 103 dB normal dynamic range (NDR) may be still less than the front-end can deliver (refer to the *RVP900 User's Manual - Digital Receiver and Signal Processor*).

To alleviate the bottleneck IFD performance to the system dynamic range, an extended dynamic range receiver mode was introduced (which only works for single-polarization radar). Two channels with specified gain separation working in parallel and the filtered time series data from the downconverter are transferred out of the IFD, and are downstreamed to a Linux PC to extract relative gain and phase delay. The gain-phase information is used to correct the FIR filter, before merging the two channels together to form the final dynamic-range extended (single pol) channel. Due to the limitation of the communication bandwidth limit, this mode is limited for a single-pol radar receiver only. Its dynamic range was documented in the *RVP900 User's Manual - Digital Receiver and Signal Processor* as 106 dB (see Figure 31 on page 131) using the zauto utility.

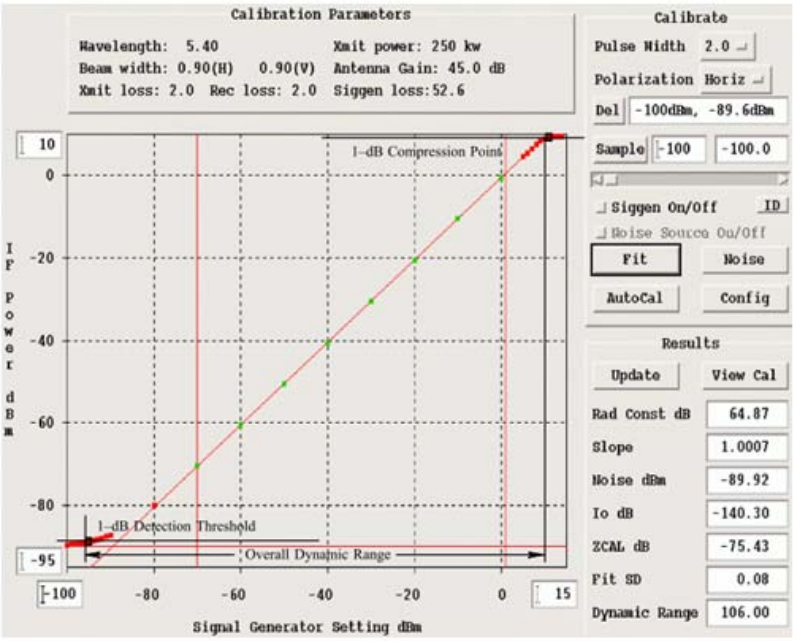


Figure 31 Dynamic Range of (Legacy) Single-Polarization Extended Receiver Mode

5.5 Dual-Polarization RVP Dynamic Range Expansion

To implement dynamic range expansion without using a back-end PC to process time series data received from limited bandwidth Ethernet, the

extension logic must be implemented in the IFD for either single-polarization or dual-polarization radar, as shown in Figure 32 on page 132.

or

WDR is an extension of NDR of the RVP900, as shown in Figure 32 on page 132.

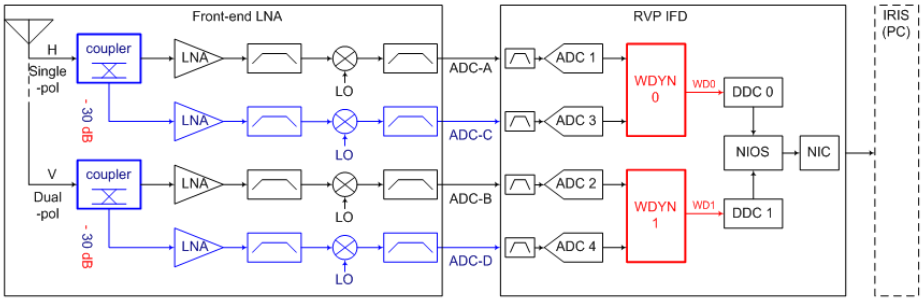


Figure 32 Simplified System Block Diagram of Dynamic Range Extension of RVP IFD for Single-Polarization and Dual-Polarization Radar

Each of the front-end input signals is split into two channels through a coupler: a high gain channel (Hi) and a low gain channel (Lo). The gain separation is hardwired (fixed) at about 30 dB. Considering the standard receiver channel dynamic range can be 106 dB, with targeted overall dynamic range of 120 dB, a 20~30 dB separation is a reasonable choice. The signal goes through a LNA, bandpass filter, front-end downconverter, and another bandpass filter to feed the RVP IFD ADC input.

5.5.1 IFD ADC Terminal Definitions

For single-pol wide-dynamic range (WDR):

- ADC-A as High (Hi) gain channel, ADC-C as Low (Lo) gain channel

For dual-polarization WDR:

- ADC-A as High (Hi) gain channel, ADC-C as Low (Lo) gain channel of H-polarization
- ADC-B as High (Hi) gain channel, ADC-D as Low (Lo) gain channel of V-polarization

The definition of terminals for legacy single-polarization only WDR:

- ADC-A as High (Hi) gain channel, ADC-B as Low (Lo) gain channel

Other traditional NDR operation:

- ADC-A for single-polarization
- ADC-A as H and ADC-B as V for dual-polarization

The ADC input is protected through a TVS and passes through an anti-aliasing bandpass filter before the downconverter. ADC samples pass through a multiplexer (not shown in Figure 32 on page 132). To be compatible with a standard receiver mode, the ADC outputs may be routed to digital downconverter (DDC) or passed through a newly added WDR extension block. The direct ADC outputs and the WDR outputs can still be viewed in the dspx utility as monitor samples (Ch1~Ch4, WD0, WD1). Limited by the current FPGA DSP block, the WDR outputs and the downstream DDC can only be changed from 16 bits to 18 bits for the standard receiver mode; a potential limitation factor to the maximum dynamic range achievable. The DDC outputs feed the IRIS process running on a PC through the IFD on-board NIOS microprocessor and the Ethernet network interface (NIC).

5.6 Architecture

Details of the one-channel (single-polarization) WDR logic in Figure 32 on page 132 are redrawn in Figure 33 on page 133. The ADC input of the IFD passes through the signal strength (saturated or weak signal) monitor logic at the same time it passes through the 7-tap Fractional Delay Filters (FDF), switching logic between Hi and Lo channels. It then passes through the gain and delay detection monitor logic in IFD, communicating through Ethernet controlled by NIOS controller, to a PC running periodic channel monitoring/update.

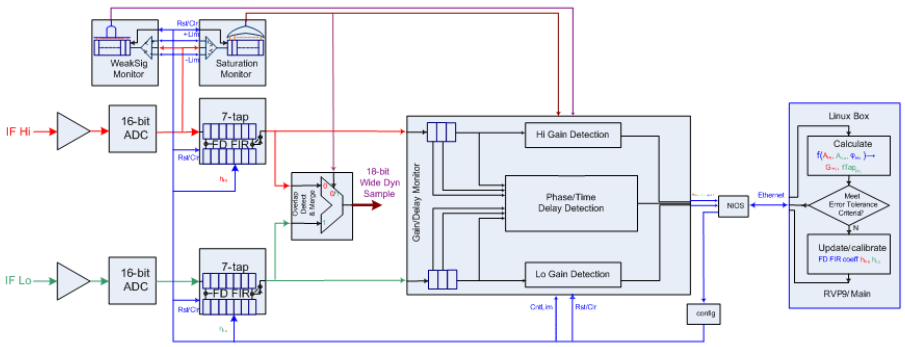


Figure 33 Basic Architecture of Single-Channel Wide Dynamic Range Extension

5.7 Algorithms

The new extended dynamic range receiver mode can be viewed as a feedback control loop: statistical channel status are monitored in FGA, update decision is made in the Linux PC to relieve FPGA from expensive computing and the FDF refresh coefficients are fed back to the IFD. Hence the algorithms can be categorized as two parts: status monitoring and system updating, which are implemented together by IFD's FPGA and Linux box's watchdog thread.

5.7.1 ADC DC/Zero Drift Detection

For Hi/Lo relative gain and delay detection, pure AC signals will be assumed. But the on-board ADC actually generates a DC-shifted AC due to both the ADC chip itself and front-end analog channel drifting:

$$H_{adc}(i) = H_{dc} + H(i);\ \ L_{adc}(i) + L_{dc} + L(i)$$

Figure 34 Equation 2

where H_{adc} and L_{adc} are H_{dc} and L_{dc} shifted pure AC signal sample H and L at i-th ADC clock for Hi and Lo channel respectively. Due to the nature of the drift, we assume the DC level can be viewed as a virtual constant during the operation time (a watchdog cycle, default 100 mS).

By accumulating enough non-saturated samples in the FPGA and averaging in Linux watchdog thread, the actual DC drift can be calculated as:

$$H_{dc} = \sum_{i=1}^{N_{drift}} H_{adc}(i) \Big/ N_{drift} \ ;\ \ L_{dc} = \sum_{i=1}^{N_{drift}} L_{adc} \Big/ N_{drift} \ ;\ \ N_{drift} \le NMAX_{drift}$$

Figure 35 Equation 3

where N_{drift} is the actual number of samples accumulated, which should always be less than or equal to the maximum number configured $NMAX_{drift}$.

5.7.2 Relative Gain Detection

Although relative gain separation is a fixed configuration for a specific design, the gain itself and the separation can be drifted due to component replacement (with different tolerance) and temperature drift. By accumulating enough samples in the FPGA, the actual signal $_{RMS}$ and amplitude can be obtained:

$$A_h = \sqrt{2}H_{RMS} = \sqrt{2\sum_{i=1}^{N_{gfd}} H^2(i) \Big/ N_{gfd}} \; ; \; A_L = \sqrt{2}L_{RMS} = \sqrt{2\sum_{i=1}^{N_{gfd}} L^2(i) \Big/ N_{gfd}} \; ; \; N_{drift} \quad NMAX_{gfd}$$

Figure 36 Equation 4

where A_H and A_L is the amplitude, H_{RMS} and L_{RMS} the $_{RMS}$ value, $\sum H^2$ and $\sum L^2$ the accumulated square value of Hi and Lo channel (AC) signal respectively, and N_{gfd} is the number of samples accumulated in a "good" overlap region of Hi and Lo channels, which should always be less than or equal to the maximum total number configured $NMAX_{gfd}$.

Since the signals are DC drifted, the monitored squared terms should be corrected before valid gain separation can be calculated. From Figure 3 on page 44:

$$H^2_{adc}(i) = (H_{dc} + H(i))^2 \approx H^2_{dc} + H^2(i), \text{ hence}$$
$$H^2(i) = H^2_{adc}(i) - H^2_{dc}; \; L^2(i) = L^2_{adc}(i) - L^2_{dc}$$

Figure 37 Equation 5

where the cross product of DC and AC is statistically insignificant (cancelled positive and negative value when accumulated) when enough samples accumulated. The relative gain separation ratio is:

$$gainHL = A_H / A_L = \sqrt{\sum_{i=1}^{N_{gfd}} H^2(i) \Big/ \sum_{i=1}^{N_{gfd}} L^2(i)}$$

$$= \frac{\sqrt{\sum_{i=1}^{N_{gfd}} H_{adc}^2(i) - \left[\sum_{i=1}^{N_{drift}} H_{adc}(i) / N_{drift}\right]^2 N_{gfd}}}{\sqrt{\sum_{i=1}^{N_{gfd}} L_{adc}^2(i) - \left[\sum_{i=1}^{N_{drift}} L_{adc}(i) / N_{drift}\right]^2 N_{gfd}}}$$

Figure 38 Equation 6

which can also be quantified as separation dB: *20log10*(*gainHL*) to facilitate user configuration and status monitor/display.

5.7.3 Relative Delay Detection

The Hi and Lo channels may have different delays due to component mismatch, aging factor, and temperature coefficient especially in RF modules. The relative phase delay will cause spiking noise during transition between the channels, so it must be monitored and corrected.

5.7.3.1 Basic Delay Detection (±90 degree range)

Using 1-lag cross correlation of the two channel samples, $H(i)$ and $L(i)$ at the i-th ADC clock and the samples at previous clock $H(i-1)$ and $L(i-1)$, the relative delay of Hi channel to Lo channel can be expressed as a fractional ADC sample *fdHL*:

$$fdHL = 1/(2\pi f_N)\ a\sin\left[\frac{\sum_{i=2}^{N_{gfd}} (H(i-1)L(i) - H(i)L(i-1))}{2\sqrt{\sum_{i=1}^{N_{gfd}} H^2(i)}\ \sqrt{\sum_{i=1}^{N_{gfd}} L^2(i)}\ \sin(2\pi f_N)}\right]$$

Figure 39 Equation 7

where f_N is the normalized IF center frequency (f_{IFc}/f_{ADC}, that is, over ADC sample frequency, with boundary 0~0.5), the numerator inside the *a*sin function is the accumulated correlation of Hi and Lo channel AC samples, and the root square terms in the denominators are related to the amplitude of Hi and Lo channel AC signals.

DC drift has no effect statistically to the cross correlation term in Equation 7 on page 136, but the magnitude terms are affected as in Equation 6 on page 136. Considering DC drift, the fractional delay can be corrected as:

$$fdHL = 1/(2\pi f_N)\ a\sin\left[\frac{\sum_{i=2}^{N_{gfd}}(H_{adc}(i-1)L_{adc}(i) - H_{adc}(i)L_{adc}(i-1))}{2\sqrt{\sum_{i=1}^{N_{gfd}}H_{adc}^2(i) - \left[\sum_{i=1}^{N_{drift}}H_{add}(i)/N_{drift}\right]^2_{N_{gfd}}}\sqrt{\sum_{i=1}^{N_{gfd}}L_{adc}^2(i) - \left[\sum_{i=1}^{N_{drift}}H_{add}(i)/N_{drift}\right]^2_{N_{gfd}}}\ \sin(2\pi f_N)}\right]$$

Figure 40 Equation 8

Since the *a*sin function used (*a*sin2 cannot be used as it requires quadrature IQ information), the above equation puts an upper limit to the delay at a specific frequency:

$$2\pi f_N|fdHL| \le (\pi/2),\ \text{or}\ \ |fdHL| \le 1/4f_N$$

Figure 41 Equation 9

When Nyquist folded bandpass sampling is used, such as 60MHZ IF and 100MHZ sampling, the fN in eq.8 is actually folded to (1-fN) therefore the delay normally interpreted (for no-folded sampling) as "lead" or "lag" should be reversed due to the sin(2?fN) term.

The fractional delay can also be converted to or from delay in absolute time unit such as nano-second (nS) to facilitate the user configuration and status display, by simply multiplying or dividing the delay in samples by the ADC clock frequency.

5.7.3.2 Extended Delay Detection (±180 degree range)

RF components in the front-end receiver may cause Hi/Lo relative phase delay in IF (to IFD's ADC inputs) to be anywhere between -180 degree to +180 degree, while eq.9 indicates that the basic detection can only handle -90 to +90 degree.

To cover the full range, ±180 degree phase delay detection, the quadrature information of the angle must be found. An easy way would be to construct two separate I and Q filters for both the high and low gain channels, but would significantly increase the implementation cost (FPGA area budget). When the angle detected from the basic asin term is known, the actual phase can be deduced as:

$$\phi = (\cos>0)\,?\ \text{asin}:(\sin>0)\,?\,(180-\text{asin}):(-180-\text{asin})$$

Figure 42 Equation 10

So if we can derive cosine info from the incoming ADC wave, we can cover the phase detection to ±180 degree. Note that only the sign of the cosine term, not necessary its accurate value is needed to identify the quadrature correctly, as long as the quadrature crossing critical points, namely the ±90 boundaries can be identified uniquely.

The cos term can be derived by using derivatives of the incoming waveform stream, especially in baseband sampling, where the normalized frequency is low, or equivalently, the sampling frequency is far higher than signal frequency - as a result, many samples per cycle can be sampled and used to calculate the derivative; or the sampling period, T, is relatively very short, as defined mathematically:

$$\cos(t) = \frac{\sin(t+\Delta t)-\sin(t)}{\Delta t},\ \Delta t \to 0,\ \Delta t\ as\ T$$

Figure 43 Equation 11

But in RF/IF signal processing, quite often bandpass sampling, where low sampling frequency is used, even processing at aliasing frequency, the normalized frequency is still very high. Figure 44 on page 139 shows an example, where a 60MHz IF signal is sampled at 72MHz. Only 6 samples per cycle can be obtained even at the relatively lower aliasing frequency (12MHz). Therefore the assumption in Equation 11 on page 138 is not valid and considerable error will be resulted for the derivative, as shown in Figure 44 on page 139: the phase error between the ideal derivative (when T is tiny, shown in cyan) and the actual one (when T is corresponding to 60 degree, shown in red) is corresponding to about half of the sampling period.

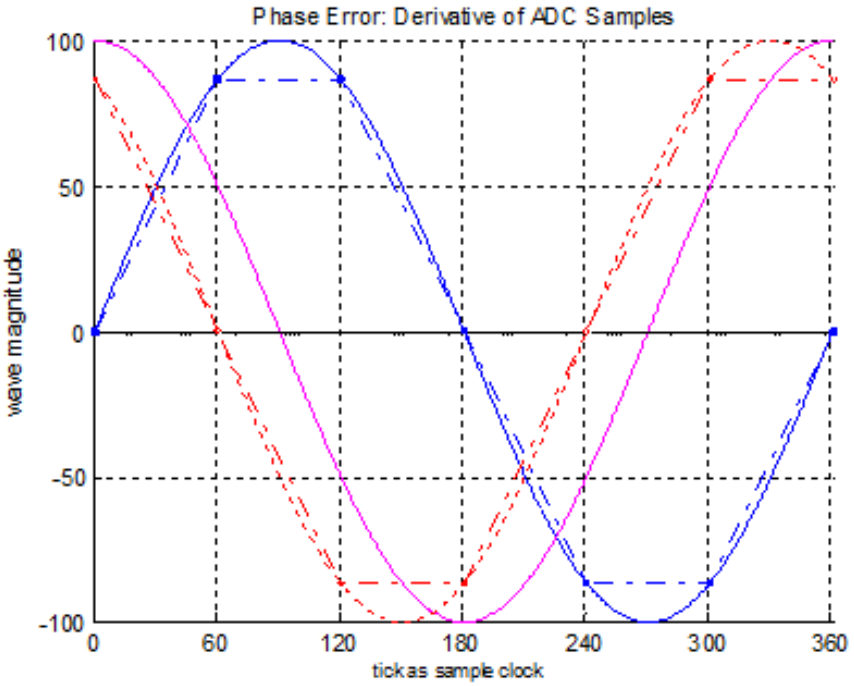


Figure 44 Phase Error: Derivative of ADC Samples

NOTE	The challenge for derivative cosine detection from ADC wave when bandpass sampling with high normalized frequency. Ideal/ADC wave of the alias 12MHz (in blue solid/dash); ideal derivative (cos) of 12MHz wave (in cyan); actual derivative (in red dash) and fitting (in red dot).

The derivative error around the critical boundary-crossing points (i.e. ±90 degree) can be reduced by correlating along two complimentary directions:

$$\cos(Da) = \left(\frac{(HL1+HL2)/(2\pi f_N)}{2\sqrt{\sum_{i=1}^{N_{gfd}} H_{adc}^2(i) - \left(\sum_{i=1}^{N_{drift}} H_{adc}(i) / N_{drift} \right)^2 N_{gfd}} \sqrt{\sum_{i=1}^{N_{gfd}} L_{adc}^2(i) - \left(\sum_{i=1}^{N_{drift}} L_{adc}(i) / N_{drift} \right)^2 N_{gfd}} \sin(2\pi f_N)} \right)$$

where

$$HL1 = \sum_{i=2}^{N_{gfd}} \{ H_{adc}(i)[L_{adc}(i) - L_{adc}(i-1)] - H_{adc}(i-1)[L_{adc}(i+1) - L_{adc}(i)] \}$$

$$HL2 = \sum_{i=2}^{N_{gfd}} \{ [H_{adc}(i) - H_{adc}(i-1)]L_{adc}(i) - [H_{adc}(i+1) - H_{adc}(i)]L_{adc}(i-1) \}$$

Figure 45 Equation 12

Figure 46 on page 141 shows the ±180 degree phase detection coverage. Since only the sign of cosine term is used to identify whether the phase is to the left or right of the qudrature plane, not the absolute (acos) value, so the results shown is not surprising. Although the acos value around the phase 0 and ±180 degree is far off from the actual (about 30 degree error), but the quadrature (left/right plane) can still be correctly identified based on the sign of cosine. For example, around phase angle 0, the acos produces value as about 30 degree instead of 0, but the sign of cos(0) and cos(30) are the same, i.e. positive (+); around the phase ±180 degree, the acos produces value as around 150 degree instead of 180, but both have the same sign in terms of cos so they will not affect the quadrature identification either. Around the critical ±90 degree, where the sign of cos term is abruptly switching, the derivative method produces smooth angle transition, error nearly as zero, as clearly shown.

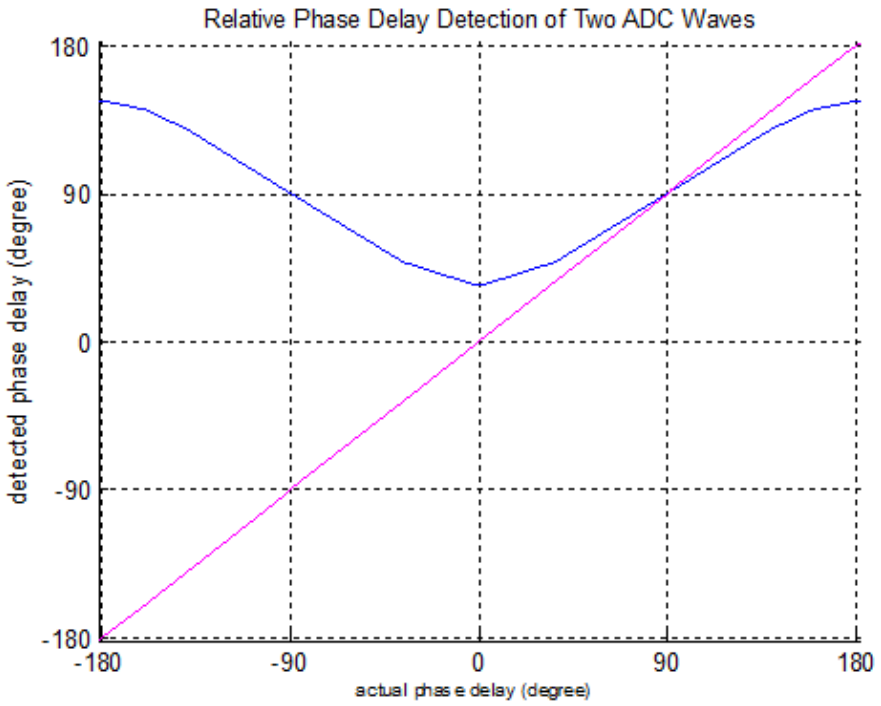


Figure 46 Relative Phase Delay Detection of Two ADC Waves

NOTE	Quadrature identification improvement using complimentary cross-correlations. The cos term for quadrature detection (in blue) identifies the critical boundaries at +-90 degree, and used together with asin term to detect the correct phase delay value (in cyan) within the full-range of +-180 degree.

5.7.4 Saturation and Overlap Detection

5.7.4.1 Saturation, Noise-floor and Overlap

Some design parameters affect the dynamic range, as shown in Figure 47 on page 142.

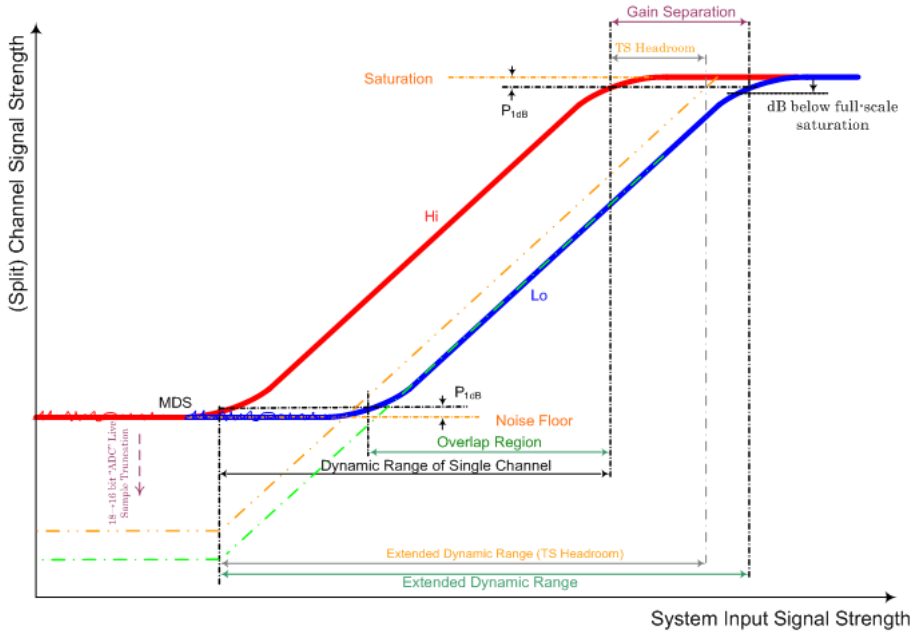


Figure 47 Saturation and Headroom Configuration Effect to Dynamic Range

A user configurable parameter, "dB below full-scale saturation" (dBblSat) is used, instead of fixed P1dB, for internal saturation detection to better handle effects of excessive harmonics when channel deeply saturated. The IFDR used is more general and its on-board antialiasing filter doesn't filter out some harmonics (ref RVP9 Schematics 5.1~5.5 of 46). When deeply saturated, harmonics, such as 3rd-order (180MHZ for IF 60MHZ), when sampled at 81MHZ, its alias (image) of 18MHZ is very close to 60MHZ IF's alias 21MHZ, both are within the wide bandwidth of the 7-tap FDF. As discussed later, the FDF design focus is to handle gain and fractional delay detection and compensation, unlike its down-stream matched-filter (in the downconverter), which has much narrower bandwidth. This parameter should be setup based on the front-end receiver's operational signal range, and generally between -1~-2dB, but also affected by another design parameter, such as "TS headroom" as discussed below. As shown in the figure, it will affect the achievable dynamic range.

Another parameter, weak signal level can be derived if the overlap region is specified.

$$saturationLevelDB = fullDB - dBblSat$$
$$weakLevelDB = saturationLevelDB - overlapDB$$

Figure 48 Equation 13

Both saturation and overlap detection are implemented in FPGA with little involvement of the Linux PC side except initial user configuration of the thresholds. Overlap region, as a user configurable input should be specified narrower than the actual system overlap region, and should ensure the weakDB is at least 1dB above the noise floor for the Lo channel, where the two channels started to overlap with each other.

The saturation and weak signal detections work in a similar way but for different purposes: They both use a sliding window to keep status information of the most recent 100 samples from Hi channel. Each incoming sample is compared against preconfigured (both positive and negative, converted from dB to binary to facilitate FPGA implementation) thresholds and the bits are logically combined to form two logic output bits: sat and weak:

$$sat = \overset{100}{\underset{i=1}{|}} satbit(i); \qquad satbit(i) = \left(H_{adc}(i) > satLevel\right) \mid \left(H_{adc}(i) < -satLevel\right)$$

$$weak = \overset{100}{\underset{i=1}{\&}} weakbit(i); \quad weakbit(i) = -weakLevel \; < H_{adc}(i) < \; weakLevel$$

Figure 49 Equation 14

When Hi channel is deemed as saturated (sat logic "true"), the Lo channel output will be used to generate the final WDR output; otherwise, Hi channel output is always used. Only non-saturated (sat bit as false) samples are accumulated for ADC DC drift detection, and only samples in a "good" overlap region (not too strong, too weak, i.e. both sat and weak bits are false) are accumulated for gain and delay detection.

5.7.4.2 TS Headroom

Users may want to control the output of the WDR to be at a specific gain (separation from Hi channel) instead of using what detected, which as shown is affected by at least environmental factors. Figure 50 shows the effect of the parameter "TS headroom", which affects the achievable dynamic range and the lowest level of WDR output.

When TS headroom specified, the Lo and Hi channel FDF gain is initialized as:

$$gainLo(dB) = gainSep(dB) - Headroom(dB);$$
$$gainHi(dB) = -Headroom(dB)$$

Figure 50 Equation 15

As illustrated in Figure 47 on page 142, the Lo channel may no longer have a normalized gain of 1, unless default headroom (equal to gain separation) is used. Or visualized from another perspective, when default headroom used, the merged signal is equivalent to shifting the Hi channel signal downwards (right-bottom) by GainSeparation while keeping the Lo channel gain as unity; otherwise, the merged signal level is upwards (left-top) shifted above Lo channel (Lo gain >1). In both cases, the relative gain separation between Hi and Lo must be kept. During operation, the detected relative gain (as in Equation 6 on page 136) is used to adjust Hi channel gain, while Lo channel gain is kept as initialized.

5.7.4.3 Dynamic Range and Live ADC Monitoring

The legacy utility dspx-Pr plot "live" 16-bit ADC samples, before they enter into the downconversion channel. For debug/service purpose, the outputs of the WDR's FDF (WD0 and WD1) are also routed as "virtual" ADC samples to be displayed together with other real ADC outputs (A, B, C, D) as shown in Figure 32 on page 132, Since the WD0/WD1 are 18-bit, with wider dynamic range, they must be truncated as 16-bit when signals are too weak, as shown in Figure 47 on page 142. The WD signal level can be derived from Hi channel by offset of headroom when signal level is lower than threshold:

$$\begin{aligned} &if \quad WD(dBm) > trunc(dBm) \qquad WD(dBm) = WD(dBm); \\ &else\ WD(dBm) = Hi(dBm) - Headroom(dBm) \end{aligned}$$

Figure 51 Equation 16

For the 16-bit ADC, the NDR has about 106dB dynamic range and noise level of -90dBm, as shown in Figure 31 on page 131, so generally the threshold can be configured as about -90dBm.

5.7.5 Fractional Delay FIR Optimization

Fractional delay FIR (FDF) filters are used to make the Hi and Lo channel "aligned" in gain and phase before its output being selected for WDR output.

5.7.5.1 Cover the Spectrum: BPF, HPF and LPF Tradeoffs

To be cost-effective with constrained resource (DSP blocks in the FPGA), a short (7-tap) FDF is selected after search by simulation. It can be used as either low-pass (LPF), band-pass (BPF) or high-pass (HPF) filter based on

the actual normalized frequency to cover the full spectrum (0~0.5), as shown in Table 1.

Table 11 FDF used as LPF, BPF and HPF

Type of FIR	LPF	BPF	HPF
Normalized Frequency	< 0.05	0.05 ~ 0.30	> 0.30

The 7-tap FDF can be generalized as:

$$y(k) = \sum_{i=0}^{6} h(i)x(k-i)$$

Figure 52 Equation 17

where h(i), i=0~6 are the coefficients of the FDF, x and y, respectively, the input (ADC output) and output of the FDF. The design of the coefficients is to approximate the response of the ideal filters:

LPF:

$$h(i) = gain \cdot \frac{\sin(\omega_c(i-fd)}{\pi(i-fd)};$$

$$H(\omega) = \begin{cases} gain \cdot e^{-j\omega \cdot fd}, & 0 \le |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases}$$

Figure 53 Equation 18

BPF:

$$h(i) = gain \cdot \left(\frac{\sin(\omega_{c1}(i-fd)}{\pi(i-fd)} - \frac{\sin(\omega_{c2}(i-fd)}{\pi(i-fd)} \right);$$

$$H(\omega) = \begin{cases} gain \cdot e^{-j\omega \cdot fd}, & \omega_{c1} \le |\omega| \le \omega_{c2} \\ 0, & \omega_{c2} \le |\omega| \le \pi \ or \ |\omega| \le \omega_{c1} \end{cases}$$

Figure 54 Equation 19

HPF:

$$h(i) = gain \cdot \left(\delta(i-fd) - \frac{\sin(\omega_c(i-fd)}{\pi(i-fd)} \right);$$

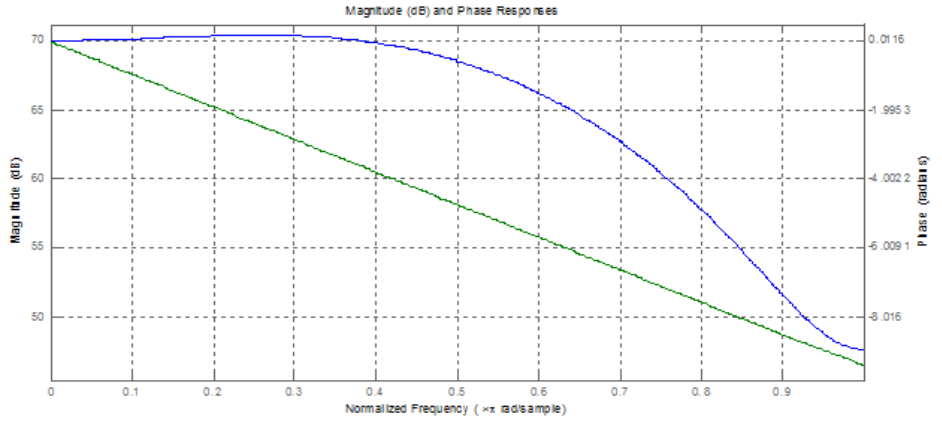
$$H(\omega) = \begin{cases} gain \cdot e^{-j\omega \cdot fd}, & \omega_c \le |\omega| \le \pi \\ 0, & |\omega| \le \omega_c \end{cases}$$

Figure 55 Equation 20

where fd the fractional delay, ? = 2?f as the angular frequency of normalized frequency f, ?c = 2?fc as cutoff angular frequency at

normalized frequency fc, H(?) as the frequency response of the filter. Common characteristics of the filter can be described in frequency domain in four terms:

(a) FDF as BPF



(b) FDF as HPF

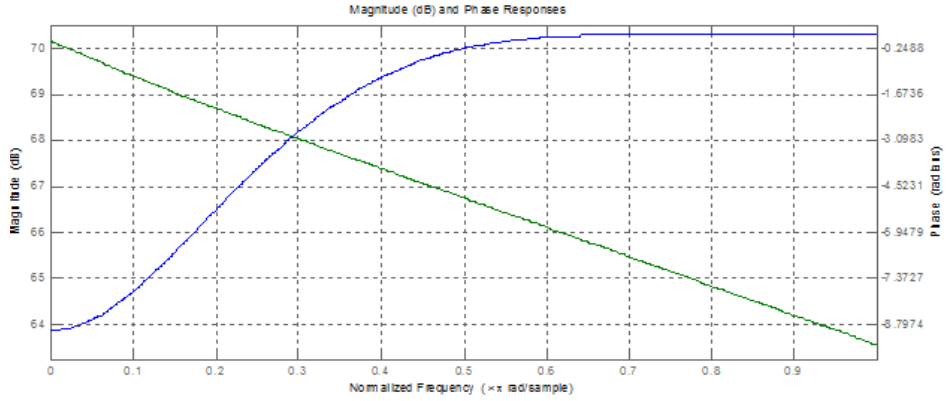
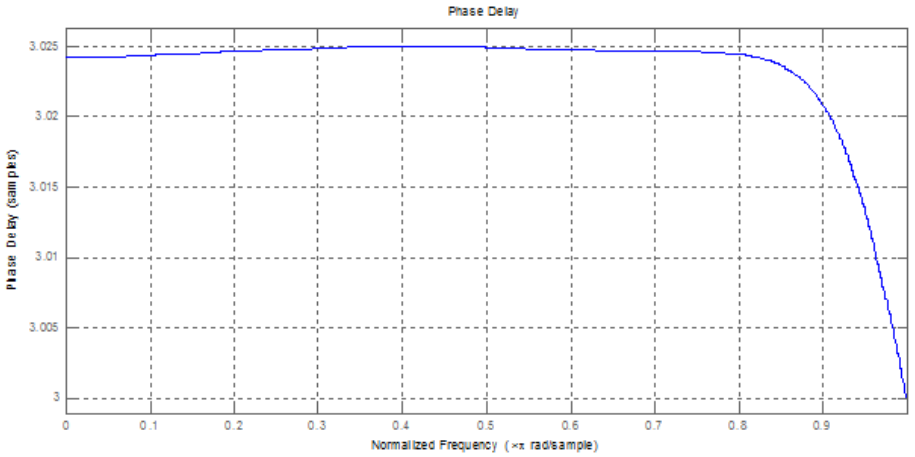


Figure 56 Comparison of Magnitude and Phase Response of (a) BPF and (b) HPF

(a) FDF as BPF



(b) FDF as HPF

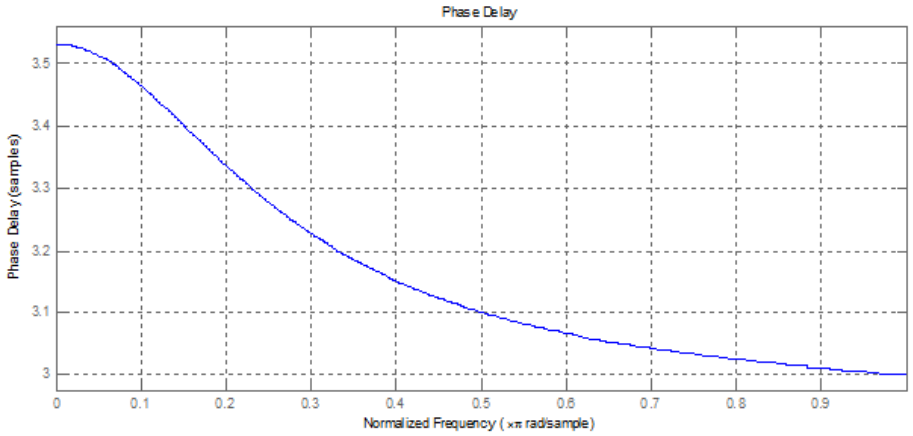
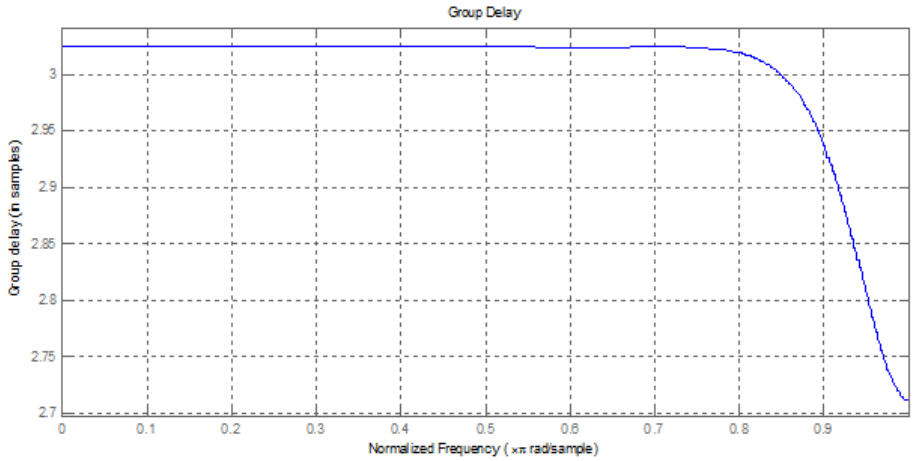


Figure 57 Comparison of Phase Delay Response of (a) BPF and (b) HPF

(a) FDF as BPF



(b) FDF as HPF

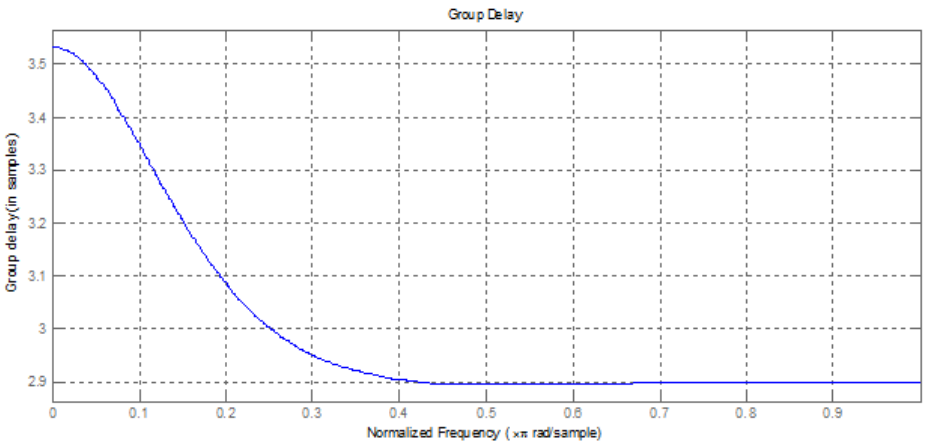


Figure 58 Comparison of Group Delay Response of (a) BPF and (b) HPF

NOTE	Frequency is normalized over Nyquist frequency.

$$Magnitude: \quad |H(\omega)|$$

Figure 59 Equation 21

$$Phase: \quad \theta(\omega) = \arg[H(\omega)]$$

Figure 60 Equation 22

$$PhaseDelay: \quad \tau_p(\omega) = -\frac{\theta(\omega)}{\omega}$$

Figure 61 Equation 23

$$GroupDelay: \quad \tau_g(\omega) = -\frac{d}{d\omega}\theta(\omega)$$

Figure 62 Equation 24

These filters are approximated using traditional windowed method: Hamming is used for LPF and HPF while Blackman used for BPF. It is interesting to observe the simulated characteristics of BPF and HPF for the same fractional delay (0.05), as shown in Figure 63 for magnitude |H(?)| and phase ?(?), Figure 64 for phase delay ?p and Figure 65 for group delay ?g, notice that the frequency is normalized over Nyquist frequency in the figures (twice of that if normalized over sampling frequency).

The most noticeable difference between BPF and HPF is the phase delay, group delay as well as the relation between them. Although both type of filters show similar "linear" phase response in Figure 63, the BPF, designed with a 0.34 normalized frequency (over Nyquist), shows as nearly constant around its center frequency, while the HPF, designed for a 0.8 normalized frequency (over Nyquist), although shows nearly constant group delay within its band, its phase delay is non-linear. The behavior can be illustrated by comparing HPF with LPF, from which the former is derived by shifting center frequency by k?:

$$\underset{\theta(\omega) = -c\omega}{LPF} \qquad \underset{\theta(\omega) = -c\omega - k\pi}{HPF}$$

Figure 63 Equation 25

$$\tau_p(\omega) = -\frac{\theta(\omega)}{\omega} = c \qquad \tau_p(\omega) = -\frac{\theta(\omega)}{\omega} = c + \frac{k\pi}{\omega}$$

Figure 64 Equation 26

$$\tau_g(\omega) = -\frac{d}{d\omega}\theta(\omega) = c \qquad \tau_g(\omega) = -\frac{d}{d\omega}\theta(\omega) = c$$

Figure 65 Equation 27

More detailed phase delay and group delay of HPF is illustrated in Figure 58 on page 148 around its bandwidth (0.6~1 centered at 0.8 in this case). The relationship between phase delay (detected based on Equation 8 on page 137) and the group delay (feedback to design FDF based on Equation 20 on page 145 plus window shaping) around the center of the HPF bandwidth:

$$fd = fdHL\,\frac{f_N}{0.5 - f_N}$$

Figure 66 Equation 28

Using Equation 28 on page 150, the feedback control of FDF can converge within reasonable number of iterations, but the residue power (the difference between Hi and Lo channel output) is higher than that of using BPF or LPF. An all-pass FDF[16] can be designed without the shortcoming, but significantly more DSP blocks from FPGA are needed, which may be out of the design constraints of WDR.

5.7.5.2 Optimize Frequencies for BPF

If the FDF is used as a BPF (or LPF which covers only very narrow bandwidth) instead of HPF, the system can have a better performance. The normalized frequency range is 0.05~3.00.

With known IF frequency fIF, ADC frequency fADC, and the bandwidth of the IF signal fBW, the FDF can be designed as a BPF limited by:

$$f_N = \begin{cases} f_{IF}/f_{ADC} & \text{if } f_{IF} < f_{ADC}/2 \\ 1 - f_{IF}/f_{ADC} & \text{if } f_{ADC}/2 < f_{IF} < 1 \end{cases}$$

Figure 67 Equation 29

$$f_{BWN} = f_{BW} / f_{ADC}$$

Figure 68 Equation 31

$$(0.05 + f_{BWN}/2) \le f_N < (0.30 - f_{BWN}/2)$$

Figure 69 Equation 32

where fBWN is the normalized IF bandwidth. In the case of often-used Nyquist folded sampling (as in equation 25), for known fIF, the sampling frequency can be derived as:

$$\frac{f_{IF} + f_{BW}/2}{0.95} \le f_{ADC} < \frac{f_{IF} - f_{BW}/2}{0.70}$$

Figure 70 Equation 33

For a known 60MHZ IF with different bandwidth, the optimal ADC sampling frequency to make FDF as BPF is listed in Table 12.

Table 12 Optimized ADC Sample Frequency for 60MHZ IF

fBW MHZ		0.25	0.5	1	2	3	4	5	6	7	8	9	10
fADC	max	86	85	85	84	84	83	82	81	81	80	79	79
	min	63	63	64	64	65	65	66	66	67	67	68	68

5.7.6 Error Tolerance and FDF Update

The relative gain and fractional delay detected are compared against user specified tolerances (see dspx setup), default as 1/1000 (gain ratio-1 or a fractional sample). If either is out of the limit, FDF update is triggered automatically each watchdog cycle during a 100-cycle time frame after the power up or every 100 watchdog cycles after the initial period. The 100 cycles, or 10 seconds (based on default watchdog period) is selected based on the assumption/observation that the changing of relative gain and delay is very slow, such as that caused by temperature drift and aging.

The FDFs of the Hi and Lo channel of a WDR block are updated based on the relative gain gainHL (equation 34) and delay fdHL (equation 35) detected:

$$gainH = gainH / gainHL$$

Figure 71 Equation 34

$$gainL = gainL, \ (keep)$$

Figure 72 Equation 35

$$fdH \quad = fdH \begin{cases} + fdHL/2, & if\ LPF/BPF \\ - fdHL/2, & if\ HPF \end{cases}$$

Figure 73 Equation 36

$$fdL \quad = fdL \begin{cases} - fdHL/2, & if\ LPF/BPF \\ + fdHL/2, & if\ HPF \end{cases}$$

Figure 74 Equation 37

where gainH, gainL is the gain, and fdH, fdL the fd used to design the FDF (Equation 18 on page 145, Equation 19 on page 145, Equation 20 on page 145). Notice that there is a phase delay reversal of HPF relative the LPF or BPF. The Hi channel gain is scaled to be the same as that of Lo channel (kept unchanged) so that within the overlap region, the Hi and Lo channel are "identical" to facilitate the switching to WDR output at transition point.

5.8 Use of DSPX Utility or Manual User Configuration and Status Monitoring

The receiver mode and relevant parameters can be configured either with factory default or by a user via dspx utility tool.

5.8.1 Dspx -Mb

WDR related setup question: the Rx IF frequency will be used for 7-tap FDF design.

```
Burst Pulse and AFC
-------------------
Tx Intermediate Frequency: 60.0000 MHz
Rx Intermediate Frequency: 60.0000 MHz
```

Figure 75 WDR Related Setup Questions: Dspx-Mb

5.8.2 Dspx -Mc

WDR related setup questions: ADC clock and Rx Mode.

```
IFD synthesized system clock: 81 MHz

#     Rx Mode Description
-   ----------------------
0   Standard single channel
1   --Reserved--
2   Legacy RVP900/2005 WDN Compatibility
3   Standard dual channel

Default receiver mode: 3
```

Figure 76 WDR Related Setup Questions: Dspx-Mc

The selection of IFD clock also affects the WDR internal FDF design. The selection of IF and IFD clock affect the normalized frequency of the FDF.

RxMode: when 0 or 3 (for 1-pol or 2-pol) selected, user can proceed further to use Mp command to select whether to use normal (NDR) or wide-dynamic range design (WDR); Mode 2 is for the legacy single-pol only dynamic range expansion (different implementation).

5.8.3 Dspx -Mp

WDR related setup questions: whether to useWDR or NDR and WDR related parameters (if WDR selected).

```
Use extended dynamic range: YES
IFD New Wide Dynamic Range Parameters
    Overlap for gain/delay monitor: 30.00 dB
    Chan 1 Gain separation & max deviation: 27.9632 dB,
    0.1000 dB
    Chan 1 Hi/Lo Delay & max deviation: -19.4227 deg,
    0.1000 deg
    Chan 1 Use default headroom (gain separation): YES
    Chan 1 dB below full scale for saturation monitor:
    -1.2000 dB
    Chan 1 starting for dspx-Pr truncation correction at:
    -90.0000 dBm
    Chan 2 Gain separation & max deviation: 27.8268 dB,
    0.1000 dB
    Chan 2 Hi/Lo Delay & max deviation: -178.8414 deg,
    0.1000 deg
    Chan 2 Use default headroom (gain separation): YES
    Chan 2 dB below full scale for saturation recovery:
    -1.2000
    Chan 2 starting for dspx-Pr truncation correction at:
    -90.0000 dBm
```

Figure 77 WDR Related Setup Questions: Dspx-Mp

"`Use extended dynamic range: YES`—If the user intended to use the new WDR, answer as "y" or just enter. All the questions below (IFD New Wide Dynamic Range Parameters) display and the number of channels is decided based on user's selection in Mc: that is, if Mode 0 then only one channel; if Mode 3 then two channels (labeled as Chan 1 and Chan 2 in follow-up questions). If the answer is "no", all the WDR specific questions will be skipped (not shown).

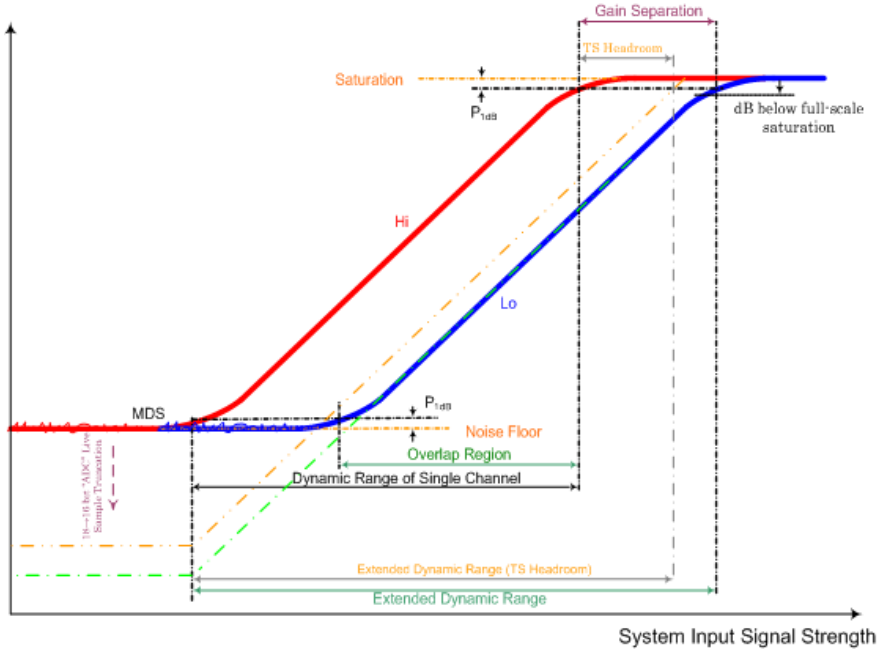


Figure 78 Illustration of WDR Setup Parameters

Parameters related to WDR are shown in Figure 47 on page 142. When first-bootup, the parameters are shown as pre-defined, whenever user changed any of the parameters and "`save`", the parameters will be shown next time as new default values.

"`Gain separation & max deviation`". Gain separation as shown is the relative gain difference between Hi and Lo channel. It should be default as the front-end receiver (or test bench) nominal separation, but may have been changed due to component replacement and environmental issues (such as aging). The max deviation (absolute value) represents the threshold value: when the detected gain separation is over this limit, the FDF will be updated with a new set of filter coefficients. Generally this number should be small (such as 0.5dB). Detected values are shown when run `rvp9 -showRxMode` (see below), so users can re-configure those parameters to make the `Delta` close to zero. So even when those parameters unknown (uncertain) for the first-time bootup, system can still detect the actual "`Gain separation`", but the `Delta` shown in `rvp9 -showRxMode` may be far from zero - then users are recommended to re-run dspx -Mp to reconfigure the values.

"`Hi/Lo Delay & max deviation`". The relative phase delay between Hi and Lo channel as designed/calibrated value of the front-end receiver. The max deviation represents the threshold value: when the detected delay separation is over the limit, the FDF will be updated with a new set of filter coefficients. Similar to "`Gain separation & max deviation`", the detected values are shown in `rvp9 -showRxMode` and users are

recommended to set the values based on what displayed to make the `Delta` close to zero.

"`Use default headroom (gain separation): YES`". The "`headroom`" is shown in Figure 78 on page 155, can be used to control the actual gain of the WDR output (or the gain separation between WDR output and the Hi channel), and it should be less than (or equal to) "`Gain separation`" specified above. Default headroom is equal to the gain separation, and in this case, the Lo channel has a normalized gain of 1. Users are recommended to accept default (answer "`y`" or enter). If not (answer "`n`"), the following question is displayed:

```
TS headroom above Hi channel saturation: 27.9632 dB
```

With the default already displayed, user can change it to other values. In this case, users should also check (and maybe change) the "`dB below full scale`" setup (below), since it may affect the behavior of saturated channel(s) - the Lo channel will have a normalized gain > 1 and the transition around saturation point (in zauto utility) may not be smooth if those two parameters are not configured properly.

"`dB below full scale for saturation monitor: -1.2000 dB`", the parameter is shown in Figure 78 on page 155 (top-right corner), as a threshold to view the ADC output as saturated and to start processing saturated signals. Actual value depends on the front-end receiver and the signal range, generally between -2.0 to -1.0dB, but maybe just a fraction of dB if non-default headroom used. When channel deeply saturated, excessive harmonics (or alias images) from receiver still pass through IFD's on-board anti-aliasing filter, and the short-length FDF is designed with wide bandwidth with focus on gain/phase adjustment, unlike the down-stream matched-filter in the downconverter.

"`Starting for dspx-Pr truncation correction at: -90.0000 dBm`". As shown in Figure 78 on page 155 (bottom-left), WDR internal logic uses 18-bit, but the ADC is 16-bit. The dspx-Pr shows "`live`" ADC samples with 16-bit resolution, the 18-bit is out of its normal range. For debug/service purpose, the 18-bit is "truncated" as 16-bit if the signal is too weak (default as -90dBm, corresponding to a normal 16-bit dynamic range). For the weak signals, the WDR output shown in dspx-Pr is derived from the Hi channel (sepDB or headroom below).

5.8.4 Dspx -Pr

Used to view "live" ADC samples, not the packed IQ data after downconversion. No change in terms of usage for WDR, EXCEPT that if the front-end output (ADC input) has wider dynamic range (than that 16-bit can represent). As stated in dspx-Mp setup question: "`starting for`

`dspx-Pr truncation correction at: -90.0000 dBm`", WDR internal FDF output (WD0 and WD1 will be derived from Hi channel (Ch1 and Ch2) if (and only if) signal is weaker than specified threshold due to the limitation of 16-bit representation. So users are recommended to use zauto and/or ascope to check system noise floor - both zauto and ascope use full resolution floating point packed IQ data (after the downconversion).

5.9 STATUS MONITORING

WDR real-time status can be displayed as in Figure 79 on page 157 with current Rx mode and channels.

```
Standard dual channel -
   Channel: 0 (Gain dB, Delay deg) Sep(27.9632 -19.4227)
   Delta(0.0000 -0.0000) Err(-0.0079 -0.0385)
   Channel: 1 (Gain dB, Delay deg) Sep(27.8268 -178.8414)
   Delta(-0.0000 0.0000) Err(0.0029 -0.0071)
```

Figure 79 WDR Related Monitoring Display: RVP9-ShowRxMode

Three pairs of values are displayed in the same format:

`(Gain dB, Delay deg)`

with the first value corresponding to relative gain and second to relative delay. At the start-up, the monitor runs every watch-dog cycle (about 0.1 sec); after 100 watch-dog cycles (10 sec), the monitor will update the info every 100 watch-dog cycles (10 sec).

`Sep (gain dB, delay deg)`, which displays the relative gain and fractional delay detected during the last monitoring period. Even with default value (factory-default or last user setup), the system can still detect (and display) the actual separation, so it can still be used as reference for users to reconfigure parameters using dspx-Mp command.

`Delta (gain dB, delay deg)`, which displays the difference between currently detected (gain, delay) and the user configured values: "`Gain separation`", and "`Hi/Lo Delay`" in dspx-Mp. When the differences are out of the pre-defined limits, or the `max deviation` configured, the pair will be followed by "!!!" as a warning so that the user can re-configure the default configuration values. Users should use actually detected `Sep (gain, delay)` values after the initial power up time frame (10 seconds by default) for the re-configuration. Once re-configured, the Delta pair is expected to be or very close to `(0,0)`.

`Err(gain, delay)`, which displays the gain and delay errors - the difference between what detected in the last period and what ideally they should be: the Hi and Lo channel should be perfectly matched (after FDF auto-adjustment): (0dB, 0deg) for a smooth transition between Hi and Lo channel when merging as output to the downstream downconverter.

If the signal strength of Hi channel is out of the overlap region (Figure 78 on page 155), no correlation information can be used for FDF adjustment, the display will change from 3 pairs of values to:

```
NOT ENOUGH OVERLAP SAMPLES DETECTED


Standard dual channel -
   Channel: 0 NOT ENOUGH OVERLAP SAMPLES DETECTED
   Channel: 1 NOT ENOUGH OVERLAP SAMPLES DETECTED
```

Figure 80 WDR Related Monitoring Display: RVP9-ShowRxMode When Operate Out of Overlap Region

5.10 TESTING

To verify the design concepts and algorithms, a simple passive splitter network was used in the earlier stage, which has much-needed dynamic range (depending on the signal generator), and later a "strip-down" version of the actual front-end receiver (only the IF portion) was constructed (by PNY from HEL) for a more realistic testing environment, but with lower dynamic range (higher than noise-level of IFDR itself). Preliminary system verification test done on actual receiver and radar system conducted by PNY and RKE from HEL are listed at the end.

Figure 81 Simple Test-Bench: Splitter, 1-inch Connector, 4-inch and 8-inch Cables as Extra Parts for Delay Simulation

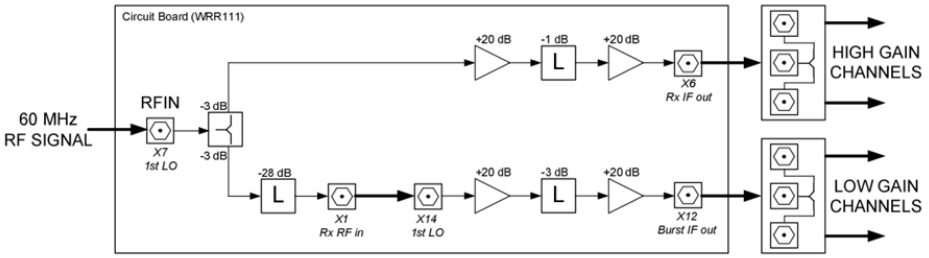


Figure 82 Block Diagram of the 60 MHz WDR Test Board (Coaxial Cables are Marked with Thick Arrow)

Unless explicitly specified afterwards, all test results shown in this document are based on the simple bench (Figure 81 on page 158). The splitter is used to split the signal generator output to two signals: Hi and Lo, a 20 dB attenuator is used to separate the Hi and Lo channel, a 4-inch or 8-inch cable is used to emulate different delays along the Hi or Lo channel, and a 1-inch connector is used to connect the short cable to the 36-inch cable along the Hi or Lo channel.

Limited by the test bench signal generator (upper power output limit 13 dBm) and loss of 3.8 dBm by the splitter, it is physically impossible to test the max potential dynamic range, so comparative measure (done in late stage on bench illustrated in Figure 82 on page 159) and projection are used - all different receiver modes (standard, existing 1-pol dynamic range and the new 2-pol dynamic range expansion) are tested under the same environment.

As discussed in section 3.6, FDF update period is intentionally longer than a watchdog cycle (100 cycles after the initial power on), so be patient with changing configuration parameters - wait until updated - natural changing of gain/delay would not be as fast as bench-testing.

5.10.1 Dspx Configuration

A user can configure major parameters via dspx utility tool, as shown in Section 5.7.5 Fractional Delay FIR Optimization on page 144:

- Mc command is used to specify the receiver mode to be used. Either mode 0 or mode 3 will power related WDR channel(s) if "using wide dynamic range" question in Mp is answered as "yes". If neither is selected, relevant DSP blocks in the FPGA are powered down. IFD (ADC) clock selection should be considered together with Rx IF frequency.
- Mb command is used to specify Rx IF frequency, it should be considered together with ADC frequency.

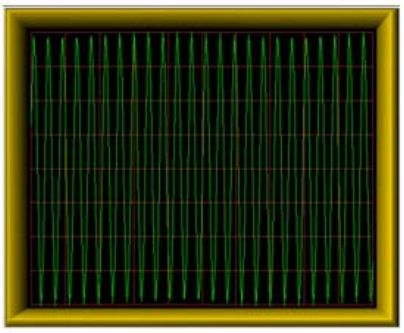
- Mp command is used to specify dynamic range mode parameters such as overlap, gain and delay separation and error tolerance

5.10.2 Dspx Plot

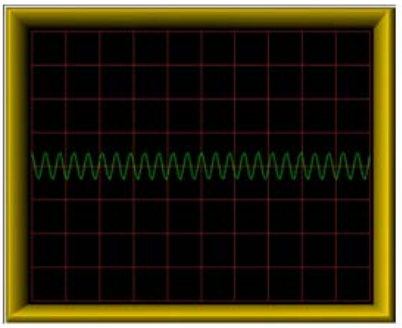
The test bench shown in Figure 81 on page 158 is configured as: after splitter and the two 36-inch cables to Hi and Lo channels, a 20 db attenuator is attached to Lo channels, while a 4 inch cable connected to the Hi channel through a 1-inch connector (total 5 inches added). This will be the default configuration for all tests below unless explicitly specified otherwise.

Figure 83 on page 160 shows the waveforms captured when the signal generator output is set as 0 dBm. The channel Ch1 (or the Hi) becomes -3.79 dBm due to the splitter gain drop, and the Lo channel becomes -23.68 dBm (instead of -20 dBm if without the splitter gain drop). The "composite" or "merged" channel WD0 has nearly the same signal strength as Lo channel. In the discussion below, to be consistent (or to compare) with the standard receiver mode, which doesn't use splitter, the signal strength after the splitter, instead of that immediately from the signal generator, a -3.79 dBm difference, is used as the (IFD ADC channel) signal input level.

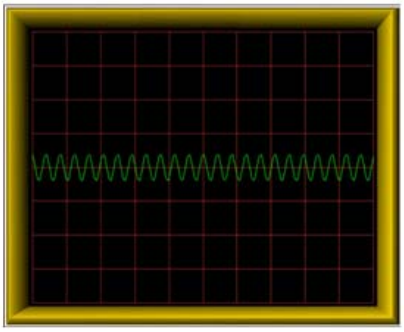
(a) IF: Ch1



(b) IF: Ch2



(c) IF: WDo



(d) Display/Traverse IF Channels

```
[jge@bou-breeze widedyn]$ dspx
Digital Signal Processor 'Chat' Mode
Checking for code upgrades... Okay
(Type ^C to exit Chat Mode)

Total:-3.79 dBm,  Filtered:-3.73 dBm,  MidSamp:-3.73 dBm
Zoom:x4,  Navg:3,  Start:0.00 usec (0.00 km),  Span:2 usec  IF:Ch2
Total:-23.68 dBm,  Filtered:-23.63 dBm,  MidSamp:-23.63 dBm
Zoom:x4,  Navg:3,  Start:0.00 usec (0.00 km),  Span:2 usec  IF:Ch3
Total:-70.74 dBm,  Filtered:-85.80 dBm,  MidSamp:-85.81 dBm
Zoom:x4,  Navg:3,  Start:0.00 usec (0.00 km),  Span:2 usec  IF:Ch4
Total:-71.24 dBm,  Filtered:-89.39 dBm,  MidSamp:-89.21 dBm
Zoom:x4,  Navg:3,  Start:0.00 usec (0.00 km),  Span:2 usec  IF:Ch5(TXB)
Total:-70.24 dBm,  Filtered:-91.77 dBm,  MidSamp:-92.07 dBm
Zoom:x4,  Navg:3,  Start:0.00 usec (0.00 km),  Span:2 usec  IF:WD0
Total:-23.63 dBm,  Filtered:-23.64 dBm,  MidSamp:-23.64 dBm
Zoom:x4,  Navg:3,  Start:0.00 usec (0.00 km),  Span:2 usec  IF:WD1
Total:-90.69 dBm,  Filtered:-107.58 dBm,  MidSamp:-107.59 dBm
```

Figure 83 Dspx Pr plot: Waveforms and Gain Drop of Splitter

Using the Pr plot, the ADC channel "noise" or DC/zero offset/drift can also be monitored, with a signal input -100dBm. The IFD used in the test environment has shown 16-bit ADC value around 87 for the Hi channel, which corresponding to 6~7 bit loss of accuracy if not monitored and cancelled in the algorithms for gain and delay detection.

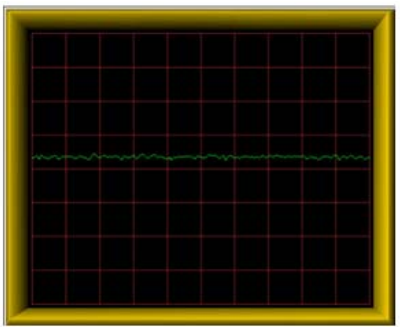


Figure 84 Pr plot: Ch1 (Hi) Channel DC Offset

5.10.3 RVP9 - ShowRxMode Status Monitoring

Running status of the RVP WDR mode can be monitored via rvp9 - showRxMode, including: startup (after initial power up) gain and delay, update if necessary during the startup and follow-up normal operation. Gain and delay delta (comparing against user's pre-configuration) and error (comparing against ideal 0) are also displayed.

5.10.3.1 Monitoring and Update Period

To reduce the bandwidth consumption by the monitor, the status is displayed every seven watchdog cycles (700ms by default), unless there is an update request at a specific watchdog cycle, at which the update monitor/display will overwrite the normal status. Since the update after initial power up will be based on statistics collected from 100 watchdog cycles, be patient in a lab-bench test environment to observe the changes.

5.10.3.2 Relative Delay Test At Various Frequencies

The relative delay between Hi and Lo channel depends on hardware configuration and environmental factors, such as cable length mismatch, aging and temperature. The test-bench is used to test the delay detection

algorithm, especially the performance of the FDF acting as a BPF or a HPF.

Table 13 Relative Hi-Lo Delay Tested using 72 MHZ ADC

Channel Setup	Hi	No extra	1"+4"	1"+8"	No extra	
	Lo	No extra	No extra		1"+4"	1"+8"
IF 20 MHZ		-0.3166	0.3235	0.8381	-0.9555	-1.4691
IF 60 MHZ		-0.2826	0.3472	0.8492	-0.9124	-1.4201

Table 13 shows delay tests using 72 MHZ ADC sampling frequency for 20 MHZ and 60 MHZ IF under various delay configurations. In both 20 MHZ and 60 MHZ IF cases, the 7-tap FDF is used as a BPF, although for 60 MHZ Nyquist folded "bandpass" sampling is used: 60/72 = 0.833, 1-0.833 = 0.167 "normalized" frequency. The delay can be viewed as directly related to the cable length.

For the 60 MHZ case, when no extra cable is added to either the Hi or Lo channel, the delay (-0.2826 nS) is caused by the 20 dB attenuator (which is about 1-inch in length). When an extra 4-inch cable is added to the Hi, through a 1-inch connector (threads conversion), delay is changed to 0.3472 nS. This means that the extra 5-inch added cable/connector caused a delay of 0.3472 - (-0.2826) = 0.6298 nS. When another 8-inch cable is used to replace the 4-inch cable, the added delay is: 0.8492 - 0.3472 = 0.5020 nS, which is corresponding to the extra 4-inch cable length. The ratio of 0.6298 / 0.5020 = 1.2545 is corresponding to the cable length ratio 5/4=1.25.

When the ADC sampling frequency is changed to 100 MHZ, for the 20 MHZ IF case, the 7-tap FDF is still used as a BPF, but for the 60 MHZ, it is used as a HPF. The delay tests are listed in Table 4. Comparing with Table 13, the delay detection seems to work fine. But as simulation indicated (discussed separately), with a reasonably wide bandwidth, HPF may cause higher residue noise than BPF so we should try to use BPF when ADC (and/or IF) frequency optimization is possible.

Table 14 Relative Hi-Lo Delay Tested using 100 MHZ ADC

Channel Setup	Hi	1"+4"	No extra
	Lo	No extra	1"+4"
IF 20 MHZ		0.3270	-0.9548
IF 60 MHZ		0.3473	-0.9147

5.10.3.3 Convergence Test At 60 MHZ IF and 72 MHZ ADC

The speed, or the number of iterations of feedback from the instance when the gain and/or delay are out of specified tolerance to the time they are both within tolerance, or say to converge, depends on the bandwidth of the IF and the FDF used (LPF, BPF or HPF). The test is conducted to verify that the system can converge at reasonable speed.

Assume IF as 60MHZ and use factory default ADC frequency 72MHZ, the two FDFs in Hi and Lo channel act as BPF, the feedback loop can converge very quickly, most time 1~2 watchdog cycles. Bench-test shows that the system converged instantly after power on initialization (1 watchdog cycle). Notice that the test was conducted using signal generator of pure sine wave. For wide band signal, the convergence likely will take more cycles as observed by using bandlimited noise simulation (see MATLAB simulator in SVN).

5.10.3.4 Convergence Test At 60 MHZ IF and 100 MHZ ADC

Using 100 MHZ, the highest ADC sampling frequency achievable in RVP900 for 60 MHZ IF, the FDF acts as a HPF. The relative delay detection is "hacked" to accommodate the difference between phase delay and group delay. The feedback loop will converge at a slower speed (more iterations than the case of LPF or BPF), and the merged output may have higher residue power (noise), depending on the IF bandwidth and actual relative delay.

5.10.3.5 Convergence Test At Other Frequencies

The converging speed depends on the normalized IF center frequency, or whether FDF acting as LPF, BPF or HPF, and the bandwidth of the IF. Section 3.5 discussed the performance of FDF, which is directly related to the converge speed, and suggested the boundary of BPF frequencies (eq.29 to eq.32 and Table 2). Equivalent non-Nyquist-fold frequencies, such as 12MHZ IF, 72MHZ ADC (equivalent to 60MHZ IF, 72MHZ ADC), and 40MHZ IF, 100MHZ ADC (equivalent to 60MHZ IF, 100MHZ ADC) are tested showing the same converging behavior.

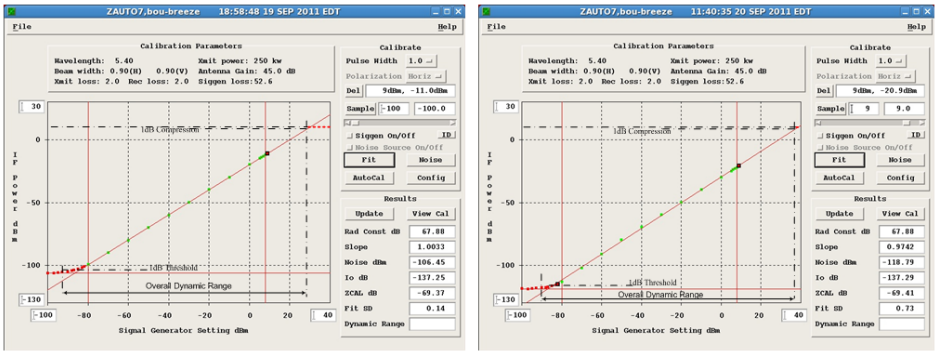
5.10.4 Zauto: Noise Level and Dynamic Range

The noise level and dynamic range of the new receiver modes can be tested using zauto utility tool. But the signal generator used for the test in the lab cannot drive strong enough output (dBm) to reach the saturation level of Lo channel so projection is used to estimate the overall dynamic range with curve fitting. The dynamic range of the new mode is compared with that of existing (single-pol only) wide range mode and the standard mode to verify the testing methodology.

5.10.4.1 At 60 MHZ IF 72 MHZ ADC

As in Section 5.10.3.3 Convergence Test At 60 MHZ IF and 72 MHZ ADC on page 163, at these frequencies, the FDFs act as BPFs. Figure 85 on page 164 shows the zauto plots of the new receiver mode with two different gain separations: 20 dB and 30 dB, both show promising dynamic range.

If 20 dB separation is used, when Hi reaches 28 dBm, Lo could be 8 dbm (saturated); but if 30 dB separation, Lo would only be -2 dBm, which is 10 dBm below saturation level and hence it will not reach the full potential overall dynamic range — or, to get the potential full dynamic range, the Hi channel has to bear up to 38 dBm, severely over saturated.

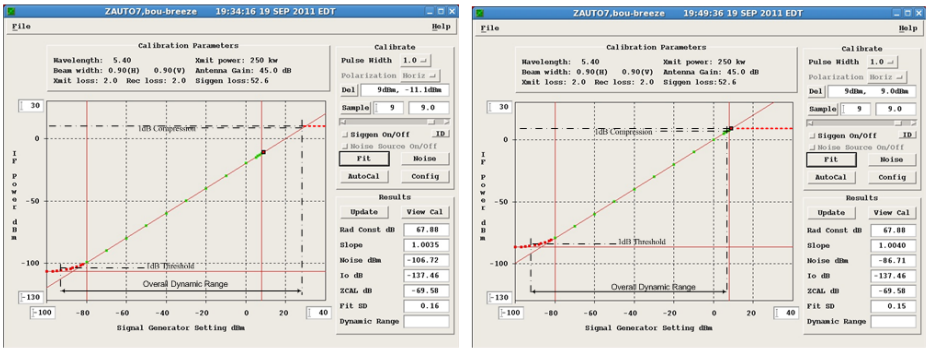


(a) 20 dB Gain Separation (b) 30 dB Gain Separation

Figure 85 Zauto Plots of New Dynamic Expansion Mode at 60 MHZ IF, 72 MHZ ADC (a) with gain separation of 20 dB, (b) with gain separation of 30 dB

The existing (single-polarization only) wide dynamic range receiver mode and standard mode are tested based on the same test-bench setup, as shown in Figure 85 (a) and (b) respectively, to compare with the new dynamic expansion mode shown in Figure 86 (a), all with a 20 dB gain separation.

It can be seen that the new mode has the similar noise level as the existing one, both about 20 dB below the standard mode, therefore a 20 dB increase in dynamic range over the standard mode, which was documented as 106 dB (also duplicated and projected in (b) in Figure 86 on page 165).



(a) Existing Expansion Mode (b) Standard Mode

Figure 86 Zauto Plots of Existing (Single-Polarization only) Dynamic Expansion Mode (a) and Standard Mode (b) at 60 MHZ IF, 72 MHZ ADC

5.10.4.2 At 60 MHZ IF 100 MHZ ADC

As in Section 5.10.3.4 Convergence Test At 60 MHZ IF and 100 MHZ ADC on page 163, the FDF acts as a HPF at these frequencies. Figure 87 on page 166 shows the zauto plot of the new receiver mode with a gain separation of 20 dB.

Comparing with (a) in Figure 85 on page 164, both the noise level and the dynamic range seem unchanged. But refer to Section 5.7.5 Fractional Delay FIR Optimization on page 144, using FDF as HPF may introduce higher residue (noise) power, and the feedback control loop may need more iterations to converge also, which is not visible in the zauto plot. In general, whenever possible, the frequencies should be optimized to set FDF as BPF (or LPF) to achieve better performance.

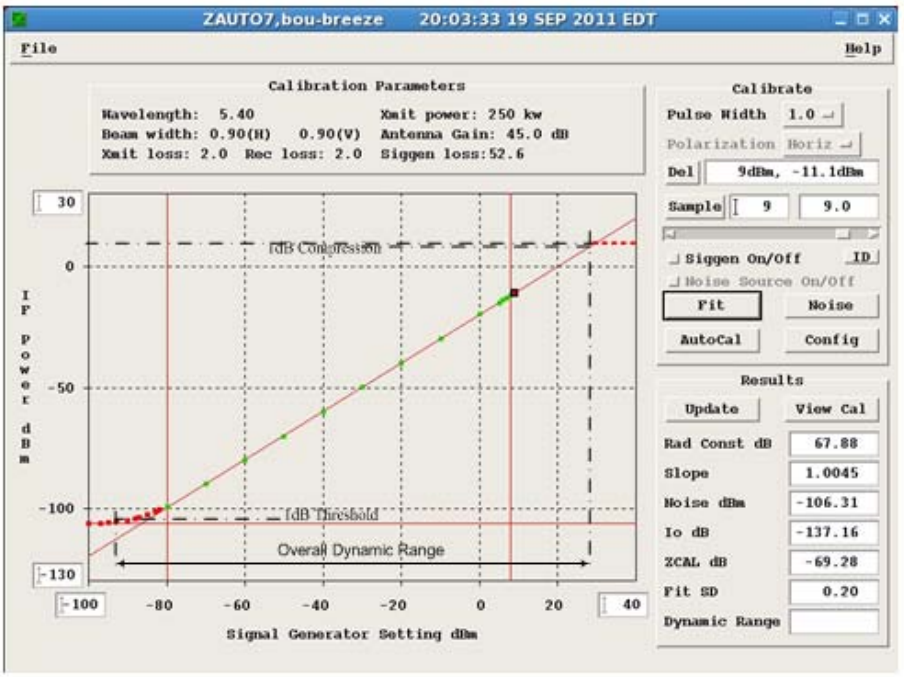


Figure 87 Zauto Plot of New Receiver Mode at 60 MHZ IF and 100 MHZ ADC

5.10.4.3 At 5600MHZ RF 2uS Pulse 81 MHZ ADC (Actual Receiver)

Zauto test from HEL radar lab on actual receiver is shown in Figure 88 on page 167.

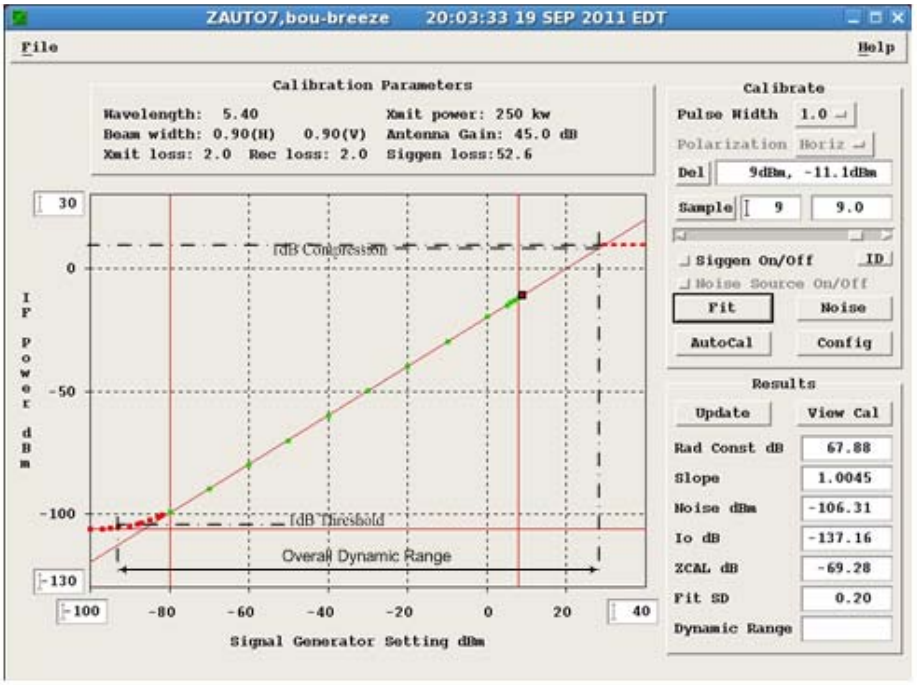


Figure 88 Zauto Plot of Actual Radar Receiver at 5600 MHZ RF, 2us CW Pulse, 81 MHZ ADC (PNY, HEL Lab)

5.10.5 Ascope: Noise Level and Dynamic Range

The dynamic range of the new receiver mode is also tested by using ascope utility tool for various frequency configurations. The dynamic range can be derived from the peak of the total power (0dB reference) to the lowest detectable signal power or noise level, and the smooth transition between Hi and Lo channels can also be monitored around the Hi channel saturation level.

5.10.5.1 Smooth Transition Around Saturation of Hi

The saturation level of the Hi channel is 8dbm, and at this level the Lo is 20dB below, -12dBm when using 20dB gain separation. After setting the signal generator at a reasonable output level, such as 0dBm for a while so that the IFD's overlap detection logic has detected and compensated the gain and delay differences of the two channels, adjusting the generator output from 11.5dBm to 12.5dBm (so that the Hi channel saturation is crossed) with smallest step size of the generator (0.1dBm), the monitored

ascope total power is listed in Table 15, which demonstrated that the transition from Hi to Lo is smooth.

Table 15 Input sweeping to cross Hi channel saturation

Generator (dBm)	11.6	11.7	11.8	11.9	12.0	12.1	12.2	12.3	12.4
ascope (dB)	-20.31	-20.22	-20.12	-20.02	-19.93	-19.83	-19.75	-19.64	-19.55

Another indication of the smooth transition can be monitored via rvp9 -showRxMode. A 0.1dBm less or more than 12dBm signal generator output is displayed from normal three-pairs of (gain, delay) to "No VALID SAMPLES DETECTED" since after Hi is saturated, there is no overlap sample for the internal logic to monitor.

5.10.5.2 At 60 MHZ IF 72 MHZ ADC

Figure 89 on page 169 shows that ascope captured for the new wide dynamic range expansion receiver mode, the existing dynamic range expansion mode, and the standard receiver mode at 6.2 dBm (10 dBm from signal generator minus the loss of 3.8 dBm from the splitter) input to RVP.

The Mag Scope for the new mode (top of Figure 89 on page 169) is still about 22 dB below the maximum (0 dB), the upper end of the dynamic range, which is similar to that of the existing single-pol only wide range design, as shown in the middle of Figure 89 on page 169. The SNRs shown are also very close. With the same setup, the standard receiver (non-dynamic range expansion) mode is also tested as shown at the bottom of Figure 89 on page 169. Since the actual input to RVP is 6.2 dBm, which is about 2 dBm below the saturation point of the channel, the number shown on the top-left corner, -2.04 dB makes sense. But comparing with either the new or existing expansion mode, the dynamic range expansion design has more "room", about 20 dB more to reach saturation, or our expected increase of dynamic range.

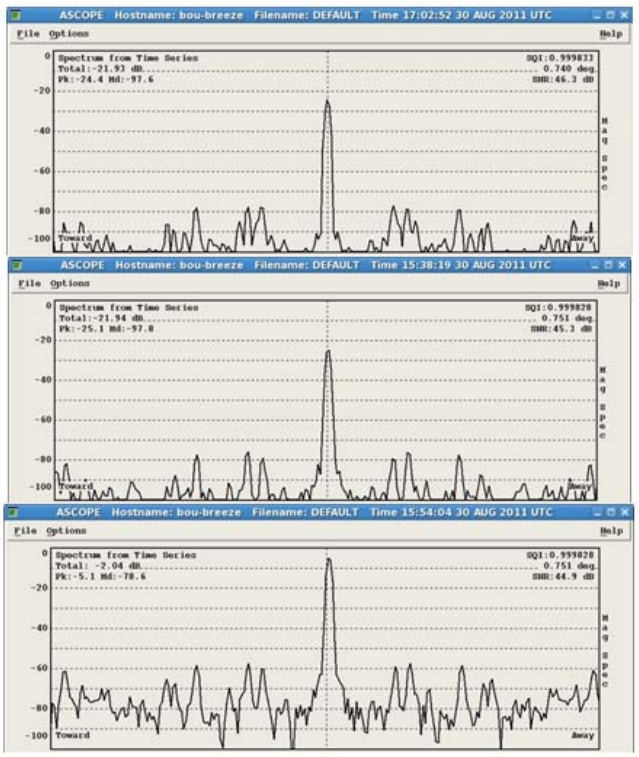
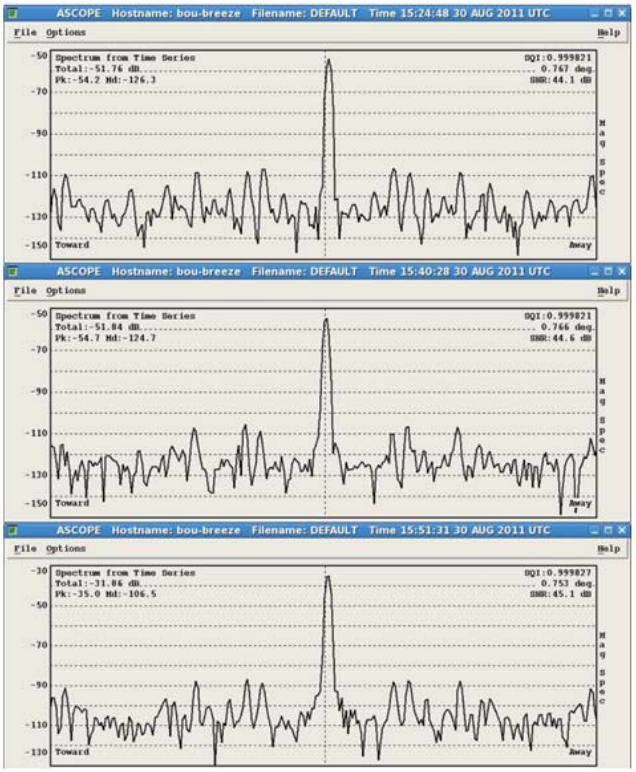


Figure 89 Ascope Plot of New, Existing Wide Dynamic Range, and Standard Receiver Mode at 60 MHZ IF and 72 MHZ ADC, 6.2 dBm Input

When the signal is strong, the noise level shown in the captures in Figure 89 on page 169 may not reflect the real noise level. A reduced signal strength, -20 dBm, is used to test all three designs, and their ascope captures are shown in Figure 90 on page 170. The median of noise levels are indicators of their expected dynamic ranges for the new, old dynamic range and non-WDR designs: 126, 124, 106 dB.



New Dynamic Range

Existing Wide Dynamic Range

Standard Receiver Mode

Figure 90 Ascope Plot of New, Existing Wide Dynamic Range, and Standard Receiver Mode at 60 MHZ IF and 72 MHZ ADC, -20dBm Input

"In practice, a coherent signal at -4 dB SNR can easily be measured when 25 or more pulses are used" (refer to the *RVP900 User's Manual - Digital Receiver and Signal Processor*). The new and existing wide dynamic range expansion modes are tested with very weak sign level at -90dbm, as shown in Figure 91 on page 171. Note that at this level the SNR measurements are not accurate. By using this coherent signal measurement, we may expect a 4 dB higher dynamic range.

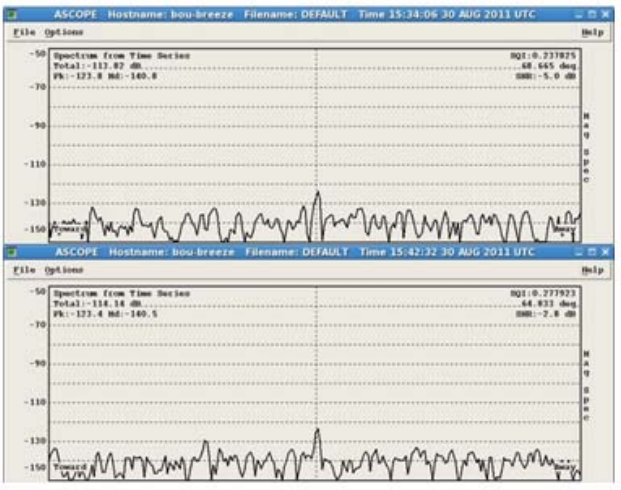


Figure 91 Ascope Plot of New and Existing Wide Dynamic Range Receiver Mode at 60 MHZ IF and 72 MHZ ADC, -90dBm Input

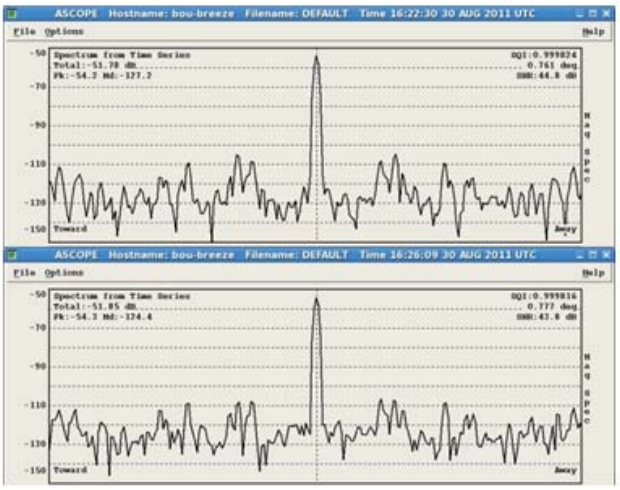


Figure 92 Ascope Plot of New and Existing Wide Dynamic Range Receiver Mode at 60 MHZ IF and 100 MHZ ADC, -20dBm Input

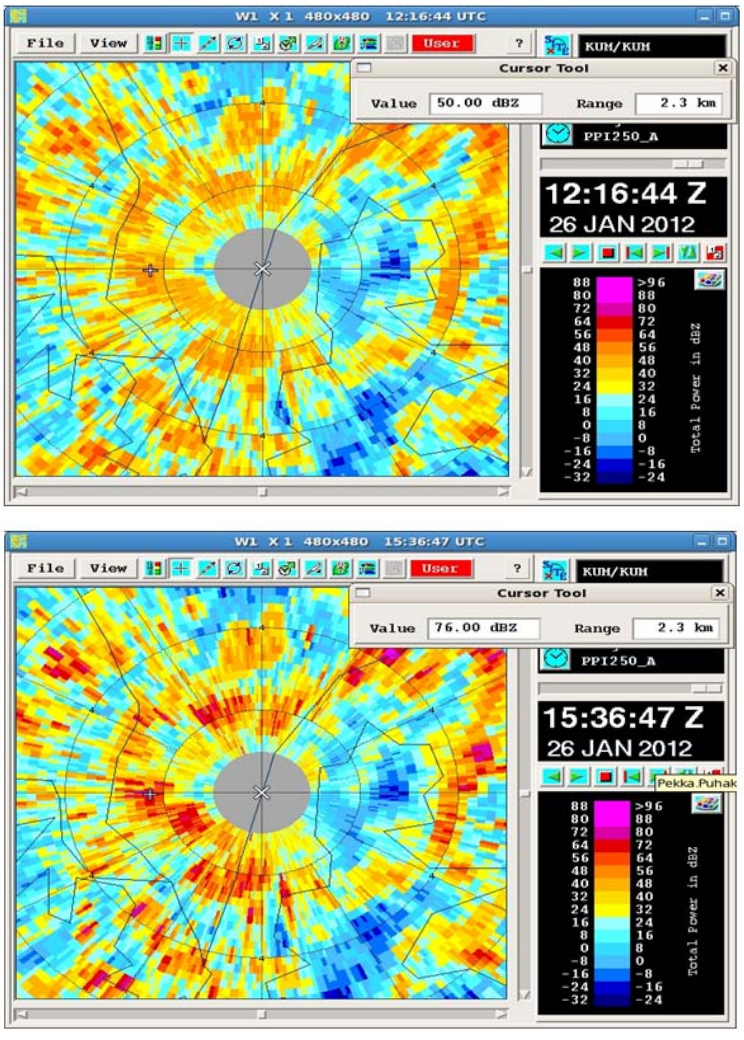
5.10.5.3 At Other Normalized Frequencies

For zauto test, equivalent non-Nyquist-fold frequencies, such as 12MHZ IF, 72MHZ ADC (equivalent to 60MHZ IF, 72MHZ ADC), and 40MHZ IF, 100MHZ ADC (equivalent to 60MHZ IF, 100MHZ ADC) are tested showing similar ascope plots to the equivalent Nyquist-fold frequencies. Figure 92 on page 171 shows the case for 60MHZ IF 100MHZADC. See

Section 5.7.5 Fractional Delay FIR Optimization on page 144 for implications of using FDF as LPF, BPF or HPF and possible optimizations.

5.10.6 Kumpula Radar Operational Test (HEL Lab)

Figure 93 on page 172 shows the NDR and WDR comparison tests conducted by RKE in Finland: WDR has a 26dBz increase over NDR in actual operation.



(a) NDR

(b) WDR

Figure 93 (a) NDR and (b) WDR Comparison Test Conducted on 26 Jan 2012 by Keranen Reino RKE in HEL Office

5.11 SUMMARY

The architecture of the new wide dynamic range receiver is designed to effectively use the constrained FPGA resources in the current RVP900. Its detection module correlates the Hi and Lo gain channel, feeds statistical information to the watchdog thread running on a Linux PC via high-speed Ethernet link. The relative gain and delay of Hi and Lo channel can then be periodically adjusted when necessary by the feedback control from the watchdog thread. Each wide dynamic channel (for each pol) can thus be designed with only a few DSP blocks.

The relevant detection algorithms and FDF design tradeoffs have been mathematically analyzed, simulated and tested. The actual implementation of most of the algorithms is a hardware-firmware co-design effort: the parameters (or sub-modules) used in the algorithms are from the hardware (FGPA) implemented monitor circuit, and math intensive calculations are done in software/firmware in the Linux PC.

The dynamic range of the new receiver modes has been verified in a lab environment as promising. Using the common zauto and ascope utilities, the new modes are shown to have compatible dynamic range and noise level as the existing single-pol only wide dynamic range mode: both have 20dB higher than standard receiver mode when 20~30dB gain separation is used. Smooth transition between Hi and Lo channel around the Hi channel saturation point has also been verified.

CHAPTER 6

CALIBRATION

6.1 Calibration Considerations

Polarization systems require additional calibration as compared to conventional systems. There are three aspects to the calibration:

- dBZ_o measurement in both channels for dBZ and dBT calibration
- gdr measurement for Z_{dr} calibration
- xdr measurement for LDR calibration

6.1.1 dBZ_o Calibration for dBZ

The RVP900 supports separate calibration of both polarization channels. Measurement of dBZ_o for each channel of a dual polarization system is identical to the conventional radar case described in *RVP900 User's Manual - Digital Receiver and Signal Processor*. Note that for a single-channel switching system, the only difference between the horizontal and vertical signal paths occurs after the high power switch, that is, differential insertion loss of the switch itself and any differential insertion loss of the waveguides and feed after the switch. This means that for single-channel switching systems it may be sufficient to calibrate at one polarization and then adjust the calibration of the other channel by the differential gain GDR (see below).

6.1.2 GDR Calibration for Z_{dr}

The Z_{dr} Offset is the dB value of the relative gain between the co-polarized channels including both transmitter and receiver gain, that is:

$$ZDROffset = 10LOG\ \frac{g_v^r g_v^t}{g_h^r g_h^t} \quad \text{and} \quad gdr = \frac{g_v^r g_v^t}{g_h^r g_h^t}$$

GDR is input into the processor as a dB value. However, for analyses in this chapter, the linear gdr value is sometimes more convenient.

In principle, if dBZ_o could be calibrated perfectly in both channels, measurement of GDR would not be required. In practice, this is not possible because dBZ_o cannot be calibrated to an absolute accuracy sufficient for Z_{dr}, that is, to 1/16th of a dB. Therefore, the RVP900 uses the GDR approach.

Since GDR includes both transmitter and receiver differential gains, accurate calibration requires that an actual target be observed. One way to do this is as follows:

- Set the GDR to be 0 dB using your application software (for example, for Vaisala IRIS systems in the setup utility RVP section). Disable clutter filtering for Z_{dr} in either your application software (by selecting filter 0) or explicitly in the RVP900 TTY setups mp section.
- Place the antenna at 90 degrees elevation (vertical incidence) during moderate to heavy rain. The melting layer should be at a height that is well above the recovery zone of the T/R and in the antenna "far zone". A melting layer higher than 2 km is suggested, but the specific characteristics of the radar should be considered.
- Collect Z_{dr} data at vertical incidence while the antenna is rotating in azimuth.
- Use a separate application program to average the Z_{dr} values around a full 360 degrees at each range bin (height). Generate a plot of 360-average Z_{dr} vs height.
- You should observe that the average Z_{dr} values in regions of strong signal (>20 dB SNR) below the bright band are approximately constant with height. This is the value that should be used in your application software for GDR.
- Enter the value and repeat the calibration to verify that the average Z_{dr} is now 0 dB.

The rationale for this approach is as follows. When viewed at vertical incidence, rain should have a Z_{dr} of 0 dB since the drops will all appear circular. The reason for averaging over 360 degrees is to cancel-out effects from sidelobe contamination from nearby ground targets and other artifacts of the antenna/feed/radome system. For example the radome may have an obstruction light on the top. Some of these artifacts can be minimized by assuring the weather targets are strong, that is, heavy rain is preferred for this calibration.

6.1.3 Offset Calibration for LDR

XDR is the dB value of the relative gain between the co- and cross-receiver channels for LDR measurements. Analogous to GDR, it is defined as the dB value of the ratio of the vertical to horizontal receiver gains, for example:

$$XDR = 10LOG\frac{g_v^r}{g_h^r} \quad \text{and} \quad xdr = \frac{g_v^r}{g_h^r}$$

There are three techniques for calibration of XDR:

- Solar method
- Signal generator method with connection to waveguide
- Linear feed horn remote radiator method

It is recommended for the transmitter to be off for all of these methods.

6.1.3.1 Solar Method

Use the sun to measure LDR. The measured value of LDR is then the XDR offset. LDR should be measured in fixed mode for both LDRH and LDRV. The values should be reciprocal (for example, +1 dB and -1 dB). Use the average of the absolute value if they are not precisely reciprocal (for example, for +1.4 and -1.2 use 1.3). Finally after inputting the XDR value, retest to verify that the sun has been properly corrected to have zero LDR.

6.1.3.2 Signal Generator Method with Connection to Waveguide

Connect a signal generator with a splitter to both channels and measure XDR directly. This does not account for any effects that are before the coupler (for example, waveguide, feed, radome, antenna gain).

6.1.3.3 Linear Feed Horn Remote Radiator Method

Use a calibrated linear feed horn with an RF source located several hundred meters from the radar. Maximize the H channel return and measure the response using the RVP900 pr command "Filtered" power in the "Primary Channel". Now rotate the feed horn to vertical and maximize the power in the "Secondary Channel". The difference in dB is XDR. Note that signal multi–path effects could bias the results from this technique.

In all cases it is recommended that for the calibration, XDR be set to 0 dB in the application user software and that the RVP900 TTY setups be configured as follows:

- Noise correction enabled for LDR and noise sample taken prior to the measurements (with care not to sample with a test signal turned–on or while looking at the sun).
- Clutter correction disabled for LDR.

6.2 Z_{dr}

The **zcal** utility is an alternative to the **zauto** utility for entering and displaying the LOG receiver calibration numbers in the calibration file. Zcal can be useful when first setting up a system, before final calibration. It is also the only way to reset reference calibration information. Reference information is applicable only on systems that automatically run calibration. If a new calibration deviates too much from the reference, it is not used. This prevents loss of data if the signal generator fails.

Zcal requires no graphics interface. You enter the calibration numbers which have been determined in some other manner. For an RVP6, these numbers consist of a slope and an intercept in the linear mapping between averaged A/D converter values and dBZ. For the RVP7 and later, just the intercept is required. There are separate calibrations for each pulse width and polarization as applicable. A thorough discussion of the LOG receiver channel calculations is covered in the *Signal Processor User's Manual*.

6.2.1 Invoking Zcal

Command

```
zcal
```

6.2.2 Zcal Commands and Prompts

Zcal displays the stored and reference calibration information for each pulse width, then prompts you to enter a command. The number of pulse widths may vary, depending on your system. For a large number of pulse widths, you will want to use a large width terminal window. All dates shown are in local time as configured on your computer.

```
------ Horizontal Calibration ------
```

	Sto 1.0us	Ref 1.0us	
Slope:	0.5084	0.5084	dB/LSB
Cal:	-39.44	-39.44	dBZ
Std Dev:	0.00	0.00	
Log Noise:	0.00	0.00	A/D
Lin. Noise:	0.00	0.00	A/D
Linear RMS:	0.00	0.00	A/D
I0:	0.00	0.00	dB
Flag:	0	0	
Cal Time:	15:56:52	15:12:40	
Cal Date:	12 OCT 2002	10 DEC 2001	
Siggen Date:	15 JAN 2000	15 JAN 2000	
Siggen ID:	19550	19550	
Current Siggen ID:		19550	
Current Siggen cal date:		15 JAN 2000	

```
ZCAL command (? for help):
```

The table below lists the commands that you can enter at the prompt:

Cal	Enter a new calibration reflectivity
Exit or Quit	Exit from the utility
ID	Set the signal generator ID
List	Show the current numbers on the screen again.
Polar	Change polarizations (only on dual pol radars)
Refer	Set reference to stored values
Slope	Enter a new calibration slope
Write	Write the calibration file
?	Display help on the commands

6.2.3 Changing LOG Receiver Calibration Numbers

To change the calibration reflectivity:

1. Type `cal` at the command prompt. Zcal displays the prompt:

   ```
   Enter pulse width to change
   ```

2. Enter the pulse width in microseconds (for example, 1, 1.0, 1.00). Zcal displays the prompt:

   ```
   Enter new calibration number
   ```

3. Enter the calibration reflectivity (dBZo) in dB. Typically this is a negative dB number, such as -35, which corresponds to the minimum detectable dBZ at 1 km.

   Zcal redisplays the calibration information, showing the new calibration reflectivity that you entered.

To change the calibration slope:

1. Type `slope` at the command prompt. Zcal displays the prompt:

   ```
   Enter new calibration slope
   ```

2. Enter the calibration slope, usually a positive number in the range 0.2 to 0.6.

   Zcal redisplays the calibration information, showing the new calibration slope that you entered.

To change the reference values:

1. Type `refer` at the command prompt.

   Zcal copies the current calibration to the reference calibration and redisplays the calibration information with the new reference information settings. You should issue this command when you are happy with the current calibration. IRIS/Open always uses the current calibration when configuring the DSP. It uses the reference calibration only when performing an automatic calibration. If the new calibration deviates too much from the reference, it is not used.

To write the calibration file:

1. Type `write` at the command prompt.

   Zcal writes the calibration information to the file, then prints the message:

   ```
   File updated successfully.
   ```

To switch polarizations:

Zcal writes displays information for only one polarization at a time. If you radar is capable of transmitting in either horizontal or vertical polarization, then both channels can be calibrated. This is normally done in **zauto**. To switch which polarization is display, type "polar".

CHAPTER 7

QUALITY CONTROL

7.1 Thresholding of Polarization Parameters

The thresholding of polarization parameters by the processor eliminates bins with weak or uncertain signals. Note that the thresholding can be disabled if it is desired to see all of the data regardless of the data quality.

All of the polarization parameters are based on power ratios. The RVP900 requires that each power term in a ratio pass a signal-to-noise test similar to the log power test. For example, there are up to four different powers that can be calculated (alternating dual-channel case) so the tests for each of these are:

$$\frac{<|S_{hh}|^2>}{N_h} > N_{thresh}$$

$$\frac{<|S_{hv}|^2>}{N_h} > N_{thresh}$$

$$\frac{<|S_{vv}|^2>}{N_h} > N_{thresh}$$

$$\frac{<|S_{vh}|^2>}{N_h} > N_{thresh}$$

where the linearized threshold that is input as the dB LOG threshold, that is:

$$N_{thresh} = 10^{LOGthresh/10}$$

For example, a valid LDRH requires both a valid S_{hh} and a valid S_{vh}. The parameters RhoH and PhiH have the same requirement since they are the magnitude and phase of the cross-correlation function which is based on S_{hh} and S_{vh}.

There are two exceptions:

- **Z_{dr}**

  Z_{dr} requires that both Shh and Svv pass the signal-to-noise tests noted above. However, Z_{dr} can be additionally thresholded by any of the other threshold parameters (LOG, SIG, SQI, CSR) similar to a standard moment. See *RVP900 User's Manual - Digital Receiver and Signal Processor* for a description of the standard moment thresholding.

- **PhiDP for single channel alternating case**

  PhiDP requires that both Shh and Svv pass the signal-to-noise tests noted above. In the single channel alternating case, PhiDP must also satisfy the additional test that the Doppler velocity at the range bin must be valid, that is, not thresholded by its own criteria. This is because the algorithm for PhiDP in this case essentially subtracts the phase change due to the Doppler velocity. If the Doppler velocity is uncertain, the algorithm cannot produce reliable results.

CHAPTER 8
REFERENCES

1. Aydin, K., and V.Giridhar, 1992: C-band Dual-Polarization Radar Observables in Rain. *J. of Atmospheric and Oceanic Technology*, **9**, 383-390.
2. Bringi, V.N., T.D. Keenan, and V. Chandrasekar, 2001: Correcting C-Band Radar Reflectivity Data for Rain Attenuation: A Self-Consistent Method with Constraints. *IEEE Transactions on Geoscience and Remote Sensing,* **39**, 1906-1915.
3. Bringi, V. N., and V. Chandrasekar, 2001: Polarimetric Doppler Weather Radar: Principles and Applications. *Cambridge UP*, 636.
4. Carey, L.D., S.A. Rutledge, D.A. Ahijevych, and T.D. Keenan, 2000: Correcting Propagation Effects in C-Band Polarimetric Radar Observations of Tropical Convection using Differential Phase. *J. of Applied Meteorology*, **39**, 1405-1433.
5. Chandrasekar, V., and Y. Wang, 2009: Adaptive Specific Differential Phase at Dual-Polarization Radar International Application Published under the Patent Cooperation Treaty. WO 2009/114868 A1.
6. Gorgucci, E., V. Chandrasekar, and L. Baldini, 2006: Correction of X-Band Radar Observation for Propagation Effects Based on the Self-Consistency Principle. *J. of Atmospheric and Oceanic Technology*, **23**, 1668-1681.
7. Gorgucci, E., and V. Chandrasekar, 2005: Evaluation of Attenuation Correction Methodology for Dual-Polarization Radars: Application to X-Band Systems. *J. of Atmospheric and Oceanic Technology*, **22**, 1195-1206.
8. Gorgucci, E., G. Scarchilli, P.F. Meischner, M. Hagen, and V. Chandrasekar, 1998: Intercomparison of Techniques to Correct for Attenuation of C-Band Weather Radar Signals. *J. of Applied Meteorology*, **37**, 845-853.

9. Gourley, J.J., P. Tabary, and J. Parent du Chatelet, 2006: Empirical Estimation of Attenuation from Differential Propagation Phase Measurements at C Band. *J. of Applied Meteorology and Climatology*, **46**, 306-317.
10. Harker, B.J., 2006: Dynamic Range Enhancements in Radar Systems. *Proc. of 3rd EMRS DTC Technical Conference,* Edinburg.
11. Harker, B.J., 2008: Dynamic Range Enhancements for Radars and RF Systems. *Proc. of 5th EMRS DTC Technical Conference,* Edinburg.
12. Hitschfeld, W., and J. Bordan, 1954: Errors Inherent in the Radar Measurement of Rainfall at Attenuating Wavelenghts. *J. of Applied Meteorology*, **11**, 58-76.
13. Houze, R.A., 1997: Stratiform Precipitation in Regions of Convection: A Meteorological Paradox? *Bulletin of the American Meteorological Society*, **78.10**, 2179-2196.
14. Jameson, A. R., 1991: The Effect of Temperature on Attenuation-Correction Schemes in Rain Using Polarization Propagation Differential Phase Shift. *J. of Applied Meteorology*, **31**, 1106-1118.
15. Keenan, T.D., K. Glasson, F. Cummings, T.S. Bird, J. Keeler, and J. Lutz, 1998: The BMRC/NCAR C-Band Polarimetric (C-POL) Radar System. *J. of Atmospheric and Oceanic Technology*, **15**, 871-886.
16. Laakso, T.I., Jan. 1996: Splitting the Unit Delay. *IEEE Signal Processing Magazine*, **13.1**, 30-60.
17. Le Bouar, E., J. Testud, and T.D. Keenan, 2001: Validation of the Rain Profiling Algorithm "ZPHI" from the C-Band Polarimetric Weather Radar in Darwin. *J. of Atmospheric and Oceanic Technology*, **18**, 1819-1837.
18. Lim, S., V. Chandrasekar, and V.N. Bringi, 2005: Hydrometeor Classification System Using Dual Polarization Radar Measurements: Model Improvements and In Situ Verification. *IEEE Transactions on Geoscience and Remote Sensing,* **43.4**, 792-801.
19. Liu, H., and V. Chandrasekar, 2000: Classification of Hydrometeors Based on Polarimetric Radar Measurements: Development of Fuzzy Logic and Neuro-Fuzzy Systems, and In Situ Verification. *J. of Atmospheric and Oceanic Technology*, **17.2**, 140-164.
20. Mehenni, A, E. Obligis, and J. Testud, 1999-2002: Technique for Estimating Rainfall from Meteorological Radar with Polarization Diversity European/French. Worldwide and U.S: Patents PCT/FR99/ 00134, WO99/ 38028, US 6473026 B1.
21. Panov, S., R. Keranen, and V. Chandrasekar, 2008: Assessment of the Polarimetric Attenuation Correction Implementation in the RVP8 Signal Processor. *5th European Conference on Radar in Meteorology and Hydrology*, Helsinki, Finland.

22. Park, H., A.V. Ryzhkov, D.S. Zrnic, and K.-E. Kim, 2008: The Hydrometeor Classification Algorithm for the Polarimetric WSR-88D: Description and Application to an MCS. *Weather and Forecasting*, **24**, 730-748.
23. Park, S.-G., V.N. Bringi, V. Chandrasekar, M. Maki, and K. Iwanami, 2005: Correcting of Radar Reflectivity and Differential Reflectivity for Rain Attenuation at X-Band, Part I: Theoretical and Empirical Basis. *J. of Atmospheric and Oceanic Technology*, **22**, 1621-1632.
24. Park, S.-G., M. Maki, K. Iwanami, V.N. Bringi, and V. Chandrasekar, 2005: Correcting of Radar Reflectivity and Differential Reflectivity for Rain Attenuation at X-Band, Part II: Evaluation and Application. *J. of Atmospheric and Oceanic Technology*, **22**, 1633-1655.
25. Puhakka, T. et al., 2006: University of Helsinki Research Radar Setup. 4*th European Conference on Radar Meteorology and Hydrology*, Barcelona, Spain.
26. Ryzhkov, A.V., T.J. Schuur, D.W. Burgess, P.L. Heinselman, S.E. Giangrande, and D.S. Zrnić, 2005: The Joint Polarization Experiment: Polarimetric Rainfall Measurements and Hydrometeor Classification. *Bulletin of the American Meteorological Society*, **86.6**, 809-24.
27. Ryzhkov, A.V., P. Zhang, D. Hudak, J.L. Alford, M. Knight, and J.W. Conway, 2007: Validation of Polarimetric Methods for Attenuation Correction at C-Band. *Proc. of the 33rd Conference on Radar Meteorology*, American Meteorological Society, Cairns, Australia.
28. Schuur, T. et al., 2003: Observations and Classifiction of Echoes with Polarimetric WSR-88D Radar. Rep. NOAA, Norman, OK, University of Oklahoma. *Web. <http://www.cimms.ou.edu/~schuur/jpole/JPOLE_HCA_report_pdf.pdf>*.
29. Smyth, T.J., and A.J. Illingworth, 1998: Correction for Attenuation of Radar Reflectivity using Polarization Data. *Quarterly Journal of the Royal Meteorological Society*, **124**, 2393-2415.
30. Straka, J.M., D.S. Zrnić, and A.V. Ryzhkov, 2000: Bulk Hydrometeor Classification and Quantification Using Polarimetric Radar Data: Synthesis of Relations. *J. of Applied Meteorology*, **39.8**, 1341-1372.
31. Vulpiani, G., P. Tabary, J. Parent-Du-Chatelet, and J.J. Gourley, 2006: Extensive Comparison of Various Candidates for Operational Attenuation Correction with a C-Band Polarimetric Radar. *Proc. of the 4th European Conference on Radar in Meteorology and Hydrology*, Barcelona, Spain.

32. Vulpiani, G., P. Tabary, J. Parent-du-Chatelet, O. Bouquet, M.-L. Segond, and F.S. Marzano, 2007: Hail Detection using a Polarimeric Algorithm at C-Band: Impact on Attenuation Correction. *Proc. of the 33rd Conference on Radar Meteorology*, American Meteorological Society, Cairns, Australia.
33. Waldvogel, A., B. Federer, and P. Grimm, 1979: Criteria for the Detection of Hail Cells. *J. of Applied Meteorology*, **18.12**, 1521-1525. See also Foote, G.B. at al ibid.
34. Wang, Y., and V. Chandrasekar, 2009: Algorithm for Estimation of the Specific Differential Phase. *J. of Atmospheric and Oceanic Technology*, **26**, 2565-2578.
35. Zrnić, D.S., A.V. Ryzhkov, J.M. Straka, Y. Liu, and J. Vivekanandan, 2001: Testing a Procedure for Automatic Classification of Hydrometeor Types. *J. of Atmospheric and Oceanic Technology*, **18.6**, 892-913.
36. Cypress Semiconductor Corp., 2005: Dual Slope Dynamic Range Expansion. *Web: <http://www.industry.su/Cypress/02/ dual_slope_dynamic_range_expansion_technical_brief_14.pdf>*.
37. Radio Frequency Systems: Technical Description RF Cables. *Web: <http://www.rfsworld.com/userfiles/pdf/technical-description-rfcables.pdf>*
38. The Vision and Image Science Lab, Israel Institute of Tech: Adaptive Sensitivity. *Web: <http://visl.technion.ac.il/research/isight/AS/>*.

APPENDIX A
GLOSSARY

This appendix contains the glossary.

CSU	Colorado State University
CSU-CHILL	Radar research facility at CSU http://lab.chill.colostate.edu/chill-technical.html)
DP	Dual Polarimetric
FFT	A major signal processing mode of rvp900 based on Fast Fourier Transform; see also PPP
GC/AP	Ground clutter and anomalous propagation
GUI	Graphical user interface
IF	Intermediate frequency
IRIS	Interactive Radar Information System
$IRIS_ROOT	Entry point of the IRIS/RDA install tree
JPOLE	Joint Polarization Experiment
K_{dp}	Specific Differential Phase Shift
MBF	Membership function
ML	Melting layer (height)
NSSL	National Severe Storms Laboratory, Oklahoma, USA
PhiDP	Differential Propagational Phase Shift
PPP	A major signal processing mode of rvp900 based on pulse pair correlations; see FFT

RDA	Radar Data Acquisition software used within the RVP900
RhoHV	Correlation Coefficient
RS	Rule Strength
RVP900	Digital IF Receiver and Signal Processor
STAR	Simultaneous Transmit and Receive
Z	Reflectivity (Z_H - horizontal, Z_V - vertical, Z_{HV}-horizontal and vertical)
Z_{dr}	Differential Reflectivity

INDEX

C
Calibration
    log receiver 180
D
Dual polarization
    calibration 35, 175
    modes 14
    notation 14
    radar systems 12
    standard moments 24
    thresholding 183
Dual polarization , modes
    alternating dual channel 23
    alternating single channel 24
    fixed transmit 18
    simultaneous transmit 20
K
KDP
    description 9
L
LDR
    description 11
LOG
    receiver
        calibration 180
P
PhiDP
    description 9
R
RhoHV, description 11
S
STAR 20
Z
ZDR
    description 9

www.vaisala.com

M211452EN