NCAR/UNIDATA

CfRadial Data File Format

Proposed
CF-compliant netCDF Format
for Moments Data for RADAR and LIDAR
in Radial Coordinates

Version 1.0, draft 11

Mike Dixon, Wen-Chau Lee

Bob Rilling , Chris Burghart,

Joe Van Andel, Dennis Flanigan

EOL, NCAR

2011-01-18

1 Introduction

1.1 Purpose

The purpose of this document is to specify a CF-compliant netCDF format for radar and lidar moments data in radial (i.e. polar) coordinates.

The intention is that the format should, as far as possible, comply with the CF conventions for gridded data. However, the current CF 1.4 convention does not support radial radar/lidar data. Therefore, extensions to the conventions will be required.

The current CF conventions are documented at:

http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.4

1.2 Extensions to the CF convention

This convention introduces the following extensions to CF:

1. The following axis attribute types:
    - axis = "radial_azimuth_coordinate";
    - axis = "radial_elevation_coordinate";
    - axis = "radial_range_coordinate";
2. Additional standard units. The following need to be added:
    - dB (ratio of two values, in log units. For example, ZDR).
    - dBm (power in milliwatts, in log units)
    - dBZ (radar reflectivity in log units)
3. Additional standard names.

CfRadial files will be CF compliant, with the above extensions.

1.3 Strict use of variable and attribute names

However, because of the inherent complexity of radial radar and lidar data, the CfRadial format requires extra strictness, as compared to CF in general, in order to keep it manageable. There are so many metadata variables in CfRadial that it is essential to require **strict adherence** to the **dimension names** and **metadata variable names** exactly as specified in this document, in addition to their standard names. It is not practical to expect an application to search for standard names for metadata variables, since this makes the code unnecessarily complex.

Since this is a completely new format, there is no requirement to support legacy data sets which are less strictly defined.

Note that this strictness requirement only applies to **metadata** variables. The **moments data** fields will be handled as usual in CF, where the **standard name** is the **definitive guide** to the contents of the field. Suggested standard names for radar variables not yet supported by CF are listed in section 6.

One exception to this is the dimensions used to specify string variable lengths. String length dimensions may be added as needed. This document refers to the string length dimension as 'string_length', but any suitable dimension may be used in its place. See section 4.2.

1.4 _FillValue value attribute

The CF documentation mentions that the **missing_value** attribute is deprecated, and that **_FillValue** should be used instead. However, the CF document also states that **missing_value** is supported for backward compatibility purposes.

CfRadial will use the **_FillValue** attribute to indicate missing values.

Note: the CF documentation mentions that the **missing_value** attribute is deprecated, and that **_FillValue** should be used instead. However, the CF document also states that **missing_value** is supported for backward compatibility purposes.

Therefore, it is recommended that software readers check for **missing_value** as well.

1.5 Required vs. optional variables

If not otherwise stated, the inclusion of a variable is mandatory.

Some variables are optional. If so, this is stated in this document. If a variable is omitted, a reader should set the variable to missing everywhere.

2 Data Content Overview

2.1 The nature of radar and lidar moments data

As a radar or lidar scans (or points) the data fields (or **moments**) are produced over an entity specified by a time interval or angular interval.

We refer to this entity as a **ray**, **beam** or **dwell**. In this document we will use the term **ray**.

For a given ray, the field data are computed for a sequence of **ranges** increasing radially away from the instrument. These are referred to as range **gates**.

In most cases, the spacing between the range gates is constant along the ray, although this is not necessarily the case. For example, some NOAA radars have gate spacings of 75m, 150m and 300m. Therefore, we need to handle the cases for which the range gate spacing is **variable**.

2.2 Coordinate system for moments data arrays

The moments data to be handled by this format is represented in 2 principal dimensions:

- **time**: rays have monotonically increasing time
- **range**: bins have monotonically increasing range

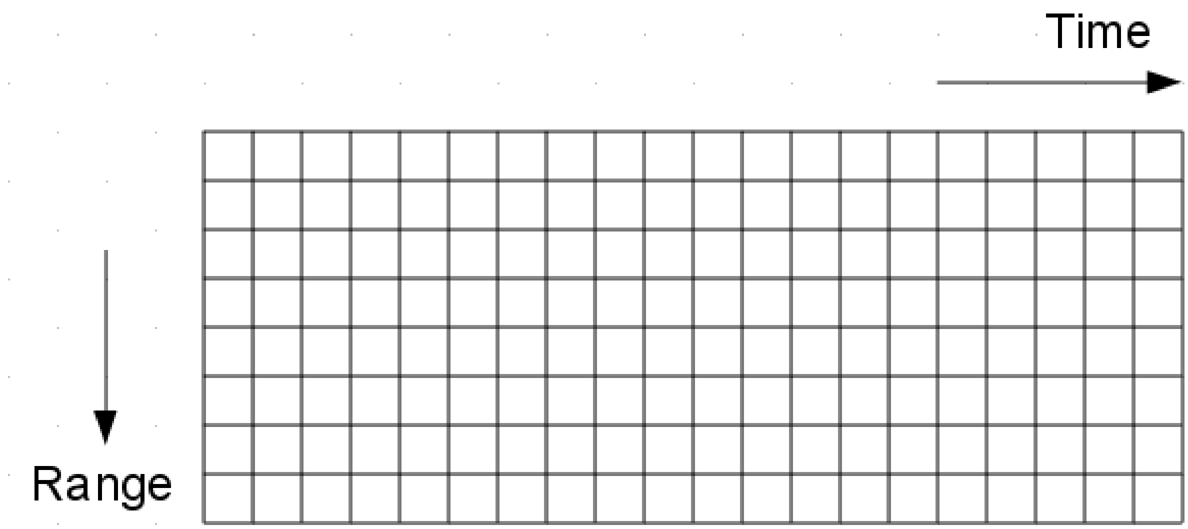

Figure 2.1 Data organization in time and range

The primary coordinate is **time** and the secondary coordinate is **range**.

The field data are stored in contiguous 1-D arrays, in which the range index varies the fastest, and the time index the slowest.

2.3 Geo-reference variables

A subset of the metadata variables in CfRadial are used to locate a radar or lidar measurement in space.

These are:

- range
- elevation
- azimuth
- latitude
- longitude
- altitude

See sections 4.4, 4.5 and 4.7 for details on these variables.

For moving platforms, extra variables are required for geo-referencing.

These are:

- heading
- roll
- pitch
- rotation
- tilt

See section 4.8 for details on these variables.

The mathematical procedures for computing data location relative to earth coordinates are described in detail in section 7.

2.4 Sweep indexes – a "pseudo" third dimension

A set of two or more related sweeps, typically a complete 3-D radar or lidar scan, is referred to as a **volume**.

A **volume scan** is comprised of one or more **sweeps**.

Scanning may be carried out in a number of different ways. For example:

- horizontal scanning at fixed elevation (PPI mode)
- vertical scanning at constant azimuth (RHI mode)
- antenna not moving, i.e. constant elevation and azimuth (staring or pointing)
- aircraft radars which rotate around the longitudinal axis of the aircraft (e.g. ELDORA)

For each of these modes a **sweep** is defined as follows:

- PPI mode: a sequence of rays at fixed elevation angle
- RHI mode: a sequence of rays at fixed azimuth angle
- pointing mode: a sequence of rays over some time period, at fixed azimuth and elevation

The **volume** may therefore be logically divided into **sweeps**. In CfRadial, we do not separate the sweeps in the stored field data arrays. Rather, we store arrays of **start** and **stop indexes**, which identify the rays that belong to each sweep. Some recorded rays may be in the transition region between defined sweeps, i.e. they may not belong to any sweep. For these rays we set the 'antenna_transition' flag to 1.

2.5 Constant range geometry per volume

For a single volume, the CF/radial convention requires that the number of range bins, and the ranges for each of the bins, be non-varying in time. Therefore, within a volume

- the number of range bins is the same for all rays
- the range geometry is the same for all rays.

If the raw data range geometry **changes over time** within a volume, the data to be represented must be re-sampled using a common **time-invariant** range geometry for the volume.

2.6 No grid mapping variable

The data in this format is saved in the native coordinate system for RADARs and LIDARs, i.e. radial (or polar) coordinates, with the instrument at the center.

A grid mapping type is not required, because the geo-reference variables contain all of the information required to locate the data in space.

For a *stationary* instrument, the following are stored as **scalar variables** (see section 4.5):

- latitude
- longitude
- altitude

Position and pointing references for *moving* platforms must take the following motions into account (see section 4.8):

- platform translation
- platform rotation

2.7 Calibration information

Radars must be calibrated to ensure that the moments data are accurate. Calibration for some types of lidar may also be applicable.

A radar may have multiple sets of calibration parameters. Generally a separate calibration is performed for each transmit **pulse width.** Separate calibrations may be performed for other reasons as well. CfRadial supports storing multiple sets of calibration parameters, using the **radar_calibration** and **lidar_calibration** conventions.

The calibration applicable to a specific ray is indicated by the **calibration_index** variable.

2.8 Compression

The netCDF 4 library supports files in the following formats:

- classic
- 64bit offset
- netcdf4
- netcdf4 classic

The **netcdf4** format is built on HDF5, which supports compression. Where data are missing or unusable, the data values will be set to a constant well-known **_FillValue** code. This procedure, combined with the use of the **netcdf4** format, provides efficient compression.

It is therefore recommended that the netcdf4 option be used whenever possible, to keep data sets as small as possible.

3 Convention hierarchy

The CF/Radial convention comprises a **base** convention, along with a series of optional **sub-conventions** for specific purposes.

At the time of writing, the following conventions are envisaged:

Convention name	Type	Description
CF/Radial	Base	Radial data extension to the CF convention. Contains all necessary information for interpreting and displaying the data fields in a geo-referenced manner
instrument_parameters	Optional	Parameters common to both radar and lidar instruments
radar_parameters	Optional	Parameters specific to radars
lidar_parameters	Optional	Parameters specific to lidars
radar_calibration	Optional	Calibration values for radars
lidar_calibration	Optional	Calibration values for lidars
platform_velocity	Optional	Velocity of the platform, in multiple dimensions
geometry_correction	Optional	Corrections to the geometry of the data set

If a netCDF file conforms to a base convention and one or more sub-conventions, these are concatenated in the Conventions attribute as a space-delimited string.

The following are some examples:

- “CF/Radial instrument_parameters”
- “CF/Radial instrument_parameters radar_parameters radar_calibration”
- “CF/Radial lidar_parameters platform_velocity”

4 *CF/Radial* base convention

The base *CF/Radial* convention covers the minimum set of elements which are required to describe a radar/lidar data set sufficiently for basic display and plotting. *CF/Radial* is a specialization of *CF*.

NOTE on units: in the following tables, for conciseness, we do not spell out the units strings exactly as they are in the netCDF file. The following abbreviations apply:

Units string in netCDF file	Abbreviation in tables
"degrees per second"	degrees/s
"meters per second"	m/s

4.1 Global attributes

Attribute name	Type	Convention	Description
Conventions	string	CF	Conventions string will specify Cf/Radial, plus selected sub-conventions as applicable
title	string	CF	Short description of file contents
institution	string	CF	Where the original data were produced
references	string	CF	References that describe the data or the methods used to produce it
source	string	CF	Method of production of the original data
history	string	CF	List of modifications to the original data
comment	string	CF	Miscellaneous information
instrument_name	string	CF/Radial	Name of radar or lidar
site_name	string	CF/Radial	Name of site where data were gathered
scan_name	string	CF/Radial	Name of scan strategy used, if applicable
platform_is_mobile	string	CF/Radial	"true" or "false"

4.2 Dimensions

Dimension name	Description
time	The number of rays. This dimension is optionally UNLIMITED
range	The number of range bins
sweep	The number of sweeps
frequency	Number of frequencies used
string_length **	Length of char type variables.

**** Note**: any number of ‘string_length’ dimensions may be created and used. For example, you may declare the dimensions ‘string_length”, ‘string_length_short’ and ‘string_length_long’, and use them appropriately for strings of various lengths. These are only used to indicate the length of the strings stored, and have no effect on other parts of the format.

4.3 Global variables

Variable name	Dimension	Type	Comments
volume_number	none	int	Volume numbers are sequential, relative to some arbitrary start time, and may wrap
platform_type	(string_length)	char	Options are: *“fixed”*, *“vehicle”*, *“ship”*, *“aircraft”*, *“aircraft_fore”*,*“aircraft_aft”*, *“aircraft_tail”*,*“aircraft_belly”*, *“aircraft_roof”*,*“aircraft_nose”*, *“satellite_orbit”*, *“satellite_geostat”*
instrument_type	(string_length)	char	Options are: “radar”, “lidar”
primary_axis	(string_length)	char	Options are: *“axis_z”*, *“axis_y”*, *“axis_x”* See section 7 for details.
time_coverage_start	(string_length)	char	UTC time of first ray in file. Format is: yyyy-mm-ddThh:mm:ssZ
time_coverage_end	(string_length)	char	UTC time of last ray in file. Format is: yyyy-mm-ddThh:mm:ssZ

4.4 Coordinate variables

Variable name	Dimension	Type	Units	Comments
time	(time)	double	seconds	Coordinate variable for time. Time at center of each ray, in fractional seconds since start_time.
range	(range)	float	meters	Coordinate variable for range. Range to center of each bin.

4.4.1 Attributes for time coordinate variable

Attribute name	Type	Value
standard_name	string	"time"
long_name	string	"time in seconds since volume start"
units	string	"seconds since *yyyy-mm-dd*T*hh:mm:ss*Z", where the actual start_time is stated

4.4.2 Attributes for range coordinate variable

Attribute name	Type	Value
standard_name	string	"projection_range_coordinate"
long_name	string	"range_to_measurement_volume"
units	string	"meters"
spacing_is_constant	string	"true" or "false"
meters_to_center_of_first_gate	float	Set to start range in meters
meters_between_gates	float	Set to gate spacing in meters Only applicable if spacing_is_constant is "true"
axis	string	"radial_range_coordinate"

4.5 Location variables

Note: for *stationary* platforms, these are *scalars*, and for *moving* platforms they are *vectors* with time.

Variable name	Dimension	Type	Units	Comments
latitude	none or (time)	double	degrees_north	Latitude of instrument. For a stationary platform, this is a scalar. For a moving platform, this is a vector.
longitude	none or (time)	double	degrees_east	Longitude of instrument. For a stationary platform, this is a scalar. For a moving platform, this is a vector.
altitude	none or (time)	double	meters	Altitude of instrument, above mean sea level. For a stationary platform, this is a scalar. For a moving platform, this is a vector.
altitude_agl	none or (time)	double	meters	Altitude of instrument above ground level. For a stationary platform, this is a scalar. For a moving platform, this is a vector. Omit if not known.

4.6 Sweep variables

Variable name	Dimension	Type	Units	Comments
sweep_number	(sweep)	int		The number of the sweep, in the volume scan. Starts at 0 each volume scan.
sweep_mode	(sweep, string_length)	char		Options are: *"sector"*,*"coplane"*,*rhi"*, *"vertical_pointing"*,*"idle"*, *"azimuth_surveillance"*, *"elevation_surveillance"*, *"sunscan"*,*"pointing"*, *"manual_ppi"*,*"manual_rhi"*
fixed_angle	(sweep)	float	degrees	Target angle for the sweep. elevation in most modes azimuth in RHI mode

Variable name	Dimension	Type	Units	Comments
sweep_start_ray_index	(sweep)	int		Index of first ray in sweep, relative to start of volume. 0-based.
sweep_end_ray_index	(sweep)	int		Index of last ray in sweep, relative to start of volume. 0-based.
target_scan_rate	(sweep)	float	degrees/s	Intended scan rate for this sweep. The actual scan rate is stored according to section 4.7. This variable is optional. Omit if not available.

NOTE: this section must always exist, even if a volume contains only a single sweep. The reason is that the sweep_mode and sweep_fixed_angle are necessary for fully understanding the sweep strategy.

4.7 Sensor pointing variables

Variable name	Dimension	Type	Units	Comments
azimuth	(time)	float	degrees	Azimuth of antenna, relative to true north.
elevation	(time)	float	degrees	Elevation of antenna, relative to the horizontal plane.
scan_rate	(time)	float	degrees/s	Antenna scan rate. Set to negative if counter-clockwise in azimuth or decreasing in elevation. Positive otherwise. Omit if not known.
antenna_transition	(time)	byte		1 if antenna is in transition, i.e. between sweeps, 0 if not. Omit if not known. If variable is omitted, the transition will be assumed to be 0 everywhere.

4.7.1 Attributes for azimuth(time) variable

Attribute name	Type	Value
standard_name	string	“azimuth angle”
long_name	string	“azimuth angle from true north”
units	string	“degrees”

Attribute name	Type	Value
axis	string	“radial_azimuth_coordinate”

4.7.2 Attributes for elevation(time) variable

Attribute name	Type	Value
standard_name	string	“elevation angle”
long_name	string	“elevation angle from horizontal”
units	string	“degrees”
axis	string	“radial_elevation_coordinate”

4.8 Moving platform geo-reference variables

For *moving* platforms, the following additional variables will be included to allow geo-referencing of the platform in earth coordinates. See section 7 for further details.

Variable name	Dimension	Type	Units	Comments
heading	(time)	float	degrees	Heading of the platform relative to true N, looking down from above.
roll	(time)	float	degrees	Roll about longitudinal axis of platform. Positive is left side up, looking forward.
pitch	(time)	float	degrees	Pitch about the lateral axis of the platform. Positive is up at the front.
drift	(time)	float	degrees	Difference between heading and track over the ground. Positive drift implies track is clockwise from heading, looking from above. NOTE: not applicable to land-based moving platforms.
rotation	(time)	float	degrees	Angle between the radar beam and the vertical axis of the platform. Zero is along the vertical axis, positive is clockwise looking forward from behind the platform.

Variable name	Dimension	Type	Units	Comments
tilt	(time)	float	degrees	Angle between radar beam (when it is in a plane containing the longitudinal axis of the platform) and a line perpendicular to the longitudinal axis. Zero is perpendicular to the longitudinal axis, positive is towards the front of the platform.

4.9 Moments field data variables

Each 2-D moments field variable has the dimensions of **(time, range).**

The field data will be stored using one of the following:

netCDF type	Byte width	Description
ncbyte	1	scaled signed integer
short	2	scaled signed integer
int	4	scaled signed integer
float	4	floating point
double	8	floating point

The netCDF variable name is interpreted as the short name for the field.

Field data variables have the following attributes:

<table>
<tr><th>Attribute name</th><th>Type</th><th>Convention</th><th>Description</th></tr>
<tr><td>long_name</td><td>string</td><td>CF</td><td>Longer name describing the field</td></tr>
<tr><td>standard_name</td><td>string</td><td>CF</td><td>CF standard name for field</td></tr>
<tr><td>units</td><td>string</td><td>CF</td><td>Units for field</td></tr>
<tr><td>_FillValue</td><td>same type as field data</td><td>CF</td><td>Used if data are missing at this range bin</td></tr>
<tr><td>scale_factor</td><td>float</td><td>CF</td><td rowspan="2">Float value =
(integer value) * scale_factor
+ add_offset</td></tr>
<tr><td>add_offset</td><td>float</td><td>CF</td></tr>
<tr><td>coordinates</td><td>string</td><td>CF</td><td>See note below</td></tr>
</table>

NOTE: the “coordinates’ attribute lists the variables needed to compute the location of a data point in space.

For stationary platforms, the coordinates attribute should be set to:

“*elevation azimuth range*”

For moving platforms, the coordinates attribute should be set to:

“*elevation azimuth range heading roll pitch rotation tilt*”

5 Sub-conventions

The base *CF/Radial* convention, as described above, covers the minimum set of netCDF elements which are required to locate radar/lidar data in time and space.

The following sub-conventions augment the base convention with additional information for various purposes.

5.1 The *instrument_parameters* sub-convention

This convention stores parameters relevant to both radars and lidars.

Variables in this convention will have the string attribute **meta_group**, set to the value "**instrument_parameters"**.

Variable name	Dimension	Type	Units	Comments
frequency	(frequency)	float	Hz	List of operating frequencies, in Hertz. In most cases, only a single frequency is used.
follow_mode	(sweep, string_length)	char		options are: *"none"*, *"sun"*, *"vehicle"*, *"aircraft"*, *"target"*, *"manual"*
pulse_width	(time)	float	seconds	
prt_mode	(sweep, string_length)	char		Pulsing mode Options are: *"fixed"*, *"staggered"*, *"dual"*
prt	(sweep) if prt constant (time) if dual prt (sweep, prt) if staggered prt	float	seconds	Pulse repetition time. For fixed prt, this is set per sweep. For dual prt, this is set per time, because prt changes every ray. For staggered prt, for each sweep list the prt values used, using the prt dimension.
prt_ratio	(sweep)	float		For dual/staggered prt mode. This is set per sweep.
polarization_mode	(sweep, string_length)	char		Options are: *"horizontal"*, *"vertical"*, *"hv_alt"*, *"hv_sim"*, *"circular"*

Variable name	Dimension	Type	Units	Comments
nyquist_velocity	(time)	float	m/s	Unambiguous velocity
unambiguous_range	(time)	float	meters	Unambiguous range
scan_rate	(time)	float	degrees/s	Antenna scan rate Set to missing if not available.
n_samples	(time)	int		Number of samples used to compute moments

The instrument_parameters convention also specifies one extra, but optional, attribute – sampling_ratio - for each field variable.

The number of samples used to compute the moments may vary from field to field. In the table above, n_samples refers to the maximum number of samples used for any field. The sampling_ratio is computed as the actual number of samples used for any field, divided by n_samples.

Attribute name	Type	Description
sampling_ratio	float	n_samples for this field divided by n_samples specified for each time (see table above)

If this attribute is missing, its value will be assumed to be 1.0.

5.2 The *radar_parameters* sub-convention

This convention handles parameters specific to radar platforms. Variables in this convention will have the string attribute **meta_group**, set to the value "**radar_parameters**".

Variable name	Dimension	Type	Units	Comments
radar_antenna_gain_h	none	float	dB	Nominal antenna gain, H polarization
radar_antenna_gain_v	none	float	dB	Nominal antenna gain, V polarization
radar_beam_width_h	none	float	degrees	Antenna beam width H polarization
radar_beam_width_v	none	float	degrees	Antenna beam width V polarization
radar_measured_transmit_power_h	(time)	float	dBm	Measured transmit power H polarization
radar_measured_transmit_power_v	(time)	float	dBm	Measured transmit power V polarization

5.3 The *lidar_parameters* sub-convention

This convention handles parameters specific to lidar platforms. Variables in this convention will have the string attribute **meta_group**, set to the value "**lidar_parameters"**.

Variable name	Dimension	Type	Units	Comments
lidar_beam_divergence	none	float	milliradians	Transmit side
lidar_field_of_view	none	float	milliradians	Receive side
lidar_aperture_diameter	none	float	cm	
lidar_aperture_efficiency	none	float	percent	
lidar_peak_power	none	float	watts	
lidar_pulse_energy	none	float	joules	

5.4 The *radar_calibration* sub-convention

Variables in this convention will have the string attribute **meta_group**, set to the convention name "**radar_calibration"**.

5.4.1 Dimensions

Dimension name	Description
r_calib	The number of calibrations available

5.4.2 Variables

The meaning of the designations used in the calibration variables are as follows for dual-polarization radars:

- '**h**': horizontal channel
- '**v**': vertical channel
- '**hc**': horizontal co-polar (h transmit, h receive)
- '**hx**' **–** horizontal cross-polar (v transmit, h receive)
- '**vc**': vertical co-polar (v transmit, v receive)
- '**vx**' **–** vertical cross-polar (h transmit, v receive)

For single polarization radars, the '**h**' quantities should be used.

Variable name	Dimension	Type	Units	Comments
r_calib_index	(time)	byte		index for the calibration that applies to each ray
r_calib_time	(r_calib, string_length)	char	UTC	e.g. 2008-09-25 T23:00:00Z
r_calib_pulse_width	(r_calib)	float	seconds	Pulse width for this calibration
r_calib_receiver_bandwidth	(r_calib)	float	MHz	Bandwidth of receiver, nominally: 1.0 / pulse_width
r_calib_ant_gain_h	(r_calib)	float	dB	Derived antenna gain H channel
r_calib_ant_gain_v	(r_calib)	float	dB	ditto, V channel
r_calib_xmit_power_h	(r_calib)	float	dBm	Transmit power H channel
r_calib_xmit_power_v	(r_calib)	float	dBm	ditto, V channel
r_calib_two_way_waveguide_loss_h	(r_calib)	float	dB	2-way waveguide loss measurement plane to feed horn H channel
r_calib_two_way_waveguide_loss_v	(r_calib)	float	dB	ditto, V channel
r_calib_two_way_radome_loss_h	(r_calib)	float	dB	2-way radome loss H channel
r_calib_two_way_radome_loss_v	(r_calib)	float	dB	ditto, V channel
r_calib_receiver_mismatch_loss	(r_calib)	float	dB	Receiver filter bandwidth mismatch loss
r_calib_radar_constant_h	(r_calib)	float	m/mW dB units	Radar constant H channel
r_calib_radar_constant_v	(r_calib)	float	m/mW dB units	ditto, V channel
r_calib_noise_hc	(r_calib)	float	dBm	Measured noise level H co-pol channel
r_calib_noise_vc	(r_calib)	float	dBm	ditto, V co-pol channel
r_calib_noise_hx	(r_calib)	float	dBm	ditto, H cross-pol
r_calib_noise_vx	(r_calib)	float	dBm	ditto, V cross-pol
r_calib_receiver_gain_hc	(r_calib)	float	dB	Measured receiver gain H co-pol channel
r_calib_receiver_gain_vc	(r_calib)	float	dB	ditto, V co-pol channel

Variable name	Dimension	Type	Units	Comments
r_calib_receiver_gain_hx	(r_calib)	float	dB	ditto, H cross-pol
r_calib_receiver_gain_vx	(r_calib)	float	dB	ditto, V cross-pol
r_calib_base_1km_hc	(r_calib)	float	dBZ	reflectivity at 1km for SNR=0dB H co-pol channel
r_calib_base_1km_vc	(r_calib)	float	dBZ	ditto, V co-pol channel
r_calib_base_1km_hx	(r_calib)	float	dBZ	ditto, H cross-pol
r_calib_base_1km_vx	(r_calib)	float	dBZ	ditto, V cross-pol
r_calib_sun_power_hc	(r_calib)	float	dBm	Calibrate sun power H co-pol channel
r_calib_sun_power_vc	(r_calib)	float	dBm	ditto, V co-pol channel
r_calib_sun_power_hx	(r_calib)	float	dBm	ditto, H cross-pol
r_calib_sun_power_vx	(r_calib)	float	dBm	ditto, V cross-pol
r_calib_noise_source_power_h	(r_calib)	float	dBm	Noise source power H channel
r_calib_noise_source_power_v	(r_calib)	float	dBm	ditto, V channel
r_calib_power_measure_loss_h	(r_calib)	float	dB	Power measurement loss in coax and connectors H channel
r_calib_power_measure_loss_v	(r_calib)	float	dB	ditto, V channel
r_calib_coupler_forward_loss_h	(r_calib)	float	dB	Coupler loss into waveguide H channel
r_calib_coupler_forward_loss_v	(r_calib)	float	dB	ditto, V channel
r_calib_zdr_correction	(r_calib)	float	dB	corrected = measured + correction
r_calib_ldr_correction_h	(r_calib)	float	dB	corrected = measured + correction
r_calib_ldr_correction_v	(r_calib)	float	dB	corrected = measured + correction
r_calib_system_phidp	(r_calib)	float	degrees	System PhiDp, as seen in drizzle close to radar
r_calib_test_power_h	(r_calib)	float	dBm	Calibration test power H channel
r_calib_test_power_v	(r_calib)	float	dBm	ditto, V channel

Variable name	Dimension	Type	Units	Comments
r_calib_receiver_slope_hc	(r_calib)	float		Computed receiver slope, ideally 1.0 H co-pol channel
r_calib_receiver_slope_vc	(r_calib)	float		ditto, V co-pol channel
r_calib_receiver_slope_hx	(r_calib)	float		ditto, H cross-pol
r_calib_receiver_slope_vx	(r_calib)	float		ditto, V cross-pol

5.5 The *lidar_calibration* sub-convention

Variables in this convention will have the string attribute **meta_group**, set to the value **“lidar_calibration”**.

At the time of writing, this convention has not been defined.

5.6 The *platform_velocity* sub-convention

For *moving* platforms, the following additional variables will be included to indicate the velocity of the platform at each time.

Variables in this convention will have the string attribute **meta_group**, set to the value **“platform_velocity”**.

Variable name	Dimension	Type	Units	Comments
eastward_velocity	(time)	float	m/s	EW velocity of the platform. Positive is eastwards.
northward_velocity	(time)	float	m/s	NS velocity of the platform. Positive is northwards.
vertical_velocity	(time)	float	m/s	Vertical velocity of the platform. Positive is up.
eastward_wind	(time)	float	m/s	EW wind at the platform location. Positive is eastwards.
northward_wind	(time)	float	m/s	NS wind at the platform location. Positive is northwards.
vertical_wind	(time)	float	m/s	Vertical wind at the platform location. Positive is up.
heading_rate	(time)	float	degrees/s	Rate of change of heading
roll_rate	(time)	float	degrees/2	Rate of change of roll of the platform
pitch_rate	(time)	float	degrees/s	Rate of change of pitch of the platform.

5.7 The *geometry_correction* sub-convention

The following additional variables will be included to quantify errors in the georeference data for the platform. These are constant for a data set.

Variables in this convention will have the string attribute **meta_group**, set to the value **"geometry_correction"**.

Variable name	Dimension	Type	Units	Comments
azimuth_correction	none	float	degrees	correction to azimuth values
elevation_correction	none	float	degrees	correction to elevation values
range_correction	none	float	degrees	correction to range values
longitude_correction	none	float	degrees	correction to longitude values
latitude_correction	none	float	degrees	correction to latitude values
pressure_altitude_correction	none	float	meters	correction to pressure altitude values
radar_altitude_correction	none	float	meters	correction to radar altitude values
eastward_ground_speed_correction	none	float	m/s	correction to EW ground speed values
northward_ground_speed_correction	none	float	m/s	correction to NS ground speed values
vertical_velocity_correction	none	float	m/s	correction to vertical velocity values
heading_correction	none	float	degrees	correction to heading values
roll_correction	none	float	degrees	correction to roll values
pitch_correction	none	float	degrees	correction to pitch values
drift_correction	none	float	degrees	correction to drift values
rotation_correction	none	float	degrees	correction to rotation values

Variable name	Dimension	Type	Units	Comments
tilt_correction	none	float	degrees	correction to tilt values

6 Standard names

To the extent possible, CfRadial uses standard names already defined by CF.

Section 6.1 lists the proposed standard names for metadata variables, and section 6.2 lists the proposed standard names for moments data.

6.1 Proposed standard names for metadata variables

Variable name ***Standard name***	**Units**	**Already supported in CF?**
altitude_agl *altitude_above_ground_level*	meters	no
altitude_correction *altitude_correction*	meters	no
altitude *altitude*	meters	yes
antenna_transition *antenna_is_in_transition_between_sweeps*	unitless	no
azimuth_correction *azimuth_angle_correction*	degrees	no
azimuth *beam_azimuth_angle*	degrees	no
drift_correction *platform_drift_angle_correction*	degrees	no
drift *platform_drift_angle*	degrees	no
eastward_velocity_correction *platform_eastward_velocity_correction*	m/s	no
eastward_velocity *platform_eastward_velocity*	m/s	no
eastward_wind *eastward_wind_speed*	m/s	yes
elevation_correction *beam_elevation_angle_correction*	degrees	no
elevation *beam_elevation_angle*	degrees	no
time_coverage_end *data_volume_end_time_utc*	seconds	no
fixed_angle *target_fixed_angle*	degrees	no
follow_mode *follow_mode_for_scan_strategy*	unitless	no

Variable name ***Standard name***	**Units**	**Already supported in CF?**
frequency *radiation_frequency*	s-1	no
heading_change_rate *platform_heading_angle_rate_of_change*	degrees	no
heading_correction *platform_heading_angle_correction*	degrees	no
heading *platform_heading_angle*	degrees	no
instrument_name *name_of_instrument*	unitless	no
instrument_type *type_of_instrument*	unitless	no
latitude_correction *latitude_correction*	degrees	no
latitude *latitude*	degrees_east	no
lidar_aperture_diameter *lidar_aperture_diameter*	meters	no
lidar_aperture_efficiency *lidar_aperture_efficiency*	unitless	no
lidar_beam_divergence *lidar_beam_divergence*	radians	no
lidar_constant *lidar_calibration_constant*	unitless	no
lidar_field_of_view *lidar_field_of_view*	radians	no
lidar_peak_power *lidar_peak_power*	watts	no
lidar_pulse_energy *lidar_pulse_energy*	joules	no
longitude_correction *longitude_correction*	degrees	no
longitude *longitude*	degrees_east	no
northward_velocity_correction *platform_northward_velocity_correction*	m/s	no
northward_velocity *platform_northward_velocity*	m/s	no
northward_wind *northward_wind*	m/s	yes

Variable name ***Standard name***	**Units**	**Already supported in CF?**
nyquist_velocity *unambiguous_doppler_velocity*	m/s	no
n_samples *number_of_samples_used_to_compute_moments*	unitless	no
pitch_change_rate *platform_pitch_angle_rate_of_change*	degrees	no
pitch_correction *platform_pitch_angle_correction*	degrees	no
pitch *platform_pitch_angle*	degrees	yes
platform_is_mobile *platform_is_mobile*	unitless	no
platform_type *platform_type*	unitless	no
polarization_mode transmit_receive_*polarization_mode*	unitless	no
prt_mode *transmit_pulse_mode*	unitless	no
pressure_altitude_correction *pressure_altitude_correction*	meters	no
primary_axis *primary_axis_of_rotation*	unitless	no
prt *pulse_repetition_frequency*	/s	no
prt_ratio *multiple_pulse_repetition_frequency_ratio*		no
pulse_width *transmitter_pulse_width*	seconds	no
radar_antenna_gain_h *nominal_radar_antenna_gain_h_channel*	dB	no
radar_antenna_gain_v *nominal_radar_antenna_gain_v_channel*	dB	no
radar_beam_width_h *half_power_radar_beam_width_h_channel*	degrees	no
radar_beam_width_v *half_power_radar_beam_width_v_channel*	degrees	no
radar_transmit_power_h *radar_transmit_power_h_channel*	dBm	no
radar_transmit_power_v *radar_transmit_power_v_channel*	dBm	no

Variable name ***Standard name***	**Units**	**Already supported in CF?**
range_correction *range_to_center_of_measurement_volume_correction*	meters	no
range *range_to_center_of_measurement_volume*	meters	no
roll_correction *platform_roll_angle_correction*	degrees	no
roll *platform_roll_angle*	degrees	yes
rotation_correction *beam_rotation_angle_relative_to_platform_correction*	degrees	no
rotation *beam_rotation_angle_relative_to_platform*	degrees	no
r_calib_antenna_gain_h *calibrated_radar_antenna_gain_h_channel*	dB	no
r_calib_antenna_gain_v *calibrated_radar_antenna_gain_v_channel*	dB	no
r_calib_base_dbz_1km_hc *radar_reflectivity_at_1km_at_zero_snr_h_co_polar_channel*	dBZ	no
r_calib_base_dbz_1km_hx *radar_reflectivity_at_1km_at_zero_snr_h_cross_polar_channel*	dBZ	no
r_calib_base_dbz_1km_vc *radar_reflectivity_at_1km_at_zero_snr_v_co_polar_channel*	dBZ	no
r_calib_base_dbz_1km_vx *radar_reflectivity_at_1km_at_zero_snr_v_cross_polar_channel*	dBZ	no
r_calib_coupler_forward_loss_h *radar_calibration_coupler_forward_loss_h_channel*	dB	no
r_calib_coupler_forward_loss_v *radar_calibration_coupler_forward_loss_v_channel*	dB	no
r_calib_index *calibration_data_array_index_per_ray*	unitless	no
r_calib_ldr_correction_h *calibrated_radar_ldr_correction_h_channel*	dB	no
r_calib_ldr_correction_v *calibrated_radar_ldr_correction_v_channel*	dB	no
r_calib_noise_hc *calibrated_radar_receiver_noise_h_co_polar_channel*	dBm	no
r_calib_noise_hx *calibrated_radar_receiver_noise_h_cross_polar_channel*	dBm	no
r_calib_noise_vc *calibrated_radar_receiver_noise_v_co_polar_channel*	dBm	no

Variable name ***Standard name***	**Units**	**Already supported in CF?**
r_calib_noise_vx *calibrated_radar_receiver_noise_v_cross_polar_channel*	dBm	no
r_calib_noise_source_power_h *radar_calibration_noise_source_power_h_channel*	dBm	no
r_calib_noise_source_power_v *radar_calibration_noise_source_power_v_channel*	dBm	no
r_calib_power_measure_loss_h *radar_calibration_power_measurement_loss_h_channel*	dB	no
r_calib_power_measure_loss_v *radar_calibration_power_measurement_loss_v_channel*	dB	no
r_calib_pulse_width *radar_calibration_pulse_width*	seconds	no
r_calib_radar_constant_h *calibrated_radar_constant_h_channel*	(m/mW)dB	no
r_calib_radar_constant_v *calibrated_radar_constant_v_channel*	(m/mW)dB	no
r_calib_receiver_gain_hc *calibrated_radar_receiver_gain_h_co_polar_channel*	dB	no
r_calib_receiver_gain_hx *calibrated_radar_receiver_gain_h_cross_polar_channel*	dB	no
r_calib_receiver_gain_vc *calibrated_radar_receiver_gain_v_co_polar_channel*	dB	no
r_calib_receiver_gain_vx *calibrated_radar_receiver_gain_v_cross_polar_channel*	dB	no
r_calib_receiver_mismatch_loss *radar_calibration_receiver_mismatch_loss*	dB	no
r_calib_receiver_slope_hc *calibrated_radar_receiver_slope_h_co_polar_channel*	unitless	no
r_calib_receiver_slope_hx *calibrated_radar_receiver_slope_h_cross_polar_channel*	unitless	no
r_calib_receiver_slope_vc *calibrated_radar_receiver_slope_v_co_polar_channel*	unitless	no
r_calib_receiver_slope_vx *calibrated_radar_receiver_slope_v_cross_polar_channel*	unitless	no
r_calib_sun_power_hc *calibrated_radar_sun_power_h_co_polar_channel*	dBm	no
r_calib_sun_power_hx *calibrated_radar_sun_power_h_cross_polar_channel*	dBm	no
r_calib_sun_power_vc *calibrated_radar_sun_power_v_co_polar_channel*	dBm	no

Variable name ***Standard name***	**Units**	**Already supported in CF?**
r_calib_sun_power_vx *calibrated_radar_sun_power_v_cross_polar_channel*	dBm	no
r_calib_system_phidp *calibrated_radar_system_phidp*	degrees	no
r_calib_test_power_h *radar_calibration_test_power_h_channel*	dBm	no
r_calib_test_power_v *radar_calibration_test_power_v_channel*	dBm	no
r_calib_time *radar_calibration_time_utc*	unitless	no
r_calib_two_way_radome_loss_h *radar_calibration_two_way_radome_loss_h_channel*	dB	no
r_calib_two_way_radome_loss_v *radar_calibration_two_way_radome_loss_v_channel*	dB	no
r_calib_two_way_waveguide_loss_h *radar_calibration_two_way_waveguide_loss_h_channel*	dB	no
r_calib_two_way_waveguide_loss_v *radar_calibration_two_way_waveguide_loss_v_channel*	dB	no
r_calib_xmit_power_h *calibrated_radar_xmit_power_h_channel*	dBm	no
r_calib_xmit_power_v *calibrated_radar_xmit_power_v_channel*	dBm	no
r_calib_zdr_correction *calibrated_radar_zdr_correction*	dB	no
scan_name *name_of_antenna_scan_strategy*	unitless	no
scan_rate *antenna_angle_scan_rate*	unitless	no
site_name *name_of_instrument_site*	unitless	no
spacing_is_constant *spacing_between_range_gates_is_constant*	unitless	no
sweep_end_ray_index *index_of_last_ray_in_sweep*	unitless	no
sweep_mode *scan_mode_for_sweep*	unitless	no
sweep_number *sweep_index_number_0_based*	unitless	no
sweep_start_ray_index *index_of_first_ray_in_sweep*	unitless	no

Variable name ***Standard name***	**Units**	**Already supported in CF?**
sweep_unambiguous_range *unambiguous_range_for_sweep*	meters	no
threshold_field_name *name_of_data_field_for_thresholding*	unitless	no
threshold_value *value_applied_to_threshold_field*	unitless	no
tilt_correction *beam_tilt_angle_relative_to_platform_correction*	degrees	no
tilt *beam_tilt_angle_relative_to_platform*	degrees	no
time *time*	seconds	no
time_coverage_start *data_volume_start_time_utc*	unitless	no
unambiguous_range *unambiguous_range*	meters	no
vertical_velocity_correction *platform_vertical_velocity_correction*	m/s	no
vertical_velocity *platform_vertical_velocity*	m/s	no
vertical_wind *upward_air_velocity*	m/s	yes
volume_number *data_volume_index_number*	unitless	no

6.2 Standard names for moments variables

Standard name	Short name	Units	Already in CF?
equivalent_reflectivity_factor		dBZ	yes
linear_equivalent_reflectivity_factor		Z	no
radial_velocity_of_scatterers_away_from_instrument		m/s	yes
spectrum_width		m/s	no
log_differential_reflectivity_hv	ZDR	dB	no
log_linear_depolarization_ratio_hv	LDR	dB	no
log_linear_depolarization_ratio_h	LDRH	dB	no
log_linear_depolarization_ratio_v	LDRV	dB	no
differential_phase_hv	PHIDP	degrees	no
specific_differential_phase_hv	KDP	degrees/km	no
cross_correlation_ratio_hv	RHOHV		no
log_power	DBM	dBm	no
log_power_co_polar_h	DBMHC	dBm	no
log_power_cross_polar_h	DBMHX	dBm	no
log_power_co_polar_v	DBMVC	dBm	no
log_power_cross_polar_v	DBMVX	dBm	no
linear_power	PWR	mW	no
linear_power_co_polar_h	PWRHC	mW	no
linear_power_cross_polar_h	PWRHX	mW	no
linear_power_co_polar_v	PWRVC	mW	no
linear_power_cross_polar_v	PWRVX	mW	no
signal_to_noise_ratio	SNR	dB	no
signal_to_noise_ratio_co_polar_h	SNRHC	dB	no
signal_to_noise_ratio_cross_polar_h	SNRHX	dB	no
signal_to_noise_ratio_co_polar_v	SNRVC	dB	no
signal_to_noise_ratio_cross_copolar_v	SNRVX	dB	no
normalized_coherent_power	NCP		no

7 Computing the data location from geo-reference variables

Weather radars and lidars rotate primarily about a *principal axis* (e.g., plan-position-indicator mode in ground-based radar about the zenith), slew about a secondary axis, orthogonal to the primary axis (e.g., range-height-indicator in ground-based radar), or slew on a plane by changing both primary and secondary axis (e.g., COPLANE in ground-based radar). In the ground-based radar convention, a point in space relative to a radar is represented in a local spherical coordinate systems $\mathbf{X}_i$ by three parameters, range (r), azimuth (λ), and elevation (ϕ). A ground-based radar is assumed “leveled” with positive (negative) elevation, ϕ, above (below) a *reference plane* (a leveled plane orthogonal to the principal axis and containing the radar). The azimuth angle, λ, is the angle on the reference plan increases clockwise from the True North (TN) following the Meteorological coordinate convention (e.g., TN is 0 degree and East is 90 degree). Further processing and manipulating radar data (e.g., interpolation, synthesis, etc) typically are performed in a Cartesian coordinate systems $\mathbf{X}$ (a right-handed XYZ, geo-reference, coordinate systems) where Y is North and X is East (Fig. 7.1). Hence, a coordinate transformation between $\mathbf{X}_i$ (radar sampling space) and $\mathbf{X}$ (geo-reference space) is required.

Based on the principal axes, remote sensors can be classified into three types, X, Y, or Z type. The purpose of this chapter is two-fold: (1) to define a consistent terminology, and (2) to derive coordinate transformation matrices for each type of remote sensor. Many sensors (e.g. fixed ground radars) are of the Z-type, have a fixed location, are leveled and are aligned relative to True North (TN). Dealing with such sensors is much simpler than for those on moving platforms. Therefore, they will be dealt with first, and the more complicated treatment of all three types of remote sensor mounted on moving platforms will be covered in the later sections.

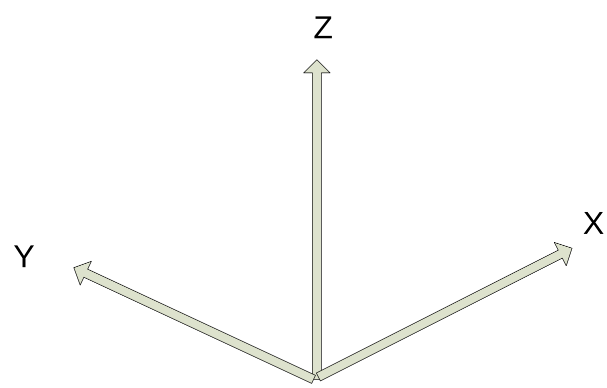

Figure 7.1: Right-handed XYZ coordinate system.

7.1 Special case – ground-based, stationary and leveled sensors

Z-type sensors (radars and lidars) rotate primarily about the vertical (Z) axis, and the reference plane is a horizontal plane passing through the sensor. The Y axis is aligned with TN, and the X axis points East.

Azimuth angles (λ) are positive clockwise looking from above (positive principal axis), with 0 being TN.

Elevation angles (ϕ) are measured relative to the horizontal reference plane, positive above the plane and negative below it.

7.1.1 LIDARs

For LIDARs, the assumption is generally made that propagation of the beam is along a straight line, emanating at the sensor. The coordinate transformation between X_i (r, λ, ϕ) and **X** (x, y, z) is as follows:

$$x = x_0 + r \cos\phi \sin\lambda$$

$$y = y_0 + r \cos\phi \cos\lambda$$

$$z = z_0 + r \sin\phi$$

where

x is positive east

y is positive north

(x_0, y_0, z_0) are the coordinates of the sensor relative to the Cartesian grid origin and the azimuth angle (λ) is the angle clockwise from TN.

The sensor location is specified in longitude, latitude and altitude in the CfRadial format. Locations in the earth's geo-reference coordinate system are computed using the sensor location and the (x,y,z) from above, using normal spherical geometry.

7.1.2 RADARs

The propagation of radar microwave energy in a beam through the lower atmosphere is affected by the change of refractive index of the atmosphere with height. Under average conditions this causes the beam to be deflected downwards, in what is termed 'Standard Refraction'. For most purposes this is adequately modeled by assuming that the beam is in fact straight, relative to an earth which has a radius of 4/3 times the actual earth radius. (Rinehart 2004.)

For a stationary and leveled, ground-based radar, the equations are similar to those for the LIDAR case, except that we have one extra term, the height correction, which reflects the beam curvature relative to the earth.

The height above the earth's surface for a given range is:

$$h = \sqrt{r^2 + R'^2 + 2rR'\sin(\phi)} - R' + h_0$$

See Rinehart 2004, chapter 3, for more details.

The (x,y) location for a given range is:

$$x = x_0 + r \cos\phi \sin\lambda$$

$$y = y_0 + r \cos\phi \cos\lambda$$

where x is positive east, y is positive north, and remembering that azimuth is the angle clockwise from true north.

7.2 Moving platforms

For moving platforms, the metadata for each beam will include:

- longitude of instrument
- latitude of instrument
- altitude of instrument
- rotation and tilt of the beam (see section 7)
- roll, pitch and heading of the platform
- platform motion (U_G, V_G, W_G)
- air motion (U_{air}, V_{air}, W_{air})

For ground-based moving platforms (e.g., Doppler on Wheels), the earth-relative location of the observed point is:

$$x = x_0 + r\cos\phi\sin\lambda$$

$$y = y_0 + r\cos\phi\cos\lambda$$

$$h = \sqrt{r^2 + R^2 + 2rR'\sin\phi} - R' + h_0$$

Note that for airborne radar platforms, correcting for refractive index does not apply. Therefore, for airborne radars, use the straight line equations for LIDARs.

Refer to the sections below for the computation of elevation (ϕ) and azimuth (λ) relative to earth coordinates.

Then apply the following equations, as before, to compute the location of the observed point.

$$x = x_0 + r\cos\phi\sin\lambda$$

$$y = y_0 + r\cos\phi\cos\lambda$$

$$z = z_0 + r\sin\phi$$

7.3 Coordinate transformations for the general case

This section details the processing for the general case.

Instruments which do not fall under section 7.3 above must be handled as a general case.

7.3.1 Coordinate systems

In addition to the previously-defined $\mathbf{X_i}$ and $\mathbf{X}$ coordinate systems, the following intermediate right-handed coordinate systems need to be defined to account for a moving, non-leveled platform:

- $\mathbf{X}_a$: platform-relative coordinates, +Y points to heading, +X points to the right side, +z is orthogonal to the reference plane XY.
- $\mathbf{X}_h$: leveled, platform heading-relative coordinates, +Y points heading, +X points heading+90°, and Z points up (local zenith).

The goal here is to derive transformations from $\mathbf{X_i}$ to **X** via $\mathbf{X}_a$ and $\mathbf{X}_h$.

7.3.2 The earth-relative coordinate system

The earth-relative coordinate system, **X**, is defined as follows, X is East, Y is North, and Z is zenith. Azimuth angle, λ, is defined as positive *clockwise* from North (i.e., meteorological angle) while elevation angle, ϕ, is defined positive/negative above/below the horizontal plane at the altitude (h_0) of the remote sensor.

7.3.3 The platform-relative coordinate system

The general form of the mathematic representation describes a remote sensing device mounted on a moving platform (e.g., an aircraft, see Figure 7.2). This figure depicts the theoretical reference frame for a moving platform. (We use the aircraft analogy here, but the discussion also applies to water-borne platforms and land-based moving platforms.)

The platform-relative coordinate system of the platform, $\mathbf{X}_a$, is defined by the right side, (X_a), the heading, (Y_a), and the zenith, (Z_a).

The origin of $\mathbf{X}_a$ is defined as the location of the INS on a moving platform.

The platform-relative coordinate system is defined by 3 rotations in the following angles: pitch (P), roll (R) and heading (H). These angles are generally measured by an inertial navigation system (INS).

The platform moves relative to X, based on its heading H, and the drift D, caused by wind or current. (D is 0 for land-based platforms). The track T is the line of the platform movement over the ground.

NOTE: -see Lee et al. (1994) for further background on this topic, and on the corrections to Doppler velocity for moving platforms. Usually, the platform INS and the sensor may not be collocated. See Lee et al. (1994) for discussions on how to compensate for the Doppler velocity due to the relative motion between these two.

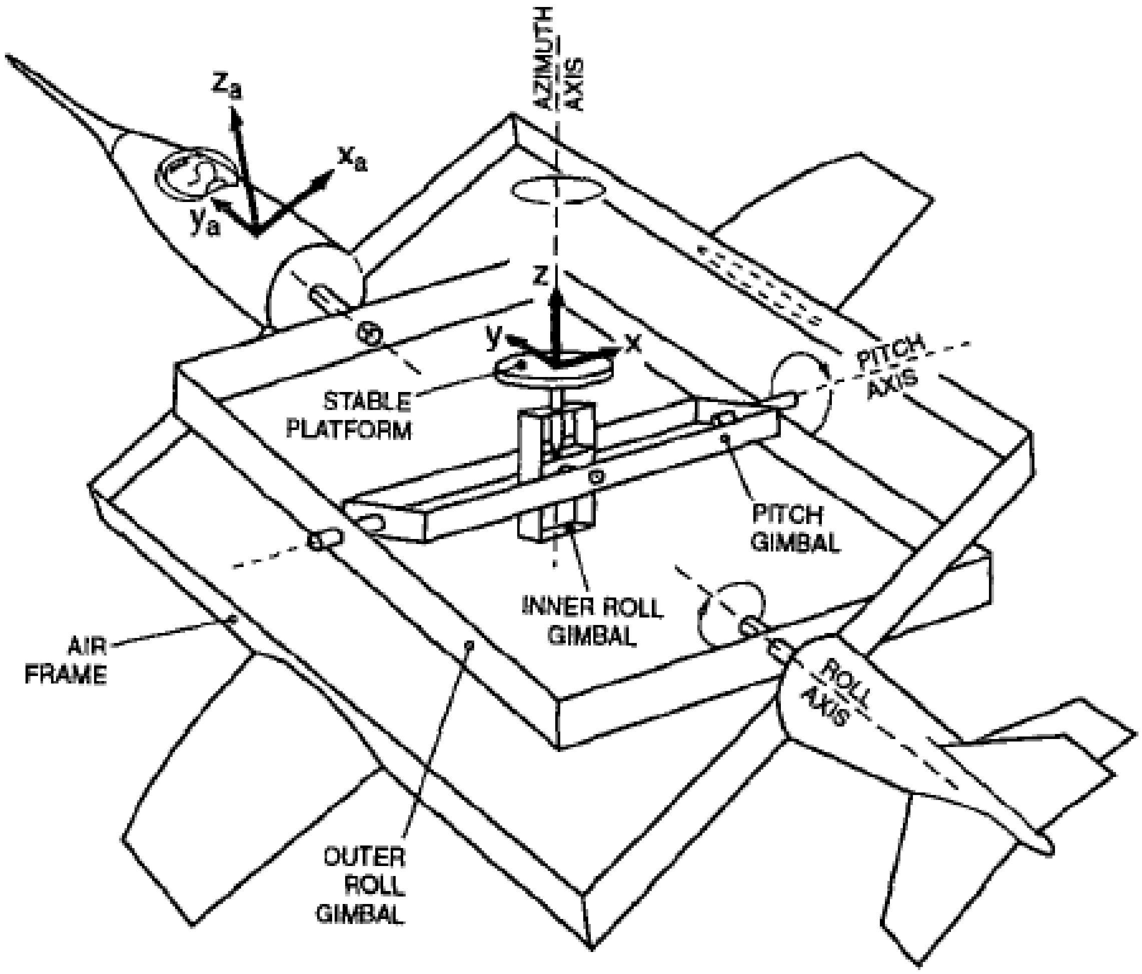

Figure 7.2 Moving platform axis definitions and reference frame (reproduced from Lee et al., 1994,originally from Axford, 1968) ©American Meteorological Society. Reprinted with permission.

Figures 7.3 a through c show the definitions of heading, drift, track, pitch and roll.

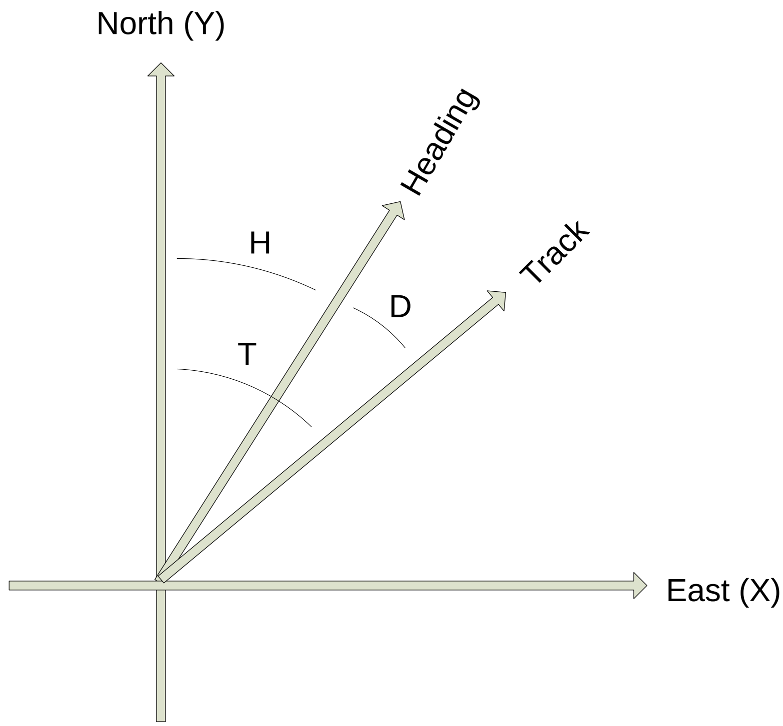

Figure 7.3(a): Definition of heading, drift and track.

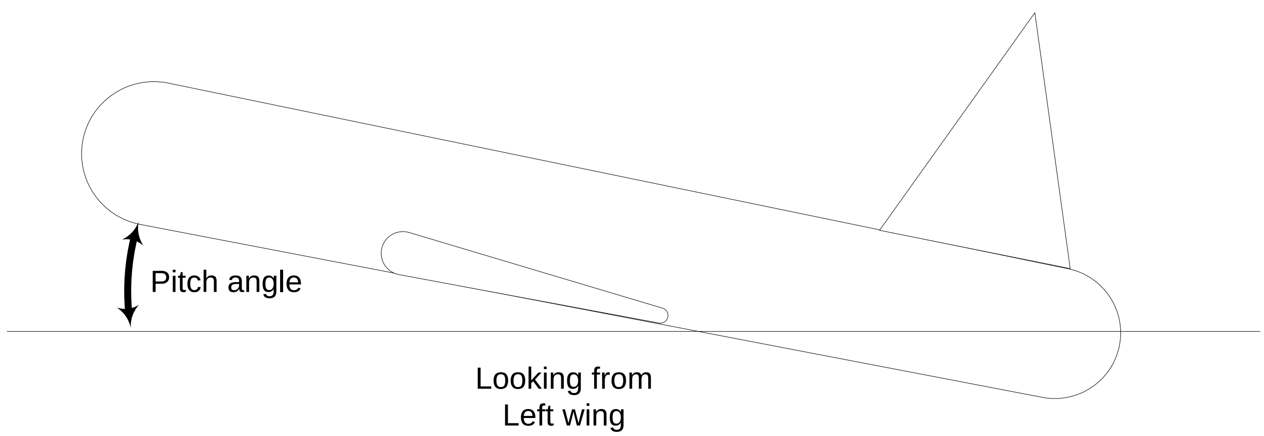

Figure 7.3(b): Definition of pitch

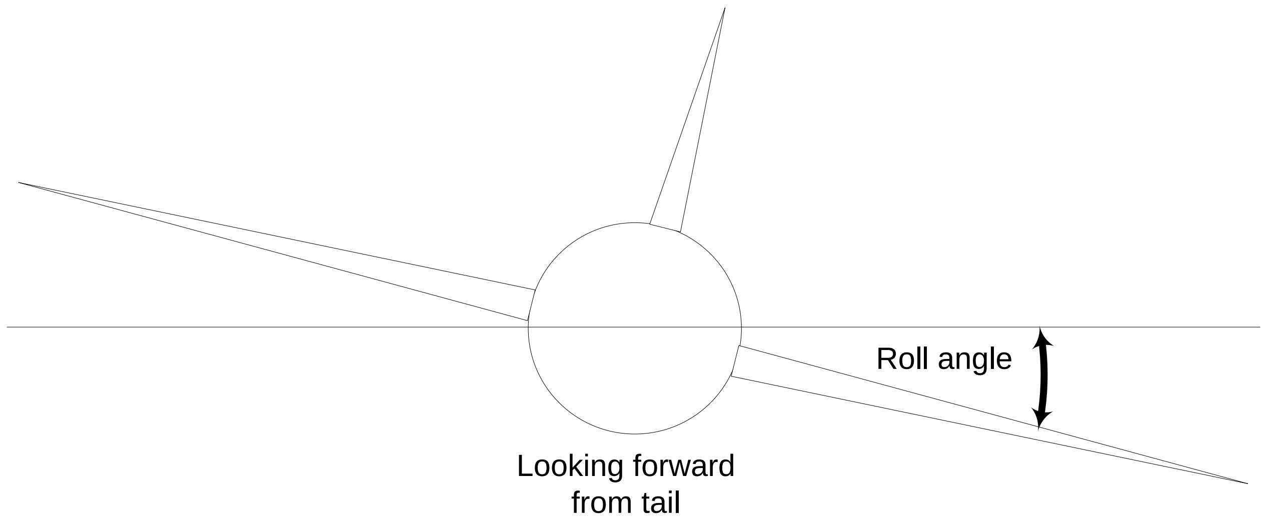

Figure 7.3(c): Definition of roll

7.3.4 The sensor coordinate system

In the sensor coordinate system, $\mathbf{X}_i$, each gate location is characterized by a range, *r*, a rotation angle, θ, and a tilt angle, τ. Following the ground-based radar convention, the rotation angle, θ, is the angle projected on the reference plane, positive *clockwise* from the third axis (counting from the principal axis in $\mathbf{X}_a$) looking *towards the sensor* from the positive principal axis. The tilt angle, τ, is the angle of the beam relative to the reference plane. A beam has a positive/negative τ depending on whether it is on the positive/negative side of the reference plane, using the principal axis to determine the sign. Each gate location (r, θ, τ) in $\mathbf{X}_i$ can be represented in (r, λ, ϕ) in **X**.

Table 7.1: Characteristics of 3 types of sensors.

Sensor Type	Type X	Type Y	Type Z
Principal Axis	X_a	Y_a	Z_a
Reference Plane	Y_aZ_a	Z_aX_a	X_aY_a
0° Rotation Angle	$+Z_a$	$+X_a$	$+Y_a$
90° Rotation Angle	$+Y_a$	$+Z_a$	$+X_a$
Examples	Ground-based radar/lidar, aircraft nose radar, downward scanning radar on Global Hawk, NOAA P3 lower-fuselage radar and C-band scatterometer	Tail Doppler radars on NOAA P3 and NSF/NCAR ELDORA	EDOP, Wyoming Cloud Radar

7.4 Coordinate transformation sequence

The following transformations are carried out to transform the geometry from the instrument-based (**X**$_i$) to the earth-based coordinate system (**X**):

- translate from **X**$_i$ to **X**$_a$
- rotate from **X**$_a$ to **X**

7.4.1 Transformation from X_i to X_a

The details of this step depend on the sensor type: Z, Y or X (Table 7.1)

7.4.1.1 Type Z sensors

The characteristics are:

- the primary axis is Z_a
- the reference plane is (X_a, Y_a)
- the rotation angle θ is 0 in the (Y_a, Z_a) plane, i.e. along the +Y axis. Rotation increases clockwise from +Y, when looking from above (i.e. from +Z)
- the tilt angle (τ) is 0 in the (X_a, Y_a) plane, positive above it (for +Z_a) and negative below it.

The transformation to X_a coordinates is:

$$\begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix} = r \begin{pmatrix} \sin\theta\cos\tau \\ \cos\theta\cos\tau \\ \sin\tau \end{pmatrix}$$

7.4.1.2 Type Y sensors

The characteristics are:

- the primary axis is Y_a
- the reference plane is (Z_a, X_a)
- the rotation angle θ is 0 in the (Z_a, X_a) plane, i.e. along the +X_a axis. Rotation increases clockwise from +X, when looking from +Y.
- the tilt angle (τ) is 0 in the (Z_a, X_a) plane, positive for +Y_a.

Note that the definition of θ is different from the convention defined in Lee et al. (1994)[1]. Let θ' is the rotation angle defined in Lee et al. (1994), θ=mod(450°- θ').

[1] The rotation angle, θ', defined in previous airborne tail Doppler radar convention (Lee et al. 1994) was positive clockwise looking from the tail toward the nose of an aircraft (i.e., looking from the -Y_a-axis) that has been the convention for airborne tail Doppler radars. However, this

The transformation to X_a coordinates is:

$$\begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix} = r\begin{pmatrix} \cos\theta\cos\tau \\ \sin\tau \\ \sin\theta\cos\tau \end{pmatrix}$$

7.4.1.3 Type X sensors

The characteristics are:

- the primary axis is X_a
- the reference plane is (Y_a, Z_a)
- the rotation angle θ is 0 in the (Y_a, Z_a) plane, i.e. along the +Z_a axis. Rotation increases clockwise from +Z_a, when looking from +X_a.
- the tilt angle (τ) is 0 in the (Y_a, Z_a) plane, positive for +X_a.

The transformation to X_a coordinates is:

$$\begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix} = r\begin{pmatrix} \sin\tau \\ \sin\theta\cos\tau \\ \cos\theta\cos\tau \end{pmatrix}$$

7.4.2 Rotating from X_a to X

Rotating $\mathbf{X}_a$ to **X** requires the following 3 steps:

- remove the roll R, by rotating the x axis around the y axis by –R.
- remove the pitch P, by rotating the y axis around the x axis by –P.
- remove the heading H, by rotating the y axis around the z axis by +H

The transformation matrix for removing the roll component is:

$$M_R = \begin{pmatrix} \cos R & 0 & \sin R \\ 0 & 1 & 0 \\ -\sin R & 0 & \cos R \end{pmatrix}$$

The transformation matrix for removing the pitch component is:

$$M_P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos P & -\sin P \\ 0 & \sin P & \cos P \end{pmatrix}$$

The transformation matrix for removing the heading component is:

convention is different from that used in the ground-based radars. The r and τ were defined the same way in the current convention.

$$M_H = \begin{pmatrix} \cos H & \sin H & 0 \\ -\sin H & \cos H & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

We apply these transformations consecutively:

$$X = M_H M_P M_R X_a$$

$$M_H M_P M_R = \begin{pmatrix} \cos H & \sin H & 0 \\ -\sin H & \cos H & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos P & -\sin P \\ 0 & \sin P & \cos P \end{pmatrix} \begin{pmatrix} \cos R & 0 & \sin R \\ 0 & 1 & 0 \\ -\sin R & 0 & \cos R \end{pmatrix}$$

$$= \begin{pmatrix} \cos H \cos R + \sin H \sin P \sin R & \sin H \cos P & \cos H \sin R - \sin H \sin P \cos R \\ -\sin H \cos R + \cos H \sin P \sin R & \cos H \cos P & -\sin H \sin R - \cos H \sin P \cos R \\ -\cos P \sin R & \sin P & \cos P \cos R \end{pmatrix}$$

$$= \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

7.5 Summary of transforming from X_i to X

We combine the above 2 main steps for transform all the way from the instrument coordinates to earth coordinates:

7.5.1 For type Z radars:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} r \begin{pmatrix} \sin\theta \cos\tau \\ \cos\theta \cos\tau \\ \sin\tau \end{pmatrix}$$

$$= r \begin{pmatrix} m_{11} \sin\theta \cos\tau + m_{12} \cos\theta \cos\tau + m_{13} \sin\tau \\ m_{21} \sin\theta \cos\tau + m_{22} \cos\theta \cos\tau + m_{23} \sin\tau \\ m_{31} \sin\theta \cos\tau + m_{32} \cos\theta \cos\tau + m_{33} \sin\tau \end{pmatrix}$$

7.5.2 For type Y radars:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} r \begin{pmatrix} \cos\theta\cos\tau \\ \sin\tau \\ \sin\theta\cos\tau \end{pmatrix}$$

$$= r\begin{pmatrix} m_{11}\cos\theta\cos\tau + m_{12}\sin\tau + m_{13}\sin\theta\cos\tau \\ m_{21}\cos\theta\cos\tau + m_{22}\sin\tau + m_{23}\sin\theta\cos\tau \\ m_{31}\cos\theta\cos\tau + m_{32}\sin\tau + m_{33}\sin\theta\cos\tau \end{pmatrix}$$

7.5.3 For type X radars:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} r \begin{pmatrix} \sin\tau \\ \sin\theta\cos\tau \\ \cos\theta\cos\tau \end{pmatrix}$$

$$= r\begin{pmatrix} m_{11}\sin\tau + m_{12}\sin\theta\cos\tau + m_{13}\cos\theta\cos\tau \\ m_{21}\sin\tau + m_{22}\sin\theta\cos\tau + m_{23}\cos\theta\cos\tau \\ m_{31}\sin\tau + m_{32}\sin\theta\cos\tau + m_{33}\cos\theta\cos\tau \end{pmatrix}$$

7.5.4 Computing earth-relative azimuth and elevation

We can then compute the earth-relative azimuth and elevation as follows:

$$\lambda = \tan^{-1}(x/y)$$

$$\phi = \sin^{-1}(z/r)$$

7.6 Summary of symbol definitions

$\mathbf{X}_i$: instrument-relative coordinate system, (r, θ, τ) or (r, λ, ϕ)

$\mathbf{X}_a$: platform-relative coordinate system (x_a, y_a, z_a) – see figure 7.2

$\mathbf{X}_h$: coordinate system relative to level platform (no roll or pitch) with heading H.

X: earth-relative coordinate system (x, y, z), x is positive east, y is positive north, z is positive up.

H: heading of platform (see figure 7.3)

T: track of platform (see figure 7.3)

D: drift angle (see figure 7.3)

P: pitch angle (see figure 7.3)

R: roll angle (see figure 7.3)

λ: azimuth angle

ϕ: elevation angle

θ: rotation angle

τ: tilt angle

r: range

h: height

h_0: height of the instrument

R': pseudo radius of earth = $(4/3)6374\,km$

8 Change log

8.1 Version 1.0, draft7: 2010-06-16

This was the first version posted to the web.

8.2 Version 1.0, draft8: 2010-11-01

The following changes were made for draft 8:

Changes to section 1.4

Changed missing_value to _FillValue.

Changes to section 2.6

Removed the use of the **grid mapping** variable, since this is not really necessary for this format. The location information is provided in the location variables and is independent of any grid mapping.

Changes to section 3

Removed the ‘frequency_list’ sub-convention. The list of frequencies used is now a required member of the ‘instrument_parameters’ sub-convention.

Changes to section 4.2

Relaxed the requirement to have a single dimension to handle string length. Any number of dimensions may be added for strings of different lengths. Readers of the data must check the dimension used by each string variable to determine its length.

The ‘frequency’ dimension is now required if the ‘instrument_parameters’ sub-convention is used. The wavelength variable has been removed, and is replaced by a list of 1 or more frequencies.

Changes to section 4.3

Added “aircraft” to the platform type list.

Moved ‘instrument_type’ up into the base convention, from the ‘instrument_params’ sub-convention.

Changed start_time variable to time_coverage_start, and end_time to time_coverage_end, to conform to the NetCDF Attribute Convention for Dataset Discovery (ACDD).

Changes to section 4.5

‘altitude_agl’ is now an optional variable. Omit this variable if not known.

Changes to section 4.6

Changed ‘sweep_fixed_angle’ to ‘fixed_angle’.

Added ‘target_scan_rate’ as an optional variable. Omit this variable if not known.

Changes to section 4.7

Changed ‘antenna_transition’to be optional. Omit this variable if not known. It will then be assumed that the antenna is not in transition for any ray.

Added ‘scan_rate’ variable, which is optional. Omit this variable if the values are not known.

Changes to section 4.9

Changed missing_value to _FillValue.

Changes to section 5.1

Changed sweep_follow_mode to follow_mode.

Changed sweep_prf_mode to prt_mode.

Added prt_ratio.

Changed sweep_polarization_mode to polarization_mode.

Changes to section 5.2

Changed radar_transmit_power_h to radar_measured_transmit_power_h.

Changed radar_transmit_power_v to radar_measured_transmit_power_v.

Moved radar_receiver_bandwidth into 5.4.2.

Removed Changewavelength. Only frequencies are specified.

Added the option for more than 2 PRTs.

Relaxed the requirements – if a variable is not specified, a default value will be assumed. These are specified in the document.

Added linear reflectivity (Z) as a field with a proposed standard name.

Calibration as a variable with attributes?

Radar rotation direction? Sign to indicate dirn?

8.3 Version 1.0, draft 10: 2011-01-15

The following changes were made for draft 10:

Changes to section 7

Both the explanations and equations were modified. The concept of the instrument coordinate system ($\mathbf{X}_i$) was added.

9 References

Axford, D. N., 1968: On the accuracy of wind measurements using an inertial platform in an aircraft, and an example of a measurement of the vertical structure of the atmosphere. *J. Appl. Meteor.*, 7, 645-666.

Lee, W., P. Dodge, F. D. Marks Jr. and P. Hildebrand, 1994: Mapping of Airborne Doppler Radar Data. *Journal of Oceanic and Atmospheric Technology*, 11, 572 – 578.

Rinehart, R. E., 2004: Radar for Meteorologists, Fourth Edition. *Rinehart Publications.* ISBN 0-9658002-1-0