RDA 8.05.10 Release Notes (1 Aug 2004)

These release notes cover changes made to the SIGMET Radar Data Acquisition platform, including primarily the RVP8 and RCP8 products. The last public release was RDA-8.05.8 dated 30 June 2004. If you are upgrading from an earlier version please also read the release notes that have been published since then. Note that RDA-8.05.9 was skipped in the public sequence.

New Features

1. The **Mp** setup question *Apply amplitude correction based on Burst/COHO* is now shown only when the **Mb** setup question *PhaseLock to the burst pulse* is set to *YES*. This serves as a reminder that amplitude correction is only available when burst pulse processing is also enabled. In addition, the maximum amplitude correction has been increased from 2dB to 6dB, i.e., timeseries data will be corrected even when the transmit power is fluctuating by as much as ±6dB per pulse.

2. The (I,Q) data that are used to compute noise samples now flow through the timeseries API, thus allowing them to be examined by user code and stored as part of an overall timeseries archive. Noise sample data are collected at 256 ranges and consist of at least 256 consecutive pulses. The taskID of this collection of samples is assigned the special name "`__NOISESAMPLE`".

3. The upper bound on (I,Q) "headroom" correction (defined in the **Mp** menu) has been reduced from 6dB to 5dB because of concerns about numerical stability when too much extrapolation is attempted.

4. Transmit waveforms generated by the RVP8/Tx card are now corrected for the inherent *sin(f)/f* amplitude-vs-frequency response of the TxDAC. This results in flatter FM chirp waveforms from the card. In addition, a new setup question in the **Mz** section lets you supply additional adhoc frequency band compensation:

   `FM Chirp manual spectrum flattener: 0.00 %-per-MHz`

   The units of this parameter are: percentage amplitude compensation per megaHertz of deviation from the IF center frequency. Positive values result in a relative boost of higher frequencies across the time span of the waveform. When the overall frequency response of your Tx/Rx is not flat, you can tune this parameter to null out the error based on feedback from either the **Ps** spectrum plot or an oscilloscope display of the Tx pulse.

5. The space key may now be used to alternate between **Pa** plots even when the waveform optimization hunt is running. This allows you to flip between plots to watch the progress in different representations.

6. Tuning parameter #2 (frequency breakpoint) for the non-linear FM transmit waveform can now range from –1 to +2, rather than –1 to +1. The interval from –1 to +1 behaves just like before, but the extended range from +1 to +2 causes the "Y" endpoint to move down from the corner. This allows a broader class of waveforms to be investigated for

theoretical purposes, but in practice these additional waveforms do not actually improve the PSL or ISL levels.

7. The RCP8 now supports dual antennas that are mounted at different elevation angles on the same mechanical pedestal. Only one of the two antennas is "active" at any given time, but control can be switched back and forth between them and all Pedestal/Earth angles will be handled and reported properly. The following new setup questions have been added to "Site Misc":

```
Pedestal has an auxiliary second antenna: YES
  AZ offset from Primary-->Secondary:   0.06 deg
  EL offset from Primary-->Secondary:  88.30 deg
  Use secondary antenna beginning w/IRIS mode: 2
  Secondary antenna uses (180+AZ)&(180–EL) : YES
```

In this example, a secondary antenna is mounted 88.3 degrees above the primary unit. The primary antenna is the one that is defined and used everywhere else in the RCP8. Thus, the physical orientation of the primary antenna matches the angles reported from the physical AZ/EL sensors (plus possible fixed offsets from the "Axis" command), and the overall limits of travel are expressed in terms of the primary antenna angles. Note that we assume a traditional Elevation-over-Azimuth pedestal mount in order to use this dual-antenna feature, and also that the azimuth of the two antennas is nearly identical (within one degree of each other). Depending on the angle separation between the two antennas you may want to consider the upper one "reversed" from the lower via the transformation AZ⟶180+AZ and EL⟶180–EL.

Antenna selection is made automatically according to the requested IRIS mode, and thus can be made transparent to the rest of an operational system. The secondary antenna will be used whenever one of the higher-numbered IRIS modes is selected, in which case the new logic variable **sAltAnt** will also be TRUE. This status variable can be combined with logic equations and timers to control whatever sequence(s) of operations need to be performed when switching between antennas. You may also override **sAltAnt** in order to manually switch antennas based on other logic criteria.

8. The RCP8 "Help View" printout now includes the additive angle offsets that convert measured pedestal angles into actual antenna angles. The RCP02 used to report this same information, but it was never ported into the RCP8.

9. The RCP8 "Monitor ADC" command is now available for live monitoring of the analog input voltage levels from either the I/O-62 PCI card or the IO62CP backpanel.

10. You may now use the two relays that are mounted on the I/O-62 PCI card even when an IO62CP backpanel is selected. The following *softplane.conf* stanza can be added by hand, or automatically via *softplane –resave* .

```
# Relays from IO62 (requires jumper settings on the IO62
# which may conflict with using the above TTY/Trig lines).
#    Relay1  –  NO:J17      Arm:J18      NC:J15
#    Relay2  –  NO:J11,3    Arm:J11,4    NC:J11,2
#
splConfig.Io62[0].Opt.Cp.Relay1 = ""
splConfig.Io62[0].Opt.Cp.Relay2 = ""
```

11. The RCP8 now supports pedestals that are driven using stepper motors rather than DC motors. The following new questions appear in the *VServo* command for each axis:

```
Generate stepper motor drive control signals: YES
  Stepping calibration:  100.0 clicks/degree
```

When stepper motors are enabled, the following new *softplane.conf* logical control signals will also be active and can be hooked to outputs of the I/O-62 PCI card:

- **`cStepCntAZ`** & **`cStepCntEL`** : These are active-high pulses having a duration of at least 5µsec, at a rate that is proportional to the desired angular velocity.
- **`cStepUpAZ`** & **`cStepUpEL`** : These are active-high levels which indicate the direction of the desired angular rotation. "Up" means clockwise for azimuth and upward for elevation.

The stepper motor control signals will be inhibited whenever the RCP8 is shutdown or has been placed in LOCAL antenna mode. The signals are also inactive if the built-in antenna simulator is running on the selected axis, or if neither a position nor velocity servo are selected.

12. The options for the RCP8 toplevel *Control* command have been renamed slightly to be more clear. The choices now are: *Lines*, *Logic*, *VarMisc*, *VarADC*, and *VarAnt*.

13. The Boolean variable names representing analog input voltages now begin with "adcN" (rather than "aN"), where "N" represents the slot number of the variable.

14. A powerful new set of Boolean variables have been added to the RCP8 to allow logic equations to make decisions based on details of the antenna motion. A new setup menu is accessed via *Control VarAnt* as follows:

```
RCP> control VarAnt
Antenna Pos/Vel Test Variable Definitions
----------------------------------------
Antenna Logic Variable #0  is defined: YES
  Apply test on AZIMUTH (else ELEVATION): YES
  Apply test to VELOCITY (else POSITION): YES
  Test the DESIRED value (else MEASURED): YES
  Use PEDESTAL coordinates (else EARTH) : YES
  Test for GREATER-THAN (else LESS-THAN): YES
  Comparison test value:   6.00  deg,deg/sec
```

This menu defines up to sixteen Boolean status variables which are TRUE when any of various conditions exist at the radar antenna. In the above example we define a variable called **`ant0_desPedVelAZ_gt_6.00`** which detects when the desired pedestal AZ velocity is greater than six degrees/second. The variable name is created automatically so that it can then be used in logic equations via *Control Logic* . This class of variables is very general and can test either the AZ or EL axes for a desired or measured position or velocity, using either a greater-than or less-than test on either the pedestal or Earth coordinates.

Since these variables can detect virtually any combination of antenna commands and status, one can imagine many creative uses for them. For example, suppose we wish to

drive a pedestal whose azimuth axis uses synchronous motors that can only run at 2, 4, or 6 RPM according to two control lines (C1,C0) equal to (0,1), (1,0), and (1,1). Further suppose that control line C2 requests a single reverse rotation speed, and that C3 causes an axis brake to be applied. We start by defining four antenna sense variables:

```
ant0_desPedVelAZ_gt_0.00
ant1_desPedVelAZ_gt_18.00
ant2_desPedVelAZ_gt_30.00
ant3_desPedVelAZ_lt_0.00
```

The *ant0* variable will be TRUE whenever a positive azimuth velocity is requested, and *ant1* and *ant2* detect when that requested velocity exceeds 3RPM and 5RPM (i.e., the midpoints between the available higher motor speeds). Likewise, the *ant3* variable detects when any negative rotation is being requested. We can now combine these into the following logic equations:

```
# V4 indicates when motion is requested in either direction
  v4 = ant0_desPedVelAZ_gt_0.00 | ant3_desPedVelAZ_lt_0.00
# Timer #1 will apply the brake after 1.5 sec of inactivity
  t1_extend_1.50 = v4
# Timer #0 allows 0.5 sec delay for the brake to release
  t0_retard_0.50 = v4
# Construct the two forward motion controls in (V1,V0)
  v1 = ant1_desPedVelAZ_gt_18.00
  v0 = (ant0_desPedVelAZ_gt_0.00 & !v1) | ant2_desPedVelAZ_gt_30.00
# Gate the motion/brake requests with the timers
  c0 = v0 & t0_retard_0.50
  c1 = v1 & t0_retard_0.50
  c2 = ant3_desPedVelAZ_lt_0.00 & t0_retard_0.50
  c3 = !t1_extend_1.50
```

Whenever motion is requested the brake immediately releases while both the forward direction (C1,C0) lines and the reverse direction C2 line are delayed 0.5 seconds. Then, when the desired velocity returns to zero, the antenna will coast for 1.5 seconds after which the brake will be reapplied.

15. You may now specify up to two independent I/O-62 cards to to handle signal I/O for the RCP8. The “primary” card is the one whose solder-side flexribbon connector is plugged into the component-side connector of the “secondary” card. Both cards are fully usable via signals mapped through *softplane.conf* , but only the primary card can be used for implicit I/O such as A/D inputs, tachometer and synchro inputs, motor drive outputs, etc. Please be sure to choose your pin assignments accordingly. The following new questions in the *Site Misc* menu select zero, one, or two I/O-62 cards:

```
Primary   I/O-62 PCI card (-1:None) :  0
Secondary I/O-62 PCI card (-1:None) :  1
```

Bug Repairs

1. Pulse-to-pulse amplitude correction was not working properly with simulated data.
2. The “headroom” correction of power levels beyond the saturation level of the IFD was not working properly for the (I,Q) data stream being written to the timeseries API. The **Pr** and SNOISE powers *were* being corrected properly, however.

3. New RVP8/Tx output waveforms were not being resynthesized right away after changing the IF in the **Mb** menu (the changed IF setting would take effect only after something else caused the waveforms to be reloaded).

4. The question *Process incoming servo control packets* has been added back into the RCP8 *Site Host* menu. This feature was accidently removed in RDA-8.04.8.

5. A bug was repaired in the RDA kernel module in which blocked RVP8 threads were not necessarily being awakened when certain IOCTLs were performed. This bug surfaced only as a result of changes recently made to the IRIS/DSP driver; so there should not have been any real operational problems with the previous driver and RVP8 working together. The new V4.2 kernel module will automatically be loaded on reboot, or you can load it into the currently running kernel via stop/start of rdasys.

6. A bug was repaired in the DSP Driver's *DspResetFifo()* routine that was causing a timeout with the RVP8 roughly 1% of the time. Customers who have written their own socket interface to the RVP8 using DspExport should take a look at our routine, and should use the new "RKFF" socket command. Customers who have written their own application code using our *dspw_reset()* function with the fifo reset flag TRUE, should recode their applications to call *DspResetFifo()*.

RDA 8.05.8 Release Notes (30 June 2004)

These release notes cover changes made to the SIGMET Radar Data Acquisition platform, including primarily the RVP8 and RCP8 products. The last public release was RDA-8.05.6 generated on 2 June 2004. If you are upgrading from an earlier release, please read those notes also. Note that RDA-8.05.7 was skipped in the public sequence.

New Features

1. Both the RVP8 and RCP8 will now shutdown automatically whenever the IRIS antenna library gets shutdown via **qant**.

2. Four new setup variables have been added to the WSR88D DCU setup section of the RCP8, allowing multiplicative scale factors to be applied to AZ/EL Pos/Vel.

3. The RVP8 **Pr** plot now includes a LOG-Only option, in addition to the Samples-Only and Samples-Plus-LOG plots versus range. This allows you to see the receiver's echo response more clearly in cases where you want to focus attention on it alone. This is especially handy when viewing echoes from compressed Tx waveforms, since the Rx samples from any target will always span a wide interval of ranges and thus tend to clutter up the plot.

4. This is the first RDA release to officially support the new Rev.C RVP8/Rx and Rev.E/F RVP8/IFD hardware. These RVP8 digital receiver components have many improved features such as doubled sample rate (72MHz vs 36MHz), an integrated CAT-5E link (rather than fiber/coax), a third IF input for dual polarization and other out-of-the-box applications, plus full "reflash" capability for field upgrades. Production shipments of this new hardware will begin in August.

5. Improvements have been made in how the RVP8 represents the phase of all live (I,Q) data that are acquired from the IFD and RVP8/Rx card. It used to be that the absolute phase of both the burst pulse and receiver samples would slew predictably according to the values of the IFD acquisition clock, the radar intermediate frequency, and the PRT. While there was nothing really wrong with this (since phase measurements are relative rather than absolute) it did have the confusing effect that the absolute phase of a CW input precisely at IF would be rotating.

   To improve on this, the RVP8 now corrects all (I,Q) phases to be relative to a continuous sinusoid at IF, much like a traditional analog receiver/mixer would do. Coherent signals will thus be seen as having fixed absolute phase at all ranges and at all PRFs. A further benefit of this change is that the DC offsets measured by SNOISE are more meaningful, and indicate the presence of coherent leakage into the receiver system. These offsets can be monitored in GPARM output words #7 and #8. Note that the RVP8 does not presently subtract these offsets from the incoming (I,Q) data because blindly doing so would break many existing procedures, e.g., CW calibration of the radar. We will add this option in a future release after receiving feedback from our research customers and coming up with a proper design.

6. We have upgraded the Intel Integrated Performance Primitives (IPP) library to the latest version 4.0. This library is shipped with all RDA systems and provides many of the low level signal processing routines that are used by the RVP8 compute threads. New IPP header files can be found in the RDA *include* directory, and new PDF documentation is in *config_template/extraspdf*. This new IPP release provides improved performance on hyper-threaded processors, along with several new features that the RVP8 will begin using in the near future. The IPP version is printed in the RVP8 startup terminal and should now be "`v4.0 4.0.19.77`" rather than "`v2.0 gold SP1 2.0.6.39`".

Important: Please be sure to install a new *tplates.tgz* file when you carry out this RDA upgrade. That file is normally not needed for every upgrade, but will be required this time because the new IPP libraries are contained within it. You must also run *ldconfig* as *root* after the upgrade to cache the locations of the new libraries. All of these steps only need to be done once.

7. The *rda/intelipp/ipp_example.c* demo program now takes a *–cpu* flag that causes it to measure the CPU speed using the Intel IPP library methods.

8. The clutter filter specifications loaded by *dspw_filterSpecs()* now include the option of overriding the default data window (from the last SOPRM command) with a specific data window to be used whenever that filter is selected. For example, Filter #1 could be setup to use a Blackman window even though a Hamming window is being used everywhere else. This optional window field can also be viewed/changed in the **Mf** menu.

9. The RCP8 can now correctly read data from the Seatex INU (Inertial Navigation Unit). The antenna position stabilization is not available yet.

Bug Repairs

1. The RVP8/Tx card would fail to output transmit pulse waveforms when the Acquisition Clock frequency (**Mc** menu) and TxClk/VCXO frequency (**Mz** menu) were not precise rational multiples of each other.

2. Two bugs were repaired in ORDA BATCH mode. The High-PRF data were not being processed from exactly the correct set of pulses, and Reflectivity output data were not being supplied from the Low-PRF samples when the High-PRF data were unavailable.

3. The Clutter Microsuppression ("pop" clutter filter) feature was not working properly. All bins would be zeroed when the filter was enabled. This bug was introduced in the 8.04.2 release from 19 November 2003.

4. The PLL lock detection has been improved on the RVP8/Tx card. The lock state that is reported in the **V** command had sometimes been incorrect.

5. The last point of data in a **Pr** plot of samples versus range was sometimes incorrect.

6. Velocities were inverted when compressed transmit waveforms were used.

7. The legacy form of the RVP8 LFILT opcode was not properly ignoring bits in the upper half of each filter selection word, causing Filter-7 to be incorrectly selected in such cases.

Those high bits used to select the log clutter filter in the RVP6, but that feature is no longer required nor supported in the RVP8.

8. When using **antx** to interface to the RCP8 over the multicast socket interface, the “help listall” command did not print correctly. This problem had several different causes, depending on the speed of your computer and network. It was solved in the antenna library by increasing the chat mode buffer from 10K to 20K, and by transmitting network chat packets in groups of 20.

RDA 8.05.6 Release Notes (2 June 2004)

These release notes cover changes made to the SIGMET Radar Data Acquisition platform, including primarily the RVP8 and RCP8 products. The last public release was RDA-8.05.3 generated on 15 April 2004. If you are upgrading from an earlier release, please read those notes also.

Bug Repairs

1. In the **tsarchive** program, the TX Phase and Polarization bits were not recorded as part of the pulse header structure. This was added to both the recording and to the **tsview** display program.
2. The RVP8 *THRESH* command was not properly setting the threshold control flags for individual parameters. Only the low byte of the 16-bit flag was being stored.
3. The RVP8 *RBACK* command was not properly returning the parameters set by the *THRESH* command. The SQI and WSP fields were swapped in the readback data.
4. During trigger blanking the RVP8 was occasionally producing short CPIs (rays having very few pulses) at the leading and trailing edges of the blanked intervals. This has been repaired.

RDA 8.05.5 Release Notes (12 May 2004)

These release notes cover changes made to the SIGMET Radar Data Acquisition platform, including primarily the RVP8 and RCP8 products. The last public release was RDA-8.05.3 generated on 15 April 2004. If you are upgrading from an earlier release, please read those notes also. Note that RDA-8.05.4 was skipped in the public sequence.

New Features

1. The RCP8 now supports analog voltage inputs that can be used within logic equations as well as for Q-BITE status. There are ten analog voltage channels which are continuously sampled by the RCP8. If the *softplane.conf* file indicates that an I/O-62 board is being used directly, then channels 0–7 represent the eight differential voltages on pins 13/34, 14/35, etc., and channels 8&9 are unused. Likewise, if an IO62-CP backpanel is plugged in, then channels 0–9 represent the ten differential voltages on J8 pins 1/14, 2/15, etc.

   The analog voltages can be used to create up to sixteen different logic variables, each of which is defined by some combination of input voltages being compared with a threshold voltage. The variables are defined in the *Control ADC* menu:

```
RCP> control adc
Analog Input Test Variable Definitions
--------------------------------------
A/D Logic Variable #0  is defined: YES
  Description of A00 variable: 'HiWater '
  Input summation term #1: A3
  Input summation term #2: -A8
  Input summation term #3: Zero
  Input summation term #4: Zero
  Test for ( A3-A8  >  -5.00 Volts )
```

   This defines a logic variable named *A00_HiWater* which will be true whenever the voltage difference between analog input channels #3 and #8 exceeds negative five volts. Up to four analog input channels can be summed to create each logic variable.

   The analog inputs can also be sent to the host computer every 2.5 seconds by bundling them into a Q-BITE packet consisting of ten 3-byte values. Each value holds a 21-bit signed integer representing the corresponding channel's measured input in millivolts. The overall packet length is SYNC+ID+(3x10)+END = 33 bytes, and the packets are configured using the *site host* menu:

```
RCP> si ho
Host Computer Setups
--------------------
...
RCP8 transmits analog voltage Q-BITE packets: YES
  ID of analog voltage Q-BITE packets: 0x0F
```

2. The *rdaflash* utility now prints progress messages while it is programming and verifying each board. This gives some reassuring feedback during lengthy (several minute) cycles.

3. The RCP8 will now default to a shutdown state whenever its power-up diagnostics fail in any way. This matches the way the RCP02 used to work. Also, Bit-5 of Byte-7 of the internal BITE packet is now set whenever a power-up error occurred.

Bug Repairs

1. The RCP8 was driving its *cPWidth[3:0]* output lines with the 2-bit binary representation of the current pulsewidth, rather than with a 1-of-4 single-bit decoded format. This has been fixed, and now all four output bits are properly driven, i.e., one and only one will be high at any given time.

Important: If your are using the cPWidth output lines in the previous incorrect format, you will have to make some minor changes to your softplane.conf file. What's most likely is that cPWidth[0] is being used as a single control bit in a dual pulsewidth system, in which case simply inverting it with a "~" prefix will restore proper operation.

2. The RVP8 was not blanking triggers properly based on AZ/EL angle when those angles were not being sampled live by the RVP8 itself. This has been fixed so that sectors are properly blanked regardless of the AZ/EL angle source.

3. The *cTrigBlank* softplane output signal is now generated properly by the RVP8. This signal used to be low all the time.

RDA 8.05.3 Release Notes (15 Apr 2004)

These release notes cover changes made to the SIGMET Radar Data Acquisition platform, including primarily the RVP8 and RCP8 products. The last release was RDA–8.05.2 generated on 29 March 2004. If you are upgrading from an earlier release, please read those notes also.

New Features

1. The RVP8 can now design optimized non-linear FM transmit waveforms that have clean bandlimited envelopes and superb range sidelobe suppression. Figure-1 shows an actual **Pr** plot of a 40µsec, 5MHz optimized waveform generated by the RVP8/Tx card and fed into the IFD. Compare this plot with the non-optimized linear FM waveform shown in Figure-3 of the previous 8.05.2 Release Notes.

   In this example, the ideal Tx waveform has a Peak Sidelobe Level (PSL) of –76.7dB and an Integrated Sidelobe Level (ISL) of –62.3dB. The measured testbench performance is several dB short of this, probably because of the uncompensated analog bandpass filters on the RVP8/Tx and IFD. These filters have several tenths of a dB of amplitude ripple as well as minor deviations from linear phase within the 5MHz signal bandwidth. The effect is that the sampled analog waveform is not quite identical to the ideal waveform. We will be developing code to apply appropriate pre-emphasis of the transmit waveform in a future RVP8 release.

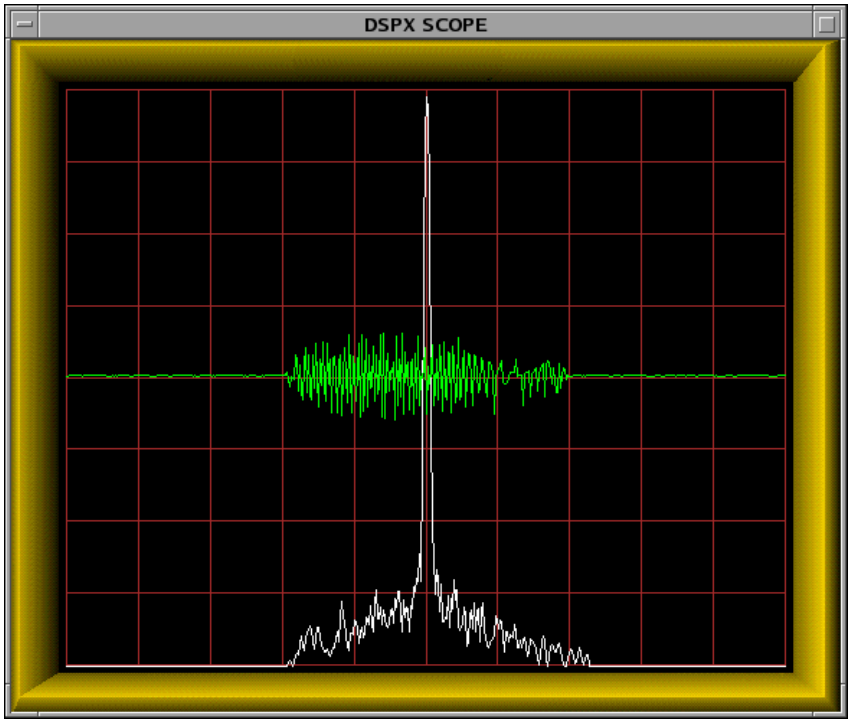

Fig.1 : *IFD Sampling of Optimized Compressed RVP8/Tx Waveform Displayed in **Pr** Plot*

2. The RVP8 now supports a new “Plot Ambiguity” **Pa** command in which compressed transmit waveforms can be designed, studied, and optimized. Figure-2 shows one form

of **Pa** plot in which the magnitude of the Tx/Rx range sidelobes are drawn on a log scale having 10dB ticks. Only half of the horizontal plot is shown because the zero-Doppler response (white plot) can safely be assumed to be symmetric. The pulse length is 30μsec, bandwidth is 3MHz, PSL is –61.2dB and ISL is –50.8dB.

Also shown in yellow and green are the Tx/Rx responses when the overall waveform is modified by a 50KHz target Doppler shift. Real weather targets would never have such a large Doppler component, but the **Pa** menu allows you to study its effect anyway.

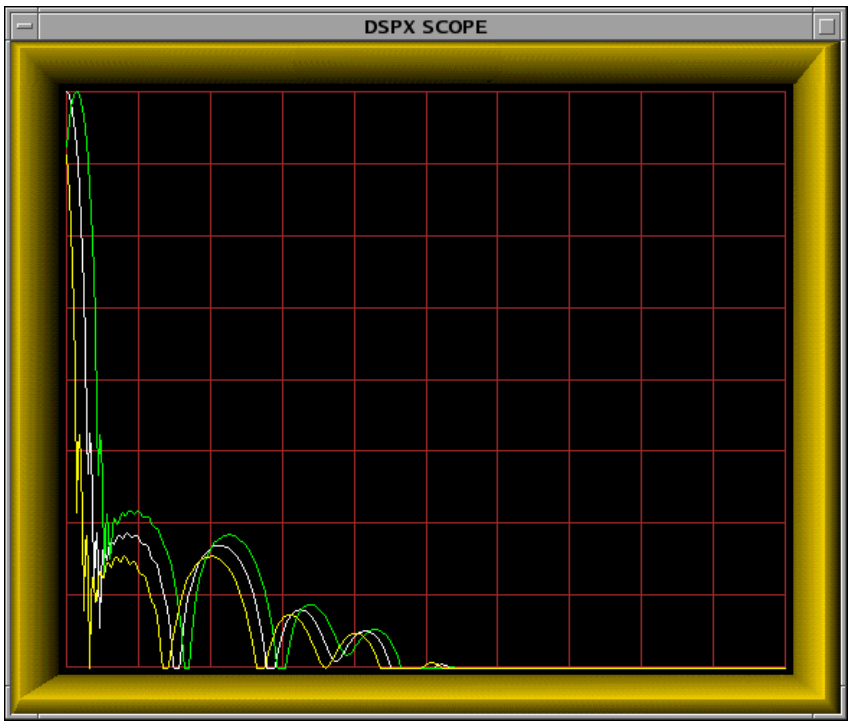

Fig.2 : *Ambiguity Diagram of a Compressed Tx Pulse Displayed in **Pa** Plot*

An alternate form of **Pa** plot of the same Tx waveform is shown in Figure-3. The horizontal axis again represents time, but now spans the entire duration of the pulse. Three different plots are drawn, hence the vertical axis is interpreted differently in each case:

- The instantaneous frequency across the full length of the pulse is shown in white. The vertical scale is normalized to hold the overall frequency span, which is also shown numerically in the **Pa** TTY output.
- The waveform baseband phase is shown in green, and is normalized so that the vertical axis holds the full span of values. Note that the phase, which generally spans a few thousand degrees, is “unwound” in this plot so that you can see its behavior nicely.
- The amplitude of the Tx waveform envelope is shown in yellow. It is drawn using a linear vertical scale which occupies only the middle half of the plot. This is to avoid creating too much “plotting clutter” in the corners.

From Figure-3 we can see that the waveform consists of a linear FM chirp that occupies about 87% of the pulse length. The frequency remains nearly constant in the leading and trailing edges, hence the overall label "Non-Linear FM". The central chirp is contained within a somewhat larger amplitude envelope that applies full scale power within the middle 82% of the pulse, and also provides bandlimited shaping of the leading and trailing edges.

This waveform was designed using the "$" automatic search-and-optimize command in the **Pa** menu. For a given pulse length and bandwidth of the Tx waveform, this command allows you to try thousands of combinations of FM shape and amplitude shape, searching for the combination that minimizes the sum of PSL and ISL (in dB). This gives the best overall waveform for weather radar observations in which both the PSL and ISL are important.

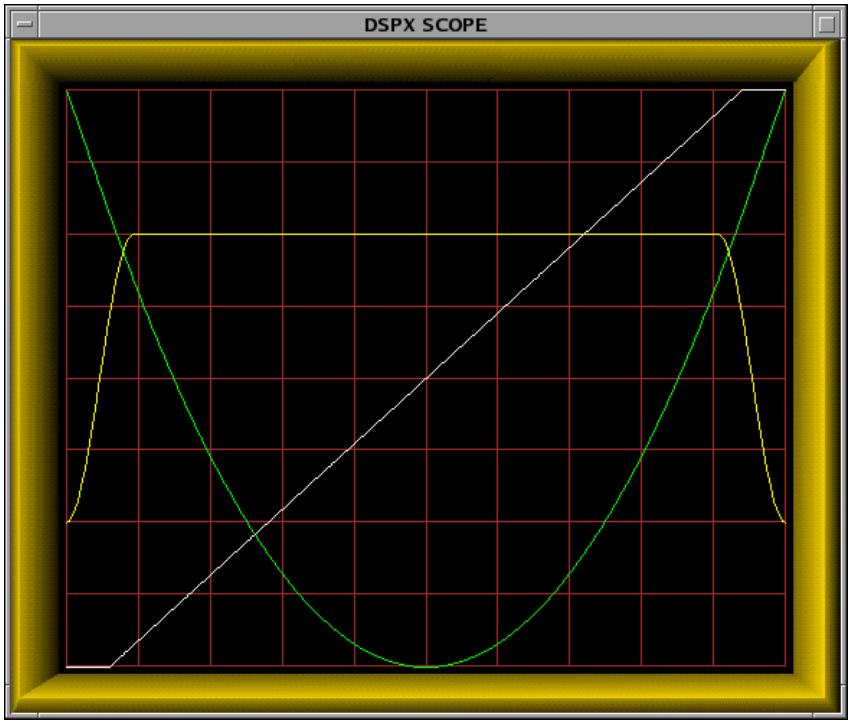

Fig.3 : *Frequency (white), Phase (green), and Amplitude (yellow) of a Compressed Tx Pulse Displayed in **Pa** Plot*

3. Improvements have been made in the ability to generate RVP8 triggers whose start times depend on their PRT. The start times defined in the **Mt<n>** menu now consist of a fixed time offset, plus a fraction of the previous PRT, plus a fraction of the next PRT. What's new here is that you now specifically select which of the two PRTs to use, e.g.,

```
Start: 10.00 usec + (-0.500000 * prevPRT) + (0.000000 * nextPRT)
```

By typing one, two or three new values at the prompt, you can change the fixed offset, previous fraction, and next fraction for each trigger's start time. Note that the fractional values are only shown when at least one of them is nonzero.

In the above example, the trigger will start 10μsec after the midpoint of the previous pulse and the current pulse. The start time is computed as a negative fraction of the previous PRT, and thus positions the pulse properly in the negative time region. If we wanted a pulse that was half way between the current pulse and the next one, we would use +0.5 times *nextPRT* instead.

Important Licensing Change

1. Effective in RDA release 8.05.3 the time series archiving program called **tsarchive** requires an RDA license to run. If you are a customer who has purchased this feature, please request a license from SIGMET. To do this: First Set your sitename in the **setup** License section. Then type "show_machine_code". Then email the output to sigmet so we can generate your license.

Bug Repairs

1. Several Makefiles in the RDA tree were accidently using 'rm –r' where 'rm –f' should have been used.
2. Turning AFC On/Off from the **Mb** menu would not take effect until that setting was saved and the RVP8 was restarted.

RDA 8.05.2 Release Notes (29 Mar 2004)

These release notes cover changes made to the SIGMET Radar Data Acquisition platform, including primarily the RVP8 and RCP8 products. The last release was RDA–8.05.1 generated on 8 March 2004. If you are upgrading from an earlier release, please read those notes also.

We are very happy to announce that 8.05.2 is the first release that supports pulse compression on the RVP8. This is an exciting new addition that will open the door to using low power coherent transmitters more extensively in the weather radar community. There are many future directions for this project, so we expect it to be somewhat of a "work in progress" as customers begin using compressed waveforms over the next few months. We look forward to your observations and suggestions.

New Features

1. The RVP8 now supports the new Rev.D RVP8/Tx card, which has improved PLL signal integrity for generating the master synthesis clock. The change will be most noticeable in phase modulated systems where the Tx waveforms are being compared with other radar system clocks over time periods greater than ten milliseconds.

2. The RVP8/Tx now synthesizes your requested output frequencies exactly, not merely to the nearest 1-Hz or so. Thus, if you're locking the 81MHz VCXO to a 10MHz external reference and you ask for a 60MHz Tx channel output, an oscilloscope will show the 10MHz and 60MHz signals locked dead together.

3. The RVP8/Tx internal API has been completely rewritten to support arbitrary waveform synthesis in a very general manner. Each major mode defines a *txWaveformDesign()* entry point that produces a complex valued baseband modulation for the transmitted IF carrier. That same routine also generates the matched receiver coefficients. The RVP8 handles all of the details of mixing these complex values Up/Down to accommodate the real Transmitter/Receiver, as well as applying pulse-to-pulse Tx phase modulation and creating the received (I,Q) data stream.

4. RVP8/Tx waveforms can now be loaded in a few tens of microseconds. Previously they would require several milliseconds to load up.

5. The amplitude weighting of the leading and trailing edges of RVP8/Tx pulses is much more precisely controlled than before. The envelope is defined by the *Bandwidth of transmit pulse* parameter in the **MTn** menu for each pulsewidth. Please note that it is the ***total*** time duration of the synthesized pulse (including the rolloff at each edge) that is defined by the *Pulselength of transmit pulse* parameter. This convention is generally easier to apply than some of the alternatives such as "3dB points" or "full power span". You may need to tweak some of your old pulselength settings and Tx trigger widths to comply with this new definition.

6. The **Pb** plotting command now has the option of showing the ideal baseband transmitted waveform when a compressed pulse is being synthesized. This is primarily useful as a quick visual check on the correctness of the waveform design routines. An example is

shown in Figure-1 for a 40µsec linear FM pulse having a bandwidth of 4MHz. The real part of the complex waveform produced by *txWaveformDesign()* is displayed. Note that the IF "carrier" is not present in this baseband modulation.

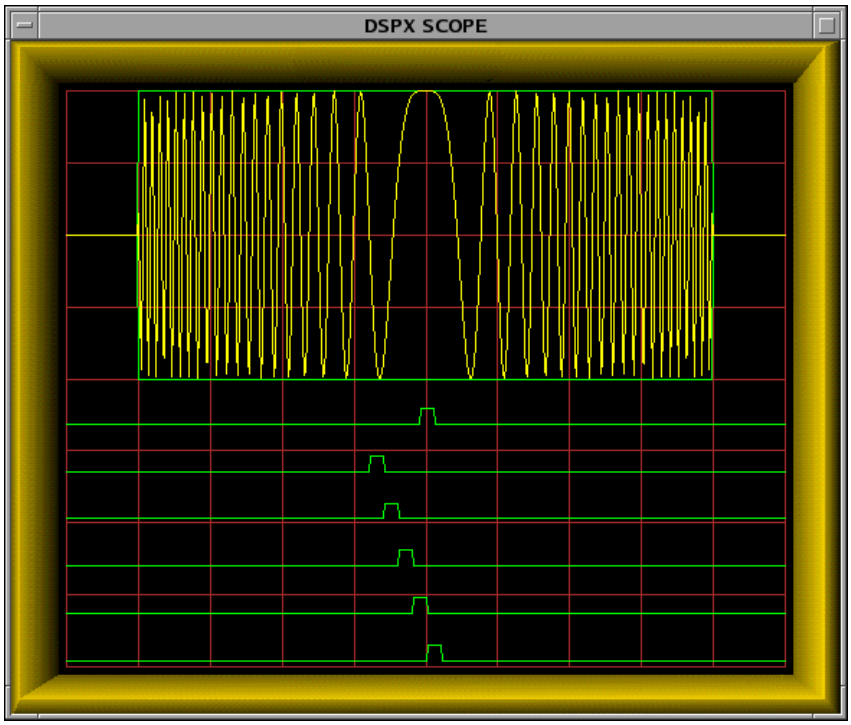

Fig.1 : *Real Part of Complex Linear-FM Tx Waveform Displayed in* ***Pb*** *Plot*
Pulselength: 40 sec, Bandwidth: 4MHz

7. The **Ps** plotting command now can be used to examine the ideal transmit spectrum and actual received spectrum of compressed pulses. An example is shown in Figure-2 for a 60MHz, 40µsec linear FM pulse having a bandwidth of 2MHz. The energy in the pulse is both sharply contained within and uniformly distributed over the 2MHz frequency interval centered on the IF carrier. This demonstrates the ability of synthesized transmit waveforms to remain cleanly within their allocated bounds, which is essential for weather radars operating in tight urban environments.

8. The **Pr** plotting command now can be used to examine the mainlobe response as well as range sidelobes of compressed pulses. Figure-3 shows a snapshot obtained by feeding a delayed RVP8/Tx pulse into the IFD and plotting over a 100µsec window. The plot effectively shows the magnitude of the waveform's zero-Doppler ambiguity function. Peak sidelobe levels of approximately –50dB are obtained using the simple linear FM pulse.

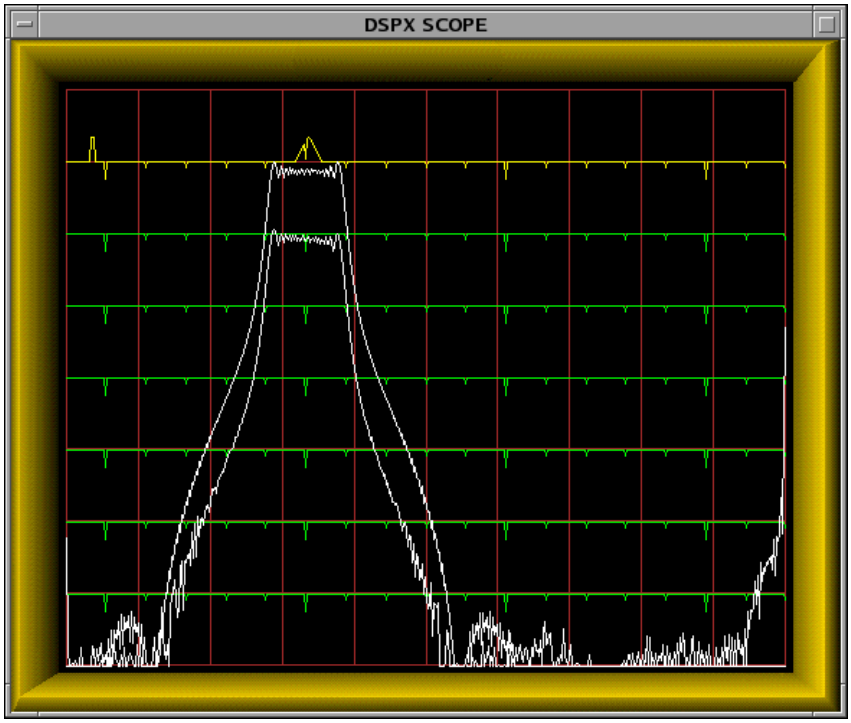

Fig.2 : *Ideal and Actual Linear-FM Spectrum Displayed in **Ps** Plot*
Pulselength: 40 sec, Bandwidth: 2MHz

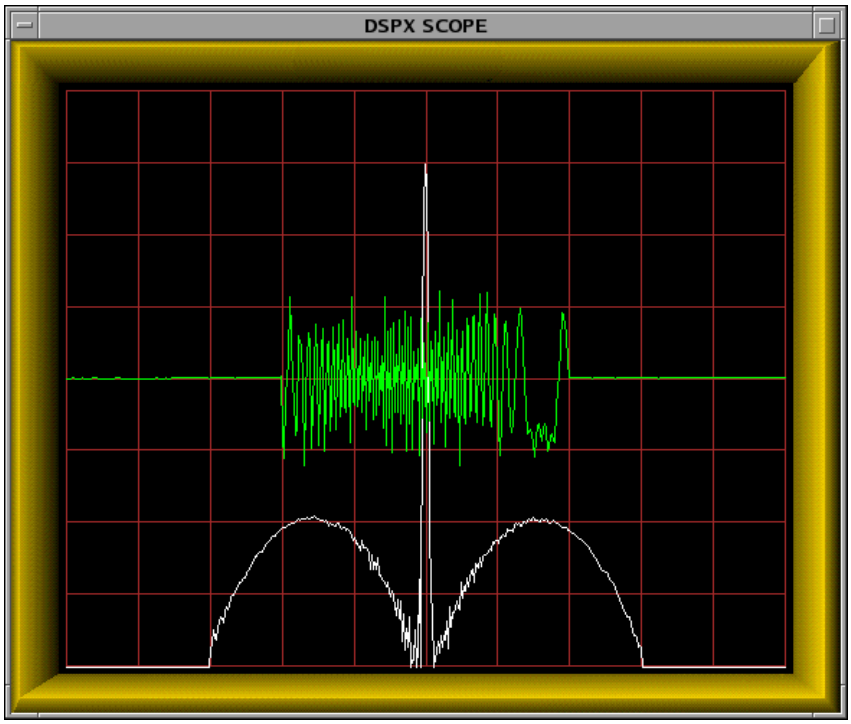

Fig.3 : *Actual IFD Sampling of Compressed RVP8/Tx Waveform Displayed in **Pr** Plot*
Pulselength: 40 sec, Bandwidth: 5MHz

Future Directions for Pulse Compression

1. Within the next 2-3 weeks we expect to release a general optimization package for designing compressed Tx waveforms, including a class of non-linear FM pulses that will have significantly improved range sidelobes compared with linear FM.
2. As we learn more about our customer's specific transmitters we will introduce adaptive pre-emphasis software to automatically compensate for frequency dependent as well as time dependent nonlinearities in the radar hardware.

Bug Repairs

1. A cross-checking bug was repaired that would cause the RVP8 to crash with a floating point exception whenever an undefined trigger bank was selected for running by custom code. Many thanks to Nathan Parker at MIT Lincoln Lab for helping zap this puzzling bug.
2. The "L" and "R" keys in the **Pb** (Plot Burst) menu now shift the RVP8/Tx waveform in addition to the normally shifted triggers. Previously the Tx waveform did not move.

RDA 8.05.1 Release Notes (8 Mar 2004)

These release notes cover changes made to the SIGMET Radar Data Acquisition platform, including primarily the RVP8 and RCP8 products. The last release was RDA–8.05 generated on 24 February 2004. If you are upgrading from an earlier release, please read those notes also.

New Features

1. There is now a *fBandWidthMHz* element in the extended GPARM structure *dspExParm* returned by the RVP8. It indicates the receive FIR bandwidth for traditional pulsed radar applications; but also returns the Tx bandwidth when the RVP8 is using a compressed pulse (Rx bandwidth is not well defined in the latter case).
2. Additional comments have been added to *rvp8main/open/txsubs.c* which describe the internal operation of the RVP8/Tx card interface.

Bug Repairs

1. External trigger inputs were not working properly in the RVP8. This has been broken since the 8.04.8 release.

RDA 8.05 Release Notes (24 Feb 2004)

These release notes cover changes made to the SIGMET Radar Data Acquisition platform, including primarily the RVP8 and RCP8 products. The last release was RDA–8.04.8 generated on 2 February 2004. If you are upgrading from an earlier release, please read those notes also.

New Features

1. The *vecFloatIQFromPackIQ()* library function has been speeded up by about a factor of six. This is good news for all software that recreates floating point timeseries data from their packed 16-bit encoded representations (e.g., timeseries import and playback).

2. Additional bits have been added to the DSP_HDR_FLAGS ray header flags to convey whether each ray was computed within an alternate PRFSECT "pie slice" sector, and if so, which one.

3. GPARM output word #49 now conveys the standard deviation of the noise measured in both the vertical and horizontal receive channels. Units are tenths of deciBels relative to the mean noise level.

4. Several changes have been made to the RVP8 setup menus in preparation for supporting pulse compression. You now choose in **Mz** whether an output channel of the RVP8/Tx card will supply an actual transmit waveform:

```
Chan A – 0:Unused, 1:FixedFreq, 2:TxWaveform : 2
  Output power level : 0.0 dBm
```

   and if so, then the specifications for that transmit waveform are chosen for each pulse width in the **Mt<n>** menu:

```
Tx Waveform – 0:CWPulse, 1:NonLinFM : 0
  Pulse compression tuning params: 0.0000, 0.0000, 0.0000
  Bandwidth of transmit pulse: 5.00 MHz
  Pulsewidth of transmit pulse : 1.00 usec
  Zero offset of transmit pulse: 0.00 usec
```

   Thus, all of the tuning parameters for the synthesized Tx waveform now reside in one place for each pulsewidth. The three optional parameters for pulse compression will be passed into whatever Tx waveform synthesis routine has been defined for the current major mode, hence, their meaning will vary.

⚠ **Warning: If you are presently using a gated RVP8/Tx output waveform to fire your transmitter, please change the Tx channel mode to "2" (TxWaveform), and copy the Pulsewidth/Bandwidth specs into each Mt<n> menu so that it matches the width of the trigger that used to gate the pulse. Your gated Tx pulse will then be the same as before.**

5. The ORDA panel CW-Test-Mode can now be requested at any time, and it's effect is to force RF_GATE to a continuous On (low) output level. Because RF_GATE can now be activated at any time, it has been removed from the special triggers that are inhibited by the RxProtect handshake, i.e., MOD_DISCHARGE and RF_PULSE_START are now the only protected triggers.

6. The RVP8 *RBACK* opcode can now return the Minimum and Maximum values of the optional I/O-62 A/D converter, sampled over at least one complete pulse period. Sixteen bit signed outputs represent the full range of the A/D converter. When the *RVP88D* backpanel is connected, the first Min/Max pair samples the *LOGVideo* input, the second pair samples the *CathodePulse* input, and the third and fourth pairs sample internal levels.

7. The RVP8 now supports a new enhanced version of the LFILT command that provides "Clutter Maps", i.e., the much greater flexibility of allowing filter choices to depend on antenna angle as well as range. This lets you specify a 2D or even 3D table of clutter filter selections that are dynamically selected during live data processing.

   The RVP8 maintains an internal array of up to 1024 different filter-versus-range tables, each of which is keyed to a particular solid angle AZ/EL sector. Each enhanced LFILT command fills in one of these slots with a filter selection table similar to that of the legacy command, except that the number of range bins is specified explicitly and eight bits are used for filter selection rather than three. Then, for each live ray being processed, the RVP8 applies clutter filters according to the filter slot whose solid sector includes the midpoint AZ/EL of the ray. If the antenna angle of the ray does not fall within any of the defined filter sectors, then the all-pass filter (#0) will be used at all ranges.

   The low and high angle limits in each filter slot are inclusive; hence, the pair (0x000, 0xFFFF) would span the full 360-degree circle with no gaps. Also, the filter array can be sparse (not all slots filled in), and have overlapping sectors (in which case the highest numbered slot that spans a ray's AZ/EL midpoint will be used). Choosing the highest numbered encompassing slot is a subtle but important detail that allows complex regions to be defined as a layered hierarchy of overlapping sectors. For example Slot-0 might define a default 360-degree filter-versus-range table, while Slot-1 defines special filtering within 0-90 degrees that is further modified by Slot-2 filters in a 40-50 degree span. A 45-degree ray would then be filtered according to Slot-2, a 60-degree ray would use Slot-1, and a 100-degree ray would use Slot-0.

   If the CLR bit is set in the opcode, then no additional arguments follow and the entire internal filter array (all slots) will be invalidated. The result is that no clutter filter will be applied to any of the processed data, regardless of Range, AZ, or EL. Moreover, loading a given slot with a table consisting of zero bins of filter data will invalidate just that one slot. This allows some data to be removed from the table without having to resort to a complete CLR operation.

   The legacy LFILT command is equivalent to calling the enhanced command first with the CLR bit set, followed by a second call that writes the legacy filter choices into slot #0 using AZ/EL limits that cover all of space. Thus, the legacy behavior is obtained as a special case of the enhanced mechanism. The new driver entry point *dspw_filt3D()* is available to create these new filter specifications in the RVP8.

8. The RVP8's RBACK command now uses an 8-bit (rather than 4-bit) field to choose the type of data that are being read back. This change was necessary because many new readback types have been added to the RVP8.

9. The defining parameters for the RVP8 clutter filters can now be accessed and modified via the DSP driver. Use the new *dspr_filterSpecs()* and *dspw_filterSpecs()* entry points to read and modify any of the clutter filter definitions.

10. The DSP driver is now able to read and modify some of the RVP8 burst pulse processing flags. The new BPOPTS opcode and *dspw_burstPulseOpts()* entry point may be used to control whether the signal processor is phase locking its (I,Q) data to the measured burst phase, and whether pulse-to-pulse amplitude correction is being applied. Likewise, the present states of those bits can now be read by *dspr_exParm()* .

Bug Repairs

1. Direct AZ/EL angle insertion from the RCP8/DCU was not working properly. The angles were being inserted with incorrect times, resulting in wildly thrashing values being read from the IRIS antenna driver. Now that this is fixed, you can get higher quality angles on the RCP8/DCU computer by configuring Setup/RCP to use "Native RCP8" as the "Antenna angle insertion source".

2. An RVP8 bug was repaired that could cause a spurious Floating Point Exception when *userTriggerOutputs()* was called by customized major mode code. This did not affect any of the "as delivered" RVP8 operating modes (PPP, FFT, RPH, and BATCH).