Riverside Research Institute

To: Biomed group
From: Jeff Ketterling
Date: Updated February 19, 2002
Subject: LabVIEW programs as of now

The LabVIEW programs written so far are briefly discussed, including the subprograms. These programs evolve quickly and may not reflect the most current version. Most programs have notes included in the diagram window of the VI. Theses notes indicate what the programs are doing. Some are more straightforward than others. The programs will be listed as I now have them divided into folders. Some calls to subVIs are made between the divisions I have set up. The programs listed under the Main programs headings have a user friendly interface and are meant for a specific purpose. The Subprograms sections list the support programs used by the Main programs.

---

1 ViewData

1.1 Main programs

1.1.1 *AlterHeader.vi*

Allows header info in EYE files to be altered on a field by field basis. Can change any number of selected EYE files in a single directory.

1.1.2 *BinFFT.vi*

This is a sub-program used by SpecFilter.vi. It is where most of the processing occurs.

1.1.3 *BytetoBMP.vi*

Convert an 8 bit byte stream file into a BMP image. Used for images created for prostate work but can be adapted for other uses.

1.1.4 *InterpSecImage.vi*

This program is now obsolete. This VI converts a sector scan into a polar type of display. The implementation is rather slow because the intensity values are interpolated to give a final image evenly spaced in x and y coordinates.

1.1.5 *SpecFilter.vi*

This program is a method of reducing speckle when creating B-mode images from EYE files. Take an FFT of each scan line of the eye file, separate the transmission band into several bins, convert back to time domain, add the bins, and then display the image.

1.1.6 *ViewEye.vi*

Multipurpose program to view an eye file and blow up certain features. A section of the original data can be boxed to give a 3-D rotational view of the data. The data can also be viewed as individual lines. If the image is a sector scan, a conversion is made to display it as a true sector scan image. Can also select to see a pop up window with all the parameters in the EYE file header.

1.1.7 *ViewPicts.vi*

Little program to view a sequence of images as a series of bitmaps. If the images are sector scans, they are displayed accordingly. The name of the file is overlaid on the image in the upper right hand corner. User chooses a directory and then the EYE files to view as Picts.

1.2 Subprograms

1.2.1 *ReadEye.llb*

This library contains VIs to extract info from the EYE files. Right now they can also read Ron's Cornell linear scan format to a limited extent. The following VI's extract info from the header as it is lumped in our EYE format: *A/DInfo.vi, CalibrationInfo.vi, HeaderInfo.vi, OtherInfo.vi, PatientInfo.vi, RFInfo.vi, ScanInfo.vi, ScanSystemInfo.vi, TGC&RF_ADInfo.vi, TherapyInfo.vi, Time/DateInfo.vi, TissueType.vi, TransducerInfo.vi*. There are also some more specialized VIs.

1. *MakeHeader.vi*

   Assemble all header info into a string format which can be appended to an EYE file.

2. *ReadHead.vi*

   Read all the info from the header and makes it available in clusters corresponding to the info sections just listed above. Also output the whole 1024 byte header.

3. *ReadEyeFile.vi*

   Reads info from the EYE file and makes it available as a 2-D array with the columns corresponding to scan lines. Also makes some other variable available.

4. *RonLinear.vi*

   Like *ReadEyeFile.vi* but for Cornell linear scan format.

5. *SetStringLength.vi*

   Sets the length of a string to equal a specified number of characters. Used to make the text fields the appropriate length in the header.

6. *ViewLine.vi*

   Makes a single line from an EYE file available for viewing.

1.2.2 *SectorDisplay.llb*

This library contains the VIs used to convert a rectangular array of data into a sector image.

1. *AxisRanges.vi*

   Pull out the max and min x and y values of the sector image once it has been converted from polar to rectangular. Used to scale the data and represent distance on the x and y axis.

2. *BytestoArray.vi*

   Convert 8 bit values to scaled values. Need to know high and low scaling parameters used to create byte file.

3. *CheckForSec.vi*

   Looks at file header to see if EYE file is a sector scan. If so outputs a boolean argument. One output for if it is a sector scan and another for if it is a scan greater than 180 degrees.

4. *ConvertToSec.vi*

   Converts an image to a sector format. Outputs the z values as an array and a bitmap.

5. *InterpAxis.vi*

   Used when forming a new sector image by interpolating along the x and y axis. Called by *InterpSecImage.vi* and therefore not used anymore.

6. *InterpHoles.vi*

   Take a freshly mapped sector image and fill in "holes" where no value was inserted. Fills in a hole with an interpolated value along the horizontal axis (cross range direction).

7. *LogScaleImage.vi*

   Takes the absolute value of the image data and compresses it logarithmically. Then scales the data so a value of 128 is still 128 after scaling.

8. *MakePict.vi*

   Convert a data array to a Pict format. Have choice of how to scale data. Can use logarithmic compression of linear scaling. All methods scale back to 256.

9. *OverlaySectorRoi.vi*

   Overlays a rectangular ROI onto the sector image. The reverse is not set up. That is, an ROI cannot be chosen on the sector image which then maps back to a rectangle on the original non-sector image.

10. *OverSample.vi*

    Use this for Hitachi data when converting to sector format. The RF data is first reduced down and then each scan line is duplicated twice before converting to a sector format. This fills out the image at the far ranges and leads to a better looking image.

11. *PictFromEye.vi*

    Make a Pict of the EYE file data and overlay the name of the EYE file on the image.

12. *PolarToRect.vi*

    Convert the original polar coordinates of the scan data into rectangular (X, Y) coordinates. The RF data can also be reduced to speed up image conversion. This is done by reducing the rows (RF pts) by some user selected value.

13. *RescaleAxis.vi*

    Re-scale the (X, Y) data that has been converted from polar coordinates so that the max X and Y values correspond to the number of rows and columns, respectively, in the final sector image. The (X, Y) data now spans from zero to the number of rows or columns and the data can be dropped into an empty matrix with the indices corresponding to the re-scaled (X, Y) values. Note (X, Y) gets rounded off when dropped into a display matrix.

14. *ROIEdges.vi*

    Determine the coordinates of the rectangular edges of the ROI. Used to overlay the corresponding ROI on the sector image.

15. *ScaleAxisBack.vi*

    Take indices in sector image and convert them back to (X, Y) coordinates or to indices from original data array.

16. *SectorOverlay.vi*

    This is where the sector image is inserted into an empty array. Uses pixel locations to replace empty elements with image values. Image has “holes” filled in with interpolation by InterpHoles.vi and is then smoothed with *SmoothImage.vi*.

17. *SmoothImage.vi*

    Rough smoothing of sector image. Can either take an average of the surrounding points or take the max value of the surrounding points.

2 FreqAnalysis

2.1 Main programs

2.1.1 *1-DSpecViewAndFit.vi*

Allows a spectrum to be taken for the selected ROI. The resulting spectrum can be used to make a CAL file or to find fit parameters for a specified frequency range. The CAL file and fit range can be read from the file header or input by the user. The resulting fit parameters can be saved to a *Spectra.txt* file with the format currently in use. The user can specify if the data should be forced to a certain length FFT. This may be important if a CAL is subtracted from the data. The ROI needs to be twice the length of the cal file.

2.1.2 *2-DSpecView.vi*

Will perform 2-D spectra analysis on the selected ROI. Can subtract a CAL from the data if desired.

2.1.3 *AOIParams.vi*

Allows sliding window data taken with *WholeImageROIFit.vi* to be analyzed with a cylindrical Area of Interest (AOI) or a free drawn AOI. The parameter values (midband fit, slope, and intercept) in the AOI are averaged to give a single fit value for the AOI. Can look at any of the fit images and the results come out for all three.

2.1.4 *BiopsyBatch.vi*

Process prostate EYE file data from B&K machines either on a case by case basis or as a batch. Biopsy AOI is taken based on a straight line in the sector co-ordinate system which is then converted back to the raw data form. AOI can either be between two parallel lines or a fixed number of points (normally 110). Batch processing is either done with all EYE files in a single directory or from a text list of files. Data is saved in the Spectra.txt format.

2.1.5 *Color.vi*

Open parameter image files and lookup table, then color code image.

2.1.6 *ConAgent.vi*

Process contrast agent data.

2.1.7 *ScatProps.vi*

Compute CQ^2 properties. Program is rough and not ready for any particular use.

2.1.8 *SectorBiopsy.vi*

Program is similar to BiopsyBatch.vi but is geared more towards data from the Hitachi. A rectangular ROI is define in the sector image and converted to the raw data format. The region of the ROI is then extracted from the data as a sub-set and sliding FFT analysis is performed on it, creating a parameter image of the ROI. An average value is then found for the ROI. The Hitachi cross calibration file is used for the analysis. Program can be run on a case by case basis or in batch mode. Results are saved in Spectra.txt.

2.1.9 *WholeImageROIFit.vi*

Analyze an EYE file with a sliding ROI and return the resulting slope, intercept, and mid-band fit data. A fit is also made to the Cal data and this can be subtracted from the final results. The program is presently geared to make parameter images that cover all the scan lines with a 64 pt window that increments by 8 RF points. The program can be easily modified, though, to allow the size or the ROI and how may points should be in the X and Y direction of the final image. Data can be saved in a variety of formats and is put in a folder named Param Images in the same directory as the EYE files.

2.2 Subprograms

2.2.1 *FreqAnalysis.llb*

1. *1DROISpectrum.vi*

   Take the 1-D power spectrum of the input signal. Data is passed through a Hamming window and then zero padded before taking the FFT. The output can be converted to dB or left as "raw" data. Phase information is available if ever needed.

2. *2DROISpectrum.vi*

   Take the 2-D FFT of a ROI. A CAL file can be subtracted from the end result.

3. *AOICoords.vi*

   Determine the indices for the AOI selected with *AOIParams.vi*. The program returns the row numbers, the start column, and number of points to take for the AOI.

4. *AttenCorr.vi*

   Small program to find attenuation correction as a function of range. Default α=0.5 db/cm MHz. Can be used for both parameter images or RF data. Output is an array to correct midband, $2\alpha x f_c$, and one to correct slope, $2\alpha f_c$.

5. *ColorEncode.vi*

   Overlay color for likelihood of cancer on parameter images. Window size can be chosen.

6. *ExtractName.vi*

   Extract the name and case number from the input file name. Also check to see how many digits are used to specify the case number.

7. *FindLine.vi*

   Find the indices of a line in sector scan ref frame and convert to indices in the raw data ref frame. Can also use three input points to convert a box in the sector format to its equivalent in the raw data reference frame.

8. *FindSectorBox.vi*

   Convert a box in sector format to one in the raw data system. Essentially converting (x, y) to (r, θ). Where (x, y) is a box in sector format and (r, θ) represents the raw data where r is RF point (range) and θ is scan line (angle).

9. *FitSpectrum.vi*

   Fit the selected frequency band using linear regression and return slope, intercept, mid-band fit, and error values.

10. *MakeEllipse.vi*

    Find the coordinates of an ellipse based on the input arguments taken from three cursors in a graph window. One cursor defines the center of the ellipse and the other two the major and minor axis.

11. *MovableROIFit.vi*

    Moves a ROI through a whole image and returns the fit parameters. The fit parameters for a CAL file are also returned if needed. The program can take both a single ROI or scan through a whole image. User selects ROI size and number of boxes in both the X and Y directions or the ROI size and start coordinates.

12. *PIByteFiles.vi*

    Create byte stream files from parameter image arrays. Geared towards B&K data right now. Byte files are scaled to the input ranges for mid-band, intercept, and slope.

13. *ProstSpecFormat.vi*

    Take data in *SpecCurveData.vi* and convert it to a text file where each variable is a fixed character width. Needed to prepare data for prostate data base.

14. *ROIAverage.vi*

    Find average attenuation values for a region of interest in *SectorBiospy.vi.* For Hitachi data, also find average cross calibration for ROI.

15. *ScaleParamData.vi*

    Scale data to a range of zero to some upper bound. Need to enter max and min range values for the scaling.

16. *ScatTheory.vi*

    Algorithm to compute CQ^2 or spectral parameters from CQ^2.

17. *SpecCurveData.vi*

    Assemble ROI and fit parameters into a string of characters which can be written to *Spectra.txt*

18. *SubSpecOneCase.vi*

    If looking at a single case, subtract the CAL parameters from the raw fit region of the ROI spectrum. This VI is needed because the output of the FFT and fit comes in a format to handle a sliding window FFT. Need to extract the data for the single ROI case.

19. *UnscaleParamData.vi*

    Scale 8 bit parameter images back to their true dB spectral parameter values.

20. *ZeroPad.vi*

    Pad zeros onto the data before taking an FFT. User selects a power of 2 or the nearest power of 2.

3 ProstateDataAcq

Many of the programs described here are similar to what is discussed in Sec. 4. The main differences are that the VIs in this section do not use any motion control hardware and data acquisition is done with a GaGe board rather than with an Acqiris board.

3.1 Main programs

3.1.1 *CopyToZip.vi*

A simple program that copies files from the Current Scans folder to Previous Scans folder and also copies the files to a zip disk. The program can also send an email message back to RRI if the computer used to take the data is networked.

3.1.2 *ProstateGetData.vi*

Similar to *AcqControl.vi* in Sec. 4.2.5. This is the master control for prostate data acquisition. Parameters are entered that get saved in the EYE file. Data is acquired and saved in RAM until a save to disk option is selected. Thumbnail images of all the scans can be viewed as data acquisition is in progress. In a research version of the software, more detailed options are available that permit RTV calibration and data acquisition from a single scan line across multiple frames.

3.2 Subprograms

3.2.1 *Gage.llb*

This library contains the VIs used to control the GaGe board. VIs are also used from the library of programs installed with GaGe software.

1. *GageSampInterval.vi*

   Convert sampling rate to GaGe input parameter.

2. *GageTrigLevel.vi*

   Convert trig level to GaGe input parameter.

3. *GageVolScale.vi*

   Convert voltage scale to GaGe input parameter.

4. *SetGagePrams.vi*

   Set up all GaGe input parameters.

5. *SetUpGage.vi*

   Oscilloscope like program that acquires data from the GaGe board and displays it. This VI is used in the research version of the software to find the scan line with the maximum voltage. *ProstateGetData.vi* does not use this VI in the clinical version of the software.

3.2.2 ProstateScanSystem.llb

1. *CountConfig.vi*

   Set up counter on the NI 6601 counter/timer card to take a frame of data or to acquire calibration data. Works with both a Hitachi system or a B&K system (3535 or Hawk).

2. *DigGageData.vi*

   This is where the GaGe data is acquired and transferred to memory. Parameters for each scan are also recorded along with the data. TGC data can be saved if using a B&K.

3. *FrameTrigger6601.vi*

   For frame acquisition, the counter is set up to generate an output pulse on the falling edge of a vector. The counter is armed by the frame signal. For the Hitachi it comes from the machine; for the B&K it comes from another counter. The source input of the counter is actually routed internally by the 6601 card to the gate input. Hardware arming of the trigger occurs on PFI line 30. Note that hardware arming is available on the dedicated counter/timer cards from NI, but not on multi-purpose A/D cards.

4. *FTPData.vi*

   This program allows data to be FTPed back to RRI. Also sends an email message that data is being sent. Not very elaborate error checking and has not been tested remotely yet.

5. *NumberMultFile.vi*

   Add a scan number to root file name.

6. *ProstateParams.vi*

   Set up a cluster of parameters for each scan. These are used to generate the header.

7. *ProstateThumbnails.vi*

   Create small thumbnails of each scan for quick viewing.

8. *QuitMode.vi*

   Option window to verify user really wants to exit software. If FTP is available, it also gives an FTP option.

9. *RemNoiseRec.vi*

   Used for B&K Hawk only. Subtracts out "noise" from the scan lines. First two lines of each from are coherent noise. First line for odd lines, second line for even lines.

10. *SaveGageData.vi*

    Save all acquired data to eye files.

11. *ScanParams.vi*

    All machine specific parameters are selected and kept track of in this vi.

12. *VectorTrigger6601.vi*

    The vector trigger is more straightforward than frame acquisition. Vectors go to the source of the counter, the frame signal to the gate. An output is generated by counting the number of vectors after the falling edge of each frame sync. This allows an output trigger to occur at the same vector sync for every frame. This program will work with normal NI timer also, although it may need to be modified slightly.

4 Scanning Control

4.1 Main programs

4.1.1 *mastercontrol.vi*

This VI acts as the control panel for the different functions of data acquisition. Several options are presented: the motors can be manually controlled, the motors can be moved to an (X, Y, Z) position, the motors can be sent back to the $(0, 0, 0)$ position, or a scan mode can be selected. The options for motor control open another control window to perform the desired function. To perform a scan, the scan mode is chosen, and then the button marked **GO** needs to be pushed.

4.2 Subprograms

Some of the subprograms can nearly act as stand alone programs. However, because some global arguments are used and parameters get passed between subprograms, they will not normally function on their own.

4.2.1 *DP110.llb*

This contains the controls for the Acqiris DP 110. The low level programs are in the LabVIEW directory in the *Instr.lib* folder.

1. *conMem.vi*

   Set up segments and points per segment of digitizer.

2. *conSamp.vi*

   Set up sampling rate and delay of digitizer.

3. *conTrig.vi*

   Set up trigger of digitizer.

4. *conVert.vi*

   Set up coupling and voltage scale of digitizer.

5. *RFWindow.vi*

   Determine where to place plot cursor to indicate what data will be captured.

6. *SampInterval.vi*

   Convert numerical sampling rate control into an actual value which can be input into the DP100.

7. *SetUpDigi.vi*

   Set up the Acqiris digitizer to take data. Used like an oscilloscope. Can select voltage scale, delay, and time base. If using a therapy burst, can also select those parameters. A red line on the digitizer traced shows what data will be captured during the scan.

8. *SingleSeq.vi*

   Take a single sequence of data and return as a true voltage.

4.2.2 Panametrics

The control of the 5900PR is largely taken from VIs provided by Panametrics.

1. *5800PRSet.llb*

   Can act as stand alone program. Checks status of 5900PR and then changes settings if needed.

2. *5800PR.llb*

   5800PRStatus.llb check current settings, *5900RW.llb* can read or write to 5900, and *GPIB Error Report.llb* reports an error if one came up.

4.2.3 *MM200.llb*

This library contains programs to perform basic movement of the motion system via the MM2000 motion control card. There is no need to discuss them because they are named to reflect their function.

4.2.4 *Ctrlpanl.llb*

This library also contains programs to perform basic movement of the motion system, and many of the features are duplicates of the *MM200.llb* library. However, the subVI named *MM2KCTRL.VI* is worth noting. It allows manual movement or incremental movement of the motors. This VI forms the core of the motion control accessed through *mastercontrol.vi*.

4.2.5 *ScanSystem.llb*

This library contains the meat and bones of the scan system. As mentioned earlier, the programs are able to operate as stand alone units to only some extent because global variables are used.

1. *33120PRF.vi*

   Set the frequency on a HP33120A function generator. Used to set up the prf signal on the SYNC output

2. *33123oSyncStat.vi*

   Allows the SYNC output of a HP33120A function generator to be turned off and on.

3. *AcqControl.vi*

   This vi provides the options for performing a scan. The look of the front panel changes depending on if it is a 1-D or 2-D scan. The appropriate parameters are available for the user to input. When the parameters are ready for use, a scan is commenced by pushing **Start Scan**. Pushing **Exit** ends the execution of the program.

   Global variables are extensively used. This helps reduce the clutter of wiring and passing arguments between subprograms. The program starts by setting up the PRF source which is an external function generator. The user can select an option to set up the Panametrics P/R (**Set Panametrics**). The digitizer (**Set Digitizer**) needs to be set up before taking data. More detailed instructions on the use of the software available.

4. *CaseParams.vi*

   Assembles a cluster of arguments for each scan performed. Used so to keep track of parameters when saving data.

5. *ContMotion.vi*

   This vi assembles a program to perform a continuous motion scan with triggers at specified intervals. Used by *SendProgram.vi*.

6. *DigData.vi*

   This VI controls the actual scan and data acquisition. First, the counters are stopped if using an internal PRF or if the therapy transducer is being used. Then the Acqiris is set up to take a batch of sequence data. Counters are then started or the motors are moved to provide the PRF triggers for the Acqiris. Motion can either be a continuous sweep or in incremental steps, depending on the type of scan. After the data is acquired, the counters are again stopped, the therapy voltage, if used, is set to zero and the motors are returned to their start position. The data is then assembled into a 2-D matrix with each column a single scan line.

7. *EXTPRFCount.vi*

   This VI sets up the counters to divide down the PRF and to send out a delayed pulse to the therapy transducer. Counter 0 provides the divide down and counter 1 provides the delay.

8. *IncSteps.vi*

   This vi assembles a program to perform an incremental motion triggers at a specified increment. Used by *SendProgram.vi*.

9. *InputHeader.vi*

   Panel that opens prior to saving EYE files. Contains the fields in the header and permits changes to be made to some of them.

10. *IntPRFCount.vi*

    Sets up counters on the A/D board to generate a PRF. Counter 0 is the PRF and counter 1 divides down the signal. The divided signal can trigger the therapy. This VI is no longer used. Instead *EXTPRFCount.vi* is used because the PRF is always from an external source. If setting up a system with no motion control, this VI could be useful.

11. *MM2KManCTRL.VI.vi*

    Allows manual control of the motors. By depressing the controls, motor can be moved in any direction. The position is updated as the motor moves.

12. *MotionGlobals.vi*

    This VI contains all the global variables. It is used by many of the programs in this section.

13. *MoveToXYZ.vi*

    Moves motors to the specified (X, Y, Z) position.

14. *NumberFileName.vi*

    Used for labeling 2-D scans. Allows a number to be added to the case name.

15. *SaveData.vi*

    Save the data from a scan in eye files. Uses arguments assembled in *CaseParams.vi* to set up values in the header. User can alter any header parameter and then save the data. If a 2-D scan was used, data files are automatically numbered.

16. *SendProgram.vi*

    Sends a motion program to the MM2000. The program is assembled based on input parameters sent to the motion board and is then compiled by the MM2000 card. Two types of programs can be sent, one for a continuous motion with trigger pulses at specified intervals, or one for incremental motion with a trigger pulse at each incremental move.

17. *SetPid&FindPos.vi*

    Set the parameters for the motors such as velocity. Also find the current position and write it to the global variables.

18. *ShowThumbs.vi*

    Displays a 1-D array of Picts as a 2-D array. User specifies number of columns.

19. *Therapy&Scan.vi*

    Allows therapy to be fired for longer durations than just 2000 cycles of HP burst mode. Calls *Therapy.vi* between taking A-Scans. User specifies A-scans before and after therapy.

20. *Therapy.vi*

    Allows therapy transducer to be fired for some specified number of seconds. Makes a call to *TherapyBurst.vi* to set up the function generator in a gated mode. Counters 0 and 1 are used to gate the therapy function generator for the desired length of time.

21. *TherapyBurst.vi* Sends GPIB commands to adjust the settings of the function generator used to control the therapy transducer.

22. *Thunmbnails.vi*

    Make thumbnail images of all the scans. The thumbnails can be viewed as an array by selecting the appropriate option where available.

5 TransducerCal

5.1 Main programs

5.1.1 *BeamCal.vi*

Takes EYE files from pulse measurements with a hydrophone and computes beamwidth and other parameters. Some of the parameters need to be manually entered.

5.1.2 *FindBandwidth.vi*

Open a cal fine (two column format) and determine bandwidth and center frequency.

5.1.3 *PulseCal.vi*

Allows calibration of pulses from a transducer using the FDA definitions of various parameters. Need to enter a few parameters from when the data was taken. Then can take an EYE file and select a scan line that contains the pulse information. By selecting the region of the line that has the pulse, all the relevant parameters are generated.

5.1.4 *TestTrans.vi*

Will perform a 1-D or 2-D scan and allow for characterizing the beam pattern of a transducer. Uses many of the same VIs that were found in Sec. 4. Data will be presented in both a 3-D form and also 2-D. Still need to work out details.

5.2 Subprograms

5.2.1 *CalTransducer.llb*

1. *ChoosePram.vi*

   Finds selected parameter for scan data. Can find pos, neg, peak-peak, and rms voltages or single pulse parameters.

2. *Derate&MI.vi*

   Find derated parameter values for ultrasound pulse. Takes inputs from *IntParams.vi*.

3. *FindHydSens.vi*

   Find a sensitivity of a hydrophone for a specified frequency. Can choose a calibration from one of our hydrophones.

4. *IntParams.vi*

   Finds the intensity parameters such as I_{ta} and I_{pa} for an ultrasound pulse.

5. *MoveandWait.vi*

   Move motor and wait for specified time duration.

6. *MultiPlane.vi*

   Used when taking 3-D data of a transducer. Allows for multiple scan plane data sets by incrementing up (vertical) in the z-direction after each scan and saving the data to a spreadsheet file.

7. *PulsePrams.vi*

   Finds pulse from an RF signal and computes the pulse parameters using *FindHydSens.vi*, *IntParams.vi*, and *Derate&MI.vi*.

8. *RapidScan.vi*

   Takes data in a continuous sweep rather then in incremental steps. Speeds up the data collection.

9. *ScanTrans.vi*

   Performs scans for *TestTrans.vi*. Moves motors along desired axis in an incremental fashion, stopping briefly at each point to take data. Idea is to use a hydrophone to sample a CW wave from a source and then to extract parameters from the wave to characterize the source. May want to allow a burst from source. Can use a pre-trigger or delay on Acqiris to take the data properly. Will need ability to display trace from Acqiris.

6 3DdataPrep

6.1 Main programs

6.1.1 *3DVisPrep.vi*

This program allows data stored in scan planes (either spreadsheet format or RGB BMP) to be converted to several types of output files. The output options include conversion to 8 bit byte files for each plane or a single byte file with all the data. The spreadsheet data can also be converted to 8 bit BMP files.

7 Annular Array

7.1 Main programs

An assortment of programs to compute the SIR from an annular array.