SPIRE

SUBJECT: SPIRE QLA User Requirements

PREPARED BY: Tanya Lim

DOCUMENT No: **SPIRE-RAL-PRJ-000999**

ISSUE: **Draft** 0.3 **Date:** **24 January 2002**

APPROVED BY: **Date:**

ICC Software Manager
(Steve Guest)

ICC Manager
(Ken King)

Instrument Scientist
(Bruce Swinyard)

AIV Manager
(Dave Smith)

Distribution

Steve Guest	RAL
Ken King	RAL
Bruce Swinyard	RAL
Dave Smith	RAL
Sunil Sidher	RAL

Change Record

ISSUE	DATE
Draft 0.2	07 November 2001
Draft 0.3	24 January 2002

TABLE OF CONTENTS

FIGURES

TABLES

Glossary

AIV	Assembly Integration Verification
BDA	Bolometer Detector array
BSM	Beam Steering Mechanism
CDMS	Command and Data Management System
CQM	Cryogenic Qualification Model
EGSE	Electrical Ground Support Equipment
FS	Flight Spare
FTS	Fourier Transform Spectrometer
GUI	Graphical User Interface
HCSS	Herschel Common Science System
IDL	Interactive Data Language
ILT	Instrument Level Tests
PFM	Proto-Flight Model
QLA	Quick Look Analysis
RTA	Real Time Assessment

1. SCOPE

This document describes the top level user requirements for SPIRE QLA. The concept of the QLA system is described in the introduction, with the assumptions made in section 4. The following sections then describe general requirements followed by functional requirements. Many functional requirements are specific functionalities of QLA required for specific tests. Until the test plan is mature it is unlikely that this section will be complete although the basic requirements should be covered.

2. DOCUMENTS

2.1 Applicable Documents

AD1	EGSE Users Requirements Document	FIRST-SPI-DOC-000102
AD2	SPIRE CQM Instrument Level Test Plan	SPIRE-RAL-DOC-001049

2.2 Reference Documents

RD1	Packet Structure ICD	SCI-PT-IF-07527
RD2	SPIRE Instrument CQM Test Requirements	SPIRE-RAL-NOT-000389
RD3	SPIRE Instrument AIV Plan	SPIRE-RAL-DOC-000410
RD4	Minutes of the SPIRE QLA Workshop	SPIRE-RAL-MOM-001012

3. INTRODUCTION

SPIRE Quick Look Analysis (QLA) will be a tool mainly used to perform a near real time assessment of the scientific quality of the SPIRE instrument output using the telemetry data. To do this, QLA will generate displays, which are triggered either automatically or through user interaction.

The main use of QLA will occur during the instrument test phases where it is used for near real time assessment of instrument output. Typical uses include displaying detector output as an image e.g. for peak-up, displaying data changing with time e.g. a single detector output during chopping , plotting values against each other, e.g. mirror position and detector output, and monitoring equipment e.g. test equipment and mechanism positions.

The tests will be run via test scripts with several items of test equipment outputting data required by QLA (see figure 3.1. The test scientist may wish to inspect a number of different parameters, some of which may be difficult to define in advance of the test, therefore it is desirable if, for data display purposes, QLA is a modular highly interactive system allowing the user to select (and de-select) displays and algorithm routines to be applied to the incoming data. In this document the application of an individual module to a set of data is referred to as a process, therefore it may be possible to have separate processes executing the same code at the same time e.g. a process may extract and plot the data from an individual detector and another process may be simultaneously doing the same for a different detector. The ability to have multiple processes accessing the input stream (or parallel steams) is the main baseline requirement for QLA. As the instrument testing matures, the concept of which displays are needed will also develop and the use of QLA will become less interactive and more script driven. Information received within the test data can then be used to trigger (and stop) QLA processes defined by a QLA script. QLA will have some functionality to allow near real time scientific data reduction e.g. the conversion of detector output to volts. However, the data products produced will not be of the same quality or reduced to the same extent, as those produced off-line with an interactive analysis system.

Throughout testing, instrument commissioning and operations, the QLA system must also be capable of receiving stored telemetry in a playback mode. This will allow QLA to be used as a diagnostic system for instrument problems by allowing the user to inspect the data at a rate slower than real time, therefore giving access to instrument behaviour changes which happen on short timescales. The user will also have tools within the QLA/RTA system for the visualisation of the raw telemetry and these are not usually present in interactive analysis systems.

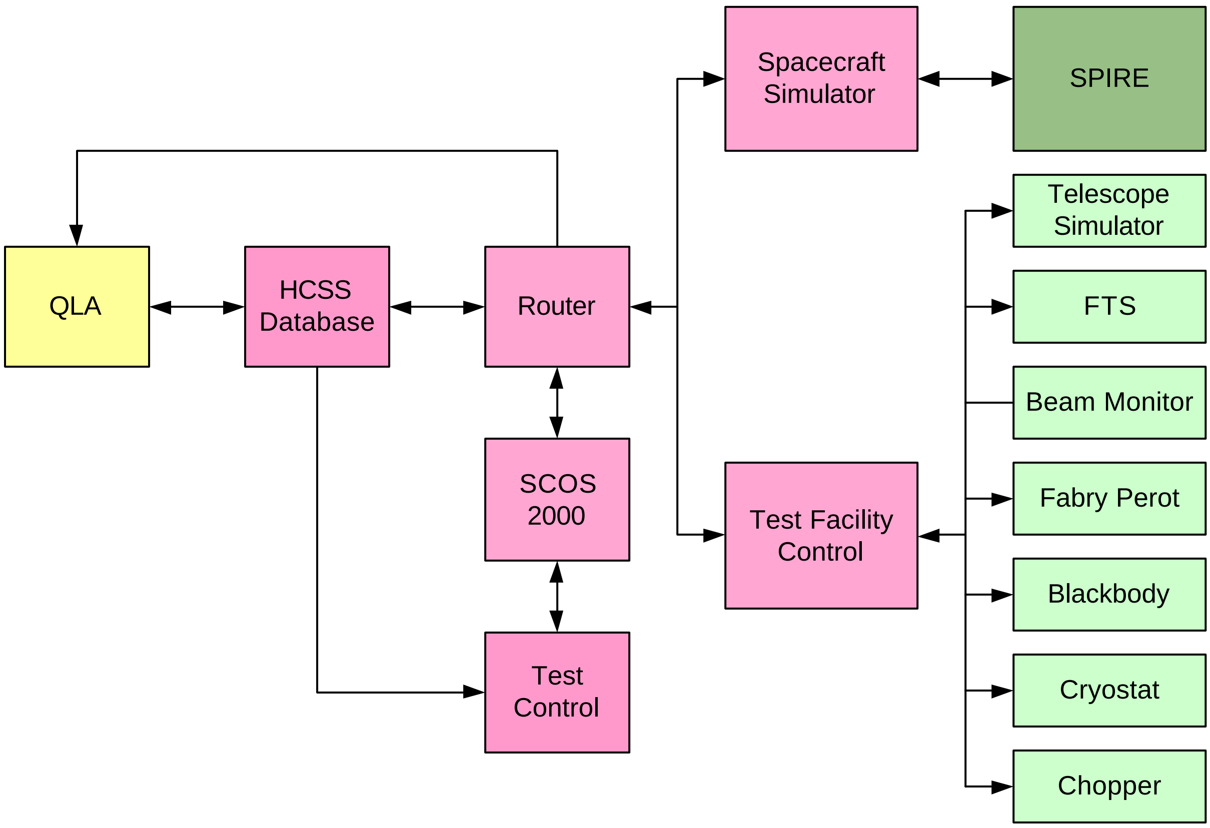

Figure 3.1: The SPIRE test setup

The use of QLA is somewhat dependent on the stage of testing and what is expected to be understood from a particular instrument model. The main stages are outlined below.

CQM testing – The Cryogenic Qualification Model (CQM) will have three bolometer detector arrays (BDAs), one in the photometer and two in the spectrometer. It will be the first model during ILT where the scientific output of the instrument can be assessed. Most of the testing will consist of functional tests and initial performance tests. The QLA system must be able to re-construct the test equipment status, instrument status (e.g. BSM position) and detector readouts on a common timeline. Basic displays needed include, selected telemetry parameter with time, and images of the three arrays.

PFM testing – The Proto-Flight Model (PFM) test programme will be designed to verify that the instrument is meeting all its performance requirements, and will be capable of carrying out the defined operating modes, hence all functionality of QLA described in this document must be present. The instrument calibration will be defined during PFM testing therefore functionality must be present to either analyse the calibration data within QLA or export the data in a suitable format for an external calibration or interactive analysis tool.

FS testing – The Flight Spare (FS) testing will consist of the full set of performance and operating mode tests. A set of calibration parameters will also be derived for the FS model.

IST – For integrated system testing QLA will be used to monitor a pre-defined set of functional and performance tests and it likely that most use of QLA during this phase will be script driven. The tests will verify that the instrument integration with the spacecraft has been successful and that the performance is satisfactory. This will be the first use QLA with the actual spacecraft command and data management (CDMS) system.

Spacecraft Commissioning – During the spacecraft commissioning the same set of functional and performance tests will be carried out (in real time?) and QLA will be used to monitor the output.

PV and Operations – after the commissioning phase QLA will no longer be needed for nominal operations, however the system will remain operational to allow any instrument problems to be investigated. These investigations may take the form of playback of spacecraft data or may be the verification of a new operating mode on the flight spare.

This note aims to define a core set of requirements for all these phases. The main input to this was the result of the discussion held at the SPIRE QLA workshop (RD4), and the draft set of requirements from the EGSE user requirements document (AD1).

4. ASSUMPTIONS

The QLA system will have to correlate in time the data coming from the various subsystems and test equipment, e.g. facility chopper and detector readouts. It is assumed that there is sufficient information coming from these systems to enable this to take place.

QLA will need a mechanism to determine when to start and stop processes automatically. It is assumed that it is possible to command the DPU to generate event packets or set telemetry parameter values as required which will then be used by QLA as a trigger.

QLA will be able to store reduced data in the local HCSS (QLA-UR-ST-02). This data must be accessible by any authorised user, which may include users external to RAL. Authorised users should be able to make queries on the stored data e.g. via a database interface and retrieve data in the form of files (ascii and fits being the most likely formats). It is assumed that this functionality is not provided by QLA.

QLA must be able to update tables within the HCSS while retaining the previous version within the HCSS (QLA-UR-ST-07). It is assumed that this functionality is available in the HCSS.

5. GENERAL REQUIREMENTS

A set of general requirements applying to all types of data handling and display are given in this section.

5.1 Interfaces

SPIRE QLA will be primarily a system designed to facilitate the understanding of data from the ground testing of the instrument and much of the functionality will be based on near real time data assessment. Therefore one of the main drivers for QLA during testing will be the need for speed. This will be ensured by an interface to the router which bypasses the ingestion process of the HCSS. There will also need to be an interface to the HCSS to allow QLA to be used in playback mode and to allow data to be selected either in the form of packets or data frames. The HCSS interface will also allow the retrieval of calibration files either delivered by sub-systems or generated from previous tests.
During the commissioning phase both RT and playback will be required and QLA is required to receive telemetry from the operational HCSS. During normal operations, QLA will mainly be used for playback and it is unlikely that it will be used in RT.

QLA-UR-IF-01	QLA shall provide an input interface to the real-time telemetry data stream.
QLA-UR-IF-02	An interface to the router must exist which will allow QLA to retrieve multiple packet streams from the router.
QLA-UR-IF-03	An interface must exist to the HCSS which will allow QLA to receive either RT or playback data from the HCSS.
QLA-UR-IF-04	QLA shall accept all telemetry in the form of source packets or data frames *The different types of packets will be defined in the Packet Structure ICD (RD1)*
QLA-UR-IF-05	The interface shall allow multiple processes to make requests at the same time. *QLA/IA may run as several processes operating in parallel*.
QLA-UR-IF-06	It shall be possible for QLA to be simultaneously receiving packet streams and data frame streams. *It is conceivable that different QLA processes running simultaneously may be more suited to one data type than the other*.
QLA-UR-IF-07	The interface shall be able to handle telemetry source packets at the maximum rate they are generated by the instrument and additional test equipment (50 packets/s (TBC) or 400 kbps (TBC)). *The instrument may generate data at a rate exceeding the average on-board data rate (100 kbps, TBC) in bursts, during testing. The test equipment may also generate data at a higher rate. The interface should be able to transfer data at this maximum rate to prevent the QLA running behind the real-time telemetry*.
QLA-UR-IF-08	It shall be possible to select types of telemetry source packet or data frame. The packet selection shall be by APID and/or packet type and subtype. The data frame selection is TBD. *Selection of packets will reduce the performance overhead of dealing with unnecessary packets*.
QLA-UR-IF-09	The interface shall allow a request for the last telemetry packet, of a given type, to become available.
QLA-UR-IF-10	The interface shall allow a request for the next telemetry packet of a given type to become available. *In this way the QLA software may continue to use the latest available telemetry data*.
QLA-UR-IF-11	The interface shall allow a request for telemetry packets over a given time range relative to the current time. *This will allow QLA to `look back' over a time range to generate products*.
QLA-UR-IF-12	The interface shall allow a request for telemetry packets or data frames over a given absolute time range. *This will allow QLA to generate products related to a given time period*
QLA-UR-IF-13	The interface shall allow playback data to requested based on time of acquisition of the data.
QLA-UR-IF-14	It shall be possible to playback at a rate of at least 5 times the normal data acquisition rate.
QLA-UR-IF-15	The interface shall be able to request any data resident in the HCSS database based on a query. *QLA/IA will need to access other data (e.g. calibration files, parameter data, spacecraft data*

	etc. in many ways, based on the object model defined for the HCSS database.
QLA-UR-IF-16	The interface shall convert telemetry source packets or data frames into data structures compatible with the analysis software environment used by QLA/IA. *For example, the interface could create an IDL data structure from each packet that can then be accessed from the QLA/IA software without knowledge of the form in which the data arrived at the interface.* *No conversion of data values into engineering units will be provided – this is a function of the QLA/IA software.*
QLA-UR-IF-17	QLA shall be able to interface with the external data sources (router or HCSS) either on the local machine or over a network (via TCP/IP)

5.2 Data Extraction

QLA-UR-DE-01	QLA shall be able to convert datation information in the science data streams into a standard acquisition time (UTC, TBC) for each data sample.
QLA-UR-DE-02	QLA shall be able to extract any parameter, or set of parameters from a science data stream, along with its/their associated acquisition time(s). *It is expected that a sample of each set of parameters can be associated with a single acquisition time (TBC).*
QLA-UR-DE-03	QLA shall be able to extract multiple samples of a parameter, or set of parameters, along with their acquisition times, from a science data stream
QLA-UR-DE-04	It shall be possible to 'interpolate' the data samples extracted from one science data stream to the values that would be expected at the acquisition times extracted from another science data stream. *This will allow two sets of science data to be correlated in time, before display.*

5.3 Display

QLA-UR-DI-01	It shall be possible to plot data to a display in various forms. *At least 2D plot, 3D plot, histogram, contour, scrolling time series shall be provided* *Plotted points may be as points with or without joining lines*
QLA-UR-DI-02	It shall be possible to display multiple plots on a single display *At least 8 plots should be displayable on a single display*
QLA-UR-DI-03	It shall be possible from the GUI to manipulate the display *For example it should be possible to zoom in/out on a section of the display (and have the data redrawn at the new scale), change the scale of the display, change the text of titles, add comments with arrows pointing to part of the display etc)*
QLA-UR-DI-04	It shall be possible to make a hardcopy, in colour, of any display
QLA-UR-DI-05	It shall be possible to display a key indicating the source of each set of data on a plot

5.4 Data Plotting

QLA-UR-PL-01	It shall be possible to plot extracted parameters against acquisition time *It is expected that at least 10 parameters may be overlayed on each plot (using different point/line types and/or colours)*
QLA-UR-PL-02	It shall be possible to plot samples of one extracted parameter against a second extracted parameter. *It is expected that at least 10 parameters may be displayed in this way on a single plot (using different line types/colours*
QLA-UR-PL-03	It shall be possible to display error bars (in any axis) on any plot

5.5 Storage

All telemetry from the tests is stored in the HCSS database, therefore there is no requirement for QLA to store telemetry data. Data processed by QLA may be either stored or discarded depending on the nature of the test e,g. the ground testing will produce data from interactive operations such as peaking up, which does not have a requirement to be stored for future use. Therefore the QLA design must allow for not all data to be stored.

Data stored in the HCSS database must be accessible either outside QLA or by applications within QLA. It would be desirable if the facility to retrieve stored reduced data from the database is as user friendly as possible. There should also be facilities for temporary data storage on a disk, to house temporary files created while QLA is running and to store data reduced by QLA that needs further analysis e.g. within IDL.

QLA must have functionality to store data automatically. This will generally consist of adding new tables (or objects) to the database containing the processed results of an individual test. It would be desirable if the database update facility could also include expanding existing data within the database e.g. the database may contain a table of the daily detector temperature readings and other related information where today's value could be added. Storage activity may be triggered either by an automated process or by user interaction.

QLA-UR-ST-01	A disk storage area should be available for QLA to place output files. *This is likely to be on the QLA machine although it would not affect functionality if this is a networked disk*.
QLA-UR-ST-02	QLA must be able to store reduced scientific data in the HCSS database. *The specific cases are described in the QLA usecases.*
QLA-UR-ST-03	QLA must be able to store reduced scientific data to a disk storage area.
QLA-UR-ST-04	The HCSS I/F shall allow writing of data to the HCSS database, when authorised. The interface will not check the authorisation. It will provide the relevant information to HCSS to authorise the transaction.
QLA-UR-ST-05	Automatic storage of data must be possible. *This could be by writing a file or by making an input to the database depending on the process. Note there will be operations done by QLA which will produce data that is not required to be stored.*
QLA-UR-ST-06	QLA must be able to add new data to a pre-existing table within the HCSS database. *For example today's temperature reading.*
QLA-UR-ST-07	QLA must be able to update a calibration table in the HCSS. *This would involve replacing a pre-existing calibration table as the 'current' version while keeping the 'previous' versions for traceability reasons.*
QLA-UR-ST-08	No query on the local HCSS database should take more than 30 seconds (TBC). *This sets a requirement on the design of the data storage within the database but is no additional details are given here as this is a matter for implementation.*
QLA-UR-ST-09	Data retrieved from the HCSS for further processing should be stored in a file format supported by IDL.

6. FUNCTIONAL REQUIREMENTS

The following requirements are general requirements for functionalities that should be available all times when QLA is running.

6.1 General Functionality

QLA-UR-GF-01	It shall be possible to run QLA either interactively or via a script.
QLA-UR-GF-02	It shall be possible to run multiple processes simultaneously

QLA-UR-GF-03	It shall be possible to start and stop QLA processes from a command line without inhibiting the running of other processes.
QLA-UR-GF-04	It shall be possible to start and stop QLA processes automatically. *This could be a call from a script or from another process. An example of this would be a plot appearing or being updated at certain defined points within a test.*
QLA-UR-GF-05	It shall be possible to select parameters for extraction. *As the contents of the packets are known, this could be implemented as a selectable list from a GUI (or GUIs say for each packet type TBD) or from the command line. Selected parameters are then used by other processes e.g. plotted as a time series.*
QLA-UR-GF-06	It shall be possible to select a single, a user defined set or a pre-defined set of detectors. *For certain tests a detector or set of detectors need to be selected. The minimum requirement is that this is possible from the command line although it would be desirable if the user had the option of selecting from a GUI display showing detector number and array position. We should be able to select sets of detectors based on Bias electronics, ADC converter used or any other grouping (TBD)*
QLA-UR-GF-07	It shall be possible to select the number of plots for parameter time series plot window. *The idea would be to allow for any number of parameters to be selected and allow the window to be split up in to the number of plots required, typically this may be up to about 4.*
QLA-UR-GF-08	It shall be possible to plot a selected parameter as a time series *There are two options: a window of fixed time length with the data scrolling off the beginning, or with a fixed window containing all the data, which slowly becomes compressed as more data is added. The 'time' used will probably be the acquisition time. It is unlikely that any graphical manipulation of the plot widow is required.*
QLA-UR-GF-09	It shall be possible to display selected parameters as a scrolling lists. *This could be done either as a separate window defined by the command line or selectable using a GUI which was called from the command line.*
QLA-UR-GF-10	It shall be possible to plot two parameters. *User selects two parameters, either through command line or possibly GUI lists to plot against each other e.g. detector output vs BSM position. The implementation of this requires the ability to correlate data acquisition times as information arrives at different rates.*
QLA-UR-GF-11	It shall be possible to plot detector output against detector number. *For some purposes we will want to monitor detector output of all detectors in an array. One way of doing this is with a two dimensional plot with detector number on the x-axis and parameter on the y-axis. Here a histogram plot is probably most useful.*
QLA-UR-GF-12	It shall be possible to plot detector output as an image corresponding to physical or optical layout. *This may be applicable to both raw and processed data. Note the physical layout is obtained simply by the detector geometry, the optical layout is the layout of the detector beam FWHM projected on to the sky.*
QLA-UR-GF-13	Retrieve and plot stored parameters v time. *We may wish to retrieve long-term output from previous tests to see how the current results fit in to the trend. This will also be an IA requirement.*

6.2 Demodulation

Near real time de-modulation is one of the core requirements of QLA. The signal on the detectors may be demodulated in one of three ways. It may be modulated by the external chopper, by the beam steering mechanism or by the internal calibrators. The form of modulation may vary e.g step functions, triangular and sawtooth patterns.

QLA-UR-SR-01	It shall be possible for QLA to demodulate a signal modulated by the external chopper *Several algorithms (TBD) may be used and could be selectable, the most simple of which would be to subtract the off signal from the on signal.*
QLA-UR-SR-02	It shall be possible for QLA to demodulate signals from either of the internal calibrators *Several algorithms (TBD) may be used and could be selectable, the most simple of which would be to subtract the off signal from the on signal.*

QLA-UR-SR-03	It shall be possible for QLA to demodulate a signal modulated by the beam steering mechanism. *Several algorithms (TBD) may be used and could be selectable, the most simple of which would be to subtract the off signal from the on signal.*
QLA-UR-SR-04	Subtract off from on signal per cycle. In addition to the demodulation algorithm used we need a' per cycle' method of demodulating for a running display of the on-off signal. *This would require combing the detector readouts at each chop position then subtracting (What about the calibrators?).*
QLA-UR-SR-05	Display de-modulated signal. *There shall be three possible displays: a time-series plot with one point per chop cycle, building up in real time. This time series plot could be a single plot or three plots stacked vertically, one showing chop position, one detector signal and one the de-modulated signal; a display of the de-modulated signal as a list of signal values which is updated every chop cycle.*
QLA-UR-SR-06	Display signal vs chop position, *This would address triangular and saw-tooth chopping. Should be possible both as a time series and an integrated plot over one cycle. The integrated plot could be updated in a pre-defined time period or a pre-defined number of chops or once per test (TBD). We may want to select individual detectors or make an average across an entire array.*
QLA-UR-SR-07	Store time series of signal and chopper position/BSM position/calibrator output in the database along with relevant HSK parameters for all detectors. The user should be able to select additional storage to a file. *The time series is stored to allow easy access to processed data (volts) for testing alternative demodulation algorithms.*
QLA-UR-SR-08	Produce an overall response for each detector for the each chop frequency and store in the database. *Many chop frequencies will be tested. The signal will need to be de-modulated, then, if done on a per cycle basis, averaged. I guess a selection of favourite averaging algorithms could be available here. We might also wish to store other related information relating to the expected optical load.*
QLA-UR-SR-09	Retrieve previous/selected frequency and response values, either single detector or overall value. *A series of tests will be conducted where the detector responses are obtained at different chop frequencies (and patterns?) Once the series is complete we will want to then construct the frequency/response plot.*
QLA-UR-SR-10	Display the selected detector(s) response vs frequency. *A single plot window should be sufficient, one or more detectors may be displayed.*
QLA-UR-SR-11	Analyse frequency/response. *The data needs to be analysed to determine the characteristic parameter: f_{3db} plus possibly 'shape of curve'. This will involve plotting response (y-axis) against frequency (x-axis), fitting the curve, then finding the f_{3db} value by determining half the intercept with the y-axis.*

6.3 Alignment Tests

The following requirements relate to the functionality required for specifically for optical alignment tests and peaking up the signal.

QLA-UR-OP-01	Create image of de-modulated detector signal on selected detector array(s) *For optical alignment we move the beam and change the focus therefore we need a visual representation of the whole array. For the photometer, up to all three arrays could be displayed simultaneously.*
QLA-UR-OP-02	Create 3-D projection of the signal on an array. *This could either be in the form of a contour map or surface.*
QLA-UR-OP-03	Add the beam location to the detector array image. *Put the location of the centre of the input beam from the telescope on an image of the detector array. To know the expected beam location we will need the output from the BSM and presumably the telescope simulator. It is not clear exactly how QLA will do this.*
QLA-UR-OP-04	Select a slice. *To visually aid peak up it would be useful for the user to select at least two1-D*

	slices across either the array image, or the 3-D projection. This should be done via GUI.
QLA-UR-OP-05	Plot slice across array. *The detector signal is then interpolated along these slices and plotted as signal vs position graphs, this may be separate plots on the same plot axis or two separate graphs.*

6.4 Noise Tests

QLA-UR-DN-01	Calculate running estimates of noise by averaging. *This will be required when running noise tests. Running noise estimates will be calculated from the time series for an individual or group of detectors. To calculate the running estimates the user will select an interval (either by time or number of samples). After each interval, the mean and standard deviation (noise) of the detector time series is calculated.*
QLA-UR-DN-02	Calculate running estimates of noise using power spectrum. *This will be required when running noise tests. Running noise estimates will be calculated from the time series for an individual or group of detectors. To calculate the running estimates the user will select an interval (either by time or number of samples). After each interval the detector time series is Fourier transformed to the frequency domain. The power spectrum is displayed and the running noise estimate is calculated by either integrating under the power spectrum or by measuring the power at selected spot frequencies. These frequencies should be defined either ahead of testing or selected at the start of testing to allow a consistent database to be formed.*
QLA-UR-DN-03	Display running noise estimates *This will be required when running noise tests. This could take the form of a split plot window, with one plot showing the running detector time series and the other one showing the running noise estimate.*
QLA-UR-DN-04	Display final noise power spectrum for a selected detector. *Once a test is complete the noise at each frequency is calculated by an FFT on the complete time series. This noise vs frequency plot is then displayed.*
QLA-UR-DN-05	Store the final power spectrum. *Essentially this could be a two column table, this is related to the test by use of a test number made up of the test ID and the test sequence number TBS.*
QLA-UR-DN-06	Recover and display long-term noise values for a selected detector. *A period of time (days-months) is input as a query and the noise estimates at the pre-defined frequencies is recovered. The display can either be in the form of a plot or list selected by the user.*
QLA-UR-DN-07	Retrieve standard/previous power spectrum for a detector and display. *It would be desirable to be able to display the power spectrum from a previous test alongside the one produced by a current test.This retrieval can take place before, during or after a test and could be done with one command.*

6.5 SMEC

QLA-UR-SM-01	Extract correct SMEC position *Extract the SMEC position at the time of the detector samples by interpolating SMEC positions using the SMEC velocity information.*
QLA-UR-SM-02	Re-construct interferogram
QLA-UR-SM-03	Plot interferogram *Will be done for selected detecto(s)r at the end of each scan*
QLA-UR-SM-04	Transform *Convert the interferogram to a spectrum*
QLA-UR-SM-05	Plot spectrum. *The plot can either be signal vs wavelength (TBD), frequency or wavenumber (user chooses).*

7. ANALYSIS

This section describes offline (as opposed to real time) functionality as required by the execution of individual tests. It will be necessary for the QLA environment to support the use of scripts. These scripts will dictate how the components described in this section are put together.

7.1 Load Curve Tests

Although the production of load curves can be done offline, it would be desirable to have the following requirements met by the near real time part of the system.

QLA-UR-DF-01	Convert detector data to voltages. *Each detector signal must be converted from the extracted bit pattern. It must be converted to a number, this number is then divided by the gains within the system then the offset is added to give the actual output voltage of the detector.*
QLA-UR-DF-02	Determine mean detector voltage. *I'm assuming for a load curve test that we step through each input current taking several detector readouts at each step. If this is the case a single voltage will have to be determined by averaging the measured voltages for that step.*
QLA-UR-DF-03	Extract VI. *This is only required for load curve tests. The input currents exist in H/K packets. The detector readouts and current values need to be put on a common timescale, then for each current step the each detector voltage must be determined. (Note all detectors must be extracted for purposes of storing this reduced data)*
QLA-UR-DF-04	Display VI for any selected detector. *Specifically plots a Voltage (Y-axis) against current (X-axis curve for a single detector. The V and I values will have to be generated from the test results.*
QLA-UR-DF-05	Display VI for a selected set of detectors. *Put a set of VI curves for an individual array, or selected part an array in to one plot window.*
QLA-UR-DF-06	Store VI values. *It is envisaged that each test will produce one or more load curves for each detector in each array. These should be stored for later retrieval with a pre-defined set of H/K parameters (TBD) which need to be defined in advance of testing.*
QLA-UR-DF-07	Retrieve stored VI values *The user should be able to select one or more previous data sets. The minimum requirement is that this is done via a command but a selection GUI giving information about each stored set would be desirable.*
QLA-UR-DF-08	Plot retrieved VI values on same graph as current VI curve. *After a load curve test we may wish to compare graphically with a previous test, this is not needed in real time.*

7.2 Spectrometer Tests

QLA-UR-SM-06	Re-construct LVDT position for Moire step *This should be apparent from the organisation of the TM from the MCU. This is required to determine how the LVDT changes with optical position as the LVDT is expected to be non-linear.*
QLA-UR-SM-07	Plot LVDT vs fringe position. *This will be used to check how reliable the LVDT values wrt fringe position.*
QLA-UR-SM-08	Plot actual position vs time. This is to look for linearity.
QLA-UR-SM-09	Plot difference between the LVDT position and re-constructed fringe position against fringe position. *The plot in QLA-UR-SM-02 is likely to be close to a straight line. The difference plot allows us to look at the small deviations from this linearity.*
QLA-UR-SM-10	Plot velocity error v time or scan position (or Delta T v step no.) *This can only be done using the SMEC in trace mode as these values are not in the science packets. (TBC).*
QLA-UR-SM-11	Plot spectra Scans can be combined in the spectral domain and the built up scan plotted. *It is not yet clear if interferogram scans can be combined before transformation to the spectral domain.*
QLA-UR-SM-12	Find ZPD Scan using internal calibrator + shutter to define input signal. *QLA will have to use*

	fitting algorithms (TBD) to do this.
QLA-UR-SM-13	Characterise ZPD *Once we've found the ZPD on the optical axis, the shift off the optical axis should be a parabolic shape and this shape must be defined for the calibration. Therefore we need to find the ZPD for each detector then we will need to fit the overall shape, this may be done offline.*
QLA-UR-SM-14	Store ZPD Will have a different ZPD for each detector, could be a 2 column table.
QLA-UR-SM-15	Extract optical path difference per detector sample. *Convert mechanical position to OPD for each detector.*
QLA-UR-SM-16	Calibrate interferogram *Many steps, are needed here (TBD).*
QLA-UR-SM-17	Phase shift interferogram.
QLA-UR-SM-18	Combine interferograms *This will need defining as the tests progress but we certainly need the ability to de-glitch and phase shift.*
QLA-UR-SM-19	Interferogram storage Storage. *We will want to store both interferograms and spectra from each stage in the processing, hence if we decide on the new calibration scheme we can recover the data at the stage needed. We will also want to store the relevant HSK parameters and this will be defined ahead of testing.*
QLA-UR-SM-20	Construct Data Cube For operating mode tests we will want to be doing stuff with images and constructing data cubes.The only functionality QLA is required to have is the data cube construction, further analysis is likely to take place elsewhere.

7.3 Spectrum Reconstruction

QLA-UR-SP-01	Import a model spectrum for RSRF determination. To determine the RSRF the interferogram shape from the detector readouts needs to be divided by the expected source shape i.e. the Fourier transform of the input black body spectrum.It is assumed that this spectrum is generated externally.
QLA-UR-SP-02	Determine spectral response function. For the photometer a facility FTS will be used giving a known input spectrum. At each point in this spectrum a single signal value will have to be derived, then the signal values put together to form the detected scan. The RSRF is determined by simply dividing the output scan with the input spectrum. Although this is likely to be done offline, QLA should be able to assoviate the detector data with the facility FTS position and give an output file in a suitable format for easy RSRF construction.
QLA-UR-SP-03	Store RSRF Analysed data stored for each detector, this may require storage in the HCSS from data reduced externally to QLA.

7.4 Mechanism Tests

Cooler and Shutter tests need to be included

QLA-UR-MC-01	ILT,IST	Reconstruct BSM pointing. *We will need to scan the BSM off the centre of the array and determine where the image falls in relation to BSM position.The required QLA functionality for this is stated elsewhere in these requirements.*

7.5 Other Functionality

The following tests do not yet have clearly defined requirements on QLA.

QLA-UR-OF-01	Monitor VI Monitor VI (R_{op}) v time (while carrying out any other operation) – e.g. to check for warming of parts of instrument by mechanism or calibrator operation
QLA-UR-OF-01	Measure crosstalk Both optical and electrical crosstalk will be measured at the subsystem level, we will need to measure these at system level.

QLA-UR-OF-01	Glitch analysis It is not yet clear what glitch analysis will be needed for ILT.