SALT Pipeline Software Modules - Plan and Status

Petri Vaisanen, Steve Crawford, Ted Williams, Éric Depagne – June 20, 2015.

This document is *not* a design document for the pipeline. It is merely a brief summary of the main modules and work needed, as well as an approximate status report.

It is important that SALT pipeline development is managed and executed as a real project with specific user requirements, a design plan, controlled resource allocations and time-lines including plans for testing and verification before release. This very first step has not been done yet due to lack of resources.

We estimate *3 man-months* for writing up a detailed plan for each of the modes (excluding NIR for now, which needs more time). Some of these can be collected from pieces of documentation written for parts of data reduction tasks over the years. An HRS pipeline design document is in preparation, but is proceeding slowly due to other commitments of all individuals working on it.

The time estimates below for coding of the pipelines for instrument modes are still very crude estimates. A better estimate can only be done once the proper design documents are finalized.

Common requirement for all modes (needs major work from a programmer):

- supportable: based on Python, though algorithms and tasks can be called, or assimilated, from other existing software tools.
- the pipeline for a given mode works both in black-box automatic mode where final data is delivered to user, and as a stand-alone installation with options of running steps interactively.
- packaging needs to be professional standard so that export and installation is hassle-free over a range of platforms at Partner institutions.
- fully documented with user manuals

All modes assume that the preliminary pipeline reductions are completed, i.e. gain, overscan, bias, mosaic and cross-talk corrections.

RSS Fabry-Perot Software Modules

first versions can support data plane only, but second (standard) version should support bad pixel and variance planes also

1) existing 2-D cosmic ray identification and flagging/cleaning; 3-D scheme would be a good second level enhancement

2) collect and improve calibration ring analysis routines

3) wavelength calibration (existing code just needs improvement)

4) night sky fitting and subtraction (continuum and line)

5) flat fielding with stellar photometry (when enough star density), frame normalization, flux calibration (to ~10%)

6) FP ghost measurement and subtraction (existing code just needs improvement)

7) seeing/IQ normalization (convolution/deconvolution)

8) Line profile fitting and display (existing code needs generalization)

9) kinematic mapping (existing code needs enhancement)

10) data product assembly and visualization tool

Time estimate: 6 months

RSS long-slit software modules

- Software for many of the steps exists already, but they have not been tested and verified nor generalized enough to work reliably. (TBD: have to indicate somehow which steps need the most work still)

1) Cosmic ray identification and flagging/cleaning (tool exists but needs to be generalized to work in all situations)

2) Spectroscopic flat-fielding (pixel-to-pixel and illumination, the latter strategy needs to be defined before coding)

3) Arc line identification and wavelength calibration (tool using RSS model as basis exists but needs to be generalized to work for all lamps and configurations, needs a library to be built)

4) 2D frame rectification (working tool exists if previous step worked)

5) Sky background subtraction (needs definitions and coding, requires both automated and easy-to-use interactivity)

6) Spectral extraction (needs definitions and coding, requires both automated and easy-to-use interactivity)

7) Flux calibration (needs definitions and coding, a library to be built)

Time estimate: 6 months total including both data reduction expert and programmer, most of the work will be the generalization, testing and packaging of current pieces of code. Some of this work directly benefits MOS mode work and will likely cut down the time needed for the latter.

RSS MOS long-slit software modules

- After initial steps MOS follows a similar philosophy as LS, the steps are not repeated below
- While some of the steps described below have code and tasks written already, they are far from the quality of a general use pipeline.

1) Indentify slits (code has been developed to do this, but not properly tested or generalized)

2) Separate slits (as above)

3) Follow LS-recipe module steps (these have not been adapted to work with MOS yet; largest differences will come from flat fielding, slit distortion corrections, and the spatial offsets of the slits meaning that wavelength ranges are different for each slit, libraries will have to be built to make this automatic)

4) Re-assemble calibrated MOS data.

Time estimate: 3 months total including both data reduction expert and programmer; many parts of the work can be done concurrently with the LS-mode coding. Most of the MOS-specific work will likely be the generalization and testing of the slit-finding and separation tools.

RSS Polarimetric software modules [TBD. Steve Potter]

1) xxx

2) yyy

Time estimate: 6 months. This does *not* include work to be done during the commissioning period of just integrating Ken Nordsieck's basic polarimetric tasks to our system.

HRS software modules

- The HRS pipeline will be designed to work for LR, MR and HR modes.
- HS mode will be discussed separately.
- The preliminary data processing steps include HRS specific steps e.g. to reorder the data frames, correction of the overscan, etc. and subtract scattered light

1) Creation of all master calibration frames (can already be done, but not tested)

2) Tracing the orders and fit cross order profiles (using a formatcheck frame and the model of instrument, prototype works for MR)

3) Optimally extract the orders, correct for blaze and tilt, and remove cosmic rays (prototype for MR works, not tested)

4) Wavelength calibration: extract lines from ThAr, fit for each order, apply solution (prototype works for MR mode with ~500 m/s accuracy, no proper testing yet)

5) Subtract sky background (not attempted yet)

6) Flux calibration using standard stars (not planned yet)

7) Merge orders (not planned yet)

Time estimate: 9-12 months total including both data reduction expert and programmer. While some of the individual steps above already have existing and working code for at least the MR mode, it is practically *ad hoc* work without a design document. The largest amount of work will likely be designing, implementing *and testing* the optimal ways of achieving the intended velocity accuracy per mode.

RSS NIR software modules

- *There is no plan yet*. It is assumed many of the RSS long-slit modules could be modified and used for the purpose. Such NIR spectra are typically taken in nodding mode with multiple spectra obtained at different locations to help accurate sky-subtraction. However, it now appears an IFU unit will be delivered. There is no experience within SALT Operations or SAAO for developing IFU data pipelines, in fact, not even experience with reducing IFU data. We will be in contact with Marsha Wolf and also investigate available packages at other observatories.

Time estimate: 24 months total including both data reduction expert and programmer, a very crude estimate without any detailed knowledge of the problem yet.

SALTICAM software modules

Only slot-mode pipeline is to be supported at this stage. It already exists. After flat-fielding issues are sorted some time in the future (it is a P2 SALT project) normal imaging could be pipeline processed as well to provide e.g. automated photometry.

Time estimate: 3 months total including both data reduction expert and programmer, if it is decided normal imaging and frame-transfer modes need a pipeline. This is work *after* flat-fielding is understood.