Software/Hardware Development Items for MMTAO

Draft 1

Items I'm aware of and have at least taken token steps toward either fixing (or developing plans to fix), in rough order of importance.

1) Bring **NGS PCR** code base to a **manageable, understandable state** (without breaking it)

- Crucial for future development; important for us to understand how the system works.
- Confusing dfead code & unused versions of files removed from tree and archived
- Build against libraries in-tree only (some libraries are built in-tree and then copied elsewhere manually for dynamic linking. This is a pitfall waiting to happen.
- Refactored and pruned source tree is then built & tested on sky to verify operation
- Re-enable PID loop control (had been commented out for 2 years!)
- Check-in validated tree into version control (currently SVN on SORAL)
- Current PCR running from AO-GUI is Vidhya's reference version. Once we are ready, switch over to this codebase by default.

Status: 80% complete.

2) Try to get 28x26 bin-by-3 images pulled into PCR via **fiber optic frame grabber card** versus the old AIA 'scsi' card

- Initial tests promise to **solve freezing WFSC problem** – single most infuriating aspect of MMTAO operations and **major cause of non-weather-related downtime**.
- Camera images show flux but in addition to stair-stepping (images are supposed to be 26x26), the images are partially scrambled.
- Looks a bit like de-interlacing problem but isn't. Counts look OK so ADC is OK too. Need to understand how sciemeasure treats this mystery FO board and how images are dumped into the framegrabber
- Next step: use standard EDT tools to pull images and try to get any kind of info from scimeasure. Measure known pattern with camera in lab and get the readout image regridded appropriately. If 28x26 readouts simply doesn't work, we can try pulling 80x80 frames as LGS did and rebin the data in PCR.

Status: 50% complete

3) Along the way to debugging #2... **how does our camera controller work**? Major uncertainty about how the camera controller is configured. We get frames at 537 Hz with no averaging or waiting, but this corresponds to no known standard programming mode of the scimeasure controller. Do we have custom firmware? LGS used this camera. How did they do this?

- Requires sitting down with the camera, possibly with oscilloscope, reverse engineer if necessary how we are clocking this FPA out.
- Gaining new control over the frame rate is one way to improve performance. We should in theory be able to run the mirror up to 800 Hz with a significant gain in Strehl, the mirror should respond favorably. Easiest solution is to run at a high master rate, like 5000 Hz, then add integration cycles to get lower rates. Eg. 500 Hz would be achieved by running the master clock at 5000 Hz and integrating for 10 cycles before dumping.
- I was shocked; this is critical knowledge that appears to have been lost.

Status: 20% complete. Know precise questions to ask! Probably needs 1-5 days of hard poring over documents, controller, or getting the ear of someone at Scimeasure, 1-2 days to document fully.

4) **AO Operator interface simplification**.   Current operation of the mirror is a numbingly-endless back and forth clicking between AO-GUI and XADSEC. Easy to screw up. Using remote procedural calls in AO-GUI, with a listener in XADSEC, it should be possible for AO-GUI to update menu items and functions in XADSEC, leading us in the 'nothing goes wrong' path to the elusive "one button" closed loop AO operation.
Status: 20% (developed 'hello world' proof of concept using TCL/TK 8.x and IDL 8)

5) **Weird PCR behavior** to understand and fix if it points to a problem.

- PCR has a massive **memory leak** when in closed loop mode. Once we solve the WFSC freeze, it is likely that the loop will next break when all memory has been exhausted.
- **PCR spins 100% on the CPU in open loop** mode.   Closed loop mode seems to be >50% idle. Odd. Why is this?

Status: 10%

6) Feed-forward of **accelerometer** input and **IR tip/tilt** input direct from ARIES' guide probe.

- Some code already available in PCR. Where do accelerometers stand with Keith?
- Where does the tip/tilt probe stand with ARIES? Tested once using standard imaging detector, did manage to provide guiding capability at 1 Hz. Need GenIII controller on spectrometer permenently to free up a GenII controller for the tip/tilt FPA.

Status: 20%, some code available, but overall design needs to be documented and updated

7) **Reconstructors**. How many of the 56 modes are we fruitfully correcting and how many are just noise? Re-introduce old familiar topics such as modal gain and so forth.
Status: 5% (planning phase)

8) The **seeing monitor** hasn't worked in years. Do we bother? From a facilities point of view, this seems like something we should have working.
Status: 1% (know where the code is but that's it)

9) **PCR suffers to some degree from jitter**, since the realtime loop is running on a soft-realtime system. Some steps could improve the existing system, however standard RT patches exist for the Linux kernel to **provide hard realtime support**, which could still be slaved to the camera frames. We just need the mirror commands to go out **exactly** on time.
Status: 5% design phase

Craig Kulesa
11 June 2015