Automated Phase Retrieval Control Update
April 30, 2009
Rick Burruss

Introduction

The Palomar Adaptive Optics group and Gene Serabyn agreed in February 2009 to work together to resurrect the Automated Phase Retrieval Control scripts (APRC, aka MGS) written by Sidd Bikkannavar in 2007. These scripts automate the tasks of acquiring Pharo defocused white light images and processing the images on computers at JPL using the Modified Gerchberg-Saxton (MGS) routines. Using just two defocused Pharo images taken at equal separations before and after the Pharo focal plane, the MGS routines measure intensity variations to estimate the non-common path wave front error (WFE). In 2007 these APRC scripts typically reduced the non-common path WFE from 150 nm to 40 nm[1], where the 150 nm is typically the best that can be done by hand tuning PalAO centroid offsets using a few Zernike polynomials. When each iteration of the MGS routines are finished at JPL, the inverse of the estimated error is fit into the current DM control map (flatmap), and this new map is written to disk and sent back to Palomar as the basis for the next iteration. The entire process took nearly 10 minutes per iteration in 2007, and usually takes 3 iterations to reach the 40 nm results.

Sidd showed that the 40 nm WFE is split between 18 nm of controllable spatial frequency errors and 29 nm of uncontrollable high frequency error. Note that these higher spatial frequency errors will be controllable by the P3K PalAO upgrade.

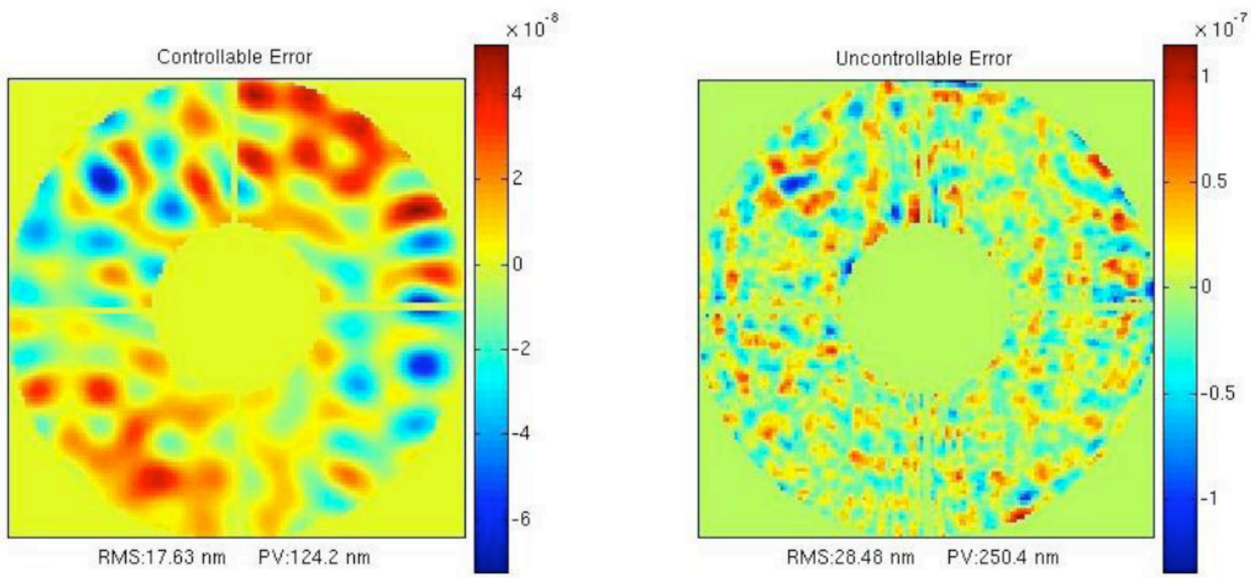


Fig 1 Controllable (left) and uncontrollable (right) components of the final MGS WFS results in 2007

The APRC Strehl ratios of the PalAO internal white light source measured in Pharo science images improved by 10-15% over images that were tuned by hand. While the APRC method produced PalAO control maps that led to

---

1 Bikkannavar et al., “Autonomous Phase Retrieval Control for Calibration of the Palomar Adaptive Optics System”, SPIE 7015 (2008)

excellent open loop results, unfortunately we were never able to derive centroid offsets from these control maps, and so were not able to use these results in closed loop or on sky. A short report by Troy & Roberts in April 2007[2] listed a few other remaining issues and bugs in the APRC scripts. We hoped then to resolve these issues before releasing APRC as a standard PalAO procedure, but this was never completed. Following is a detailed report on the work done to update the APRC scripts so that they can be used by Palomar staff in place of the hand tuning method.

Changes at Palomar

There have been a number of changes to PalAO since April 2007 that impact the APRC work:

- The PalAO bench was realigned to allow for larger science instruments and some of the optics were recoated in 2008. This means that the DM influence functions measured for the MGS routines in 2007 are no longer valid and need to be redone.
- The PalAO stimulus package (white light source) was replaced by an all-reflective design in 2008. Since the relay design is 1:1, the required MGS parameters for PalAO should be unchanged. The optical surfaces of the diamond turned OAPs ($\lambda/4$) may introduce surface errors into the MGS calculations that were not present in 2007.
- The PalAO flatmap procedures and filenames are slightly different than in 2007.
- The aousr and aprc user accounts at both Palomar and JPL have either been changed or deleted. Secure shell versions between Palomar and JPL have changed.

Summary of APRC Work Done

The APRC routine is started from Palomar (harbor) via a matlab script as before (aprc_tao.m), but the entire process is now contained in one step:

```
%   Normally, the user enters only the iteration number as an input.
%   The routine will create directory structures based on the current
%   date.
%      Example: >>aprc_tao(1)
%
%   The user may force a date in 'YYYY-MM-DD' format as the
%   second input at the command line.
%      Example: >>aprc_tao(4,'2007-03-27')
```

Following is a summary of the work done:

1. Refined secure shell connections between Palomar and JPL so that the entire APRC iteration is run in one step

- Removed the APRC accounts at both Palomar (harbor) and at JPL s383 computers (not needed)
- Resurrected the AOUSR account at JPL
- Established MORGAI (Sidd's computer) as the computer at JPL to process MGS
- Created ssh keys for the aousr accounts on both harbor and s383 computers (ie: hooligan) to allow for password free connections going both directions
- ***Wrote csh scripts to control the remote login and matlab startup from harbor to morgai***
    - morgai.s connects from hooligan to morgai and passes the iteration number
    - run_aprc.s starts matlab on morgai and runs aprc_jpl.m with the passed iteration number
    - init.s is run on hooligan if iteration = 1 in order to setup the necessary directory structures (solution to scp incompatibilities)

---

[2] Troy, Roberts, "Automated Phase Retrieval Control Testing", (April 2007)

2. Copied the high level MGS scripts from Sidd's home directory into the JPL aprc directory (/proj/palao/aprc/dev) so that they can be edited and run "locally".

3. The current PalAO flatmap is copied from its default location into "control_00" on both harbor and hooligan if iteration = 1

4. Fixed the JPG output so that the APRC results are easier to see on harbor. All of the secure shell connections are opened through X window tunnels so that matlab windows created by APRC scripts at JPL are forwarded to harbor (useful for influence measurements when run from Palomar)

5. Identified and fixed the bug that prevented proper creation of centroid offsets from derived control maps (IDL was converting to arcseconds). Added a prompt to the end of aprc_tao.m that asks the user if they want to create centroid offsets (Y/N), which they will usually do after ~ 3 iterations when the WFE is minimized. This part of APRC loads the necessary PalAO parameters, turns on the high order loop on the white light source, and extracts 30 seconds of average centroid positions. 6 files are then written to disk

    - /proj/aocp/tables/cent_offsets/co_mgs (co_mgs2, co_mgs4)
    - /proj/aocp/tables/cent_offsets/co_mgs_<fdate>_<iter> (co_mgs2_<fdate>_<iter>, co_mgs4_<fdate>_<iter>)

    where co_mgs2 is (co_mgs / 2) and co_mgs4 is (co_mgs / 4)

6. Added a matlab script to create the centroid offsets at any time given the control map and the date

    ```
    %   make_cents(map,fdate)
    %
    %   map = control map# (ie: for control_04, map = 4)
    %   fdate = date path to control map (ie: fdate = '2009-04-21')
    %
    %   example:  make_cents(4,'2009-04-21')
    ```

7. Found that iteration #1 needs to be repeated until focus is completely removed before proceeding to iteration #2, and that moving Pharo is the best & easiest long term solution. If the final focus term > +/- 120 nm then the APRC script reports to the screen that the Pharo focus screw needs to be turned, and gives the proper direction and amount (analyze_wfs.m). Iteration #1 is then run again until this focus term is < +/- 120 nm.

8. Found that the ideal diameter for FFT centering in make_img.m is 80 rather than 96 (improves final WFE by 5 nm)

9. Created new influence gain matrix (see procedure below). The original code poked a grid of 16x16 actuators over 9 iterations at different locations to map the entire DM. The poke height was 300 nm, which led to the phase wrapping problem reported by Sidd on one side of the DM. I tried other poke heights (290, 270, 250, 200) and found the 250 nm pokes produced the best WFE results with no phase wrapping to deal with. This influence gain matrix is now the file in use for APRC..

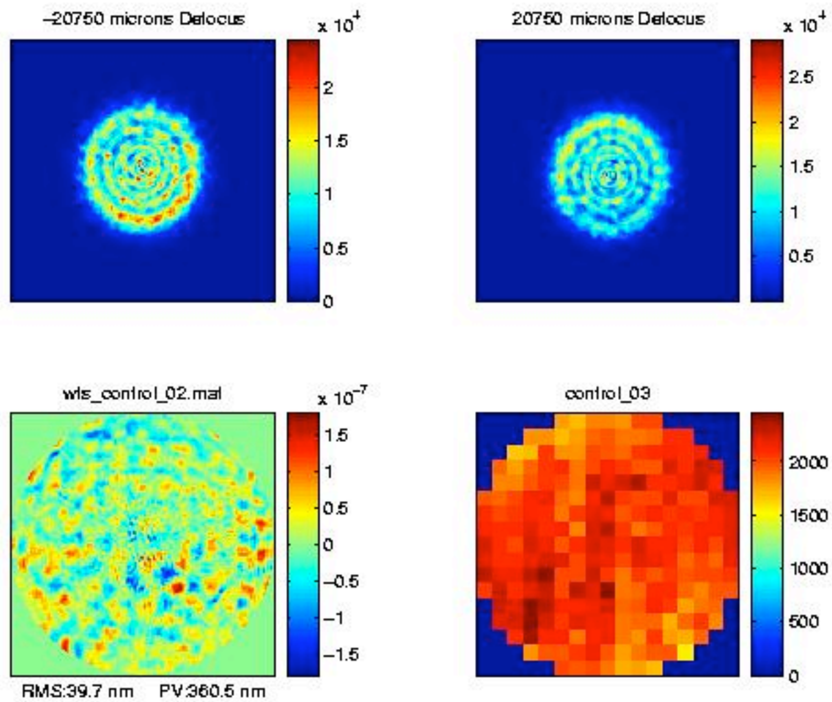
10. Cleaned up pharo control matlab code to optimize efficiency. Optimized white light power and order of palao and pharo commands.

11. Found that the "gain_factor" parameter in make_control.m does not change the resulting WFE, only the time it takes to reach the final result. 0.75 remains as the best value.

12. Found that Pharo's bad pixel map is unchanged since this data was taken in 2003, so it is still in use now.

Results

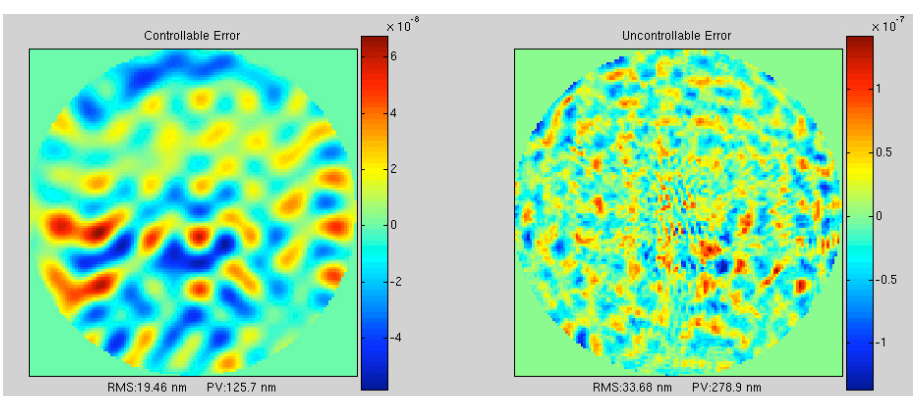
APRC is now fully operational at Palomar, and is run from a single matlab command (an iteration). Each iteration takes about 5 minutes to complete, and when finished an image appears on the screen that looks like this



The top 2 images are defocused raw pharo images. The bottom left image is the estimated wave front error, in this case the residual error is 39 nm total. The bottom right image is the control map which is automatically copied into the current white light flatmap. 3 iterations are usually required to reach the best WFE in most cases, although fewer iterations are required if the process has been run within the past 12 hours or so.

Palomar staff has been trained to run this script from harbor, and trained to focus the Pharo camera by hand during iteration number 1.

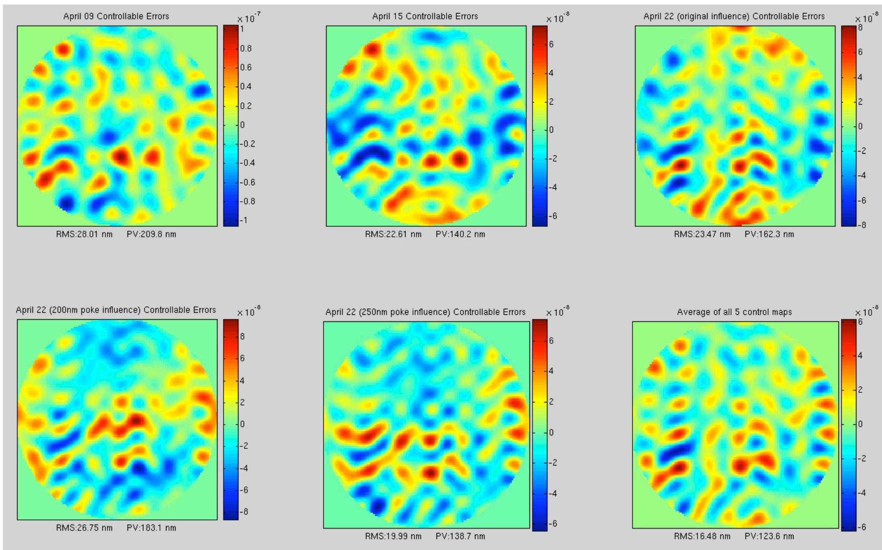
This image shows the split between controllable and uncontrollable spatial error from April data, verifying that mgs is working well.



Remaining Work

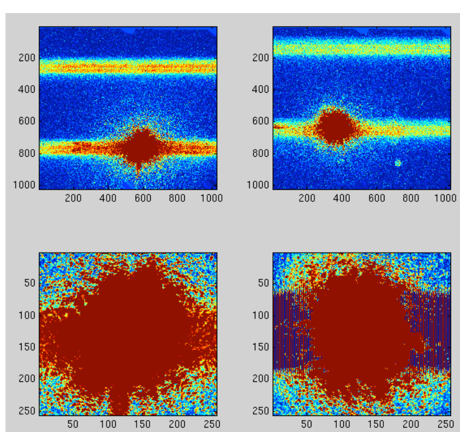
1. Need to fix bug in make_cents.m and aprc_tao.m where time is not calculated correctly after daylight savings. Harbor matlab version is too old, does not allow newer datenum functions, so need to get time from ao telemetry (ao_status)

2. After looking closely at the controllable WFE from final APRC results taken over a few days, it is clear that we can do better. (see image below). Cathy Ohara thinks we should spread the poke pattern out when measuring influence of the DM by using fewer pokes in a grid and taking more data. I plan to do this in June 2009.

   These images show the controllable error taken over a few days in April 2009. The last image in the lower right is an average of the 5 results, which we should be able to control. We think the current influence measurements overlap with each other, so spreading the pokes out should help reduce the final error by 10-20 nm. So it is realistic to expect WFE results as low as 15-20 nm in total.

3. We found odd patterns in intermittent raw pharo data after the initial centering algorithm was finished (see below). The error occurs in CentroidCenter.m at some point during the FFT conversion to or from the image plane. I was not able to find the problem, but it does not appear to cause WFE residual. This error appeared in March and April data, but was not present in May. Cathy thinks is due to science camera anomaly, perhaps bad pixels or elevated readnoise, but I was not able to verify this.

Influence Procedure

1. Send current flatmap_wl to morgai in fdate/poke/control_00

2. Run make_poke_patterns. Leave dpath, dfile, oldflat, and dstem as is. The default poke height is 300 nm. Gain is the multiplier to change this value. Gain = 1 -> 300 nm, Gain = 0.833333 -> 250 nm (what we want)

3. Transfer these to harbor after creating fdate/poke

4. on harbor, run auto_pokes

5. Send these pokes back to JPL

6. Edit & run make_img_pokes (make sure fdate is correct)

7. Edit & run palao_mgs_pokes2. Check each of these in Sidd's gui

8. Edit & run make_sensitivity (make sure fdate, sname, and mpath are correct)

MGS GUI

1. addpath /home/sbikkann/MGS_WFS
   addpath /home/sbikkann/matlab

2. load ('wfs_p1.mat')

3. mgs_gui_rick(wfs)

4. remove zernikes

5. select region with mouse drag box, choose color for replacement (red)

6. fit zernikes (~ 50 – 80) or apply filter 11, 47; 80, 11

7. save prior and/or save wfs_p1 wfs

8. palao_mgs_pokes2.m (you can specify which priors to run [4, 6, 8]

To Split Between Controllable and Uncontrollable Errors

1. cd to WFS directory

2. addpath /home/sbikkann/MGS_WFS
   addpath /home/sbikkann/matlab

3. load the wfs that you want   load ('wfs_control_05');

4. wfs=mgs_gui_rick(wfs)   ->   remove zernikes

5. set low pass filter to 8 cycles / pupil @ 0% rolloff

6. Apply and save

7. figure(1),subplot(1,2,1),simage(wfs.prior,1,1)

8. temp=zernike_remove(wfs.opd_mgs,1:3);

9. figure(1),subplot(1,2,2),simage(temp,1,1) for combined error

   or

   figure(1),subplot(1,2,2),simage(temp-wfs.prior,1,1) for just the uncontrolled error

10. if on a mac, grab the image using preview and save as jpg