UK Astronomy Technology Centre

Gemini A&G wavefront processing system: calibration for, and using, ospLib

Steven Heddle
21st January 1999 wfs_sbh_002/01

---

1 Introduction

This document attempts to outline how ospLib functions should be used to obtain the control matrix which is applied to the measured centroid positions to get the Zernike coefficients. The basic principles used are described, as well as the functions used to make corrections to null positions, and calculate the control matrix offline from stored data. The data used, their formats and creation, are also described.

2 Basic principles

The basic principle is that we have a number of frames of calibration data showing Shack-Hartmann spots, whose Zernike coefficients are known precisely and are a function of the displacements of the spot positions from null positions obtained when no aberrations are present. Using this information we can calculate the Zernike coefficients of a frame whose spots form an arbitrary set of displacements within their subapertures, in terms of Zernike functions which span the space described by our calibration data.

First we consider a simplified version of our problem where during our calibration we may obtain images aberrated in terms of only a single Zernike polynomial, for each of the polynomials that we wish to correct for. We adopt the convention that matrices are represented by upper-case letters, and vectors represented by lower-case letters, both emboldened as part of text.

In terms of linear algebra, the Zernike polynomials are orthogonal, and constitute a basis set for the space spanned by them.

The problem may be basically formulated as follows:

$$\mathbf{s} = \mathbf{R}\,\mathbf{a} \tag{1}$$

where $\mathbf{s}$ is the column vector of signals (e.g. x and y displacements of centroid positions from their nulls in the subapertures of the wavefront sensors), $\mathbf{R}$ is the reconstructor matrix relative to the basis set of single Zernike polynomials, and $\mathbf{a}$ is the column vector of Zernike coefficients corresponding to the various amounts of aberration due to each of the Zernike polynomials in the basis set $\{Z_i\}$.

Using an on-axis object and on-axis wfs we impose known aberrations on the primary mirror and measure the response of the wfs. Population of the reconstructor matrix may proceed a row at a time by dialling-up each of the individual Zernike functions we wish to correct for to some known degree (equal to the function's Zernike coefficient) and recording the centroid positions in each of the subapertures. This represents a set of solutions to the forward problem defined above, which is to determine the centroid positional responses given a Zernike function stimulus. We can then use least squares methods and singular valued decomposition (SVD) to determine

the control matrix $\mathbf{C}$ which is the solution to the inverse problem of determining the Zernike coefficients given the set of centroid positional responses, which may be defined as below:

$$\mathbf{a} = \mathbf{C}\,\mathbf{s} \tag{2}$$

Thus to obtain the vector of Zernike coefficients $\mathbf{a}$ for an input frame, we multiply the vector of spot displacements $\mathbf{s}$ obtained from that frame by the control matrix $\mathbf{C}$, which is relative to the basis set of single Zernike polynomials. Clearly

$$\mathbf{C} = \mathbf{R}^{-1}, \mathbf{R} = \mathbf{C}^{-1}. \tag{3}$$

3 Generalising to arbitrary basis functions

The problem is complicated by the realities of our situation. In practice we will be more likely to achieve basis functions which each are some linear combination of Zernike polynomials. For instance, it is probable that when the primary is called upon to impose some amount of abberation corresponding to a single Zernike polynomial, the result (as characterised by the Zernike coefficients measured by the HRWFS) will be some linear combination of Zernike polynomials. Thus when we populate the reconstuctor matrix a row a time by imposing our basis functions we obtain a reconstructor matrix $\mathbf{R}'$ which is with respect to basis functions $\{Z'_n\}$.

These may be expressed in terms of the Zernike polynomials like so:

$$\begin{aligned} Z'_1 &= f_{11}Z_1 + f_{21}Z_2 + f_{31}Z_3 + \ldots & (4)\\ Z'_2 &= f_{12}Z_1 + f_{22}Z_2 + f_{32}Z_3 + \ldots & (5)\\ &\vdots \end{aligned}$$

where the f_{ij} constitute the elements of a matrix $\mathbf{F}$ which expresses the relationship between the Zernike functions Z_i and the basis functions Z'_i.

$$[F] = \begin{bmatrix} f_{11} & f_{12} & f_{13} & \cdots \\ f_{21} & f_{22} & f_{23} & \cdots \\ f_{31} & f_{32} & f_{33} & \cdots \\ \vdots & & & \end{bmatrix} \tag{6}$$

Considering a vector of spot displacements $\mathbf{s}$, the frame from which it was obtained shows abberations in terms of its Zernike coefficients $\mathbf{a} = \mathbf{C}\ \mathbf{s}$, or in terms of its arbitrary basis function coefficients $\mathbf{a}' = \mathbf{C}'\ \mathbf{s}$. Thus we may express the aberration $\phi(r,\theta)$ as

$$\begin{aligned} \phi(r,\theta) &= a_1Z_1 + a_2Z_2 + a_3Z_3 + \ldots & (7)\\ &= a'_1Z'_1 + a'_2Z'_2 + a'_3Z'_3 + \ldots & (8)\\ &= a'_1(f_{11}Z_1 + f_{21}Z_2 + f_{31}Z_3 + \ldots) + \\ &\quad a'_2(f_{12}Z_1 + f_{22}Z_2 + f_{33}Z_2 + \ldots) + \\ &\quad \ldots & (9) \end{aligned}$$

$$
\begin{aligned}
&= (a'_1 f_{11} + a'_2 f_{12} + a'_3 f_{13} + \ldots) Z_1 + \\
&\quad (a'_1 f_{21} + a'_2 f_{22} + a'_3 f_{23} + \ldots) Z_2 + \\
&\quad (a'_1 f_{31} + a'_2 f_{32} + a'_3 f_{33} + \ldots) Z_3 + \ldots \\
\Leftrightarrow [a] &= [F][a'] \qquad (10) \\
&= [F]\,([C'][s]) \\
\Leftrightarrow [F]\,[C'] &= [C] \qquad (11)
\end{aligned}
$$

Thus we may calculate the Zernike coefficients in a straightforward manner either by obtaining the control matrix with respect to the Zernike polynomials (equation 11), or by multiplying the coefficients with respect to the arbitrary basis by $\mathbf{F}$ (equation 10).

This is satisfactory to a certain extent, but we have arrived at either of these solutions by inverting $\mathbf{R}'$– consequently we have estimates of the fitting variances for each of the arbitrary basis functions (obtained from the SVD- see next section) rather than for the single Zernike polynomials as required. It would be preferable if we could establish the matrix transformation to obtain $\mathbf{R}$ from $\mathbf{R}'$, and perform the inversion using SVD on $\mathbf{R}$ instead.

We note

$$
[I] = [C][C]^{-1} = [F][F]^{-1} = [C'][C']^{-1} \Rightarrow [C][C]^{-1} = [F]\left([C'][C']^{-1}\right)[F]^{-1} = \qquad (12)
$$

4 Matrix inversion using SVD

5 Necessary data

What we need in terms of calibration data is 1) The file of null positions for the array. The name of this file is specified in the .ini file, and the file itself is an ascii file listing the null positions of each subaperture on a separate line, the coordinates space separated, relative to the coordinate system defined as follows:

and in a definite order - left to right along the rows, a row at a time starting from the bottom. Subapertures which are unused have -1 -1 for the coordinates, and this is all that needs to be done henceforth regarding these subapertures. I mentioned to trim the null positions, start with a file of nominal null positions, do a measurement to get centroid positions relaive to these null positions, and is the measurement is on frame which shows nulled spots, the centroid positions represent the required corrections to be made to the null file. I have been and am working to automate this trimming using a function which was written earlier to do this, but until I have finished this, it is appropriate to give an example file of nominal null positions, for a 6x6 array of subaps nominally perfectly centred, with the corner subaps unused, for an 80x80 array, spots on a 10 pixel pitch, which I believe is our situation- example follows (NB this has 2 pixels subtracted from the current working nullfile and not 1.5 as I had inexpertly estimated yesterday)

```
-1 -1
25.5 15.5
35.5 15.5
45.5 15.5
55.5 15.5
-1 -1
15.5 25.5
```

```
25.5 25.5
35.5 25.5
45.5 25.5
55.5 25.5
65.5 25.5
15.5 35.5
25.5 35.5
35.5 35.5
45.5 35.5
55.5 35.5
65.5 35.5
15.5 45.5
25.5 45.5
35.5 45.5
45.5 45.5
55.5 45.5
65.5 45.5
15.5 55.5
25.5 55.5
35.5 55.5
45.5 55.5
55.5 55.5
65.5 55.5
-1 -1
25.5 65.5
35.5 65.5
45.5 65.5
55.5 65.5
-1 -1
```

So this is the starting point for the nullfile, which should exist before trimming using ospNull-Correction, or by seeing the numbers from a measure on an image of nulled spots. Alternatively get the null positions from WaveLab, make any coordinate system shift, and use an accurate nullfile directly.

2) A dark image which should be a full frame (i.e. 80x80) float FITS file.

3) A flat field, which should also be stored as a full frame float FITS file. This is less important perhaps, and we can progress using the default FITS file of uniform ones (which by the way is created during ospInit if the file specified for multiplicative offsets in the .ini file does not exist).

4) the basis set of calibration frames- brief summary of salient points: For a max of x Zernike corrections we need x calibration frames. These frames should form an orthogonal basis-if they don't there will be Zernikes we cannot correct, but this may well be a function of the primary in any case- for SVD it doesn't matter, as it will make the best of a bad job regardless of whether the psce is over, perfectly or under specified. The Zernike composition of the calibration frames must be known, and will be prompted for in ospNewCalibrate (just run the executable), and saved in a file, with the calibration filenames saved in another file, so this tedious process only needs to be done once. Thus using this calibration frames we can quickly calculate control matrices for e.g. smaller numbers of Zernikes (typically 3 !) or different thresholds. It doesn't matter which order the calibration frames (which each correspond to an arbitrary basis function) are entered-the stage I have included which transforms the recontstructor matrix and thus control matrix so

that they are relaitive to single Zernike basis functions takes care of this. Oh yes, the calibration frames should be full frame, float FITS images (unscrambled like those above) without any background or flat field correction or thresholding- this takes place in the calibration routine.

6 Correction of the nullfile

7 Performing calibration using prepared frames of known Zernike composition

I think I am already on top of this- the typing is just to populate files of the type that you suggest, which are then read as part of the calibration function. Two files are created, one with the name prompted for (e.g. testbb in the 15Jan1999 ftpdir which stores the Zernike composition of the basis functions. This is written out by ospWriteMatrixToFile which ensures that it is in the correct format for ospReadMatrixFromFile later- this data is the matrix F whose inverse postmultiplies the reconstructor from the arb. basis functions to give the reconstructor in terms of the single Zernike functions in the correct order. The format is simple and the file may compiled directly if required. I will explain the format with regard to the seven lines of example testbb:

```
Written by write_matrix_to_file.
Next line shows the row and column dimensions.
3 3
Remember row data should be separated by a single space
5.000000 0.000000 0.000000
0.000000 5.000000 0.000000
0.000000 0.000000 5.000000
```

First second and fourth lines are simply arbitrary comments less than 80 characters long, which are as above when the file is written by ospWriteMatrixToFile. The third line is the row and column dimensions of the matrix, space separated integers. In the case of F the matrix is square, and the dimensions are equal to the number of basis functions which is also equal to the number of Zernike coefficients by which these functions are to be characterised. At this stage, use all the data available, as we can specify the order of the control matrix to be calculated by the wfsSpecific-¿np parameter read in from the .ini file. i.e. if we have 14 basis functions in terms of 14 Zernike coefficients, populate a 14x14 F matrix, as we can later use the same data to calculate control matrices in terms of single Zernike functions for any np less than 14. The space separated floats are the Zernike composition of the basis functions arranged in columns, with the column data starting with the coefficient of tip, then tilt, then the coefficients of the other corrections in order, so that column one corresponds to basis function 1 with five units of Z1, zero units of Z2 and zero units of Z3, with Z1 = tip, Z2 = tilt, Z3 = focus, column 2 corresponds to basis function 2 with etc.

The other file created has the same name, but a .ims extension, and stores the filenames of the FITS images corresponding to the basis function coefficients stored in the other file. The name of each file and the entering of the basis function coefficients are interleaved to ensure correspondence. An example of such a file is given by the 6 lines of testbb.ims:

```
unscrambled
```

```
float
3
data/z1_perf5.fits
data/z2_perf5.fits
data/z3_perf5.fits
```

The first two lines record the type of the files. At present only unscrambled and float images can be used, but some provision has been made to allow scrambled and unsigned_short_int as valid entries in the future. The third line is the number of basis functions and must be equal to the dimensions of the matrix in the other file. This parameter is used when reading these files if they exist already. The remaining lines show the names of the FITS images of the basis functions, and their paths relative to $cwd. Remember, it does not matter which order the basis functions are entered so long as their Zernikes compositions correspond to them. The coefficients of the Zernike compositions must be in the correct order, but this should be ensured by the prompting from within ospNewCalibrate.

When running the calibration, the first prompt asks for the name of a file where zernike composition for each basis function exists - if that file does not exist, we move onto the second prompt which gives the option of trying another filename (in case the name has been entered incorrectly) or creating a file of that name and the associated file with a .ims extension, and populating them by supplying the prompted data. If the file does exist, it and its associated file are read and used to calculate a control matrix, with no further text input.

8 Summary of files input and output as part of the calibration process

Brackets around a complete entry mean it has already been read in as part of ospInit.

8.1 ospInit

In:

- .ini file (e.g.pwfs.ini), text file, which identifies these files

- nullfile (e.g. idealoff6x6_32.dat.3) text file, described above,

- subtractive offsets for dark (e.g data/pwfs.sub), float full frame FITS image

- multiplicative offsets for flat field (e.g. data/pwfs.mult)

- control matrix (e.g. data/pwfs.control), text file, format as testbb above

- vector of fitting variances from SVD part (e.g. pwfs.fvars), text file

If the file of fitting variances does not exist, default values of 99 are used. No file is written out. If the control matrix does not exist, default values of zero are used. No file is written out. These latter defaults are adeequate so long as calibration takes place immediately!

Out:

- If the subtractive offset file does not exist, one is created during ospInit and written out, with uniform values of zero. Name as specified in .ini file. - If the multiplicative offsets file does not exist, one is created during ospInit and written out, with uniform values of 1. Name as specified in .ini file.

8.2 ospNullCorrection

In: (- nullfile, already read in during ospInit and stored as part of context structure)

(- subtractive offsets as read in above, but already stored as part of context structure))

(- multiplicative offsets as read in above, but already stored as part of context structure))

- full frame float FITS image showing nulled spots

Out:

- nullfile, corrected, name as read in during ospInit.

- nullfile, uncorrected, name as above but with .bak suffix

An option exists to not apply subtractive and multiplicative offsets, e.g. if a nulled image already has had thes corrections applied.

8.3 ospNewCalibrate

In:

- file containing list of full frame float FITS images of basis functions

- file containing matrix of Zernike coefficients of these basis functions, (the [F] matrix) for,at as testbb above.

- the aforementioned full frame float FITS images

(- subtractive offsets as read in above, but already stored as part of context structure))

(- multiplicative offsets as read in above, but already stored as part of context structure))

Out: - control matrix, name as specified in .ini file, format as testbb above

- vector of fitting variances, text file, name as specified in .ini file

8.4 ospMeasure, ospFGMeasure, ospCoAdd

None