PUAKO notes #05:
Data analysis facility

O. Beltramo-Martin[*]

January 6, 2021

Contents

1 Rationale

PUAKO possesses several graphical facilities to help you analyzing your results. The first one is the *psfStats* object that is automatically instantiating when grabbing the AO telemetry or doing forward or hybrid reconstruction. The second one is the *statisticalAnalysis* object that is called by the user to read the header of all fits files containing the reconstructed PSF that are saved when using the native PUAKO function to perform the reconstruction over multiple files (*getRecPsf* or *getPrimePsf*). I will present those two objects in the following an detail how calculations are implemented to estimate PSF figure of merits or the AO error breakdown.

[*]olivier.beltramo-martin@lam.fr

2 The psfStat class

When one has initiated the `psfr` (forward reconstruction) or `psfp` (PRIME) of PUAKO, visual comparisons with the sky or simulated image can be done by typing

```
displayResults(p.psfr,'fov',100,'fontsize',20)
```

where `p.psfr` is the psfr property of the PUAKO object. Matlab will display five different figures showing 1D profiles (x,y cuts, azimuthal PSF and OTF profiles) as well as a 2D comparison of the PSF. The user can specify `fov` as the field of view in pixels and the font size for legends and ticks labels.
To analyze more quantitative results, one wants to compare properties `p.psfr.psf` and `p.trs.sky` that are both *psfStats* object. Practically, this class has several properties that are automatically estimated, such as the Strehl-ratio (`p.trs.sky.SR`), the x/y FWHM (`p.trs.sky.FWHMx`, `p.trs.sky.FWHMy`), the aspect ratio (`Ellipticity`) or the encircled energy (`EncircledEnergy`). On top of that, this class gives access to model-fitting facilities for adjusting a Gaussian or Moffat model over the provided image, or scale this latter over a reference image given as an input. The call of *psfStat* is the following

```
pStat = psfStats(im,pupil,wvl,Samp,psInMas,src,'psfResolution',size(im,1),
'flagMoffat',false,'flagGaussian',false,'includeDiffraction',false,'im_ref',[],'fitbg',false)
```

where

- `im` is the image/PSF the user wants to characterize, which can be either a sky or simulated or reconstructed PSF.
- `pupil` is the telescope pupil what serves deriving the telescope transfer function to calculate the Strehl-ratio.
- `Samp` is the PSF sampling ratio given in $\lambda/(2D)$ units.
- `psInMas` is the detector pixel scale in mas.
- `src` is a structure that must contains fields `src.x` and `src.y` so as to define the position in arcsec of each source in the image.
- `psfResolution` is an optional parameter to crop the original image if necessary.
- `flagMoffat` and `flagGaussian` are set to true, the *psfStat* class will adjust respectively an asymmetric Moffat and Gaussian model on the image. The user can choose to fit one of the model, both or none by default. Moreover, if `includeDiffraction` is set to true, the model is convolved with the diffraction pattern due to the pupil.
- `im_ref` is a reference image that, if given as input, will serve to adjust the astrometry and photometry of `im` over it. If `fitbg` is set to true, the algorithm fits a constant background level as well.

When calling the *psfReconstruction* class (see PUAKO note #03), PUAKO automatically performs a scaling of the reconstructed PSF whose result can be found in the field `p.psfr.psf.im_fit`, which is the one compared with the reference image when running the function *displayResults*. Moreover, retrieved parameters are stored in `p.psfr.psf.catalogs.ref` that is a structure with several fields, such as `x`,`y` and `flux` as the estimated x/y position (in mas) and flux, as well as associated 3-σ precision and Fraction of Variance Unexplained (FVU) [3].

Finally, when requesting the *psfStat* to adjust a Gaussian or a Moffat model, the user will access the retrieved images in fields `MoffatImage` and `GaussianImage`, as well as residual map in `MoffatResidual` and `GaussianResidual`. Estimates and 3-σ precision are stored in respectively `MoffatParam` and `MoffatStd` (change Moffat by Gaussian). For sake of readiness of results, the user can check the field `catalogs.moffat` (or gaussian) to get the estimated position and flux (`x`,`y`, `flux`) with precision, the FWHM (`fwhm_x`, `fwhm_y`) as well as the aspect ratio (`aspectRatio`) and the FVU.
The user is not supposed to instantiate the *psfStats* object by himself as it is called automatically when running PUAKO, but I encourage you to play with it to really capture its full powerfullness.

3 The statisticalAnalysis class

The *statisticalAnalysis* object is an independent class from PUAKO that consists simply in reading fits header of all reconstruction results stored in saving folders that must be given when instantiating the object

```
sa = statisticalAnalysis({path_save1,path_save2})
```

that will read all files in path_save1 and path_save2. The list of paths can be extended to infinite; it is the responsibility of the user to not loading too many data once and save its memory. All the properties that are stored in files header are reported in the PUAKO notes #02. When the *statisticalAnalysis* is instantiated, three options are possible

1. **Versus plot**. Permits to draw any field versus any other by typing

   ```
   sa.displayPlot(fieldx,fieldy,'sortby',[],'fontsize',20,'legendx',fieldx,'legendy',
   fieldy,'xyline',false,'Color','b','Marker',[],'MarkerSize',7,'thresx',[],'threshy',[],
   'scale',linear','legendLoc','northwest','polyFitOrder',0),
   ```

   where

   - `fieldx` and `fieldy` are the fields respectively in abscissa and ordinates axis, such as 'SKYSR' and 'RECSR' to draw the reconstructed Strehl-ratio as function of the measured one for instance. The user can threshold the values to be plotted by filling fields `thresx` and `thresy`. Moreover, the labels that appear on the figure can be changed using `legendx` and `legendy` that are set to the field name by default. The color, type and size of markers can be set with `Color`, `Marker` and `MarkerSize`. Finally, the function does include error bars if they are available and the user can change the scale to 'logx', 'logy' or 'loglog' using the `scale` field.
   - `sortby` is an optional field to distinguish with different colors and markers samples of different categories. For instance , `sortby` can be set to 'SYSFREQ' to analyze the effect of the loop frequency, 'AOMODE' to distinguish LGS and NGS case or 'DATE' for separating results acquired in different nights. The code automatically attribute different color and marker type to each categories. The legend location can be changed using `legendLoc`.
   - if `xyline` is set to true, a straight line $x = y$ will be added in dashed black line on the figure. Moreover, if the user enters a value for `polyFitOrder`, the figure displays polynomial fits in $\text{fieldx}^{\text{polyFitOrder}}$ (polyFitOrder can be negative or fractional) for each sorted category.

   See the PUAKO notes #01, #02, #03 and #04 to see some illustrations obtained with this function.

2. **Histograms plot**. Allows to draw histograms of one or two specific fields by typing

```
sa.displayHistogram(field,'field2',[],'sortby',[],'fontsize',20,'legend1',[],'legend2',
[],'Color','b','thres',[],'legendLoc','northwest','xlab',field),
```

where

- `field` and `field2` (optional) are the fields the figure will show histograms.
- `sortby` is an optional field to distinguish different categories. IF there are four category, the figure will be composed by four sub-plots giving histograms of field values (field2 if given). The color is managed automatically. The user can change the abscissa label with `xlab` as well as the fields name in the legend box with `legend1` and `legend2`.

See the PUAKO note #02 to see some illustrations obtained with this function.

3. **Error breakdown analysis**. Shows pie charts of averaged AO error terms over observing categories (system frequency, AO mode, date,...) using

```
sa.displayAOerrorBreakdown('obj_name',[],'date',[],'resultsof','psfr','sortby',[],
'fontsize',20,'separatedFig',true),
```

where fields `obj_name` and `date` (optional) serve to specify the file name ('n0004' for instance) and the date ('20130801') the user wants to analyze the error budget. If not provided, PUAKO will provide an averaged error breakdown over all files it has read the header. If so, it remains possible to define categories, to be given using `sortby` ('AOMODE' for instance) for which PUAKO will display an error breakdown, in separated figures (default) or in sub-plots by putting `separatedFig` as true. Finally, the user can display the error breakdown calculated using either the forward PSFR (`sortby` is psfr) or the hybrid PSFR (`sortby` is prime).

Some illustration are given in Sect. 6.

4 PSF metrics estimation

4.1 Strehl ratio

The Strehl-ratio is calculated using the function *getStrehRatio* in the *psfTools.m* file and requires to be fed with the image, the pupil model and the PSF sampling in $\lambda/(2D)$ units. The Strehl-ratio is derived as a ratio of OTF as follows

$$\mathrm{SR} = \frac{\iint \tilde{h}(\boldsymbol{\rho}/\lambda)d^2\boldsymbol{\rho}/\lambda}{\iint \tilde{h}_{\mathrm{DL}}(\boldsymbol{\rho}/\lambda)d^2\boldsymbol{\rho}/\lambda}, \tag{1}$$

where $\tilde{h}$ is the image OTF and $\tilde{h}_{\mathrm{DL}}$ the diffraction-limit OTF deduced from the pupil model. The function operates the following steps:

1. **Get the image OTF**. The image OTF is derived from the FFT2 algorithm applied to the image which is preliminary finely recentered.
2. **Get the Diffraction-limit OTF**. If the PSF sampling is larger than 1, the diffraction-limit OTF is calculated from the pupil model and then zero-padded to obtain the same pixel scale in the conjugated plan comparatively to the image. For undersampling, the code calculates the Nyquist-sampled OTF from the

pupil model at a high-resolution to obtain the same field of view in the focal-plane. The high-resolution PSF is then interpolated to get the right pixel scale and the OTF is obtained from the FFT2 of this PSF.

3. **Calculate the Strehl-ratio** as the ratio of the two integrals.

4. **Estimate the error bars**. To estimates those, we must rewrite the SR calculation as a ratio of PSF maxima. I note $\mathcal{M}_{\text{PSF}}$ as the image maximum, $\mathcal{F}_{\text{PSF}}$ as its flux and $\mathcal{M}_{\text{DL}}$ the Diffraction-limit PSF whose energy is unitary. Eq. 1 can be rewritten as follows

$$\text{SR} = \frac{\mathcal{M}_{\text{PSF}}}{\mathcal{F}_{\text{PSF}} \times \mathcal{M}_{\text{DL}}}. \tag{2}$$

Considering a Gaussian read-out noise of variance $\sigma^2_{\text{e-}}$ and a photon noise, the 5-σ uncertainties are deduced from

$$\begin{aligned} \frac{\sigma_{\text{SR}}}{\text{SR}} &= \frac{\sigma_{\mathcal{M}}}{\mathcal{M}_{\text{PSF}}} + \frac{\sigma_{\mathcal{F}}}{\mathcal{F}_{\text{PSF}}} \\ \sigma_{\mathcal{M}} &= 5 \times \sqrt{\sigma^2_{e-} + \mathcal{M}_{\text{PSF}}} \\ \sigma_{\mathcal{F}} &= 5 \times \sqrt{n^2_{\text{otf}}\sigma^2_{e-} + \mathcal{F}_{\text{PSF}}} \end{aligned} \tag{3}$$

where n^2_{otf} is the total number of pixels.

4.2 Full width at half maximum

The FWHM can be estimated using different options when calling the function *getFWHM* in *puakoTools.m*. First of all, the PSF is interpolated to a to factor r higher resolution ($r = 4$ by default).

1. **Cutting method**. This is the straightforward method that consists simply in detecting the pixels for which the intensity becomes half lower than the maximum. This gives the FWHM in x and y axis.

2. **Contour method (default)**. This technique utilizes the Matlab native function *contour* (ellipsoidal fit). Comparatively to the previous method, this one estimates the PSF FWHM along the major and minor axis regardless the PSF orientation.

3. **Gaussian or Moffat fit**. A Gaussian or a Moffat model is adjusted over the image and the FWHM is deduced from the retrieved parameters

$$\begin{aligned} \widehat{I_G}(x) = I_0 \exp(-0.5x^2/\alpha^2) &\implies \text{FWHM}_G = 2 \times \alpha \times \sqrt{(2 \times log(2))} \\ \widehat{I_M}(x) = I_0(1 + x^2/\alpha^2)^{-\beta} &\implies \text{FWHM}_M = 2 \times \alpha \times \sqrt{(2^{1/\beta} - 1)} \end{aligned} \tag{4}$$

If $\Delta\theta$ is the pixel scale, the uncertainty on the FWHM estimates is derived as

$$\sigma^2_{\text{FWHM}} = \frac{(\Delta\theta)^2}{2r^2} \tag{5}$$

4.3 Fraction of variance unexplained

Given an image I and a reference image I_{ref}, the FVU calculation implemented in PUAKO is the following

$$\mathrm{FVU} = \frac{\sum_{i,j}\left(I(i,j) - I_{\mathrm{ref}}(i,j)\right)^2}{\sum_{i,j}\left(I_{\mathrm{ref}}(i,j) - \bar{I}_{\mathrm{ref}}\right)^2}, \tag{6}$$

where $\bar{I}_{\mathrm{ref}}$ is the mean value of I_{ref}. This metrics is interesting to capture how good is the reconstruction overall spatial frequencies and comparatively to other determination techniques.

5 AO error breakdown

Among all the functionalities of PUAKO, one is dedicated to assess the wavefront error breakdown using the *errorBreakdown.m* file by typing

```
wfe = errorBreakDown(class,'display',false)
```

where `class` is a top-level class *telemetry*, *psfReconstruction* or *PRIME*. This function is automatically called in *defineHeaderFromResults* when saving the reconstructed PSF and the wave-front error terms are saved in the fits header. The output `wfe` is a structure containing the Maréchal Strehl-ratio, the Parenti Strehl-ratio [4], the total residual error in nm and all error terms. The residual error is decomposed on several independent error contributors as follows

$$\sigma^2_{\varepsilon} = \sigma^2_{\mathrm{stat}} + \sigma^2_{\perp} + \sigma^2_{\mathrm{alias}} + \sigma^2_{\mathrm{jitter}} + \sigma^2_{\mathrm{ntt}} + \sigma^2_{\mathrm{lag}} + \sigma^2_{\mathrm{n}} + \sigma^2_{\Delta} \tag{7}$$

where

- σ^2_{stat} is the static error defined from the static aberration map δ_{stat} that includes residual NCPA, differential field-dependent static aberrations, static fitting error and additional static modes (see PUAKO note #03 and #04) as follows:

$$\sigma^2_{\mathrm{stat}} = \iint_{\mathcal{P}} \delta_{\mathrm{stat}}(\boldsymbol{r}) d^2\boldsymbol{r}. \tag{8}$$

  Note that if you use PRIME to identify additional static modes, they will be accounted for in this calculation, while they won't if you use forward PSF reconstruction.

- $\sigma^2_{\perp}$ is the DM fitting error caused by the limited number of DM actuators

$$\sigma^2_{\perp} = \log\left(-\frac{\lambda^2}{2\pi}\frac{\iint \tilde{h}_{\mathrm{DL}}(\boldsymbol{\rho}/\lambda)\exp\left(-0.5\mathcal{D}_{\perp}(\boldsymbol{\rho})\right) d^2\boldsymbol{\rho}/\lambda}{\iint \tilde{h}_{\mathrm{DL}}(\boldsymbol{\rho}/\lambda) d^2\boldsymbol{\rho}/\lambda}\right) \tag{9}$$

  Regarding whether you use the PSF reconstruction or PRIME error breakdown, PUAKO will use the r_0 value estimates from the telemetry or the image respectively.

- $\sigma^2_{\mathrm{alias}}$ refers to the aliasing error that propagates through the AO loop. This error is derived using Eq. 9 by substituting $\mathcal{D}_{\perp}$ by $\mathcal{D}_{\mathrm{alias}}$. If one uses the image-assisted error breakdown, PUAKO will take into account the image-based r_0 and g_{al} estimates (see PUAKO note #04).

- σ^2_{tt} is the residual tip-tilt error that is evaluated from

$$\sigma^2_{\mathrm{jitter}} = \sigma^2_{\mathrm{tt}} - \sigma^2_{\mathrm{ntt}}, \tag{10}$$

where σ^2_{tt} is obtained with Eq. 9 by putting $\mathcal{D}_{\text{tt}}$ instead of the fitting phase structure function. The tip-tilt noise error is obtained from

$$\sigma^2_{\text{ntt}} = \int \kappa^{\text{tt}}_{\text{ntf}}(f)df \times \sigma^2_\eta \tag{11}$$

with $\kappa^{\text{tt}}_{\text{ntf}}$ is noise rejection transfer function of the tip-tilt loop and σ^2_η the noise variance estimated from tip-tilt DM commands (see PUAKO note #02). Again, if one utilizes PRIME, the jitter calculation will take into account the g_{tt} estimate (see PUAKO note #04).

- σ^2_{lag} is the bandwidth error including any phase reconstruction error and evaluated from

$$\sigma^2_{\text{lag}} = \sigma^2_{\text{ao}} - \sigma^2_{\text{n}}, \tag{12}$$

where σ^2_{ao} is obtained with Eq. 9 by considering $\mathcal{D}_{\text{ao}}$ in place of the fitting phase structure function. When using the modal gain estimation of PRIME, $\mathcal{D}_{\text{ao}}$ is obtained by multiplying modal gains with corresponding modal phase structure function as discussed in the PUAKO note #04. The noise error σ^2_{n} is obtained from Eq. 11 when using the high-order loop noise transfer function and the noise variance estimated from the high-order DM commands instead.

- σ^2_Δ is the anisoplanatism error calculated as similarly as Eq. 9 with the anisoplanatic structure function $\mathcal{D}_\Delta$ (see PUAKO note #03). This calculation relies on the MASS-DIMM C_n^2 profile in the PSF reconstruction framework. If PRIME applied, the C_n^2 profile is the one adjusted during the model-fitting process (see PUAKO note #04).

Finally, the Maréchal Strehl-ratio at the wavelength λ is obtained using the classical expression

$$\text{SR}_{\text{mar}} = \exp\left(-\frac{4\pi^2\sigma^2_\varepsilon}{\lambda^2}\right) \tag{13}$$

while the Parenti expression calculates firstly the tip-tilt excluded Strehl-ratio

$$\text{SR}_{\text{ho}} = \exp\left(-\frac{4\pi^2(\sigma^2_\varepsilon - \sigma^2_{\text{jitter}})}{\lambda^2}\right) \tag{14}$$

and derives an alternative Strehl-ratio estimates using

$$\text{SR}_{\text{par}} = \frac{\text{SR}_{\text{ho}}}{1 + 4\pi^2\sigma^2_{\text{jitter}}\lambda^2} + \frac{1 - \text{SR}_{\text{ho}}}{1 + (D/r_0)^2}. \tag{15}$$

This latter approximation should be slightly more accurate, especially for large residual wavefront errors.

6 Application to Keck data

I report in Figs. 1, 2 and 3 pie charts presenting the AO error breakdown, either for isolated cases or overall the 304 data sets (78 LGS data acquired on a single night, 226 NGS data distributed over 4 nights). Error terms are calculated using the forward reconstruction as [resented in Sect. 5. This kind of plots allows to identify at a glance what are the critical parameters in the AO performance. For instance, the case 'n0070' has a very large jitter of 284 nm comparatively to the mean value of 123 nm, which is due to the telescope wind shake as presented in the PUAKO note #03. Moreover, in LGS mode, we have a very large jitter of 300 nm that we retrieve on the PSF shape as well. We have also a noise excess comparatively to the NGS case that is almost entirely explained by the noise on the tip-tilt WFS. Finally, NCPA residual were particularly dominant in the error breakdown for unknown reasons so far (discussions on-going).

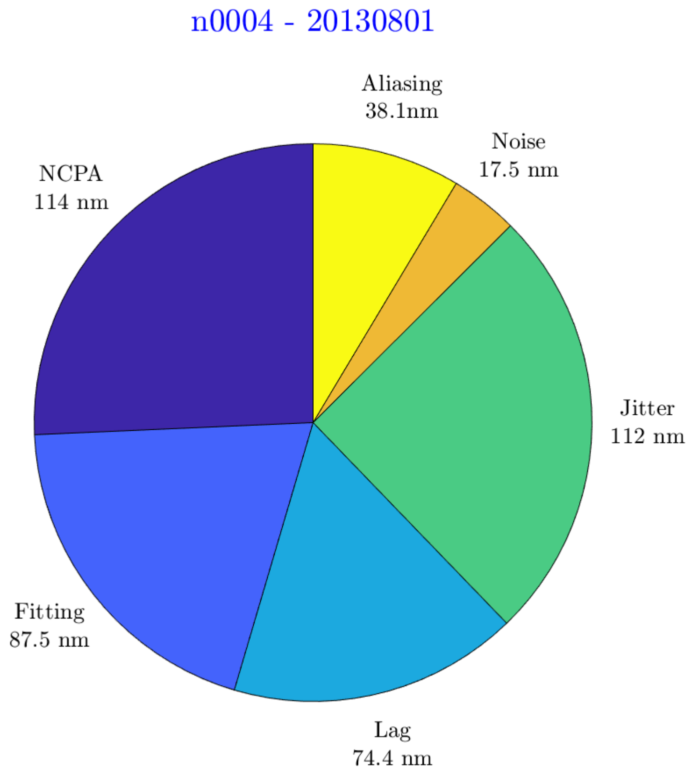
About NCPA, residuals are particularly high-level, up to 180 nm with LGS, while we expect a residual wavefront of rather 50 nm. I don't know whether these high-values are specific to these observing nights or not; however the Strehl-ratio estimation using the Maréchal approximation remains consistent with the image Strehl-ratio, which claims that the total amount of residual wavefront is well assessed.
About the fitting error, the assumption that the AO pattern is a circular region of diameter twice the D cut-off frequency leads to have the approximation $\sigma^2_\perp = 0.26(d/r_0)^{5/3}$, which looks meaningful regarding [5, 2].
About the lag error, I have retrieved mean values reported in Tab 1 that illustrates how the servo-lag errors decreases with respect to the AO loop frequency as expected [1]. Again, numbers look to comply with [5]. Note that these errors include any the phase reconstruction errors as well (modal filtering, non-linear effects,...), that are assessed to fwe tens of nm [5].
About the jitter, we get a residual in mas of

$$\sigma_{\text{jitter}}\text{ (mas)} = \frac{2\times 3600\times 180\times 1e3\times\sigma_{\text{jitter}}\text{ (nm)}}{\pi D}, \tag{16}$$

which is 5 mas in NGS mode and 12 mas in LGS mode for a 10 m telescope, to be compared to 34 mas and 44 mas for the diffraction respectively in H and K band.

Table 1: Servo lag error (including reconstruction error) regarding the AO loop frequency.

Loop frequency (Hz)	41	61	149	438	1054
Lag error (nm)	260	205	175	107	70



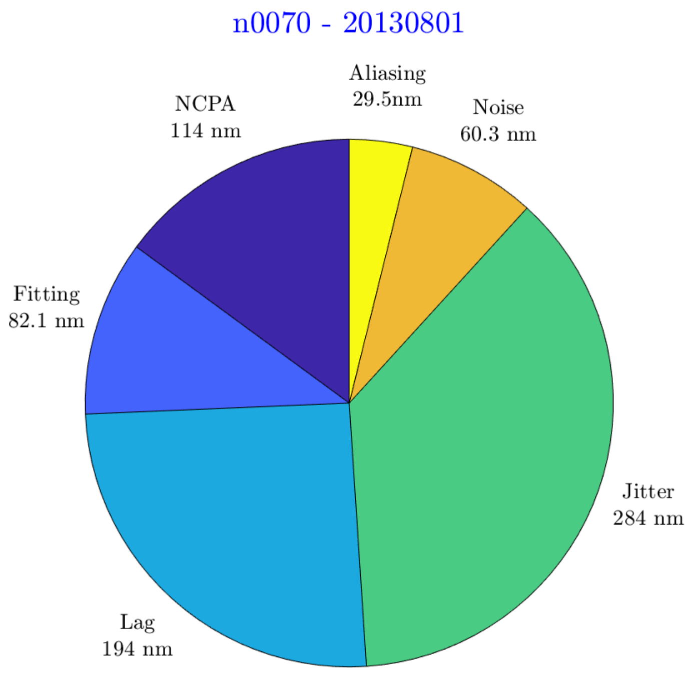


Figure 1: Error breakdown charts for two NGS data sets acquired on August 1st 2013.

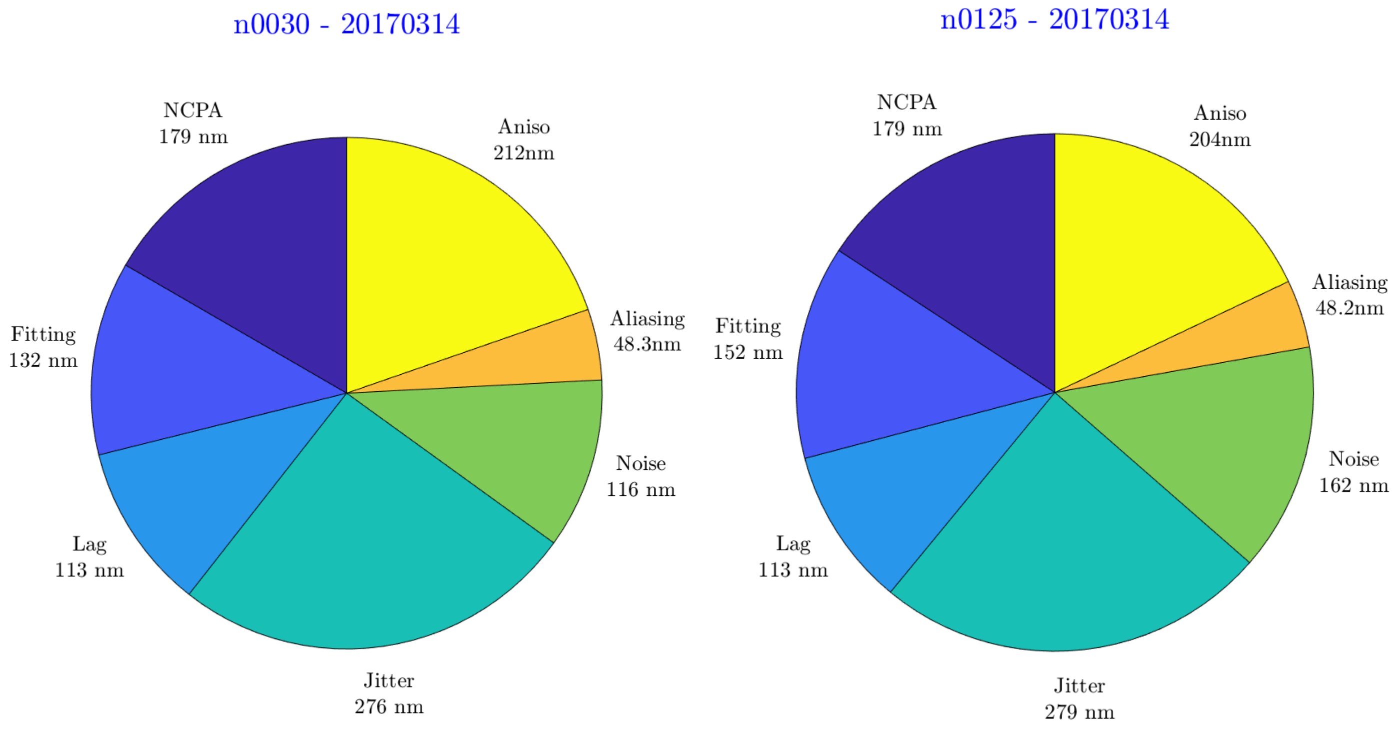


Figure 2: Error breakdown charts for two LGS data sets acquired on August 1st 2013.

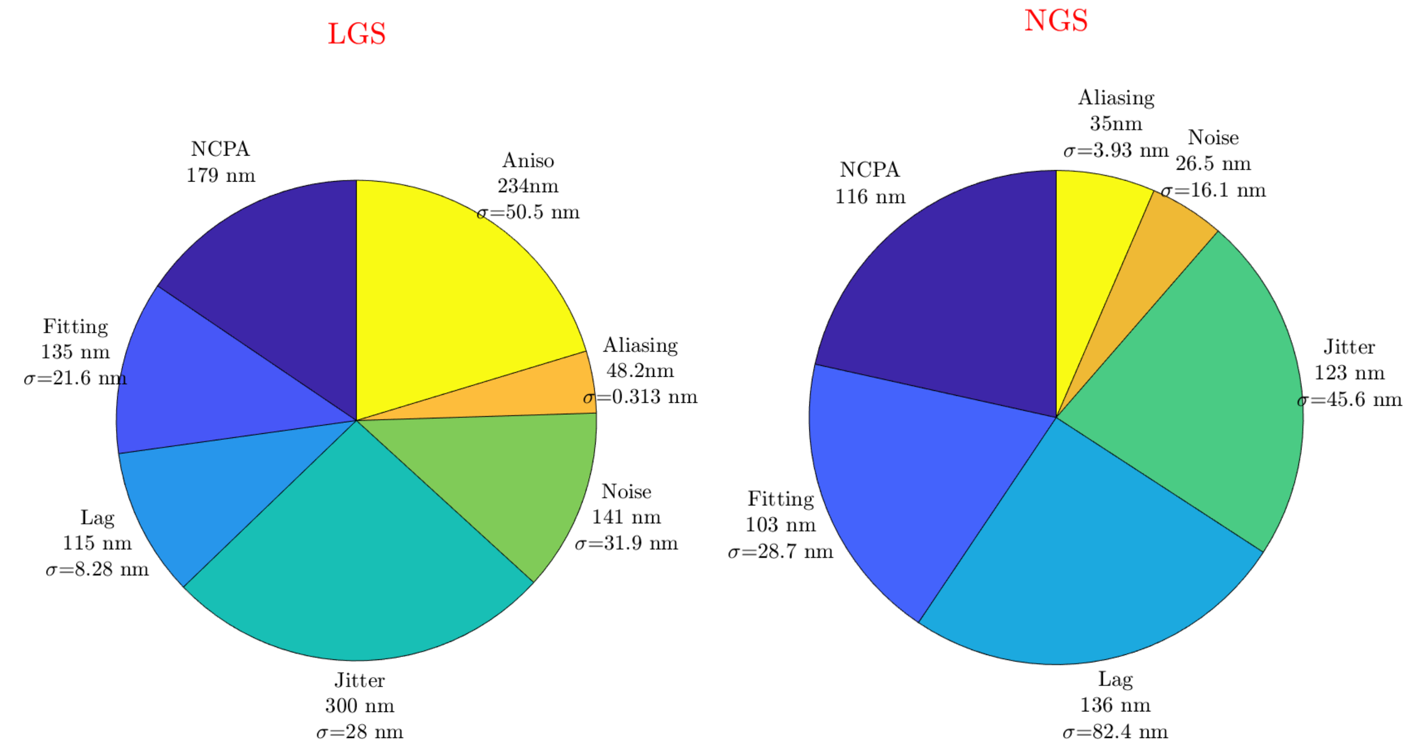


Figure 3: Averaged error breakdown obtained in NGS and LGS mode overall data sets acquired during the four engineering nights (304 data sets). Values of σ indicate the standard-deviation of errors over all error terms estimates.

References

[1] D. P. Greenwood. Mutual coherence function of a wave front corrected by zonal adaptive optics. *Journal of the Optical Society of America (1917-1983)*, 69:549, April 1979.

[2] John W. Hardy. *Adaptive Optics for Astronomical Telescopes*. 1998.

[3] I. R. King. Accuracy of measurement of star images on a pixel array. *Publications of the Astronomical Society of the Pacific* , 95:163–168, February 1983.

[4] R. R. Parenti and R. J. Sasiela. Laser guide-star systems for astronomical applications. *Journal of the Optical Society of America A*, 11:288–309, January 1994.

[5] P. Wizinowich. Progress in laser guide star adaptive optics and lessons learned. In *Adaptive Optics Systems III*, volume 8447 of *Proc. SPIE* , page 84470D, July 2012.