A New Software Tool to Predict Astrometric Errors for ELTs

Trupti Ranka[a], Sireesha Chamarthi[b], Arun Surya[c], Matthias Schöck[d], and Jessica Lu[e]

[a]Giant Magellan Telescope Organization, Pasadena, USA
[b]Indian Institute of Astrophysics, Bangalore, India
[c]University of California, San Diego, USA
[d]Thirty Meter Telescope International Observatory, Pasadena, USA
[e]University of California, Berkeley, USA

ABSTRACT

Extremely Large Telescopes (ELTs) have precision requirements of a few tens of micro-arcsec for differential astrometry science cases. Each ELT project has its own astrometric error budget taking into consideration the specific design parameters of the observatory. A version of the Thirty Meter Telescope (TMT) astrometry error budget has previously been established and the details were presented at SPIE 2016.[1] In this paper, we briefly revisit this error budget analysis. The main focus of this paper is a new python-based astrometry calculator which was developed for a more user-friendly application of the error budget. It facilitates direct evaluation of, and comparison between different scenarios such as absolute vs differential astrometry; dense vs spare observation fields; science fields with and without reference objects, etc. The details of the astrometry calculator and its general functions are described. A few example science sensitivity studies are presented and the procedure of estimating astrometric errors for other observatories is outlined.

Keywords: Astrometry, error budgets, analysis software, ELTs

1. INTRODUCTION

Observations with the existing ground and space-based telescopes have led to number of important discoveries in broad areas of astrometry. A few important discoveries include: a) discovery of a supermassive black hole at the Galactic center using the W. M. Keck Observatory,[2] b) characterising proper motions of dwarf galaxies and kinematic studies of globular clusters using Gaia[3] c) direct imaging of exoplanets using Very Large Telescope (VLT)[4] and Gaia.[5] Existing 8-10 meter class telescopes demonstrated astrometric error capabilities of around 150 micro arcsec with the help of Adaptive Optics. Astrometry with ELTs is expected to improve upon these results and and enable a wide range of new discoveries.

One of the prime objectives of these telescopes is to improve the astrometric precision capabilities. To do so it is essential to estimate the precision to a range of few tens of micro arcsec. Astrometry with ground based telescopes is limited, among other factors, by errors due to the non-uniformity of the Point Spread Function (PSF) across the field, the temporal instability of the PSF and errors induced by the optical distortions and their incomplete correction.[6] These errors are to be constrained to three orders of magnitude less than the diffraction limit. In order to do so, the astrometric error budget is to be evaluated by taking into account error sources such as opto-mechanical errors, focal plane errors, atmospheric refraction errors, residual turbulence after AO correction, along with reference source and catalog errors. The astrometric error is also influenced by the observation field parameters like the science object of interest, the number of field stars, reference stars and science objects in the field of view, wavelength range and integration time.

In this work, we focus on the Thirty Meter Telescope and its astrometric capabilities. The TMT adaptive optics system, NFIRAOS, is designed to enable diffraction limited imaging and integral field spectroscopy at near infrared wavelengths with the IRIS instrument.[7,8] The science cases for IRIS require an astrometric precision of 50 μas in a 100 s exposure for differential astrometry over a 30" field of view in H-band ($\lambda = 1.65 \mu m$).[6]

A detailed astrometry error budget analysis for TMT-IRIS has previously been reported.[1] Using this astrometric error budget as a foundation, we developed a new web-based astrometry calculator that allows future telescope users to predict the astrometric error for their specific science case. While the calculator is currently implemented only for TMT, it is general enough that it could be employed for other ELTs as well. The error budget analysis reported earlier was established as an offline-spreadsheet version [1]. The calculator consists of a python script and a web interface where observation parameters can be fed based on an interactive Graphical User Interface (GUI) wherein the contributing parameters can be varied based on science interests to estimate the achievable precision. The error in the astrometric observations primarily depends on the mode of the observation (absolute or differential astrometry). The observing and the site conditions are also taken into account for estimating the error. Apart from this, the science of interest and the instrument parameter inputs are also evaluated for the contribution to the error. The astrometry calculator is thus divided into two sections for science inputs and the engineering inputs. The web-based astrometric calculator and its inputs and outputs are described in Section 2. Section 3 describes the underlying python implementation and additional functionality not implemented in the web interface. In Section 4, we use the astrometry calculator on a few example science cases. Section 5 describes how the astrometry calculator can be adapted for other telescopes and conclusions are presented in Section 6.

2. THE ASTROMETRY CALCULATOR WEB INTERFACE

Astrometric error budgets are extremely complex and significant effort is required to make them accessible and usable by astronomers preparing observations. We implemented an interactive web platform where users can input various observational scenarios and compare the resulting astrometric error. Using the interactive platform on the web, the user will be able to compare the astrometric error evaluated for various observation scenarios. The calculator allows the user to vary the input parameters and obtain the final estimated error in the astrometric observations. The astrometric error is primarily categorised based on the type science case of interest. The error is fundamentally evaluated as a variance in the estimated position of science object from the nominal position. As an overall view, the science cases are grouped into absolute and differential astrometry. Differential astrometric cases are further classified depending upon the position of the science objects in the field relative to each other or relative to other field objects. To first order, the astrometric error depends on whether the science case requires absolute or differential astrometry.

1. **Absolute Astrometry** The science object position is estimated in the sky coordinate system. In case there are no astrometric reference stars in the science camera field of view, the science target position is measured with respect to the natural guide stars with known sky positions.

2. **Differential astrometry of science objects with respect to each other** This category estimates the variance in position of science objects with respect to each other. Science applications that fall in this classification include the studies of binary stars and exoplanet observations. This analysis requires an estimate of the motion of objects in relation to one another.

3. **Differential astrometry with respect to reference objects** Applications that evaluate the variance in position of science objects in relation to reference stars fall under this group. This classification can be utilised for studies involving the motion of stars around the Galactic centre or jet ejection in star forming region.

The astrometric observations and error analysis are influenced by various factors along with the science field of interest. For errors induced by the telescope and the instrument, the sources of error are divided into 5 categories: 1) reference object catalogue errors, 2) refraction errors 3) atmospheric errors 4) opto-mechanical errors and 5) focal plane errors. The observational terms which are science specific are categorised and displayed in the science input panel. The terms which are related to the telescope and the instrument are displayed in the engineering input panel. The web-based calculator starts with default values for the inputs in the science and engineering panels. The defaults for the engineering input panel variables are already set for TMT with the IRIS instrument. Ideally, for evaluating a science case for TMT, an observer needs to modify only variables in the science input panel. However, the user can also evaluate the sensitivity of the astrometry budget to

telescope/instrument parameters by modifying the terms in the engineering input panel. A list of the factors taken into consideration in the science and the engineering input panel is given below.

2.1 Science Input Panel

The The science input panel is used to fill in the details of the observation. The star field is defined in terms of science objects, reference objects and field objects. The user needs to define the number of these objects and their average signal to noise ratio. The objects are defined as follows.

1. **Science Objects** These are the objects in the field for which the positions and motions are to be determined.[9]

2. **Reference objects** These Objects are objects with known coordinates that are available from other observations (e.g. the Hipparcos or Gaia catalogs, or the masers around the Galactic Center) and that are visible in the science field.[9]

3. **Field Objects** Field objects are all objects that are visible in the field, in addition to but in particular also including science or reference objects.[9]

The Signal to Noise Ratio (SNR) of the field objects also contribute to the error. The visual magnitude of the field objects is used in order to estimate the SNR of the objects utilising the TMT Exposure Time Calculator (ETC). Along with this, the user can vary the distance between the science and the reference stars. The user can vary the average distance amongst the references and average distance amongst the Natural Guide Stars NGS. The scale at which the science object moves is also given as an input parameter. This might be the size of the orbit of a star moving around the Galactic center of a binary star orbit, or the distance over which an object moves in a straight-ish line over the epochs of a set of observations. A screenshot of the GUI of the tool is presented in Fig. 1 . The user can also define the filter used during observation and the integration time for the observation
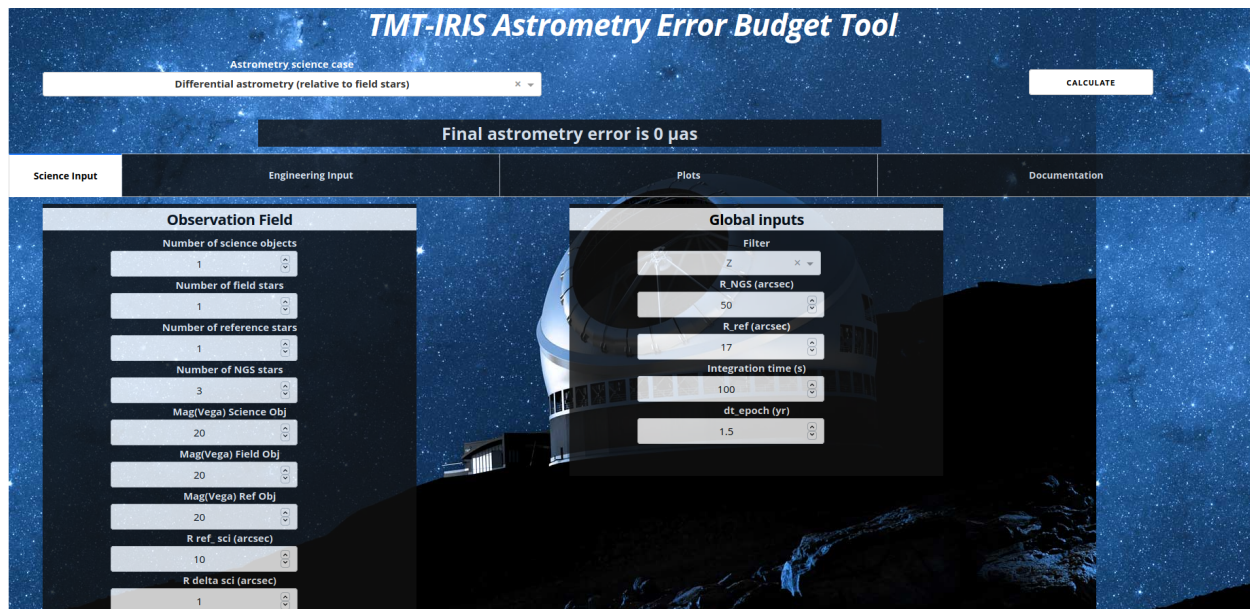
Figure 1. Science input panel of the tool. The science input is broadly classified into two categories: observation field and the global inputs. The user has the option to vary the number of stars (science objects, field stars, reference stars, Natural Guide Stars (NGS)) in the field of view along with their corresponding magnitudes. Apart from this, the user can also vary the distance between the science and the reference objects (R-ref_sci). The R-delta sci term in the field represents the relevant scale of the science object movement. The global inputs section allows the user to vary the filter used, the average separation of the NGSs (R_NGS)) along with the average separation of the reference stars (R_ref). Apart from this, the global inputs also include the integration time (seconds) and the time since the catalog reference epoch.

2.2 Engineering Input Panel

The engineering input panel handles the instrument errors as well as the catalog errors. The GUI of the panel is presented in Fig. 2 .

1. **Opto-mechanical errors** These error are due to optical and mechanical parts of the telescope and the instruments. The terms include quasi static distortions of the structure, optical surface errors, vibration errors, motor rotator errors, positioning errors of the natural guide star probe arm. All of these errors are much larger than the astrometric precision requirement of the telescope, but they can be removed through careful calibration, coordinate transformations and post processing.[6] It is only the uncertainty of our knowledge of the distortions that causes an astrometric error.

2. **Reference Star Catalogue** These errors are due to the uncertainty of the positions of the reference stars. They include uncertainties in colour of the stars, variability of magnitude of the stars, uncertainty in proper motion of the stars and gravitational lensing.[6]

3. **Refraction errors** These errors are due to chromatic and achromatic refraction of the light. Difference in zenith angle within the field cause achromatic refraction. This error scales linearly with zenith angle and can be taken out in post processing. Chromatic errors are due to the wavelength dependence of refraction, some of which is corrected by the atmospheric dispersion corrector. Refraction errors can be a big contribution to the total astrometric error.[6]

4. **Residual turbulence errors** These errors are due to atmospheric turbulence after adaptive optics (AO) correction. These errors cause irregularities in the PSF. The halo effect error occurs in dense fields when the core of one object overlaps the halo of another object.
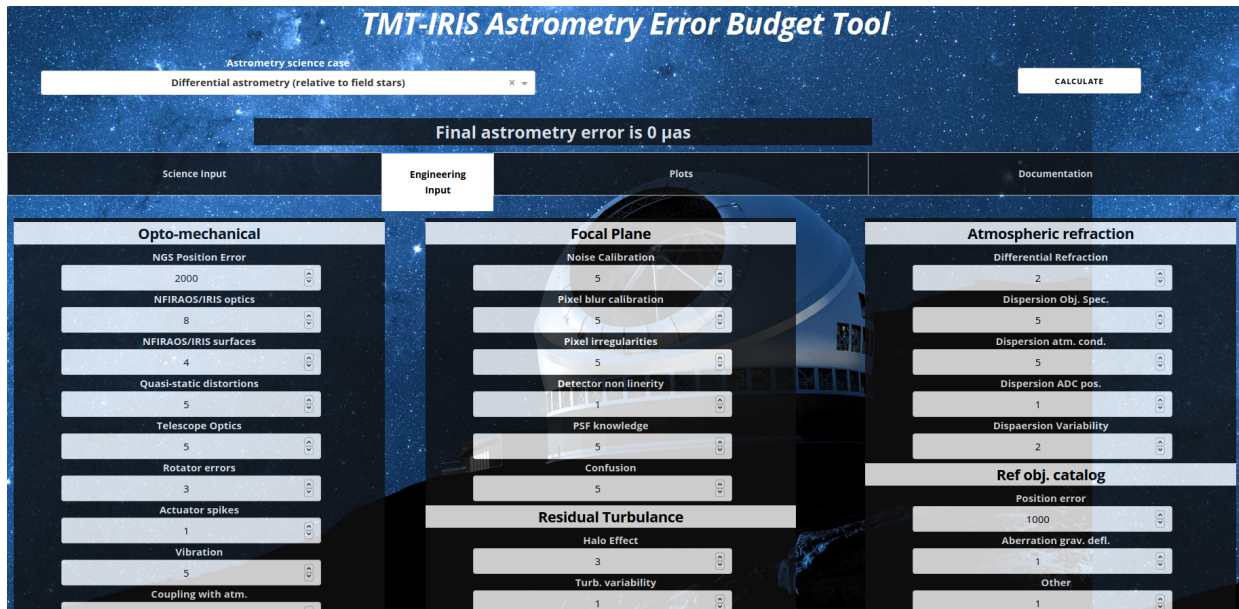
Figure 2. Engineering Input Panel of the tool. The classification includes opto-mechanical, focal plane residual turbulence, atmospheric refraction and reference object catalogue errors. The user can vary any of these parameters for the corresponding science case. The user can choose the astrometry science case from the drop down menu in the top left corner of the GUI, and then vary input parameters from the science and the engineering input fields. The final error estimated is displayed after clicking the calculate button in the top right corner of the window.

5. **Focal plane errors** The errors that fall in this category are photon noise, pixel irregularities, confusion due to unresolved sources in dense fields and detector non-linearities.

3. ERROR BUDGET CALCULATOR IN PYTHON

3.1 Code Structure

The python code is split in two parts. The GUI interface and the astrometry error budget calculator. The GUI is developed using the python Dash libraries.[10] The calculator is made of simple function calls. Python dictionaries are used to set inputs and outputs to the function calls. Intermediate results are also stored in python dictionaries. The GUI sets the inputs for the calculator by modifying the input parameter dictionaries. Depending on the type of astrometry observation selected and the user inputs, the calculator functions calculate the astrometry error. The calculator is self contained. It can be executed separately by invoking python in a terminal. This allows users to automate calculations for large numbers of science cases by wrapping the calculator in additional python scripts. The use of the Dash libraries allows users to modify the GUI easily without having prior javascript experience. Only intermediate python programming experience is required to modify the GUI. 3 shows the interaction between different parts of the code. The observation field inputs are specified in terms of their magnitude w.r.t to Vega. The TMT Exposure Time Calculator (ETC) is used to convert the magnitude and exposure time to the corresponding SNR for the observation field. To reduce the computation time, it is easier if the whole ETC is not invoked for each unique calculation. Hence a data set of SNR as a function of magnitude and exposure time is created for different wavelength filters. The SNR for a given magnitude and exposure time is found by interpolating on this data set. The limitation of this method is that if the interpolation point is far outside the data set, then a new dataset will be needed for more accurate results. The detailed design and implementation of the IRIS ETC is described in Walth et al. The astrometry calculator code is made public in the git repository at the following Github Repo in the footnote *. An instance of using the calculator directly in python interactive shell is shown below

```
from inputs import *
from error_calculator import Error_calculator
error = Error_calculator(global_inputs, field, sigma_sci, sigma_NGS,
  RefObjNCatErr, 'Absolute_Astrometry')
```

The arguments global_inputs, field ,sigma_sci, sigma_NGS, RefObjNCatErr are dictionaries with default values for TMT-IRIS imported from inputs file. They map to the GUI panel and can be modified for a given observation.

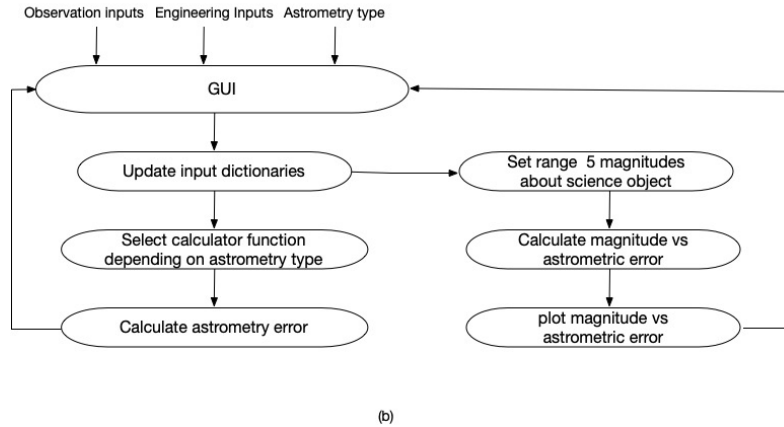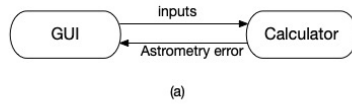* Github Repo: https://github.com/chamarthisireesha/TMTAstrometry

Figure 3.   a) Block diagram of GUI and Astrometry calculator interaction b) Detailed block diagram representing the interaction between different parts of the code.

3.2 Plotting tab

The tool has a plotting tab which plots the change in astrometric error with change in SNR/magnitude of the science object. The plotting tab is tested for various science cases like differential and absolute astrometry. Figure 4 shows the plot for differential astrometry error with respect to reference stars for magnitude of the science object varied between 18 and 21. All the parameters in the calculator apart from the science object magnitude are held at default values.
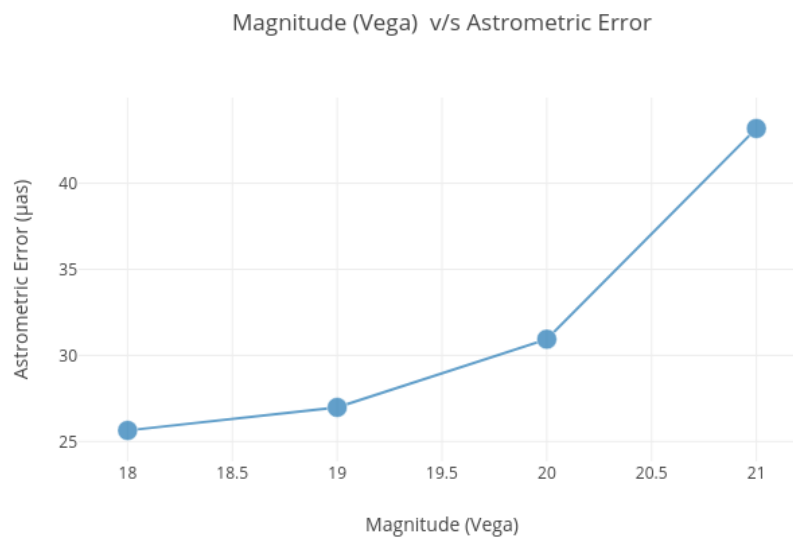

Figure 4.   Plot showing astrometry error Vs magnitude of the science object for differential astrometry with reference stars.

The main calculator is written as a simple python function and can be invoked directly at the command line in the python shell environment. A user can then write scripts to do batch analyses of astrometry for various science cases and sensitivity studies w.r.t. certain input parameters. The code can be found in the git repository. Below is an example of using the calculator directly in a python interactive shell. The output is shown in figure 5. We set the astrometry type to absolute astrometry. The observation is set for Z band and 1 science object. We plot the effect of varying number of reference stars. Initially, the error drops drastically with increase in number if reference stars. After 20 reference stars the error becomes less sensitive to the number of reference stars. Thus, it is possible to analyze the sensitivity of astrometric error as a function of the number of reference stars.

```python
from inputs import *
import numpy as np
from error_calculator import Error_calculator
import matplotlib.pyplot as plt
Nref = np.arange(3, 100, dtype=int)
c=0
Nref.shape
error = np.zeros(Nref.shape)
field['NSci'] = 1
for k in Nref:
        field['Nref'] = k
        output = Error_calculator(global_inputs, field, sigma_sci, sigma_NGS,
        RefObjNCatErr, 'Absolute_Astrometry')
        error[c] = output['Astrometry_error']
        c=c+1
plt.plot(Nref, error)
plt.xlabel('Number_of_reference_stars', fontsize=14)
plt.ylabel('arcseconds', fontsize=14)
plt.show()
```
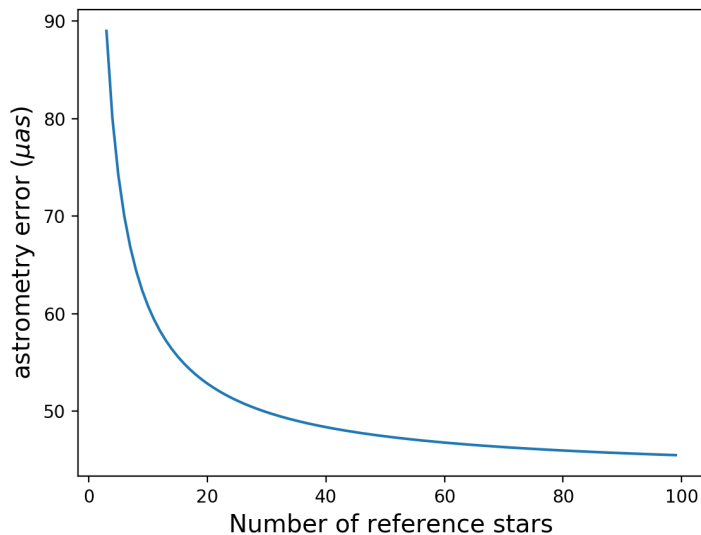

Figure 5. Plot showing astrometry error Vs number of reference stars for absolute astrometry.

Science Inputs	
Field	*Value*
Number of science objects	1
Number of field stars	10000
Number of reference stars	10
Number of Natural Guide Stars used by NFIRAOS	3
SNR of science object	200
SNR of field stars	680
SNR of reference stars	4800
Distance between science object and the reference object (arcsec)	3
Relevant scale of the science object movement (arcsec)	1
Filter used	K
Average separation of Natural Guide Stars (arcsec)	50
Average separation of reference objects (arcsec)	17
Time since catalog reference epoch (year)	1.5
Integration time (seconds)	3600
Engineering Inputs	
All the engineering inputs are the default values used by IRIS	

Table 1. Observational field values used for astrometry error estimation

4. APPLICATIONS OF THE ASTROMETRY CALCULATOR

An example of analysing the astrometry error budget for observations of the Galactic centre with TMT-IRIS is presented. A detailed astrometry budget with simulated images of the Galactic centre is provided in Ref. 11. We demonstrate how the python astrometry calculator can complement such studies which are already underway for all ELTs.

Galactic centre observations of stellar proper motions are an example of differential astrometry with respect to reference stars within the IRIS field of view. We start with the default parameters for TMT-IRIS which are present in the tool. The experimental setup includes an exposure time of 3600 s in a K-band filter ($\lambda \sim 2\mu$m). Based on the IRIS exposure time calculator, this yields a peak-pixel SNR of 4800 at K=16 and 680 at K=20, which is used to set the SNR in the astrometry calculator for the reference stars and science target, respectively. The astrometric error from source confusion is also a strong function of the stellar density. We utilize the scaling between K magnitude and the source confusion term presented in Ref. 11. Table 1 shows the complete set of input variables for the Galactic centre calculation.

The resulting total astrometric precision and the contribution from confusion is shown for stars of different K-band brightness in Figure 6. The astrometry error for brighter science objects ($K \leq 15$) stays at $\approx 55\mu as$. For brighter science objects the astrometry error is dominated by separation between reference stars. For fainter science objects ($K > 15$) confusion error starts to dominate the total astrometry error.
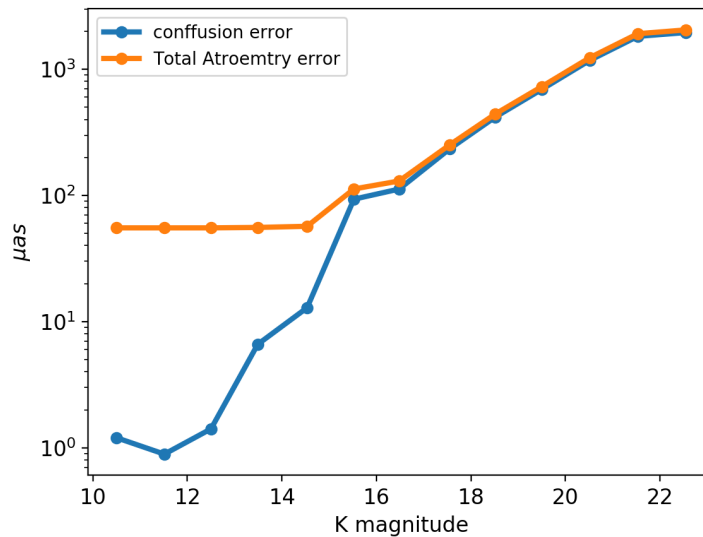
Figure 6. Astrometry error Vs magnitude in the K band for differential astrometry related to field objects

5. ASTROMETRY ANALYSIS FOR OTHER ELTS

Large ground based telescopes typically have similar sources of astrometric error like opto-mechanical distortions, detector related non linearities and imperfections, atmospheric distortions etc. The magnitudes and sensitivity of each astrometric error term depend on the specific design of the telescope and the instrument and the site where it is built. The method of arriving at the astrometric error estimates for other ELTs thus uses the same underlying idea of calculating and scaling variances of different sources depending on the science case and the telescope design.

The python code of the calculator can be modified for use with other ELTs such as the European Extremely Large Telescope (EELT) and the Giant Magellan Telescope (GMT) and their instruments. Detailed astrometry errors for the EELT with the Multi-adaptive optics Imaging Camera for Deep Observations (MICADO) are given by Trippe et al.[12] The paper discusses 10 sources of systematic errors and the intrinsic error due to the SNR of the observation. The approach for the astrometry error budget calculation is analogous to the approach used in the astrometry analysis for TMT-IRIS. The standard deviation of the astrometric error for each term is calculated and then summed up in quadrature. Detailed simulations for various science cases are carried out for these sources of error to come up with standard deviations on the error. It is assumed that the images are post processed, with appropriate coordinate transforms applied. The coordinate transforms take care of spatially linear error terms hence such terms are not counted in the error budget for the EELT-MICADO astrometry error.[12] Table 2 lists a possible mapping of the astrometric errors derived from these simulations onto the terms used in our analysis tool, demonstrating how the tool can be used for an EELT astrometry error budget.

We see that the EELT-MICADO instrument observations can be analysed with our software by sorting errors into appropriate categories. The observation field inputs in the software can be modified to match the EELT site and telescope parameters.

EELT-MICADO Error sources[12]	Equivalent terms in TMT error budget
Detector sampling/binning	Pixel blur, confusion
Geometric distortion	Static opto-mechanical distortions
Plate scale instabilities, instrumental rotations	Dynamic opto-mechanical distortions
Achromatic differential refraction	Achromatic differential refraction
Chromatic differential refraction	Chromatic differential refraction
AO guide star measurement errors	NGS position errors
Atmospheric differential tilt jitter	Diff. TTJ: plate scale, Diff. TTJ: higher order
Anisoplanatism of the MAORY AO	PSF irregularities, Halo effect
Galaxies as astrometric calibrators	Reference object and catalog errors
Sky-projected pixel scale	Reference object and catalog errors
Intrinsic Astrometry error	SNR of science, reference and field objects

Table 2. Astrometry error terms for EELT with MICADO and a possible mapping onto TMT error budget terms

6. CONCLUSION

This paper presents details of a new web-based astrometry error budget calculator. The errors are evaluated taking into consideration various science and instrument factors. The primary goal of the approach is to develop an easy-to-use interactive calculator that can be used to rapidly explore other science cases and sensitivities than those covered in the existing TMT astrometry error budget,[1] and to extend it to other ELTs. It is an open source calculator that can be customised for additional inputs or for other instruments. In its current state, the calculator is set up and auto-configured for TMT astrometry error analyses. In future work, simulations of additional science cases, telescopes and instrumentation can be included, which will results in more general applicability to astrometric error analysis.

ACKNOWLEDGMENTS

REFERENCES

[1] Schöck, M., Andersen, D., Rogers, J., Ellerbroek, B., Chisholm, E., Dunn, J., Herriot, G., Larkin, J., Moore, A., Suzuki, R., et al., "Flowdown of the TMT astrometry error budget (s) to the IRIS design," in [*Ground-based and Airborne Instrumentation for Astronomy VI*], **9908**, 9908AD, International Society for Optics and Photonics (2016).

[2] Ghez, A. M., Klein, B. L., Morris, M., and Becklin, E. E., "High proper-motion stars in the vicinity of sagittarius A*: Evidence for a supermassive black hole at the center of our galaxy," *The Astrophysical Journal* **509**, 678–686 (dec 1998).

[3] GAIA Collaboration, Helmi, A., van Leeuwen, F., and et.al, "GAIA data release 2 - kinematics of globular clusters and dwarf galaxies around the milky way," *A&A* **616**, A12 (2018).

[4] Kasper, M., Amico, P., Pompei, E., Ageorges, N., Apai, D., Argomedo, J., Kornweibel, N., and Lidman, C., "Direct imaging of exoplanets and brown dwarfs with the VLT: NACO pupil-stabilised lyot coronagraphy at 4 μm," *The Messenger* **137**, 8–13 (2009).

[5] Perryman, M., Hartman, J., Bakos, G. Á., and Lindegren, L., "Astrometric exoplanet detection with GAIA," *The Astrophysical Journal* **797**, 14 (nov 2014).

[6] Schöck, M., Do, T., Ellerbroek, B., Herriot, G., Meyer, L., Suzuki, R., Wang, L., and Yelda, S., "Developing performance estimates for high precision astrometry with TMT,"

[7] Larkin, J., Moore, A., Wright, S. A., Wincentsen, J. E., Anderson, D., Chisholm, E., Dekany, R. G., Dunn, J., Ellerbroek, B., Hayano, Y., et al., "The infrared imaging spectrograph (iris) for TMT: Instrument overview 2016," SPIE-The International Society for Optical Engineering (2016).

[8] Herriot, G., Andersen, D., Atwood, J., Boyer, C., Byrnes, P., Caputa, K., Ellerbroek, B., Gilles, L., Hill, A., Ljusic, Z., et al., "NFIRAOS: first facility AO system for the thirty meter telescope," in [*Adaptive Optics Systems IV*], **9148**, 914810, International Society for Optics and Photonics (2014).

[9] Schöck, M., Do, T., Ellerbroek, B., Herriot, G., Meyer, L., Suzuki, R., Wang, L., and Yelda, S., "TMT top down astrometry error budget," **TMT.AOS.TEC.12.039.DRF03** (2014).

[10] Inc., P. T., "Collaborative data science." https://dash.plot.ly/?_ga=2.244913478.2115395488.1584253998-429566853.1584253998 (2015).

[11] Yelda, S., Meyer, L., Ghez, A., and Do, T., "Astrometry in the galactic center with the Thirty Meter Telescope," (10 2013).

[12] Trippe, S., Davies, R., Eisenhauer, F., Schreiber, N. M. F., Fritz, T. K., and Genzel, R., "High-precision astrometry with MICADO at the European Extremely Large Telescope," *Monthly Notices of the Royal Astronomical Society* **402**, 1126–1140 (02 2010).