

# Отчет научного сотрудника лаборатории информатики

Шергина В.С. за 2022г.

10 декабря 2022 г.

Оглавление:

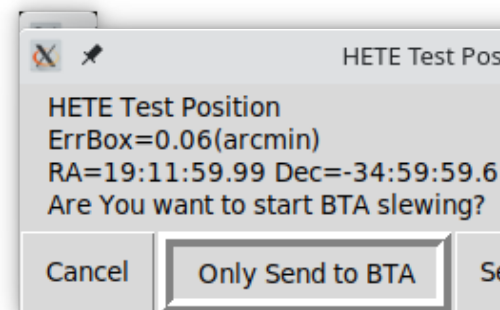
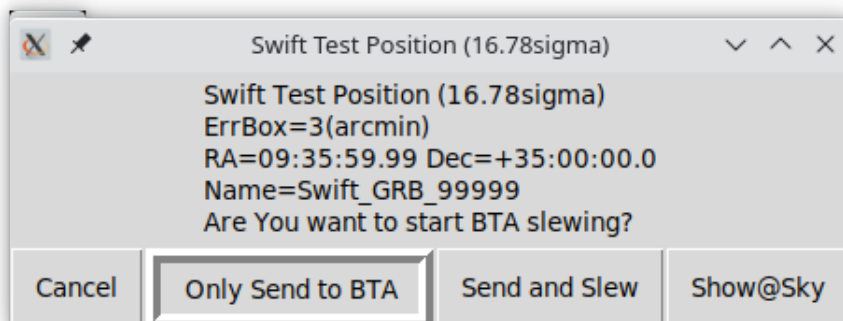
- Модернизация GCN для телескопов CAO.
- Работы по АСУ БТА.
- Работы по TCS Цейсс-1000.
- Подготовка к замене основных ОС.

## Модернизация GCN для телескопов CAO.

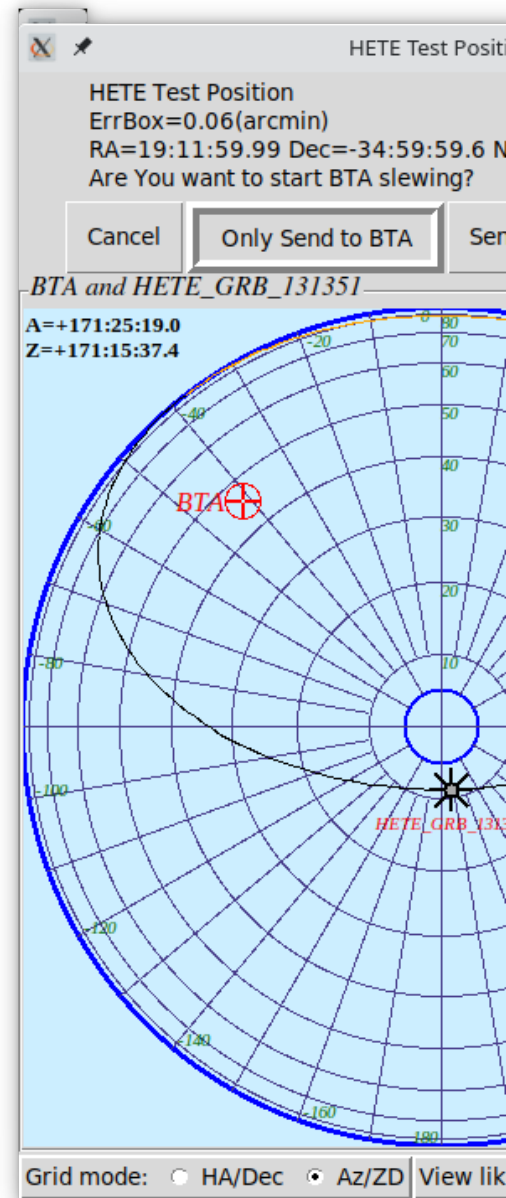
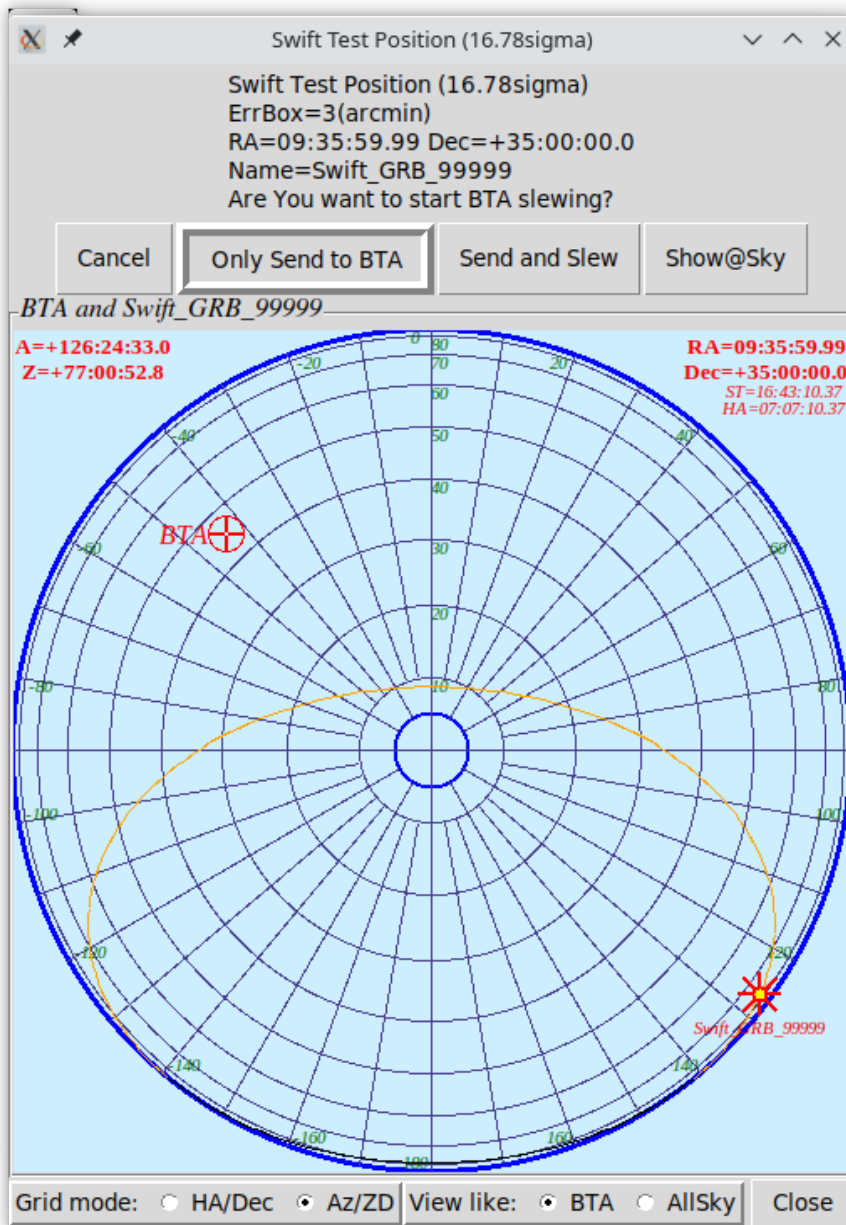
В 2014-м году было разработано клиентское программное обеспечение для работы с системой серверов GCN/TAN. Оно предназначалось для установки на компьютере наблюдателя чтобы оперативно наводить телескопы CAO по координатам GRB-событий. Это МО было написано на Python2 и реально эксплуатировалось на компьютере удалённых наблюдений Цейсс-1000. В прошлом году закончилась официальная поддержка Python2 и теперь основным в различных ОС является Python3.

Для дальнейшей эксплуатации тексты программ были переделаны под Python3 и его библиотеки. При этом было устранено несколько выявленных при эксплуатации недочётов. Например: улучшена работа с тайм-аутами и взаимодействие параллельных thread-ов с Tcl/Tk интерпретатором, а так же теперь приложение не даёт себя «иконифаить» (чтобы всё время оставаться на экране).

Но главные переделки коснулись интерфейса. Появилась дополнительная кнопка «Show@Sky» (т. е. «показать на небе»).



При нажатии на неё появляется схематическое текущее положение источника на небе вместе с текущим положением выбранного телескопа.

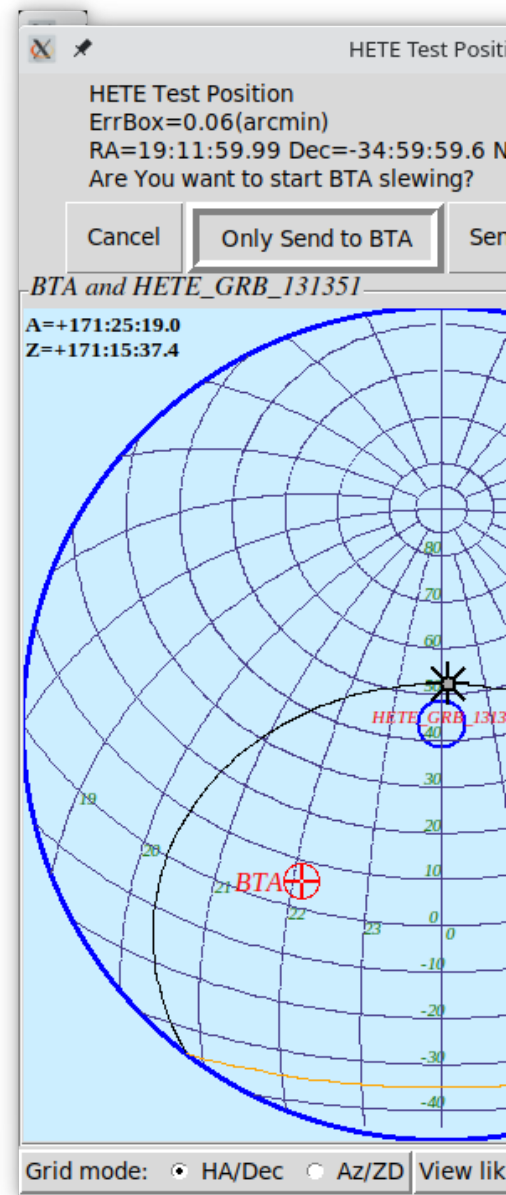
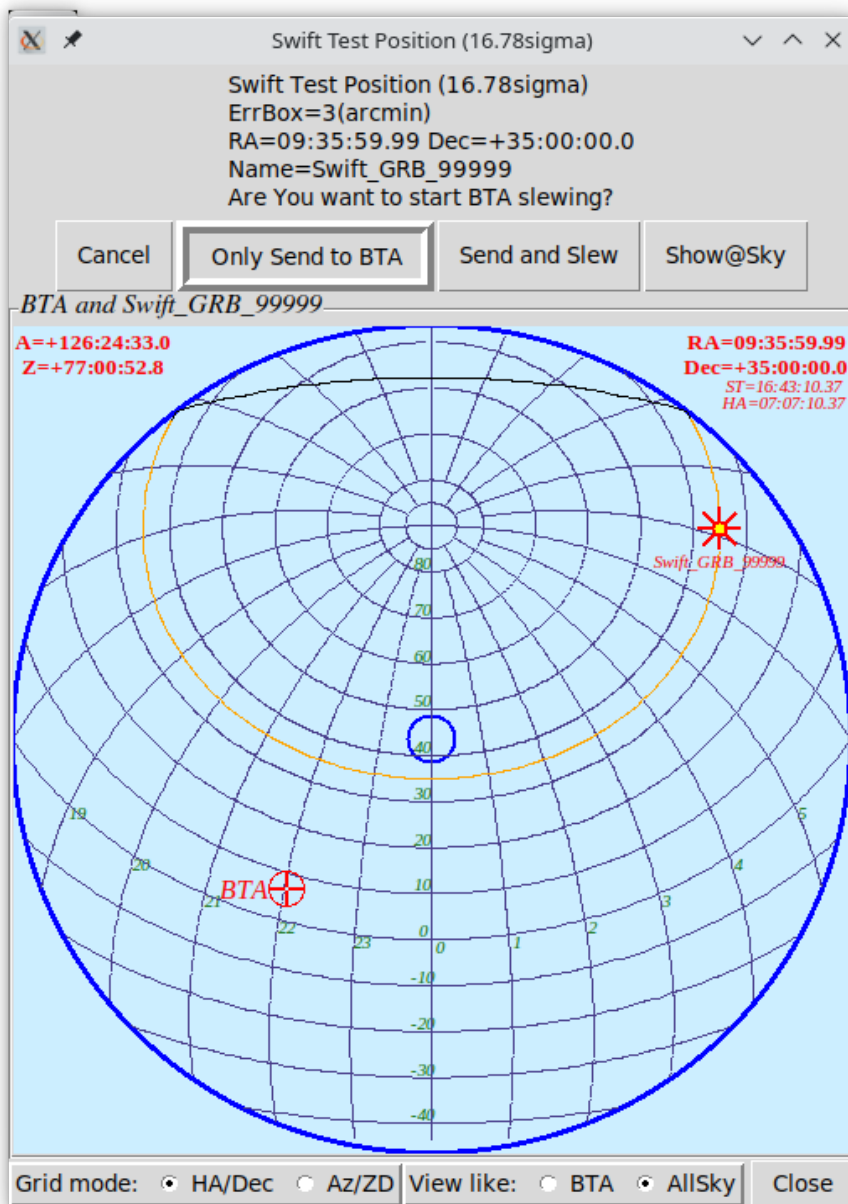


Источник наблюдаемый (над горизонтом).

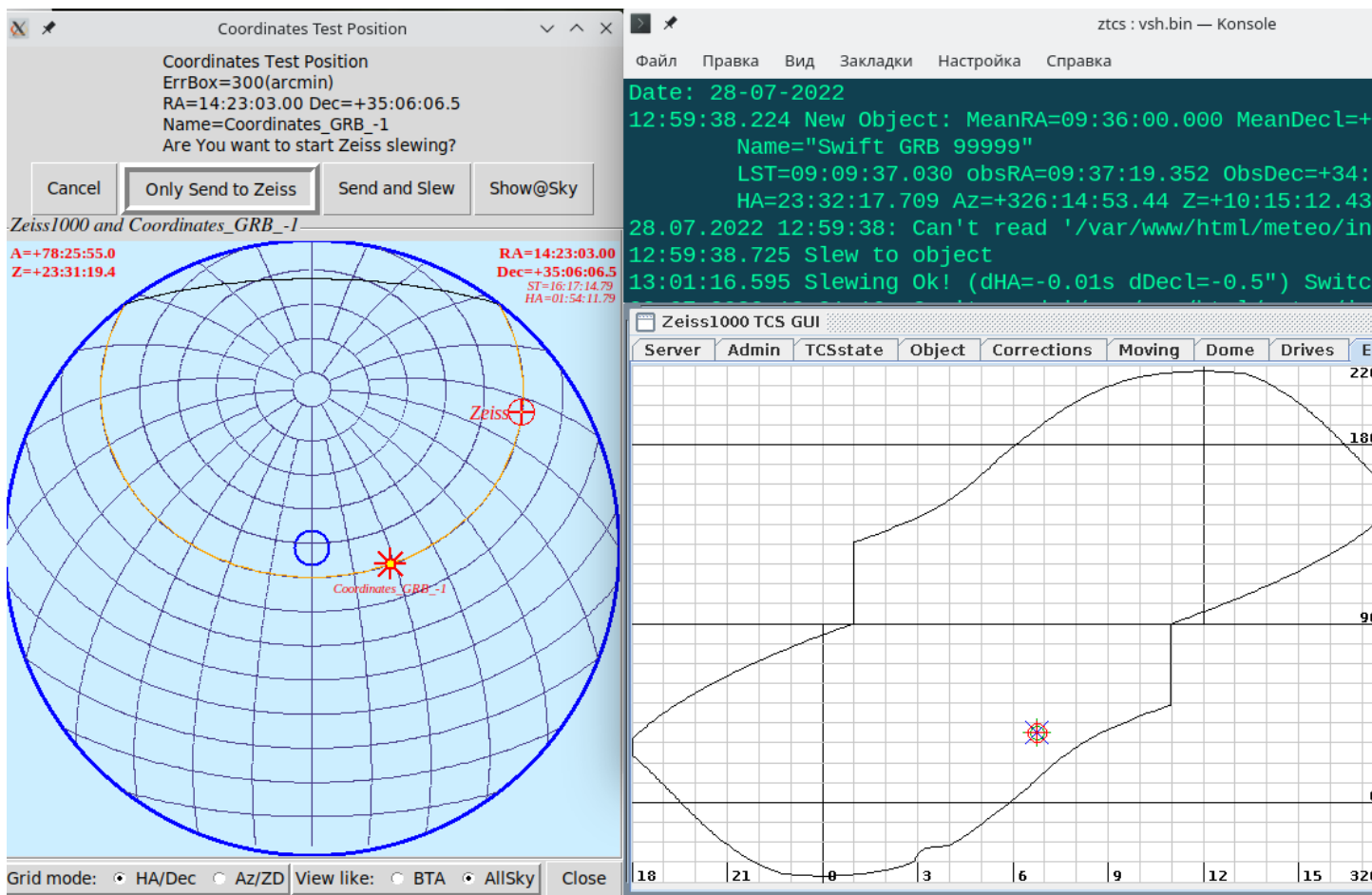
Источник не наблюдаемый (под горизонтом)

Кроме того прочерчена линия по которой будет происходить суточное движение источника.

Ниже расположены кнопки позволяющие выбирать тип сетки: «Азимут/Z» или «Н.А./Decl.», а также стиль представления неба: как принято в интерфейсах АСУ БТА (юг сверху) или как на снимках камер AllSky (юг внизу).



Каждые 5 минут изображение будет обновляться демонстрируя актуальные положения источника и телескопа.



Пример запуска телескопа Цейсс-1000 по координатам тестового источника «Swift GRB»

(на очереди следующий, тоже тестовый - «Coordinates GRB»)

## Работы по АСУ БТА.

В этом году вышел из строя РЕР-контроллер азимута. Ввиду того что автор прошивки контроллера из САО давно уволился, процесс ремонта (С.Синянским и В.Михайловым) оказался довольно сложным и длительным. Для того чтобы БТА не простаивал, Э.Емельянов разрабатывал временное решение на основе микроконтроллера для считывания датчика угла азимута и передачи кодов в систему управления.

Сначала консультировал эту разработку по части имитации CAN-протокола РЕР-контроллера, затем отслеживал последствия этой имитации в системе управления БТА. Работало хуже чем оригинал, но терпимо — БТА наблюдал.

Консультировал Михайлова при восстановлении прошивки РЕР-контроллера из найденных исходных текстов. При попытках вернуть контроллер в азимут (не с первого раза) отслеживал результат в системе управления и давал заключение что не так.

В процессе все этих работ ещё выяснилось что стал хуже работать датчик угла

азимута (не держал нуль-пункт). В запасе имелся только один датчик с другой разрядностью. При окончательной установке РЕР-контроллера датчик заменили.

Внёс правки в главную программу системы управления чтобы значение координаты азимута было правильным. Дополнительное улучшение - что теперь не нужен концевик «-А», который от рождения БТА служил старшим разрядом кода.

В этом году механики, по неосторожности, сломали датчик уровня масла. Система начала непрерывно сигнализировать операторам что бак практически пуст. По просьбе последних внёс правку в их интерфейс *bta\_oper* чтобы отключить эту функцию.

Также операторы жаловались что иногда, при положении телескопа на посадочной площадке в горизонте, дверь стакана ПФ невозможно открыть и приходится снова дёргать телескоп.

Провел анализ архивных записей системы *bta\_trace* который показал что управление приводом **Z** по команде «Стоп» совсем отключается, телескоп идет на выбег и реально останавливается в случайном месте. При дальнейшем разбирательстве выяснил что так сейчас работает схема РК (которая многократно переделывалась за последние годы). При снятии сигнала «Наведение» SEW-привод **A** не отключается и останавливает телескоп нормально (по RAMP), а управление SEW-привода **Z** блокируется.

Разработал и внёс правки в главную программу системы управления, чтобы при останове на посадочной площадке, сигнал «Наведение» снимался с задержкой, давая SEW-приводу **Z** время отработать по RAMP.

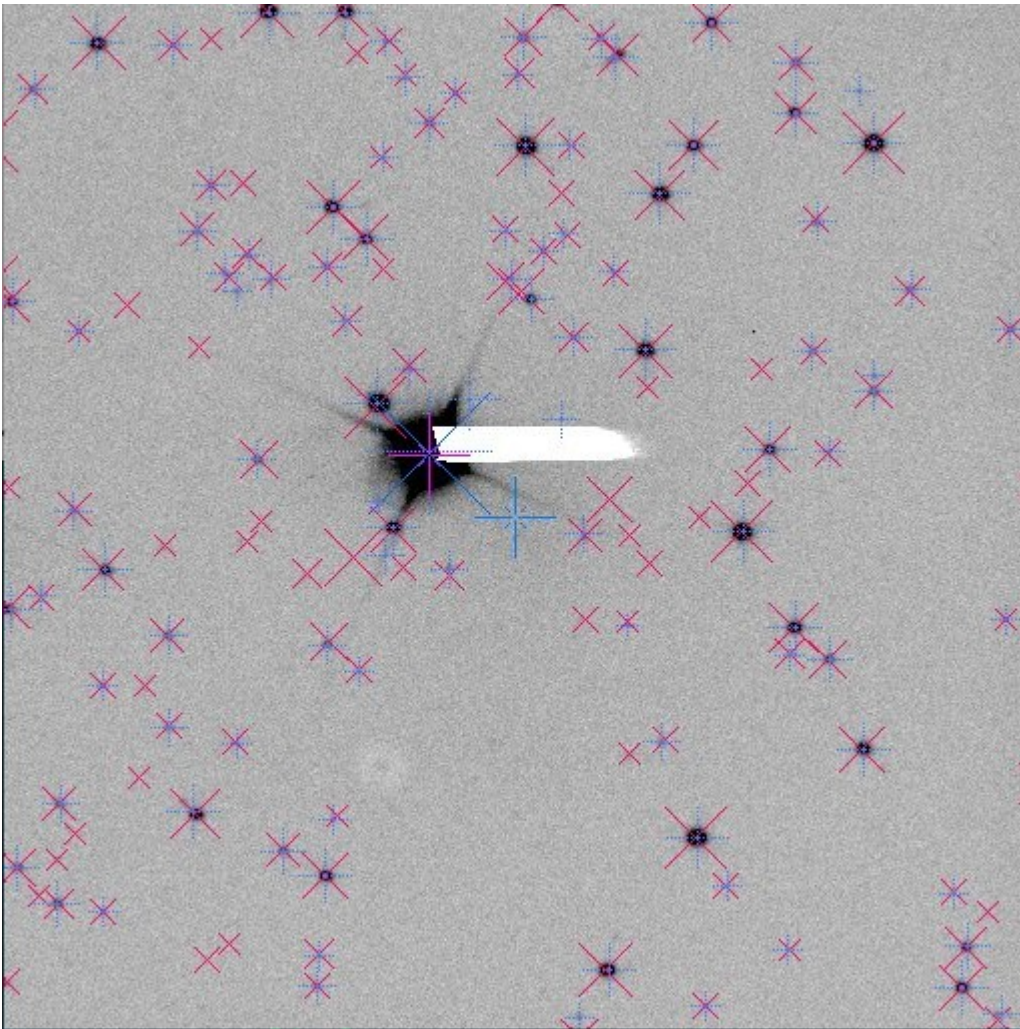
## **Работы по системе управления Цейсс-1000.**

Вносились мелкие правки в пользовательский интерфейс системы управления. (Например: произвольный порядок параметров при вводе нового объекта, изменения в работе с релейными блоками KBX-110 и Муха).

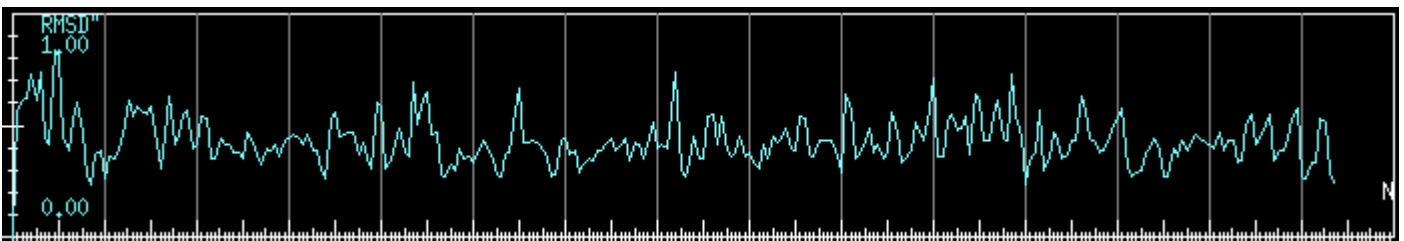
Поправки наведения (Pointing Model) не менялись от запуска в эксплуатацию новой системы управления, т.е. почти десять лет. В октябре сего года Э.Емельянов, проводя приборно-отладочные работы, выделил несколько часов и провёл технические наблюдения. Т.е. снял порядка 145 звезд из Тучо-2 по всему небу, и в прямой моде и с перекладкой. Приёмник с полем ~3.4' (512x512). По два снимка на каждый объект.

Для обработки этого набора из 150 + 142 FITS-файла написаны скрипты на основе, разработанной несколько лет назад, универсальной программы WCS-привязки FITS-изображений *fist\_wcs*. Она позволяет дополнять недостающие FITS-параметры из командной строки.

В результате получена таблица смещений в формате подходящем для программы расчёта методом наименьших квадратов коэффициентов Pointing Model для Цейсс-1000 *zeiss\_lsqm*.

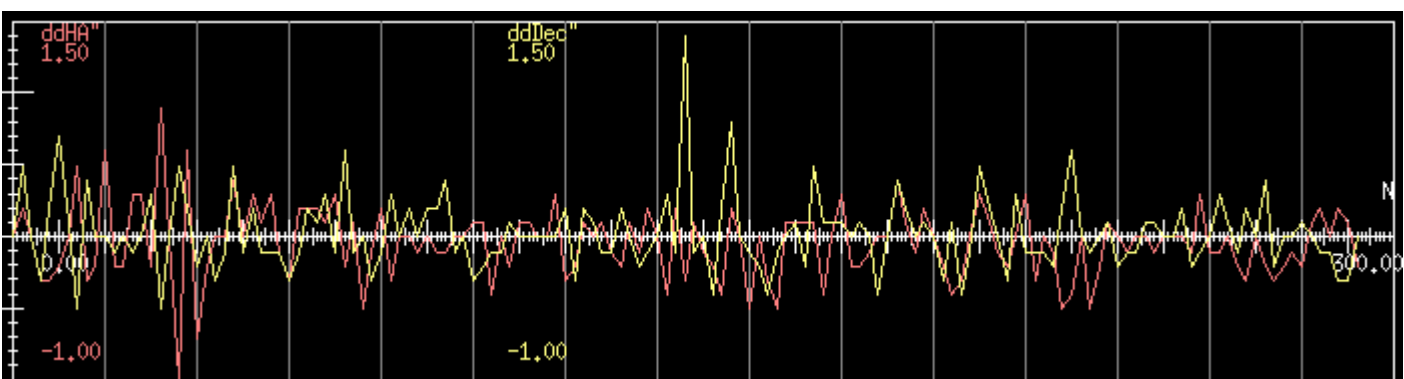


Использование *fist\_wcs* даёт возможность оценить точность полученных данных, поскольку она работает по всем звёздам в поле и дополнительно выдаёт среднеквадратическую ошибку  $\sigma$  (RMSD) соответствия звёзд каталогу.



Видно что точность в пределах 0.3-0.7" (в среднем 0.5").

Поскольку каждый объект снимался два раза через 10s, можно посмотреть ещё одну оценку точности метода — разность между смещениями на двух снимках.



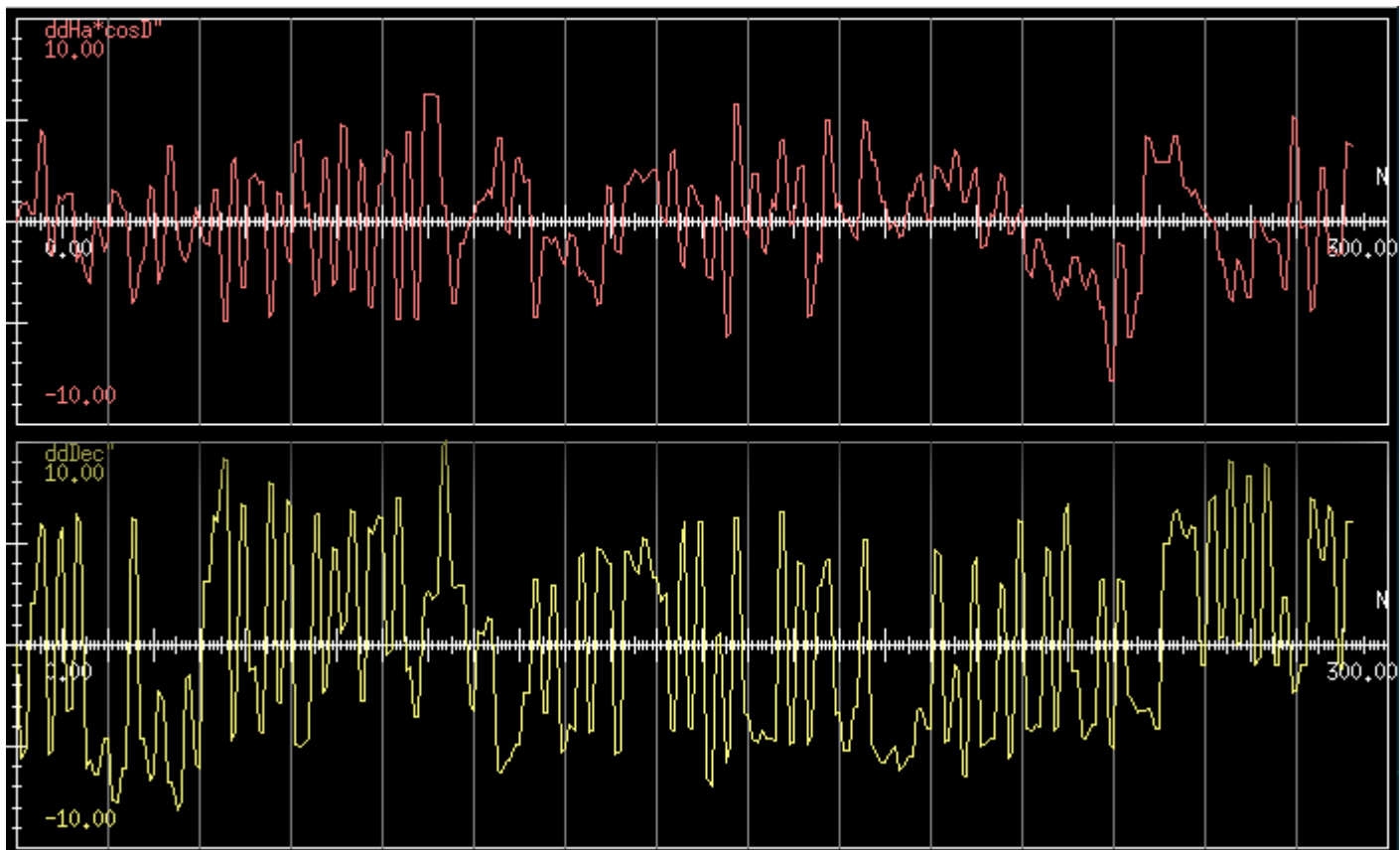
Здесь тоже видно что большая часть укладывается в  $\pm 0.7''$ . А «отскоки» - это качество ведения телескопа за 10s.

Из-за маленького поля приёмника наблюдения производились с включённой Pointing Model, т. е. телескоп отработывал текущие коэффициенты, а смещения в таблице являются дополнением к текущим поправкам. Поэтому был разработан скрипт на основе **awk**, который вычислял по формулам Pointing Model текущие поправки и добавлял их к смещениям в таблице. Полученная таблица (как бы из наблюдений без поправок) обсчитывалась **zeiss\_lsqm**.

Новые коэффициенты	Текущие коэффициенты
IN=-77.36	IN=-73.93
ID=-50.16	ID=-38.7
CH=-105.91	CH=-92.1
NP=7.94	NP=1.25
MA=-4.42	MA=-12.47
ME=-65.85	ME=-66.09
TF=1.44	TF=6.30
DAF=90.24	DAF=94.44
GWH1=28.91	GWH1=31.70
GWH1phi=1.0	GWH1phi=0.0
GWH2=5.21	GWH2=5.35
GWH2phi=4.5	GWH2phi=3.8
GWD1=1.82	GWD1=12.21
GWD1phi=69.0	GWD1phi=65.0
GWD2=-1.27	GWD2=-4.45
GWD2phi=-28.0	GWD2phi=-20.0

При наблюдениях не определялось никакой начальной точки привязки, просто центр используемого приёмника. Поэтому, в коэффициенты CH и ID вошла возможная ошибка установки приёмника относительно прежней точки привязки.

На этих графиках представлены невязки аппроксимации.



Обращает на себя внимание «пила» в полученных ошибках. Внимательный анализ показал что есть заметная корреляция между направлением переезда телескопа и знаком ошибки. Т.е. это очень похоже на **люфт** привода или оптики, по крайней мере по оси Dec1 (около 10"). По оси Н.А. видимо тоже есть, но раза в два меньше. Это же показывают и среднеквадратические ошибки аппроксимации:  $\sigma_{HA}=2.9$   $\sigma_{Dec}=4.99$ . Разумеется это снижает точность получаемых коэффициентов.

## Подготовка к замене основных ОС.

В прошлом году в ЦЕРНе закрылся проект основной ОС используемой на серверах CAO - *Scientific Linux*. Встал вопрос на какой дистрибутив ОС *Linux* ориентироваться в дальнейшем. Например - зарубежный или российский.

Поскольку в начале года мне поменяли рабочую машину на новую с двумя дисками, решил поставить и осваивать параллельно две максимально новые системы. В качестве зарубежного дистрибутива выбрал только что появившийся CentOS Stream 9. В расчёте что за год апдейты приведут его из «сыроватого» в нормальное состояние.

Из российских первоначально поставил РЕД ОС как максимально похожую на *Scientific*. Но оказалось что она, судя по версиям пакетов и компонентов, соответствует RedHat7, т.е. уже слегка устарела (даже за этот год версия РЕД ОС сменилась только с 7.3.1 на 7.3.2). Дистрибутивы Astra Linux и ALT Linux мы не рассматривали изначально как «не RedHat-овские».

Поэтому мне из российских подошла только ROSA Linux, т.к. она с одной стороны основана на пакетной системе rpm/dnf и системное администрирование как в



RedHat, а с другой - у неё есть свободный и постоянно обновляемый дистрибутив ROSA Fresh.

Под обеими системами устанавливал и проверял приложения и сервисы которые использовались в прежних системах.

Далее в каждую из этих новых 64-разрядных систем адаптировал основную часть матобеспечения разработанного за последних пару десятков лет в старых 32-разрядных. Разумеется проверить можно было только то что не требует физических интерфейсов. Например системы управления БТА и Цейсс-1000 только в режиме моделирования.

Не всё получалось успешно. Например в CentOS+KDE неправильно работает Eclipse (т. е. нельзя вести новые разработки на Java), в ROSA не удалось запустить созданные QtCreator клиентские интерфейсы волоконного спектрографа.