Implementation of photometry, automation and remote control in the GROWTH-India telescope

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Atharva Patil

IISER PUNE

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

October, 2017

Supervisor: **Dr. Varun Bhalerao**

Certificate

This is to certify that this dissertation entitled Implementation of photometry, automation and remote control in the GROWTH-India telescopetowards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by **Atharva Patil** at Indian Institute of Science Education and Research under the supervision of **Dr. Varun Bhalerao**, Associate Professor, Department of Physics, IIT Bombay , during the academic year 2017-2018.

Bhalerao

Dr. Varun Bhalerao

Committee:

Dr. Varun Bhalerao

Dr. Prasad Subramanian

This thesis is dedicated to my parents.

Declaration

I hereby declare that the matter embodied in the report entitled Implementation of photometry, automation and remote control in the GROWTH-India telescope are the results of the work carried out by me at the Department of Physics, IIT Bombay, under the supervision of **Dr. Varun Bhalerao** and the same has not been submitted elsewhere for any other degree.

Patil

Atharva Patil

Acknowledgments

This thesis wouldn't have been possible without the support of many individuals.

I would primarily like to thank my supervisor, Dr. Varun Bhalerao, who not only gave me the opportunity to work with him, but provided me with invaluable guidance throughout the whole thesis.

I thank Viraj and Hrishikesh, who always kept me on toes and provided me with a helping hand throughout. I would like to thank Dr. G C Anupama for her guidance at IIA. I am indebted to Tarun Sharma, IIA and Sonam Jorphail, IAO for the assistance they provided me with, concerning the actual hardware. I would also thank the friendly staff at IAO, Hanle for keeping me going in the frigid conditions at Ladakh. I am also thankful to the Department of Science and Technology (DST), India for funding my visit to Hanle. I also thank Sujay Mate and Arvind B for being there whenever small obstacles had to be overcome. I would like to thank my batchmates working in astronomy, Sumeet Kulkarni, Varun Srivastava and Ketan Rikame for their little tips and freshening chats during the tea sessions.

I thank Dr. Prasad Subramanian for being my local advisor at IISER, Pune and providing me with valuable suggestions whenever required. I would also like to thank Dr. Suddhasatta Mahapatra, at IIT-Bombay for providing me with a space in his lab. Finally, the support provided by my friends at IISER-Pune, group members at IIT-Bombay and my family helped me to keep marching forward.

Abstract

The study of electromagnetic counterparts to gravitational waves holds immense potential for breakthroughs in our understanding of short gamma ray bursts, sites of r-process nucleosynthesis, and even cosmology. The first discovery of a gravitational wave source with an electromagnetic counterpart, GW170817, has opened the floodgates to this rich new field. Indian efforts to discover such sources will be led by the upcoming wide-field GROWTH-India telescope: India's first robotic telescope. The aim of this thesis is to completely automate the telescope in order to enable such searches. This includes the development of a robust client-server architecture for telescope remote control, a versatile client for automation, and image processing and archival pipelines at the server.

Contents

Abstract **xi**

1 Introduction **1**

1.1 Electromagnetic followup of Gravitational wave 3

1.2 GROWTH 4

2 Photometry **7**

2.1 Pre-existing code 7

2.2 Modifications made to the code 10

2.3 Transient search 15

2.4 Web interface 15

3 Telescope automation **19**

3.1 RTS2 20

3.2 Telescope components 21

3.3 Development of Filter wheel driver 23

3.4 Modifying the Camera driver 25

3.5 Development of telescope driver 25

3.6 Remote control from CREST 27

3.7 Client-Server model 27

4 Ongoing and future work **29**

Chapter 1

Introduction

On the 14th of September, 2015, mankind discovered the direct existence of gravitational waves for the first time. The signal named 'GW150914' was detected by two LIGO detectors at Hanford and Livingston in USA. The signal has been inferred two have been caused by the merger of two black holes of masses 35 $\mathrm{M}_\odot$ and 30 $\mathrm{M}_\odot$, radiating around 3 $\mathrm{M}_\odot$ of gravitational waves in the form of gravitational waves to form a black hole of hole of mass 62 $\mathrm{M}_\odot$ [6].

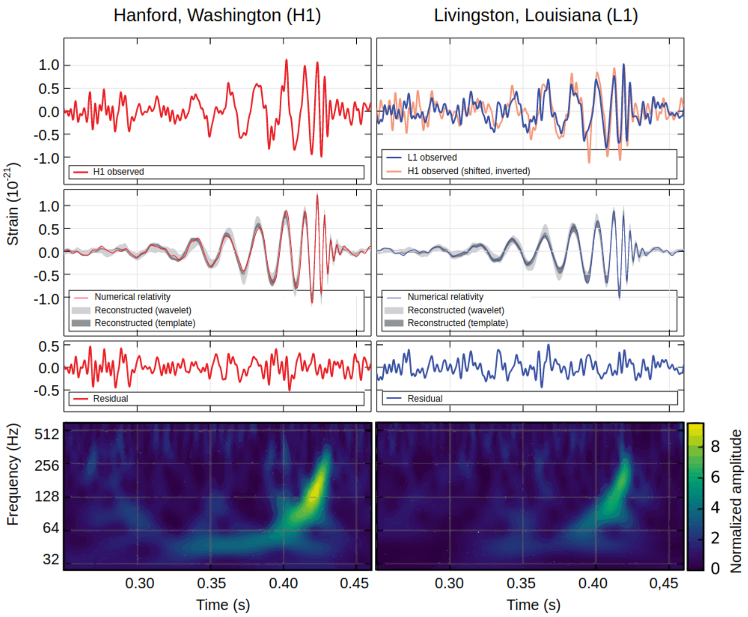


Figure 1.1: GW1508914 signal measured at LIGO, Hanford and LIGO, Livingston

This verification of the existence of gravitational waves further bolstered Einstein's theory of general relativity. This signal was followed by 3 more such signals [10] [5] as a result of black hole mergers. Since the events were results of black hole mergers, no clear electromagnetic counterparts were observed.

However on 17th of August, 2017, the Advanced LIGO and Advanced Virgo detectors detected the very first gravitational waves from binary neutron star (BNS) merger. What followed next was a successful follow up of this signal (GW170817) for a counterpart in the electromagnetic spectrum (EM170817) by a host of observatories all over the world [4]. This EM170817 is an evidence that electromagnetic counterparts exist for gravitational waves from a BNS merger [11].

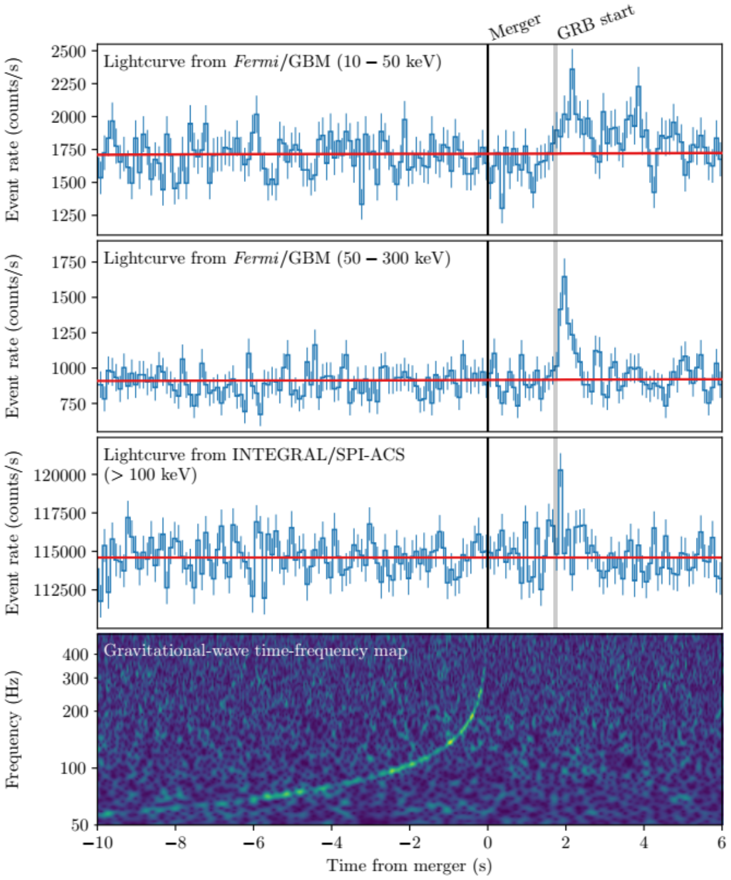


Figure 1.2: GW170817 signal measured at LIGO, Livingston

1.1 Electromagnetic followup of Gravitational wave

The motivation for searching the electromagnetic counterparts is driven by multiple reasons like reduction of the signal to noise ratio for a confident Gravitational wave detection, identification of merger redshifts, thus allowing independent measurements of various cosmological parameters. More importantly, it helps in better analyzing the astrophysical processes involved in the event.

Following up such electromagnetic counterparts can be challenging since localization is difficult using LIGO detectors and transients tend to fade away fast. However, the addition of the VIRGO detector has helped the case and the upcoming LIGO, India project will help in localization further.

Sources of the electromagnetic counterparts of Gravitational waves

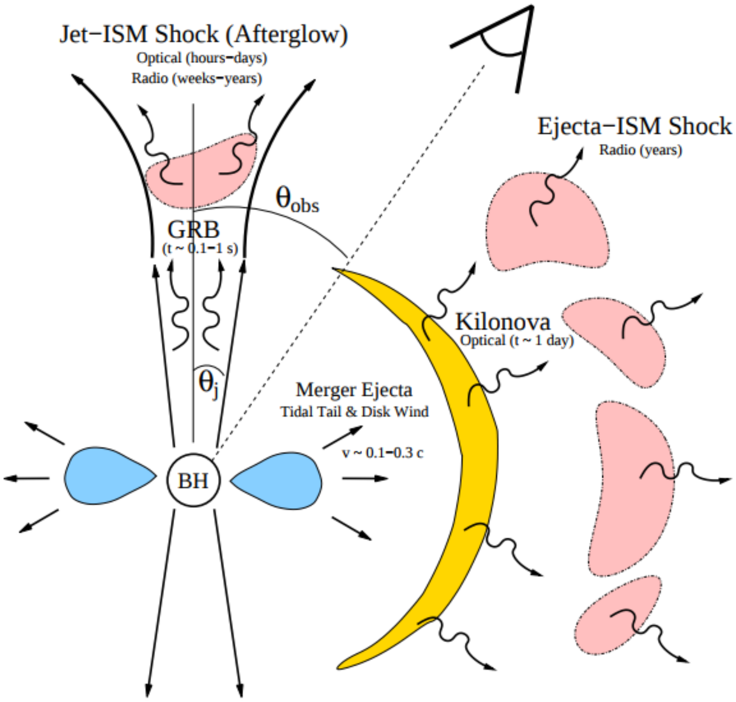


Figure 1.3: Figure depicting the sources of gamma, optical and radio emission from a binary neutron star merger [14]

Following are the most promising sources for the electromagnetic counterparts of gravitational waves (also depicted in figure 1.3:

- Short duration Gamma-ray bursts (SGRB)
- Radio/Optical Afterglow
- Kilonova

1.2 GROWTH

GROWTH-Global Relay of Observatories watching transients happen [1] is an international collaboration studying the physics of fast-changing events in the cosmos like supernovae, neutron stars or black hole mergers, and near-earth asteroids. The goal is to observe and study the transients as fast as possible within the first few days due to the inherently rapid fading of transients. There are currently 18 observatories worldwide 1.4 as a part of this collaboration, including 3 Indian observatories:

- Girawali observatory, Pune (Optical)
- Giant Metrewave Radio Telescope, Pune (Radio)
- Himalayan Chandra Telescope, Ladakh (Optical-Infrared)

The way the collaboration basically works is that as daylight approaches the observatories observing the transients, the task of observation is relayed to other observatories which are still experiencing a night. Hence, the task of observation is relayed to other observatories to maximize the time spent observing the target of interest.

1.2.1 GROWTH-India telescope

The GROWTH-India telescope is an upcoming 0.7m telescope at Hanle, Ladakh. The site is situated at an altitude of 4,500 meters (14,764 feet) and will be connected via a satellite link

[1] http://growth.caltech.edu

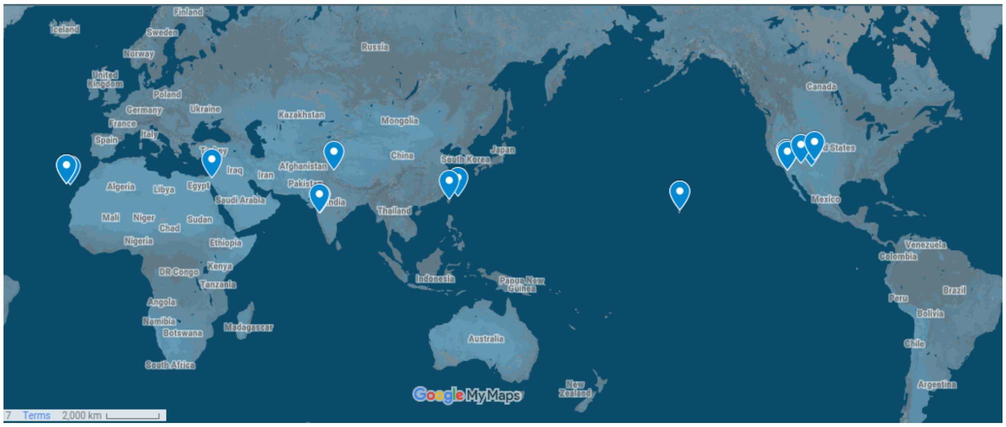


Figure 1.4: Location of GROWTH telescopes worldwide

to the Centre For Research and Education in Science and Technology (CREST), Bangalore which belongs to the Indian Institute of Astrophysics (IIA), Bangalore. The data received at IIA will then be relayed to IIT, Bombay for processing. The telescope will have a 3σ sensitivity of 17 magnitude at 40 Mpc in 'g' and 'r' bands respectively. This makes it capable of detecting events like EM170817 all the way to 200 Mpc, the horizon distance of advanced LIGO.

Chapter 2

Photometry

In order to find the transients in any data, we first have to process the images to extract all stars and measure their brightness in units of 'magnitudes'. The measured magnitude of a star is calculated from the 'counts' in an image, which are proportional to flux, but there is an unknown constant of proportionality. Zero point is defined as the magnitude of a star that gives one data count in the image.

$$m = \mathrm{ZP} - 2.5 \log(\mathrm{counts}) \tag{2.1}$$

A previous semester project of mine calculated zero point for a FITS file. In the first half of the current thesis, I am building on that code to find transients from new data.

2.1 Pre-existing code

While my own code successfully calculated zero points, it was intended to be a learning exercise. For proceeding further, we switched to a publicly available code for zero point calculations. This code had several advantages: it was well developed and tested extensively with real data, had existing interfaces to various databases for getting reference magnitudes of stars, and had a robust method for estimating goodness of fit for zero point calculation.

The output of the code involved the following:

- The zero point of the image
- The scatter in the data
- Fraction of outliers
- Graph of our magnitudes v/s catalog magnitudes
- All the above for each filter of the proprietary databases

It does this using a minimization algorithm described in the very next section 2.1.1.

2.1.1 Minimization algorithm

The basic principle on which the whole code works is maximisation of the likelihood function (or minimization of negative of log likelihood). The offset of observed magnitude from the catalog magnitude gives the zero point for the given image. This difference in magnitudes, i.e. delta magnitude (δ_{mag} from here on), is used to calculate the zero point in the image. Firstly, we assume that the delta magnitude values assume a Gaussian shape distributed as

$$f_x = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2} \tag{2.2}$$

where

$$x = \delta_{\text{mag}} \tag{2.3}$$

$$\mu = \text{median}\,(\delta_{\text{mag}}) \tag{2.4}$$

$$\sigma = \sqrt{(\text{error in distribution})^2 + (\text{error in data})^2} \tag{2.5}$$

The log likelihood values where then minimized using the Nelder Mead algorithm [16], with suitable initial guesses based on the data.

Nelder Mead algorithm explained briefly

The Nelder-Mead algorithm is a downhill simplex method applied to functions to find a local minima or maxima. A simplex is the notion of a triangle generalized to arbitrary dimensions. The simplex used is a ‘n+1’ vertice shape where ‘n’ is the number of variables in the function. For the above case, the simplex used is a tetrahedron, a shape of 4 vertices for the function with 3 variables.

A ‘n+1’ number of guesses is provided for the n-variable function. These form the ’n+1’ vertices of the simplex. This simplex goes ’downhill’ towards a local minima via 3 changes to the simplex:

- Expansion
- Contraction
- Shrinkage

The worst point (‘W’) is replaced by a reflection of the worst point through the centroid of the other points (‘R’). If ‘R’ is better than ‘W’, the simplex has correctly moved closer to a minima. The simplex is then expanded along this direction if the new expanded point (‘E’) is better than ‘R’ (as shown in the top half of the figure 2.1) and the process continues for this new simplex formed with ‘W’ replaced by ‘E’ otherwise ‘W’ is replaced by ‘R’ (as shown in the bottom half of the figure 2.1). If however, ‘R’ is worse off than ‘W’ then a contracted point (‘C’) is calculated closer to the centroid of the simplex. ‘C’ replaces ‘W’ if it is better than ‘W’ (as shown in the top half of the figure 2.2). Finally, if ‘C’ isn’t better than ‘W’, then all the points except the best point (‘B’) are shrunk towards ‘B’ (as shown in the bottom half of the figure 2.2).

Hence the data of catalog magnitudes v/s observed magnitude is fit with a line offset by the zero point value.

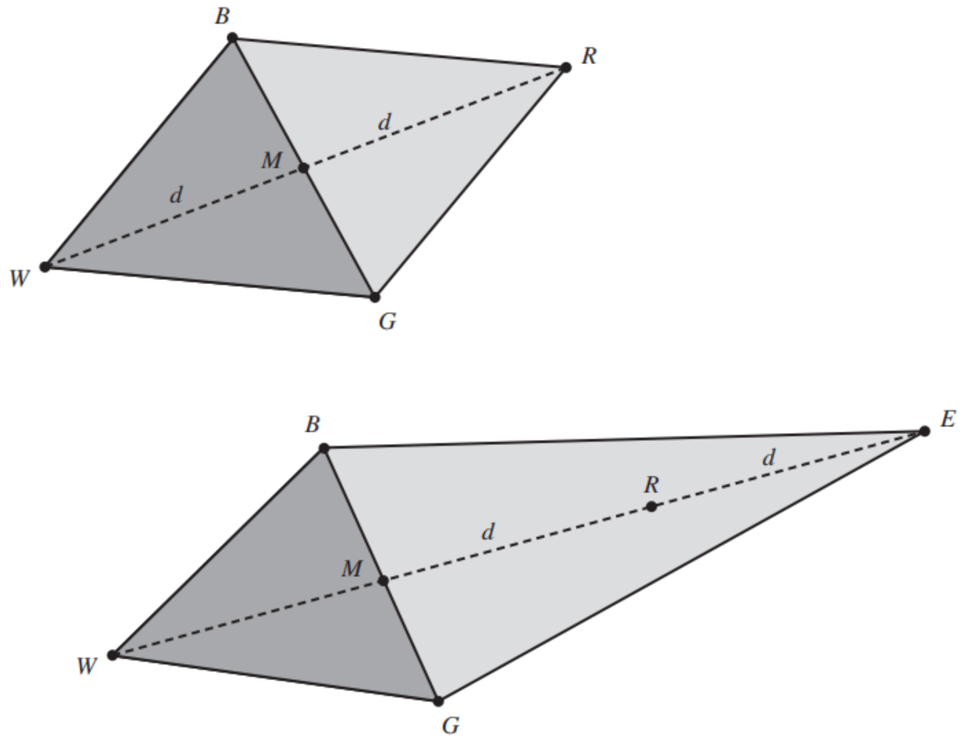


Figure 2.1: Nelder-Mead algorithm with triangle as the simplex. The top figure shows the reflection of the worst point in the simplex and the bottom triangle shows the extended point used if it is better than the reflected point [13]

2.2 Modifications made to the code

2.2.1 Input of FITS

A modification was made such that there's a check whether a FITS image or source table is input into the code. If the plain FITS file is the input, source extraction is performed on the file using sextractor [8] and a table of sources is obtained.

SExtractor explained briefly

SExtractor is a software widely used by the astronomy community to catalog objects in their images. The sources in the image are assumed to be composed of two components, the source

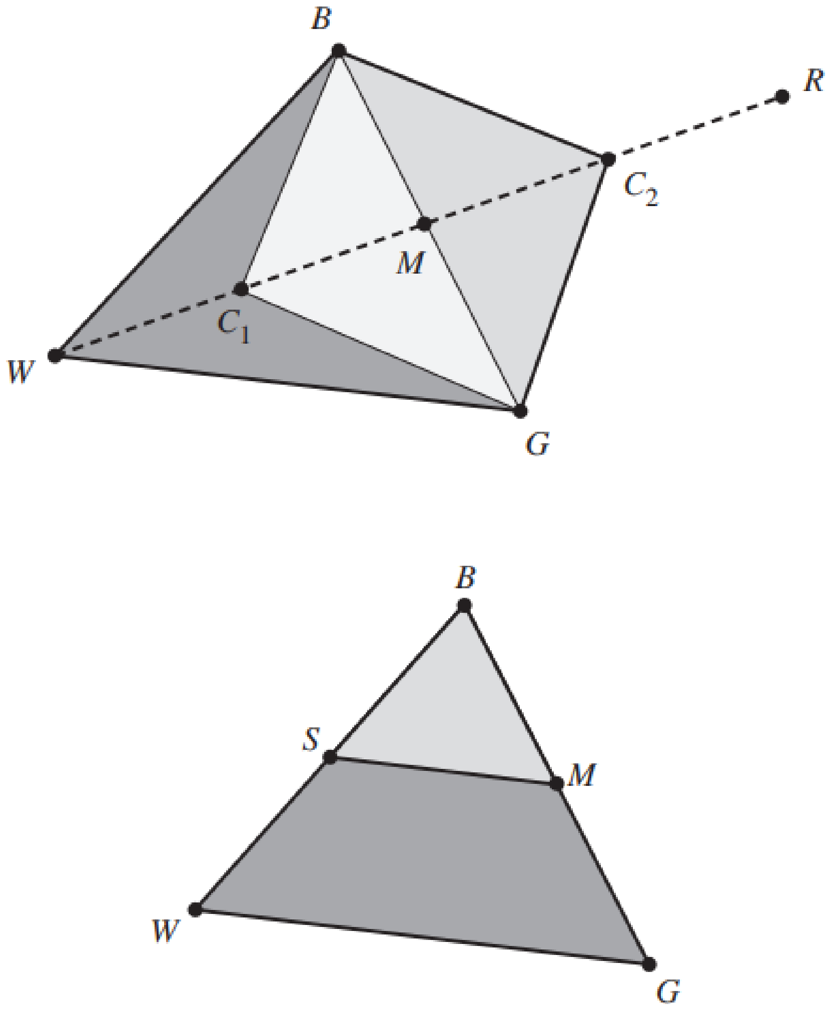


Figure 2.2: Nelder-Mead algorithm with triangle as the simplex. The top figure shows the contraction of the triangle towards the centroid(M). The bottom figure shows the shrinking of the triangle towards the best point (B) [13]

signal, and the background flux. The background flux is estimated first. This is done by recursively sigma-clipping the field and value outside of 3-sigma is taken to be the background value. After background estimation, the detection of a luminous source takes place by either of 2 methods. It happens via finding the peak brightness generally for luminous objects like stars or by thresholding, that is finding the source using a contiguous area of pixels for more extended sources. Figure 2.3 shows the bright sources extracted in such a way. The number of such sources found is extensive and not all of them correspond to actual physical entities. These falsely detected sources are majorly removed via cross-matching described in section 2.2.2.

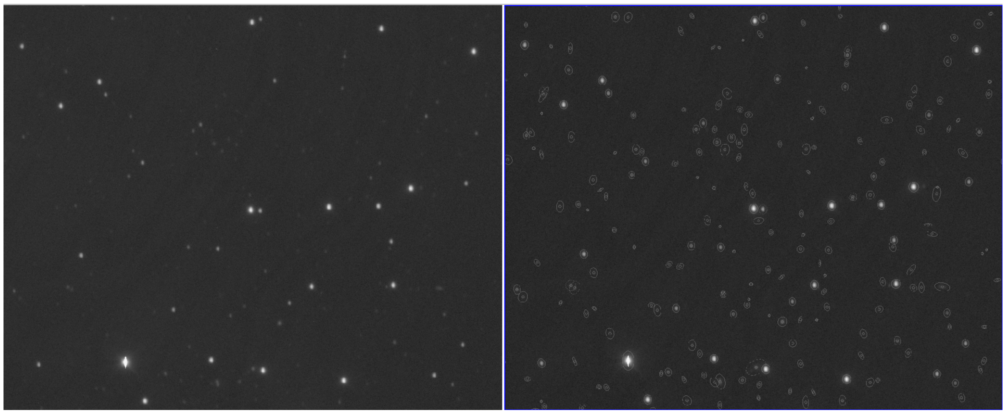


Figure 2.3: The left half of the figure is a sample sky image upon which sextractor is used. The sources which are then found are shown as circles in the right half of the image.

2.2.2 Integration with online databases

There exist standard catalogs with photometry done for a large part of the sky or even the whole sky. We use two such standard catalogs, SDSS [7] and USNO [15]. These catalogs were incorporated into the code with PAN-STARRS DB incorporation being the next. These are the catalogs which are being used to cross-match our observed data with and obtain the catalog magnitudes for the corresponding RA and DEC in our data table.
Multiple ways to cross match were thought of and tried, like:

- download a specific radius of catalog encompassing the image and cross match locally with it
- upload the whole source table, cross-match everything online
- query every single source with the on line db of catalogs (e.g. using services like Vizier)

Multiple libraries like pyvo, astroquery, etc. were also tested for the depending upon the above scenarios. However, based on efficiency and recommendation from the astropy community, for this particular use-case, the web service Vizier was chosen to be used via the astroquery.Xmatch module. This method provided cross-matched catalogs in the order of seconds. This was smaller in comparison to others.

Hence, scatter plots (like figure 2.4) of our magnitude v/s the catalog magnitudes were obtained for each filter of SDSS and USNO catalog

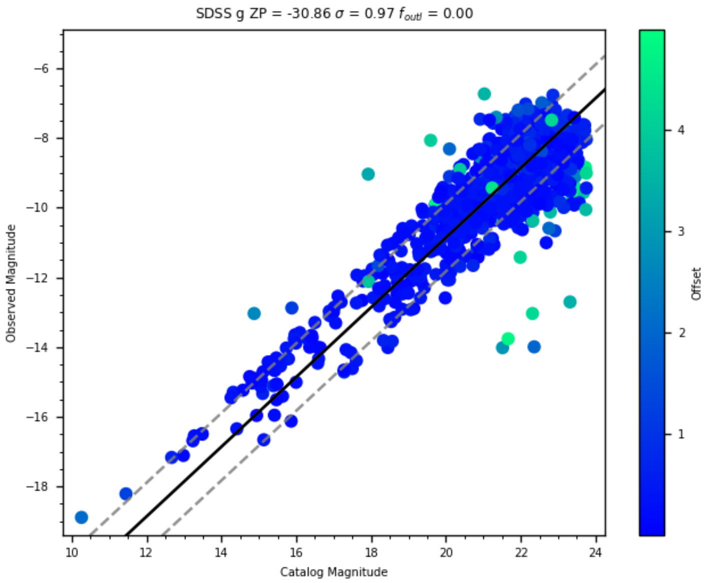


Figure 2.4: Observed magnitude v/s Catalog magnitude for ‘g’ filter of SDSS catalog. The scatter is colour-coded by the offset, i.e., the angular distance between an observed data point and catalog data point in arcseconds. The solid black line represents the best fit obtained via linear regression. The dotted gray line represents 1σ for this fit.

2.2.3 Data filtering

To further improve the results of the zero point calculation 2 cuts were applied:

- offset cuts
- min-max cuts

The offset cuts specified the radius of the data object above which the cross-matched object is not to be considered suitable for the zero point calculation. Typically high offset objects seem to increase the spread of the scatter because they are unlikely to be the same object (or likely to be physically unrelated). Hence the offset cuts are applied. Although stringent cuts could have been applied at the time of cross match itself, it is kept up to the user to define this value later on. This reduced the data so as to exclude the far off catalog objects being matched with the data objects.

The other cuts applied to the data, i.e., the min-max cuts specified the catalog magnitude limits beyond which the data would not be cross-matched. This is because even the star catalogs have magnitude limits. For the calculation of zero point, the cleanest sample space is desired. It has to be least impacted by the statistical and systematic errors of the star catalog and thus the cuts seemed necessary for application.

The impact of these cuts on the uncut data can be seen in the figure 2.5.

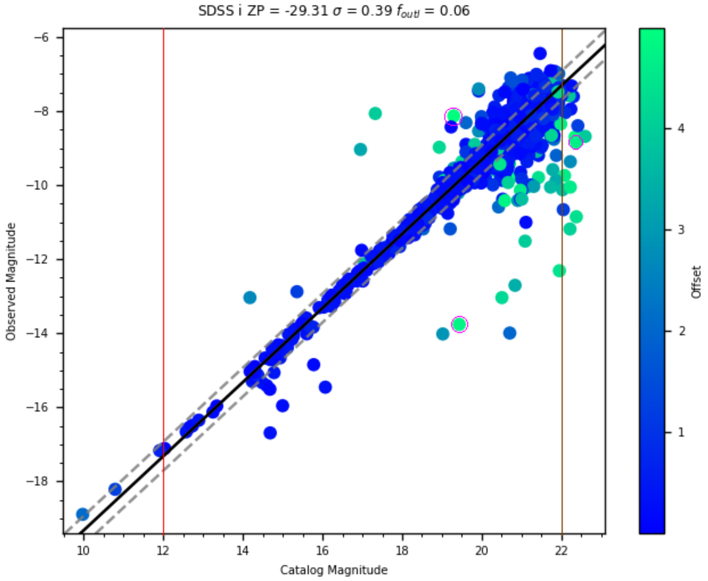


Figure 2.5: Observed magnitude v/s Catalog magnitude for ‘i’ filter of SDSS catalog. The scatter is color-coded by the offset, i.e., the angular distance between an observed data point and catalog data point in arcseconds. The solid black line represents the best fit obtained via linear regression. The dotted gray line represents 1σ for this fit. The red line represents the minimum magnitude cut applied and the blue line represents the maximum magnitude cut applied. The points to the left of the red line and to the right of the blue line would be filtered out. The pink circle represents the points to be filtered out based on their high offsets.

2.3 Transient search

After finding out the zero point for the image, the focus was moved onto finding transients in the image. Two parameters were considered initially:

- Δ_{mag}, the change in the brightness of a star
- statistical significance of the change in brightness

This Δ_{mag} differs from the earlier δ_{mag} used for zero point calculation. This new Δ_{mag} is the deviation of the old δ_{mag} from the zero point calculated earlier.

$$\Delta_{\text{mag}} = \delta_{\text{mag}} - ZP \tag{2.6}$$

It gives an idea about how much the object in our image has brightened up (or dimmed down) in comparison to the corresponding catalog object. The more the difference, the better the chance of that object being a transient.
An alternative parameter instead of the above delta magnitude is the statistical significance($\chi_{\Delta_{\text{mag}}}$) given by

$$\frac{\Delta_{\text{mag}}}{\sigma_{\Delta_{\text{mag}}}} \tag{2.7}$$

The more significant the delta magnitude, the better chance of the object being a real transient rather than a statistical false alarm. These two parameters are then plotted(figure 2.6) against each other to find possible transients. A problem faced here is that the error term ($\sigma_{\Delta_{\text{mag}}}$) in this calculation is being dominated by the zero point error which doesn't help in differentiating between the objects. Hence to improve the error, stricter cuts are being enforced on zero point calculation to make a cleaner subset of data.

2.4 Web interface

I supervised two undergraduate students from IIT-Bombay, Viraj and Hrishikesh, to build a graphical user interface. Based on experience with other transient detection projects, a web portal was selected as the right GUI. The coding was done in Python, Flask [2], HTML. The web portal as seen in the figure 2.7 here currently takes as input the FITS table of sources

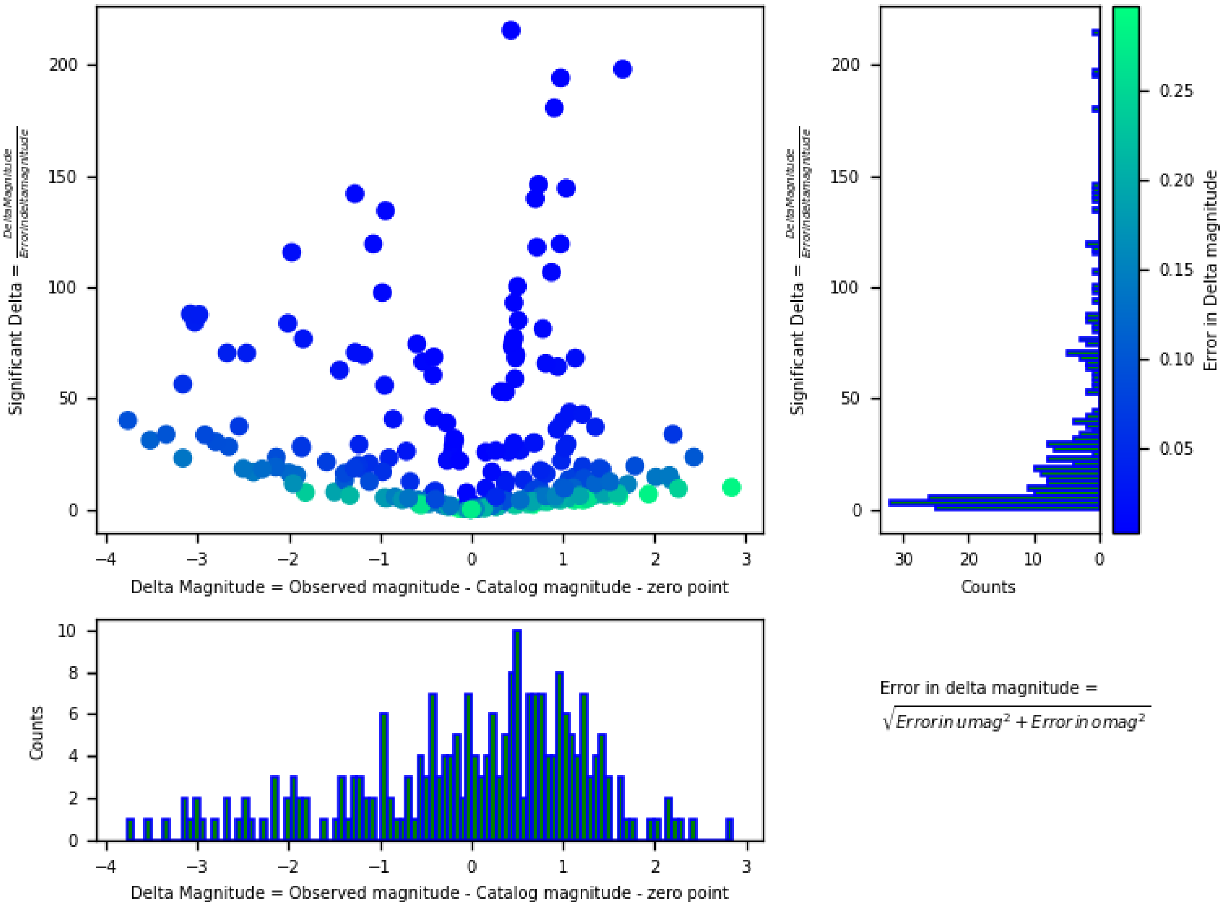
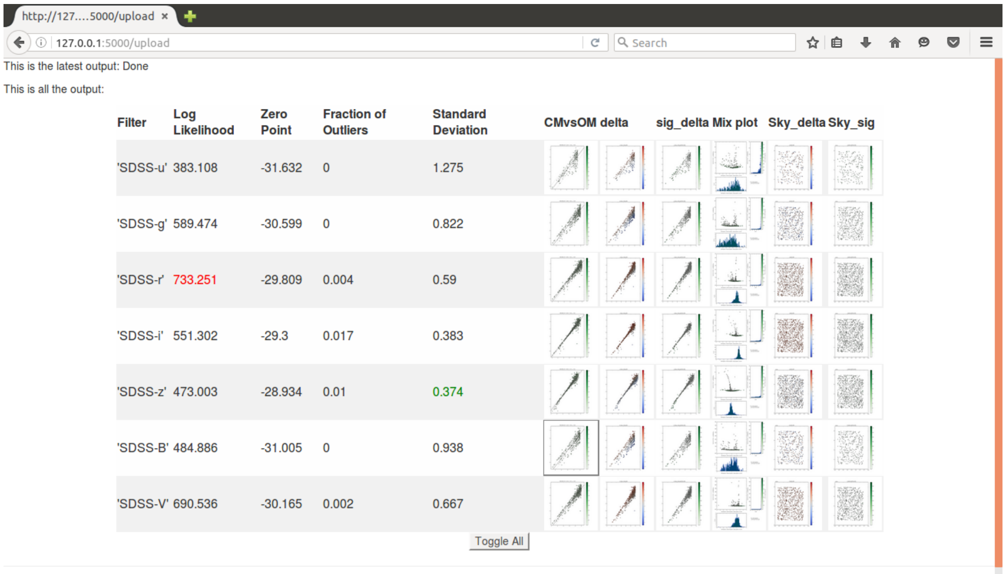


Figure 2.6: The scatter plot displays Δ_{mag} vs $\chi_{\Delta_{\mathrm{mag}}}$. The histogram on the top right gives a distribution of $\chi_{\Delta_{\mathrm{mag}}}$ while the histogram on the bottom left gives a distribution of Δ_{mag}

for the tested data, performs the linear regression to find out the zero point and outputs most of the detailed graphs along with the values.



This is the latest output: Done

This is all the output:

Filter	Log Likelihood	Zero Point	Fraction of Outliers	Standard Deviation	CMvsOM	delta	sig_delta	Mix plot	Sky_delta	Sky_sig
'SDSS-u'	383.108	-31.632	0	1.275						
'SDSS-g'	589.474	-30.599	0	0.822						
'SDSS-r'	733.251	-29.809	0.004	0.59						
'SDSS-i'	551.302	-29.3	0.017	0.383						
'SDSS-z'	473.003	-28.934	0.01	0.374						
'SDSS-B'	484.886	-31.005	0	0.938						
'SDSS-V'	690.536	-30.165	0.002	0.667						

Figure 2.7: Screenshot of the web interface

Chapter 3

Telescope automation

Need for automation

Most telescopes are built at remote sites. The GROWTH-India telescope is being set up in the frigid. The concerned officials may be required to intervene at inconvenient times upon receiving the triggers. Some objects need to be monitored regularly, and these objects may be present in large number. This is time consuming, inefficient and may also lead to human errors. An automated remote telescope is one that not only performs scheduled tasks but is also able to perform certain tasks remotely on request; without human intervention. Such a telescope will help to solve the above problems.

Historical perspective of automation

The first attempts to robotize telescopes were developed by astronomers after electromechanical interfaces to computers became common at observatories. Various automated instrument sequencers were employed on solar expeditions in the late 1800s. In the 1960s, a major robotic telescope was developed by Art Code et. al. at Wisconsin. At the same time, a Boller and Chivens telescope was automated at Kitt peak national observatory; but it was found to be unreliable. Later, more telescopes have been successfully automated worldwide [9].

3.1 RTS2

3.1.1 Observatory control software

A typical observatory generally has different software for controlling the different devices in an observatory, like Dome control software for controlling the dome (eg: maxdome2), Telescope mount control software (eg:PWI3), Camera control software (eg: MaximDL), etc. Such observatory control softwares might be commercial (and mostly close sourced) or free to use [12].

3.1.2 Choice of RTS2

RTS2 (Remote telescope control software, 2nd version) is an open source, observatory control software designed majorly in C++ for Linux based operating systems, developed by Petr Kubanek. Reasons for selecting RTS2:

- Open source in nature:
  Thus easier to 'mould' the software as per our needs.
- Superior target scheduling
- One standardized software to control multiple telescopes, thus making it scalable in the future.
- Already implemented in various observatories spanning 6 continents across the world
- Command line controllable:
  Making it easier to operate via SSH on the low bandwidth satellite link between IAO and IIA.
- Drivers pre-built for a wide variety of devices
- Previous experience in RTS2

3.2 Telescope components

3.2.1 Telescope mount

The mount being used is a PlaneWave Instruments' 0.7m Alt-Az with an F ratio: F/6.5 and 10 arcsecond RMS pointing accuracy [3].

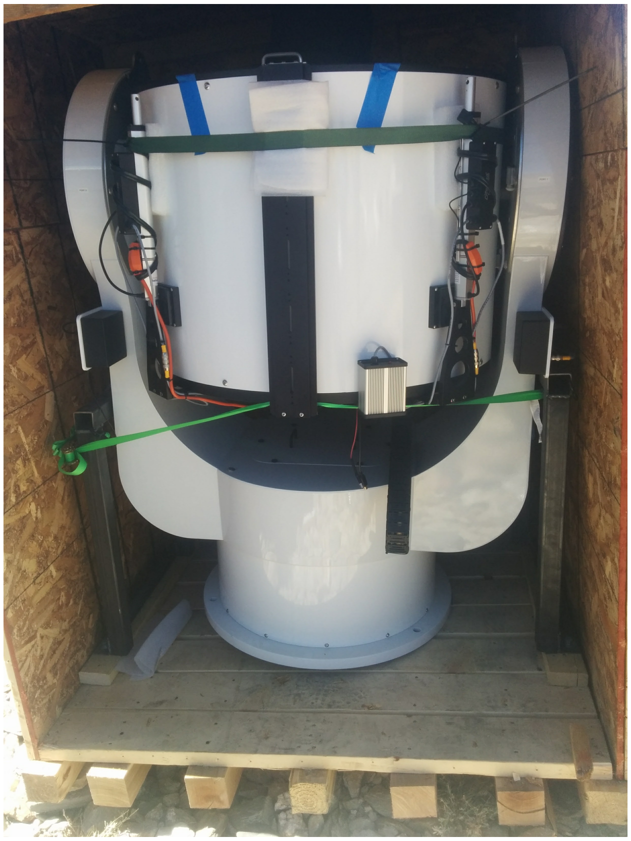


Figure 3.1: An image of the planewave mount at Hanle, Ladakh

3.2.2 Andor CCD

The camera to be used for the GROWTH-India telescope is the Andor iKon-XL, 230 model. It has a 4096 x 4108 resolution back-illuminated sensor which is thermo-electrically cooled. An important criterion for the selection of this sensor is its fast, low noise readout speeds up to 4MHz [1].

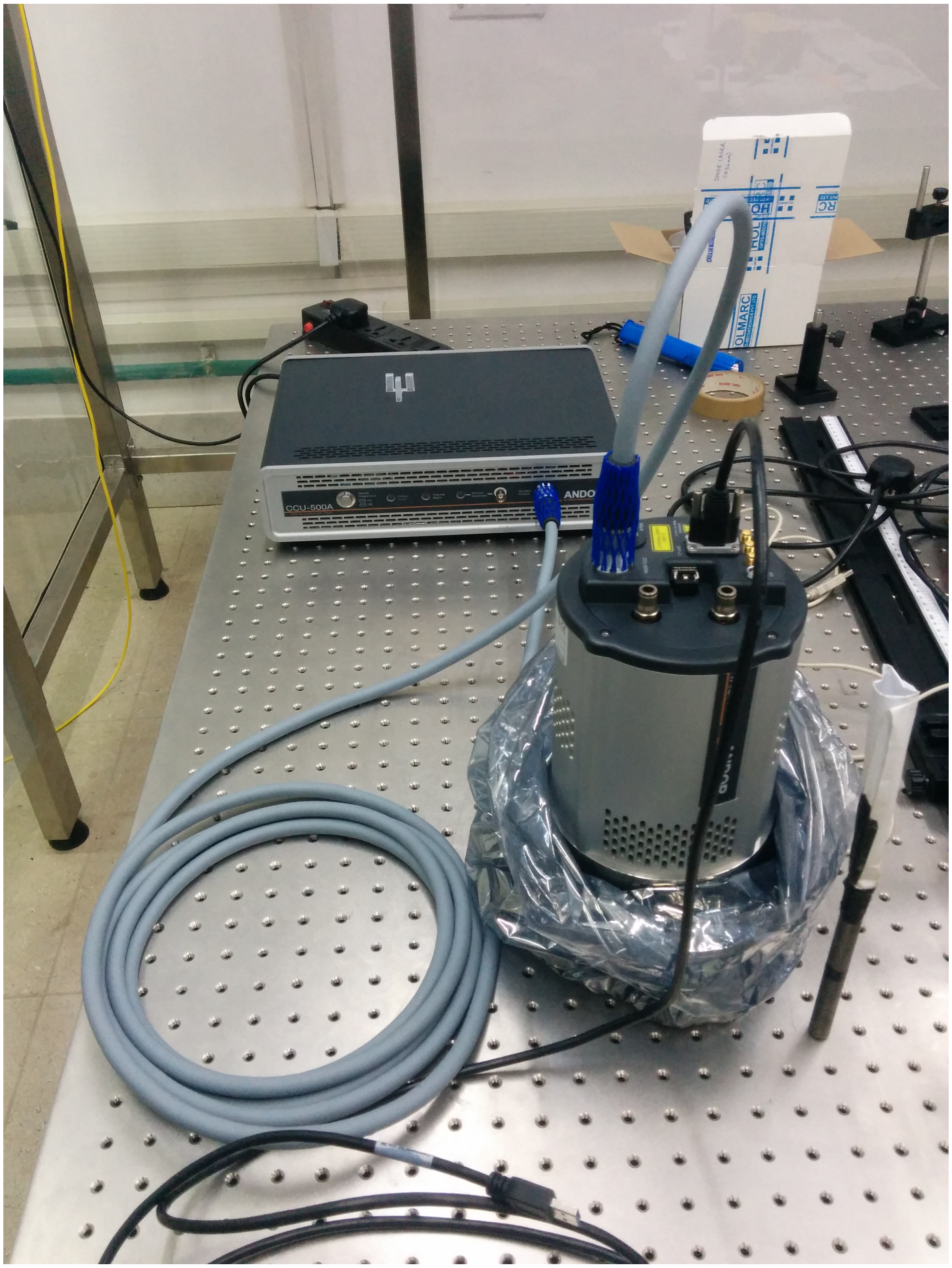


Figure 3.2: Initializing the filter wheel

3.2.3 Filter wheel and Dome controller

The control circuit for the Filter wheel unit had been built by Tarun Sharma, Ph.D. at IIA, Bangalore. It used a serial to USB connection to communicate with a computer system. The

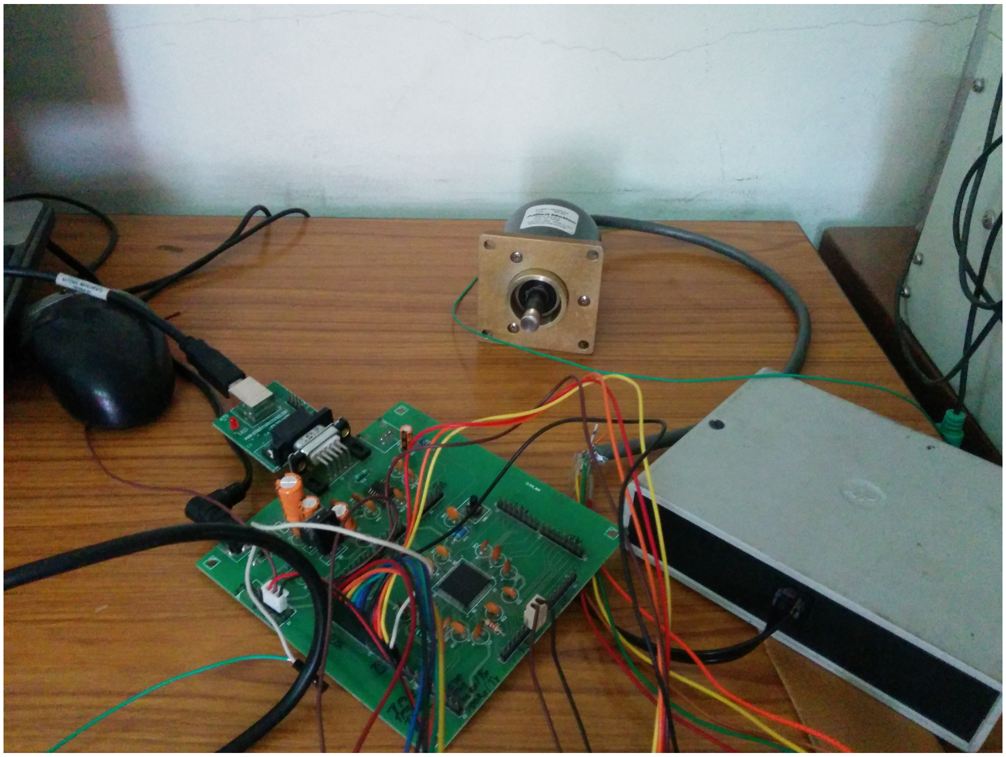


Figure 3.3: Setup of the filter wheel controller

dome controller is Diffraction limiteds' MaxDome II Observatory Dome Control System. It too communicates with the computer via a serial to USB communication.

3.3 Development of Filter wheel driver

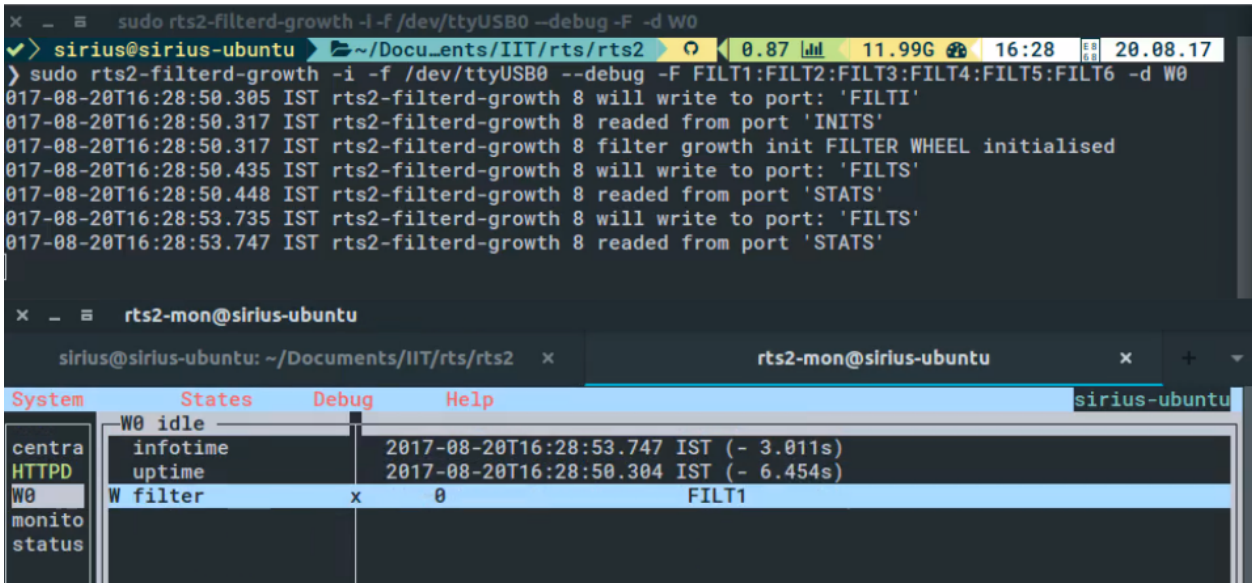
RTS2 being open source, even the drivers and communication libraries developed for each of the observatory components are open source. This helps a user to build drivers for unsupported devices using standard communication protocols (eg: serial connection or TCP protocol).

This being a custom-made filter wheel unit, didn't have plug and play drivers for the RTS2 OCS. However since it used the standard serial port communication, the necessary libraries already existed in RTS2. Hence, I wrote a driver in C++ and tested it by changing the filters using RTS2 as shown in figures 3.4 and 3.5 below.

Figure 3.4 shows two windows: one running the filter wheel driver and other running the RTS2 monitor (a terminal based GUI to control RTS2). 'FILTI' denotes the initialization command given to the Filter wheel every time the filter wheel driver starts. 'INITS' is the response from the filter wheel upon successful initialization. Similarly, 'FILTS' is issued to check the current status and 'STATS' gives an indication that everything works as expected.

Figure 3.5 shows the same two windows as above with the 'FILT3' command being given via the computer to the Filter wheel. This command made the filter wheel rotate to the filter in 3rd position from the initial position (1st in this case).

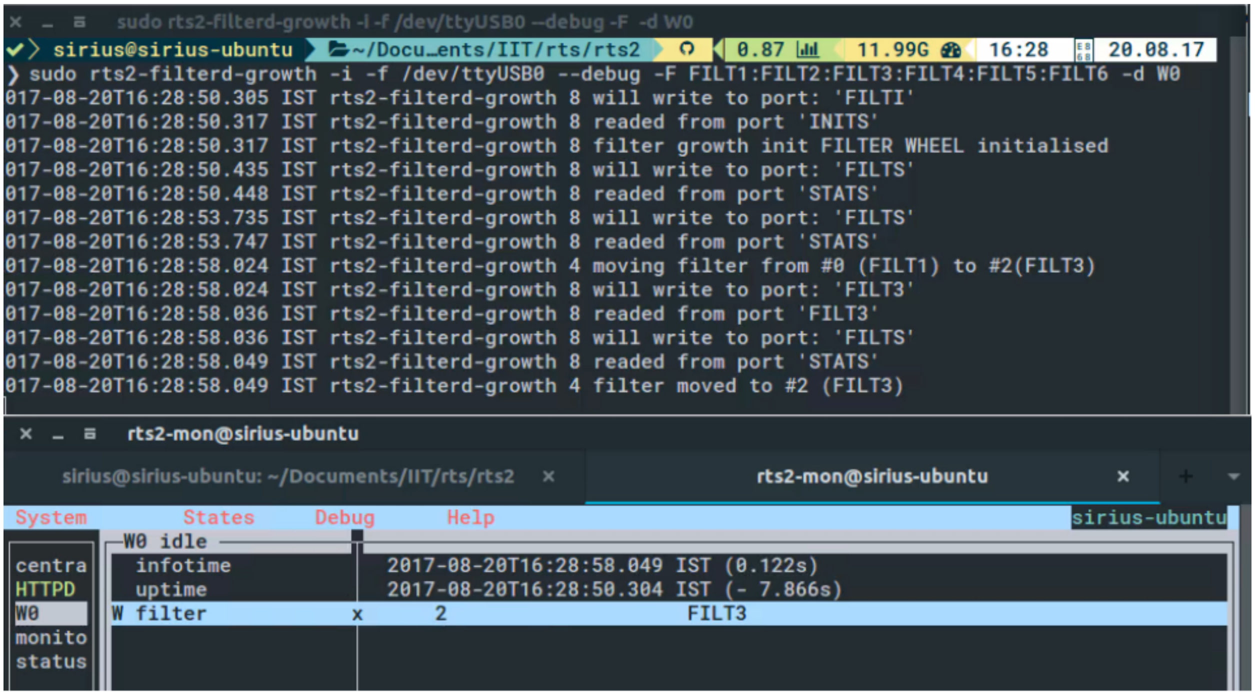
Such commands for changing filters, initializing the filter wheel, homing the filter wheel and checking its status were incorporated into the driver built for RTS2.



```
❯ sudo rts2-filterd-growth -i -f /dev/ttyUSB0 --debug -F FILT1:FILT2:FILT3:FILT4:FILT5:FILT6 -d W0
017-08-20T16:28:50.305 IST rts2-filterd-growth 8 will write to port: 'FILTI'
017-08-20T16:28:50.317 IST rts2-filterd-growth 8 readed from port 'INITS'
017-08-20T16:28:50.317 IST rts2-filterd-growth 8 filter growth init FILTER WHEEL initialised
017-08-20T16:28:50.435 IST rts2-filterd-growth 8 will write to port: 'FILTS'
017-08-20T16:28:50.448 IST rts2-filterd-growth 8 readed from port 'STATS'
017-08-20T16:28:53.735 IST rts2-filterd-growth 8 will write to port: 'FILTS'
017-08-20T16:28:53.747 IST rts2-filterd-growth 8 readed from port 'STATS'
```

```
        W0 idle
centra    infotime         2017-08-20T16:28:53.747 IST (- 3.011s)
HTTPD     uptime           2017-08-20T16:28:50.304 IST (- 6.454s)
W0      W filter        x      0                   FILT1
monito
status
```

Figure 3.4: Initializing the filter wheel



```
× _ ≡   sudo rts2-filterd-growth -l -f /dev/ttyUSB0 --debug -F -d W0
✔ > sirius@sirius-ubuntu > ~/Docu…ents/IIT/rts/rts2 >     0.87     11.99G     16:28     20.08.17
> sudo rts2-filterd-growth -i -f /dev/ttyUSB0 --debug -F FILT1:FILT2:FILT3:FILT4:FILT5:FILT6 -d W0
017-08-20T16:28:50.305 IST rts2-filterd-growth 8 will write to port: 'FILTI'
017-08-20T16:28:50.317 IST rts2-filterd-growth 8 readed from port 'INITS'
017-08-20T16:28:50.317 IST rts2-filterd-growth 8 filter growth init FILTER WHEEL initialised
017-08-20T16:28:50.435 IST rts2-filterd-growth 8 will write to port: 'FILTS'
017-08-20T16:28:50.448 IST rts2-filterd-growth 8 readed from port 'STATS'
017-08-20T16:28:53.735 IST rts2-filterd-growth 8 will write to port: 'FILTS'
017-08-20T16:28:53.747 IST rts2-filterd-growth 8 readed from port 'STATS'
017-08-20T16:28:58.024 IST rts2-filterd-growth 4 moving filter from #0 (FILT1) to #2(FILT3)
017-08-20T16:28:58.024 IST rts2-filterd-growth 8 will write to port: 'FILT3'
017-08-20T16:28:58.036 IST rts2-filterd-growth 8 readed from port 'FILT3'
017-08-20T16:28:58.036 IST rts2-filterd-growth 8 will write to port: 'FILTS'
017-08-20T16:28:58.049 IST rts2-filterd-growth 8 readed from port 'STATS'
017-08-20T16:28:58.049 IST rts2-filterd-growth 4 filter moved to #2 (FILT3)
```

```
× _ ≡   rts2-mon@sirius-ubuntu
    sirius@sirius-ubuntu: ~/Documents/IIT/rts/rts2  ×          rts2-mon@sirius-ubuntu          ×   +
System        States     Debug      Help                                          sirius-ubuntu
        ┌W0 idle ────────────
centra  │ infotime          │ 2017-08-20T16:28:58.049 IST (0.122s)
HTTPD   │ uptime            │ 2017-08-20T16:28:50.304 IST (- 7.866s)
W0      │W filter          x│     2                     FILT3
monito  │                   │
status  │                   │
```

Figure 3.5: Rotating the filter wheel

3.4 Modifying the Camera driver

The Andor camera had drivers available for Linux based OSs. Even though Andor cameras were, in general, supported by RTS2, the iKon camera wasn't able to communicate with RTS2. GDB (GNU Project Debugger) was then used to triangulate the problem which turned out to be the improper reading of the sensor dimensions by RTS2. The problem was then finally solved by me with the help of Petr. Dark flats were then obtained, as shown in the figure 3.6. Hence, the camera was able to successfully take exposures using RTS2.

3.5 Development of telescope driver

The planewave mount had a proprietary software and driver package for communicating with a Windows-based operating system. Also, RTS2 didn't have specific drivers made for this particular mount. However, the windows based driver had a socket based communication

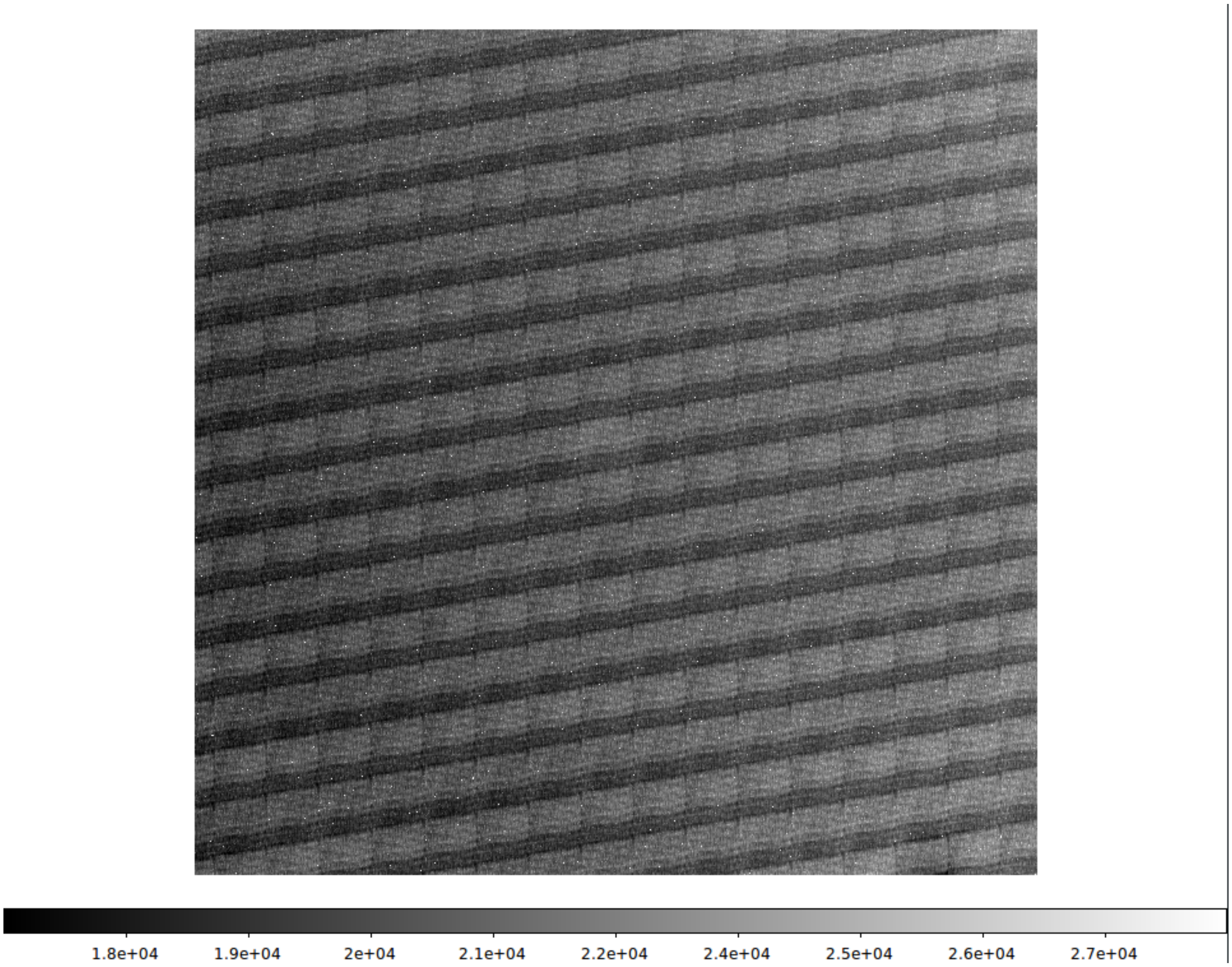


Figure 3.6: A screenshot of a dark frame captured by the andor camera, displayed in a DS9 window

option as well. Since all the commands needed to operate the mount were issuable using socket commands, the architecture as described in the figure 3.5 was decided to be used.

Initial tests were performed to simply send a 'hello world' text over the socket protocol using RTS2. Next, the actual driver was developed with the help of Viraj and Hrishikesh. This was done by translating the socket commands into abstract commands using which RTS2 works.

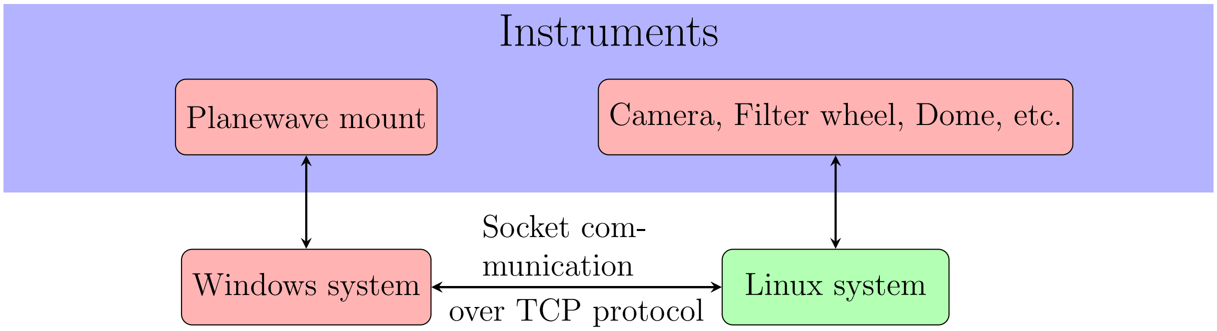

Instruments
Planewave mount
Camera, Filter wheel, Dome, etc.
Windows system
Socket communication over TCP protocol
Linux system

Figure 3.7: Connection between systems and instruments. The mount is directly connected to a windows os based system. The Linux system with RTS2 installed on is connected to all the other components except the mount. The Windows and Linux system talk to each other via socket commands.

3.6 Remote control from CREST

A visit to the CREST campus helped me understand the network architecture used for the communications between CREST, Bangalore and the Indian Astronomical Observatory (IAO), Ladakh. It helped us understand the limitations imposed by a low bandwidth network connection and its influence on the data pipeline.

3.7 Client-Server model

The GROWTH-India telescope is based on a client-server model. The remote Linux system(shown in green in figure 3.5) at Leh will act as the client. A priority based target list will be provided by the server at IIT, Bombay.

3.7.1 Finite state machine of the client side

On the client side of this model sits a python code (developed in an earlier semester) whose basic structure is given in figure 3.7.1

The code begins with going from the sleep state to idle if the telescope is simply turned on or to routine observation if any routine observations are to be performed by the telescope. If a

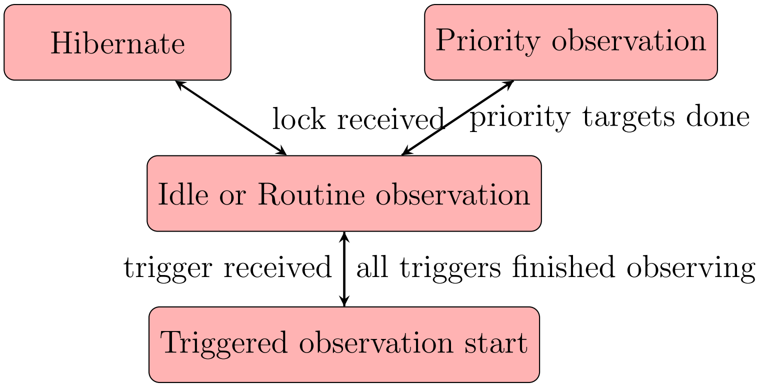

Hibernate
Priority observation
lock received
priority targets done
Idle or Routine observation
trigger received
all triggers finished observing
Triggered observation start

Figure 3.8: Main code layout

'lock' is found, the telescope priorities the ongoing observation and won't get disturbed from any trigger that the server may send. Only after these priority observations are completed, it goes back to performing the routine observations. When in this state, if a trigger sent by the server is received by the client, it will queue the triggers received.

3.7.2 Server code

The server-side code is developed on top of a Flask-based server created by Sujay Mate, IISER Pune.

Chapter 4

Ongoing and future work

4.0.1 Data pipeline

A data pipeline is being set up for the data to be obtained from the GROWTH-India telescope. This involves reduction of data at Hanle, retrieval of this data at CREST, Bangalore and relaying this data for archival at IIT, Bombay. The mechanism for this pipeline is currently being implemented as per the outline displayed in the figure 4.0.1.

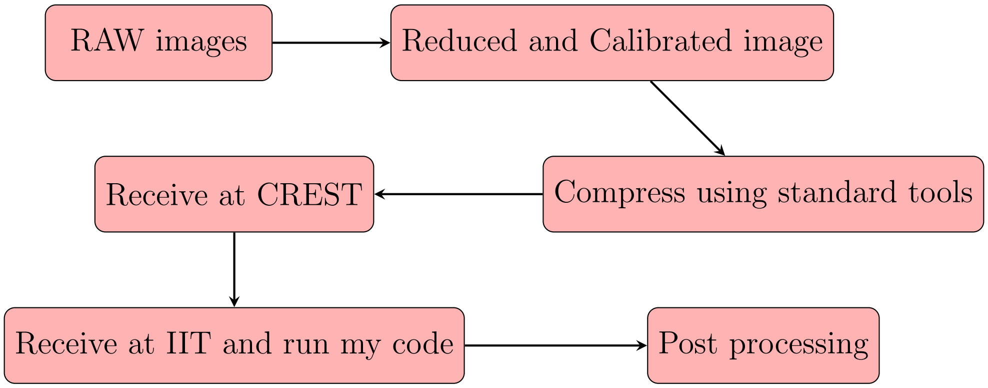

RAW images
Reduced and Calibrated image
Compress using standard tools
Receive at CREST
Receive at IIT and run my code
Post processing

Figure 4.1: Outline of the data pipeline

4.0.2 Implementation of automation in the telescope

I have been given the complete responsibility of ensuring the remote operability of this telescope. For this purpose, a visit to the site is planned in the month of November 2017. The major goal of this visit is to install and test the automation code in the systems at Hanle after the telescope has been physically set up and tested. This includes testing whether all the components, that is the mount, camera, filter wheel, dome controller work as expected with the main python code.

List of Abbreviations

BNS	Binary Neutron Star
σ	Standard Deviation
Mpc	Megaparsec
FITS	Flexible Image Transport System
DS9	SAOImage DS9 is an astronomical imaging and data visualization application.
SDSS	Sloan Digital Sky Survey
USNO	United States Naval Observatory
HTML	Hypertext Markup Language
FLASK	Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine
RTS2	Remote Telescope control Software
PWI3	Planewave Interface 3
TCP	Transmission Control Protocol
GUI	Graphical user interface
Alt-Az	Altitude-Azimuth
GDB	GNU Debugger

Bibliography

[1] Andor camera specifications. http://www.andor.com/pdfs/specifications/Andor_iKon-XL_230_Specifications.pdf.

[2] Flask web framework. http://flask.pocoo.org/.

[3] Planewave cdk 700 specifications. http://planewave.com/products-page/cdk700/0-7m-cdk-telescope-system/.

[4] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, et al. Gw170817: Observation of gravitational waves from a binary neutron star inspiral. *Phys. Rev. Lett.*, 119:161101, Oct 2017.

[5] Benjamin P Abbott, R Abbott, TD Abbott, F Acernese, K Ackley, C Adams, T Adams, P Addesso, RX Adhikari, VB Adya, et al. Gw170814: A three-detector observation of gravitational waves from a binary black hole coalescence. *Physical Review Letters*, 119(14):141101, 2017.

[6] Benjamin P Abbott, Richard Abbott, TD Abbott, MR Abernathy, Fausto Acernese, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, RX Adhikari, et al. Observation of gravitational waves from a binary black hole merger. *Physical review letters*, 116(6):061102, 2016.

[7] Christopher P Ahn, Rachael Alexandroff, Carlos Allende Prieto, Scott F Anderson, Timothy Anderton, Brett H Andrews, Éric Aubourg, Stephen Bailey, Eduardo Balbinot, Rory Barnes, et al. The ninth data release of the sloan digital sky survey: first spectroscopic data from the sdss-iii baryon oscillation spectroscopic survey. *The Astrophysical Journal Supplement Series*, 203(2):21, 2012.

[8] Emmanuel Bertin and Stephane Arnouts. Sextractor: Software for source extraction. *Astronomy and Astrophysics Supplement Series*, 117(2):393–404, 1996.

[9] Alberto Javier Castro-Tirado. Robotic autonomous observatories: a historical perspective. *Advances in Astronomy*, 2010, 2010.

[10] Justin D Finke and Soebur Razzaque. The binary black hole merger rate from ultraluminous x-ray source progenitors. *Monthly Notices of the Royal Astronomical Society*, 472(3):3683–3691, 2017.

[11] M. M. Kasliwal, E. Nakar, L. P. Singer, D. L. Kaplan, D. O. Cook, A. Van Sistine, R. M. Lau, C. Fremling, O. Gottlieb, J. E. Jencson, S. M. Adams, U. Feindt, K. Hotokezaka, S. Ghosh, D. A. Perley, P.-C. Yu, T. Piran, J. R. Allison, G. C. Anupama, A. Balasubramanian, K. W Bannister, J. Bally, J. Barnes, S. Barway, E. Bellm, V. Bhalerao, D. Bhattacharya, N. Blagorodnova, J. S. Bloom, P. R. Brady, C. Cannella, D. Chatterjee, S. B. Cenko, B. E. Cobb, C. Copperwheat, A. Corsi, K. De, D. Dobie, S. W. K. Emery, P. A. Evans, O. D. Fox, D. A. Frail, C. Frohmaier, A. Goobar, G. Hallinan, F. Harrison, G. Helou, T. Hinderer, A. Y. Q. Ho, A. Horesh, W.-H. Ip, R. Itoh, D. Kasen, H. Kim, N. P. M. Kuin, T. Kupfer, C. Lynch, K. Madsen, P. A. Mazzali, A. A. Miller, K. Mooley, T. Murphy, C.-C. Ngeow, D. Nichols, S. Nissanke, P. Nugent, E. O. Ofek, H. Qi, R. M. Quimby, S. Rosswog, F. Rusu, E. M. Sadler, P. Schmidt, J. Sollerman, I. Steele, A. R. Williamson, Y. Xu, L. Yan, Y. Yatsu, C. Zhang, and W. Zhao. Illuminating Gravitational Waves: A Concordant Picture of Photons from a Neutron Star Merger. *ArXiv e-prints*, October 2017.

[12] Petr Kubánek, Martin Jelínek, John French, Michal Prouza, Stanislav Vítek, Alberto J Castro-Tirado, and Victor Reglero. The rts2 protocol. In *Advanced Software and Control for Astronomy II*, volume 7019, page 70192S. International Society for Optics and Photonics, 2008.

[13] John H Mathews, Kurtis D Fink, et al. *Numerical methods using MATLAB*, volume 4. Pearson London, UK:, 2004.

[14] Brian D Metzger and Edo Berger. What is the most promising electromagnetic counterpart of a neutron star binary merger? *The Astrophysical Journal*, 746(1):48, 2012.

[15] David G Monet, Stephen E Levine, Blaise Canzian, Harold D Ables, Alan R Bird,

Conard C Dahn, Harry H Guetter, Hugh C Harris, Arne A Henden, Sandy K Leggett, et al. The usno-b catalog. *The Astronomical Journal*, 125(2):984, 2003.

[16] John A Nelder and Robert Mead. The downhill simplex algorithm. *Computer Journal*, 7(S 308), 1965.