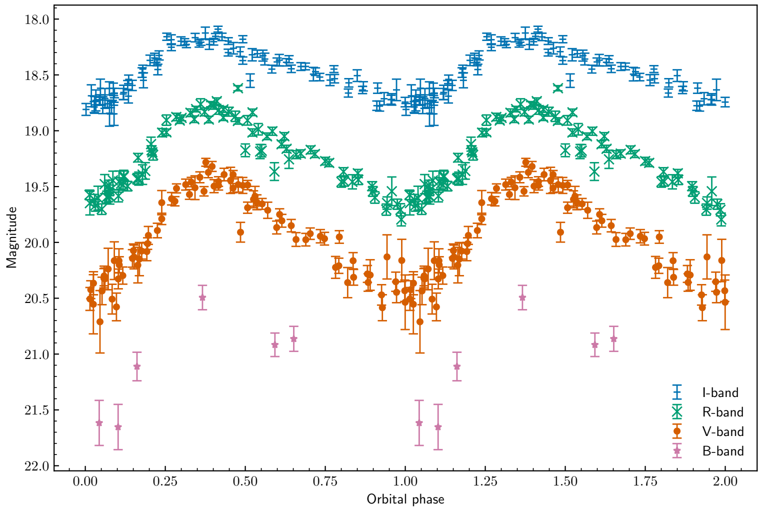
Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

Anders Tidemann

Robotic optical observations of compact binary millisecond pulsars

Master's thesis in MSc in Physics
Supervisor: Manuel Linares
June 2023



NTNU
Norwegian University of
Science and Technology

Anders Tidemann

Robotic optical observations of compact binary millisecond pulsars

Master’s thesis in MSc in Physics
Supervisor: Manuel Linares
June 2023

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

NTNU
Norwegian University of
Science and Technology

Acknowledgements

This thesis marks the end of 5 years of studying physics and the end of my MSc in Physics at the Department of Physics of the Norwegian University of Science and Technology (NTNU).

I want to express my gratitude to the people who have helped me achieve what I have accomplished. First and foremost, I would like to thank Prof. Manuel Linares for being my supervisor over the past year. I greatly appreciate your patience whenever I needed extra time, for creating a thesis that has been enjoyable to work on, and for providing valuable feedback on my thesis work and writing. This would not have been possible without your support.

I would also like to extend my thanks to the LOVE-NEST group for the bi-weekly meetings. It was a great opportunity to learn more about neutron stars and explore beyond the academic life of a student. I truly appreciate the feedback and help that all of you have provided over the past year.

Additionally, I want to express my gratitude to Dr. Mark Kennedy and Dr. Alice K. Harding for our fruitful conversations about my thesis work. Your insights have been invaluable.

I would like to thank the Department of Physics for giving me the chance to write a master's thesis and for providing me with a computer during this year of research. The subjects I've had the opportunity to study here have been fascinating, and I've thoroughly enjoyed my time as a student in this department for the past five years.

Finally, I want to give a big thank you to my friends and family for their continuous support throughout my years in Trondheim. Life as a student would have been tougher without all of you by my side. I also want to extend my gratitude to Elias O. Slettevoll and Konrad E. R. Jervell. The countless hours we've spent working and engaging in extracurricular activities together have brought me immense happiness.

Anders Tidemann

Trondheim, 7th of June, 2023

Abstract

This master's thesis focuses on the robotic optical observations of the compact binary millisecond pulsar PSR J1048+2339, obtained using the Joan Oró Telescope (TJO). As part of the thesis work, a pipeline to streamline the observations taken with the TJO was developed. This pipeline was applied to observations of PSR J1048+2339 taken with the TJO over a four-month period in 2020. We present the findings from these observations, complemented by data gathered from the Nordic Optical Telescope (NOT).

The thesis also presents the theoretical background for the study of neutron stars, covering topics such as their formation, physical parameters, and evolution. We discuss how the field of observational astrophysics is shaped by modern robotic optical observing methods, including the use of sky surveys and small robotical telescopes. Building upon this, we develop and implement a user-friendly pipeline and demonstrate its practical advantages for observers utilizing the TJO.

Furthermore, we present the results of optical photometric observations of the compact binary millisecond pulsar PSR J1048+2339 obtained from both the TJO and the NOT. The data reveal consistent single-peaked asymmetrical light curves over the course of four months of observations. The single-peaked nature of the light curve points to an irradiated companion. We find a strong asymmetry; the flux maximum is observed about 0.1 orbital cycles before superior conjunction of the companion star. This might be the result of a swept-back intra-binary shock. We speculate that this disparity may lead to a hotspot with its highest temperature occurring at phase $\phi \simeq 0.45$.

Sammendrag

Denne masteroppgaven fokuserer på robotiske optiske observasjoner av den kompakte dobbeltstjerne-millisekundpulsaren PSR J1048+2339, utført ved hjelp av Joan Oró-teleskopet (TJO). Som en del av masterarbeidet ble en programvare produsert for å effektivisere observasjoner tatt med TJO. Denne programvaren ble anvendt på data av PSR J1048+2339 samlet inn i løpet av en periode på fire måneder i 2020 av teleskopet. Vi presenterer funnene fra disse observasjonene, supplert med data samlet inn fra Nordic Optical Telescope (NOT).

Oppgaven utforsker den teoretiske bakgrunnen for nøytronstjerner og dekker emner som nøytronstjernedannelse, ulike fysiske parametere og nøytronstjerners utvikling. Vi undersøker hvordan feltet for observasjonsastrofysikk påvirkes av moderne, robotiske, optiske observasjonsmetoder, inkludert bruk av heldekkende undersøkelser av nattehimmelen (sky surveys) og små robotiske teleskoper. Fra dette utgangspunktet utvikler og implementerer vi en brukervennlig programvare og demonstrerer dens praktiske fordeler for observatører som benytter TJO.

Videre presenterer vi resultatene av optiske fotometriske observasjoner av den kompakte dobbeltstjerne-millisekundpulsaren PSR J1048+2339, utført både ved hjelp av TJO og NOT. Dataene viser konsekvente, asymmetriske lyskurver med ett enkelt maksimum i løpet av de fire månedene med observasjoner. Dette enkle makimumet i lyskurven tyder på at kompanjongstjernen er bestrålt av millisekundpulsaren, mens asymmetrien kan være resultatet av et sjokk mellom stjernene som følge av interaksjonen mellom pulsarvinden og kompanjongvinden. Grunnet den sirkulære bevegelsen til stjernene om hverandre kan dette sjokket flyttes mot siden av kompanjongstjernen som er vendt vekk fra bevegelsesretningen. Lyskurvene viser ulike grader av asymmetri mellom ulike bånd, noe som potensielt kan tilskrives den underliggende ellipsoidale modulasjonen som forventes av en deformert kompanjonstjerne. Vi spekulerer i om dette kan skyldes en varm flekk, som følge av pulsarstrålingen, med høyest temperatur ved fase $\phi \simeq 0.45$.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

There are many different celestial objects that are interesting to study. Among them, neutron stars display a wide range of observable phenomena, and feature extreme physical properties, which make them of great interest to astrophysicists. These neutron stars are dense objects that are rapidly rotating (spin period $P \sim 1.44\text{ms} - 10\text{s}$ [10, 57]) with strong magnetic fields (surface field strength $B \sim 10^7 - 10^{15}\text{G}$ [10, 57]). Neutron stars are among the densest objects in the universe (their cores can reach densities $\rho \sim 10^{15}$ g cm^{-3} [46]), much denser than anything that we can create on Earth. By studying them we can gain a better understanding of the fundamental properties of matter in such extreme conditions.

In addition to being some of the densest objects in the universe, neutron stars also exhibit a range of interesting and complex phenomena. For example, they can emit beams of electromagnetic radiation from their magnetic poles allowing us to study the behaviour of radiation and matter in extreme magnetic fields. They can form binary systems with other stars, leading to phenomena such as accretion and mass transfer. It is also possible to observe mergers between neutron stars which give rise to gravitational waves, a phenomenon that can lead to a better understanding of the neutron star interior [71] and the origin of heavier nuclei [8].

1.1 Neutron stars and pulsars

In 1967, at the Mullard Radio Astronomy Observatory in Cambridge, a radio telescope was installed. The instrument's objective was to study the angular structure of compact radio sources, which are celestial objects that emit radio waves and have a small angular size. To achieve this, the instrument observed the scintillation effect, which occurs due to the irregular structure of the interstellar medium [48]. The instrument was sensitive enough to detect human-made noise, however among all the noise Jocelyn Bell, the student of Tony Hewish, discovered a periodic signal. It was first thought to be weak and sporadic interference, but the signal seemed to be originating from a specific location in the sky. Later in the year, systematic investigations revealed that the signal pulsed at a stable $P = 1.337\text{s}$. With the lack of astrophysical knowledge of any celestial object capable of producing such a stable periodic signal, the team named the source Little Green Men, or LGM-1 for short [74]. This was the first discovery of a pulsar, a rapidly rotating magnetized neutron star.

1.1.1 Formation of neutron stars

Stars undergo a series of changes over their lifetime, including changes in their size, temperature, and composition. At the end of their lives, stars can explode in a display of light and energy or simply fade away. The ultimate fate of a star is determined by its mass. For low-mass stars like our Sun, the end of their life is relatively peaceful. They eventually run out of fuel, expand into a

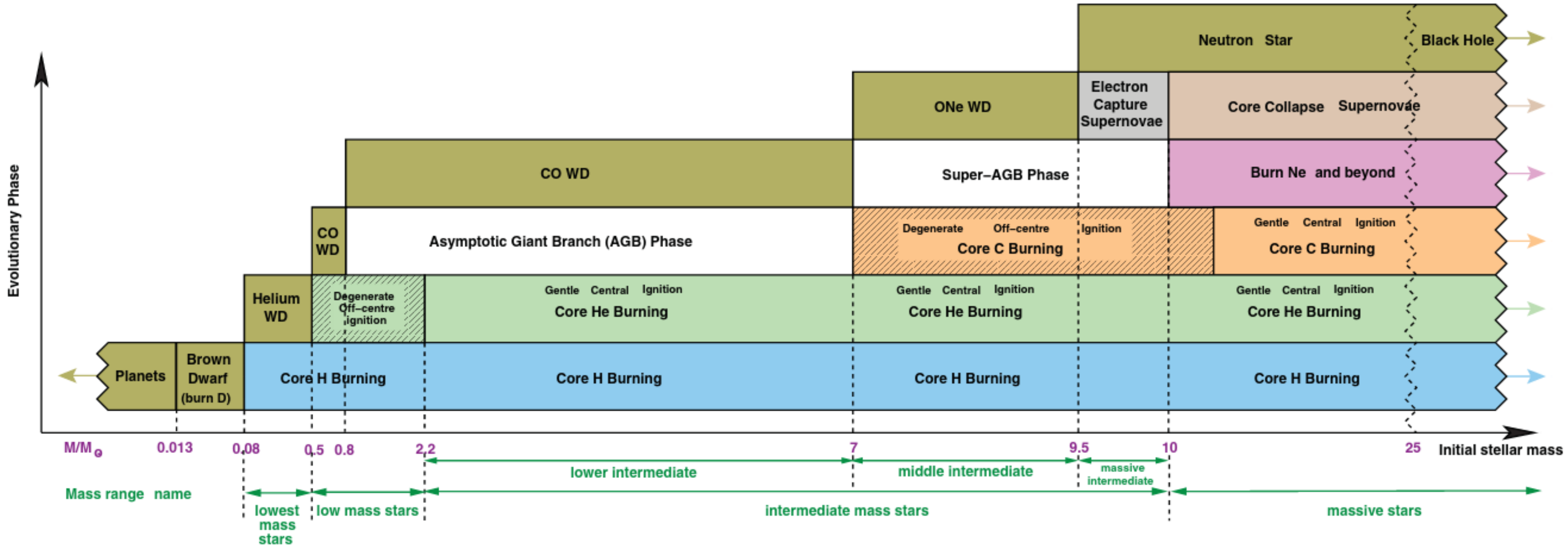


Figure 1.1: This figure from Karakas et al. (2014) [56] depicts how the evolution of stars is affected by the initial mass. It determines the nuclear burning phases and the destiny of the remnant. Here we can see that for masses $9.5M_\odot$ - $25M_\odot$, the resulting remnant will be a neutron star. Note that the masses given in the figure are indicative only and that the real physical limits will vary depending on certain details, such as the composition of the star.

red giant, and then shed their outer layers to become a white dwarf [3]. However, high-mass stars, $> 9.5M_\odot$ where $M_\odot$ is the mass of our Sun ($M_\odot \approx 1.99 \times 10^{33}$g), have a much more dramatic fate. Stars that have a lower mass than this will usually shed their envelope before the core reaches the critical Chandrasekhar mass of $1.4M_\odot$.

Stars of an initial mass greater than or equal to $10M_\odot$ will end their life in a core collapse supernova (ccSN). These massive stars will go through the fusion of lighter elements until they produce iron. The iron will then sink to the bottom of the gravitational potential and form an iron core [74]. As iron is the most stable element, any further fusion processes would not release any energy. This then removes the outwards thermal pressure that was present during the fusion of lighter elements, and the core contracts until the electron-degeneracy pressure can support the star. When this degenerate iron core reaches the Chandrasekhar limit the self-gravity becomes too strong and the core collapses and contracts further. Upon further contraction, the pressure increases and the resulting increase in temperature allows for the photo-disintegration of heavy nuclei into free neutrons and helium. As the density in the core increases, the free electrons get forced together with the protons to create neutrons via inverse beta-decay, producing neutrons and neutrinos. When the core reaches nuclear density, the equation of state stiffens and any infalling matter is sent outwards in an explosive fashion. What remains depends on the initial mass of the main-sequence star. The supernova remnant would be a black hole if the remaining core, after the ccSN, is massive enough to overcome the neutron degeneracy pressure. The condition for this occurrence is an initial mass exceeding about $25M_\odot$ (see Fig. 1.1). A star with an initial mass in the approximate range $10M_\odot - 25M_\odot$ will thus lead to the formation of a neutron star.

1.1.2 Neutron star structure

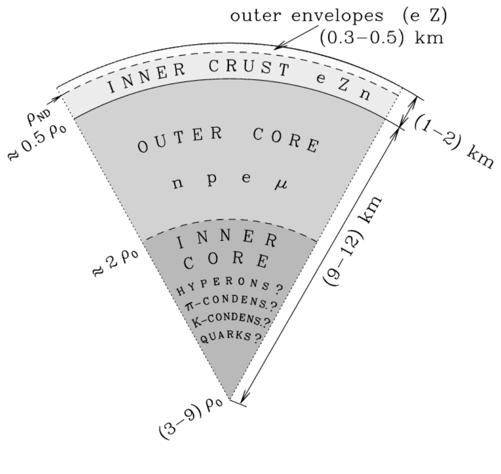


Figure 1.2: Graphic summary of the different layers of a neutron star [46].

In the neutron star atmosphere one can find densities in the range of a few grams g cm^{-3}, whereas towards the center it is possible to reach densities of 10^{15}g cm^{-3} In order to properly model the insides of neutron stars one wishes to relate this density with the pressure, temperature and the generated magnetic field, in what is known as the equation of state [60].

The structure of a neutron star consists of an atmosphere and 4 inner layers; the outer crust, inner crust, outer core and inner core. The atmosphere is the thinnest layer at about $\sim 0.1 - 100$ cm, with a density range of $\sim 0.1 - 10$ g cm^{-3}. In comparison, the atmosphere of a regular star usually has a height of $\sim 10^8$ cm and density of $\sim 10^{-7}$ g cm^{-3} [77]. It is found that the atmosphere is made up of plasma, and that the thickness varies with temperature, with hotter surface temperatures leading to thicker atmospheres. As in other stars, the atmosphere is where the thermal electromagnetic radiation that we observe is created, and this radiation contains important information on the neutron stars physical quantities such as mass and radii [46].

The layer beneath the atmosphere is the outer crust, which ranges from the densities $\rho \sim 10\,\mathrm{g\,cm^{-3}}$ to $\rho \sim 10^{11}\,\mathrm{g\,cm^{-3}}$. The outer crust consists of iron ^{56}Fe in a cubic lattice. As the density increases to $\rho \sim 10^4$ g cm^{-3} deeper in the outer crust these iron atoms become fully ionized as a result of the electron pressure, and at densities of $\sim 10^7$ g cm^{-3} the crust undergoes neutronization. This is a process where the atoms experience electron capture and the nuclei structure becomes more neutron rich [19]. As we approach a couple hundred meters deep into the outer crust and densities of $\rho = \rho_{ND} \sim 10^{11}$ g cm^{-3} we encounter a phenomenon called neutron drip, where the neutrons begin to "drip" out of the nuclei to produce free neutrons. This is the point at which the outer crust ends and the inner crust begins.

The inner crust is about $1 - 2\,\mathrm{km}$ thick, and ranges from densities ρ_{ND} at the top to $0.5\rho_0$ at the bottom (where $\rho_0 = 2.8 \times 10^{14}\,\mathrm{g\,cm^{-3}}$ is the nuclear saturation density [46]). At $\rho > 4 \times 10^{11}\,\mathrm{g\,cm^{-3}}$, the free neutrons will form a neutron gas outside of the nuclei. Indeed, the inner crust consists of free neutrons, electrons and nuclei rich in neutrons, however the amount of free neutrons increases as density increases. As we get closer to the crust-core transition, and densities approach that of atomic nuclei, the nuclear attraction and Coulomb repulsion (together also known as Coulomb frustration) will start to warp matter into convoluted shapes such as slabs, cylinders and spheres [49, 106]. This part of the crust is known as the "pasta"-phase. Even though the "pasta"-layer is a smaller part of the total crust it still constitutes a large fraction of the total mass of the crust [64].

Inside the crust we find the core of the neutron star. Much like the crust this is also divided into an inner and an outer part. The outer part of the core is several kilometers thick and the density ranges from $0.5\rho_0$ to $2\rho_0$. It is composed of nucleons, electrons and muons, often referred to as a $npe\mu$ composition [59]. In order to determine the state of this matter one employs electric neutrality and beta equilibrium, where beta equilibrium is the equilibrium concerning the beta decay of neutrons and inverse processes. The $npe\mu$-plasma components are heavily degenerate, with the muons and electrons forming almost ideal Fermi gases. The neutrons and protons, which interact via nuclear forces, constitute a strongly interacting Fermi liquid and can exist in a superfluid state [46].

The very center of a neutron star is known as the inner core. This is where the density $\rho > 2\rho_0$, and its radius stretches several kilometers. The inner core can reach densities up to $10 - 15$ times the nuclear matter density, ρ_0. The composition of the inner core is not fully agreed upon, and there are different hypotheses as to what it can consist of depending on the model used to build the equation of state. Looking at Figure 1.2 we can see four exotic states of matter which may exist in the core; hyperons, pion condensation, kaon condensation and quark matter [46].

1.1.3 Magnetic fields

In modern astrophysics, magnetic fields have a prominent role and are responsible for a variety of energetic phenomena, spanning from solar flares to fast radio bursts. The strongest magnetic fields are found in neutron stars, and this magnetic field partly determines how we view the neutron star, be it as a dim, cooling neutron star, radio pulsar or a magnetar. It is possible to estimate the initial magnetic field strength through the conservation of magnetic flux through the stellar surface. As the dying star collapses, the surface area becomes smaller, but the number of magnetic field lines that goes through this surface area is conserved. From the assumption of conservation of magnetic flux we get $B_{NS} = (R_0/R_{NS})^2 B_0$ (where $R_{NS} = 10$ km). If we insert the values found for the Sun ($B_0 = B_\odot = 1$G and $R_0 = R_\odot$) we get a magnetic field strength $B \sim 10^{10}$ G. The typical magnetic field strength of a young neutron star is in the range $10^{11}\mathrm{G} - 10^{13}$ G, with some exceptions, so this is a simple estimate which falls 1-3 orders of magnitude short [41].

1.1.4 Rotation

It is possible to estimate the rate of rotation of a neutron star through the assumption of conservation of angular momentum. Given the for angular momentum scaling, $L \propto {}^{R^2}\!/_{P}$ (where R and P are the radius and spin period of the neutron star, respectively), we can find the relation between the spin-period before and after the collapse that creates the neutron star to be

$$P_{NS} = \frac{R_{NS}^2}{R_0^2} P_0. \tag{1.1}$$

If a star with a core the size and rotational period of the Sun were compressed into a neutron star, we get, by inserting $P_\odot = 27$ d (here we have chosen the Carrington period of the Sun [75]), $R_\odot = 6.9 \times 10^{10}$ cm and the canonical radius of a neutron star, $R_{NS} = 10\,\mathrm{km}$, a period of

$P_{NS} = 0.7$ ms. This value does not seem to correlate well with observed spin-periods which are found to be varying from 23.5 s down to 1.4 ms [17]. This is only an estimate and there are several reasons as to why we do not observe newly born neutron stars with spin periods of this order. For instance, as the neutron is born it undergoes an initial strong spin-down phase which can be due to magnetic breaking and interactions with the in-falling matter from the supernova. Additionally, the radius of the core of massive stars is likely smaller or the spin period longer, leading to higher spin periods of newly born neutron stars.

Spin-down

As neutron stars rotate, so does their magnetic field. The rotation of the magnetic field leads to beaming of electromagnetic radiation originating in the magnetic poles of the neutron star. The electromagnetic radiation radiates away part of the angular momentum of the star, and the result is a slow and continuous spin-down. One usually denotes the spin-down of a neutron star by the symbol $\dot{P}$, which is the derivative of the period P with respect to time. The spin-down rate varies from system to system, but it has been shown that neutron stars that have shorter spin-periods P usually have smaller spin-down rates (see Fig. 1.3). Usually the spin-down rates are within the range of $\dot{P} \sim 10^{-21} - 10^{-9}$ [23].

There are several important quantities that can be calculated using the values of P and $\dot{P}$. One of these quantities is the spin-down luminosity. Given a moment of inertia I, the rotational energy of an object is given as

$$E = \frac{I\Omega^2}{2} = \frac{2\pi^2 I}{P^2}, \tag{1.2}$$

where $\Omega = \frac{2\pi}{P}$ is the angular frequency of rotation. If we differentiate this with respect to time, we get the change in rotational energy, more commonly known as spin-down luminosity, as a function of P and $\dot{P}$;

$$-\dot{E} = -\frac{dE}{dt} = \frac{4\pi^2 I\dot{P}}{P^3}. \tag{1.3}$$

It is important to note that this is not a measured luminosity, but the measured decrease in rotational energy of the neutron star. This change in rotational energy is what sets the energy budget on the neutron stars emitted electromagnetic radiation. If we assume that $-\dot{E}$ is approximately equal to the luminosity of the magnetic dipole radiation, it is possible to set a lower limit for the surface magnetic field strength of the star (B). From [23] it is given that

$$B^2 = \frac{3c^3 IP\dot{P}}{2 \cdot 4\pi^2 R^6 \sin^2 \alpha} \tag{1.4}$$

where α is the angle between the spin and the magnetic axes of the neutron star. This equation will have its lowest value when $\sin \alpha = 1$, thus yielding a lower bound on the magnetic field strength at the surface

$$B > \left(\frac{3c^3 IP\dot{P}}{2 \cdot 4\pi^2 R^6} \right)^{1/2}. \tag{1.5}$$

Finally, under the same assumption that the radiative power $P_{rad} \approx -\dot{E}$, no significant change in the inclination angle α, and the assumption that the initial period of the star was much shorter than the current period, we can calculate the characteristic age to be

$$\tau \equiv \frac{P}{2\dot{P}}. \tag{1.6}$$

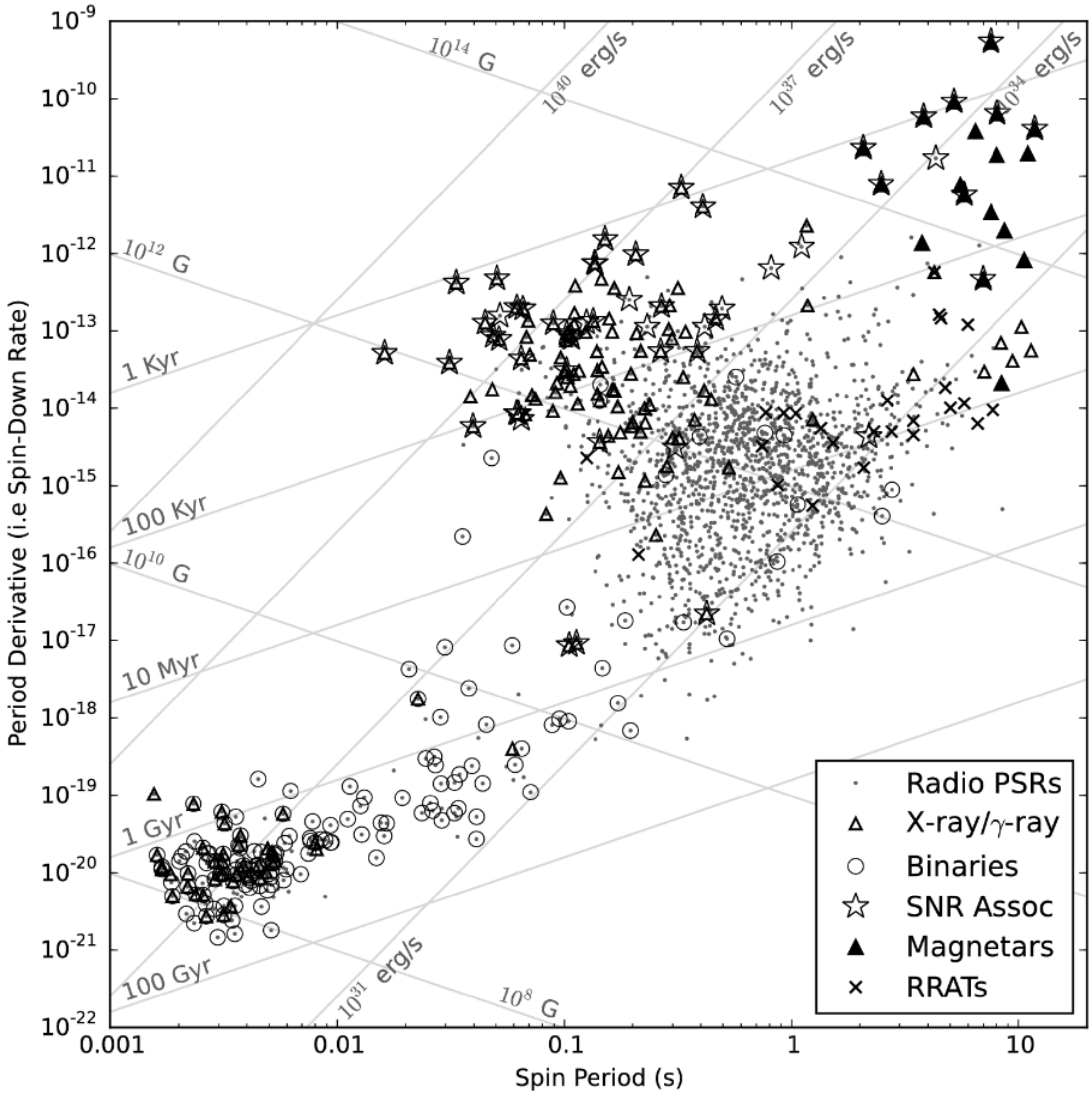


Figure 1.3: P-$\dot{P}$ diagram [23]. It is useful in order to visualize the lives of pulsars and the information we can gather through the properties of the spin period P and its time derivative $\dot{P}$. The spin-down luminosity, minimum magnetic field strength and characteristic age (that are outlined in section 1.1.4) are plotted as straight lines, and the position of each star gives insight in the values of these physical properties.

Spin up

Following the initial discovery of neutron stars, another intriguing finding emerged: the identification of systems in which the spin period of the neutron stars was observed to be decreasing instead of increasing. This is made possible due to the accretion of matter and angular momentum onto the neutron star. This either occurs when the orbital separation decreases due to gravitational wave radiation, or as the companion increase in size due to stellar evolution. In either scenario the star will begin to fill the Roche-lobe, the region around a star in binary systems where matter is gravitationally bound to that star. The matter then overflows from the Roche-lobe of the companion into the Roche-lobe of the neutron star, passing through the L1 Lagrangian point, and forms an accretion disc of stellar matter around the neutron star. The matter falls onto the star carrying angular momentum from the inner-most Keplerian orbits, resulting in the transfer of angular momentum onto the neutron star. Another mode of transferring angular momentum is through the magnetic field lines connected to the inner disc. As we can see in Figure 1.3 the younger neutron star population have spin periods 0.1s and the older population have shorter periods in the millisecond range. It can be inferred that the spin-up process is slow and that pulsars with low millisecond periods have been spun up by accretion. This spin-up process is known as “recycling” and during the process we can observe the systems as low-mass X-ray binaries.

These low-mass X-ray binaries are thought to be the evolutionary predecessor of millisecond pulsars. This theory is backed by the discovery of transitional millisecond pulsars, neutron stars that interchange between a radio millisecond pulsar state and an accretion powered X-ray pulsar such as PSR J1023+0038 [5], PSR J1824-2452I [76] and PSR J1227-4853 [87]. These observations of binary systems undergoing a transition from X-ray accretion-powered systems to rotation-powered radio millisecond pulsars provide strong evidence supporting the idea that millisecond pulsars are old and slowly rotating neutron stars that have been spun up by accretion in a low-mass X-ray binary system [67].

1.1.5 Pulsars

A pulsar is a rapidly rotating, highly magnetized neutron star that emits electromagnetic radiation in at least one band (most often radio). This electromagnetic radiation can be emitted from the magnetic poles of the neutron star through the acceleration of charged particles along the magnetic field lines [65], and the direction of the electromagnetic beam is determined by the magnetic axis of the pulsar (see Fig. 1.4). To be able to observe a pulsar as a pulsating source the magnetic axis has to be misaligned to the rotational axis of the neutron star, and a part of the electromagnetic beam needs to be visible from Earth.

The family of pulsars is often divided into 3 sub-categories; rotation-powered, accretion-powered and magnetars. Each displays different physical qualities when observed as the underlying physics differ. Rotation-powered pulsars are pulsars where the pulsar’s rotational energy powers electromagnetic emission. This causes the pulsar to lose angular momentum and slowly spin down. As this emission is observable in radio wavelengths, these pulsars are more commonly known as radio pulsars.

Accretion-powered pulsars exhibit X-ray luminosities of $\sim 10^{37}\,\mathrm{erg\,s^{-1}}$ [42], which can be accounted for by accretion. As matter from the companion accretes onto the magnetic poles of the neutron star, X-ray hot spots are produced on the poles, and as the neutron star rotates we can observe periodic X-ray radiation, much like the periodicity of radio emission from radio-pulsars. This X-ray emission leads to the more commonly used name, X-ray pulsars.

The strongest magnetized objects in the universe are magnetars. This neutron star population houses magnetic fields of $B \sim 10^{14} - 10^{16}$G, and it is thought that the fields originate from a dynamo process that occurs in the collapse of the parent star. This process amplifies the magnetic field by a factor $\sim 100 - 1000$ [74]. As a result of the measurements of the spin-down rates of the magnetars, it is thought that the magnetar-population is very young by astronomical standards, thus implying that the magnetic field strength has not yet had the time to decay away [57]. In Figure 1.3 we see that magnetars populate the area of high spin periods ($>$ 1s and high period derivatives ($> 10^{-14}$ second per second). Most other pulsars are observed with spin periods of a few hundred milliseconds, so this suggest that magnetars experience heavy magnetic braking just after birth [57].

Applications of pulsars

Accurate measurements of pulsar timing, which involve the measurement of the arrival time of photons emitted from the pulsar, form the basis for extracting valuable information about various scientific applications of pulsars. It is possible to form pulsar clocks from the pulsar timing that can rival atomic clocks in terms of long term stability, however pulsars are less stable over short time scales.

The locations of pulsars can be determined to extreme accuracy. This can be applied to a range of scientific and practical applications. When the Pioneer and Voyager probes where sent out into outer space there were plaques included to show the position of the Sun relative 14 pulsars. These 14 pulsars were are depicted on the plaques showing both their positions and periods. As the pulse-period in pulsars evolve in time it would thus be possible for an extra-terrestrial civilization to locate the solar system, but also the approximate epoch of when the probes were launched [69]. Another application of the pin-point locations of pulsars can be the use of X-ray pulsars as a navigation system. As mentioned, the timing stabilities of pulsars can be compared to atomic clocks, and when comparing the spacecraft arrival times of pulses from well known pulsars to arrival times at well known reference points, one can triangulate the position of the spacecraft much like one triangulates the position of a car here on Earth [9] This was in fact proven as a viable theory by NASA back in 2018 [107] using the NICER-module aboard the International Space Station through the Station Explorer for X-ray Timing and Navigation Technology (SEXTANT) enhancement. Through observations of the proper motion of pulsars in a galactic cluster, we can determine the parameters for the physical properties of the cluster ([79]). It is also possible to track the motion of pulsars close to Sagittarius A* in order to learn more about how super-massive black holes warp the space-time surrounding them. Finally, as the electromagnetic emission travel towards Earth it will interact with the ionized interstellar medium. Through measuring the delay in arrival time of pulses compared to predicted arrival time it is possible to calculate the dispersion measure along the line of sight. Given independent measurements of the distance to the pulsar, we are given a way of mapping the interstellar medium.

1.2 Millisecond pulsars

Millisecond pulsars (MSPs) are a subclass of pulsars that are characterized by their extremely short spin periods (P = 1.4–30 ms [10]) and low spin-down rates ($\dot{P} = 10^{-18} - 10^{-22}\,\mathrm{s\,s^{-1}}$ [10]). The first millisecond pulsar was discovered in 1982 by Backer et al. [7] with a period of 1.588 ms. Prior to these findings, the consensus was that pulsars originated from supernovae and possessed periods larger than 20ms [67]. After the initial discovery, it was suggested that these MSPs were formed through a process called "recycling", where material from a low-mass companion accretes onto a slowly rotating, old neutron star (as discussed in Section 1.1.4). The transfer of angular momentum during this event has the potential to theoretically increase the rotation rate of the pulsar to an order of $\sim$ 100 Hz. This process has also made the millisecond pulsars more massive than younger neutron stars [10].

Looking at the millisecond pulsars in our galaxy, it is possible to divide the population into two groups based on location; the globular cluster population and the Galactic field population (204 MSPs in globular clusters, and 386 in the Galactic field [61]).

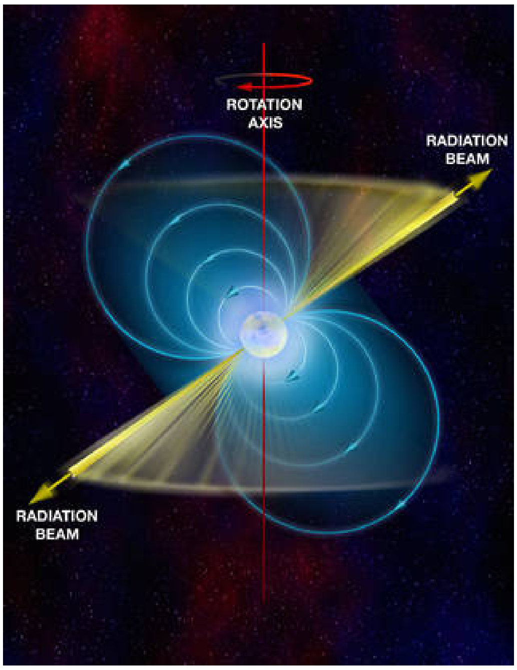


Figure 1.4: Depiction of a rotating neutron star beaming electromagnetic radiation from its magnetic poles that sweep the sky like a lighthouse. Credit: Bill Saxton, NRAO/AUI/NSF

Millisecond pulsars have been observed in multiple different wavelengths such as gamma ray, X-ray and radio. It is also possible to discover MSPs in the optical depending on the nature of the system. About 80% of all MSPs are found in binary systems [10], with the majority of them believed to have white dwarf companions [66]. However, we have observed both main-sequence and brown dwarf counterparts, and these systems are often subjected to optical observations. Through the use of spectrospcopy and photometry, we can determine the properties of the binary system, such as the mass of the neutron star and companion [93], the orbital parameters [93, 30], pulsar wind interactions

[73] and the nature of the companion star.

1.2.1 Spiders

Spiders are a part of the compact binary millisecond pulsar population, where the pulsar wind is strong enough to ablate the companion, or eject the transferred matter from the companion. There are two groups of spiders; redbacks and black widows, aptly named after spider species where the females eat their partners. Spiders have very energetic millisecond pulsars with spin-down luminosities of $\dot{E} \sim 10^{34} - 10^{35}$ erg s^{-1}. Both spiders have similar pulsars, the difference lies in the nature of their companions. Black widows have low-mass brown-dwarf companions of $M_c \sim 0.01 M_\odot$ [14], whereas the companions of redbacks are main sequence stars, of $\sim 0.2 M_\odot$ [14]. Both spiders have compact orbits with typical orbital periods of less than a day, and currently there are more than 40 known black widow systems and about 20 redbacks in the Galactic field [10].

Spiders usually display irregular eclipses of their radio signals as a result of matter being ejected into the surrounding space by the pulsar wind [14]. The matter is ejected due to the intensity of the pulsar wind in spider MSP [10]. The study of these eclipses is crucial for understanding how the irradiated companion lose mass, the characteristics of the medium causing the eclipses and understanding the interactions between the pulsar wind and the eclipse medium [81]. The continuous ablation of the stellar matter is thought to be the explanation for why there exists isolated MSPs. It is thought that some black widow systems could completely erode away the star, which explains how the isolated MSPs appear to have evolved in binary systems [103].

Since the majority of recently discovered spider systems, particularly redbacks which have brighter companion stars, are situated far from the Galactic plane where interstellar extinction is minimal, optical telescopes have excellent access to study them [63]. These optical observations of spiders gives insight in the properties of the companion star and the dynamics of the pulsar wind. Given the non-thermal nature of the MSP emission, the majority of the optical light detected originates from the companion star.

When observing compact binary MSPs in the optical band, we expect orbital modulation in the light curves. The light curves will often display one of two different modulations depending on the physical parameters of the system; ellipsoidal modulation with two peaks per orbital phase or a sinusoidal modulation with one peak per orbital phase [10]. For an non-irradiated companion, we expect there to be two flux minima per orbital phase due to the non-spherical shape of the companion star [14]. These occur when the companion is in quadrature with the neutron star, and the companion is deformed due to tidal distortion. If the companion is irradiated, the apparent flux will be at a maximum when the companion is in superior conjunction with the neutron star, and minimum when in inferior conjunction. In order to observe the orbital modulation, we need to observe systems where the orbital inclination angle of the system is such that we do not see the system from above. An orbital inclination angle of $i = 0°$ would yield a flat light curve with no modulations, whereas $i = 90°$ would yield the largest amplitude modulation.

1.2.2 Intra-binary shocks

The optical light curves of redbacks and black widows often show evidence of companion irradiation. In the past, it was believed that the irradiation of the companion star resulted from direct isotropic "photon" heating. However, with the increase in spider population over the past decade, we have discovered several bright, nearby systems providing high quality optical data on the irradiation of the companion. It became clear that the direct heating models could not explain all the observed phenomena, such as asymmetric light curves, color patterns that do not show evidence of direct heating and irradiation luminosities larger than the spin-down luminosity of the pulsar [86].

There are two theories that account for indirect irradiation of the companion; intra-binary shocks and the magnetic field of the companion channeling the pulsar wind onto parts of the surface [86]. The intra-binary shock assumes heating through the collision of the isotropic stellar wind and the pulsar wind [10]. The relativistic pulsar wind is most likely not isotropic, but rather equatorially concentrated [97]. Prompt post-shock pair-production reprocesses the pulsar radiation and heats the companion [86]. The shocked pulsar wind produces synchotron emission, which can be observed as a X-ray modulation, and gamma rays are expected to be produced as a result of inverse compton scattering.

The geometry of the intra-binary shock could affect the observed light curves. This geometry may vary depending on factors such as the strength of the companion wind and the orbital velocity of the companion. The pulsar wind power is factored in due to the companion wind being driven by the pulsar irradiation. In 2016, Romani & Sanchez [86] modelled the intra-binary shock geometry, and the results can be seen in Figure 1.5. Here we can see that the orbital sweep back for low v_{rel}, meaning an intra-binary shock close to the companion, can lead to the shock wrapping around the companion. Radiation would then heat the companions trailing side, leading to asymmetric light curves (as can be seen in the right part of Figure 1.5).

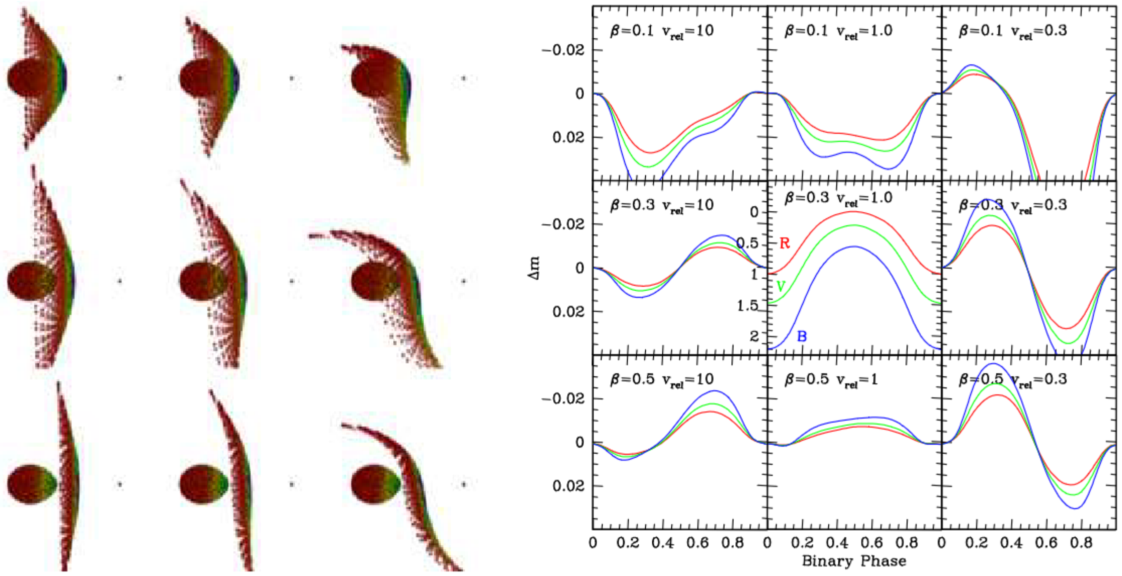


Figure 1.5: The results of modelling the intra-binary shock geometry in Romani & Sanchez 2016 [86]. Here β is the wind momentum flux ratio and v_{rel} is the ratio between the companion wind speed and the orbital velocity. Left: The geometry of the intra-binary shock for different values of β and v_{rel}. Right: Shows the resulting light curve for the corresponding intra-binary shock geometry.

1.3 PSR J1048+2339

The focus of this thesis is on one particular redback millisecond pulsar, PSR J1048+2339 (hereinafter referred to as J1048). This binary system was discovered by [28] during a 327 MHz search of unidentified *Fermi* sources to be included in the 3FGL *Fermi* LAT catalog. It was discovered when targeting the *Fermi* source 3FGL J1048.6+2338. From radio timing and multiwavelength observations, J1048 was found by Deneva et al. 2016 [30] to have a spin-period of $P = 4.66$ms, an orbital period of 6h and a spin-down luminosity of $\dot{E} = 1.2 \times 10^{34}$ erg s^{-1} [108]. Given a dispersion measure of $DM = 16.6543$ pc cm^{-3}, this places the system at a distance of 1.7kpc from Earth [31]. The systematic velocity is $\gamma = -24(8)$kms^{-1} and the secondary semi-amplitude $K_2 = 376(14)$kms^{-1}, where the numbers in the parentheses is the 1σ uncertainty on the values [93]. Using the values of P and K_2 it makes it possible to find the mass function of the system. Strader et al. (2019) [93] make use of this and the companion mass function from Deneva et al. (2016) [30] and find the minimum mass of the neutron star to be $M_{NS} = 1.96(22)M_{\odot}$ and companion mass $M_c = 0.38(4)M_{\odot}$.

Yap et al. (2019) [108] preformed observations of J1048 in March and April 2018 with the Lulin 1m telescope (LOT) and the 2m Liverpool Telescope. Figure 1.6 depicts the light curves from the observations. The first two nights of observations show a clear ellipsoidal modulation of the light curve with two maxima per orbital phase, which is expected for a non-irradiated system with a Roche-lobe filling companion. The next run of observations with the LOT showed an increase in the received flux from J1048, corresponding to a magnitude change of about 1 in the *r'*-band and 0.6 in the *g'*-band. The light curves then exhibited only one maximum per orbital phase, indicating that the system was then dominated by pulsar wind heating. For direct pulsar irradiation we expect the peak to be at superior conjunction, which following the convention used for these light curves is at phase $\phi = 0.75$. Yap et al. (2019) [108] found that the maximum flux is located at $\phi = 0.65$ (but they do not specify in which band this is measured).

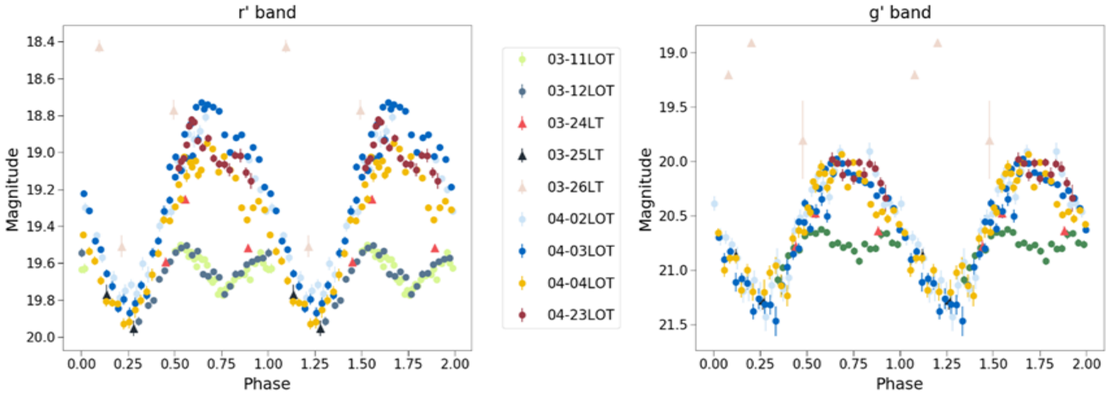


Figure 1.6: Light curves of J1048 from March and April 2018 from Yap et al. (2019) [108]. We can see a clear ellipsoidal and sinusoidal modulation as a result of the non-existence and existence of pulsar wind heating, respectively.

Observations at other wavelengths have given insight to the geometry of the system and its environment. Radio observations have shown radio eclipses around the superior conjunction, but also small eclipses at other orbital phases, indicating that the systems gaseous surroundings are dynamic and change short time scales [30]. Furthermore, in a recently published paper Clark et. al (2023) [22] detected gamma ray eclipses from J1048 for the first time. This leads to further restrictions on the minimum orbital inclination angle, as the pulsars gamma ray emissions are fully eclipsed. The minimum inclination was found to be $i^{min} = 80.4\,\text{deg}$. Finally in 2021, Zanon et al. performed multiwavelength observations of the system in order to search for a residual accretion disc or intra-binary shock emission. They observed extended emission close to the inner Lagrangian point, thought to originate from an intra-binary shock between the relativistic pulsar wind and the matter leaving the companion star [73].

Chapter 2

Robotic optical telescopes

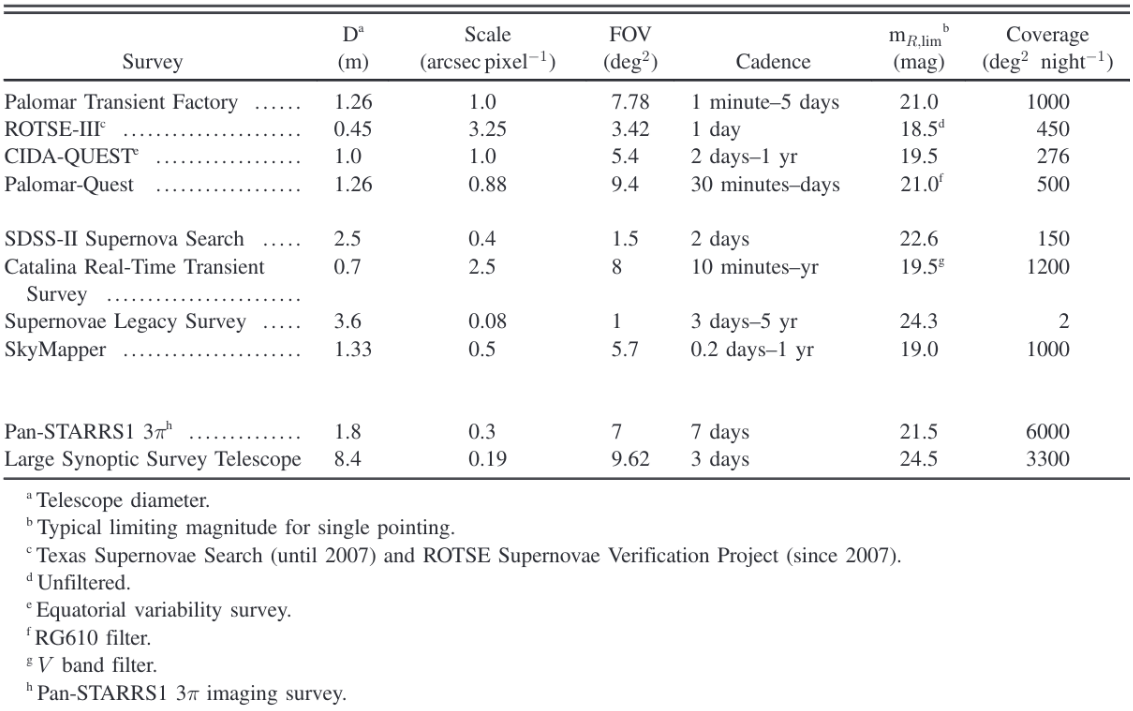
The methods of observing space from Earth have developed a lot since the first telescopes were used by Galileo to discover the moons of Jupiter [99]. Today there are numerous different types of telescopes used to observe different phenomena. Neutron stars can be observed in multiple different wavelengths, each giving new information regarding the observed systems. Most of these telescopes are required to be operated by an astronomer during the night, and as the number of detected explosive transients and highly variable objects increase, these traditional follow-up telescopes do not suffice. Thus for decades, astronomers have made use of robotic telescopes and sky surveys in order to observe these events.

2.1 Time-domain revolution

Time-domain astronomy is the study of how astronomical objects change over time. This is a branch of astronomy that has been practiced since antiquity, when the planetary motions were mapped by ancient astronomers [100]. In modern times, time-domain astronomy mainly concerns the observation and mapping of variable objects outside of our solar system. It includes many fields within astronomy, nanemly stellar evolution, supernovae, GRBs, the Solar system and cosmology.

The observable parameter space defines which physical parameters and conditions can be observed. For astronomy that can include the range of observable wavelengths, the flux limit (depth) or solid angle coverage [37]. The opening of the time-domain in astronomy leads to an extra axis in the observational parameter space. Every time technology opens new domains of exploration within a field, we are likely to discover new phenomena [47]. The study of novae, supernovae, gamma-ray bursts, and AGN in the temporal domain provides fresh ways to comprehend the intense stages of star and galaxy evolution in astrophysics. Some phenomena can only be studied in the time domain, such as accretion and various cosmic explosions, and without time-domain astronomy it would be difficult to gain any proper insight into these phenomena [35].

Even though the term time-domain astronomy is nothing new, the methodology of it has been subject to the advancements in technology. At the end of the 1800s there was a development in how observational astronomy was practiced. Photographical plates replaced the eye as the receiver in telescopes, making it possible to have longer exposures in order to capture fainter objects and to accurately log observed changes [12]. The plates also had a larger field of view due to the only limiting factors being the size of the plate of what was practical to handle. From 1890 to 1990 there were taken roughly 450,000 images currently stored in the Harvard College Observatory plate collection [45]. These provide an approximately uniform full-sky coverage in the hundred year period, and can give great insight into the changes of variables on the night sky. A project to reduce and digitize these plates was begun in 2001, with the DASCH project (Digital Access to a Sky Century @ Harvard) [45]. Even though extreme phenomena have short time-scales of hours to months, the astronomical time-scales are vastly longer. Having access to the digitized plates from the DASCH project opens up the possibility to conduct studies on a great deal of objects on



Survey	D[a] (m)	Scale (arcsec pixel^{-1})	FOV (deg^2)	Cadence	$m_{R,\mathrm{lim}}$[b] (mag)	Coverage (deg^2 night^{-1})
Palomar Transient Factory	1.26	1.0	7.78	1 minute–5 days	21.0	1000
ROTSE-III[c]	0.45	3.25	3.42	1 day	18.5[d]	450
CIDA-QUEST[e]	1.0	1.0	5.4	2 days–1 yr	19.5	276
Palomar-Quest	1.26	0.88	9.4	30 minutes–days	21.0[f]	500
SDSS-II Supernova Search	2.5	0.4	1.5	2 days	22.6	150
Catalina Real-Time Transient Survey	0.7	2.5	8	10 minutes–yr	19.5[g]	1200
Supernovae Legacy Survey	3.6	0.08	1	3 days–5 yr	24.3	2
SkyMapper	1.33	0.5	5.7	0.2 days–1 yr	19.0	1000
Pan-STARRS1 3π[h]	1.8	0.3	7	7 days	21.5	6000
Large Synoptic Survey Telescope	8.4	0.19	9.62	3 days	24.5	3300

[a] Telescope diameter.
[b] Typical limiting magnitude for single pointing.
[c] Texas Supernovae Search (until 2007) and ROTSE Supernovae Verification Project (since 2007).
[d] Unfiltered.
[e] Equatorial variability survey.
[f] RG610 filter.
[g] V band filter.
[h] Pan-STARRS1 3π imaging survey.

Table 2.1: A summary of certain sky surveys taken from Rau et al. 2009 [82].

time-scales which are longer than other modern sky surveys by a factor of 5.

In the 1990s, astronomy changed from being a data-poor science to an extremely data-rich science. This change was mainly led by the introduction of digital sky surveys [36]. The amount of data produced is expected to double every $\sim 1-2$ years following Moore's law, reflecting the development of technology that produces the data [35]. Digital sky surveys are now the biggest data producers in astronomy, and open the time-domain for systematic exploration through full sky coverage and a cadence of a few days [34] (Table 2.1). The upcoming Large Synoptic Survey Telescope (LSST), which is expected to have first scientific light in august 2024, will alone produce 15 TB worth of data each night [55].

With the large amounts of data obtained by the wide-field synoptic surveys, they are able to contribute greatly to the discovery and understanding of optical transient events. In order to support this it is required to have good cyber-infrastructure consisting of data processing pipelines, archiving, automatic transient event classification, distribution of event-notifications and more. Due to the fleeting nature of transient events it is important that the scientific community is notified in real time of the type of transient event in order to increase the likelihood of a necessary follow-up using higher cadence telescopes [35].

The methods of time-domain astronomy will be further improved upon in the coming decades, with upgrades to the existing surveys and construction of new ones. These days surveys produce tens to hundreds observational alerts for transient events per night. With the constuction of the LSST, which will survey the entire southern sky with a cadence of 3 nights on average, there is an expectancy that it will produce 10^6 observational alerts each night ([24]). Therefore rapid event-classification may be quite important as we peer into the future of time-domain astronomy, in order to decide which events are worthy of the available follow-up resources. This is a challenging task due to most optical transients looking the same when discovered - a change in received flux in a star-like object ([35]). However, through multi-wavelength observations of the night sky it will be easier to quickly determine the class of optical transient. The upcoming Square Kilometer Array, which is a radio telescope, or the Fermi Gamma-ray Space Telescope are examples of telescopes which may help quickly identifying the nature of an optical transient, by finding counterparts at other wavelengths.

2.2 Optical transients

Optical transients occur in the optical wavelength range and exhibit sudden, temporary changes in brightness. These transient phenomena can originate from a diverse range of astronomical sources, including supernovae, gamma-ray bursts, tidal disruption events, and stellar flares. They provide invaluable opportunities for astronomers to study dynamic processes and understand the underlying physical mechanisms at play in the Universe. By carefully observing and characterizing optical transients, scientists can unravel crucial information about the nature of celestial objects, their evolution, and the dynamic processes shaping our cosmic environment. As discussed in the previous section, advancements in observational techniques, such as wide-field surveys and rapid response networks, have greatly facilitated the detection and monitoring of these transient events, enabling us to explore the transient sky more comprehensively than ever before. In the following section we will cover some of these interesting transient phenomena.

2.2.1 Supernovae

Supernovae (SN) are cosmic explosions that mark the cataclysmic deaths of massive stars, releasing an immense amount of energy ($\sim 10^{51}$ erg) and creating some of the most luminous transient objects in the Universe. Studying SNe is crucial for a multitude of reasons. Supernovae provide a unique window into the life cycles of stars and the processes that govern stellar evolution. They shed light on the formation of compact objects like neutron stars and black holes, as well as the mechanisms responsible for the synthesis and dispersal of heavy elements essential for the formation of planets and the emergence of life. The predominant contemporary application involves the use of Type Ia SNe as standard candles to gauge cosmic distances accurately [24].. From redshift and distance measurements using Type Ia SNe, it was discovered that the expansion of the Universe is accelerating. [85].

Type Ia supernovae are thought to be the product of white dwarfs increasing their mass to the Chandrasekhar limit [13]. This requires the white dwarf to be in a binary system, however the nature of the companion is uncertain. Among the numerous intriguing proposed scenarios, two emerge as particularly noteworthy. The single degenerate model proposes that the companion is a hydrogen-rich companion where the mass transfer occurs through Roche-lobe overflow [13] and the white dwarf accretes mass until it reaches the threshold for thermonuclear ignition [88]. The double degenerate model proposes that the companion is also a white dwarf. The merging of the two objects will occur as a consequence of angular momentum losses through the emission of gravitational waves. The resulting mass from the merger may be close to the Chandrasekhar mass threshold, leading to a supernova event [84]. With the introduction of the LSST, meaning more detected transient events every night, and better transient classification, it will hopefully be possible to further constrain the nature of these progenitors.

The study of SNe through sky surveys and prompt follow-up observations holds significant importance in determining the specific classification of the SNe. Optical spectroscopy of the events at the time of peak brightness makes it possible to separate the SNe into types Ia, Ib, Ic and II ([92]). The observed differences in the spectra and light curves of SNe can be attributed to the size of the star's envelope and the extent to which it was shed before the explosion occurred ([24]). These differences can be linked to the different physical parameters of the progenitors. The distinguishing factors are the presence or absence of hydrogen in the spectra. Spectra of type I SNe do not display any hydrogen features, and type Ib and Ic differ from Ia due to a notable Si absorption line for type Ia. The lack of hydrogen lines is attributed to the loss of the hydrogen envelope in the progenitor previous to the explosion ([24]). The presence of hydrogen features is evidence of a type II SN, and together with type Ib/c they constitute the SNe which are the result of a core-collapse (covered in section 1.1.1). The examination of core-collapse SNe is crucial to gain insight into the concluding stages of massive-star evolution and the formation of neutron stars and black holes [83].

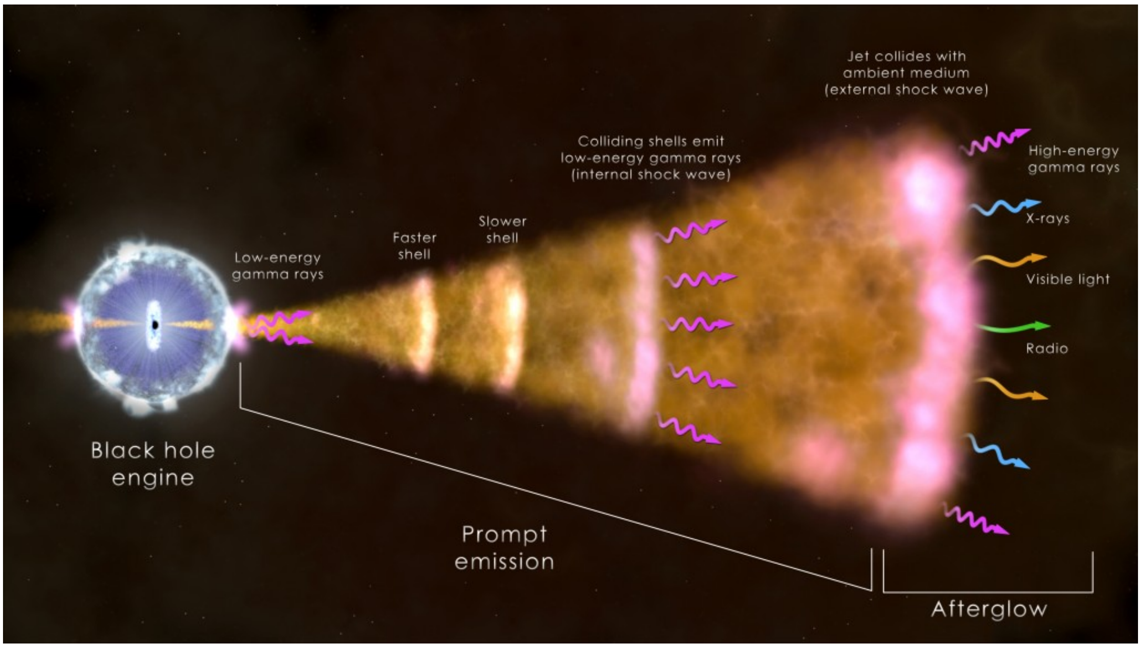


Figure 2.1: An illustration of a long gamma ray burst after the death of a massive star. Here the black hole drives a particle jet into space. Electromagnetic radiation is emitted from hot gas near the black hole, internal shock collision within the jet, and from the jets interaction with its surroundings. Credits: NASA

2.2.2 Gamma-ray bursts

Gamma-ray bursts (GRBs) are captivating astronomical events, representing the most luminous explosions known in the universe with the release of $\sim 10^{51} - 10^{53}$ ergs in a matter of seconds [80]. The origin and mechanism behind these bursts are subjects of intense research and debate within the scientific community [70]. The first GRBs were discovered in the late 60s by four separate Vela satellites [58], and have later been discovered to be distributed isotropically on the sky, pointing to an extragalactic origin [16]. On average there are a few occurances of GRBs throughout the universe per day [70] (with about one observed daily), and they can be divided into to short and long GRBs. This classification depends on the duration of the GRB, where the cut-off is at 2 seconds. Long-duration GRBs are connected to the core-collapse of massive stars leading to SNe. In fact, all observed SNe that are connected to GRBs are type Ic [24]. There is an alternative model which considers magnetars as the cause of a portion of long GRBs [102]. Magnetar progenitors can more easily explain activity post the prompt onset of the GRB, such as X-ray flares. Short GRBs are thought to come from the merger of two compact objects, like two neutron stars or a neutron star and a black hole [39]. The first evidence supporting this theory came in 2017 when LIGO detected the gravitational waves (GW 170817A) from a neutron star - neutron star merger, with consequent short-GRB detection (GRB 170817A) by Fermi-GBM [2].

In 1997 the Dutch-Italian satellite BeppoSAX discovered an x-ray counterpart of the long gamma ray burst GRB970228 [25]. BeppoSAX was able to accurately determine the position of the GRB, and shortly after the x-ray observation, an optical afterglow was identified [104]. This afterglow is thought to originate from the interactions between the ultra-relativistic jet and the surrounding environment, such as the interstellar medium [72]. It has a spectral shape covering most of the electromagnetic spectrum, from gamma-rays to radio, and can last seconds to months. The emission is found to have a temporal change in its flux, following a power-law [80]. As the jet expands into the interstellar medium, we expect there to be two shocks; one running back into the jet, and one outwards into the surrounding environment. The shocked external medium produces synchotron radiation as the electrons are accelerated [43], and this is the majority of the observed afterglow radiation (Fig 2.1). The light curves and spectrum of the afterglow are determined by the environment the jet expands into, which again is closely linked to the nature of the progenitor

[72].

2.2.3 Kilonovae

Compact binary mergers, involving at least one neutron star (NS) such as NS-NS mergers and black hole (BH)-NS mergers, are accompanied by a diverse range of electromagnetic signals spanning a wide range of wavelengths. Previous to the gravitational wave event GW 170817A, where two neutron stars merged, it was anticipated that these mergers would also have an isotropic counterpart called a kilonova [71]. Following the event, the first kilonova was observed by multiple different observatories [2], thus confirming the presence of this supernova-like phenomena. Mergers of NS-NS/NS-BH systems will eject a fraction of matter at sub-relativistic velocities [62], and these ejecta create a neutron-rich environment where it is possible for heavy, unstable nuclei to form via rapid neutron capture (r-process) [8]. The r-process occurs in extreme astrophysical environments characterized by high temperatures, high densities, and a large number of free neutrons. The unstable nuclei will begin to decay towards stablility, thus providing a heat source for the expanding envelope [62]. The transient event has a luminosity comparable to that of a supernova at about $\dot{E} \sim 10^{40} - 10^{41}\mathrm{ergs}^{-1}$, and peaks in the optical and infra-red wavelengths [96]. The lifetime of kilonovae are dependant on the abundance of the different nuclei that are created during the r-process, but it is expected to be from days to weeks [62, 8, 71].

2.3 Small robotic telescopes

A robotic telescope has the ability to function independently without any supervision during night time. Ordinary telescopes usually have an on-site astronomer to carry out the observations, but with the removal of this human interaction we reduce costs and open up a number of new possibilities. Professional optical telescopes are usually installed at high altitudes to reduce cloud coverage, light pollution, and atmospheric absorption, in order to achieve better conditions for observing [33]. With robotic telescopes one can now build telescopes that are even more isolated than before, which might lead to better observing conditions.

Robotic telescopes are important tools in the era of time-domain astronomy. One of the main advantages of having robotic telescopes over traditional manned telescopes is the reaction time to unpredictable transient events. As wide field synoptic surveys detect short-lived variations in the night sky, it is important to react quickly enough to be able to detect the transient in multiple wavelengths and with high-cadence. Being able to observe other variable objects over timescales of seconds to years with minimal human interaction and easily coordinating concurrent observations with other robotic telescopes, both ground- and space-based, are great benefits of robotic telescopes [24]. As the data on transients increase the need for fully automatic observational techniques and reduction methods increase as well [95].

A global network consisting of robotic telescopes can improve the follow-up of transient events. It is a complex endeavour as only subsets of telescopes are able to cover certain events. This is due to visibility constraints or weather blocking the view of the transient. It will be necessary to improve upon the early classification of such events in order to determine which telescopes are better for following it up [11]. An example would be a gamma-ray burst (GRB), where the photon rate in the afterglow of the GRB scales as t^{-1}. From experience we known that the slew time of telescopes scales roughly to the mirror diameter D [11]. This yields a photon rate proportional to D^{-1}. As the detection area of a telescopes scales as D^2, this results in the amount of photons detected being proportional to D. Thus, for a GRB, it would be important to direct the larger, slower telescopes to the event rather than the smaller, faster ones.

As the detected number of transient events increase, it will become increasingly important to distinguish which events are worth following up. The reason being that some events will turn out to be relatively common. A useful approach would involve acquiring prior information on various telescope and instrument combinations. This information can aid in determining the probability of a given instrument setup's ability to differentiate between potential classifications of a transient

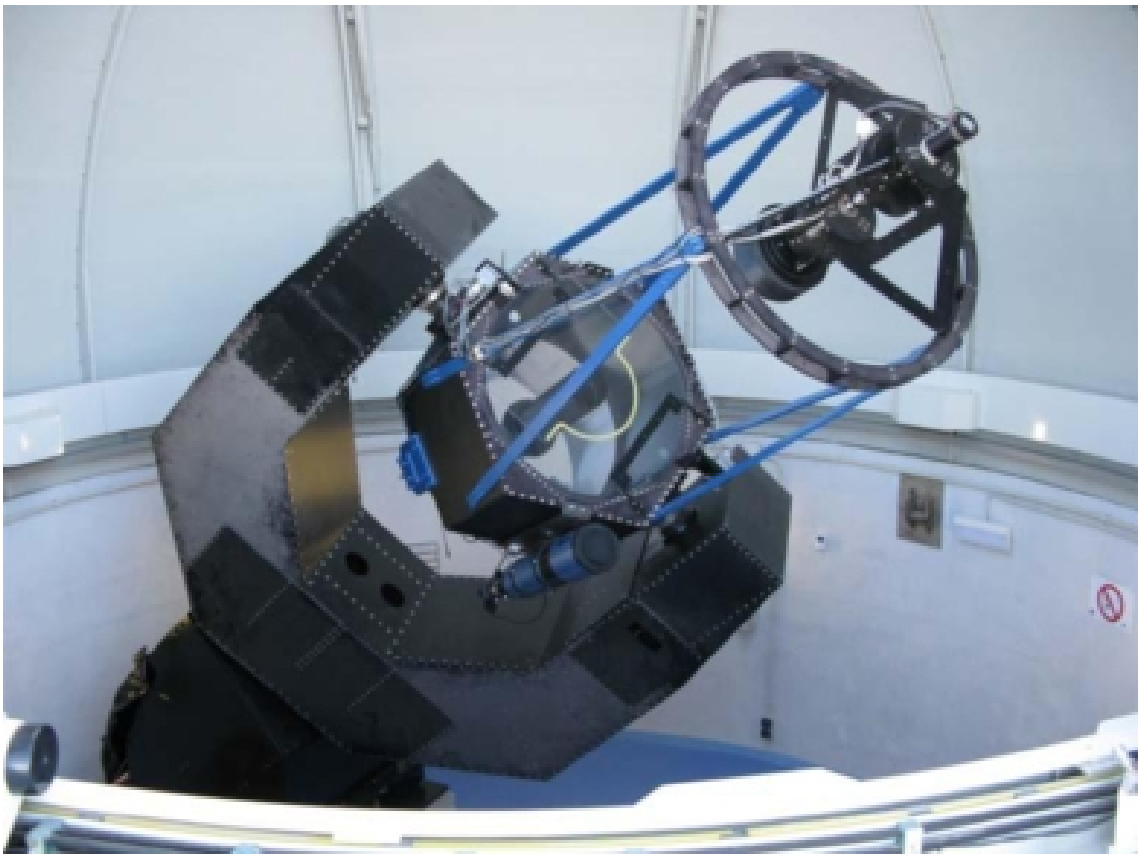


Figure 2.2: The Joan Oró Telescope as seen mounted inside the dome. It is mounted with an equatorial fork mount which can point anywhere that is 5 degrees above the horizon. Image: [52]

event through observation [11]. This will be increasingly important when the LSST sees first scientific light, as it will scan the entire southern sky every 3 nights and will release 10^6 observational alerts per night [24].

2.4 The Joan Oró Telescope

The Joan Oró Telescope (hereafter referred to as TJO from the catalan name "Telescopi Joan Oró") is a 0.8-m diameter telescope which is installed at the Montsec Observatory in the Catalan Pre-Pyrenees [40]. It is mounted on an equatorial fork mount which can point anywhere on the night sky, that is 5 degrees above the horizon, in less than 30 seconds (see Fig. 2.2). It is the largest telescope in Catalonia, and the development of the telescope and its infrastructure was a result of the collaborative endeavors of Professor Joan Oró and a group of amateur astronomers committed to promoting science in Catalonia [101]. The telescope is able to operate completely unattended. This is a result of a hard requirement of robotic operations as the site is quite isolated and the specific needs for the science cases [53]. The observers deliver proposals for targets to observe. If they are given access to the telescope it is necessary to enter the targets, instrument set-up and any constraints, such as weather, into an online scheduler. The telescope is operating on a software which analyzes all the current active proposals and automatically produces a plan for the observations each night. Once the observations are taken, they get uploaded into an online database ready for the observer to download.

The TJO conducts a variety of observations with diverse scientific goals. Being able to provide high-cadence and continuous observations, but also being able to rapidly react to transient events, are both grounds for why the TJOs main scientific endeavours is in the field of time domain astronomy ([53]). The TJO requires flexibility in the scheduling for each night in order to properly react to any time-critical events such as GRBs, novae, supernovae and others.

The TJO has a 0.8 m primary mirror which is mounted with a focal length of F/9.6 ([52]). The telescope has no active guiding available, however the tracking of the telescope is accurate for 5 minute exposures within 1 arcsecond. The general pointing accuracy of the telescope is better than 1 arcmin for nearly all directions, however it is worse for low elevation. The main instrument of the telescope is mounted in the Ritchey-Chrétien focus. The telescope itself is covered by a 6.15-m diameter dome.

The main instrument that is mounted in the TJO, as of december 2018, is LAIA ([50]). It is an optical imager that consists of two parts; the charge-coupled device (CCD) and the filter wheel. The filter wheel for LAIA can hold a total of 12 separate filters, but as of writing there are 10 installed. These are photometric Johnson-Cousin filters for the U-, B-, V-, Rc- and Ic-band, as well as the newly installed SDSS-filters g', r', i' and z', as well as an H_α-filter [51]. LAIA is created to cover a field of view of 27.3 x 27.3 arcminutes, however the focal plane of the TJO has a circular field of view of 30 arcminutes. This, as well as the pick-off mirror for TJOs spectrographic instrument ARES, are sources for vignetting in the final image created (seen in Fig. 2.3). The data collected by the TJO is automatically reduced by the telescope using same night blanks and flat field images.

2.5 TJO-pipeline

The TJO-pipeline is software written in Python 3.10 that has been developed as a part of this master thesis. The goal of the software is to streamline the data collection from the TJO. Due to the nature of the scheduler of the TJO, it is difficult to know when there are new observations in one of your active proposals. Therefore, previous to the TJO-pipeline the data collection consisted of checking the database if new data had been uploaded, and then if there were new data, download and unzip them into into your preferred directory. Further, the images downloaded would suffer from the vignetting mentioned in section 2.4. The main feature of the TJO-pipeline is to automate the actions necessary in order to have the data ready for further processing.

The TJO uploads three files each night it performs observations on science targets belonging to one of your proposals. The filenames and contents are as follows (with YYYYMMDD being the date of the observations):

1. YYYYMMDD_catalog.tar.gz
   (a) YYYYMMDD_rawdata.txt containing information on the rawdata (time of observation, elevation, etc.).
   (b) YYYYMMDD_reddata.txt containing information on the reduction process, number of stars and FWHM for each reduced file.
   (c) .dat file for each reduced file containing information on every detected object.
2. YYYYMMDD_rawdata.tar.gz
   (a) Non-reduced fits files.
3. YYYYMMDD_reddata.tar.gz
   (a) Reduced fits files.

The TJO-pipeline consists of five parts; DOWNLOAD, CHARACTERIZE, ANALYZE, TJOPIPELINE and CONFIG. Each of these parts, excluding CONFIG, is programmed in order to run in three different modes, date, full proposal and archive mode. The date mode is used to run the scripts on a specific date in a specific proposal where you have observations. Full proposal will perform the same tasks as in date mode, but for every date where there are observations within a proposal. Archive mode works similarly to full proposal mode, but is set up in order to be run through cron-job, whereas

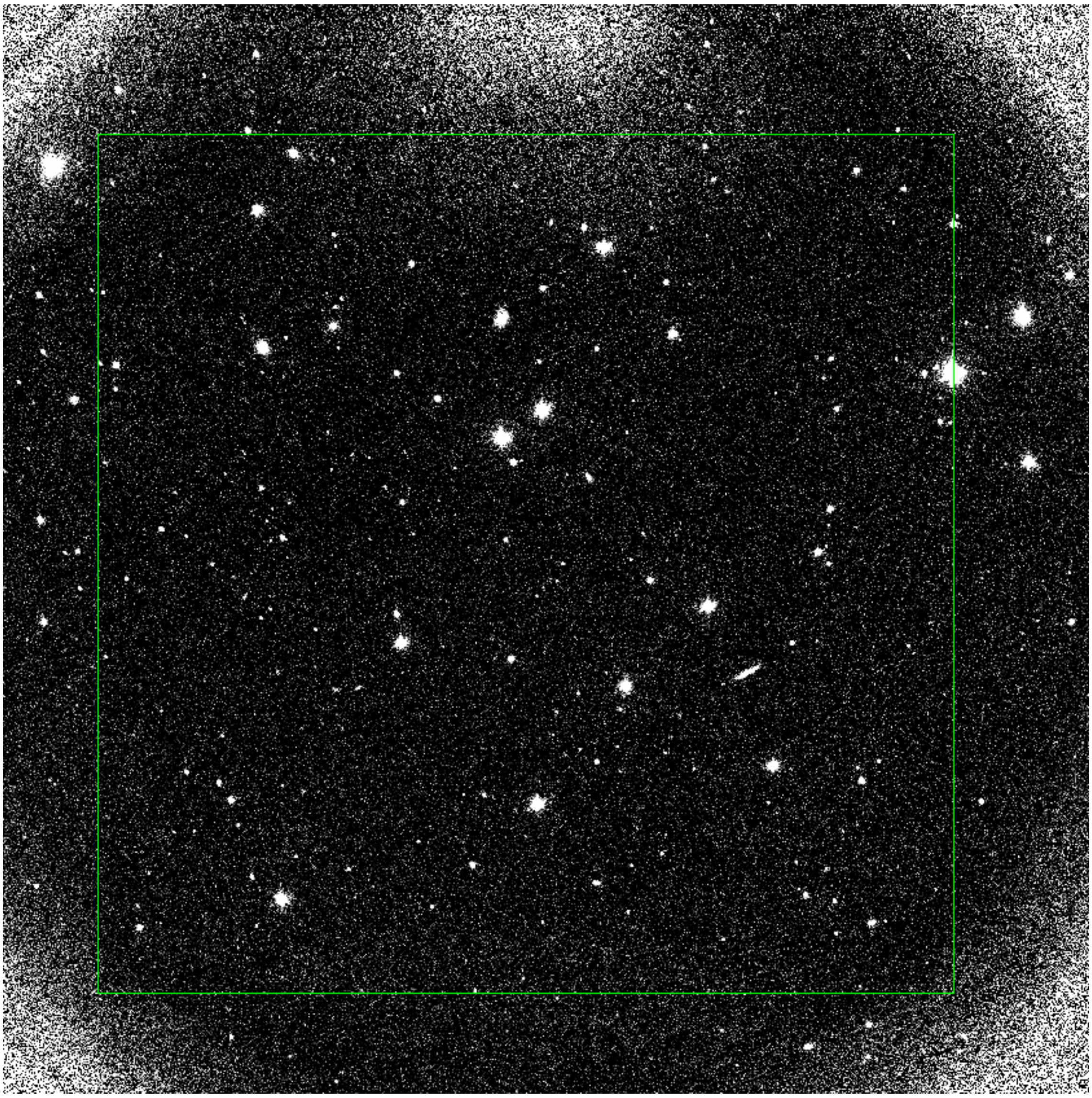


Figure 2.3: A screenshot of one of the images in DS9 showcasing the vignetting and the area that the images are trimmed to. The corners are vignetted due to the size of the TJO focal plane being smaller than the size of the CCD, and the additional source of vignetting seen in the top middle is due to the ARES pick-off mirror.

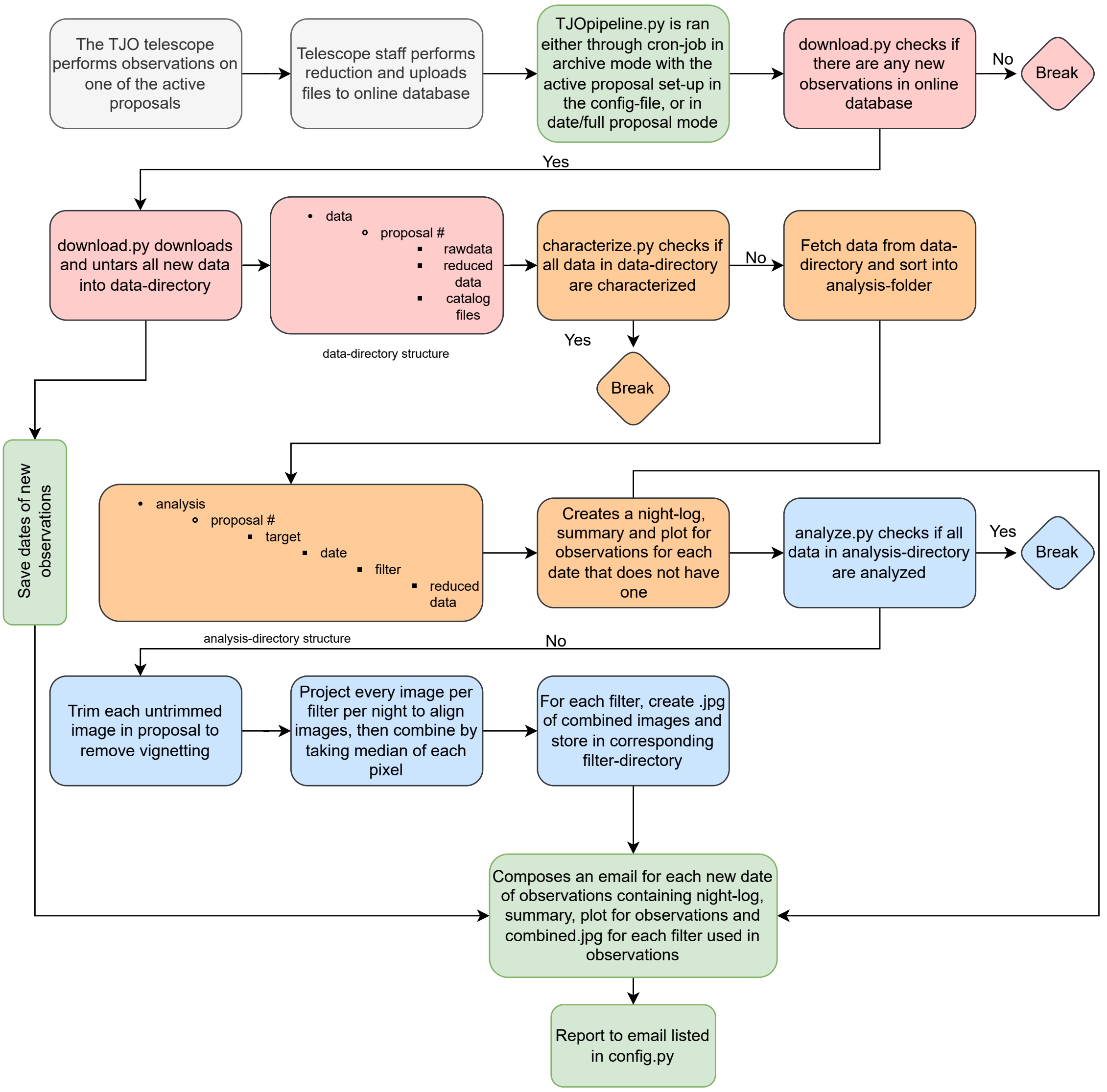
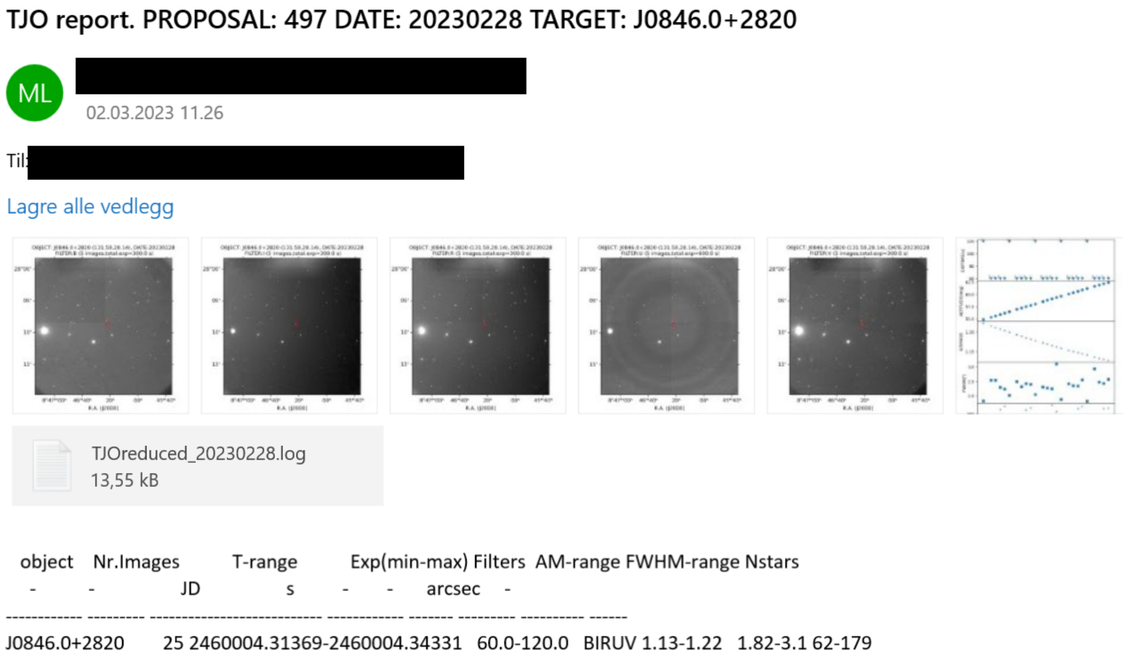

The TJO telescope performs observations on one of the active proposals
Telescope staff performs reduction and uploads files to online database
TJOpipeline.py is ran either through cron-job in archive mode with the active proposal set-up in the config-file, or in date/full proposal mode
download.py checks if there are any new observations in online database
No
Break
Yes
download.py downloads and untars all new data into data-directory
data
proposal #
rawdata
reduced data
catalog files
data-directory structure
characterize.py checks if all data in data-directory are characterized
No
Fetch data from data-directory and sort into analysis-folder
Yes
Break
Save dates of new observations
analysis
proposal #
target
date
filter
reduced data
analysis-directory structure
Creates a night-log, summary and plot for each observations for each date that does not have one
analyze.py checks if all data in analysis-directory are analyzed
Yes
Break
No
Trim each untrimmed image in proposal to remove vignetting
Project every image per filter per night to align images, then combine by taking median of each pixel
For each filter, create .jpg of combined images and store in corresponding filter-directory
Composes an email for each new date of observations containing night-log, summary, plot for observations and combined.jpg for each filter used in observations
Report to email listed in config.py

Figure 2.4: A flowchart showing the inner workings of the TJO-pipeline. The chart shows the steps from observations done by the TJO to the products of the pipeline being reported to the user. The individual parts of the pipeline are color-coded.

date and full proposal mode works through user inputs. In order to run the pipeline, you need all parts of the software in an empty directory you wish to download your data to and to fill out the necessary details in CONFIG. The pipeline will create any directory necessary for the software to function. Then the pipeline will be ready to run through the command line using either of these calls; *python TJOpipeline.py -m x*, with *x* being the mode you want to run it in, *d* (date), *f* (full proposal) or *a* (archive).

2.5.1 TJOpipeline and config

The TJOPIPELINE script is a wrapper that runs through all the integral parts of the pipeline, DOWNLOAD, CHARACTERIZE, and ANALYZE, in this order. The pipeline will run if there are any new observations done with the telescope or if it detects images that have not undergone the full data processing within the pipeline. Given there are new observations, and the pipeline is done running through the separate scripts, TJOPIPELINE will compose an email with a summary of the information that is created throughout the script, and send it to the user. An example of this email can be seen in Figure 2.5. The script can be run through a command line call as seen in the



TJO report. PROPOSAL: 497 DATE: 20230228 TARGET: J0846.0+2820

ML [redacted]

02.03.2023 11.26

Til: [redacted]

Lagre alle vedlegg

TJOreduced_20230228.log
13,55 kB

```
 object   Nr.Images       T-range          Exp(min-max) Filters  AM-range FWHM-range Nstars
   -          -              JD            s         -      -       arcsec    -
------------ --------- --------------------------- ------------ ------- --------- ---------- ------
J0846.0+2820      25 2460004.31369-2460004.34331   60.0-120.0   BIRUV 1.13-1.22   1.82-3.1 62-179
```

Figure 2.5: An example of an email reported by the pipeline. The email provides a short summary of the nights observations, a .log file containing detailed information on each image taken, quick look images for the filters used, and a plot that visualizes the observations. Examples of the plot of observations and the combined images can be seen in Figures 2.7 and 2.8.

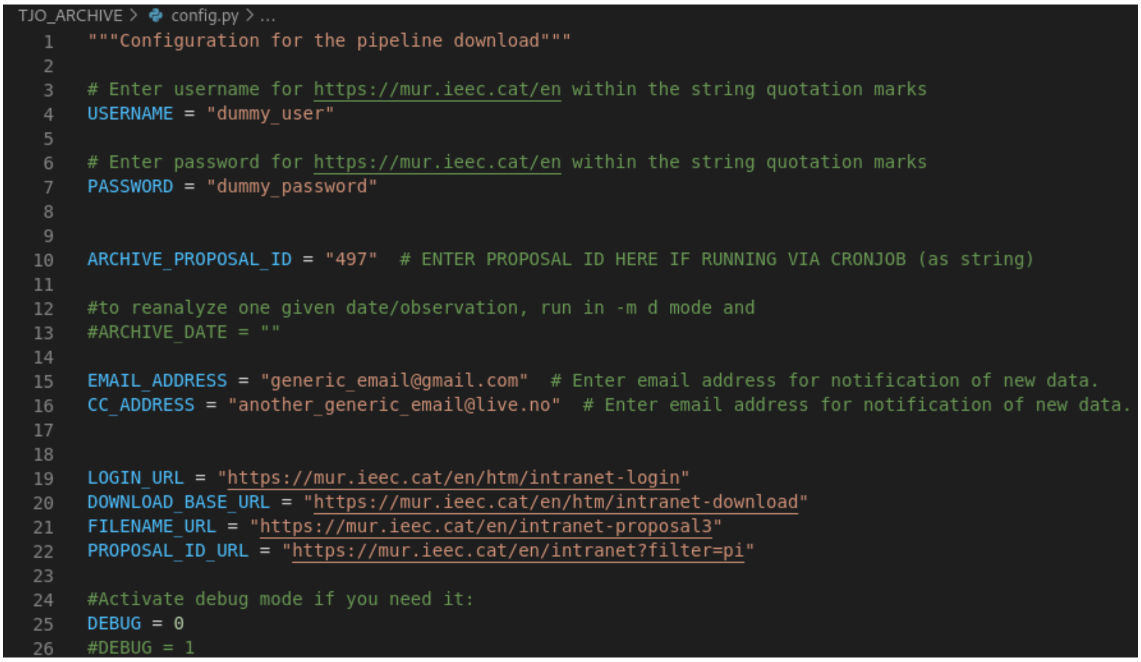
previous paragraph.

The CONFIG file is necessary in order to run the pipeline efficiently, and can be seen in Figure 2.6. It contains the necessary URLs for the online database that the other parts of the pipeline requires to complete their tasks. Users will have to change PROPOSAL_ID_URL depending on their position in the proposal, whether they are the PI or a collaborator. In Figure 2.6, CONFIG is set-up for a PI. It is also necessary for the user to enter log-in credentials for the online database and an email address used for reporting the observations of the night. The file also contains a variable that will make the software verbose in order to more easily debug the code.

2.5.2 Download

The DOWNLOAD script's main task is to download the data uploaded by the TJO staff to the online database. The script initiates a session to maintain specific parameters throughout requests on the website, allowing seamless access to each URL without the need to send a new log-in request for every action. The online database requires log-in in order to access the data, so DOWNLOAD has to send a POST-request [20] containing the log-in credentials of the user. These credentials are taken from CONFIG. DOWNLOAD then scrapes the names of the proposals, and the dates of observations within the proposals, which the user has in the online database. This is done through a GET-request [20] and by using a parser. We then find the proposals within the element-information for the specific URL. Currently, this part of the pipeline breaks if there are changes done on the URL we access, such as changing the dimensions of the table. This will be fixed before an eventual release of the pipeline to the public. Then, depending on the choice of mode the script is run through, the user enters which proposal they wish to download data from, and the date within for the date mode. DOWNLOAD will go through the files available under a proposal, and if there are any new observations taken that are yet to be downloaded it will download these, unzip and save them in a data directory containing directories for each proposal. This directory-structure can be seen in the flowchart in Figure 2.4.

When running TJOPIPELINE in archive mode the new observations are reported in the form of an



```
"""Configuration for the pipeline download"""

# Enter username for https://mur.ieec.cat/en within the string quotation marks
USERNAME = "dummy_user"

# Enter password for https://mur.ieec.cat/en within the string quotation marks
PASSWORD = "dummy_password"


ARCHIVE_PROPOSAL_ID = "497"  # ENTER PROPOSAL ID HERE IF RUNNING VIA CRONJOB (as string)

#to reanalyze one given date/observation, run in -m d mode and
#ARCHIVE_DATE = ""

EMAIL_ADDRESS = "generic_email@gmail.com"  # Enter email address for notification of new data.
CC_ADDRESS = "another_generic_email@live.no"  # Enter email address for notification of new data.


LOGIN_URL = "https://mur.ieec.cat/en/htm/intranet-login"
DOWNLOAD_BASE_URL = "https://mur.ieec.cat/en/htm/intranet-download"
FILENAME_URL = "https://mur.ieec.cat/en/intranet-proposal3"
PROPOSAL_ID_URL = "https://mur.ieec.cat/en/intranet?filter=pi"

#Activate debug mode if you need it:
DEBUG = 0
#DEBUG = 1
```

Figure 2.6: A screenshot of the config.py file. The file consists of all the necessary user information for the pipeline to run. The log-in credentials are necessary in order to download the data from the online database, ARCHIVE_PROPOSAL_ID is for the pipeline is running in archive mode and EMAIL_ADDRESS is to correctly report the results to the user.

email to the user. DOWNLOAD will compare the data available in the online database with what is already in the data directory. If there are any new observations, it will save the dates in an array and return this array when it has completed the tasks listed above.

2.5.3 Characterize

CHARACTERIZE is a script that characterizes the TJO data that has been downloaded from the online database. The script will find and save in log files the targets, filters, exposure times and the time of observations, and save these in a directory named night_logs. It also sorts the data into a structure of proposal, target, date and filters for that date. This data structure can be seen in Figure 2.4. Further characterization happens as the script creates two .log files; one summary of the observations of the night, and one containing detailed information regarding each science image collected. It will also create a plot that gives information on the observations such as; exposure time, altitude, airmass, FWHM and number of stars. An example of this plot can be seen in Figure 2.7.

The observations will be available in the online database for a year, and after that they will be made public and removed from the database. To access the data after this it is necessary to request the staff at the telescope to upload the observations. The data would then be available for downloading, however there would be no catalog file. The catalog file contains all the information that would be added to the summary and detailed .log file. In order to create the summary and detailed .log files without the catalog it is necessary to access the headers of the FITS-files. The headers contain all the information regarding the FITS-files, and are accessed using the FITS library which is part of the *AstroPy* package and stored. The pipeline will extract all the information from the headers even if the catalog-file is available, and then append the extra information available in the catalog files.

2.5.4 Analyze

The final part of the pipeline is ANALYZE. The photometric observations taken by the TJO are uploaded in both raw and reduced format, meaning that it is possible to begin differential aperture photometry as soon as the data are downloaded since the reduced files have been bias, dark and flat-field calibrated. However, in Section 3 the issue of vignetting in the images was explained. The main task of analyze is to remove this vignetting and to combine each image taken in a given filter for that night. The combined image is useful for getting a quick look at how the observations were without having to inspect each image individually, and to see if the source was detected in the different passbands (see Fig. 2.8).

To have good images to work with, the removal of the vignetting is helpful. As the vignetting is quite consistent in every image and filter, one image was visually inspected to determine what size the photometric images should be cropped to. It was found that cropping to $x = [360 : 3580]$ and $y = [380 : 3610]$ would result in a vignette-free image (see Fig. 2.3). In order to prevent images from being cropped twice, we check the size of each image and compare to the expected standard size in images taken with the LAIA instrument at the TJO. This is also to remove the possibility to crop an image which was windowed, where only part of the CCD is used.

For each night of observations, ANALYZE will combine images in every filter used that night and save the combined image as a separate FITS-file. In order to do so, the pipeline chooses the first image taken in the filter it is currently combining and projects all the other images on top of that through the wcs-coordinates in the images. This is done due to the TJO not having auto-guiding capabilities, and this results in small differences in the object's x- and y-position in the images. The projection is done using WCS_PROJECT from the *CCDproc* python library [27]. Once all images in the filter are aligned, they are combined with COMBINER from the same library. In addition, the pipeline applies the wcs-coordinates of the original image, which all others were projected onto, to the final combined image. Finally, the pipeline creates a .jpg of the combined image to be attached to the email that is reported back to the user.

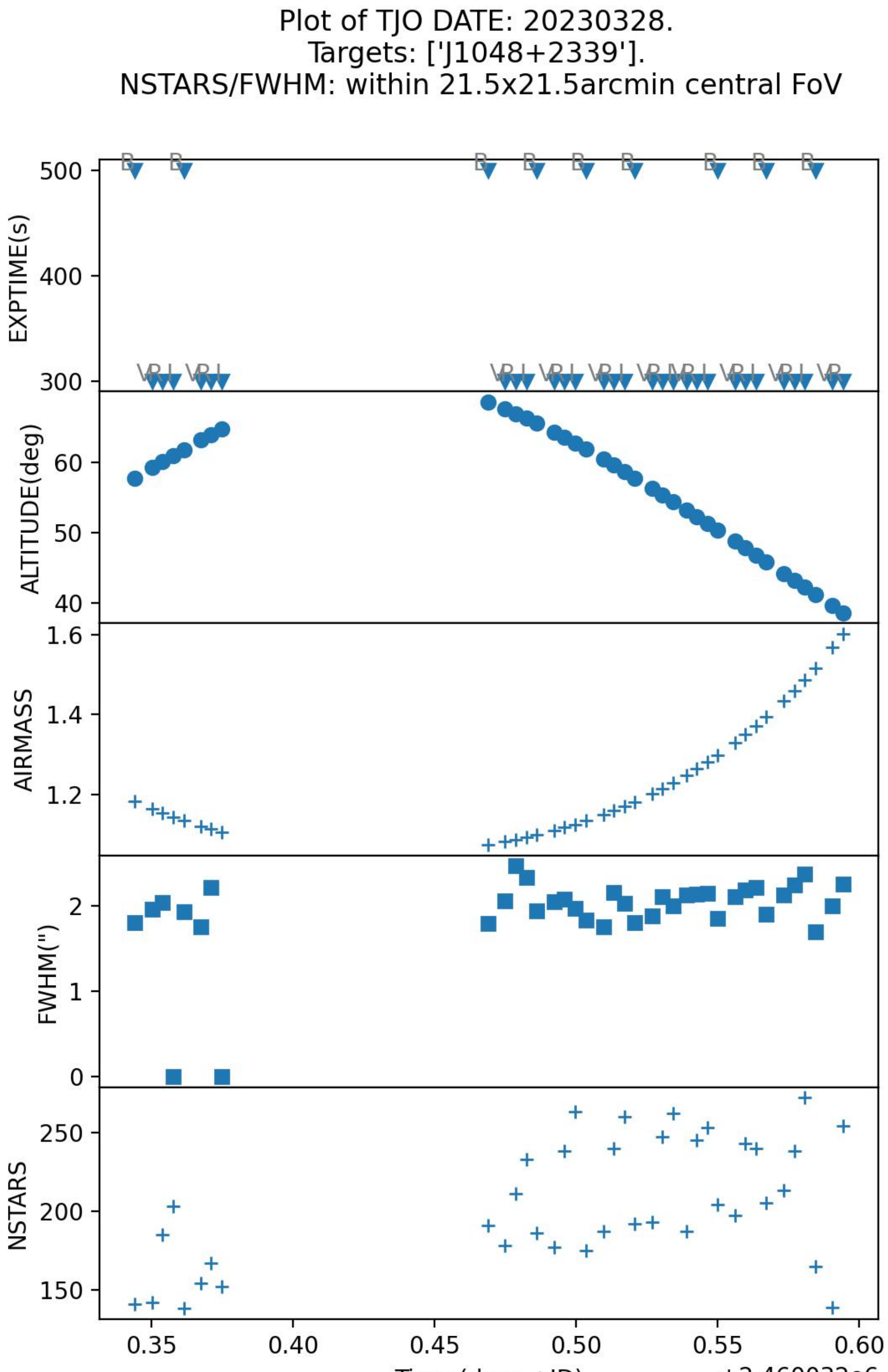


Figure 2.7: A plot created by the CHARACTERIZE part of the pipeline. The script will read out the exposure times, altitudes and airmass from the detailed information gathered in the log file, and FWHM and NSTARS from the catalog files, and plot them with the time of observation.

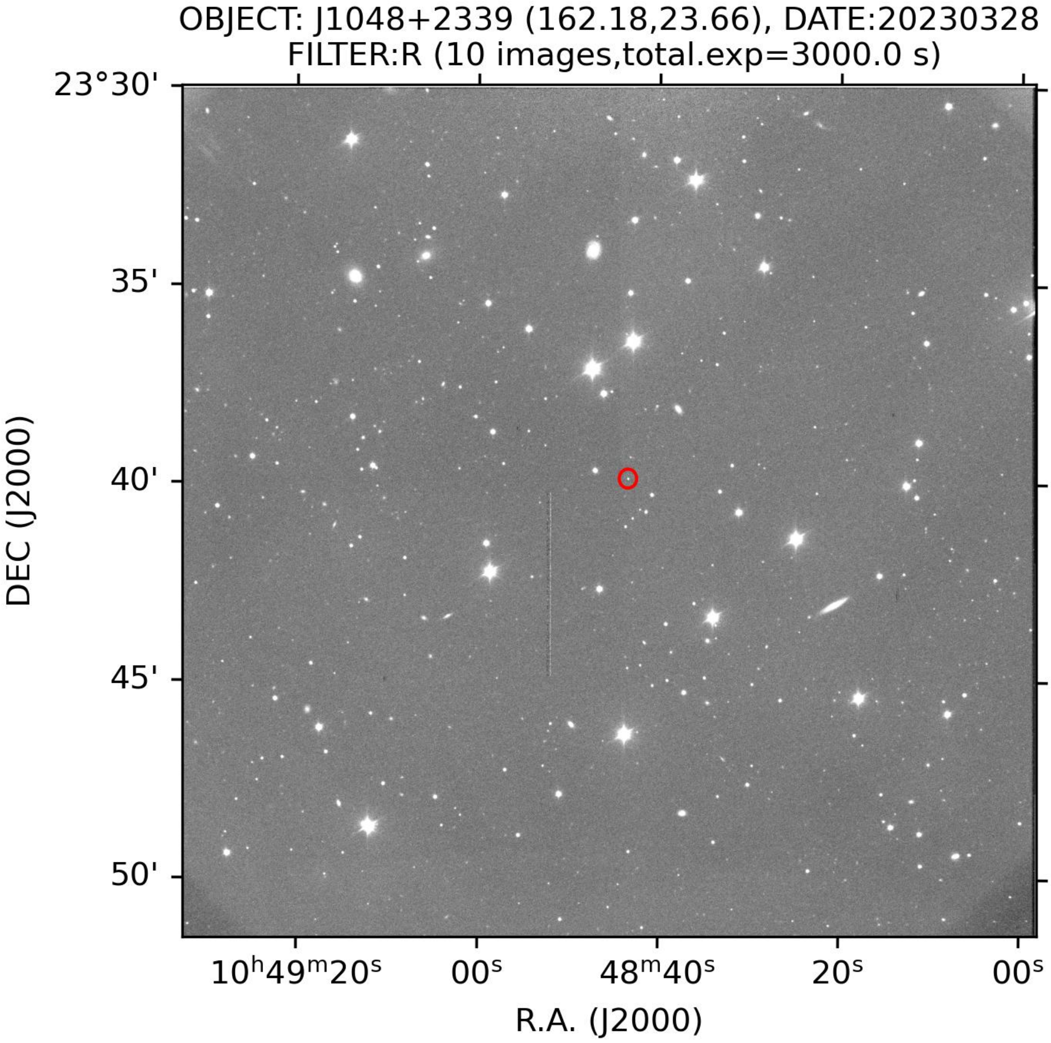

OBJECT: J1048+2339 (162.18,23.66), DATE:20230328
FILTER:R (10 images,total.exp=3000.0 s)
23°30'
35'
40'
45'
50'
DEC (J2000)
10h49m20s
00s
48m40s
20s
00s
R.A. (J2000)

Figure 2.8: An example image of how the combined FITS-file looks like. The image gives information regarding the observations, namely the filter, the amount of images in the specific filter and total exposure time. The image gives us a quick look at the quality of the observations taken that night. A red circle is added to quickly recognize the location of the target, and makes it possible to easily determine if the target was detected.

Chapter 3

Optical photometry with the Joan Oró Telescope

In this chapter, we present the data utilized for our master's thesis, which were obtained using the TJO telescope during the period from 15th of February 2020 to 8th of May 2020. We will delve into the methodology employed to acquire these data and the theoretical framework required to interpret the subsequent results.

3.1 Photometry

3.1.1 Magnitude

In astrophysics, the brightness of a celestial object is often quantified using the magnitude scale, a numerical system where lower numbers correspond to brighter objects and higher numbers indicate dimmer objects. The magnitude is a dimensionless number defined for each passband. This convention can be traced back to the ancient Greek astronomers who classified stars into six magnitude groups. In this system, stars of magnitude 1 were considered the 20 brightest, while stars at magnitude 6 were barely visible to the naked eye [54].

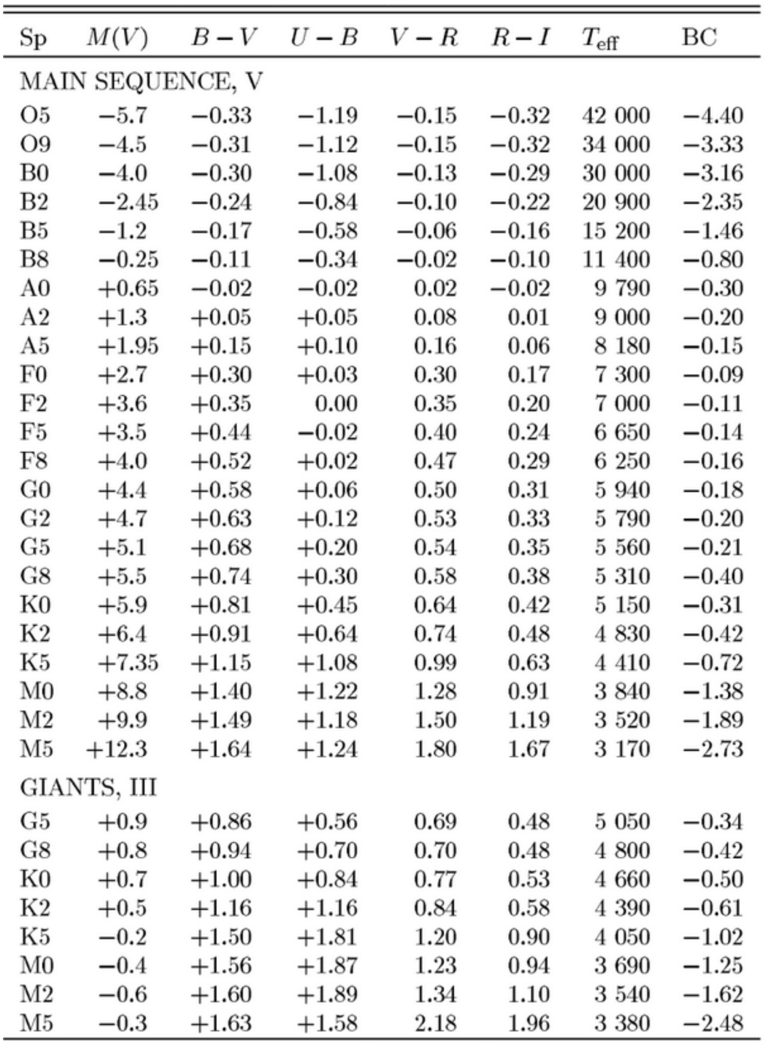
In modern times it was found through photometric measurements that the first magnitude stars were about 100 times brighter than the sixth magnitude stars. This led to the magnitude scale being logarithmic [54]. Therefore, if we assume that object 1 is 100 times brighter than object 2 ($F_1 = 100F_2$, where F is the flux of the object in units of Wm^{-2}) and that the magnitude difference is 5 we can find

$$m_1 - m_2 = 5 = -x \log_{10}\left(\frac{F_1}{F_2}\right) = -x \log_{10}(100) = -2x. \tag{3.1}$$

Here the negative sign is to ensure that the brighter object is related to the lower numerical magnitude. Therefore, to determine the magnitude m_1 of an object, we use a reference object with magnitude m_{ref}, along with the measured fluxes F_1 and F_{ref} of both objects as observed from Earth, yielding

$$m_1 - m_{ref} = -2.5 \log_{10}\left(\frac{F_1}{F_{ref}}\right). \tag{3.2}$$

Since this is the formula for the magnitude as observed from earth, it is known as the apparent magnitude. The absolute magnitude of the object is usually lower, depending on its distance from Earth.



Sp	$M(V)$	$B-V$	$U-B$	$V-R$	$R-I$	T_{eff}	BC
MAIN SEQUENCE, V							
O5	−5.7	−0.33	−1.19	−0.15	−0.32	42 000	−4.40
O9	−4.5	−0.31	−1.12	−0.15	−0.32	34 000	−3.33
B0	−4.0	−0.30	−1.08	−0.13	−0.29	30 000	−3.16
B2	−2.45	−0.24	−0.84	−0.10	−0.22	20 900	−2.35
B5	−1.2	−0.17	−0.58	−0.06	−0.16	15 200	−1.46
B8	−0.25	−0.11	−0.34	−0.02	−0.10	11 400	−0.80
A0	+0.65	−0.02	−0.02	0.02	−0.02	9 790	−0.30
A2	+1.3	+0.05	+0.05	0.08	0.01	9 000	−0.20
A5	+1.95	+0.15	+0.10	0.16	0.06	8 180	−0.15
F0	+2.7	+0.30	+0.03	0.30	0.17	7 300	−0.09
F2	+3.6	+0.35	0.00	0.35	0.20	7 000	−0.11
F5	+3.5	+0.44	−0.02	0.40	0.24	6 650	−0.14
F8	+4.0	+0.52	+0.02	0.47	0.29	6 250	−0.16
G0	+4.4	+0.58	+0.06	0.50	0.31	5 940	−0.18
G2	+4.7	+0.63	+0.12	0.53	0.33	5 790	−0.20
G5	+5.1	+0.68	+0.20	0.54	0.35	5 560	−0.21
G8	+5.5	+0.74	+0.30	0.58	0.38	5 310	−0.40
K0	+5.9	+0.81	+0.45	0.64	0.42	5 150	−0.31
K2	+6.4	+0.91	+0.64	0.74	0.48	4 830	−0.42
K5	+7.35	+1.15	+1.08	0.99	0.63	4 410	−0.72
M0	+8.8	+1.40	+1.22	1.28	0.91	3 840	−1.38
M2	+9.9	+1.49	+1.18	1.50	1.19	3 520	−1.89
M5	+12.3	+1.64	+1.24	1.80	1.67	3 170	−2.73
GIANTS, III							
G5	+0.9	+0.86	+0.56	0.69	0.48	5 050	−0.34
G8	+0.8	+0.94	+0.70	0.70	0.48	4 800	−0.42
K0	+0.7	+1.00	+0.84	0.77	0.53	4 660	−0.50
K2	+0.5	+1.16	+1.16	0.84	0.58	4 390	−0.61
K5	−0.2	+1.50	+1.81	1.20	0.90	4 050	−1.02
M0	−0.4	+1.56	+1.87	1.23	0.94	3 690	−1.25
M2	−0.6	+1.60	+1.89	1.34	1.10	3 540	−1.62
M5	−0.3	+1.63	+1.58	2.18	1.96	3 380	−2.48

Table 3.1: The absolute magnitude in the V-band, colors, effective surface temperatures and bolometric corrections of the different spectral types of main sequence stars and giants [26].

3.1.2 Color index

After determining the magnitude of an astronomical object in different passbands, it becomes possible to calculate its color index. The color index is a numerical representation that provides information about the object's spectral energy distribution. It is calculated by subtracting the magnitude of the object observed in one passband from its magnitude observed in another passband. The color index not only provides information about the object's color but also about its temperature and spectral type, particularly in the case of stars. By comparing the magnitudes in different passbands, we are able to determine the color index and infer the temperature of a star's surface. Hotter stars have lower color indices, appearing bluer, while cooler stars have higher color indices, appearing redder. This relationship between color index and temperature is based on the principles of blackbody radiation, where hotter objects emit more of their radiation at shorter (bluer) wavelengths. Furthermore, the color index can also help determine the spectral type of a star, which provides information about its chemical composition and evolutionary stage. Different types of stars, such as main sequence stars and giants have characteristic color indices associated with their spectral features as can be seen in Table 3.1. By analyzing the color index, we can make inferences about a star's spectral type and gain insights into its properties, such as its luminosity and stage of stellar evolution [109].

In addition to providing information about an object's color, temperature and spectral type, the color index is also important when studying the effects of extinction. Extinction is the process

Date	Images	Exposure time (s)	Filters	Air-mass range
14.02.2020	4	300.0	BIRV	1.32-1.39
16.02.2020	18	300.0	BIRV	1.08-1.17
18.02.2020	73	300.0	IRV	1.05-1.9
19.02.2020	74	300.0	IRV	1.05-1.86
24.02.2020	9	300.0	IRV	1.13-1.23
02.03.2020	5	300.0	IRV	1.3-1.38
13.03.2020	9	300.0	IRV	1.07-1.11
14.03.2020	5	300.0	IRV	1.1-1.13
18.03.2020	8	300.0	IRV	1.05-1.07
20.03.2020	19	300.0	IRV	1.05-1.21
22.03.2020	3	300.0	IRV	1.11-1.12
23.03.2020	5	300.0	IRV	1.06-1.06
26.03.2020	3	300.0	IRV	1.05-1.09
27.03.2020	4	300.0	IRV	1.07-1.09
28.03.2020	6	300.0	IRV	1.05-1.17
09.04.2020	9	300.0	IRV	1.09-1.15
08.05.2020	11	300.0	IRV	1.11-1.21

Table 3.2: A summary of the observations with the TJO in the period 14th of February 2020 to 8th of May 2020. The table displays the observation data, number of images taken, exposure time per image, filters used that night, and the air-mass range during the night. A light-curve plot of sub-sets of the observational period can be seen in Figure 3.3.

of absorption and scattering of light as it passes through the interstellar or intergalactic medium. The interstellar medium is composed of dust and gas, and these intervening materials can diminish the intensity in specific wavelengths, causing a shift in the observed colors of astronomical objects. By comparing the observed color index of an object with the intrinsic color index, we are able to quantify the amount of extinction along the line of sight, and is calculated by the equation

$$E(B-V) = (B-V)_{observed} - (B-V)_{intrinsic} \tag{3.3}$$

The difference between the two is known as color excess, and provides an estimate of the amount of reddening caused by the extinction [91]. This also creates the possibility to find the intrinsic color index by using dust maps, maps which trace dust reddening as a function of distance and angular position in the sky, such as the one released by Green et al. 2019 [44].

3.2 Observations and data analysis

As mentioned in section 1.3, the focus of this thesis is one particular redback, PSR J1048+2339 (hereafter referred to as J1048). We observed J1048 using the TJO telescope on 19 different nights from the 15th of February 2020 to the 8th of May 2020. The images were taken with the LAIA instrument and each image was exposed for 300 seconds in the Johnson-Cousin filters B, I, R and V. The bias, dark and flat-field corrections are performed by the TJO team as part of their pipeline. The data were downloaded using the TJOpipeline software (see Sec. 2.5), which produced log files for each night, making it possible to get a quick overview of the data. In total, we acquired 85 images in the I-band, 86 in the R-band, 84 in the V-band and 6 in the B-band, giving full orbital coverage in the I-, R- and V-bands.

The magnitudes of J1048 in the different filters were obtained by performing differential photometry on the data taken by the TJO. This is a technique where we transform the counts measured by the CCD of the telescope into a number on the magnitude scale. In order to do this we need to extract the counts for our target and reference stars using the ultracam pipeline software [32]. We select our target and reference stars using the function *setaper* which allows the user to set apertures around the points of interest of which the counts should be extracted (see Fig. 3.1). Then we run *reduce* with the required parameters. This was repeated for each night of observations until we had data files containing information, such as the time of observation, counts, errors, etc., on every star chosen through *setaper*.

Assuming that the measured counts on the CCD are proportional to the received flux (F) from the object, it is possible to rewrite equation 3.2 in terms of the detected counts. In order to find the magnitude it is necessary to compare our target to what is known as a reference star which has a known magnitude. Through inserting the values for a given filter for the reference star magnitude, and the counts for both the target and the reference star, we can find the corresponding magnitude for our target in this filter. For the determination of the magnitude of J1048, a total of 6 reference stars were used in the calculation, as seen in Figure 3.1. A set of reference stars are used in order to lower the error on the final magnitude and provide higher precision photometry, compared to what could be achievable with a single reference star. In order to construct a "master" magnitude for the reference stars, we need to add up the apparent fluxes from each star using their previously determined magnitudes. This is done by solving equation 3.2 for F_1 and assuming F_{ref} is the zero-point flux for that passband. This yields

$$m_{tot} = -2.5 \log_{10}\left(\sum_{i=2}^{n_{ref}+1} 10^{-0.4 m_i}\right) \tag{3.4}$$

where m_{tot} is the "master" magnitude, n_{ref} is the number of reference stars chosen, i corresponds to the numbered stars in Figure 3.1 and m_i is the known magnitude of the star. Each of these previously known magnitudes have an accompanying error, and from error propagation we get that the total error on m_{tot} is given by

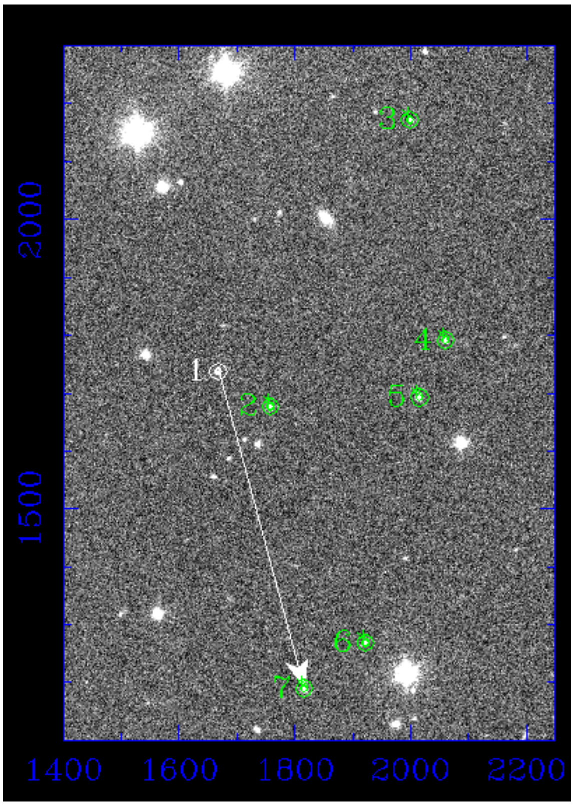


Figure 3.1: A screenshot taken from the differential photometry performed by the ULTRACAM pipeline software. The target, PSR J1048+2339, is noted as star 1 in the image, and the reference stars are stars 2 through 7. The arrow from star 1 to star 7 is a link between the two apertures in order to ensure correct tracking of apertures as the function *reduce* from the ULTRACAM pipeline software extracts the counts for each image taken during a night of observations.

$$\varepsilon_{tot} = \sqrt{\sum_{i=2}^{n_{ref}+1}\left(\frac{\varepsilon_i}{m_i \cdot \sum_{j=2}^{n_{ref}+1} 1/m_j}\right)^2}, \tag{3.5}$$

where ε_i denotes the errors corresponding to the magnitude of star i. Here we can see that for a larger number of reference star, the error contribution from each reference star will be smaller.

In order to select the reference stars, the magnitudes of the stars in the field were found using the Pan-STARRS1 Surveys [18]. We narrowed down the available stars by only considering stars with magnitudes slightly lower than the expected magnitude of J1048. This would ensure that the reference stars were visible simultaneously as J1048 in case of cloud coverage or other factors which could influence the seeing. This resulted in the choice of the six reference stars seen in Figure 3.1, with magnitudes as seen in Table 3.3. It is also important that the received flux from the reference

Reference star	mag$_B$	mag$_V$	mag$_R$	mag$_I$
2	18.63 ± 0.05	17.10 ± 0.02	16.23 ± 0.02	15.39 ± 0.03
3	18.62 ± 0.04	17.99 ± 0.02	17.61 ± 0.02	17.25 ± 0.02
4	18.06 ± 0.04	17.30 ± 0.01	16.85 ± 0.02	16.45 ± 0.02
5	18.84 ± 0.05	17.29 ± 0.02	16.42 ± 0.02	15.35 ± 0.03
6	18.76 ± 0.04	17.82 ± 0.02	17.27 ± 0.02	16.74 ± 0.02
7	17.76 ± 0.04	16.93 ± 0.02	16.44 ± 0.02	15.99 ± 0.02

Table 3.3: The Pan-STARRS 1 magnitude values with uncertainties of the reference stars chosen for the photometry of PSR J1048+2339 [18]. Here the numbered reference stars correspond to the numbers in Figure 3.1.

stars remain stable throughout our observations to make sure that the measured variations in the apparent magnitude of our target is indeed intrinsic and not due to variations in our reference stars. Figure 3.2 is a plot of the folded light curves of the chosen reference stars, using orbital ephemeris of J1048 ([31]) in order to easily examine if there are any variations over the orbit which can inflict errors onto the light curves of J1048. There are slight deviations in reference star 2, 3 and 5 in the B-band, but the error is consistent and leads to the conclusion that these are stable stars capable of producing accurate light curves since all of our reference stars have RMS-values of less than 0.03 in every passband (see Fig. 3.2).

The photon arrival times at the observatory are not useful for accurate timing work as the location changes throughout Earths year-long orbit, and thus the light will travel differently. To assure the best possible precision in the light curves, it is necessary to apply a barycentric correction to the dataset to account for the orbital motion of Earth and to calculate the photon arrival time at a standard location, the solar system's barycenter[1]. Using the AstroPy library it is possible to streamline the transformation by using their *time*-package. We create appropriate **Time** and **SkyCoord** objects using the location of the telescope and the logged times of observations, and the right ascension and declination of our target. Using these objects it is simple to calculate the difference in photon arrival times between the Solar system barycenter and observatory by using the light_travel_time function. This creates a **TimeDelta** object which we can add to our **Time** object to give us the barycentric corrected times of observations.

3.2.1 Light curves

To create light curves we apply Equation 3.2 to every image taken in the time period and calculate the differential magnitude of J1048 using the measured counts from our target and the sum of all counts in the reference stars. Then, adding the total of the known magnitudes yields the apparent magnitude of our target for every observation. All of the magnitude data are then folded using the equation

$$\phi = \frac{T - T_0}{P_{orb}} - \text{int}\left(\frac{T - T_0}{P_{orb}}\right) \tag{3.6}$$

where $T > T_0$ is the time of observation, T_0 is the epoch of superior conjunction of the pulsar and P_{orb} is the orbital period. T_0 and P_{orb} are gathered from the orbital ephemeris found by Deneva et al. 2021[31] where $T_0 = 56637.598174(2)$ MJD and $P_{orb} = 0.25051915(3)$d. This resulted in the light curve shown in Figure 3.4. The companion is in inferior conjunction at orbital phase $\phi = 0$, and superior conjunction at $\phi = 0.5$. The light curve displays a single peak per orbit in every band, pointing towards the evidence of an irradiated companion (as discussed in Section 1.2). Thus we find evidence that the system did not undergo any transitions from an irradiated system to an non-irradiated system over the period of our observations. Further evidence for this can be seen in Figure 3.3, where the observations from the observational period is displayed separately. Here the data are divided into sub-periods of observations, and each sub-period is folded on the same T_0 and P_{orb} as before.

[1] The barycenter is the center mass point of a system, and is the point that all celestial objects in our Solar system orbit around.

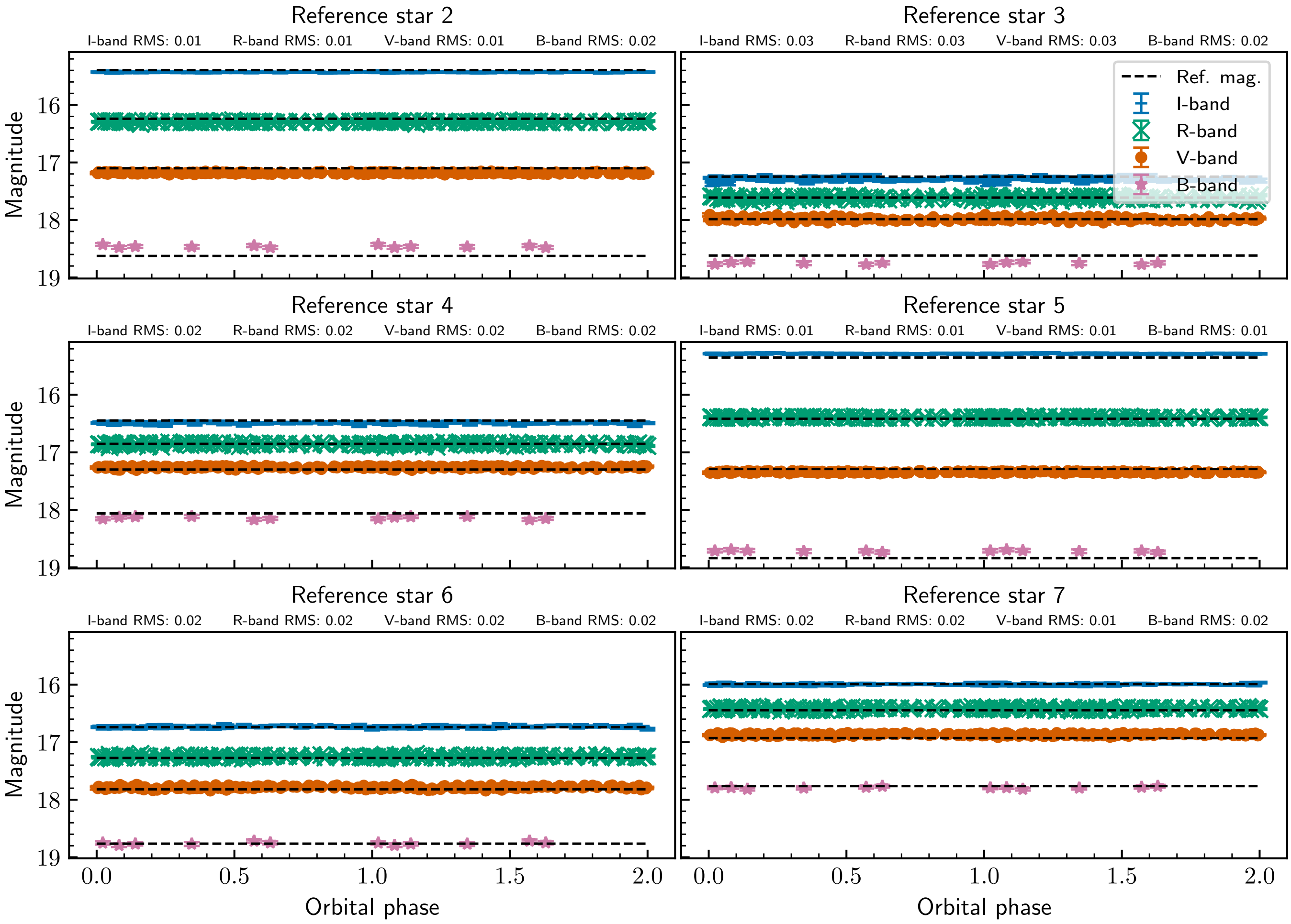


Figure 3.2: Folded light curves of the reference stars used in the magnitude calculation of PSR J1048+2339. The data are folded on top of the orbital phase of PSR J1048+2339 in order to examine if there are any intrinsic variations in the reference stars which can impact the light curves of the neutron star binary.

Filter	mag_{max}	ϕ_{max}	$\Delta\phi$ $(0.5 - \phi_{max})$	$\Delta\phi$ in minutes
I	18.16 ± 0.02	0.372 ± 0.010	0.128 ± 0.014	46 ± 5
R	18.79 ± 0.02	0.407 ± 0.010	0.093 ± 0.014	34 ± 5
V	19.42 ± 0.02	0.401 ± 0.007	0.099 ± 0.011	36 ± 4

Table 3.4: The resulting maximum magnitude and corresponding orbital phase value ϕ_{max} found by fitting parabolas to the dataset. $\Delta\phi$ in minutes found using P_{orb} [31].

When the side of a companion facing the neutron star is being irradiated, the expectation is for the companion to appear brightest when in superior conjunction. However, J1048 does not adhere to this expectation, rather the maxima of the sinusoidal modulation in the light curve is shifted towards phase $\phi = 0$ by ~ 0.1 orbital phases. To quantify this, we fit parabolas of the form $p(\phi) = A(\phi - \phi_{max}) + B$ to the light curve data in order to measure the orbital phase shift, and how much it deviates from the expected $\phi = 0.5$. To do the latter, it is necessary to calculate the uncertainty of the current orbital epoch through error propagating Equation 3.6. This yields an uncertainty of $\varepsilon_\phi = 0.0091$. The results are displayed in table 3.4. These results correspond to the observed phase shifts in Yap et al. 2019 [108], who reported a phase shift of $\Delta\phi = 0.1$ (can be seen in Figure 1.6).

3.2.2 Colors and irradiation

Upon calculating the magnitudes in different bands, we can determine the colors exhibited by our system at various orbital phases. The analysis of these colors offers valuable insights into the thermal properties of the neutron star companion. In Section 3.1.2, we outlined an important factor: the observed color does not match the intrinsic color due to the influence of extinction caused by the interstellar medium. To accurately determine the temperatures of the companion on its day and night side, it is crucial to correct the apparent magnitudes for the significant influence of interstellar extinction. By accounting for this, we ensure that our temperature estimations are not skewed and accurately reflect the thermal properties of the neutron star companion.

In order to account for the extinction we make use of the interstellar dust map created by Green et al. 2019 [44]. Entering the right ascension and declination into their online query map will yield the reddening or g'-r' color excess for a given distance from Earth. In the direction of J1048, this reddening is found to be $E(g - r) = 0.02$ magnitudes at any distance above 0.3kpc (recall that J1048 is at a distance of 1.7kpc). It is thus necessary to translate this reddening to the color excess in the colors V-I and V-R. This is achieved by finding the relation between the two separate extinctions using the table for the extinction coefficients in Green et al. 2019 [44], and the extinction coefficients in Table 6 of Schlafly & Finkbeiner 2011 [89]

$$a = \frac{E(V - I)}{E(g - r)}, \qquad b = \frac{E(V - R)}{E(g - r)}. \tag{3.7}$$

Here $E(V - I)$, $E(V - R)$ and $E(g - r)$ is found by subtracting the extinction coefficients found in the tables of Green et al. and Schlafly & Finkbeiner from each other. This yields the coefficients $a = 1.37$ and $b = 0.636$. Multiplying the reddening found from the interstellar dust map for J1048 ($E(g - r) = 0.02$) with the coefficients a and b yields the V-I and V-R color excess, and from this we can find the intrinsic color indices. It is then possible to find the temperature of the day and night side of the companion by cross-referencing the value of our color indices with Table 5 in Pecaut & Mamajek 2013[2] [78]. The resulting color-values and temperature values can be found in Table 3.5. Here the color-values are found using the magnitude values from the binned light curves (which can be seen in Figure 3.5) at phases $\phi = 0$ and $\phi = 0.4$, with propagated uncertainties. It is important to note that the uncertainty in the night side temperatures are quite large. This

[2] Note that this is an approximate method and the derived intrinsic colors correspond to the spectral energy distribution at phases $\phi = 0$ and $\phi = 0.4$, which includes light from different parts of the star due to the inclination of the system.

Color	Δmag_{day}	Δmag_{night}	T_{day} (K)	T_{night} (K)	T_{irr} (K)	$L_{irr}(\mathrm{erg\,s^{-1}})$
V-R	0.63 ± 0.06	0.9 ± 0.2	4600 ± 230	3890 ± 500	3850 ± 650	$(3.7 \pm 2.5) \times 10^{33}$
V-I	1.26 ± 0.06	1.8 ± 0.2	4440 ± 150	3850 ± 200	3600 ± 371	$(2.9 \pm 1.2) \times 10^{33}$

Table 3.5: The results calculated from the two colors, V-R and V-I. The colors at the point of maximum magnitude ($\phi = 0.4$), and minimum magnitude ($\phi = 0$), T_{day} and T_{night} found from table 5 in Pecaut & Mamajek 2013 [78], T_{irr} calculated by Equation 3.8 and L_{irr} calculated by Equation 3.9.

is due to the large uncertainty in the V-band magnitude at phase $\phi = 0$, which are due to poor image quality resulting from the limitations of the telescope used. The chosen exposure time was insufficient to capture a high-quality image, and as a result, the target object was simply not bright enough during this orbital phase. Due to the discrete nature of Table 5 in Pecaut & Mamajek 2013 [78], it is not possible to get exact values for the temperatures and their uncertainties in the different cases. The listed temperatures and uncertainties are therefore an estimate for the day and night side in order to further calculate the irradiation temperature of the companion.

In compact binary millisecond pulsars, the irradiation temperature refers to the increase in temperature of the pulsar's companion star as a result of the emitted radiation from the pulsar. Modelling the optical observations of spider companions can give insight into the fundamentals of the pulsar wind [105]. One example is to estimate the amount of radiating power from the pulsar wind which hits the companion, by comparing the temperatures of the day and night side, and calculate the irradiation temperature. To do this we compare the radiation of black-bodies and assume that the difference between the black-body radiation in the day and night side is all due to the irradiation from the pulsar [15]. This yields the formula

$$T_{irr} = \left(T_{day}^4 - T_{night}^4\right)^{-4}. \tag{3.8}$$

For the different colors we find the estimated irradiated temperature to be; $T_{irr}^{(V-R)} = 3850\mathrm{K}$ and $T_{irr}^{(V-I)} = 3600\mathrm{K}$.

Furthermore, the irradiation temperature of the companion star provides crucial insights into the associated irradiated luminosity. The irradiated luminosity represents the amount of energy received by the companion star from the pulsar radiation and high-energy particle emission. Finding this value and comparing it to the spin-down luminosity of J1048 can give insight into the geometry and efficiency of the pulsar wind and pulsar radiation. We can use the formula given by Breton et al. 2012 [15] to calculate the irradiating luminosity:

$$L_{irr} = 4\pi a^2 \sigma T_{irr}^4. \tag{3.9}$$

Here a is the orbital separation, calculated by Kepler's third law using the masses for the pulsar and the companion [93] and the orbital period [31], σ is the Stefan-Boltzmann constant, and T_{irr} is the calculated irradiated temperature from Equation 3.8. Inserting the values yield irradiated luminosities of; $L_{irr}^{(V-R)} = 3.7 \times 10^{33}\mathrm{erg\,s^{-1}}$ and $L_{irr}^{(V-I)} = 2.9 \times 10^{33}\mathrm{erg\,s^{-1}}$.

The apparent colors for one orbital phase are shown in Figure 3.6. These were calculated by the use of a binned light curve, where the magnitudes in each filter within a bin of 0.05 orbital phases were averaged (see Fig. 3.5). The color curves further confirm that the companion of J1048 is asymmetrically heated, as we can observe the minimum color value occuring at orbital phases $\phi < 0.5$. A smaller color value implies hotter surface temperatures. The color curves also display a large uncertainty at and around phase $\phi = 0$. It is important to note that these are apparent color curves, meaning that they have not been corrected for extinction, and are mainly to investigate the trend in color, and thus the surface temperature of the companion.

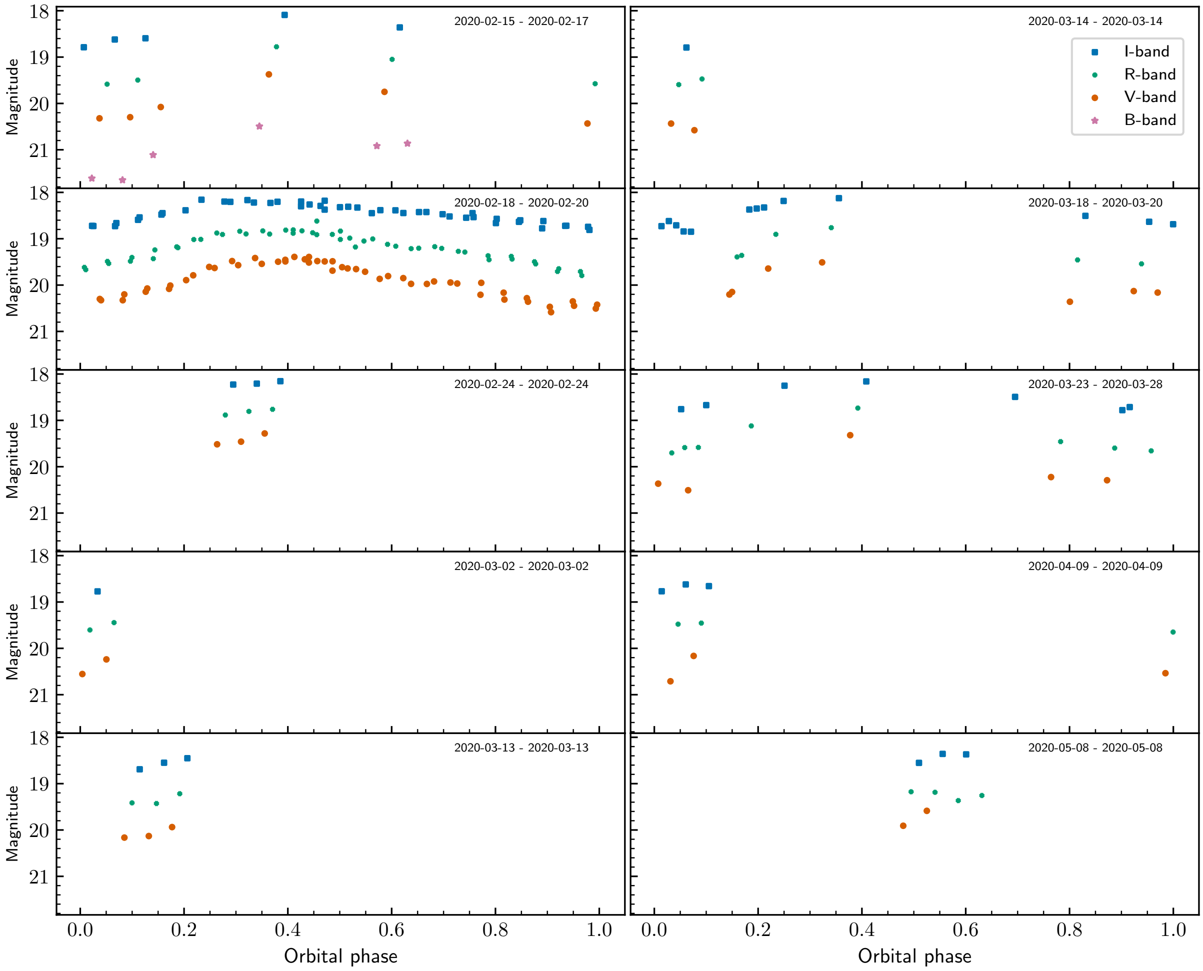


Figure 3.3: Light curves showing different time periods of the observations taken with the TJO in the period 15.02.2020 to 08.05.2020. Though the dataset is sparse in the later months of the observational period, one can infer a sinusoidal modulation due to irradiation of the companion for the observations taken in February and March.

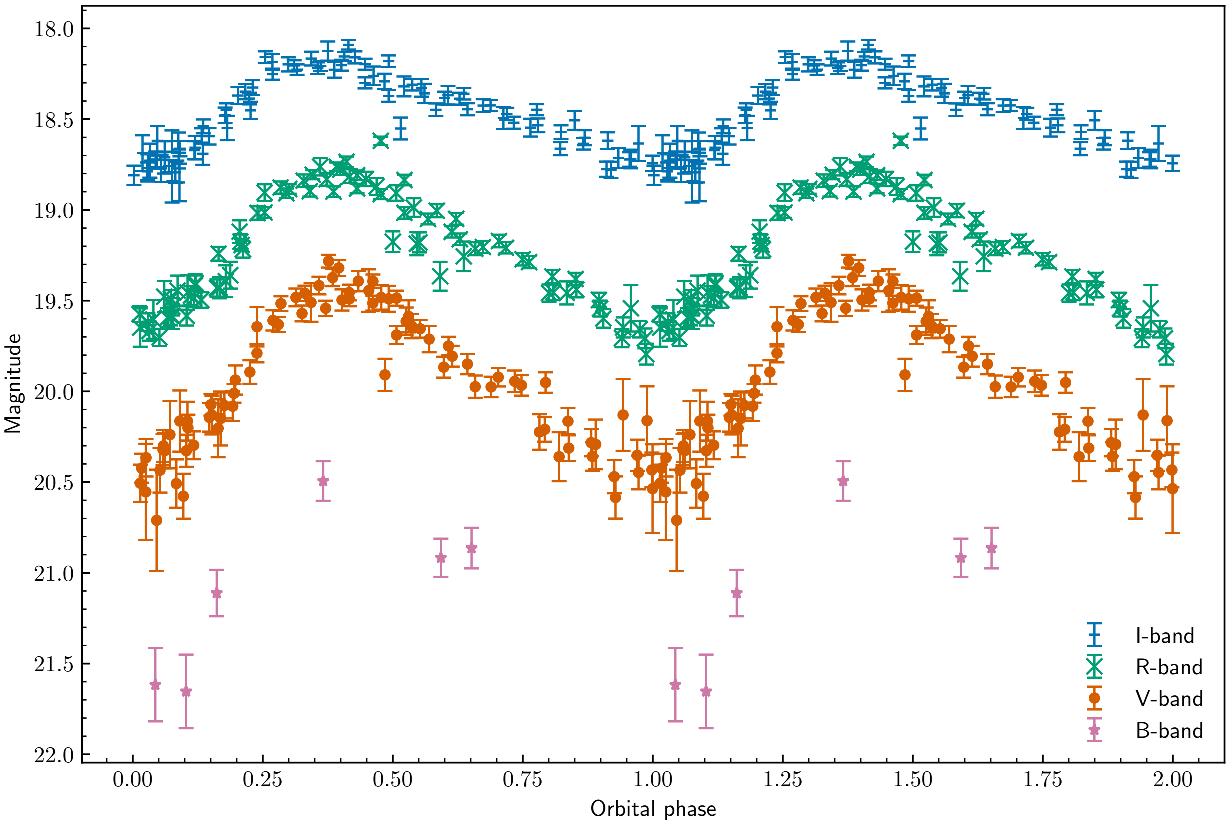


Figure 3.4: Light curves of PSR J1048+2339 in the Johnson-Cousin filters B, V, R and I taken by the TJO in the period of 15th of February 2020 to 8th of May 2020. The data are folded with an orbital period of 6 hours and $T_0 = 56637.598174$. The binary displays evidence for irradiation throughout the observational period. The companion is in superior conjunction at $\phi = 0.5$, however the point of minimum magnitude happens earlier in the orbit ($\phi \sim 0.4$). The light curves thus present an indication that the companion is asymmetrically irradiated.

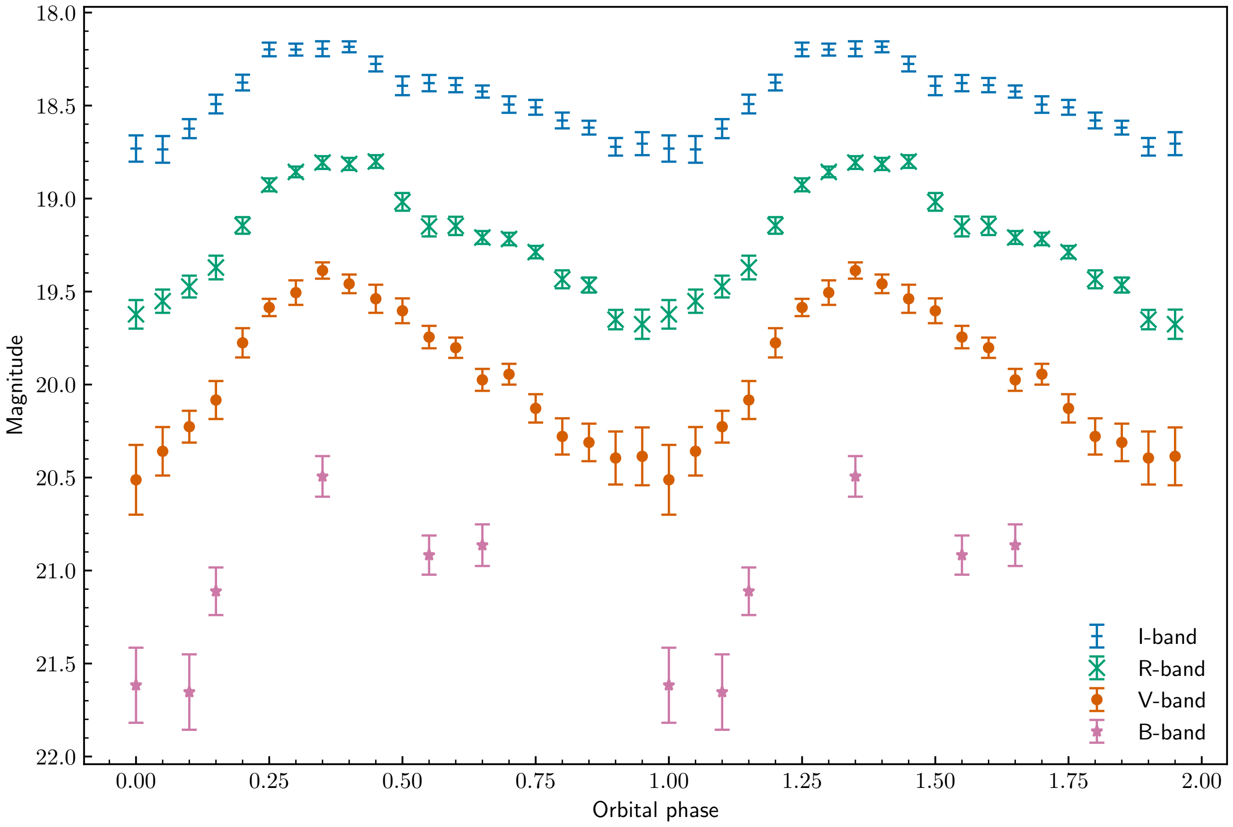
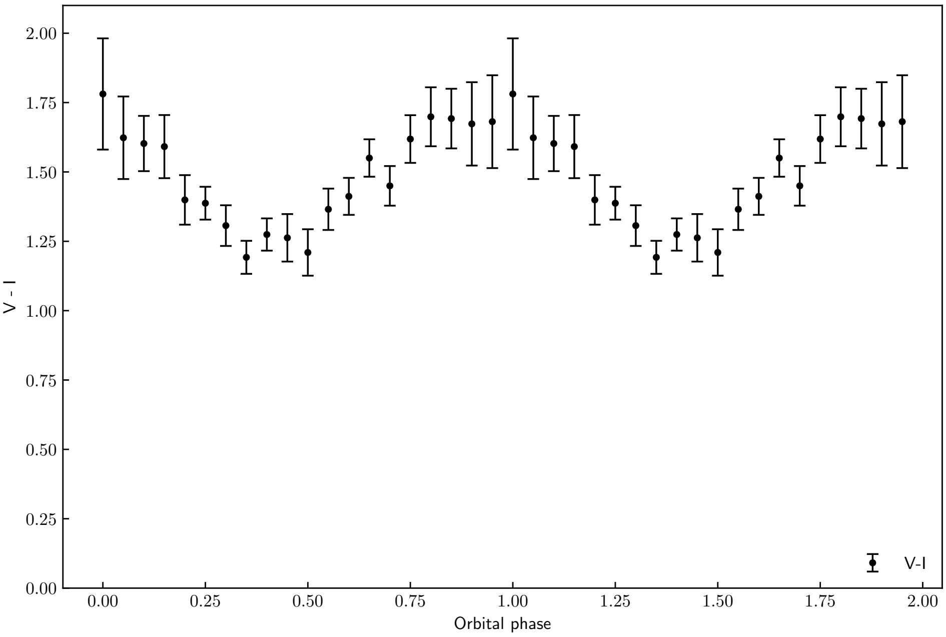
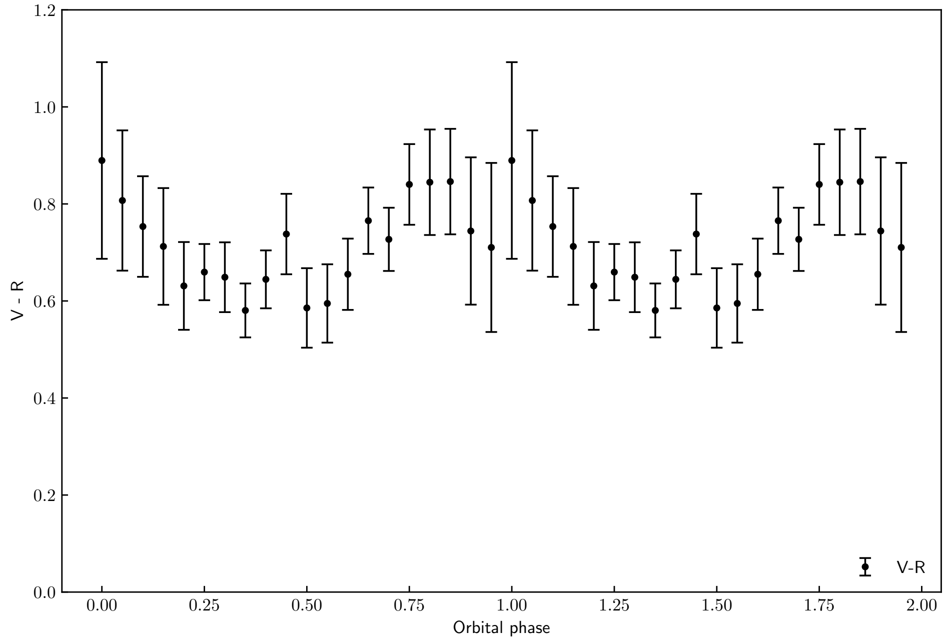


Figure 3.5: Binned light curve of PSR J1048+2339 with a binning of 0.05 using data taken by the TJO. The binned light curve is created in order to reduce noise and enhance the visibility of trends and variations in the data. The light curve displays evidence for asymmetric heating of the companion, with the minimum magnitude being shifted $\Delta\phi = 0.1$ orbital phases.



(a) V-I color curve.



(b) V-R color curve.

Figure 3.6: Apparent color curves for our two calculated colors, V-I and V-R, for PSR J1048+2339. They are calculated using a binned light curve of J1048 with a bin of 0.05. The color curves display asymmetric heating of the companion, similar to what is observed in the light curves.

Chapter 4

Optical photometry with the Nordic Optical Telescope

4.1 Nordic Optical Telescope

The Nordic Optical Telescope (NOT) is a 2.56m diameter telescope. To ensure a broad field of view with optimal optical quality and eliminate spherical aberration, the Nordic Optical Telescope (NOT) features a Ritchey-Chrétien optical system coupled with an alt-azimuth mounting configuration. Situated at the Roque de los Muchachos Observatory in La Palma, the NOT has been the primary optical telescope for northern-hemisphere observations for Nordic astronomers since 1989 [38]. The telescope was originally funded by Norway, Sweden, Denmark and Finland, but the initiative welcomed the University of Iceland's participation in 1997. The telescope and the enclosure was designed to deliver the best possible image quality for the site [6].

Among the design choices for better image quality was an active optics system, which was installed at the NOT in 1992 [4]. The active optics system implemented in the NOT facilitates controlled deformation of the primary mirror by fine-tuning the load forces acting on the mirror through a network of 45 metallic bellows positioned along its axial support. The goal is to remove any aberrations originating in the telescope itself. This is done through assuring the alignment of the primary and secondary mirrors to prevent a decentering coma, and the removal of low-order aberation [4]. High quality imaging is achieved by measuring the aberrations using a wavefront sensor and subsequent tuning of the mirror.

The NOT currently features 5 different instruments; ALFOSC (for UV-optical imaging, low-resolution spectroscopy and polarimetry), FIES (high-stability and high-resolution optical spectroscopy), NOTCam (Near-IR imaging and low-resolution spectroscopy), MOSCA (High-resolution, high-efficiency UV-optical imaging) and StanCam (Standby CCD-imager) [1]. The data analyzed in this chapter were aquired using ALFOSC (The Andalucia Faint Object Spectrograph and Camera). It has a 2048x2064 CCD, and a field of view of 6.4x6.4 arcminutes in imaging mode. The instrument has a permanent set-up consisting of 4 wheels that contain grisms and slits for spectroscopy and filters for photometry. The wheels include SDSS ugriz-filters and Johnson-Cousin UBVRI-filters.

Apart from serving as a platform for scientific research to address the objectives of the scientific community, the NOT also dedicates a portion of its observation time to educating the upcoming generation of astronomers. They offer on-site courses tailored for groups of PhD and MSc students in order to teach how to observe with the NOT [38]. There is also a focus on becoming proficient at data reduction and learning how to formulate effective observing time applications. There is also the possibility for remote observing, therefore providing the experience to university courses without taking the students to La Palma. This is offered by the course FY3215 Observational astrophysics at NTNU, and provides hands-on observing experience for the students.

4.2 Observations and data analysis

The data-set is composed of photometric images of the redback system PSR J1048+2339, taken on the night of the 18th of February 2020 with the NOT. Each image was exposed for 60 seconds with a binning of 2x2. The filter was changed in-between every exposure, alternating between the SDSS g'-, r'- and i'-filter. The total duration of the observations were 3 hours and 26 minutes. During this time, 55 images were acquired in the r'- and i'-bands, while 54 images were obtained in the g'-band. These observations took place during the same night as one of the better covered nights in the TJO data-set, and are therefore complimentary to the previously presented results.

4.2.1 Data reduction

In order to create high quality photometric light curves it is necessary to perform bias and flat field corrections on the data provided by the observations from the NOT. This data reduction is performed in IRAF (Image Reduction and Analysis Facility) using the *ccdred*-package. The CCD sensor inherently introduces read-out noise while reading each pixel. The bias correction consists of subtracting a bias frame from each image in the data-set, and the flat frames. This bias frame is created by performing a 0 second exposure, meaning every measured count will be a result of the generated read-out noise. As the read-out noise will differ from image to image, it will beneficial to acquire multiple bias frames, and combine them into a master bias. To do so we make use of the built in function *zerocombine*, which takes in the filenames of the bias frames as a parameter, and returns an averaged, pixel by pixel, master bias frame. This can then be subtracted from every image using the function *ccdproc*, with the bias correction parameter set to true, which subtracts each pixel value by the corresponding pixel value in the master bias.

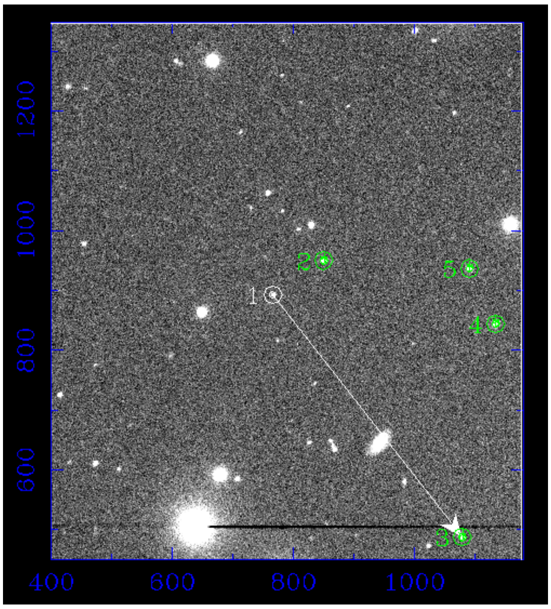


Figure 4.1: A screenshot taken from the differential photometry performed by the ULTRACAM pipeline software on data from the Nordic Optical Telescope. The target, PSR J1048+2339, is noted as star 1 in the image, and the reference stars are stars 2 through 5. The arrow from star 1 to star 5 is a link between the two apertures in order to ensure correct tracking of apertures as the function *reduce* from the ULTRACAM pipeline software extracts the counts for each image.

In addition to this constant noise, the images may exhibit non-uniform pixel responses from several sources. Some examples include pixel dropouts in the CCD, variable pixel gain, misuse of the detector leading to artifacts or simply dust particles on the lens [90]. This noise is consistent for every image, so therefore we aim to obtain a flat-field image by capturing a uniformly illuminated section of the sky that exhibits the consistent noise and then apply this calibration frame to the source images, correcting for the observed aberrations. The imperfections can come from dust or scratches on the different filters, therefore the flat field correction is done for each filter separately. Typically, one would acquire multiple flat frames, and combine these into a master flat for each filter. This is achieved through the function *flatcombine*, which combines every flat in a filter to one by taking the median value in each pixel location. For our data, we end up with three master flats for the filters g', r' and i'. These would then be normalized and each of the science images is divided by the normalized master flat in their respective filters by the function *ccdproc*, now with the flat correction parameter set to true. By doing this, the variations in pixel sensitivity and illumination across the image are compensated for, resulting in a more accurate representation of the true object being imaged.

Figure 4.2a shows that the images have a black border surrounding the exposed part of the CCD. This causes issues with the ULTRACAM extraction of counts if there are negative counts in the frame, which can happen at these fringes due to the bias subtraction. Therefore it is necessary to trim the image in order to prevent this from occuring. This is done by once again using the function *ccdproc*. We set the trimming section to be $x = [160 : 960]$ and $y = [70 : 970]$. After trimming and data reduction, the resulting image can be seen in Figure 4.2b.

The extraction of counts from the photometric images, and the subsequent data analysis in order to create light curves, is done similarly as in Chapter 3. The counts of the target and reference stars are extracted with the ULTRACAM pipeline software. The apertures chosen in *setaper* are shown in Figure 4.1. The chosen reference stars are the same as the reference stars $2 - 5$ in Figure 3.1, where stars 6 and 7 are omitted due to not being in the field of view. After setting the apertures, we run *reduce* with the required parameters to acquire the data files containing the information necessary for further analysis.

4.2.2 Light curves

We apply Equation 3.2 to the measured counts in the target and the sum of the reference stars, and add the total of the known magnitudes, to calculate the apparent magnitude of J1048 in the SDSS g'-, r'- and i'-band. We perform a barycentric correction on the observation times, in order to go from MJDs to TDB. The magnitude data are then folded using Equation 3.6 with the same orbital ephemeris from Deneva et al. 2021 [31] used in Chapter 3. The resulting light curve can be seen in Figure 4.3.

Filter	mag$_{max}$	ϕ_{max}	$\Delta\phi$ $(0.5 - \phi_{max})$	$\Delta\phi$ in minutes
i'	18.361 ± 0.004	0.366 ± 0.004	0.134 ± 0.010	48 ± 4
r'	18.618 ± 0.005	0.395 ± 0.002	0.105 ± 0.009	38 ± 3
g'	19.363 ± 0.003	0.414 ± 0.002	0.085 ± 0.009	31 ± 3

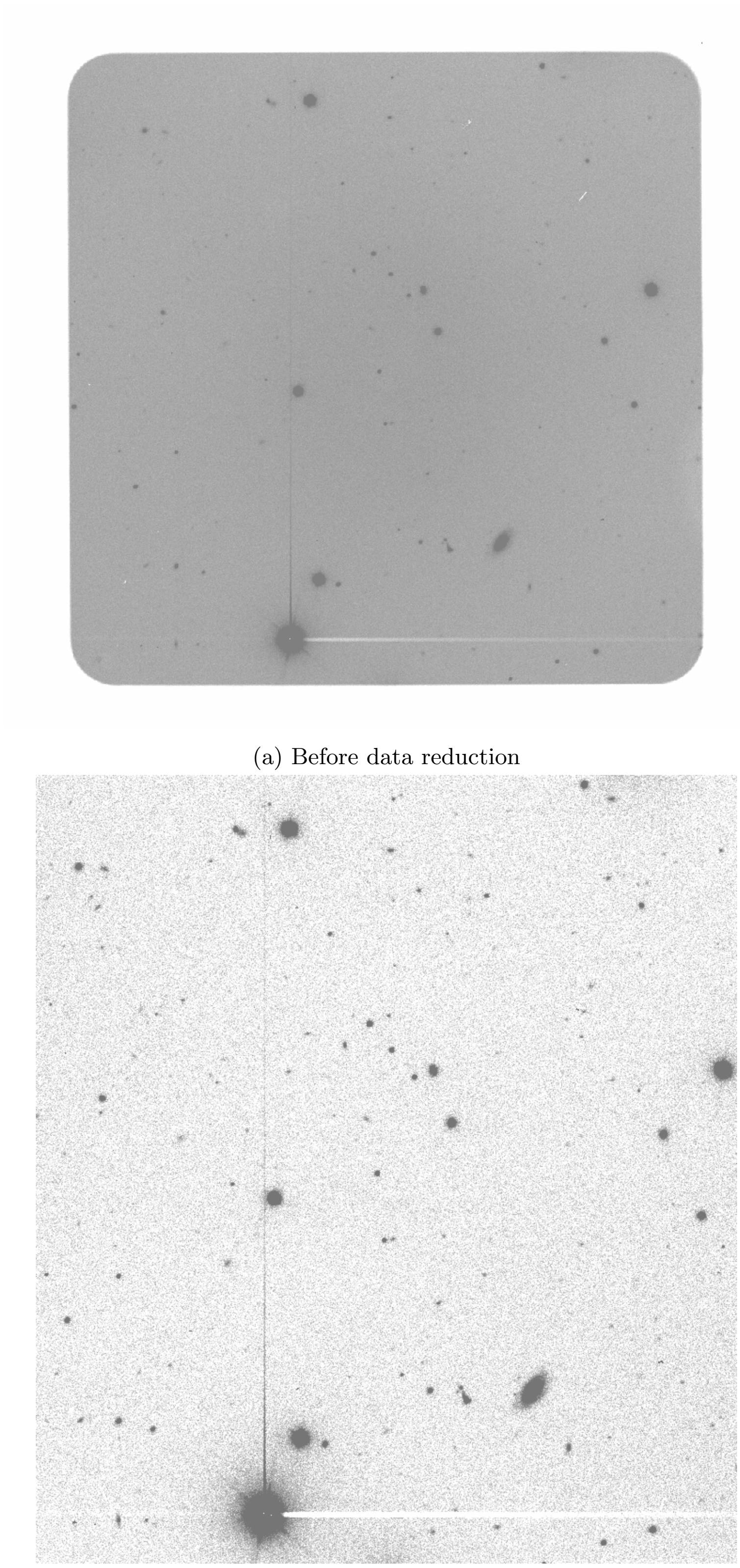
Table 4.1: The resulting maximum magnitude and corresponding orbital phase value ϕ_{max} found by fitting parabolas to the dataset. $\Delta\phi$ in minutes found by multiplying the phase values with $P_{orb} = 0.25051915(3)$ [31] and transforming from days to minutes.

The light curves demonstrate a similar behavior ast that seen in the TJO data, where each orbit exhibits only one maximum, with a noticeable shift towards lower orbital phases. However, due to the relatively short observational period of 3 hours and 26 minutes, we have incomplete coverage of the orbital phases. Nonetheless, the data obtained during this period have smaller uncertainties compared to the TJO data, enabling more precise measurements of the phase shift in different passbands. Fitting parabolas of the form $p(\phi) = A(\phi - \phi_{max}) + B$ to the light curves results in a ϕ_{max} of; $\phi_{max}^{(i')} = 0.366 \pm 0.004$, $\phi_{max}^{(r')} = 0.395 \pm 0.002$ and $\phi_{max}^{(g')} = 0.414 \pm 0.002$. The results are summarized in table 4.1. The phase shift is even more prominent in the SDSS i'-band than in the Johnson-Cousin I-band, which was measured at $\Delta\phi = 0.128 \pm 0.014$. It also interesting to note the large difference in shifts between the different passbands, in addition to the large phase shift from phase $\phi = 0.5$;

$$\phi_{max}^{(g')} - \phi_{max}^{(r')} = 0.019 \pm 0.003,$$
$$\phi_{max}^{(g')} - \phi_{max}^{(i')} = 0.048 \pm 0.004,$$
$$\phi_{max}^{(r')} - \phi_{max}^{(i')} = 0.029 \pm 0.004.$$

4.2.3 Color and irradiation

We wish to determine the color and corresponding surface temperature of J1048 using data obtained from the NOT. We focus on the g'-r' color, due to the nature of in table 5 of Pecaut & Mamajek



(a) Before data reduction

(b) After data reduction

Figure 4.2: Photometric images taken by the Nordic Optical Telescope in the SDSS r'-band. (a) The image previous to data reduction, where the border where the CCD was not exposed is visible. (b) The same image, after the data reduction. Here the center gradient is non-existent due to the flat-field correction, and there is generally less noise in the image as a result of the bias subtraction. The visible border in (a) is trimmed away to prevent negative counts as a result of the bias subtraction.

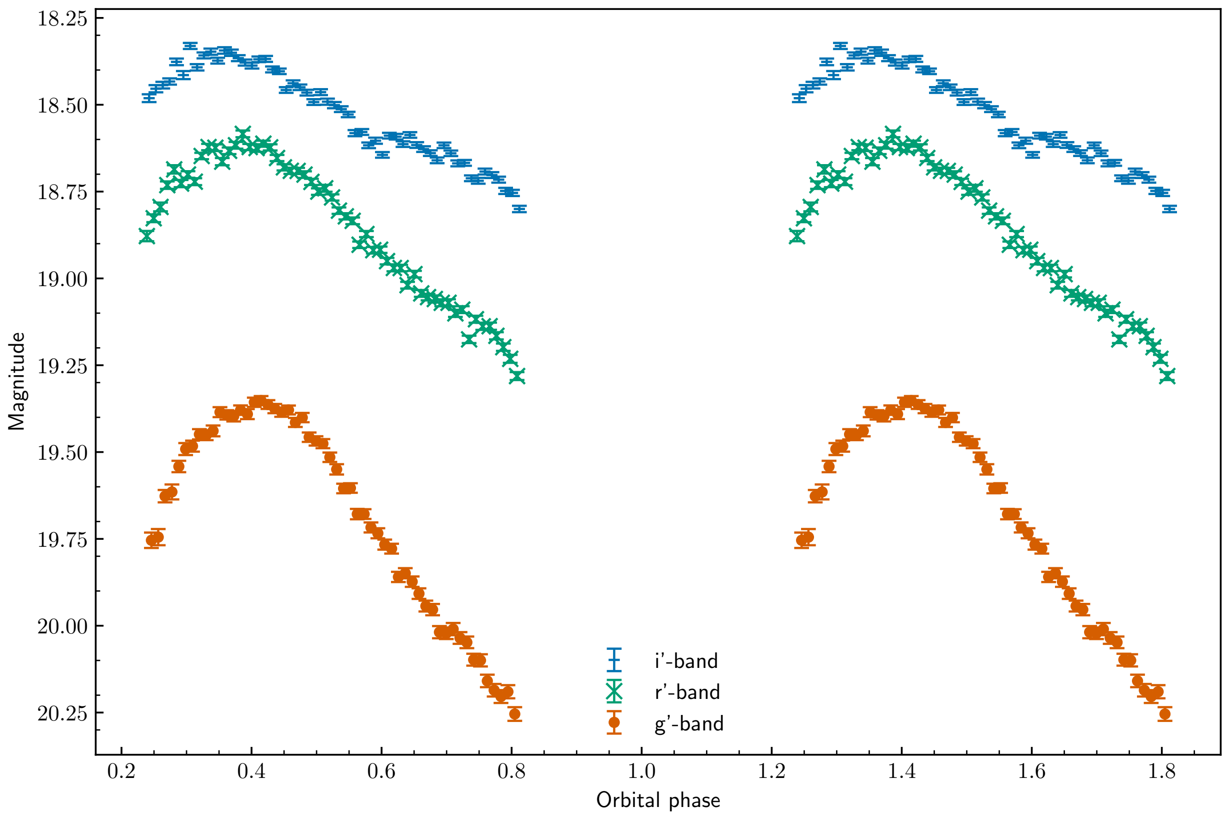


Figure 4.3: Light curves of PSR J1048+2339 in the SDSS g'-, r'- and i'-filters created with photometric images taken by the NOT the 18th of February 2020. The data is folded with an orbital period of 6 hours and $T_0 = 56637.598174$ MJD.

Color	Δmag_{day}	T_{day} (K)	T_{night} (K)	T_{irr} (K)	$L_{irr}(\mathrm{erg\,s^{-1}})$
g'-r'	0.715 ± 0.015	5135 ± 35	3850 ± 200	4670 ± 121	$(8.1 \pm 0.8) \times 10^{33}$

Table 4.2: The results calculated from the color, g'-r'. The color at orbital phase ($\phi = 0.4$), T_{day} found from table 5 in Pecaut et al. 2013 [78], T_{irr} calculated by Equation 3.8 and L_{irr} calculated by Equation 3.9.

2013 [78]. To account for extinction, we utilize the interstellar dust map [44] and subtract the color excess of $E(g' - r') = 0.02$ magnitudes from the apparent color to obtain the intrinsic color of the companion. To facilitate a meaningful comparison of colors, we perform this calculation at the same phase to the color analysis of the TJO data ($\phi = 0.4$).

At a phase of $\phi = 0.4$, we determine the intrinsic color to be $g' - r' = 0.715 \pm 0.015$. We then consult a table containing color-temperature relations and find that the corresponding surface temperature for this orbital phase is $T_{day} = 5135 \pm 35$K. It is unfortunately not possible to obtain a night-side temperature of the companion due to the phase coverage being incomplete. In order to be able to calculate an irradiation temperature and following irradiation luminosity we will assume a night-side temperature equal to what was found in chapter 3; $T_{night} = 3850 \pm 200$K. By using Equation 3.8, this then yields an irradiation temperature of $T_{irr} = 4670$K. Inserting T_{irr} into Equation 3.9, with the orbital separation of $a = 1.54463833 \times 10^{11}$cm, gives an irradiation luminosity of $L_{irr} = 8.1 \times 10^{33}\mathrm{erg\,s^{-1}}$. These results are summarized in Table 4.2.

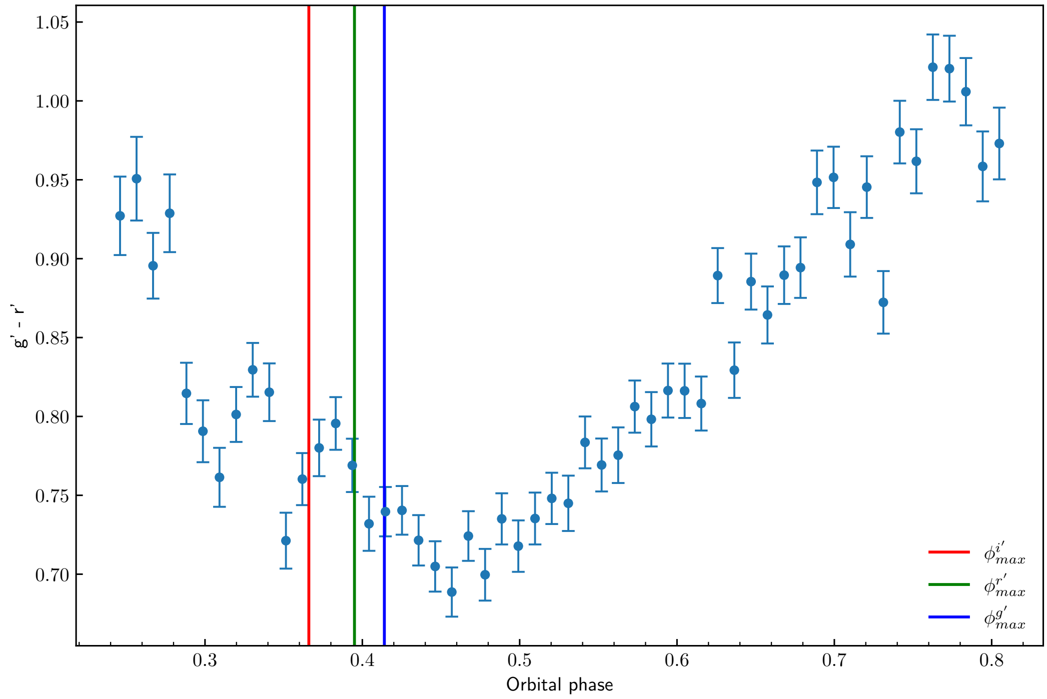


Figure 4.4: g'-r' color curve created with magnitude data of PSR J1048+2339 taken by the Nordic Optical Telescope. The curve displays evidence of asymmetric heating of the companion, with a minimum in the color curve at $\phi \simeq 0.45$. The ϕ_{max} value for the different filters are plotted for clarity.

A color curve plot is produced of all the available magnitude data created from the NOT observations. This can be seen in Figure 4.4. Due to the observations altering between filters for each new image, this was calculated through subtracting subsequent images in the r'-band from the g'-

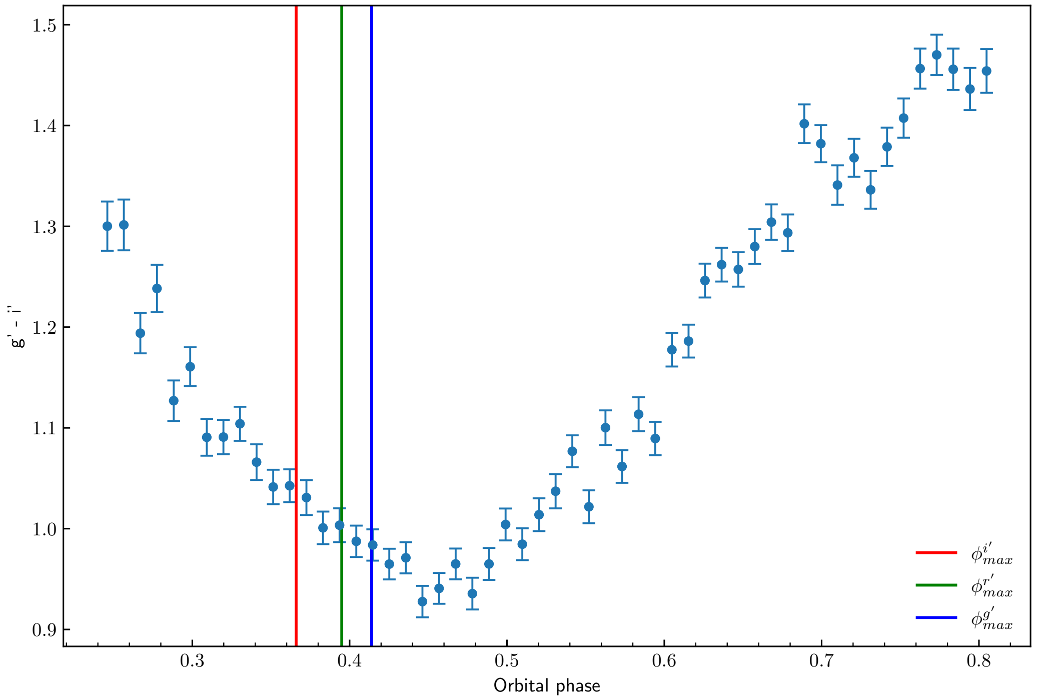


Figure 4.5: g'-i' color curve created with magnitude data of PSR J1048+2339 taken by the Nordic Optical Telescope. The curve displays evidence of asymmetric heating of the companion, with a minimum in the color curve at $\phi \simeq 0.45$. The ϕ_{max} value for the different filters are plotted for clarity.

band. The figure shows a clear indication that the companion is asymmetrically heated, however one interesting feature of the plot is that the point of highest temperature (smallest color index) does not coincide with the point of minimum magnitude in either the g'- or r'-band. The same is observed for the colors g'-i' and r'-i', which can be seen in Figures 4.5 and 4.6. These color curves display the same features as the g'-r' color, where the smallest color index do not coincide with the point of minimum magnitude. When comparing all of the color curves to those from the TJO data, we can see a similar trend where the smallest color index is around $\phi = 0.45$.

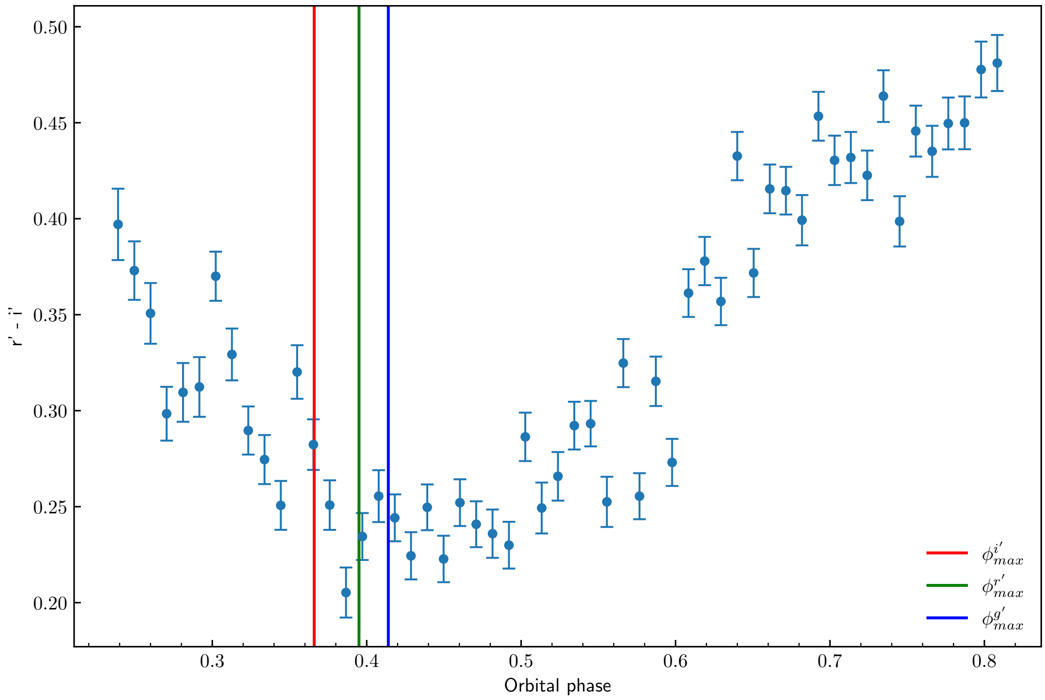


Figure 4.6: r'-i' color curve created with magnitude data of PSR J1048+2339 taken by the Nordic Optical Telescope. The curve displays evidence of asymmetric heating of the companion, with a minimum in the color curve at $\phi \simeq 0.45$. The ϕ_{max} value for the different filters are plotted for clarity.

Chapter 5

Discussion

5.1 Variable irradiation

We have analyzed the optical data for the redback system PSR J1048+2339 with two separate telescopes. Observations made by Yap et al. (2019) [108] and Cho et al. 2018 [21] both show modal changes in the system between an ellipsoidal modulation, pertaining to non-irradiated systems, and a sinusoidal modulation, usually found in irradiated systems. Since the observations from Yap et al. and Cho et al. are separated in time, it can be inferred that the system can periodically change between the modes. Our folded light curves from the TJO and NOT data displayed in chapters 3 and 4 show evidence that the system is irradiated. We can see in Figure 3.4 that a few of the data-points lie outside the main asymmetric light curve. The points corresponding to an increase in flux could be explained through flaring events, while a reduction in observed flux could be explained by a decrease in irradiation. With the help of Figure 3.3 we find that the images with a decrease in observed R-band flux (seen around phase $\phi \sim 0.6$) originate from the same night of observations, the 8th of May 2020 (see Fig. 5.1). The change is of about $\sim +0.2$ mag, which is not consistent with the change in magnitude observed during a modal change in 2018 [108] (see Sec. 1.3). The data show a similar change in the V- and I-band magnitudes, however they are not as consistent as for the R-band. This observed decrease in flux might be due to short time-scale variations in the irradiation of the companion, due to change in the heating mechanisms (i.e. small changes in the companion wind strength could displace an intra-binary shock leading to an increase or a decrease in irradiation (see Sec. 1.2.2)). In the observation by Yap et al., they discovered a decrease in r'-band magnitude of 0.3 at phase $\phi = 0.55$ about two weeks prior to observing J1048 in a new modal state. Given that our data shows a decrease of 0.2 magnitudes, it is possible that what we observed is the beginning of a change from sinusoidal to ellipsoidal modulation.

Interestingly, when describing the theoretical background for the sudden change from ellipsoidal modulation to sinusoidal modulation in 2018, Yap et al. interpreted the observed X-ray modulation and decrease in magnitude as the existence of an intra-binary shock. This was dismissed since in order to match the observations, the system would have to be observed almost edge-on. Gamma ray eclipses were detected for J1048 previously in 2023 [22] leading to a constraint on the minimum orbital inclination of $i > 80.4°$. Therefore the theory could hold true, and the change in optical modulation could be due to a change in the magnetic field of the companion resulting in the intra-binary shock moving towards or away from the companion.

5.2 Asymmetric light curves

One of the main results presented in Chapters 3 and 4 are the large phase shifts in the optical light curves (see Table 3.4 and 4.1). These phase shifts suggest that the irradiation from J1048 is not concentrated on the front-facing side of the companion, but instead shifted towards its trailing side. Asymmetric light curves like these appear to be common among redback systems.

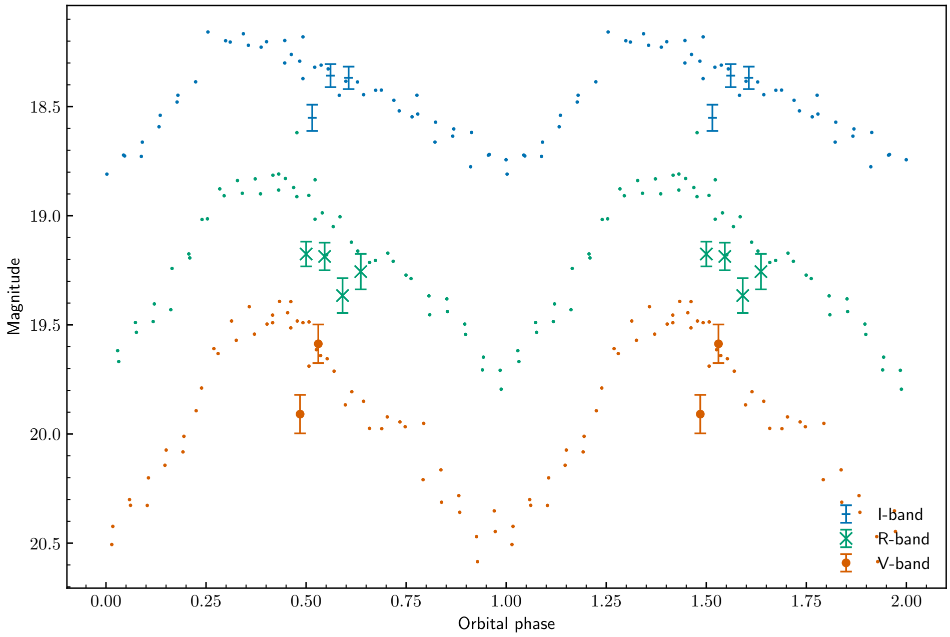


Figure 5.1: The observations taken the 8th of May 2020 plotted with the full coverage obseravations taken the 18th and 19th of February 2020. Here we can clearly see the decrease in R-band flux compared to the data from February.

We can find a reported phase shift in Schroeder & Halpern (2014), who report a phase shift of $\phi = -1.4 \times 10^{-2} \pm 5 \times 10^{-4}$ for the redback PSR J2215+3115, about 7 times smaller than the values reported in Tables 3.4 and 4.1. Asymmetry has also been observed in two transitional millisecond pulsars (tMSPs), systems which transition between accretion-powered and radio-powered pulsar states [10]. Both PSR J1023+0038 [94] and PSR J1227-4853 [94, 29] have been observed displaying asymmetrical light curves, where PSR J1227-4853 has a reported phase shift of $\Delta\phi = 0.06$ [29]. All of these light curves show a maximum flux shifted towards lower phases, providing evidence that the a point on trailing side of the companion is hotter than the point directly facing the neutron star.

There are several possible theories to describe the nature of the phase shift in the light curves, but there is no mechanism that is widely accepted. One possible explanation is that the asymmetry is due to the intra-binary shock (IBS), which arises when the relativistic pulsar wind interacts with the companion wind or ejected matter. The intra-binary shock can wrap around the companion given the right wind and orbital parameters [86]. The companion is then heated by the non-thermal X-ray emission produced in the IBS, and due to being swept back across the trailing side of the companion, the trailing side will be radiated. In Romani & Sanchez's work, their models simulating this swept back IBS demonstrate good consistency with the observed data. From Figure 1.5 it can be inferred that the v_{rel} of the system is $v_{rel} < 1$ as this is what leads to the asymmetric IBS.

Another possible theory is the heat distribution from a direct heating model, where the energy is re-radiated at the exact location where it was deposited [15]. This model alone is incapable of accounting for the asymmetric light curves, but the dynamics on the surface of the companion may be able to produce them. In the direct heating model, the side that faces the neutron star will become strongly heated, creating a large temperature gradient between the day and night side of the companion. This may create strong, circulatory winds as the significant temperature difference between the day and night sides provides enough energy to drive winds and allow the flow of heat through the atmosphere, where the spin of the companion drives the circulation within the atmosphere [94].

5.2.1 Passband-specific phase shifts

In the data aquired by both the TJO and NOT, we find that the phase shifts are not equal from filter to filter. The longer wavelengths display a larger shift from the expected phase of $\phi = 0.5$. The largest difference in ϕ_{max} can be found for $\phi_{max}^{(g')} - \phi_{max}^{(i')} = 0.048 \pm 0.004$. Similar observations can be seen for the two previously mentioned tMSPs, where light curves of the i'- and g'- band are significantly more shifted than those of the u'-band. Even though the sinusoidal modulations are produced due to irradiation, the ellipsoidal variation will superposed on top. Indeed, Stringer et al. (2021) [93] mentions that the redder bands will have a larger relative contribution the ellipsoidal variation to the light curve shape. Conversely, we may expect smaller phase shifts from bluer passbands than those in our observations. This fraction of ellipsoidal variation in the light curves is related to the orbital inclination of the system, as high inclinations means observing a larger change in the projected area of the tidally distorted companion.

As colors are directly related to the surface temperature of the star, color curves could help determine the phase where we observe the highest surface temperature. In the color plot based on data from the NOT (see Figs. 4.4, 4.5 and 4.6) we see the that the phase of minimum magnitude do not coincide with the minimum in the color curve for any ϕ_{max} and filter combination. However, the separate color plots do coincide will with each other with a minimum color value at $\phi \simeq 0.45$. Thus, we find evidence that the hottest spot on the companion surface is not seen at the orbital phase of highest flux in the observed passbands, but rather later in the orbit (but still earlier than phase $\phi = 0.5$). As discussed in the previous paragraph, the bluer passbands are show a smaller phase shift than for the red. If we observed the target in shorter wavelength filters, this could result in the phase shift converging towards the phase of minimum color $\phi \simeq 0.45$.

For a theory consisting of a direct heating model with no heat distribution, we would expect the point of maximum flux, and minimum color index, of an irradiated companion to be at phase $\phi = 0.5$. As we have observed, this is not the case for J1048, but rather we observe a minimum color index and maximum flux appearing on the trailing side of the companion. Due to observations of other redbacks and tMSPs showing similar phase shifts toward lower orbital phases, this could mean that the asymmetry originates from an intra-binary shock sweeping over the trailing side of the companion, given that these systems also display evidence of the presence of an intra-binary shock. This explanation for the asymmetry is qualitatively consistent with the models of Romani & Sanchez [86], where they modeled the how intra-binary shock geometry affects light curves (see Fig. 1.5).

5.3 Irradiation temperature and luminosity

In chapters 3 and 4 we presented the day and night side temperatures as a result of the analysis of the companion color. The parts of the results of this analysis are summarized in Table 5.1.

Telescope	Date	T_{day} (K)	T_{night} (K)	T_{irr} (K)	$L_{irr}(\mathrm{erg\,s^{-1}})$
TJO	Feb-May 2020	$4290 - 4830$	$3490 - 4390$	$3200 - 4500$	$1.2 \times 10^{33} - 6.2 \times 10^{33}$
NOT	18th of Feb 2020	$5100 - 5170$		$4550 - 4790$	$7.3 \times 10^{33} - 8.9 \times 10^{33}$

Table 5.1: A summary of the results of the color analysis performed in chapters 3 and 4.

We can see a discrepancy in the values of T_{irr} and L_{irr} between the TJO and the NOT data. This is a result of the difference in the day side temperatures found from the two telescopes. The main uncertainty of the color values is from the differential photometry, which was done similarly for both data-sets. Therefore, it is possible that this discrepancy is due to the discrete table that was used for finding the temperatures, as there is only one other discrete temperature in-between the temperature values found in this work. Furthermore, the calculated color indices for the TJO were found by the use of the binned light curves. These can have artificially increased the color-value due to the averaging done. Therefore, it could be beneficial to compare the NOT data, with the

TJO data taken the same night. In doing so, we find a slightly V-R color of ~ 0.57 at phase $\phi \simeq 0.4$. This results in a day side temperature closer to $T_{day} = 4830$ K, which equals a K3V spectral type companion, whereas the NOT data equals a K2V spectral type.

Given the spin-down luminosity of the system, it is possible to estimate how efficiently the binary converts the pulsar energy to observable optical radiation. We define the heating effciency as $\epsilon = L_{irr}/|\dot{E_{int}}|$, where L_{irr} is the irradiation luminosity calculated from the irradation temperature and $\dot{E_{int}}$ is the intrinsic spin-down luminosity of the millisecond pulsar. Inserting the intrinsic spin-down luminosity of J1048 and the irradiated luminosities we find efficiencies of

$$\begin{aligned}\epsilon_{V-R} &= 0.31,\\ \epsilon_{V-I} &= 0.24,\\ \epsilon_{g'-r'} &= 0.67.\end{aligned}$$

Comparing these numbers to the irradiation efficiency measured in black widow systems (Table A1 in [68]), we find that the observed effiencies of J1048 seem to correspond well with the data. Interesting to note that there are black widow systems that display an irradiation efficiency of more than 1. This could be a result of the intrinsic spin-down luminosity not radiating isotropically. The angular distribution of $\dot{E}$ is found to be proportional to $\sin^4\theta$, where θ is the angle to the rotational axis [98], and the spin-down luminosity is therefore strongest along the equator and dissipates towards the poles.

5.4 Conclusion

In this master thesis we have presented the theoretical background on neutron stars and robotic optical observing using sky surveys and small telescopes. This has been applied to the development and implementation of a pipeline to be used with the Joan Oró Telescope (TJO). We have also analyzed data from the TJO and the Nordic Optical Telescope on the compact binary millisecond pulsar PSR J1048+2339.

The TJO-pipeline was developed to integrate with the Joan Oró telescope in order to streamline the observations and make it easier as an observer to get quick information regarding the observations taken with the telescope. The pipeline has been up and running since early February, and has proven useful with the notifications to observers whenever there are new observations.

We have presented the results of optical photometric observations of PSR J1048+2339 taken both by the TJO and the NOT. The data show consistent asymmetrical light curves, pointing to the presence of a swept back intra-binary shock. The light curves show non-identical phase shift for the different passbands applied to the observations, which might be due to the underlying ellipsoidal modulation being more protrudent in the red filters. We have found that the color curves do not show minima where the light curves have their maxima, and discussed that this might be due to a hotspot which has the highest temperature at phase $\phi = 0.45$. Finally, the irradiating luminosity of J1048 was found to be similar to those observed in black widow systems. Future observations of J1048 should incorporate bluer passbands in order to assess if the phase shift will eventually converge towards the phase of highest temperature.

Bibliography

[1] URL: http://www.not.iac.es/instruments/.

[2] B. P. Abbott et al. 'GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral'. In: 119.16, 161101 (Oct. 2017), p. 161101. DOI: 10.1103/PhysRevLett.119.161101. arXiv: 1710.05832 [gr-qc].

[3] Leandro G Althaus et al. 'Evolutionary and pulsational properties of white dwarf stars'. In: *The Astronomy and Astrophysics Review* 18 (2010), pp. 471–566.

[4] T. Andersen et al. 'Active Optics on the Nordic Optical Telescope'. In: *European Southern Observatory Conference and Workshop Proceedings*. Vol. 42. European Southern Observatory Conference and Workshop Proceedings. June 1992, p. 311.

[5] Anne M. Archibald et al. 'A Radio Pulsar/X-ray Binary Link'. In: *Science* 324.5933 (June 2009), pp. 1411–1414. DOI: 10.1126/science.1172740. URL: https://doi.org/10.1126%5C%2Fscience.1172740.

[6] Arne Ardeberg. *Some properties of the Nordic Optical Telescope*. 1989.

[7] D. C. Backer et al. 'A millisecond pulsar'. In: 300.5893 (Dec. 1982), pp. 615–618. DOI: 10.1038/300615a0.

[8] Jennifer Barnes. 'The Physics of Kilonovae'. In: *Frontiers in Physics* 8, 355 (Oct. 2020), p. 355. DOI: 10.3389/fphy.2020.00355.

[9] Werner Becker, Mike G Bernhardt and Axel Jessner. 'Autonomous spacecraft navigation with pulsars'. In: *arXiv preprint arXiv:1305.4842* (2013).

[10] Sudip Bhattacharyya, Alessandro Papitto and Dipankar Bhattacharya. *Millisecond Pulsars*. Springer, 2022.

[11] M. F. Bode and W. T. Vestrand. 'Small and Robotic Telescopes in the Era of Massive Time-Domain Surveys'. In: *New Horizons in Time Domain Astronomy*. Ed. by Elizabeth Griffin, Robert Hanisch and Rob Seaman. Vol. 285. Apr. 2012, pp. 235–238. DOI: 10.1017/S1743921312000658.

[12] IS Bowen. 'Astronomical optics'. In: *Annual Review of Astronomy and Astrophysics* 5.1 (1967), pp. 45–66.

[13] Timothy D. Brandt et al. 'The Ages of Type Ia Supernova Progenitors'. In: 140.3 (Sept. 2010), pp. 804–816. DOI: 10.1088/0004-6256/140/3/804. arXiv: 1002.0848 [astro-ph.GA].

[14] R. P. Breton et al. 'Discovery of the Optical Counterparts to Four Energetic Fermi Millisecond Pulsars'. In: 769.2, 108 (June 2013), p. 108. DOI: 10.1088/0004-637X/769/2/108. arXiv: 1302.1790 [astro-ph.HE].

[15] R. P. Breton et al. 'KOI 1224: A FOURTH BLOATED HOT WHITE DWARF COMPANION FOUND WITHiKEPLER/i'. In: *The Astrophysical Journal* 748.2 (Mar. 2012), p. 115. DOI: 10.1088/0004-637x/748/2/115. URL: https://doi.org/10.1088%5C%2F0004-637x%5C%2F748%5C%2F2%5C%2F115.

[16] Michael S. Briggs et al. 'BATSE Observations of the Large-Scale Isotropy of Gamma-Ray Bursts'. In: 459 (Mar. 1996), p. 40. DOI: 10.1086/176867. arXiv: astro-ph/9509078 [astro-ph].

[17] Manisha Caleb et al. 'Discovery of a radio-emitting neutron star with an ultra-long spin period of 76 s'. In: *Nature astronomy* 6.7 (2022), pp. 828–836.

[18] K. C. Chambers et al. *The Pan-STARRS1 Surveys.* 2019. arXiv: 1612.05560 [astro-ph.IM].

[19] Nicolas Chamel and Pawel Haensel. 'Physics of neutron star crusts'. In: *Living Reviews in relativity* 11 (2008), pp. 1–182.

[20] Rakesh Vidya Chandra and Bala Subrahmanyam Varanasi. *Python requests essentials.* Packt Publishing Birmingham, UK, 2015.

[21] Patricia B. Cho, Jules P. Halpern and Slavko Bogdanov. 'Variable Heating and Flaring of Three Redback Millisecond Pulsar Companions'. In: 866.1, 71 (Oct. 2018), p. 71. DOI: 10.3847/1538-4357/aade92. arXiv: 1809.00215 [astro-ph.HE].

[22] CJ Clark et al. 'Neutron star mass estimates from gamma-ray eclipses in spider millisecond pulsar binaries'. In: *Nature Astronomy* (2023), pp. 1–12.

[23] James J Condon and Scott M Ransom. *Essential radio astronomy.* Vol. 2. Princeton University Press, 2016.

[24] C. M. Copperwheat et al. 'Liverpool telescope 2: a new robotic facility for rapid transient follow-up'. In: *Experimental Astronomy* 39.1 (Mar. 2015), pp. 119–165. DOI: 10.1007/s10686-015-9447-0. arXiv: 1410.1731 [astro-ph.IM].

[25] E. Costa et al. 'Discovery of an X-ray afterglow associated with the γ-ray burst of 28 February 1997'. In: 387.6635 (June 1997), pp. 783–785. DOI: 10.1038/42885. arXiv: astro-ph/9706065 [astro-ph].

[26] Arthur N Cox. *Allen's astrophysical quantities.* Springer, 2015.

[27] Matt Craig et al. *astropy/ccdproc: v1.3.0.post1.* Dec. 2017. DOI: 10.5281/zenodo.1069648. URL: https://doi.org/10.5281/zenodo.1069648.

[28] HT Cromartie et al. 'Six new millisecond pulsars from arecibo searches of fermi gamma-ray sources'. In: *The Astrophysical Journal* 819.1 (2016), p. 34.

[29] D. de Martino et al. 'Multiwavelength observations of the transitional millisecond pulsar binary XSS J12270-4859'. In: 454.2 (Dec. 2015), pp. 2190–2198. DOI: 10.1093/mnras/stv2109. arXiv: 1509.02765 [astro-ph.HE].

[30] JS Deneva et al. 'Multiwavelength observations of the redback millisecond pulsar J1048+2339'. In: *The Astrophysical Journal* 823.2 (2016), p. 105.

[31] JS Deneva et al. 'Timing of Eight Binary Millisecond Pulsars Found with Arecibo in Fermi-LAT Unidentified Sources'. In: *The Astrophysical Journal* 909.1 (2021), p. 6.

[32] V. S. Dhillon et al. 'ULTRACAM: an ultrafast, triple-beam CCD camera for high-speed astrophysics'. In: 378.3 (July 2007), pp. 825–840. DOI: 10.1111/j.1365-2966.2007.11881.x. arXiv: 0704.2557 [astro-ph].

[33] Vik Dhillon. *Observatory sites.* 2013. URL: http://www.vikdhillon.staff.shef.ac.uk/teaching/phy217/telescopes/phy217_tel_sites.html.

[34] George Djorgovski. 'Astrophysics in the era of massive time-domain surveys'. In: *The third hot-wiring the transient universe workshop.* 2014, p. 215.

[35] S. G. Djorgovski et al. 'Exploring the Time Domain with Synoptic Sky Surveys'. In: *New Horizons in Time Domain Astronomy.* Ed. by Elizabeth Griffin, Robert Hanisch and Rob Seaman. Vol. 285. Apr. 2012, pp. 141–146. DOI: 10.1017/S1743921312000488. arXiv: 1111.2078 [astro-ph.IM].

[36] S. George Djorgovski et al. 'Sky Surveys'. In: *Planets, Stars and Stellar Systems. Volume 2: Astronomical Techniques, Software and Data.* Ed. by Terry D. Oswalt and Howard E. Bond. 2013, p. 223. DOI: 10.1007/978-94-007-5618-2_5.

[37] SG Djorgovski et al. 'Searches for rare and new types of objects'. In: *arXiv preprint astro-ph/0012453* (2000).

[38] Anlaug Amanda Djupvik and Johannes Andersen. 'The Nordic Optical Telescope'. In: *Highlights of Spanish astrophysics V.* Springer, 2010, pp. 211–218.

[39] David Eichler et al. 'Nucleosynthesis, neutrino bursts and γ-rays from coalescing neutron stars'. In: 340.6229 (July 1989), pp. 126–128. DOI: 10.1038/340126a0.

[40] David Fernández et al. 'Sky quality at the Montsec Astronomical Observatory and commissioning of the Joan Oró Telescope'. In: ().

[41] Lilia Ferrario and DT Wickramasinghe. 'Magnetic fields and rotation in white dwarfs and neutron stars'. In: *Monthly Notices of the Royal Astronomical Society* 356.2 (2005), pp. 615–620.

[42] Pranab Ghosh. *Rotation and accretion powered pulsars*. Vol. 7. World Scientific, 2007.

[43] Jonathan Granot and Re'em Sari. 'The Shape of Spectral Breaks in Gamma-Ray Burst Afterglows'. In: 568.2 (Apr. 2002), pp. 820–829. DOI: 10.1086/338966. arXiv: astro-ph/0108027 [astro-ph].

[44] Gregory M. Green et al. 'A 3D Dust Map Based on iGaia/i, Pan-STARRS 1, and 2MASS'. In: *The Astrophysical Journal* 887.1 (Dec. 2019), p. 93. DOI: 10.3847/1538-4357/ab5362. URL: https://doi.org/10.3847%2F1538-4357%2Fab5362.

[45] Jonathan Grindlay et al. 'Opening the 100-year window for time-domain astronomy'. In: *Proceedings of the International Astronomical Union* 7.S285 (2011), pp. 29–34.

[46] Paweł Haensel, Aleksander Yu Potekhin and Dmitry G Yakovlev. *Neutron Stars 1*. Springer, 2007.

[47] Martin Harwit. 'The number of Class A phenomena characterizing the universe'. In: *Quarterly Journal of the Royal Astronomical Society* 16 (1975), pp. 378–409.

[48] A Hewish et al. 'Observation of a rapidly pulsating radio source (reprinted from Nature, February 24, 1968)'. In: *Nature* 224.5218 (1969), p. 472.

[49] CJ Horowitz et al. 'Dynamical response of the nuclear "pasta" in neutron star crusts'. In: *Physical Review C* 72.3 (2005), p. 035801.

[50] IEEC. *https://www.ieec.cat/content/18/laia/*. 2023. URL: https://www.ieec.cat/content/18/laia/.

[51] IEEC. *New filters for the Joan Oró Telescope of the Montsec Observatory*. Jan. 2023. URL: https://www.ieec.cat/content/media/news/en/1049.

[52] IEEC. *Telescope and dome*. 2023. URL: https://www.ieec.cat/content/18/telescope-and-dome/.

[53] IEEC. *TJO Generalities*. 2023. URL: https://www.ieec.cat/content/18/generalities/.

[54] James Jeans. *Astronomy and cosmogony*. CUP Archive, 1929.

[55] Mario Jurić et al. 'The LSST data management system'. In: *arXiv preprint arXiv:1512.07914* (2015).

[56] Amanda I Karakas and John C Lattanzio. 'The Dawes review 2: nucleosynthesis and stellar yields of low-and intermediate-mass single stars'. In: *Publications of the Astronomical Society of Australia* 31 (2014), e030.

[57] Victoria M Kaspi and Andrei M Beloborodov. 'Magnetars'. In: *Annual Review of Astronomy and Astrophysics* 55 (2017), pp. 261–301.

[58] Ray W. Klebesadel, Ian B. Strong and Roy A. Olson. 'Observations of Gamma-Ray Bursts of Cosmic Origin'. In: 182 (June 1973), p. L85. DOI: 10.1086/181225.

[59] James M Lattimer and Maddappa Prakash. 'The physics of neutron stars'. In: *science* 304.5670 (2004), pp. 536–542.

[60] JM Lattimer and M Prakash. 'Neutron star structure and the equation of state'. In: *The Astrophysical Journal* 550.1 (2001), p. 426.

[61] Jongsu Lee et al. 'A Comparison of Millisecond Pulsar Populations between Globular Clusters and the Galactic Field'. In: 944.2, 225 (Feb. 2023), p. 225. DOI: 10.3847/1538-4357/acb5a3. arXiv: 2302.08776 [astro-ph.HE].

[62] Li-Xin Li and Bohdan Paczyński. 'Transient Events from Neutron Star Mergers'. In: 507.1 (Nov. 1998), pp. L59–L62. DOI: 10.1086/311680. arXiv: astro-ph/9807272 [astro-ph].

[63] M. Linares. 'Super-Massive Neutron Stars and Compact Binary Millisecond Pulsars'. In: *Multifrequency Behaviour of High Energy Cosmic Sources - XIII. 3-8 June 2019. Palermo*. Dec. 2020, 23, p. 23. DOI: 10.22323/1.362.0023. arXiv: 1910.09572 [astro-ph.HE].

[64] CP Lorenz, DG Ravenhall and CJ Pethick. 'Neutron star crusts'. In: *Physical Review Letters* 70.4 (1993), p. 379.

[65] Duncan R Lorimer. 'Binary and millisecond pulsars'. In: *Living reviews in relativity* 11.1 (2008), pp. 1–90.

[66] S. C. Lundgren et al. 'Optical Studies of Millisecond Pulsar Companions'. In: 458 (Feb. 1996), p. L33. DOI: 10.1086/309920.

[67] R. N. Manchester. 'Millisecond Pulsars, their Evolution and Applications'. In: *Journal of Astrophysics and Astronomy* 38.3, 42 (Sept. 2017), p. 42. DOI: 10.1007/s12036-017-9469-2. arXiv: 1709.09434 [astro-ph.HE].

[68] D. Mata Sánchez et al. 'A black widow population dissection through HiPERCAM multiband light-curve modelling'. In: 520.2 (Apr. 2023), pp. 2217–2244. DOI: 10.1093/mnras/stad203. arXiv: 2301.07132 [astro-ph.HE].

[69] Gregory L Matloff. 'Interstellar message plaques'. In: *Deep-Space Probes: To the Outer Solar System and Beyond* (2005), pp. 177–184.

[70] Peter Meszaros. 'Gamma-ray bursts'. In: *Reports on Progress in Physics* 69.8 (2006), p. 2259.

[71] Brian D Metzger. 'Kilonovae'. In: *Living reviews in relativity* 20.1 (2017), p. 3.

[72] Davide Miceli and Lara Nava. 'Gamma-Ray Bursts Afterglow Physics and the VHE Domain'. In: *Galaxies* 10.3 (May 2022), p. 66. DOI: 10.3390/galaxies10030066. arXiv: 2205.12146 [astro-ph.HE].

[73] A. Miraval Zanon et al. 'Evidence of intra-binary shock emission from the redback pulsar PSR J1048+2339'. In: 649, A120 (May 2021), A120. DOI: 10.1051/0004-6361/202040071. arXiv: 2103.06317 [astro-ph.HE].

[74] Joonas Nättilä and Jari JE Kajava. 'Fundamental physics with neutron stars'. In: *arXiv preprint arXiv:2211.15721* (2022).

[75] VN Obridko and BD Shelting. 'On the negative correlation between solar activity and solar rotation rate'. In: *Astronomy Letters* 42 (2016), pp. 631–637.

[76] A. Papitto et al. 'Swings between rotation and accretion power in a binary millisecond pulsar'. In: *Nature* 501.7468 (Sept. 2013), pp. 517–520. DOI: 10.1038/nature12470. URL: https://doi.org/10.1038%5C%2Fnature12470.

[77] GG Pavlov et al. 'Neutron star atmospheres'. In: *NATO ASI Series C Mathematical and Physical Sciences-Advanced Study Institute* 450 (1995), pp. 71–90.

[78] Mark J. Pecaut and Eric E. Mamajek. 'Intrinsic Colors, Temperatures, and Bolometric Corrections of Pre-main-sequence Stars'. In: 208.1, 9 (Sept. 2013), p. 9. DOI: 10.1088/0067-0049/208/1/9. arXiv: 1307.2657 [astro-ph.SR].

[79] ES Phinney. 'Pulsars as probes of newtonian dynamical systems'. In: *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences* 341.1660 (1992), pp. 39–75.

[80] Tsvi Piran. 'Gamma-ray bursts and the fireball model'. In: *Physics Reports* 314.6 (1999), pp. 575–667.

[81] E. J. Polzin et al. 'Study of spider pulsar binary eclipses and discovery of an eclipse mechanism transition'. In: 494.2 (May 2020), pp. 2948–2968. DOI: 10.1093/mnras/staa596. arXiv: 2003.02335 [astro-ph.HE].

[82] Arne Rau et al. 'Exploring the Optical Transient Sky with the Palomar Transient Factory'. In: 121.886 (Dec. 2009), p. 1334. DOI: 10.1086/605911. arXiv: 0906.5355 [astro-ph.CO].

[83] Arne Rau et al. 'Exploring the Optical Transient Sky with the Palomar Transient Factory'. In: 121.886 (Dec. 2009), p. 1334. DOI: 10.1086/605911. arXiv: 0906.5355 [astro-ph.CO].

[84] A. Rebassa-Mansergas et al. 'Where are the double-degenerate progenitors of Type Ia supernovae?' In: 482.3 (Jan. 2019), pp. 3656–3668. DOI: 10.1093/mnras/sty2965. arXiv: 1809.07158 [astro-ph.SR].

[85] Adam G. Riess et al. 'Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant'. In: 116.3 (Sept. 1998), pp. 1009–1038. DOI: 10.1086/300499. arXiv: astro-ph/9805201 [astro-ph].

[86] Roger W. Romani and Nicolas Sanchez. 'INTRA-BINARY SHOCK HEATING OF BLACK WIDOW COMPANIONS'. In: *The Astrophysical Journal* 828.1 (Aug. 2016), p. 7. DOI: 10.3847/0004-637x/828/1/7. URL: https://doi.org/10.3847%5C%2F0004-637x%5C%2F828%5C%2F1%5C%2F7.

[87] Jayanta Roy et al. 'Discovery of Psr J1227-4853: A Transition from a Low-mass X-Ray Binary to a Redback Millisecond Pulsar'. In: 800.1, L12 (Feb. 2015), p. L12. DOI: 10.1088/2041-8205/800/1/L12. arXiv: 1412.4735 [astro-ph.HE].

[88] Evan Scannapieco and Lars Bildsten. 'The Type Ia Supernova Rate'. In: 629.2 (Aug. 2005), pp. L85–L88. DOI: 10.1086/452632. arXiv: astro-ph/0507456 [astro-ph].

[89] Edward F. Schlafly and Douglas P. Finkbeiner. 'MEASURING REDDENING WITH SLOAN DIGITAL SKY SURVEY STELLAR SPECTRA AND RECALIBRATING SFD'. In: *The Astrophysical Journal* 737.2 (Aug. 2011), p. 103. DOI: 10.1088/0004-637x/737/2/103. URL: https://doi.org/10.1088%5C%2F0004-637x%5C%2F737%5C%2F2%5C%2F103.

[90] James A. Seibert, John M. Boone and Karen K. Lindfors. 'Flat-field correction technique for digital detectors'. In: *Medical Imaging 1998: Physics of Medical Imaging*. Ed. by James T. Dobbins and John M. Boone. Vol. 3336. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. July 1998, pp. 348–354. DOI: 10.1117/12.317034.

[91] SG Sichevskiy, AV Mironov and O Yu Malkov. 'Classification of stars with WBVR photometry'. In: *Astronomische Nachrichten* 334.8 (2013), pp. 832–834.

[92] Stephen J. Smartt. 'Progenitors of Core-Collapse Supernovae'. In: 47.1 (Sept. 2009), pp. 63–106. DOI: 10.1146/annurev-astro-082708-101737. arXiv: 0908.0700 [astro-ph.SR].

[93] Jay Strader et al. 'Optical spectroscopy and demographics of redback millisecond pulsar binaries'. In: *The Astrophysical Journal* 872.1 (2019), p. 42.

[94] J. G. Stringer et al. 'Optical photometry of two transitional millisecond pulsars in the radio pulsar state'. In: 507.2 (Oct. 2021), pp. 2174–2191. DOI: 10.1093/mnras/stab2167. arXiv: 2107.11306 [astro-ph.HE].

[95] T. Sun et al. 'The Changing Event Survey: Transients, Space Debris & NEOs'. In: *Revista Mexicana de Astronomia y Astrofisica Conference Series*. Vol. 53. Revista Mexicana de Astronomia y Astrofisica Conference Series. Sept. 2021, pp. 166–168. DOI: 10.22201/ia.14052059p.2021.53.34.

[96] Masaomi Tanaka et al. 'Properties of Kilonovae from Dynamical and Post-merger Ejecta of Neutron Star Mergers'. In: 852.2, 109 (Jan. 2018), p. 109. DOI: 10.3847/1538-4357/aaa0cb. arXiv: 1708.09101 [astro-ph.HE].

[97] A. Tchekhovskoy, A. Spitkovsky and J. G. Li. 'Time-dependent 3D magnetohydrodynamic pulsar magnetospheres: oblique rotators.' In: 435 (Aug. 2013), pp. L1–L5. DOI: 10.1093/mnrasl/slt076. arXiv: 1211.2803 [astro-ph.HE].

[98] Alexander Tchekhovskoy, Alexander Philippov and Anatoly Spitkovsky. 'Three-dimensional analytical description of magnetized winds from oblique pulsars'. In: 457.3 (Apr. 2016), pp. 3384–3395. DOI: 10.1093/mnras/stv2869. arXiv: 1503.01467 [astro-ph.HE].

[99] *The three Galileos the man, the spacecraft the telescope*. Vol. 220. Astrophysics and Space Science Library. Jan. 1997. DOI: 10.1007/978-94-015-8790-7.

[100] *Time domain astronomy*. Accessed: 2023-04-21. 2023. URL: https://www.cfa.harvard.edu/research/topic/time-domain-astronomy.

[101] Josep M Trigo-Rodrıguez et al. 'Asteroids, Comets and Meteorite-Dropping Bolides Studied from The Montsec Astronomical Observatory'. In: *Assessment and Mitigation of Asteroid Impact Hazards: Proceedings of the 2015 Barcelona Asteroid Day*. Springer. 2017, pp. 243–256.

[102] V. V. Usov. 'Millisecond pulsars with extremely strong magnetic fields as a cosmological source of γ-ray bursts'. In: 357.6378 (June 1992), pp. 472–474. DOI: 10.1038/357472a0.

[103] E. P. J. van den Heuvel and J. van Paradijs. 'Fate of the companion stars of ultra-rapid pulsars'. In: 334.6179 (July 1988), pp. 227–228. DOI: 10.1038/334227a0.

[104] J. van Paradijs et al. 'Transient optical emission from the error box of the γ-ray burst of 28 February 1997'. In: 386.6626 (Apr. 1997), pp. 686–689. DOI: 10.1038/386686a0.

[105] Guillaume Voisin et al. 'A model for redistributing heat over the surface of irradiated spider companions'. In: *Monthly Notices of the Royal Astronomical Society* 499.2 (Sept. 2020), pp. 1758–1768. DOI: 10.1093/mnras/staa2876. URL: https://doi.org/10.1093%5C%2Fmnras%5C%2Fstaa2876.

[106] Gentaro Watanabe et al. 'Simulation of transitions between "pasta" phases in dense matter'. In: *Physical review letters* 94.3 (2005), p. 031101.

[107] Alexandra Witze. 'NASA test proves pulsars can function as a celestial GPS'. In: *Nature* 553.7688 (2018), pp. 261–263.

[108] Yee Xuan Yap et al. 'Face changing companion of the redback millisecond pulsar PSR J1048+ 2339'. In: *Astronomy & Astrophysics* 621 (2019), p. L9.

[109] Martin V Zombeck. *Handbook of space astronomy and astrophysics*. Cambridge University Press, 2006.

Appendix

Source code for the TJO-pipeline.

A TJOpipeline.py

```
import email
import os
import config
import sys
import argparse
import smtplib
from os.path import basename
from email.mime.application import MIMEApplication
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email.utils import COMMASPACE, formatdate
import numpy as np
import download
import characterize
import analyze


"""def send_mail(send_from, send_to, subject, text, files=None, server="127.0.0.1"):
    # assert isinstance(send_to, list)

    smtpObj = smtplib.SMTP("smtp.office365.com", 587)
    smtpObj.ehlo()
    smtpObj.starttls()
    smtpObj.login(send_from, "ywt6nrbx2n8")

    msg = MIMEMultipart()
    msg["Subject"] = subject
    msg["From"] = send_from
    msg["To"] = send_to

    msg.attach(MIMEText(text))

    for f in files:
        with open(f, "rb") as fil:
            part = MIMEApplication(fil.read(), Name=basename(f))
        # After the file is closed
        part["Content-Disposition"] = 'attachment; filename="%s"' % basename(f)
        msg.attach(part)

    smtpObj.sendmail(send_from, send_to, msg.as_string())
    smtpObj.close()
"""


#ML, May2023: you can now activate debug mode from config.py
if (config.DEBUG==1): print("DEBUG mode active, higher verbosity")


parser = argparse.ArgumentParser()

parser.add_argument(
    "-m",
    "--mode",
    help="Enter which mode you wish the software to run in; full download of proposal (f), specific date proposal (d) or archive mode (a) for running through Cronjob",
)

parser_args = parser.parse_args()

MODE = parser_args.mode

```

```
if not MODE:
    print(
        "Which mode do you wish to run the software in? (Full download (f) or date download (d). To run in archive mode, enter (q) and run script with argument '-m a'.)"
    )
    MODE = str(input())

if MODE == "q":
    print("Exiting software...")
    sys.exit()
elif MODE != "f" and MODE != "a" and MODE != "d":
    print("The entered mode does not exist. Exiting software...")
    sys.exit()

if __name__ == "__main__":
    new_dates = download.main()

    if (config.DEBUG==1): print("length new_dates:",len(new_dates))

##ML CHARACTERIZE ONLY IF NEW DATES
    if len(new_dates):
        if (config.DEBUG==1): print("ready to run characterize.py")
        os.system(f"python characterize.py -m {MODE}")

##ML ANALYZE ONLY IF NEW DATES
    if len(new_dates):
        if (config.DEBUG==1): print("ready to run analyze.py")
        os.system(f"python analyze.py -m {MODE}")


if len(new_dates):
    for DATE in new_dates:
        PROPOSAL_ID = config.ARCHIVE_PROPOSAL_ID
        proposal_path = f"analysis/{PROPOSAL_ID}"
        targets = os.listdir(proposal_path)
        for target in targets:
            dates = os.listdir(f"{proposal_path}/{target}")
            if DATE in dates:
                email_str = ""
                rootdir = f"analysis/{PROPOSAL_ID}/{target}/{DATE}"
                email_str = f"'TJO report. PROPOSAL: {config.ARCHIVE_PROPOSAL_ID} DATE: {DATE} TARGET: {target}' -a analysis/{config.ARCHIVE_PROPOSAL_ID}/{target}/{DATE}/TJOreduced_{DATE}.log "
                #ML: add night plot now produced in characterize.py:
                if os.path.exists(
                    f"data/{config.ARCHIVE_PROPOSAL_ID}/{DATE}_reddata/night.jpg"
                ):
                    email_str += f"-a data/{config.ARCHIVE_PROPOSAL_ID}/{DATE}_reddata/night.jpg "



                for filter in next(os.walk(rootdir))[1]:
                    if os.path.exists(
                        f"analysis/{config.ARCHIVE_PROPOSAL_ID}/{target}/{DATE}/{filter}/combined_{filter}.jpg"
                    ):
                        email_str += f"-a analysis/{config.ARCHIVE_PROPOSAL_ID}/{target}/{DATE}/{filter}/combined_{filter}.jpg "
#Attach another file:
#                   if os.path.exists(
#                       f"analysis/{config.ARCHIVE_PROPOSAL_ID}/{target}/{DATE}/{filter}/combined_{filter}.jpg"
#                   ):
#                       email_str += f"-a analysis/{config.ARCHIVE_PROPOSAL_ID}/{target}/{DATE}/{filter}/combined_{filter}.jpg "


                email_str += f"{config.EMAIL_ADDRESS}, {config.CC_ADDRESS} "
                if os.path.exists(
                    f"analysis/{config.ARCHIVE_PROPOSAL_ID}/{target}/{DATE}/TJOsummary_{DATE}.log"
                ):
                    email_str += f"< analysis/{config.ARCHIVE_PROPOSAL_ID}/{target}/{DATE}/TJOsummary_{DATE}.log"
                #else? what if .log is not found?
                text_file = open("latest_email.txt", "w")
                text_file.write("mailx -n -s " + email_str)
                text_file.close()
                os.system("mailx -n -s " + email_str)
```

B download.py

```
from codecs import utf_16_encode
from encodings import utf_8
import os
import sys
import argparse
from urllib import response
import numpy as np
import tarfile

import requests
from requests import Response, Session
from bs4 import BeautifulSoup as BS
import config

from urllib3.exceptions import InsecureRequestWarning

parser = argparse.ArgumentParser()

parser.add_argument(
    "-m",
    "--mode",
    help="Enter which mode you wish the software to run in; full download of proposal (f), specific date proposal (d) or archive mode (a) for running through Cronjob",
)

parser_args = parser.parse_args()

requests.packages.urllib3.disable_warnings(category=InsecureRequestWarning)


def login(sess, url, username, password):
    """Login to the server"""
    response = sess.post(
        url,
        data={"user": username, "pass": password},
    )
    return response


def scrape_proposals(sess, url):
    """Scrapes website to look for proposals in order to determine if there is any new proposals and for input checks"""
    response = sess.get(
        url,
    )
    soup = BS(response.text.encode("utf-8"), features="lxml")
    soup_table = soup.find("table", class_="listado2")

    tag_td = np.array(soup_table.find_all("td"), dtype=object)
    shaped_tag_td = np.reshape(tag_td, np.shape(tag_td)[0])

    if url.endswith("pi"):
#           n = 6
        n=8      #ML number of columns in table - 1
    elif url.endswith("collaborator"):
#           n = 7
        n=9     #ML number of columns in table - 1 ?
    proposals = np.array([])
```

```python
    for i in range(1, len(shaped_tag_td) // n):
        i = i * n
        tmp_string = shaped_tag_td[i]
        proposal_id = str(tmp_string)[-8:-5]
        proposals = np.append(proposals, proposal_id)
    return response, proposals


def scrape_filenames(sess, url, proposal_id):
    """Scrapes website to look for filenames for downloadable files and for input checks"""
    response = sess.get(
        url,
        params={"proposal_id": proposal_id},
    )
    soup = BS(response.text, features="lxml")

    tag_a = np.array(soup.find_all("a"), dtype=object)
    shaped_tag_a = np.reshape(tag_a, np.shape(tag_a)[0])

    filenames = np.array([])
    dates = np.array([])
    for i, string in enumerate(shaped_tag_a):
        if "tar.gz" in string:
            filenames = np.append(filenames, string)
            dates = np.append(dates, string[:8])
    return response, filenames, dates


def download_file(sess, url, proposal_id, filename):
    """Download file from URL with params"""
    response = sess.get(
        url,
        params={"proposal_id": proposal_id, "file": filename},
        stream=True,
    )
    return response


def untar(proposal_id, filename):
    """Untars the file and removes the tared file from the directory"""
    tar = tarfile.open(f"data/{proposal_id}/{filename}", mode="r")
    tar.extractall(path=f"data/{proposal_id}/")
    tar.close()
    os.remove(f"data/{proposal_id}/{filename}")
    return


def save_file(response, dir_name, filename):
    """Save a file to a given directory"""
    if len(response.content):
        if not os.path.exists(dir_name):
            os.makedirs(dir_name)

        with open(f"{dir_name}/{filename}", "wb") as f:
            f.write(response.content)


def main():
    """Main function"""
    sess = Session()
    sess.verify = False

    login_response = login(
        sess=sess,
        url=config.LOGIN_URL,
        username=config.USERNAME,
        password=config.PASSWORD,
    )

    _, proposal_id_lst = scrape_proposals(
        sess=sess,
        url=config.PROPOSAL_ID_URL,
    )

    MODE = parser_args.mode

    if not MODE:
        print(
            "Which mode do you wish to run the software in? (Full download (f) or date download (d). To run in archive mode, enter (q) and run script with argument '-m a'.)"
        )
        MODE = str(input())

    if MODE != "f" and MODE != "a" and MODE != "d":
        print("The entered mode does not exist. Exiting software...")
        return

    """Part of pipeline for downloading full datasets of proposals"""

    if MODE == "f":
        print("Available proposals:", proposal_id_lst)
        print("Enter proposal id:")
        PROPOSAL_ID = str(input())
        if PROPOSAL_ID == "q":
            print("Exiting software")
            return

        if (config.DEBUG==1): print("PROPOSAL_ID, proposal_id_lst:",PROPOSAL_ID, proposal_id_lst)

        while PROPOSAL_ID not in proposal_id_lst:
            print("Invalid proposal id, please try again:")
            PROPOSAL_ID = str(input())
            if PROPOSAL_ID == "q":
                print("Exiting software")
                return

        _, filenames_lst, _ = scrape_filenames(
            sess=sess,
            url=config.FILENAME_URL,
            proposal_id=PROPOSAL_ID,
        )
        k = 0
        for FILENAME in filenames_lst:
            if os.path.exists(f"data/{PROPOSAL_ID}/{FILENAME[:-7]}"):
                k += 1
                print(f"{FILENAME} already downloaded.")
            elif os.path.exists(f"data/{PROPOSAL_ID}/{FILENAME}"):
                untar(PROPOSAL_ID, FILENAME)
            else:
                print("Downloading file:", FILENAME)
                download_response = download_file(
                    sess=sess,
                    url=config.DOWNLOAD_BASE_URL,
                    proposal_id=PROPOSAL_ID,
                    filename=FILENAME,
                )
                if len(download_response.content) == 0:
                    print(f"{FILENAME} was empty.")
                else:
                    save_dir = f"data/{PROPOSAL_ID}"
                    save_file(download_response, save_dir, FILENAME)
                    untar(PROPOSAL_ID, FILENAME)
        if k == len(filenames_lst):
            print("No new files found")
            return np.array([])
        else:
            return np.unique(new_dates)

    """Part of pipeline for downloading specific dates of proposals"""

    if MODE == "d":
        print("Available proposals:", proposal_id_lst)
        print("Enter proposal id:")
        PROPOSAL_ID = str(input())

        if PROPOSAL_ID == "q":
            print("Exiting software")
            return

        if (config.DEBUG==1): print("PROPOSAL_ID, proposal_id_lst:",PROPOSAL_ID, proposal_id_lst)

        while PROPOSAL_ID not in proposal_id_lst:
            print("Invalid proposal id, please try again:")
            PROPOSAL_ID = str(input())
            if PROPOSAL_ID == "q":
                print("Exiting software")
                return

        _, filenames_lst, date_lst = scrape_filenames(
            sess=sess,
            url=config.FILENAME_URL,
            proposal_id=PROPOSAL_ID,
        )
        k = 0
        print("Available dates:", np.unique(date_lst))
        print("Enter date (YYYYMMDD):")
        DATE = str(input())

        if DATE == "q":
            print("Exiting software")
            return

        while DATE not in date_lst:
            print("Invalid date, please try again:")
            DATE = str(input())
            if DATE == "q":
                print("Exiting software")
                return

        indices = np.argwhere(DATE == date_lst)
        for i in indices.flatten():
            FILENAME = filenames_lst[i]
            if os.path.exists(f"data/{PROPOSAL_ID}/{FILENAME[:-7]}"):
                k += 1
                print(f"{FILENAME} already downloaded.")
            elif os.path.exists(f"data/{PROPOSAL_ID}/{FILENAME}"):
                untar(PROPOSAL_ID, FILENAME)
            else:
                print("Downloading file:", FILENAME)
                download_response = download_file(
                    sess=sess,
                    url=config.DOWNLOAD_BASE_URL,
                    proposal_id=PROPOSAL_ID,
                    filename=FILENAME,
                )
                if len(download_response.content) == 0:
                    print(f"{FILENAME} was empty.")
                else:
                    save_dir = f"data/{PROPOSAL_ID}"
                    save_file(download_response, save_dir, FILENAME)
                    untar(PROPOSAL_ID, FILENAME)

        if k == len(filenames_lst):
            print("No new files found")
            return np.array([])
        else:
            return np.unique(new_dates)

    """Part of pipeline for archive mode, to be run with Cronjob"""

    if MODE == "a":
        PROPOSAL_ID = str(config.ARCHIVE_PROPOSAL_ID)
        if PROPOSAL_ID == "q":
            print("Exiting software")
            return

        if (config.DEBUG==1): print(”PROPOSAL_ID, proposal_id_lst:”,PROPOSAL_ID, proposal_id_lst)

        if PROPOSAL_ID not in proposal_id_lst:
            print(
                ”Invalid proposal id inserted into config file, please insert a valid id and try again”
            )
            return

        _, filenames_lst, _ = scrape_filenames(
            sess=sess,
            url=config.FILENAME_URL,
            proposal_id=PROPOSAL_ID,
        )
        k = 0
        new_dates = np.array([])
        for FILENAME in filenames_lst:
            if os.path.exists(f”data/{PROPOSAL_ID}/{FILENAME[:-7]}”):
                k += 1
            elif os.path.exists(f”data/{PROPOSAL_ID}/{FILENAME}”):
                untar(PROPOSAL_ID, FILENAME)
            else:
                new_dates = np.append(new_dates, FILENAME[0:8])
                print(”Downloading file:”, FILENAME)
                download_response = download_file(
                    sess=sess,
                    url=config.DOWNLOAD_BASE_URL,
                    proposal_id=PROPOSAL_ID,
                    filename=FILENAME,
                )

                if len(download_response.content) == 0:
                    print(f”{FILENAME} was empty.”)
                else:
                    save_dir = f”data/{PROPOSAL_ID}”
                    save_file(download_response, save_dir, FILENAME)
                    untar(PROPOSAL_ID, FILENAME)

        if k == len(filenames_lst):
            print(”No new files found”)
            return np.array([])
        else:
            return np.unique(new_dates)


if __name__ == ”__main__”:
    main()
```

C characterize.py

```
import os
import shutil
import argparse
import numpy as np
from astropy.table import unique, Table, Column, MaskedColumn, hstack
from astropy.io import fits, ascii
import matplotlib.pyplot as plt
import warnings

warnings.filterwarnings(”ignore”, category=DeprecationWarning)
import config
import glob #added by ML to select files carefully
from astropy.coordinates import SkyCoord  #added by ML to read/convert RA,DEC format
import astropy.units as u  #added by ML to read/convert RA,DEC format
from astropy.coordinates import Angle

#ML May 2023, changes to general_night_log in characterize.py to
```

```
#remove blank spaces from strings that store:
#
#Object (e.g. 'MAXI J1820+070' had blank spaces, i have NOT reprocessed that)
#Filter (new SDSS filters had blank spaces in header keyword)
#Obstype (e.g. 'Sky Flat' had blank spaces)

#ML May 2023, added DEBUG flag in config.py
#if (config.DEBUG==1): MORE OUTPUT PRINTED IN ALL MODES

#ML: most science object reduced images are TJO*imc.fits
#ML: but a few cases have TJO*iml.fits files (failed astrometry??)


#ML: major changes in characterize on March 1st, 2023:

#produce night.jpg plots when catalog files are not available
#(calculating airmass and setting nstars=fwhm=0)

#modified copy call to exclude iml.fits files from being copied to
#analysis folder

#grab information from reddata.txt only for imc.fits files (again,
#exclude iml.fits files)


#ML: added more printed output by default to mode 'd' (#ML debug):
#mode d: more output


parser = argparse.ArgumentParser()

parser.add_argument(
    ”−m”,
    ”−−mode”,
    help=”Enter which mode you wish the software to run in; full download of
        proposal (f), specific date proposal (d) or archive mode (a) for running
        through Cronjob”,
)

parser_args = parser.parse_args()

def plot_night(table,outfile,date,targets):

    resdpi = 200 #resolution of jpeg plot
    fig = plt.figure(figsize=(6,9))

    gs = fig.add_gridspec(5, hspace=0)
    axs = gs.subplots(sharex=True, sharey=False)
    fig.suptitle(f”Plot of TJO DATE: {date}.\nTargets: {targets}.\n NSTARS/FWHM:
        within {xsize:.1f}x{ysize:.1f}arcmin central FoV”)
#if you want to add more info in title: \n(0 when not available: no catalog?)

    style = dict(size=10, color='gray') #for labels

    axs[0].plot(table[”jd”], table[”exptime”], 'v')
    axs[0].set(ylabel='EXPTIME(s)')
#    axs[0].text(2459965.40641,120, 'TEST−U')
    for i in range(len(table[”jd”])):
        axs[0].text(table[”jd”][i],table[”exptime”][i], table[”filter”][i],ha='
            right',**style)

#     print(table.info)

#     elevatio = Angle(str(table[”elevatio”]+' degrees'))
#     print(elevatio)
    altitude = np.zeros(len(table[”elevatio”]))
    jday = np.zeros(len(table[”elevatio”]))

    for i in range(len(table[”elevatio”])):
        temp = table[”elevatio”][i].split(”:”)
        elev = float(temp[0]) + int(temp[1]) / 60.0 + int(temp[2]) / 3600.0
        altitude[i] = elev #added by ML: save in degrees
        jday[i] = table[”jd”][i]

    axs[1].plot(jday,altitude , 'o')
    axs[1].set(ylabel='ALTITUDE(deg)')

    axs[2].plot(table[”jd”], table[”airmass”], '+')
    axs[2].set(ylabel='AIRMASS')

    axs[3].plot(table[”jd”], table[”FWHM”], 's')
    axs[3].set(ylabel='FWHM(\”)')

    axs[4].plot(table[”jd”], table[”NSTARS”], '+')
    axs[4].set(ylabel='NSTARS')

    plt.xlabel(”Time (days +JD)”)

# Hide x labels and tick labels for all but bottom plot.
    for ax in axs:
        ax.label_outer()

    #save to file with the desired resolution
    plt.savefig(outfile, dpi=resdpi)
    plt.close(fig)


def slicer_vectorized(a, start, end):
    b = a.view((str, 1)).reshape(len(a), −1)[:, start:end]
    return np.fromstring(b.tostring(), dtype=(str, end − start))


def general_night_log(path):
    ”””
    Input: A path of where the night logs are to be taken from. Can either be rawfiles or reduced files.

    Creates general night logs from the headers of the FITS files.

    Output: a Table for further use with the AstroPy library with all info from headers in each column.

    ”””

   # ML. changed these lines to make sure that we only select TJO∗.fits files
   #ML: you cannot use only imc files since night logs are also created from raw data

#     file_lst = np.array(os.listdir(path))
    file_names=[]
    if (config.DEBUG==1): print(path,'/TJO∗.fits')

    #This is needed to strip off the path fron the filenames
    for file in glob.glob(str(path+'/TJO∗.fits')):
        file_names.append(os.path.relpath(file,path))

    if MODE != ”a” and MODE != ”f”: print(”file_names:”,file_names)

    file_lst = np.array(file_names)
    size = np.shape(file_lst)
    if MODE != ”a” and MODE != ”f”: print(”size,file_lst:”,size,file_lst)

    log_info = {
        ”filename”: Column(
            np.zeros(size), name=”filename”, dtype=”str”, format=”%s”, unit=”−”
        ),
        ”obstype”: Column(
            np.zeros(size), name=”obstype”, dtype=”str”, format=”%s”, unit=”−”
        ),
        ”object”: Column(
            np.zeros(size), name=”object”, dtype=”str”, format=”%s”, unit=”−”
        ),
        ”filter”: Column(
            np.zeros(size), name=”filter”, dtype=”str”, format=”%s”, unit=”−”
        ),
        ”date−obs”: Column(
            np.zeros(size), name="date-obs", dtype="str", format="", unit="-"
        ),
        "time-obs": Column(
            np.zeros(size), name="time-obs", dtype="str", format="", unit="-"
        ),
        "jd": Column(np.zeros(size), name="jd", dtype="float", format="%.6f", unit="-"),
        "elevatio": Column(
            np.zeros(size), name="elevatio", dtype="str", format="", unit="-"
        ),
        "ra": Column(np.zeros(size), name="ra", dtype="str", format="", unit="-"),
        "dec": Column(np.zeros(size), name="dec", dtype="str", format="", unit="-"),
        "instrume": Column(
            np.zeros(size), name="instrume", dtype="str", format="%s", unit=""
        ),
        "naxis1": Column(
            np.zeros(size), name="naxis1", dtype="int", format="%i", unit="pix"
        ),
        "naxis2": Column(
            np.zeros(size), name="naxis2", dtype="int", format="%i", unit="pix"
        ),
        "camtemp": Column(
            np.zeros(size), name="camtemp", dtype="float", format="%.1f", unit="C"
        ),
        "exptime": Column(
            np.zeros(size), name="exptime", dtype="float", format="%.1f", unit="s"
        ),
        "defocus": Column(
            np.zeros(size), name="defocus", dtype="float", format="%.1f", unit="-"
        ),
        "focuspos": Column(
            np.zeros(size), name="focuspos", dtype="float", format="%.2f", unit="-"
        ),
        "domestat": Column(
            np.zeros(size), name="domestat", dtype="str", format="%s", unit="-"
        ),
    }

    table = Table(log_info)
    for i, file in enumerate(file_lst):
        hdu_lst = fits.open(f"{path}/{file}")

        for cols in table.colnames:
            if cols == "filename":
                table[cols][i] = file
            elif cols == "object" or cols == "filter" or cols == "obstype":
                table[cols][i] = hdu_lst[0].header[cols].replace(" ", "")  #ML test remove blank spaces from string
            else:
                table[cols][i] = hdu_lst[0].header[cols]

    return table


def create_summary(table, path, PROPOSAL_ID, DATE):
    array = {
        "object": Column([], name="object", dtype="str", format="%s", unit="-"),
        "Nr.Images": Column([], name="Nr.Images", dtype="int", format="", unit="-"),
        "T-range": Column([], name="Trange", dtype="str", format="%s", unit="JD"),
        "Exp(min-max)": Column(
            [], name="Exp(min-max)", dtype="str", format="%s", unit="s"
        ),
        "Filters": Column([], name="Filters", dtype="str", format="%s", unit="-"),
        "AM-range": Column([], name="AM-range", dtype="str", format="%s", unit="-"),
        "FWHM-range": Column(
#ML:            [], name="FWHM-range", dtype="str", format="%s", unit="pixel"
            [], name="FWHM-range", dtype="str", format="%s", unit="arcsec"
        ),
        "Nstars": Column([], name="Nstars", dtype="str", format="%s", unit="-"),
    }

    obj = str(np.unique(np.array(table[”object”]))[0])
    nr_images = len(np.array(table[”object”]))
    t_range = str(
        str(np.round(np.min(np.array(table[”jd”])), 5))
        + ”–”
        + str(np.round(np.max(np.array(table[”jd”])), 5))
    )
    exposure_times = str(
        str(np.round(np.min(np.array(table[”exptime”])), 1))
        + ”–”
        + str(np.round(np.max(np.array(table[”exptime”])), 1))
    )
    filter_lst = np.unique(np.array(table[”filter”]))
    filters = ””
    for filter in filter_lst:
        filters += str(filter)

    cat_path = f”data/{PROPOSAL_ID}/{DATE}_catalog”
    if os.path.exists(cat_path):
        airmass_range = str(
            str(np.round(np.min(np.array(table[”airmass”])), 2))
            + ”–”
            + str(np.round(np.max(np.array(table[”airmass”])), 2))
        )
        fwhm_range = str(
            str(np.round(np.min(np.array(table[”FWHM”])), 2))
            + ”–”
            + str(np.round(np.max(np.array(table[”FWHM”])), 2))
        )
        nstars = str(
            str(np.min(np.array(table[”NSTARS”])))
            + ”–”
            + str(np.max(np.array(table[”NSTARS”])))
        )
    else:
        airmass = np.zeros(len(table[”elevatio”]))
        for i in range(len(table[”elevatio”])):
            temp = table[”elevatio”][i].split(”:”)
            elev = float(temp[0]) + int(temp[1]) / 60.0 + int(temp[2]) / 3600.0
            airmass[i] = 1 / np.sin(elev * np.pi / 180)
        airmass_range = (
            str(np.round(np.min(airmass), 2)) + ”–” + str(np.round(np.max(airmass),
                2))
        )
        fwhm_range = ”–”
        nstars = ”–”

    temp_table = Table(array)
    temp_table.add_row(
        np.array(
            [
                obj,
                nr_images,
                t_range,
                exposure_times,
                filters,
                airmass_range,
                fwhm_range,
                nstars,
            ],
        )
    )
    logsum = open(path, ”w+”)
    logsum.write(
        str(
            temp_table[
                ”object”,
                ”Nr.Images”,
                ”T–range”,
                ”Exp(min–max)”,
                ”Filters”,
                ”AM–range”,
                "FWHM-range",
                "Nstars",
            ]
        )
    )
    logsum.close()
    # ascii.write(temp_table, path, format="fixed_width_two_line")


def characterize(PROPOSAL_ID, DATE):
    # CCD trim/crop region in pixels:
    xmin = 360
    xmax = 3580
    ymin = 380
    ymax = 3610

    # Pixel scale in arcsec/pixel:
    pscale = 0.4

    # Standard image size:
    naxis1 = 4096
    naxis2 = 4108

    # Trim/crop size in arcmin:
    global xsize,ysize
    xsize = (xmax - xmin) * pscale / 60.0
    ysize = (ymax - ymin) * pscale / 60.0
    red_path = f"data/{PROPOSAL_ID}/" + str(DATE) + "_reddata"

    #Create corresponding night sub-directory in the night_logs/ directory:
    if not os.path.exists("night_logs"):
        os.makedirs("night_logs")
    if not os.path.exists(f"night_logs/{PROPOSAL_ID}"):
        os.makedirs(f"night_logs/{PROPOSAL_ID}")

    #If the night log for this night is not there, create it:
    if not os.path.exists(f"night_logs/{PROPOSAL_ID}/TJOnight_{DATE}.log"):
        #Use preferentially raw data directory:
        raw_path = f"data/{PROPOSAL_ID}/" + str(DATE) + "_rawdata"
        #If _rawdata is not there, use _reddata instead (**):
        if not os.path.exists(raw_path):
            raw_path = red_path

        if (config.DEBUG==1): print("ready to create night log from raw data at ", raw_path)

        #This creates the night log AND STORES TABLE for further processing
        raw_table = general_night_log(raw_path)
        raw_table.sort(["obstype", "jd"])

        if (config.DEBUG==1): print(raw_table,DATE) #ML debug

        #This dumps the contents of raw_table into the night log:
        #ML WARNING: THIS IS CALLED raw_table BUT IT COULD CONTAIN A TABLE BUILT FROM _reddata (see ** above)
        ascii.write(
            raw_table,
            f"night_logs/{PROPOSAL_ID}/TJOnight_{DATE}.log",
            format="fixed_width_two_line",
        )
    #If the night log for this night IS there, read it:
    else:
        raw_table = ascii.read(f"night_logs/{PROPOSAL_ID}/TJOnight_{DATE}.log")

    #Set analysis path to PROPOSAL_ID:
    analysis_path = f"analysis/{PROPOSAL_ID}"
    #Create analysis path to PROPOSAL_ID if it was not there:
    if not os.path.exists(analysis_path):
        if (config.DEBUG==1): print("analysis path does not exist, creating:", analysis_path)
        os.makedirs(analysis_path)

    #ML: create another night log, from _reddata. WHY?
    if (config.DEBUG==1): print("ready to create night log from red data at ", red_path)
    #ML: keep table in temp_red_table:
    temp_red_table = general_night_log(red_path)
    #ML: remove non-Science (calibration) entries:
    rm_lst = np.argwhere(temp_red_table["obstype"] != "Science")
    temp_red_table.remove_rows(rm_lst)
    #Sort according to filename:
    temp_red_table.sort("filename")

    #Build path to YYYYMMDD_reddata.txt:
    cat_path = f"data/{PROPOSAL_ID}/{DATE}_catalog"
    redtxt_path = (
        f"data/{PROPOSAL_ID}/" + str(DATE) + "_catalog/" + str(DATE) + "_reddata.txt"
    )

#ML: now defining arrays and tables before we branch according to cat/red presence.
#ML: goal is to have some of the same table elements available for plotting (FWHM, airmass, nstars) even if empty

#ML: this is the array/table that stores the info we grab from reddata.txt:
    array = {
        "rawfile": Column(
            [], name="rawfile", dtype="str", format="%s", unit="-"
        ),
        "biasfile": Column(
            [], name="biasfile", dtype="str", format="%s", unit="-"
        ),
        "darkfile": Column(
            [], name="darkfile", dtype="str", format="%s", unit="-"
        ),
        "flatfile": Column(
            [], name="flatfile", dtype="str", format="%s", unit="-"
        ),
        "redfile": Column(
            [], name="redfile", dtype="str", format="%s", unit="-"
        ),
        "catfile": Column(
            [], name="catfile", dtype="str", format="%s", unit="-"
        ),
        "nstars0": Column(
            [], name="nstars0", dtype="int", format="%i", unit="-"
        ),
        "fwhm0": Column(
            [], name="fwhm0", dtype="float", format="%.5g", unit="arcsec"
        ),
    }
    extra_red_info = Table(array)


#ML: this is the array/table that stores the info we grab from .dat files in catalog folder
    last_array = {
        "FWHM": Column(
            [], name="FWHM", dtype="float", format="%.2g", unit="arcsec"
        ),
        "NSTARS": Column([], name="NSTARS", dtype="int", format="%i", unit="-"),
        "airmass": Column(
            [], name="airmass", dtype="float", format="%.3g", unit="-"
        ),
    }
    last_table = Table(last_array)

#ML from the Sextractor manual:
#Extraction flags: FLAGS
#FLAGS contains 8 flag bits with basic warnings about the source extraction process, in order of increasing concern.
#SExtractor Documentation, Release 2.24.2
#Table 1.3: FLAGS description
#ValueMeaning
#1 aperture photometry is likely to be biased by neighboring sources or by more than 10% of bad pixels in any aperture
```

```
#2 the object has been deblended
#4 at least one object pixel is saturated
#8 the isophotal footprint of the detected object is truncated (too close to an image boundary)
#16 at least one photometric aperture is incomplete or corrupted (hitting buffer or memory limits)
#32 the isophotal footprint is incomplete or corrupted (hitting buffer or memory limits)
#64 a memory overflow occurred during deblending
#128 a memory overflow occurred during extraction


    if os.path.exists(cat_path):
        if os.path.exists(redtxt_path):
#array and extra_red_info were defined here. now above
            with open(redtxt_path) as f:
                i = 0
                for line in f:
                    txt_arr = np.array(line.split())
                    if len(txt_arr) == 8 and txt_arr[4].endswith("imc.fits"):
                        if (config.DEBUG==1): print(DATE, txt_arr)  #ML debug
                        #ML: In some exceptional cases there is no good photometry and num_stars="--", fwhm="nan"
                        #That gave invalid format error and program crashes
                        #Next lines work around that by setting them to 0
                        if txt_arr[6] == "--":
                            txt_arr[6] = 0
                        if txt_arr[7] == "nan":
                            txt_arr[7] = 0.0

                        extra_red_info.add_row(txt_arr)
            extra_red_info.sort("redfile")
        else:
            print(f"No reddata.txt file found for the date {DATE}")

#       cat_lst = os.listdir(cat_path) #old version
        if (config.DEBUG==1): print(cat_path) #ML debug

#ML I have introduced here another selective list so that we only
#list the files we are looking for: catalog TJO*imc_trl.dat files

        cat_lst = glob.glob(str(cat_path+'/TJO*imc_trl.dat'))
        if (config.DEBUG==1): print(cat_lst) #ML debug

        cat_lst.sort()

        if (config.DEBUG==1): print("len cat_lst",len(cat_lst)) #ML debug
        if (config.DEBUG==1): print("len temp red table",len(temp_red_table))#ML debug
        if len(cat_lst) > 0:
#last_array and last_table were defined here. now above
#           k = 0
            for i in range(len(cat_lst)):
#               if f"{cat_lst[i]}".endswith(".txt"):
#                   k += 1
                if f"{cat_lst[i]}".endswith("imc_trl.dat"):
                    loadcat = ascii.read(f"{cat_lst[i]}")
                    mask = (
                        (loadcat["X_IMAGE"] > xmin)
                        & (loadcat["X_IMAGE"] < xmax)
                        & (loadcat["Y_IMAGE"] > ymin)
                        & (loadcat["Y_IMAGE"] < ymax)
                    )
                    maskgood = (
                        (loadcat["X_IMAGE"] > xmin)
                        & (loadcat["X_IMAGE"] < xmax)
                        & (loadcat["Y_IMAGE"] > ymin)
                        & (loadcat["Y_IMAGE"] < ymax)
                        & (loadcat["FLAGS"] == 0)          #ML beta!: adding this line to use only good photometry. See meaning of flags above
                    )
                    trimmed_cat = loadcat[mask]             #ML: stores stars in central part of CCD (avoid vignetting)
                    trimmed_cat_good = loadcat[maskgood]    #ML: stores stars in central part of CCD with good photometry (FLAGS=0)

                    ngood = len(trimmed_cat_good)           #ML: if this is 0 we might run into problems: runtime warning (mean of empty slice)

#                    print("i:",i) #ML debug
#                    temp = temp_red_table["elevatio"][i - k].split(":")
                    temp = temp_red_table["elevatio"][i].split(":")
                    elev = float(temp[0]) + int(temp[1]) / 60.0 + int(temp[2]) / 3600.0

                    nclean = len(trimmed_cat)              #ML: still counting all stars in central part, no matter how good photometry is

                    if ngood==0:
                        print("NO GOOD PHOTOMETRY IN CENTRAL PART OF CCD!",cat_lst[i])
                        FWHM = 0.0
                    else:
                        FWHM = 3600 * np.mean(trimmed_cat_good["FWHM_WORLD"])      #ML: now averaging only over good photometry stars to get FWHM

                    if MODE != "a" and MODE != "f": print("in CAT: clean, good:",nclean,ngood) #ML debug

                    airmass = 1 / np.sin(elev * np.pi / 180)

                    last_table.add_row([FWHM, nclean, airmass])
            if (config.DEBUG==1): print("lengths",len(temp_red_table),len(extra_red_info),len(last_table)) #ML debug
            red_table = hstack([temp_red_table, extra_red_info, last_table])
    else:
        print(f"No catalog folder found for the date {DATE}")

#ML: changed next lines so that we always get airmass and FWHM, nstars, even if empty=0 when no catalog is available

#        red_table = temp_red_table
#        print(temp_red_table, extra_red_info, last_table)
        ntab=len(temp_red_table)
        if (config.DEBUG==1): print("ntab:",ntab) #ML debug


        for i in range(ntab):

            #Catalog photometry files not available here, we will set these to 0
            FWHM=0.0
            nclean=0

            #Since airmass was not read from txt file, we calculate it from altitude ('elevatio'):
#            print(temp_red_table["elevatio"][i])
            temp = temp_red_table["elevatio"][i].split(":")
            elev = float(temp[0]) + int(temp[1]) / 60.0 + int(temp[2]) / 3600.0
            airmass = 1 / np.sin(elev * np.pi / 180)

            last_table.add_row([FWHM, nclean, airmass])

        red_table = hstack([temp_red_table, last_table])

        if (config.DEBUG==1): print("red_table:\n",red_table)


    target_indices = np.argwhere(red_table["obstype"] == "Science")
    targets = np.unique(np.array(red_table["object"][target_indices]))
    if (config.DEBUG==1): print("targets:",targets)

    #ML: This will produce a plot of AIRMASS, ALTITUDE, FWHM, etc
    #Exclude calibrations, bias, etc:
    science_indices = np.argwhere(red_table["obstype"] == "Science").flatten()
    science_red_table = red_table[science_indices]
    #Build path where plot will be saved:
    plotpath = f"data/{PROPOSAL_ID}/{DATE}_reddata/night.jpg"
    #Produce plot:
    plot_night(science_red_table,plotpath,DATE,targets)


    for target in targets:
        target_path = f"{analysis_path}/{target}"
        if not os.path.exists(target_path):
            os.makedirs(target_path)

    if len(targets) > 1:
        for target in targets:
            target_indices = np.argwhere(red_table["object"] == str(target)).flatten()
            new_red_table = red_table[target_indices]
            analysis_date_path = f"analysis/{PROPOSAL_ID}/{target}/{DATE}"
            if not os.path.exists(analysis_date_path):
                os.makedirs(analysis_date_path)
            if not os.path.exists(f"{analysis_date_path}/TJOreduced_{DATE}.log"):
                ascii.write(
                    new_red_table,
                    f"{analysis_date_path}/TJOreduced_{DATE}.log",
                    format="fixed_width_two_line",
                )

            filenames = np.array(new_red_table["filename"]).flatten()
            filters = np.array(new_red_table["filter"]).flatten()
            for i, filter in enumerate(filters):
                if not os.path.exists(f"{analysis_date_path}/{filter}"):
                    os.makedirs(f"{analysis_date_path}/{filter}")
                #ML: added endswith to exclude iml.fits files from copy
                if not os.path.exists(f"{analysis_date_path}/{filter}/{filenames[i]}"):
                    if filenames[i].endswith("imc.fits"):
                        shutil.copy2(
                            f"data/{PROPOSAL_ID}/{DATE}_reddata/{filenames[i]}",
                            f"{analysis_date_path}/{filter}/{filenames[i]}",
                        )
            summary_path = f"{analysis_date_path}/TJOsummary_{DATE}.log"
            if not os.path.exists(summary_path):
                create_summary(new_red_table, summary_path, PROPOSAL_ID, DATE)

    else:
        target = targets[0]
        analysis_date_path = f"analysis/{PROPOSAL_ID}/{target}/{DATE}"
        if not os.path.exists(analysis_date_path):
            os.makedirs(analysis_date_path)
        if not os.path.exists(f"{analysis_date_path}/TJOreduced_{DATE}.log"):
            ascii.write(
                red_table,
                f"{analysis_date_path}/TJOreduced_{DATE}.log",
                format="fixed_width_two_line",
            )

        filenames = np.array(red_table["filename"])
        filters = np.array(red_table["filter"])
        for i, filter in enumerate(filters):
            if not os.path.exists(f"{analysis_date_path}/{filter}"):
                os.makedirs(f"{analysis_date_path}/{filter}")
            #ML: added endswith to exclude iml.fits files from copy
            if not os.path.exists(f"{analysis_date_path}/{filter}/{filenames[i]}"):
                if filenames[i].endswith("imc.fits"):
                    shutil.copy2(
                        f"data/{PROPOSAL_ID}/{DATE}_reddata/{filenames[i]}",
                        f"{analysis_date_path}/{filter}/{filenames[i]}",
                    )
        summary_path = f"{analysis_date_path}/TJOsummary_{DATE}.log"
        if not os.path.exists(summary_path):
            create_summary(red_table, summary_path, PROPOSAL_ID, DATE)
```

```python


def main():

    # CCD trim/crop region in pixels:
    xmin = 360
    xmax = 3580
    ymin = 380
    ymax = 3610

    # Pixel scale in arcsec/pixel:
    pscale = 0.4

    # Standard image size:
    naxis1 = 4096
    naxis2 = 4108

    # Trim/crop size in arcmin:
    xsize = (xmax - xmin) * pscale / 60.0
    ysize = (ymax - ymin) * pscale / 60.0

    #ML: Make it global to filter output in characterize() depending on MODE
    #ML: Now most output is printed according to config.DEBUG
    global MODE
    MODE = parser_args.mode

    if not MODE:
        print(
            "Which mode do you wish to run the software in? (Full download (f) or date download (d). To run in archive mode, enter (q) and run script with argument '-m a'.)"
        )
        MODE = str(input())

    if MODE == "q":
        print("Exiting software...")
        return
    elif MODE != "f" and MODE != "a" and MODE != "d":
        print("The entered mode does not exist. Exiting software...")
        return

    # Creates an array with all the available proposals downloaded using the TJO_download script
    proposal_id_lst = np.array(os.listdir("data"))

    """Part of pipeline for characterization of full proposal dataset"""

    if MODE == "f":
        print("Available proposals:", proposal_id_lst)
        print("Enter proposal id:")
        PROPOSAL_ID = str(input())
        if PROPOSAL_ID == "q":
            print("Exiting software")
            return
        while PROPOSAL_ID not in proposal_id_lst:
            print("Invalid proposal id, please try again:")
            PROPOSAL_ID = str(input())
            if PROPOSAL_ID == "q":
                print("Exiting software")
                return

#ML: CHANGED these lines to exclude extraneous files/folders inside data/PROPOSAL_ID
#WARNING: This will break if PROPOSAL_ID does not have 3 characters
#          folder_lst = np.array(os.listdir(f"data/{PROPOSAL_ID}"))
        folder_lst = np.array(glob.glob(str('data/'+PROPOSAL_ID+'/????????_*')))
#          date_lst = np.unique(slicer_vectorized(folder_lst, 0, 8))
        date_lst = np.unique(slicer_vectorized(folder_lst, 9, 17))

        for DATE in date_lst:
            characterize(PROPOSAL_ID, DATE)

    """Part of pipeline for characterization of specific date of proposal"""

    if MODE == "d":
        print("Available proposals:", proposal_id_lst)
        print("Enter proposal id:")
        PROPOSAL_ID = str(input())
        if PROPOSAL_ID == "q":
            print("Exiting software")
            return
        while PROPOSAL_ID not in proposal_id_lst:
            print("Invalid proposal id, please try again:")
            PROPOSAL_ID = str(input())
            if PROPOSAL_ID == "q":
                print("Exiting software")
                return

        # Creates an array containing all folders in specific proposal

#ML: CHANGED these lines to exclude extraneous files/folders inside data/PROPOSAL_ID
#WARNING: This will break if PROPOSAL_ID does not have 3 characters
#        folder_lst = np.array(os.listdir(f"data/{PROPOSAL_ID}"))
        folder_lst = np.array(glob.glob(str('data/'+PROPOSAL_ID+'/????????_*')))
#        date_lst = np.unique(slicer_vectorized(folder_lst, 0, 8))
        date_lst = np.unique(slicer_vectorized(folder_lst, 9, 17))



        print("Available dates:", date_lst)
        print("Enter date (YYYYMMDD):")
        DATE = str(input())

        if DATE == "q":
            print("Exiting software")
            return

        while DATE not in date_lst:
            print("Invalid date, please try again:")
            DATE = str(input())
            if DATE == "q":
                print("Exiting software")
                return

        characterize(PROPOSAL_ID, DATE)

    if MODE == "a":
        PROPOSAL_ID = config.ARCHIVE_PROPOSAL_ID
        if PROPOSAL_ID not in proposal_id_lst:
            print("Invalid proposal id in config file. Exiting software...")
            return
        folder_lst = np.array(os.listdir(f"data/{PROPOSAL_ID}"))
        """DATE = config.ARCHIVE_DATE
        if len(DATE):
            characterize(PROPOSAL_ID, DATE)
        else:"""

#ML: CHANGED these lines to exclude extraneous files/folders inside data/PROPOSAL_ID
#WARNING: This will break if PROPOSAL_ID does not have 3 characters
#        folder_lst = np.array(os.listdir(f"data/{PROPOSAL_ID}"))
        folder_lst = np.array(glob.glob(str('data/'+PROPOSAL_ID+'/????????_*')))
#        date_lst = np.unique(slicer_vectorized(folder_lst, 0, 8))
        date_lst = np.unique(slicer_vectorized(folder_lst, 9, 17))

        for DATE in date_lst:
            characterize(PROPOSAL_ID, DATE)


if __name__ == "__main__":
    main()
```

D analyze.py

```
# from distutils.command.config import config
import os
import shutil
import pandas as pd
import argparse
import numpy as np
from astropy.table import unique, Table, Column, MaskedColumn, hstack
from astropy.io import fits, ascii
from ccdproc import ImageFileCollection
from astropy.nddata import CCDData
import ccdproc
import matplotlib.pyplot as plt
from ccdproc import Combiner, wcs_project
import config

from ccdproc.utils.sample_directory import sample_directory_with_files

from astropy.visualization import ZScaleInterval  #added by ML for image plotting purposes
import glob #added by ML to select files carefully
from astropy.wcs import WCS #added by ML to plot images in WCS
from matplotlib.patches import Circle, Rectangle #added by ML to add markers on images
from astropy.coordinates import SkyCoord  #added by ML to read/convert RA,DEC format
import astropy.units as u  #added by ML to read/convert RA,DEC format

parser = argparse.ArgumentParser()

#ML May 2023, changes to analyze.py to remove blank spaces from
#strings that store (here for plotting purposes only):
#
#Object (e.g. 'MAXI J1820+070' had blank spaces, i have NOT reprocessed that)
#Filter (new SDSS filters had blank spaces in header keyword)

#ML May 2023, added DEBUG flag in config.py
#if (config.DEBUG==1): MORE OUTPUT PRINTED IN ALL MODES

#ML: most science object reduced images are TJO*imc.fits
#ML: but a few cases have TJO*iml.fits files (failed astrometry??)

#reprojection fails if no astrometry is present (like in iml.fits files)

#I have encountered problems when iml.fits files are present in the
#analysis/ folders, but now this should be avoided since
#characterize.py does not copy those iml.fits files over

#ML: major changes in analyze on March 1st, 2023:

#added a limit nmax=100, so that dates/filters with MANY images are
#not reprojected and combined. otherwise we may crash due to memory
#issues (and becomes very slow in any case)


parser.add_argument(
    ”–m”,
    ”—mode”,
    help=”Enter which mode you wish the software to run in; full download of proposal (f), specific date proposal (d) or archive mode (a) for running through Cronjob”,
)
parser_args = parser.parse_args()

def plot_fits(fitsfile, jpgfile, resdpi, raobj, decobj,radius,title):
    #This function plots a fits image with a circle on top
    #
    #INPUTS:
    #fitsfile: name of fits file. image is assumed at extension [0]
    #resdpi: DPI resolution of the jpeg created
    #raobj: R.A.(J2000) of center of circle, in degrees
    #decobj: DEC(J2000) of center of circle, in degrees
    #radius: radius of circle, in degrees
    #title: title string on top of plot
    #
    #OUTPUT:
    #jpgfile: name of jpeg file where the image will be saved

    #read first extension and its WCS info>
    hdu = fits.open(fitsfile)[0]
    wcs = WCS(hdu.header)

    #read image data and setup z scale
    data = fits.getdata(fitsfile)
    zscale = ZScaleInterval()

    #plot in WCS coords, gray scale and zscale
    fig=plt.subplot(projection=wcs)
    plt.imshow(zscale(data), cmap="gray")

    #This will add a circle at the target/s RA,DEC. radius in degrees.
    circle1=plt.Circle((raobj,decobj),radius, color='r',fill=False, transform=fig.get_transform('fk5'))
    plt.gca().add_patch(circle1)

    #Tune
    plt.title(title,fontsize=10)
    plt.xlabel("R.A. (J2000)")
    plt.ylabel("DEC (J2000)")

    #save to file with the desired resolution
    plt.savefig(jpgfile, dpi=resdpi)

    return 1

def main():

    # CCD trim/crop region in pixels:
    xmin = 360
    xmax = 3580
    ymin = 380
    ymax = 3610

    # Pixel scale in arcsec/pixel:
    pscale = 0.4

    # Standard image size:
    naxis1 = 4096
    naxis2 = 4108

    # Trim/crop size in arcmin:
    xsize = (xmax - xmin) * pscale / 60.0
    ysize = (ymax - ymin) * pscale / 60.0

    #ML: for plotting purposes:
    resdpi = 300        #resolution of jpeg (combined) plot
    radcirc = 0.004     #radius of circle around target (degrees)

    #ML: maximum nr of images to reproject and combine:
    nmax = 100


    MODE = parser_args.mode

    if not MODE:
        print(
            "Which mode do you wish to run the software in? (Full download (f) or date download (d). To run in archive mode, enter (q) and run script with argument '-m a'.)"
        )
        MODE = str(input())

    if MODE == "q":
        print("Exiting software...")
        return
    elif MODE != "f" and MODE != "a" and MODE != "d":
        print("The entered mode does not exist. Exiting software...")
        return

    # Creates an array with all the available proposals downloaded using the TJO_download script
    proposal_id_lst = np.array(os.listdir("analysis"))

    if MODE == "f":
        print("Available proposals:", proposal_id_lst)
        print("Enter proposal id:")
        PROPOSAL_ID = str(input())
        if PROPOSAL_ID == "q":
            print("Exiting software")
            return
        while PROPOSAL_ID not in proposal_id_lst:
            print("Invalid proposal id, please try again:")
            PROPOSAL_ID = str(input())
            if PROPOSAL_ID == "q":
                print("Exiting software")
                return

        targets = np.array(os.listdir(f"analysis/{PROPOSAL_ID}"))
        date_df = pd.DataFrame()
        for i, target in enumerate(targets):
            temp_df = pd.DataFrame(
                {target: np.array(os.listdir(f"analysis/{PROPOSAL_ID}/{target}"))}
            )
            date_df = pd.concat([date_df, temp_df], axis=1)

        flat_dates = np.ndarray.flatten(np.array(date_df))
        flat_dates = flat_dates[~pd.isnull(flat_dates)]
        flat_dates.sort()

#        print("flat dates:",flat_dates) #ML debug
#        print("date_df:",date_df) #ML debug

        for DATE in flat_dates:
            for j, target in enumerate(targets):
                dates_for_target = date_df[target]
                dates_for_target = np.array(
                    dates_for_target[~pd.isnull(dates_for_target)]
                )
                if DATE in dates_for_target:
                    rootdir = f"analysis/{PROPOSAL_ID}/{target}/{DATE}"
#                    print("rootdir:",rootdir) # ML debug
                    for item in os.listdir(rootdir):
#                        print("item:",item) # ML debug
                        if os.path.isdir(f"{rootdir}/{item}"):
#                           files_in_dir = np.array(os.listdir(f"{rootdir}/{item}"))
#                            file_locs = np.core.defchararray.add(
#                                f"{rootdir}/{item}/", files_in_dir
#                            )
# ML. changed these lines to make sure that we only select TJO*imc.fits files
                            file_locs = np.array(glob.glob(str(rootdir+'/'+item+'/TJO*imc.fits')))
                            reprojected = []
                            totexp=0.  #ML cumulative exposure
                            nima=0     #image counter
                            nimatot=len(file_locs) # ML: count total nr. oof images in this data/filter. if more than nmax, we will skip reprojection
                            print(DATE,item,"we have in total nr images:",nimatot) #ML debug
                            for i, file in enumerate(file_locs):
#                                print("file1:",file) #ML debug
#                                print("DATE1",DATE) #ML debug
                                hdu_lst = fits.open(file)
                                #ML: Read and save sky coordinates of target:
                                RA=hdu_lst[0].header["OBJRA"]    # Target RA in hh:mm:ss (J2000)
                                DEC=hdu_lst[0].header["OBJDEC"] # Target DEC in dd:am:as (J2000)
                                ltarget=SkyCoord(RA,DEC, unit=(u.hourangle,u.deg), frame='icrs')

                                #Read more info from header:
                                OBJECT=hdu_lst[0].header["OBJECT"].replace(" ", "") # object name
                                EXPTIME=hdu_lst[0].header["EXPTIME"] # exposure time (seconds)
                                FILTER=hdu_lst[0].header["FILTER"].replace(" ", "") # filter
#                                DATE=hdu_lst[0].header["DATE-OBS"] # date (of last exposure in sequence)
                                #count images and total exposure:
                                totexp = totexp + EXPTIME
                                nima = nima + 1

#                                print(RA,DEC)
#                                print(ltarget.ra,ltarget.dec)

                                if (
                                    hdu_lst[0].header["naxis1"] == naxis1
                                    and hdu_lst[0].header["naxis2"] == naxis2
                                ):
                                    print(f"Cropping and analyzing {DATE}...")
                                    image = CCDData.read(file)
                                    trimmed = ccdproc.trim_image(
                                        image[xmin:xmax, ymin:ymax]
                                    )
                                    saveto = file
                                    trimmed.write(saveto, overwrite=True, format="fits")
                                    if not os.path.exists(
                                        f"{rootdir}/{item}/combined_{item}.fits"
                                    ):
                                        if i == 0:
                                            image0 = trimmed.wcs
                                        proj_image = wcs_project(trimmed, image0)
                                        reprojected.append(proj_image)
                                else:
                                    if nimatot>nmax:
#                                        print("More than",nmax," images, will not reproject",nimatot) #ML debug
                                        continue

                                    elif not os.path.exists(
                                        f"{rootdir}/{item}/combined_{item}.fits"
                                    ):
                                        print(
                                            f"Projecting image from the date {DATE}..."
                                        )
                                        image = CCDData.read(file, fix=False)
                                        print("DATE:",DATE) #ML debug
                                        print("file:",file) #ML debug
#                                        return # ML debug

                                        if i == 0:
                                            image0 = image.wcs
                                        proj_image = wcs_project(image, image0)
                                        reprojected.append(proj_image)
                            if nimatot > nmax:
                                print("More than",nmax,"images, will not combine/plot",nima) #ML debug
                                continue
                            elif not os.path.exists(
                                f"{rootdir}/{item}/combined_{item}.fits"
                            ):
                                combiner = Combiner(reprojected)
                                average = combiner.median_combine()
                                average.wcs = image0
                                savepath = f"{rootdir}/{item}/combined_{item}.fits"
                                average.write(savepath, overwrite=True)
                                plotpath = f"{rootdir}/{item}/combined_{item}.jpg"
                                print(savepath,plotpath)
                                #formatted plot w title:
                                plotitle = str("OBJECT: %s (%.2f,%.2f), DATE:%s\nFILTER:%s (%i images,total.exp=%.1f s)" % (OBJECT, ltarget.ra.degree, ltarget.dec.degree, DATE, FILTER, nima, totexp))
#                                print(savepath,plotpath,resdpi,ltarget.ra.degree,ltarget.dec.degree,radcirc,plotitle)
                                plot_fits(savepath,plotpath,resdpi,ltarget.ra.degree,ltarget.dec.degree,radcirc,plotitle)

    if MODE == "d":
        print("Available proposals:", proposal_id_lst)
        print("Enter proposal id:")
        PROPOSAL_ID = str(input())
        if PROPOSAL_ID == "q":
            print("Exiting software")
            return
        while PROPOSAL_ID not in proposal_id_lst:
            print("Invalid proposal id, please try again:")
            PROPOSAL_ID = str(input())
            if PROPOSAL_ID == "q":
                print("Exiting software")
                return

        targets = np.array(os.listdir(f"analysis/{PROPOSAL_ID}"))
        date_df = pd.DataFrame()
        for i, target in enumerate(targets):
            temp_df = pd.DataFrame(
                {target: np.array(os.listdir(f"analysis/{PROPOSAL_ID}/{target}"))}
            )
            date_df = pd.concat([date_df, temp_df], axis=1)

        flat_dates = np.ndarray.flatten(np.array(date_df))
        flat_dates = flat_dates[~pd.isnull(flat_dates)]
        flat_dates.sort()
        print("Available dates:", flat_dates)
        print("Enter date (YYYYMMDD):")
        DATE = str(input())

#        print("flat dates:",flat_dates) #ML debug
#        print("date_df:",date_df) #ML debug

        if DATE == "q":
            print("Exiting software")
            return

        while DATE not in flat_dates:
            print("Invalid date, please try again:")
            DATE = str(input())
            if DATE == "q":
                print("Exiting software")
                return

        for i, target in enumerate(targets):
            dates_for_target = date_df[target]
            dates_for_target = np.array(dates_for_target[~pd.isnull(dates_for_target)])
            if DATE in dates_for_target:
                rootdir = f"analysis/{PROPOSAL_ID}/{target}/{DATE}"
                for item in os.listdir(rootdir):
                    if os.path.isdir(f"{rootdir}/{item}"):
#                        files_in_dir = np.array(os.listdir(f"{rootdir}/{item}"))
#                        file_locs = np.core.defchararray.add(
#                            f"{rootdir}/{item}/", files_in_dir
#                        )
# ML. changed these lines to make sure that we only select TJO*imc.fits files
                        file_locs = np.array(glob.glob(str(rootdir+'/'+item+'/TJO*imc.fits')))
                        reprojected = []
```

```
                            totexp=0.  #ML cumulative exposure
                            nima=0     #image counter
                            nimatot=len(file_locs) # ML: count total nr. oof images in this data/filter. if more than nmax, we will skip reprojection
                            print("we have in total:",nimatot) #ML debug
                            for i, file in enumerate(file_locs):
#                                print("file1:",file) #ML debug
#                                print("DATE1",DATE) #ML debug
                                hdu_lst = fits.open(file)
                                #ML: Read and save sky coordinates of target:
                                RA=hdu_lst[0].header["OBJRA"]    # Target RA in hh:mm:ss (J2000)
                                DEC=hdu_lst[0].header["OBJDEC"] # Target DEC in dd:am:as (J2000)
                                ltarget=SkyCoord(RA,DEC,unit=(u.hourangle,u.deg),frame='icrs')

                                #Read more info from header:
                                OBJECT=hdu_lst[0].header["OBJECT"].replace(" ", "") # object name
                                EXPTIME=hdu_lst[0].header["EXPTIME"] # exposure time (seconds)
                                FILTER=hdu_lst[0].header["FILTER"].replace(" ", "") # filter
#                                DATE=hdu_lst[0].header["DATE-OBS"] # date (of last exposure in sequence)
                                #count images and total exposure:
                                totexp = totexp + EXPTIME
                                nima = nima + 1

#                                print(RA,DEC)
#                                print(ltarget.ra.degree,ltarget.dec.degree)

                                if (
                                    hdu_lst[0].header["naxis1"] == naxis1
                                    and hdu_lst[0].header["naxis2"] == naxis2
                                ):
                                    print(f"Cropping and analyzing {DATE}...")
                                    image = CCDData.read(file)
                                    trimmed = ccdproc.trim_image(
                                        image[xmin:xmax, ymin:ymax]
                                    )
                                    saveto = file
                                    trimmed.write(saveto, overwrite=True, format="fits")
                                    if not os.path.exists(
                                        f"{rootdir}/{item}/combined_{item}.fits"
                                    ):
                                        if i == 0:
                                            image0 = trimmed.wcs
                                        proj_image = wcs_project(trimmed, image0)
                                        reprojected.append(proj_image)
                                else:
                                    if nimatot>nmax:
#                                        print("More than",nmax," images, will not reproject",nimatot) #ML debug
                                        continue

                                    elif not os.path.exists(
                                        f"{rootdir}/{item}/combined_{item}.fits"
                                    ):
                                        print(
                                            f"Projecting image from the date {DATE}..."
                                        )
                                        image = CCDData.read(file, fix=False)
                                        print("DATE:",DATE) #ML debug
                                        print("file:",file) #ML debug
#                                        return # ML debug

                                        if i == 0:
                                            image0 = image.wcs
                                        proj_image = wcs_project(image, image0)
                                        reprojected.append(proj_image)
                            if nimatot > nmax:
                                print("More than",nmax,"images, will not combine/plot",nima) #ML debug
                                continue
                            elif not os.path.exists(
                                f"{rootdir}/{item}/combined_{item}.fits"
                            ):
                                combiner = Combiner(reprojected)
                                average = combiner.median_combine()
                                average.wcs = image0
                                savepath = f"{rootdir}/{item}/combined_{item}.fits"
                                average.write(savepath, overwrite=True)
                                plotpath = f"{rootdir}/{item}/combined_{item}.jpg"
#                                print(savepath,plotpath)
                                #formatted plot w title:
                                plotitle = str("OBJECT: %s (%.2f,%.2f), DATE:%s\nFILTER:%s (%i images,total.exp=%.1f s)" % (OBJECT,ltarget.ra.degree,ltarget.dec.degree,DATE,FILTER,nima,totexp))

                                plot_fits(savepath,plotpath,resdpi,ltarget.ra.degree,ltarget.dec.degree,radcirc,plotitle)



    if MODE == "a":
        PROPOSAL_ID = config.ARCHIVE_PROPOSAL_ID
        if PROPOSAL_ID == "q":
            print("Exiting software")
            return
        while PROPOSAL_ID not in proposal_id_lst:
            print("Invalid proposal id, please try again:")
            PROPOSAL_ID = str(input())
            if PROPOSAL_ID == "q":
                print("Exiting software")
                return
        targets = np.array(os.listdir(f"analysis/{PROPOSAL_ID}"))
        date_df = pd.DataFrame()
        for i, target in enumerate(targets):
            temp_df = pd.DataFrame(
                {target: np.array(os.listdir(f"analysis/{PROPOSAL_ID}/{target}"))}
            )
            date_df = pd.concat([date_df, temp_df], axis=1)

        flat_dates = np.ndarray.flatten(np.array(date_df))
        flat_dates = flat_dates[~pd.isnull(flat_dates)]
        flat_dates.sort()

        if (config.DEBUG==1): print("flat dates:",flat_dates) #ML debug
        if (config.DEBUG==1): print("date_df:",date_df) #ML debug

        for DATE in flat_dates:
            for j, target in enumerate(targets):
                dates_for_target = date_df[target]
                dates_for_target = np.array(
                    dates_for_target[~pd.isnull(dates_for_target)]
                )
                if DATE in dates_for_target:
                    rootdir = f"analysis/{PROPOSAL_ID}/{target}/{DATE}"
#                    print("rootdir:",rootdir) # ML debug
                    for item in os.listdir(rootdir):
#                        print("item:",item) # ML debug
                        if os.path.isdir(f"{rootdir}/{item}"):
#                            files_in_dir = np.array(os.listdir(f"{rootdir}/{item}"))
#                            file_locs = np.core.defchararray.add(
#                                f"{rootdir}/{item}/", files_in_dir
#                            )
# ML. changed these lines to make sure that we only select TJO*im?.fits files
                            file_locs = np.array(glob.glob(str(rootdir+'/'+item+'/TJO*im?.fits')))
```

```python
                            reprojected = []
                            totexp=0.  #ML cumulative exposure
                            nima=0     #image counter
                            nimatot=len(file_locs) # ML: count total nr. oof images in this data/filter. if more than nmax, we will skip reprojection
                            for i, file in enumerate(file_locs):
                                hdu_lst = fits.open(file)
                                #ML: Read and save sky coordinates of target:
                                RA=hdu_lst[0].header["OBJRA"]    # Target RA in hh:mm:ss (J2000)
                                DEC=hdu_lst[0].header["OBJDEC"] # Target DEC in dd:am:as (J2000)
                                ltarget=SkyCoord(RA,DEC, unit=(u.hourangle,u.deg), frame='icrs')

                                #Read more info from header:
                                OBJECT=hdu_lst[0].header["OBJECT"].replace(" ", "")  # object name
                                EXPTIME=hdu_lst[0].header["EXPTIME"] # exposure time (seconds)
                                FILTER=hdu_lst[0].header["FILTER"].replace(" ", "")  # filter
#                                DATE=hdu_lst[0].header["DATE-OBS"] # date (of last exposure in sequence)
                                #count images and total exposure:
                                totexp = totexp + EXPTIME
                                nima = nima + 1

#                                print(RA,DEC)
#                                print(ltarget.ra,ltarget.dec)

                                if (
                                    hdu_lst[0].header["naxis1"] == naxis1
                                    and hdu_lst[0].header["naxis2"] == naxis2
                                ):
                                    image = CCDData.read(file)
                                    trimmed = ccdproc.trim_image(
                                        image[xmin:xmax, ymin:ymax]
                                    )
                                    saveto = file
                                    trimmed.write(saveto, overwrite=True, format="fits")
                                    if not os.path.exists(
                                        f"{rootdir}/{item}/combined_{item}.fits"
                                    ):
                                        if i == 0:
                                            image0 = trimmed.wcs
                                        proj_image = wcs_project(trimmed, image0)
                                        reprojected.append(proj_image)
                                else:
                                    if nimatot>nmax:
                                        if (config.DEBUG==1): print("More than",nmax," images, will not reproject",nimatot) #ML debug
                                        continue

                                    elif not os.path.exists(
                                        f"{rootdir}/{item}/combined_{item}.fits"
                                    ):
                                        image = CCDData.read(file, fix=False)
                                        if (config.DEBUG==1): print("DATE:",DATE) #ML debug
                                        if (config.DEBUG==1): print("file:",file) #ML debug
                                        if i == 0:
                                            image0 = image.wcs
                                        proj_image = wcs_project(image, image0)
                                        reprojected.append(proj_image)
                            if nimatot > nmax:
                                if (config.DEBUG==1): print("More than",nmax," images, will not combine/plot",nima) #ML debug
                                continue
                            elif not os.path.exists(
                                f"{rootdir}/{item}/combined_{item}.fits"
                            ):
                                combiner = Combiner(reprojected)
                                average = combiner.median_combine()
                                average.wcs = image0
                                savepath = f"{rootdir}/{item}/combined_{item}.fits"
                                average.write(savepath, overwrite=True)
                                plotpath = f"{rootdir}/{item}/combined_{item}.jpg"
                                if (config.DEBUG==1): print(savepath,plotpath)
                                #formatted plot w title:
                                plotitle = str("OBJECT: %s (%.2f,%.2f), DATE:%s\n FILTER:%s (%i images,total.exp=%.1f s)" % (OBJECT,ltarget.ra.degree,ltarget.dec.degree,DATE,FILTER,nima,totexp))

                                plot_fits(savepath,plotpath,resdpi,ltarget.ra.degree,ltarget.dec.degree,radcirc,plotitle)


if __name__ == "__main__":
    main()
```

NTNU

Norwegian University of
Science and Technology