universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

"Spectroscopic Analysis of Cool Stars: Software Development and Activity Analysis."

verfasst von / submitted by

Fabian Haberhauer, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2022 / Vienna, 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 861
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Astronomie
Betreut von / Supervisor:	Mag. Dr. Theresa Rank-Lüftinger

Spectroscopic Analysis of Cool Stars: Software Development and Activity Analysis.

Fabian Haberhauer, BSc

Abstract

This works aims at creating a software tool in the form of a graphical application with the purpose of visualising and analysing stellar spectra, from both observational data and synthetic models. The software is based on an earlier code by Kochukhov (2007, 2018), but written in Python instead of IDL, with the main motivation of being freely available for research and educational purposes, while also focusing on adhering to certain principles of software design, aimed at promoting future development and expansion of the capabilities of this software. The core functionality of this powerful application includes the ability to manipulate various parameters of synthetic spectra and visualise their effects on the spectral profile. These parameters include: radial velocity, rotational velocity broadening ($v \sin i$) and macroturbulent broadening (v_{mac}).

Another main function of this new software is to measure equivalent widths and integrated flux in user-defined passbands. These measurement methods are used to analyse stellar activity indicators (Ca II H&K, Hα, Ca II IR triplet) in a sample of cool stars (mostly spectral types G to M) and investigate the relation of stellar activity to other parameters like effective temperature or surface gravity and metallicity. The variability of these activity indicators across different observation epochs for the same stars are also analysed by means of Lomb-Scargle periodograms.

From the available data, there is now strong evidence for a clear correlation of stellar activity to any of the investigated fundamental stellar parameters, with the exception of a minor increase of some activity indicators with decreasing effective temperature, but this is believed by other authors to be an artifact of missing bolometric flux calibration. Periodic changes in stellar activity is found in a small number of stars, but in many cases this analysis would benefit from a larger amount of data.

Zusammenfassung

Diese Arbeit befasst sich mit der Entwicklung und Erprobung eines Programms zur graphischen Darstellung, Manipulation und wissenschaftlichen Analyse von spektroskopischen Daten, sowohl beobachteter Sternspektren als auch mit Modellrechnungen erzeugter synthetischer Spektren. Das Programm selbst baut dabei auf einer früheren Software von Kochukhov (2007, 2018) auf, ist aber im Gegensatz dazu in der Programmiersprache Python verfasst, da diese, anders als das vom Vorläufer verwendete kostenpflichtige IDL, frei zugänglich ist und das neuentwickelte Programm daher kostenlos für wissenschaftliche Zwecke und zur universitären Ausbildung zur Verfügung gestellt werden kann. Bei der Entwicklung des neuen Programms wird zudem ein Fokus auf diverse Prinzipien der Softwareentwicklung gelegt, welche die Langlebigkeit und zukünftige Ausbaufähigkeit des Programms gewährleisten und fördern.

Die Grundfunktionalität des neuentwickelten Programms liegt neben der erwähnten Darstellung von Sternspektren bei der Manipulationsmöglichkeit diverser Parameter synthetischer Spektren, deren Effekt auf das Erscheinungsbild des Spektrums dargestellt werden. Diese Parameter beinhalten unter Anderem: Radialgeschwindigkeit, Rotationsverbreiterung ($v \sin i$) und Macroturbulenzen (v_{mac}). Eine weitere Hauptfunktion ist die Messung von Äquivalenzbreiten und integrierten Flüssen in benutzerdefinierten Filtern.

Diese genannten Messmethoden werden anschließend verwendet, um anhand einer Auswahl an kühlen Sternen (hauptsächlich Spektraltypen G bis M) stellare Aktivitätsindikatoren zu vermessen und mit den gewonnenen Werten die möglichen Beziehungen dieser zu anderen stellaren Parametern wie Effektivtemperatur, Oberflächenschwerebeschleunigung und Metallizität festzustellen. Ebenso untersucht wird die zeitliche Veränderlichkeit dieser Aktivitätsindikatoren für einzelne Sterne durch Anwendung von Lomb-Scargle Periodogrammen, wodurch in einzelnen Fällen eine ebensolche Variabilität festgestellt werden konnte, obgleich in vielen Fällen eine eindeutige Aussage anhand der vorliegenden Daten nur schwer möglich ist.

Thanks

I would like to take this opportunity to express my infinite gratitude to my supervisor Mag. Dr. Theresa Rank-Lüftinger, without whom none of this would have been possible. Very special thanks goes also to Dr. Sudeshna Boro Saikia for the invaluable additional support provided, as well as the large volume of data for me to analyse, even though she could not be listed as an official co-supervisor due to bureaucratic reasons. I am looking forward to continue working with these two amazingly supportive people and great researchers.

Contents

Chapter 1

Introduction

Spectroscopy is arguably one of the most important tools for determining physical properties of stars, from basic parameters like temperature, surface gravity, chemical abundances and radial velocity, to turbulence and rotational velocity, as well as more complex properties such as stellar activity. Especially the latter is of growing interest for various fields within astronomy and astrophysics, from stellar astrophysics all the way to star-planet interactions and habitability studies. While stars of spectral types G and hotter are quite well studied in that regard, stars of spectral types K and cooler are underrepresented, even though they are overabundant in the universe and due to their long lifetimes have lately been at a focus for exoplanet habitability. A major limiting factor for habitability around low-mass stars is their activity, which can result in outbursts leading to increased radiation exposure of any potentially habitable planets. A solid understanding of this activity is thus beneficial to a wider field of research, and correlating cool star activity to other stellar parameters would enhance our understanding of exoplanet habitability around a wide stellar mass range, as well as providing a better insight into astrophysical stellar processes, such as magnetic field generation. (Güdel, 2007)

1.1 Motivation

In order to efficiently analyse stellar spectroscopic data in general, it is imperative to have a capable and easy-to-use software to visualise observed spectra, compare them to simulated synthetic spectra, as well as to analyse the observed spectra. Such a software would ideally be freely available for both research and educational purposes and work on most common platforms (i.e. operating systems). It should also be easily expandable and adaptable, accept third-party extensions and generally be designed with a long service life and possible future community contribution in mind.

The author of this work has himself experienced the hassle of having to deal with costly licensed (subscription based) specialised software in undergraduate and graduate astronomy education. This usually means one licence is installed on a university server and students have to work via remote access. It would be preferable to have a software solution that allows for free and uncomplicated distribution among students to use on their own systems. The educational benefits of a software tool that allows for manipulation of stellar parameters in synthetic stellar spectra are considerable. It will allow astronomy students to directly visualise the impact of those parameters on the shape of spectral lines in a hands-on manner, as opposed to just seeing examples in a static lecture. Assignments to manually fit parameters like radial velocity and stellar rotation velocity to a given real-world observed spectrum could also be easily carried out using such a software, especially when it is able to be executed on the students' own computers, without the need for remote access to a server.

Besides the educational aspect, the second and arguably even more important purpose of such a software tool will be astronomical research. The aim is to develop a freely available software that can be used by professional astronomers to analyse spectra, and can easily be expanded to add new features in the future. The inspiration for this software and it's features was `binmag` (Kochukhov, 2007, 2018), and many of the core modules are essentially ported from `binmag` to the new software.

The decision was made to develop this new software in *Python*[1], a language which has seen a steady increase in popularity, both in the astronomical scientific community as well as in general. The name of the newly developed software was chosen to be SynSPEC.py, which stands for **Syn**thetic **Spec**tra in **py**thon, as visualising and manipulating synthetic (and observed) stellar spectra is the main objective of the software, besides analysis of observed spectra.

1.2 Software development

1.2.1 Existing software solutions

Commonly used powerful software tools for reduction and analysis of spectroscopic data, like `binmag` (Kochukhov, 2007, 2018)and *Spectroscopy Made Easy (SME)* (Valenti and Piskunov, 1996, 2012; Piskunov and Valenti, 2017) are written in *IDL*, as are a number of smaller tools for e.g. reduction tasks like continuum normalisation. Most of the current functions of SynSPEC.py(except the passband measurements, see section 2.4.1.1) are already implemented in `binmag`, whereas some functions of `binmag` (specifically automatic parameters fitting and composite binary spectra) are currently not implemented in SynSPEC.py. *SME* has even more abilities, like e.g. fitting of stellar parameters including effective temperature and surface gravity, as well as element abundances, among many more capabilities. However, the *IDL* programming language is only available at a subscription-based regular payment. This effectively creates a barrier to those who wish to use the functionality of these tools. Additionally, both *IDL* itself and the mentioned tools can be challenging to correctly set up and use, especially for an inexperienced user.

For those reasons, it was deemed necessary to develop SynSPEC.py, which fulfills the requirements of being freely available and easy to execute on any operating system and has many advantages in the human-readability[2] of the code, which can drastically improve maintainability. Furthermore, since *Python* is already seeing widespread and increasing use in the

[1] `https://www.python.org/`

[2] i.e. how easy the code can be understood by a human, see `https://en.wikipedia.org/wiki/Computer_programming#Readability_of_source_code`

professional astronomical community, developing SySPECpy in this language removes the need to deal with yet another programming language; it is more efficient to improve in one general purpose language than to have only basic skills in multiple ones.

1.2.2 Requirements for a new software

While it would be sufficient for the scope of this thesis to create a software tool that delivers all the functionality necessary to thoroughly analyse the given spectroscopic data (see chapter 3), this would be of very limited use to the scientific community as a whole. In order to potentially be expandable and usable beyond this work, the code should be designed following a number of principles commonly summarised under the umbrella term *clean code* (see Martin, 2008). These code design paradigms are meant to significantly increase the longevity, maintainability, expandability and comprehensibility of a software. This includes, e.g. extensive refactoring wherever feasible, a high degree of modularity (see also section 2.1), code documentation, as well as general best-practise coding principles, such as *DRY* ("Don't Repeat Yourself") and *YAGNI* ("You ain't gonna need it") among others. Specific to *Python*, this work will adhere to the paradigms of PEP8[3], PEP20[4] and (to a lesser extant) PEP484[5] as much as feasible.

1.2.3 Object-orientation

Object-orientated programming (OOP) is native to Python and in many cases a useful choice for anything more complex than a short script. However, it should be applied in a reasonable way, its advantages and disadvantages weighed in each case. "Blind" application of classes for everything may result in a too complicated interface for simple applications. Thus, this work used what might be called a "mixed approach", keeping low-level modules like file I/O handling at a functional level[6], while using classes for high-level parts of the code, e.g. everything related to the graphical user interface (see section 2.2) and handling spectra with parameters once loaded in the main application (see section 2.7.1).

1.2.4 Usage of existing packages

A major advantage of python is the plenitude of existing packages, which can easily be imported and used in a new script, as opposed to the need of writing "boilerplate" code for simply tasks each time. If the package is sufficiently well developed (which is true for almost all packages used in this work), this may also offer a considerable advantage in performance, since in the back-end, computation-intensive operations can come in the form of C-compiled extensions, which are naturally superior in performance to purely interpreted python code. Some of these packages have had a development cycle of multiple decades by a large pro-

[3] `https://www.python.org/dev/peps/pep-0008/`
[4] `https://www.python.org/dev/peps/pep-0020/`
[5] `https://www.python.org/dev/peps/pep-0484/`
[6] Technically, Python follows the "everything's an object"-approach, meaning there hardly is a purely functional way of programming in Python, since even functions themselves are (first-class) object, as are simple integers or floats. This has certain advantages which are beyond this discussion. In this work, anything restricted to relatively simple built-in classes and objects like `numpy` arrays (which are also classes) is referred to as non-OOP.

fessional development team (usually open-source) and will thus also be a lot more stable as well as the aforementioned superior performance, compared to self-implemented code. For all these reasons, it is usually advantageous to use existing packages wherever possible, as long as it does not make it more cumbersome than implementing a new solution.

The following packages were used to a large extent in this work:

- numpy (`https://numpy.org/`)
- scipy (`https://scipy.org/`)
- astropy (`https://www.astropy.org/`)
- wxpython (`https://www.wxpython.org/`)
- matplotlib (`https://matplotlib.org/`)

1.3 Previous research on activity of cool stars

Although the *Mount Wilson* program (Wilson, 1968; Duncan et al., 1991; Baliunas et al., 1995) measured the chromospheric activity of stars down to a spectral type of M2 (Baliunas et al., 1995), theirs analysis focused more on K5 to F stars. This is less surprising considering their method of quantifying chromospheric activity was based on the Ca II H&K lines (the so-called “S-index” (Vaughan et al., 1978; Duncan et al., 1991; Baliunas et al., 1995)), which are of decreasing usefulness for stars cooler than mid-M, due to relatively little flux being emitted at those wavelengths for such stellar effective temperatures (Cincunegui et al., 2007). However, even though the *Mount Wilson* program started more than half a century ago, S-index measurements taken using different instruments are still calibrated to the *Mount Wilson* values. It is thus very useful to have a software tool which is able to measure the S-index (among other stellar activity diagnostics) using spectra from different spectrographs.

A possible alternative activity indicator is the Hα line, and there is evidence for a correlation between the activity in Ca II H&K lines and Hα (Giampapa et al., 1989; Strassmeier et al., 1990; Robinson et al., 1990; Montes et al., 1995), although Cincunegui et al. (2007) later showed that this correlation might not be as clear.

Important contributions for activity of late-type stars using the Ca II H&K lines were also made by Wright et al. (2004); Mittag et al. (2013); Astudillo-Defru et al. (2017); Perdelwitz et al. (2021) and recently Boro Saikia et al. (2018) provided an exhaustive compilation of chromospheric activity in cool stars, including an analysis of activity-cycle periods, the latter with methodology very similar to this work (see section 3.5).

The aim of this thesis is thus twofold: Firstly, chapter 2 will discuss the development of SySPEC.py and how this software can be used to carry out various spectroscopic measurements. Secondly, chapter 3 will cover the actual activity analysis performed using SySPEC.py and the scientific results thereof.

Chapter 2

Description of the Software

2.1 Main application vs. standalone modules

A main objective of this work was to create SySPEC.py in the form of an application with a graphical user interface, or GUI. The point of this GUI application is to have all functionality, like opening and displaying data, comparing it to adaptable simulated data, taking measurements, etc. available in one place and usable as a full-fledged application, without the need for any programming experience to use it. This means that all actions can be performed by a mouse-click (or drag) and all results are immediately visible, either as a change in the plot or as numerical values. All of this is also especially useful when SySPEC.py is used educationally, as students are able to intuitively work with the software without the need for previous programming skills, so the focus can be directly on the scientific question rather than having to deal with setting up some code first.

In contrast to that, a more experienced user may be in a position where it is advantageous to be able to use certain functionalities like e.g. measurements or file I/O in the context of a *Python* script, specifically tailored to a certain task. A good example of this would be the automated analysis of a larger amount of data, first using the `spectrum_loader` module to open the files containing this data, then using the `measurements` sub-package to perform the actual analysis, like e.g. measuring equivalent widths. If the individual datasets do not need to be viewed by the operator and all that is desired is the measurement output in the form of an output file, it is not necessary and would in fact hinder performance to employ a graphical interface. For this reason and similar cases, a lot of the core functionality of SySPEC.py is available in modules or sub-packages, which can be loaded into an independent script without the need to run the actual GUI application. See the package documentation for a full description of the available modules and functions.

2.2 General GUI layout

The layout of the graphical interface of SySPEC.py is inspired by the *IDL* widget code `binmag` (Kochukhov, 2007, 2018). This will enable a relatively easy transition for anyone who has previously been working with `binmag`, while also being intuitive for new users. All standard operations, like opening files, are found in logical places.

The main window features a menu bar (as usual in most desktop applications, discussed in detail in section 2.3), the main content panel (plot or notes) and the panel tabs at the bottom. The combination of tabs and corresponding content panels is hereinafter referred to as a “notebook”, in accordance with the naming conventions of *wxPython*. The notebook containing the notes and plot panels with tab selection at the bottom of the window will be called the main notebook. This notebook currently has two types of panels, namely the `notes panel` and the `plot panel`.

2.2.1 Notes Panel

The `notes panel` is currently the default panel the user will see upon launching the application. It simply consists of a large text box whose content can be edited by the user. The result of any measurements taken in a plot panel will also be written into these notes, together with the name of the spectrum the measurement was performed upon. Any changes, automatic or by the user, to the text of the notes panel are immediately saved to a `.txt` file in the application directory. It is planned for future versions to extend the features of the notes panel.

2.2.2 Plot Panel

The `plot panel` provides the core functionality of SySPEC.py. It consists of the large plot window where spectral data is plotted, including the control buttons for the plot, and the so-called `adjustment notebook`. This is the second use of tabs in SySPEC.py, each open spectrum (file) has a corresponding tab. The name of the tab is identical to the file name from which the spectrum was loaded.

2.2.2.1 Plotting spectral data

To display a spectrum in the plot window, select the file containing the data in the file open dialog, which can be found in the File menu, see section 2.3.1.1 for details. SySPEC.py currently supports a number of different file formats, listed in section 2.5. The correct format (default is `.prf`) must be selected in the bottom right of the dialog window, otherwise the files will not be displayed in the selected folder. The dialog box itself is the standard file open dialog native to the operating system (Windows, Linux or MacOS). After loading is completed (which may take a few moments for larger files), the spectrum is plotted in the plotting window. The color is assigned automatically but can be changed in the `adjustment notebook`. Plotted spectra can also be hidden there. More options are found in the plot menu, see section 2.3.1.3.

For synthetic spectra with a line list in the header, the spectral lines will be annotated (i.e. labelled) in the plot. The number of line labels is reduced if the currently visible range contains to many labelled lines to plot all annotations. The lines to be labelled are chosen based on line prominence, as well as relative depth. This relative depth is also visualised with a red bar next to the label itself. An example of this is shown in Figure 2.1.

The limits of the plot are set automatically. The x-axis limits are based on the wavelengths

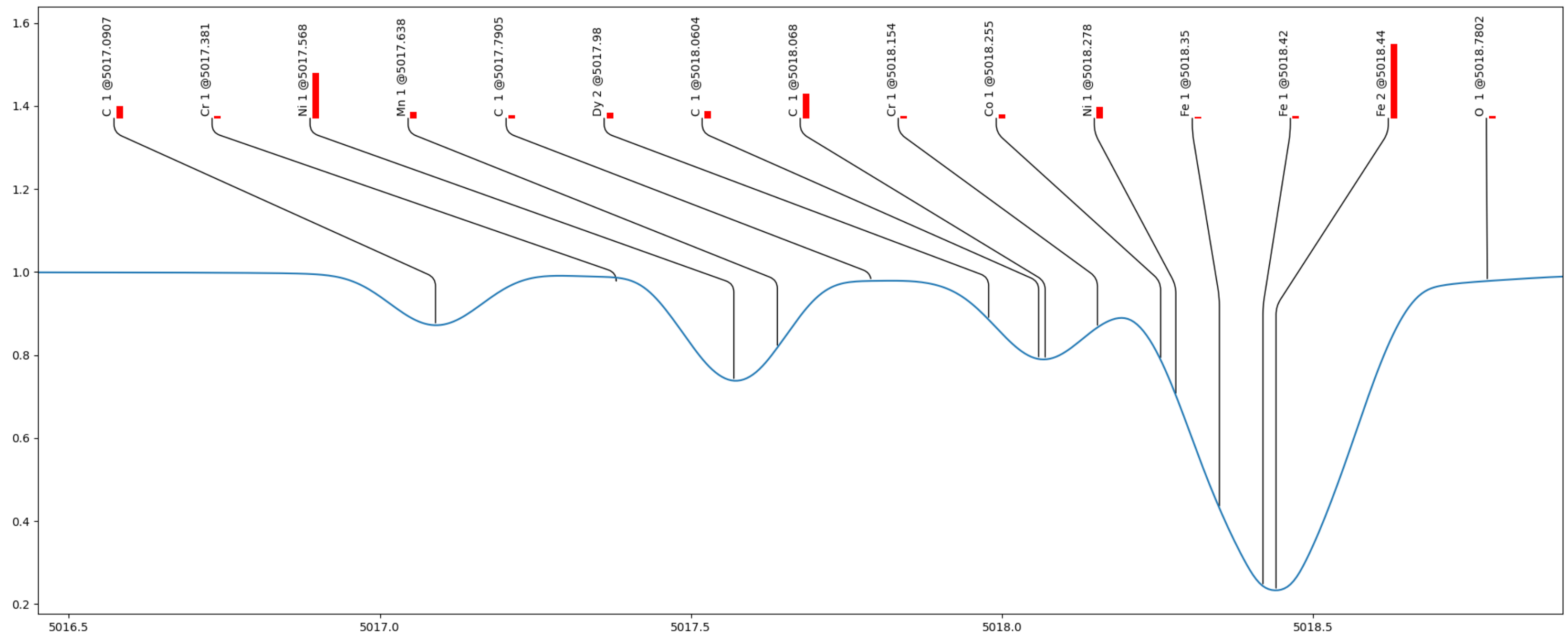

Figure 2.1: Example of a synthetic spectrum plotted with SySPEC.py. Note the labels annotating the spectral lines. The red bars next to the label text are scaled based on the relative depth of the spectral line. These can be effectively used to determine the contribution of each line in a case where two or more lines are blended. For example the second and third from the right in this plot (“`Fe 2 @5018.44`” and “`Fe 1 @5018.42`”), the Fe 2 line is mainly responsible for the visible absorption line, whereas the contribution of Fe 1 is negligible.

of the plotted spectra, while the y-axis is by default limited to 0 to 1, assuming the spectra will be normalised. This can be changed manually in the plot menu, see section 2.3.1.3 for details.

2.2.2.2 The `adjustment notebook`

The panels of the `adjustment notebook` contain multiple controls for adjusting (hence the name) a number of parameters in the spectrum, as well as controlling the appearance of the plot itself (like e.g. line colour). The layout of these controls is very similar to `binmag` (Kochukhov, 2007, 2018). The controls available for each spectrum depend on the type of the spectrum, i.e. observational or synthetic. More control options are available in the latter case, like adjusting v_{mac} and $v \sin i$ of the synthesis. Observed spectra have an additional control for vertical offset, usually to account for less-than-optimal continuum normalisation. The radial velocity v_{rad} can be adjusted in any type of spectrum, although it is recommended to adjust this parameter in the observation rather than the synthesis, especially when comparing a small number of observations to a variety of synthetic spectra. This also has the benefit, that newly loaded syntheses don't have to be adjusted all the time.

2.3 Menu bar

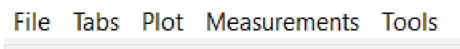

Figure 2.2: The menu bar at the top of the application window.

The menu bar (shown in Figure 2.2), as is common in most applications, is a part of the graphical user interface located along the top of the application window, used to control key features of the program, such as opening files or performing measurements. Is is designed to appear familiar and natural to the user, as the structure follows that of many applications, with functions specific to SynSPEC.py in between. For example, the user would probably expect to open a file via the File > Open menu, so that is where this function will be located. The implementation is done using *wxPython*'s built-in menu functionality, using a flexible system of "menu builders" defined in section 2.3.2.

2.3.1 Structure

The menu bar currently consists of 5 main (drop-down) menus, namely:

2.3.1.1 File

The File menu currently contains the following menu items:

- File > Open to launch the (standard builtin OS-specific) file open dialogue, to load spectra from files. The file format has to be selected in the dialogue, see section 2.5 for supported formats. This menu item is only enabled when a plot panel is selected (i.e. not the Notes panel).

- File ≫ About for basic information about SySPEC.py.
- File ≫ Close to close the application. Alternatively just close the window.

More menu items, like the ability to save sessions, are under consideration for future versions.

2.3.1.2 Tabs

The Tabs menu allows the user to switch between the main panels, currently meaning the Tabs ≫ Notes and Tabs ≫ Plot panel. This has the same effect as switching those panels at the corresponding tabs on the bottom of the application window. The menu item Tabs ≫ New Panel is currently a placeholder, but will allow the addition of another panel, for example a second plot window, in future versions.

2.3.1.3 Plot/Notes

The third menu switches its content and title based on the currently selected (bottom) panel. When the notes panel is selected, the Notes menu is currently empty. When the plot panel is selected, the Plot menu contains the following menu items:

- Plot ≫ Reset scale to reset the view of the plot window so all loaded spectra fit inside the plotting space. This is primarily meant as a "safety feature", in case the "Home" button on the plot navigation bar does not perform correctly, which is a known issue.
- Plot ≫ y-limits to manually set the lower and upper limits for the y-axis, mostly for cases of unnormalised data, or data containing emission features rising well above the continuum level.
- Plot ≫ Show all to set all loaded spectra to be shown.
- Plot ≫ Hide all to set all loaded spectra to not be shown.

2.3.1.4 Measurements

The Measurements menu is created dynamically upon launching the application. It's contents depend on the available measurement modules, see section 2.4. This menu is only available when a plot panel is selected.

2.3.1.5 Tools

The Tools menu currently only contains a placeholder for a future Tools ≫ Preferences menu, and the functionality to incorporate radial velocity correction values defined in a separate file, Tools ≫ fix vrad (beta).

2.3.2 Menu builders

The menu is constructed via a so-called *menu builder*, inspired by the *Working With Menus*[1] article in the *wxPyWiki*. In general, a menu builder is a `Tuple`, consisting of four items:

[1] `https://wiki.wxpython.org/WorkingWithMenus`

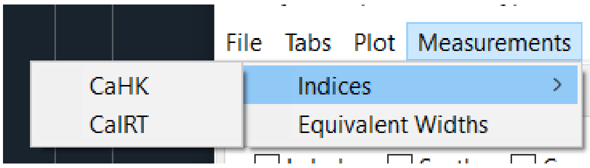

Figure 2.3: Example of a sub-menu. Note that the Measurement menu shown here is created dynamically and may thus appear slightly different depending on the loaded measurement modules, as discussed in section 2.4.

1. The menu item ID; in anything but some predefined special cases like File » Open or File » Close, this will be `wx.ID_ANY` (which is equal to `-1`). The special cases are used to seamlessly integrate with OS-specific inter-application standard keyboard shortcuts, such as ctrl + O for File » Open under Windows.

2. The caption, which is the text that will be displayed in the menu.

3. The text that will be displayed in the status bar when hovering over the respective menu item. Can be an empty string.

4. The handler is the callable that will process the event when the user clicks the menu item. It must have a single argument `event`, which can in many cases be ignored in the handler itself, but has to be included for combatibility purposes with *wxPython*. Alternatively, this can be a list containing other builders, in which case a sub-menu will be created, see Figure 2.3 for an example of a sub-menu. An abstract example of this can be found in listing 2.1. The handler can also be `None`, in which case the menu item will be without a function.

One or (usually) multiple menu builders are combined in a list for each top-level menu, such as File. This is combined with the caption of this top-level menu into a tuple. This tuple corresponds to the total menu, meaning a menu as a whole can e.g. be returned by a function. This is for example how the Tabs and Measurements menus are created. Finally, a separator line can be added in a menu by inserting `None` instead of a builder in the list of menu items.

```
(wx.ID_ANY, "&foo", "1", [
    (wx.ID_ANY, "&bar", "2", None),
    None,
    (wx.ID_ANY, "&baz", "3", None)
    ])

```

Listing 2.1: Example of a builder containing two other builders. This will create a menu item "foo" which has two submenus "bar" and "baz" separated by a line, with status bar help messages of "1", "2" and "3" respectively. In this example, no handlers are specified, so clicking those menu item would not actually in any actions.

2.4 The measurements system

This section describes the various measurements which can be performed in SySPEC.py. Measurements in this context are defined as calculations performed on a loaded spectrum without altering the data itself. Examples of such measurements implemented in the code include the estimation of equivalent widths for spectral lines, integration of the flux in a given spectral window (= passband) and the combination of multiple passbands into a single value. Measurements are usually performed on observed spectra, but may also be applied to synthetic spectra. In most cases, the spectrum currently selected in the `adjustment notebook` (see section 2.2.2.2) will be used in the calculation (some measurements may be applied to all currently loaded spectra).

All types of measurement have their respective menu item in the `measure` menu of the main menu bar, sometimes with sub-menus below them for different variations of that measurement. Each measure menu item corresponds to a module in the `measure` subpackage. These modules contain most of the logic[2] required for performing, visualising and saving their measurement, and integrate with the main menu via a `register` function, which will be discussed in detail in section 2.4.2. The measurements system thus fulfills the requirement of modularity, since various new measurement methods can be added as modules in the future, including the possibility of third-party extensions and plugins. See section 2.4.2 for instructions of how to create your own measurement module. Some of the existing measurement modules may also be used as standalone modules, as the can easily be imported from the measure subpackage using python's standard import syntax. The functionality of that module can then be used outside the context of the main application. A good example of this is the `Passbands` module, see section 2.4.1.1.

2.4.1 Existing measurements

2.4.1.1 Passband

The Passband module provides functionality for quantitative flux measurements within so-called passbands, or filters. The input data is assumed to be of the same 2D-array format internally used throughout this application, with the first column being the wavelength and the second column the intensity. The latter may or may not be normalised, however some types of passband (see below) assume normalisation. Multiple passbands may be combined into a single measurement, called an index (see section 2.4.1.1), after the prototypical "S-Index" (Vaughan et al., 1978; Duncan et al., 1991).

This module is also designed to be used in a "standalone" way as part of a script, without the main application running, using python's standard import syntax. If used in that way, it is recommended to load spectral data from various file formats using the matching function from the `utils.spectrum_loader` module.

Passbands A Passband is defined by it's central wavelength, full-width half-maximum (FWHM) and shape. In the case of simple geometric passbands, the shape can directly be

[2] In some cases, certain properties, methods and functions may be required from another module, such as `utils.spectra` or `ui_aux`.

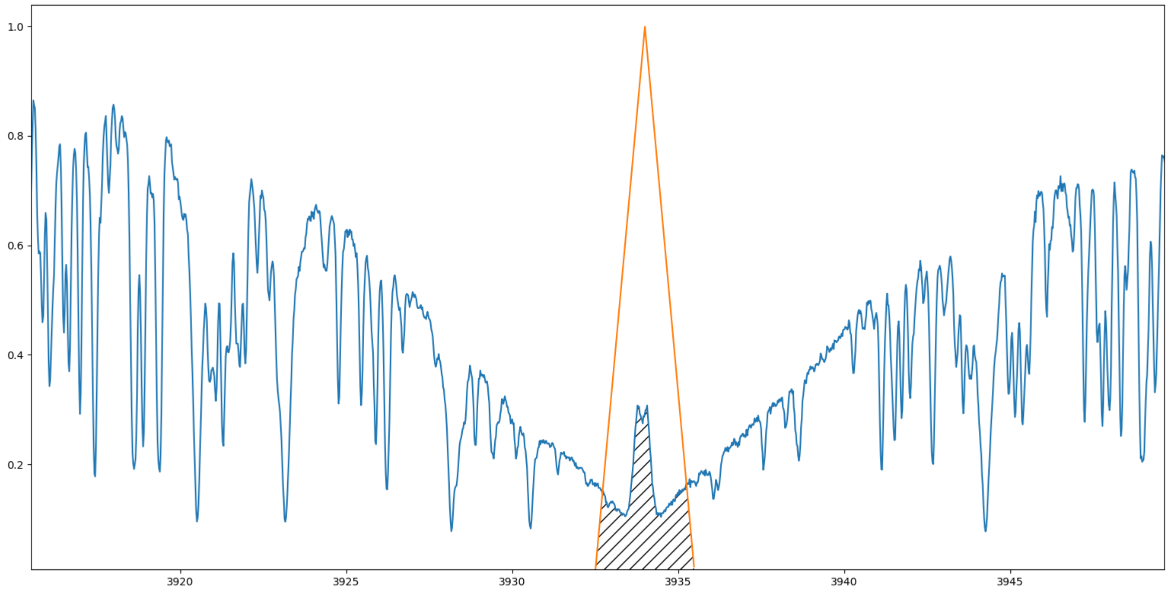

Figure 2.4: Example of a narrow triangular filter centered on an emission feature. The shaded area indicates the area of integration.

inferred from the other two parameters. Currently, only rectangular (`RectPassband`) and (isosceles) triangular (`TriPassband`) passbands are implemented, however it is planned to add support for custom filter curves in the future, to include standard photometric filters. Figure 2.4 shows an example of a triangular passband, Figure 2.5 shows an index measurement (see below) consisting of rectangular passbands. In the rectangular case, integration is simply performed under the data, in an interval defined by start and end points, calculated from central wavelength and fwhm. For this reason, it does not matter whether the data is normalised or not, it just has to be kept in mind when interpreting the results. In the case of the triangular passband, however, the integration is performed under the combined curve of the filter and the data, usually meaning the left- and rightmost parts are integrated under the curve while the center (between the intersection points of data and filter) is integrated under the data. The triangle itself is defined as having a height of one, meaning the use of unnormalised data may yield imprecise results. In all cases of passbands which are not the trivial rectangular case, the filter curve will be interpolated onto the wavelength grid given by the data. In all cases, integration is performed using numpy's implementation of Simpson's rule for numerical integration.

Initialisation of a rectangular or triangular passband instance is simply done via the two aforementioned parameters **central wavelength** and **FWHM**. These two float values must be in the same unit, and also consistent with the unit of the x-axis of the data which will be used to perform the integration. It is important to note that the passband instance is defined *without* the data, meaning only the "shape" of the filter will be saved. This means a passband instance can be defined once and used on multiple datasets (assuming consistent units) without any of the information from those datasets persisting in the passband instance.

To perform the data integration, each passband subclass has a specific `integrate` method.

The first argument of this method is the data to be integrated, in the form of a two-dimensional array-like structure with the first column consisting of the wavelengths and the second column of the corresponding intensities. The wavelength range must cover at least the range used by the filter, otherwise an error is raised. The passband provides two additional attributes after initialisation, `start` and `stop`. These values correspond to starting and ending wavelengths used by the respective passband instance, in the same units as the values given for initialisation. These attributes can thus be used to check before calling the integration, whether a given data array covers the range of the passband. This work focuses on the analysis of high-resolution spectra ($R > 60\,000$) and care should be taken when using low resolution data. Another problematic case might occur in a dataset which has at least one value before the starting and after the ending value of the passband, but no values in between, i.e. no datapoints within the filtercurve. Such a case may occur when using a particular narrow filter on a dataset with low spectral resolution. For example, some of the passbands used in this work were as narrow as 2 Å; at those values, a resolution below $R \sim 10\,000$ might cause a runtime error, if less than 2 data points (i.e. pixels) end up within the passband's boundaries.

Indices An Index is a measurement, consisting of an arbitrary combination of passband measurements, named after the prototypical "S-Index" (Duncan et al., 1991; Baliunas et al., 1995). Usually, the calculation of the final index value involves dividing by a continuum flux (also measured using a passband), so the result is dimensionless; although custom indices may be defined that result in a dimensional quantity. Figure 2.5 shows an example of an index measurement. All information to construct an index, such as the passbands involved and the formula for calculating the final value are defined in a YAML file specific to each index. It is also possible to include instrument-specific constants, such as those needed for the "S-index". In that case, the instrument must be specified for the data used in the evaluation of the index measurement (for example in the for of a FITS-header keyword `INSTRUM`). If coefficients are used but no instrument information is given, or if the instrument given does not have coefficient values specified in the YAML file or instrument information is passed to an index not using coefficients, an exception is raised. To evaluate the index measurement on a given dataset, simply call the index object. As with the passband objects it consists of, the index object only holds general information upon creation; the data is not stored in the instance, but passed for evaluation in each call. When calling an index object, the arguments are data (in the same format as specified for the passbands), optionally `coeff_key` (instrument name) and optionally axis. The axis argument can be used to plot the filters into an existing matplotlib axis object, mostly intended for debugging and creating visualisations. The following will be plotted if axis is given: The filter curve (or boundaries for rectangular passbands), the zero level and the area which is integrated will be marked with a hatched fill.

As the calculation for the final index value is specified as a string (see below for a detailed discussion), this string must be evaluated by the python interpreter in order to perform the calculation. Since python's `eval` function does have some potential security loopholes, a number of measures are taken to mitigate this risk. Firstly, the string containing the calculation is compiled. This allows for `names` to be extracted. These names may only be a) one of the names of a passband that the index is using, or b) one of the coefficients.

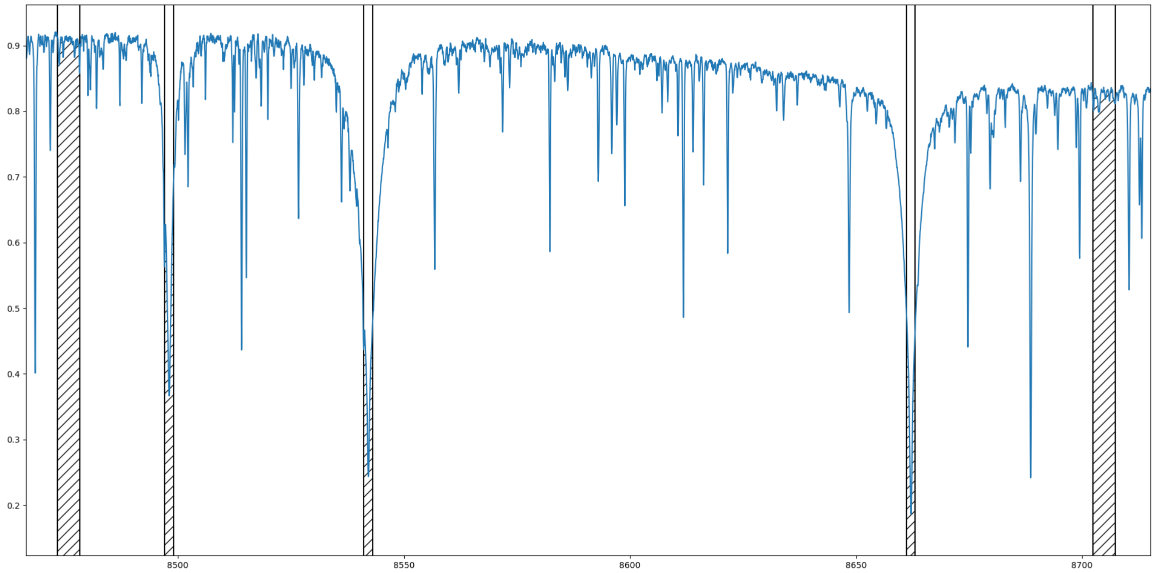

Figure 2.5: Example of an index measurement defined by Marsden et al. (2014) consisting of three narrow rectangular passbands centered on Ca II lines and two broader rectangular passbands to measure continuum levels. Again note the shaded area in the passbands showing the area that will be integrated.

Any other name or expression will raise an exception. The passband and coefficients are then replaced with floats before the string is evaluated. In the case of passband names, this numerical value is the integrated flux from the given data, while in the case of coefficients, the corresponding coefficient value for the given instrument name is taken from the config file. Thus, the string to be evaluated should only contain numerical values and operations. Finally, the `__builtins__` are set to an empty dictionary, and an empty dict is also passed for the `locals` argument of `eval`. All this should keep the risk of using `eval` at an acceptably low level.

Defining custom indices It is possible and intended to define custom index measurement profiles. If a YAML file containing all necessary information to construct the index is placed in the `measure.profiles` folder, the index will be auto-recognised and added to the measurements tab of the main application. Of course, if the `measure` subpackage is used in a "standalone" way (as described in section 2.4.1.1), the config file can be placed anywhere, if the path is specified when initialising the respective `Index` object.

A full example of the structure for such a YAML config file is shown in Listing 2.2. The first part called `Setup` contains all passbands used in the index. Each passband is defined by three parameters: `type`, `cent` and `fwhm`. `type` currently supports either `rect` for rectangular passbands or `tri` for triangular ones. `cent` is the central wavelength and `fwhm` the full-width half-maximum, as described above. In the example shown, all wavelength values are in Ångström. The second part is `Coeffs`, which are instrument-specific coefficients used in the calculation of the index. This part is optional and only used if such coefficients are used in the

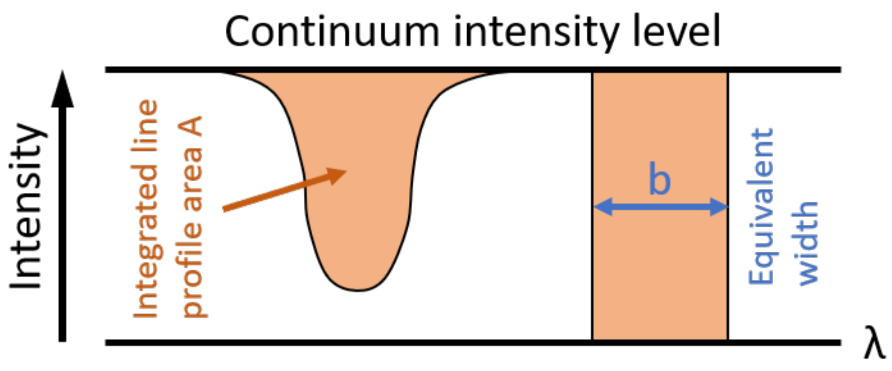

Figure 2.6: The principle behind the "equivalent width".The width b is chosen such that the shaded areas A are equal.

calculation. Since the coefficients are instrument-specific[3], they must be specified for each instrument used, as shown in the example with `NARVAL` and `ESPaDOnS`. The spelling must match that used in the `INSTRUM` field of the header from the file containing the observational data. The last part of the config file is `Calc`, the "cooking recipe" for the index. It consists of the names of the passbands defined in the `Setup` part as well as those of the coefficients (if any) defined in the `Coeffs` part, and any mathematical operator, as shown in the example. One blank space must be placed before and after each passband or coefficient name, to ensure successful recognition by the compiler. Lastly, any number of commentary lines starting with `#` can be added to the config file as well.

2.4.1.2 Equivalent widths

An important and often used characteristic of a spectral line is its so-called equivalent width, which is defined as the width of a hypothetical rectangle with height of unity and an area equal to the integrated area of the spectral line (Palla and Stahler, 2008), see Figure 2.6 for an illustration of this general principle. Equation 2.1 shows the mathematical definition of the equivalent width W_λ, where F_λ denotes the flux as a function of wavelength and F_C the continuum level (see below for a deatiled discussion of F_C). The `equivalent_widths` module provides functionality to perform such measurements, either integrated into the main application or as a standalone module.

$$W_\lambda = \int \left(1 - \frac{F_\lambda}{F_C}\right) d\lambda \tag{2.1}$$

When used within the main application (via the Measurements menu), a separate dialog window will open and two vertical lines will appear in the plot window. These lines can be dragged with the mouse and allow the manual definition of the boundaries for the spectral line whose area is to be integrated for estimating its equivalent width. The wavelength values

[3] Non-instrument specific constant values can simply be added to the `Calc` formula.

```yaml
---
Setup:
  CaH:
    type: tri
    cent: 3968.47
    fwhm: 1.09
  CaK:
    type: tri
    cent: 3933.66
    fwhm: 1.09
  CaR:
    type: rect
    cent: 4001
    fwhm: 20
  CaV:
    type: rect
    cent: 3901
    fwhm: 20
Coeffs:
  NARVAL:
    a: 12.873
    b: 2.502
    c: 8.877
    d: 4.271
    e: 1.183E-3
  ESPaDOnS:
    a: 7.999
    b: -3.904
    c: 1.15
    d: 1.289
    e: -.069
# a...H, b...K, c...R, d...V, e...offset
Calc: ( a * CaH + b * CaK ) / ( c * CaR + d * CaV ) + e
...
```

Listing 2.2: Example of a YAML config file used to define an Index measurement.

will appear in the dialog box, and can also be entered there instead of dragging the lines. After the boundaries are defined, the Calculate button will perform the integration and the result will be displayed in the corresponding field. The Save button will write the result together with the name of the spectrum used for the calculation to the notes panel.

To use the equivalent widths in a standalone way, import the `equivalent_width` function from the `equivalent_widths` module and refer to its documentation.

Continuum level An important consideration when calculating equivalent widths is the correct estimation of the continuum level (F_C in eq. 2.1). Especially when comparing multiple observation with less than perfect continuum reduction or lower signal to noise ratios, care has to be taken in deciding whether an apparent change in a spectral line's equivalent width is caused by an actual change in the line profile/depth or just by a flawed continuum level estimation. In the current implementation, the continuum is calculated simply as the average flux value between the first and last data point given. This is not ideal, since it might easily be affected by noise. Thus, the user is encouraged to visually check the start and end points for any extreme noise when using the equivalent width function, and to adjust accordingly if deemed necessary. It is planned to implement a more robust way of estimating the continuum level in future versions.

2.4.2 How to register

In order to register a custom measurement module for integration in the GUI of the main application, the minimum requirements are as follows:

- The `.py` file containing the module must be in the `measure` subfolder of the package.
- The module must contain a function called `register`, which takes exactly one argument, the `parent`, which in the case of the measurement system will be the main frame of the application containing the menu bar.
- This `register` function must return a valid `builder` to create the menu item and bind it to the desired action. See section 2.3.2 for a detailed discussion of builders.
- Since the builder requires a handler, this handler must be defined and returned in the context of the register function. The handler must take exactly one argument, `event`. Usually, this event is not used in the handler function, but must be included for compatibility with *wxPython*. In most cases, the handler will contain the necessary logic to trigger the calculation associated with the measurement and save the result, or to launch a custom dialog which may then perform the measurement (as is the case in the equivalent width module for example, see section 2.4.1.2). In the simplest case, the handler method simply wraps around the `.Show()` method of a `wx.Dialog` object or subclass, passing the `parent` argument from the register function containing the application main frame, with all other logic performed at this dialog.

In this context some useful convenience methods are:

- `parent.save_to_notes(str)` ... save a string to the noted panel.

- `parent.current_spectrum` ... gives access to the currently selected spectrum, see the documentation of the `spectrum` class and its subclasses for available properties.
- in the two examples given above, it is assumed that `parent` is the main frame passed to the register function.

2.5 Loading spectral data

Spectroscopic data comes in many different file formats. The most common is some sort of ASCII table, often including various headers and sometimes footers. The format of the data table itself is usually straightforward, with the first column representing the wavelength and the second one the flux or intensity. This is sometimes followed by additional columns for things like error levels or unnormalised flux, both can be ignored if all that is desired is a "quick and dirty" look at the data. The other common general format is FITS (Wells et al., 1981), a well-known and widely used data format in the astronomical community. In the context of spectroscopic data, FITS files are usually used to represent observational data, rather than synthetic spectra, where the ASCII format dominates. FITS files are much more advanced than simple ASCII tables, with the possibility of multiple header-data-units to store additional data and meta-data, which can be very useful in data analysis.

Unfortunately, there is no real consensus on how exactly to represent spectroscopic data in a FITS file, and there exist many different ways of doing so. This presents a challenge for developing a stable file parser that is able to correctly load many different "flavours" of spectroscopic FITS files. See section 2.5.1 for a detailed discussion on this topic.

The following commonly used files formats are currently supported by SySPEC.py:

- FITS files (`.fit`, `.fits`, `.FITS`) containing observational data
- ASCII tables (`.obs`, `.s`) with headers no longer than two lines
- *Synth3/Synthmag/Synmast* (Kochukhov, 2007, 2012) synthetic spectra (`.out`)
- *Synth3/Synthmag/Synmast* integrated synthetic spectra (`.prf`)

Each of those formats has a separate loader function within the `utils.spectrum_loader` module. Those functions all follow the same basic structure: They take the file path as a parameter and return a `(header, data)` tuple, where `header` may in turn be a tuple containing multiple information, or a dictionary containing header fields, depending on the files format.

2.5.1 FITS files

The trivial case for spectroscopic data in a FITS (Wells et al., 1981) file is one header-data-unit ("HDU" hereafter) containing metadata in the header and a simple table of wavelengths and intensities in the data section. However, many FITS files containing observational data have multiple HDUs, where usually the first contains general metadata about the observation and no data section, and the second HDU contains the actual spectrum, with table-specific information in the header section. Additional HDUs are sometimes present, with different purposes such as providing uncertainties or unnormalised data, though these might also come in additional columns of the second HDU's data table. Another application for multiple HDU is spectropolarimetric data in the Stokes-QUV format.

The actual handling of FITS files is performed using `astropy.io.fits`, which usually has no problem returning `numpy` arrays from a HDU data section. These are often structured arrays, rather than simple ones, which makes column indexing more practical. However, some FITS files use the `recarray` format, which is currently converted to a regular structured array in a very "hacky" way. Better handling of this case is planned.

The `loadfits` function provides additional arguments for returning columns other than the first two (usually if the unnormalised spectrum is desired) and for also including an uncertainty column, if present in the data. This functionality is currently not used in the main application, but may be useful if the loader module is imported as a standalone utility. Note that the described functions are still under development and not fully tested. A detailed explanation of how to use the `loadfits` function and it's additional parameters is provided in the documentation.

2.5.2 `.prf` files

The `.prf` format (Kochukhov, 2007) is used for synthetic spectra after disk integration (see section 2.6). This means the spectrum in a `.prf` file can be used without any further processing. The file structure is as follows:

1. *VALD* (Piskunov et al., 1995; Kupka et al., 1999) header (ASCII table) containing spectral line data
2. Number of wavelengths (data points) and the limb darkening value
3. The spectrum as a wavelength, intensity ASCII table
4. Values for vsini and vmac
5. Values for fluxes (red, blue, center)

The *VALD* header is handled by a separate loader, which returns a structured array containing the list of spectral lines and, optionally, also returns the file object and keyword dictionary, both are used if there is more information to read from a file after the *VALD* header (such is the case for e.g. `.prf` files).

2.5.3 Synth3 (`.out`) files

The `.out` format generated by *Synth3* (Kochukhov, 2007, 2012) provides a synthetic spectrum pre-disk integration. The integration is performed once upon loading the spectrum from the corresponding file and again each time a parameter like $v \sin i$ or v_{mac} is changed. The advantage of this is that the resulting synthetic spectrum will be more accurate compared to using only Gaussian broadening on an already integrated spectrum. The downside is a slightly increased calculation time whenever a parameter is changed. However, with the current implementation, the whole integration takes a maximum of a few seconds (tested on a lower-end computer) even for a detailed spectrum with resolution of 10^6 over a spectral range of more than 5000 Å, and comparatively less for smaller spectra, so this added calculation time is not considered a large issue.

The file structure of the `.out` format is similar to the `.prf` format described in the previous section, also starting with a *VALD* header containing spectral line data (Kupka et al., 1999). This is followed by the values for μ, which is the cosine of the angle between the outward normal and the line of sight (Valenti and Piskunov, 1996). Each μ value (typically *Synth3* will produce 7 different values) corresponds to one column in the following table of intensities, which is to be integrated. These are again split into a small section of continuum intensities, followed by the intensities for each wavelength, as given by the resolution and spectral range of the synthesis, which can be up to several 10^5.

2.5.4 Processing of observational data

Observational spectroscopic data (from FITS files or ASCII tables) is passed through a number of auxiliary functions, before being returned by the loader. These functions process the data, mostly to ensure correct formatting. The following steps are performed:

1. Ensure a wavelength axis exists (2D spectrum). If not, try to create one using information from the header. If no wavelength axis exists and there is no header, an exception will be raised.
2. Ensure proper orientation (row-first) as standard in *Numpy*.
3. Ensure only wavelength and intensity columns are returned (remove any additional columns).
4. Restore the spectral continuity (i.e. remove "Echelle-loops" from overlapping orders) of the data.
5. Ensure the wavelength axis is (likely) in Ångström. This is currently done by multiplying by 10 if the smallest value of the wavelength column is below 1000. It is recognised that this implementation may cause issues when using data from different regimes of the electromagnetic spectrum. However, it works well for all the data used in this work. It is planned to have an option to configure the behaviour of this function in future versions.

2.6 Disk integration

While it might be sufficient in some cases to include the effects of rotational ($v \sin i$) and macroturbulent (v_{mac}) broadening by applying simple convolutions (Doppler in the case of $v \sin i$ and Gaussian in the case of v_{mac}) to the already integrated synthetic spectrum using an approach which might be called "post-production", much more precise results can be achieved by re-integrating the spectrum for each change in these two parameters. The macroturbulent velocity is in that case interpreted as the radial-tangential anisotropic macroturbulence (Gray, 2005). Integration is performed by the `utils.rds3v.py` module, the code in which is adapted from `rds3v.pro` by Kochukhov (2007).

The input for the disk integration is a `.out` file (see section 2.5.3), which will usually include "intensity spectra at 7 angles [μ] between the line of sight and normal to the stellar surface" (Kochukhov, 2007), however the implementation allows for any number of μ values greater than one. Numerical integration is then performed by summing over annuli weighted by their relative areas. Within each annulus, the intensity is convolved with $v \sin i$ using a kernel constructed with respect to the velocity at the edge of each annulus and with v_{mac} using separate radial and tangential kernels. The relative weights of these two macroturbulence kernels are currently hard-coded to be equal. This could easily be changed in future versions, however as argued by Gray (2005), this assumption should be justified in most cases.

The result of the integration is then re-normalised. Gaussian instrumental broadening (in the form of $\lambda/\Delta\lambda$) can also be specified, as well as the resolution on which the spectrum is interpolated (again as $\lambda/\Delta\lambda$, default is 10^6). All velocities are assumed to be given in $\mathrm{km\,s^{-1}}$.

2.7 Additional auxiliary modules

This section provides a short overview of various other module found in the `utils` subpackage. Most of these are used to outsource one or more functions and/or classes containing logic used in multiple places in SynSPEC-py; or to make the main module better structured.

2.7.1 `spectra`

The data and meta-data from spectra, synthetic or observed, once loaded from a file, is handled using an object-oriented (OOP) approach. This is useful not only to conveniently store data (i.e. the spectrum itself) and meta-data (i.e. the header) in one place, but also to have certain parameters, such as e.g. radial velocity, stored alongside and to handle the adaptions to the data which come with changing parameters, like shifting the spectrum when the radial velocity is changed. The `spectra` module contains a number of classes, as well as auxiliary functions, to facilitate this.

The `Spectrum` class provides most of the basic functionality common to all types of spectra, like plotting and changing simple parameters, such as radial velocity. All parameters are implemented as `@property` and stored using a `Parameter` class and a `WeakKeyDictionary` to assign them to the correct instance of a `Spectrum` or a subclass. The `Parameter` class is useful, because it allows to implement a general logic to handle "getter" and "setters" for the `@property`. This is used for checking if the assigned value falls between predefined boundaries (e.g. no negative values are allowed for rotational velocity or macroturbulence), as well as for triggering re-calculation and re-plotting of the spectrum whenever a parameter has changed.

2.7.2 `ui_aux`

The `ui_aux` module contains a number of auxiliary classes and functions to support the creation and handling of the GUI in the main application using `wxPython`. The central class is `AdjustmentElement`, which provides a container for a functional element and a static text label next to it. The functional element is determined and handled by a variety of subclasses:

- `SpinElement` for "spin"-type controls, which are a combination of a numerical text box and two small "up" and "down" buttons to increase or decrease the numerical value inside the box.
- `CheckElement` for a check-box that can be ticked or unticked.
- `ButtonElement` for a simple clickable button.
- `BoxElement` for a editable textbox.
- `ColourElement` for a "colour picker" control, which launches the OS-native colour dialog box. Used e.g. to change the colour of a line in a plot.

Integration into the `parent` container is inherited from the superclass. Some subclasses, namely `CheckElement` and `ButtonElement`, have a built-in label in the control itself. In those cases, the extra label of `AdjustmentElement` is deactivated. Each subclass must also define a `value` property to access the value (text, numerical, or Boolean) stored in the control. The property can be read-only or read and write. This will also handle any necessary type conversion in both directions.

2.7.3 `annotations`

This module essentially holds the `PercentBar` class, as well as some additional code to produce a testing environment for that class. The sole purpose of this class is to provide the rectangular bars visualising the relative depths of annotated spectral lines when displaying a synthetic spectrum, as shown in Figure 2.1. To accomplish this, the `PercentBar` class heavily modifies the behavior of the box native to `matplotlib` annotations. This would normally be used to construct a frame around the text of an annotation, but here this box is shifted away from the text and given a height corresponding to the relative depth of the spectral line to be labelled. The inspiration for this feature is also taken from `binmag` (Kochukhov, 2007, 2018).

2.7.4 `gbroad`

The `gbroad` module holds only one function, `gauss_broad`, which is used to smooth a spectrum by convolution with a Gaussian function with a given FWHM. This was transcribed directly from the corresponding IDL function in `binmag` (Kochukhov, 2007, 2018). For the convolution, only the intensity values are needed, so the spectrum is given and returned as a 1D array.

2.7.5 `rebin`

Similar to the previously discussed module, `rebin` is used only to provide functionality for rebinning a spectrum to a specified resolution using interpolation (cubic by default). The basic logic is again taken from `binmag` (Kochukhov, 2007, 2018), but this function has been modified to make good use of the existing `scipy.interpolate` module, which achieves better performance and accuracy. The module currently also contains some code to visualise the different results of this rebinning using different methods and resolutions.

2.7.6 `restore_cont_extended`

Spectra from Echelle-type spectrographs usually have the issue of overlapping orders, meaning if stored in a simple 2D table, the wavelength axis is not continuous. This can lead to multiple issues: Firstly, when the spectrum is plotted without considering this, the plot will show lines running diagonally over the spectrum, where two orders overlap, an effect referred to as "Echelle-loops" in this work. While this is only a cosmetic issue, removing these lines leads to a better usability of the plot. Secondly, the ends of an Echelle order are often subject to much increased noise, which can make it harder to see the actual signal of an overlapping order. Thirdly, and most important from a technical standpoint, when performing automated measurements on the spectrum, a non-continuous wavelength axis could lead to unexpected issues. Such measurements might also accidentally use the high-noise part at the end of an order, likely rendering the result of the measurement useless. For all the reasons mentioned, it is thus a good idea to remove these overlaps when loading spectral data.

The `restore_cont_extended` module provides the `restore_continuity` function to perform exactly that. This function will loop over a 2D spectrum and recognise discontinuities in the wavelength axis. Wherever two orders overlap, the (possibly noisy) ends of the orders are clipped and the remaining overlapping section is handled by the `_mk_middle_range` subroutine. In the current implementation, it simply uses data from the order with more data points and thus higher resolution, as that may vary slightly between two adjacent orders. It is planned to investigate rebinning the data from both orders to obtain better signal-to-noise ratio in future versions of SySPEC.py.

Chapter 3

Data analysis and results

Having established the necessary software toolset, I used SySPEC.py to analyse high-resolution spectra and determine the activity level of cool main sequence stars. The data comprises of archival datasets taken using the high-resolution echelle spectrographs NARVAL and ESPaDOnS. NARVAL at TBL (Telescope Bernard Lyot) is a twin of ESPaDOnS at CFHT (Canada-France-Hawaii Telescope). The characteristics of these two instruments are listed in the table below. A detailed description of the instruments is given by Aurière (2003) and Donati et al. (2006) respectively.

Instrument	**NARVAL**	**ESPaDOnS**
Location	Pic du Midi (France)	Mauna Kea (Hawai'i)
Telescope mirror	2.03 m	3.6 m
Instrument type	echelle spectrograph and spectropolarimeter	
Wavelength range	370-1000 nm	370-1050 nm
Spectral resolution	65 000	68 000-81 000

The data used is taken from the POLARBASE database and it comprises of slightly more than 1000 stars. Out of these, 82 were selected for detailed analysis, based on the following criteria:

- Spectral type of mid-F or later (Spectral classifications were taken from *Simbad*[1])
- Main-sequence star
- At least 30 usable observations

In a second selection, the number of stars was reduced to 75, based on a closer look at the available data, mostly in terms of data quality, where some had to be rejected. The full list of all 75 stars is given in the appendix in Table A.1. Data quality was assessed by visual inspection. Finally, after a preliminary analysis (as discussed below), the dataset was further reduced to 45 stars. In terms of spectral type (again taken from *Simbad*), these include:

[1] `https://simbad.u-strasbg.fr/simbad/`

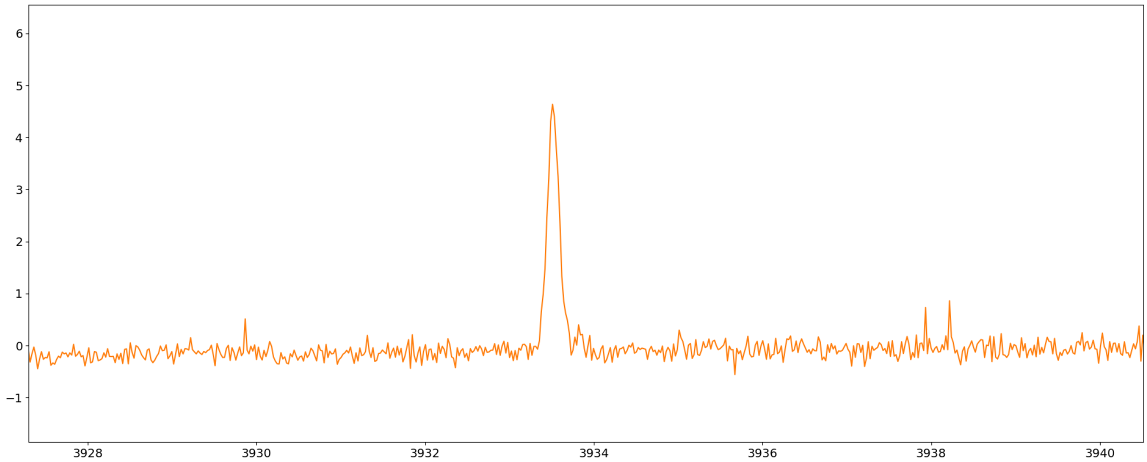

Figure 3.1: Example of a particularly high Ca II K emission observed in *Ross 154*. Also note the continuum level is partially below zero, indicating a problem with continuum normalisation of this spectrum, which can lead to erroneous results for activity indicators.

- 14 M dwarfs
- 12 K dwarfs
- 11 G solar-type stars
- 6 F main-sequence stars
- 1 A1IV star for comparison
- 1 K2V+DA binary[2]

All of these stars were analysed with respect to the activity indicators discussed in the following section. Out of this sample, 9 stars with either a significant variability in their activity or a strong non-thermal chromospheric emission are looked at in a detailed analysis in sections 3.4 and 3.5.

3.1 Measuring stellar activity

3.1.1 S-index

One of the most commonly used indicators for stellar activity is the so-called S-index (Vaughan et al., 1978; Duncan et al., 1991; Baliunas et al., 1995), which is measuring the flux in the Ca II H&K lines, normalised to the continuum on both sides of those lines. The complete formula is given by Wright et al. (2004):

$$S = \frac{aF_H + bF_K}{cF_R + dF_V} + e \tag{3.1}$$

[2]The only binary star was included because there was a large dataset of good quality available, and the measured activity indicators show an interesting periodic pattern (see section 3.5).

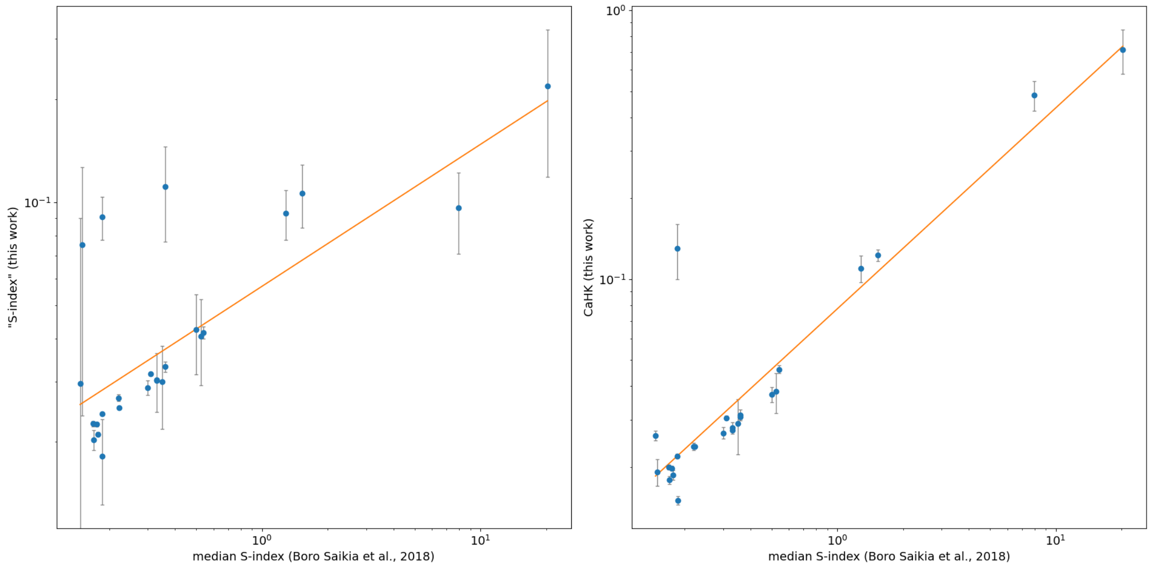

Figure 3.2: Correlation between the median S-index from Boro Saikia et al. (2018) and the S-(left panel) and CaHK (right panel) indices measured in this work. The CaHK index on the right is clearly more consistent to the literature values. Each data point represents one star, the values are taken as the median over all spectra for each star, the error bars are the standard deviation (only for values from this work). The correlation coefficient for the fit shown in both plots increases from $R^2 = 0.44$ to $R^2 = 0.86$ when the CaHK index is used instead of the S-index.

where F_H and F_K are the integrated fluxes in two triangular passbands (see also Figure 2.4) with an FWHM of 1.09 Å centered on the cores of the Ca II H&K lines, and F_R and F_V the integrated fluxes in two rectangular passbands with an FWHM of 20 Å on either side of the two lines to normalise the measurement to the continuum. The coefficients a, b, c, d and e are instrument-specific. Values for these for both NARVAL and ESPaDOnS were taken from Marsden et al. (2014).

3.1.2 CaHK index

After attempting to measure the S-index in the available data and comparing these preliminary results to the literature, it was found that there was a systematic issue in estimating the S-index, likely related to the normalisation in the spectra and the implementation of the triangular passbands used in the S-index. This was especially apparent for more active stars with very high emissions in the Ca II H&K lines, which exceed the continuum level by a factor of up to 10. An example of such a case is shown in Figure 3.1. It was therefore decided to introduce the "CaHK index" which is defined in a similar way to the classical S-index, but avoids the issue of triangular passbands. It is given by the following formula:

$$Ca_{HK} = \frac{F_H + F_K}{F_R + F_V} \tag{3.2}$$

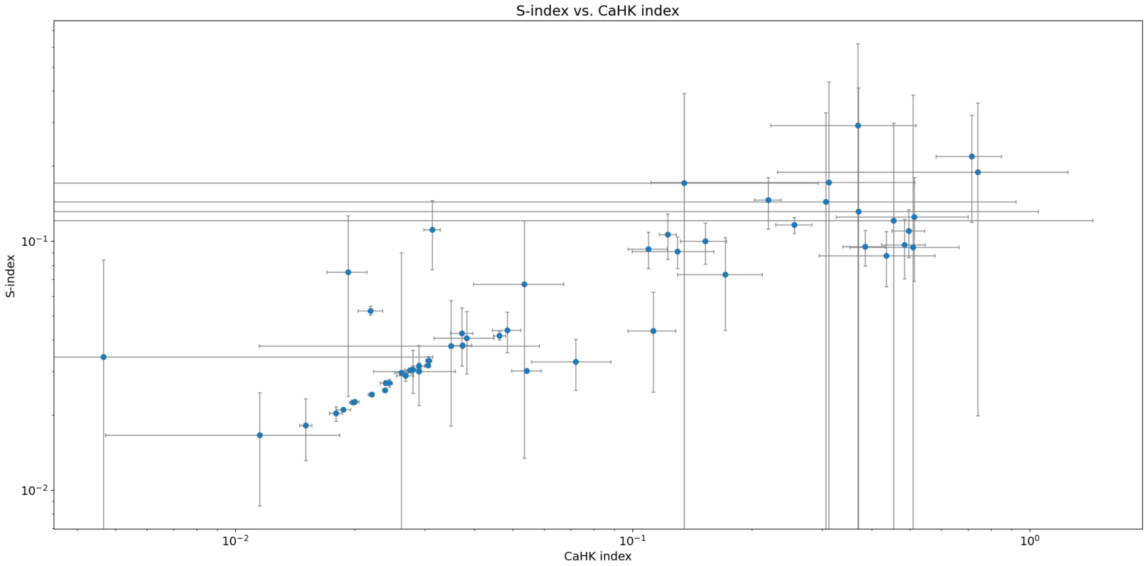

Figure 3.3: Correlation between the S- and CaHK indices measured in this work. Each data point represents one star, the values are taken as the median over all spectra for each star, the error bars are the standard deviation.

where F_H and F_K are the integrated fluxes in two rectangular passbands with an FWHM of 2.18 Å centered on the cores of the Ca II H&K lines and F_R and F_V are defined identically to the S-index (see above). This formula is very similar to that of the S-index, but used rectangular instead of triangular passbands with the same 2.18 Å base width, and also omitting the coefficients used to calibrate the calculation to different instruments.

In order to compare the quality of the S- and CaHK index measurements, their correlation with known literature values was investigated. For 27 stars, a median S-index value was taken from Boro Saikia et al. (2018) and plotted against the S- and CaHK index values from this work (Figure 3.2, both axes logarithmic). As becomes apparent in these plots, the correlation between the CaHK index and the literature values is much better and consistent over two orders of magnitude, with the exception of one outlier. The still existing discrepancy in measured values of around one order of magnitude is thought to be caused by the absence of calibrated coefficients in the CaHK index as well as by the slightly different passband profiles used to measure the flux in the Ca II H&K lines. Based on this consistency, it was decided to rely on this CaHK index as a proxy for the S-index.

Additionally, the correlation between the S-index and CaHK index values measured in this work was looked at and is shown in Figure 3.3. As can be seen in this plot, the two indices do correlate relatively well for low values, which is consistent with the assumption that higher activity levels caused the issues seen in measuring the S-index.

3.1.3 Other indicators

Since the S- and CaHK indices are measured in a part of the spectrum which has shown to be susceptible to high levels of noise compared to other parts of the visual spectrum and is also close to the edge of the spectral range of the instruments used, it has proven to be somewhat difficult to use the S-index as a reliable measurement. Thus, two other, non instrument-specific, "indices" were added, which lie in more "quiet" parts of the visual spectrum: the Ca II IRT index and the Hα-index, both defined in Marsden et al. (2014), and references therein. The formula for these indices are as follows:

$$Ca_{IRT} = \frac{F_{8498} + F_{8542} + F_{8662}}{F_V + F_R} \tag{3.3}$$

$$H\alpha = \frac{F_{H\alpha}}{F_V + F_R} \tag{3.4}$$

where F_{8498}, F_{8542} and F_{8662} denote the integrated flux in 2 Å wide rectangular passbands, each centered on one of the lines of the Ca II IR Triplet , at 8498, 8542 and 8662 Å respectively. $F_{H\alpha}$ is the integrated flux of a 3.6 Å wide rectangular passband centered on the Hα line. F_V and F_V in both cases are integrated fluxes in two rectangular passbands to the left and right of the Ca II IR Triplet and the Hα line, respectively, to normalise the measurements to the continuum in that part of the spectrum. Note that F_V and F_R are different in each of the three indices. All definitions used here were taken from Marsden et al. (2014).

3.2 Consistency and limitations of activity indicators

3.2.1 Correlations

In order to investigate the correlation of the three different activity indicators ("indices") used in this work, the values measured for each index were plotted against each other (figures 3.4, 3.5 and 3.6). In these plots, each data point represents one star. The x and y values are the median of all measurements of that particular star for the respective index, the error bars are the standard deviation. Figure 3.4 shows the CaHK index plotted against the CaIRT index (both axes are logarithmic). The correlation in this case is rather vague, although the large uncertainties for higher values make it difficult to give a qualitative statement. There is however somewhat of a trend to be seen in the plot: stars with higher value for the CaHK index always have a higher value in the CaIRT index as well.

A much clearer correlation is observed in the relation between the CaHK and Hα indices, as seen in Figure 3.5 (again both axes are logarithmic). At a value of around 1.2 for the Hα index and 0.5 for the CaHK index, there is a sharp turn-off, after which the CaHK index values do not increase significantly for an increased Hα index, although only 8 stars fall into this region. A possible explanation for this is that some stars in this sample (see section 3.4) feature a very high Hα emission, which can be up to a factor of 2 or 3 higher than the emission in the Ca II H&K lines, an effect that would also be represented in the corresponding index values. The correlation between the CaIRT and Hα indices (Figure 3.6, both axes logarithmic) is also decent, again with a larger spread towards higher values, which may be explained with higher Hα emissions as mentioned in the previous paragraph.

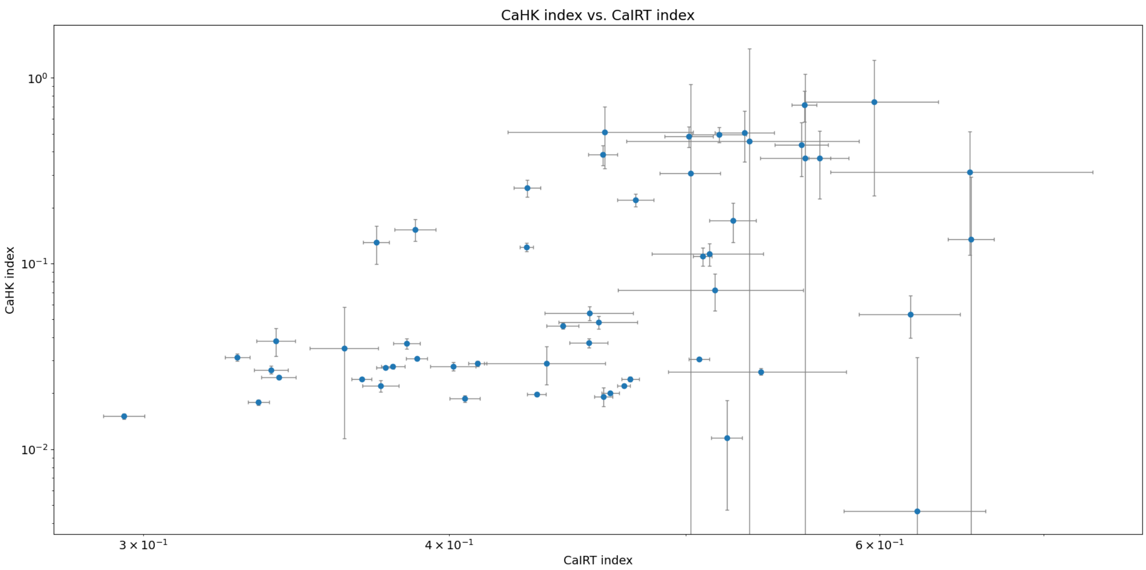

Figure 3.4: CaHK index versus CaIRT index, both axes logarithmic. Each dot is the median for one star, error bars indicate standard deviation.

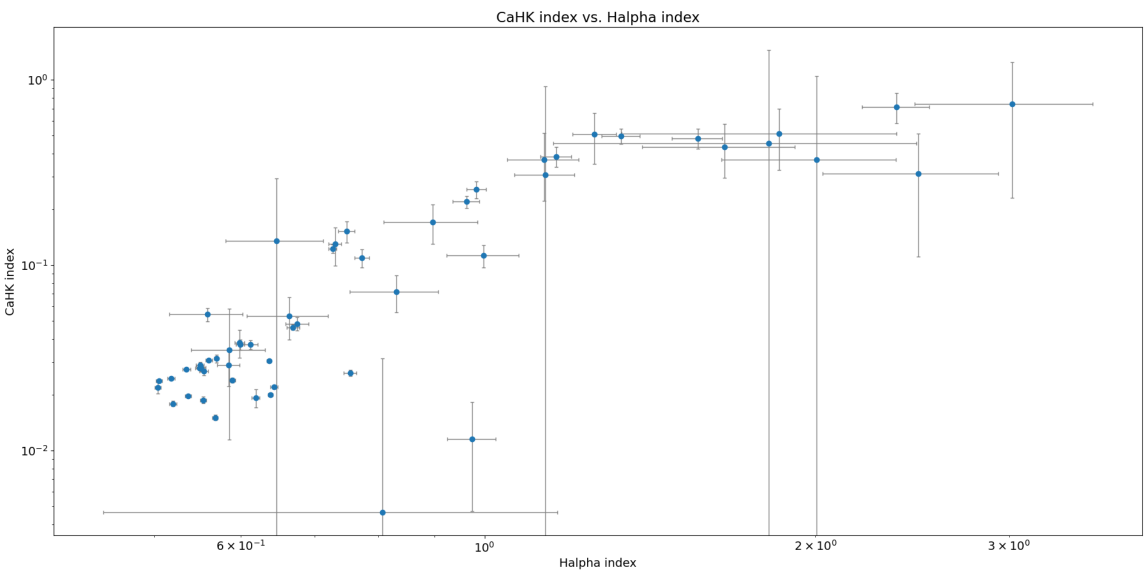

Figure 3.5: CaHK index versus Hα index, both axes logarithmic. Each dot is the median for one star, error bars indicate standard deviation.

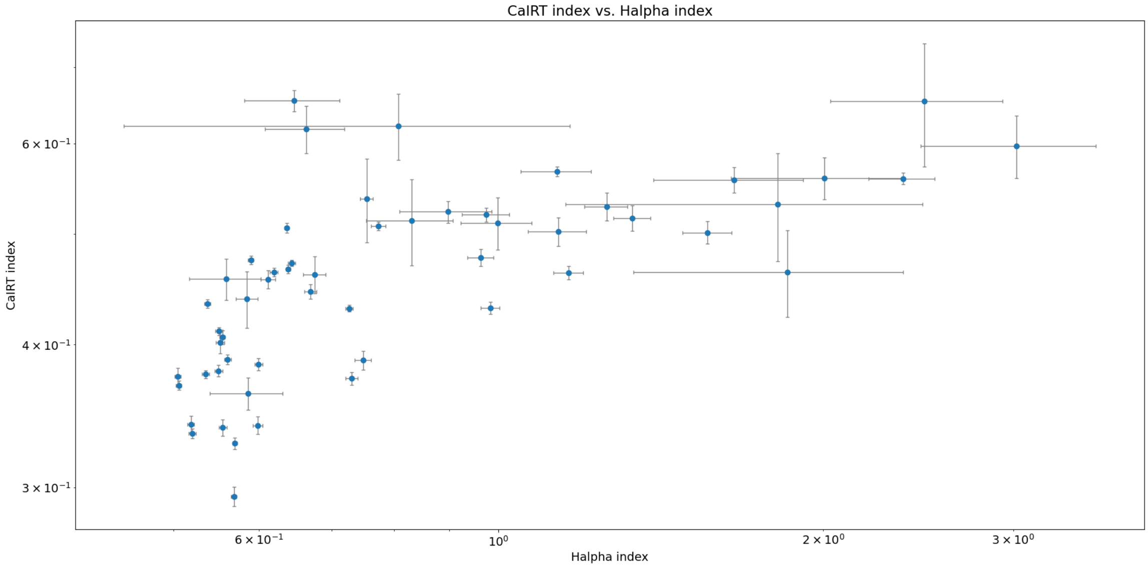

Figure 3.6: CaIRT index versus Hα index, both axes logarithmic. Each dot is the median for one star, error bars indicate standard deviation.

3.2.2 Hα width

Another limitation might occur in the Hα index: The rectangular passband ($F_{H\alpha}$ in equation 3.4) used to measure the flux in the Hα line as defined by Marsden et al. (2014) is 3.6 Å wide. However, in some extreme cases (as shown in section 3.4) where Hα is found in emission, the line profile might be significantly wider - as much as 15 Å - than the passband. The result of this is that the flux in that line is only partially captured by this method. Care should thus be taken when comparing the results from stars with such broad emissions, as the Hα index from Marsden et al. (2014) might not be applicable to those cases.

3.3 Stellar activity across stellar parameters

Figures 3.7, 3.8 and 3.9 show the measured values for the activity indicators (CaIRT, Hα, CaHK index) defined in section 3.1 against different stellar parameters. Figure 3.7 shows effective temperature (T_{eff}), Figure 3.8 surface gravity ($\log g$) and Figure 3.9 metallicity [Fe/H]. Each value is the median of all available measurements for each star, the error bars are the standard deviation. The values for T_{eff}, $\log g$ and [Fe/H] are taken from SIMBAD[3]. A data point is only plotted if a value for that parameter was available on SIMBAD.[4]

When plotted against T_{eff} (Figure 3.7), the measured activity indicators are at a somewhat constant relatively low level for temperatures above ∼4300 K. This is least pronounced for the CaIRT index (top panel), where the spread is higher. Especially the Hα index (middle panel) has a considerable spread at temperatures below ∼3500 K, which can be explained

[3] `https://simbad.u-strasbg.fr/simbad/`

[4] The full list of stars used to create these plots is given in the appendix in Table A.1.

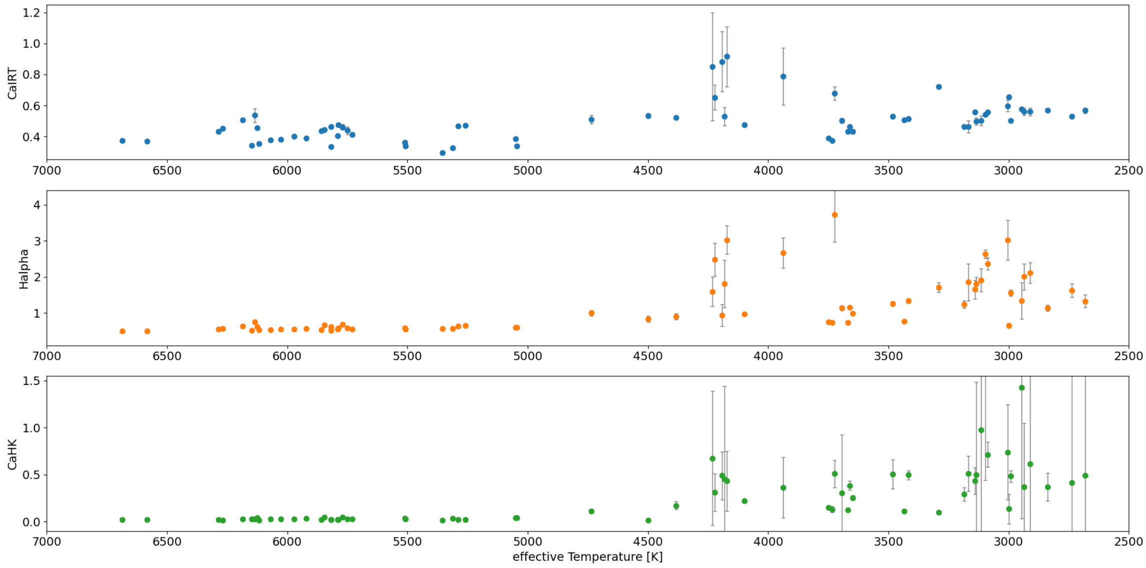

Figure 3.7: CaIRT index (top), Hα index (middle) and CaHK index (bottom) versus stellar effective temperature. Values for T_{eff} were taken from SIMBAD.

by the more prevalent emission case in this temperature range (see below). The CaHK index (bottom panel) starts to become a lot more uncertain below this temperature, which is likely because stars with T_{eff} below ∼3500 K have very little flux in the wavelengths where the CaHK index is measured, resulting in poor signal to noise ratios. In general, it can be stated that activity levels seem to increase below ∼4300 K. However, as discussed by Boro Saikia et al. (2018), this trend is likely an effect of different convection zone depths and is not seen any more when the values are corrected using the bolometric flux of these stars. It is planned to include these bolometric corrections in future work, in order to get a more realistic picture of the relation between activity indicators and stellar properties.

Since this analysis was focused on main-sequence stars, values for $\log g$ will inherently be clustered around 4-4.5, as seen in Figure 3.8. No clear trend of activity indicators could be observed in these plots, with the exception of the CaIRT index (top panel), where values seem to increase somewhat for both higher and lower values of $\log g$, but the amount of data and the uncertainties in some of these values make it difficult to determine any trends. Metallicities (Figure 3.9) of the analysed stars mostly range between −0.2-0.3. The Hα index (middle panel) shows some indication of an upwards trend towards lower metallicities. The CaHK index (bottom panel) has no clear trend here, almost all higher values are subject to large error bars. The CaIRT index (top panel) has a larger spread and again a minor increase towards lower metallicities, but no significant trend.

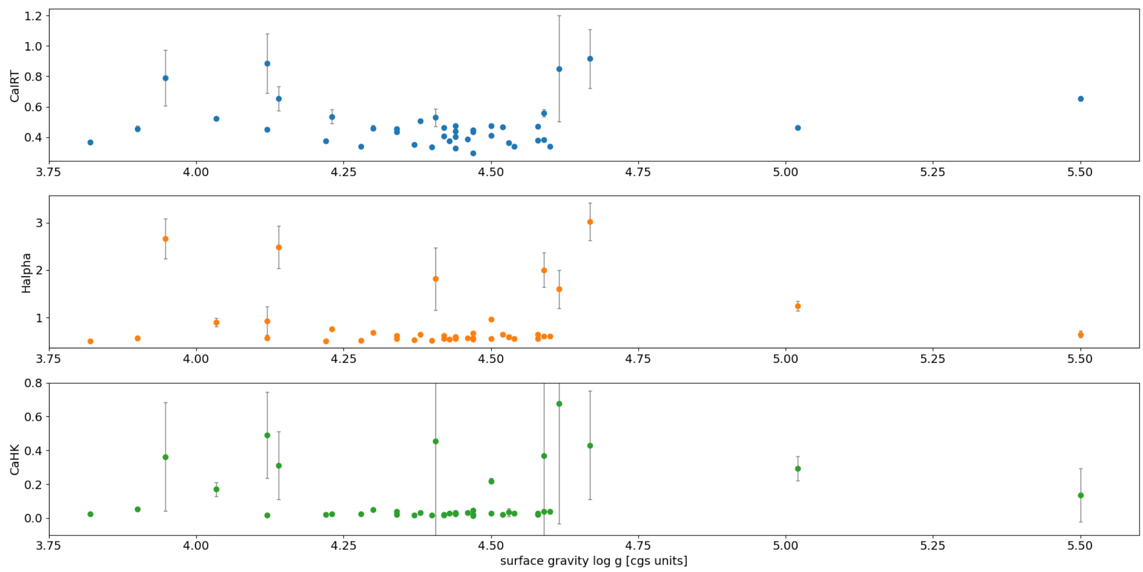

Figure 3.8: CaIRT index (top), Hα index (middle) and CaHK index (bottom) versus surface gravity. Values for $\log g$ were taken from SIMBAD.

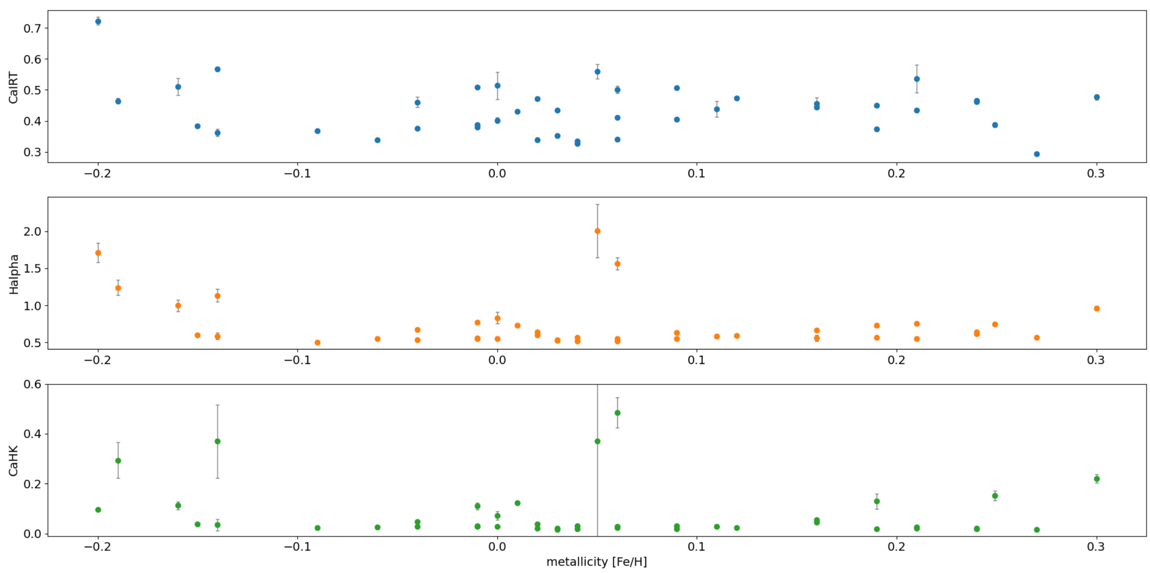

Figure 3.9: CaIRT index (top), Hα index (middle) and CaHK index (bottom) versus metallicity [Fe/H]. Values for [Fe/H] were taken from SIMBAD.

3.4 Emission in cool stars

While non-thermal chromospheric emissions are found (e.g. in the Ca II H&K lines) in spectral types as early as F and are indeed the basis for activity indicators such as the CaHK index, a more interesting case exists for some stars, mostly of K and M type. Commonly denoted with an *e* suffix in the spectral type, these stars show prominent emission profiles for lines such as Hα , the Ca II IR Triplet and Ca II H&K . The following subsections will briefly discuss six individual stars as case studies to show different lines shapes and intensities of emission in cool stars of various stellar parameters and properties. A summary of these values is given in table 3.1.

3.4.1 V2129 Oph

A young (1.5 ± 0.5 Myr) classical T Tauri star (CTTS), *V2129 Oph* has a mass of around $1.35 \pm 0.15\,M_\odot$ and a rotation period of 6.53 d. The star has been classified as spectral type K5 and is part of a binary system (the secondary is not considered significant for the scope of this analysis), and is still accreting mass from a circumstellar disk (Sousa et al., 2021). The effective temperature has been estimated by Franchini et al. (2014) at 4223 ± 40 K and the surface gravity at $\log g = 4.14 \pm 0.06$. The rotational velocity is $v \sin i = 14.9 \pm 0.9\,\mathrm{km\,s^{-1}}$ (Torres et al., 2006) and the stellar radius approximately $2.6\,R_\odot$ (Strassmeier, 2009).

The shape of the Hα line (Figure 3.10, top left panel) is similar to the previously discussed star, although both the "shoulders" and the asymmetry are not as pronounced, but still visible. The drop on the blue side is only present in some observations (e.g. in the green spectrum in the plot), and the maximum height is a lot more variable at around $4 - 7I_c$. In some cases (e.g. the red spectrum in the plot), the shape of the emission line is almost a pure Gaussian. The Ca II H&K lines (Figure 3.10, bottom left panel) are also in emission, up to $5I_c$ and featuring a slight double peak. The plot only shows one spectrum for this wavelength range, because the other two spectra used in this Figure had excessive noise in this region. The lines of the Ca II IR Triplet (middle left panel in Figure 3.10) are quite interesting in this case: Emission is clearly present, but confined to a very sharp line in the core, while the broader absorption profile is clearly visible around the emission feature. The height of the emission peak is subject to some variability, while the absorption profile stays remarkably constant. This appears consistent with the work of Sousa et al. (2021), who also reported variability of various emission lines, which they contribute to magnetically controlled accretion.

3.4.2 DG Tau

DG Tau is a T-Tauri Young Stellar Object (YSO) of spectral type K5-M0 with a mass of $0.67\,M_\odot$, a luminosity of $1.7\,L_\odot$, an estimated effective temperature of 2790 ± 250 K (Kruger et al., 2011) and an age of ∼0.3 Myr (Kitamura et al., 1996; Güdel et al., 2007), which has an energetic jet (Lavalley et al., 1997; Lynch et al., 2013) and a protoplanetary disk with radial disk wind (Güdel et al., 2018).

The Hα line (Figure 3.10, top right panel) shows a very broad (FWHM of 5-10 Å) emission profile with strongly asymmetric "shoulders" and a distinct sharp drop close to the center

Name	Type	Mass $[M_\odot]$	Radius $[R_\odot]$	Age [Myr]	T_{eff} [K]	$\log g$	$v \sin i$ $[\text{km s}^{-1}]$	rot.period [d]	references
V2129 Oph	T Tauri, K5	1.35 ± 0.15	2.6	1.5 ± 0.5	4223 ± 40	4.14 ± 0.06	14.9 ± 0.9	6.53	6, 9, 13, 23
*DG Tau	T Tauri, K7	0.67	–	~0.3	2790 ± 250	–	24.7 ± 0.7	–	1, 7, 11
BP Tau	T Tauri, K7	~0.8	2	6 ± 4	4060 ± 100	4.5 ± 0.2	10.2 ± 1.8	–	2, 3
HK Aqr	BY Dra, M0Ve	0.57 ± 0.07	0.53 ± 0.04	40 ± 5	3800	–	70.4 ± 0.1	0.431	4, 14, 17
Ross 154	EV, M3.5Ve	0.177 ± 0.004	0.200 ± 0.008	<1000	3248 ± 67	5.22 ± 0.12	3.5 ± 0.5	–	8, 22, 25
YZ CMi	EV, M4.5Ve	0.308	0.324	–	2945	4.28	6.6 ± 0.8	2.8	18, 19
*HD 189733	BY Dra, K2V	0.82	0.76	~4300	5050 ± 50	4.56 ± 0.03	vsini	rotperi	21, 16
*HN Peg	G0V	1.085 ± 0.091	1.002 ± 0.018	257 ± 46	5961	4.45	12.81	4.6	12, 5, 20, 15, 10
*†V471 Tau	K2V+WD	0.864 ± 0.029	0.816 ± 0.042	–	4980 ± 10	4.50 ± 0.05	91 ± 1	0.5212	24, 26

Table 3.1: Collection of parameters of the stars discussed in sections 3.4 and 3.5.
References: 1...Kitamura et al. (1996), 2...Simon et al. (2000), 3...Errico et al. (2001), 4...Barnes et al. (2004), 5...Valenti and Fischer (2005), 6...Torres et al. (2006), 7...Güdel et al. (2007), 8...Wargelin et al. (2008), 9...Strassmeier (2009), 10...Martínez-Arnáiz et al. (2010), 11...Kruger et al. (2011), 12...Eisenbeiss et al. (2013), 13...Franchini et al. (2014), 14...Gaidos et al. (2014), 15...Boro Saikia et al. (2015), 16...Boyajian et al. (2015), 17...Elliott et al. (2016), 18...Newton et al. (2017), 19...Fouqué et al. (2018), 20...Giribaldi et al. (2019), 21...Route (2019), 22...Kuznetsov et al. (2019), 23...Sousa et al. (2021), 24...Kővári et al. (2021), 25...Pineda et al. (2021), 26...Muirhead et al. (2022)
*...periodogram analysis performed
†...numbers for mass, radius etc. in this case are given for the primary K2V component of the binary; rotation period is the orbital period of the binary

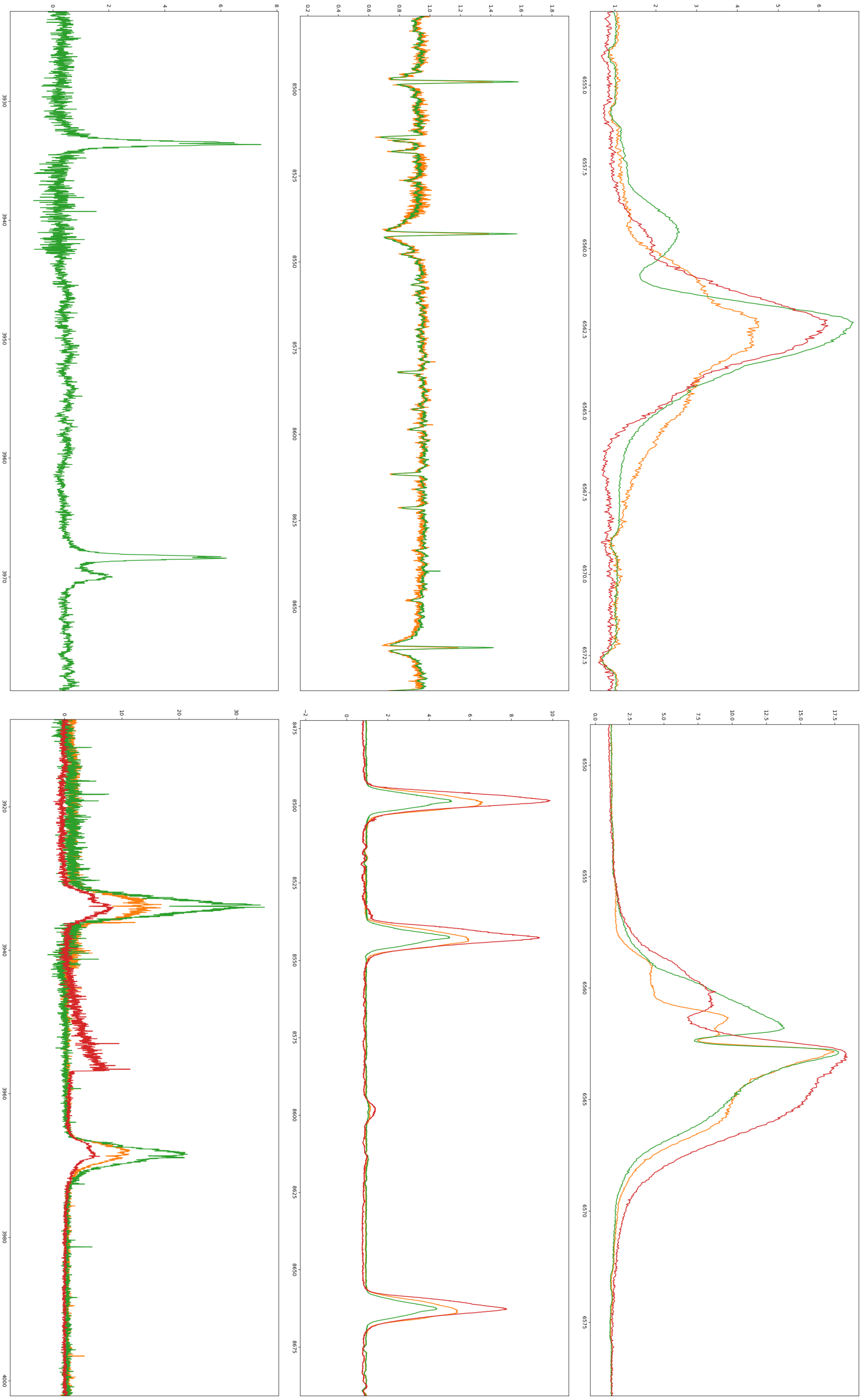

Figure 3.10: Left: V2129 Oph, right: DG Tau. Top: Hα, middle: Ca Triplet, bottom: Ca H&K.

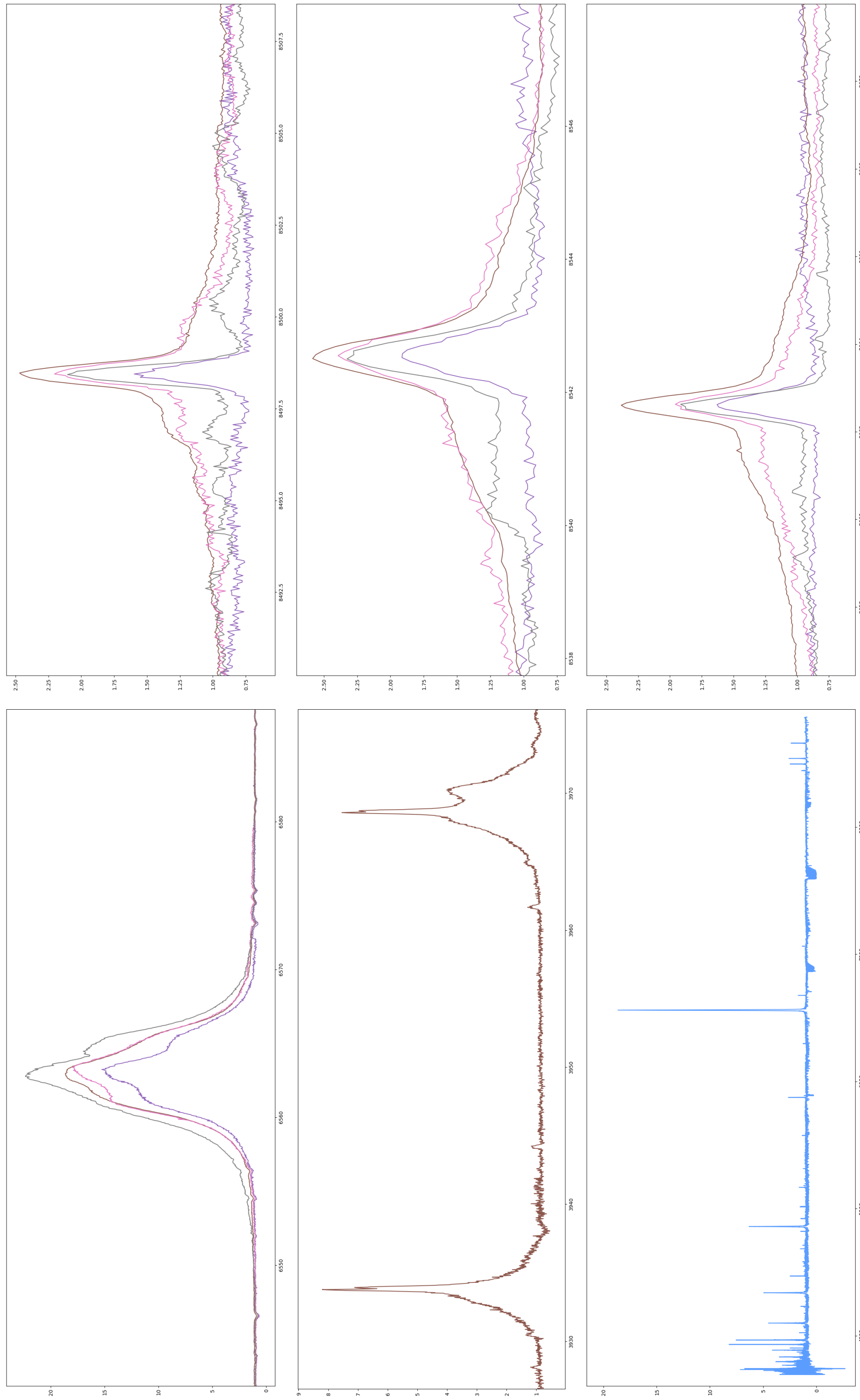

Figure 3.11: BP Tau. Top left: Hα, middle: Ca H&K, bottom left: total normalised spectrum, Right, top to bottom: Ca Triplet.

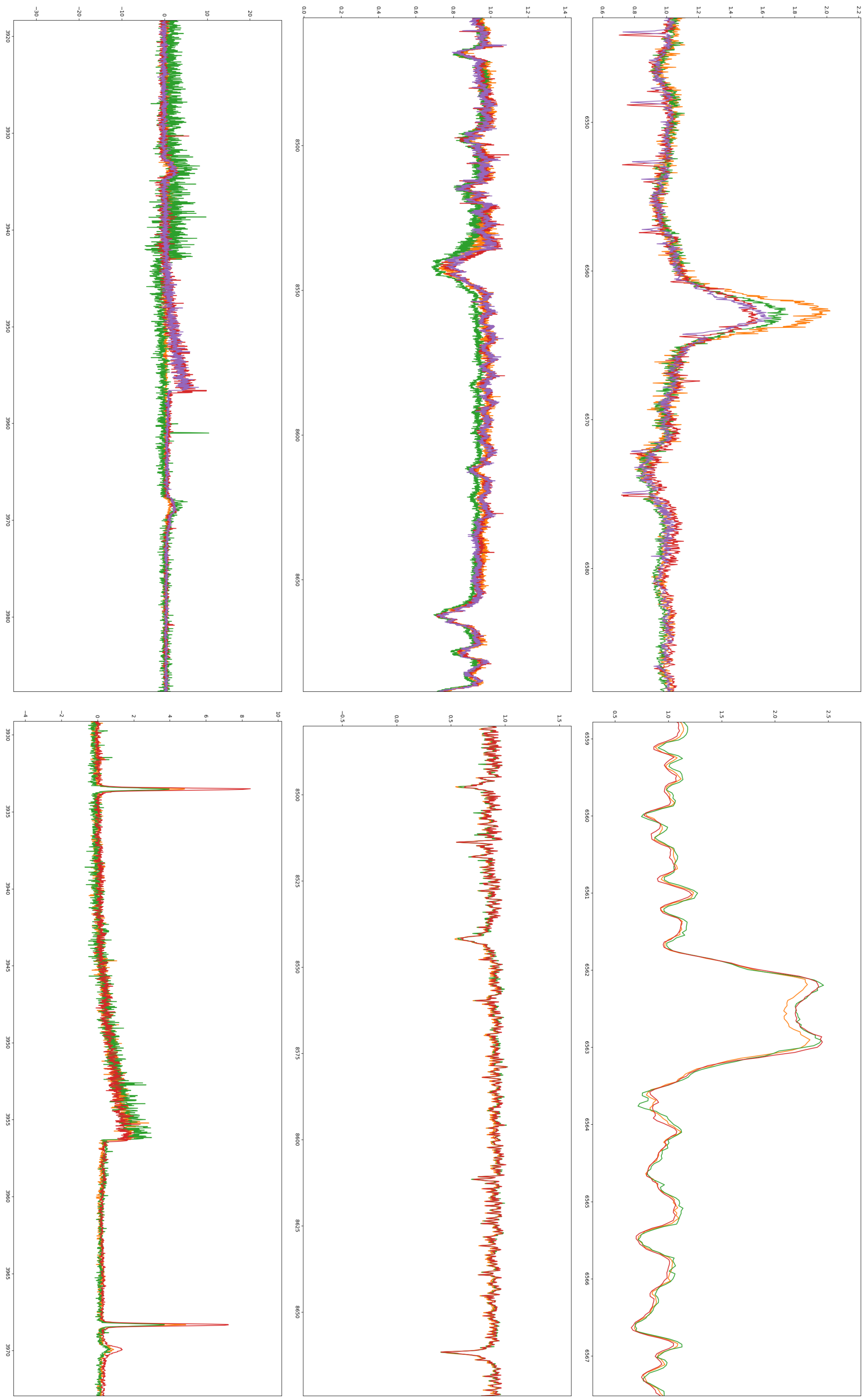

Figure 3.12: Left: HK Aqr, right: Ross 154. Top: Hα, middle: Ca Triplet, bottom: Ca H&K.

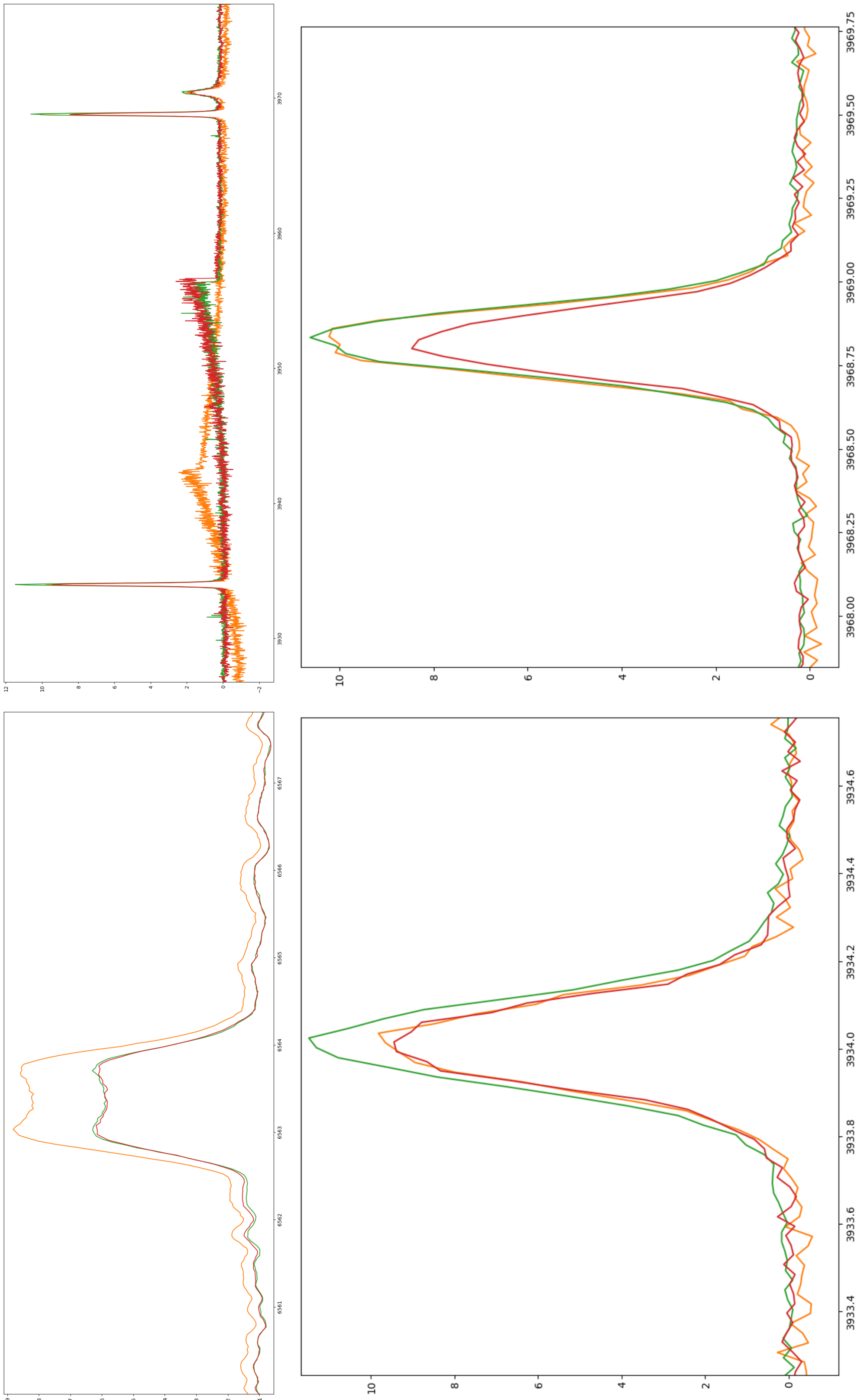

Figure 3.13: YZ CMi. Top left: Hα, top right: Ca H&K, bottom: Ca H&K individual.

on the blue side. The exact location of this drop relative to the line center varies between different observation epochs, as does the strength of the "shoulders" (compare the green and orange spectra in the plot). The observed asymmetry in these shoulders may appear to be somewhat correlated to the location of the drop (as seen in the red spectrum of the plot). The unusual shape of the Hα line might be related to the jets and outflows and/or disk interactions that have been observed. The maximum height of the emission line stays relatively constant at around $17.5 I_c$ throughout different observations, which is intriguing given the significant change of the overall line profile. The Ca II H&K lines (bottom panel of Figure 3.10) also have strong emission and a much higher degree of variability, going between 5 and $30 I_c$. The lines of the Ca II IR Triplet behave similarly, varying between 5 and $10 I_c$, as seen in the middle panel of Figure 3.10. There also seems to be a slight asymmetry found in these line profiles. Interestingly, there appears to be an exact anti-correlation between the maximum height of the Ca II H&K and Ca II IR Triplet lines in this star, the cause of which remains unknown. This star was also analysed with respect to the time-variability of its activity indicators, see below.

3.4.3 BP Tau

The first of three T Tauri stars analysed in this work, *BP Tau* has a spectral type of K7, effective temperature of 4060 K and rotational velocity of $10.2 \pm 1.8\,\mathrm{km\,s^{-1}}$. The star has a mass of around $0.8\,\mathrm{M_\odot}$, a radius of around $2\,\mathrm{R_\odot}$ and a luminosity of $0.93\,\mathrm{L_\odot}$. An accretion disk has also been observed, which causes a large portion of the star's luminosity, and flares from changing accretion rates have been reported as well (Errico et al., 2001). The star's age is not easy to constrain, but a value of 6 ± 4 Myr was given by Simon et al. (2000). It is also an X-Ray source, the believed origin of which is shocks in accreting material (Schmitt et al., 2005).

The strong emission lines are immediately notable in the spectrum (Figure 3.11, bottom left panel), the most prominent of which is Hα (top left panel), with a height of up to $20 I_c$, the highest relative value in this sample. Hα lines are also very broad, with clearly notable asymmetric "shoulders", possibly caused by accreting material. A variability in Hα intensity of almost a factor of 2 between different observations is also seen. The Ca II H&K lines as well show significant emission of up to $8 I_c$, with slight double peaks and minor asymmetry in the wings, but considerably less variability. The lines of the Ca II IR Triplet are in emission (as seen in all three left panels of Figure 3.11), with a similar strong level of variability with time and come correlation to the variability seen in Hα , although there are irregularities (the maximum emission height is not found in the same observation for the Ca II IR Triplet and Hα , but for the minimum this is the case, and there is a clear trend). The wings of the Ca II IR Triplet lines are also interesting, in some observations showing clear asymmetry with the same trend as found in all other emission lines of this star, while in other observations the line boundaries are much sharper, in some cases even showing a hint of shallow absorption around the core emission.

3.4.4 HK Aqr

HK Aqr is a BY Draconis variable red dwarf of spectral type M0Ve, with a mass of $0.57 \pm 0.07\,\mathrm{M_\odot}$, a radius of $0.53 \pm 0.04\,\mathrm{R_\odot}$ and a luminosity of $0.05\,\mathrm{L_\odot}$ (Gaidos et al., 2014). Its age has been estimated via cluster membership to be around 30-40 Myr by Elliott et al. (2016). The star is fast-rotating with a $v \sin i$ of $70.4 \pm 0.1\,\mathrm{km\,s^{-1}}$ and a rotation period of only $0.431\,22 \pm 0.000\,06\,\mathrm{d}$, as given by Barnes et al. (2004), who also report differential rotation and the presence of large spots. Indication of prominence oscillations were found by Leitzinger et al. (2016). Values for effective stellar temperature of 3800 K and metallicity of [Fe/H] 0.27 were taken from Gaidos et al. (2014).

The data for this particular star available for this work in general is subject to high levels of noise, and some likely pipeline-related issues with continuum normalisation towards the blue end of the spectrum. Hα (Figure 3.12, top left panel) exhibits emission up to about $2I_c$, following an almost Gaussian profile, with some variability across time. The Ca II IR Triplet (middle left panel) is in absorption, with no notable irregularities. The Ca II H&K lines (bottom left panel) in these spectra are barely visible, showing minor emission levels.

3.4.5 Ross 154

Ross 154 is an eruptive variable X-ray flare star with a relatively strong magnetic field of $2.6 \pm 0.3\,\mathrm{kG}$ (Wargelin et al., 2008). At a mass of $0.177 \pm 0.004\,\mathrm{M_\odot}$ and a radius of $0.200 \pm 0.008\,\mathrm{R_\odot}$, it is a otherwise a typical red dwarf of spectral type M3.5Ve, with an effective temperature of $3248 \pm 67\,\mathrm{K}$ (Pineda et al., 2021). It is possibly the oldest star of the small sample discussed here, and the upper limit of its age was given by Wargelin et al. (2008) at 1 Gyr, who also state the rotational velocity $v \sin i$ at $3.5 \pm 0.5\,\mathrm{km\,s^{-1}}$. Values for surface gravity $\log g = 5.22 \pm 0.12$ and metallicity [Fe/H] -0.20 ± 0.36 were used from Kuznetsov et al. (2019).

Again, the noise level is above average compared to the rest of the dataset used in this work, and there are some normalisation problems at shorter wavelengths. Hα (Figure 3.12, top right panel) shows a broad emission profile up to around $2I_c$ and also has a pronounced double peak, but only minor variability. While the Ca II IR Triplet is only in minor absorption, the Ca II H&K lines show significant emission up to $4 - 8I_c$ between different observations (middle and bottom left panels of Figure 3.12, respectively).

3.4.6 YZ CMi

Similar to the previous star, *YZ CMi* is a M4.5Ve type eruptive variable, but with a somewhat higher mass and radius of $0.308\,\mathrm{M_\odot}$ and $0.324\,\mathrm{R_\odot}$ and a rotation period of 2.8 d (Newton et al., 2017). Fouqué et al. (2018) mention a rotational velocity of $v \sin i = 6.6 \pm 0.8\,\mathrm{km\,s^{-1}}$ and the values for $T_{\mathrm{eff}}$and $\log g$of 2945 K and 4.28 respectively were taken from Hojjatpanah et al. (2019). It is a white-light and Hα flare star according to Maehara et al. (2021), and Bondar and Katsova (2018) reported a long-period activity cycle of 27.5 yr.

Hα (top left panel in Figure 3.13) is found in the form of a broad double-peaked emission line, with a variable maximum height of around $6 - 9I_c$ for different observations. The Ca II H&K lines (all other panels) also show strong emission up to about $10I_c$. Interestingly, the variability there is not consistent with either the variability of Hα nor between the two lines themselves. The Ca II IR Triplet is not really usable for this star (with the available data) due to high levels of noise, but there is some indication of emission in the cores.

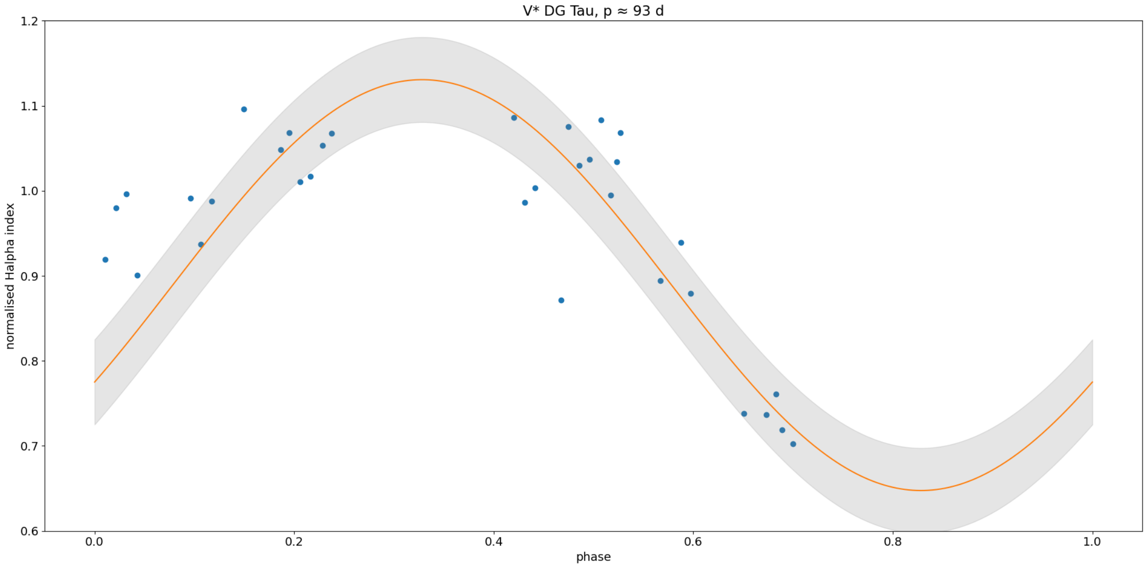

Figure 3.14: DG Tau. Phased plot of the Hα index.

3.5 Time-analysis of measured activity

The spectra of 45 stars where a large quantity of high-quality data was available were also analysed with respect to temporal variability of their activity indicators. Four of these stars with particularly interesting results are presented below. The time series analysis was performed using the `LombScargle` class[5] available as part of the `astropy.timeseries` subpackage, which provides Lomb-Scargle periodograms for the detection of periodic signals. Frequency and power of each dataset were acquired with the `LombScargle.autopower` method and the fitted sinusoidal lines in the plots below were calculated using the `LombScargle.model` method. The plots in this section all show a phased plot of the individual measurements for each index, normalised to the median of each data set. To avoid filling this section with too many plots, those marked with an "A" prefix are found in the appendix.

3.5.1 DG Tau

A general description of this young K7 star is provided above in section 3.4.2. The time series analysis for the Hα and CaIRT indices both yield a best-fit period of around 93 d. As can be seen in the phased plots in Figure 3.14, the Hα index has a much higher amplitude of more than ±10%, while the CaIRT (Figure A.2) index varies only by around ±8%, which is still a lot compared to some other variabilities discussed below. Interestingly, the two indices appear to be out of phase by almost 180°, meaning the maximum of the Hα index occurs roughly at the minimum of the CaIRT index. This is somewhat consistent with the change in the emission profiles discussed in section 3.4.2 and might be a result of these two lines being produced in different part of the stellar chromosphere. The CaHK index is shown in Figure A.3, with a best-fit frequency of 121 d, but again the trend is not very clear.

[5] `https://docs.astropy.org/en/stable/timeseries/lombscargle.html`

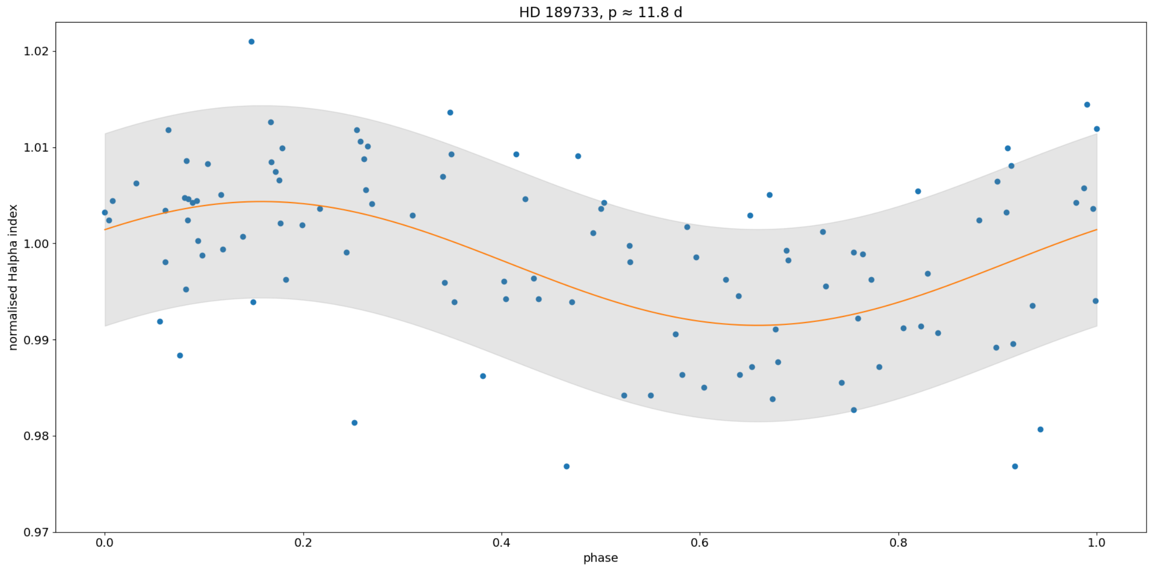

Figure 3.15: HD 189733. Phased plot of the Hα index.

3.5.2 HD 189733

The hot-Jupiter-hosting K2V star *HD 189733* has a mass of $0.82\,\mathrm{M}_\odot$, a radius of $0.76\,\mathrm{R}_\odot$, $T_{\mathrm{eff}} = 5050 \pm 50\,\mathrm{K}$ and a metallicity of $[\mathrm{Fe/H}] = -0.03 \pm 0.04$ (Route, 2019). It has received much attention for its suspected *star-planet-interaction* (e.g. Cauley et al., 2018), but the existence of such interactions has been doubted in more recent studies (e.g. Route, 2019). It is also known as a BY Dra type variable with a period of 12.04 d (Samus' et al., 2017).

A period very close to this value could be observed in the Hα index (Figure 3.15) at 11.8 d. However, this value was only seen after extending the Nyquist range to 5, meaning the frequencies found in this way are not very reliable and could be a result of overfitting. For the CaIRT index (Figure 3.16), a period of ∼30 d was the best fit at a Nyquist factor of 2. Increasing this value did not result in any significant peaks in the power-frequency plot in the ∼12 d range. In the case of the CaHK index (Figure A.1), the best-fit frequency was much higher at 534 d and not a clear detection at all, with quite large spread in the data points. Increasing the Nyquist range did not yield any other frequency.

3.5.3 HN Peg

HN Peg is a 257±46 Myr old G0V star (Eisenbeiss et al., 2013) and considered a "young solar analogue" (Boro Saikia et al., 2015). A mass of $1.085 \pm 0.091\,\mathrm{M}_\odot$, a radius of $1.002 \pm 0.018\,\mathrm{R}_\odot$ and a luminosity of $0.062 \pm 0.033\,\mathrm{L}_\odot$ are given by Valenti and Fischer (2005), values for T_{eff} of 5961 K, $\log g \approx 4.45$ and a [Fe/H] metallicity of −0.06 were taken from Giribaldi et al. (2019). Boro Saikia et al. (2015) report a rotation period of 4.6 d, and a rotational velocity of $v \sin i \approx 12.81\,\mathrm{km\,s^{-1}}$ was found by Martínez-Arnáiz et al. (2010).

A best-fit frequency for the Hα and CaIRT indices was found at 319 d and 171 d respectively (phased plots are shown in Figures 3.17 and 3.18). The disagreement between these two

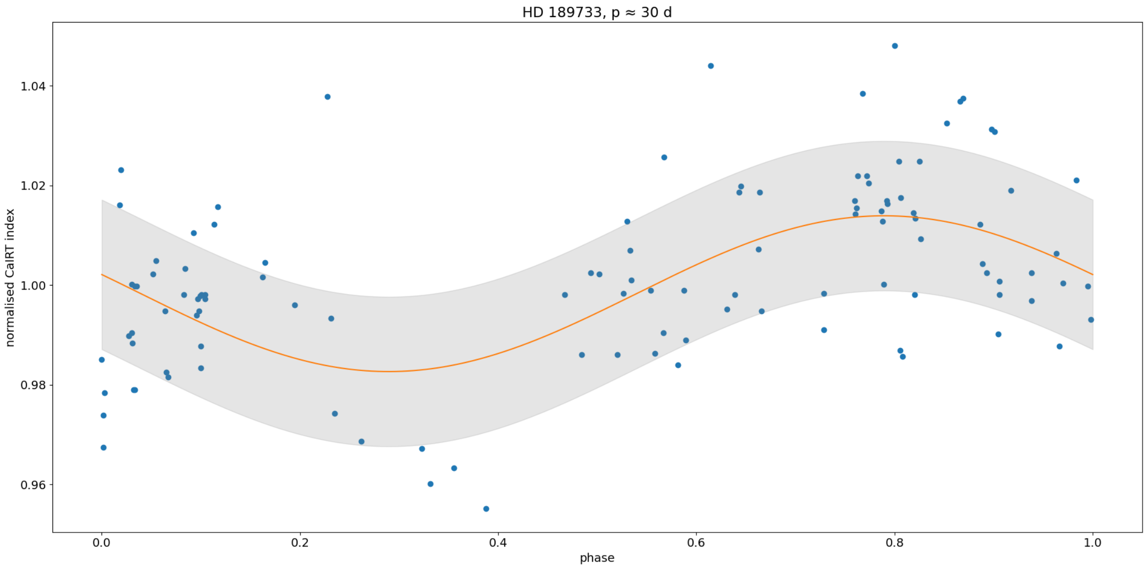

Figure 3.16: HD 189733. Phased plot of the CaIRT index.

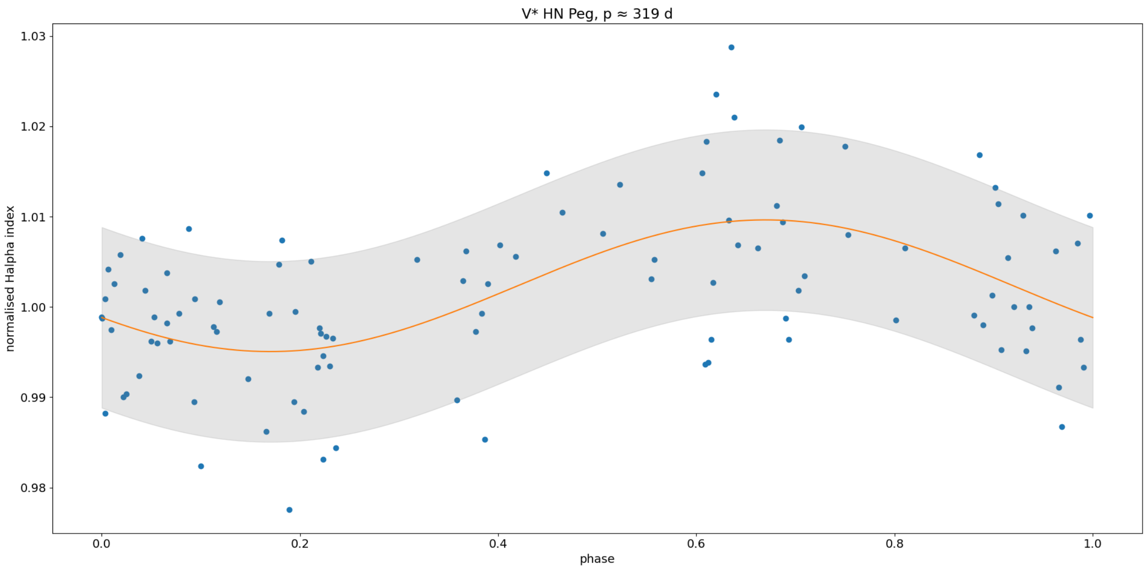

Figure 3.17: HN Peg. Phased plot of the Hα index.

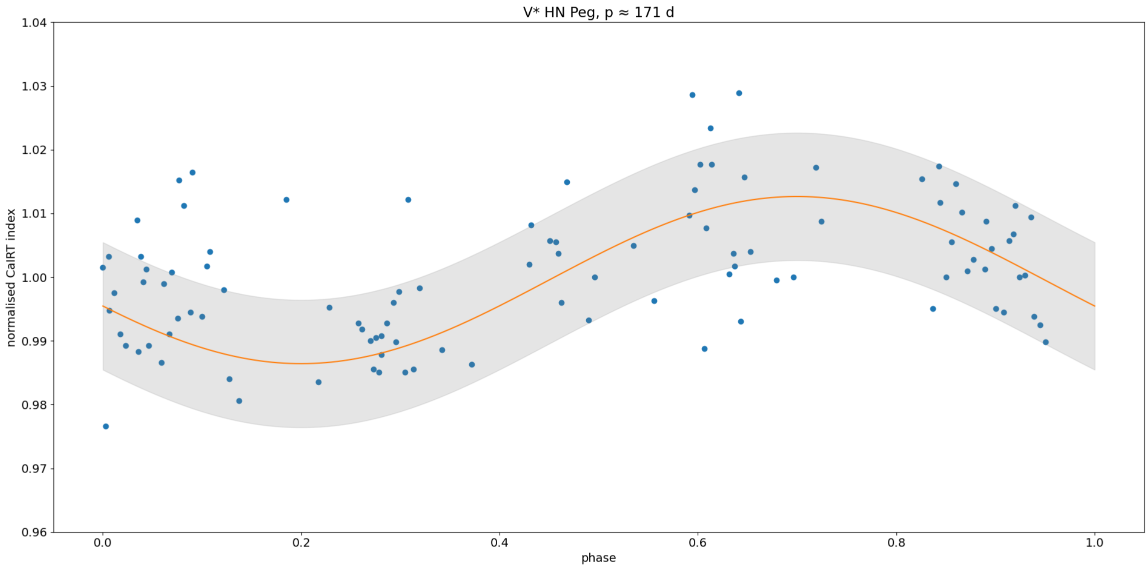

Figure 3.18: HN Peg. Phased plot of the CaIRT index.

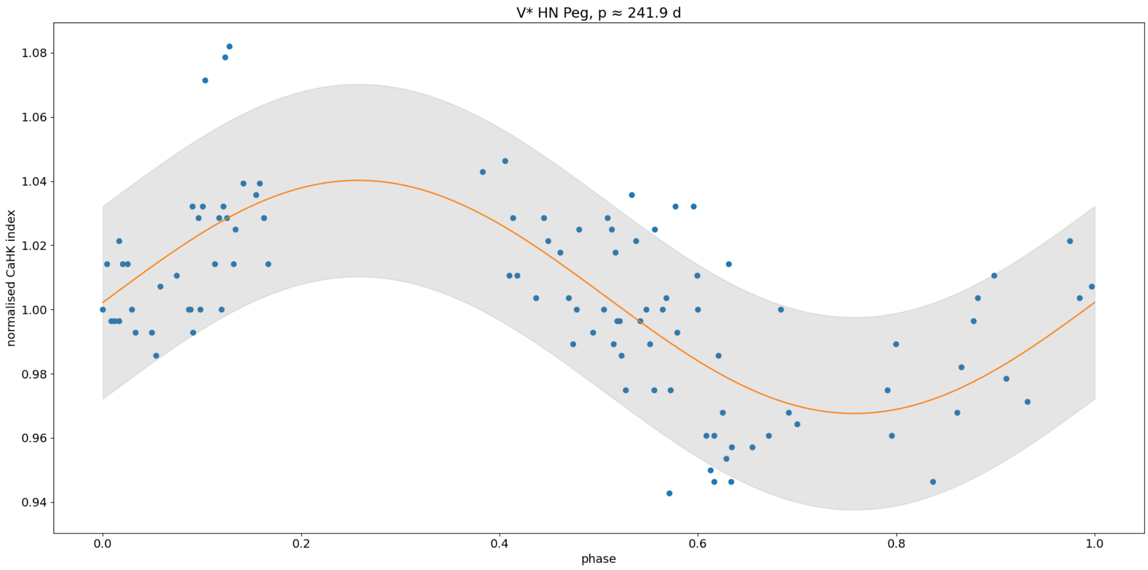

Figure 3.19: HN Peg. Phased plot of the CaHK index.

values is somewhat reduced by the fact that the frequency-power plot for each index shows significant side-peaks at the best-fit frequency of the other index, meaning the discrepancy is likely due to noise in the data and the relatively low number of data points. The amplitude of both variabilities is rather low however, at around ±1% in each case. The CaHK index (Figure 3.19) yields a period of around 242 d, with an amplitude of almost ±4%, and out of phase with the other two indices. The variation of the Ca II H&K lines was also studied by Baliunas et al. (1995), who report a period of 6.2 ± 0.2 yr. This period does correlate to a secondary peak in the power-frequency plot from this work, but more data over a longer time span would be needed to make a better assessment of this periodicity.

3.5.4 V471 Tau

The post-common-envelope (Muirhead et al., 2022) pre-cataclysmic (Kővári et al., 2021) binary *V471 Tau* consists of a $0.864 \pm 0.029\,M_\odot$ K2V main-sequence star and a $0.797 \pm 0.016\,M_\odot$ white dwarf in a 0.015 au orbit (Kővári et al., 2021; Muirhead et al., 2022). The K2 component has a luminosity of $0.41 \pm 0.01\,L_\odot$, a radius of $0.816 \pm 0.042\,R_\odot$ and a $v \sin i$of $91 \pm 1\,\mathrm{km\,s^{-1}}$. Furthermore, the effective temperature is 4980 ± 10 K and the surface gravity $\log g = 4.50 \pm 0.05$. (Kővári et al., 2021; Muirhead et al., 2022). The presence of the close white dwarf was shown to have an influence on the magnetic activity of the main-sequence star (Kővári et al., 2021) and activity is increased on the side pointed towards the white dwarf (Zaire et al., 2021). The K2V star is also subject to differential rotation, that possibly varies with time and the magnetic interaction might be the cause for observed eclipse timing variations (Zaire et al., 2021).

Regarding the time-series analysis of the measured activity indicators in this work, no satisfying result could be achieved with the single-frequency fit used in all previous cases, for both the Hα and CaIRT indices. However, after extending the fit to $n = 2$ Fourier terms, a much better fit could be produced. Especially the CaIRT index (phased plot in Figure A.4) shows good agreement with a 2-frequency fit, the main period of which was found to be 0.35 d. The Hα index (Figure A.5) yields a similar period at 0.3 d, although there appear to be additional signals in the data which are not well covered by the fit (see left half of the plot). The CaHK index is sufficiently fitted using a single Fourier term (Figure A.6), a 2-term-fit (Figure A.7) does not produce much improvement. The best-fit period is 0.52 d, which matches remarkably well with the 0.5212 d orbital period found by Muirhead et al. (2022). A higher number of Fourier terms did not result in any improvement in either case.

Since the best-fit periods for the Hα and CaIRT indices do not match the orbital period of the binary system (although at the same order of magnitude), the phased plots for these indices were re-done using the 0.5212 d orbital period (Muirhead et al., 2022) and $n = 2$ Fourier terms, the results can be seen in Figures A.8 and A.9, respectively. The fit in this case is quite good, especially for the CaIRT index (Figure A.9). The orbital frequency was also present as a significant secondary peak in the power-frequency diagrams. More data points for this system, especially spread over a longer time interval, might yield a best-fit frequency closer to the orbital frequency.

Chapter 4

Discussion & Conclusion

The first primary objective of this work was to port the core functionality of `binmag` (Kochukhov, 2007, 2018) from *IDL* to *Python*. The result of this development process is SynSPEC.py, a GUI-based *Python* application which fulfills the functional requirements and works cross-platform, without depending on non-freely available software. The capabilities implemented in the current version include: The ability to visualise synthetic spectra including annotating spectral lines and the relative contribution of multiple lines to a blended spectral line, manipulating parameters of a synthetic spectrum such as $v \sin i$, $v_{mac}$and v_{rad}, and calculating line equivalent widths. Furthermore, SynSPEC.py is able to load and visualise observed spectra in a wide variety of common formats. While not (currently) supporting all functions of `binmag`, some new functions were added, primarily aimed at measuring activity indicators in stellar spectra. These measurements were then applied to a sample of cool main-sequence stars with varying levels of chromospheric activity, mostly of spectral types G, K and M, some of which feature strong non-thermal chromospheric emission.

The estimation of stellar activity indicators with SynSPEC.py worked reasonably well, given that it is a newly developed software and especially the parts of the code used for these measurements were implemented from scratch. The application is running mostly stable now, but would still benefit from more future testing of the software in general and verification of the results yielded by the measurements. It is planned to continue this development process of SynSPEC.py and to add more functionality and improve the existing one in the future. Additionally, the creation and distribution of a stable standalone executable version of the software is planned.

For the application of measuring stellar activity indicators, the performance of the software was mostly satisfying. The obstacles encountered during the data analysis were mostly caused by limitations in the quality of the data itself, as well as the pre-treatment that came with the data, like continuum normalisation and radial velocity correction, both of which were sub-optimal in some cases. One significant issue was the implementation of the S-index, which was partially solved by introducing the CaHK index as a proxy, as discussed in detail in section 3.1.2. It is planned to fix this and properly implement S-index measurements as part of future work, while also improving the calibration of the measurements to existing literature values. It might also be worth exploring the application of different activity indicators already in use by other authors, especially to some of the stars analysed in this work which feature particularly strong emission lines (e.g. Hα) that might only be partially captured by some methods like the narrow-band indices used in this work.

The results of the activity analysis were compared to various stellar parameters like effective temperature, surface gravity or metallicity, in order to investigate any possible trends or relations. A small number of individual stars was looked at in more detail, including three young T Tauri type stars, along with main sequence K and M stars. Some of these stars show very strong levels of emission in the Ca II H&K, Hα and Ca II IR Triplet lines, oftentimes with some degree of variability across different observation epochs. In a few cases, where a large number of individual spectra was available, this variability was further investigated by means of Lomb-Scargle periodogram analysis. In two cases, a period was found that matches with values found by other authors, although large uncertainties were involved in each detection.

While some of the stars that were analysed with respect to the time-variability of their activity indicators already show interesting patterns of variability, in many cases this analysis would greatly benefit from an increased amount of data points spread over a large time span, to detect long-period variability, but also clustered in a short interval like a few days with multiple observations per night, to investigate shorter periods. An important consideration in this context is the risk of "over-fitting" a signal, i.e. seemingly detecting a time-variability that does not actually exist, but is a product of noisy data. The estimation of a false-alarm-probability for detected frequencies could be a possible solution to this, as would again be a larger amount of data. With a few exceptions, many of the variabilities detected in this work only show an amplitude of a few percent. Without proper estimation of uncertainties and the aforementioned false-alarm-probability, these frequencies should only be taken as preliminary results for a possible pre-selection of candidate targets for a more detailed investigation of variable activity.

This work has shown that SySPEC.py is a working software tool that can efficiently be used to analyse spectroscopic data, but has also brought the focus to some aspects of this software, that could be improved and their functionality expanded over the course of future development. These include: support for multiple formats of synthetic spectra, combination of two spectra to a composite binary spectrum, full support of spectropolarimetric data, built-in periodogram analysis, direct access to line databases and creation of synthetic spectra directly in the GUI application, automatic estimation/fitting of stellar parameters, re-processing of observed spectra (e.g. continuum re-normalisation) and much more. It is also planned to turn the software into a full *Python* package as well as a standalone GUI executable and distribute it for research and educational purposes.

Bibliography

N. Astudillo-Defru, X. Delfosse, X. Bonfils, T. Forveille, C. Lovis, and J. Rameau. Magnetic activity in the HARPS M dwarf sample. The rotation-activity relationship for very low-mass stars through R'$_{HK}$. A&A, 600:A13, April 2017. doi: 10.1051/0004-6361/201527078.

M. Aurière. Stellar Polarimetry with NARVAL. In J. Arnaud and N. Meunier, editors, *EAS Publications Series*, volume 9 of *EAS Publications Series*, page 105, January 2003.

S. L. Baliunas, R. A. Donahue, W. H. Soon, J. H. Horne, J. Frazer, L. Woodard-Eklund, M. Bradford, L. M. Rao, O. C. Wilson, Q. Zhang, W. Bennett, J. Briggs, S. M. Carroll, D. K. Duncan, D. Figueroa, H. H. Lanning, T. Misch, J. Mueller, R. W. Noyes, D. Poppe, A. C. Porter, C. R. Robinson, J. Russell, J. C. Shelton, T. Soyumer, A. H. Vaughan, and J. H. Whitney. Chromospheric Variations in Main-Sequence Stars. II. ApJ, 438:269, January 1995. doi: 10.1086/175072.

J. R. Barnes, D. J. James, and A. Collier Cameron. Differential rotation and star-spot evolution on HK Aqr in 2001 and 2002. MNRAS, 352(2):589–599, August 2004. doi: 10.1111/j.1365-2966.2004.07949.x.

N. I. Bondar and M. M. Katsova. Long-Term Cyclic Variability of YZ CMi in the Context of Solar and Stellar Physics. *Geomagnetism and Aeronomy*, 58(7):910–915, December 2018. doi: 10.1134/S0016793218070046.

S. Boro Saikia, S. V. Jeffers, P. Petit, S. Marsden, J. Morin, and C. P. Folsom. Variable magnetic field geometry of the young sun HN Pegasi (HD 206860). A&A, 573:A17, January 2015. doi: 10.1051/0004-6361/201424096.

S. Boro Saikia, C. J. Marvin, S. V. Jeffers, A. Reiners, R. Cameron, S. C. Marsden, P. Petit, J. Warnecke, and A. P. Yadav. Chromospheric activity catalogue of 4454 cool stars. Questioning the active branch of stellar activity cycles. A&A, 616:A108, August 2018. doi: 10.1051/0004-6361/201629518.

Tabetha Boyajian, Kaspar von Braun, Gregory A. Feiden, Daniel Huber, Sarbani Basu, Pierre Demarque, Debra A. Fischer, Gail Schaefer, Andrew W. Mann, Timothy R. White, Vicente Maestro, John Brewer, C. Brooke Lamell, Federico Spada, Mercedes López-Morales, Michael Ireland, Chris Farrington, Gerard T. van Belle, Stephen R. Kane, Jeremy Jones, Theo A. ten Brummelaar, David R. Ciardi, Harold A. McAlister, Stephen Ridgway, P. J. Goldfinger, Nils H. Turner, and Laszlo Sturmann. Stellar diameters and temperatures - VI. High angular resolution measurements of the transiting exoplanet host stars

HD 189733 and HD 209458 and implications for models of cool dwarfs. MNRAS, 447(1): 846–857, February 2015. doi: 10.1093/mnras/stu2502.

P. Wilson Cauley, Evgenya L. Shkolnik, Joe Llama, Vincent Bourrier, and Claire Moutou. Evidence of Magnetic Star-Planet Interactions in the HD 189733 System from Orbitally Phased Ca II K Variations. AJ, 156(6):262, December 2018. doi: 10.3847/1538-3881/aae841.

C. Cincunegui, R. F. Díaz, and P. J. D. Mauas. Hα and the Ca II H and K lines as activity proxies for late-type stars. A&A, 469(1):309–317, July 2007. doi: 10.1051/0004-6361:20066503.

J. F. Donati, C. Catala, J. D. Landstreet, and P. Petit. ESPaDOnS: The New Generation Stellar Spectro-Polarimeter. Performances and First Results. In R. Casini and B. W. Lites, editors, *Solar Polarization 4*, volume 358 of *Astronomical Society of the Pacific Conference Series*, page 362, December 2006.

Douglas K. Duncan, Arthur H. Vaughan, Olin C. Wilson, George W. Preston, James Frazer, Howard Lanning, Anthony Misch, Jean Mueller, David Soyumer, L. Woodard, Sallie L. Baliunas, Robert W. Noyes, Lee W. Hartmann, Alain Porter, Cornelis Zwaan, Frans Middelkoop, Rene G. M. Rutten, and Dimitri Mihalas. CA II H and K Measurements Made at Mount Wilson Observatory, 1966–1983. ApJS, 76:383, May 1991. doi: 10.1086/191572.

T. Eisenbeiss, M. Ammler-von Eiff, T. Roell, M. Mugrauer, Ch. Adam, R. Neuhäuser, T. O. B. Schmidt, and A. Bedalov. The Hercules-Lyra association revisited. New age estimation and multiplicity study. A&A, 556:A53, August 2013. doi: 10.1051/0004-6361/201118362.

P. Elliott, A. Bayo, C. H. F. Melo, C. A. O. Torres, M. F. Sterzik, G. R. Quast, D. Montes, and R. Brahm. Search for associations containing young stars (SACY). VII. New stellar and substellar candidate members in the young associations. A&A, 590:A13, May 2016. doi: 10.1051/0004-6361/201628253.

L. Errico, S. A. Lamzin, and A. A. Vittone. UV spectra of T Tauri stars from the HST and IUE satellites: BP Tau. A&A, 377:557–565, October 2001. doi: 10.1051/0004-6361:20011108.

Pascal Fouqué, Claire Moutou, Lison Malo, Eder Martioli, Olivia Lim, Arvind Rajpurohit, Etienne Artigau, Xavier Delfosse, Jean-François Donati, Thierry Forveille, Julien Morin, France Allard, Raphaël Delage, René Doyon, Elodie Hébrard, and Vasco Neves. SPIRou Input Catalogue: global properties of 440 M dwarfs observed with ESPaDOnS at CFHT. MNRAS, 475(2):1960–1986, April 2018. doi: 10.1093/mnras/stx3246.

M. Franchini, C. Morossi, P. di Marcantonio, M. L. Malagnini, and M. Chavez. The FEROS-Lick/SDSS observational data base of spectral indices of FGK stars for stellar population studies. MNRAS, 442(1):220–228, July 2014. doi: 10.1093/mnras/stu873.

E. Gaidos, A. W. Mann, S. Lépine, A. Buccino, D. James, M. Ansdell, R. Petrucci, P. Mauas, and E. J. Hilton. Trumpeting M dwarfs with CONCH-SHELL: a catalogue of nearby cool host-stars for habitable exoplanets and life. MNRAS, 443(3):2561–2578, September 2014. doi: 10.1093/mnras/stu1313.

Mark S. Giampapa, Lawrence E. Cram, and Walter J. Wild. The CA II resonance lines in M dwarf stars without H-alpha emission. ApJ, 345:536–542, October 1989. doi: 10.1086/167928.

R. E. Giribaldi, M. L. Ubaldo-Melo, G. F. Porto de Mello, L. Pasquini, H. G. Ludwig, S. Ulmer-Moll, and D. Lorenzo-Oliveira. Accurate effective temperature from Hα profiles. A&A, 624:A10, April 2019. doi: 10.1051/0004-6361/201833763.

David F. Gray. *The Observation and Analysis of Stellar Photospheres*. Cambridge University Press, 2005.

M. Güdel, A. Telleschi, M. Audard, S. L. Skinner, K. R. Briggs, F. Palla, and C. Dougados. X-rays from jet-driving protostars and T Tauri stars. A&A, 468(2):515–528, June 2007. doi: 10.1051/0004-6361:20065736.

M. Güdel, C. Eibensteiner, O. Dionatos, M. Audard, J. Forbrich, S. Kraus, Ch. Rab, Ch. Schneider, S. Skinner, and E. Vorobyov. ALMA detects a radial disk wind in DG Tauri. A&A, 620:L1, December 2018. doi: 10.1051/0004-6361/201834271.

Manuel Güdel. The Sun in Time: Activity and Environment. *Living Reviews in Solar Physics*, 4(1):3, December 2007. doi: 10.12942/lrsp-2007-3.

S. Hojjatpanah, P. Figueira, N. C. Santos, V. Adibekyan, S. G. Sousa, E. Delgado-Mena, Y. Alibert, S. Cristiani, J. I. González Hernández, A. F. Lanza, P. Di Marcantonio, J. H. C. Martins, G. Micela, P. Molaro, V. Neves, M. Oshagh, F. Pepe, E. Poretti, B. Rojas-Ayala, R. Rebolo, A. Suárez Mascareño, and M. R. Zapatero Osorio. Catalog for the ESPRESSO blind radial velocity exoplanet survey. A&A, 629:A80, September 2019. doi: 10.1051/0004-6361/201834729.

Zs. Kővári, L. Kriskovics, K. Oláh, P. Odert, M. Leitzinger, B. Seli, K. Vida, T. Borkovits, and T. Carroll. A confined dynamo: Magnetic activity of the K-dwarf component in the pre-cataclysmic binary system V471 Tauri. A&A, 650:A158, June 2021. doi: 10.1051/0004-6361/202140707.

Yoshimi Kitamura, Ryohei Kawabe, and Masao Saito. Aperture Synthesis 13CO (J = 1-0) Observations of the Molecular Gas around DG Tauri: Evidence for a Dispersing Gas Disk. ApJ, 457:277, January 1996. doi: 10.1086/176728.

O. Kochukhov. BinMag: Widget for comparing stellar observed with theoretical spectra, May 2018.

O. P. Kochukhov. Spectrum synthesis for magnetic, chemically stratified stellar atmospheres. In I. I. Romanyuk, D. O. Kudryavtsev, O. M. Neizvestnaya, and V. M. Shapoval, editors, *Physics of Magnetic Stars*, pages 109–118, January 2007.

Oleg Kochukhov. Synth3: Non-magnetic spectrum synthesis code, December 2012.

Andrew J. Kruger, Matthew J. Richter, John S. Carr, Joan R. Najita, Greg W. Doppmann, and Andreas Seifahrt. Gas and Dust Toward DG Tau B and VV CrA. ApJ, 729(2):145, March 2011. doi: 10.1088/0004-637X/729/2/145.

F. Kupka, N. Piskunov, T. A. Ryabchikova, H. C. Stempels, and W. W. Weiss. VALD-2: Progress of the Vienna Atomic Line Data Base. A&AS, 138:119–133, July 1999. doi: 10.1051/aas:1999267.

M. K. Kuznetsov, C. del Burgo, Ya. V. Pavlenko, and J. Frith. Characterization of a Sample of Southern M Dwarfs Using Harps and X-shooter Spectra. ApJ, 878(2):134, June 2019. doi: 10.3847/1538-4357/ab1fe9.

C. Lavalley, S. Cabrit, C. Dougados, P. Ferruit, and R. Bacon. Sub-arcsecond morphology and kinematics of the DG Tauri jet in the [O I]λ6300 line. A&A, 327:671–680, November 1997.

M. Leitzinger, P. Odert, T. V. Zaqarashvili, R. Greimel, A. Hanslmeier, and H. Lammer. Indications of stellar prominence oscillations on fast rotating stars: the cases of HK Aqr and PZ Tel. MNRAS, 463(1):965–979, November 2016. doi: 10.1093/mnras/stw1922.

C. Lynch, R. L. Mutel, M. Güdel, T. Ray, S. L. Skinner, P. C. Schneider, and K. G. Gayley. Very Large Array Observations of DG Tau's Radio Jet: A Highly Collimated Thermal Outflow. ApJ, 766(1):53, March 2013. doi: 10.1088/0004-637X/766/1/53.

Hiroyuki Maehara, Yuta Notsu, Kousuke Namekata, Satoshi Honda, Adam F. Kowalski, Noriyuki Katoh, Tomohito Ohshima, Kota Iida, Motoki Oeda, Katsuhiro L. Murata, Masayuki Yamanaka, Kengo Takagi, Mahito Sasada, Hiroshi Akitaya, Kai Ikuta, Soshi Okamoto, Daisaku Nogami, and Kazunari Shibata. Time-resolved spectroscopy and photometry of M dwarf flare star YZ Canis Minoris with OISTER and TESS: Blue asymmetry in the Hα line during the non-white light flare. PASJ, 73(1):44–65, February 2021. doi: 10.1093/pasj/psaa098.

S. C. Marsden, P. Petit, S. V. Jeffers, J. Morin, R. Fares, A. Reiners, J. D. do Nascimento, M. Aurière, J. Bouvier, B. D. Carter, C. Catala, B. Dintrans, J. F. Donati, T. Gastine, M. Jardine, R. Konstantinova-Antova, J. Lanoux, F. Lignières, A. Morgenthaler, J. C. Ramìrez-Vèlez, S. Théado, V. Van Grootel, and BCool Collaboration. A BCool magnetic snapshot survey of solar-type stars. MNRAS, 444(4):3517–3536, November 2014. doi: 10.1093/mnras/stu1663.

Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall PTR, USA, 1 edition, 2008. ISBN 0132350882.

R. Martínez-Arnáiz, J. Maldonado, D. Montes, C. Eiroa, and B. Montesinos. Chromospheric activity and rotation of FGK stars in the solar vicinity. An estimation of the radial velocity jitter. A&A, 520:A79, September 2010. doi: 10.1051/0004-6361/200913725.

M. Mittag, J. H. M. M. Schmitt, and K. P. Schröder. Ca II H+K fluxes from S-indices of large samples: a reliable and consistent conversion based on PHOENIX model atmospheres. A&A, 549:A117, January 2013. doi: 10.1051/0004-6361/201219868.

D. Montes, M. J. Fernandez-Figueroa, E. de Castro, and M. Cornide. Excess Hα emission in chromospherically active binaries. A&A, 294:165–176, February 1995.

Philip S. Muirhead, Jason Nordhaus, and Maria R. Drout. Revised Stellar Parameters for V471 Tau, A Post-common Envelope Binary in the Hyades. AJ, 163(1):34, January 2022. doi: 10.3847/1538-3881/ac390f.

Elisabeth R. Newton, Jonathan Irwin, David Charbonneau, Perry Berlind, Michael L. Calkins, and Jessica Mink. The Hα Emission of Nearby M Dwarfs and its Relation to Stellar Rotation. ApJ, 834(1):85, January 2017. doi: 10.3847/1538-4357/834/1/85.

Francesco Palla and Steven W Stahler. *The formation of stars: Steven W. Stahler and Francesco Palla*. Wiley-VCH, 2008. ISBN 3527405593.

V. Perdelwitz, M. Mittag, L. Tal-Or, J. H. M. M. Schmitt, J. A. Caballero, S. V. Jeffers, A. Reiners, A. Schweitzer, T. Trifonov, I. Ribas, A. Quirrenbach, P. J. Amado, W. Seifert, C. Cifuentes, M. Cortés-Contreras, D. Montes, D. Revilla, and S. L. Skrzypinski. CARMENES input catalog of M dwarfs. VI. A time-resolved Ca II H&K catalog from archival data. A&A, 652:A116, August 2021. doi: 10.1051/0004-6361/202140889.

J. Sebastian Pineda, Allison Youngblood, and Kevin France. The M-dwarf Ultraviolet Spectroscopic Sample. I. Determining Stellar Parameters for Field Stars. ApJ, 918(1):40, September 2021. doi: 10.3847/1538-4357/ac0aea.

N. E. Piskunov, F. Kupka, T. A. Ryabchikova, W. W. Weiss, and C. S. Jeffery. VALD: The Vienna Atomic Line Data Base. A&AS, 112:525, September 1995.

Nikolai Piskunov and Jeff A. Valenti. Spectroscopy Made Easy: Evolution. A&A, 597:A16, January 2017. doi: 10.1051/0004-6361/201629124.

Richard D. Robinson, Lawrence E. Cram, and Mark S. Giampapa. Chromospheric H alpha and CA II Lines in Late-Type Stars. ApJS, 74:891, December 1990. doi: 10.1086/191525.

Matthew Route. The Rise of ROME. I. A Multiwavelength Analysis of the Star-Planet Interaction in the HD 189733 System. ApJ, 872(1):79, February 2019. doi: 10.3847/1538-4357/aafc25.

N. N. Samus', E. V. Kazarovets, O. V. Durlevich, N. N. Kireeva, and E. N. Pastukhova. General catalogue of variable stars: Version GCVS 5.1. *Astronomy Reports*, 61(1):80–88, January 2017. doi: 10.1134/S1063772917010085.

J. H. M. M. Schmitt, J. Robrade, J. U. Ness, F. Favata, and B. Stelzer. X-rays from accretion shocks in T Tauri stars: The case of BP Tau. A&A, 432(2):L35–L38, March 2005. doi: 10.1051/0004-6361:200500014.

M. Simon, A. Dutrey, and S. Guilloteau. Dynamical Masses of T Tauri Stars and Calibration of Pre-Main-Sequence Evolution. ApJ, 545(2):1034–1043, December 2000. doi: 10.1086/317838.

A. P. Sousa, J. Bouvier, S. H. P. Alencar, J. F. Donati, E. Alecian, J. Roquette, K. Perraut, C. Dougados, A. Carmona, S. Covino, D. Fugazza, E. Molinari, C. Moutou, A. Santerne, K. Grankin, É. Artigau, X. Delfosse, G. Hebrard, and SPIRou Consortium. Star-disk interaction in the T Tauri star V2129 Ophiuchi: An evolving accretion-ejection structure. A&A, 649:A68, May 2021. doi: 10.1051/0004-6361/202140346.

Klaus G. Strassmeier. Starspots. A&A Rev., 17(3):251–308, September 2009. doi: 10.1007/s00159-009-0020-6.

Klaus G. Strassmeier, Francis C. Fekel, Bernard W. Bopp, Robert C. Dempsey, and Gregory W. Henry. Chromospheric CA II H and K and H alpha Emission in Single and Binary Stars of Spectra Types F6–M2. ApJS, 72:191, January 1990. doi: 10.1086/191414.

C. A. O. Torres, G. R. Quast, L. da Silva, R. de La Reza, C. H. F. Melo, and M. Sterzik. Search for associations containing young stars (SACY). I. Sample and searching method. A&A, 460(3):695–708, December 2006. doi: 10.1051/0004-6361:20065602.

J. A. Valenti and D. A. Fischer. VizieR Online Data Catalog: Spectroscopic properties of cool stars. I. (Valenti+, 2005). *VizieR Online Data Catalog*, art. J/ApJS/159/141, November 2005.

J. A. Valenti and N. Piskunov. Spectroscopy made easy: A new tool for fitting observations with synthetic spectra. A&AS, 118:595–603, September 1996.

J. A. Valenti and N. Piskunov. SME: Spectroscopy Made Easy, February 2012.

A. H. Vaughan, G. W. Preston, and O. C. Wilson. Flux measurements of Ca II and K emission. PASP, 90:267–274, June 1978. doi: 10.1086/130324.

B. J. Wargelin, V. L. Kashyap, J. J. Drake, D. García-Alvarez, and P. W. Ratzlaff. X-Ray Flaring on the dMe Star, Ross 154. ApJ, 676(1):610–627, March 2008. doi: 10.1086/528702.

D. C. Wells, E. W. Greisen, and R. H. Harten. FITS - a Flexible Image Transport System. A&AS, 44:363, June 1981.

O. C. Wilson. Flux Measurements at the Centers of Stellar H- and K-Lines. ApJ, 153:221, July 1968. doi: 10.1086/149652.

J. T. Wright, G. W. Marcy, R. Paul Butler, and S. S. Vogt. Chromospheric Ca II Emission in Nearby F, G, K, and M Stars. ApJS, 152(2):261–295, June 2004. doi: 10.1086/386283.

B. Zaire, J. F. Donati, and B. Klein. Magnetic field and activity phenomena of the K2 dwarf V471 Tau. MNRAS, 504(2):1969–1988, June 2021. doi: 10.1093/mnras/stab1019.

Appendix A

Additional tables and figures

Table A.1: All stars used in the preliminary and final analysis of this work. All values are taken from SIMBAD.

Star	**Spectral type**	**radial velocity** [$\mathrm{km\,s^{-1}}$]	T_{eff} [K]	$\log g$ [cgs units]	**[Fe/H]** [dex]	$v \sin i$ [$\mathrm{km\,s^{-1}}$]
107 Psc	K1V	-33.619 ± 0.001	5259	4.58	0.02	1.2
111 Tau	F8V	37.7 ± 0.2	6184	4.38	0.09	16
14 Cet	F5V	11.487 ± 0.001	6583	3.82	-0.09	4.357
18 Sco	G2Va	11.764	5818	4.4	0.04	2.056
51 Peg	G2IV	-33.165 ± 0.001	5817	4.42	0.24	4.99
54 Psc	K0.5V	-32.898 ± 0.001	5289	4.52	0.24	2.5
61 UMa	G8V	-5.18 ± 0.08	5507	4.54	-0.06	3.26
BD+05 378	K6Ve	5.75 ± 0.37	4100	4.5	0.3	40 ± 4
BD+13 2618	M0V	-12.52 ± 0.03	3481	–	–	9.6 ± 0.9
BD+16 2708	M3V	-7.219 ± 0.001	3649	–	–	–
BD+36 2322B	M4.5V	-6.78 ± 0.29	3136	–	–	39.2 ± 0.7
BD+44 2051B	M6.0V	70.02 ± 0.21	3000	5.5	–	8.0 ± 0.7
BD+61 195	M1.5V	-6.05	3750	–	0.249	8
bet UMa	A1IVps	-13.1 ± 0.1	9620	3.9	0.16	46
chi01 Ori	G0V	-15.48 ± 0.18	6027	4.58	-0.01	10.79
G 13-22	M4.5Ve	-12.88 ± 0.15	2937	4.59	0.05	6.1 ± 0.7
G 154-44	M4.0Ve	-32.69 ± 0.03	2947	–	–	4.3 ± 0.7
G 208-45	M6V	5.46 ± 0.13	2736	–	–	7.9 ± 0.7
LP 731-58	M6.5V	2.18 ± 0.02	2683	5.05	0.04	3.3 ± 0.7
GSC543-620	M4.5V	-10.82 ± 0.24	3096	–	–	49.9 ± 0.9
HD 102195	K0V	2.05	5311	4.44	0.04	–
HD 175726	G0V	10.322 ± 0.002	6069	4.43	-0.04	13.6
HD 189733	K2V	-2.317 ± 0.001	5044	4.6	0.02	4.5
HD 190771	G2V	-25.133 ± 0.002	5751	4.44	0.11	6.1
HD 209290	M0.5V	18.25	3668	–	0.01	2.68
BP Tau	K5/7Ve	15.76 ± 0.13	4171	4.668	–	12.74 ± 0.36
V410 Tau	K3Ve	20.66 ± 0.39	4382	4.034	–	78.84 ± 0.53
HD 285968	M2.5V	26.236 ± 0.001	3435	–	-0.01	–
HD 36395	M1.5Ve	8.56	3734	–	0.19	2.5 ± 0.3
HD 46375	G9V	-0.979 ± 0.001	5355	4.47	0.27	4.3
HD 52265	G0V	53.838 ± 0.001	6136	4.23	0.21	3.912
HD 56124	G0	22.569 ± 0.001	5857	4.47	0.03	1.5 ± 0.5

Continued on next page

Table A.1 – continued from previous page

Star	**Spectral type**	**radial velocity** [km s^{-1}]	$T_{\rm eff}$ [K]	$\log g$ [cgs units]	**[Fe/H]** [dex]	$v \sin i$ [km s^{-1}]
pi.01 UMa	G0.5V	-12.826 ± 0.002	5921	4.46	−0.01	11.21
HD 76151	G2V	32.036	5788	4.44	0.12	3.58
HD 9986	G2V	-21.024 ± 0.001	5791	4.42	0.09	1
HD 22468	K2:Vnk	-15.3 ± 0.1	4736	3.35	−0.16	34.5
5 LMi	F7Vs	4.212 ± 0.002	6147	4.28	0.06	9.0 ± 0.5
HD 179949	F8V	-24.619 ± 0.001	6287	4.34	0.21	6.451
kap01 Cet	G5V	19.359 ± 0.001	5730	4.5	0.06	5.86
ksi Boo	G7Ve+K5Ve	1.59 ± 0.03	5511	4.53	−0.14	3.26
BD+57 793	F5V	-11.91 ± 2.36	6685	4.22	–	20.3 ± 2.0
ups And	F9V	-28.662 ± 0.001	6269	4.12	0.19	10.73
AA Tau	K5Ve	17.37 ± 0.15	4181	4.406	–	14.18 ± 0.37
BD+20 2465	dM3	12.43	2991	–	0.06	4.1 ± 0.7
AF Lep	F8V(n)k:	20.61 ± 0.51	6126	4.34	0.55	54.7 ± 0.5
CU Cnc	M5Ve	-39.342 ± 3.232	3141	–	–	–
DF Tau	M3Ve	16.46 ± 0.34	3937	3.948	–	31.26 ± 0.32
DG Tau	K6Ve	15.4 ± 0.6	–	–	–	24.7 ± 0.7
DK Tau	K7e+K7e	16.2 ± 0.2	4231	4.616	–	17.36 ± 0.47
51-15	M6.5Ve	10.62 ± 0.11	2809	–	−0.36	11.4 ± 0.7
EK Dra	G1.5V	-20.687 ± 0.004	5845	4.47	0.16	16.8 ± 0.5
eps Eri	K2V	16.376	5049	4.59	−0.15	4.08
EV Lac	M4.0V	0.287 ± 0.002	3167	–	–	5.9 ± 0.7
GL Vir	M4.5Ve	6.0 ± 1.5	2910	4.64	0.01	15.6 ± 0.8
CD-35 10525	K7Ve	-3.6 ± 1.3	4192	4.12	–	–
HK Aqr	M0Ve	4.35 ± 0.13	3693	–	–	69.4 ± 0.7
HN Peg	G0V+	-16.723 ± 0.001	5972	4.44	–	12.81
MT Ori	K2	21.86 ± 1.07	–	–	–	–
OT Ser	M1.0V	6.831 ± 0.003	3417	–	–	5.2 ± 1.0
R Leo	M7-9e	14.1 ± 0.6	–	–	–	–
R Sct	K4/5pec	43.8 ± 0.6	4500	–	−0.47	–
RT Cyg	M2-7e	-116.0 ± 4.5	–	–	–	–
R Vir	M3.5-7e	-26.6 ± 0.7	–	–	–	–
TW Hya	K6Ve	12.335 ± 0.002	3725	–	–	5.4 ± 0.7
TZ Ari	M4.5V	-28.34 ± 0.01	2837	–	−0.14	4.4 ± 0.7
U Psc	M4e	−35	3292	0.76	−0.2	–
V1005 Ori	M0Ve	19.16 ± 0.06	3661	–	–	8.7 ± 1.6
Ross 154	M3.5Ve	-10.494 ± 0.002	3184	5.02	−0.19	5.2 ± 0.9
SR 9	K5e	-7.1 ± 1.3	4223	4.14	–	14.7 ± 0.9
V374 Peg	M3.5Ve	-4.79 ± 0.09	3114	–	–	36.9 ± 0.7
HD 209458	F9V	-14.743 ± 0.001	6118	4.37	0.03	3.7
V471 Tau	K2V+DA	37.4 ± 0.5	–	–	–	–
V889 Her	G2V	-23.0 ± 0.3	5769	4.3	−0.04	39.0 ± 0.5
YZ CMi	M4.0Ve	26.66 ± 0.03	3088	–	–	6.6 ± 0.8
Wolf 47	M5V	-5.76 ± 0.02	3003	–	–	12.5 ± 0.7

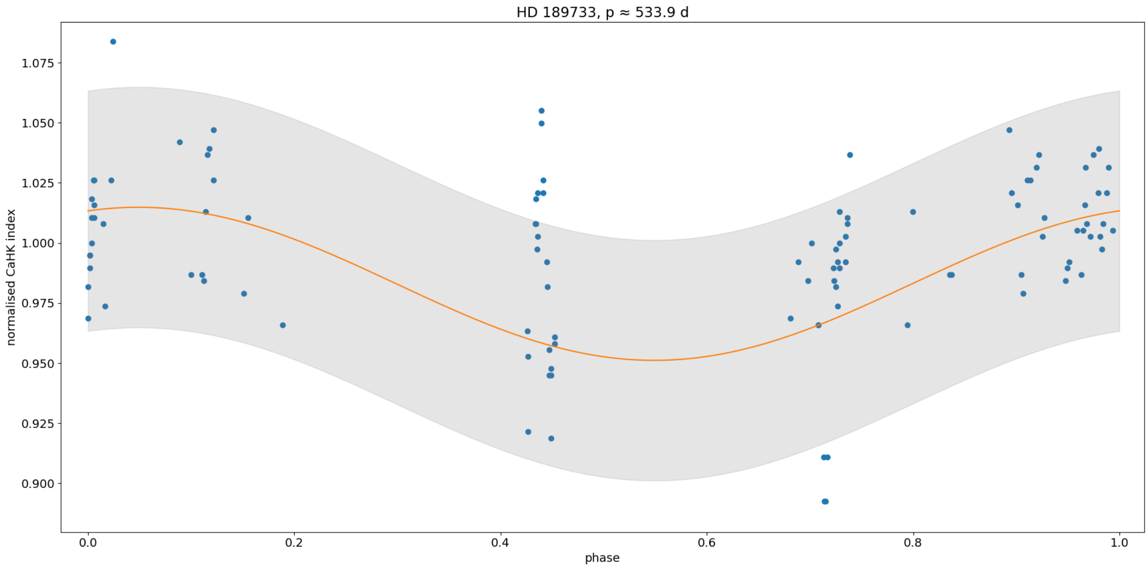

Figure A.1: HD 189733. Phased plot of the CaHK index.

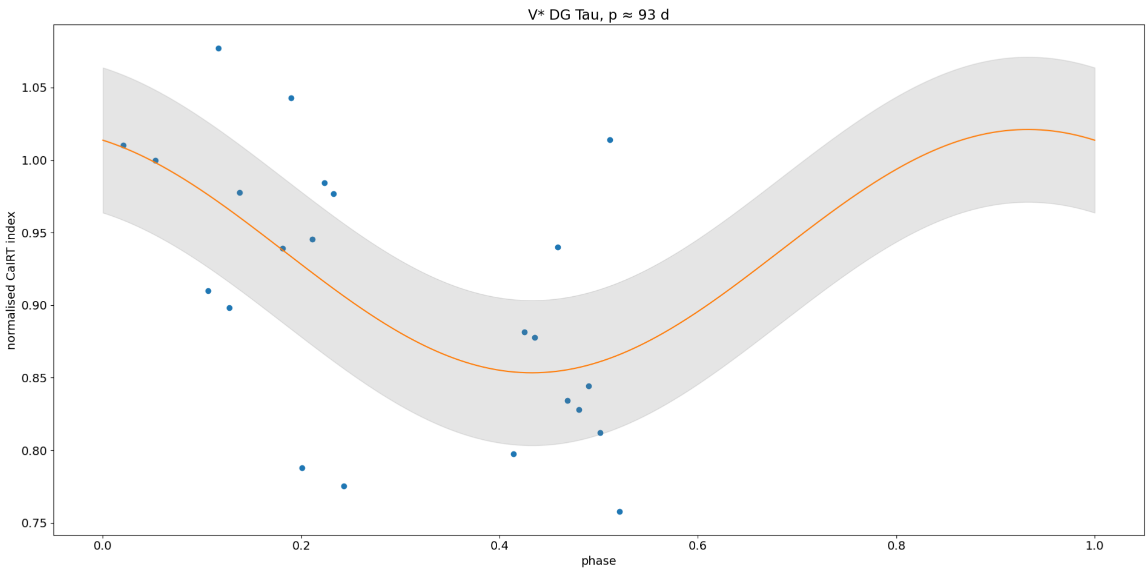

Figure A.2: DG Tau. Phased plot of the CaIRT index.

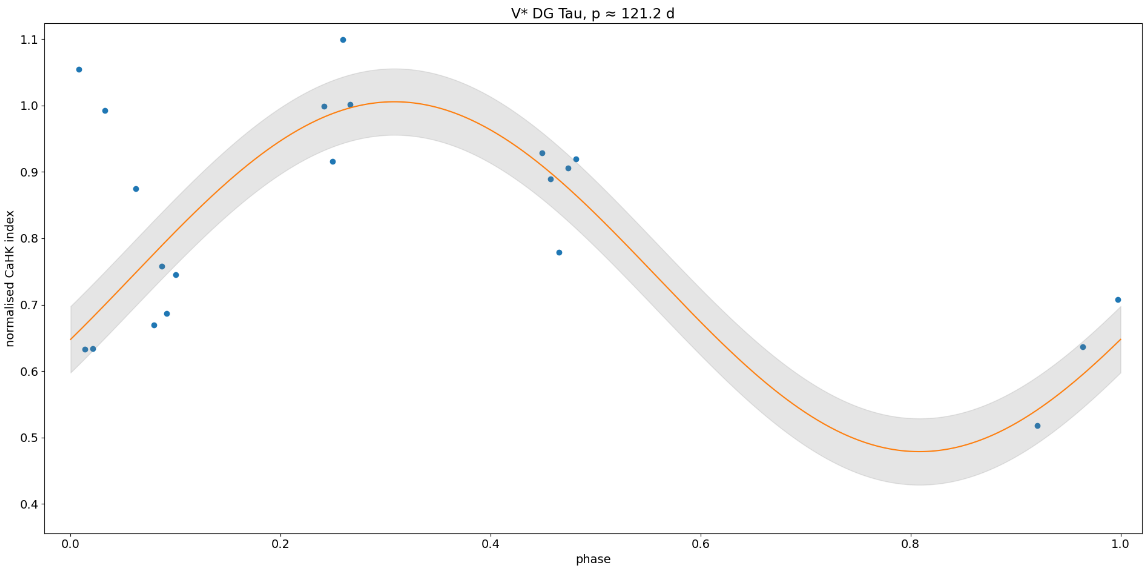

Figure A.3: DG Tau. Phased plot of the CaHK index.

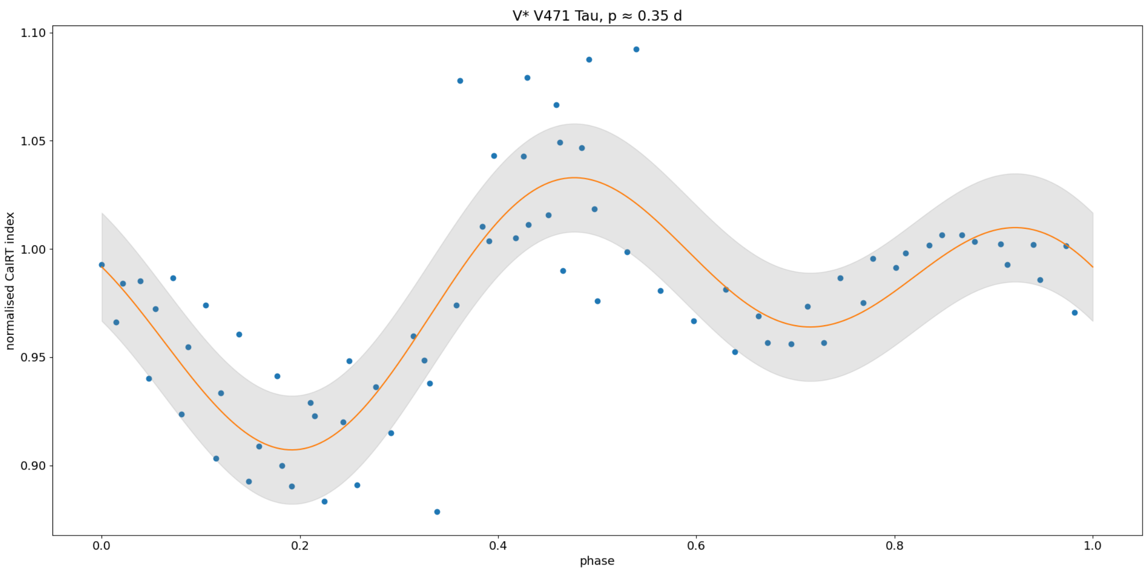

Figure A.4: V471 Tau. Phased plot of the CaIRT index with $n = 2$ Fourier terms.

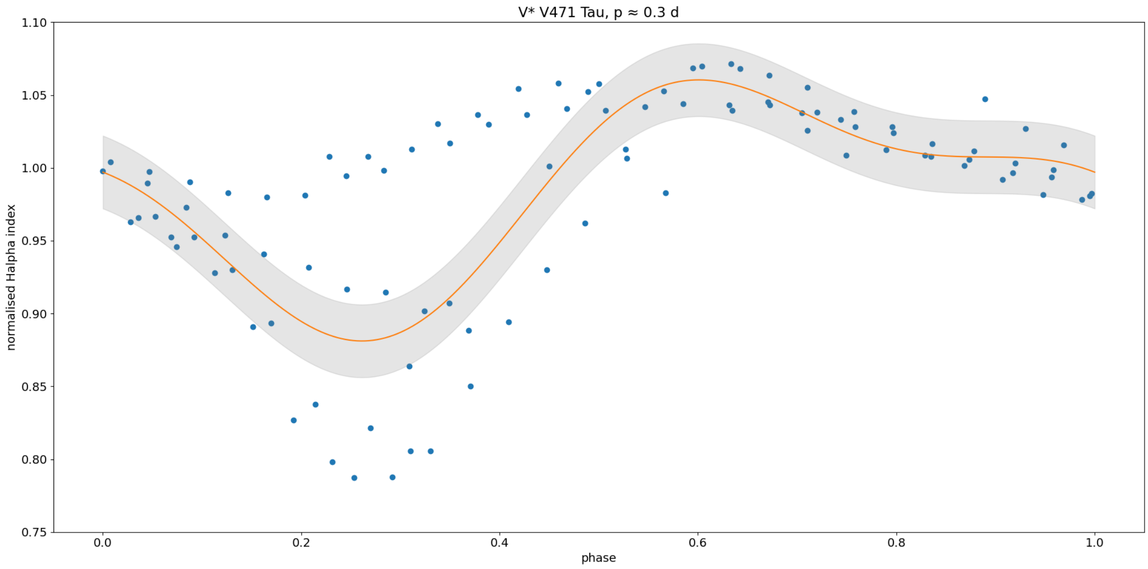

Figure A.5: V471 Tau. Phased plot of the Hα index with $n = 2$ Fourier terms.

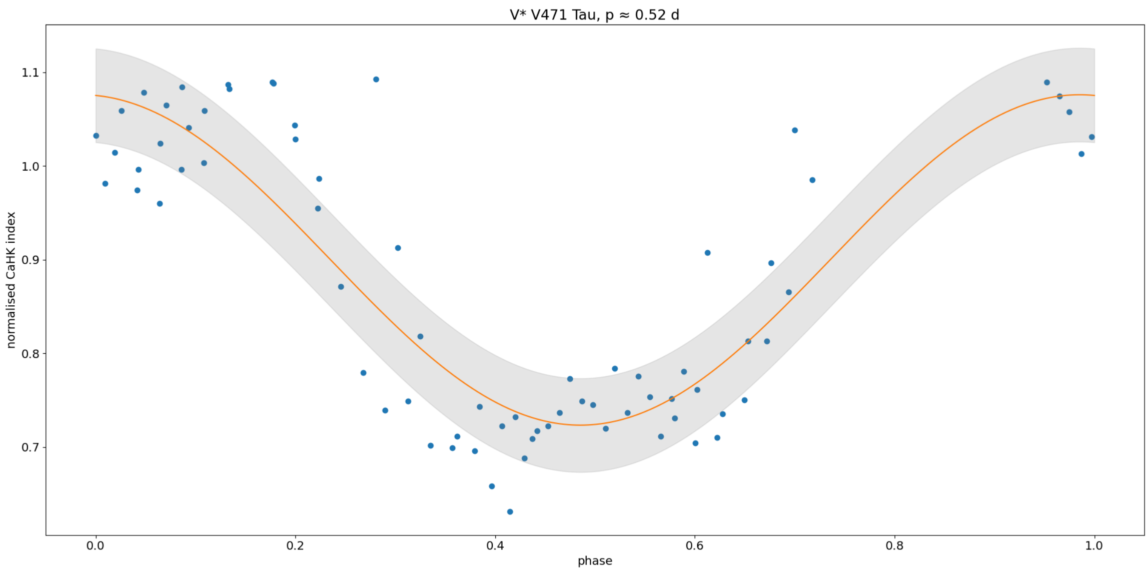

Figure A.6: V471 Tau. Phased plot of the CaHK index with $n = 1$ Fourier term.

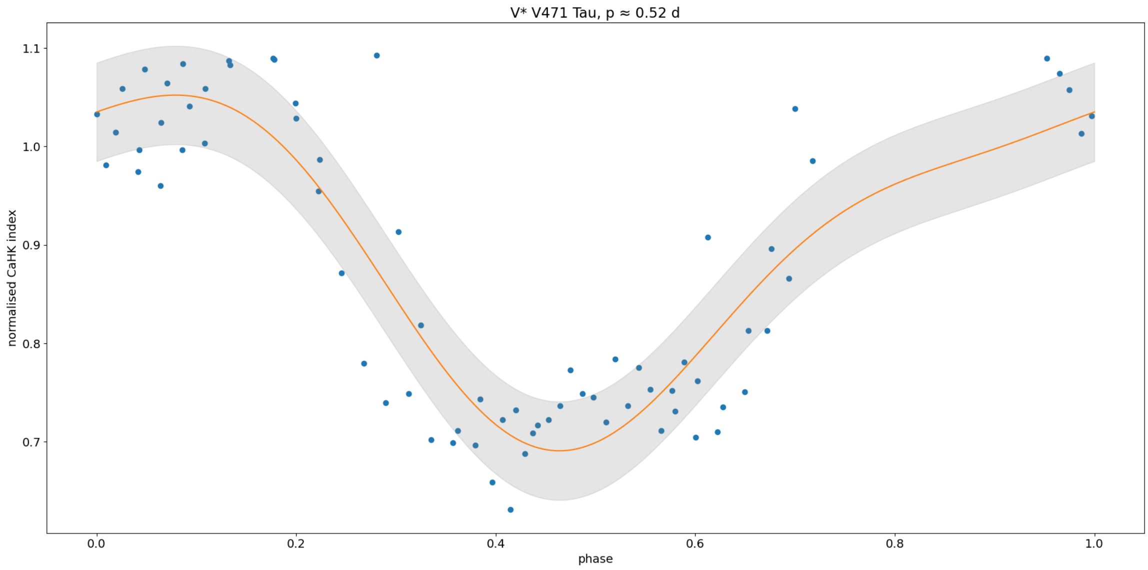

Figure A.7: V471 Tau. Phased plot of the CaHK index with $n = 2$ Fourier terms.

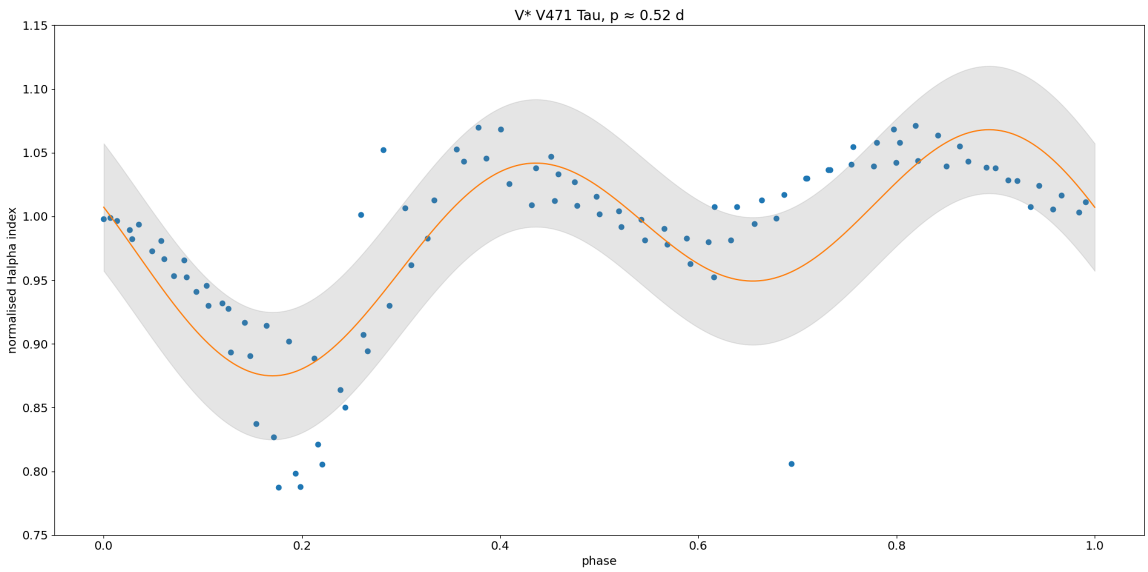

Figure A.8: V471 Tau. Phased plot of the Hα index with $n = 2$ Fourier terms. The main frequency is taken from the orbital period of the system.

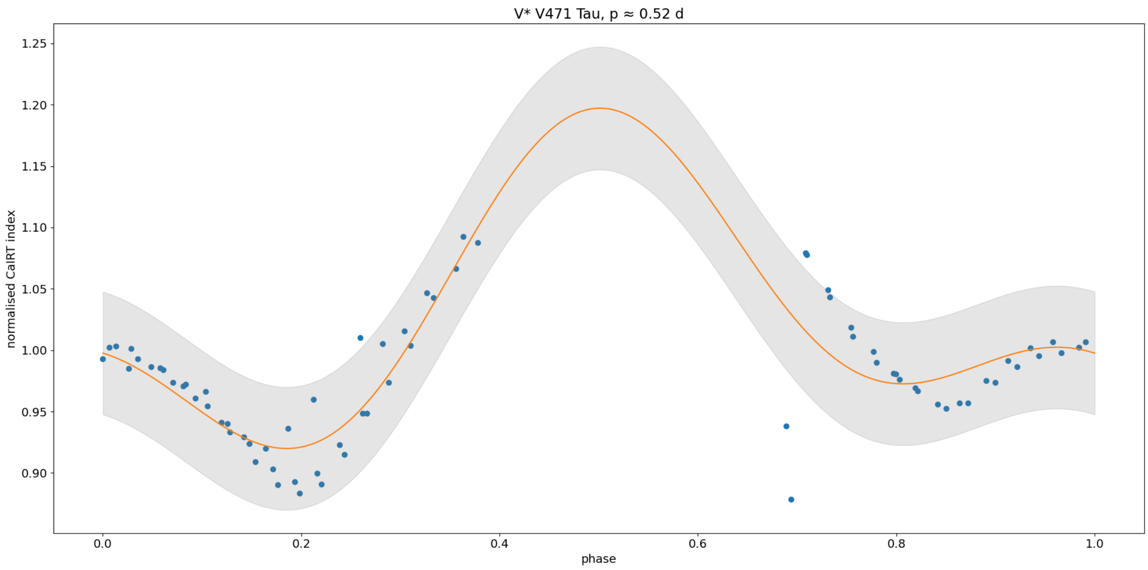

Figure A.9: V471 Tau. Phased plot of the CaIRT index with $n = 2$ Fourier terms. The main frequency is taken from the orbital period of the system.