Study on period search methods of variable stars: Application to ASAS and CRTS databases

Thesis submitted to
Cochin University of Science and Technology
in partial fulfillment of the requirements
for the award of the degree of

Doctor of Philosophy

Shaju K.Y.

Department of Physics
Cochin University of Science and Technology
Kochi - 682022

November 2013

Study on period search methods of variable stars: Application to ASAS and CRTS databases
Ph.D. thesis in the field of Physics

Author :
Shaju K.Y.
Department of Physics
Cochin University of Science and Technology
Kochi - 22
shajuky@gmail.com

Research Supervisor :
Prof Ramesh Babu T.
Department of Physics
Cochin University of Science and Technology
Kochi - 22
rbt@cusat.ac.in

Front cover : Light curves of normal RR Lyrae, Blazkho RR Lyrae and Delta Scuti (*Courtsey : CRTS, Caltech*)
Back cover : Light curves of contact, detached and semi-detached eclipsing binaries (*Courtsey : CRTS, Caltech*)
Cover design by : Prof.Arun V.

Scientists are searching for the ultimate truth...

Dedicated to
Deepa, Ruchira and Rithikesh

CERTIFICATE

Certified that the work presented in this thesis is a bonafide work done by Mr.Shaju K.Y., under my guidance in the Department of Physics, Cochin University of Science and Technology and that this work has not been included in any other thesis submitted previously for the award of any degree.

Kochi
November, 2013

Prof.Ramesh Babu T.
(Supervising Guide)

DECLARATION

I hereby declare that the work presented in this thesis is based on the original work done by me under the guidance of Prof.Ramesh Babu T., Professor, Department of Physics, Cochin University of Science and Technology and has not been included in any other thesis submitted previously for the award of any degree.

Kochi
November, 2013

Shaju K.Y.

Contents

List of Figures

List of Tables

List of abbreviations

- **ASAS -** **A**ll **S**ky **A**utomated **S**urvey
- **CRTS -** **C**atalina **R**eal-**T**ime **T**ransient **S**urvey
- **CS -** **C**ubic **S**pline
- **GLSP -** **G**eneralized **L**omb-**S**cargle **P**eriodogram
- **HJD -** **H**eliocentric **J**ulian **D**ate
- **HRD -** **H**ertzsprung-**R**ussell **D**iagram
- **JD -** **J**ulian **D**ate
- **LC -** **L**ight **C**urve
- **LSP -** **L**omb-**S**cargle **P**eriodogram
- **MCS -** **M**odified **C**ubic **S**pline
- **OGLE -** **O**ptical **G**ravitational **L**ensing **E**xperiment
- **PDF -** **P**robability **D**ensity **F**unction
- **PDM -** **P**hase **D**ispersion **M**inimization
- **PLC -** **P**hased **L**ight **C**urve
- **SigSpec - Sig**nificant **Spec**trum

Acknowledgments

I was fortunate to work under the guidance of Prof.Ramesh Babu T, who had exceptionally deep knowledge and wide research experience, which helped me throughout the period of research. I would like to express him my deep and sincere gratitude for guiding me with the right suggestions, at the right moments. His approachability, proficiency, logical way of thinking and the way of tackling difficult situations in lucid manner has helped me immensely during the research work.

I would like to express my sincere gratitude to Prof.V.C.Kuriakose, my Doctoral Committee member and other theory group professors Prof.M.Sabir and Dr.Titus K. Mathew for being there as a constant source of support and providing help whenever I needed them most.

I take this opportunity to thank Prof.Pradeep, Head of the Department of Physics and also the previous Heads of Department Dr.Godfrey Louis, Dr.M.R. Anantharaman, Dr.K. P. Vijayakumar, Dr.V C Kuriakose and Dr.Ramesh Babu T, Prof.M.Sabir for providing necessary facilities for my research work. I also thank my M.Phil. teachers Dr.Sudha Kartha and Dr.Jayalekshmy, Ph.D. course work coordinator Dr.Junaid Bushiri and all other faculty members for all the support provided. The help and cooperation by the office staff of this department and library staff are also gratefully acknowledged.

I express my sincere gratitude to Dr.Ninan Sajeeth Philip, St.Thomas College, Kozhencheri, who inspired me by discussing some aspects of light curve analysis. I remember Dr.Piet Reegen, University of Vienna,, Austria, who is the author of SigSpec, for the useful discussions, valuable suggestions and help extended during the numerical works. I do hereby express my heart-felt condolence for his untimely, pre-mature and sad demise to his family. His departure from this world happened at a time, when our collaborative work was at the pinnacle and it was a huge loss to the scientific community and personally to me.

I thank Dr.Andrew Drake and the CRTS team for giving the data for my research work. The CSS survey is funded by the National Aeronautics and Space Administration under Grant No. NNG05GF22G issued through the Science Mission Directorate

Near-Earth Objects Observations Program. The CRTS survey is supported by the U.S.National Science Foundation under grants AST-0909182 and AST-1313422.

I express my sincere gratitude to Prof.Ajit Kembavi,IUCAA, Pune for the warm hospitality extended during my visit there. I also express my sincere gratitude to Dr.Ranjan Gupta, Dr.Ranjeev Misra and Dr.Deepankar Bhatacharya. My discussions with Dr.Balakrishna N, Statistics department, CUSAT, was very effective and I am thankful for his friendly approach.

Also I am grateful for the useful discussions with Jogesh C. Babu, Bob Stellingwerf, Zechmeister, Min-Su Shin, Roman Baluev, K.Sriram and Shanti Priya. Our thanks are due to Ajit Kembavi, IUCAA, Pune and V.C.Kuriakose, IRC, CUSAT, Kochi, for the computational facilities provided. Thanks to those people who published their time series analysis tools and user manuals on the web.

I thankfully remember the good times I had with the research scholars of Physics Department and other departments, with whom, the interactions made me few years younger. I express my deep sense of gratitude to Dr.Jayadevan, Dr.Radhakrishnan, Dr.Nijo, Priyesh, Vivek, Tharanath, Saneesh, Anoop, Nima, Bhavya, Lini, Neelam, Prasia, Jishnu and Dinto for the love and consideration they showered on me. Days have been lovely due to the presence of Theory team, OED team, Optics team, Thin film team and the hostel mates, who filled the hours with happiness. I sincerely thank Prof. Arun V, for the exquisite cover design.

I am extremely grateful to my wife Deepa, who has been unbelievably supportive, without which it would have been impossible to lead such a long student life as I had. I am also thankful to my daughter Ruchira for giving publicity to my research work among her classmates and constantly encouraging me to discover an extra solar planet. Nobody is so happier than me, when my son Rithikesh was born at the end of my F.D.P. duration.

I thank the previous manager Fr.Jose Stephen Menachery C.M.I, present manager Fr.John Paliakkara C.M.I. and Principal Fr.Dr.Jose Thekkan C.M.I., of Christ College, Irinjalakuda, Thrissur for permitting me to do research under F.D.P. program and also for many other official helps. Also I thank the physics department Head, teaching and non-teaching staff and entire teaching, non-teaching staff and office staff of Christ College, Irinjalakuda, Thrissur. I gratefully acknowledge the financial support (KLCA008/2009/F.I.P.) provided by UGC, Bangalore, under Faculty Development Program.

Shaju K. Y.

Preface

Study on variable stars is an important topic of modern astrophysics. After the invention of powerful telescopes and high resolving powered CCD's, the variable star data is accumulating in the order of peta-bytes. The huge amount of data need lot of automated methods as well as human experts. This thesis is devoted to the data analysis on variable star's astronomical time series data and hence belong to the inter-disciplinary topic, Astrostatistics.

For an observer on earth, stars that have a change in apparent brightness over time are called variable stars. The variation in brightness may be regular (periodic), quasi periodic (semi-periodic) or irregular manner (aperiodic) and are caused by various reasons. In some cases, the variation is due to some internal thermo-nuclear processes, which are generally known as intrinsic variables and in some other cases, it is due to some external processes, like eclipse or rotation, which are known as extrinsic variables. Intrinsic variables can be further grouped into pulsating variables, eruptive variables and flare stars. Extrinsic variables are grouped into eclipsing binary stars and chromospherical stars. Pulsating variables can again classified into Cepheid, RR Lyrae, RV Tauri, Delta Scuti, Mira etc. The eruptive or cataclysmic variables are novae, supernovae, etc., which rarely occurs and are not periodic phenomena. Most of the other variations are periodic in nature.

Variable stars can be observed through many ways such as photometry, spectrophotometry and spectroscopy. The sequence of photometric observations on variable stars produces time series data, which contains time, magnitude and error. The plot between variable star's apparent magnitude and time are known as light curve. If the time series data is folded on a period, the plot between apparent magnitude and phase is known as phased light curve. The unique shape of phased light curve is a characteristic of each type of variable star. One way to identify the type of variable star and to classify them is by visually looking at the phased light curve by an expert. For last several years, automated algorithms are used to classify a group of variable stars, with the help of computers.

Research on variable stars can be divided into different stages like observation, data reduction, data analysis, modeling and classification. The modeling on variable stars helps to determine the short-term and long-term behaviour and to construct theoretical models (for eg:- Wilson-Devinney model for eclipsing binaries) and to derive stellar properties like mass, radius, luminosity, temperature, internal and external structure, chemical composition and evolution. The classification requires the determination of the basic parameters like period, amplitude and phase and also some other derived parameters. Out of these, period is the most important parameter since the wrong periods can lead to sparse light curves and misleading information.

Time series analysis is a method of applying mathematical and statistical tests to data, to quantify the variation, understand the nature of time-varying phenomena, to gain physical understanding of the system and to predict future behavior of the system. Astronomical time series usually suffer from unevenly spaced time instants, varying error conditions and possibility of big gaps. This is due to daily varying daylight and the weather conditions for ground based observations and observations from space may suffer from the impact of cosmic ray particles.

Many large scale astronomical surveys such as MACHO, OGLE, EROS, ROTSE, PLANET, Hipparcos, MISAO, NSVS, ASAS, Pan-STARRS, Kepler,ESA, Gaia, LSST, CRTS provide variable star's time series data, even though their primary intention is not variable star observation. Center for Astrostatistics, Pennsylvania State University is established to help the astronomical community with the aid of statistical tools for harvesting and analysing archival data. Most of these surveys releases the data to the public for further analysis.

There exist many period search algorithms through astronomical time series analysis, which can be classified into parametric (assume some underlying distribution for data) and non-parametric (do not assume any statistical model like Gaussian etc.,) methods. Many of the parametric methods are based on variations of discrete Fourier transforms like Generalised Lomb-Scargle periodogram (GLSP) by Zechmeister(2009), Significant Spectrum (SigSpec) by Reegen(2007) etc. Non-parametric methods include Phase Dispersion Minimisation (PDM) by Stellingwerf(1978) and Cubic spline method by Akerlof(1994) etc.

Even though most of the methods can be brought under automation, any of the method stated above could not fully recover the true periods. The wrong detection of period can be due to several reasons such as power leakage to other frequencies which is due to finite total interval, finite sampling interval and finite amount of data. Another problem is aliasing, which is due to the influence of regular sampling. Also spurious periods appear due to long gaps and power flow to harmonic frequencies is an inherent problem of Fourier methods. Hence obtaining the exact period of variable star from it's time series data is still a difficult problem, in case of huge databases, when subjected to automation. As Matthew Templeton, AAVSO, states "Variable star data analysis is not always straightforward; large-scale, automated analysis design is non-trivial". Derekas et al. 2007, Deb et.al. 2010 states "The processing of huge amount of data in these databases is quite challenging, even when looking at seemingly small issues such as period determination and classification".

It will be beneficial for the variable star astronomical community, if basic parameters, such as period, amplitude and phase are obtained more accurately, when huge time series databases are subjected to automation. In the present thesis work, the theories of four popular period search methods are studied, the strength and weakness of these methods are evaluated by applying it on two survey databases and finally a modified form of cubic spline method is introduced to confirm the exact period of variable star. For the classification of new variable stars discovered and entering them in the "General Catalogue of Variable Stars" or other databases like "Variable Star Index", the characteristics of the variability has to be quantified in term of variable star parameters.

Chapter 1 gives a brief account of variable stars, their types, sampling constraints in astronomical time series, challenges in astro-time series analysis and some problems faced under automation. A short review of the existing period search methods are also included. Some of the characteristic light curves are also shown.

Chapter 2 reviews the parametric period search methods such as Generalised Lomb-Scargle periodogram(GLSP) and Significant Spectrum(SigSpec). We show the results of applying GLS periodogram and SigSpec on some sample data and justify the need for improvement of methods, while doing automation. For our convenience, we have re-coded SigSpec package into FORTRAN and is given in appendix A.1.

Chapter 3 reviews the non-parametric period search methods such as Phase Dispersion Minimisation and Cubic Spline method. Cubic spline method is modified with unequally spaced knots, which is used for period confirmation. The Modified Cubic Spline(MCS) method is coded in FORTRAN and is given in appendix B.1.

Chapter 4 shows the results of sequential application of PDM method and MCS method, for the period search of ASAS (All Sky Automated Survey) database. The new results include improved periods, entirely different periods and harmonic periods, when compared to the published catalogue. The results are tabulated and the corresponding light curves are shown.

Chapter 5 shows the results of application of the SigSpec/GLSP followed by MCS for the period search of CRTS (Catalina Real-Time Transient Survey) database. The results are tabulated and the corresponding light curves are plotted.

Chapter 6 summarizes the substantial findings of the works presented in the thesis and suggests future scopes of the work.

- ***Appendix A.1*** contains the SigSpec method used for the automated period search, written in FORTRAN. This program can be run by the script in ***appendix C.1***
- ***Appendix B.1*** contains the modified cubic spline method written in FORTRAN, used for the automated period confirmation.
- ***Appendix B.2*** contain phase folding program written in FORTRAN, which is needed for the automated running of MCS.
- ***Appendix C.1*** is a bash script, used for the automated running of *SigSpec* program written in ***appendix A.1***.
- ***Appendix C.2*** is a bash script used to select the best period from the output file produced by ***sigspecf.for*** given in ***appendix A.1***.
- ***Appendix C.3*** is a bash script used to automate the running of MCS program given in ***appendix B.1***.
- ***Appendix C.4*** is bash script for checking the time series input files for any possible errors and to clean the data.

List of Publications

In refereed journals

- "Photometric study of hot Contact binaries in SMC", D. Shanti Priya, K. Sriram, K.Y. Shaju and P. Vivekananda Rao, *Bull. Astr. Soc. India* (2013) 41, 159172

- "Cubic spline analysis with unevenly spaced knots : Confirmation of exact period of variable stars", Shaju K.Y. and Ramesh Babu Thayyullathil.(Manuscript under preparation).

Conference presentations

- "Automated period detection from variable stars time series database", Shaju.K.Y., Piet Reegen and Ramesh Babu Thayyullathil, ASI Conference, Raipur,Chatheesgad, 6-8 Febraury 2010.

- "Automated period detection from variable stars time series data", Shaju.K.Y., Piet Reegen and Ramesh Babu Thayyullathil, 11th Asian-Pacific Regional IAU Meeting 2011: APRIM - 2011, Chiang-Mai, Thailand, 21-26 July 2011.

- "Re-Analysis of ASAS Database - Detection of new variable stars"Shaju K.Y. and Ramesh Babu Thayyullathil, ASI meeting, Thiruvananthapuram, Kerala 20-22 February 2013.

Chapter 1

Variable stars and Light curves

1.1 Introduction

In this chapter, variable stars are briefly introduced, types of variable stars are described, some typical light curves are shown, various astronomical surveys, which produces variable star data are mentioned, astronomical time series are introduced. Finally various period search methods are briefly discussed. Some common problem faced by the data analyst, while doing period search are discussed.

1.2 Variable stars

For an observer on earth, stars that have a change in apparent brightness over time are called variable stars. Theoretically saying, all stars becomes variable stars at least a few times during their evolution. All stars display variations in brightness during birth time and also at the death time. In between this evolution, stars will change its position from the main sequence to the final stage, in the H-R diagram. Even though our parent star, Sun exhibits minor spectroscopic and flare type variations, Sun is not considered as a variable star for us. But finally Sun will move from the main sequence to become a red giant star and even well before that the human beings has to be anticipate about this, for sustaining 'life'in the universe. Studying the variations on Sun-like stars will help to predict the future evolution of our Sun from the current state.

In olden days, a star is considered variable, if the magnitude variation is detectable to human eye and the order of the period is average human life. But with the advent of modern sophisticated observation and detection techniques, the magnitude of variable stars under study is extended to vary from 0.001 to 20 and the period from seconds to several years. Also the depth of observation extends into other galaxies and globular clusters.

The variation in brightness may be regular (periodic), quasi periodic (semi-periodic) or irregular manner (aperiodic) and are caused by various reasons. In some cases, the variation is due to some internal thermo-nuclear processes, which are generally known as intrinsic variables and in some other cases, it is due to some external processes, like eclipse or rotation, which are known as extrinsic variables. Intrinsic variables can be further grouped into pulsating variables, eruptive variables and flare stars. Extrinsic variables are grouped into eclipsing binary stars and chromospherical stars. Pulsating variables can again categorized into Cepheids, RR Lyrae, RV Tauri, Delta Scuti, Mira etc, whose light variations are periodic in nature. The eruptive or cataclysmic variables are novae, supernovae, etc., which rarely occurs and also not a periodic phenomenon. Most of the other variations are periodic in nature.

International Astronomical Union (IAU) Division G Commission 27 (variable star) and 42 (close binary star) are exclusively for Variable Stars and responsible for maintaining General Catalogue of Variable Stars (GCVS), located at Sternberg Astronomical Institute, Moscow. Information Bulletin of Variable Stars (IBVS), Konkoly, Hungary periodically produces supplements to GCVS. The latest edition of the General catalogue of Variable Stars (GCVS, 2013 [68]) lists nearly 47,969 variable stars. There are some dedicated variable star groups in other parts of the world like AAVSO (American Association of Variable Star Observers), BAAVSS (British Astronomical Association Variable Star Section).

1.2.1 Nomenclature of variable stars

Historically there were many schemes for naming variable stars, one of the most popular method of naming a variable star was using the Roman capital letters, starting from R to Z (R,S,T,U,V,W,X,Y,Z), followed by Latin name of the constellation. For example, W UMa (W Ursae Majoris). Thus initially there were only 9 options for each constellation. Then the scheme is extended into RR to RZ, SS to SZ, etc., with the condition that the second alphabet should be higher than the first alphabet. Then total options became 54. Further extension of the scheme into AA to AZ, BB to BZ, etc., omitting J, total options became 334 for each constellation. Most of the variable star types are popular by this scheme. In this thesis, we use the position of star, RA$\pm$DEC combination as variable star-ID and use the above discussed scheme to identify the type of variable star.

1.2.2 Importance of variable star research

Variable stars are speaking to us, about the mysteries of universe, as [43] says, by changing their magnitude in a particular way. Those who study variable stars, tries to understand the content of the speech, and interprets the hidden secrets of the stars and universe. Our universe is not static and every change in the universe in any scale is important in understanding the structure of universe. Determining the short-term and long-term behaviour of stars is important in stellar astrophysics. Constructing theoretical models (Wilson-Devinney model for eclipsing binaries) helps to derive stellar properties like mass, radius, luminosity, temperature, internal and external structure, chemical composition and evolution. Studying sun-like variable stars will help us to predict the evolution of our sun, the variations on which will greatly affect the human life on earth. These kind of predictions will be much benefited for the future generations on earth. Another important usage is to classify the variable stars into different categories, which is an indication of our broad knowledge about the universe.

1.2.3 Astronomical time series

Variable stars can be observed through many ways such as photometry, spectrophotometry and spectroscopy. A sequence of photometric observations produce a time series with three columns data with time, magnitude and error. SExtractor(Source Extractor) (`astromatic.net/software/sextractor`) is a program that builds a catalogue of objects from an astronomical image. ISIS(image-subtraction) (`http://www2.iap.fr/users/alard/package.html`) is a package to process a series of CCD images using the image subtraction method to generate time series and light curves. By analysing this astronomical time series, the period, amplitude and the phase of variable star oscillations can be estimated. Spectroscopic observations on variable stars reveal the spectral type, chemical composition, luminosity class etc.

Astronomical time is usually recorded in Julian date(JD) format or other variations of it like Modified Julian date(MJD) or Heliocentric Julian date(HJD). The Julian date format is “JD” followed by 7 digits, then a decimal point, followed by 6 digits. JD is continuous count of days and fractions since noon at Greenwich on 1 January 4713 BC, before which, it is believed that no scientific observations exist. JD 2450000.000000 corresponds to year 1995 and 2460000.000000 corresponds to year 2023, hence for observations during these years, JD245 is usually not recorded and only the remaining 4 digits, decimal point and remaining 7 digits are only recorded. Apparent magnitude (mostly V magnitude) is measured by comparing brightness with magnitude of a nearby non-variable star of known magnitude. Error is related with various conditions of measuring magnitudes. In astronomy, an 'epoch' is a moment in time for which celestial coordinates or orbital elements are specified. In the case of celestial coordinates, the position at other times can be computed by taking into account precession and proper motion.

1.2.4 Time series analysis

Time series analysis is the application of mathematical and statistical tests on time varying data, to quantify the variation through the extraction of some parameters, and to use that parameters to learn something about the behaviour of the system. In other words, the goal of time-series analysis is to gain some physical understanding of the system under observation: what makes the system time variable?, what makes this system different than other systems with similar variability?, etc. Once the current state of the system is well understood, then the next aim is to forecast or predict future behavior of the system. If the system deviates from the prediction, then the reasons for the deviation has to be found out and corrections has to be added recursively for further forecasting of time varying systems. In the real world, there are many types of time series, like economic time series, marketing time series, climate time series, time series in communication and digital signal processing etc. There are many statistical methods and standard tools like R-package, SPSS, PSPP etc. for analysing these time series.

1.2.5 Problems with astronomical time series

The astronomical time series is significantly different from the usual statistical time series mainly due to the unevenly spaced time instants, varying error conditions and also due to the possibility of big gaps. The majority of astronomical measurements cannot be taken continuously and evenly over long periods of time due to several reasons. For ground based observations, daily varying daylight and the weather conditions are unavoidable sources of varying errors. If the observation is taken from space, the measured magnitude may suffer from the impact of cosmic ray particles on CCD. These kinds of stray light corruptions occasionally producing data points beyond repair [62], [26]. Also when observations taken at different times and various geographically located telescopes are combined, then also long time gaps and normalisation problem can appear in the resulting time series. Hence the commonly available statistical time series analysis methods can't be directly applied on astronomical time series and extract correct information from it. Obtaining the exact period and other parameters of variable star from it's luminosity-time series data is still a difficult problem, in case of huge databases, when subjected to automation [25].

1.2.6 Light curve (LC) and Phased Light curve (PLC)

A light curve (LC) is a X-Y plot of a variable star's apparent magnitude versus time, with the time is plotted on the X-axis and the inverted magnitude is plotted on the Y-axis. Then the increase along the Y-axis shows increase in brightness and the maxima (highest points) correspond to the brightest magnitudes the star attains. The time required for one complete oscillation of the light curve is known as the period T.

If the period T is known or assumed, then the time series can be folded on period T, so that the phase is given by

$$\text{phase} = \frac{(t_k - t_0)}{T} - \text{Integer part of } \left(\frac{t_k - t_0}{T}\right) = \text{Fractional part of } \left(\frac{t_k - t_0}{T}\right)$$

where t_0 is the initial epoch of the variable star observation and t_k is the time instants of observations. Phase is defined as the fractional portion of the number of periods, which have elapsed since a given epoch. The value of phase Φ now will be in between 0.0(start of a period) and 1.0(end of a period). Now the plot between phase and inverted magnitude is known as phased light curve (PLC). Usually phase is extended beyond 0.0 or 1.0, by simply repeating the structure, so that any discontinuity at the boundaries can be checked.

If the folded period T is exact, good shaped, minimum scattered PLC is obtained. The shape of PLC is a unique characteristic of each type of variable star. Familiarising the shape of the PLC and identifying the variability type is one of the criterion for showing the experience of a variable star researcher.

1.2.7 Astronomical variability surveys

There exist huge amount of observational time series data of variable stars, from various survey observations, which are given below. Even though some of these survey data are collected for different purposes, they also produce variable star time series data. The data analysis on these time series can lead to the discovery of interesting new objects and variable stars. In the near future, more data are expected to come from many surveys and astro-time series data analysis is expected to be a prospectus research area.

- ASAS (All Sky Automated Survey – detection of photometric variability) (`astrouw.edu.pl/asas/`).

- MACHO (MAssive Compact Halo Objects – search for dark matter by gravitational lensing) (`macho.anu.edu.au/`).
- EROS (Expérience pour la Recherche d'Objets Sombres – search for and study of dark stellar bodies by their gravitational microlensing effects on stars) (`eros.in2p3.fr/`).
- OGLE (Optical Gravitational Lensing Experiment – search for dark matter by gravitational microlensing) (`ogle.astrouw.edu.pl/`).
- ROTSE (Robotic Optical Transient Search Experiment – searching gamma-ray bursts)(`rotse.net/`).
- The PLANET Collaboration (Probing Lensing Anomalies NETwork – detecting and characterising microlensing anomalies) (`bustard.phys.nd.edu/MPS/`).
- MISAO Project (Multitudinous Image-based Sky-survey and Accumulative Observations – making use of images in the world for new object discoveries and data acquisition of known objects)(`aerith.net/misao/`).
- Pan-STARRS(Panoramic Survey Telescope And Rapid Response System - asteroids, comets, variable stars)(`pan-starrs.ifa.hawaii.edu/public/`).
- PASS (Permanent All Sky Survey)(`iac.es/proyecto/pass/`).
- XO Project (Photometric search for Jovian planets transiting very bright stars)(`http://www-int.stsci.edu/ pmcc/xo/index.shtml`).
- LINEARdb (The LINEAR Survey Photometric Database)(`https://astroweb.lanl.gov/lineardb/`).
- NSVS (Northern Sky Variability Survey) (`http://skydot.lanl.gov/nsvs/nsvs.php`).
- HATNet (Hungarian-made Automated Telescope) (`https://www.cfa.harvard.edu/ gbakos/HAT/index.html`).
- SuperWASP (Wide Angle Search for Planets) (`superwasp.org/index.html`).
- MOA (Microlensing Observations in Astrophysics)(`www.physics.auckland.ac.nz/moa/`).
- MEGA (Microlensing Exploration of the Galaxy Andromeda – microlensing search targeting M31) (`http://user.astro.columbia.edu/ arlin/MEGA/`).

- SAVS (Semi-Automatic Variability Search) (astri.uni.torun.pl/ gm/SAVS/).
- LSST (Large Synoptic Survey Telescope - variable sources, Transient alerts)(lsst.org/lsst).
- ESA mission Gaia (Global Astrometric Interferometer for Astrophysics – three-dimensional map of the Milky Way) (sci.esa.int/science-e/www/area/index.cfm?fareaid=1).
- HIPPARCOS (HIgh Precision PARallax COllecting Satellite – mission of the European Space Agency) (rssd.esa.int/index.php?project=HIPPARCOS&page=index).
- CoRoT (COnvection ROtation et Transits planetaires) (esa.int/Our_Activities/Space_Science/COROT).
- Kepler mission by NASA (http://kepler.nasa.gov/).
- The CRTS(Catalina Real-Time Transient Survey – large scale synoptic survey, to detect and classify all types of variability's in the sky) (crts.caltech.edu).

1.2.8 Taxonomy of variable stars

Variable stars are grouped according to the astrophysical reasons for variability, light curve shapes and other characteristic parameters. Generally variable stars divided into two groups: intrinsic and extrinsic.

The stars, which vary in brightness, due to some internal process belong to intrinsic groups. Stars in this group vary in brightness as they expand and contract, heat and cool. Examples are eruptive variables like supernovae, novae, dwarf novae and pulsating variables.

If the brightness variation is due to some external influence, they belong to extrinsic groups, which include eclipsing binary and rotating variables.

The intrinsic groups can be again categorized into periodic, non-periodic and semi-periodic variables. The periodic variables are those stars, whose brightness varies in a regular, repeated way as a function of time. Intrinsic periodic variables are the pulsating stars like Cepheid, RR Lyrae stars etc. All extrinsic variables are generally periodic.

Non-periodic or irregular variable stars are those stars, whose brightness varies irregularly with time, such as supernovae, novae etc. Irregular variables have no apparent periodicity in their light curves

Semi-periodic variables are stars having some irregularity and also periodicity. Example is Z UMa. The semi-regular class of variable stars are long period variables whose light curves exhibit additional complexities beyond those of the well-behaved regular variables. For instance, a semi-regular variable might have an average period of 150 days, meaning that on an average, successive maxima are roughly 150 days apart. Here some maxima separation might be 100 days, while others might be 200 days.

The most abundant types of variable stars are eruptive variables, pulsating variables, rotating variables, cataclysmic variables, and eclipsing binary systems. According to fourth edition of GCVS catalogue, there are 7 types of variable stars and within each type, there exist many sub-types[68] as shown below.

- Eruptive (FU, GCAS, I, IA, IB, IN, INA, INB, INT, IT, IN(YY), IS, ISA, ISB, RCB, RS, SDOR, UV, UVN, WR)
- Pulsating (ACYG, BCEP, BCEPS, CEP, CEP(B), CW, CWA, CWB, DCEP, DCEPS, DSCT, DSCTC, GDOR, L, LB, LC, M, PVTEL, RPHS, RR, RR(B), RRAB, RRC, RV, RVA, RVB, SR, SRA, SRB, SRC, SRD, SXPHE, ZZ, ZZA, ZZB),
- Rotating (ACV, ACVO, BY, ELL, FKCOM, PSR, SXARI)
- Cataclysmic (explosive and nova-like) variables (N, NA, NB, NC, NL, NR, SN, SNI, SNII, UG, UGSS, UGSU, UGZ, ZAND)
- Eclipsing binary systems (E, EA, EB, EW, GS, PN, RS, WD, WR, AR, D, DM, DS, DW, K, KE, KW, SD)
- Intense variable X-ray sources (X, XB, XF, XI, XJ, XND, XNG, XP, XPR, XPRM, XM)
- Other symbols (BLLAC, CST, GAL, L:, QSO, S, *, +, :)
- The new variability types (ZZO, AM, R, BE, LBV, BLBOO, EP, SRS, LPB)

for more details see GCVS [68]. Several types of variable stars are briefly described below with some of the characteristic light curves.

1.2.9 Pulsating variables

Pulsating stars undergo periodic expansion and contraction of their surface layers called pulsations. The pulsations may be radial or non-radial. A radially pulsating star remains spherical in shape, while in the case of non-radial pulsations the star's shape periodically deviates from a sphere, and even neighboring zones of its surface may have opposite pulsation phases. Depending on the period, mass and evolutionary status of the star, and also on the scale of pulsational phenomena, the pulsating variables can be again sub-classified into Mira's and semi regular variables, Cepheids, RV Tauri stars, RR Lyrae, RV Tauri, δ-Scuti etc (*astro.utoronto.ca/ percy/var.html*).

- Cepheids: These stars have periods of 1-70 days with amplitudes of variation from 0.1 to 2.0 magnitudes. Cepheids obey a strict period-luminosity relationship, with higher luminosity Cepheids having longer periods. Therefore, by measuring the period of a Cepheid variable, its luminosity is obtained and from the relation between apparent brightness and luminosity, the distance between observer and Cepheid is calculated. This is the method used to measure distance to the galaxies and globular clusters and hence Cepheids are known as standard candles of the universe. The typical light curve of a Cepheid is shown in Figure 1.1.

- RR Lyrae stars: These stars have periods of 0.2-1.2 days with amplitudes of variation from 0.3 to 2 magnitudes. These pulsating variables are white giant stars. These can again sub-divided into RRab, RRc etc. The typical light curve of a normal RR Lyrae is shown in Figure 1.2.

- RV Tauri stars: These stars have periods of 30-150 days with amplitudes of variation up to 3.0 magnitudes. They are yellow supergiants.

- Long Period Variables (Mira): These stars have periods of 80-1000 days with amplitudes of variation from 2.5 to 5.0 magnitudes. They are giant red variables.

- Delta Scuti variables: They are low amplitude variables. Some have amplitudes of nearly one magnitude and regular light curves like some of the RR Lyrae stars and Cepheid variables. Others have complex LC's and multiple periods with milli-magnitude light variations. The pulsations of delta scuti stars are important in studying the interior structure of the star and comes under asteroseismology. The typical light curve of a delta scuti is shown in Figure 1.3.

- Semi regular stars: These stars have periods of 30-1000 days with amplitudes of variation from 1.0 to 2.0 magnitudes. They are giants and supergiants displaying periodicity superimposed with intervals of irregular light variation.

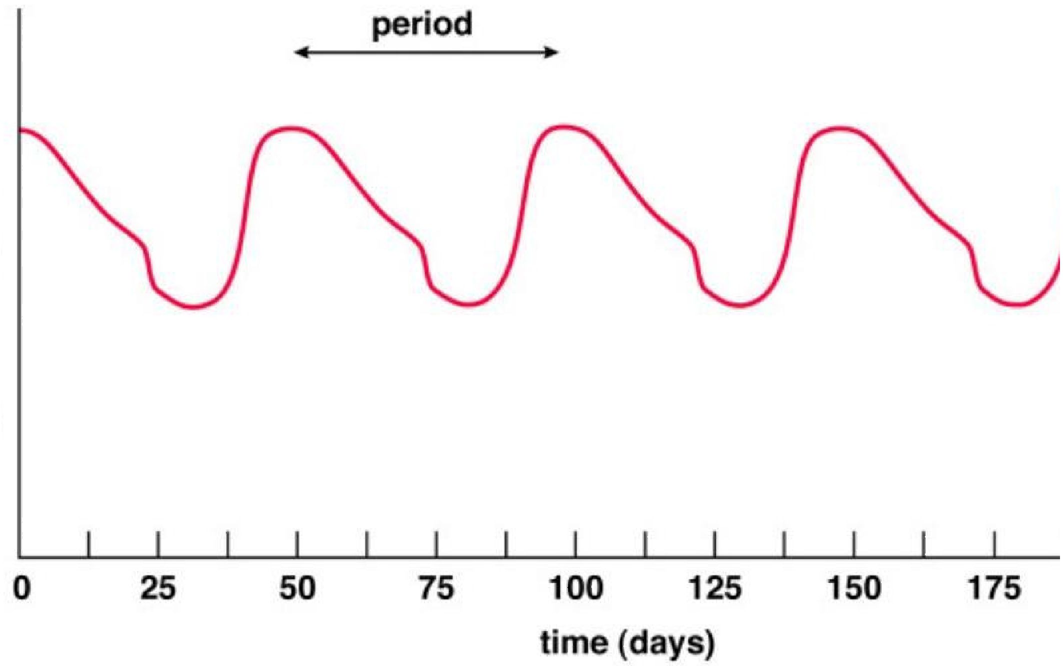

Figure 1.1: Light curve of Cepheid: Courtsey Google

- Small-Amplitude Pulsating Red Giants (SAPRGs): These stars have periods of 5-100 days with amplitudes of variation from 0.05 to 1 magnitude. As their name (also called small-amplitude red variables) suggests, these stars are red giants. Due to their instability, a majority of red giants physically expand and contract (pulsate) periodically as a result of convective processes. The pulsations may be radial or non-radial and these stars may be multi-periodic.

Type	Period in days	Amplitude in mag
Cepheid	1-70	0.1-2.0
RR Lyrae	0.2-1.2	0.3 to 2
RV Tauri	30-100	up to 3.0
Mira	80-1000	2.5-5.0
Semiregular	30-1000	1.0-2.0

Table 1.1: Typical period and amplitude range of some type of variable stars

1.2.10 Eruptive stars

These stars show brightness variation because of violent processes and flares occurring in their chromospheres and coronae, usually accompanied by shell ejections or

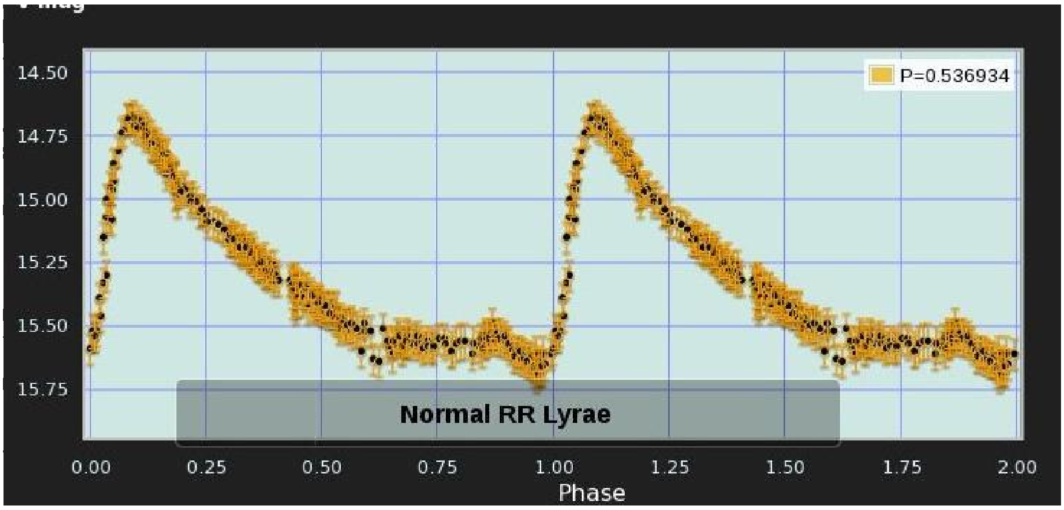

Figure 1.2: Light curve of RR Lyrae: Courtsey CRTS

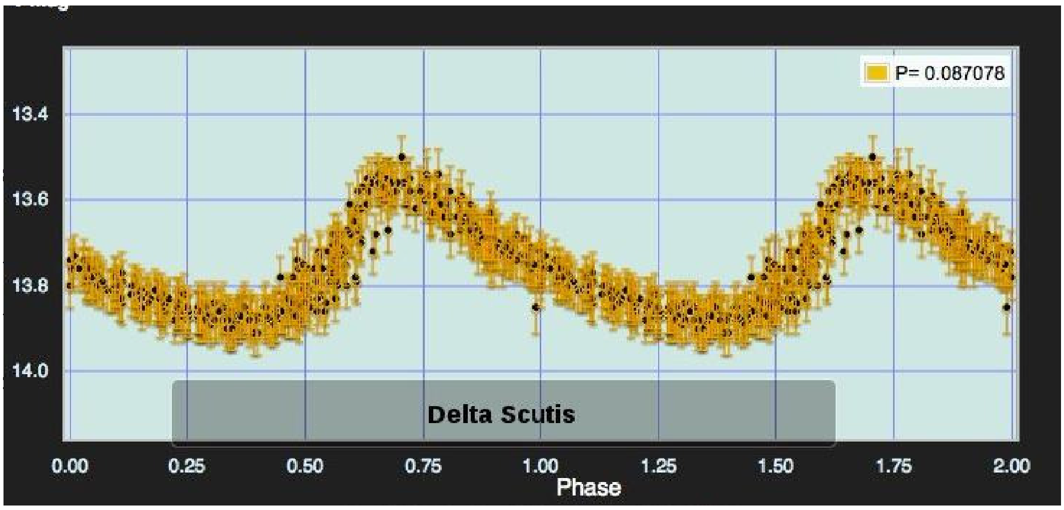

Figure 1.3: Light curve of δ-Scuti : Courtsey CRTS

mass outflow in the form of stellar winds of variable intensity and/or by interaction with the surrounding interstellar medium. The most common types of eruptive variables are: Supernovae, Novae, Recurrent Novae, Dwarf Novae, Symbiotic Stars, R Coronae Borealis Stars, Flare Stars, T Tauri variables.

- Supernovae: These stars show sudden, dramatic, and final magnitude increases as a result of a catastrophic stellar explosion. Thus, there is no period, and amplitudes of variation are 20+ magnitudes.

- Novae: These close binary systems consist of a main sequence, Sun-like star and a white dwarf. They increase in brightness by 7 to 16 magnitudes in a matter of one to several hundred days. After the outburst, the star fades slowly to its initial brightness over several years or decades. Periods are typically 1-300+ days, and amplitudes of variation are 7-16 magnitudes.

- Recurrent Novae: These objects are similar to novae, but have two or more slightly smaller-amplitude outbursts during their recorded history. Periods are 1-200+ days, and amplitudes of variation are 7-16 magnitudes.

- Dwarf Novae: These are close binary systems made up of a Sun-like star, a white dwarf, and an accretion disk surrounding the white dwarf. The accretion disk "erupts" every few weeks. The typical light curve of a Dwarf Novae is shown in Figure 1.4.

- Symbiotic Stars: These close binary systems consist of a red giant and a hot blue star, both embedded in nebulosity. They show nova-like outbursts, up to three magnitudes in amplitude, and are semi-periodic.

- R Coronae Borealis Stars: These are rare, luminous, hydrogen-poor, carbon-rich, variables that spend most of their time at maximum light, occasionally fading as much as nine magnitudes at irregular intervals. They then slowly recover to their maximum brightness after a few months to a year.

- Flare Stars: Also known as UV Ceti stars, these are intrinsically faint, cool, red, main-sequence stars that undergo intense outbursts from localized areas of the surface. The result is an increase in brightness of two or more magnitudes in several seconds, followed by a decrease to its normal minimum in about 10 to 20 minutes.

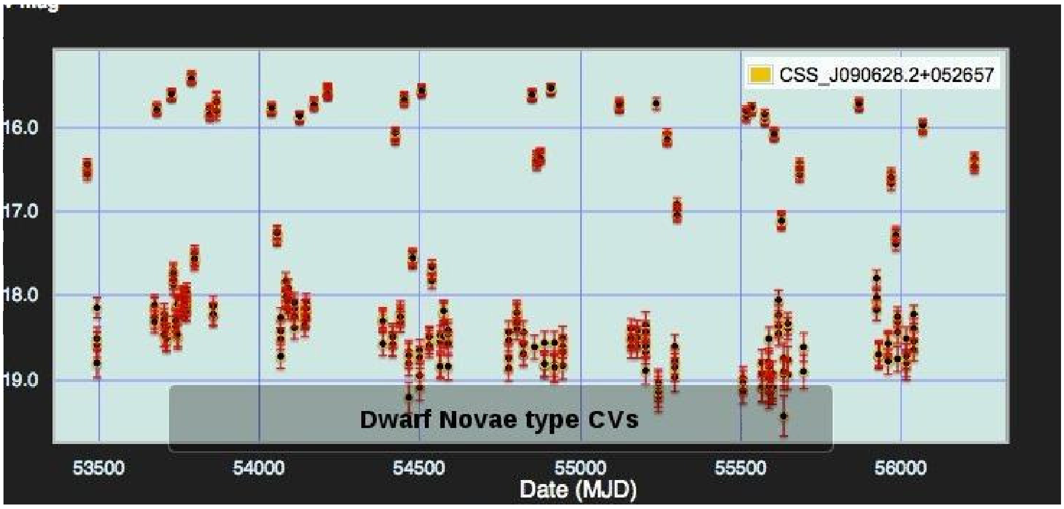

Figure 1.4: Light curve of dwarf novae : Courtsey CRTS

1.2.11 Rotating variables

These variables have non-uniform surface brightness and/or ellipsoidal shapes, whose variability is caused by axial rotation with respect to the observer. The non-uniformity of surface brightness distributions may be caused by the presence of spots or by some thermal or chemical inhomogeneity of the atmosphere caused by a magnetic field whose axis is not coincident with the rotation axis. These stars can be subdivided into various types. eg: pulsars, elliptical stars and magnetic variables. In rotating variable stars, variation in brightness is usually small and results in the rotation of the star exposing dark or bright spots, or patches (”starspots”) on its surface.

1.2.12 Cataclysmic (explosive and nova-like) variables

Cataclysmic variables show outbursts caused by thermonuclear bursts on their surface layers (novae) or deep in their interiors (supernovae). It is often referred as “nova-like” for variables that show nova-like outbursts caused by rapid energy release in the surrounding space (UG-type stars) and also for objects not displaying outbursts but resembling explosive variables at minimum light by their spectral characteristics. Most of cataclysmic variables are close binary systems which components have strong mutual influence on the evolution of each star. Generally hot, dwarf component is surrounded by an accretion disk made from the matter lost by its cooler and more extended companion. Eg: dwarf novae, classic novae and supernovae.

1.2.13 Eclipsing variables

Two or more stars revolving around common centre of mass, sometimes eclipse one another, for an observer on earth. In the case of two component stars or binary stars, the orbital plane should be close to the observer's line of sight so that the components periodically eclipse each other. Consequently, the observer finds changes of the apparent combined brightness of the system with the period coincident with that of the components orbital motion. There are three ways of classifying eclipsing binary systems taking into account shape of the light curve, physical characteristics of components (their luminosity classes) and degree of their Roche lobe filling as follows.

- EA (Detached eclipsing binary, Algol, $\beta-$Persei) Binaries with spherical or ellipsoidal components. Since they are detached, it is possible to specify, from their light curves, the moments of the beginning and end of the eclipses. Between eclipses the light remains almost constant because of reflection effects or physical variations. Secondary minima may be absent. An extremely wide range of periods is observed, from 0.2 to 10000 days. Light amplitudes are also quite different and may reach several magnitudes. The typical light curve of an Algol is shown in Figure 1.7.

- EB (Semi detached eclipsing binary, $\beta-$Lyrae) These are eclipsing binaries having ellipsoidal components and light curves for which it is impossible to specify the exact times of onset and end of eclipses, since they are semi-detached. The secondary minimum is observed in all cases, whose depth is considerably smaller than that of the primary minimum. The periods are larger than 1 day. The light amplitudes are usually less than 2 magnitude in V band. The typical light curve of a $\beta-$Lyrae is shown in Figure 1.6.

- EW (Contact eclipsing binary, W Ursae Majoris) These are having periods shorter than 1 day and consisting of ellipsoidal components almost in contact. From the light curves, it is impossible to specify the exact times of onset and end of eclipses, since they are contact binaries. The depths of the primary and secondary minima are almost equal. The light amplitudes are usually less than 0.8 mag in V band. The typical light curve of a W UMa is shown in Figure 1.5.

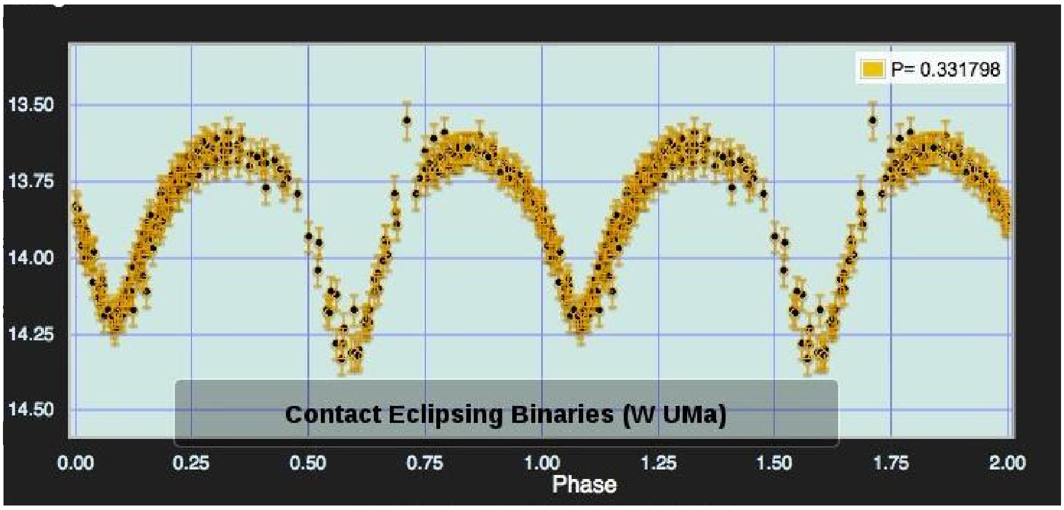

Figure 1.5: Light curve of contact eclipsing binary : Courtsey CRTS

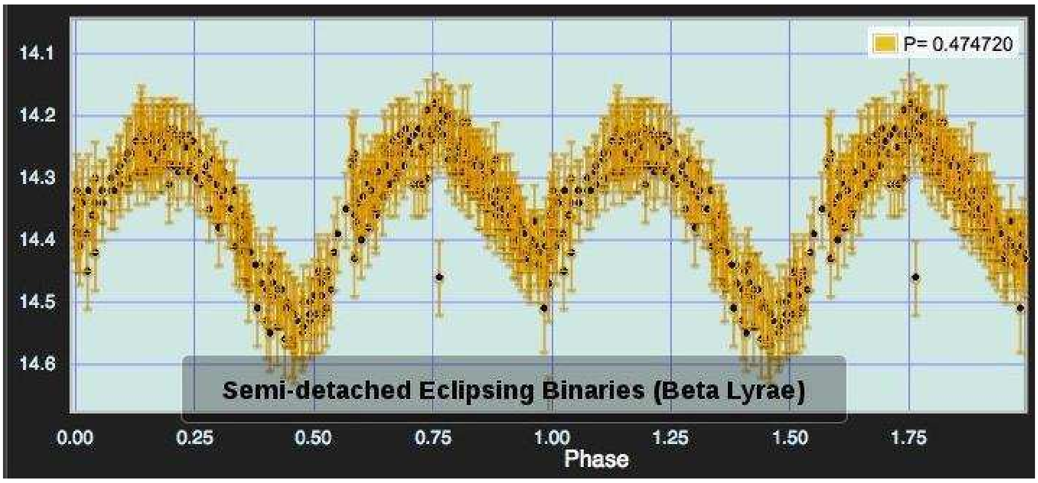

Figure 1.6: Light curve of semi detached eclipsing binary : Courtsey CRTS

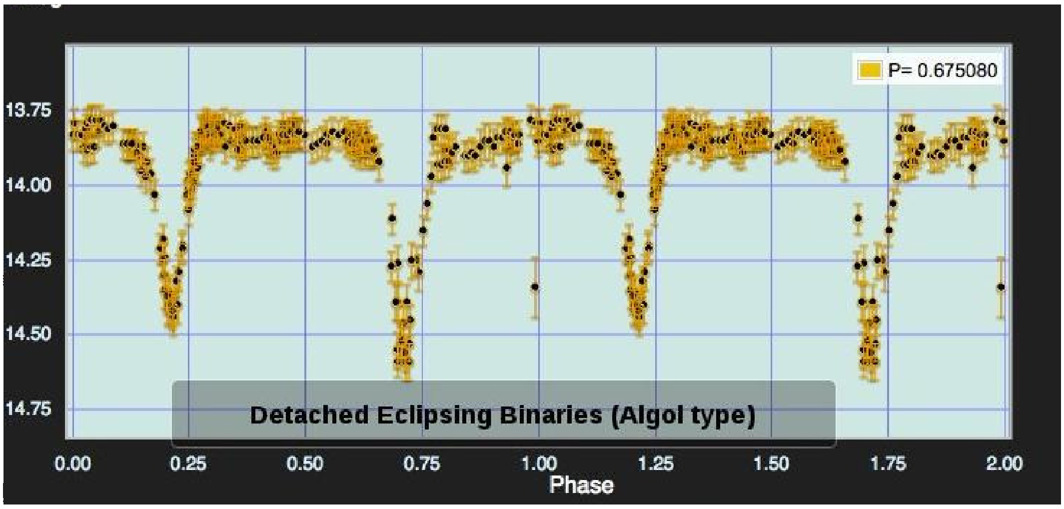

Figure 1.7: Light curve of detached eclipsing binary : Courtsey CRTS

1.2.14 X-ray variables

These are formed by close binary systems which are sources of strong variable X-ray emission, which can't be attributed to any other variable star mechanism and most often, they are optically variable too. The primary component is a hot compact object (white dwarf, neutron star, or a black hole). The X-ray emission is caused by matter falling onto the compact object or its accretion disc. The X-rays then irradiate the companion star causing a variety of effects such as bursts, spectral variations or even eclipses.

1.2.15 Unique variables

In addition to the variable-star types described above, rest of the variables belong to unclassified variables, which do not have a generalized behaviour.

1.3 Various period search methods

Efficiency of period search methods can be evaluated according [42] to the following criteria:

- Is the method based on the Fourier analysis or based on the folding process with trial periods?
- Is the method depend on prior information of the statistical distributions of time series?

- How much percentage of data is used by the method?
- Is the method addresses the non-harmonicity, multi-periodicity, drifts, trends etc., which may contribute to frequency dependent spectra?
- How the irregularities in the data spacing (especially long gaps) influence spectra?
- How the regularities in the input data (periodic gaps) influence spectra?
- How the method treats the gaps in phase distributions?
- How precisely the period estimates can be computed?
- Can the method be used for searching complex patterns on the phase diagram?
- Is it possible to use weights to take into account different quality of data points?
- How robust is the method against outliers in the data?
- Is the numerical computation allows easy programming and modifications?
- What is the average time, required for the analysis of one time series data?

It is found that designing a period search method, which will consider all the above criteria, will be extremely complicated. Most of the existing period search methods are based on a compromise between speed, flexibility, robustness and sensitivity of the algorithms.

Most of the approaches existing at present to extract frequencies from a variable star's time series data are listed at the Geneva University website[1]. Many of them are based on the Discrete Fourier Transform (DFT) or variations of it, which can be regarded as a correlation between the measured time series and trigonometric functions – sines and cosines, with frequency as the independent parameter. The consideration of cosine and sine to represent a two-dimensional Fourier vector defines the Fourier space, and the normalisation of the cosine and sine covariances provides the length of the Fourier vector to return the signal amplitude. A plot of amplitude versus frequency is termed as the amplitude spectrum of the time series. Peaks in an amplitude spectrum indicate frequencies where the data set correlates better with the trigonometric functions than elsewhere, and the idea of period detection is to assume that the highest peaks indicate signals produced by the star, whereas the lower peaks are due to random measurement errors and are frequently called

[1] `obs.unige.ch/~eyer/VSWG/tools.html`

noise. The simplest method to distinguish between signal and noise is to average the amplitudes over a certain frequency range and to compare the peak amplitude to this environmental mean (Breger et al. [17]). One of the potential drawbacks of employing such a signal-to-noise ratio is that it depends on the frequency range used for averaging and – more critically – on (yet) unresolved signal components hidden in the noise.

The problem of finding the appropriate noise level in the amplitude spectrum – or the degree of randomness in a time series – was addressed by several authors providing different solutions. Some introduce corrections to the DFT itself, such as the Lomb-Scargle Periodogram ([49], [70]) or the Date Compensated DFT ([34]), others apply statistical methods such as ANOVA analysis ([72],[73]). Several variations of Lomb-Scargle Periodogram exists, such as Generalized Lomb-Scargle Periodogram (**GLS**) introduced by Zechmeister [82], which is also a very useful implementation. Francois Mignard's FAMOUS(`ftp://ftp.obs-nice.fr/pub/mignard/Famous/` is a method, which uses a sinusoidal model to fit the data with the amplitude coefficients being either constant or a polynomial in time. The string-length method was introduced by Lafler and Kinman ([46]), which minimized the length of the light curve. A completely different way of period detection is the systematic examination of phased light curves modulo different periods and to determine the best-fitting period by minimising the intrinsic scatter of the phase plot. The most common formal representation is the Phase Dispersion Minimization (**PDM**) by Stellingwerf ([77]). This method does not require any initial assumption on the shape of the periodicity and works also for non-trigonometric signals.

There are some useful software packages like

- PERIOD04 (`www.univie.ac.at/tops/Period04/`)
- PERANSO (`www.tonnyvanmunster.ipage.com/peranso/downloads.htm`)
- AVE (`www.astrogea.org/soft/ave/aveint.htm`)
- Vstar (`www.citizensky.org/content/vstar`)
- VARTOOLS (`www.astro.princeton.edu/~jhartman/vartools/vartools1.202.tgz`)
- MUFRAN (`www.konkoly.hu/tifran`)

But they require human supervision and intervention in between, so that they become extremely time-consuming if applied to huge time series databases.

1.4 Problems with detected periods

Whatever be the period detection technique, if the detected period is exact, the intrinsic scatter of a phased light curve will attain a minimum and gives better light curves. Most of the methods described above are effective upto certain extend in finding the "true" or "exact" periods, but the final results are mixed with spurious, alias or harmonic periods with highly significant statistic. The irregular sampling can eliminate the aliasing and well defined statistic may eliminate spurious periods. The sub-harmonic averaging introduced in *PDM* by Stellingwerf([77]) in the latest version, can eliminate the wrong harmonic periods. But when a common method is applied to all data files in a database, under automation, the individual monitoring of time series data becomes time consuming and laborious and most of these problems persist in the output.

In the case of eclipsing binaries, the work by [25] shows that the improvement in the fifth decimal place in the period value, can give well defined phased light curve, atleast in some cases.

Several authors select few known (for eg: Huijse([41]) choose a subset of 3 types) types of variable stars from a database and achieved the high automation accuracy. But our attempt is to design a general method and use the total database for the period search on any type of variable stars.

This thesis work is an attempt to confirm the exact period of any type of variable stars, with the modified cubic spline method, when applied to huge databases, under automation. The detected exact period permits to identify the type and amount of variability and also to deduce many other important astrophysical parameters. The Fourth Variable Star Working Group meeting held at Geneva Observatory, Switzerland in 2005, specified about the period search benchmarks [33][2], which includes the format of time series data, various period search methods, classification methods and tools and documents. Several papers (Blomme([14]),Dubath([31]) and Debosscher[26]) state that pre-processing such as de-trending the time series and removing the outliers will improve the period detection. [26]

1.5 Sampling constraints and Period search range

For an ideal time series, the data should be sampled in a moderate way. The under sampling may cause to skip the real period and oversampling wastes CPU time. Even sampling causes aliasing (daily observation causes 1 cycles per day alias) and uneven sampling or gaps produce spurious effects. Most of the statistical time series

[2] `obswww.unige.ch/~eyer/VSWG/tools.html`

analysis tools developed without anticipating the non-uniform sampling as in the case of astronomical observations. As per Templeton M. [3], the total time span of the time series and the average sampling interval are the factors deciding the period search range and resolution. In astronomical time series, the observers have no control over the observation and has to deal with the obtained time series.

1.6 Conclusion

In this chapter, the scope and importance of variable star research and period analysis methods are discussed. Even with all constraints on astro-time series data, the data analyst has to recover exact information through time series analysis, which require statistical, numerical and computational skills are required, in addition to the domain knowledge from astrophysics. Also the automated period search methods are required in the modern era of peta-bytes of variable star data.

[3] aavso.org

Chapter 2

Parametric period search methods

2.1 Introduction

In this chapter, the popular period search method Lomb-Scargle periodogram (LSP) is derived in two different ways. First the LSP is derived as least squares trigonometric fitting, similar to the efforts by Scargle [70] and Zechmeister [82] and secondly, on the basis of discrete Fourier transform(DFT) and probability theory, as pointed out by Reegen [62]. Then the equations for generalised Lomb-Scargle periodogram (GLSP) and Significant Spectrum (SigSpec) are given. The three methods LSP, GLSP and SigSpec are used for the period search, and the results are given in chapter 4 and chapter 5. The SigSpec method is coded in FORTRAN and is given in Appendix A.1. The script for automated running of SigSpec is given in Appendix C.1 and C.2.

In order to analyse gaped and distorted astronomical time series data, the widely used method is the periodogram analysis. The periodogram is like an amplitude power spectrum, which gives the dominant frequencies (and hence periods) present in the unevenly spaced and gaped time series data. The periodogram was first introduced by Arthur Schuster [71] as an estimate of the spectral intensity of a signal. Later it was re-defined by Barning [11], Lomb [49], and then modified by Scargle [70] and is known as LSP. There exist many variations of LSP, which are used for period detection of variable stars and also for identifying exact signal frequencies in many types of time-varying processes.

2.2 Periodogram as least-squares sine fitting

Consider the time series t_k, y_k, ϵ_k, where t_k is the time [1], y_k is the apparent magnitude of the variable star, ϵ_k is the error in the magnitude measurement and $k = 1 \ldots N$, where N is the total number of measurements. Consider the most general oscillatory or periodic model function [2], which is also suitable for the periodic time series.

$$y(t_k) = A\cos(\omega t_k - \theta) \tag{2.1}$$

where $\omega = 2\pi f$ so that f is the frequency corresponding to the variable star's oscillations or variations (so that period T=$1/f$, A is the amplitude of variation and θ is the phase of the oscillation. By taking $a = A\cos\theta$ and $b = -A\sin\theta$, the model function eq.(2.1) becomes,

$$y(t_k) = a\cos\omega t_k + b\sin\omega t_k \tag{2.2}$$

Note that both the model functions eq.(2.1) and eq.(2.2), are mathematically same. The least-squares error $E(\omega)$, which is the squared difference between the data y_k and the model function $y(t_k)$ is,

$$E(\omega) = \sum_{k=1}^{N}[y_k - y(t_k)]^2 \tag{2.3}$$

has to be minimized for the best fit. For the minimum $E(\omega)$ the first partial derivatives should vanish and we have,

$$\frac{\partial E}{\partial a} = -\sum_{k=1}^{N} 2[y_k - y(t_k)]\cos\omega t_k \quad = \quad 0 \tag{2.4}$$

$$\frac{\partial E}{\partial b} = -\sum_{k=1}^{N} 2[y_k - y(t_k)]\sin\omega t_k \quad = \quad 0 \tag{2.5}$$

[1] Usually the observational astronomical time stamping t_k is in Julian date (JD) format as discussed in 1

[2] Zechmeister[82] introduced an additional offset c in the model $y(t_k) = a\cos\omega t_k + b\sin\omega t_k + c$. Instead of the offset c, the arithmetic mean $\bar{y}$ of the data may be substituted, then the mean subtracted data has to be used for computation of periods.

$$\sum_{k=1}^{N} y_k \cos \omega t_k = \sum_{k=1}^{N} y(t_k) \cos \omega t_k \tag{2.6}$$

$$\sum_{k=1}^{N} y_k \sin \omega t_k = \sum_{k=1}^{N} y(t_k) \sin \omega t_k \tag{2.7}$$

Substituting from eq.(2.2), eq.(2.6) and eq.(2.7) becomes,

$$\sum_{k=1}^{N} y_k \cos \omega t_k = \sum_{k=1}^{N} (a \cos \omega t_k + b \sin \omega t_k) \cos \omega t_k \tag{2.8}$$

$$\sum_{k=1}^{N} y_k \sin \omega t_k = \sum_{k=1}^{N} (a \cos \omega t_k + b \sin \omega t_k) \sin \omega t_k \tag{2.9}$$

These equations can be written as,

$$YC = aCC + bCS \tag{2.10}$$

$$YS = aCS + bSS \tag{2.11}$$

where the following notations[3] are used.

$$YC = \sum_{k=1}^{N} y_k \cos \omega t_k, YS = \sum_{k=1}^{N} y_k \sin \omega t_k \tag{2.12}$$

$$CC = \sum_{k=1}^{N} \cos^2 \omega t_k, SS = \sum_{k=1}^{N} \sin^2 \omega t_k \tag{2.13}$$

$$CS = \sum_{k=1}^{N} \cos \omega t_k \sin \omega t_k \tag{2.14}$$

The eq.(2.10) and eq.(2.11) can be written in matrix form as,

$$\begin{bmatrix} CC & CS \\ CS & SS \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} YC \\ YS \end{bmatrix} \tag{2.15}$$

Solving for a and b, we get,

$$a = \frac{YC.SS - YS.CS}{CC.SS - CS.CS} = \frac{YC.SS - YS.CS}{|D|} \tag{2.16}$$

$$b = \frac{YS.CC - YC.CS}{CC.SS - CS.CS} = \frac{YS.CC - YC.CS}{|D|} \tag{2.17}$$

[3]Similar notations are used by Barning[11], Lomb [49], Scargle[70], Zechmeister [82]

where

$$|D| = \begin{vmatrix} CC & CS \\ CS & SS \end{vmatrix} = CC.SS - CS^2 \tag{2.18}$$

The minimum error $E(\omega)$ from eq.(2.3) can be written as,

$$\begin{aligned} E(\omega) &= \sum_{k=1}^{N} [y_k - y(t_k)]\,[y_k - y(t_k)] &(2.19)\\ &= \sum_{k=1}^{N} [y_k - y(t_k)]\,y_k - \sum_{k=1}^{N} [y_k - y(t_k)]\,y(t_k) &(2.20)\end{aligned}$$

Using eq.(2.4) and eq.(2.5) the second term in the above equation should vanish and we get

$$\sum_{k=1}^{N} [y_k - y(t_k)]\,y(t_k) = 0 \tag{2.21}$$

Therefore eq.(2.20) becomes,

$$\begin{aligned} E(\omega) &= \sum_{k=1}^{N} [y_k - y(t_k)]\,y_k &(2.22)\\ &= \sum_{k=1}^{N} [y_k - (a\cos\omega t_k + b\sin\omega t_k)]\,y_k &(2.23)\\ &= YY - aYC - bYS &(2.24)\end{aligned}$$

where $YY = \sum_{k=1}^{N} y_k^2$. Substitute for a and b from eq.(2.16) and eq.(2.17), we get,

$$E(\omega) = YY - \left(\frac{YC^2.SS + YS^2.CC - 2YS.CS.YC}{|D|}\right) \tag{2.25}$$

The normalised periodogram is given by,

$$\begin{aligned} P(\omega) = \frac{YY - E(\omega)}{YY} &= \left(\frac{YC^2.SS + YS^2.CC - 2YS.CS.YC}{|D|\,.YY}\right) &(2.26)\\ &= \frac{1}{YY.\,|D|}\left[SS.YC^2 + CC.YS^2 - 2.CS.YC.YS\right] &(2.27)\end{aligned}$$

where $|D|$ is given by eq.(2.18). One essential property of periodogram is that it should be invariant [4] under time translation[70]; i.e. if the time t_k is replaced by

[4] This is also an inherent property of Fourier transform

$(t_k - \tau)$, the periodogram should remain unchanged, where τ is a suitable offset time. Thus to make eq. (2.27) time invariant and also in order to facilitate the statistical description of the LSP spectrum, we will choose τ such that

$$CS_\tau = 0 \tag{2.28}$$

After the introduction of τ, for every term, τ is used as subscript. Under the condition $CS_\tau = 0$, the eq.(2.27) reduces to,

$$P(\omega) = \frac{1}{YY_\tau . |D|}\left[SS_\tau . YC_\tau^2 + CC_\tau . YS_\tau^2\right] \tag{2.29}$$

and the eq. (2.18) becomes,

$$|D| = CC_\tau . SS_\tau \tag{2.30}$$

Substituting eq.(2.30) in eq.(2.29)

$$P(\omega) = \frac{1}{YY_\tau}\left[\frac{YC_\tau^2}{CC_\tau} + \frac{YS_\tau^2}{SS_\tau}\right] \tag{2.31}$$

$$P(\omega) = \frac{1}{\sum_{k=1}^{N} y_k^2}\left[\frac{\left[\sum_{k=1}^{N} y_k \cos\omega\left(t_k - \tau\right)\right]^2}{\sum_{k=1}^{N} \cos^2\omega\left(t_k - \tau\right)} + \frac{\left[\sum_{k=1}^{N} y_k \sin\omega\left(t_k - \tau\right)\right]^2}{\sum_{k=1}^{N} \sin^2\omega\left(t_k - \tau\right)}\right] \tag{2.32}$$

by using the notations from eq.(2.14). This is the same periodogram given by Scargle [70], except the normalisation factor $1/\left(\sum_{k=1}^{N} y_k^2\right)$ instead of 1/2. Also when time t_k is replaced by $(t_k - \tau)$, the time translation invariance is achieved as in Scargle [70]. Then expression for τ is given by eq.(2.35).

2.2.1 Derivation of time-shift τ

From (2.28) $CS_\tau = 0$ implies,

$$CS_\tau = \sum_{k=1}^{N} \cos\omega(t_k - \tau)\sin\omega(t_k - \tau) = 0 \tag{2.33}$$

$$\sum_{k=1}^{N}(\cos\omega t_k \cos\omega\tau + \sin\omega t_k \sin\omega\tau) \times (\sin\omega t_k \cos\omega\tau - \cos\omega t_k \sin\omega\tau) = 0 \tag{2.34}$$

Multiplying and simplifying, we get,

$$\tan 2\omega\tau = \frac{\sum_{k=1}^{N} \sin 2\omega t_k}{\sum_{k=1}^{N} \cos 2\omega t_k} \tag{2.35}$$

The effect of introducing time delay τ is that this pair of sinusoid would be mutually orthogonal at sample times t_k, and also adjusted for the potentially unequal powers of these two basis functions, to obtain a better estimate of the power at a frequency ω. The frequency ω with maximum $P(\omega)$ is taken as the strongest frequency component, corresponding to the exact period.

2.3 Discrete Fourier transform

In this section, we review the basics of Fourier analysis, Fourier coefficients, Fourier transform and Discrete Fourier transform (DFT) of a time series.

2.3.1 Fourier series and Fourier transform

Any finite single valued periodic function $f(t)$, which is either continuous or possesses a finite number of finite discontinuities can be represented as the sum of trigonometric functions as,

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right) \right] \tag{2.36}$$

This is known as Fourier series for $f(t)$. The coefficients a_0, a_n and b_n are known as Fourier coefficients and are given by

$$a_0 = \frac{2}{T} \int_0^T f(t) dt \tag{2.37}$$

$$a_n = \frac{2}{T} \int_0^T f(t) \cos\left(\frac{2\pi nt}{T}\right) dt \tag{2.38}$$

$$b_n = \frac{2}{T} \int_0^T f(t) \sin\left(\frac{2\pi nt}{T}\right) dt \tag{2.39}$$

Fourier series can be expressed in exponential form also as,

$$f(t) = \sum_{n=-\infty}^{+\infty} d_n e^{i\frac{2\pi nt}{T}} \tag{2.40}$$

where $2d_n = a_n - ib_n$, $d_0 = \frac{a_0}{2}$. Substituting for a_n and b_n, we get,

$$d_n = \frac{1}{T}\int_0^T f(t)e^{-i\frac{2\pi nt}{T}}dt. \tag{2.41}$$

Using eq.(2.41) in eq.(2.40) we get

$$f(t) = \sum_{n=-\infty}^{+\infty}\left[\frac{1}{T}\int_0^T f(t')e^{-i\frac{2\pi nt'}{T}}dt'\right]e^{i\frac{2\pi nt}{T}}. \tag{2.42}$$

Since $f(t)$ is periodic, the limits 0 to T can be replaced by $-T/2$ to $-T/2$, so that,

$$f(t) = \sum_{n=-\infty}^{+\infty}\left[\frac{1}{T}\int_{\frac{-T}{2}}^{\frac{T}{2}} f(t')e^{-i\frac{2\pi nt'}{T}}dt'\right]e^{i\frac{2\pi nt}{T}} \tag{2.43}$$

When time period $T \to \infty$, the pulse will not be repeated for a finite time. The pulse under this condition behaves like an isolated pulse. As $T \to \infty$, the frequency $\nu = \frac{1}{T} \to 0$. The frequency becomes infinitesimally small and may be written as $\Delta\omega$. Then, $\omega = \frac{2\pi n}{T}$ and $\Delta\omega = \frac{2\pi}{T}$.

$$f(t) = \sum_{n=-\infty}^{+\infty}\left[\frac{\Delta\omega}{2\pi}\int_{\frac{-T}{2}}^{\frac{T}{2}} f(t')e^{-i\omega t'}dt'\right]e^{i\omega t}. \tag{2.44}$$

In the limit $T \to \infty$, the sum represented by $\sum_n$ is replaced by an integral as the unit change in n produces infinitesimal changes in ω and the eq.(2.44) becomes,

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} d\omega \left[\int_{-\infty}^{+\infty} f(t')e^{-i\omega t'}dt'\right]e^{i\omega t} \tag{2.45}$$

This is known as Fourier integral and let the term inside the square bracket is denoted by $F(\omega)$. Then,

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{i\omega t}d\omega \tag{2.46}$$

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt \tag{2.47}$$

and $F(\omega)$ is the Fourier transform of $f(t)$.

2.3.2 Impulse function

The impulse function or Dirac function is defined such that,

$$\delta(x) = 0, x \neq 0 \tag{2.48}$$

$$\int_{-\infty}^{+\infty} \delta(x)\, dx = 1 \tag{2.49}$$

This $\delta(x)$ function can be used in case of peak-like functions. Some important properties of the $\delta(x)$ function are,

- *Shifting property*

$$\int_{-\infty}^{+\infty} \delta(x-a) f(x)\, dx = f(a) \tag{2.50}$$

- *Symmetry property*

$$\delta(x) = \delta(-x) \tag{2.51}$$

- *Scaling property*

$$\delta(ax) = \frac{1}{|a|}\delta(x) \tag{2.52}$$

2.3.3 Fourier transform (FT)

Time series can be viewed from two complementary standpoints:

- The time domain
- The frequency domain

The observer views and records the variable star magnitude variations in the time domain. But the mathematical analysis becomes simpler under frequency domain. This is true with most of the phenomena in the nature. Any complex signal can be fully described in either of these domains and we can go between the two domains with the help of Fourier transform.

Let $f(t)$ be arbitrary complex valued generalized function of a real argument t. Then its Fourier transform is defined as,

$$F(\nu) = \int_{-\infty}^{+\infty} f(t)\, e^{-i2\pi t\nu} dt \tag{2.53}$$

In general $F(\nu)$ is also complex valued generalized function, but the argument ν is still real. Corresponding inverse Fourier transform is defined as,

$$f(t) = \int_{-\infty}^{+\infty} F(\nu)\, e^{i2\pi\nu t} d\nu \tag{2.54}$$

Thus $f(t)$ are now described as integrals (or sums in case of discrete valued functions) over the peaks in $F(\nu)$. The Fourier transform is thus an algorithm to analyse peaks $F(\nu)$ and inverse Fourier transform synthesizes $f(t)$ from peaks.

In the case of variable star observations, the obtained data is in the form of sampled or discretised waveforms, and hence finite sums has to be used instead of integrals, then it is known as DFT.

2.3.4 Fourier coefficients of disrete-time Fourier series

A real, $N-$periodic, discrete-time signal $y[n]$ can be represented by a linear combination of the complex exponential signals

$$e^{j0\omega_0 n} = 1, e^{j\omega_0 n}, e^{j2\omega_0 n}, \cdots, e^{j(N-1)\omega_0 n}$$

as

$$y[n] = \sum_{k=0}^{N-1} c_k e^{jk\omega_0 n} \tag{2.55}$$

In these expressions,$j = \sqrt{-1}$, and the discrete-time fundamental frequency is $\omega_0 = \frac{2\pi}{N}$. This discrete-time Fourier series representation provides notions of frequency content of discrete-time signals, and it is very convenient for calculations involving linear, time-invariant (LTI) systems because complex exponentials are eigenfunctions of LTI systems. The complex coefficients $c_0, c_1, \cdots, c_{N-1}$can be calculated from the expression

$$\{c_k\} = \frac{1}{N} \sum_{n=0}^{N-1} y(n) e^{-jk\omega_0 n}, k = 0, 1, 2, \cdots, N-1 \tag{2.56}$$

The $c_k's$ are called the spectral coefficients of the signal $y[n]$. A plot of $|c_k|$ versus k is called the magnitude spectrum of y[n], and a plot of $arg|c_k|$ versus k is called the phase spectrum of $y[n]$. These plots, particularly the magnitude spectrum, provide a picture of the frequency composition of the signal. Notice that the spectral coefficients repeat as k is varied. In particular, for any value of k,

$$\begin{aligned}
\{c_{k+N}\} &= \frac{1}{N}\sum_{n=0}^{N-1} y(n)e^{-j(k+N)\omega_0 n} && (2.57)\\
&= \frac{1}{N}\sum_{n=0}^{N-1} y(n)e^{-jk\omega_0 n}e^{-jN\frac{2\pi}{N}n} && (2.58)\\
&= \frac{1}{N}\sum_{n=0}^{N-1} y(n)e^{-jk\omega_0 n} && (2.59)\\
&= c_k && (2.60)
\end{aligned}$$

The Fourier coefficients c_k can be split as

$$a(\omega) = \sum_{k=0}^{N-1} y_k \cos\omega t_k \tag{2.61}$$

$$b(\omega) = \sum_{k=0}^{N-1} y_k \sin\omega t_k \tag{2.62}$$

The corresponding variances are,

$$\left\langle a^2\right\rangle(\omega) = \left\langle y^2\right\rangle\left(\sum_{k=0}^{N-1}\cos\omega t_k\right)^2 \tag{2.63}$$

$$\left\langle b^2\right\rangle(\omega) = \left\langle y^2\right\rangle\left(\sum_{k=0}^{N-1}\sin\omega t_k\right)^2 \tag{2.64}$$

2.4 Time series and Probability theory

In this section, assume that the time series is generated by a Gaussian random process with population variance [5] $\left\langle y^2\right\rangle$. In time-domain, the characteristic equation for Gaussian distributed probability density function (here after PDF) $\phi(y_k)$ is,

$$\phi(y_k) = \frac{1}{\sqrt{2\pi\left\langle y^2\right\rangle}}e^{-\left(\frac{y_k^2}{2\left\langle y^2\right\rangle}\right)} \tag{2.65}$$

where $\sqrt{\left\langle y^2\right\rangle}$ is the standard deviation σ or $\left\langle y^2\right\rangle$ is the variance σ^2 as the usual notation. The discrete Fourier transform (DFT) of the time series produces a 2-

[5] This assumption is for using the definition of false alarm probability, where the null hypothesis is that time series data consists of white noise only

dimensional vector (a, b), in the frequency domain, known as Fourier coefficients. The Fourier coefficients can be interpreted as a periodic sequence of same periodicity as the discrete time series data. These Fourier coefficients are not orthogonal to each other or in other words they are generally correlated. Hence the PDF in the frequency domain will be bi-variate function of these coefficients. The Fourier coefficients with unit normalisation, in the frequency domain are,

$$a(\omega) = \sum_{k=1}^{N} y_k \cos \omega t_k \tag{2.66}$$

$$b(\omega) = \sum_{k=1}^{N} y_k \sin \omega t_k \tag{2.67}$$

Here normalizations are taken as unity and summation is changed from 1 to N for simplicity. Rotating the Fourier space coordinates by an angle θ_0 will transform the correlated Fourier coefficients a and b into uncorrelated coefficients α and β with zero covariance.

$$\alpha(\omega, \theta_0) = \sum_{k=1}^{N} y_k \left[\cos(\omega t_k - \theta_0)\right] \tag{2.68}$$

$$\beta(\omega, \theta_0) = \sum_{k=1}^{N} y_k \left[\sin(\omega t_k - \theta_0)\right] \tag{2.69}$$

the variance of α and β are,

$$\left\langle \alpha^2 \right\rangle (\omega, \theta_0) = \left\langle y^2 \right\rangle \sum_{k=1}^{N} \left[\cos(\omega t_k - \theta_0)\right]^2 \tag{2.70}$$

$$\left\langle \beta^2 \right\rangle (\omega, \theta_0) = \left\langle y^2 \right\rangle \sum_{k=1}^{N} \left[\sin(\omega t_k - \theta_0)\right]^2 \tag{2.71}$$

If two variables are uncorrelated, their covariance $\langle \alpha\beta \rangle$ should vanish. Thus for the covariance to be zero,

$$\langle \alpha\beta \rangle = \left\langle y^2 \right\rangle \sum_{k=1}^{N} \left[\cos(\omega t_k - \theta_0)\right] \times \sum_{l=1}^{N} \left[\sin(\omega t_l - \theta_0)\right] = 0 \tag{2.72}$$

Solving for θ_0, we get [6]

$$\tan 2\theta_0 = \frac{2\sum_{k=1}^{N} \sin \omega t_k \sum_{k=1}^{N} \cos \omega t_k}{(\sum_{k=1}^{N} \cos \omega t_k)^2 - (\sum_{k=1}^{N} \sin \omega t_k)^2} \tag{2.73}$$

Since (α, β) are the corresponding two uncorrelated Gaussian variables with variances $\langle \alpha^2 \rangle$ and $\langle \beta^2 \rangle$ respectively, their combined bi-variate Gaussian distributed PDF is given by,

$$\phi(\alpha, \beta) = \frac{1}{2\pi\sqrt{\langle \alpha^2 \rangle \langle \beta^2 \rangle}} e^{-\frac{1}{2}\left(\frac{\alpha^2}{\langle \alpha^2 \rangle} + \frac{\beta^2}{\langle \beta^2 \rangle}\right)} \tag{2.74}$$

Thus in order to make the Fourier coefficients uncorrelated of each other, we have to rotate Fourier space coordinates through an angle θ_0. This rotation transforms the Fourier coefficients (a, b) into (α, β) with zero covariance, as required. The rotation can be easily implemented as subtraction of θ_0 from θ. The Cartesian coordinates (α, β) are transformed into polar coordinates (A, θ) as area element transforms as $(d\alpha\ d\beta) \Rightarrow (A\ dA\ d\theta)$.

In order to obtain the transformation, consider the DFT of a time series y_k consists of both amplitude A and phase angle θ. To transform the eq.(2.74) from Cartesian coordinates into polar coordinates (A, θ), substitute,

$$\alpha = \frac{A}{2} \cos(\theta - \theta_0) \tag{2.75}$$

$$\beta = \frac{A}{2} \sin(\theta - \theta_0) \tag{2.76}$$

Then the Jacobian determinant, which does the transformation is,

$$\begin{vmatrix} \frac{\partial \alpha}{\partial A} & \frac{\partial \alpha}{\partial \theta} \\ \frac{\partial \beta}{\partial A} & \frac{\partial \beta}{\partial \theta} \end{vmatrix} \tag{2.77}$$

which become after substituting from eq. (2.76) in eq. (2.77),

$$\frac{A}{4} \cos^2(\theta - \theta_0) + \frac{A}{4} \sin^2(\theta - \theta_0) = \frac{A}{4}$$

Then eq. (2.74) in polar coordinates becomes,

$$\phi(A, \theta | \omega) = \frac{A}{4} \frac{1}{2\pi\sqrt{\langle \alpha^2 \rangle \langle \beta^2 \rangle}} e^{-\frac{1}{2}\left(\frac{A^2 \cos^2(\theta - \theta_0)}{4\langle \alpha^2 \rangle} + \frac{A^2 \sin^2(\theta - \theta_0)}{4\langle \beta^2 \rangle}\right)}$$

[6]Note that this is same as eq.(2.35), which shows that the time translation in time domain (Cartesian space) corresponds to rotation in the frequency domain (Fourier space).

Inside the parenthesis, $(A, \theta|\omega)$, the left side of $|$ symbol has random variables and right side constants.

On simplifying, we get,

$$\phi(A, \theta|\omega) = \frac{A}{8\pi\sqrt{\langle\alpha^2\rangle\langle\beta^2\rangle}} e^{-\frac{A^2}{8}\left(\frac{\cos^2(\theta-\theta_0)}{\langle\alpha^2\rangle}+\frac{\sin^2(\theta-\theta_0)}{\langle\beta^2\rangle}\right)} \tag{2.78}$$

where θ is the phase.

Here $\phi(A, \theta|\omega)$ is a bi-variate PDF of amplitude and phase, which is difficult to manipulate mathematically. Hence we will convert it into $\phi(A|\omega, \theta)$, which is a uni-variate function of amplitude by changing the normalization condition.

$$\iint \phi(A, \theta|\omega)\, dA d\theta = 1 \tag{2.79}$$

into

$$\int \phi(A|\theta, \omega)\, dA = 1 \tag{2.80}$$

The normalisation condition is that the PDF, when integrated over complete amplitude range (from 0 to ∞), has to be unity. Using the transformation equation from bi-variate to uni-variate as,

$$\phi(A|\theta, \omega) = \frac{\phi(A, \theta|\omega)}{\phi(\theta|\omega)} \tag{2.81}$$

Using the result $\int_0^\infty ye^{-(\kappa y^2)}dy = \frac{1}{2\kappa}$ and substitute eq.(2.78) in eq.(2.79), we get,

$$\begin{aligned} \phi(\theta|\omega) &= \int_0^\infty \phi(A, \theta|\omega)\, dA \qquad (2.82) \\ &= \frac{1}{2\pi}\left(\frac{\sqrt{\langle\alpha^2\rangle\langle\beta^2\rangle}}{\langle\beta^2\rangle\cos^2(\theta-\theta_0)+\langle\alpha^2\rangle\sin^2(\theta-\theta_0)}\right) \end{aligned}$$

$$\phi(A|\theta, \omega) = \frac{A\left[\langle\beta^2\rangle\cos^2(\theta-\theta_0)+\langle\alpha^2\rangle\sin^2(\theta-\theta_0)\right]}{4\left[\langle\alpha^2\rangle\langle\beta^2\rangle\right]} \times e^{-\frac{A^2}{8}\left(\frac{\cos^2(\theta-\theta_0)}{\langle\alpha^2\rangle}+\frac{\sin^2(\theta-\theta_0)}{\langle\beta^2\rangle}\right)} \tag{2.83}$$

Let,

$$R^2 = \frac{\langle\alpha^2\rangle\langle\beta^2\rangle}{\langle\beta^2\rangle\cos^2(\theta-\theta_0)+\langle\alpha^2\rangle\sin^2(\theta-\theta_0)} \tag{2.84}$$

then,

$$\phi\left(A|\theta,\omega\right) = \frac{A}{4R^2}e^{-\frac{A^2}{8R^2}} \tag{2.85}$$

Here ϕ is a function of random variable A only, since θ and ω are constants. Thus eq.(2.85) represents the probability density of amplitude for a particular frequency, at a constant phase. It will be interesting to note that, geometrically eq.(2.84) represents an ellipse[7] in polar coordinates, with semi-major axis $\sqrt{\langle\alpha^2\rangle}$ and semi-minor axis $\sqrt{\langle\beta^2\rangle}$. Thus from eq.(2.84),

$$R(\theta) = \sqrt{\frac{\langle\alpha^2\rangle\langle\beta^2\rangle}{\langle\beta^2\rangle\cos^2(\theta-\theta_0)+\langle\alpha^2\rangle\sin^2(\theta-\theta_0)}} \tag{2.86}$$

Here θ_0 represents the orientation of the ellipse in Fourier space, which in turn depends on the frequency. The normalisation of uncorrelated random variables α and β gives,

$$\alpha\left(\omega,\theta_0\right) = \sum_{k=1}^{N} y_k\left[\cos\left(\omega t_k-\theta_0\right)\right]$$

$$\beta\left(\omega,\theta_0\right) = \sum_{k=1}^{N} y_k\left[\sin\left(\omega t_k-\theta_0\right)\right] \tag{2.87}$$

with variances,

$$\left\langle\alpha^2\right\rangle\left(\omega,\theta_0\right) = \left\langle y^2\right\rangle\sum_{k=1}^{N}\left[\cos\left(\omega t_k-\theta_0\right)\right]^2$$

$$\left\langle\beta^2\right\rangle\left(\omega,\theta_0\right) = \left\langle y^2\right\rangle\sum_{k=1}^{N}\left[\sin\left(\omega t_k-\theta_0\right)\right]^2 \tag{2.88}$$

From these equations, the semi-major and semi-minor axes of the ellipse are

[7]For an ellipse $\frac{y^2}{a^2}+\frac{y^2}{b^2}=1$,put $x=r\cos\theta$ and $y=r\sin\theta$, then $r^2\left(\frac{b^2\cos^2\theta+a^2\sin^2\theta}{a^2b^2}\right)=1$ so that $r^2=\frac{a^2b^2}{b^2\cos^2\theta+a^2\sin^2\theta}$

chosen as,

$$\alpha_0(\omega, \theta_0) = \sqrt{\frac{\langle \alpha^2 \rangle}{\langle y^2 \rangle}} = \sqrt{\sum_{k=1}^{N} \cos^2(\omega t_k - \theta_0)}$$

$$\beta_0(\omega, \theta_0) = \sqrt{\frac{\langle \beta^2 \rangle}{\langle y^2 \rangle}} = \sqrt{\sum_{k=1}^{N} \sin^2(\omega t_k - \theta_0)} \tag{2.89}$$

Substituting these equations into eq.(2.86), we get,

$$\frac{1}{R^2} = \frac{1}{\langle y^2 \rangle} \left(\frac{\cos^2(\theta - \theta_0)}{\alpha_0^2} + \frac{\sin^2(\theta - \theta_0)}{\beta_0^2} \right) \tag{2.90}$$

θ_0, α_0 and β_0 denotes the 2-dimensional PDF for Gaussian noise in Fourier space and periodogram is based on these quantities.

2.4.1 Cumulative Distribution Function (CDF) and False Alarm Probability (FAP)

The Cumulative Distribution Function (CDF) of a continuous random variable is described as the probability that a variate Y takes on a value less than or equal to a number y and is defined as,

$$\text{Prob}(Y \leq y) = \int_{-\infty}^{y} f(t)dt$$

where f is the PDF. Hence CDF is obtained by integrating the eq. (2.85),

$$\Phi(A|\omega, \theta) = \int_0^A \phi(a|\omega, \theta)\, da = \int_0^A \frac{a}{4R^2} e^{-\frac{a^2}{8R^2}} da \tag{2.91}$$

let $x = a^2, dx = 2ada$, then upper limit becomes A^2,

$$\Phi(A|\omega, \theta) = \frac{1}{8R^2} \int_0^{A^2} e^{-\frac{x}{8R^2}} dx = 1 - e^{\left(-\frac{A^2}{8R^2}\right)}$$

Scargle in (1982) and later by Horne & Baliunas (1986) defined False-Alarm Probability assuming that, the power at a given frequency is exponentially distributed and also based on the null hypothesis that, the time series data is only due to white noise. If the frequency ω in the power spectrum is derived from normally distributed random noise (white noise), the probability $P(\omega)$ is of amplitude z or higher is,

$$\text{Prob}(P(\omega) \geq z) = e^{-z}$$

Thus z is the largest peak out of M_i independent frequencies. Then the probability that the particular frequency is smaller than z is $(1 - e^{-z})$. Then the probability that each M_i independent frequency is smaller than z is $(1 - e^{-z})^{M_i}$. Then the False-Alarm Probability (FAP) is defined as the probability that atleast one out of M_i independent peaks is expected to be larger than the given value z as $\Phi_{FAP} = 1 - (1 - e^{-z})^{M_i}$. Thus if false alarm probability is small, the null hypothesis is invalid, and the corresponding signal peak is not due to noise. Thus the false-alarm probability for an amplitude level to exceed a value A,

$$\Phi_{FAP}\left(A|\omega,\theta\right) = 1 - \left(1 - e^{-\left(\frac{A^2}{8R^2}\right)}\right) = e^{\left(-\frac{A^2}{8R^2}\right)} \tag{2.92}$$

This is the false-alarm probability for an amplitude level to exceed a value A, at phase θ and frequency ω.

2.4.2 Periodogram

The periodogram of a DFT amplitude spectrum is defined as the negative logarithm of false alarm probability,

$$\begin{aligned} P\left(A|\omega,\theta\right) = -\log_e \Phi_{FAP}\left(A|\omega,\theta\right) = \frac{A^2}{8R^2} \\ = \frac{A^2}{8\left\langle y^2\right\rangle}\left(\frac{\cos^2\left(\theta-\theta_0\right)}{\alpha_0^2} + \frac{\sin^2\left(\theta-\theta_0\right)}{\beta_0^2}\right) \end{aligned} \tag{2.93}$$

by using eq.(2.90). The Cartesian representation of eq.(2.93) is,

$$\begin{aligned} P\left(a,b|\omega\right) &= \frac{1}{2\left\langle y^2\right\rangle}\left(\frac{A^2\cos^2\left(\theta-\theta_0\right)}{4\alpha_0^2} + \frac{A^2\sin^2\left(\theta-\theta_0\right)}{4\beta_0^2}\right) \\ &= \frac{1}{2\left\langle y^2\right\rangle}\left(\frac{\alpha^2}{\alpha_0^2} + \frac{\beta^2}{\beta_0^2}\right) \end{aligned} \tag{2.94}$$

where substitution is made from eq.(2.76). Then,

$$P\left(\omega\right) = \frac{1}{2\left\langle y^2\right\rangle}\left[\frac{\left(\sum_{k=1}^{N} y_k\left[\cos\left(\omega t_k-\theta_0\right)\right]\right)^2}{\sum_{k=1}^{N}\cos^2\left(\omega t_k-\theta_0\right)} + \frac{\left(\sum_{k=1}^{N} y_k\left[\sin\left(\omega t_k-\theta_0\right)\right]\right)^2}{\sum_{k=1}^{N}\sin^2\left(\omega t_k-\theta_0\right)}\right] \tag{2.95}$$

where $\langle y^2 \rangle$ is the variance. This is the periodogram (θ_0 is given by eq. (2.73), which is re-defined by Scargle (1982), and the major difference is that here, the normalization factor is $\frac{1}{2\langle y^2 \rangle}$, whereas in Scargle definition the normalisation factor is $\frac{1}{2}$. Also we can see that the time translation invariance in the Cartesian space corresponds to the rotational invariance in Fourier space. In eq. (2.73), if θ_0 is replaced by $(\omega\tau)$, the time shift is obtained as in eq. (2.35). Once exact period or frequency is known, the next two important parameters are phase and amplitude of the signal.

2.5 Phase θ

Using the notation, $(\omega t_k - \theta) \equiv \phi_k$, the least square fit for amplitude A and phase θ is given by the condition

$$\frac{\partial}{\partial A} \sum_{k=1}^{N} \{y_k - A\cos\phi_k\}^2 = 0 \tag{2.96}$$

$$\frac{\partial}{\partial \theta} \sum_{k=1}^{N} \{y_k - A\cos\phi_k\}^2 = 0 \tag{2.97}$$

From eq.(2.96), we get,

$$2\sum_{k=1}^{N} \{y_k - A\cos\phi_k\} \times (-)\cos\phi_k = 0$$

$$\sum_{k=1}^{N} \{-y_k \cos\phi_k + A\cos^2\phi_k\} = 0$$

$$\sum_{k=1}^{N} y_k \cos\phi_k = A\sum_{k=1}^{N} \cos^2\phi_k \tag{2.98}$$

From eq.(2.97),

$$2\sum_{k=1}^{N} \{y_k - A\cos\phi_k\} \times A\sin\phi_k = 0$$

$$\sum_{k=1}^{N} y_k \sin\phi_k = A \sum_{k=1}^{N} \cos\phi_k \sin\phi_k \tag{2.99}$$

Dividing eq.(2.98) by eq.(2.99), we get,

$$\frac{\sum_{k=1}^{N} y_k \cos\phi_k}{\sum_{k=1}^{N} y_k \sin\phi_k} = \frac{\sum_{k=1}^{N} \cos^2\phi_k}{\sum_{k=1}^{N} \cos\phi_k \sin\phi_k}$$

Cross multiplying and simplifying, we get,

$$\sum_{k=1}^{N} y_k \cos\phi_k \sum_{l=1}^{N} \cos\phi_l \sin\phi_l - \sum_{k=1}^{N} y_k \sin\phi_k \sum_{l=1}^{N} \cos^2\phi_l = 0$$

$$\sum_{k,l=1}^{N} y_k \left\{\cos\phi_k \cos\phi_l \sin\phi_l - \sin\phi_k \cos^2\phi_l\right\} = 0$$

$$\sum_{k,l=1}^{N} y_k \cos\phi_l \left\{\cos\phi_k \sin\phi_l - \sin\phi_k \cos\phi_l\right\} = 0$$

$$\sum_{k,l=1}^{N} y_k \cos\phi_l \sin(\phi_l - \phi_k) = 0$$

substituting back the notation $\phi_k \equiv (\omega t_k - \theta)$ and $\phi_l \equiv (\omega t_l - \theta)$

$$\sum_{k,l=1}^{N} \{\sin\omega\,(t_l - t_k)\}\, y_k \cos(\omega t_l - \theta) = 0$$

which finally reduces to,

$$\sum_{k,l=1}^{N} \{\sin\omega\,(t_l - t_k)\}\, y_k \left[\cos\omega t_l \cos\theta + \sin\omega t_l \sin\theta\right] = 0$$

$$\sum_{k,l=1}^{N} \{\sin\omega\,(t_l - t_k)\}\, y_k \cos\omega t_l \cos\theta + \sum_{k,l=1}^{N} \{\sin\omega\,(t_l - t_k)\}\, y_k \sin\omega t_l \sin\theta = 0$$

$$\tan\theta = -\frac{\sum_{k,l=1}^{N}\{\sin\omega\left(t_l - t_k\right)\} y_k \cos\omega t_l}{\sum_{k,l=1}^{N}\{\sin\omega\left(t_l - t_k\right)\} y_k \sin\omega t_l}$$

$$\tan\theta = -\frac{\sum_{k,l=1}^{N}\{\sin\omega t_l \cos\omega t_k - \cos\omega t_l \sin\omega t_k\} y_k \cos\omega t_l}{\sum_{k,l=1}^{N}\{\sin\omega t_l \cos\omega t_k - \cos\omega t_l \sin\omega t_k\} y_k \sin\omega t_l}$$

$$\tan\theta = -\frac{\{\sum_{l=1}^{N}\sin\omega t_l \cos\omega t_l\}\sum_{k=1}^{N}\cos\omega t_k - \{\sum_{l=1}^{N}\cos^2\omega t_l\}\sum_{k=1}^{N}\sin\omega t_k}{\{\sum_{l=1}^{N}\sin^2\omega t_l\}\sum_{k=1}^{N}\cos\omega t_k - \{\sum_{l=1}^{N}\cos\omega t_l \sin\omega t_l\}\sum_{k=1}^{N}\sin\omega t_k}$$

replacing l index by k and apply -ve sign

$$\tan\theta = \frac{\{\sum_{k=1}^{N}\sin\omega t_k \cos\omega t_k\}\sum_{k=1}^{N}\cos\omega t_k - \{\sum_{k=1}^{N}\cos^2\omega t_k\}\sum_{k=1}^{N}\sin\omega t_k}{\{\sum_{k=1}^{N}\cos\omega t_k \sin\omega t_k\}\sum_{k=1}^{N}\sin\omega t_k - \{\sum_{k=1}^{N}\sin^2\omega t_k\}\sum_{k=1}^{N}\cos\omega t_k} \tag{2.100}$$

From which, the phase can be calculated.

2.6 Amplitude A

If frequency ω and phase θ is known, the amplitude can be calculated from eq.(2.98),

$$A = \frac{\sum_{k=1}^{N} y_k \cos\phi_k}{\sum_{k=1}^{N}\cos^2\phi_k} \tag{2.101}$$

or from eq.(2.99),

$$A = \frac{\sum_{k=1}^{N} y_k \sin\phi_k}{\sum_{k=1}^{N}\sin\phi_k \cos\phi_k} \tag{2.102}$$

where $\phi_k \equiv (\omega t_k - \theta)$.

2.7 Generalized Lomb-Scargle periodogram (GLSP)

Zechmeister(2009) introduced a variation of Lomb-Scargle periodogram as,

$$P\left(\omega\right) = \frac{1}{\sigma^2}\left[\frac{\left[\sum_{k=1}^{N} y_k \cos\omega\left(t_k - \tau\right)\right]^2}{\sum_{k=1}^{N}\cos^2\left(\omega t_k - \tau\right)} + \frac{\left[\sum_{k=1}^{N} y_k \sin\omega\left(t_k - \tau\right)\right]^2}{\sum_{k=1}^{N}\sin^2\omega\left(t_k - \tau\right)}\right] \tag{2.103}$$

where σ is the standard deviation and the normalisation factor is $1/\sigma^2$. The time shift τ is defined by,

$$\tan 2\omega\tau = \frac{\sum_{k=1}^{N} \sin 2\omega t_k - 2\sum_{k=1}^{N} \cos \omega t_k \sum_{k=1}^{N} \sin \omega t_k}{\sum_{k=1}^{N} \cos 2\omega t_k - \left(\sum_{k=1}^{N} \cos \omega t_k\right)^2 + \left(\sum_{k=1}^{N} \sin \omega t_k\right)^2} \tag{2.104}$$

2.8 Spectral Significance (SigSpec)

The principle of *SigSpec* (Significance Spectrum) is described by Reegen (2007). The method is based on the analytical solution for the frequency-domain Probability Density Function (PDF) of white noise, which depends on amplitude, frequency and phase. Instead of the common signal-to-noise ratio criterion, the spectral significance of a DFT amplitude spectrum is defined as,

$$Sig = \frac{N \log e}{\langle y^2 \rangle} \left[\left(\frac{a_{zm} \cos \theta_0 + b_{zm} \sin \theta_0}{\alpha_0} \right)^2 + \left(\frac{a_{zm} \sin \theta_0 - b_{zm} \cos \theta_0}{\beta_0} \right)^2 \right] \tag{2.105}$$

where $\langle y^2 \rangle$ is the variance and $\omega = 2\pi f$, where f is the frequency. The zero mean corrected Fourier coefficients are given by,

$$a_{zm} = \frac{1}{N} \sum_{k=1}^{N} y_k \left[\cos \omega t_k - \frac{1}{N} \sum_{l=1}^{N} \cos \omega t_l \right] \tag{2.106}$$

$$b_{zm} = \frac{1}{N} \sum_{k=1}^{N} y_k \left[\sin \omega t_k - \frac{1}{N} \sum_{l=1}^{N} \sin \omega t_l \right] \tag{2.107}$$

where α_0 and β_0 are given by,

$$\alpha_0 = \sqrt{\frac{2}{N^2} \left\{ N \sum_{k=1}^{N} \cos^2 (\omega t_k - \theta_0) - \left[\sum_{l=1}^{N} \cos (\omega t_l - \theta_0) \right]^2 \right\}} \tag{2.108}$$

$$\beta_0 = \sqrt{\frac{2}{N^2} \left\{ N \sum_{k=1}^{N} \sin^2 (\omega t_k - \theta_0) - \left[\sum_{l=1}^{N} \sin (\omega t_l - \theta_0) \right]^2 \right\}} \tag{2.109}$$

and θ_0 is given by,

$$\tan 2\theta_0 = \frac{N\sum_{k=1}^{N}\sin 2\omega t_k - 2\sum_{k=1}^{N}\cos\omega t_k \sum_{k=1}^{N}\sin\omega t_k}{N\sum_{k=1}^{N}\cos 2\omega t_k - \left(\sum_{k=1}^{N}\cos\omega t_k\right)^2 + \left(\sum_{k=1}^{N}\sin\omega t_k\right)^2} \tag{2.110}$$

From the output file produced by *SigSpec*, the period with maximum significance is chosen as the exact period. According to Reegen (2007), one of the advantage of *SigSpec* is that it takes into effect the phase factor, which permits to identify the most reliable period without any statistical bias. Through an iterative procedure, *SigSpec* detects the most significant frequency component and performs the corresponding pre-whitening to find the next dominant frequency. Thus *SigSpec* is also suitable for multi-periodic search, especially in Astro-seismological applications.

2.8.1 Normalisation factor

In the first LSP derivation the normalisation factor is $1/\left(\sum_{k=1}^{N} y_k^2\right)$ whereas the Scargle [70] normalisation factor is 1/2. In the second derivation of LSP, when the mean subtracted data is used, the normalisation factor is $1/2\left\langle y^2\right\rangle$. In literature[70],[11], we can see various forms for the normalisation factor and corresponding arguments that, which normalisation factor will correctly retrieve the signal from the noise. In the above two derivations, two theoretically possible normalisation factors are obtained.

2.9 Application of parametric methods

In order to show the failure of these parametric methods in some cases, we select the first star $005759+0034.7$ from the ASAS *a2perlc* database, which is an eclipsing binary with published period of 1.596059 days. The time series data is shown in Figure (2.1). The published light curve is shown in the Figure (2.2). The PDM / GLSP / SigSpec gave almost half of the published period and light curve from GLSP is shown in Figure (2.3). GLSP give a period of 0.797991 days, PDM give a period of 0.789295 days and SigSpec give a period of 0.798033530207 days. Hence all three methods could not find the exact period, and so these methods need improvement, while automating. The Θ statistic from PDM is shown in the Figure (2.4), the GLSP periodogram peak is shown in Figure (2.5) and the SigSpec spectral significance is shown in Figure (2.6).

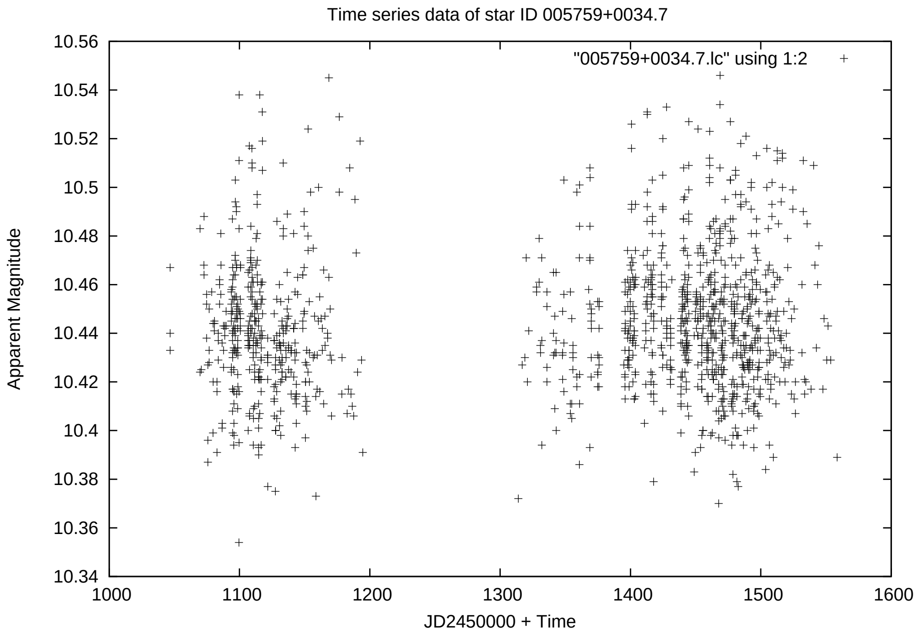

Figure 2.1: Time series data of $005759 + 0034.7$: Courtesy ASAS2

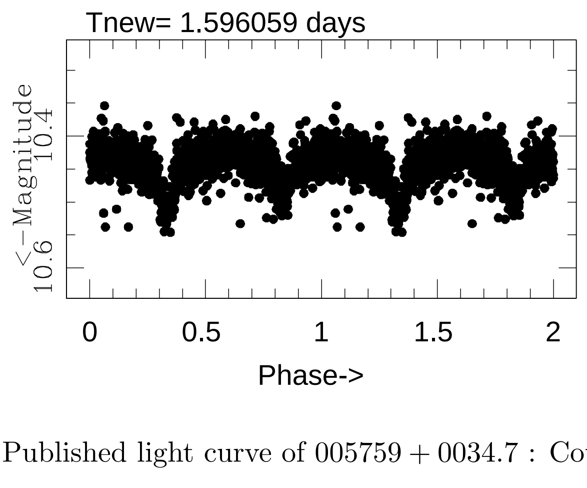

Figure 2.2: Published light curve of $005759 + 0034.7$: Courtesy ASAS2

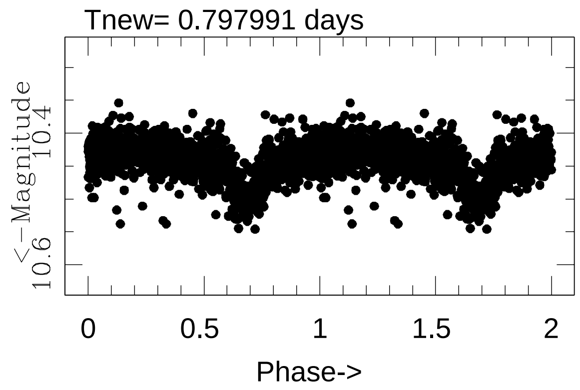

Figure 2.3: Light curve of $005759 + 0034.7$ obtained from GLSP. The light curve misses one peak, compared to the Figure 2.2. PDM and SigSpec give similar light curves.

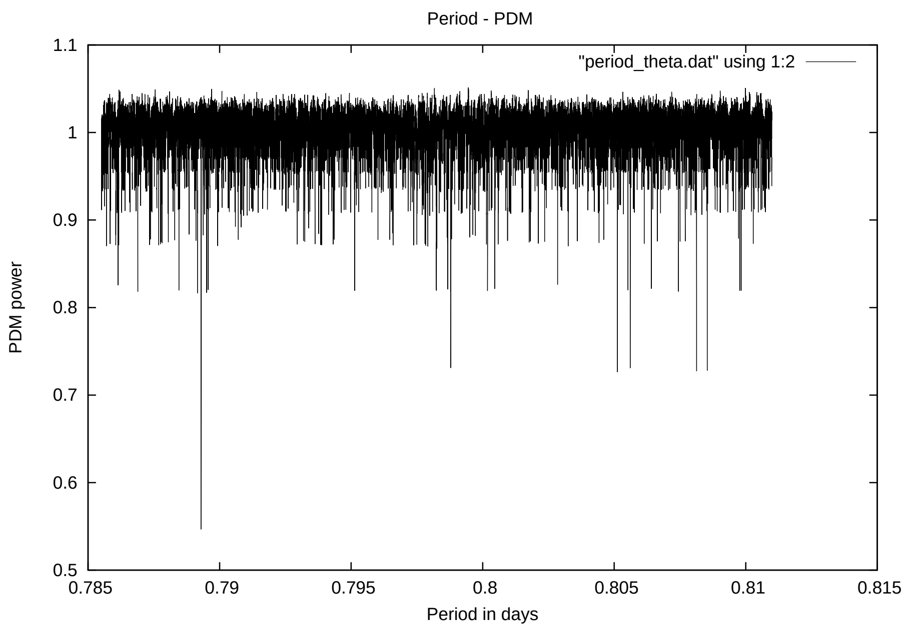

Figure 2.4: The PDM Θ statistic of $005759 + 0034.7$, the minimum is around 0.78 days

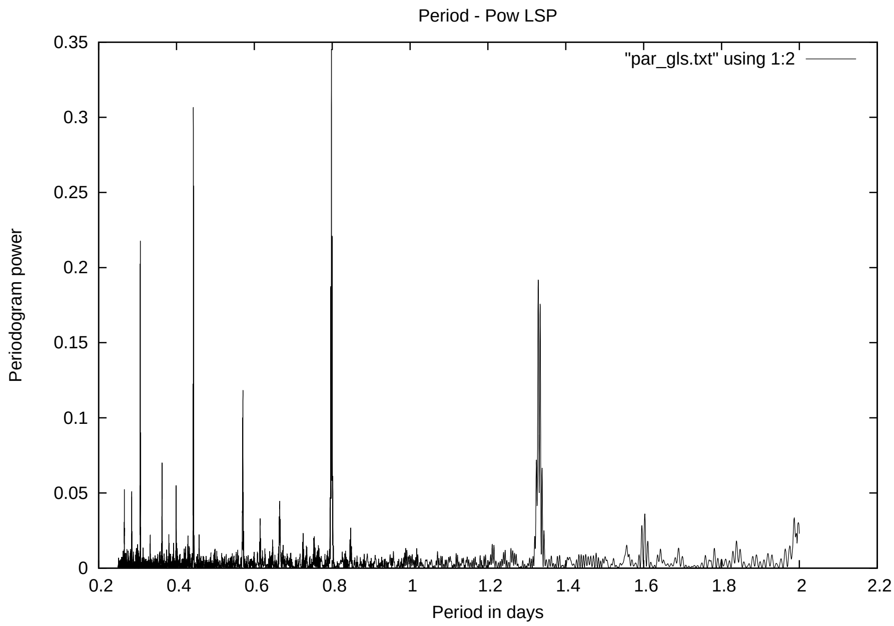

Figure 2.5: The GLSP periodogram of $005759 + 0034.7$, the peak is around 0.8 days

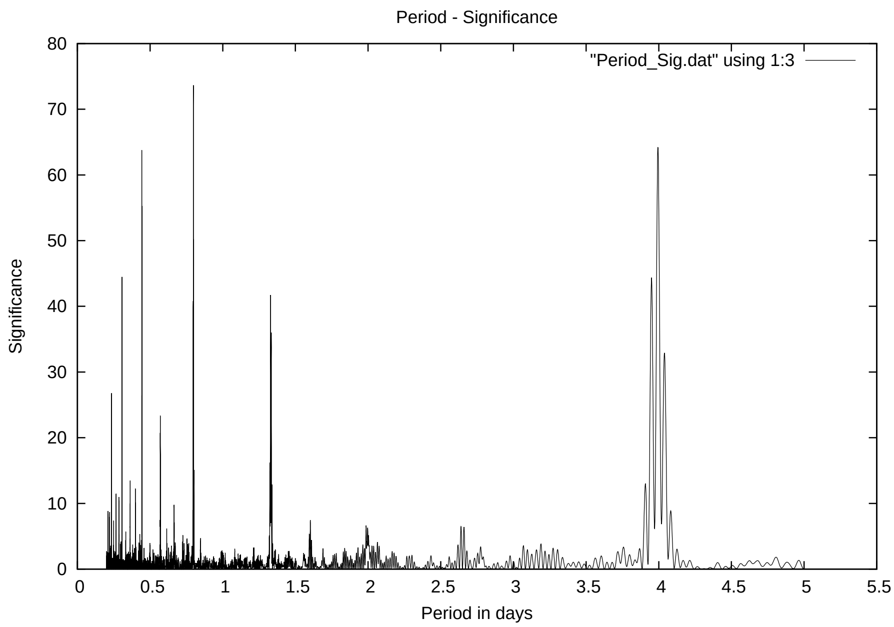

Figure 2.6: The SigSpec Significance of $005759 + 0034.7$, the maximum is around 0.8 days

2.10 Conclusion

In this chapter, the LSP is derived in two ways, first as Least squares trigonometric fitting and secondly on the basis of discrete Fourier transform(DFT) and probability theory. Then the features of GLSP and SigSpec methods are compared. Finally the results based on LSP, GLSP and SigSpec on sample data are compared, which shows the need for improvement in period detection. The SigSpec method is coded in FORTRAN. The GLSP FORTRAN program can be downloaded from (http://www.astro.physik.uni-goettingen.de/ zechmeister/gls.php) and *SigSpec* C program can be downloaded from (http://homepage.univie.ac.at/peter.reegen/download.html).

Chapter 3

Non-Parametric period search methods

3.1 Introduction

In this chapter, another two popular period search methods

- Phase dispersion minimisation (PDM),
- Method based on cubic spline (CS) interpolation,

are discussed. The principle of both methods are almost similar, in the sense that, both tries to draw a smooth mean curve, through the phase-folded light curve on an assumed period and the period corresponding to the minimum scattered light curve is chosen as the best possible period. Finally the cubic spline interpolation method is modified with unevenly spaced knots and used for period confirmation. The modified cubic spline (MCS) method is coded in FORTRAN and is given in Appendix B.1. The script for automated running of MCS is given in Appendix C.3

3.2 Phase dispersion minimization (PDM)

According to Stellingwerf [77], in the PDM method, the time series data (t_k, y_k) with N observations, is folded over a trial period and this phased data is divided into several bins (optimum 10 bins). It is a least square fit to the mean curve, which is determined by the data. The total variance of the data is given by,

$$\sigma^2 = \frac{\sum_{k=1}^{N}(y_k - \bar{y})^2}{N-1} \tag{3.1}$$

Here the mean $\bar{y} = \frac{1}{N}\sum_{k=1}^{N} y_k$. If M is the number of bins, then the variances in each bin ${s_j}^2, j = 1, \ldots, M$ are calculated as

$$s_j^2 = \frac{\sum_{i=1}^{n_j}(y_i - \bar{y})^2}{n_j - 1} \tag{3.2}$$

where n_j is the number of data in j^{th} bin and bin mean $\bar{y} = \frac{1}{n_j}\sum_{k=1}^{n_j} y_k$. The overall variance for all the bins is given by,

$$S^2 = \frac{\sum_{j=1}^{M}(n_j - 1){s_j}^2}{\sum_{j=1}^{M}(n_j - M)} \tag{3.3}$$

where S is s_j summed over M bins. The approximate value of period is obtained by minimizing the statistic, over a fixed number of data points. The PDM statistic Θ is the ratio between the sum of the bin variances and the total variance of the data.

$$\Theta = \frac{S^2}{\sigma^2} \tag{3.4}$$

The Θ value is found for trial periods continuously in a range, as in other methods and the period corresponding to minimum value of Θ is taken as the best period. Θ corresponds to the dispersion along the mean curve through the phased data. This method is also known as 'Fourierogram'.

In order to speed up the computation, the PDM method initially runs through a rough cut values and then the same process is repeated around the approximate period, with a finer spacing and by including the whole data. Thus more accurate period is obtained by minimising the same statistic Θ. The PDM software can be downloaded from (stellingwerf.com/rfs-bin/index.cgi?action=PageView&id=34).

3.3 Cubic spline interpolation

The mathematical concept of cubic spline interpolation is adopted from the engineer's tool, which is used to draw smooth curve through a number of points, with any complex pattern. The spline consists of weights attached to a flat surface at the points to be connected. A flexible strip is then bent across each of these weights, which produces a smooth curve. In mathematical spline, the weights are the coefficients of the cubic polynomial, used to interpolate the numerical data. These coefficients bend the line, so that it passes smoothly through all the points. The basic idea is to fit the data with a piecewise polynomial function of degree three.

3.3.1 Linear Spline

For a tabulated function $f_k \equiv f(t_k), k = 0, \dots N$ a spline is a piecewise polynomial between each pair of points, whose coefficients are determined with the help of neighboring points. The contributions from nearby points are included in order to guarantee global smoothness in the interpolated function, up to some order of derivatives. If there are $N+1$ points, there are N intervals and out of these, consider one particular interval (t_k, t_{k+1}). The Lagrange's piecewise linear interpolation formula in that interval is [80],

$$f = Af_k + Bf_{k+1} \tag{3.5}$$

where

$$A \equiv \frac{t_{k+1} - t}{t_{k+1} - t_k} \tag{3.6}$$

$$B \equiv 1 - A = \frac{t - t_k}{t_{k+1} - t_k} \tag{3.7}$$

In the case of linear spline, if all the piecewise polynomials are combined for all intervals, the fit is poor because, the first derivatives of the interpolating polynomial functions need not be continuous at the piece boundaries.

3.3.2 Natural cubic spline

The aim of cubic spline interpolation is to get an appropriate polynomial interpolation function, which is continuous and both the first and second derivatives are continuous. Thus cubic spline interpolation will give a smoother function better than piecewise linear or quadratic interpolations.

Consider a function $f(t)$, which is tabulated at $N+1$ points such that $f_k = f(t_k)$, where $k = 0, 1, \dots, N$. Let the values for the functions second derivatives $f_k^{''}$ are known. Then within each interval (t_k, t_{k+1}), a cubic polynomial term is added on the right-hand side of eq.(3.5), whose second derivative varies linearly from a value $f_k^{''}$ on the left, to a value $f_{k+1}^{''}$ on the right. Such a function can be constructed by replacing eq. (3.5) by[80],

$$f = Af_k + Bf_{k+1} + Cf_k^{''} + Df_{k+1}^{''} \tag{3.8}$$

where A and B are given in eq.(3.6) and eq.(3.7) and C and D are given by,

$$C \equiv \frac{1}{6}(A^3 - A)(t_{k+1} - t_k)^2 \tag{3.9}$$

$$D \equiv \frac{1}{6}(B^3 - B)(t_{k+1} - t_k)^2 \tag{3.10}$$

Since A and B are linearly dependent on t, then C and D (through A and B) have cubic $t-$dependence. Also $f^{''}$ is the second derivative of the new interpolating polynomial. By taking the first derivative of eq. (3.8) with respect to t, and substituting for $\frac{dA}{dt}, \frac{dB}{dt}, \frac{dC}{dt}$ and $\frac{dD}{dt}$ we get,

$$\frac{df}{dt} = \frac{f_{k+1} - f_k}{t_{k+1} - t_k} - \frac{3A^2 - 1}{6}(t_{k+1} - t_k)f_k^{''} + \frac{3B^2 - 1}{6}(t_{k+1} - t_k)f_{k+1}^{''} \tag{3.11}$$

and the second derivative is,

$$\frac{d^2 f}{dt^2} = Af_k^{''} + Bf_{k+1}^{''} \tag{3.12}$$

Since $A = 1$ at t_k and $A = 0$ at t_{k+1} , and $B = 0$ at t_k and $B = 1$ at t_{k+1}, eq.(3.12) shows that $f^{''}$ is the tabulated second derivative, and also that the second derivative will be continuous across the boundary between two intervals (t_{k-1}, t_k) and (t_k, t_{k+1}). The most important idea of a cubic spline is to ensure the continuity across the boundaries, by matching the first derivative (slope) $f^{'}(t_k)$ and the second derivative (curvature) $f_k^{''}$. The required equations are obtained by setting eq. (3.11) evaluated for $t = t_k$ in the interval (t_{k-1}, t_k) and equating to the same equation evaluated for $t = t_k$, but in the interval (t_k, t_{k+1}). This gives,

$$\begin{aligned} \frac{t_k - t_{k-1}}{6} y^{''}(t_{k-1}) + \frac{t_{k+1} - t_{k-1}}{3} y^{''}(t_k) + \frac{t_{k+1} - t_k}{6} y^{''}(t_{k+1}) = \\ \frac{y(t_{k+1}) - y(t_k)}{t_{k+1} - t_k} - \frac{y(t_k) - y(t_{k-1})}{t_k - t_{k-1}} \end{aligned} \tag{3.13}$$

for $k = 1, 2, \ldots, N-1$. These are $(N-1)$ linear equations in the $(N+1)$ unknowns $f_i^{''}, i = 0, 1, \ldots, N$. For having $(N+1)$ unknowns and a unique solution, the boundary conditions are set at t_1 and t_N by equating both $f(t_1)$ and $f(t_N)$ equal to zero, giving the natural cubic spline, which has zero second derivatives on both of its boundaries. Thus the solution for $f_k^{''}$, where $k = 0, 1, \ldots, N$ is obtained and this can be substituted back into eq. (3.8) to give the cubic interpolation formula in each interval (t_k, t_{k+1}).

3.3.3 B-splines

Instead of using the functional form eq. (3.8) and also using the constraints ($f_k^{'}$ continuous at interval boundaries) to solve for $f_k^{''}$ for getting the natural interpolating splines, it is possible to use a set of piecewise cubic polynomials defined on some sub-interval of $t_0), t_1, \ldots t_N$, which are by construction, continuous upto the second derivative at the boundaries of intervals. They would form a set of basis functions, since linear combinations of these functions would also satisfy the continuity properties at the boundaries between adjacent intervals.

A B-spline or basis spline [23] is a piecewise polynomial function of degree k in the variable t. It is defined over a range $(t_0 \leq t \leq t_N)$, $N = k + 2$. The points t_i, where the values of the function are tabulated for $f_i = f(t_i)$ are known as knots or break-points and the knots must be in ascending order. Each piece of the function is a polynomial of degree k between and including adjacent knots. A B-spline is a continuous function at the knots and when all knots are distinct, its derivatives are also continuous up to the derivative of degree $k - 1$.

For any given set of knots, the B-spline is unique. The usefulness of B-splines lies in the fact that any spline function of degree k on a given set of knots can be expressed as a linear combination of B-splines. Now to construct cubic spline over the whole range $[t_0, t_N]$, it is only needed to match the sum of the basis functions with tabulated values of f_i at the interpolating nodes $t_i, i = 0, 1, \ldots, N$. The B-splines are the basis functions that satisfy the specified continuity conditions. For the whole range interpolation, this can be achieved by defining $B_0(t)$ as,

$$B_0(t) = \begin{cases} 0, \quad t \leq (t_0 - 2h) \\ \frac{1}{6}(2h + (t - t_0))^3, \quad (t_0 - 2h) \leq t \leq (t_0 - h) \\ \frac{2h^3}{3} - \frac{1}{2}(t - t_0)^2(2h + (t - t_0)), \quad (t_0 - h) \leq t \leq t_0 \\ \frac{2h^3}{3} - \frac{1}{2}(t - t_0)^2(2h - (t - t_0)), \quad t_0 \leq t \leq (t_0 + h) \\ \frac{1}{6}(2h - (t - t_0))^3, \quad (t_0 + h) \leq t \leq (t_0 + 2h) \\ 0, \quad t \geq t_0 + 2h \end{cases} \tag{3.14}$$

where $h = (t_{k+1} - t_k) = (t_N - t_0)/N$ is the width between interpolating nodes (here assumed to be equal). Thus B_0 has non-zero values over four intervals and satisfy the continuity conditions at the boundaries of the intervals. i.e. $B_0, B_0^{'}$ and $B_0^{''}$ are continuous at $-2h, -h, 0, h$ and $2h$. If the $B_0(t)$ functions are shifted to the right by k nodes, we will get $B_k(t) = B_0(t - kh + t_0)$.

The cubic spline function, $S_3(t)$ for the whole interval $[t_0, t_N]$ is written as the linear combination of the B_k's,

$$S_3(t) = \sum_{k=-1}^{N+1} c_k B_k(t) \tag{3.15}$$

The sum is from -1 to $N+1$, since B_{-1} is non-zero in the interval (t_0, t_1), and B_{N+1} is also non-zero in the interval (t_{N-1}, t_N). In order to uniquely define a cubic spline, 4 conditions (coefficients) have to be specified and hence in the whole interval $[t_0, t_N]$, there are N intervals and total of $4N$ conditions are to be specified. The continuity conditions are automatically satisfied in the $(N-1)$ interior points, since the B_k's satisfy the continuity conditions (i.e. $3(N-1)$ conditions). The other requirement is that S_3 must match the tabulated points, i.e. $S_3(t_k) = f(t_k)$ for $k = 0, 1, \ldots, N$.i.e. ($N+1$ conditions). Now there are two more unspecified conditions, which are obtained by taking $S_3^{''}(t_0) = S_3^{''}(t_N) = 0$ (i.e.,the curvature equals zero at the boundaries, which is the condition for natural spline). Now $S_3(t)$ at t_k is,

$$S_3(t_k) = c_{k-1}B_{k-1}(t_k) + c_k B_k(t_k) + c_{k+1}B_{k+1}(t_k) + c_{k+2}B_{k+2}(t_k) = f(t_k) \tag{3.16}$$

All other B_k's are zero. From the definitions of $B_k(t)$ and $B_0(t)$, we get,

$$\begin{aligned} B_k(t_k) = B_0(t_0) &= \frac{2h^3}{3} && (3.17) \\ B_{k-1}(t_k) = B_0(t_0 + h) &= \frac{h^3}{6} && (3.18) \\ B_{k+1}(t_k) = B_0(t_0 - h) &= \frac{h^3}{6} && (3.19) \\ B_{k+2}(t_k) = B_0(t_0 - 2h) &= 0 && (3.20) \end{aligned}$$

Substituting into eq.(3.16), the recurrence relation for the coefficients c_k is,

$$c_{k-1} + 4c_k + c_{k+1} = \frac{6}{h^3} f(t_k) \tag{3.21}$$

for $k = 0, 1, \ldots, N$. At the boundaries, let $S_3^{''}(t_0) = S_3^{''}(t_N) = 0$. Differentiating $S_3(t)$ gives,

$$S_3^{''}(t) = \sum_{k=-1}^{N+1} c_k B_k^{''}(t) \tag{3.22}$$

To find the second derivatives of B_k, differentiate B_0 twice and we get

$$B_0^{''}(t) = \begin{cases} 0, & t \leq (t_0 - 2h) \\ 2h + (t - t_0), & (t_0 - 2h) \leq t \leq (t_0 - h) \\ -2h - 3(t - t_0), & (t_0 - h) \leq t \leq t_0 \\ -2h + 3(t - t_0), & t_0 \leq t \leq (t_0 + h) \\ 2h - (t - t_0), & (t_0 + h) \leq t \leq (t_0 + 2h) \\ 0, & t \geq t_0 + 2h \end{cases} \tag{3.23}$$

Hence,

$$\begin{aligned} 0 = S_3^{''}(t_0) &= c_{-1}B_{-1}^{''}(t_0) + c_0 B_0^{''}(t_0) + c_1 B_1^{''}(t_0) + c_2 B_2^{''}(t_0) \\ &= c_{-1}B_0^{''}(t_0 + h) + c_0 B_0^{''}(t_0) + c_1 B_0^{''}(t_0 - h) \\ &= hc_{-1} - 2hc_0 + hc_1 \\ &= c_{-1} - 2c_0 + c_1 \end{aligned} \tag{3.24}$$

From eq.(3.16),

$$c_{-1} + 4c_0 + c_1 = \frac{6}{h^3} f(t_0) \tag{3.25}$$

Subtract (3.24) from (3.25), we get,

$$c_0 = \frac{1}{h^3} f(t_0) \tag{3.26}$$

Similarly we can find

$$c_N = \frac{1}{h^3} f(t_N) \tag{3.27}$$

The matrix equation to solve for the coefficients $c_0, \ldots, c_N$, becomes
$$\begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 1 & 4 & 1 & \ldots & 0 & 0 & 0 \\ 0 & 1 & 4 & \ldots & 0 & 0 & 0 \\ . & . & . & \ldots & . & . & . \\ 0 & 0 & 0 & \ldots & 4 & 1 & 0 \\ 0 & 0 & 0 & \ldots & 1 & 4 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ . \\ c_{N-2} \\ c_{N-1} \\ c_N \end{bmatrix} = \frac{1}{h^3} \begin{bmatrix} f(t_0) \\ 6f(t_1) \\ 6f(t_2) \\ . \\ 6f(t_{N-2}) \\ 6f(t_{N-1}) \\ f(t_N) \end{bmatrix}$$
This set of equations is tri-diagonal and can be solved in $O(N)$ operations by the tridiagonal algo-

rithm. Final two coefficients to completely determine S_3 are,

$$c_{-1} = 2c_0 - c_1 \tag{3.28}$$

$$c_{N+1} = 2c_N - c_{N-1} \tag{3.29}$$

3.3.4 Modified cubic spline (MCS) interpolation method

This section introduces cubic spline interpolation with unevenly spaced knots (modified cubic spline (MCS) interpolation) for confirming the exact period of variable star's oscillation or rotation. Given the values y_k for t_k, $k = 1, 2, \ldots N$, using the cubic splines $B_i(t)$ we can construct an interpolated function $S(t)$ as follows:

$$S(t) = \sum_{i=1}^{n_s} c_i B_i(t) \tag{3.30}$$

where n_s is the number of splines used and c_i are the undetermined coefficients. In order to construct the cubic splines, the knot points τ_i, $i = 1, 2 \ldots n_s + 4$ has to be chosen so that $\tau_{i+1} > \tau_i$. In our case, corresponding to the phase folded data, the interval $[0, 1]$ are divided into n unequal intervals in general such that $\tau_4 = 0$, $\tau_{n_s+4} = 1$, $\tau_3 = -\tau_5$, $\tau_2 = -\tau_6$ and $\tau_1 = -\tau_7$. Given the knot points, the splines can be calculated using the following general equations:

$$B_i(t) = \begin{cases} 0, \quad t < \tau_i \\ -(\tau_{i+4} - \tau_i)\left(\alpha_{1,i}(\tau_i - t)^3\right), \quad \tau_i \le t < \tau_{i+1} \\ -(\tau_{i+4} - \tau_i)\left(\alpha_{1,i}(\tau_i - t)^3 - \alpha_{2,i}(\tau_{i+1} - t)^3\right), \quad \tau_{i+1} \le t < \tau_{i+2} \\ (\tau_{i+4} - \tau_i)\left(\alpha_{4,i}(\tau_{i+3} - t)^3 + \alpha_{5,i}(\tau_{i+4} - t)^3\right), \quad \tau_{i+2} \le t < \tau_{i+3} \\ (\tau_{i+4} - \tau_i)\left(\alpha_{5,i}(\tau_{i+4} - t)^3\right), \quad \tau_{i+3} \le t < \tau_{i+4} \\ 0, \quad t \ge \tau_{i+4} \end{cases} \tag{3.31}$$

where the coefficients $\alpha_{\nu,i}$, $\nu = 1, 2, \ldots, 5$ and $i = 1, 2, \ldots, n_s$ are given by,

$$\alpha_{\nu,i} = \prod_{\beta \neq 6-\nu}^{5} \frac{1}{(\tau_{i+5-\beta} - \tau_{i+\nu-1})} \tag{3.32}$$

To confirm the period, we use cubic B-spline to represent the light curve as,

$$y(t) = \sum_{\nu=1}^{n_s} c_\nu B_\nu(t) \tag{3.33}$$

where n_s is the number of splines used to interpolate the light curve and c_ν's are the undetermined coefficients. These coefficients can be determined by minimizing the dispersion relation $\sum_k [y(t_k) - y_k]^2 = 0$, i.e. the coefficients c_μ's are such that,

$$\frac{\partial}{\partial c_\mu} \sum_k [y(t_k) - y_k]^2 = 0, \mu = 1, 2, \ldots, n_s \tag{3.34}$$

Using (3.33) in (3.34), we get,

$$\sum_\nu M_{\mu\nu} c_\nu = N_\mu \tag{3.35}$$

where,

$$M_{\mu\nu} = \sum_k B_\mu(t_k) B_\nu(t_k)$$

and,

$$N_\mu = \sum_k B_\mu(t_k) y_k$$

By solving the linear algebraic eq. (3.35), the coefficients c_μ's are obtained.

3.3.5 Application of modified cubic spline analysis

The statistical, non-parametric cubic spline method with equally spaced knots was introduced for period searching and/or confirming the exact period, by Akerlof et.al.(1994). We have modified it to unequally spaced knots as follows. The data is folded over the period, as in the PDM method and the phased data is divided into certain number of bins. The number of bins is varying from 2 to 12, which depends on the number of data points in the time series data. If the number of data is 10 or less, the number of bins is 2 and if the number of data is greater than 1000, the number of bins is 10. If the number of data is in between, the number of bins is assigned accordingly. This kind of implementation makes it sure that the bins never become empty, even though there are gaps in the folded data.

After fixing the number of bins, the overall average of the data and the bin average of the data is calculated and according to their difference, the number of knots in each bin is selected. The difference will have larger value at the peaks, which is scaled to the nearest integer value and taken as the number of knots in the bin. This procedure produces unequally spaced knots, with more number of knots around the peaks and less number of knots at the flatter regions.

For solving the linear algebraic equation (3.35), we have used the subroutine from LAPACK [1].

[1] LAPACK is a software package provided by Univ. of Tennessee, Univ. of California Berkeley, Univ. of Colorado Denver and NAG Ltd

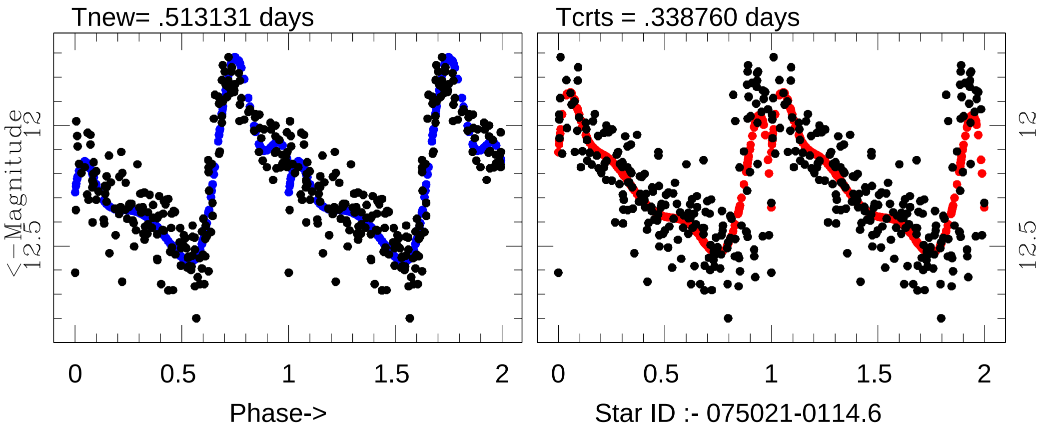

Figure 3.1: Phased light curve of the RR Lyrae star 075021-0114.6, along with the modified cubic spline curve is plotted for the both published period 0.338760 days and newly detected period 0.513131 days. It is clear that newly detected period gives better light curve

.

The method is demonstrated on the ASAS RR Lyrae star 075021-0114.6, with the number of observations 198, the published period 0.338760 days and newly detected period 0.513131 days [2]. The Figure 3.1 shows the fitted spline curve, superimposed over the phased light curve for both periods. It is clear that the light curve corresponds to the newly detected period is better. The modified cubic spline method is used only to confirm the exact period, over the PDM / GLSP/ SigSpec results, including the checking for possible double harmonics. The fitting is tested by the least square-square minimum and the corresponding period is taken as the best reliable period. The least square-square error E is,

$$E = \sum_{k=1}^{N} [y(t_k) - y_k]^2 \tag{3.36}$$

where $y(t_k)$ is the interpolated magnitude value and y_k is the observed apparent magnitude.

3.4 Conclusion

Both methods are robust enough to apply on periodic as well as semi-periodic variable star light curves. One of the disadvantage is the excess time taken by MCS method, compared to the parametric methods, which is not a considerable problem

[2]Note that all the periods shown in this thesis are in days

with the modern powerful computers. Another possibility is that, sometimes solution can get trapped into local minimum, instead of global minimum as pointed by [52]. Due to these reasons, we have used the MCS method only to confirm the exact period, instead of using it for searching the period.

Chapter 4

Application to ASAS database

4.1 Introduction

The All Sky Automated Survey (ASAS) project is dedicated to constant photometric monitoring of the whole available sky, which is approximately 10^7 stars, brighter than 14 magnitude [57], [58]. The ultimate goal is detection and investigation of any kind of the photometric variability and also to find and catalog variable stars.

ASAS survey has discovered more than 50000 variables and produced catalogue of variable stars (ACVS) of the southern hemisphere (Declination $< +28$ deg) and the photometric V-band data of the southern hemisphere until December 2009 are available. (http://www.astrouw.edu.pl/asas/?page=main).

4.2 Application to *ASAS* Database

In order to apply the modified cubic spline period confirmation method, we chose the ASAS project database[1], which contains time series data files of nearly 50,000 variable stars. The ASAS1 and ASAS2 periodic variables were classified by Eyer[33] and Richards[65]. Out of these, we chose two databases named *a2perlc* and *var3*, which contains 385 and 3169 time series data files respectively, which are cataloged by the ASAS team.

Initially the above specified time series files are subjected to PDM / GLS / SigSpec and the obtained frequencies are compared with published frequencies. The unmatched frequencies are then subjected to the modified cubic spline analysis, which finally compares the new frequency with that of published value and takes the frequency with minimum least square as the better frequency. The light curves are plotted after removing the few extreme outliers, using the *DEBiL* method by

[1] `astrouw.edu.pl/asas`

Table 4.1: Published periods from *a2perlc* are not available for comparison with newly detected periods for 4 variable stars. The corresponding new light curves are shown in the Figure 4.1

Star−ID	new T	Star−ID	new T
112422-6123.3	38.821121	114127-6216.1	5.894262
112756-6123.6	1.744592	170022-2145.0	101.40923

Devor[29]. We also visually examined the new light curves and compared with the published ones, to make sure that the new results are genuine.

We have automated the whole process of time series analysis so that no manual interruption or intervention is required in between the steps of computations involved in period search and plotting of light curves. We should able to detect nearly 5 to 6% new better periods. Some periods are improved, some new periods are entirely different from the published periods and some new periods are harmonics (integral or half-integral multiples) of the published periods.

The *a2perlc* database consists of 385 time series files. From this database we have discovered 4 variable stars, which are missing from the catalog. Their periods are given in the table 4.1 and the corresponding new light curves are shown in Figure 4.1. The table 4.2 contains the published periods from *a2perlc* database, which are compared with new different periods for 14 variable stars and these light curves are shown in Figure 4.2.The table 4.3 is the published periods from *a2perlc* database and compared with new harmonic periods for 15 variable stars and these light curves are shown in Figure 4.3. The total detection of new better periods are 33 (4+14+15).

The *var3* database, which contains 3169 time series data and we have got 245 better periods and corresponding light curves.The table 4.4 contains the published periods from *var3* database, which are compared with new periods for 200 variable stars with different periods. These 200 light curves are shown in Figure 4.4. The table 4.5 shows the published periods from *var3* database, which are compared with new periods for 45 variable stars with harmonic periods (double, triple or half etc). These 45 light curves are shown in Figure 4.5.

Detection of exact periodicity will help to identify the type of variability and classify the variable stars, in accordance with the shape and other features of the light curve.

4.3 Results from *a2perlc* and *var3* databases

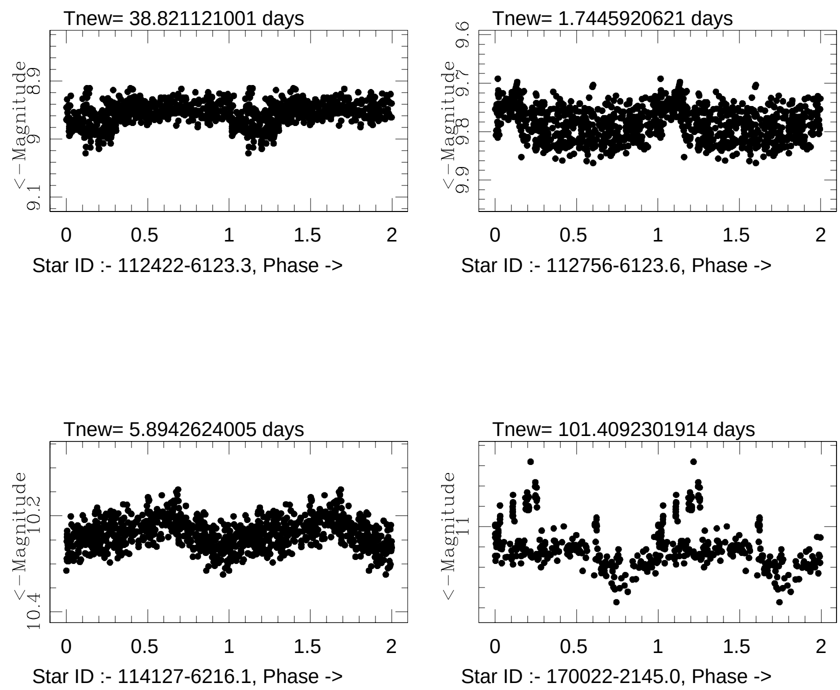

Figure 4.1: Light curve of the four stars 112422-6123.3, 112756-6123.6, 114127-6216.1 and 170022-2145.0, which are not published by ASAS in the *a2perlc* database
.

Table 4.2: Published periods from *a2perlc* database, which are compared with new different periods for 14 variable stars. The corresponding light curves are shown in the Figure 4.2

Star−ID	$Tasas$	$Tnew$	Star−ID	$Tasas$	$Tnew$
015647−0021.2	0.351447	0.542709	075021−0114.6	0.338760	0.513148
103609−5203.9	24.18321	0.956172	112301−6146.8	2.676421	3.953892
113457−6157.6	198.9680	12.51002	113916−6026.1	7.011300	0.872959
114619−6440.7	59.66783	0.980366	114726−6132.9	78.39775	5.141568
123639−6344.8	1.503000	0.600461	175630−3548.2	85.94261	0.988364
183736−0040.7	63.96056	0.983132	184435−0049.4	63.69426	0.982352
193640+0053.7	2.001650	0.666427	200208−1958.8	0.956967	22.14830

.

Table 4.3: Published periods from *a2perlc* database, which are compared with new harmonic periods for 15 variable stars. These light curves are shown in the Figure 4.3

Star−ID	$Tasas$	$Tnew$	Star−ID	$Tasas$	$Tnew$
045128-0032.7	3.618900	1.563815	045511-0101.7	3.136200	1.512603
050818-6846.8	30.43856	60.71933	050920-7027.4	37.48828	74.72229
060241-0051.7	3.857475	7.725706	065032+0000.4	7.095586	14.19883
065627-0017.4	1.177400	2.354870	104838-5245.7	1.157240	2.315714
112826-5929.4	4.499000	2.250337	113318-6306.2	0.168483	0.336968
113840-6117.9	1.195500	2.390514	113916-6026.1	7.011300	1.745793
124351-6305.2	12.51597	6.240672	124435-6331.7	2.565139	1.283260
125124-6404.7	6.281812	3.146894			

.

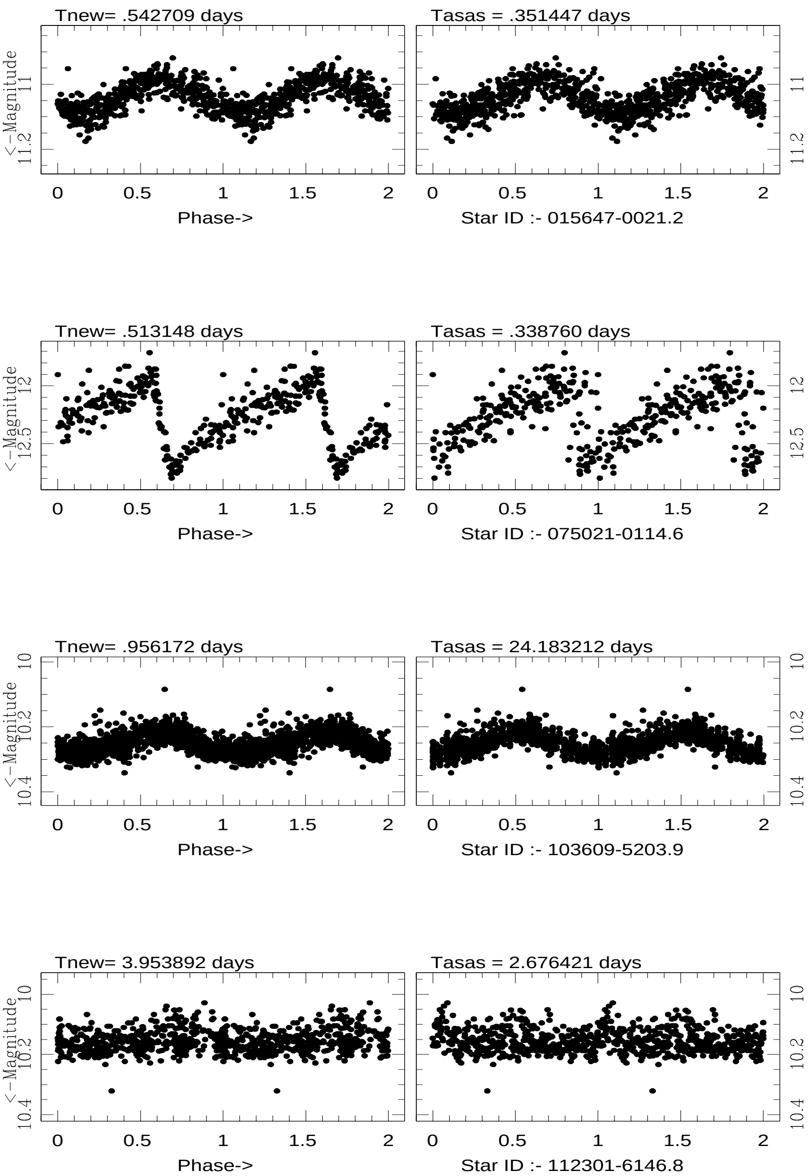

Figure 4.2: The newly detected periods and LC's are entirely different from the published periods.

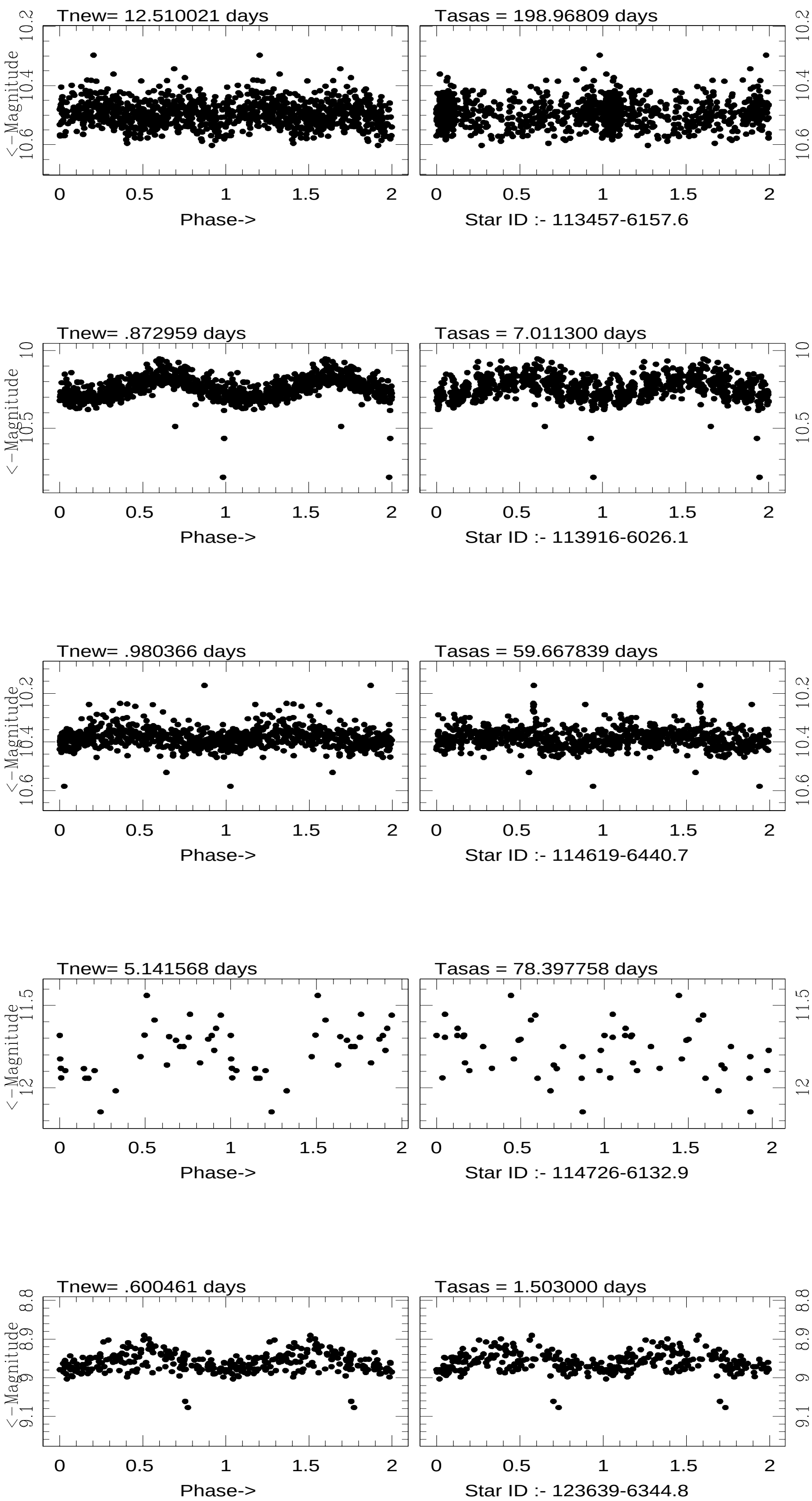

Fig. 4.2 continued ...

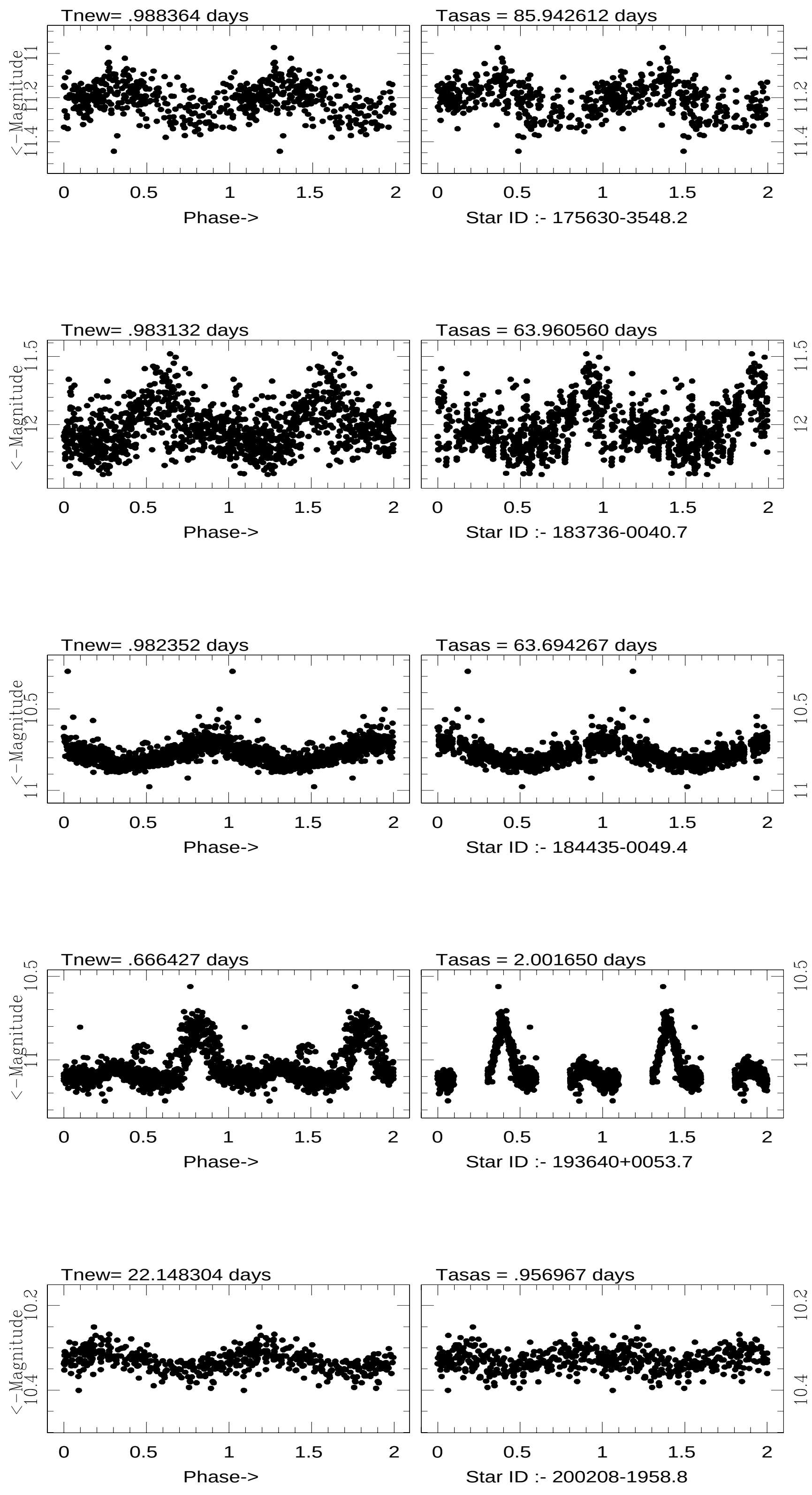

Fig. 4.2 ends here

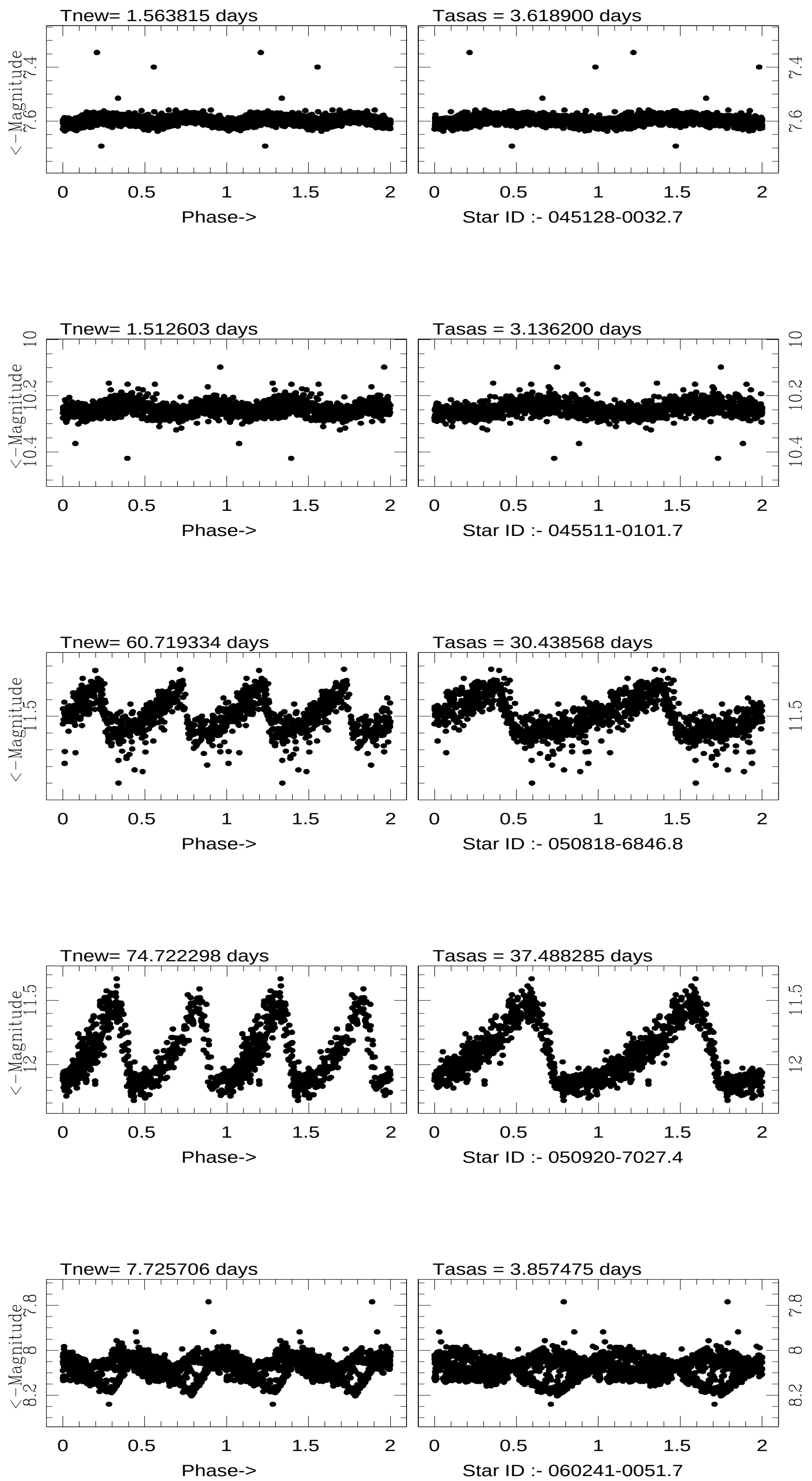

Figure 4.3: The newly detected periods are integral or half integral multiples of the published periods.

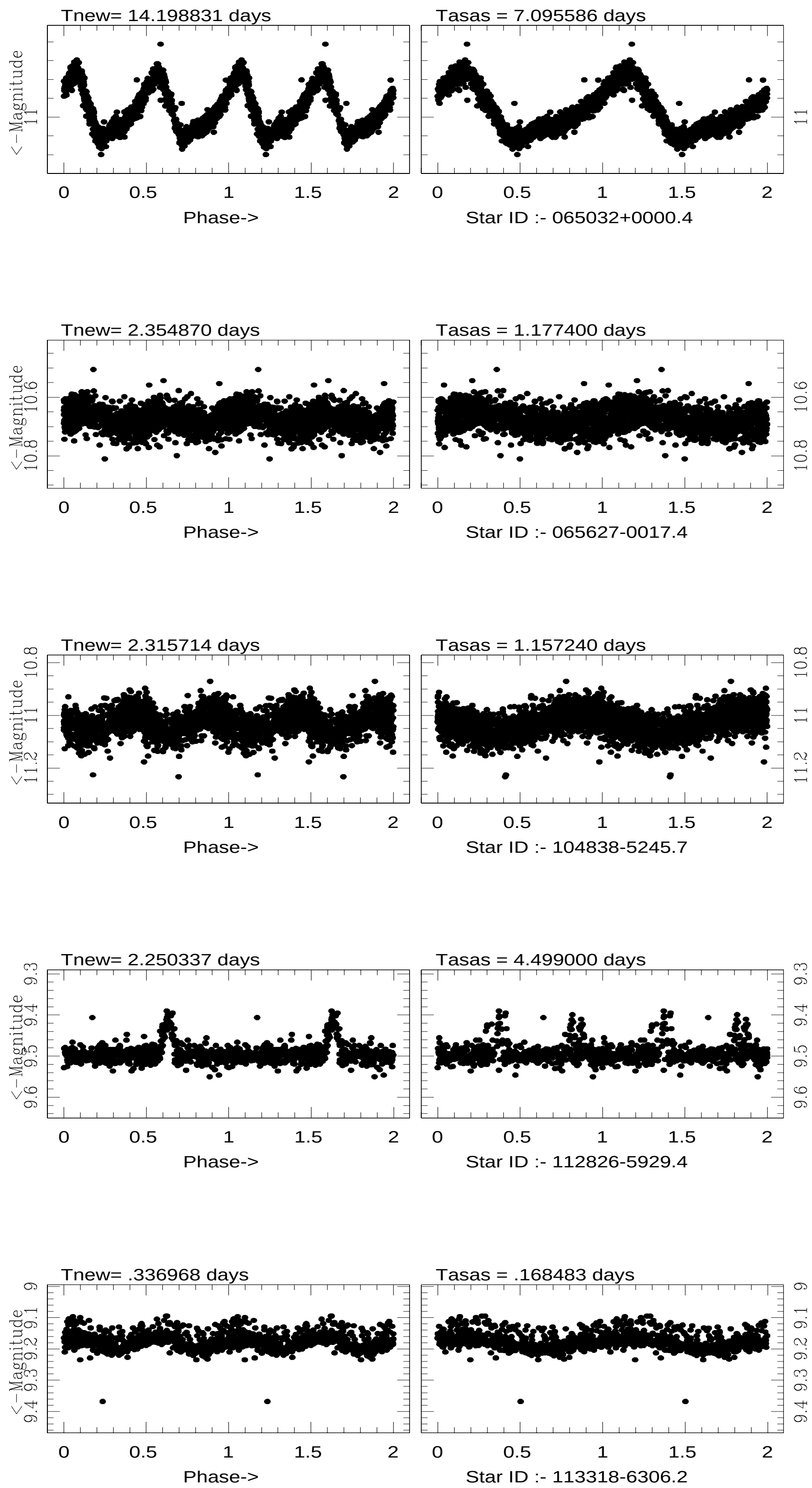

Fig. 4.3 continued ...

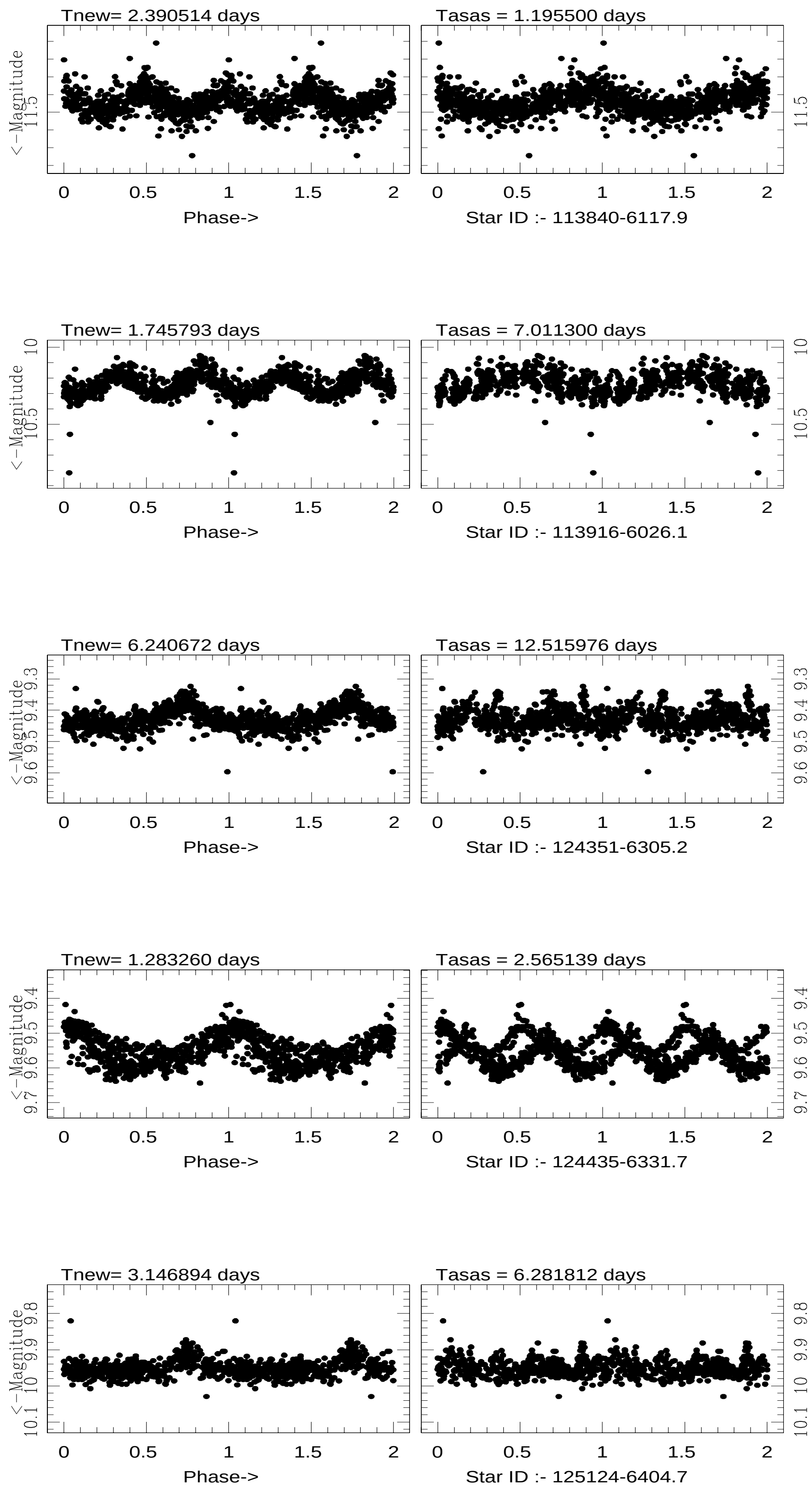

Fig. 4.3 ends here

Table 4.4: Published periods *var3* compared with new periods for 200 variable stars with different periods. The corresponding light curves are shown in the Figure 4.4

Star−ID	$Tasas$	$Tnew$	Star−ID	$Tasas$	$Tnew$
000155−6707.7	228.0000	45.07371	000239−1926.7	72.80000	136.4999
000309−1050.5	68.19999	77.82857	000316−1125.1	0.368920	0.269316
000602−3654.3	0.439110	0.784811	000636−3235.6	71.30000	134.7888
000659−0846.7	43.09999	0.974734	000739+0120.8	1.939900	0.083111
000815−2414.7	0.262427	0.393644	000819−4949.0	59.50000	146.1962
001330−7010.3	28.11000	37.68832	001513−6851.0	34.86000	0.969629
001549−7625.8	46.20000	73.67661	001654−3013.8	171.0000	142.1415
001724−7559.4	41.59999	0.975012	001854−7401.9	43.90000	0.976079
002023−2323.0	97.69999	231.4472	002102−7547.6	0.239260	0.747184
002108−4802.6	350.0000	2.000966	002334−1751.5	29.12000	223.1910
002458−7041.8	32.79000	0.329437	002656−7933.0	103.1999	192.8253
002941−2034.8	46.79999	227.6681	003510−5020.1	8.354000	1.136182
003613−5231.2	39.20000	106.2726	003615−1711.0	44.20000	0.975221
003629−2011.7	0.245848	0.197227	003629−3020.0	51.40000	0.665974
004237−6740.4	78.80000	68.72750	004409−0214.8	73.00000	266.0754
010029−4404.2	0.325583	0.197082	010431−7032.4	47.20000	109.4948
010534−1131.9	48.59999	0.066458	010906−7238.5	85.09999	241.2081
011022−5433.7	24.43000	216.4467	011339−2617.2	50.70000	72.36817
011415−0210.7	325.0000	0.332937	011725−7343.6	38.90000	198.9687
012222−2906.9	0.205090	0.170175	012438−5733.8	33.66999	0.968879
012514−4555.6	29.18000	211.2805	012850−2729.1	31.04000	0.967104
012943−4900.0	0.572510	11.74276	013404−4314.5	53.20000	0.978479
013407−2246.7	21.24000	0.956155	013608−3511.2	127.0000	101.9447
014049−6729.7	1.122800	17.85914	014224−4640.9	319.0000	281.9445
014226−3027.6	0.273712	0.377304	014318−3618.9	83.59999	0.199425
014435−7745.3	0.205717	13.61339	014508−1914.2	2.523200	1.648250
014820−8300.1	138.0000	251.8810	014945−7134.3	99.00000	0.986851
015254−7209.2	51.90000	116.3482	015417−7622.8	62.40000	101.2127
015452−6616.0	58.29999	0.989579	015757−2628.9	79.50000	195.6078
015825−2422.7	91.40000	0.333915	015827−3648.9	100.5000	211.0648
020219−2751.6	107.0000	196.0925	021029−1835.8	72.80000	111.6817
021411−8143.8	129.0000	104.3307	021741−5639.9	81.90000	185.4553
021829−6821.1	68.69999	176.0447	022808−5542.4	36.70000	235.6146
024258−2607.1	90.50000	197.4380	024415−5418.1	64.59999	225.5781

Star−ID	$Tasas$	$Tnew$	Star−ID	$Tasas$	$Tnew$
024506−4128.3	0.101698	0.127706	024528−1359.5	75.80000	0.200089
024754−4546.7	0.134506	0.155477	024840−3309.2	0.352050	0.641299
024943−6948.7	59.50000	0.980476	025347−1307.7	0.185929	0.228547
025858−1320.7	51.29999	44.43615	030615−2612.8	0.406140	0.288577
030754−2709.0	30.33000	0.967850	030954−2731.4	117.0000	206.8280
031002−4047.7	61.90000	0.981542	031115−5701.5	10.60800	0.911403
031124−3719.5	0.322599	0.243749	031320−1726.8	0.262896	0.208033
031336−0239.5	46.00000	0.976666	031555−7636.3	56.29999	67.09867
031618−2727.3	50.90000	0.111154	032038−5902.4	0.354960	0.301314
032336−1953.0	80.09999	193.6298	032719−0730.7	75.50000	181.5049
032846−6041.4	93.59999	204.9276	033219−3539.3	0.060689	0.054113
033630−3219.6	40.90000	35.87755	033719−5225.5	53.00000	221.8075
034103−3029.6	32.09000	0.967104	034136−3750.4	151.0000	0.249205
034234−1450.7	54.70000	135.4731	034418−3739.0	31.35000	0.198410
034534−1123.2	0.132551	0.180437	035027−8608.5	41.70000	108.9684
035442−2914.4	42.50000	103.7475	035504−4710.2	127.0000	145.1198
040019−0623.1	22.83000	0.197966	040031−1559.5	45.20000	53.45301
040404−3801.0	6.307000	0.863288	040418−2036.7	39.20000	0.974013
040602−0915.7	50.40000	0.978034	040717−2800.5	44.50000	0.975849
040936−8151.3	9.685000	329.9542	041038−8647.1	45.70000	82.14008
041443−1852.2	60.29999	0.981245	041521−0630.3	41.50000	0.251843
041704−1213.3	69.80000	0.982740	041957−7046.7	175.0000	0.249316
041958−1843.3	78.59999	120.3035	042142−6339.5	72.59999	90.22408
042144−1403.8	271.0000	186.2696	042351−7654.7	0.651920	1.883532
042515−5128.3	0.200940	26.11445	042525−1136.9	68.59999	144.5836
042559−3618.3	110.0000	205.3067	042640−3618.6	0.386530	0.631308
042917−8448.3	50.79999	189.2825	043358−5147.1	35.50000	188.1361
043701−2945.7	81.00000	198.8178	043812−6218.2	96.50000	212.7863
043924−5212.0	80.19999	97.73649	043930−3233.1	0.357980	4.716863
044142−3314.9	98.80000	261.4868	044325−1028.9	0.188280	0.302218
044526−4906.8	0.201264	0.167461	044704−1819.1	33.41999	215.6656
045213−4100.3	29.10000	0.167429	045426−2810.0	161.0000	186.4880
045550−2742.2	24.15000	0.960614	045742−6517.0	20.82000	178.2709
045833−7020.8	0.973410	35.66350	050247−2944.1	108.0999	148.9985
050300−5405.9	42.70000	202.6619	050324−1630.6	16.44000	180.8750
050415−6715.1	42.59999	204.6315	050737−0140.1	6.416000	0.862664

Table 4.4 continued...

Star−ID	$Tasas$	$Tnew$	Star−ID	$Tasas$	$Tnew$
050816−5013.1	63.40000	159.7113	050911−6936.2	99.00000	163.6492
051218−3438.3	20.26000	0.327305	051448−6911.5	63.50000	218.6949
051535−5309.4	80.09999	0.984621	051600−0948.6	41.20000	99.52346
051605−6006.9	1.070320	0.936245	051614−2955.1	0.156993	0.186336
051757−6808.7	410.0000	0.332503	052222−3030.7	49.40000	41.51421
052234−4635.3	0.443420	0.665069	052320−2428.1	99.40000	0.986254
052422−1020.8	54.20000	0.111069	052632−1659.4	63.09999	173.8745
052901−4821.8	52.40000	0.331986	052908−6912.3	253.0000	220.7868
052921−6847.5	104.4000	194.0727	052931−5402.8	42.20000	48.33171
053019−7302.1	161.0000	200.2298	053022−6919.7	66.90000	0.332129
053031−1303.4	74.69999	130.3753	053114−1028.8	129.0000	156.0513
053118−7900.5	11.10000	0.917295	053230−2226.2	37.40000	55.71995
053255−2646.1	57.40000	190.2430	053512−5801.2	0.104758	0.064278
053940−5934.1	0.158203	0.136602	053955−4504.2	39.20000	0.052451
054057−1810.6	0.160241	0.190926	054059−6918.6	188.0000	166.0495
054232−1914.0	33.63999	191.0070	054324−1444.0	86.69999	192.3820
054530−1903.5	57.70000	193.5326	054648−1204.0	77.69999	108.4950
054747−6022.1	79.09999	92.39866	054841−7003.2	79.09999	128.5948
055137−1432.2	1.082590	0.341310	055406−2509.2	42.09999	53.67042
055430−0343.2	279.0000	0.249587	055451−1526.2	78.59999	138.1432
055648−5925.4	74.50000	199.1041	055742−1916.3	77.50000	189.3144
060130−2527.9	90.90000	81.62194	060151−2127.7	64.40000	219.0475
060524−5729.6	127.0000	195.4714	060617−0705.9	336.0000	0.985932
060810−1406.8	49.00000	81.64902	060848−1302.4	150.0000	196.7356
125353−8727.4	54.50000	191.4580	140041−8519.4	20.20000	48.09371
140634−8522.9	26.79999	0.124526	170225−8703.3	32.45999	119.0846
171030−8623.0	222.0000	272.2056	222201−8440.0	310.0000	190.3982
231708−8548.1	79.00000	124.0942	233524−5503.1	39.79999	182.7891
233614−8331.0	86.59999	240.5640	234708−4548.9	78.90000	219.9647
235007−1417.8	84.69999	59.34230	235054−3517.8	159.0000	113.0367

Table 4.4 ends here

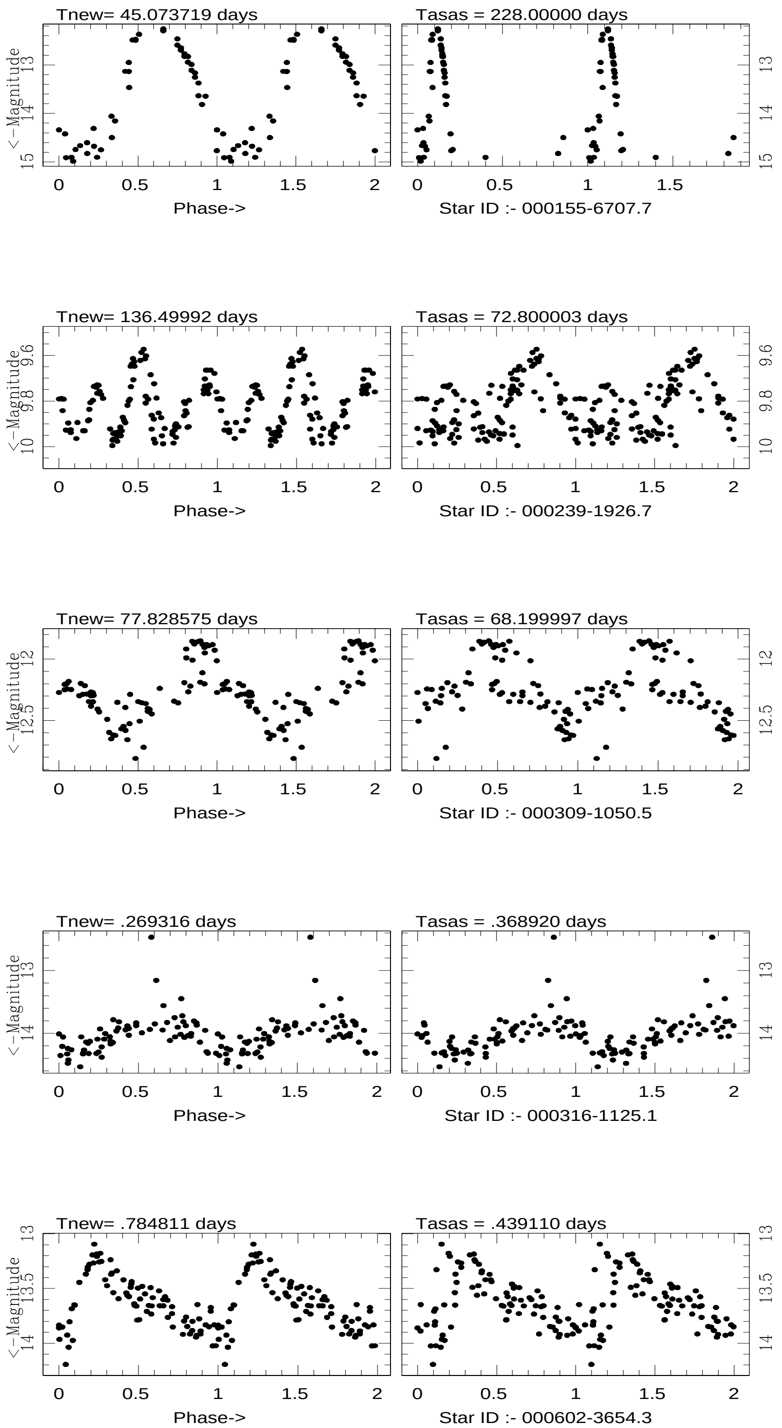

Figure 4.4: The light curves shown above are from the *var3* database. Newly detected light curves are on the left and published light curves on the right. Periods in days are given at the top of each figure. The newly detected periods are entirely different from the published periods.

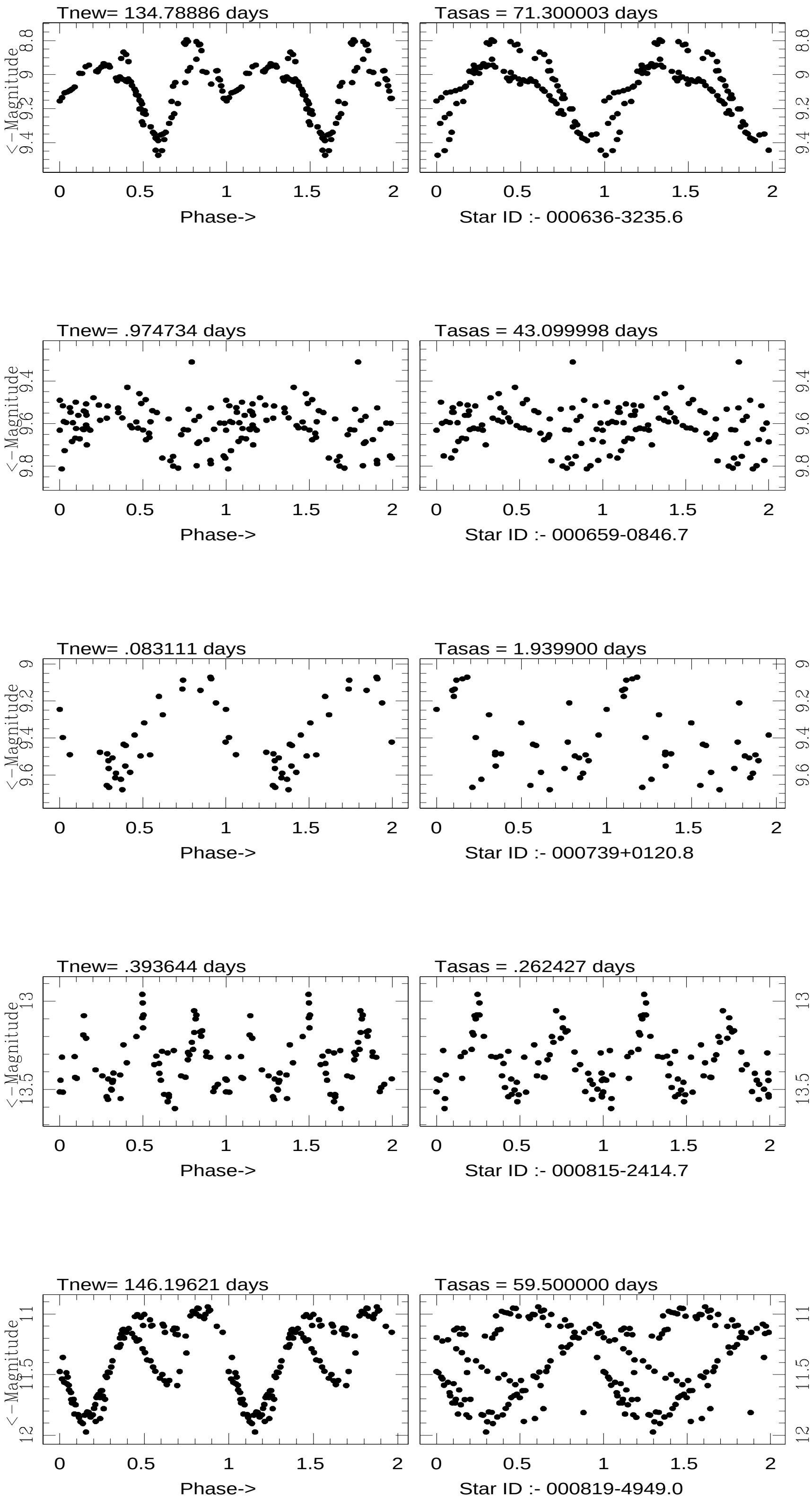

Fig. 4.4 continued...

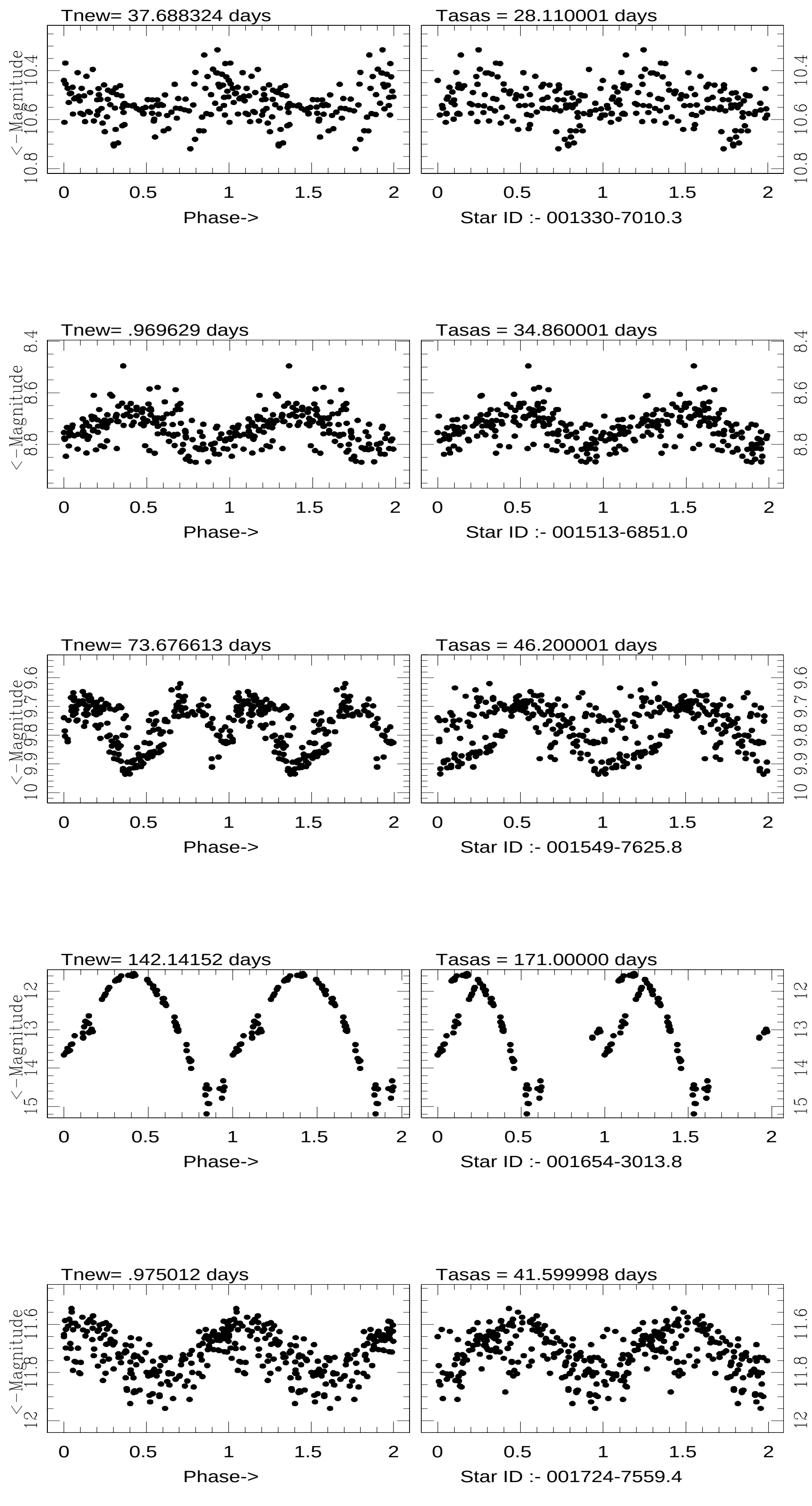

Fig. 4.4 continued...

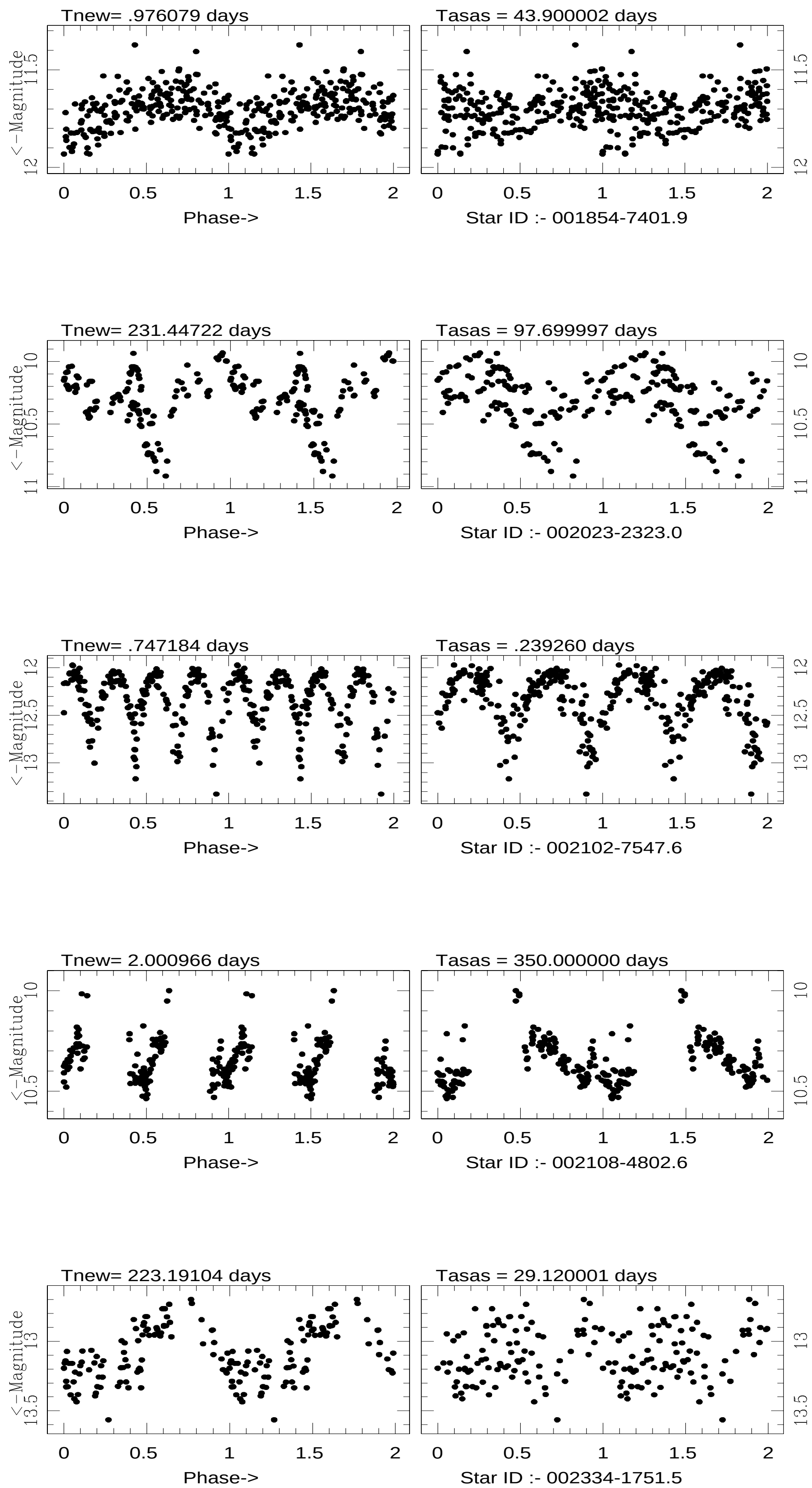

Fig. 4.4 continued...

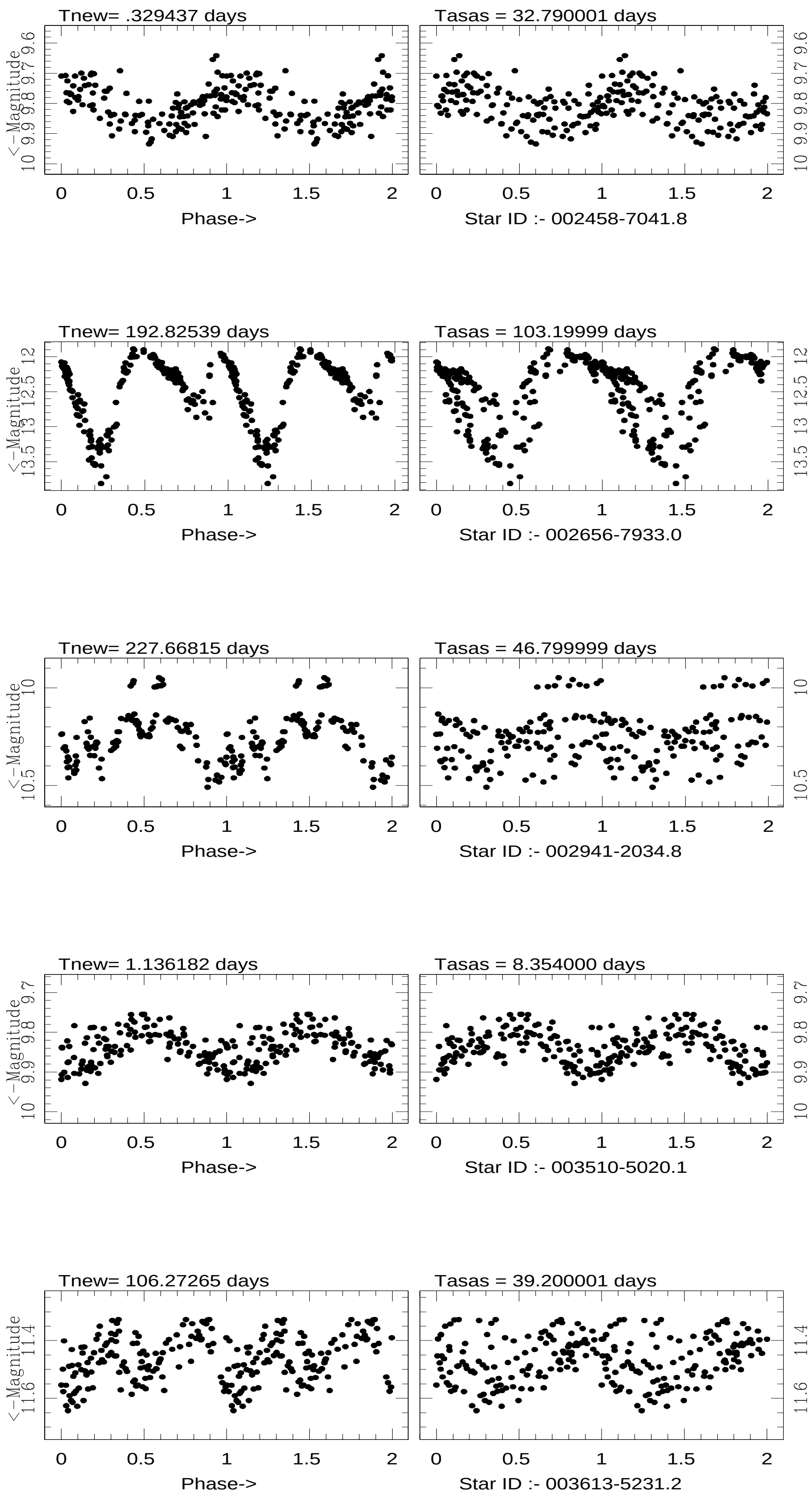

Fig. 4.4 continued...

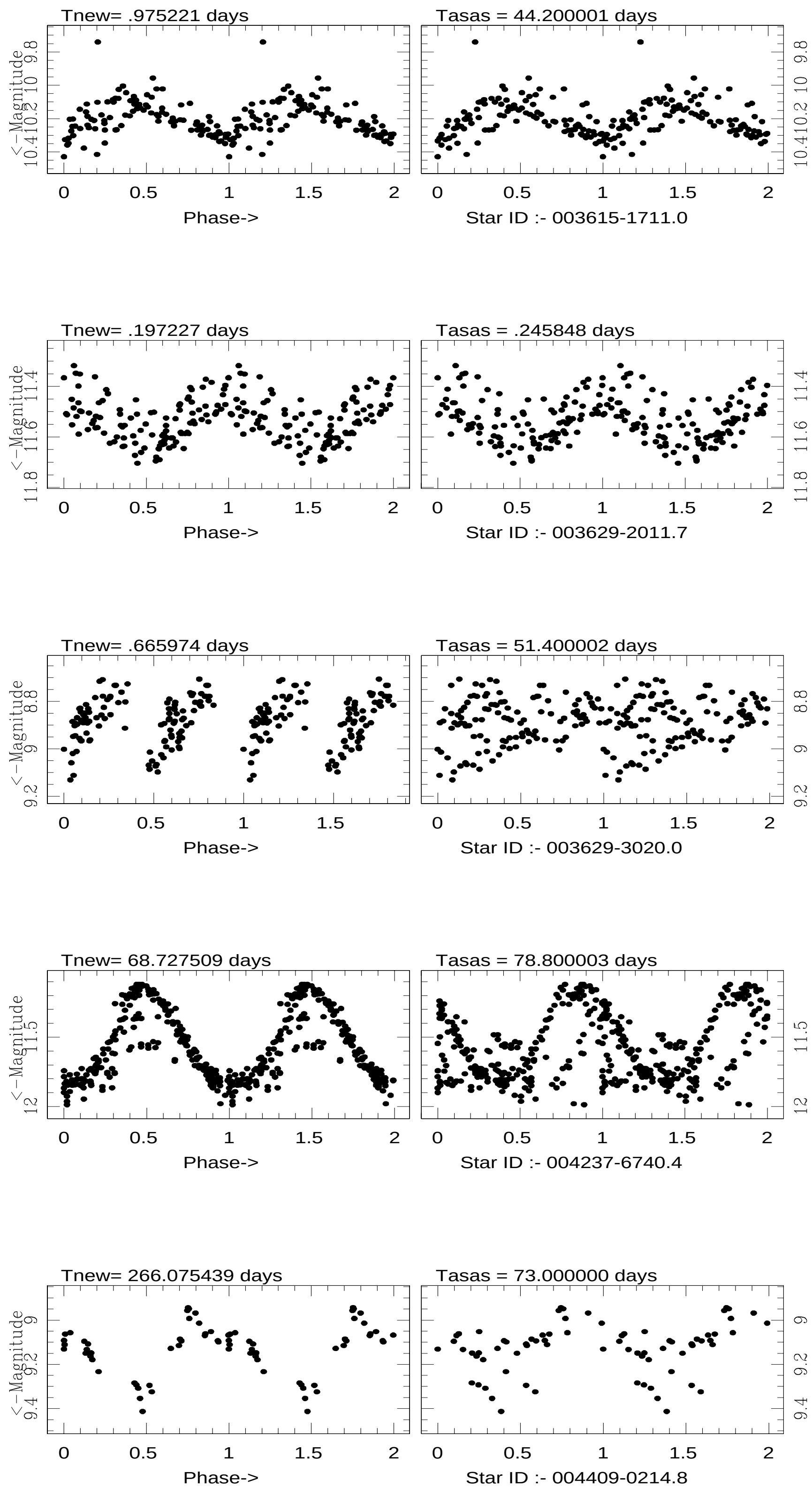

Fig. 4.4 continued...

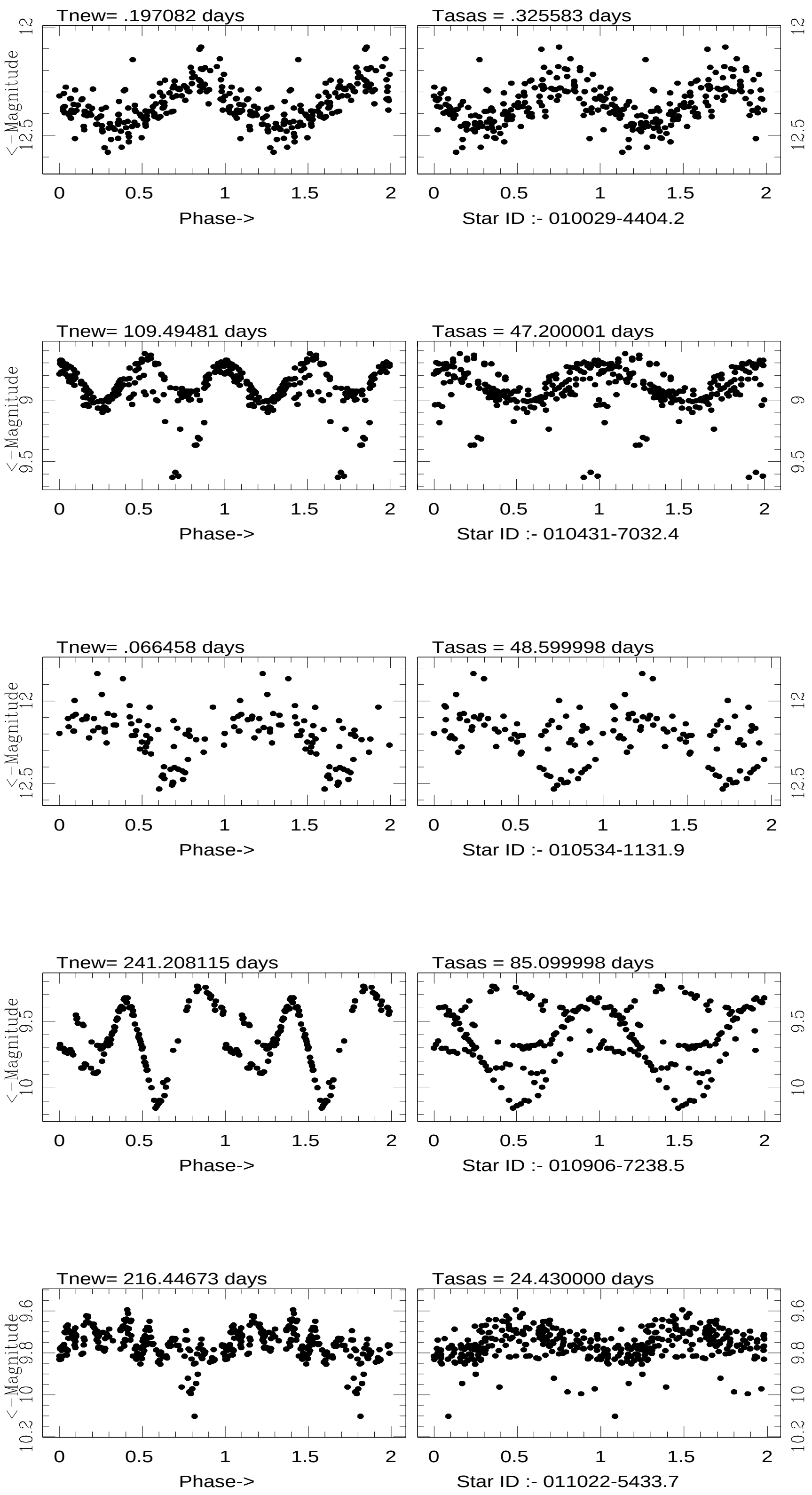

Fig. 4.4 continued...

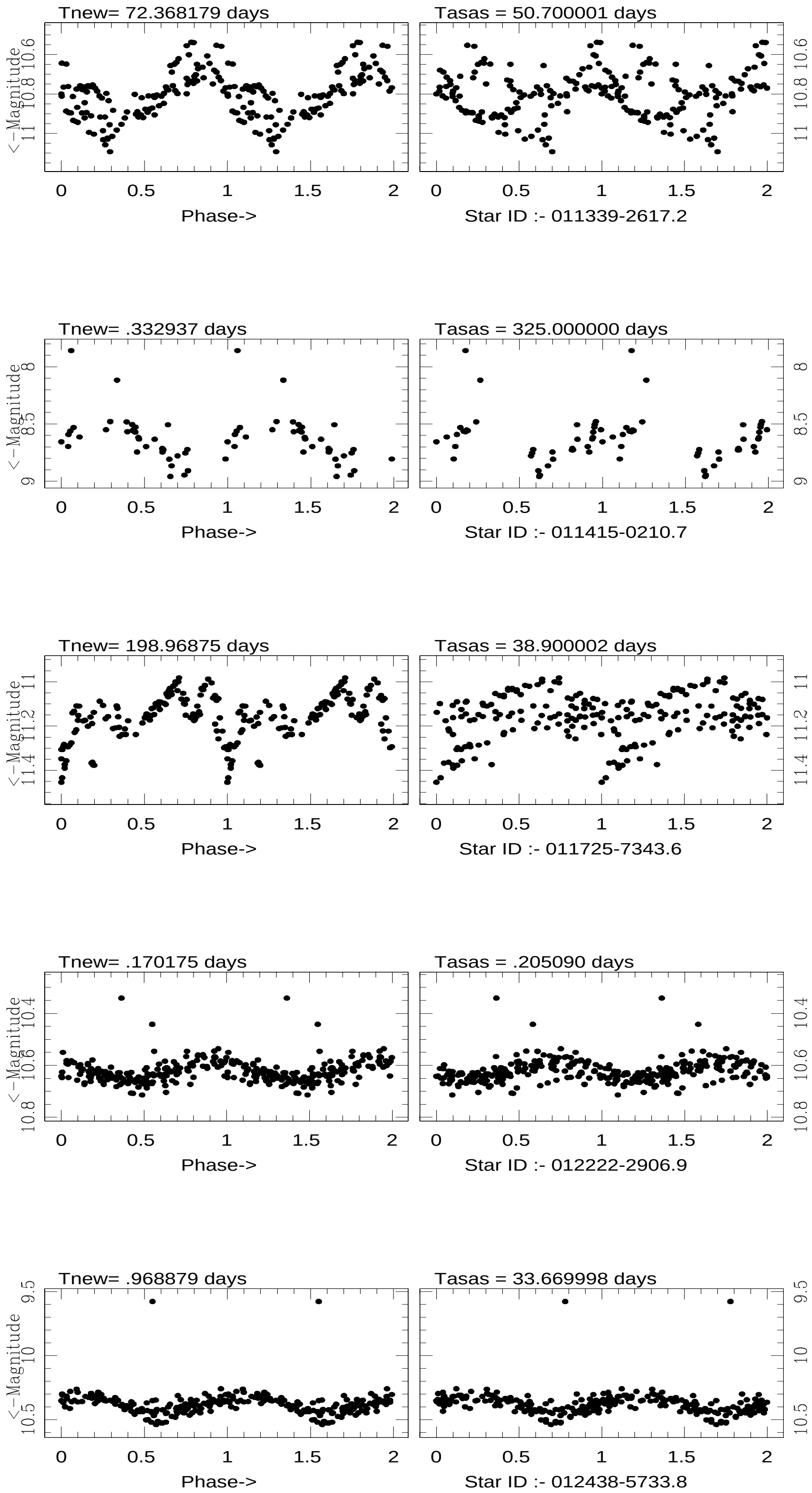

Fig. 4.4 continued...

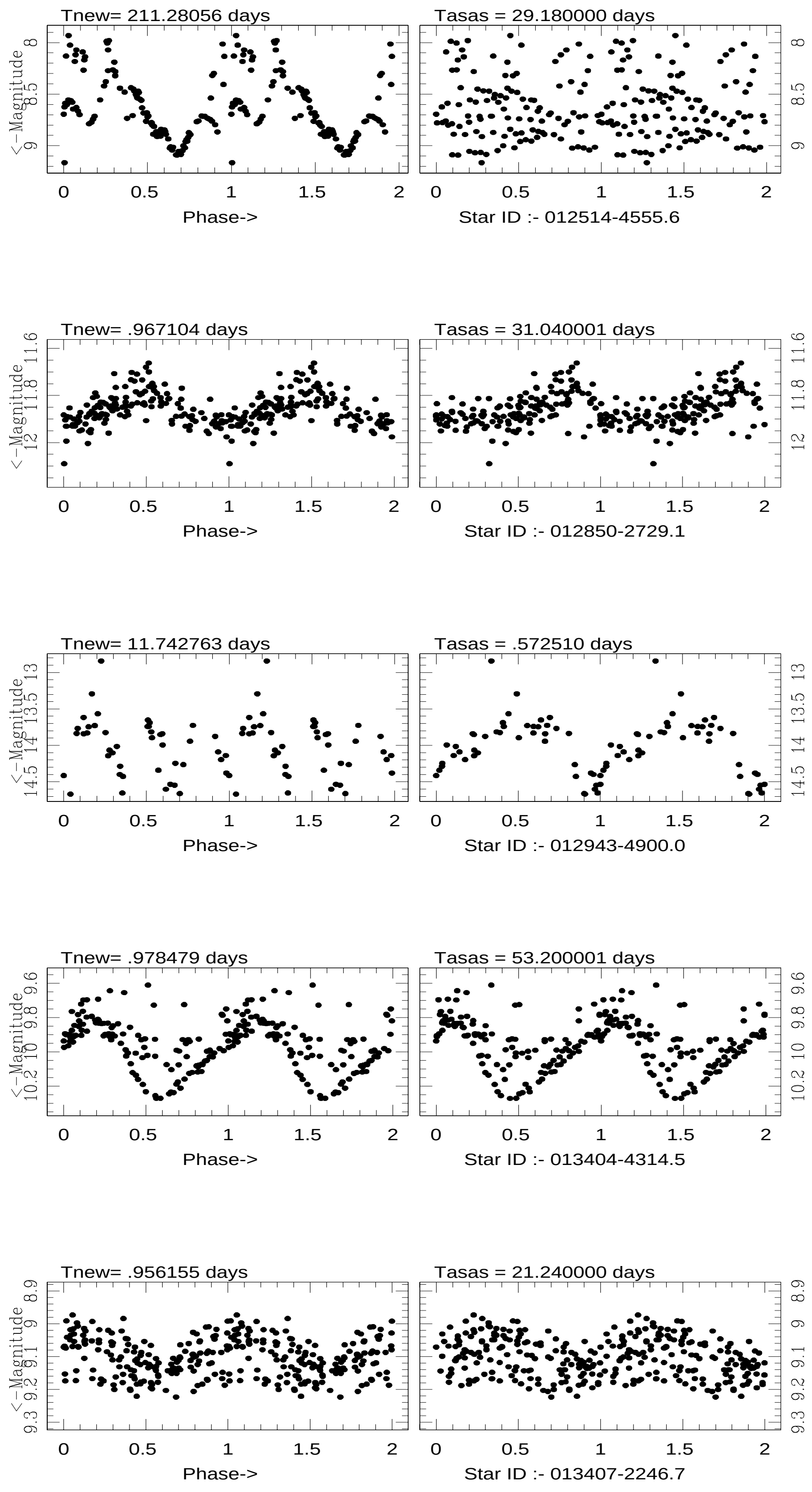

Fig. 4.4 continued...

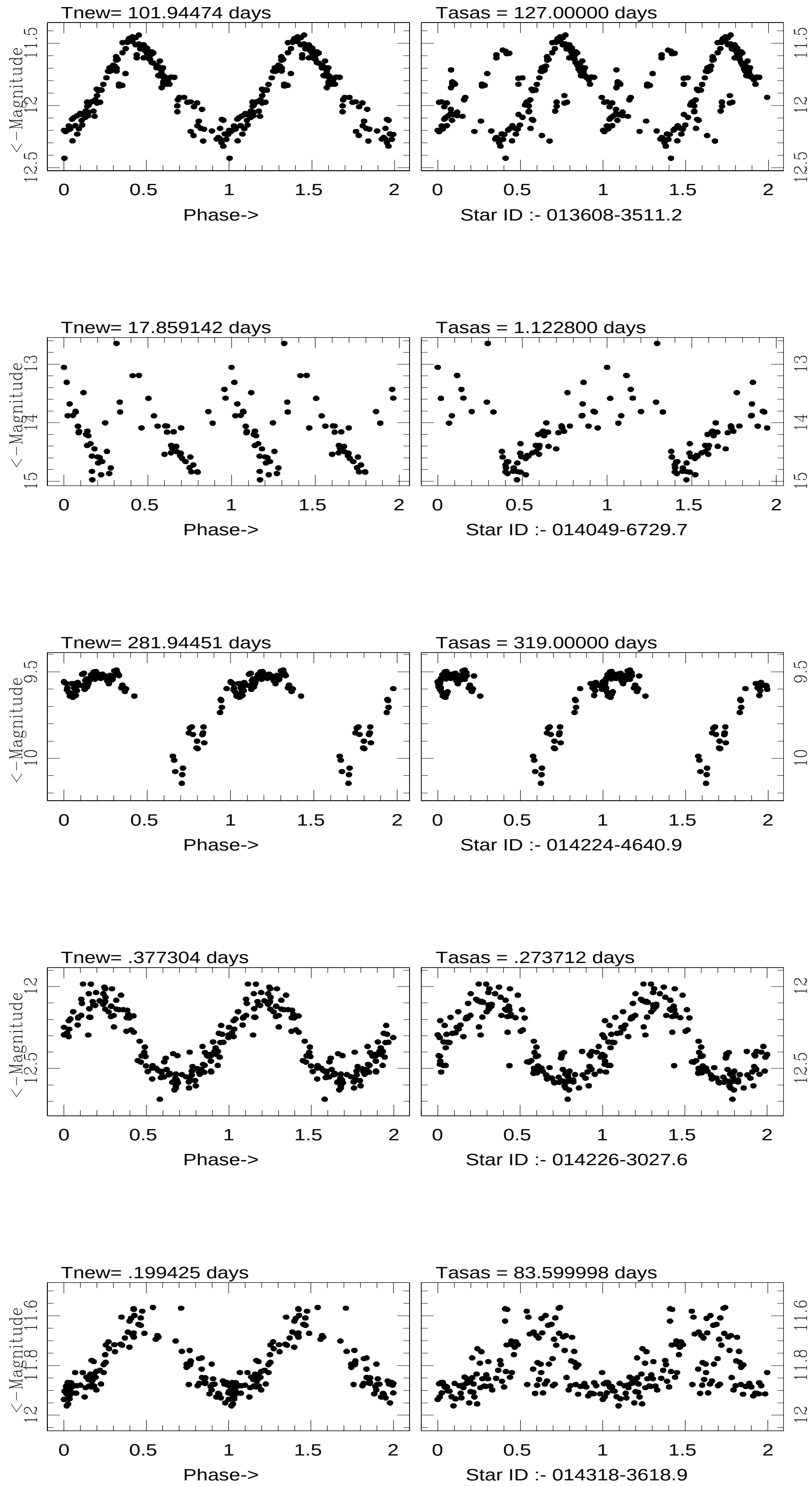

Fig. 4.4 continued...

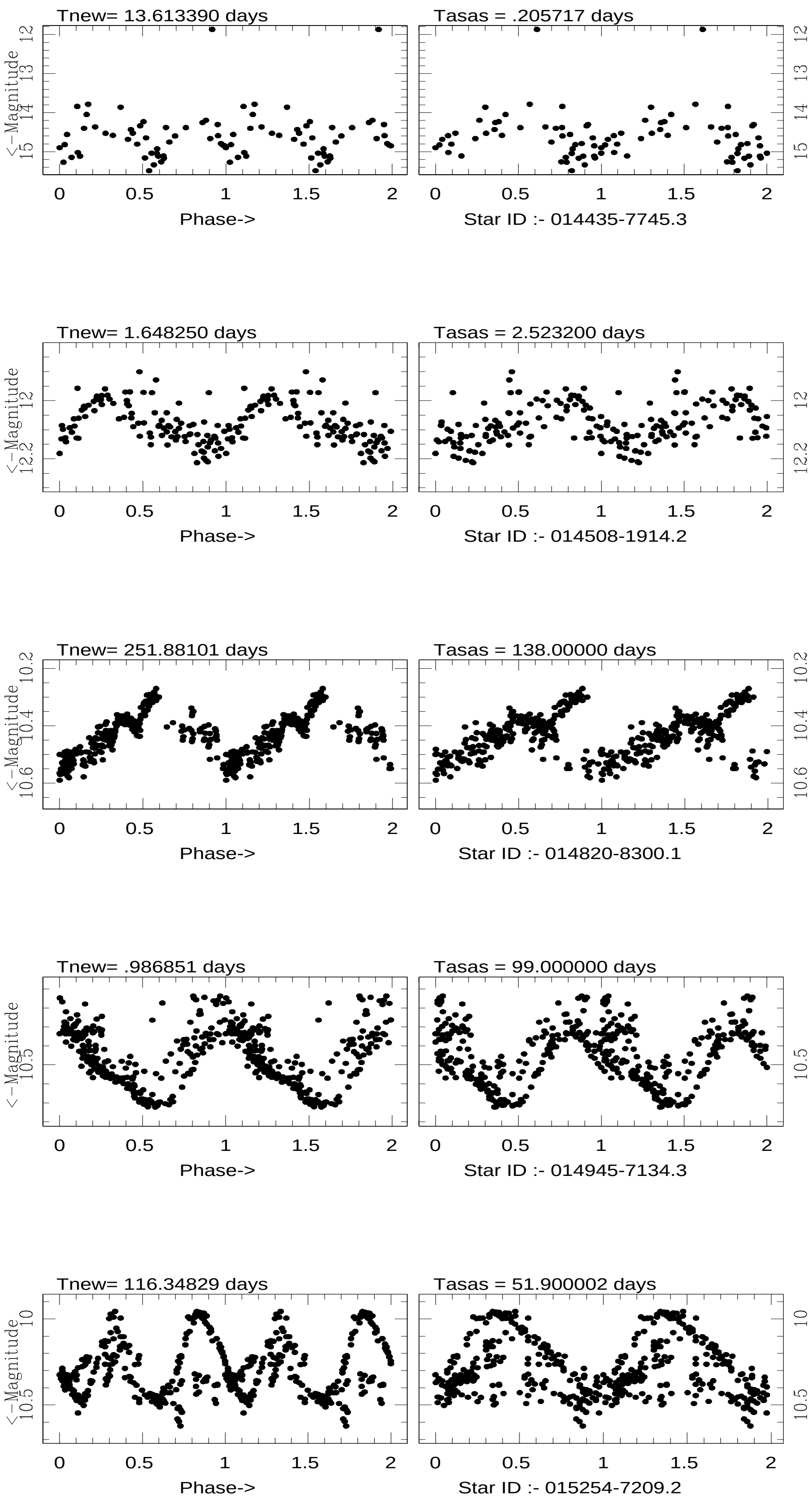

Fig. 4.4 continued...

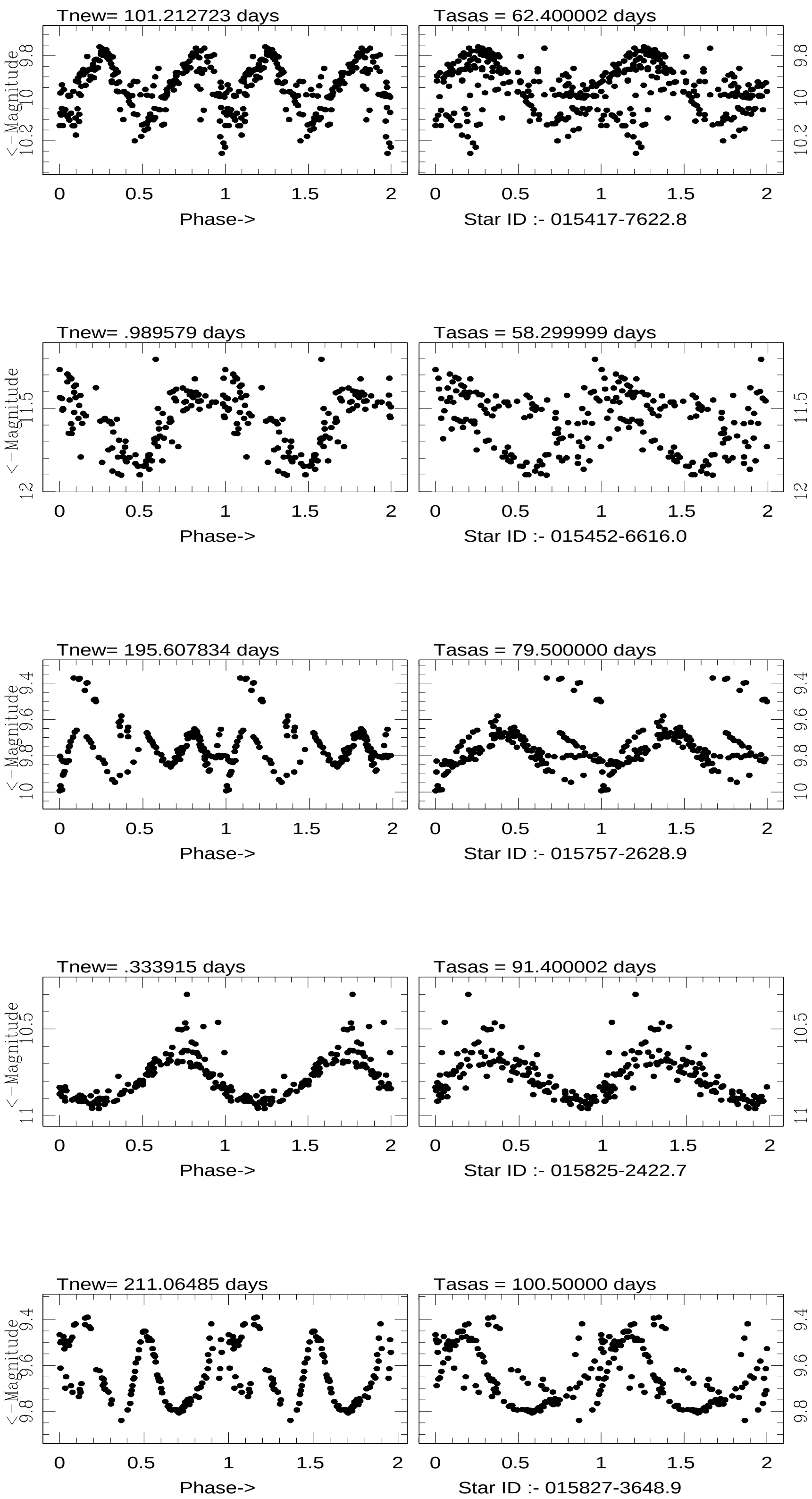

Fig. 4.4 continued...

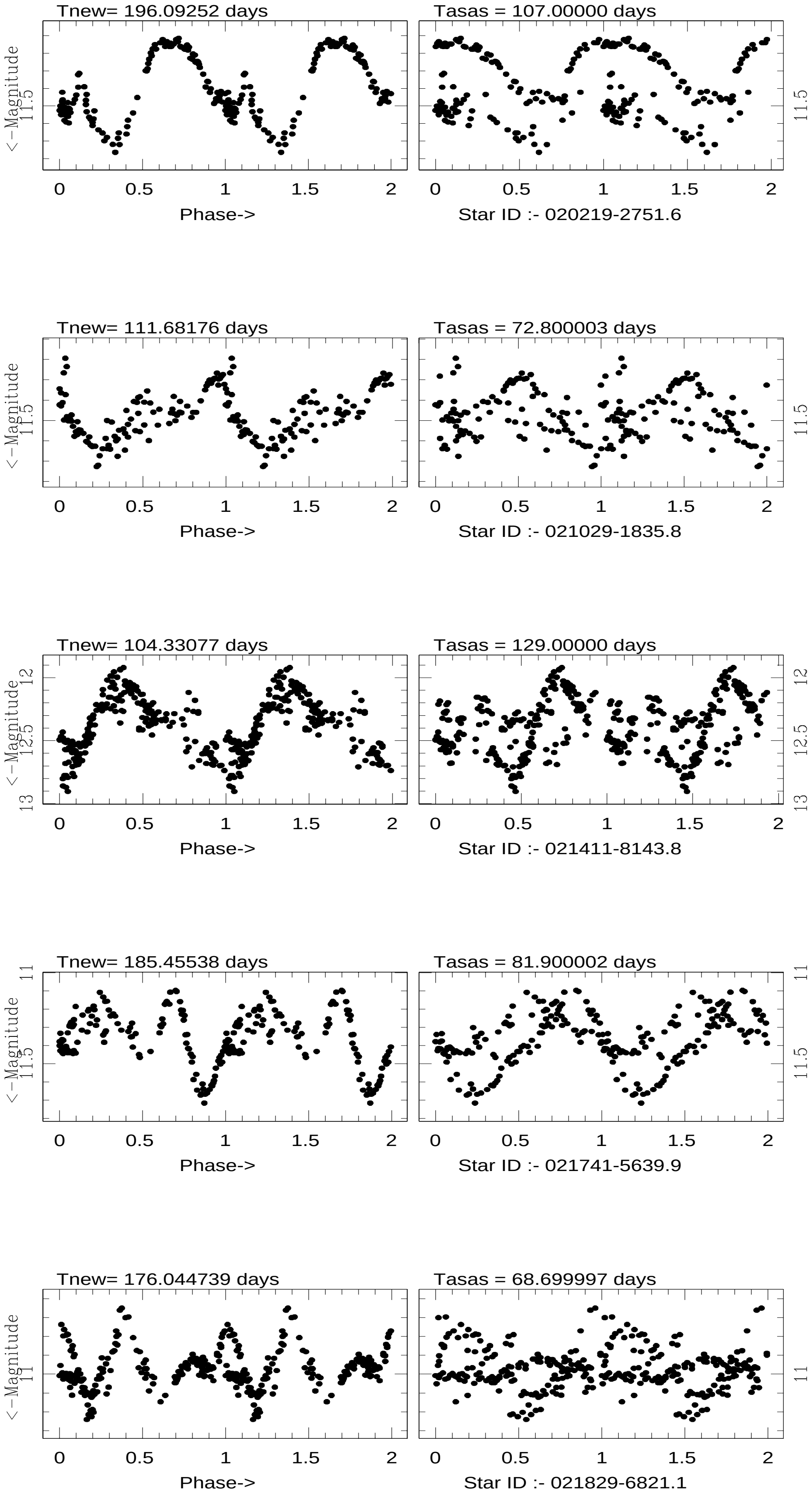

Fig. 4.4 continued...

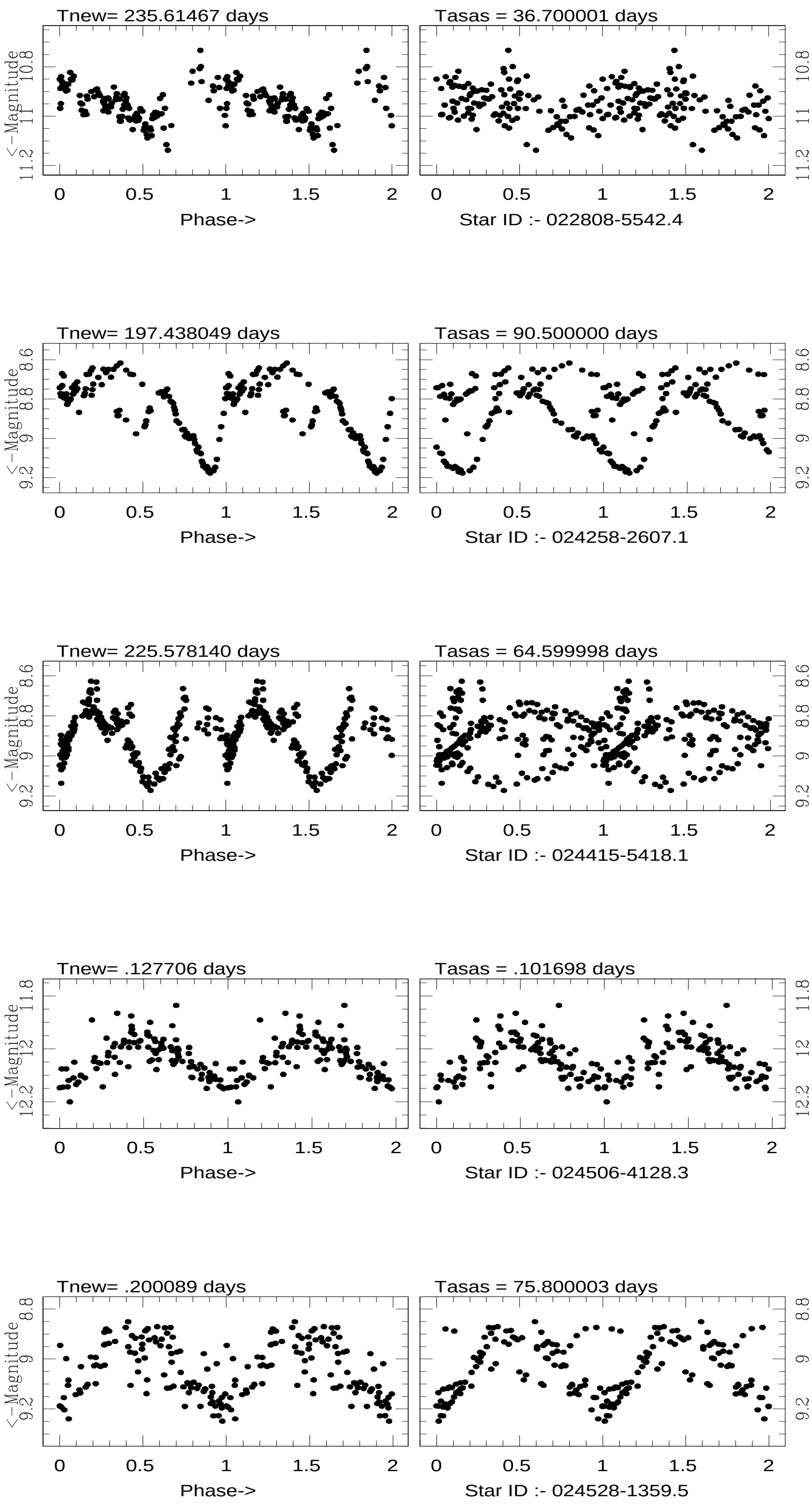

Fig. 4.4 continued...

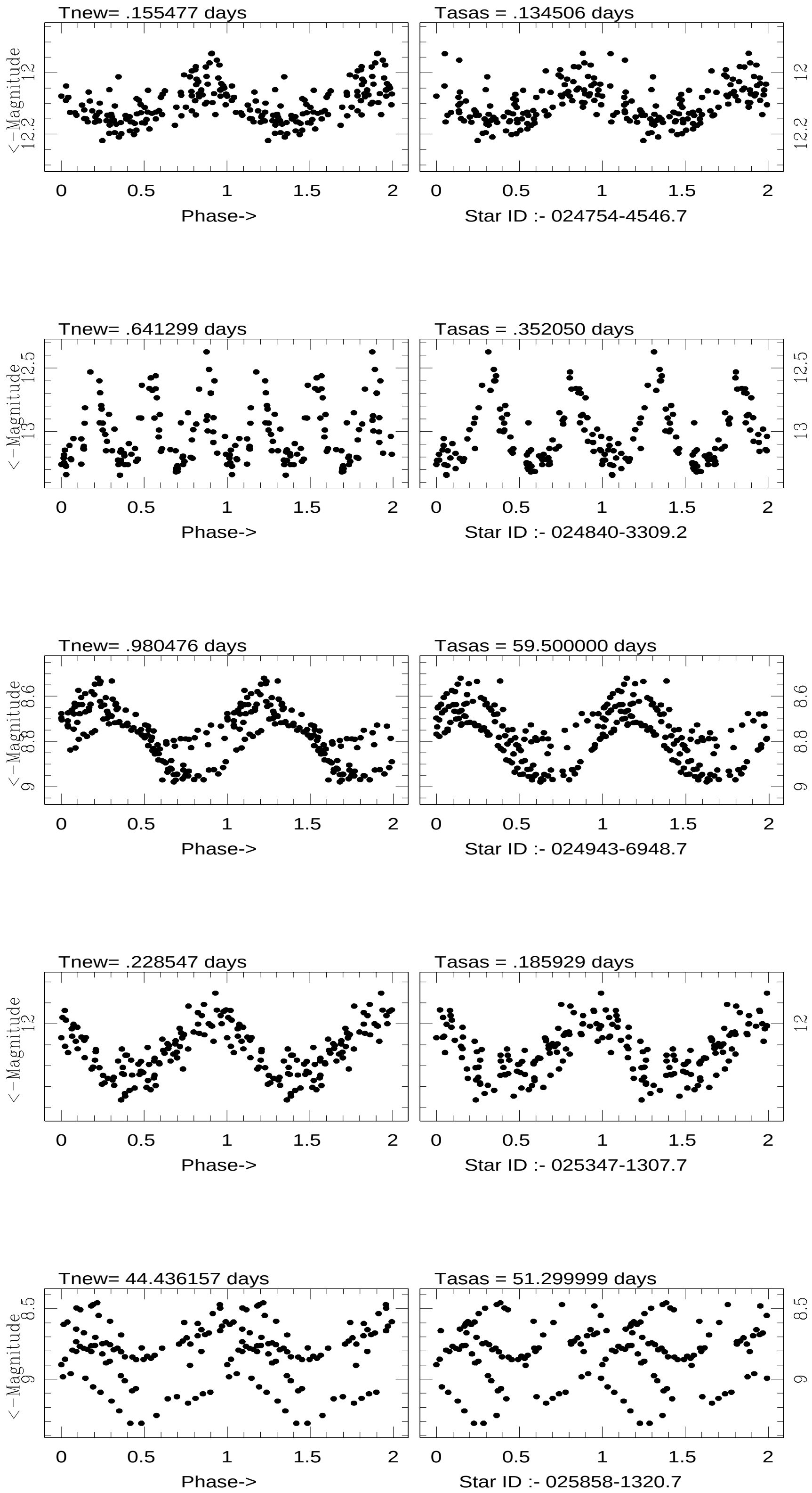

Fig. 4.4 continued...

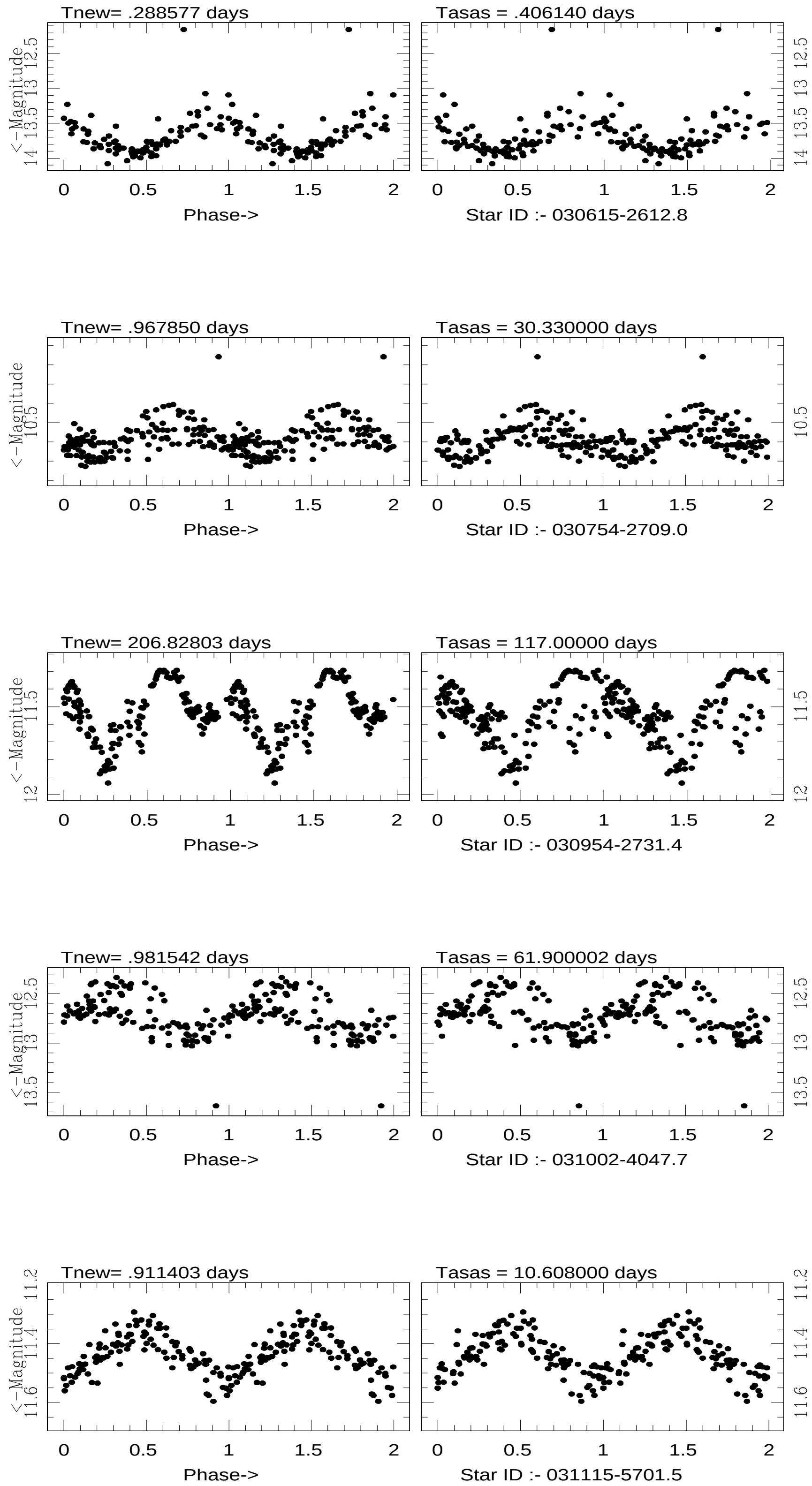

Fig. 4.4 continued...

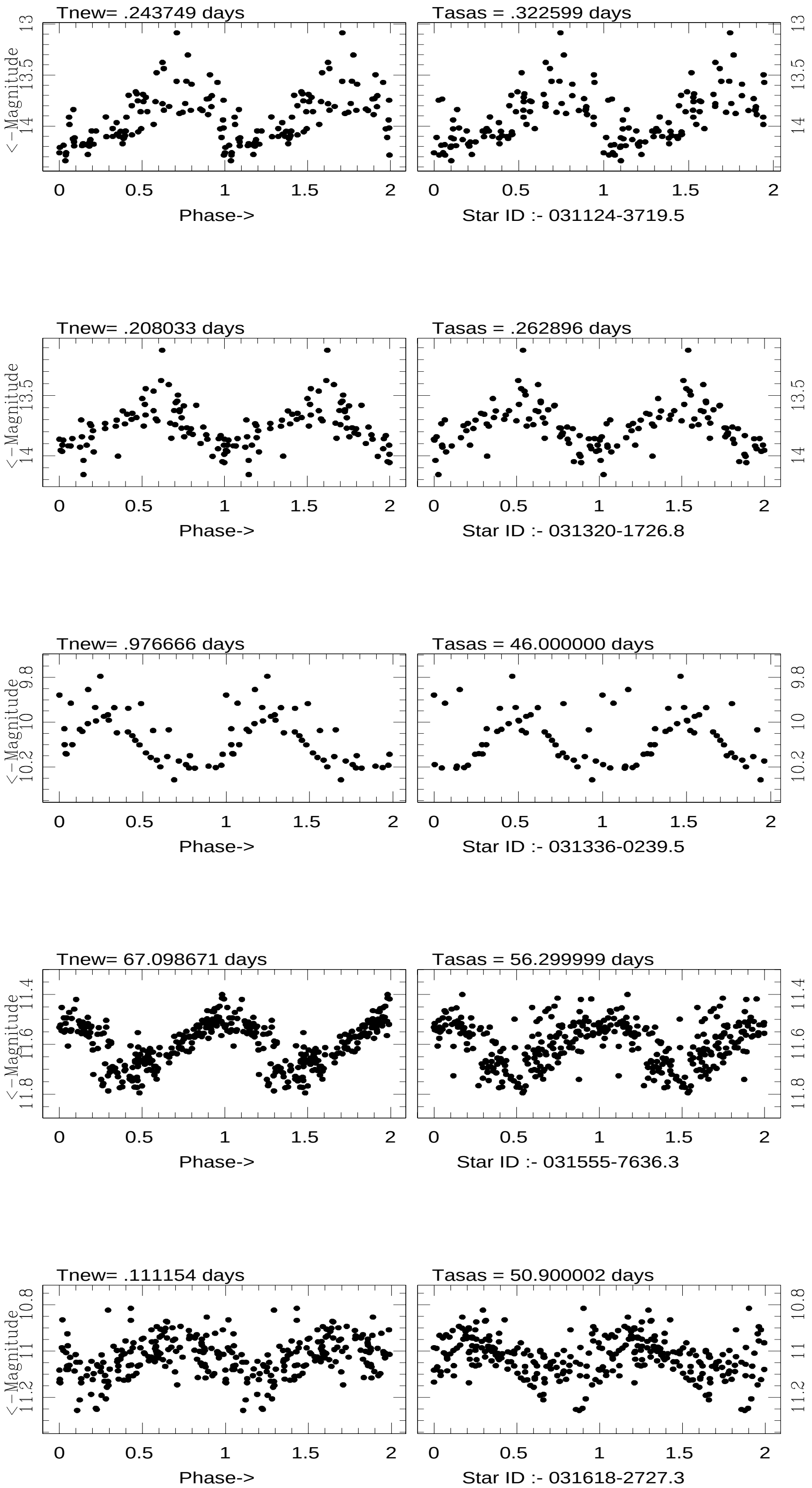

Fig. 4.4 continued...

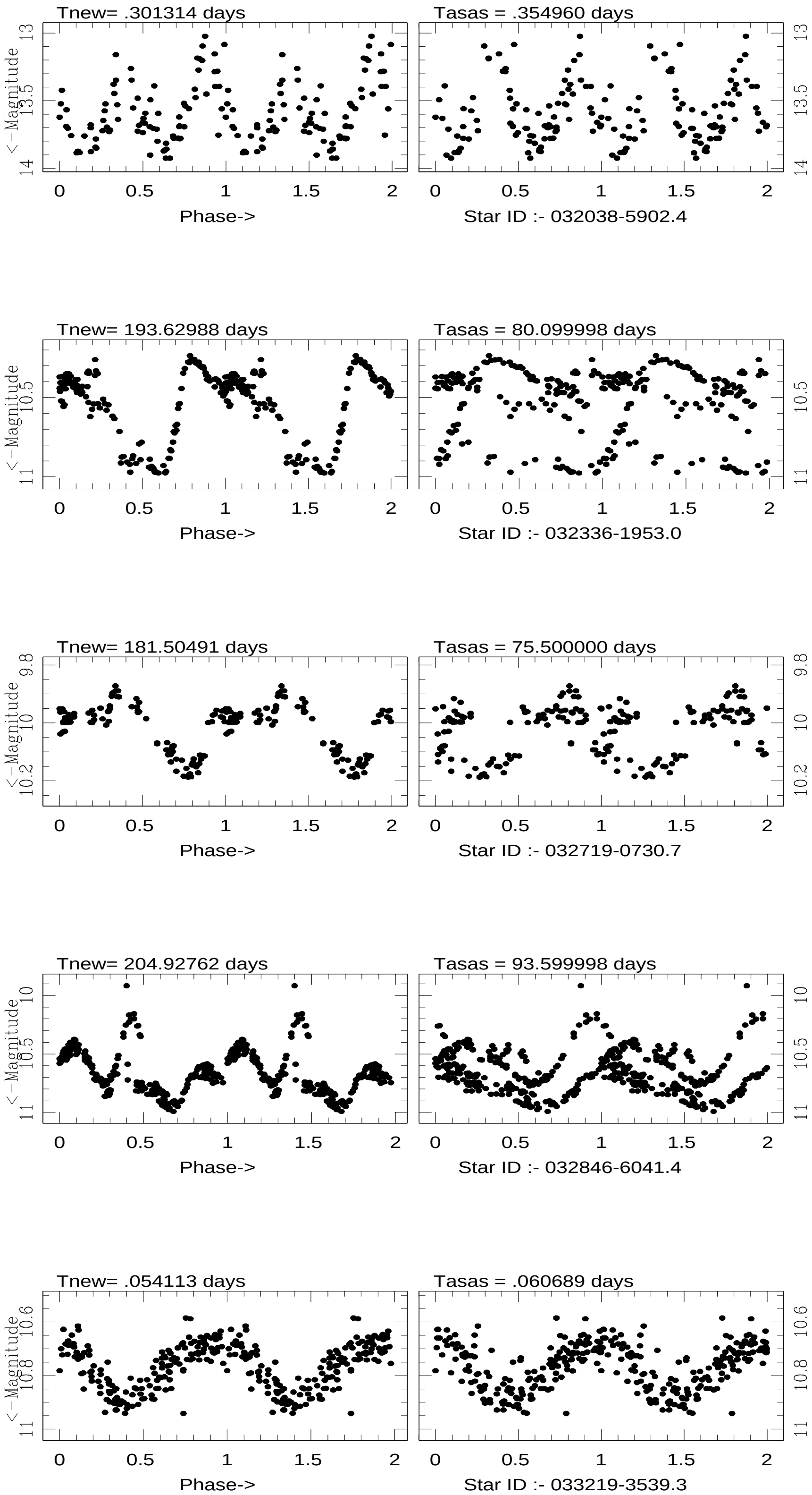

Fig. 4.4 continued...

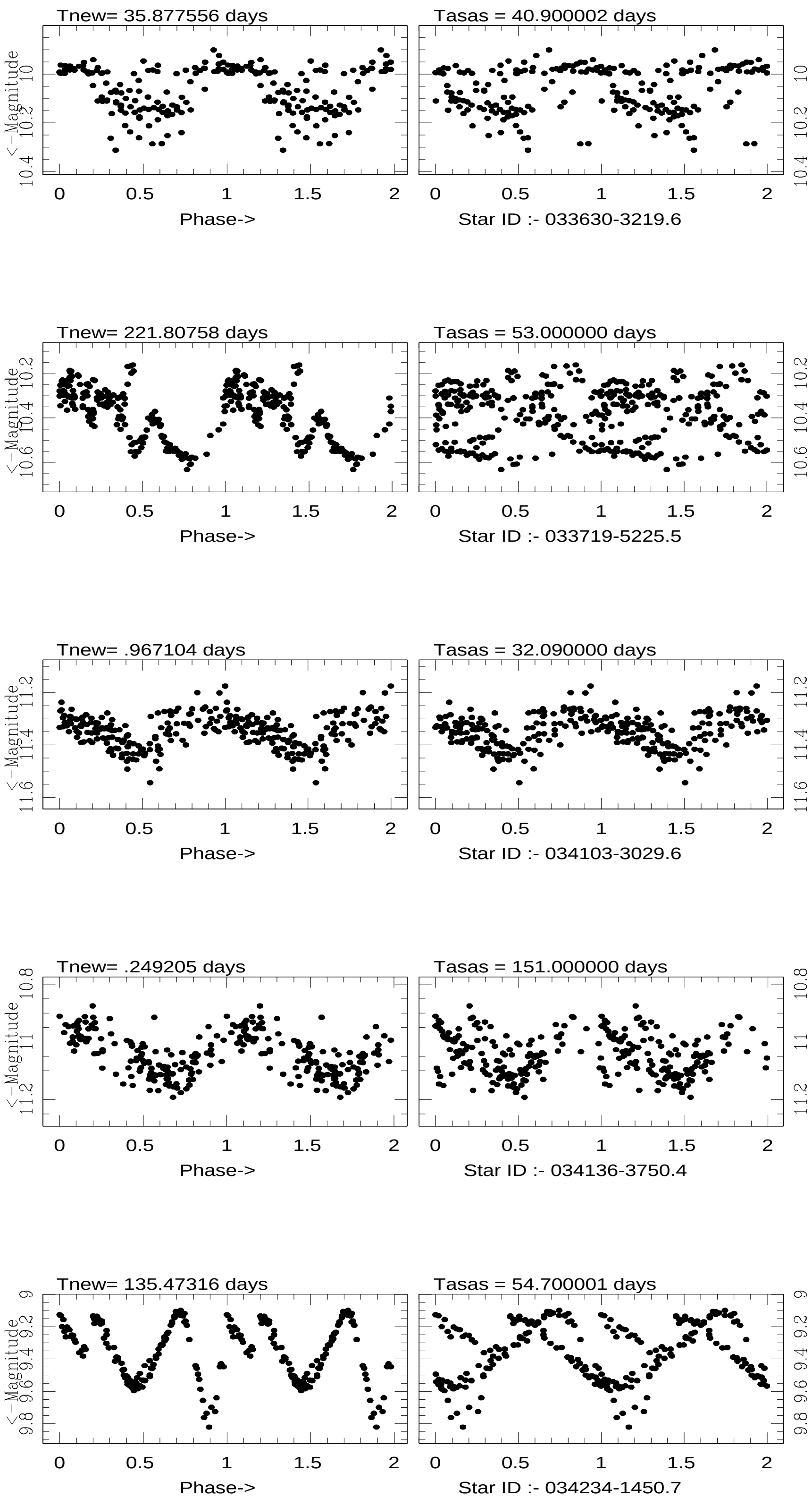

Fig. 4.4 continued...

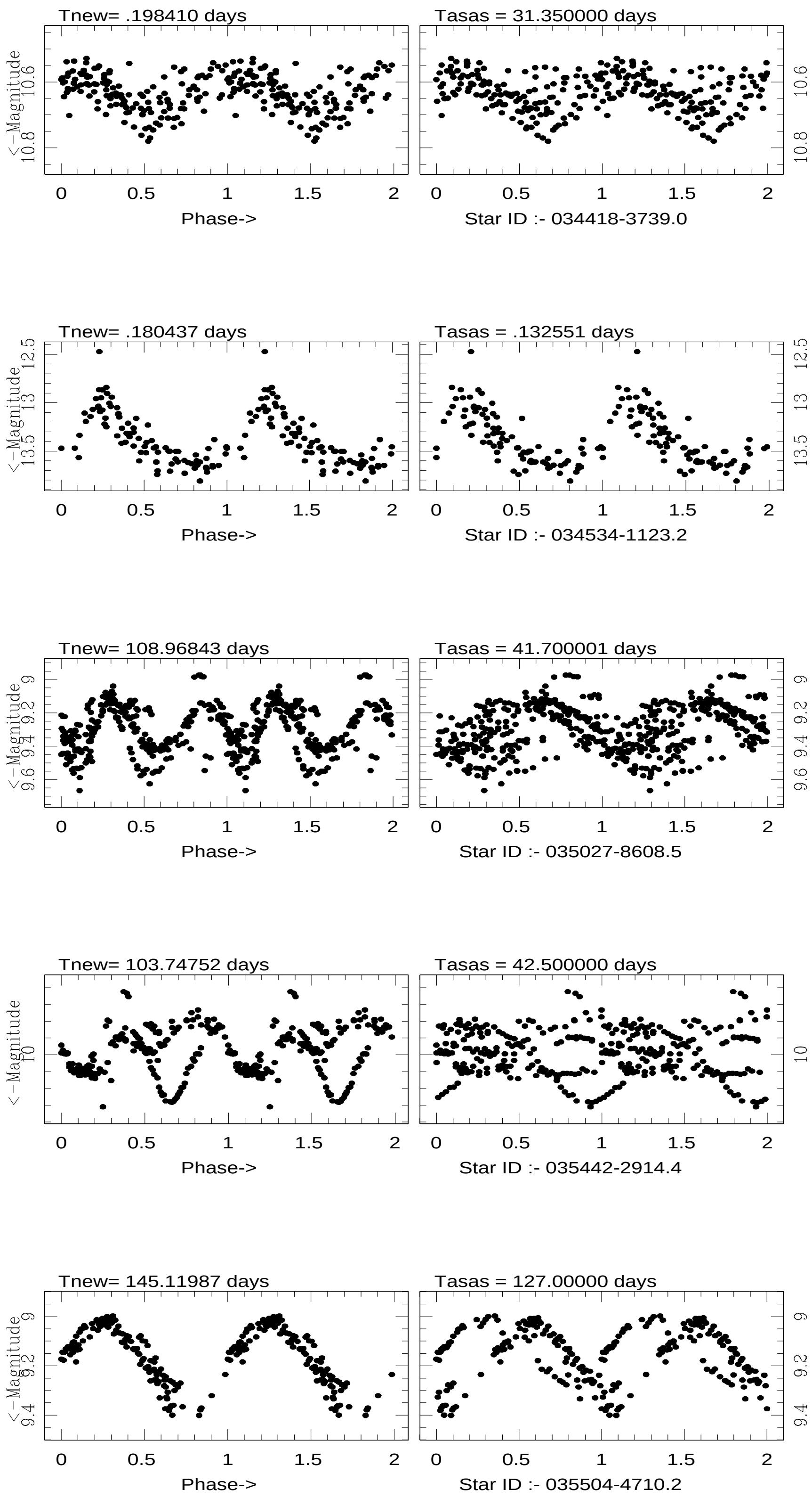

Fig. 4.4 continued...

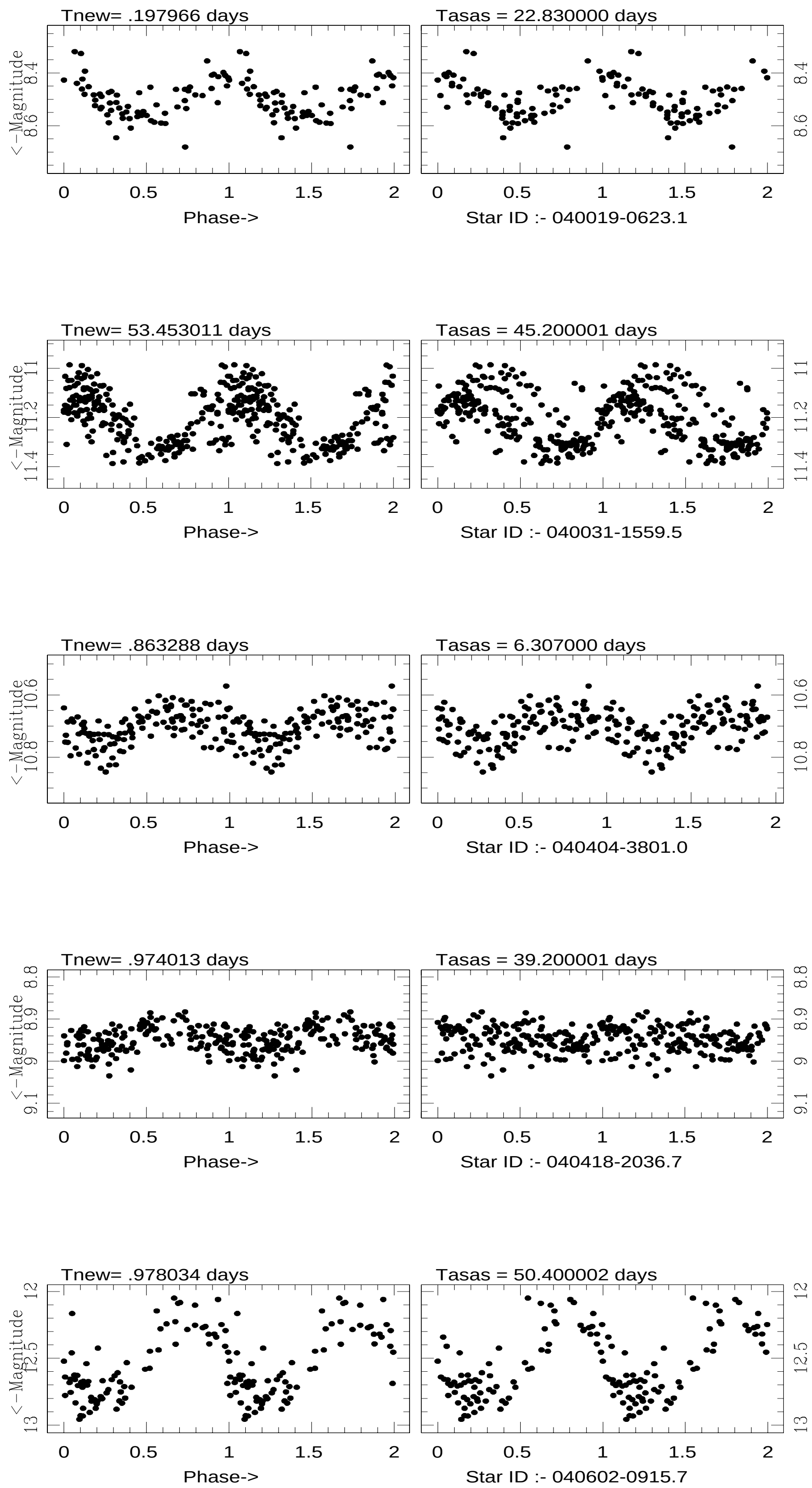

Fig. 4.4 continued...

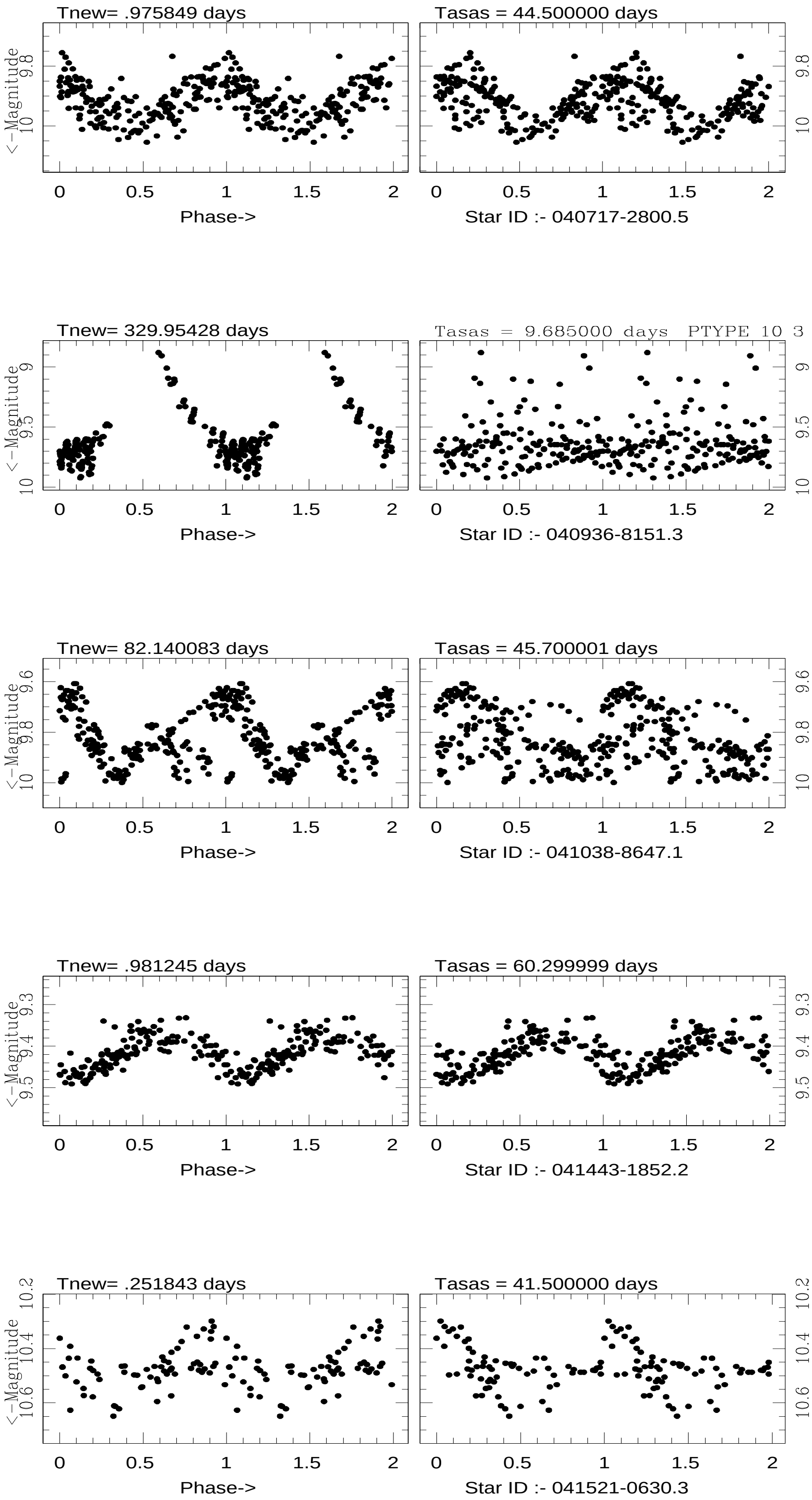

Fig. 4.4 continued...

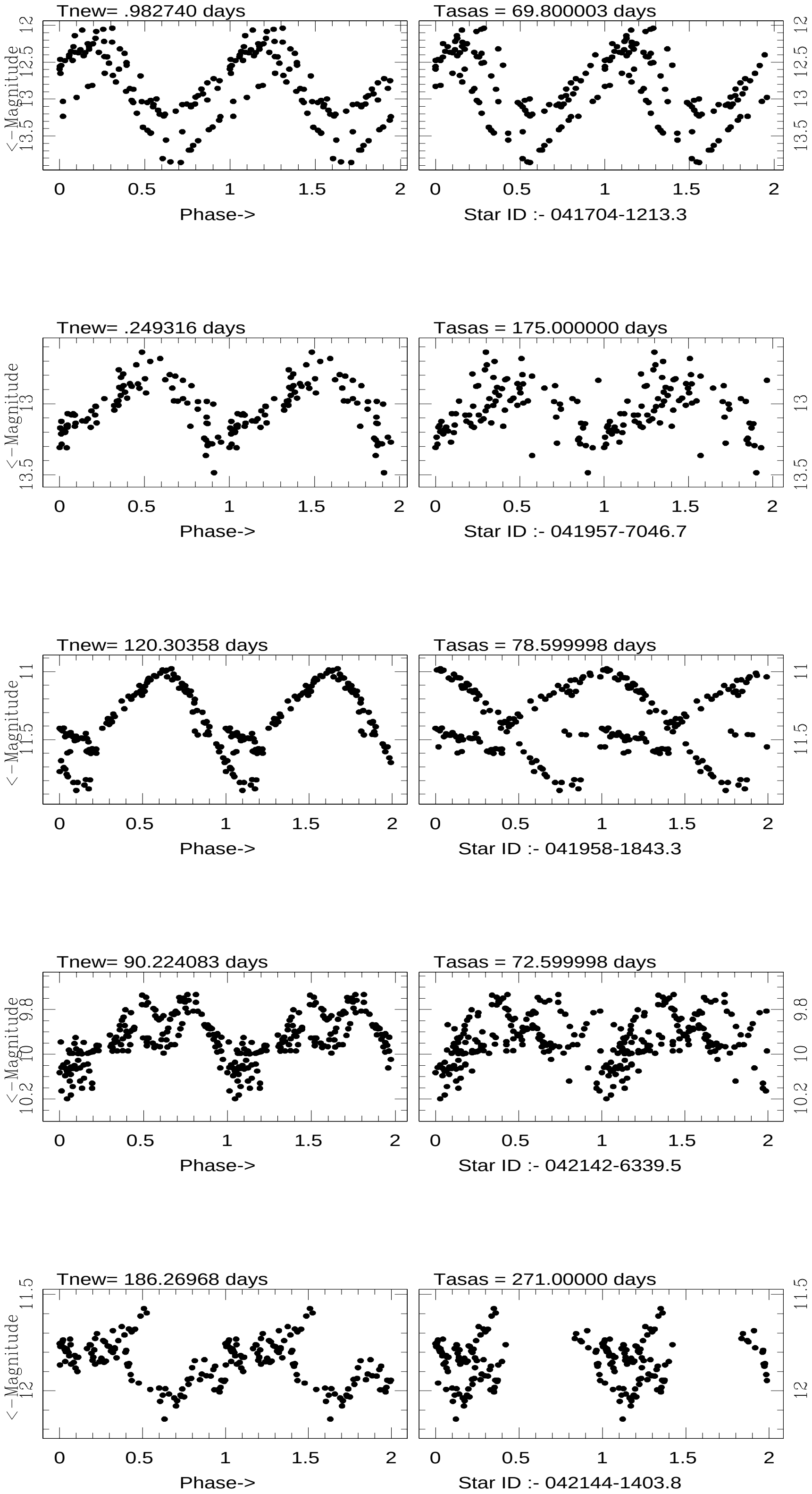

Fig. 4.4 continued...

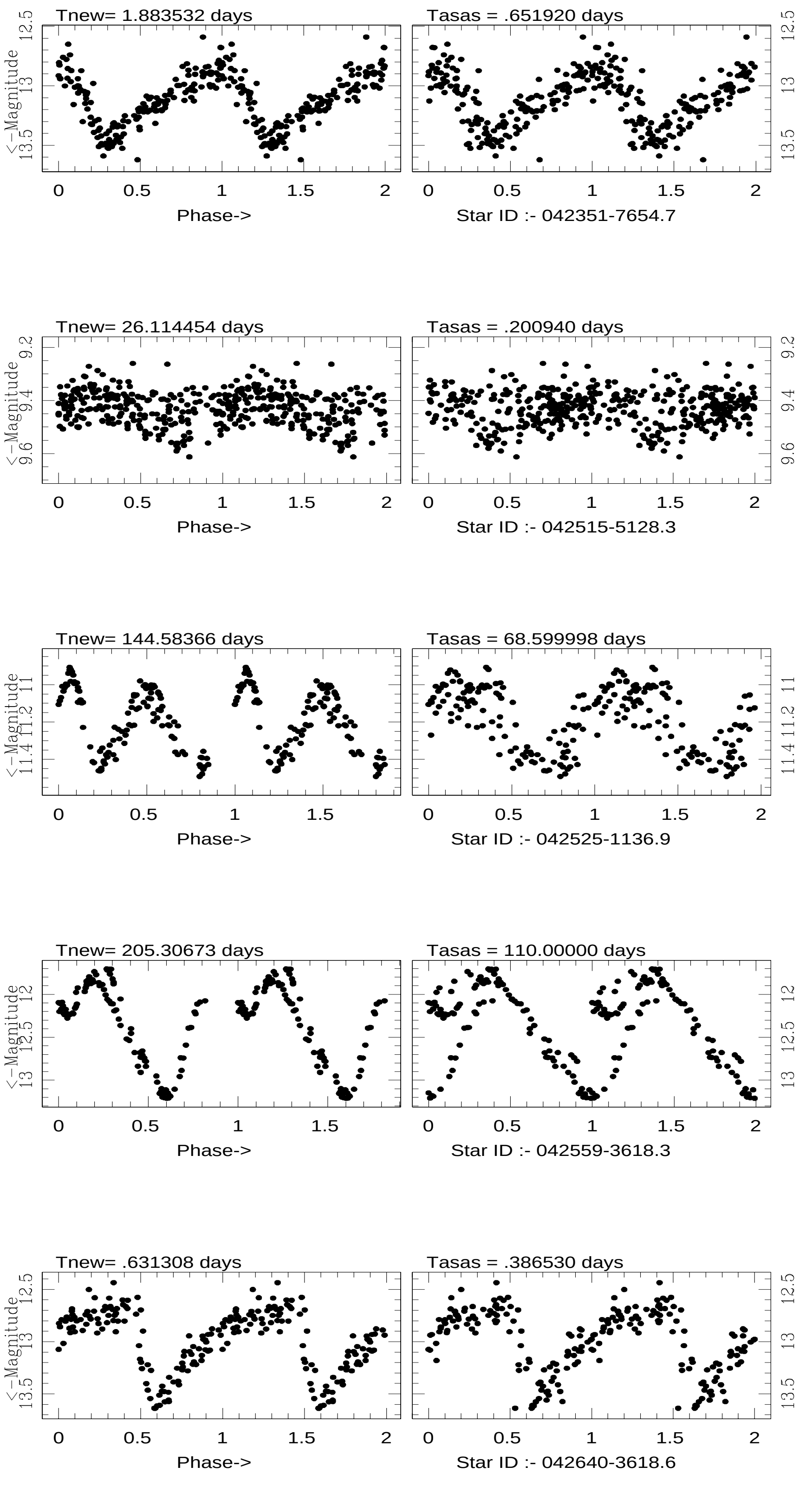

Fig. 4.4 continued...

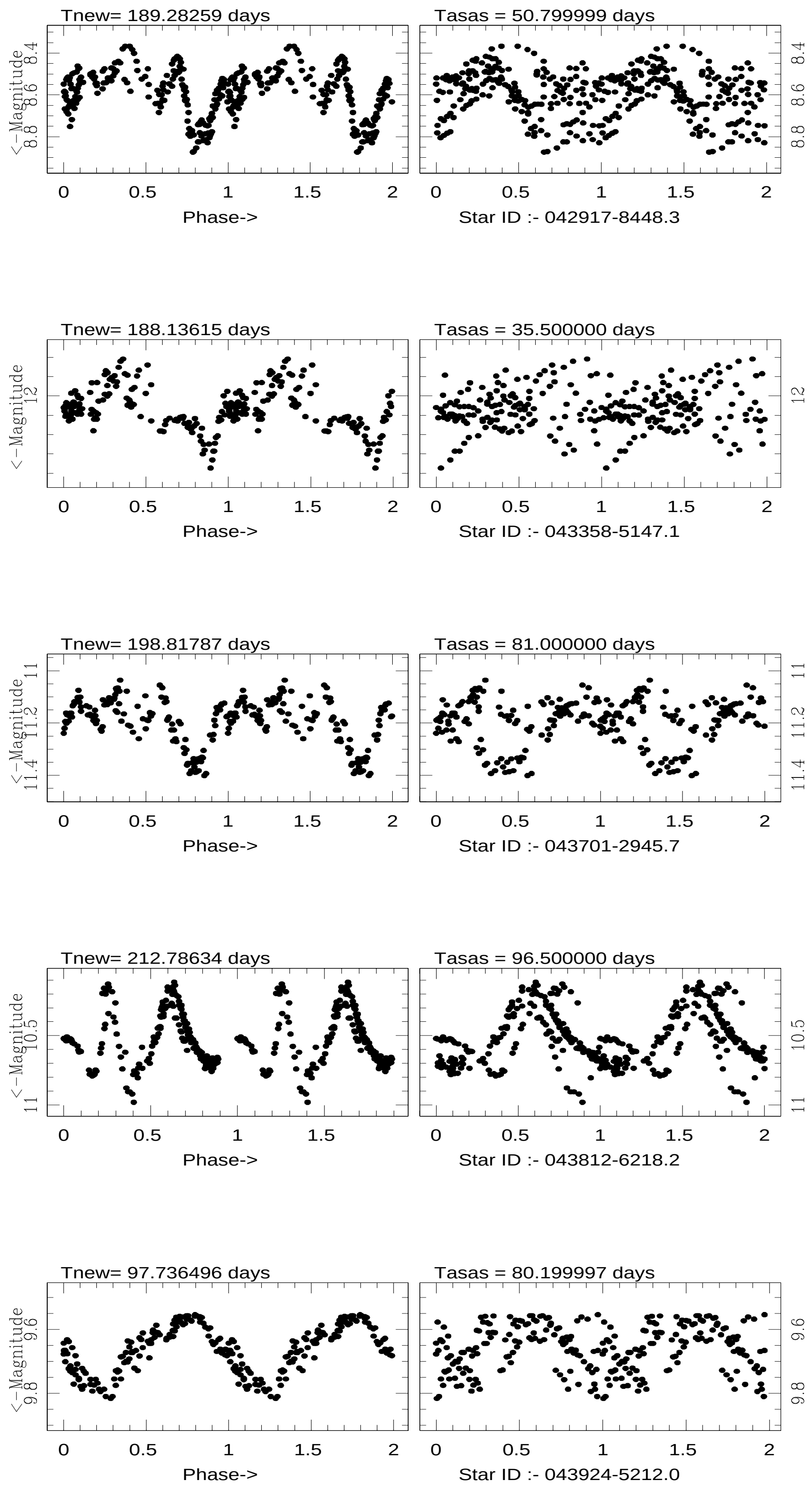

Fig. 4.4 continued...

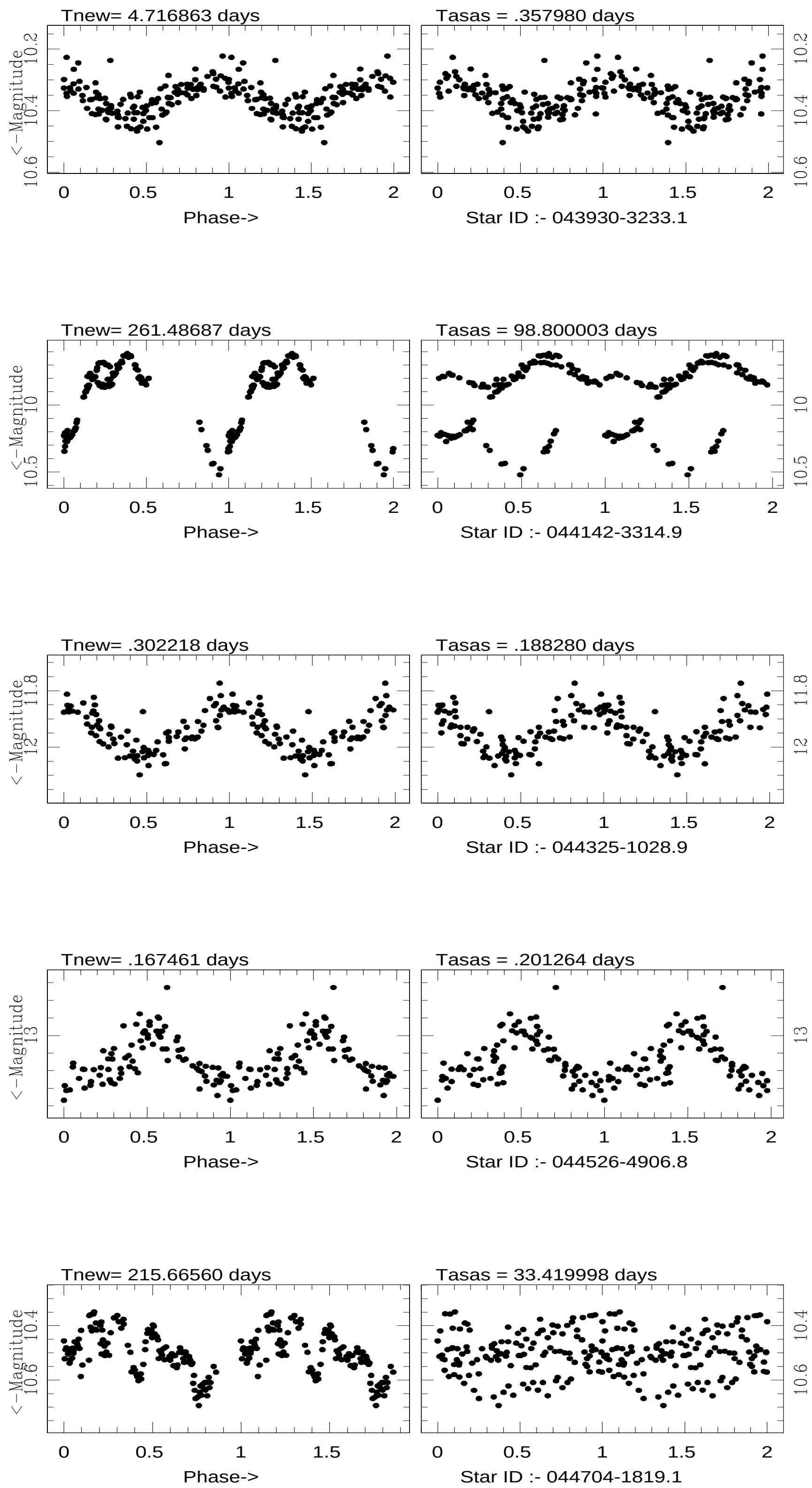

Fig. 4.4 continued...

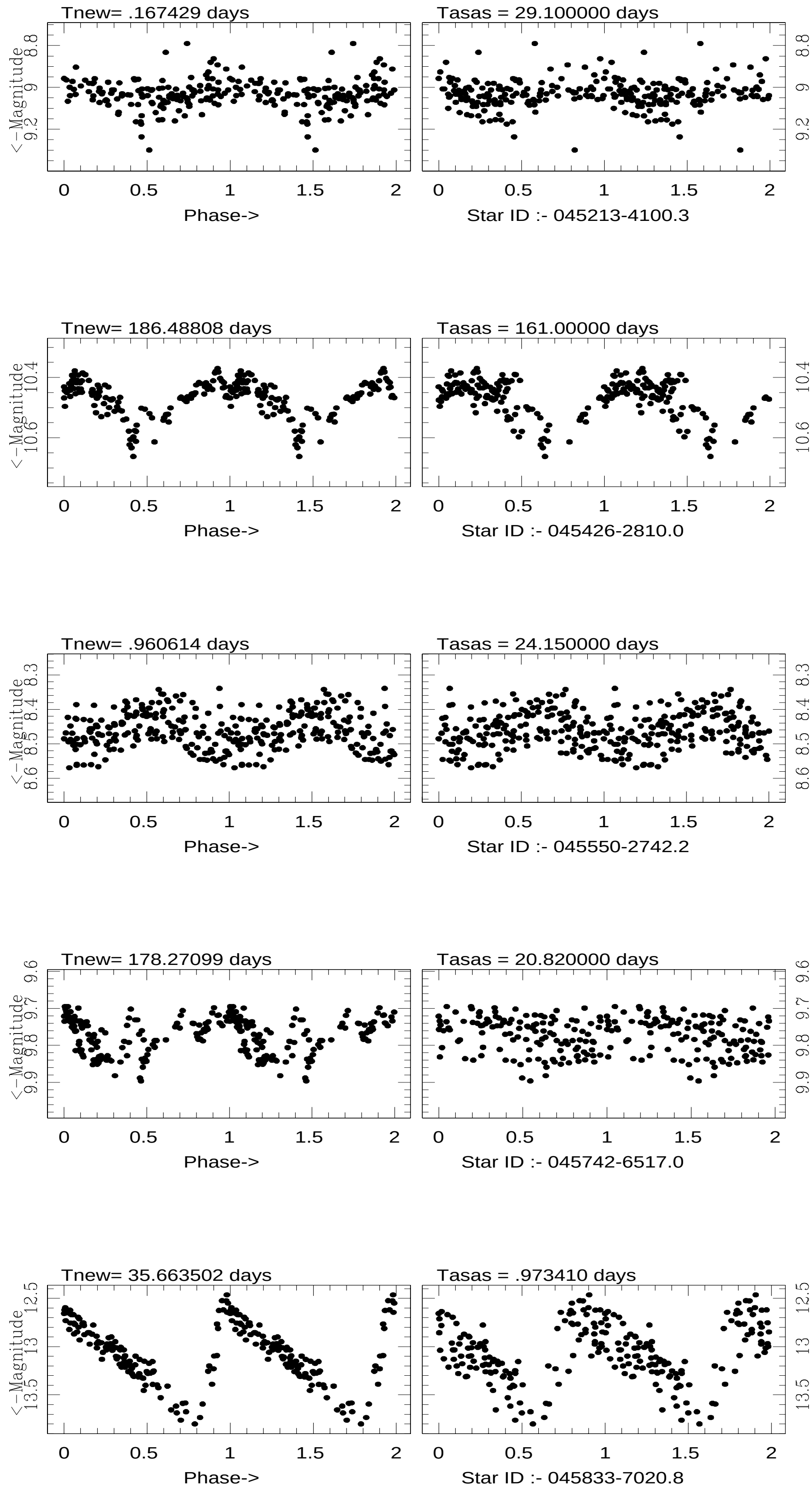

Fig. 4.4 continued...

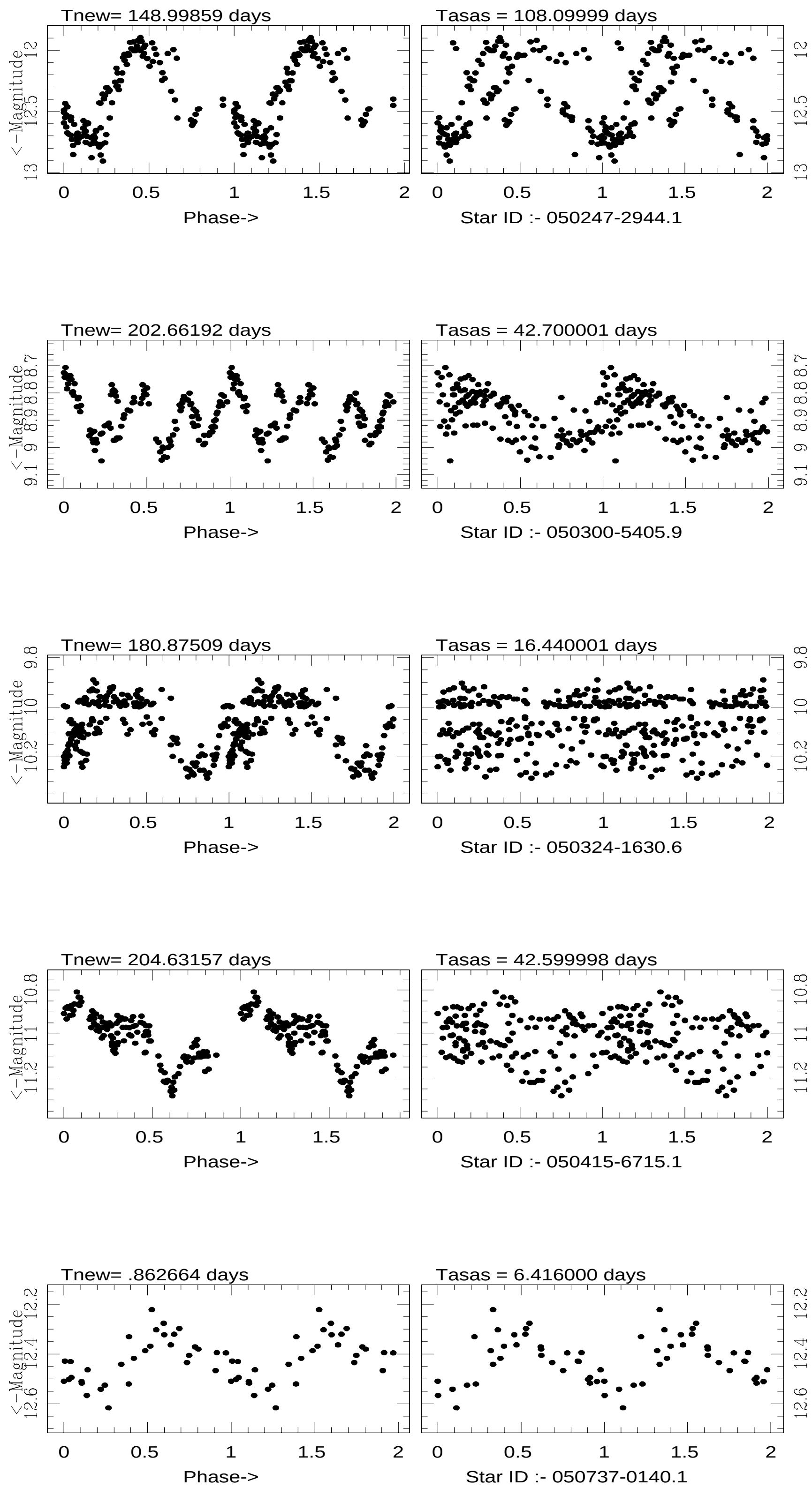

Fig. 4.4 continued...

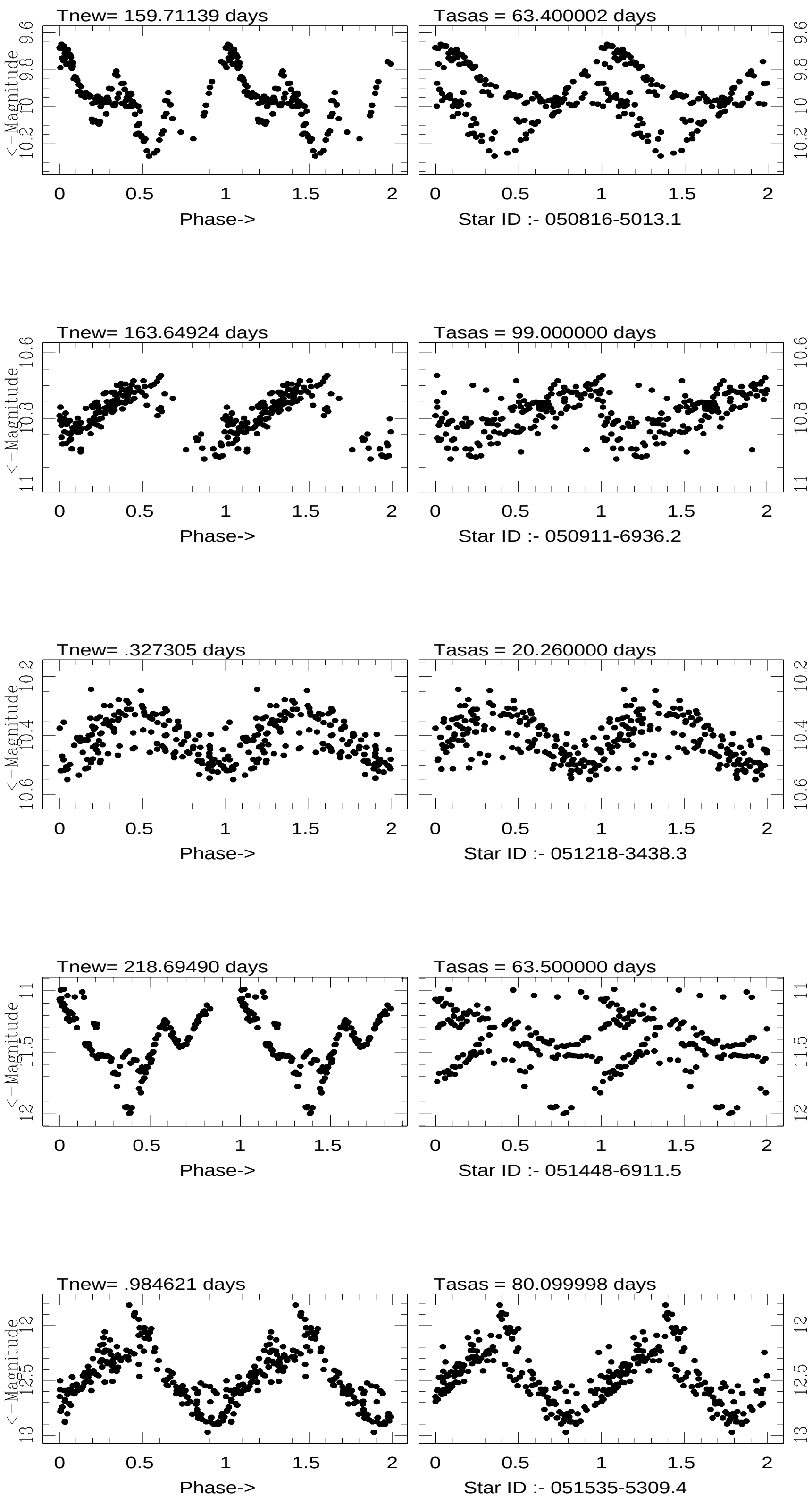

Fig. 4.4 continued...*

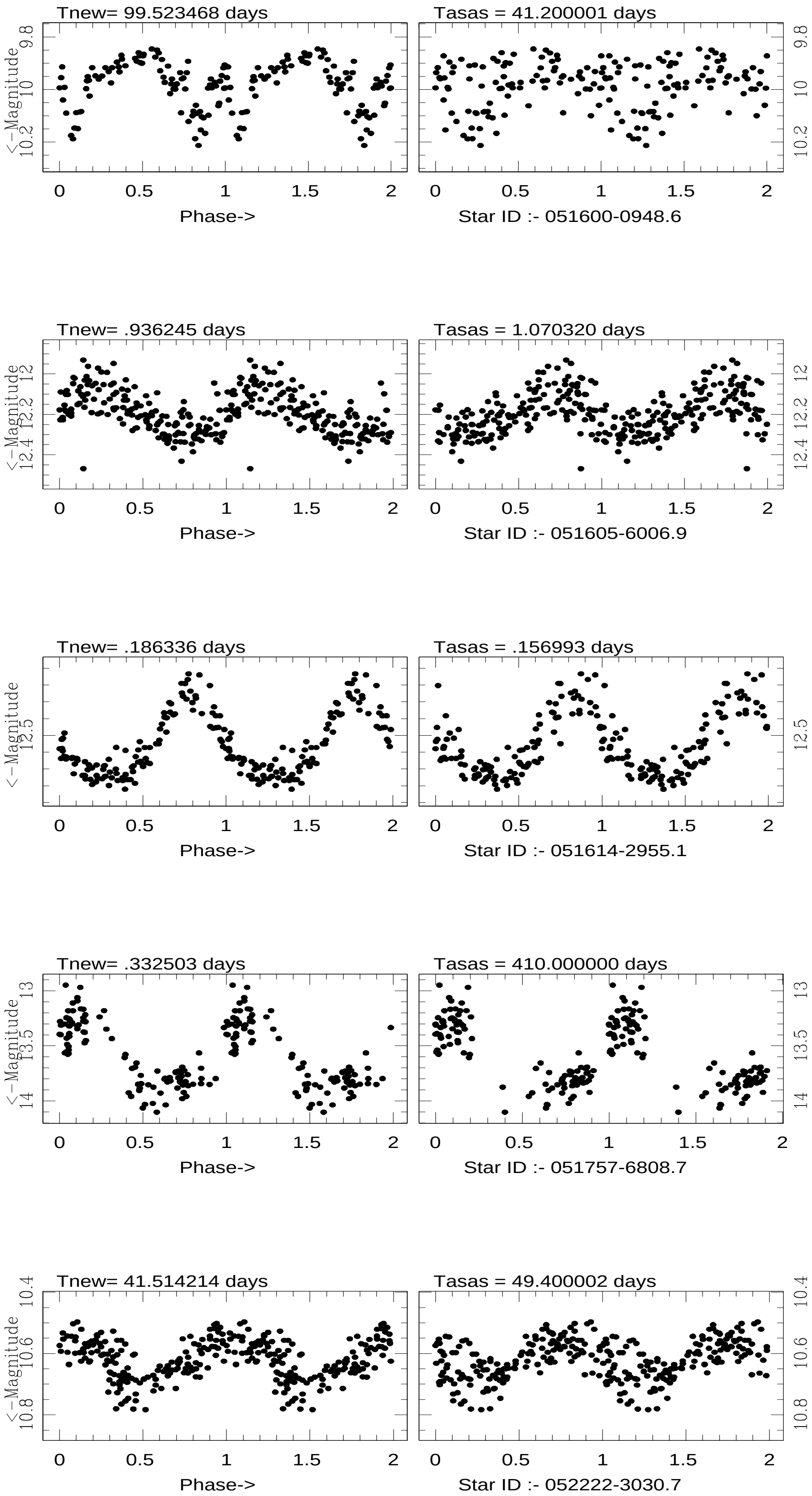

Fig. 4.4 continued...

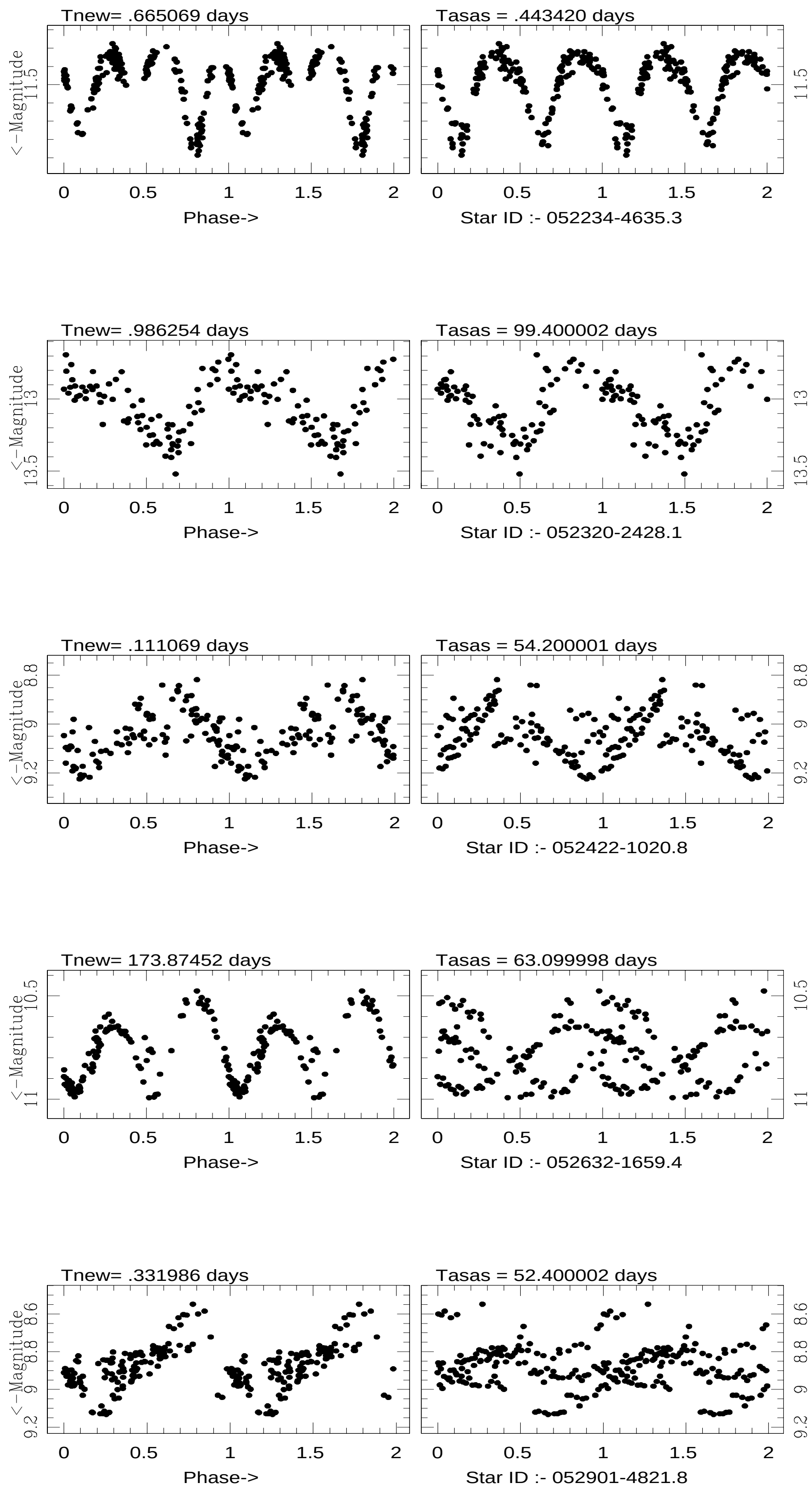

Fig. 4.4 continued...

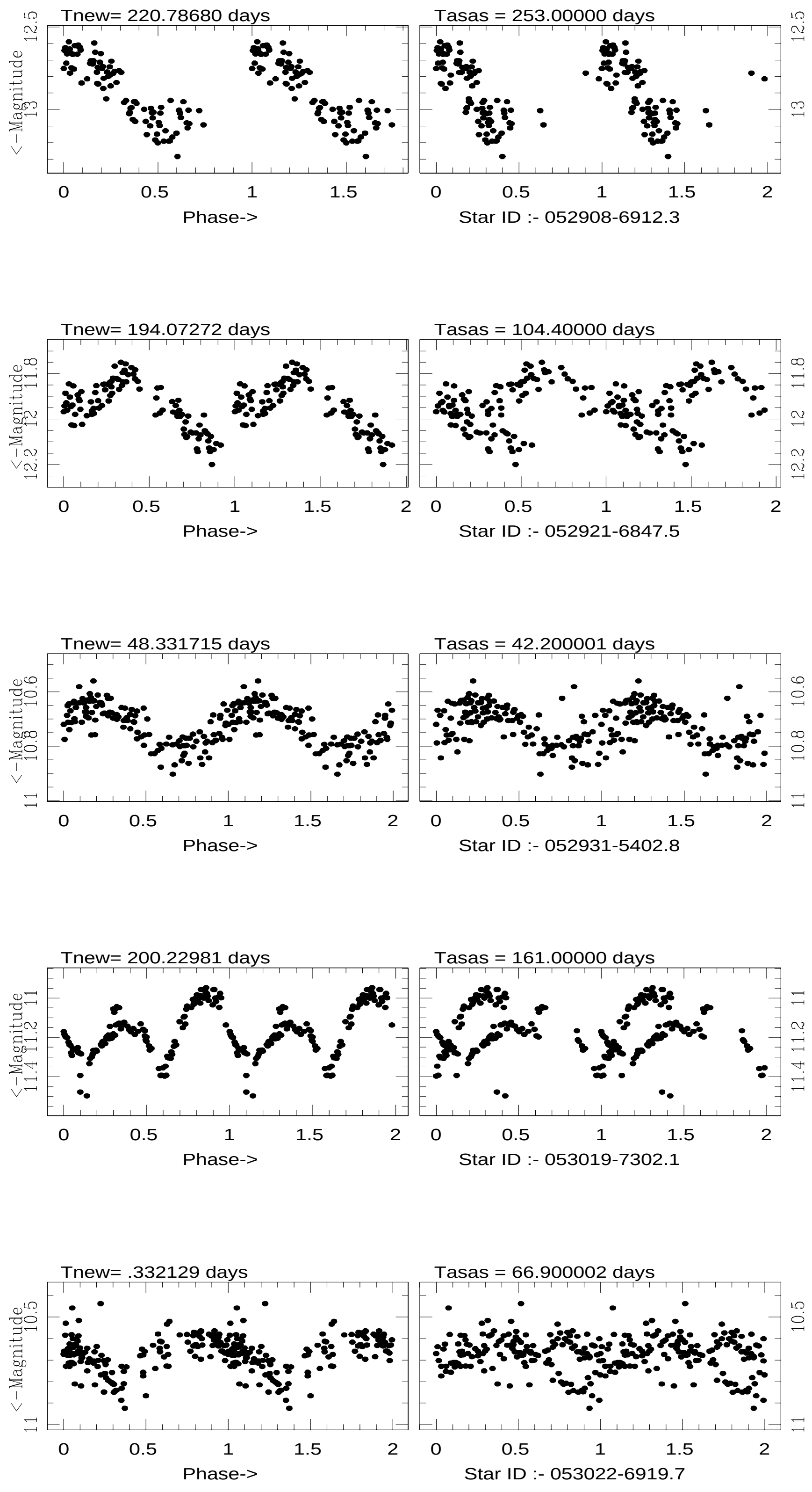

Fig. 4.4 continued...

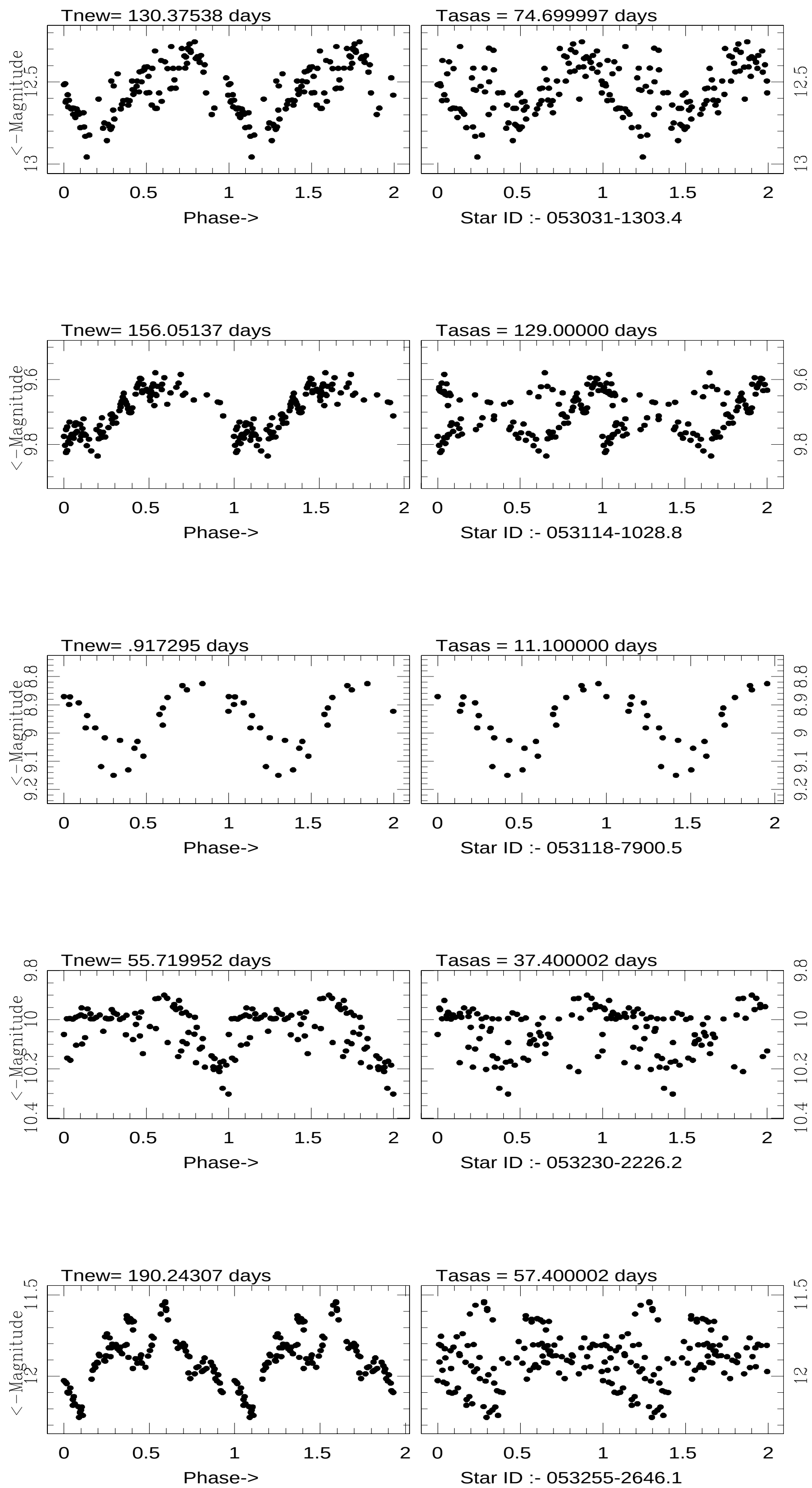

Fig. 4.4 continued...

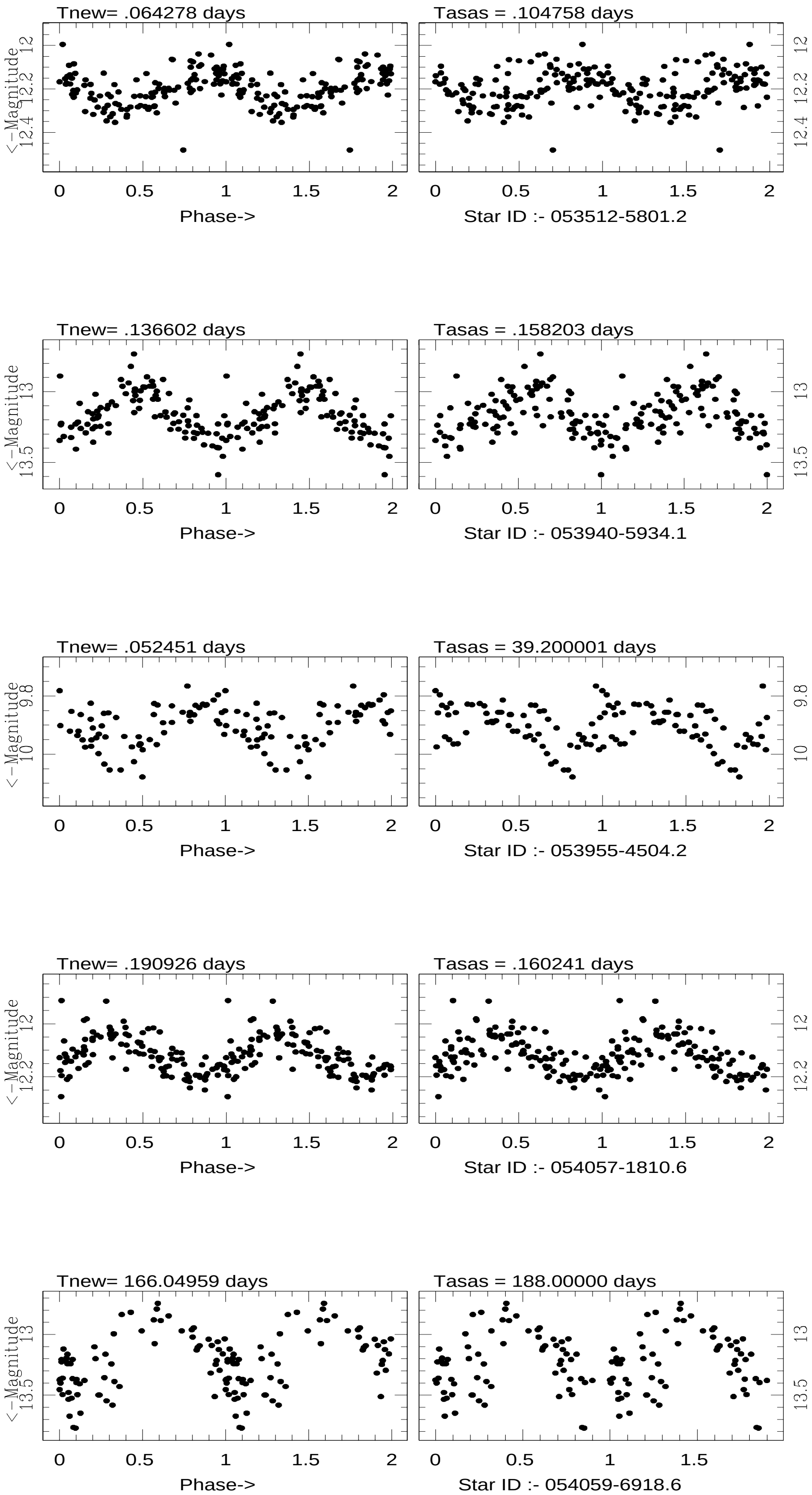

Fig. 4.4 continued...

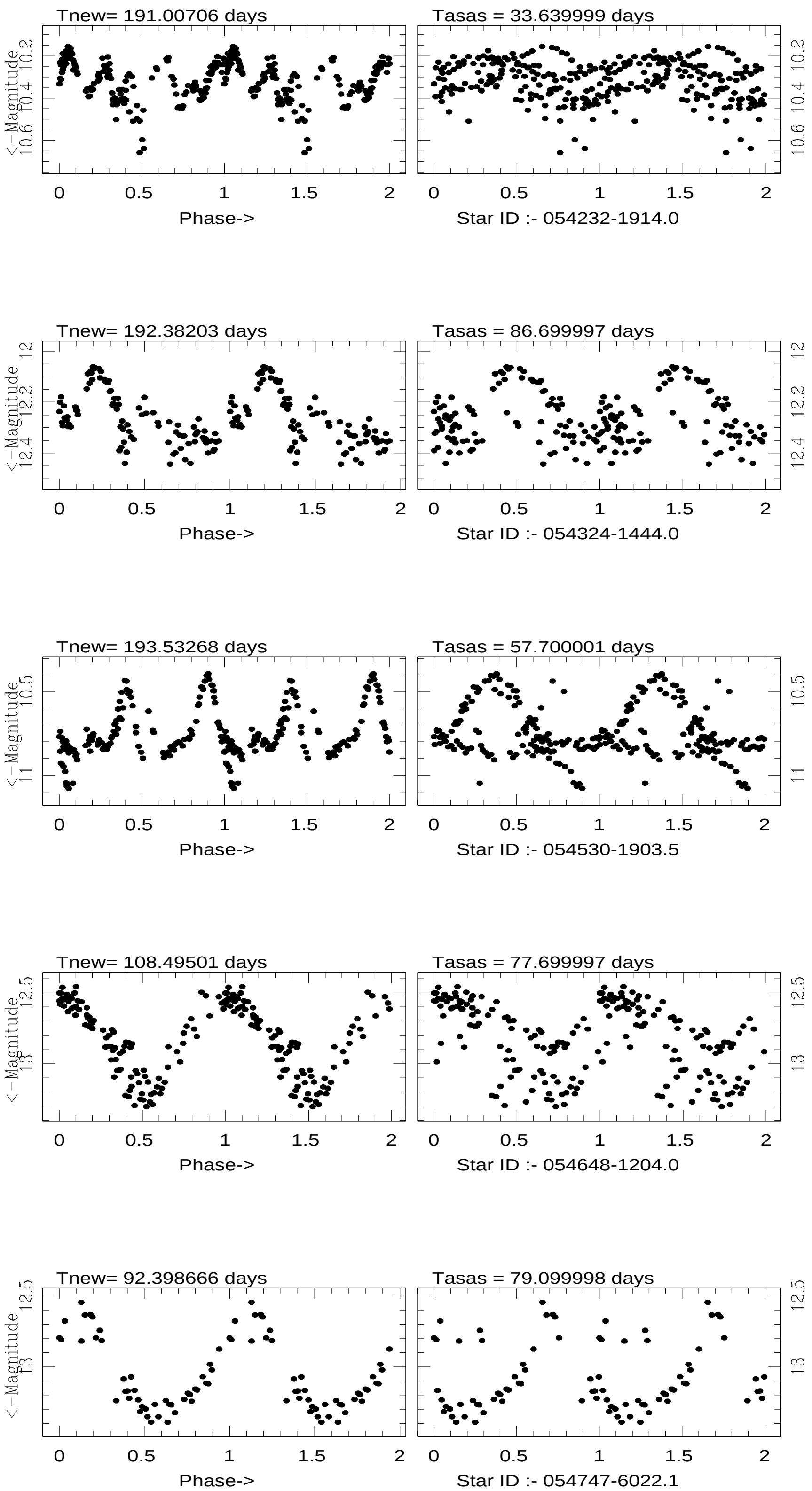

Fig. 4.4 continued...

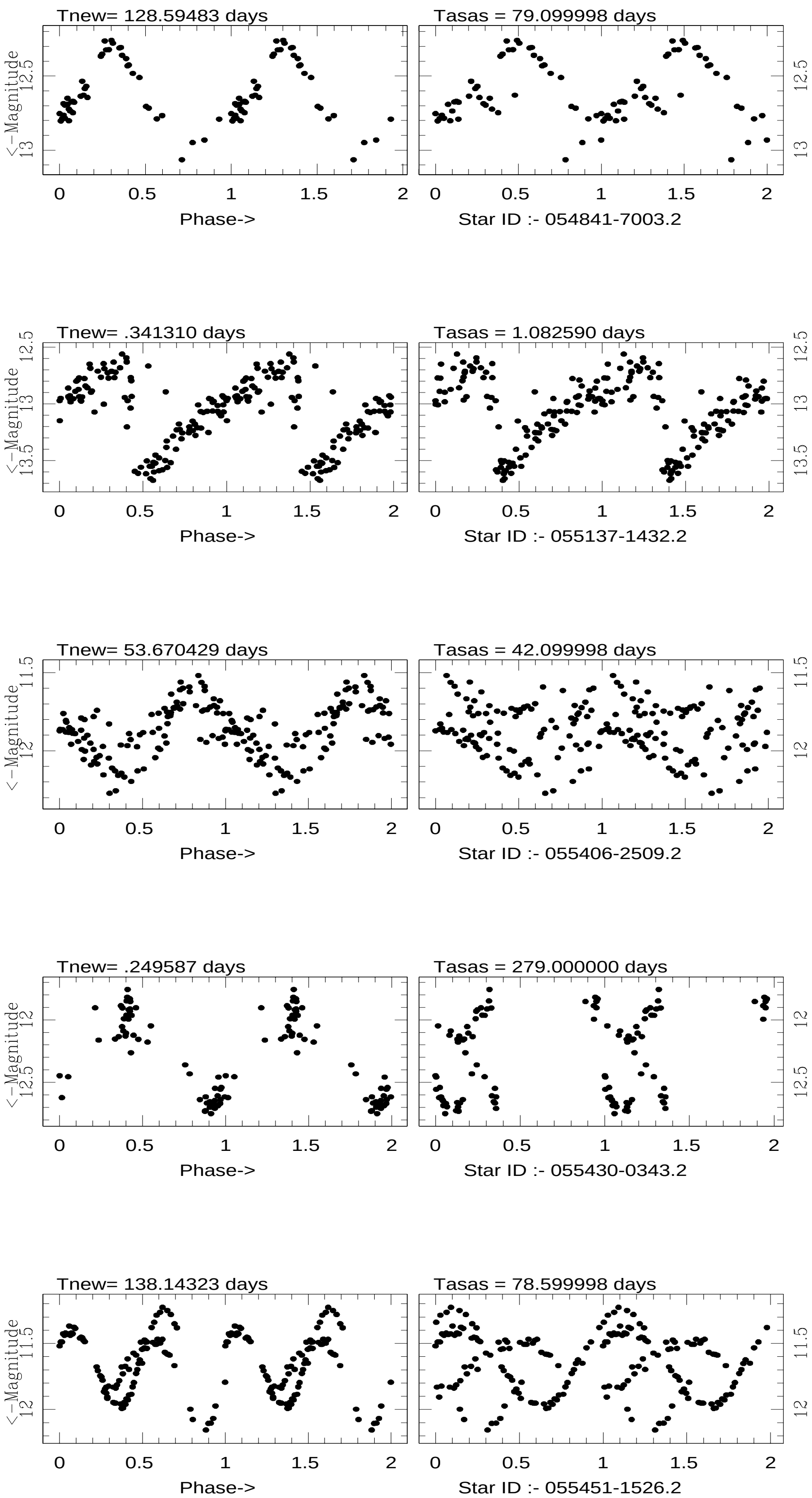

Fig. 4.4 continued...

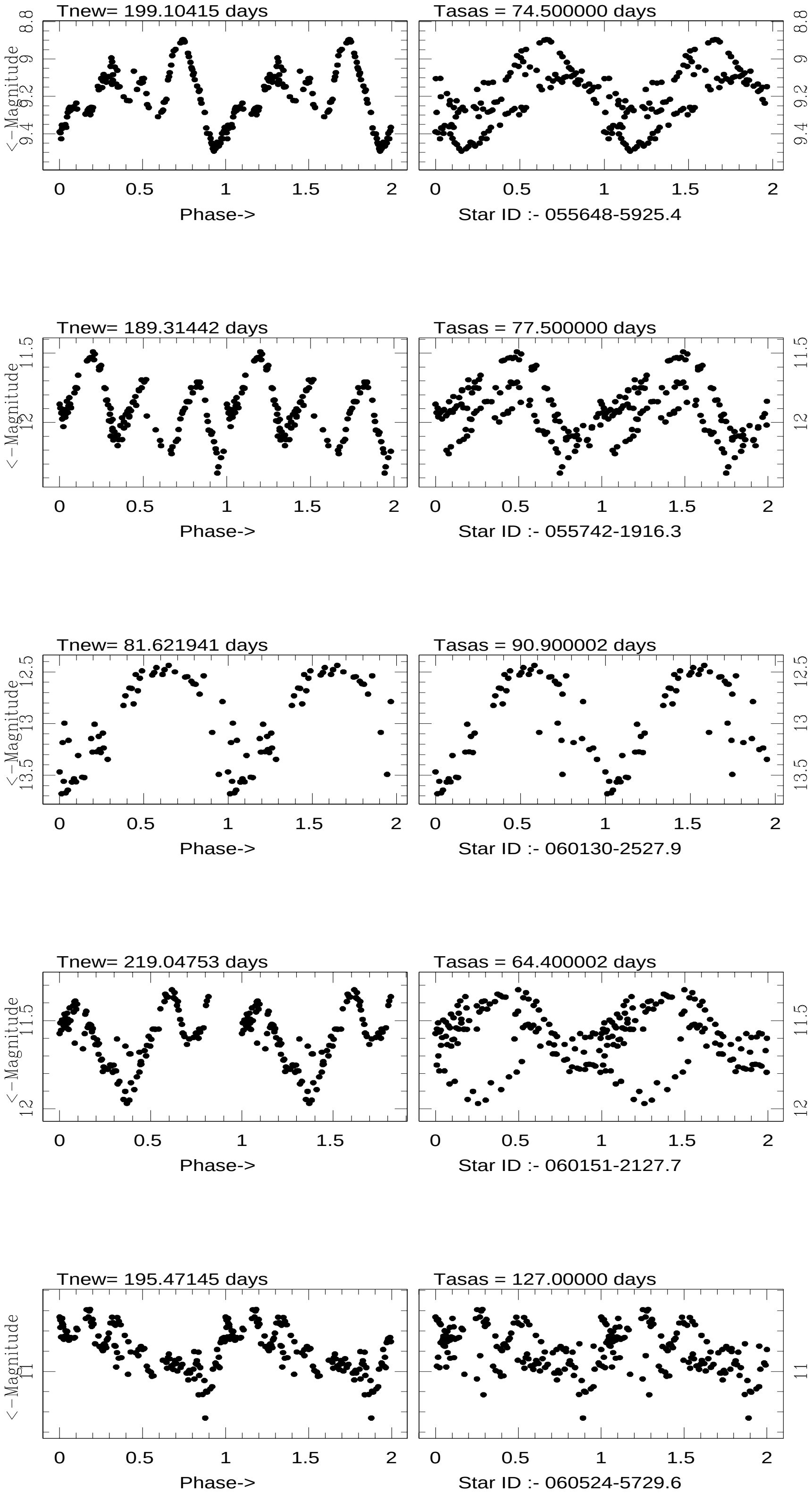

Fig. 4.4 continued...

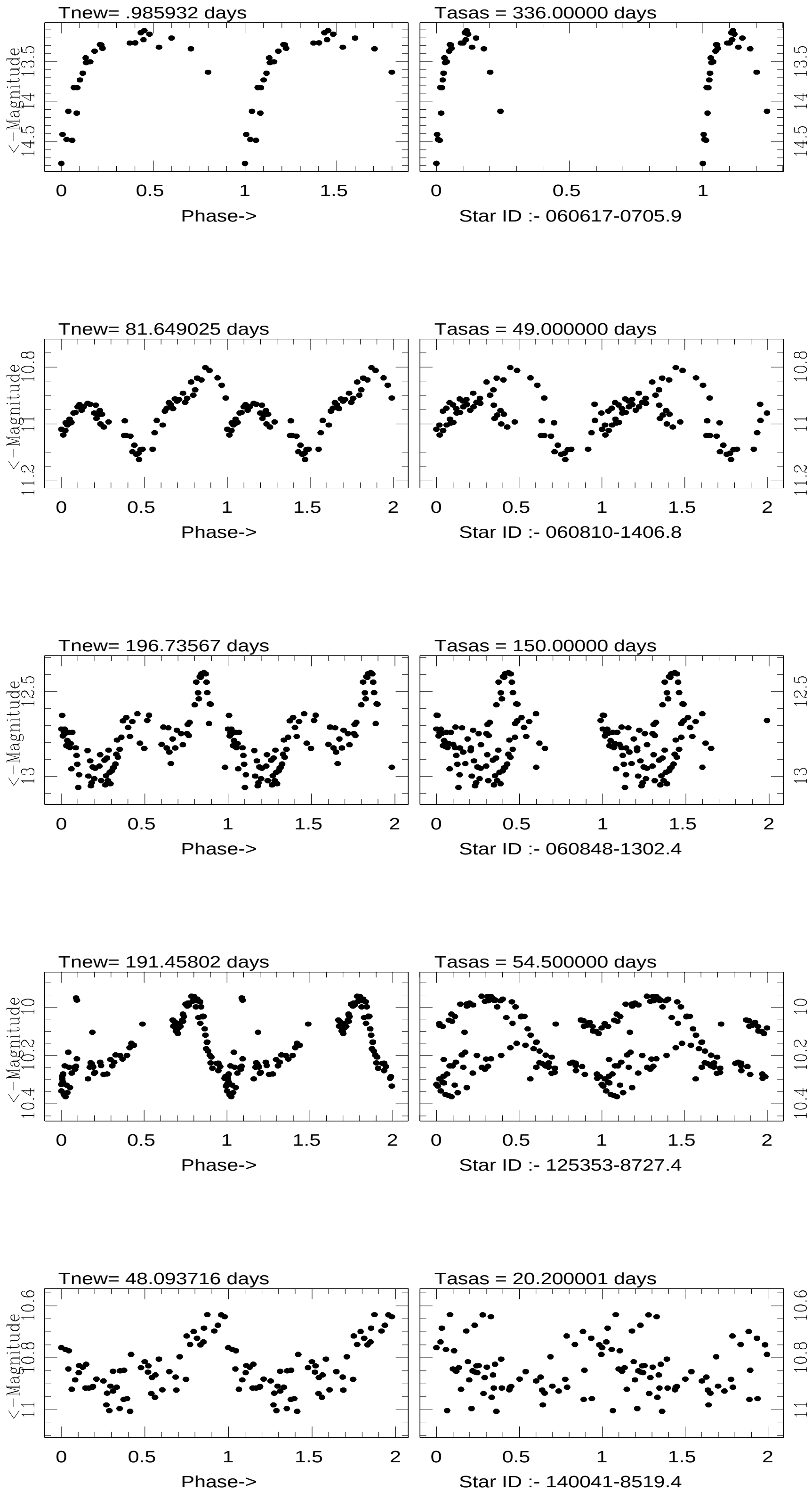

Fig. 4.4 continued...

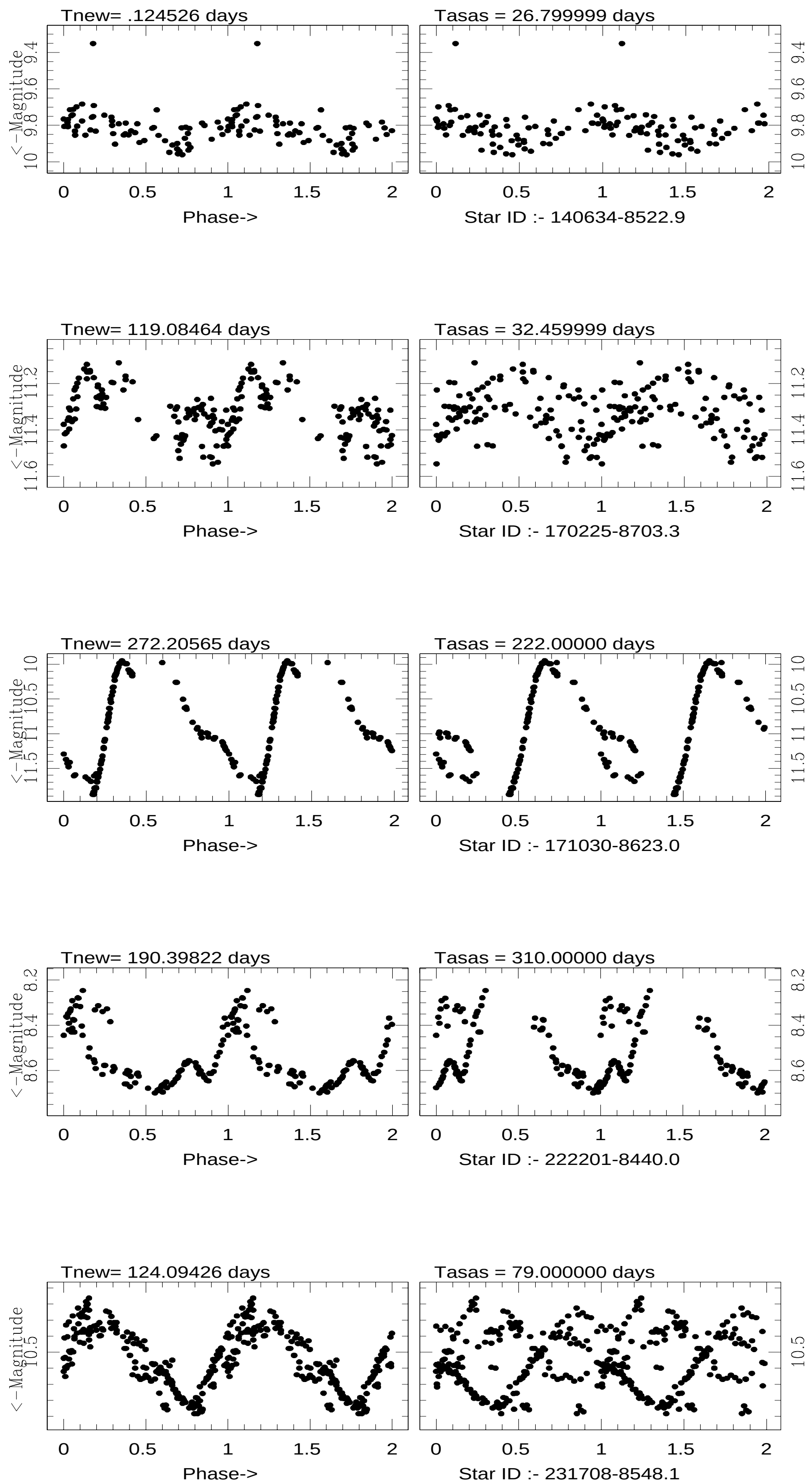

Fig. 4.4 continued...

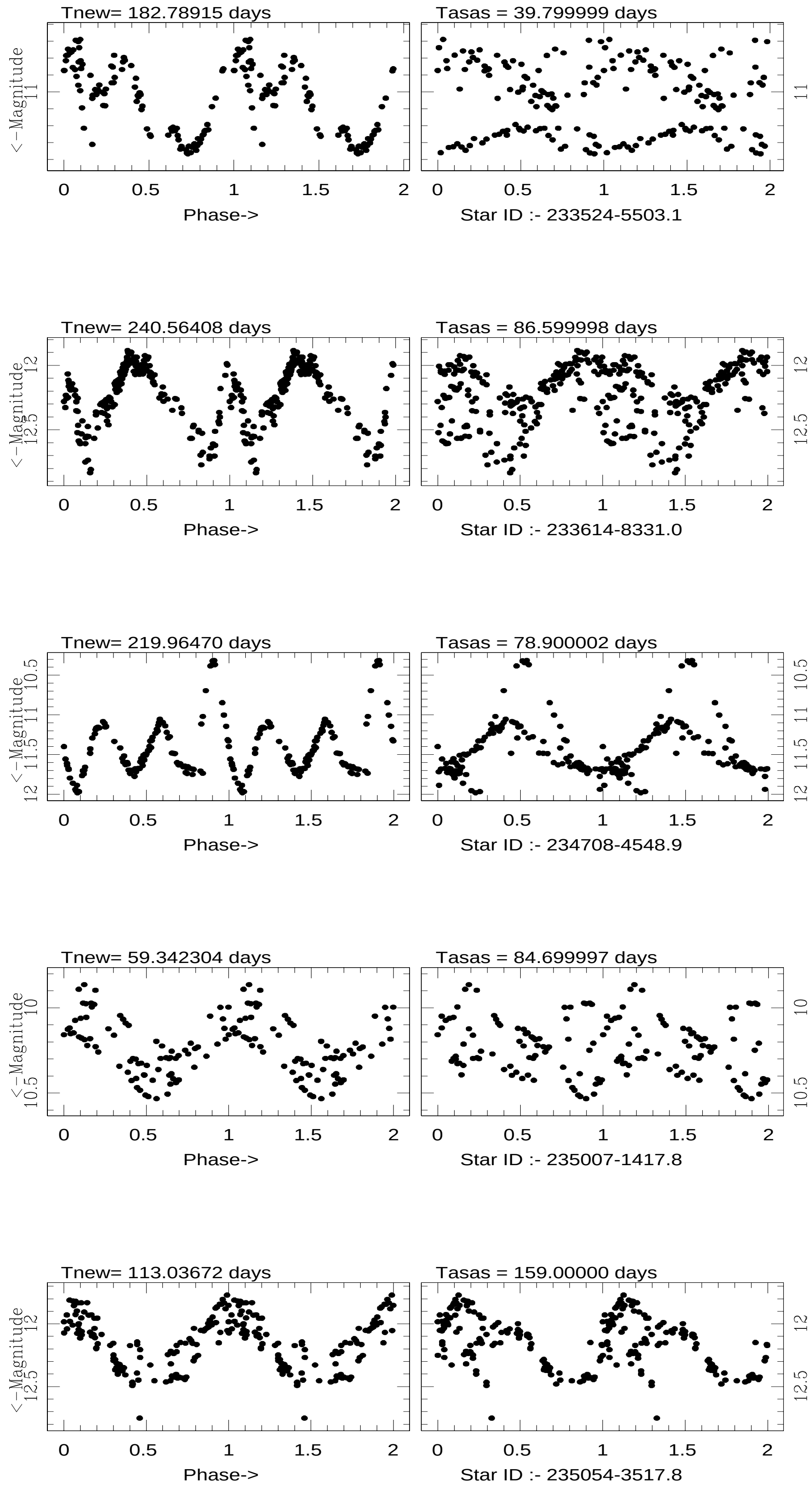

Fig. 4.4 ends here

Table 4.5: Published periods from *var3* database compared with newly detected periods for 45 variable stars with nearly harmonic periods (double, triple or half etc.). The corresponding light curves are shown in the Figure 4.5

Star−ID	$Tasas$	$Tnew$	Star−ID	$Tasas$	$Tnew$
005618−4122.0	0.335374	0.167015	051512−3523.6	0.245344	0.482269
051714−1142.0	0.236170	0.121319	053022−3234.8	0.233120	0.137078
055322−5417.9	0.245298	0.141195	000142−4229.3	32.43999	261.3117
001201−2434.0	101.1999	198.0760	001215−8312.0	27.290001	108.9761
010029−4404.2	0.325583	0.651159	010520−8152.6	48.70000	93.99372
013407−2246.7	21.24000	65.639420	013503−8255.7	49.20000	98.82673
014438−7722.2	46.50000	93.661087	014508−1914.2	2.523200	0.621234
015452−6616.0	58.29999	120.84420	015656−4149.0	2.559900	0.717422
015756−7830.5	31.15000	91.244102	020801−3619.1	66.19999	204.3466
023237−1643.6	65.19999	127.48300	030311−2660.0	66.50000	133.6750
030842−7547.4	85.69999	176.61300	032512−7839.9	50.40000	100.4208
033029−7150.4	53.50000	108.06099	033425−2537.7	106.5999	208.6163
033516−5105.8	64.50000	126.99870	040719−6026.8	65.50000	134.9145
042401−6707.1	99.90000	204.60330	042756−2912.8	49.7999	100.7442
043658−2403.0	54.79999	110.48280	051135−3120.4	21.37000	193.3045
051952−1752.4	365.0000	173.83740	052612−5336.6	36.20000	179.9676
052802−6607.4	433.0000	197.57110	052946−6905.8	100.3000	296.7796
053022−6919.7	66.90000	205.06289	053232−6549.6	53.09999	108.9712
053246−2800.8	260.0000	133.98590	053450−2704.3	43.29999	86.10275
053451−3828.8	23.79000	48.247551	053832−4547.8	101.5999	201.2456
054754−1737.6	44.00000	89.370743	055103−1930.9	69.00000	135.1347
055717−1317.3	52.00000	107.26930	230856−8425.8	80.59999	244.2481
235010−8651.1	68.69999	137.70910			

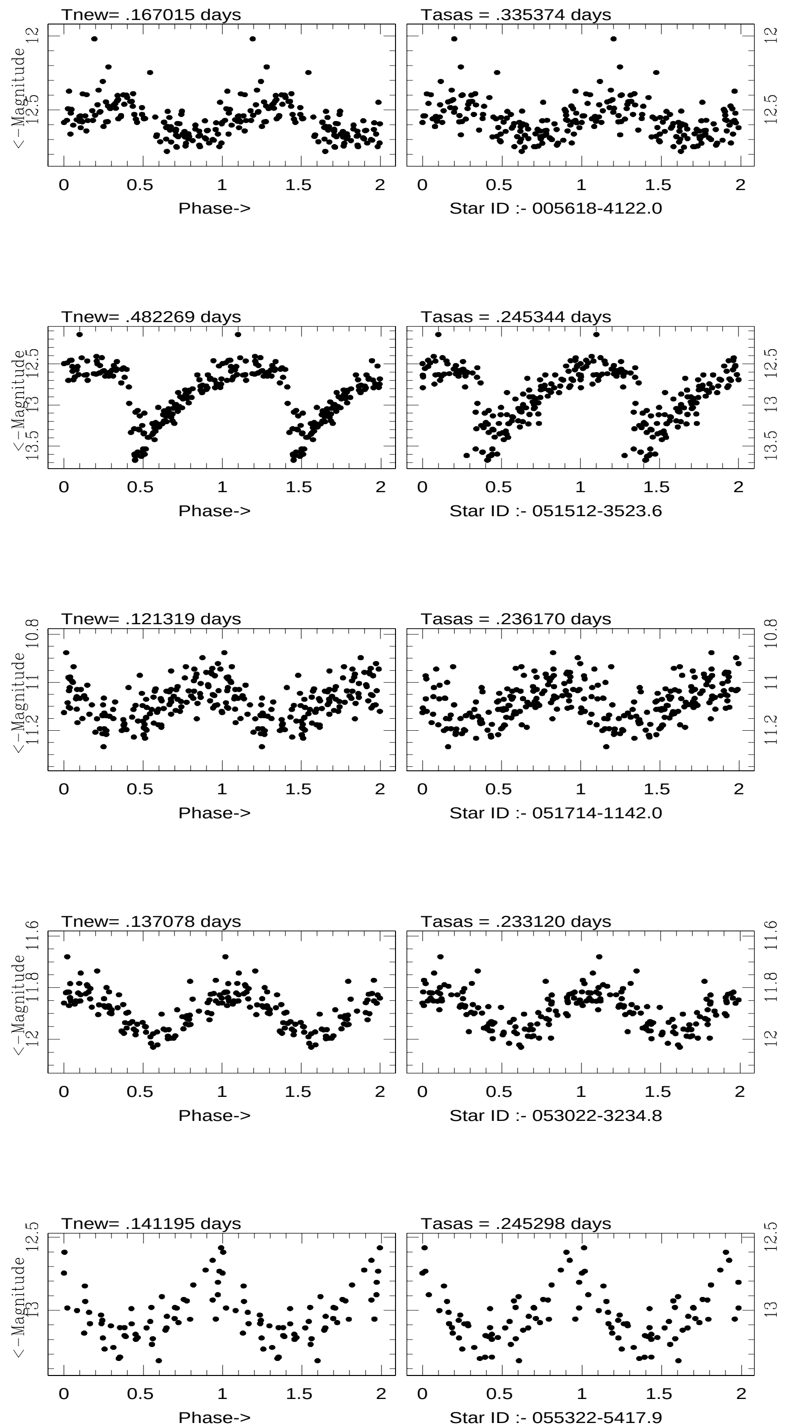

Figure 4.5: Newly detected light curves are on the left and published light curves are on the right. Periods in days are given at the top of each figure. All the above light curves are from the *var3* database. It is found that the newly detected periods are harmonics of the published periods.

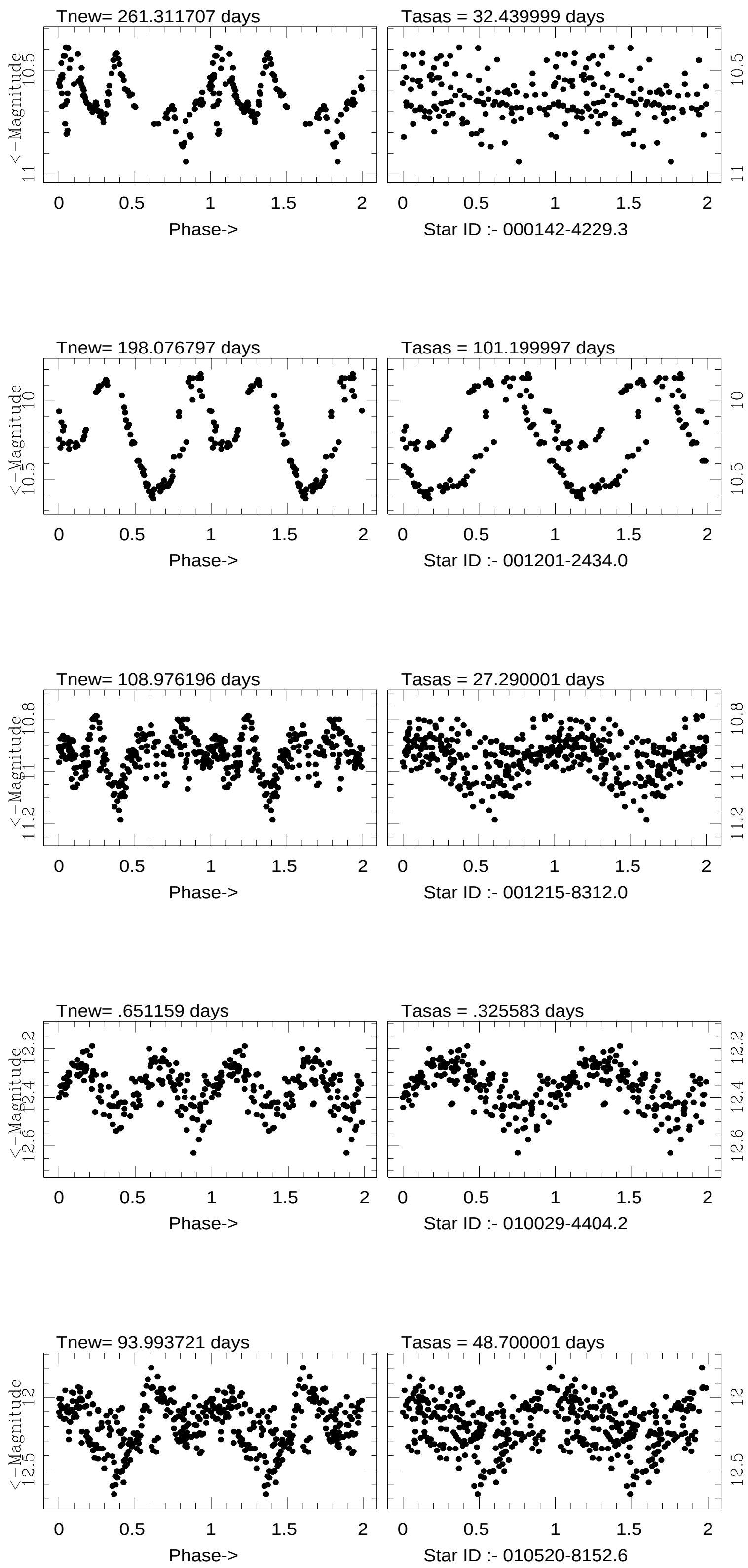

Fig. 4.5 continued...

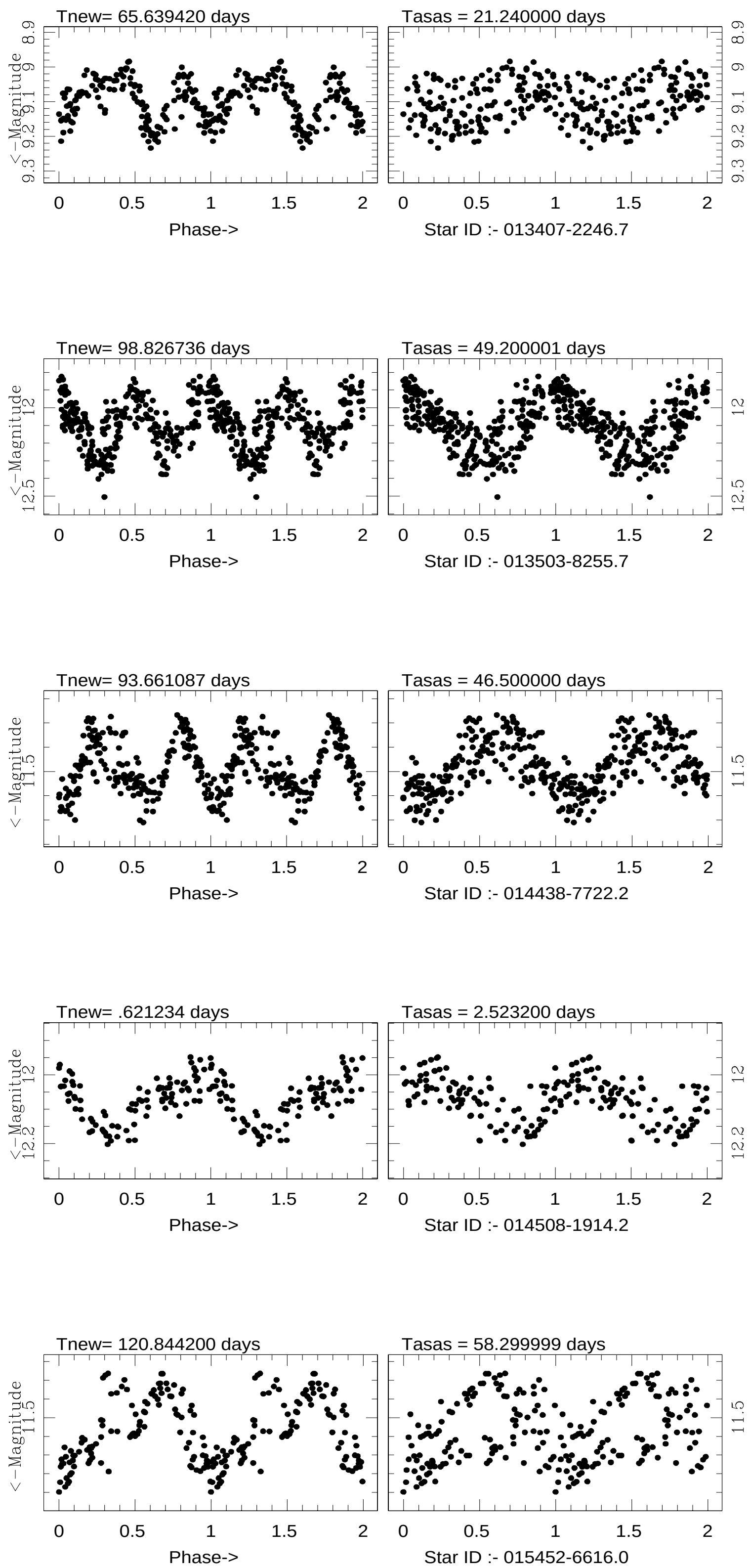

Fig. 4.5 continued...

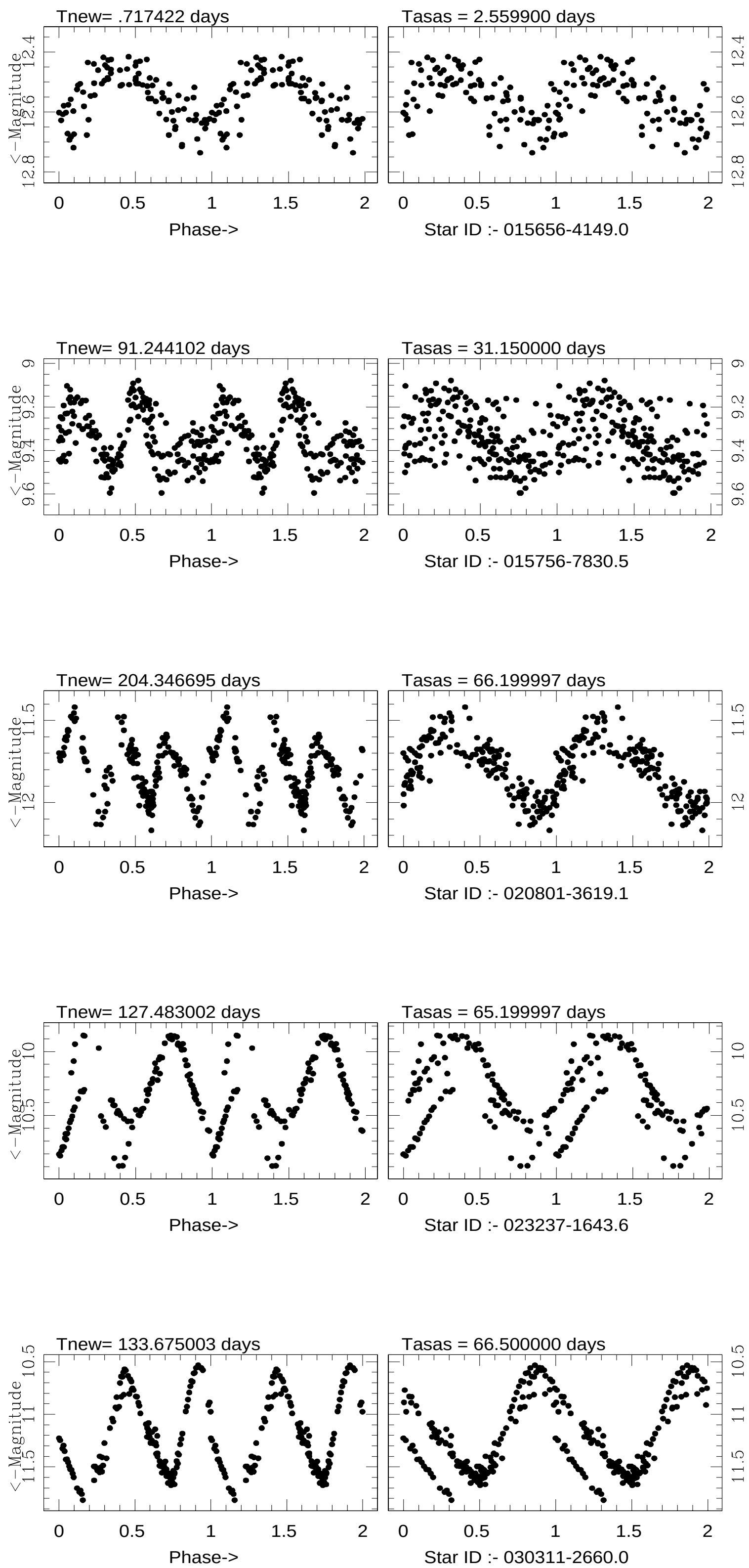

Fig. 4.5 continued...

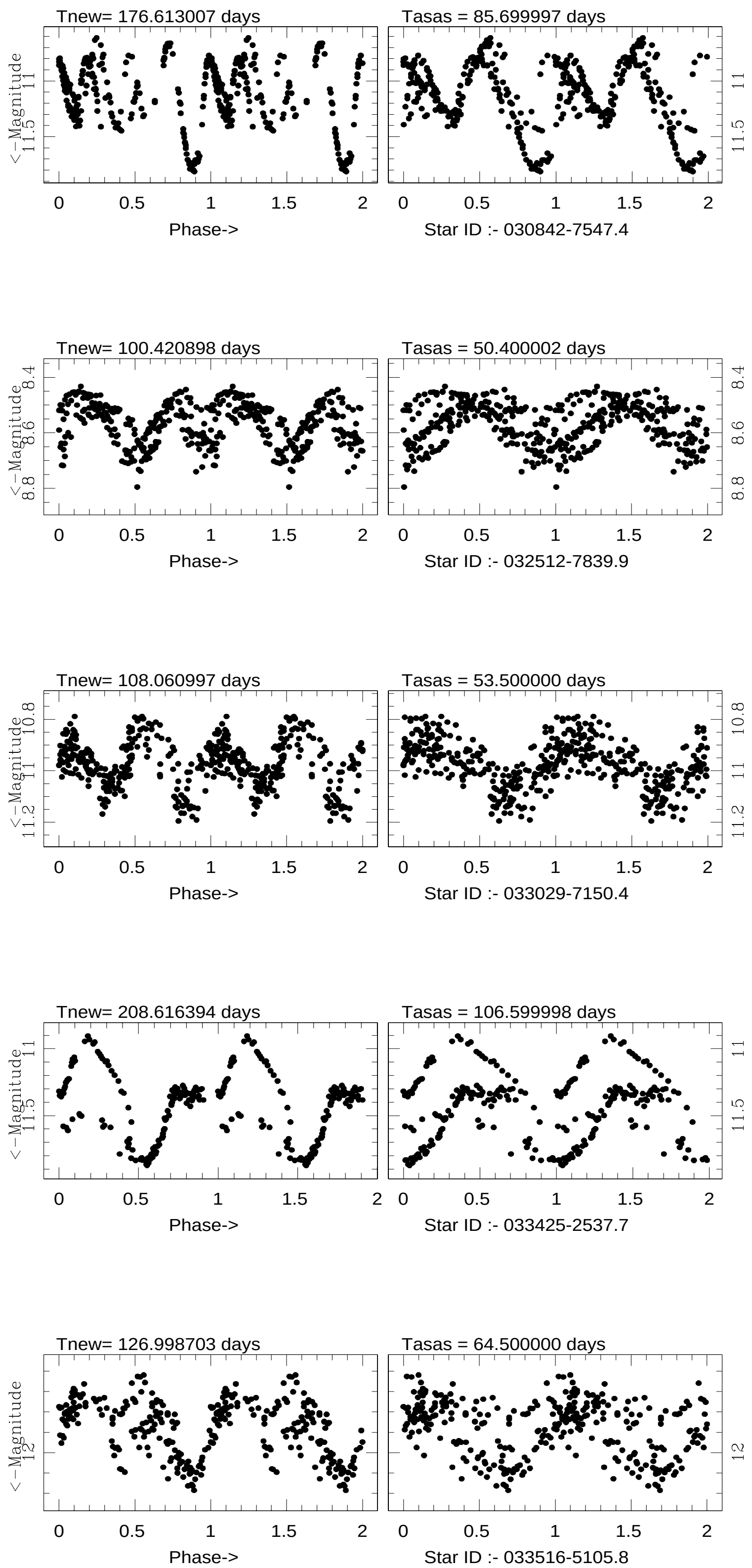

Fig. 4.5 continued...

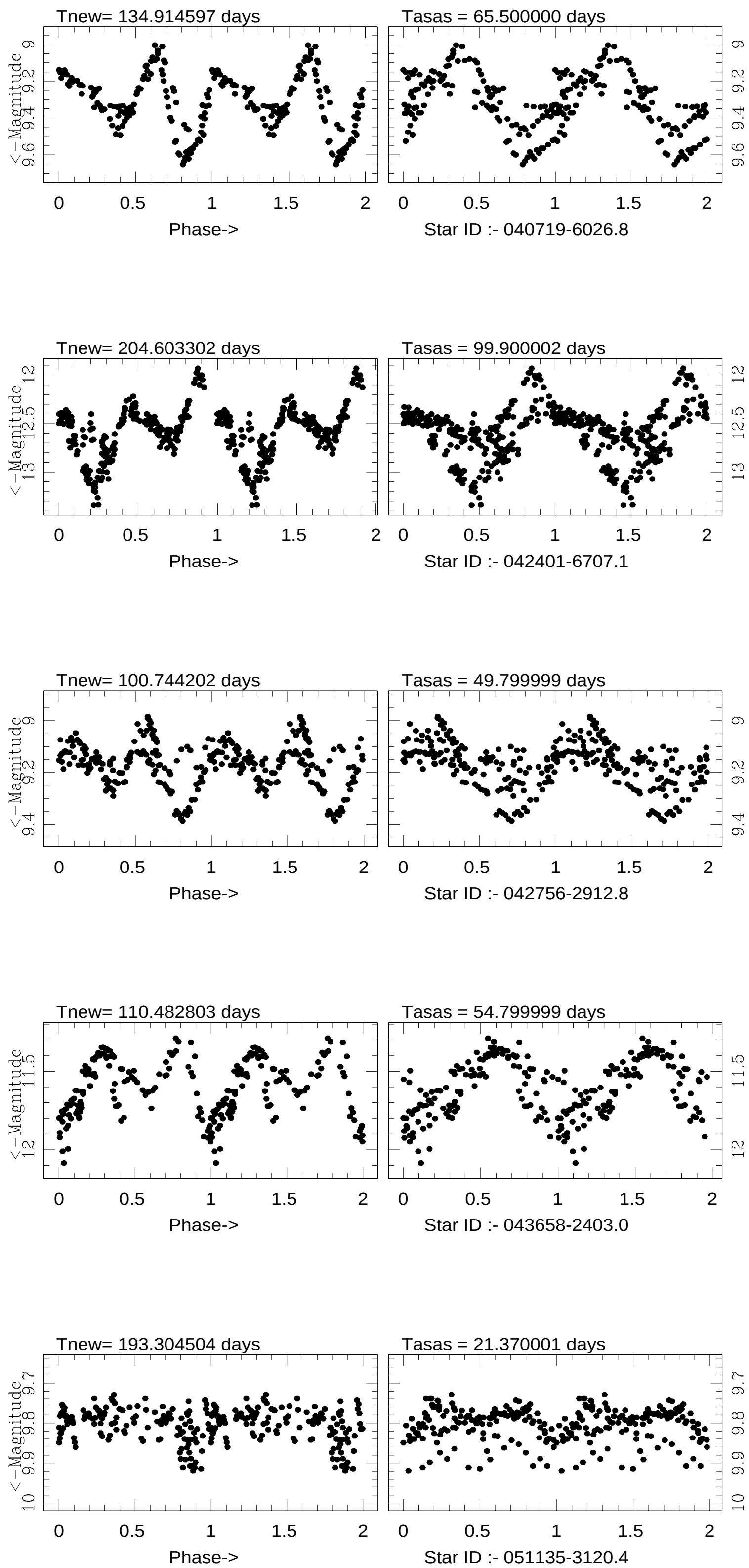

Fig. 4.5 continued...

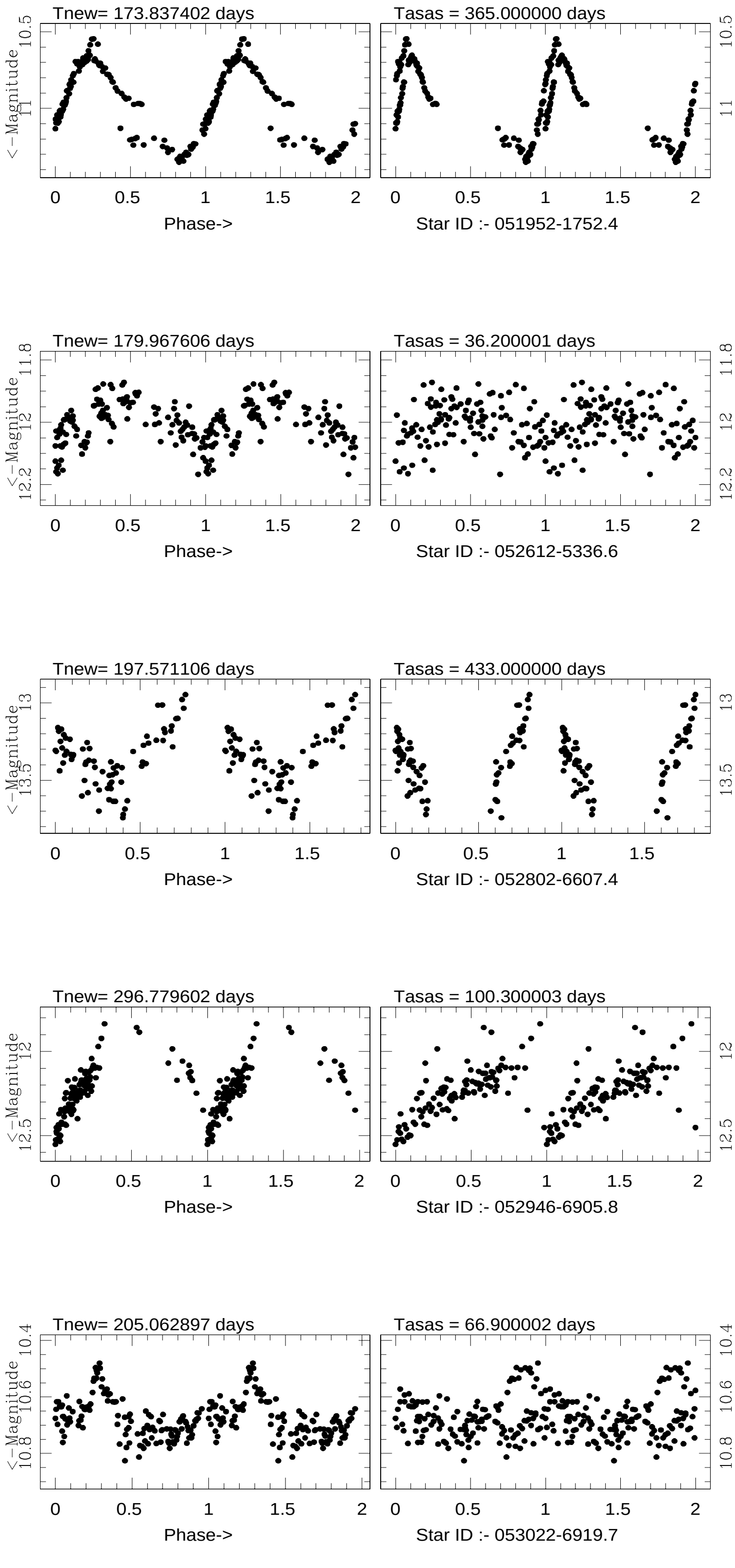

Fig. 4.5 continued...

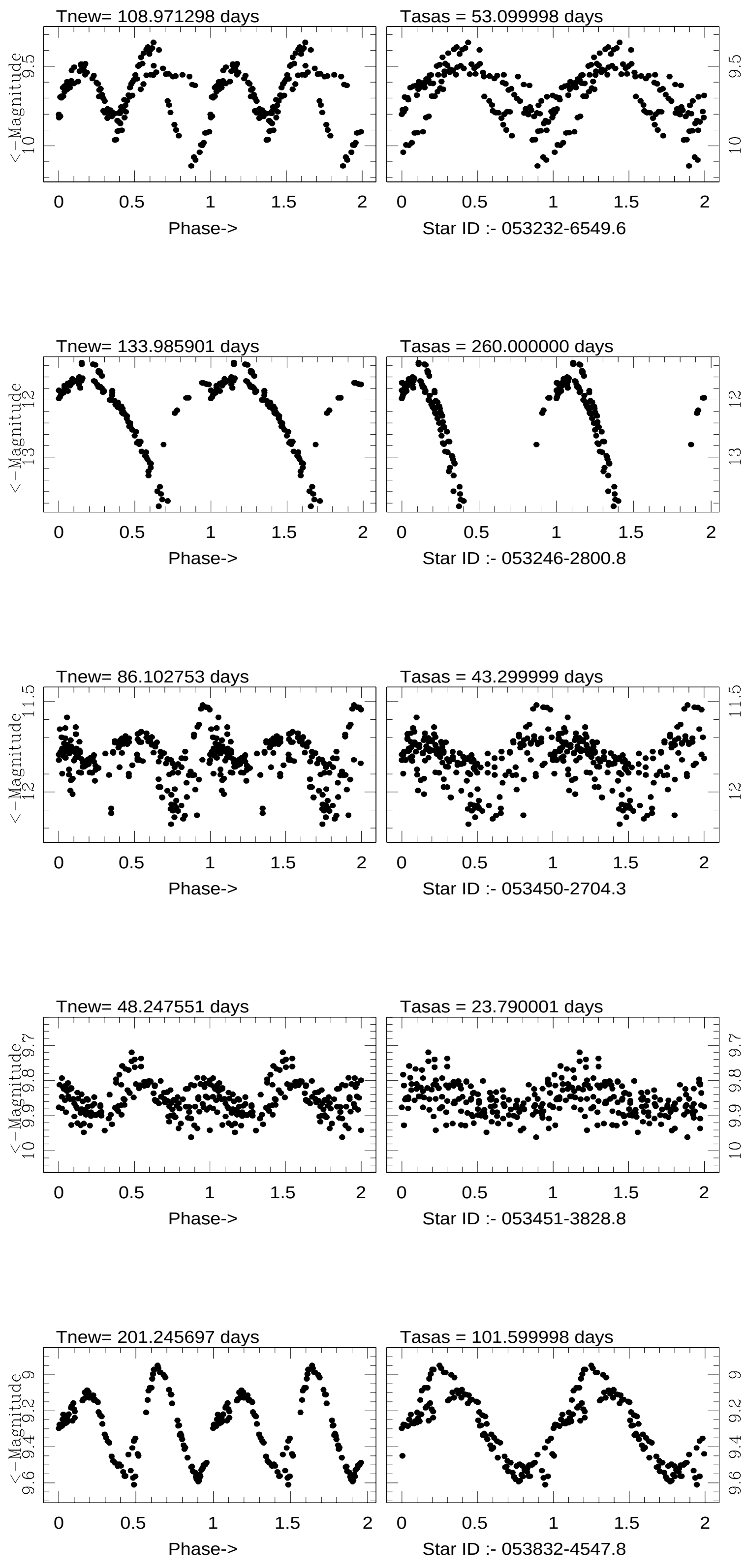

Fig. 4.5 continued...

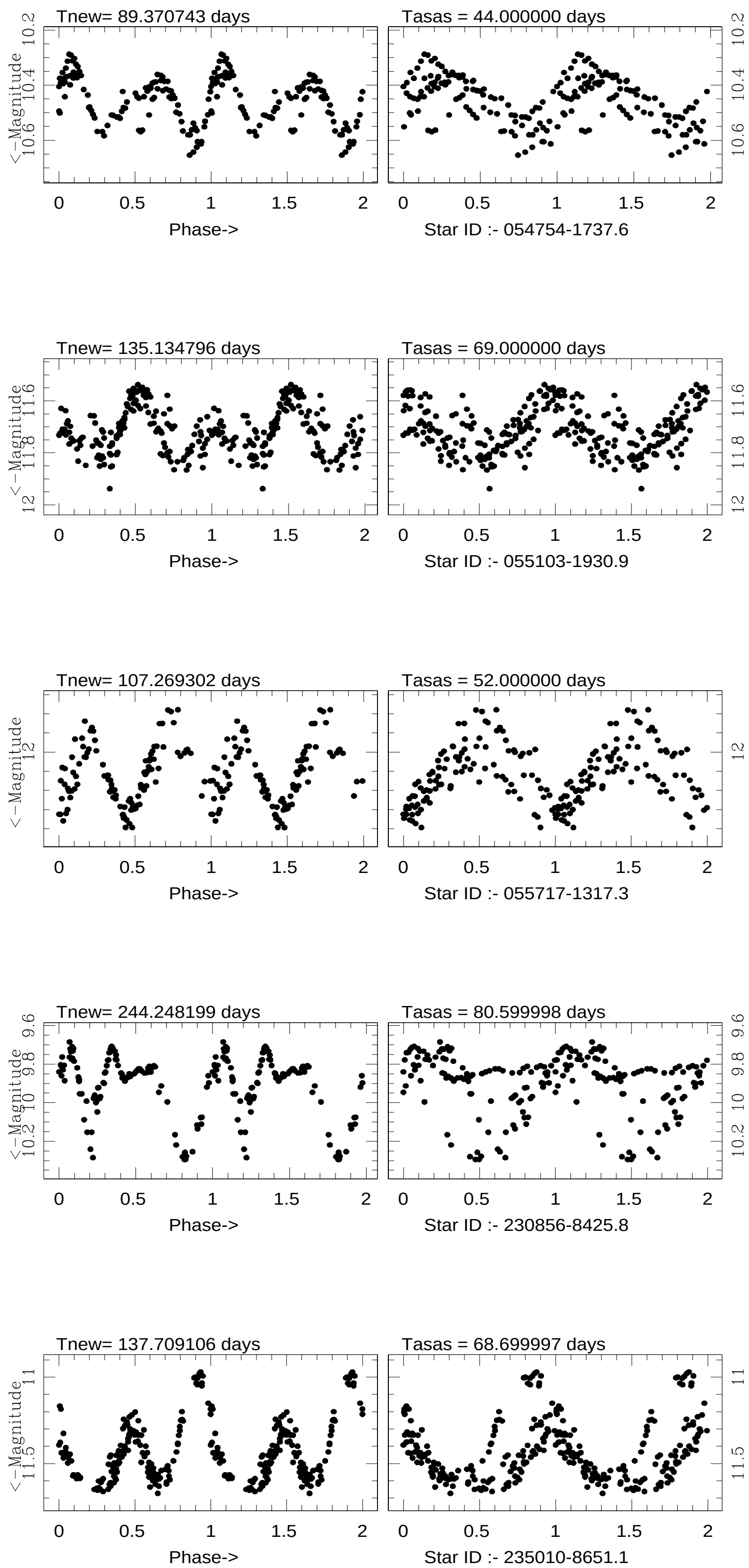

Fig. 4.5 ends here

Chapter 5

Application to CRTS database

5.1 Introduction

The Catalina Real-Time Transient Survey *CRTS* (http://crts.caltech.edu/) is a recent synoptic astronomical exploration that covers thirty three thousand square degrees of the sky in order to discover rare and interesting transient phenomena. *CRTS* that consists of all photometry from seven years of photometry taken with the three telescopes. This data release encompasses the photometry for 500 million objects ($\sim$ 40 billion measurements) with V magnitudes between 11.5 and 21.5 from an area of 33,000 square degrees. The *CRTS* team has made the entire photometric data set public as a service to the astronomical community.

5.2 Application to *CRTS* database

The *CRTS* recently released nearly 12000 RRab star's (RR Lyrae type, which oscillates in fundamental mode, Light curve shows a rapid increase to the peak, with a relatively slow decrease) time series data along with the catalogue. These stars have periods in the range between 0.4 to 0.8 days. We re-analysed the database, with this prior knowledge using SigSpec and got 44 new better periods. Also hundreds of published periods got improved from second and third decimal digital onwards. Out of these, one example for improved period is shown in the Figure (5.1) for star ID 1104076049945 (5th decimal place improvement). By the decimal place improvement of periods, the type of the star does not change in this database and hence will not be much useful for classification purpose, but definitely useful for modeling of light curves and extracting the features of individual stars.

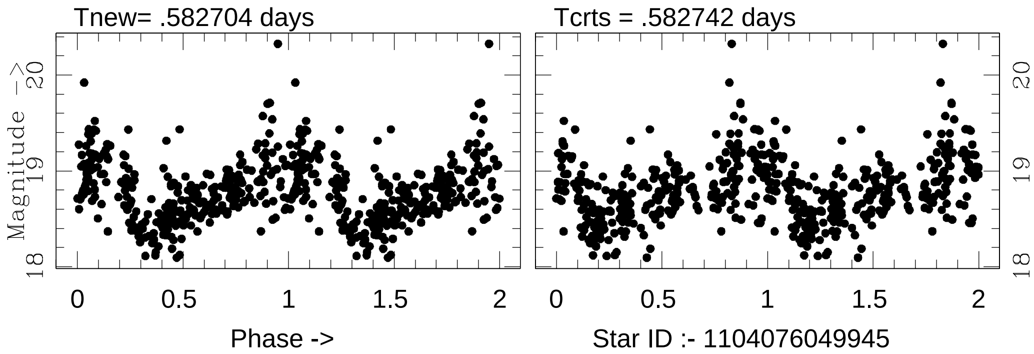

Figure 5.1: Period improvement in the 5th decimal place.

5.3 Results and Conclusion

The Since *CRTS* is a recent survey and use latest software pipeline, we have got small percentage of entirely new periods. The newly obtained period and corresponding folded light curves (left side) are shown below, along with published periods and corresponding light curves (right side) for comparison. *Tnew* is the new period detected and *Tcrts* is the *CRTS* Published period in days. As in the Chapter 4, the light curves are plotted after removing the few extreme outliers, using the *DEBiL* method by Devor[29]. Also we have visually examined the new light curves and compared with the published ones, to make sure that the new results are genuine.

Table 5.1: Newly detected periods and *CRTS* published periods for 44 variable stars. The corresponding phased light curves are shown in the Figure 5.2

Star−ID	$Tnew$	$Tcrts$	Star−ID	$Tnew$	$Tcrts$
1001010056005	0.505981	0.431235	1004073027603	0.505709	0.855231
1004085006979	0.759415	0.531987	1004089050719	0.452869	0.829589
1004090073356	0.490005	0.680319	1004091062574	0.633507	0.546297
1004093132546	0.564659	0.500184	1004116029017	0.407765	0.689753
1007029026267	0.539783	0.635778	1007033074479	0.597016	0.434509
1007047048820	0.554810	0.831158	1007084084733	0.458631	0.849127
1007114034537	0.546930	0.849455	1009032027855	0.621345	0.703380
1009087064076	0.461758	0.859941	1009088108206	0.798656	0.570580
1012055030221	0.466053	0.752115	1012073036637	0.475419	0.908466
1012079068260	0.554563	0.831816	1012089038408	0.662515	0.507995
1015081068674	0.407401	0.664602	1015105046482	0.432534	0.589502
1101083007954	0.576220	0.681087	1109044069434	0.630695	0.465836
1109068056519	0.492714	0.559968	1109093050668	0.641525	0.466515
1112066021993	0.498926	0.762737	1115054045334	0.415073	0.649162
1115067032610	0.420435	0.726873	1118027006983	0.490679	0.551302
1121064046205	0.408653	0.675226	1121085087608	0.641356	0.491003
1123062053146	0.418621	0.720000	1129082102332	0.519875	0.766972
1135002071490	0.599550	0.502907	1135081074235	0.667168	0.544994
1140076060784	0.579143	0.439200	1143037025595	0.768135	0.858729
1152047005361	0.788642	0.440379	1155055016689	0.446929	0.809873
1157022072330	0.592799	0.860527	1157028035996	0.490420	0.444617
1160021002958	0.433543	0.667280	2003191004685	0.596239	0.498063

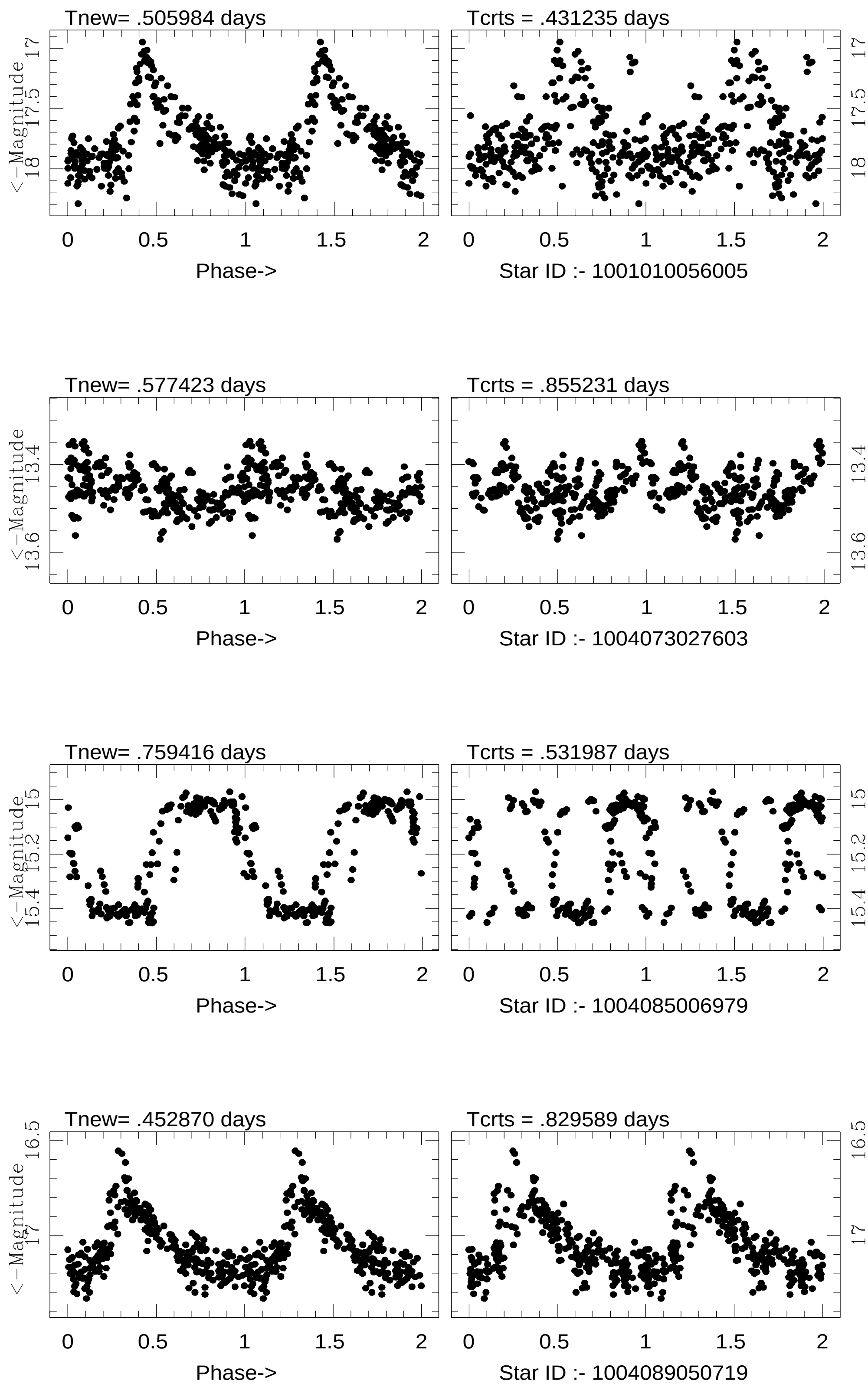

Figure 5.2: The above shown 44 light curves are obtained from the *CRTS* database. The newly detected periods give better light curves than the published periods.

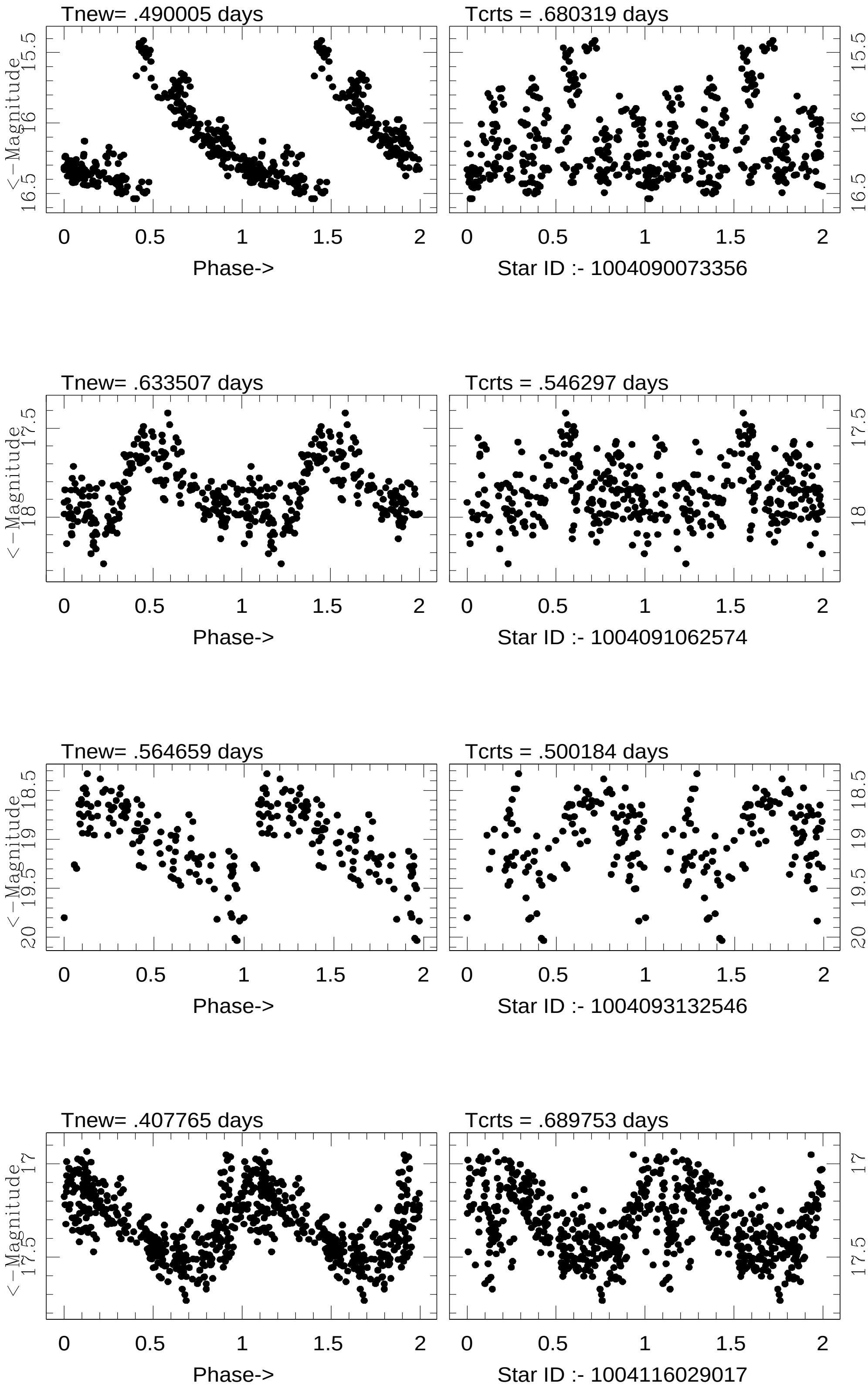

Fig. 5.2 continued...

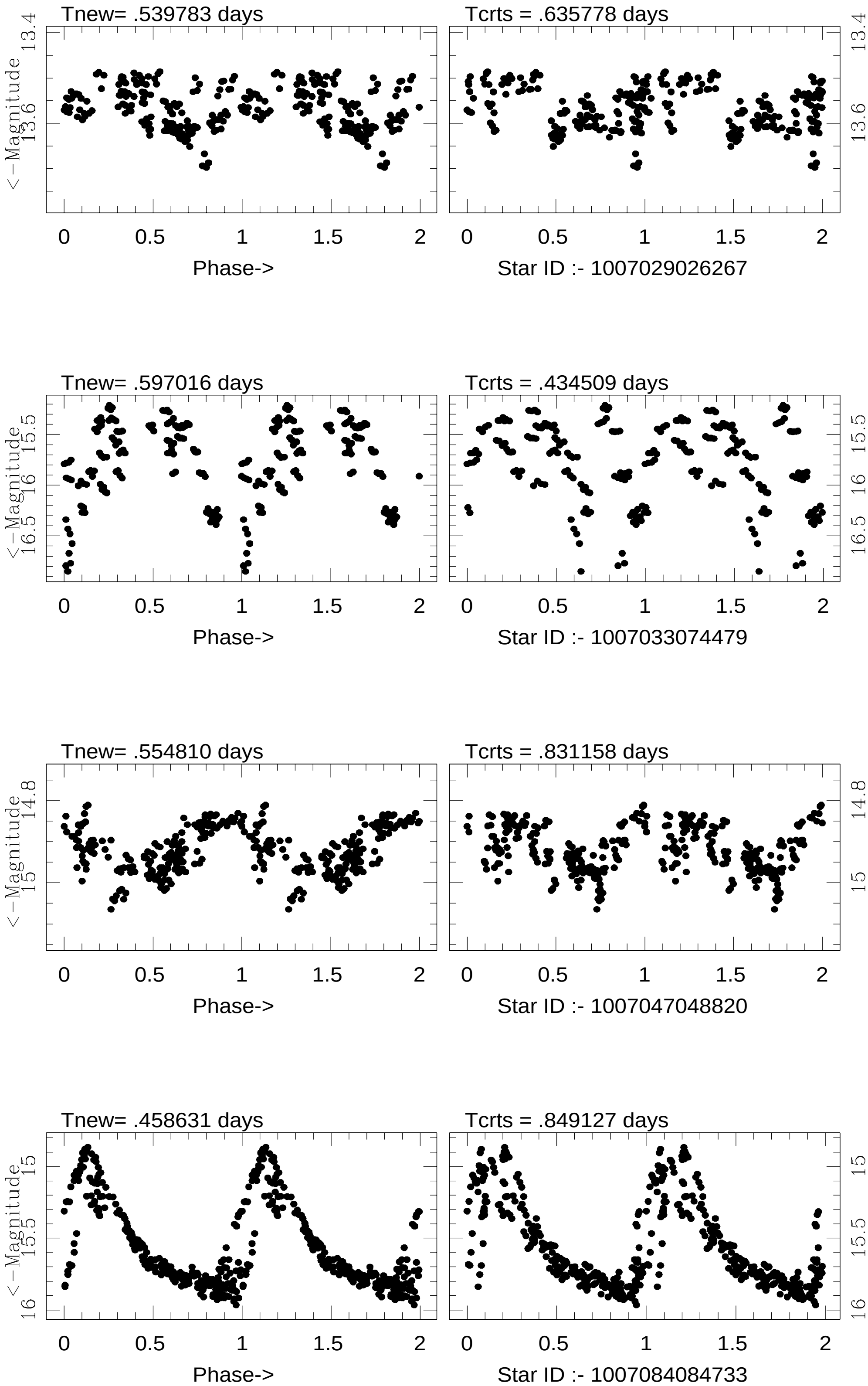

Fig. 5.2 continued...

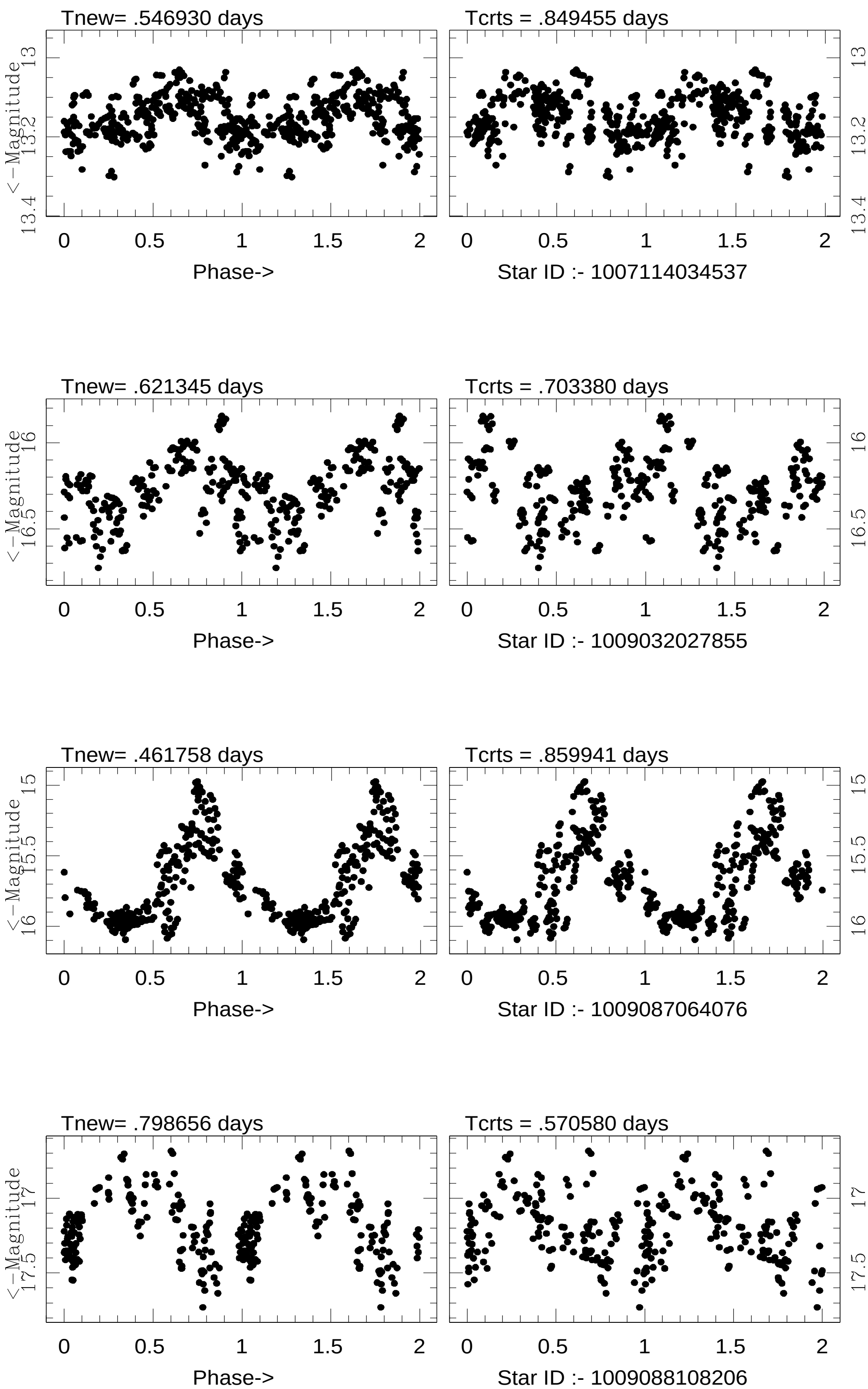

Fig. 5.2 continued...

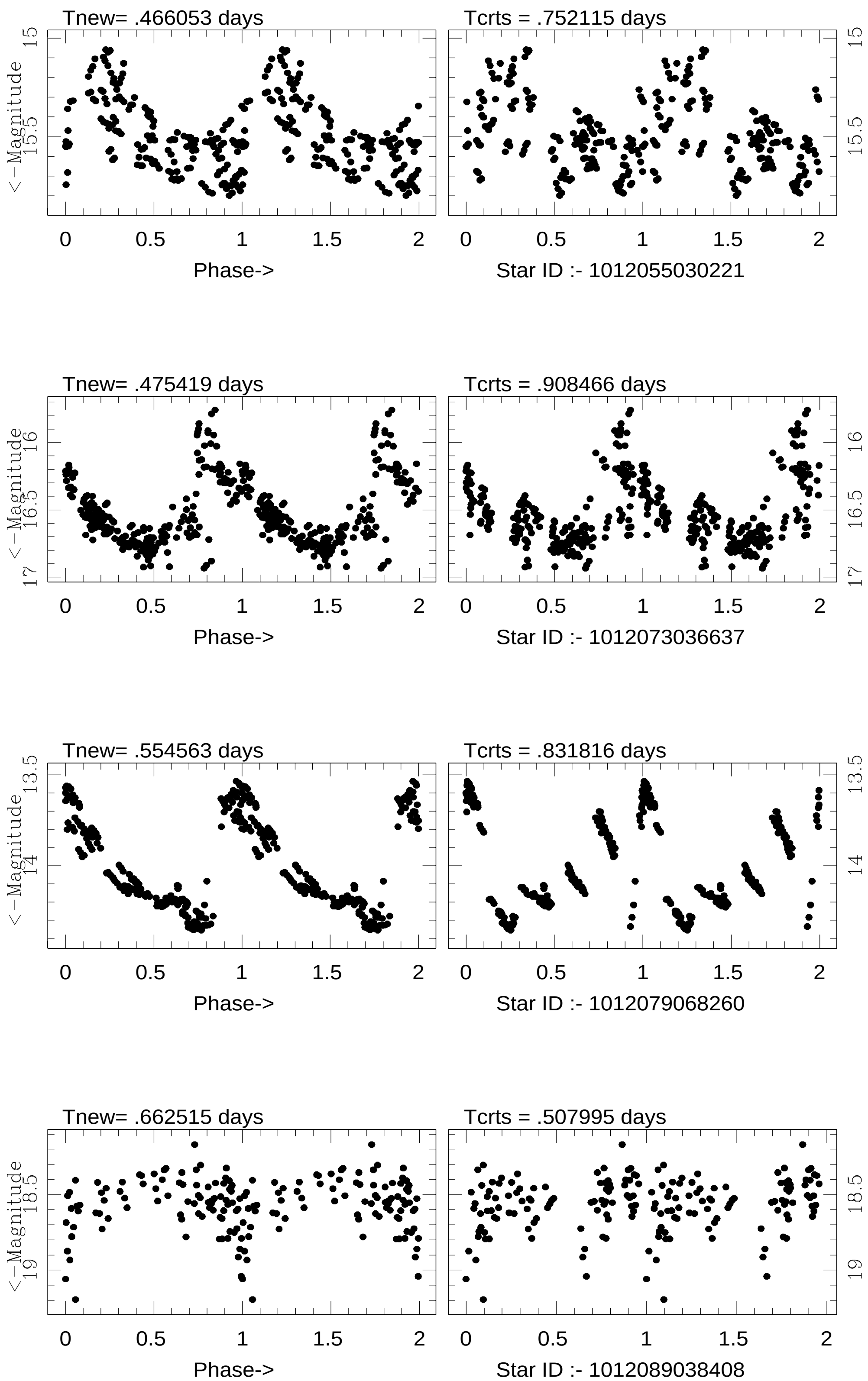

Fig. 5.2 continued...

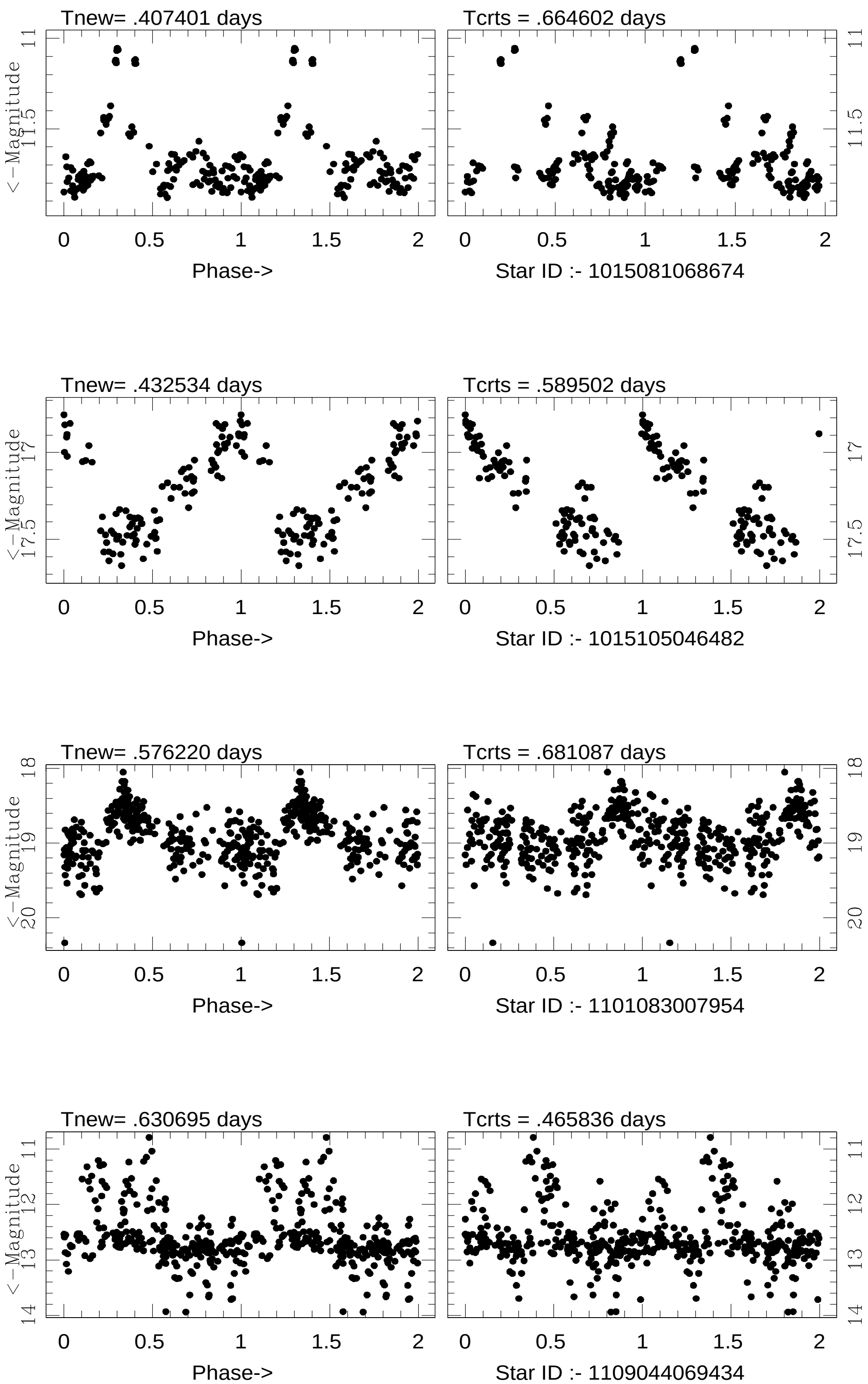

Fig. 5.2 continued...

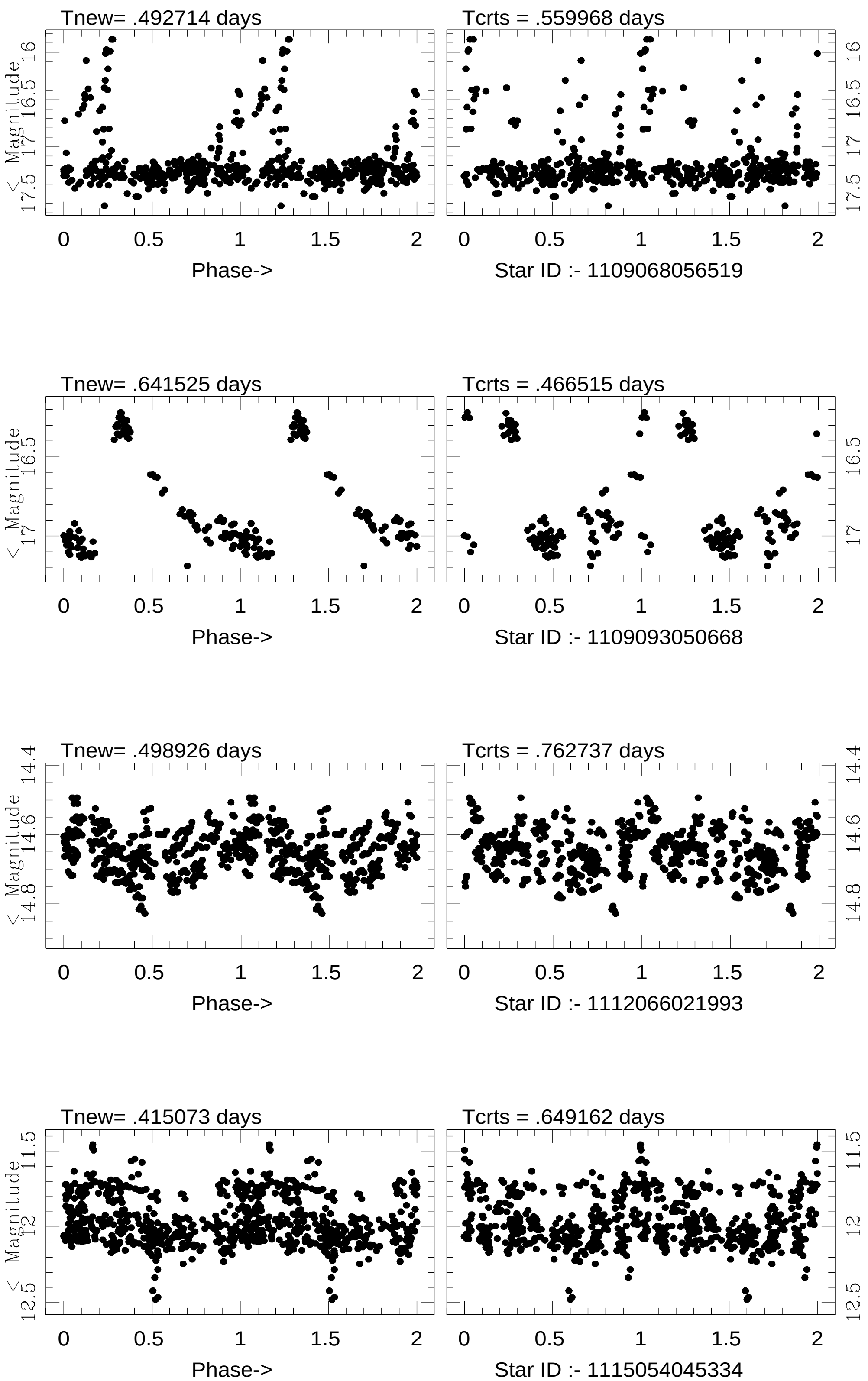

Fig. 5.2 continued...

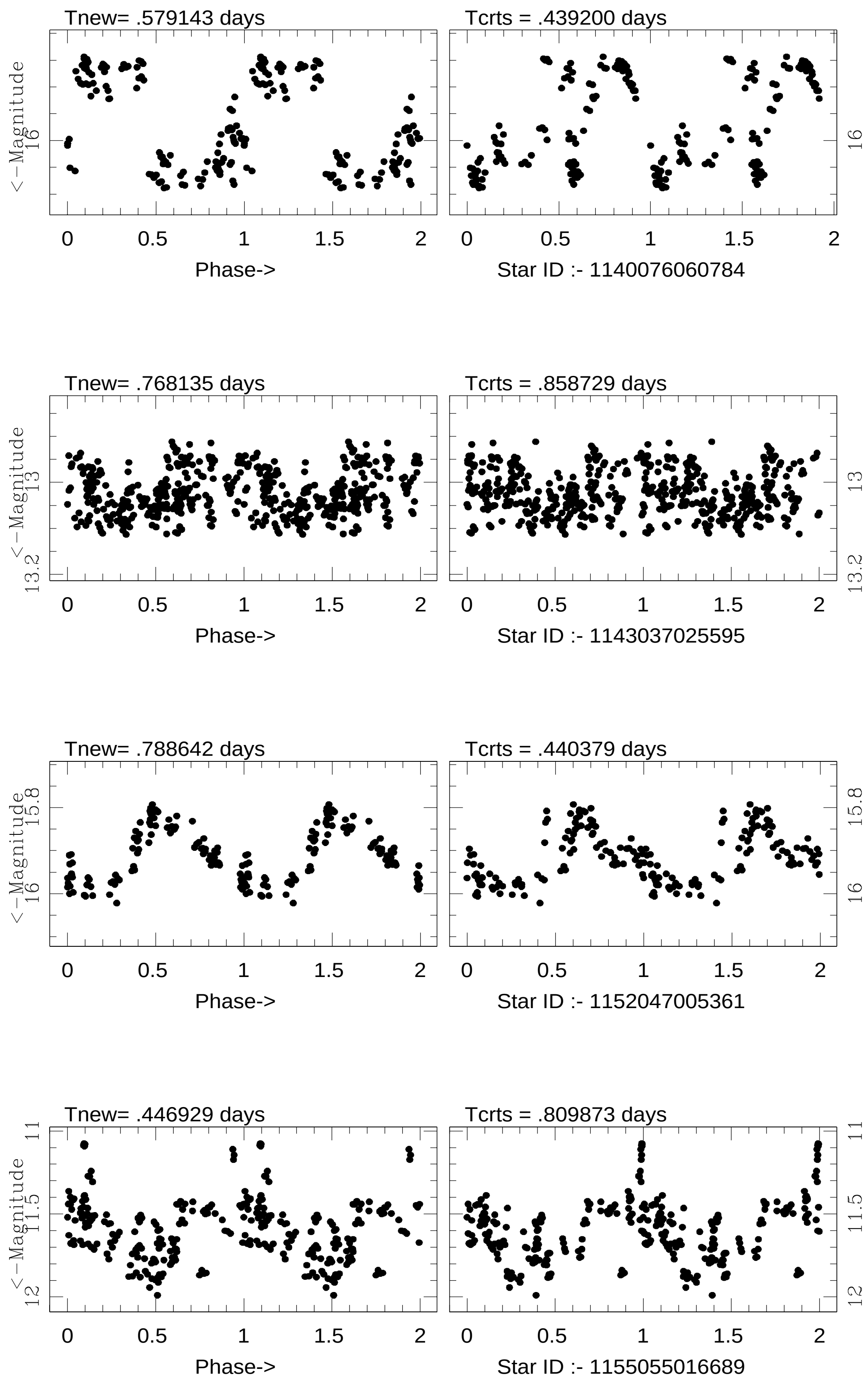

Fig. 5.2 continued...

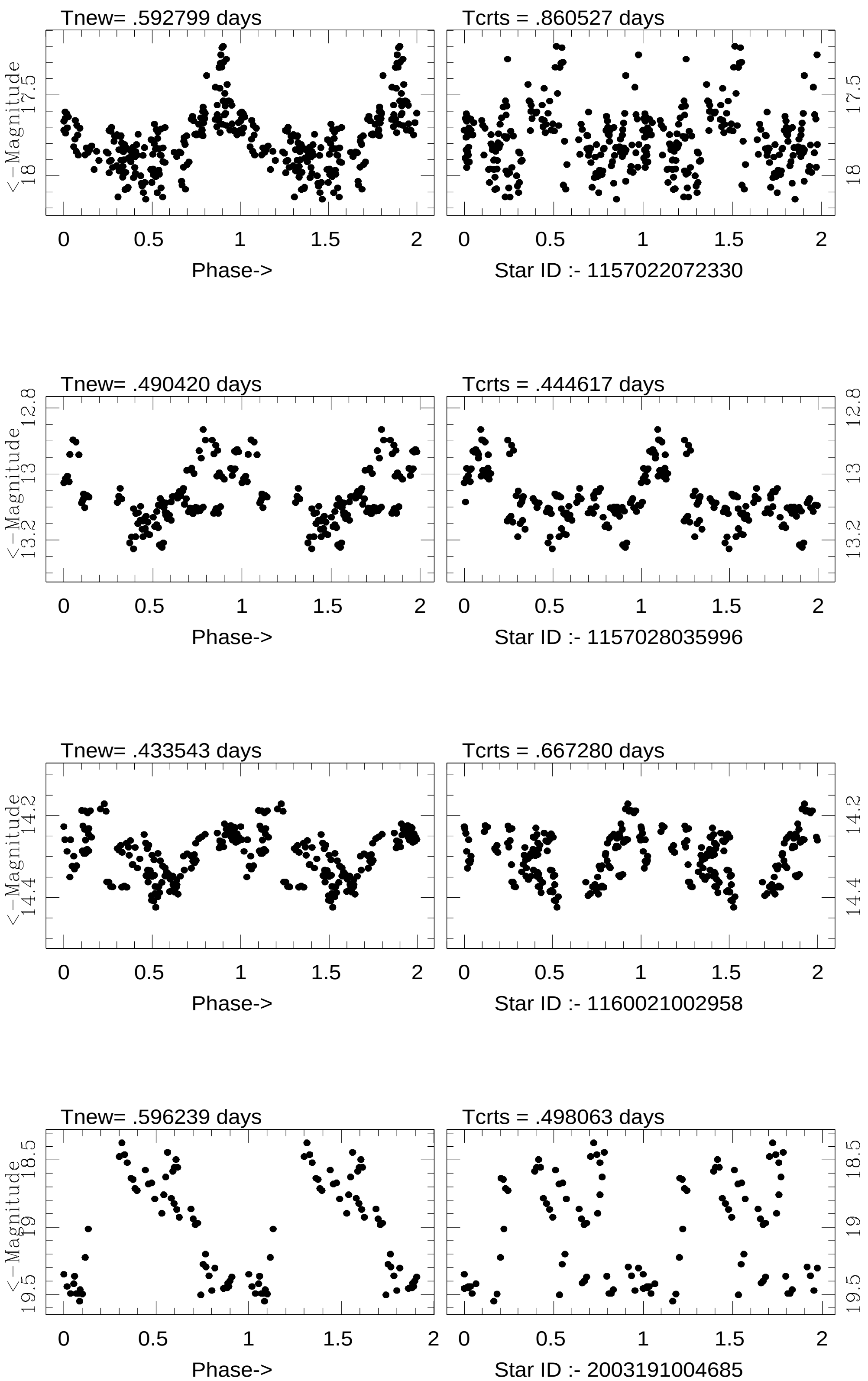

Fig. 5.2 continued...

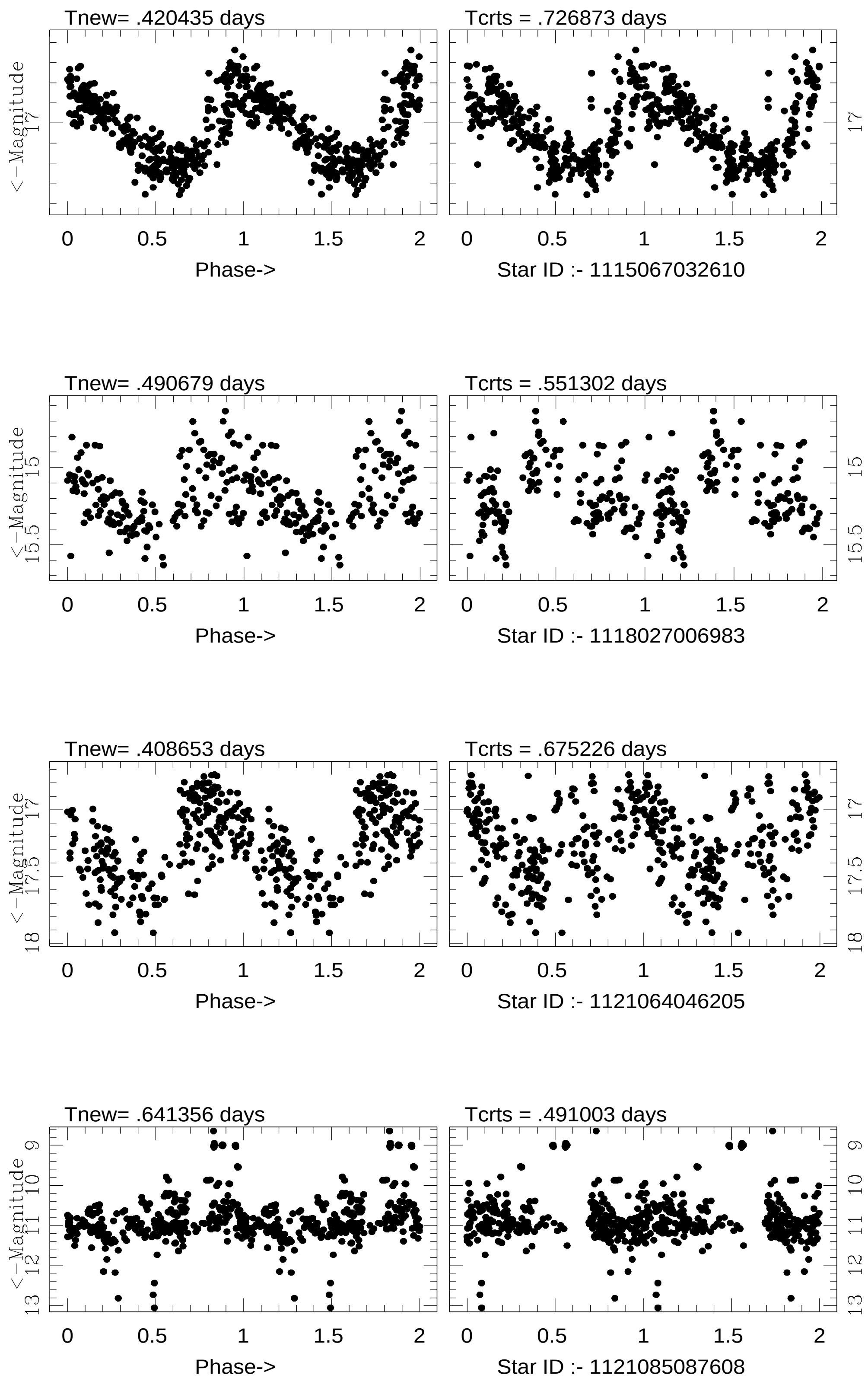

Fig. 5.2 continued...

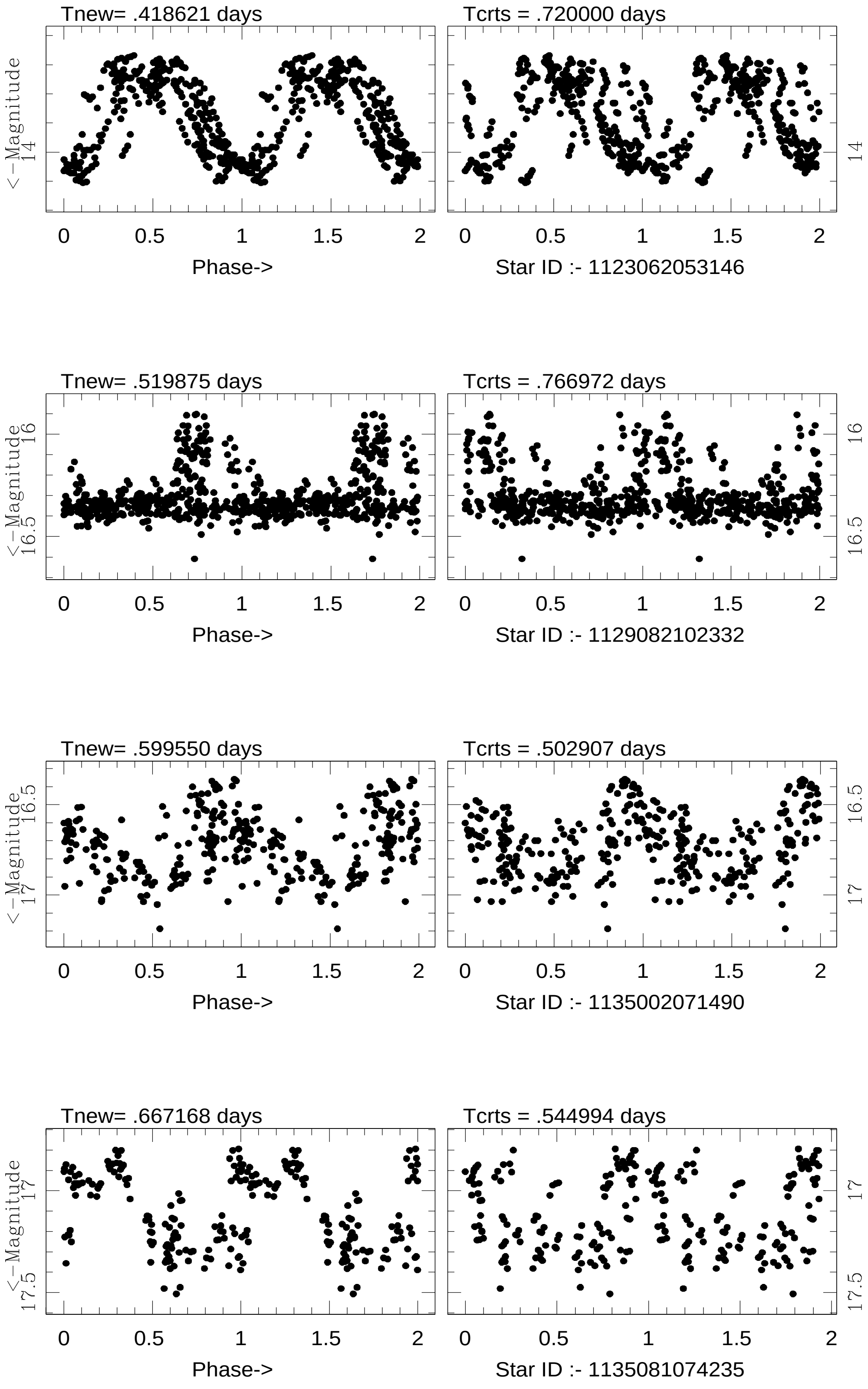

Fig. 5.2 ends here.

Chapter 6

Summary and Conclusions

6.1 Introduction

Many of the existing period search methods were studied in detail and four popular methods were compiled for automated period search of variable stars. The theory of Lomb-Scargle periodogram and cubic spline interpolation using B-splines, were studied in detail. In brief,

- *LSP/GLSP* - The derivation is consolidated in two different ways.
- *SigSpec* - Method is re-coded in FORTRAN.
- *PDM* - Automated the method.
- *MCS* - The theory modified with unevenly spaced knots, coded in FORTRAN.

The developed automated method applied on the following two databases.

- *ASAS* - Re-analysed the database.
- *CRTS* - Re-analysed the database.

Our strategy was to use the above specified first three methods on the data for the initial period search and the results are compared with the published periods. Those results, which are different from the published periods are subjected to the *MCS* interpolation method, including the harmonics of the newly detected period. The period corresponding to the minimum least square error is taken as the newly detected best period. In order to confirm this, we have visually examined all these light curves. We have applied this strategy to two databases *ASAS* and *CRTS*, which are available to the public.

It has been found that, in the case of *ASAS* database, we have detected nearly 5% better periods than the published periods. In the case of *CRTS* database, we have detected 44 entirely different periods and many decimal place improved periods. This is due to the fact that, *CRTS* uses latest classification method and also only one type of variable star data (i.e. RRab type) is used. The results obtained to us show that, the period search methods are to be improved to get 100% exact periods of variable stars, when the data is subjected to automation. This is the need of time, since large scale variability surveys are producing lot of time series data and many of these data is available to the public. The computers are getting more powerful, storage media is becoming cheaper and long lasting, Internet becoming faster, the data analysis on astro-time-series becomes a prospectus research area in the near future.

The knowledge wise required skills for this kind of data analysis research are as follows. Some domain knowledge about the variability, variable stars, astro-time-series, light curves are required. The statistical knowledge about various statistics used to assess the quality of the data as well as the obtained results are required. The numerical methods and programming in few languages are also needed. The bash or any other scripting language should be known, which will help the automation of methods, by which lot of time can be saved. Finally the interpretation of the results require some experience with the shape and other properties of the characteristic light curve of variable stars.

6.2 Summary : Comparison of 3 methods and improvement by MCS interpolation method

It has been found that the automated methods are required, for analysing thousands of light curves. Also our analysis on two databases shows that some of the published periods are not correct. This is because, usually one method is used for period confirmation. But we have used two methods in sequence, one for detection and another for period confirmation. This kind of implementation takes little more time than a single method, but some good results obtained shows that, such strategies can be used for re-analysing the published results. The table below shows the results of the 3 popular period search methods applied on *ASAS a2perlc* data, which contain 384 time series data files. The table (6.2) shows that, the modified cubic spline (*MCS*) interpolation method improved the period detection rate nearly by $5-6\%$ during automation.

Method	Success rate	+*MCS*
GLSP	81%	88%
SigSpec	85%	91%
PDM	79%	84%

Table 6.1: The sequential application of MCS improves the detection by $5-6\%$

6.3 Future prospectus

In view of above results, we can point some interesting problems that can be addressed in the future.

- Improve the method, so that full automation along with maximum success rate is obtained.
- Apply the automated methods of period detection to other available databases.
- Extract other features from the light curve.
- Extension into other time series for short term and long term periodicity predictions.
- Error analysis on the obtained periods.
- Classify the variable stars using supervised/unsupervised classification methods, Gaussian mixture model and Random forest automated classification methods, using minimum parameters.

Appendix A

Significant Spectrum - SigSpec

A.1 *SigSpec-F*

SigSpec program for finding periods of variable stars, which can be automated in combination with the script in Appendix C.1

```
! CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!  _ _ _ _ _            _ _ _ _ _
! / / / / /   SigSpec-F  / / / / /
!
! CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!
!SigSpec-F is for Period analysis of variable stars from their time series
!Expects a time series text file with 3 columns for
!time, magnitude and error as input and each separated by space or 2 columns
!Time in Julian Date.
!The extension of file is .lc. If extension is different, change it in the
!script startsigspecf.sh
!This program is written for full automation, hence nothing has to be given as input
!The script does everything. Copy and name this program as sigspecf.for
!Just keep all data files (*.lc) in a folder along with this program and script
!startsigspecf.sh
!and compile this fortran program from the same folder terminal. Use latest compiler
!gfortran -o sigspecf sigspecf.for . Uses dynamical memory allocation by fortran
!Then run the script as  ./startsigspecf.sh
!The entire files in the folder are processed one by one.
!The output is written to 'result.txt' as 7 column text file
!Output columns are freq, sig, amp, phase, rms dev, p2p scatt, csig
!Original version is written by Piet Reegen, Vienna, Austria in C Language
!SigSpec-Fortran version is written by Shaju K.Y. and Ramesh Babu Thayyullathil
!Cochin University of Science and Technology, Kochi, Kerala, India 682022
!January 2009
!For any queries or bugs, please contact shajuky@gmail.com
!Good Luck in astro computing !!!
```

```
!
        PROGRAM SIGSPECF
        IMPLICIT NONE
!Program starts here. The total number of observations read here
!The number of lines in input file is read by script startsigspecf.sh  and
!produce file 'forin.dat'
        INTEGER line
        OPEN(unit = 17,file = 'forin.dat',status = 'unknown')
        READ(17,*),line
        CALL firstfn(line)
        CLOSE(17)
        STOP
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE firstfn(N)
        IMPLICIT NONE
!       Here data input files are read
!       Initial parameters are calculated by calling Initial_Par
!       Then calls SigSpec_Cascade to proceed
!       Below number of iterations are to be given corresponding
!       to the number of frequencies to be determined as iter
!       file read subroutine is taken from GLS by Zechmeister
        INTEGER N,i,ri,nspr,iter,IO_ERR/0/
        CHARACTER name*60,line*1024
        DOUBLE PRECISION y(N),t(N),err(N),tspan
        DOUBLE PRECISION lf,uf,fs,tstart,tmin,tmax
        DOUBLE PRECISION freq,amp,theta,PI
        nspr = 0
        lf = 0.d0
        uf = 0.d0
        fs = 0.d0
        tspan = 0.d0
        tmin = 0.d0
        tmax = 0.d0
        tstart = 0.d0
!@@@@@@@@@@@@@@@@
        iter = 10 ! here give number of periods to be detected, after pre-whitening
!@@@@@@@@@@@@@@@@
   69   OPEN (unit = 18,  file = "input.txt",   status = 'old' )
        WRITE(*,*) 'The number of lines is ',N
        DO i = 1,N
   74   READ (18,'(a)',END = 73) line             ! reading line from datafile
        IF (index(line,'#').GT.0) GOTO 74          ! skip line if comment line found
        IF (lnblnk(line).EQ.0) GOTO 74             ! skip if blank line found
```

```
        READ(line,*,IOSTAT = IO_ERR) t(i),y(i),err(i) ! here actual READing
        IF (IO_ERR.NE.0 ) THEN
        READ(line,*,IOSTAT = IO_ERR) t(i),y(i) ! Reading if there are only 2 columns
        IF (IO_ERR.NE.0 ) THEN
        WRITE(6,*) "Something is wrong with input data file !!!"
        WRITE(6,*) " Please check the format of file"
        IF (err(i).EQ. 0.) IO_ERR = -2
        CLOSE(18)
        GOTO 67   !  unexpected termination of SigSpec-F, due to file problem
        ENDIF
        ENDIF
        ENDDO
 73     CLOSE (18)
        tmin = t(1)
        tmax = t(N)
        tspan = tmax-tmin
        print *, "Time Span ",tspan, "days"
        CALL Initial_Par(t,N,tspan,lf,uf,fs,nspr)
        WRITE(*,*)'Period search between ',lf, 'and',uf
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        CALL SigSpec_Cascade(t,y,N,lf,uf,nspr,fs,iter)
        WRITE(*,*)'Finished SigSpec-F Successfully'
67      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE Initial_Par(t,N,tspan,lf,uf,fs,nspr)
        IMPLICIT NONE
        INTEGER N,fi,err,h0,nspr
        DOUBLE PRECISION ufbk/-1.d0/,fres,tspan,os,t(N)
        DOUBLE PRECISION lf,uf,fs,nyfreq,nycoef
        nyfreq = 0.d0
        nycoef = 0.9d0 ! increase nycoef here for higher uf
        os = 20.d0
        fres = 1.d0/tspan
        lf = 10.d0*fres
!        IF (nycoef .LT. 0.5d0) nycoef = 0.5d0
        CALL Nyquist_Freq(t,N,tspan,nycoef,nyfreq)
        ufbk = nyfreq
       WRITE(*,*)'upper frequency limit as sigspec', ufbk
        uf = ufbk
        fs = fres / os
        h0 = INT(FLOOR((lf / fs)))
       WRITE(*,*)'h0=',h0
        lf = h0 * fs
        nspr = INT(CEILING((uf - lf)/fs + .5d0)) ! instead of ceil
```

```
        uf = lf + (nspr - 1.d0) * fs
!       nspr=nspr/20 ! For faster calculation, but less accuracy
        WRITE(*,*)'*******************************'
        WRITE(*,*)'Rayleigh frequency resolution', fres
        WRITE(*,*)'oversampling ratio', os
        WRITE(*,*)'frequency spacing', fs
        WRITE(*,*)'lower frequency limit', lf
        WRITE(*,*)'upper frequency limit', uf
        WRITE(*,*)'Nyquist coefficient', nycoef
        WRITE(*,*)'number of frequencies', nspr
        WRITE(*,*)'*******************************'
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE SigSpec_Cascade(t,y,N,lf,uf,nspr,fs,it)
        IMPLICIT NONE
!       This is the main subroutine which calls all other subroutines
        INTEGER N,i,nspr,it,im,ik,h
        DOUBLE PRECISION y(N),t(N),rsd(it),rppsc(it),period(it),fsd(it)
        DOUBLE PRECISION lf,uf,fs,sd,ppsc,sig,freq,amp,theta,torig(N)
        DOUBLE PRECISION proft0(nspr),profa0(nspr),profb0(nspr),a(3),th
        DOUBLE PRECISION yorig(N),origsd,csig(it),shift,oshift
        DOUBLE PRECISION rfreq(it),rsig(it),ramp(it),rtheta(it),ofreq
        DOUBLE PRECISION nfreq,namp,ntheta
        OPEN(unit = 19,file = 'result.txt',status = 'unknown')
!*******************************************
        sd = 0.d0
        ppsc = 0.d0
        origsd = 0.d0
        sig = 0.d0
        freq = 0.d0
        amp = 0.d0
        theta = 0.d0
        ofreq = 0.d0
        nfreq = 0.d0
        namp = 0.d0
        ntheta = 0.d0
        DO im = 1,it
         rsig(im) = 0.d0
         rfreq(im) = 0.d0
         ramp(im) = 0.d0
         rtheta(im) = 0.d0
         csig(im) = 0.d0
         rsd(im) = 0.d0
         fsd(im) = 0.d0
```

```
         rppsc(im) = 0.d0
        ENDDO
!******************************************
        CALL Stat_ZeroMean(y,N)
        CALL stat_ppscatter(y,N,ppsc)
        CALL Stat_SD(y,N,origsd)
        CALL t_Mean(t,N,shift)
        DO i = 1,N
        yorig(i) = y(i)    ! backing original time series
        torig(i) = t(i)
        ENDDO
!//////////////////////////////////////////////////////////////////////
        DO im = 1,it
        CALL Stat_ZeroMean(y,N)
        CALL stat_ppscatter(y,N,ppsc)
        rppsc(im) = ppsc
        CALL Stat_SD(y,N,sd)
        rsd(im) = sd
!******************************************
!Profile_SigSpec needs to be calculated once, serendipity
        IF (im .EQ. 1) then
        CALL Profile_SigSpec(t,y,N,lf,uf,nspr,fs,profa0,profb0,proft0)
        ENDIF
!******************************************
        CALL SigSpec_SigSpec(t,y,N,lf,nspr,fs,sd,freq,profa0,
     *  profb0,proft0,sig,im)
!        WRITE(*,*)'freq sig',freq,sig
        CALL SigSpec_MaxSig(t,y,N,freq,sig,amp,theta,fs,sd)
        WRITE(*,*)'###freq amp phase before rmsdev',freq,amp,theta
        rsig(im) = sig
        rfreq(im) = freq
        ramp(im) = amp
        rtheta(im) = theta
        oshift = shift
        WRITE(*,*)'###freq amp phase before MSN',freq,amp,theta
        CALL MultiSine_Newton(t,y,N,ramp,rfreq,rtheta,rsig,yorig,
     *  origsd,im,shift)
        shift = oshift
        WRITE(19,900)rfreq(im),rsig(im),ramp(im),rtheta(im),
     *  rsd(im),rppsc(im),csig(im)
!******************************************
        WRITE(*,*)'********************************************'
        WRITE(*,*),im,'freq,sig,amp,ph,rms,nrms,ppsc csig'
        WRITE(*,900)rfreq(im),rsig(im),ramp(im),rtheta(im),
     *  rsd(im),rppsc(im),csig(im)
        WRITE(*,*)'********************************************'
```

```
 900    FORMAT(F24.18,1X,F16.8,1X,F24.18,1X,F24.18,1X,F24.18,1X,
     *  F24.18,1X,F16.8)
        IF (im .EQ. 1) then
        csig(im) = rsig(im)
        ELSE
        CALL Sigspec_Csig(rsig,csig,im)
        ENDIF
        ENDDO
        CLOSE(19)
!*******************************************
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE amplitude(t,y,N,freq,theta,amp)
        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION theta,xcos,xsin,cos2t,sin2t,sint2,cost2
        DOUBLE PRECISION y(N),t(N),amp,ampli,freq,om,PI
          xcos = 0.d0
          xsin = 0.d0
         cos2t = 0.d0
         sin2t = 0.d0
         sint2 = 0.d0
         cost2 = 0.d0
         PI = 4.d0 * DATAN(1.d0)
         om = 2.d0 * PI * freq
!       Finding amplitude
         DO i = 1,N
         xcos = xcos + y(i) * DCOS(om * t(i)-theta)
         cos2t = cos2t + (DCOS(om * t(i)-theta))**2
         cost2 = cost2 + DCOS(om * t(i)-theta)
!       Finding amplitude another method
!         xsin = xsin + y(i) * DSIN(om * t(i)-theta)
!         sin2t = sin2t + DCOS(om * t(i)-theta) * DSIN(om * t(i)-theta)
!         sint2 = sint2 + (DSIN(om * t(i)-theta))
         ENDDO
         amp = xcos/(cos2t-cost2**2/(DBLE(N)))! eq 52 from 45
         IF (amp .LT. 0.0) THEN
          IF (theta .GT. 0.0) THEN
            theta = theta - PI
          ELSE
            theta = theta + PI
          ENDIF
         amp = -amp
         ENDIF
```

```
!         write (*,*) 'Values of amp is',amp,'phase',theta
!         amplit = xsin/(sin2t-cost2 * sint2/(DBLE(N)))
!         write (*,*) 'Values of amplit is',amplit
         END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE phase(t,y,N,freq,theta)
        IMPLICIT NONE
        INTEGER N,i,k,l,m
        DOUBLE PRECISION t(N),y(N),om,co,si,cosin,PI,freq
        DOUBLE PRECISION p1,p2,p3,cos2,sin2,theta,ycos,ysin
!        DOUBLE PRECISION num,den,theta1
        DOUBLE PRECISION csum,ssum,xcsum,xssum,cssum,c2sum,s2sum
        csum = 0.d0
        ssum = 0.d0
        xcsum = 0.d0
        xssum = 0.d0
        cssum = 0.d0
        c2sum = 0.d0
        s2sum = 0.d0
        co = 0.d0
        si = 0.d0
        ycos = 0.d0
        ysin = 0.d0
        cosin = 0.d0
        cos2 = 0.d0
        sin2 = 0.d0
        PI = 4.d0 * DATAN(1.d0)
        om = 2.d0 * PI * freq
!       Finding phase
        DO i = 1,N
        co = co + DCOS(om * t(i))! cos wt
        si = si + DSIN(om * t(i)) ! sin wt
        ycos = ycos + y(i) * DCOS(om * t(i))! y cos wt
        ysin = ysin + y(i) * DSIN(om * t(i)) ! y sin wt
        cosin = cosin + 0.5d0 * DSIN(2.d0 * om * t(i))
        cos2 = cos2 + (DCOS(om * t(i)))**2
        sin2 = sin2 + (DSIN(om * t(i)))**2
        ENDDO
        p1 = DBLE(N) * cosin - co * si
        p2 = DBLE(N) * cos2 - co * co
        p3 = DBLE(N) * sin2 - si * si
!        theta = DATAN2((p1 * ycos - p2 * ysin),(p1 * ysin - p3 * ycos))
!        IF (theta .LT. 0.d0) theta = theta + PI
!        IF (theta .GT. 2.d0 *  PI) theta = theta - PI
!        write(*,*)'p1 p2 p3  =  ',p1, p2, p3
```

```
!       write(*,*)'theta value is  =  ',theta
        DO i = 1,N
        csum = csum + DCOS(om * t(i))! cos wt
        ssum = ssum + DSIN(om * t(i)) ! sin wt
        xcsum = xcsum + y(i) * DCOS(om * t(i))! y cos wt
        xssum = xssum + y(i) * DSIN(om * t(i)) ! y sin wt
        cssum = cssum + 0.5d0 * DSIN(2.d0 * om * t(i))
        c2sum = c2sum + (DCOS(om * t(i)))**2
        ENDDO
        cssum = cssum - csum * ssum / N
        s2sum = N - c2sum - ssum * ssum / N
        c2sum = c2sum - csum * csum / N
      theta=DATAN2((cssum*xcsum-c2sum*xssum),(cssum*xssum-s2sum*xcsum))
!        write(*,*)'sigspec theta value is  =  ',theta
!       Finding phase theta directly
!        num = 0.d0
!        den = 0.d0
!        DO k = 1,N
!        DO l = 1,N
!        DO m = 1,N
!        num = num + (DSIN(om*(t(l)-t(k)))-DSIN(om*(t(m)-t(k))))*
!     5  y(k) * DCOS(om*t(l))
C        write(*,*)'Numerator value is  =  ',num
!        den = den  +  (DSIN(om*(t(l)-t(k)))-DSIN(om*(t(m)-t(k))))*
!     6  y(k) * DSIN(om*t(l))
C        write(*,*)'Denominator value is  =  ',den
!        ENDDO
!        ENDDO
!        ENDDO
!        theta1  =  -datan2(num,den)
!        IF (theta1 .LT. 0.d0) theta1 = theta1 + PI
!        IF (theta1 .GT. 2.d0* PI) theta1 = theta1-PI
!        write(*,*)'### theta1 value is  =  ',theta1
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C finding significance
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE rmsdev(t,y,N,frequ,ampl,thet,h)
        IMPLICIT NONE
        INTEGER N,i,j,k,c,h
        DOUBLE PRECISION t(N),y(N),ffreq(h),aamp(h),ttheta(h)
        DOUBLE PRECISION rms(h),PI,err,ofreq,tth,aam
        DOUBLE PRECISION frequ,ampl,thet,mrms
        OPEN(unit = 23,file = 'rms.dat',status = 'unknown')
        ofreq = 0.d0
```

```
      DO i = 1,h
       ffreq(i) = 0.d0
       aamp(i) = 0.d0
       ttheta(i) = 0.d0
      ENDDO
      k = 0
      frequ = frequ/4.d0
      DO i = 1,h
      tth = 0.d0
      aam = 0.d0
      ofreq = frequ * DBLE(k+1)
      CALL phase(t,y,N,ofreq,tth)
      CALL amplitude(t,y,N,ofreq,tth,aam)
      ffreq(i) = ofreq
      ttheta(i) = tth
      aamp(i) = aam
      WRITE(*,*)'freq',ffreq(i),'amp',aamp(i),'phase',ttheta(i)
      k = k + 1
      ENDDO
      PI = 4.d0 * DATAN(1.d0)
      DO j = 1,h
      rms(j) = 0.d0
      DO i = 1,N
      rms(j) = rms(j) + (y(i) - aamp(j) * DCOS(2.d0 * PI * ffreq(j)
     & * t(i)-ttheta(j)))**2 + (y(i)- aamp(j+1) * DCOS(2.d0 *
     & PI * ffreq(j+1) * t(i)-ttheta(j+1)))**2
      ENDDO
      ENDDO
!     sorting to get min rms value
      mrms = rms(1)
      c = 1
      DO j = 2,h
      IF ( rms(j) .LT. mrms ) THEN
      mrms = rms(j)
      c = j
      ENDIF
      ENDDO
      frequ = ffreq(c)
      ampl = aamp(c)
      thet = ttheta(c)
!      IF (rmsd .LT. err) freq(im) = freq(im) / 2.d0
      CLOSE(23)
      END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      SUBROUTINE  Profile_SigSpec(t,y,N,lf,uf,nspr,fs,profa0,
     *  profb0,proft0)
      IMPLICIT NONE
      INTEGER N,i,ri,fi,nspr
      DOUBLE PRECISION y(N),t(N)
      DOUBLE PRECISION curcos,cursin,ddcos,ddsin,dbuf
      DOUBLE PRECISION proft0(nspr),sinth0(nspr),costh0(nspr)
      DOUBLE PRECISION df,LG_E,PI,axis1(nspr),axis2(nspr)
      DOUBLE PRECISION lf,fs,sd,uf,profa0(nspr),profb0(nspr)
      df = (uf - lf) / (nspr - 1.d0)
      WRITE(*,*) 'sampling profile: initialise arrays'
      curcos = 0.d0
      cursin = 0.d0
      ddcos = 0.d0
      ddsin = 0.d0
      dbuf = 0.d0
      PI = 4.d0*DATAN(1.d0)
      LG_E = DLOG10(DEXP(1.d0))
      DO fi = 1,nspr
      sinth0(fi) = 0.d0
      costh0(fi) = 0.d0
      proft0(fi) = 0.d0
      profb0(fi) = 0.d0
      axis1(fi) = 0.d0
      axis2(fi) = 0.d0
      ENDDO
      WRITE(*,*)'sampling profile: orientation of rms error ellipse'
!*******************************************
      DO ri = 1,N
      curcos = DCOS(2.d0 * PI * lf * t(ri))
      cursin = DSIN(2.d0 * PI * lf * t(ri))
      ddcos = DCOS(2.d0 * PI * df * t(ri))
      ddsin = DSIN(2.d0 * PI * df * t(ri))
      DO fi = 1,nspr
      axis1(fi) = axis1(fi) + curcos  ! SUM cos wt
      axis2(fi) = axis2(fi) + cursin  ! SUM sin wt
      proft0(fi) = proft0(fi) + (curcos*curcos - cursin*cursin)
      profb0(fi) = profb0(fi) +  curcos * cursin
      dbuf = curcos * ddcos - cursin * ddsin
      cursin = cursin * ddcos + curcos * ddsin
      curcos = dbuf   ! cos wt cos dwt - sin wt sin dwt
!WRITE(*,*)'sampling profile: orientation of rms error ellipse',nspr * n
      ENDDO
      WRITE(6,"(A)",ADVANCE='NO')'='
      FLUSH 6
      ENDDO
```

```
!*******************************************
       DO fi = 1,nspr
       sinth0(fi) = 0.d0
       costh0(fi) = 0.d0
       sinth0(fi) = sinth0(fi) + 2.d0 * (DBLE(N) * profb0(fi) -
     * axis1(fi)*axis2(fi))!piet reegen
       costh0(fi) = costh0(fi) + DBLE(N) * proft0(fi) -
     * (axis1(fi)**2) +(axis2(fi)**2)
       ENDDO
!*******************************************
! calculates theta0
       WRITE(*,*)
       WRITE(*,*)'sampling profile: axes of rms error ellipse'
       DO fi = 1,nspr
       proft0(fi) = 0.d0
       proft0(fi) = 0.5d0 * DATAN2(sinth0(fi), costh0(fi))
       IF (proft0(fi) .LT. 0.d0) proft0(fi) = proft0(fi) + PI
       sinth0(fi) = 0.d0
       costh0(fi) = 0.d0
       ENDDO
!*******************************************
       curcos = 0.d0
       cursin = 0.d0
       ddcos = 0.d0
       ddsin = 0.d0
       dbuf = 0.d0
!*******************************************
       DO fi = 1,nspr
       axis1(fi) = 0.d0
       axis2(fi) = 0.d0
       profb0(fi) = 0.d0
       ENDDO
!*******************************************
       DO ri = 1,N
       curcos = DCOS(2.d0 * PI * lf * t(ri))
       cursin = DSIN(2.d0 * PI * lf * t(ri))
       ddcos = DCOS(2.d0 * PI * df * t(ri))
       ddsin = DSIN(2.d0 * PI * df * t(ri))
       DO fi = 1,nspr
       dbuf = curcos * DCOS(proft0(fi)) + cursin * DSIN(proft0(fi))
       axis1(fi) = axis1(fi) + dbuf
       axis2(fi) = axis2(fi) + (cursin * DCOS(proft0(fi))-curcos*
     * DSIN(proft0(fi)))
       profb0(fi) = profb0(fi) + (dbuf*dbuf)
       dbuf = curcos * ddcos - cursin * ddsin
       cursin = cursin * ddcos + curcos * ddsin
```

```
        curcos = dbuf
!       WRITE(*,*) 'sampling profile: axes of rms error ellipse',nspr*n
        ENDDO
        WRITE(6,"(A)",ADVANCE='NO')'='
        FLUSH 6
        ENDDO
!*******************************************
        DO fi = 1,nspr
        sinth0(fi) = 0.d0
        costh0(fi) = 0.d0
        costh0(fi) = costh0(fi)+DABS(profb0(fi)-(axis1(fi)**2/DBLE(N)))
        sinth0(fi)=sinth0(fi)+DABS(N-profb0(fi)-(axis2(fi)**2)/DBLE(N))
        ENDDO
        WRITE(*,*)
        WRITE(*,*)'sampling profile: finalise'
!*******************************************
        DO fi = 1,nspr
!        IF (lf + fs * fi .EQ. 0.) then
!        profa0(fi) = 0.d0
!        profb0(fi) = 0.d0
!        proft0(fi) = 0.d0
!        ELSE
        IF (DABS(costh0(fi)) .GT. DABS(sinth0(fi))) then
        profa0(fi) = DSQRT(2.d0 * DABS(sinth0(fi)) /DBLE(N))
        profb0(fi) = DSQRT(2.d0 * DABS(costh0(fi)) /DBLE(N))
        proft0(fi) = proft0(fi) + .5 * PI
        ELSE
        profa0(fi) = DSQRT(2.d0 * DABS(costh0(fi)) /DBLE(N))
        profb0(fi) = DSQRT(2.d0 * DABS(sinth0(fi)) /DBLE(N))
        ENDIF
        DO WHILE (proft0(fi) .LT. 0.)
        proft0(fi) = proft0(fi) + PI
        ENDDO
        DO WHILE (proft0(fi) .GT. PI)
        proft0(fi) = proft0(fi) - PI
        ENDDO
!       WRITE(*,*)'sampling profile: finalise',fi + 1,nspr
        ENDDO
!*******************************************
        END
!CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!!! here starts sigpspec calculation !!!!!!!!!!!!!!!!!
!CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE  SigSpec_SigSpec(t,y,N,lf,nspr,fs,sd,fmax,profa0,
     *  profb0,proft0,sigmax,nof)
        IMPLICIT NONE
```

```
        INTEGER N,i,ri,nspr,fi,nf,in,nof
        DOUBLE PRECISION y(N),t(N),fharm(nspr),sharm(nspr),fharm1(nspr)
        DOUBLE PRECISION curcos,cursin,fmax,norm,dbuf,ddcos,ddsin
        DOUBLE PRECISION sigmax,freq(nspr),LG_E,PI,lf,fs,sd,sigcut
        DOUBLE PRECISION a(nspr),b(nspr),sig(nspr),sharm1(nspr)
        DOUBLE PRECISION proft0(nspr),profa0(nspr),profb0(nspr)
!        OPEN(unit = 20,file = 'harm.dat',status = 'unknown')
!        OPEN(unit = 21,file = 'spectrum1.dat',status = 'unknown')
!        OPEN(unit = 22,file = 'spectrum2.dat',status = 'unknown')
        in = 0
        curcos = 0.d0
        cursin = 0.d0
        ddsin = 0.d0
        ddcos = 0.d0
        dbuf = 0.d0
        sigmax = 0.d0
        sigcut = 0.d0
        fmax= -1.d0
        PI = 4.d0*DATAN(1.d0)
        LG_E = DLOG10(DEXP(1.d0))
        DO fi = 1,nspr
        a(fi) = 0.d0
        b(fi) = 0.d0
        freq(fi) = 0.d0
        sig(fi) = 0.d0
        ENDDO
 !*******************************************
        WRITE(*,*)
        WRITE(*,*)'significance spectrum: Fourier Coefficients'
        DO ri = 1,N
        curcos = y(ri) * DCOS(2.d0 * PI * lf * t(ri)) /DBLE(N)
        cursin = y(ri) * DSIN(2.d0 * PI * lf * t(ri)) /DBLE(N)
        ddcos = DCOS(2.d0 * PI * fs * t(ri))
        ddsin = DSIN(2.d0 * PI * fs * t(ri))
          DO fi = 1,nspr
          a(fi) = a(fi)+ curcos
          b(fi) = b(fi)+ cursin
          dbuf = curcos * ddcos - cursin * ddsin
          cursin = cursin * ddcos + curcos * ddsin
          curcos = dbuf
!WRITE(*,*)'SigSpec spectrum: Fourier Coefficients',ri*nspr+fi,'%',n*nspr)
          ENDDO
        WRITE(6,"(A)",ADVANCE='NO')'='
        FLUSH 6
        ENDDO
!*******************************************
```

```
        WRITE(*,*)'sd = ',sd
        norm = LG_E * N / (sd*sd)
        WRITE(*,*)
        WRITE(*,*)'norm =',norm
!        freq(1) = lf
        DO fi = 1,nspr
        freq(fi) = 0.d0
        sig(fi) = 0.d0
        freq(fi) = lf + fs * in
!       IF ((freq .GT. 0.99) .and.(freq .LT. 1.01)) goto 47
        sig(fi) = norm * ((((a(fi) * DCOS(proft0(fi))+
     #  b(fi) * DSIN(proft0(fi))) /  profa0(fi))**2) + (((b(fi) *
     #  DCOS(proft0(fi)) - a(fi) * DSIN(proft0(fi))) / profb0(fi))**2))
!   47  continue
        IF (sig(fi) .GT. sigmax) then
        fmax = freq(fi)
        sigmax = sig(fi)
        ENDIF
!        if (nof .EQ. 1) WRITE(21,*),freq(fi),sig(fi)
!        if (nof .EQ. 2) WRITE(22,*),freq(fi),sig(fi)
        in = in + 1
        ENDDO
!*******************************************
        WRITE(*,*)
        WRITE(*,*)'fmax sigmax',fmax,sigmax
        sigcut = sigmax * 0.5d0
        IF (sigcut .GT. 5.d0) THEN
        DO fi = 1,nspr
        fharm1(fi) = 0.d0
        sharm1(fi) = 0.d0
        ENDDO
        nf = 0
        DO fi = 1,nspr
        IF (sig(fi) .GT. sigcut) THEN
        fharm1(fi) = freq(fi)
        sharm1(fi) = sig(fi)
        nf = nf+1
        ENDIF
        ENDDO
        WRITE(*,*)'Number of frequencies selected above cutoff',nf
        ri=1
        DO fi = 1,nspr
        IF (sig(fi) .GT. sigcut) THEN
        fharm(ri) = fharm1(fi)
        sharm(ri) = sharm1(fi)
!        WRITE(20,*),fharm(ri),sharm(ri)
```

```
        ri=ri+1
        ENDIF
        ENDDO
        ENDIF
 !      CLOSE(20)
 !      CLOSE(21)
 !      CLOSE(22)
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC      INCREASING ACCURACY OF FREQUENCY BY CHECKING FOR MAX.SIG WITH FINER
CC      NEIGHBOURING VALUES
        SUBROUTINE  SigSpec_MaxSig(t,y,N,freq,sig,amp,th,fs,sd)
        IMPLICIT NONE
        INTEGER N,i,kount
        DOUBLE PRECISION y(N),t(N)
        DOUBLE PRECISION freq,fs,prevf,psig,sig,nextf,nsig,rsig
        DOUBLE PRECISION amp,th,sd,ppsc,origfs
        prevf = 0.d0
        nextf = 0.d0
        sig = 0.d0
        psig = 0.d0
        rsig = 0.d0
        nsig = 0.d0
        amp = 0.d0
        th = 0.d0
        kount=1
        origfs = fs
        IF (freq .GT. fs) fs = fs/2.d0
        prevf = freq - fs
        CALL SigSpec_Sig(t,y,N,prevf,psig,amp,th,sd)
        CALL SigSpec_Sig(t,y,N,freq,rsig,amp,th,sd)
        nextf = freq + fs
        CALL SigSpec_Sig(t,y,N,nextf,nsig,amp,th,sd)
!NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
!      here specify accuracy of frequency
        DO WHILE (fs .GT. 0.00000000000001)
!       WRITE(*,*)'Doing this loop',kount,'times'
        fs = fs/2.d0
!       WRITE(*,*)'frequency spacing is ',fs
        IF ((rsig .ge. psig) .and. (rsig .ge. nsig)) then
          prevf = freq - fs
          CALL SigSpec_Sig(t,y,N,prevf,psig,amp,th,sd)
          nextf = freq + fs
          CALL SigSpec_Sig(t,y,N,nextf,nsig,amp,th,sd)
        ELSE IF (psig .GT. nsig) then
          nextf = freq
```

```
          nsig = rsig
          freq = freq - fs
          CALL SigSpec_Sig(t,y,N,freq,rsig,amp,th,sd)
        ELSE
          prevf = freq
          psig = rsig
          freq = freq + fs
          CALL SigSpec_Sig(t,y,N,freq,rsig,amp,th,sd)
        ENDIF
          kount=kount+1
        ENDDO
!NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
!       return frequency after maximizing sig
        sig = rsig
        fs = origfs ! backing up original fs ..otherwise low value goes
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       finding significance as sigspec program

        SUBROUTINE  SigSpec_Sig(t,y,N,freq,newsig,sigamp,th,sd)
        IMPLICIT NONE
        INTEGER N,i,hi,ri,si
        DOUBLE PRECISION y(N),t(N),norm
        DOUBLE PRECISION cossum,cossum2,sinsum,sinsum2,xcossum,xsinsum
        DOUBLE PRECISION a,b,sin2th0,cos2th0,newsig,om,LG_E,PI
        DOUBLE PRECISION a0,b0,freq,sigamp,t0,th,sd,axis1,axis2
        ri = 1
        CALL Stat_ZeroMean(y,N)
        newsig = 0.d0
        sigamp = 0.d0
        th = 0.d0
        sin2th0 = 0.d0
        cos2th0 = 0.d0
        a = 0.d0
        b = 0.d0
        cossum = 0.d0
        cossum2 = 0.d0
        sinsum = 0.d0
        sinsum2 = 0.d0
        xcossum = 0.d0
        xsinsum = 0.d0
        t0 = 0.d0
        PI = 4.d0*datan(1.d0)
        LG_E = DLOG10(DEXP(1.d0))
        om = 2.d0*PI*freq
```

```
DO ri = 1,N
cossum = cossum +  DCOS(om  * t(ri))
sinsum =  sinsum +  DSIN(om * t(ri))
cossum2 = cossum2 + DCOS(2.d0 * om * t(ri))
sinsum2 = sinsum2 +  DSIN(2.d0 * om * t(ri))
xcossum = xcossum +  y(ri) * DCOS(om * t(ri))
xsinsum = xsinsum +  y(ri) * DSIN(om * t(ri))
ENDDO
a = a + xcossum
b = b + xsinsum
sin2th0 = sin2th0 + (N * sinsum2 - 2.d0 * cossum * sinsum)
cos2th0 = cos2th0 + (N * cossum2 - (cossum**2) + (sinsum**2))
a = a / DBLE(N)
b = b / DBLE(N)
sigamp = 2.d0 * DSQRT(a**2 + b**2)
th = DATAN2(b,a)
t0 = .5d0 * DATAN2(sin2th0, cos2th0)
IF (t0 .LT. 0.d0) t0 = t0 + PI
axis1 = 0.d0
axis2 = 0.d0
cossum = 0.d0
cossum2 = 0.d0
sinsum = 0.d0
sinsum2 = 0.d0
a0 = 0.d0
b0 = 0.d0
DO ri = 1,N
cossum = cossum +  DCOS(om * t(ri) - t0)
cossum2 = cossum2 +  (DCOS(om * t(ri) - t0))**2
sinsum = sinsum + DSIN(om * t(ri) - t0)
ENDDO
sinsum2 = DBLE(N) - cossum2
axis1 = axis1 + cossum2 - (cossum**2) /DBLE(N)
axis2 = axis2 + sinsum2 - (sinsum**2) /DBLE(N)
IF (DABS(axis1) .GT. DABS(axis2)) then
a0 = 2.d0 /DBLE(N) * DABS(axis2)
b0 = 2.d0 /DBLE(N) * DABS(axis1)
IF (t0 .LT. .5d0 * PI) then
t0 = t0 + .5d0 * PI
ELSE
t0 = t0 - .5d0 * PI
ENDIF
ELSE
a0 = 2.d0 /DBLE(N) * DABS(axis1)
b0 = 2.d0 /DBLE(N) * DABS(axis2)
ENDIF
```

```
      norm = LG_E*N/(sd*sd)
      newsig = norm *(((a*DCOS(t0)+b*DSIN(t0))**2/a0) + ((a*DSIN(t0)-
     $ b * DCOS(t0))**2/b0))
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE MultiSine_Newton(t,y,N,ramp,rfreq,rth,rsig,yorig,
     *  origsd,ik,shift)
      IMPLICIT NONE
      INTEGER N,i,ri,kall,ii,ik,kount,ir,j
      DOUBLE PRECISION y(N),t(N),sd,var1,var2,rsig(ik)
      DOUBLE PRECISION var1f(ik),var2f(ik),var1amp(ik)
      DOUBLE PRECISION ramp(ik),rfreq(ik),rth(ik),var1th(ik),PI
      DOUBLE PRECISION var2amp(ik),var2th(ik),shift,cond1,cond2
      DOUBLE PRECISION yorig(N),var2a0(ik),var2b0(ik),oldsd
      DOUBLE PRECISION origsd,newsd,newppsc,torig(N)
      DO ii = 1,ik
      var1f(ii) = 0.d0
      var2f(ii) = 0.d0
      var1amp(ii) = 0.d0
      var2amp(ii) = 0.d0
      var1th(ii) = 0.d0
      var2th(ii) = 0.d0
      ENDDO
      kall = 0
      kount = 0
      oldsd = origsd
      newsd = origsd
      PI=4.d0*DATAN(1.d0)
      DO i = 1,N
      torig(i) = t(i)
      ENDDO
      shift = -shift
      CALL TS_TimeShift(t,N,shift)
      DO ir = 1,ik
      CALL Result_TimeShift(t,N,rfreq,shift,rth,ir)
      ENDDO
!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      DO WHILE (kall .EQ. 0)
      kount = kount+1
      CALL MultiSine_Derive(t,N,ramp,rfreq,rth,var1f,var2f,var1amp
     1 ,var2amp,var1th,var2th,yorig,ik)
!****************************************
      DO i = 1,N
      y(i) = yorig(i)
```

```
        ENDDO
!YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
        DO ii = 1,ik
!***************************************
        rfreq(ii) = rfreq(ii) - var1f(ii) / var2f(ii)
!***************************************
        ramp(ii) = ramp(ii) - var1amp(ii) / var2amp(ii)
        IF (ramp(ii) .LT. 0.d0) then
        ramp(ii) = -ramp(ii)
        rth(ii) = rth(ii) - PI
        ENDIF
!***************************************
        rth(ii) = rth(ii) - var1th(ii) / var2th(ii)
          DO WHILE (rth(ii) .LT. 0.d0)
          rth(ii) = rth(ii) + 2.d0 * PI
          ENDDO
        DO WHILE (rth(ii) .ge. (2.d0 * PI))
        rth(ii) = rth(ii) - 2.d0* PI
        ENDDO
!***************************************
! Pre-Whitening here
        DO ri = 1,N
        y(ri) = y(ri)-ramp(ii)*DCOS(2.d0*PI*rfreq(ii)*t(ri)-rth(ii))
        ENDDO
        cond1 = DABS(var1f(ii) / var2f(ii))
        cond2 = 0.000001d0 * (1.d0/DSQRT(rsig(ii))) / (t(N)-t(1))
        IF (cond1 .GT. cond2) kall = 0
        ENDDO
!YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
        CALL Stat_ZeroMean(y,N)
        oldsd = newsd
        CALL Stat_SD(y,N,newsd)
        IF ((1.d0 - newsd / oldsd) .LT. 0.000001) kall = 1
! Terminating Condition
        WRITE(*,*)' MultiSine fit: iteration',kount,'rms res ',newsd
        ENDDO
!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        CALL stat_ppscatter(y,N,newppsc)
        shift = -shift
        CALL TS_TimeShift(t,N,shift)
        DO ir = 1,ik
        CALL Result_TimeShift(t,N,rfreq,shift,rth,ir)
        ENDDO
        END ! MultiSine_Newton

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      SUBROUTINE MultiSine_Derive(t,N,mdamp,mdfreq,mdth,var1f,var2f
     * ,var1amp,var2amp,var1th,var2th,yorig,it)
      IMPLICIT NONE
      INTEGER N,i,ri,it,ii
      DOUBLE PRECISION y(N),t(N),yorig(N)
      DOUBLE PRECISION sine,cosine,hsin,hcos,tsin,rcos
      DOUBLE PRECISION var1f(it),var2f(it),var1th(it),var2th(it),PI
      DOUBLE PRECISION var2a0(it),var2b0(it),var1amp(it),var2amp(it)
      DOUBLE PRECISION mdfreq(it),mdamp(it),mdth(it)
      DO ii = 1,it
      ENDDO
      DO i = 1,N
      y(i) = yorig(i)
      ENDDO
      PI=4.d0*DATAN(1.d0)
!**************************************
      DO ii = 1,it
      DO ri = 1,N
      y(ri)=y(ri)-mdamp(ii)*DCOS(2.d0*PI*mdfreq(ii)*t(ri)-mdth(ii))
      ENDDO
      ENDDO
!**************************************
      CALL Stat_ZeroMean(y,N)
!MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
      DO ii = 1,it
      tsin = 0.d0
      sine = 0.d0
      cosine = 0.d0
      rcos = 0.d0
      var2a0(ii) = 0.d0
      var2b0(ii) = 0.d0
      var1th(ii) = 0.d0
      var2th(ii) = 0.d0
      var1amp(ii) = 0.d0
      var2amp(ii) = 0.d0
      var1f(ii) = 0.d0
      var2f(ii) = 0.d0
!OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
      DO ri = 1,N
      hsin = 0.d0
      hcos = 0.d0
      sine = DSIN(2. * PI * mdfreq(ii) * t(ri) - mdth(ii))
      cosine = DCOS(2. * PI * mdfreq(ii) * t(ri) - mdth(ii))
      hsin = hsin + mdamp(ii) * sine
      hcos = hcos + mdamp(ii) * cosine
```

```
        rcos = y(ri) * cosine
        var1th(ii) = var1th(ii) - mdamp(ii) * y(ri) * sine
        var1amp(ii) = var1amp(ii) - rcos
        var2a0(ii) = var2a0(ii) + cosine
        var2b0(ii) = var2b0(ii) + sine
        var2amp(ii) = var2amp(ii) + (cosine * cosine)
        var2th(ii) = var2th(ii)+(mdamp(ii)*sine)**2 + mdamp(ii)*rcos
        var1f(ii) = var1f(ii) + y(ri) * t(ri) * hsin
        var2f(ii) = var2f(ii)+(t(ri)**2)*(hsin**2+y(ri)*hcos)
        tsin  = tsin + t(ri) * hsin
        ENDDO
!OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
        var2f(ii) = var2f(ii) - (tsin ** 2) /DBLE(N)
        var2amp(ii) = var2amp(ii) - (var2a0(ii) ** 2) /DBLE(N)
        var2th(ii) = var2th(ii)-((mdamp(ii)*var2b0(ii))** 2)/DBLE(N)
        var1f(ii) = var1f(ii) * 4.d0 * PI /DBLE(N)
        var2f(ii) = var2f(ii) * 8.d0 * PI * PI /DBLE(N)
        var1amp(ii) = var1amp(ii) * 2.d0 /DBLE(N)
        var1th(ii) = var1th(ii) * 2.d0 /DBLE(N)
        var2amp(ii) = var2amp(ii) * 2.d0 /DBLE(N)
        var2th(ii) = var2th(ii) * 2.d0 /DBLE(N)
        ENDDO
!MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE Result_TimeShift(t,N,tsfreq,shift,rsth,ir)
        IMPLICIT NONE
        INTEGER N,ir
        DOUBLE PRECISION t(N),shift,PI,tsfreq(ir),rsth(ir)
        !PARAMETER(PI = 3.14159265358979323846264)
        PI = 4.d0 * DATAN(1.d0)
        rsth(ir) = rsth(ir) + 2. * PI * tsfreq(ir) * shift
        DO WHILE (rsth(ir) .LT. 0.d0)
         rsth(ir) = rsth(ir) + 2.d0 * PI
        ENDDO
        DO WHILE (rsth(ir) .ge. (2.d0 * PI))
         rsth(ir) = rsth(ir) - 2. * PI
        ENDDO
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE TS_TimeShift(t,N,shift)
        IMPLICIT NONE
```

```
        INTEGER N,i
        DOUBLE PRECISION t(N),shift
        DO i = 1,N
         t(i) = t(i)+shift
        ENDDO
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE Stat_SD(y,N,sd)
        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION y(N),cvar,sd
        sd = 0.d0
        CALL Stat_Variance(y,N,cvar)
        sd = DSQRT(cvar)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C       average is subtracting from magnitude
        SUBROUTINE  Stat_ZeroMean(y,N)
        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION y(N),ymean
        ymean = 0.d0
        CALL Stat_Mean(y,N,ymean)
        DO i = 1,N
        y(i) = y(i)-ymean
        ENDDO
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE Stat_Variance(y,N,bvar)
        IMPLICIT NONE
C       this is 1/n of sum over (x-xbar) all square
        INTEGER N,i
        DOUBLE PRECISION y(N),b(N),bbar,bvar,bi2,mean
        bi2 = 0.d0
        bbar = 0.d0
        bvar = 0.d0
        mean = 0.d0
        CALL Stat_Mean(y,N,bbar)
        DO i = 1,N
        b(i) = y(i)
```

```
        b(i) = b(i)-bbar
        bi2 = bi2+b(i)*b(i)
        ENDDO
        bvar = bi2/DBLE(N)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE Stat_Mean(y,N,ybar)
        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION y(N),ybar
C       Finding Mean of y
        ybar = 0.d0
        DO i = 1,N
        ybar = ybar+y(i)
        ENDDO
        ybar = ybar/DBLE(N)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE t_Mean(t,N,tbar)
        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION t(N),tbar
C       Finding Mean of t
        tbar = 0.d0
        DO i = 1,N
        tbar = tbar+t(i)
        ENDDO
        tbar = tbar/DBLE(N)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE stat_ppscatter(y,N,ppscatter)
        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION y(N),sumsq,ppscatter
        sumsq = 0.d0
        ppscatter = 0.d0
        DO i = 1,N-1
        sumsq=sumsq+0.25d0*(y(i+1)-y(i))**2
        ENDDO
        sumsq = sumsq+0.25d0*((y(N)-y(1))**2)
```

```
      ppscatter = DSQRT(sumsq/DBLE(N))
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE Nyquist_Coef(t,N,nyfreq,nycoef)
      IMPLICIT NONE
      DOUBLE PRECISION t(N),nycoef,nyfreq,nf
      INTEGER N,i,kount
      kount = 0
      DO i = 1,N-1
      IF ((t(i+1)-t(i)) .LE. .5/nyfreq) kount = kount+1
      ENDDO
      nycoef = (DBLE(kount)/DBLE(N))
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE Nyquist_Freq(t,N,tspan,nc,nyfreq)
      IMPLICIT NONE
      DOUBLE PRECISION t(N),nyfreq,tspan,step,test,nc,tdiff
      DOUBLE PRECISION l/99999999.d0/,u/-99999999.d0/
      INTEGER N,i,kount/0/
      nyfreq = 0.d0
      step = 0.d0
      kount = 0
      DO i = 1,N
      IF (t(i) .LT. l) l = t(i)
      IF (t(i) .GT. u) u = t(i)
      ENDDO
      DO WHILE (kount .LT. INT(N*(1.d0-nc)))
      test = u-l
      kount = 0
      DO i = 1,N-1
      tdiff = t(i+1)-t(i)
      IF ((tdiff .LT. test) .AND. (tdiff .GT. step)) test = tdiff
      ENDDO
      DO i = 1,N-1
      IF ((t(i+1)-t(i)) .LE. test) kount = kount+1
      ENDDO
      step = test
      ENDDO
      nyfreq = .5d0/step
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
!  csig finds from second freq. onwards for first freq it is same as sig

      SUBROUTINE Sigspec_Csig(rsig,csig,ct)
      IMPLICIT NONE
      INTEGER ct
      DOUBLE PRECISION rsig(ct),csig(ct)
      csig(ct-1) = rsig(ct)
      sig = -sig
      csig =  -dlog10(1.d0 - (1.d0- (10**(-csig)))*(1.d0-(10 ** -sig)))
      csig(ct) = -DLOG10(1.d0-(1.d0-(10.d0**(-csig(ct-1))))*(
     *  (1.d0-(10.d0**(-rsig(ct))))))
      IF (csig(ct) .GT. rsig(ct)) csig(ct) = rsig(ct)
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

Appendix B

Spline and phase folding programs

B.1 Spline program

Program for confirming period using modified cubic spline method, which can be automated in combination with the script Appendix C.3

```
! CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!  _ _ _ _ _            _ _ _ _ _
! / / / / /   SPLINE  / / / / /
!
! CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!spline.for is for Period confirmation of variable stars from
!the results of SigSpec/GLS/PDM
!This program is written for full automation, hence no values
!has to be given as input
!name this program as spline.for
!Keep all data files (*.lc) and the results of SigSpec/GLS/PDM
!in a folder along with
!this program and script startspline.sh
!and compile this fortran program from the same terminal, as
!gfortran -o spline spline.for in the same folder.
!Then run the script as  ./startspline.sh
!The entire files in the folder are processed one by one.
!SPLINE Fortran program is written by Shaju K.Y. and
!Ramesh Babu Thayyullathil
!Cochin University of Science and Technology, Kochi, Kerala,
!India 682022
!June 2010
!For any queries or bugs, please contact shajuky@gmail.com

        PROGRAM MAIN
```

```
        IMPLICIT NONE
!Program starts here. The total number of observations read here
!The number of observations read by script and produce file forin.dat
!based on the testing period, the data should be folded and given
!as phase.txt
!phase.txt file is produced by phase.for, given at the end of this file.
!The phase.for should be separated before compiling this file.

        INTEGER LINE,NBIN
        DOUBLE PRECISION PERIOD

        OPEN(unit = 17,file = 'forin.dat',status = 'unknown')

        READ(17,*) LINE,PERIOD

        NBIN=1
        IF (LINE .GT. 10) NBIN=2
        IF (LINE .GT. 50) NBIN=3
        IF (LINE .GT. 100) NBIN=6
        IF (LINE .GT. 500) NBIN=8
        IF (LINE .GT. 1000) NBIN=10
        IF (LINE .GT. 2000) NBIN=12
        IF (LINE .GT. 3000) NBIN=12
        IF (LINE .GT. 4000) NBIN=14
        IF (LINE .GT. 5000) NBIN=16
        CALL FIRSTFN(LINE,PERIOD,NBIN)
        CLOSE(17)
        STOP
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE FIRSTFN(ND,PERIOD,NHBIN)

      IMPLICIT NONE
      INTEGER I,ND,NHBIN,IO_ERR/0/
      INTEGER NS,NSP(NHBIN),NK,NKC,NW,BD(NHBIN)
      DOUBLE PRECISION TD(ND),YD(ND),PERIOD,DEL(NHBIN),T0,TM,DELBIN
      CHARACTER line*1024

      OPEN (unit = 18,file="phase.txt",status='unknown')

        DO I = 1,ND
        READ (18,'(a)',END = 73) line    ! reading line from datafile
        READ(line,*,IOSTAT = IO_ERR) TD(I),YD(I) ! here actual READing
        IF (IO_ERR.NE.0 ) THEN
```

```
         WRITE(6,*) "Something is wrong with input data file !!!"
         WRITE(6,*) " Please check the format of file"
         CLOSE(18)
         GOTO 67   !  INPUT FILE PROBLEM
         ENDIF
         ENDDO
73      CLOSE (18)
        T0=TD(1)
        TM=TD(ND)
!       call BINAVERAGE or KURTOSIS for unevenly spaced knots
        CALL BINAVERAGE(TD,YD,ND,T0,TM,NHBIN,DEL,NSP,NK,BD)
!       CALL KURTOSIS(TD,YD,ND,NHBIN,DEL,NSP,NK,BD)

        NS=NK-4
        NKC=5
        NW=((NS+1)*NS)/2

       CALL SUBRMS(TD,YD,ND,T0,TM,NS,NKC,NK,NW,NHBIN,DEL,PERIOD,NSP,BD)

       RETURN
67     END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       SUBROUTINE BINAVERAGE(TD,YD,ND,T0,TM,NHBIN,DELTBIN,NSP,NK,BD)

       IMPLICIT NONE
       INTEGER I,J,L,M,ND,NS,NHBIN,NK0,BD(NHBIN),NSP(NHBIN),TOTALN,NK
       INTEGER YY,NN,KURT
       DOUBLE PRECISION TD(ND),YD(ND),YDEN(NHBIN),YBINAVE(NHBIN)
       DOUBLE PRECISION FL,SL,TOTALY,YAVE,STEP,YMAX,YMIN,YDIFF,YDIF
       DOUBLE PRECISION DELTA,T0,TM,DELBIN,YBDIFF(NHBIN)
       DOUBLE PRECISION DELTBIN(NHBIN),KNOTP,YSEG

       TOTALN=0
       TOTALY=0.D0
       YAVE=0.D0
       YMAX=0.D0
       YMIN=99.D0

       DELBIN=(TM-T0)/DBLE(NHBIN)
       FL=T0
       SL=T0+DELBIN

       NK0=1 ! OFFSET
       DO I=1,ND
```

```
IF (YD(I) .LT. YMIN) YMIN=YD(I)
IF (YD(I) .GT. YMAX) YMAX=YD(I)
ENDDO
YDIFF=YMAX-YMIN
YSEG=YDIFF/5.D0 ! DIVIDING Y VALUES INTO 5
DO J=1,NHBIN
BD(J)=0
YDEN(J)=0.D0
YBINAVE(J)=0.D0
YBDIFF(J)=0.D0
NSP(J)=0.D0
DO I=1,ND
IF ((TD(I).GT.FL) .AND. (TD(I).LE.SL)) THEN
BD(J)=BD(J)+1
YDEN(J)=YDEN(J)+YD(I)
ENDIF
ENDDO
FL=SL
SL=SL+DELBIN
ENDDO

BD(1)=BD(1)+1 ! COMPENSATING FIRST DATA
YDEN(1)=YDEN(1)+YD(1)

DO J=1,NHBIN
IF (BD(J) .NE. 0) YBINAVE(J)=YDEN(J)/DBLE(BD(J))
TOTALY=TOTALY+YDEN(J)
TOTALN=TOTALN+BD(J)
ENDDO

YAVE=TOTALY/TOTALN
DO J=1,NHBIN
YBDIFF(J)=DABS(YBINAVE(J)-YAVE)*200
NSP(J)=INT(DABS(YBDIFF(J)))

IF(YBDIFF(J).GT.NINT(10.D0*YSEG)) NSP(J)=2
IF(YBDIFF(J).GT.NINT(20.D0*YSEG)) NSP(J)=2
IF(YBDIFF(J).GT.NINT(30.D0*YSEG)) NSP(J)=4
IF(YBDIFF(J).GT.NINT(40.D0*YSEG)) NSP(J)=4
IF(YBDIFF(J).GT.NINT(50.D0*YSEG)) NSP(J)=4

IF (NSP(J) .EQ. 0) NSP(J)=1
IF (NSP(J) .NE. 0)  DELTBIN(J)=DELBIN/DBLE(NSP(J))
ENDDO

NK=4
```

```
      DO I=1,NHBIN
      DO J=1,NSP(I)
      NK=NK+1
      ENDDO
      ENDDO
      NK=NK+3

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE KURTOSIS(TD,YD,ND,NHBIN,DELTBIN,NSP,NK,BD)

      IMPLICIT NONE
      INTEGER I,J,ND,ST,EN,NHBIN,NSP(NHBIN),BD(NHBIN),NK
      DOUBLE PRECISION TD(ND),YD(ND),SUM2,MEAN,KURT,T0,TM,DELBIN,FL,SL
      DOUBLE PRECISION DELTBIN(NHBIN),VARIA,SD,DIFF

      T0=TD(1)
      TM=TD(ND)
      DELBIN=(TM-T0)/DBLE(NHBIN)
      FL=T0
      SL=T0+DELBIN
      EN=0

      DO J=1,NHBIN
      BD(J)=0
      DO I=1,ND
      IF ((TD(I).GT.FL) .AND. (TD(I).LE.SL)) BD(J)=BD(J)+1
      ENDDO
      NSP(J)=1
      FL=SL
      SL=SL+DELBIN
      ENDDO
      BD(1)=BD(1)+1

      DO J=1,NHBIN
      SUM2=0.D0
      MEAN=0.D0
      VARIA=0.D0
      KURT=0.D0
      SD=0.D0

      ST=EN+1
      EN=BD(J)+ST-1
```

```
        DO I=ST,EN
        SUM2 = SUM2 + YD(I)
        ENDDO
        MEAN = SUM2/DBLE(EN-ST+1)
        DO I=ST,EN
        DIFF= YD(I) - MEAN
        VARIA=VARIA + DIFF**2
        KURT= KURT + DIFF**4
        ENDDO
        VARIA=VARIA/DBLE(EN-ST)
        SD=DSQRT(VARIA)
        KURT= KURT/ DBLE(EN-ST)
        KURT=KURT/(SD**4) - 3.D0

      IF (KURT .GT. 0.D0 ) NSP(J)=4
      IF (KURT .GT. 1.D0)  NSP(J)=3
      IF (KURT .GT. 2.D0 ) NSP(J)=1
      IF (KURT .GT. 3.D0 ) NSP(J)=1
      IF (KURT .LE. 0.D0)  NSP(J)=4
      IF (KURT .LT. -1.D0) NSP(J)=2
      IF (KURT .LT. -2.D0) NSP(J)=1
      IF (KURT .LT. -3.D0) NSP(J)=1
      IF (NSP(J) .NE. 0.D0)  DELTBIN(J)=DELBIN/DBLE(NSP(J))
      ENDDO

      NK=4
      DO J=1,NHBIN
      DO I=1,NSP(J)
      NK=NK+1
      ENDDO
      ENDDO
      NK=NK+3
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE SUBRMS(TD,YD,ND,T0,TM,NS,NKC,NK,NW,NHBIN,DEL,
     *PERIOD,NSP,BD)

      IMPLICIT NONE
      INTEGER ND,NS,NKC,NK,NW,NG,I,J,M,L,NHBIN,NSP(NHBIN)
      INTEGER NPLOT,BD(NHBIN)
      DOUBLE PRECISION TD(ND),YD(ND),T0,TM,TT(ND),YY(ND)
      DOUBLE PRECISION CS(NKC,NK),TK(NK),A(NS,NS),C(NS)
      DOUBLE PRECISION SPLINT,LSQ,INTFUN,d,v,DEL(NHBIN)
```

```
      DOUBLE PRECISION T,TSTSUM,PERIOD,DELBIN,DELPLT,LN
      OPEN (unit = 19,file="splineout.txt",status='unknown')

      LSQ = 0.D0
      LN = 0.D0
      CALL KNOT(T0,TM,TK,TD,YD,ND,NK,NHBIN,NSP,DEL,BD)
      CALL SPLCO(TK,CS,NS,NKC,NK)
      CALL SPLMAT(YD,TD,ND,A,C,NS,CS,TK,NKC,NK)
      CALL LINEQ(A,C,NS,NW)
      CALL LESQR(TD,YD,ND,C,NS,CS,TK,NKC,NK,LSQ)

10    FORMAT(F14.6,1X,F14.6)
      NPLOT=NK
      IF (NPLOT .LT. 10) NPLOT=10
      DELPLT=(TM-T0)/DBLE(NPLOT)
      DO I=1,NPLOT+1
      T=T0+DELPLT*DBLE(I-1)

      TSTSUM=INTFUN(T,C,NS,CS,TK,NKC,NK)
      WRITE(19,20)T,TSTSUM
      ENDDO
      NG=NK/3
!     CALL LENGTH(C,NS,CS,TK,NKC,NK,NG,LN)
      WRITE(6,20)PERIOD,LSQ

20    FORMAT(F14.6,1X,F14.6)
      CLOSE(19)
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE KNOT(T0,TM,TK,TD,YD,ND,NK,NHBIN,NSP,DELTBIN,BD)

      IMPLICIT NONE
      INTEGER NK,ND,I,J,K,L,U,LT,NHBIN,NSP(NHBIN),NKIJ,BD(NHBIN)
      DOUBLE PRECISION T0,TM,DELBIN,DELTBIN(NHBIN)
      DOUBLE PRECISION TK(NK),TD(ND),YD(ND),TEMP

      DELBIN=(TM-T0)/DBLE(NHBIN)
      NKIJ=4
      DO I=1,NHBIN
      DO J=1,NSP(I)
      NKIJ=NKIJ+1
      TK(NKIJ)=T0+DELBIN*DBLE(I-1)+DELTBIN(I)*DBLE(J)
      ENDDO
```

```
      ENDDO

      TK(1)=T0-TK(7)
      TK(2)=T0-TK(6)
      TK(3)=T0-TK(5)
      TK(4)=T0
      TK(NK-2)=TK(NK-3)+DELTBIN(NHBIN)
      TK(NK-1)=TK(NK-3)+2.D0*DELTBIN(NHBIN)
      TK(NK)=TK(NK-3)+3.D0*DELTBIN(NHBIN)
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE SPLCO(TK,CS,NS,NKC,NK)

      IMPLICIT NONE
      INTEGER NS,NKC,NK,I,J,K
      DOUBLE PRECISION FA,PRD
      DOUBLE PRECISION CS(NKC,NK),TK(NK)
      DO  I=1,NS
      DO  J=1,5
      PRD=1.D0
      DO  K=1,5
      IF((6-K).EQ.J)FA=1.D0
      IF((6-K).NE.J)FA=TK(I+5-K)-TK(I+J-1)
      PRD=PRD/FA
      ENDDO
      CS(J,I)=PRD
      ENDDO
      ENDDO
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DOUBLE PRECISION FUNCTION SPLINT(I,T,CS,TK,NKC,NK)

      IMPLICIT NONE
      INTEGER NKC,NK,I
      DOUBLE PRECISION T,FA
      DOUBLE PRECISION CS(NKC,NK),TK(NK)
      IF((T.LT.TK(I)).OR.(T.GE.TK(I+4)))FA=0.D0
      IF((TK(I).LE.T).AND.(T.LT.TK(I+1)))FA=-CS(1,I)*(TK(I)-T)**3

      IF((TK(I+1).LE.T).AND.(T.LT.TK(I+2)))
```

```
     * FA=-CS(1,I)*(TK(I)-T)**3-CS(2,I)*(TK(I+1)-T)**3

      IF((TK(I+2).LE.T).AND.(T.LT.TK(I+3)))
     * FA=CS(4,I)*(TK(I+3)-T)**3+CS(5,I)*(TK(I+4)-T)**3

      IF((TK(I+3).LE.T).AND.(T.LT.TK(I+4)))FA=CS(5,I)*(TK(I+4)-T)**3

      SPLINT=(TK(I+4)-TK(I))*FA
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DOUBLE PRECISION FUNCTION DSPLIN(I,T,CS,TK,NKC,NK)

      IMPLICIT NONE
      INTEGER NKC,NK,I
      DOUBLE PRECISION T,FA
      DOUBLE PRECISION CS(NKC,NK),TK(NK)
      IF((T.LT.TK(I)).OR.(T.GE.TK(I+4)))FA=0.D0
      IF((TK(I).LE.T).AND.(T.LT.TK(I+1)))FA=3.D0*CS(1,I)*(TK(I)-T)**2

      IF((TK(I+1).LE.T).AND.(T.LT.TK(I+2)))
     * FA=3.D0*CS(1,I)*(TK(I)-T)**2+3.D0*CS(2,I)*(TK(I+1)-T)**2

      IF((TK(I+2).LE.T).AND.(T.LT.TK(I+3)))
     * FA=-3.D0*CS(4,I)*(TK(I+3)-T)**2 -3.D0*CS(5,I)*(TK(I+4)-T)**2

      IF((TK(I+3).LE.T).AND.(T.LT.TK(I+4)))
     * FA=-3.D0*CS(5,I)*(TK(I+4)-T)**2

      DSPLIN=(TK(I+4)-TK(I))*FA
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE SPLMAT(YD,TD,ND,A,C,NS,CS,TK,NKC,NK)

      IMPLICIT NONE
      INTEGER ND,NS,NKC,NK,BETA,NU,I
      DOUBLE PRECISION YD(ND),TD(ND),A(NS,NS),C(NS),CS(NKC,NK),TK(NK)
      DOUBLE PRECISION SPLINT

      DO BETA=1,NS
      C(BETA)=0.D0
```

```
        DO I=1,ND
        C(BETA)=C(BETA)+YD(I)*SPLINT(BETA,TD(I),CS,TK,NKC,NK)
        ENDDO
      DO NU=1,BETA
      A(BETA,NU)=0.D0
         DO I=1,ND
         A(BETA,NU)=A(BETA,NU)+SPLINT(BETA,TD(I),CS,TK,NKC,NK)*
     $   SPLINT(NU,TD(I),CS,TK,NKC,NK)
         ENDDO
      A(NU,BETA)=A(BETA,NU)
      ENDDO
      ENDDO
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE LINEQ(A,C,N,NW)

      IMPLICIT NONE
      INTEGER N,NW,INFO
      INTEGER IPIV(N)
      DOUBLE PRECISION A(N,N),C(N),WORK(NW)
      CALL DSYSV( 'U',N,1,A,N,IPIV,C,N,WORK,NW,INFO)
!      WRITE(*,1004)INFO
1004  FORMAT(3X,'LINEQ INFO=',I3)
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE LESQR(TD,YD,ND,C,NS,CS,TK,NKC,NK,LSQ)

      IMPLICIT NONE
      INTEGER ND,NS,NKC,NK,I
      DOUBLE PRECISION TD(ND),YD(ND),C(NS),CS(NKC,NK),TK(NK),LSQ
      DOUBLE PRECISION INTFUN,YID(ND),SD,DIFF
      OPEN (unit = 29,file="splineout.txt",status='unknown')
      LSQ=0.D0
      CALL Stat_SD(YD,ND,SD)

      DO I=1,ND
      YID(I)=INTFUN(TD(I),C,NS,CS,TK,NKC,NK)
      ENDDO

      DO I=1,ND
```

```
      WRITE(29,20)TD(I),YID(I)
      ENDDO
20    FORMAT(F14.6,1X,F14.6)

      DO I=1,ND
      WRITE(31,20) YD(I),YID(I),YD(I)-YID(I)
      LSQ=LSQ+((YD(I)-YID(I))**2)
      ENDDO
      CLOSE(29)
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DOUBLE PRECISION FUNCTION INTFUN(T,C,NS,CS,TK,NKC,NK)

      IMPLICIT NONE
      INTEGER NS,NKC,NK,I
      DOUBLE PRECISION T,C(NK),CS(NKC,NK),TK(NK),SPLINT

      INTFUN=0.D0
      DO I=1,NS
      INTFUN=INTFUN+C(I)*SPLINT(I,T,CS,TK,NKC,NK)
      ENDDO
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DOUBLE PRECISION FUNCTION DINTFU(T,C,NS,CS,TK,NKC,NK)

      IMPLICIT NONE
      INTEGER NS,NKC,NK,I
      DOUBLE PRECISION T,C(NK),CS(NKC,NK),TK(NK),DSPLIN

      DINTFU=0.D0
      DO I=1,NS
      DINTFU=DINTFU+C(I)*DSPLIN(I,T,CS,TK,NKC,NK)
      ENDDO
c      DINTFU=DSINH(T)
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE LENGTH(C,NS,CS,TK,NKC,NK,NG,TLN)
```

```
      IMPLICIT NONE
      INTEGER NS,NKC,NK,NG
      DOUBLE PRECISION C(NK),CS(NKC,NK),TK(NK),TLN
      INTEGER I,J
      DOUBLE PRECISION X(NG),W(NG),T,WJ,DYBYDT,DINTFU

      CALL GAUSPT(X,W,NG)
      TLN=0.D0
      DO I=1,NK-7
          DO J=1,NG
          T=.5D0*(TK(4+I)+TK(3+I)+(TK(4+I)-TK(3+I))*X(J))
          WJ=.5D0*(TK(4+I)-TK(3+I))*W(J)
          DYBYDT=DINTFU(T,C,NS,CS,TK,NKC,NK)
          TLN=TLN+DSQRT(1.D0+DYBYDT*DYBYDT)*WJ
          ENDDO
      ENDDO
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE GAUSPT(X,W,N)

      IMPLICIT NONE
      DOUBLE PRECISION PI,Z,P1,P2,P3,DJ,PP,Z1
      INTEGER M,I,J,N
      DOUBLE PRECISION  X(N),W(N)

      PI=4.D0*DATAN(1.D0)
      M=(N+1)/2
      DO I=1,M
       Z=DCOS(PI*(I-.25D0)/(N+.5D0))
       IF(DABS(Z).LE.1.D-15)Z=0.D0
1      CONTINUE
       P1=1.D0
       P2=0.D0
       DO J=1,N
        P3=P2
        P2=P1
        DJ=DFLOAT(J)
        P1=((2.D0*DJ-1.D0)*Z*P2-(DJ-1.D0)*P3)/DJ
       ENDDO
       PP= DFLOAT(N)*(Z*P1-P2)/(Z*Z-1.D0)
       Z1=Z
       Z=Z1-P1/PP
```

```
        IF(DABS(Z-Z1).GT.5.D-15)GOTO 1
        X(I)=-Z
        X(N+1-I)=Z
        W(I)=2.D0/(1.D0-Z*Z)/PP/PP
        W(N+1-I)=W(I)
       ENDDO
       RETURN
       END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE Stat_SD(y,N,sd)

        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION y(N),cvar,sd

        sd = 0.d0
        CALL Stat_Variance(y,N,cvar)
        sd = DSQRT(cvar)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE Stat_Variance(y,N,bvar)

        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION y(N),b(N),bbar,bvar,bi2,mean

        bi2 = 0.d0
        bbar = 0.d0
        bvar = 0.d0
        mean = 0.d0

        CALL Stat_Mean(y,N,bbar)
        DO i = 1,N
        b(i) = y(i)
        b(i) = b(i)-bbar
        bi2 = bi2+b(i)*b(i)
        ENDDO
        bvar = bi2/DBLE(N)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
        SUBROUTINE Stat_Mean(y,N,ybar)

        IMPLICIT NONE
        INTEGER N,i
        DOUBLE PRECISION y(N),ybar

C       Finding Mean of y
        ybar = 0.d0
        DO i = 1,N
        ybar = ybar+y(i)
        ENDDO
        ybar = ybar/DBLE(N)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

From here use any Linear algebra routine to solve A * X = B
for example, we have used the following from LAPACK

CALL DSYSV( 'U',N,1,A,N,IPIV,C,N,WORK,NW,INFO)
*> DSYSV computes the solution to a real system of linear equations
*>    A * X = B,
*> where A is an N-by-N symmetric matrix and X and B are N-by-NRHS
*> matrices.
*>
*> The diagonal pivoting method is used to factor A as
*>    A = U * D * U**T,  if UPLO = 'U', or
*>    A = L * D * L**T,  if UPLO = 'L',
*> where U (or L) is a product of permutation and unit upper (lower)
*> triangular matrices, and D is symmetric and block diagonal with
*> 1-by-1 and 2-by-2 diagonal blocks.  The factored form of A is then
*> used to solve the system of equations A * X = B.
*> \endverbatim
*
*  Arguments:
*  ==========
*
*> \param[in] UPLO
*> \verbatim
*>          UPLO is CHARACTER*1
*>          = 'U':  Upper triangle of A is stored;
*>          = 'L':  Lower triangle of A is stored.
*> \endverbatim
*>
*> \param[in] N
*> \verbatim
```

```
*>          N is INTEGER
*>          The number of linear equations, i.e., the order of the
*>          matrix A.  N >= 0.
*> \endverbatim
*>
*> \param[in] NRHS
*> \verbatim
*>          NRHS is INTEGER
*>          The number of right hand sides, i.e., the number of columns
*>          of the matrix B.  NRHS >= 0.
*> \endverbatim
*>
*> \param[in,out] A
*> \verbatim
*>A is DOUBLE PRECISION array, dimension (LDA,N)
*>On entry, the symmetric matrix A.  If UPLO = 'U', the leading
*>N-by-N upper triangular part of A contains the upper
*>triangular part of the matrix A, and the strictly lower
*>triangular part of A is not referenced.  If UPLO = 'L', the
*>leading N-by-N lower triangular part of A contains the lower
*>triangular part of the matrix A, and the strictly upper
*>triangular part of A is not referenced.
*>
*>On exit, if INFO = 0, the block diagonal matrix D and the
*>multipliers used to obtain the factor U or L from the
*>factorization A = U*D*U**T or A = L*D*L**T as computed by
*>DSYTRF.
*> \endverbatim
*>
*> \param[in] LDA
*> \verbatim
*>          LDA is INTEGER
*>The leading dimension of the array A.  LDA >= max(1,N).
*> \endverbatim
*>
*> \param[out] IPIV
*> \verbatim
*>IPIV is INTEGER array, dimension (N)
*>Details of the interchanges and the block structure of D, as
*>determined by DSYTRF.  If IPIV(k) > 0, then rows and columns
*>k and IPIV(k) were interchanged, and D(k,k) is a 1-by-1
*>diagonal block.  If UPLO = 'U' and IPIV(k) = IPIV(k-1) < 0,
*>then rows and columns k-1 and -IPIV(k) were interchanged and
*>D(k-1:k,k-1:k) is a 2-by-2 diagonal block.  If UPLO = 'L' and
*>IPIV(k) = IPIV(k+1) < 0, then rows and columns k+1 and
*>-IPIV(k) were interchanged and D(k:k+1,k:k+1) is a 2-by-2
```

```
*>diagonal block.
*> \endverbatim
*>
*> \param[in,out] B
*> \verbatim
*>B is DOUBLE PRECISION array, dimension (LDB,NRHS)
*>On entry, the N-by-NRHS right hand side matrix B.
*>On exit, if INFO = 0, the N-by-NRHS solution matrix X.
*> \endverbatim
*>
*> \param[in] LDB
*> \verbatim
*>LDB is INTEGER
*>The leading dimension of the array B.  LDB >= max(1,N).
*> \endverbatim
*>
*> \param[out] WORK
*> \verbatim
*>WORK is DOUBLE PRECISION array, dimension (MAX(1,LWORK))
*>On exit, if INFO = 0, WORK(1) returns the optimal LWORK.
*> \endverbatim
*>
*> \param[in] LWORK
*> \verbatim
*>LWORK is INTEGER
*>The length of WORK.  LWORK >= 1, and for best performance
*>LWORK >= max(1,N*NB), where NB is the optimal blocksize for
*>DSYTRF.
*>for LWORK < N, TRS will be done with Level BLAS 2
*>for LWORK >= N, TRS will be done with Level BLAS 3
*>
*>If LWORK = -1, then a workspace query is assumed; the routine
*>only calculates the optimal size of the WORK array, returns
*>this value as the first entry of the WORK array, and no error
*>message related to LWORK is issued by XERBLA.
*> \endverbatim
*>
*> \param[out] INFO
*> \verbatim
*>INFO is INTEGER
*>= 0: successful exit
*>< 0: if INFO = -i, the i-th argument had an illegal value
*>> 0: if INFO = i, D(i,i) is exactly zero.  The factorization
*>     has been completed, but the block diagonal matrix D is
*>     exactly singular, so the solution could not be computed.
*> \endverbatim
```

```
*
*  Authors:
*  ========
*
*> \author Univ. of Tennessee
*> \author Univ. of California Berkeley
*> \author Univ. of Colorado Denver
*> \author NAG Ltd.
*
*> \date November 2011
*
*> \ingroup doubleSYsolve
*
*  =====================================================================
      SUBROUTINE DSYSV( UPLO, N, NRHS, A, LDA, IPIV, B, LDB, WORK,
     $        LWORK, INFO )
*
*  -- LAPACK driver routine (version 3.4.0) --
*  -- LAPACK is a software package provided by Univ. of Tennessee,    --
*  -- Univ. of California Berkeley, Univ. of Colorado Denver and NAG Ltd..--
*     November 2011


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

B.2 Phase folding program

Program for phase folding on given period.

```
C This is for phase folding, save as phase.for
C Compile as gfortran -o phase phase.for and put in same folder
C The file 'forin.dat' is created by the script.
        PROGRAM MAIN

        IMPLICIT NONE
        INTEGER line
        DOUBLE PRECISION period
        CHARACTER *13 fname

        OPEN(unit = 17,file = 'forin.dat',status = 'unknown')
        READ(17,*),line,period,fname
        CALL firstfn(line,period)
        CLOSE(17)
        STOP
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE firstfn(N,period)

        IMPLICIT NONE
        INTEGER N
        DOUBLE PRECISION YD(N),TD(N),period

        CALL fileread(TD,YD,N)
        CALL phasefold(TD,YD,N,period)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE fileread(t,y,N)

        IMPLICIT NONE
        INTEGER N,i,IO_ERR/0/
        CHARACTER name*60,line*1024
        DOUBLE PRECISION y(N),t(N),err(N),tspan
        DOUBLE PRECISION tstart,tmin,tmax

        tspan = 0.d0
        tmin = 0.d0
        tmax = 0.d0
        tstart = 0.d0
   69   OPEN (unit = 18,  file = "input.txt",   status = 'old' )
        DO i = 1,N
   74   READ (18,'(a)',END = 73) line    ! reading line from datafile
        IF (index(line,'#').GT.0) GOTO 74! skip line if comment line found
        IF (lnblnk(line).EQ.0) GOTO 74   ! skip if blank line found
        READ(line,*,IOSTAT = IO_ERR) t(i),y(i),err(i) ! here actual READing
        IF (IO_ERR.NE.0 ) THEN
        READ(line,*,IOSTAT = IO_ERR) t(i),y(i)
        IF (IO_ERR.NE.0 ) THEN
        WRITE(6,*) "Something is wrong with input data file !!!"
        WRITE(6,*) " Please check the format of file"
        IF (err(i).EQ. 0.) IO_ERR = -2
        CLOSE(18)
        GOTO 67   !  unexpected termination
        ENDIF
        ENDIF
        ENDDO
 73     CLOSE (18)
        tmin = t(1)
```

```
      tmax = t(N)
      tspan = tmax-tmin
 67   END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE phasefold(t,y,N,period)

      IMPLICIT NONE
      INTEGER N,i,j,bin(10)
      DOUBLE PRECISION y(N),t(N),ph(N),period,bb

      OPEN(unit = 13,file = 'phase.txt',status = 'unknown')

      DO i = 1,N
      ph(i) = ((t(i)-t(1))/period)-INT((t(i)-t(1))/period)
      WRITE(13,900),ph(i),y(i)
      ENDDO
900   FORMAT(F24.18,1X,F24.18)
      CLOSE(13)
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE  Stat_ZeroMean(y,N)

      IMPLICIT NONE
      INTEGER N,i
      DOUBLE PRECISION y(N),ymean

      ymean = 0.d0
      CALL Stat_Mean(y,N,ymean)
!     WRITE(*,*)'Value of ymean ',ymean
      DO i = 1,N
      y(i) = y(i)-ymean
      ENDDO
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE Stat_Mean(y,N,ybar)

      IMPLICIT NONE
      INTEGER N,i
      DOUBLE PRECISION y(N),ybar
```

```
C       Finding Mean of y
        ybar = 0.d0
        DO i = 1,N
        ybar = ybar+y(i)
        ENDDO
        ybar = ybar/DBLE(N)
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

Appendix C

Bash scripts used for automation

C.1 startsigspecf.sh

Script for automated running sigspecf.for

```
#!/bin/bash
 echo
 echo '******************'
 echo 'Running SigSpec-F'
 echo '******************'
rm -f forin.dat
touch forin.dat
START=$(date +%s)
echo 'Starting time :' $(date +%T)
date -R
count=$(ls *.lc | wc -l)
#searches for files with extension .lc recursively
for file in $(find $temp -type f -iname '*.lc' | sort -n); do
before="$(date +%s)"
 echo 'processing' $file 'please wait ....'
rm -f forin.dat
#creates a folder with data file name for storing
#the output and removes the extension lc
 dire=${file/.lc/}
 #mv $file ${file/.lc/}
 rm -rf $dire
 mkdir $dire
 sort -n $dire.lc -k1 -o $dire.dat
ln=`wc -l "$dire.lc" | awk '{print $1'}`
nn=0;mm=0;t1=0;t2=0;line=1
  IMM=`sed -n $line'p' $dire.dat`
  set -- "$IMM"
```

```
  IFS=" "; declare -a datas=($*)
  echo "initial time is = ${datas[$nn]}"
t1=${datas[$nn]}
  INN=`sed -n $ln'p' $dire.dat`
  set -- "$INN"
  IFS=" "; declare -a datas1=($*)
  echo "final time is = ${datas1[$mm]}"
t2=${datas1[$mm]}
lf=$( echo "scale=4; 10/($t2-$t1)" | bc -l)
uf=$( echo "scale=4; 1000*$lf" | bc -l)
###############################
### if there is + or - sign in the filename.
if [[ "$dire" =~ "+" ]]
then
cp "$file" "${dire%%+*}"
ra=${dire%%+*}
dec=${dire#*+}
fi
if [[ "$dire" =~ "-" ]]
then
cp "$file" "${dire%%-*}"
ra=${dire%%-*}
dec=${dire#*-}
fi
ra=`echo $ra | sed -e 's,./\(.*\)$,\1,g'`
echo " ra = $ra"
echo "dec = $dec"
####################################
# sorts the file on the basis of Julian
#date, first column k1
 sort -n $file -k1 -o $dire.dat
 line_num=`wc -l $dire.dat`
 line_num=${line_num%%$dire.dat}
 cp $dire.dat input.txt
echo "$line_num" >> forin.dat
# runs the main sigspec program
./sigspecf
echo
mv out.txt $dire
mv harm.dat $dire
mv spectrum*.dat $dire
mv rms.dat $dire
mv result.txt $dire
#mv phase.txt $dire
rm -f $ra
rm -f $dire.dat
```

```
rm -f forin.dat
rm -f input.txt
let count=$count-1
echo "remaining $count data files"
after="$(date +%s)"
elapsed_seconds="$(expr $after - $before)"
echo 'Time taken for '$dire' file:' $elapsed_seconds 'Seconds'
echo
done #outer loop closes
echo 'Ending Time' $(date +%T)
date -R
END=$(date +%s)
DIFF=$(( $END - $START ))
echo
echo "Total Time Taken $DIFF Seconds"
```

C.2 sigfoneperiodonly.sh

Script for selecting the best period from the output produced by sigspecf.for. Also removes 1 cycles/day alias.

```
#!/bin/bash
# this script reads periods from  file, does not run sigspecf
rm -f result_final.dat
touch result_final.dat
#searches for files with extension .lc recursively
for file in $(find $temp -type f -iname '*.lc'); do
 echo 'processing' $file 'please wait ..'
#removes the extension lc
dire=${file/.lc/}
n=0; a=2; line=1
# removing alias frequecies between cond1 and cond2
cond1=0.99
cond2=1.01
while read inputline
do
 #read periods from file
  IN=`sed -n $line'p' $dire/result.txt`
  set -- "$IN"
  IFS=" "; declare -a results=($*)
  echo "frequency is = ${results[$n]}"
  fname=`echo $dire | sed -e 's,./\(.*\)$,\1,g'`
if [ $(echo "${results[$n]} < $cond1 "|bc) -eq 1 ]
|| [ $(echo " ${results[$n]} > $cond2"|bc) -eq 1 ]; then
period=$(echo "scale=10; 1/${results[$n]}" | bc -l)
  echo "period is = $period"
```

```
 echo "$fname $period ${results[$n]}" >> result_final.dat
 break
fi
 let line=$line+1
done < $dire/result.txt #inner loop closes
done #outer loop closes
```

C.3 startspline.sh

Bash script for automated running spline.for

```
# change the name of file result-compared.dat to
#result-compared-different.dat,
# with only the periods to be checked.
#!/bin/bash
# this script is for checking the output of
#SigSpec/pdm/GLS by spline
rm -f result_final_sorted.dat
touch result_final_sorted.dat
count=$(ls *.lc | wc -l)
# input file name
filee=result-compared-different.dat
f=0
p=1
n=2
line=1
while read inputline
do
rm -f forin.dat
touch -f forin.dat
rm -f input.txt
rm -f res_spline.dat
rm -f result_spline.dat
 #read periods from file
  IN=`sed -n $line'p' $filee`
  set -- "$IN"
  IFS=" "; declare -a array=($*)
  fname=${array[$f]}
  echo "file name is = $fname.lc"
  pp=$(echo "scale=6; ${array[$p]}" | bc -l)
  echo "published period is = $pp"
  np=$(echo "scale=6; ${array[$n]}" | bc -l)
  echo "New period is = $np"
  echo 'processing' $fname 'please wait ..'
  touch forin.dat
  touch res_spline.dat
```

```
  cp $fname.lc input.txt
  sort -n input.txt -k1 -o input.txt
  line_num=`wc -l input.txt`
  line_num=${line_num%%input.txt}
  rm -f $fname/*.dat
  rm -f $fname/*.ps
  echo "$line_num $pp $fname" > forin.dat
  ./phase
  sort -k1 -n phase.txt -o phase.txt
  ./spline >> res_spline.dat
  mv splineout.txt splineoutu.txt
  mv phase.txt pphase.txt
  mv forin.dat forinu.dat
  touch forin.dat
  echo "$line_num $np $fname" > forin.dat
  ./phase
  sort -k1 -n phase.txt -o phase.txt
  ./spline >> res_spline.dat
  mv splineout.txt splineoutm.txt
  mv phase.txt mphase.txt
  mv forin.dat forinm.dat
  cp pphase.txt $fname/ph-"$pp".dat
  cp mphase.txt $fname/ph-"$np".dat
  mv splineoutu.txt $fname/sp-"$pp".dat
  mv splineoutm.txt $fname/sp-"$np".dat
  sort -k2 -n res_spline.dat -o result_spline.dat
  p=1
  nn=0
  linee=1
  IN=`sed -n $linee'p' ./result_spline.dat`
  set -- "$IN"
  IFS=" "; declare -a true_per=($*)
  echo "True period is = ${true_per[$nn]}"
  freq=$(echo "scale=6; 1/${true_per[$nn]}" | bc -l)
  fname=`echo $fname | sed -e 's,./\(.*\)$,\1,g'`
 echo "$fname ${true_per[$nn]} $freq" >> result_final_sorted.dat
 mv res_spline.dat $fname
 mv result_spline.dat $fname
 rm -f input.txt
 let count=$count-1
 echo
 echo "remaining $count files"
 let  line=$line+1
 done < $filee #outer loop closes
```

C.4 Data file cleaner script

Script for cleaning data files by removing comment lines starting with # and for replacing 'nan' in the error column, if any.

```
#!/bin/sh
# extract and replaces all gzip files to the same folder
for file in $(find $temp -type f -iname '*.lc.gz' | sort -n); do
gunzip -r $file
done
rm -f flist_tredu.txt
rm -f flist_nan.txt
# removes the comment lines beginning with #
for file in $(find $temp -type f -iname '*.lc'| sort -n); do
rm -f tempnew
rm -f tempnews
sort -k1 -o $file $file
line_num=`wc -l $file`
line_num=${line_num%%$file}
echo "line number = $line_num"
echo " removing comments from $file. wait please... "
while read line
do
case $line in
\#*)
continue
;;
esac
# set the SECTION
case $line in
SECTION*)
hdr="$line"
continue
;;
esac
pr="$hdr"
pr="$pr  $line"
echo "$pr" >> tempnew
done < $file
cp tempnew $file
n=0; a=1;line=1;e=2;linee=2
while read inputline
do
 #read periods from file
  IN=`sed -n $line'p' $file`
  set -- "$IN"
```

```
  IFS=" "; declare -a results=($*)
#  echo "time is = ${results[$n]}"
#  fname=`echo $dire | sed -e 's,./\(.*\)$,\1,g'`
  IM=`sed -n $linee'p' $file`
  set -- "$IM"
  IFS=" "; declare -a resultss=($*)
#  echo "time is = ${resultss[$n]}"
#removes nan from error column if any
if [ $(echo "${results[$e]} == "nan"" |bc) -eq 1 ]; then
results[$e]=0.001 # substitute this value
echo "$file">> flist_nan.txt
fi
if [ $(echo "${results[$n]} != ${resultss[$n]}"|bc) -eq 1 ]; then
if [ $(echo "$line < $line_num"|bc) -eq 1 ]; then
#echo "line = $line"
# star_ID  period  frequency
echo "${results[$n]} ${results[$a]} ${results[$e]}" >> tempnews
 fi
 fi
 let line=$line+1
 let linee=$linee+1
done < $file #inner loop closes
#simply flushes the last line, since it skips
 echo "${results[$n]} ${results[$a]} ${results[$e]}" >> tempnews
cp tempnews $file
line_numb=`wc -l $file`
line_numb=${line_numb%%$file}
echo "new line number = $line_numb"
if [ $(echo "$line_num != $line_numb"|bc) -eq 1 ]; then
echo "$file">> flist_tredu.txt
fi
done
rm -f tempnew
rm -f tempnews
```

Bibliography

[1] Akerlof C. et. al., *Application of cubic splines to the spectral analysis of unequally spaced data*, ApJ, **436**, 787, (1994).

[2] Akerlof C. et. al., *ROTSE All-Sky Surveys for Variable Stars. I. Test Fields*, AJ, **119**, 1901, (2000).

[3] Alan V. Oppenheim et. al. *Discrete-Time Signal Processing*, Pearson Education, (2009).

[4] Baluev R.V., *Assessing statistical significance of periodogram peaks*, MNRAS, **385**.1279B, (2008).

[5] Baluev R.V., *Detecting multiple periodicities in observational data with the multi-frequency periodogram. II. Frequency Decomposer, a parallelized time-series analysis algorithm*, arXiv1309.0100B, (2013).

[6] Baluev R.V., *Detecting multiple periodicities in observational data with the multi-frequency periodogram. I. Analytic assessment of the statistical significance*, MNRAS, **436**, 807B, (2013).

[7] Baluev R.V., *Detecting non-sinusoidal periodicities in observational data: the von Mises periodogram for variable stars and exoplanetary transits*, MNRAS, **431**, 1167B, (2013).

[8] Baluev R.V., *Distinguishing between a true period and its alias, and other tasks of model discrimination*, MNRAS, **422**, 2372B, (2012).

[9] Baluev R.V., *Search for periodicity in the observational data by means of artificial neuron networks*, yscc.conf, **230B**, (2012).

[10] Baluev R. V., MNRAS, *Detecting non-sinusoidal periodicities in observational data using multi-harmonic periodograms*, MNRAS, **395**, 1541B, (2009).

[11] Barning F.J.M., *The numerical analysis of the light-curve of 12 Lacertae*, BAN, **17**, 22, (1963).

[12] Belokurov V., et.al., *Light-curve classification in massive variability surveys - I. Microlensing*, MNRAS, **341**, pp. 13731384, (2003).

[13] Bertin E. & Arnouts, S., *SExtractor: Software for source extraction*, Astronomy & Astrophysics Supplement **317**, 393, (1996).

[14] Blomme J., et.al., *Automated Classification of Variable Stars in the Asteroseismology Program of the Kepler Space Mission.* ApJ, **713**, L204, (2010).

[15] Blomme J. et.al., *Improved methodology for the automated classication of periodic variable stars*, MNRAS **418**, 96, (2011).

[16] Bloom J.S. et.al., *Automating discovery and classification of transients and variables in the synoptic survey era*, arXiv:1104.3142v1, (2011).

[17] Breger M., et. al., *Nonradial Pulsation of the Delta-Scuti Star Bu-Cancri in the Praesepe Cluster* A&A, **271**, 482, (1993).

[18] Brian D. Warner, *A practical guide to light curve photometry and analysis*, Springer, (2006).

[19] Carbonell M., et.al., *Power spectra of gapped time series - A comparison of several methods*, A&A, **264**, 350, (1992).

[20] Clarke D., *String/Rope length methods using the Lafler-Kinman statistic*, A&A, **386**, 763, (2002).

[21] Cumming A., et.al., *The lick planet search: detectability and mass thresholds*, ApJ, **526**, 890, (1999).

[22] David M. Palmer, *A Fast Chi-squared Technique For Period Search of Irregularly Sampled Data*, arXiv:0901.1913v1, (2009).

[23] de Boor C., *A Practical Guide to Splines*, Springer, New York, (1978).

[24] Deb S. et.al., *Light curve analysis of variable stars using Fourier decomposition and principal component analysis* A&A **507**, 1729, (2009).

[25] Deb.S et.al., *Physical parameters of 62 eclipsing binary stars using the ASAS-3 dataI*, MNRAS, **412**, 1787, (2011).

[26] Debosscher J., et.al, *Automated supervised classification of variable stars. I. Methodology*, A&A, **475**, 1159, (2007).

[27] Debosscher J., et al. *Automated supervised classification of variable stars in the CoRoT programme. Method and application to the first four exoplanet fields*, A&A, **506**, 519, (2009).

[28] Deeming T.J., *Fourier analysis of unequally spaced data*, AP&SS, **36**, 137, (1975).

[29] Devor J., *Solutions for 10, 000 Eclipsing Binaries in the Bulge Fields of OGLE II Using DEBiL*, APJ, **628**, 411, (2005).

[30] Drake A.J. et al., *First Results from the Catalina Real-time Transient Survey*, ApJ, **696**, 870, (2009).

[31] Dubath P. et.al., *Random forest automated supervised classification of Hipparcos periodic variable stars*, ApJS, **11**, 216, (2009).

[32] Dworetsky M. M., *A period-finding method for sparse randomly spaced observations or How long is a piece of string?*, MNRAS **203**, 917, (1983).

[33] Eyer L. and C. Blake, *Automated classification of variable stars for All-Sky Automated Survey 1-2 data*. MNRAS, **358**, 30, (2005).

[34] Ferraz-Mello S., *Estimation of periods from unequally spaced observations*, AJ, **86**, 619, (1981).

[35] Frescura F. A. M. et.al., *Significance tests for periodogram peaks.* Arxiv preprint, 0706.2225, (2007).

[36] Graham M.J., et.al., *A comparison of period finding algorithms*, accepted MNRAS, arxiv.org/abs/1307.2209, (2013).

[37] Graham M.J., et.al., *Using conditional entropy to identify periodicity*, 2013, accepted MNRAS, arxiv.org/abs/1306.6664, (2013).

[38] Hall P. et.al., Using the periodogram to estimate period in nonparametric regression, Biometrika, **93(2)**, 411, (2006).

[39] Heck A. et.al., *On period determination methods*, AASS, **59**, 63, (1985).

[40] Horne J.H., & Baliunas S.L., *A prescription for period analysis of unevenly sampled time series*, ApJ, **302**, 757, (1986).

[41] Pablo Huijse et.al. *Period Estimation in Astronomical Time Series Using Slotted Correntropy*, IEEE signal processing letters, **18**,6,(2011)

[42] Jaan Pelt, *Astronomical Time Series Analysis : Lecture notes.*

[43] John R. Percy, *Understanding Variable stars*, Cambridge University Press, (2007).

[44] Kim Dae-Won et.al., *De-Trending time series for astronomical variability surveys*, MNRAS, **397**, 558, (2009).

[45] Kovacs G., et.al., *A trend filtering algorithm for wide-field variability surveys*, MNRAS, **356**, pp. 557567, (2005).

[46] Lafler J., & Kinman, T. D., *An RR Lyrae Star Survey with the Lick 20-inch Astrograph II. The Calculation of RR Lyrae Periods by Electronic Computer*, ApJS, **11**, 216, (1965).

[47] Li T. H., *A robust spectral analyzer for one-dimensional and multi-dimensional data analysis*, US Patent Application, 2009/0112954 A1, April (2009).

[48] Li T. H., *A nonlinear method for robust spectral analysis*, IEEE Trans Signal Process **58(5)**, 2466, (2010).

[49] Lomb N. R., *Least-squares frequency analysis of unequally spaced data*, ApSS, **39**, 447, (1976).

[50] Long J. et.al., *Optimizing Automated Classification of Periodic Variable Stars in New Synoptic Surveys* arXiv:1201.4863v2, (2012).

[51] McDonald J., *Periodic smoothing of time series*, SIAM J Sci Stat Comput, 7(2), 665, (1986).

[52] Min-Su Shin, et.al., *Efficient period search for time series photometry*, J. Korean Astronomical Society, **37**, 79, (2004).

[53] Min-Su Shin, et.al., *Detecting variability in massive astronomical time series data I. Application of an infinite Gaussian mixture model*, MNRAS, **400**, 1897, (2009).

[54] Nemec A.F.L., *A test of significance for periods derived using phase dispersion minimization techniques*, AJ, **90**, 2317, (1985).

[55] Oh H. S. et.al., *Period analysis of variable stars by robust smoothing*, J R Stat Soc C **53(1)**, 1530, (2004).

[56] Peter M. Hooper, *Period Analysis of Variable Stars: Temporal Dependence and Local Optima.*

[57] Pojmanski, G., *The All Sky Automated Survey. The Catalog of the Periodic Variable Stars in the Selected Fields*, AcA, **48**, 35, (1998).

[58] Pojmanski, G., *The All Sky Automated Survey. Catalog of about 3800 Variable Stars*, AcA, **50**, 177, (2000).

[59] Press, W. H., et.al., *Numerical Recipes in C, 2nd ed.*(Cambridge: Cambridge Univ. Press), (1992).

[60] Priestly M.B., *Spectral analysis and Time series Vol.1*, Academic press, (1981).

[61] R Development Core Team., *R: a language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, (2007).

[62] Reegen P., *SigSpec - Frequency and phase resolved significance in Fourier space*, A&A, **467**, 1353, (2007).

[63] Reimann J. D., *Frequency Estimation Using Unequally Spaced Astronomical Data*, PhD Thesis, University of California at Berkeley, (1994).

[64] Richards J., et al., *On Machine-learned Classification of Variable Stars with Sparse and Noisy Time-series Data*, ApJ, **733**, 10, (2011).

[65] Richards J., et al., *Construction of a calibrated probabilistic classification catalog: application to 50k variable sources in the all-sky automated survey*, ApJS, **203**, 32, (2012).

[66] Richards J. W., et al., *Active Learning to Overcome Sample Selection Bias: Application to Photometric Variable Star Classification*, ApJ, **744**, 192, (2012).

[67] Roberts D.H., et.al., *Time series analysis with clean I. Derivation of a spectrum*, AJ, **93**, 968, (1987).

[68] Samus N.N., et al. *General Catalog of Variable Stars* (GCVS database, Version (2012).

[69] Sarro L.M., et.al., *Automated supervised classification of variable stars*, A&A, **494**, 739, (2009).

[70] Scargle J. D., *Studies in astronomical time series analysis. II - Statistical aspects of spectral analysis of unevenly spaced data*, ApJ, **263**, 835, (1982).

[71] Schuster A. Terr.Mag. (Now J.G.R.) **3**, 13, (1898).

[72] Schwarzenberg-Czerny A., *Fast and Statistically Optimal Period Search in Uneven Sampled Observations*, ApJ, **460**, L107, (1996).

[73] Schwarzenberg-Czerny A., *The Correct Probability Distribution for the Phase Dispersion Minimization Periodogram*, ApJ, **489**, 941, (1997).

[74] Schwarzenberg-Czerny A., *Period Search in Large Datasets*, Baltic Astronomy, **7**, 43, (1998).

[75] Schwarzenberg-Czerny A., *Optimum Period Search: Quantitative Analysis*, ApJ, **516**, 315, (1999).

[76] Shanti Priya D., K. Sriram, K.Y. Shaju and P. Vivekananda Rao, *Photometric study of hot Contact binaries in SMC*, Bull. Astr. Soc. India (2013) 41, 159172.

[77] Stellingwerf R. F., *Period determination using phase dispersion minimization*, ApJ, **224**, 953, (1978).

[78] Sterken C., et.al., *Light Curves of Variable Stars: A Pictorial Atlas*, Cambridge University Press.

[79] Stetson P.B., *On the Automatic Determination of Light-Curve Parameters for Cepheid Variables*, PASP, **108**, pp. 851, (1996).

[80] Sue Liu Y., *Lecture notes*, University of Edinburgh, (geos.ed.ac.uk/ yliu23/docs/).

[81] Vanicek P., *Further development and properties of the spectral analysis by least squares*, AP&SS, **12**, 10, (1971).

[82] Zechmeister M. et.al., *The generalized Lomb-Scargle Periodogram*, A&A, **496**, 577, (2009).

[83] Zhang Z. et.al., *Robust adaptive Lomb periodogram for time-frequency analysis of signals with sinusoidal and transient components*, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP05), Vol. 4, iv-493, (2005).