A Model-Based Monitoring and Diagnosis System for a Space-Based Astrometry Mission

Aleksei Pavlov[1,*], Sven Helmer[2], and Guido Moerkotte[2]

[1] Max-Planck-Institut für Astronomie, Königstuhl 17, 69117 Heidelberg, Germany
pavlov@mpia.de
[2] Fakultät für Mathematik und Informatik, D7, 27, Universität Mannheim, 68131 Mannheim, Germany
{helmer, moerkotte}@informatik.uni-mannheim.de

Abstract. Space-based astrometry missions can achieve an accuracy that has not been possible before (by ground-based observations). However, in order to guarantee this precision, it is of the utmost importance to check the scientific quality of the data constantly. We present a model-based monitoring system, called Science Quick Look, that is able to carry out the preliminary scientific assessment. We have implemented a prototype of the system and show the results of an evaluation.

1 Introduction

Astrometry is the oldest branch of astronomy and deals with the positions, distances, and motions of stars. Apart from providing a reference frame for the observations of astronomers, astrometrical data is important for navigation and guidance systems, accurate time keeping, and supplying astrophysicists with motion and distance data.

We can employ basic trigonometry to find the distance of a far-away object S, e.g. a star, by determining how S appears to move when observed from the two ends of a baseline perpendicular to a line from the baseline's center point to the object. The largest baseline available when looking at stars is twice the distance from the Earth to the Sun, which is approximately 300 million kilometers (see Figure 1). The apparent movement of a star against background stars (which are so far away that their movement is not detectable) is called its parallax. This is the angle marked p in Figure 1. Angles are measured to a precision of arc-seconds. For example, the parallax of the star nearest to our solar system, Alpha Centauri, is $0.75''$ (three quarters of an arc-second).

The development of better instruments over time has led to a steady increase in accuracy of the obtained data. Important milestones were the development of astrolabes, sextants, telescopes, radio telescopes, and CCD (charge-coupled device) chips. The latest step was the introduction of space-based astrometry via satellites, which eliminated the blurring effects of the atmosphere. Figure 2 shows the precision that was feasible during different times in history (the values for DIVA and GAIA are predicted).

* This work was supported by a grant from the Klaus Tschira Foundation (KTS).

K.V. Andersen, J. Debenham, and R. Wagner (Eds.): DEXA 2005, LNCS 3588, pp. 920–929, 2005.

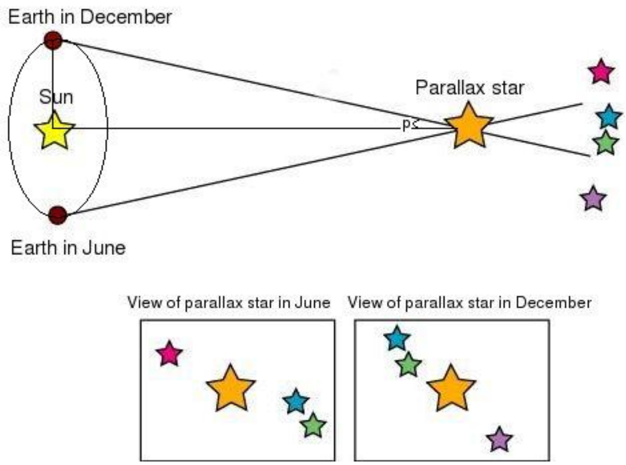


Fig. 1. Parallax of a star

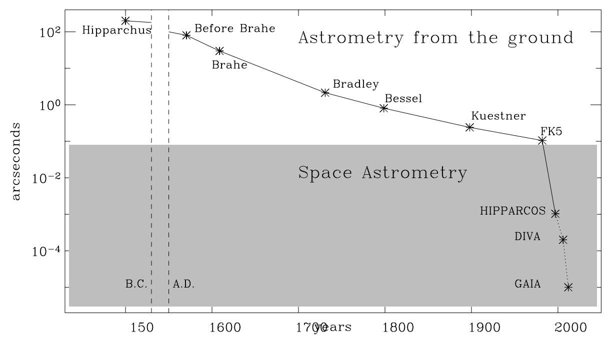


Fig. 2. Precision in arc-seconds over time

The focus of our work was to create a prototype for the monitoring and diagnosis process to assist an operator/astronomer in rapidly assessing the scientific data quality. On the one hand, we needed to generate meaningful statistics and diagnostics that reflect any changes in the system state. On the other hand, we wanted to provide expert advice on possible explanations for any problems. As C.A. Kitts notes in [11], automating the process of monitoring the operability of a spacecraft can augment or replace human decision making and thus increase reaction speeds, reduce errors and mitigate cognitive overload, enhance safety, lower costs, focus analysis and free human reasoning for more complicated tasks. However, in order to automate the monitoring process, we need to compare the data generated by the satellite to reference data. We extract this reference data from a model we developed predicting the behavior of the satellite.

When talking about monitoring operability, we have to distinguish between two different types of faults that can occur. First, we have to check the *housekeeping data*

(HK) and *attitude control system data* (ACS) to keep the satellite in orbit and on its correct course. This job is done by the ground-based space operations center. Second, we may also have faults in the instruments collecting the *scientific data*. Defects in this area do not put the satellite at immediate risk, but they may corrupt the data to such an extent that it becomes worthless for later scientific analysis. For this reason, it is mandatory to constantly analyze the scientific content of the data sent down from the satellite. In order to do this we developed a *Science Quick Look (ScQL)* monitoring framework. For the HIPPARCOS mission [1] (the first and only space-based astrometry mission up to now) the quality check of the scientific data was a very cumbersome task, since it had not been automated. Here we present an approach on how to meet this challenge by building a semi-automated monitoring and diagnosis system for the first time.

We have begun working in the framework of the DIVA project, which aimed at measuring the positions of about 35 million stars with the help of a low-cost satellite. In the meantime, DIVA has been absorbed into the larger GAIA project. At the moment, work done for DIVA is adapted to GAIA.

The remainder of the paper is organized as follows. Section 2 covers the related work. In Section 3 we give an overview of the general design of the ScQL process. The model that our monitoring system is based on is described in Section 4. Section 5 presents our monitoring and diagnosis system, while Section 6 gives a brief evaluation of our work. Section 7 contains concluding remarks.

2 Related Work

For comparing the observed data to our model and drawing conclusions from this, we rely on reasoning systems. In this section we will take a brief look at these systems. The first large group of reasoning systems comprise symptom-based approaches, in which symptoms are directly related to faults. Traditional rule-based systems represent the accumulated experience of experts in the form of empirical associations, i.e., rules that associate symptoms with their underlying faults. Examples for rule-based approaches (which include dictionary-based, tree-based, or use-case-based reasoning) can be found in [2,4,8]. Fault dictionaries are lists of symptom/fault pairs indexed by a description of the symptoms. In order to build such a dictionary, we need a specification on how a system behaves if a certain component is broken. The resulting list of fault/symptom pairs is then inverted to form our dictionary. An example of this technique can be found in [17]. Decision trees are a way to break down complex diagnostic situations hierarchically. This means that we step through a sequence of tests before arriving at a diagnostic solution. See [4] for more details. A relatively recent method is case-based reasoning, in which previously successful solutions are adapted to similar problems. Case-based systems learn by acquiring new knowledge in the form of additional case studies. Examples for these systems can be found in [16].

The second large group of reasoning systems are based on models. Here we do not rely on empirical knowledge about symptoms and faults, but on fundamental knowledge of the considered domain. Davis and Hamscher state some general rules in [4] on how to model the behavior, structure, and failures of a system from the viewpoint of troubleshooting. Qualitative models, as described in [5], are not concerned with exact quantitative values, but describe a system at a high level of abstraction. An example for a qualitative description of a parameter is that its value is increasing, decreasing, or constant. Inc-Diagnose is an algorithm developed by Ng [19] and is based on a formal theory by Reiter for diagnosis [14]. While the approaches described so far emerged from the AI community, there has also been work started from an engineering point of view, like fault detection and isolation (FDI) [3].

One thing that became clear during our study of the literature was that there is no universal method for coping with all possible situations. We favor a model-based approach, in which we can extract diagnostic clues from discrepancies between predicted behavior and observations. This approach is rather natural for a scientist (after all, the users of our system are going to be astronomers), as this is how he or she usually solves scientific problems. There are also some shortcomings of the symptom-based approaches, the main one being that it is difficult to comprehend why a system arrives at a certain conclusion. In contrast to this, a model-based approach rests upon established facts of the domain, rather than relying on empirical knowledge. Also, maintaining the knowledge for complex symptom-based systems is also a challenging task, as small changes in the design may necessitate revisions in a large part of the knowledge base.

3 General Design of Science Quick Look

3.1 Requirements of ScQL

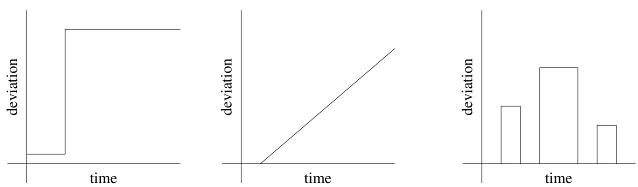
The job of the ScQL is to continuously check the correct operation of the on-board instruments. On the one hand, we have to monitor the hardware, i.e., the optics with its lenses and mirrors, the detectors with their light-sensitive areas (CCD chips), and different supply voltages. On the other hand, we have the software controlling the instruments and their output. In order to make clear what ScQL has to check here, we give a brief description of the modus operandi. The image detection software scans the raw data stream from the CCD array for star-like images. The centroiding algorithm determines the central positions of these images. Knowing this center position, the window cutting algorithm cuts out a small rectangular area around the image. This is done because transmitting the whole output of the CCD array to the ground station would use up too much of the limited data transmission rate. Then we have algorithms for identifying attitude stars. The (on-board) attitude star catalog consists of a collection of reference stars (with known positions). With the help of the reference stars the position of the satellite can be determined more accurately. This has a considerable impact on the quality of the scientific data. Last but not least, let us mention one more algorithm. Due to the rotation of the satellite, the stellar images are moving from the left to the right in the focal plane. To obtain sharp images, the shifting and reading out of the CCDs needs to be synchronized with the current rotation movement of the satellite. This is the job of the clock stroke rate adjustment. ScQL has to verify the validity of the output of all the algorithms described above.

3.2 Parts of the ScQL Process

Schematically, the process of ScQL is composed of three parts: a star transit simulator, monitoring, and diagnosis. All these parts depend heavily on the employed model. Our model takes into consideration the Galaxy, the structure of the satellite, the behavior of the satellite's components, and the scanning strategy of the satellite. The star transit simulation is responsible for describing the predicted behavior of the satellite. It is also used for generating simulated data during the development of the system (as no observed data is available yet). Monitoring produces statistics and derived parameters from observations. These values are compared to predicted ones and if the differences are too large, the system raises alarms. In the diagnosis step the symptoms generated by the monitoring process are related to faults in the system.

3.3 Faults and Residual Generation

We define a fault as a deviation of a parameter from the modeled (nominal) behavior of the satellite. We distinguish between abrupt, incipient, and intermittent faults (see Figure 3).



(a) Abrupt fault (b) Incipient fault (c) Intermittent fault

Fig. 3. Typical fault classes

The detection of faults is based on comparing the system parameters computed from observations (measurements) with the values generated by a model of the system. The difference between the two is referred to as a residual signal:

$$r_i(t) = \theta_i(t) - \hat{\theta}_i(i), \qquad 1 \le i \le k,$$

where r_i denotes the i^{th} residual, θ_i the i^{th} computed parameter from the measured system output, $\hat{\theta}_i$ the estimated (modeled) i^{th} system output parameter and k the number of residuals. The goal is to generate structured residuals to meet monitoring and diagnosis requirements. Residuals are designed to be equal or converge to zero in the fault-free case ($r_i(t) \approx 0$). A fault is described by the significant deviation of one or more residuals from zero ($\|r_i(t)\| > \eta_i > 0$, where $\eta_i \in \mathbb{R}$ denotes a threshold).

4 Star Transit Model

Before building the actual monitoring and diagnosis system, we had to develop a model for the domain. This is quite complicated, as many parameters and features have to be considered. We can only give a brief overview here, for details see [13].

The two main parts are the simulation of the sky and how it is perceived by the satellite. As we are interested in the nearest stars, our sky is practically equivalent to the Galaxy (see Subsection 4.1). For the satellite, we consider its movement (Subsection 4.2) and the behavior of its instruments (Subsection 4.3).

4.1 Simulation of the Galaxy

Modeling the distribution of the reference attitude stars is not that difficult, as they are chosen in such a way that they are homogeneously distributed throughout the sky. However, as we intend to map a much larger number of stars, we opt for a more realistic multi-component Galaxy model as described by Kharchenko et al. in [9,10]. In this model the Galaxy is viewed as a symmetrical system with respect to its rotation axis and its equatorial plane. The Galaxy is divided up into three parts consisting of the thin disk, the thick disk, and the spheroid. (The spheroid is the central part of the Galaxy, which is surrounded by a disk. The disk is divided into a thick disk, the inner ring around the spheroid, and a thin disk, the outermost, sparsely populated part of the Galaxy.) Within each group we have different density and luminosity (brightness) distributions. For simulating a complete map of the sky – including the positions and magnitudes of stars – we divide it up into 252 subareas and populate these subareas according to functions for spatial and luminosity distributions. For details on these functions see [13].

4.2 Scanning Law

When scanning the sky with a satellite, several conflicting constraints have to be considered [12]. The angle between the observed fields of view and the Sun should be at least 45° in oder to minimize straylight. The inclination of the scans on the ecliptic should be as small as possible because the parallactic effect is parallel to the ecliptic. Also, two successive scans should overlap in order to avoid unobserved regions. For DIVA the scanning law is as follows: the satellite does a complete rotation in 2 hours, the rotation axis moves slowly, circling the Sun in 56 days keeping an angular distance of 45° to the Sun.

4.3 Satellite Instruments and Data

The previous two subsections describe a theoretical view of the part of the universe that is relevant to us. When simulating the behavior of the actual satellite, we have to take into account that its instruments have a limited precision. This is reflected in the generation of the data from the simulated satellite.

The data that is produced by the satellite consists of long ribbons of data pixels. These ribbons are supplied by the CCD chips recording the movement of stars along the focal plane of the satellite. Our simulated data reflects this.

Windows containing star images are cut out and described by the following parameters: k_w, the number of the TDI (time-delayed integration) clock stroke at which the lower left corner is read into the read-out register, m_w the number of the column from which the lower left corner is read, and the type of the window (which also defines its size). Checking the validity of these parameters is one of the most important tasks of ScQL, so we focus on them. Our model currently contains a total of 30 parameters. This number will increase as we incorporate more features into our model.

5 Monitoring and Diagnosis System

5.1 Monitoring

Figure 4 shows the overall architecture of our monitoring system. The topmost layer provides a user of our system with a (graphical) user interface. It is based on IDL (Interactive Data Language) by Research Systems Inc. IDL provides tools for visualization, data analysis, and cross-platform application development [15]. Here, we only employ the user interface part. Below that, we have the Foundation Class Layer (FCL) which was originally developed by Smirnov [18] in order to simplify the creation of applications for the analysis of astronomic data. We adopted and modified the two main parts of the FCL, the Visualizer and the DataForms. The Visualizer provides a lot of easy-to-use widget tools for the visualization of large and complex data sets. DataForms is responsible for collecting the input from the user. On the lowest level, there is the actual ScQL-monitoring. It consists of an EventHandler preprocessing the user input, forwarding it to the component responsible for generating statistics and parameter estimations. In turn, the output of this component is fed into the Visualizer and a component responsible for controlling visualization parameters. The statistics-generating component (which is the most complicated part) is capable of checking the parameters, some of which were mentioned in Section 3.1. For details, interested readers are referred to [13].

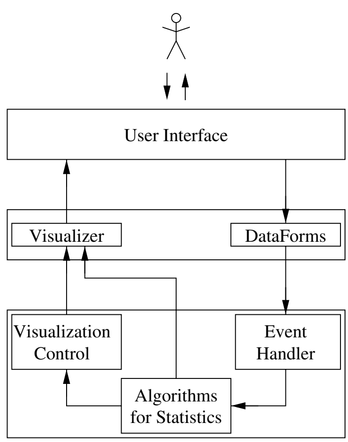


Fig. 4. Architecture of the monitoring system

5.2 Diagnosis

Our current work focuses on the monitoring system, i.e., the diagnosis system is not worked out as elaborately as its monitoring counterpart at the moment. Basically, it consists of a set of 50 rules implemented in COOL (CLIPS Object-Oriented Language). CLIPS (C Language Integrated Production System) is a tool for building expert systems [6,7]. The object-oriented approach of COOL allows us to compose the diagnosis system using modular components, which can be reused later.

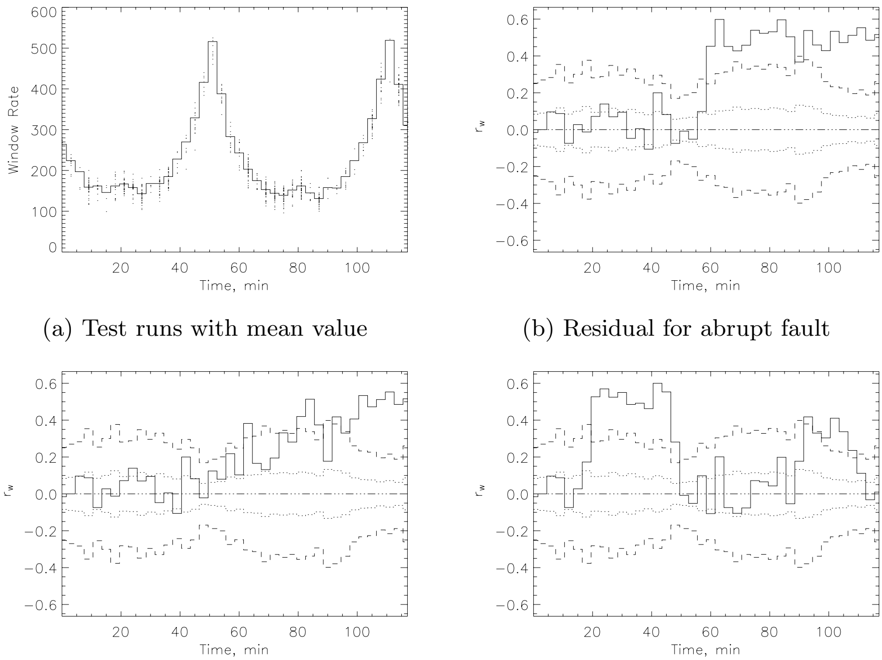
6 Short Evaluation of Our Approach

The monitoring system was tested extensively by feeding simulated star transit data into it and generating and evaluating statistics. As an example parameter, we present our evaluation results for the window rate. This parameter tells us how many cut-out windows can be found in a certain period of time in the data stream from the satellite. Figure 5(a) shows how many windows arrive in each three-minute interval. We restrict ourselves to displaying a total elapsed time of two hours and stars with a brightness down to a magnitude of $V = 10.5$ (which corresponds to approximately 1.2 million stars) to keep things legible. The peaks can be explained by the fact that the distribution of the stars is not homogeneous. The density is highest when scanning in the equatorial plane of the Galaxy (the satellite does a full rotation in two hours). Each dot in Figure 5(a) stands for one test run (a total of 50 runs were done in this case). The solid line represents the mean value of all observations.

We now introduced some faults into the simulated data (an abrupt, an incipient, and an intermittent fault) and watched the reaction of the monitoring system. The monitoring system was calibrated with the measurements shown in Figure 5(a). We determined a σ and 3σ interval around the mean value. Figure 5(b) shows the mean window rate as a reference (dash-dotted line at y=0.0). The dotted line is the σ interval and the 3σ interval is the dashed line. The solid line is the residual computed by comparing the (simulated) observation with the expected value. After approximately 60 minutes, we introduced an abrupt fault decreasing the window rate (simulating the malfunctioning of a part of the CCD array). The monitoring system reacted promptly to this event, as the residual signal went over the threshold of 3σ. As a consequence, an alarm was raised.

Figure 5(c) (for an incipient fault after 60 minutes) and Figure 5(d) (for intermittent faults between 20 to 50 and 90 to 110 minutes) show a similar picture. For the incipient fault, an alarm is raised after 60 minutes. The system goes back to normal status for a short time, only to reach a permanent alarm level after 70 minutes. For the intermittent faults the system goes into alarm status for the duration of the faults and reverts back to normal after they are over.

We should mention that during the mission the thresholds will change over time, as we collect more and more information. After half a year of observations,



(a) Test runs with mean value

(b) Residual for abrupt fault

(c) Residual for incipient fault

(d) Residual for intermittent fault

Fig. 5. Window rate

our knowledge about the sky will improve substantially (as we are able to construct a complete sphere). This will allow us to refine our model and determine thresholds that are as close as possible to the nominal case without raising false alarms.

7 Conclusion and Outlook

In order to guarantee a high level of precision when collecting scientific data during a space-based astrometry mission, the quality of the data needs to be checked constantly. The first goal of our work was to study the problem of monitoring scientific data considering the specific characteristics of a astrometry space mission. We decided to build a model-based system, as it is the most appropriate approach in our opinion. At the core of this model is a star transit simulator that mimics the behavior of the satellite and simulates its observations. We also implemented a prototype of a monitoring system that is able to process astronomical data in quasi-real time.

The results of an evaluation of our systems are very promising, so we plan to pursue further studies in this area. First of all, we want to improve the diagnosis part of our system to bring it on par with the monitoring part. Second – as the DIVA project has run out – we will adapt our approach to the next space-based astrometry mission, GAIA, whose satellite will be launched in 2012. The frameworks of DIVA and GAIA are quite similar, due to the fact that the underlying principles of operation and the basic geometry of the measurements are the same. Building a ScQL monitoring system for GAIA has become a lot easier, as an important first step has already been taken in the DIVA project.

References

1. European Space Agency. The Hipparcos and Tycho catalogues. *SP-1200*, 1997.
2. E.M. Awad. *Building Expert Systems: Principles, Procedures, and Applications*. Addison-Wesley Publishing Company, INC., 1996.
3. J. Chen and R.J. Patton. *Robust model-based fault diagnosis for dynamic systems*. Kluwer academic publishers, Boston, 1999.
4. R. Davis and W. Hamscher. *Exploring Artificial Intelligence*. Morgan Kaufmann Publishers, 1988.
5. J. de Kleer. Physics, qualitative. *Articial Intelligence*, 1992.
6. J. Giarratano and G. Riley. *CLIPS Reference Manual*. PWS Publishing Company, Boston, 1998.
7. J. Giarratano and G. Riley. *CLIPS User's Manual Volume II: Objects*. PWS Publishing Company, Boston, 1998.
8. F. Hayes-Roth, D.A. Waterman, and D. Lenat, editors. *Building Expert Systems*. West Publishing Company, 1983.
9. N. Kharchenko, S. Rybka, A. Yatsenko, and E. Schilbach. Predicted star counts and mean parallaxes down to the 23rd magnidute. *Astron. Nachr. 318*, 3:163–171, 1997.
10. N. Kharchenko and E. Schilbach. Program mega: Stellar counts and galactic models. *Baltic Astronomy*, 5:337–356, 1996.
11. C.A. Kitts. A global spacecraft control network for spacecraft autonomy research. *Proceeding of SpaceOps'96: The Forth International Symposium on Space Mission Operations and Ground Data Systems in Munich, Germany*, September 1996.
12. J. Kovalevsky. *Modern Astrometry*. Springer-Verlag Berlin Heidelberg New York, 1995.
13. A. Pavlov. *A Model-Based Monitoring System for Rapid Assessment of Payload and Spacecraft Health/Performance*. PhD thesis, Universität Mannheim, Mannheim, 2005.
14. R. Reiter. A theory of dianosis from the first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
15. Inc. 1994 Research Systems. Idl user's guide. *http://www.rsinc.com/idl/*.
16. R.C. Schank and R.P. Abelson. *Scripts, Plans, Goals and Understanding*. Erlbaum, Hillsdale, New Jersey, US, 1977.
17. J.W. Sheppard and W.R. Simpson. Improving the accuracy of diagnostics provided by fault dictionaries. In *14th IEEE VLSI Test Symposium (VTS '96)*, pages 180–186, 1996.
18. O. Smirnov. Dasha-2:improving visualization and processing of photometric data with idl objects. *Astronomical Data Analysis Software and Systems VIII*, 172:437, March 1999.
19. Hwee Tou Ng. Model-based, multiple-fault diagnosis of time-varying, continous physical devices. pages 9–15, 1990.