

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

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Dr. rer. nat.
Alexey Pavlov
aus Moskau

Mannheim, 2006

Dekan: Professor Dr. Matthias Krause, Universität Mannheim
Referent: Professor Dr. Guido Moerkotte, Universität Mannheim
Korreferent: Professor Dr. Rainer Spurzem, Universität Heidelberg

Tag der mündlichen Prüfung: 15. März 2006

Abstract

Astrometric space missions like Hipparcos, *DIVA*, Gaia have to simultaneously determine a tremendous number of parameters concerning astrometric and other stellar properties, the satellite's attitude as well as the geometric and photometric calibration of the instrument. To reach the targeted level of precision for these missions many months of observational data have to be incorporated into a global, coherent and interleaved data reduction. It is inevitable that a daily data reduction process is required in order to judge if the level of precision of the stellar, attitude and instrument parameters achieve its targeted level. This sophisticated data analysis is the in-depth scientific assessment of the quality of all observations within about 24 hours after its reception. It is based on the very complicated procedure "First Look preprocessing" (more known as a Great-Circle reduction from Hipparcos) that provides a one-dimensional, self-consistent and simultaneous solution of the attitude, the instrument calibration and celestial source parameters. For this purpose one needs to process all the 24-hours-data, a task which can be only performed at the Data Center with its computer resources. On the other hand, it is necessary and reasonable to process the observations at the ground Space Operations Center for a quick discovery of delicate changes in the spacecraft performance in the quasi-real time constraints ($15 \div 30$ min after data reception).

For this latter purpose, the concept of a model-based monitoring system has been developed that comprises activities concerning scientific data health of an astrometrical satellite which can not be guaranteed by only standard procedures applied to typical space missions. This monitoring system, called Science Quick Look (ScQL), performs the preliminary scientific assessment of the instrument and proper astrometric working of the spacecraft at the (coarse) level of precision attainable at this stage. The prototype of this software is designed in the framework of the *DIVA* project, providing monitoring, diagnostic and visualization tools. It performs the first scientific assessment of the geometric stability of the instrument and proper working of the spacecraft. The process of the monitoring is based on a model of the Galaxy, on the structure and behavior of the components of the spacecraft and its scanning strategy. The system incorporates a simulator of the observations of stars – a core of our model, that allows to mimic the work of the on-board software and to simulate star transits.

The results of an evaluation of our system look very promising, so we plan to pursue further studies in this area. As the *DIVA* project was stopped, we will adopt our approach to the next space-based astrometry mission, Gaia, which will be launched in 2012. Indeed, many aspects for the rapid assessment of payload and spacecraft health, developed in this work in the framework of *DIVA* project, are analogous to those in Gaia due to the fact that the basic principle and geometry of the measurements are the same.

A successful completion of the ScQL prototype for the *DIVA* mission provides the basis for our belief that a ScQL monitoring system for the larger project – Gaia – is

achievable in terms of the developed concept. Building a ScQL monitoring system for Gaia therefore would become a lot easier if the important steps have already been done in the DIVA project. It is evident, however, that this work has to evolve and grow, along with the concept of the Gaia satellite.

Zusammenfassung

In astrometrischen Weltraummissionen wie Hipparcos, DIVA oder Gaia muss eine Vielzahl von Parametern gleichzeitig bestimmt werden, wie astrometrische und andere stellare Eigenschaften, die Lage des Satelliten sowie die geometrische und photometrische Kalibrierung der Instrumente. Um das Präzisionsniveau zu erreichen, das in diesen Missionen angestrebt wird, müssen die über Monate gesammelten Daten in eine globale und kohärente Datenreduktion eingebunden werden. Der Prozess der täglichen Datenreduktion ist unbedingt notwendig, um einschätzen zu können, ob der angestrebte Präzisionsgrad der stellaren Parameter, der Lage des Satelliten und der Instrumente erreicht ist. Mit dieser anspruchsvollen Datenanalyse wird die Qualität aller Daten innerhalb von 24 Stunden nach deren Empfang mit wissenschaftlichen Methoden gründlich überprüft. Sie basiert auf einer komplizierten Prozedur, First Look Preprocessing genannt. Diese Prozedur liefert eine eindimensionale, selbstkonsistente und simultane Lösung für die Parameter der Lage des Satelliten, der Kalibrierung der Instrumente und der kosmischen Objekte. Diese Analyse kann nur im Data Center durchgeführt werden, weil hier die entsprechenden Computerressourcen zur Verfügung stehen. Andererseits ist es notwendig und vernünftig, die Beobachtungen auf der Erde im Space Operations Center durchzuführen, weil auf diese Weise minimale Abweichungen in der Performanz des Satelliten in Quasi-Echtzeit (15-30 min. nach Empfang der Daten) rasch entdeckt werden können.

Zu diesem Zweck wurde das Konzept eines modellbasierten Überwachungssystems entwickelt. Es umfasst alle Aspekte der Korrektheit der wissenschaftlichen Daten eines astrometrischen Satelliten, die mit den Standardprozeduren gewöhnlicher Weltraummissionen nicht garantiert werden können. Dieses Überwachungssystem, Science Quick Look (ScQL) genannt, überprüft schon im Voraus, ob das Instrument und der Satellit richtig funktionieren, auf dem Präzisionsniveau, das in diesem Stadium möglich ist. Der Prototyp dieser Software wurde im Rahmen des DIVA-Projekts entwickelt und bietet Tools für Überwachung, Diagnose und Visualisierung. Die Software führt eine erste wissenschaftliche Beurteilung der geometrischen Stabilität des Instruments und der korrekten Funktion des Satelliten durch. Der Ablauf der Kontrolle basiert auf einem Modell der Galaxie sowie auf der Struktur und dem Verhalten der Komponenten des Satelliten und seiner Strategie. Das System enthält einen Simulator für die Beobachtungen von Sternen - der Kern unseres Modells, der es ermöglicht, die Arbeit der Software an Bord zu imitieren und die Sterntrajektorien zu simulieren.

Eine Bewertung unseres Systems lieferte viel versprechende Ergebnisse, deshalb planen wir weitere Studien auf diesem Gebiet. Da das DIVA-Projekt abgebrochen wurde, werden wir unseren Ansatz bei der nächsten astrometrischen Weltraummission, Gaia, anzuwenden versuchen, die 2012 starten soll. Viele Schritte zur raschen Überprüfung von Belastung und Zustand der Raumsonde, die in dieser Arbeit für das DIVA-Projekt entwickelt wurden, entsprechen denen in GAIA, da das Grundprinzip und die Geometrie

der Messungen in beiden Projekten gleich sind.

Im Rahmen dieser Arbeit wurde die Entwicklung eines ScQL-Prototyps für DIVA erfolgreich abgeschlossen. Deshalb sind wir der Überzeugung, dass es möglich ist, innerhalb des bestehenden Konzepts ein ScQL-Überwachungssystem für Gaia zu entwickeln. Es bleibt jedoch anzumerken, dass eine solche Arbeit gemeinsam mit dem Konzept des Gaia-Satelliten wachsen und sich weiter entwickeln muss.

Preface and acknowledgments

This thesis is submitted in fulfillment of the requirements for the Doctor of Natural Sciences at the Lehrstuhl für Praktische Informatik III, Universität Mannheim, Germany. The work has been carried out in the period from January 2002 to January 2005 under the supervision of Professor Guido Moerkotte.

The thesis considers the design of a monitoring system for the rapid assessment of payload and spacecraft performance using a model-based approach. It can be applied to an astrometrical scanning satellite where the health of the scientific data can not be guaranteed by only standard monitoring procedures applied to typical space missions. The results presented in this work are generally based on the involvement into the DIVA project.

Looking back over the last three years, reminding me how this work depended on the efforts and contributions of many people. I would like to acknowledge all those people who played an integral role in helping me to reach the completion of my dissertation. Without their support and friendship I would not have had the great pleasure of writing this page.

Mannheim University

First of all, I am gratefully indebted to my supervisor Professor Guido Moerkotte for his guidance throughout this thesis. His capacity to combine the penetrating criticism and the offering direction as well as his academic experience helped me to establish the overall direction of the research and to remain focused on achieving my goal.

I would further like to thank very much Dr. Sven Helmer for his permanent advices during the endeavor of my doctoral research. His observations and comments on my drafts significantly improved the concept of my thesis and helped me to move forward with the investigation in depth.

Astronomisches-Rechen Institut (ARI)

I would like particularly to mention Dr. Elena Schilbach and Dr. Siegfried Röser for giving me the opportunity to participate in the preparation of the DIVA space mission and for their unending support during my work on this project. I am grateful for their preparedness to enlighten me on various aspects of the project, without their help the learning curve would have been very much flatter.

I am extremely grateful to Dr. Michael Biermann for helping me getting started with the task of the science quick look diagnosis and and for his great contribution.

I would like to express my gratitude to Dr. Sonja Hirte for your invaluable help as well

as informal support and encouragement.

I also extend my sincere gratitude to Dr. Ulrich Bastian who helped me to learn about the features of the astrometrical space projects during my first year at ARI. I am also grateful that he gave me the opportunity for being shortly involved in the space project *Gaia* and to visit the *Gaia* meeting in Barcelona-2003, it provided me with a wealth of information and insights.

I am grateful to Frau Helga Ballmann from ARI and Frau Simone Seeger from Mannheim University for assisting me all the time in the real world of the German society. I would have been lost without their help.

Institute of Astronomy, Russian Academy of Science (INASAN)

I would like to mention Dr. Nina V. Kharchenko and Dr. Anatoly E. Piskunov who encouraged me to burn the midnight oil, so that I could manage (almost!) to keep my deadline. I have certainly benefitted from extensive discussions with them during the last months of my stay at ARI. The invaluable help in understanding different aspects of the modern Galaxy model was highly appreciated.

Klaus Tschira Foundation

I gratefully acknowledge the financial support from the Klaus Tschira Foundation.

It would, of course, be completely amiss for me to end my acknowledgments without recognizing the immense contribution that my parents have made to my work. Despite living about 2000 km away, they are constantly in my thoughts and their love and support have been a major stabilizing force over these past three years. Their unquestioning faith in me and my abilities has helped to make all this possible and for that, and everything else, I dedicate this thesis to them.

Contents

1	Introduction	1
1.1	Sky	1
1.2	Astrometry	3
1.3	Astrometric satellites	6
1.4	DIVA-mission overview and main instrument	8
1.5	Motivation for rapid assessment of payload and spacecraft health in the framework of an astrometry mission. Science Quick Look.	11
1.6	The Goal of Ph.D. Thesis	12
1.7	Guide to the Dissertation	13
2	Monitoring systems	17
2.1	Monitoring and alarms	17
2.2	Artificial intelligence techniques for monitoring. Related work.	19
2.2.1	Symptom-based approaches	19
2.2.2	Model-Based Reasoning	22
2.2.3	Fault detection and isolation	26
2.3	Summary	26
3	From Requirements to Diagnostic Solutions	29
3.1	Requirements of the Science Quick Look (ScQL)	29
3.2	The design of the ScQL	35
3.3	Summary	41
4	Star transit simulation	43
4.1	The Sky	44
4.1.1	Equatorial coordinates; the International Celestial Reference System (ICRS)	45
4.1.2	The Ecliptic Coordinate System	47
4.1.3	Galactic Coordinates	47
4.1.4	Coordinate transformations	48
4.2	The model of Galaxy	49
4.2.1	Stellar distribution and simulation of a map of the sky	49

4.2.2	Simulation of the stellar magnitude distribution	50
4.3	A multi-component model of the Galaxy	54
4.3.1	Stellar distribution: fundamental equation of stellar statistics . . .	54
4.3.2	Luminosity functions	56
4.3.3	A sky map simulation	57
4.4	Attitude Star Catalogue	58
4.5	The Instrument	58
4.6	DIVA science data and coordinate systems	61
4.6.1	The CCD pixel stream	61
4.6.2	The body-fixed satellite system and field coordinates	62
4.6.3	Focal-plane coordinates	63
4.6.4	Pixel coordinates and grid coordinates of a star image	64
4.6.5	Geometric calibration of the instrument (grid-to-field transformation)	65
4.6.6	Reference great-circle coordinates	66
4.6.7	Windows	66
4.6.8	CCD and window numbering: specifics for k , m , n and $type$. . .	67
4.7	The nominal scanning law	68
4.8	Star transits and emulation of the work of the on-board software	70
4.8.1	The true-band sky	71
4.8.2	Attitude matrix	71
4.8.3	Electronics	71
4.8.4	Star transits	71
4.8.5	Emulation of on-board software	73
4.8.6	Prediction of SM2 star transits from SM1 detections	75
4.8.7	Star rate determination and TDI clock stroke rate adjustment . .	78
5	Monitoring proper and Diagnosis	81
5.1	ScQL monitoring dataflow	81
5.2	Monitoring proper	82
5.2.1	The scheme of the monitoring proper	82
5.2.2	Statistics, parameters and residuals.	84
5.3	ScQL diagnosis	88
5.3.1	Distinguishable faults	89
5.3.2	Indistinguishable fault	94
5.3.3	Severity of the different faults	95
5.4	Landscape of the environments	95
5.4.1	Database and mixed-language programming	95
5.4.2	ScQL monitoring and MLP	97

6	Science Quick Look prototype evaluation	103
6.1	Basic cycle	103
6.2	Transits of attitude stars	104
6.2.1	The nominal simulation	104
6.2.2	Monitoring of TDI clock stroke rate adjustment	107
6.3	All star transits	109
6.3.1	The nominal simulation in the framework of the multi- component Galaxy model. Time-varying thresholds.	109
6.3.2	Monitoring of the window rate: an abrupt, an incipient and an intermittent fault.	113
6.3.3	Monitoring of the star brightness	120
6.3.4	Monitoring of the star centroids	122
6.4	Summary	128
7	Conclusion and Outlook	131
7.1	Conclusion	131
7.2	Outlook	134
8	Appendix	135
8.1	The contribution of DIVA to astrophysics	135
8.2	Window datation and Image Parameter Set	136
8.2.1	Explanation of the data type descriptions	136
8.2.2	Windows	137
8.2.3	Image Parameter Set	141
8.2.4	Classification according to instrument type	141
8.2.5	Classification according to object type	142
8.2.6	Special window types, non-object windows	142
8.2.7	Data quantity	142
8.3	Attitude and nominal scanning law	146
8.3.1	Attitude matrix	146
8.3.2	The nominal scanning law	146
8.4	DIVA main instrument	148
8.5	Acronym Collection	153
	Bibliography	154

List of Figures

1.1	The panorama view of the sky. Credit: Knut Lundmark.	2
1.2	The Milky Way system	2
1.3	Diagram of parallax.	4
1.4	Mean errors of star positions and parallaxes in history.	6
1.5	DIVA scanning law	9
1.6	The schematic view of the optical layout of the main DIVA telescope. . .	9
1.7	The DIVA focal plane assembly: the main instrument uses a configuration of four identical CCD mosaics each consisting of 4 1k×2k CCDs, a 5th one (SM3) being added as cold redundant device.	10
1.8	The large scale interface between the ground segment (GS) and data center (DC) concerning Quick Look, Science Quick Look and First Look. 13	
2.1	Typical architecture of a monitoring system.	18
2.2	Comparison of qualitative and quantitative approaches.	25
2.3	Basic perspective of MBD [24].	27
3.1	A model-based system design.	30
3.2	Overall scheme of the daily monitoring and diagnosis.	30
3.3	The ScQL monitoring parts: SS – Sky (Galaxy) & Satellite models, O – Observations, S – symptoms, F – faults.	36
3.4	Three typical classes of the time-dependency of faults: (a) abrupt fault, (b) incipient fault, (c) intermittent fault.	38
4.1	Coordinates of the celestial objects are usually given in the equatorial system. Left: celestial sphere and equatorial coordinate system. Right: equatorial and cartesian coordinate systems (★ – the star position on the sphere).	46
4.2	Ecliptic coordinate system.	47
4.3	The brightness distribution according to [45] (◇) and [19] (solid curve). .	53
4.4	Test of the simulation of the stellar magnitude distribution: simulated (histogram) and theoretical (solid line) cumulative brightness distribution functions.	53

4.5	The luminosity functions $F_j(M_V)$ in stars/mag/pc ³ assumed in the model. The blue solid line and red dashed line are the luminosity functions of the disk main sequence and disk red giants respectively. The luminosity function of the thick disk is presented by the dotted green curve whereas the stars of the spheroid are shown by yellow dot-dashed line.	56
4.6	This shows modelled integral number of stars distributed through the Galaxy as a function of the magnitude. The shaded regions refer to the total number of stars used to simulate the <i>true sky</i> and <i>attitude star catalogue (ASC)</i>	57
4.7	The simulated map of the Galaxy for the stars with magnitude up to $M_V = 6.5$ in galactic coordinates (l, b) : lower panel – orthogonal projection, upper panel – AITOFF equal-area projection. It shows that at brighter magnitudes the sky is rather represented by an isotropic stellar density distribution.	59
4.8	A simulated map of the Galaxy for the stars with magnitude up to $M_V = 8.5$ in galactic coordinates (l, b) : orthogonal projection (lower left panel), upper panel – AITOFF equal-area projection. Lower right panel presents the Galaxy map in ecliptic coordinate system (λ, β) . Filled color polygons specify an array of color indexes and star magnitudes M_V . It shows that the features of the disk-like star distribution of our Galaxy increase with regards to a fainter magnitude. Note: an upper limit of the stellar magnitude $M_V = 8.5$ was used in this sample in order to keep the picture of the map more visible. In the DIVA simulator stars are simulated up to $M_V = 15.5$	60
4.9	The body-fixed satellite system and field coordinates (γ is the angle between two field-of-views).	63
4.10	Pixel coordinates of SM image.	65
4.11	Windows in the pixel data stream	67
4.12	The figure shows the numbering of the CCD's by n . The meaning of m is also shown.	69
4.13	The nominal scanning law of DIVA in ecliptical coordinates: we show the part of the sky scanned in 7 consecutive days (\star – coordinates of the begin of observations).	70
4.14	Flow chart of the simulation process.	72
4.15	Schematic SM1 and SM2 mosaics with readout registers indicated as red lines at the right side of CCDs. Displacements and rotations of the individual CCD chips with the mosaic are exaggerated.	73
5.1	Dataflow of the ScQL monitoring (in simulation).	82
5.2	Structure of an ScQL monitoring proper employing the MFCL.	83
5.3	The scheme for a typical DIVA application based on mixed language programming (the details see in [53]).	97

5.4	The example of the ScQL monitoring Visualizer and DataForms objects. In the top panel the rate of the window for the whole SM CCD mosaic (upper histogram, white solid line) and for SM CCD chip number 7 are presented (lower histogram, yellow solid line); the time is given in TDI clock stroke ($\delta t = 10^5$ TDI). In the bottom panel the centroid positions (C_k, C_m) in downloaded windows in scan- and cross-direction reckoned from the left corner of the window (see Table 8.5 in Appendix 8.2)	100
6.1	Basic cycle.	104
6.2	Window rate and residuals (in relative units) vs. time on the SM CCD array (CCDs mosaic) for a binsize 5 minutes (solid line histogram). Blue dotted line denotes the analytically mean value. The calculated mean window rate from the simulated observation science data flow (mean $o. = 822.89$) coincides with the value derived from the analytical model (mean $m. = 822.86$).	105
6.3	Window rate and residuals (in relative units) vs. time on the SM CCD chip for a binsize 5 minutes (solid line histogram). Blue dotted line denotes the analytically mean value. The calculated mean window rate from the simulated observation science data flow (mean $o. = 103.70$) coincides with the value derived from the analytical model (mean $m. = 102.86$).	106
6.4	Window rate and residuals (in relative units) vs. time on the SM CCD ADU chain for a binsize $\delta t = 5$ minutes (solid line histogram). The blue dotted line denotes the analytically determined mean value. The calculated mean window rate from the simulated observation science data flow (mean $o. = 52.23$) coincides well with the value derived from the analytical model (mean $m. = 51.43$).	106
6.5	The calculated (known) scan speed of the satellite with its uncertainties. In this example an abrupt increase of the true scan speed is simulated on 30 May 2005 at 01:00:00 UTC. The picture shows the determination of this effect by the ScQL software. Error bars refer to mean errors (binsize 2 min.).	107
6.6	Calculated (known) TDI clock stroke rate (left panel) and its residuals (right panel) vs time (binsize 2 min). Left panel: known TDI clock stroke rate with its uncertainties (bars). Yellow area denotes a small time interval (4 min) when the blurred images are delivered. Right panel: solid line histogram is the residuals of TDI clock stroke rate, that confirms the correctness of the work of the on-board software; dot-dashed histogram is the residuals of the TDI clock stroke rate in the case of a permanent maladjusted shifting of CCD charge. . .	108

- 6.7 An illustration of the simulation of the band sky for one run (sample). This picture shows four sky subareas (the total number of which is 252) and the band of stars for one sample of the sky map. The sky map is simulated according to the multi-component Galaxy model in a given sky band defined by the nominal scanning law. A total number of 50 runs are carried out in order to calculate the expected mean values of the star transit along the band of stars. An average total number of stars in one band is about $1.7 \cdot 10^5$ which corresponds to about $7 \cdot 10^5$ star transits in one complete rotation of the satellite. To simulate 2 hours of observations takes 16 minutes of CPU time on Dell Inspiron 5150, powered by the mobile Intel Pentium 4 processor at 3.06GHz with hyper-threading technology. 109
- 6.8 Window rate vs. time on the SM CCD array (binsize 2 min). Upper panel: (1) the black dots – nominal window rates based on 50 different samples of the band sky simulation according to the multi-components Galaxy model (see sec. 4.3 of Chapter 4) and (2) the blue solid-line histogram – the calculated predictive (mean) nominal window rate. Lower panel: the predictive window rate and its standard deviation (error-bar). 111
- 6.9 **Nominal.** Window rate vs. time on SM CCD array (binsize 2 min). Upper panel: (1) the blue solid-line with error-bars – the predictive window rate and (2) the black solid-line histogram – the simulated nominal window rate. Lower panel: (1) the black solid-line histogram – the residuals r_w^{CCDs} (see f. (5.1) in sec. 5.2.2 of Chapter 5), (2) the green dashed-line histogram and (3) red dashed-line histogram are the time-varying thresholds on the level of 1σ and 3σ correspondingly. All fluctuations of the nominal window rate residuals are in the framework of 3σ 112
- 6.10 **Abrupt fault.** Window rate vs. time on SM CCD array (binsize 2 min). Upper panel: (1) the blue solid-line with error-bars – the predictive rate window, (2) the black solid histogram – the incorrect window rate. Lower panel: (1) the black solid histogram – the residuals r_w^{CCDs} (see f. (5.1) in sec. 5.2.2 of Chapter), (2) the green dashed-line histogram and (3) red dashed-line histogram are time-varying thresholds on the level of 1σ and 3σ correspondingly. The abrupt decrease of the the whole CCD mosaic window rate is simulated as a damage (or shut down) of the four CCD chips (SM1: n=1 and n=2; SM2: n=5 and n=6) on 30 May 2005 starting at 00:10:00 UTC; the considered fault-scenario covers time interval of 2 hours [00:00:00, 02:00:00] UTC. 114

6.11 **Intermittent fault.** Window rate vs. time on SM CCD array (binsize 2 min).
 Upper panel: (1) the blue solid-line with error-bars – the predictive rate window, (2) the black solid histogram – the incorrect window rate. Lower panel: (1) the black solid histogram – the residuals r_w^{CCDs} (see f. (5.1) in sec. 5.2.2 of Chapter), (2) green dashed-line histogram and (3) red dashed-line histogram are time-varying thresholds on the level of 1σ and 3σ correspondingly. The intermittent fault of the window rate on whole CCD mosaic is simulated as a the temporarily shutdown of the four (SM1: $n=1$ and $n=2$; SM2: $n= 5$ and $n=6$) and three (SM1: $n=1$ and $n=2$; SM2: $n= 5$) CCD chips on 30 May 2005 at [00:20:00 - 00:50:00] and [01:30:00, 01:50:00] UTC correspondingly; the considered fault-scenario covers the time interval of 2 hours [00:00:00, 02:00:00] UTC. 115

6.12 **Incipient fault.** Window rate vs. time on SM CCD mosaic (binsize 2 min).
 Upper panel: (1) the blue solid-line with error-bars – the predictive rate window, (2) the black solid histogram – the incorrect window rate with the degradation of the SM CCD mosaic. Lower panel: (1) the black solid histogram – the residuals r_w^{CCDs} (see f. (5.1) in sec. 5.2.2 of Chapter), (2) green dashed-line histogram and (3) red dashed-line histogram are time-varying thresholds on the level of 1σ and 3σ correspondingly. The decrease of the the whole CCD mosaic window rate is simulated as a consequent damage (or shut down) of the CCD chips on SM2 on 30 May 2005 starting at 00:50:00, 01:00:00, 01:20:00, 01:40:00 UTC; the considered fault-scenario covers the time interval of 2 hours [00:00:00, 02:00:00] UTC. 116

6.13 The residuals of the window rate on the chip level for the case of the consequent damage (or shut down) of the chips on the SM2 mosaic (see also figure 6.12). 117

6.14 Hierarchical structure of the SM CCD array for monitoring of the science data. 118

6.15 Overview of the Gaia Astro focal plane. The two Astro telescopes share the same focal plane: Astro field-1 is the area indicated in red, Astro field-2 is outlined in green. Three functions are assigned to the focal plane system: (i) the Astrometric Sky Mapper, which detects object entering the field of view, and communicates details of the star transit to the subsequent astrometric and broad-band photometric fields; (ii) Astrometric Field (AF), devoted to the astrometric measurements; (iii) the Broad Band Photometer (BBP) which provides multi-color broad-band photometric measurements for each object. The resulting focal plane design consists of a mosaic of 180 CCDs with pixel of $10\mu m$ along scan \times $30\mu m$ across scan size (44.2 mas \times 132.6 mas). From left to right: the first columns are for the sky mappers with the following orange area of the CCDs for astrometry (analogue of DIVA SM CCDs); the last five columns are for broad-band photometry. 119

- 6.16 The decrease of the window rate on the whole CCD array (upper panel) is due to the temporal problem with the preceding FOV (PFOV) of the main instrument. The residuals of the window rate from the preceding FOV (left panel) together with the residuals of the window rate from the following FOV (FFOV) (that shows no anomaly (right panel)), identify this failure's occurrence. Note: in the case of DIVA this failure can be only identified for attitude stars. 121
- 6.17 The residuals of the window rate of the CCD array. The decrease of the window rate on the whole CCD mosaic is due to the simulation of the high background in time interval [01:00:00, 01:20:00] UTC. 122
- 6.18 Unbiased estimators $S_N(V)$ of the cumulative function for the two brightness distribution are constructed. The statistic D is defined as the maximum value of the absolute difference between two distributions (see text). 123
- 6.19 The histogram of the brightness distribution for the same time interval as in figure 6.17. The observed star histogram demonstrates an absence of faint stars starting from $V_{\text{lim}} = 9$ mag. 124
- 6.20 Along-scan C_k and across-scan C_m centroids vs. time on SM CCD mosaic for a nominal simulation (dots on the left panels); the histograms (green lines) are the average centroid position C_K and C_L (binsize 2 min). On the right panel are the appropriate residuals (black line histogram) and its thresholds (red dashed line). 125
- 6.21 The average centroid positions C_K (green line histogram) and its residuals (black line histogram) for the following rows (pairs) of CCD chips: [(1-5), (2-6)] in the case of the dual failure scenario. Dots are along-scan centroids C_k . Yellow line histograms (chip 5) are expected centroid positions C_K (and its residuals) if only the failure of the detection algorithm took place. 126
- 6.22 The same as on the figure 6.21 but for the other rows of CCD chips: [(3-7), (4-8)]. 127
- 6.23 Learning diagnostic knowledge cycle. 129

List of Tables

1.1	Space astrometry missions.	7
4.1	Predicted integral stellar density of the Galaxy	52
4.2	The geometric arrangement of the nominal DIVA SM CCDs.	76
8.1	Windows, Header Contents	138
8.2	Window, Record Contents, Part 1: Window Datation	139
8.3	Window, Record Contents, Part 2: Pixel Values	140
8.4	Image Parameter Set, Header Contents	144
8.5	Image Parameter Set, Record Contents	145
8.6	DIVA collection of parameters, Part 1 (general parameters)	148
8.7	DIVA collection of parameters, Part 2 (general parameters)	149
8.8	DIVA collection of parameters, Part 3 (data rates)	150
8.9	DIVA collection of parameters, Part 4 (data rates)	151
8.10	DIVA collection of parameters, Part 5 (data rates)	152

Chapter 1

Introduction

1.1 Sky

If it's a good, clear night, and if you aren't underneath a sky too affected by light pollution from artificial lighting, you should be able to see hundreds of stars on the sky (see fig. 1.1). All of those stars belong to our Galaxy. There are a very few things one can see which aren't stars: a band of light, clusters and galaxies. These will generally look like faint, fuzzy patches. The band of light is just the integrated light of the many stars in the Milky Way; point a telescope at any spot along it, and you will see that it resolves into rich star fields. The presence of the Milky Way is the first hint that we live in a disk-like distribution of stars.

As viewed from the picture 1.2, the Milky Way system is a spiral galaxy consisting of over 400 billion stars, plus gas and dust arranged into three general components as shown to the left:

- the halo - a roughly spherical distribution which contains the oldest stars in the Galaxy;
- the nuclear bulge and Galactic Center;
- the disk, which contains the majority of the stars, including the sun, and virtually all of the gas and dust.

The disk of our galaxy is approximately 30 *kpc*(kilo-parsec) across (or 100,000 light-years)¹. This is a huge distance; compare this to 1.29 pc, the distance to Alpha Centauri (one of the closest stars to the Sun), or to 2.7 pc, the distance to Sirius (the brightest star in Earth's sky). The solar system is situated within the outer regions of this galaxy – smaller spiral arm, called the Local or Orion Arm, well within the disk and only about 6 pc above the equatorial symmetry plane but about 8.5 kpc from the Galactic Center.

¹It is worth to note that in science the galactic scale distances are measured in parsec (1 pc = 3.26 light-years = 3.09×10^{16} meters).

Chapter 1. Introduction



Figure 1.1: The panorama view of the sky. Credit: Knut Lundmark.

This panorama, called "7,000 Stars And The Milky Way" was made under the supervision of astronomer Knut Lundmark at the Lund Observatory in Sweden. To create the picture, a mathematical distortion was used to map the entire sky onto an oval shaped image with the plane of our Milky Way Galaxy along the center and the north galactic pole at the top. 7,000 individual stars are shown as white dots, size indicating brightness.

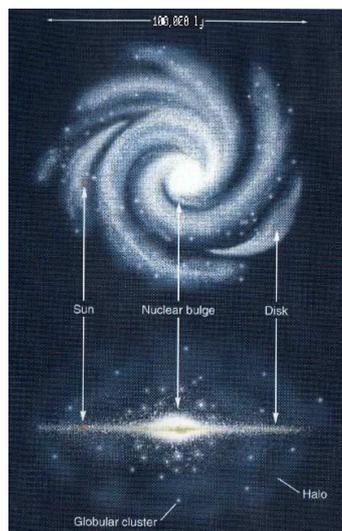


Figure 1.2: The Milky Way system

1.2 Astrometry

One of the basic parts of the knowledge that we presently have about the Sky (Universe) has been obtained by techniques of astrometry. In short, the astrometric measurements

- gave the first confirmation that stars lay at very large, but nevertheless finite, distances;
- led to William Herschel's discovery of the existence of double stars as actual physical pairs (binary systems);
- led to Nevil Maskelyne's demonstration of gravitational attraction between bodies of astronomical dimensions, and hence to a confirmation of Newtonian's law of universal gravitation;
- led to the estimation of the distance between Earth and the Sun based on the observations of the transit of Venus across the solar disc;
- led to the estimation of the distance of the stars by direct measurements (Bessel, Henderson, Struve);
- led to the prediction of the eighth planet Neptune as a result of orbital perturbations of Uranus, independently by Adams and Le Verrier (nineteen century);
- led to the discovery of white dwarfs in 1915 (it was Bessel's suggestion in 1844 that the motion of Sirius was perturbed by a faint companion).
- The first confirmation of Einstein's theory of General Relativity, from the perihelion motion of Mercury's orbit and the gravitational light deflection during solar eclipse, was also based on astrometric measurements of the very highest accuracy available at that time.

Astrometry is the domain of astronomy devoted to the determination of positions of stars (and other celestial bodies), their distance and motions. These quantities generally vary with time so that the primary goal of astrometry is to describe their motions. The stellar distance estimates are crucial to our understanding of stellar properties and underpin the whole distance network for galactic and extragalactic astronomy.

By measuring the change in a star's position as the Earth revolves around the Sun, one can determine the distance to that star (see 1.3). This change in the position is known as a star's parallax. Parallax is the apparent change in position of a star due to the actual change in the earth's position in its orbit around the sun. A photograph is taken of a star at one time during the year, and the position of the star with respect to the background stars is measured. Then a photograph is taken six months later, when the earth is on the opposite side of the sun, and the second position of the star with

Chapter 1. Introduction

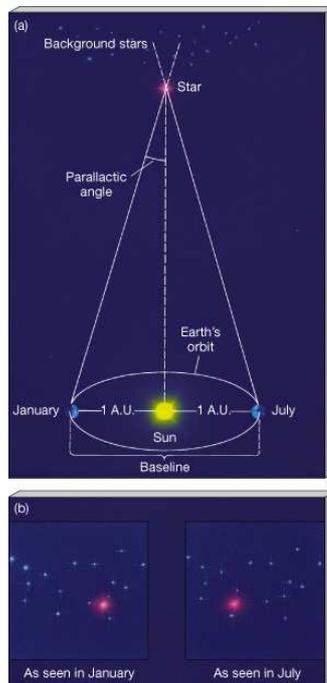


Figure 1.3: Diagram of parallax.

respect to the background stars is measured again. (This, of course, was only a general description, the actual measurements is far more complicated.)

The star will appear to move slightly with respect to the background stars, and this motion is called its parallax. Using simple geometry, we can use this parallax to calculate the star's distance from the sun. Distances to astronomical objects are very difficult to determine because the stars are very distant from us. Even with the rather big baseline available for measuring distances to stars – the diameter of the orbit of the earth that is an almost circular ellipse (length of this baseline $2 \text{ AU} = 300$ million kilometers) – the parallactic angles are extremely small.

The concept of parallax was discovered (predicted) by the ancient Greeks, who learned that the stars are very far away because they were not able to measure the parallax of any stars. It wasn't until 1838 that German Mathematician and Astronomer Friedrich Bessel was able to measure the distance to the star 61 Cygnus. In the 1830's the parallaxes for other two nearest stars, Alpha Centauri and Vega, were measured by Henderson from South Africa and by Friedrich von Struve correspondingly.

Such observations demanded enormous precision. Where a circle is divided into 360 degrees (360°), each degree is divided into 60 minutes ($60'$)– also called "minutes of arc" to distinguish them from minutes of time–and each minute contains 60 seconds of arc ($60''$). In measuring star distances, astronomers use a parsec, that is defined as the distance from the Sun which would result in a parallax of 1 second of arc as seen from

Earth². One parsec equals 3.26 light years, but as already noted, no star is that close to us. Alpha Centauri, as mentioned one of the nearest star to our solar system, has a parallax of 0.75".

While parallax observations give us information about the distance of a star, the study of proper motions tells us how these stars are moving in space. This motion of the stars with respect to each other is small due to the huge distance to even the nearest stars (but larger than the parallactic angle). For instance, the nearest star to the Sun, Proxima Centauri, moves with about 4 arcseconds per year. Although we see the motion of stars projected on the celestial sphere (in 2 dimensions), in reality, the stars are moving in 3 dimensions. Radial motion is the motion of an object along our line of sight, and this can be measured using the Doppler shift of the object's spectral lines. For astronomers, this is a "relatively" quick task (one measurement is enough). Finding the motion of an object in the other 2 dimensions, or coordinates, is a bit tougher. The motion of an object in these other two coordinates, perpendicular to the line of sight, are what is known as the proper motion. Finding the proper motion of a celestial object takes up much time and energy in the life of an astronomer because he/she has to wait long periods of time to actually observe the physical motion across the sky of the object, depending on the accuracy of astrometric measurements. Only after observing how the star moves over many years relative to background stars we can measure the proper motion of the star.

The distances and proper motions of the stars are highly valued quantities to be obtained in astronomy, so is the study of stellar parallaxes of utmost importance to astronomers. That's why the measurement of celestial angles, in particular the angles between stars has been a preoccupation of astronomers for the last few hundred years. Over the last hundred years, with only relatively small advances in astrometric precision made possible by measurements from the Earth's surface, the most important applications of astrometry have been the continuing determination of stellar distances by measuring parallaxes, the estimation of stellar velocities by measuring proper motions, and the setting up of a reference frame for the study of Earth, planetary and galactic dynamics.

So, until the end of the 19th century, before the development of what we call astrophysics, astronomy consisted only of astrometry (with its theoretical counterpart, celestial mechanics), and all observations were directed towards obtaining positions of celestial bodies. Since then astrophysics has become the most important domain of astronomy. With the extension of observations to almost all wavelengths from radio waves to gamma rays, with the use of very sensitive new receivers and the development of fast computers, remarkable progress has been made in the description and the understanding of the Universe. At that period of time, till about 1970, astrometry didn't take part in this development of astronomy, and so parallaxes and proper motions that could be obtained only by astrometric techniques. As a consequence, some basic domains

²The word "parsec" is an abbreviation and contraction of the phrase "parallax second".

Chapter 1. Introduction

of astrophysics became conspicuously uncertain in comparison with progress achieved elsewhere (see [47]). Since 1970, new techniques such as radio astrometry chronometric methods, CCD receivers and astrometric satellites have changed astrometry. Thanks to them and in particular to the latter, when exquisitely precise instruments were put on board a satellite orbiting the Earth to get above the blurring effects of the atmosphere, astrometry has become a completely renovated science.

1.3 Astrometric satellites

The tremendous success of the Hipparcos mission [2], leading to gains several orders of magnitude in precision and accuracy (see fig. 1.4), clearly shows that astrometric space missions are very attractive and appropriate. As has been mentioned, accurate astrometric measurements are required for improving our understanding of the Universe. Small errors in the measured distances of stars can therefore lead to large errors in the distances to galaxies. Astrometric measurements are the bedrock of methods for determining distances to astronomical objects. Modifications and improvements to this distance scale do impact our understanding of astrophysical phenomena both nearby and at cosmological distance scales.

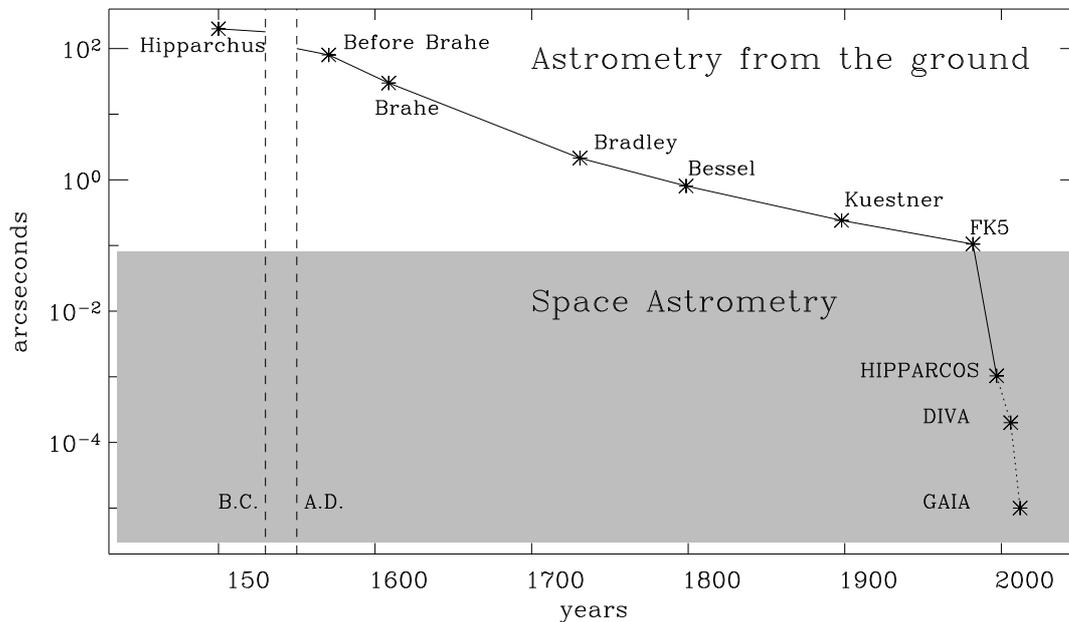


Figure 1.4: Mean errors of star positions and parallaxes in history.

The most ambitious astrometric space project is the cornerstone mission Gaia [49]

in the ESA Horizon 2000+ program. The Gaia satellite will undertake a very detailed and extensive astrometric and photometric study of our Galaxy with the primary goal to determine its formation, composition and evolution. Gaia will observe every object in the sky brighter than $V=20$ mag, that is about one billion (10^9) stars, galaxies, quasars and solar system objects, so there are also numerous supplementary science projects ranging from exo-solar planets to fundamental physics. High precision astrometry will be obtained for every object, yielding accurate positions, parallaxes and proper motions. The expected typical accuracy for parallaxes at $V=15$ mag is $\sim 10\mu$ (microarcseconds). In Gaia radial velocities will also be obtained (down to about $V=18$ th mag), thus producing a 6D phase space catalogue (three spatial and three velocity co-ordinates) for a significant fraction of our Galaxy.

DIVA [1, 57, 13] – a small astrometry satellite – has been designed to fill the gap in observations between Hipparcos and Gaia. Due to the progress in technology since the time when Hipparcos was designed, DIVA would have been able to surpass the performance of Hipparcos in every respect at a price less than a tenth of the price of Hipparcos. DIVA should have carried out a sky survey to measure positions, proper motions and parallaxes, magnitudes and colors of about 35 million stars (down to about $V=15$ th mag). It had surpassed the performance of Hipparcos in all relevant aspects: by the number of objects observed, the measurement accuracy and by its vast number of photometric and spectrophotometric data; table 1.1 presents the brief summary data of the three space astrometry missions. DIVA observation should have taken 2 years,

Table 1.1: Space astrometry missions.

Mission	Hipparcos	DIVA	Gaia
limiting magnitude	~ 12 mag	~ 16 mag	~ 20 mag
number of objects	120 000 objects	35 million objects	1 billion objects
measurement accuracies	1 mas at < 9 mag	200 μ as at 9 mag, 2 mas at 14 mag	10 μ as at 15 mag, 200 mas at 20 mag
launch	1989 (performed)	2005 (canceled)	2012 (approved)

followed by 2 years of data reduction. The final catalogue, containing positions, parallaxes, proper motions and the spectrophotometry (with stellar parameters derived such as surface temperature, gravity, and metallicity as well as extinction, within limits of accuracy) was planned to be published by 2008/2009. The detailed list of expected DIVA measurements/contributions to various fields of astronomy is shown in Appendix 8.1.

However, in December 2002 due to finding problems, the DIVA project was canceled by the German Space Agency (DLR). In a sense DIVA could have been an important precursor mission for the technology of the upcoming Gaia. Whereas these two missions differ certainly in their sheer size and in most of the details, there is a lot of underlying similarity due to the fact that the basic geometry and signal detection of the measurements are very similar.

Chapter 1. Introduction

The present work was started in the framework of the DIVA project, it focuses mostly on the science data from the astrometrical part of the instrument, leaving aside the photometric data as well as radial velocities. Proceeding from this assertion and taking into account that the DIVA observation strategy and reduction concept of the observational data has a principle similarity with Gaia, we have decided to continue building the prototype of Science Quick Look in the terms of DIVA. From now on let us concentrate on the DIVA features in order to create the prototype of a model based expert system for rapid assessment of spacecraft performance, keeping in mind that some modifications will be needed to satisfy Gaia's requirements.

1.4 DIVA-mission overview and main instrument

DIVA has into its final orbit a perigee height of 500 km and an apogee height of 71100 km. The time of revolution is 24 h, which allows the use of only one ground station for the data link. This orbit is a compromise between performance and cost. On the selected orbit, data link to the ground station is possible for 19 hours, the typical expected data rate for the transfer of scientific data was ~ 700 kbit/sec.

DIVA follows a scanning law similar to that of Hipparcos. The satellite rotates around the z-axis (fig. 1.5) in two hours. This rotation axis moves on the surface of a cone with its axis pointing towards the sun, the angle between rotation axis and the sun direction being 45 degrees. One revolution of the rotation axis on that cone takes 56 days. With this scanning law, one sky coverage is performed in about six months.

The DIVA satellite carries two instruments: an optical telescope (main instrument) and a UV telescope. The main instrument simultaneously observes two fields-of-view (FOV) separated by about 100 degrees using a single focal plane assembly for both fields. The light from the two FOVs (Preceding FOV "PFOV" (or "FOV1") and Following FOV "FFOV" (or "FOV2")) and is fed into the telescope by a beam combiner mirror, which is the entrance pupil of the system.

In fig. 1.6 the schematic view of the optical layout of the main telescope is shown. The grid on the last fold affects only a part of the focal plane (fig 1.7). This part, receiving dispersed light, houses the so-called spectroscopic CCDs (SC1 and SC2). Outside this area, in undispersed light, two CCD mosaics, called Sky Mappers (SM1 to SM2), are mounted. All mosaics are identical and each consists of 4 individual chips with 1024×2048 pixels of $13.5 \text{ micron} \times 13.5 \text{ micron}$.

Due to the rotation of the satellite the stellar images are moving from left to right in the focal plane. The integrated exposure time per mosaic transit is 1.4 sec. All CCDs are clocked synchronously with the actual rotation of the satellite, i.e. they are operated in the so-called time-delayed integration (TDI) mode. The actual rotation rate, nominally 180 arcsec/s, is determined in real-time to an accuracy better than 0.1 arcsec/s from the crossings of individual stars through SM1 to SM2.

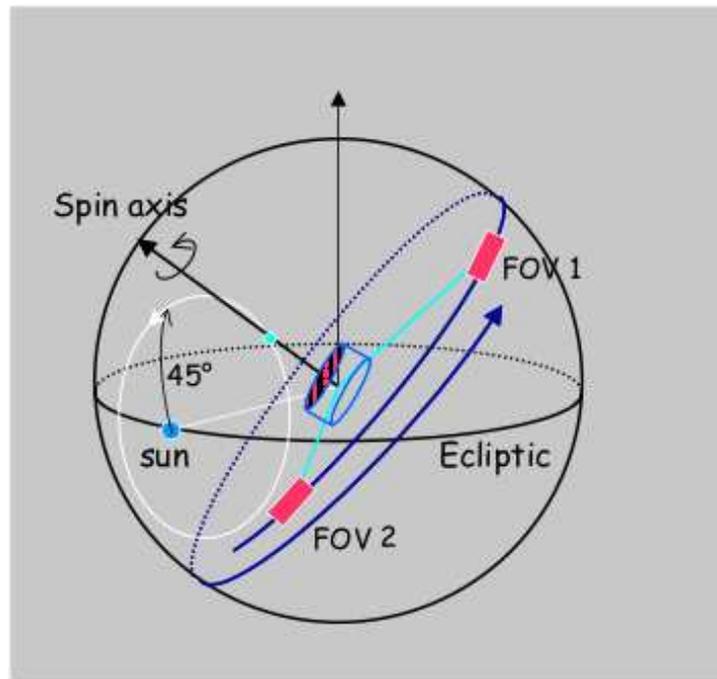


Figure 1.5: DIVA scanning law

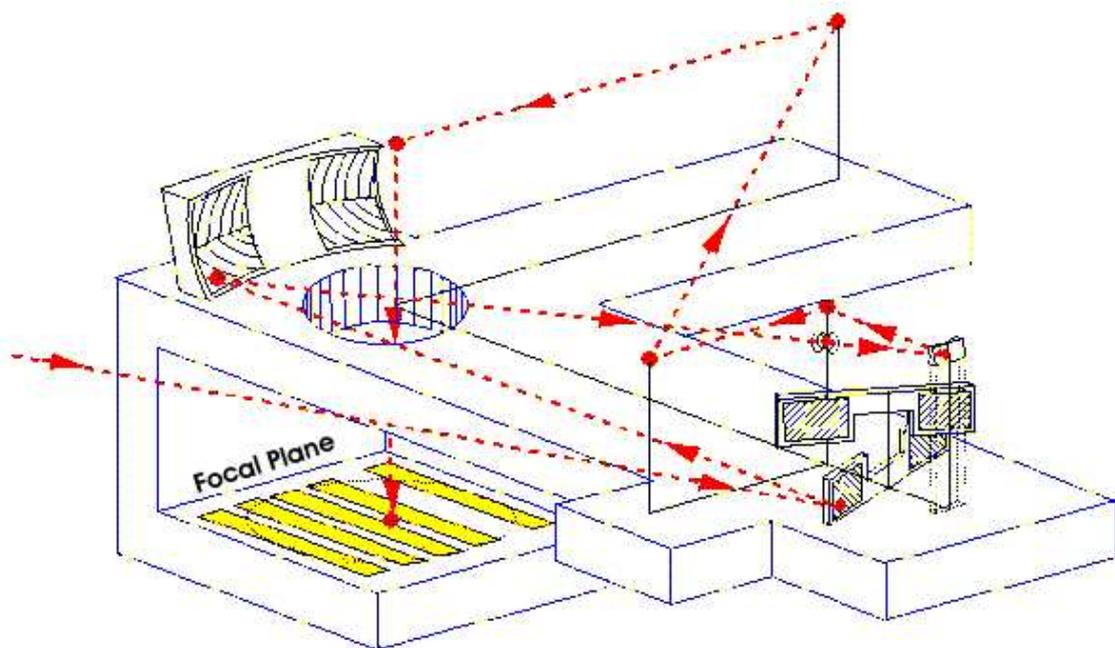


Figure 1.6: The schematic view of the optical layout of the main DIVA telescope.

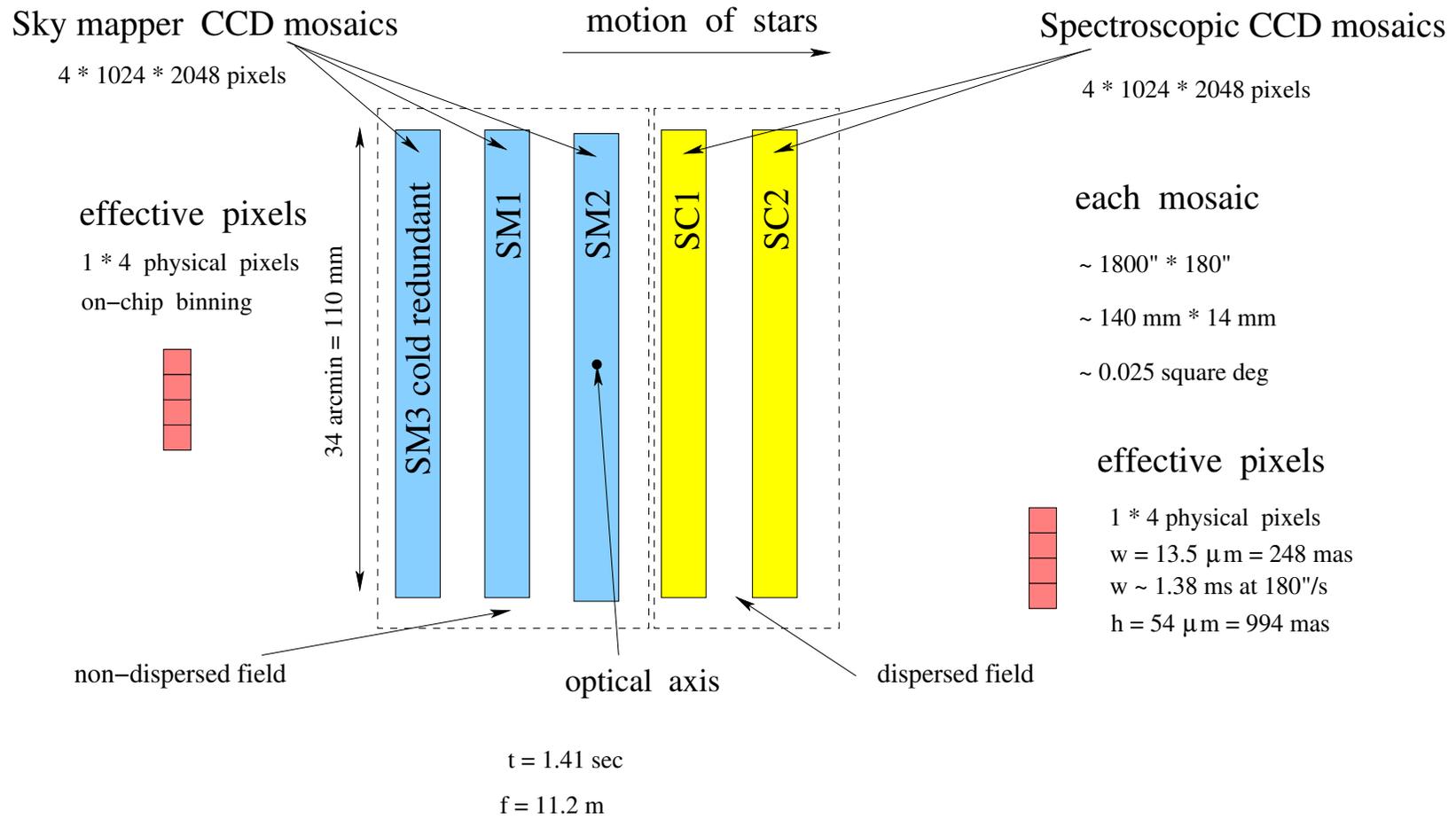


Figure 1.7: The DIVA focal plane assembly: the main instrument uses a configuration of four identical CCD mosaics each consisting of 4 $1\text{k} \times 2\text{k}$ CCDs, a 5th one (SM3) being added as cold redundant device.

After detection in SM1 the on-board software shall predict windows surrounding the stars in the continuous pixel stream of the CCDs in the SM2 and in dispersed field. Only these predicted windows around the stars in the SM2, SC1, SC2 along with the detected windows in SM1 will be transmitted to the ground.

In scan direction the full width of the central "Airy" fringe in the both the SMs and the SCs is about 1.4 arcsec at a central wavelength of 750 nm. In cross-scan direction it is 2 times larger. Therefore and because the main astrometric measurements are done along scan a four times larger pixel size in this direction is used. In the DIVA concept, this is achieved via an on-chip binning of four pixels in cross-scan direction. So, read-out noise as well as on-board data rates are reduced.

Attitude control is performed using a conventional cold-gas system. The analysis of the perturbations yielded that attitude maneuvers to keep the satellite on its scanning law are only needed at intervals larger than 10 min. Real-time attitude determination will ensure the knowledge of the attitude in all three axes to $10''$. During routine phases the use of the Sky Mapper CCDs will provide the attitude with an accuracy of better than $1''$.

The DIVA CCDs produce a pixel data stream that is much too big to be telemetered to the ground entirely. The DIVA observation strategy is an astronomically motivated procedure for selecting specific parts of that raw CCD pixel data stream for inclusion into the telemetry data stream sent to the ground. It is based on the fact that most of DIVA's raw CCD data cover almost empty, dark sky.

The essence of the idea [12] is to send only the so-called windows to the ground, small sections cut out of the bulk data stream, which are centered on detected or predicted locations of celestial objects. In our study we use a realistic model observation strategy (presented in [12]) which we discuss in the chapter 4.

1.5 Motivation for rapid assessment of payload and spacecraft health in the framework of an astrometry mission. Science Quick Look.

The spacecraft health management is a specific task of mission operations that has been enhanced through the use of automation technologies. Through careful deployment within the overall mission architecture, automation can augment or replace human decision making in order to increase reaction speeds, reduce errors and mitigate cognitive overload, enhance safety, lower costs, focus analysis and free human reasoning for strategic task [46]. Therefore, this automation requires high levels of robustness.

The space industry uses ground based monitoring software to analyze spacecraft health and detect faults. For the DIVA mission this monitoring will be performed in the framework of the Quick Look (QL) task. The QL software will help human operators to take care of the housekeeping of the satellite (bus and instrument) and attitude control

Chapter 1. Introduction

system and prepares telecommanding if necessary. It is performed in real-time or in quasi real-time at the Space Operations Center³, using only so-called *housekeeping data* (HK).

However, the DIVA(Gaia) mission differs from typical astronomical space missions in a way that the health of the science data can neither be judged nor guaranteed by means of diagnostics based only on HK data and attitude control system (ACS) data. It is mandatory to analyse also the science data. The term *science data* is used as a collective term for the bulk science data resulting from the instruments (SM-, SC-CCDs data etc.).

For this reason a sophisticated daily diagnostics (First Look, FL) of the state of the satellite is required to judge the level of precision of the stellar, attitude and instrument parameters and to achieve its targeted level. The FL is the in-depth scientific assessment of the quality of all observations within about 24 hours after its reception. It is based on the complicated procedure called First Look preprocessing (or more known as a Great-Circle reduction from Hipparcos) that provides an one-dimensional, self-consistent and simultaneous solution of the attitude, the instrument calibration and celestial source parameters. For this purpose FL needs to treat all science data that can be only performed at the Data Center with its computer resources.

However, it is reasonable to process the science data for a quick discovery of delicate changes in the spacecraft and performance at the same location where the QL task is executed, i.e. at the ground station and also in the quasi-real time constraints.

To do this we propose to develop a Science Quick Look (ScQL) monitoring, that will comprise the activities concerning science data health and alerts the operator when problems occur. Figure 1.8 depicts the scheme of the daily diagnostic with relations between the Space Operations Center and the Data Center . The ScQL will perform the preliminary scientific assessment of the instrument and proper astrometric working of the spacecraft at the (coarse) level of precision attainable at this stage. It should run in quasi real-time at the space operations center using scientific data as well as HK data.

1.6 The Goal of Ph.D. Thesis

The general goal of this research is to find ways to assist the operator-astronomer at space operations center with the complex problem of rapid assessment of the astrometrical satellite by

- ☆ **helping them to deduce the changes of the state of the system from the science data, producing statistics and diagnostics,**
- ☆ **providing expert advice on possible explanations of the problems.**

³Space Operations Center = Ground Station (e.g. GSOC – German Space Operations Center, Oberpfaffenhofen).

In particular, the focus of this study is on creating a prototype of the Science Quick Look monitoring software that should perform an initial science monitoring and diagnostics using the astrometrical science data in quasi real-time constraints. The ScQL is a simplified version of the First Look, and intended to immediately react on obvious deviations from what is expected for the science data in order to save an observation time.

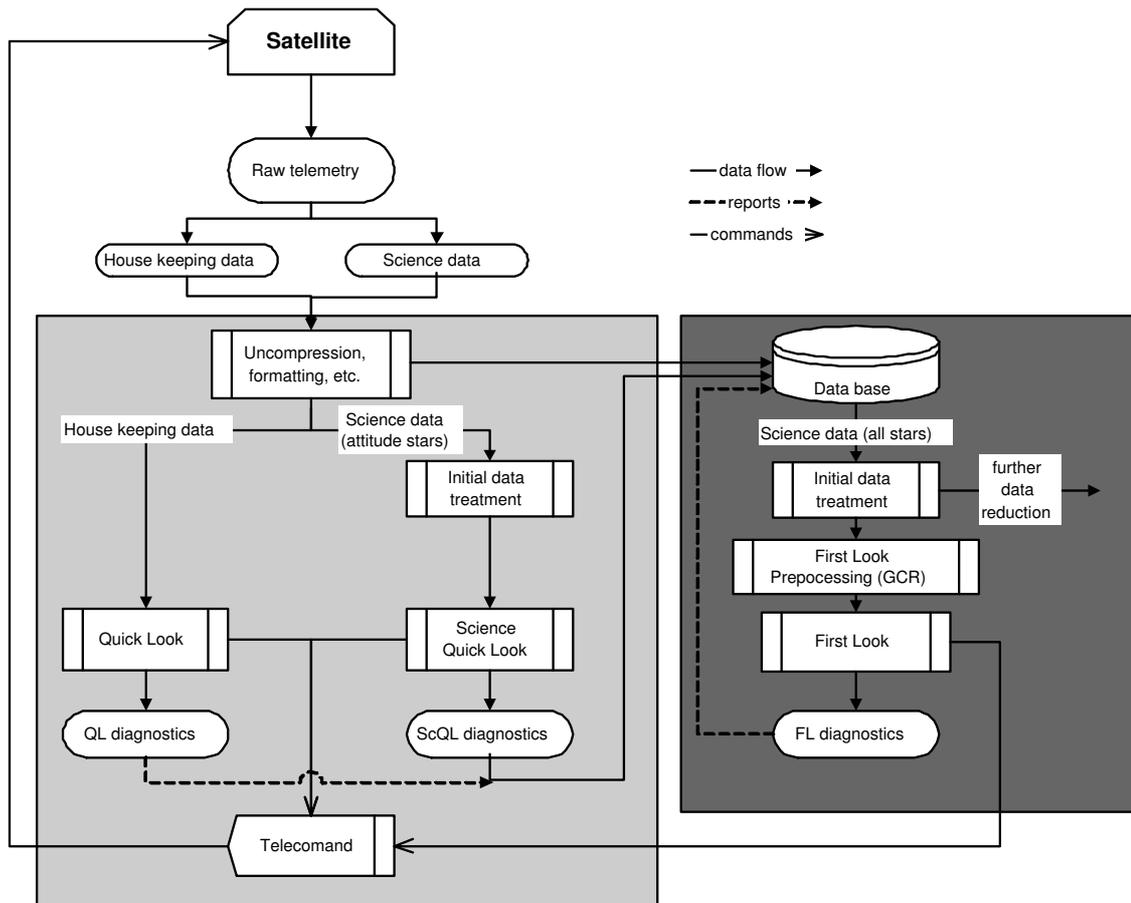


Figure 1.8: The large scale interface between the ground segment (GS) and data center (DC) concerning Quick Look, **Science Quick Look** and First Look.

1.7 Guide to the Dissertation

The dissertation is organized into 7 chapters. This first chapter has introduced the astrometrical spacecraft domain and its specific problem of the satellite health management. It is giving an overview of the astrometry and space astrometry missions such as Hippar-

Chapter 1. Introduction

cos, DIVA and Gaia. The problems of monitoring of an astrometrical scanning satellite are described and we reported about the motivation and goal of this investigation. The rest of the thesis is organized as follows:

- ❖ **Chapter 2** introduces the monitoring for dynamical systems, its objectives and typical architecture. We then discuss different artificial intelligence techniques for the monitoring and diagnosis task, focusing on the possible appropriate approaches for the task of the Science Quick Look. In particular, we describe the traditional symptom-based systems such as a ruled-based system, fault dictionaries, decision trees and a case-based system and a new type of a diagnosis system with the so-called model-based reasoning, that relies on principle knowledge of the domain. We present the main components of model-based diagnosis systems, and then generalize the results of our study from literature about the modern state of the model-based concept.
- ❖ In **chapter 3** we describe the conceptual design of the Science Quick Look monitoring. We introduce the design of the Science Quick Look, starting from setting up its monitoring requirements to the diagnostic solution. We introduce our model-based approach with three functionally different parts of the ScQL system. We proceed from general to specific, exposing the main features of the monitoring system for a scanning satellite. We define a fault notion and discuss a residual structure for correct alarm residual evaluation in the framework of our ScQL monitoring.
- ❖ The monitoring and the diagnosis are both based on a model for DIVA observations which is very complicated in the case of the scanning astrometry spacecraft. To build the complete model of the observations is extremely challenging task. The first step into this direction is the model that is explicitly implemented in the star transit simulator presented in **chapter 4**. In the framework of this simulator we define important components (sub-models) of our model, introduce and describe important entities and notions related to the sky and instrument that will be applied to the whole system.
- ❖ In **chapter 5** we describe the design of the monitoring proper and the diagnostic parts of the ScQL in detail. The landscape of the environments for a typical DIVA-application are discussed and the concept of mixed-language programming is presented. The latter is used in order to build a rather flexible and database compatible ScQL monitoring software. Also the structure of the Visualizer is given, that is applied for the ScQL monitoring proper to provide a complex widget tool and display the pre-processed (and/or simulated) science data, statistics and estimated parameters. Next, the statistics itself and estimated parameters at various levels of aggregation as well as the time-varying thresholds for alarms

are described. Using these estimated parameters, detected by the ScQL monitoring proper, we introduce an initial list of possible explanations of the detected problems in the section dedicated to ScQL diagnosis. We emphasize that some of the presented faults can not be detected and uniquely identified by our diagnosis engine; in these cases a more detailed simulation is required. This should be the task for future work.

- ❖ **Chapter 6** describes our evaluating procedure of the ScQL monitoring based on the scenario – simulation – monitoring – matching cycle for its iteratively evaluating and improving of the system. It allows to watch and validate the reaction of the ScQL monitoring. The science data with nominal parameters are simulated to derive computational solutions for a fault-free behavior of the instrument. The number of fault scenarios are shown as examples of the ScQL prototype evaluation.
- ❖ **Chapter 7** lists accomplishments, practical considerations, limitations and outlook of this work.

Chapter 1. Introduction

Chapter 2

Monitoring systems

A year spent in artificial intelligence is enough to make one believe in God.
– Linux fortune program.

2.1 Monitoring and alarms

Any physical system evolves with time, either due to its own dynamics or under the impact of external actions or events. Informally, monitoring a dynamic system can be viewed as performed by a high level module which continuously keeps track of the system, analyses all the situations encountered, communicates with human operators and suggests decisions to be taken in case of malfunction. There is very little consensus as to the architecture of the ideal monitoring system. In general, it consists of several cooperating modules (see figure 2.1):

- a **detection module** gathers elementary information (from raw data) and decides whether the evolution of the process is normal, if not an alarm is generated;
- a **diagnosis module**, responsible for one or more of the following tasks:
 - identifying characteristic situations;
 - localizing the faulty components responsible for the situation;
 - determining the primary causes of the abnormalities detected;
- a **decision module** which determines the actions which can be undertaken to reach the objective or bring back the process into normal conditions.

These modules rely on a knowledge base which contains:

- a library of detailed behavioral models of the physical system, under normal operating conditions and, when possible, in the presence of failures or in degraded modes;

Chapter 2. Monitoring systems

- the objectives to be met by the system, possibly together with plans of actions both in normal and in abnormal conditions,
- a description or several descriptions on various levels of the present state of the system, summarizing recent observations or expressing these in a language which can be understood by the operator.

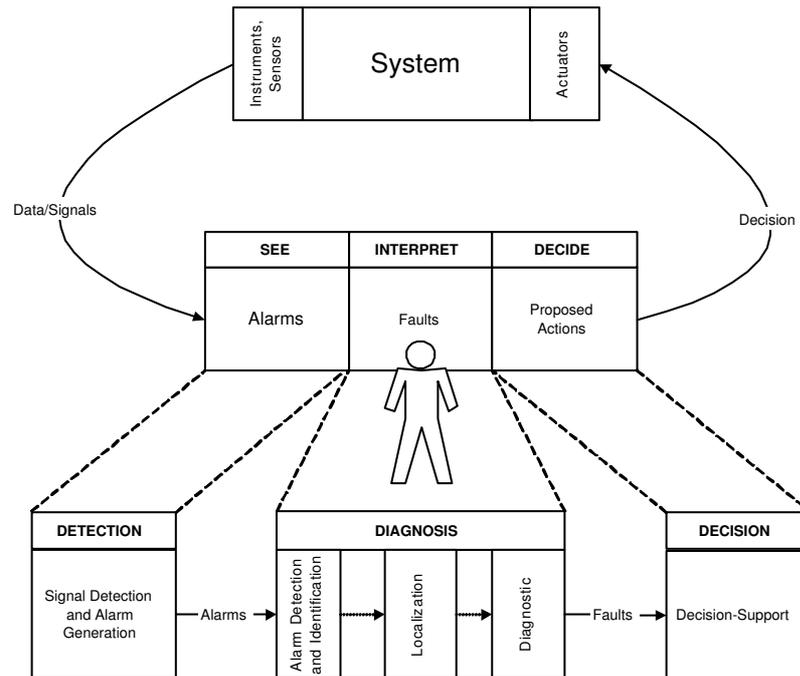


Figure 2.1: Typical architecture of a monitoring system.

The monitoring heavily relies on alarms, which rely in turn on sensor values, on the observations related to the environment, etc. In fact, the concept of "alarm" can vary from one application to another, which complicates the understanding and expression of the results. In [16] the definition of "alarm" is suggested to be adopted, after analyzing together several applications in various sectors such as nuclear power systems, telecommunication networks, etc.). This definition is based on the notion of "event" which we define first followed by the one of an alarm:

- *An event* is a piece of information extracted from continuous or discrete signals (significant variations in a message emitted by the system) or extracted from data about the context (observational data, observations related to the environment, etc.).
- *An alarm* is a discrete indicator emitted by the monitoring system on the basis of events; it is intended to trigger a human or automated reaction.

Alarms can be determined either directly by a physical mechanism following simple event detection or by calculation, comparing the observation signal to a reference one. This gives rise to the following problems:

- What should the reference be?
- Should the signal be processed? How many values should be compared?
- What means of comparison should be adopted?

Alarms are mostly intended for the operator and they are processed on-line so that the operating conditions remain as close as possible to ideal ones, allowing for the variability of input data and the natural evolution of the processes.

The purpose of monitoring is thus supervisory control and therefore the interpretation must be done in real- or quasi real-time. In this case, when the operator controls the final decisions to be undertaken, an important issue is to avoid the "cognitive overload" problem characterized by a flood of data, some (or even most) of each being redundant. The aim is hence to "intelligently" organize the information supplied to the operator. Whenever it is possible that decisions are taken with no human intervention and whenever it is important react quickly, interpretation of alarms may as well aim at automatically appropriate reactions to the evolution observed on the system.

Automated interpretation of alarms may have one of several objectives: either to determine the causes of a malfunction and provide explanations to the operator or to predict future behavior of the system so as to assess the degree of emergency of a situation, or to react automatically (by commands and/or physical mechanisms). Determination of causes is similar to the overall process of diagnosis, and therefore calls for various techniques in the field. Among the possible causes, one can distinguish *internal faults* related to equipment faults and *external faults* corresponding to input data that are outside the acceptable range. Usually, it is necessary to provide explanations, and therefore to describe explicitly some underlying implicit causal relations.

2.2 Artificial intelligence techniques for monitoring. Related work.

There is a vast amount of work published on the subjects of monitoring and diagnosis – too much to cover adequately in this section. Our objective here is to focus on the most relevant approaches that have been considered (and influenced) in the process of the design of the Science Quick Look.

2.2.1 Symptom-based approaches

The idea of exploiting knowledge of causes for the purpose of diagnosis is quite natural. A dysfunction can be described in a simply way by relations associating its original

Chapter 2. Monitoring systems

causes (component failures, etc.) to observable manifestations, or symptoms. Having a theory which models this kind of relations, the diagnostic problem consists in using the theory to seek satisfactory explanations of the observed symptoms.

Much of the literature on monitoring and diagnosis describes artificial intelligence methods of associational inference that relate symptoms directly to faults. This encompasses representations based on rules, decision trees, fault dictionaries, and the most recent case-based reasoning solving technique. Each of these methods has proven worth in various diagnostic settings, but at the same time they have several limitations.

Rule-based systems

Traditional rule-based systems have been built by accumulating the experience of the expert in the form of empirical associations – rules that associate symptoms with underlying faults (see, for example, [28, 21, 5]). The problem-solving strategy may be either forward-chaining (data-driven) or backward-chaining (goal-driven) or a combination of both. The data-driven approach is most appropriate for the task of monitoring, where it is important to respond to new readings and alarms, combining multiple pieces of evidence to assess the severity of a problem and provide the operator with its interpretation. The goal-driven approach is most often used in diagnosis where the goal is a diagnostic conclusion and the rules, through backward-chaining, seek supporting evidence for various intermediate and final conclusions. Sometimes the logical processing for a problem requires a combination of backward- and forward-chaining, called opportunistic strategy [5]. It uses forward chaining to draw conclusions from existing data and backward chaining to find data that make it possible to generate.

The rule-based approach became popular, because it permitted easy construction of expert systems by encoding heuristic knowledge in the form of **if-then** and **when-then** rules. This technique permitted rather quick and impressive demonstrations of diagnostic capability and promised that diagnostic coverage could be increased just by adding more rules. The early diagnosis systems, such as MYCIN in [20], used the heuristic knowledge for making diagnoses. However, several limitations became apparent as rule-based systems were applied to increasingly large and complex systems:

- The task of knowledge engineering (i.e., extracting experienced knowledge from experts and representing it in an appropriate form) is widely acknowledged to be the bottleneck in the building of new expert systems. The representation of the heuristics included both domain and control knowledge – a mix that becomes unmanageable when the knowledge base becomes large and necessary to revise. As the system under study becomes larger and therefore more complex, the task of knowledge engineering grows too.
- There is no guarantee that two faults (or more) that are individually diagnosable will not interact in ways that mask any or all of the symptoms. A rule-based

system may be validated on a set of test cases but still have important gaps in its knowledge base.

- Rule-based systems have no (or very little) predictive power. They cannot show what will happen if a fault is left unrepaired. Similarly, it is difficult to express and reason about temporal information, such as the evolution of symptoms of a fault.
- Small changes in the design of the system may necessitate revisions in a large part of the rule-base. So, once implemented, the systems are difficult to maintain.

Decision Trees

Decision trees provide a guide to diagnosis in a way that they write down the sequence of tests leading to a diagnostic conclusion. A decision tree is a hierarchically arranged semantic network. It is composed of nodes representing goals and links that represent decisions or outcomes. A major advantage of a decision tree, therefore, is to verify logic graphically in problems involving complex situations that result in a limited number of actions.

Decision trees are typically built manually by engineers using detailed knowledge of the system's design and its known failure modes. Such decision trees may contain not only diagnostic steps but also recommended control actions to ensure system safety.

As Davis and Hamscher [21] point out, the simplicity and efficiency that is a strength of decision trees is also an important weakness: they are a way of writing down a diagnostic strategy, but offer no indication of the knowledge used to justify that strategy. Decision trees thus lack "transparency" and are therefore difficult to update (small changes to the mechanism may require a major restructuring of the tree). Like other methods, decision trees have no predictive power to reveal the propagating effects of a fault.

Fault dictionaries

A fault dictionary is a list of symptom/fault pairs, indexed by symptom. The dictionary of the monitored system must be built for every kind and combination of faults anticipated. To do this a description of how the entire system would behave if specific components were broken in a specific way should be generated. This generation can be done by the simulation of the monitored process. The result is a list of fault/symptom pairs, which is then inverted to form a dictionary of symptom/fault pairs. An examples of this technique can be found in [63].

To an extent, fault dictionaries are forerunners of a model-based approach (see below) and overcome some of the limitations associated with rule-based systems, namely: the dictionary does not depend on experience, it can expose the effects of interacting faults, it can be regenerated mechanically if the design changes, and it is likely to cover more

Chapter 2. Monitoring systems

fault scenarios because of its systematic (though not exhaustive) treatment of the kinds and combinations of faults.

The idea behind fault dictionaries – systematic generation of symptom/fault associations – is good, but the technique has some practical limitations:

- In a single entry of the dictionary the symptoms are essentially a predicate on a snapshot of the system. This kind of matching fails to exploit temporal continuity in the evolving manifestations of a fault (this approach can be made vital by getting the right hypothesis and refuting incorrect hypotheses).
- Only the most likely failure modes of each component are considered, and therefore the combination of faults is also severely limited.

Case-Based Reasoning

A relatively recent problem solving technique is the so-called case-based reasoning (CBR). It is attracting increasing attention because it seems to directly address the problems outlined above.

A CBR technique, the origin of which is widely held by the work [59], solves new problems by adapting previously successful solutions to similar problems. It does not require an explicit domain model and so elicitation becomes a task of gathering case histories; the experience does not need to be completely understood, only recorded, the implementation is reduced to identifying significant features that describe a case; CBR systems can learn by acquiring new knowledge as cases, thus making maintenance easier. This technique in the case of monitoring of the behavior of complex physical systems, however, has also some disadvantages, including:

- It is necessary to build a library of cases before the system will be useful.
- Some values may be more significant in some contexts than others. Unless the user specifies the particular importance of those values, they will be treated with the same importance as the rest of the values.
- Cases require interpretation, as a CBR reasoner does not necessarily provide the "correct" solution for a problem; it merely provides suggestions of possible solutions.

2.2.2 Model-Based Reasoning

In response to the problems in symptom-based systems, a new type of diagnosis system was developed that did not include empirical knowledge about symptoms and diagnoses, but relied on principle knowledge of the domain. A diagnosis system of this type is called model-based, because it has a model of the diagnosis object, which describes the object (see, for example, [23] and [18]). "The model-based reasoning (MBR) process itself

can be viewed as the symbolic processing of an explicit representation of the internal workings of a system in order to predict, simulate, and/or explain the resulting behavior of the system from the structure, causality, function and behavior of its components” [69]. In other words, model based diagnostic reasoning is an interaction between prediction and observation [21]. Thus, the first step in MBR is the building of an accurate representation (model) of the system that you want to be able to reason about. Once there is a complete model and an algorithm which correctly reproduces the relationships and behavior of the system, one can get complete, precise solutions based on a set of inputs. This technique allows both simulating the behavior and making inferences about the values of unmeasured variables. By matching versus predictions rather than symptomatic patterns, model-based diagnosis also avoids combinatoric problems.

The principles of model-based diagnosis are well understood. The diversity of work in this field owes largely to the various types of models, representing many different degrees of abstraction. Our coverage of this field within this framework therefore attempts only to be representative rather than exhaustive.

Modeling for Troubleshooting (Davis & Hamscher)

Model-based diagnosis, as commonly described in the AI community [21], requires models of structure and behavior of the system/mechanism under study. One might assume that an adequate structural model is simply a complete hierarchical description of subsystems and components, and that an adequate behavioral model is a detailed simulation model that predicts all the time-varying events/changes in the mechanism. While such models do support model-based diagnosis, they are also part of the problem. As Hamscher emphasizes, the important thing about developing a model for troubleshooting is not that it uses abstractions to deal with complexity (any representation does that), but that it embodies structural and behavioral abstractions appropriate to troubleshooting.

What sort of abstractions are helpful in troubleshooting? Hamscher identifies eight principles to guide a knowledge engineer in constructing a model that makes troubleshooting feasible. Although Hamscher’s research concerns digital circuits (a form of discrete-event dynamic systems), the principles are general enough to be applied, in most cases, to the continuous-variable dynamic systems.

These principles separate into three categories: **behavior, structure, and failures**. We show the eight principles here to give interested readers a fuller understanding of Hamscher’s insights, and we recommend his article for explanations of each.

Modeling of Behavior

- The behavior of components should be represented in terms of features that are easy for the troubleshooter to observe.
- The behavior of components should be represented in terms that are stable over long periods of time or that summarize much activity into a single

Chapter 2. Monitoring systems

parameter. This is easiest for a component for which changes on its inputs always result in changes on its outputs.

- A temporally coarse behavior description that only covers part of the behavior of a component is better than not covering any at all.
- A sequential circuit should be encapsulated into a single component to enable the description of its behavior in a temporally coarse way.

Modeling of Structure

- Components in the representation of the physical organization of the circuit should correspond to the possible repairs of the actual device.
- Components in the representation of the functional organization of the circuit should facilitate behavioral abstraction.

Modeling of Failures

- An explicit representation of a given component failure mode should be used if the underlying failure has high likelihood.
- An explicit representation of a given component failure mode should be used if the resulting misbehavior is drastically simpler than the normal behavior of the component.

Qualitative models

The behavior of a physical system can be described by the exact quantitative values of its variables at each time instant. Qualitative physics is an alternative physics in which the insight concept of the system is defined within a far simpler qualitative basis, but nevertheless formal [22].

Qualitative physics provides a broad integrated and formal account of behavior – an account rich enough to enable intelligent systems to perform tasks such as diagnosis, analysis, explanation and even simulation. Fig. 2.2 illustrates the approach of qualitative physics contrasted with conventional(quantitative) physics. Both start by modeling physical situations, both end with a qualitative common sense description of the behavior. The first step in the conventional approach is to formulate and solve the differential equation to obtain a solution. The second step is to interpret this solution to obtain common sense description of the behavior. The qualitative analysis begins by formulating qualitative differential equations and then solves these. The result is a similar common sense description of the behavior and causal explanation for that behavior which is obtained "more simply".

Unlike quantitative variables, qualitative variables can only take one of a small number of values. Each qualitative value corresponds to some interval on line of the real

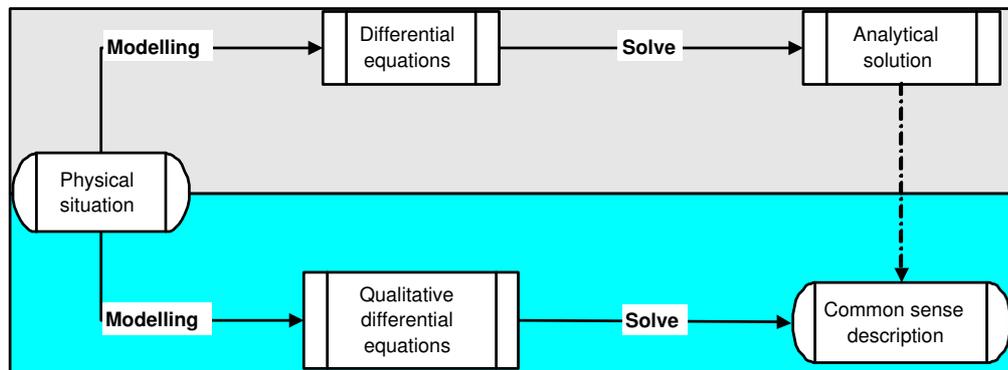


Figure 2.2: Comparison of qualitative and quantitative approaches.

numbers. *Landmark values* demarcate the boundaries between qualitatively distinguishable intervals. The most common landmark is zero, in which case three important qualitative values are positive, negative, and zero or, for a derivative, whether a quantity is increasing, decreasing, or constant.

The most commonly used qualitative simulator QSIM [48], that has been used as the core of a monitoring system called MIMIC [25], allows for implementing this type of models. The QSIM-style constraint model describes the physical device as a set of components. A set of QSIM qualitative states characterizes the device's observed faulty behavior. Two phases can be distinguished in MIMIC: monitoring and diagnosis, both of which call QSIM. The concepts of monitoring, diagnosis and simulation are associated in a cycle of hypothesis/model building/simulation/comparison. MIMIC uses a library of qualitative models including one model of normal and failure models. The fault model whose predicted behavior matches the faulty behavior then determines the set of faults in the device.

The approach based on qualitative simulation is well suited to the needs of monitoring tasks with the a high-level of abstraction where the qualitative formalisms are particular useful in representing imprecise, uncertain or incomplete knowledge.

Inc-Diagnose

In 1987 Reiter proposed a formal theory of diagnosis from first principles that reasons from system descriptions and observations of system behavior [55]. The algorithm computes all minimal diagnoses of a device, including multiple-fault diagnoses. In contrast to an abductive approach to diagnosis in which the hypothesized faults must imply or explain the symptoms, Reiter's theory requires only that a diagnosis be consistent with the system description and observations; no notion of causality is needed. However, the theory was applied only to diagnosis of digital circuits, which are representative only of physical devices having discrete, persistent states. Reiter's theory is not in conflict with

Chapter 2. Monitoring systems

the well-known model-based diagnosis work of researchers such as Davis & Hamscher [21] and deKleer & Williams [23]; rather, it gives a stronger theoretical foundation for that work.

In 1990 Ng [67] extended Reiter's algorithm to diagnose dynamic continuous devices of the kind modeled by QSIM. Since QSIM represents continuous behavior as a finite number of discrete qualitative states, it allows the diagnosis problem to be transformed from a continuous one to a discrete one. Ng's algorithm, named Inc-Diagnose, makes Reiter's approach incremental in that it permits measurement-taking at different times, intermixed with hypothesis generation. Like Reiter's approach, Inc-Diagnose can diagnose multiple faults using only a correct model of the device; Inc-Diagnose does not use fault models (see MIMIC) and it cannot predict the effects of faults.

2.2.3 Fault detection and isolation

The main objective of the fault detection and isolation (FDI) research area, widely addressed from several points of view since the early '80s, is to study methodologies for identifying and exactly characterizing possible incipient faults arising in a plant. As commonly referred to in the literature on this topic (see [17]), the term fault detection is used to describe the problem of making a binary decision either that something has gone wrong or that everything is fine. This is usually achieved designing a dynamical system (usually referred to as a *filter*) which by processing input/output data is able to detect the presence of an incipient fault and eventually to precisely isolate it generating the so-called residual signals. There are different ways which can be adopted to detect and isolate the fault. A common way, in some industrial applications, is to use *hardware redundancy*, which consists of comparing the outputs of identical hardware and perform consistence cross checks. As it can be easily realized, this approach is not always feasible in practice due to an economical impact. A solution of this kind may have a cost of the another whole system.

Using a different approach, known in FDI literature as *analytical redundancy*, the FDI task can be accomplished using analytical and functional information about the system using a mathematical model of the system (*model-based FDI*). An analytical redundancy relation (ARR) expresses a constraint among possible observations that holds when the system is working correctly. The FDI structure is the minimal set of components whose models have to be satisfied for the ARR to be satisfied. If some ARR relation does not hold, then the system is faulted and the degree to which the relations are violated are the residuals.

2.3 Summary

The primary goal of a diagnostic system is to detect anomalous system behavior and then to isolate the cause for such a behavior. One thing is very clear from a study

of the literature – there is no universal method, which could be able to cope with all kinds of systems and all possible faults. Therefore it is important to set clear diagnostic requirements beforehand, i.e. types of faults, performance attributes, and available process knowledge should be clearly identified. A model-based approach seems to be most appropriate when tackling problems in monitoring and control of a complex system such as spacecraft when the description from the first principles of the domain is more preferable. The basic concept of this approach is to use a model to predict expected behavior, and then using discrepancies between predictions and observations as diagnostic clues. This approach is rather natural for solving the problems in science, as it is the scientist's aim to investigate and describe the structures and mechanisms underlying physical processes.

There are two classes of model-based approaches:

- In the first class quantitative models are used. These approaches are based on the philosophy that a fault changes the physical parameters and hence the model of the supervised system. These methods make use of theoretical tools such as parameter estimation techniques, state estimation techniques and parity space concepts. In this approach the need for an *a priori* knowledge about the relationships between faults and changes in model parameters and states is a key point.
- The second approach is based on the use of qualitative reasoning and modeling, that are used to predict the behavior of the system in nominal conditions and in different faulty condition and the isolation of the fault is performed comparing the predicted and the actual behavior.

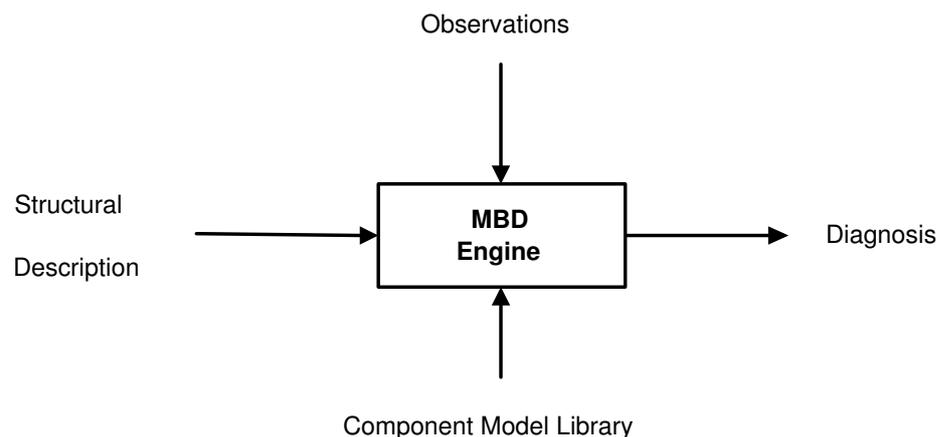


Figure 2.3: Basic perspective of MBD [24].

The concept of model-based approach, called model-based diagnosis (MBD), has evolved in two separate communities with their own terminologies and conventions:

Chapter 2. Monitoring systems

the model-based reasoning (MBR), a specialty within the AI community and the fault detection & isolation (FDI), a specialty within engineering community. Nowadays, there is an attempt to present the fundamental concepts of MBD in one consistent terminology and thus to make the field much more accessible. The authors of [24] illustrates the basic properties of MBD, discuss the differences in formal definitions, conflict treatments and basic perspective of MBD. In [24] is noted that the main characteristics of the MBD approach are an emphasis on general diagnostic reasoning engine that perform a variety diagnostic tasks, and inference of a system's global behavior from the automatic combination of local models of its components. This is illustrated in figure 2.3 where the component library used by an MBD engine describes the laws which govern the behavior of the components.

The model-based diagnosis is an active area of artificial intelligence research and it has been approved as a useful technique to monitor and automatically diagnose of spacecraft failures with a large number of components. In [14] and [70] model-based systems have been made scalable enough to be applied to systems with up to tens of thousands of components.

Chapter 3

From Requirements to Diagnostic Solutions

Monitoring and diagnostic system design starts with setting up diagnostic requirements and process knowledge.

The objective of this chapter is to set up the requirements of the monitoring system in sufficient detail to enable a rational design. We proceed from general to specific, starting here with the high level-architecture of the ScQL monitoring system and then elaborating on its components in the following chapters.

3.1 Requirements of the Science Quick Look (ScQL)

As in any other engineering design, we start the design of ScQL monitoring system with setting up its diagnostic requirements. The core of every diagnostic solution is the choice of a suitable process model (see fig. 3.1). This is conditioned by available process knowledge, given measurements, and is closely related to the requirements. The level of abstraction and accuracy of the model has to be chosen in such a way that given requirements are fulfilled.

As mentioned, a monitoring of various kinds is required to continuously check the correct operation of all aspects of the instrument, both as individual components and as a complete system. The aims of the monitoring may be generally classified by the following main requirements:

- (1) **Monitoring of the on-board instrument hardware,**
- (2) **Monitoring of the on-board instrument software.**

This monitoring is an important part of the whole daily data reduction process (figs. 3.2 and 1.8) and represents *an indirect control* of the work of the on-board hardware and software. It is based on the analysis of the observations (science data) and treated in quasi-real time on ground at the space operation center.

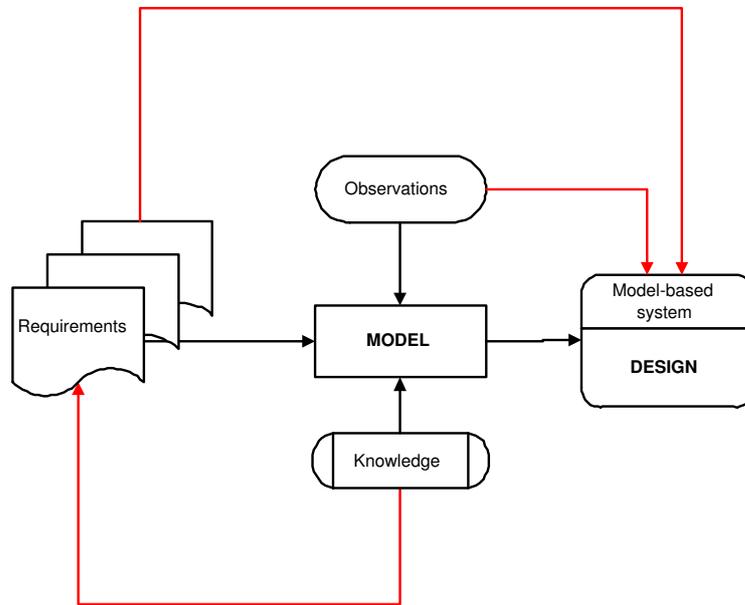


Figure 3.1: A model-based system design.

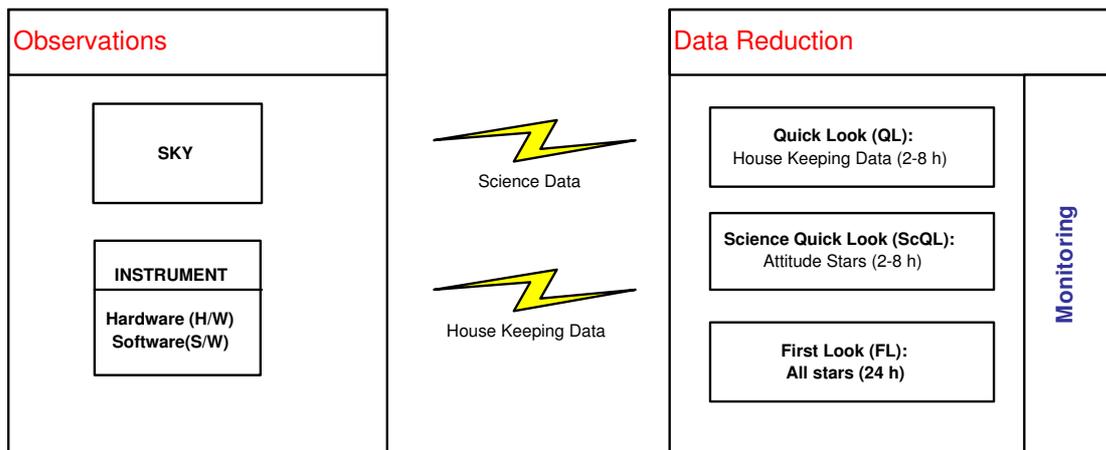


Figure 3.2: Overall scheme of the daily monitoring and diagnosis.

Data sets for the ScQL

Before we start describing the main requirements to the system in details, let us shortly present the data sets for the ScQL monitoring system. These data contain

- I.** The so-called windows (containing star images) are the central observations results produced by the main instrument. There is one window record per on-board detected or predicted star image, plus some special windows (sky background). On average there will be about 100 windows(records)/sec (for SM mosaics), but this varies between about 50 and 1500 windows(records)/sec, because the Milky Way system has a disk-like distribution of stars (i.e. our galactic disk has a higher density of stars compared to the halo).
- II.** Parts of the house keeping data: TDI timing, the values of supply voltages of the CCDs, the values of the thermal status of the CCDs.
- III.** Orbital parameters of the satellite.
- IV.** Attitude control system data: scan speed of the satellite, orientation of the satellite, etc.

A detailed description of the ScQL data sets is presented in Appendix 8.2 and discussed further in Chapter 4.

(1) Monitoring of the on-board instrument hardware

It must be possible to verify whether the hardware of the main instrument of DIVA – instrument processing unit (IPU) – is functioning correctly. This encompasses the optics of the main instrument and detectors with its light sensitive area (CCD mosaic, see figure 1.7) and supply voltages of CCD ADU chains¹.

(2) Monitoring of the instrument on-board software

It must be achieved to verify that the algorithms of on-board software of DIVA IPU is functioning correctly. Some of the following algorithms are still under construction and should be considered as preliminary ones for the tasks of the ScQL, however, they are enough developed to be included and described for the tasks of the ScQL.

- **Detection algorithm on SM1 (see fig.1.7).** The on-board image detection process performed by the instrument of the satellite constitutes a crucial element for the success of the astrometrical mission. The on-board detection ultimately defines DIVA's sky survey – its completeness, its brightness limits and its biases. The primary output of the image detection software is a list of star-like images that

¹ADU – Analog to Digital Unit that corresponds to a half of CCD chip area.

Chapter 3. From Requirements to Diagnostic Solutions

are detected in the input data stream. In [7] the different criteria for the judgment of the scientific performance of the image detection algorithm, to be used in the industrial development of the satellite, are discussed. One of the possible simple algorithms of the on-board detection for DIVA were described in [42, 41]. In [41] was applied a block-skipping method for acceleration of object detection.

- **Centroiding algorithms.** The on-board centroiding algorithm generates the centroids for the objects (images) on SM1, the coordinates of the centroid (μ_c and κ_c) are the coordinates of the centroid in cross-scan and in scan directions correspondingly. There exist two types of centroiding algorithms that define the centroids with a different precision. This precision of centroids is defined by the requirements of the consecutive tasks such as the prediction algorithm, the attitude control system, the rate of the star transit and TDI clock stroke adjustment.
 - **Coarse centroiding algorithm.** This algorithm provides the centroids of the detected object (image) on SM1 for the cutting of the window and prediction of the left-corner of the window on SM2 (there is no detection on SM2)². In this case the mean error of the centroids should be $\Delta_{\mu_c, \kappa_c} \simeq 1$ pixel in both directions because one can only cut the window with an integer number of the left-corner.
 - **Fine centroiding algorithm for attitude stars.** This algorithm provides more accurate centroids than the coarse centroiding algorithm for the objects which are identified as attitude stars. These centroids will be used further by the attitude control system for the following attitude determination, the star transit rate and TDI clock stroke adjustment.

Note: There are several methods (see [34, 35, 36, 9]) to calculate the centroids such as numerical filters, moment methods, gaussian fits and profile fits (the last one, based on the DIVA point spread function (PSF), is the most accurate method to calculate the centroids).

- **Prediction algorithm on SM2 (see fig.1.7).** This algorithm takes the centroids of the objects from SM1 as input and predicts the location of the windows surrounding the objects in the continuous pixel stream of the CCDs on SM2 and zero-order SC. The original algorithm, proposed in [12], was extended in [9] to make it more appropriate for the on-board software logic.
- **Window cutting algorithm.** This algorithm cuts out a small section of the raw CCD pixel data stream, the so called window, surrounding the objects. Only the predicted windows around the objects on SM2 & SC along with detected windows on SM1 are transferred to the ground.

²In fact, there is a very small pixel area on SM2 where the object detection should be performed.

- **Background determination algorithm.** This algorithm determines the signal level of the so-called "empty sky". The optimum way to do that in an astronomical context uses a median of the signal over extended sky areas. However, it is not possible to implement it in the on-board software, instead one will use groups of four consecutive pixels in a line to infer the background levels (see [9] for more details).
- **TDI clock stroke rate adjustment algorithm.** Due to the rotation of the satellite the stellar image are moving from left to the right in the focal plane. So, all CCDs should be clocked synchronously with the actual rotation of the satellite. They are operated in time delay integration mode and therefore have an integrated exposure time per CCD transit equal to $t_{\text{exp}} = 1.4$ sec in nominal mode. The actual rotation rate, nominally $\omega = 180''/\text{sec}$, must be determined in real time to an accuracy better than $\Delta\omega \leq 0.1''/\text{sec}$ from the derived rate of the star transit. If the shifting of the charge on the CCD does not match the actual rotation of the satellite, but this mismatch is not yet large enough to initiate the jet-firing, then a new TDI clock stroke rate must be determined which which is synchronized with scan-speed of the satellite. We propose two methods for the TDI clock adjustment from the star rate determination which can be used for the on-board TDI clock stroke rate adjustment and for on the ground control of this process in the framework of the ScQL monitoring system as well.
- **Attitude star catalogue handling.** The attitude star catalogue consists of reference stars, i.e. stars with known positions, magnitudes, proper motions and colors. These stars play an important role in the on-board attitude determination and TDI clock stroke adjustment. The complete algorithm for the on-board attitude star handling is described in [6] in general, especially designed for astronomers. But one can use the approach of [9] with a more precise description of each step of the attitude star catalogue handling that includes:
 - *Creation of the working catalogue.* It is constructed to reduce CPU time during observations and is prepared in the following steps:
 - * Calculation of the nominal scanning law attitude;
 - * Selection of all attitude stars for the next several rotation of the satellite;
 - * Computation of reference great-circle coordinates for all selected attitude stars;
 - * Transformation of the cartesian vectors into RGC longitude and latitude;
 - * Sorting stars along reference great-circle longitude.
 - *Identification of attitude stars.* With the aid of the working catalogue the identification of attitude star transits is performed according to the following steps. To reduce computational load further, only detected images which are brighter than ≈ 12.5 magnitude are considered as possible candidates for attitude stars.

Chapter 3. From Requirements to Diagnostic Solutions

- * Computation of field coordinates from pixel coordinates by applying geometric calibration;
 - * Rotation of the field coordinates into the body-fixed satellite system;
 - * Computation of the attitude matrix for the transit time;
 - * Computation of heliocentric ICRS coordinates by removing aberration;
 - * Rotation into reference great-circle coordinates and computation of RGC longitude and latitude;
 - * Computation of differences in longitude, latitude and identification.
- *Transfere to the attitude control system (ACS)*. After a transiting star is identified as an attitude star the differences between the measured and expected field coordinates are delivered to ACS.
- **Guest observer object handling.** The on-board guest observer object handling aims to predict guest observer object positions for SM1 based on the working catalogue, SM2 transits are then predicted from these SM1 by means of prediction algorithm. The procedures applied for attitude star handling are also suitable for the on-board guest observer handling with some minor differences (see [9] for details).
 - **Magnitude determination and cosmic identification.** Besides stellar positions DIVA provides photometric (and spectro-photometric) information which will be carefully determined with the on-ground reduction software. However, the on-board software will perform an estimation of magnitude for all objects for the following reasons [9]:
 - The down-link capacity of the telemetry is limited so that only a given number of pixels (windows) on the CCD can be used. Therefore, stars below a brightness limit ~ 15.5 magnitude have to be ignored.
 - Only for the brighter stars spectro-photometry will be performed, otherwise no SC window for these objects will be transferred to the ground.
 - The size of the SM windows for each object to be downloaded to the ground is determined from the magnitude estimation:
 - * $[0 : 12] \times [1 : 7]$ pixels – size of the normal window (including background, index 0);
 - * $[0 : 12] \times [1 : 17]$ pixels – size of the bright star window (including background, index 0) for stars brighter than 8 magnitude;
 - * $[1 : 28] \times [1 : 11]$ pixels – size of the guest observer object window;
 - A cosmic particle detection algorithm must be developed, so that no windows will be downloaded for areas around cosmics.

- **Compression algorithm.** As we discussed above, the data transmission rate of DIVA satellite is ~ 700 kbit/sec, therefore not allowing to download the whole CCD frames to the ground station. So, one uses the strategy that permits to transfer only the parts of the CCD mosaics, windows, that contain the necessary scientific data. However, even with this cutting the data rate could sometimes exceed the transmission rate. So, a simple loss-less data compression algorithm is desired in order to download as many scientific data as possible. For this purpose, in [9] a simple algorithm was proposed and tested. It allows to achieve the compression rate of almost a factor of 2 and the coding is simple and fast, so at the moment this algorithm is considered to be reasonable robust for the on-board data compression.

3.2 The design of the ScQL

A model-based approach for the ScQL

The task of the ScQL monitoring system is further complicated by the fact that the behavior of individual components can evolve over time and, in some cases, changes are due to the activity of other components. Thus, alarms are not independent, i.e. some are merely the consequence of the others. The distinguishing feature of the ScQL monitoring is **to collect, analyse and interpret the science data, but not only the house keeping data**. So, the analysis and interpretation of the science data is a clue to the fault detection of the components of the DIVA main instrument. The analysis of the DIVA instrument behavior can be viewed as an interaction between prediction and observations of the science data. By matching the observations against predictions rather than symptomatic patterns, model-based technique provides the simulation mechanism to cover a very broad collection of possible faults and will allow therefore to make indirect inferences about the behavior of components of the DIVA instrument.

The diagnosis of the science data is thus based on the modeling of the behavior of the DIVA with its observing scanning strategy.

There is no other way to provide the ScQL monitoring with the predicted observations (predicted science data) as to model them quantitatively. A purely qualitative model is absolutely inappropriate for this purpose because we have to keep the diagnostic accuracy and precision as high as possible to satisfy the observation requirements of DIVA as described in [11]. However, it does not mean that we will not make simplifications in our model for the science data simulation and therefore for the data analysis and diagnosis. The purely qualitative descriptions, applied usually for a qualitative simulation, when the values of variables are described in terms of their relations with a finite set of symbolic landmark values. However, it can be augmented with semi-quantitative knowledge in the form of real bounding intervals and real-valued functions. We will combine this semi-quantitative approach, that helps us to describe the behavior

Chapter 3. From Requirements to Diagnostic Solutions

of some components of the system considering only the high level of abstraction, and the quantitative model in order to derive the diagnostic ability of the ScQL monitoring. Therefore,

our approach is to combine all necessary pieces of knowledge from different abstraction levels in order to cover iteratively the task of the fault detection by the analysis and interpretation of the science data in the framework of the ScQL monitoring.

Indeed, to keep the description of some complex components of the instrument at the high level of abstraction is rather reasonable at this stage of the project. It permits to create an extended skeleton of the ScQL monitoring and make the simulation traceable in spite of the lack of the detailed qualitative model of the behavior of these components.

Schematically, the whole process of our monitoring system can be represented by three functionally different parts (see fig. 3.3):

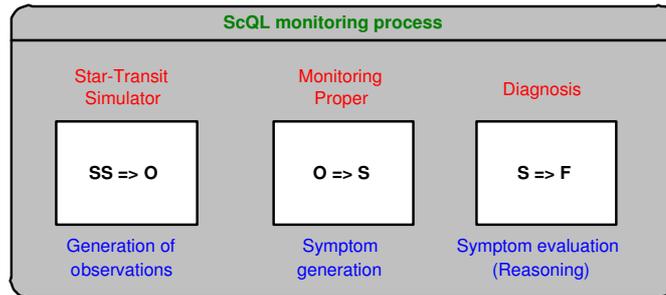


Figure 3.3: The ScQL monitoring parts: SS – Sky (Galaxy) & Satellite models, O – Observations, S – symptoms, F – faults.

★ **Star transit simulation** is used as a core of the ScQL monitoring to model the behavior of the satellite and simulate the observations by the quantitative and semi-quantitative model approach. It is described by the functional behavior of the system according to the chosen accuracy and abstraction level. This description is based on the model of the Galaxy, the structure of the DIVA satellite and its observation strategy. The star transits simulator provides the ScQL monitoring with astrometrical science data, i.e. with the data from the Sky Mapper CCD mosaics.

It is worth to mention that we consider the star (space object) as a point source on the sky and on CCDs, so we simulate the star transits in the form of suitable coordinates, centroids (see chapter 4 for details), instead of pixel images. By this approach we can generate directly the image parameter set – the output from

the *initial data treatment* (see on fig. 1.8) and avoid an explicit implementation of the first step of the scientific data reduction chain that consists of pixel data processing.

- ☆ **Monitoring proper** estimates statistics and parameters from the observations (simulated science data) and generates the symptoms (residuals) via analytical redundancy. The statistics and derived parameters are compared to their expectation values and if the findings differ significantly from the nominal ones, the system rises alarm flags and generates appropriate symptoms.

The monitoring proper performs updating the "monitoring set" composed of those parameters and their tolerances which are appropriate for the present state of the satellite.

- ☆ **Diagnosis** uses the symptoms, generated by the monitoring proper, to perform their evaluation in order to debug the abnormal (fault) situations. We define the fault as any disturbance of the observations at the scientific level of the precision. So, this third part gives the relation between the symptoms, that are represented through statistics and estimated parameters of the instrument and faulty states of the system.

The monitoring and diagnosis are both based on the model that is explicitly implemented in the star transits simulator. The role of the monitoring and diagnosis task is to generate and evaluate the symptoms derived from the analysis of the generated dataflow and to present this analysis to the operator to assist in interpreting it and diagnosis. The alarms which indicate either dysfunction or significant events are produced in the quasi-real constraint, i.e. every 15 – 30 min a new set of alarms is generated. This discrete-time constraint allows us to leave aside the problems related to real-time monitoring systems when propagation times of events, alarms and transmission paths must be taken into account. It is worth to mention that in our study **we focus on the tracking and diagnosis aspects of the ScQL monitoring system not on the advising aspect**. So, we do not develop in this work a final advising system based on diagnostics. This should be the task for future research.

Faults and residual generation

We defined above the fault as an unexpected (unpermitted) deviation of at least one characteristic property of a variable from an acceptable (nominal) behavior of the satellite. Therefore, the fault is a state that leads to a malfunction of the system. The time dependency of faults can be distinguished, as shown in figure 3.4, as abrupt faults (stepwise), incipient faults (drift like), intermittent faults. An abrupt fault appears as a quick increase in the observed symptoms, a sudden breakdown of the component, e.g. ADU chain. Incipient fault gradually develops as a function of time (slow degradation of components).

Chapter 3. From Requirements to Diagnostic Solutions

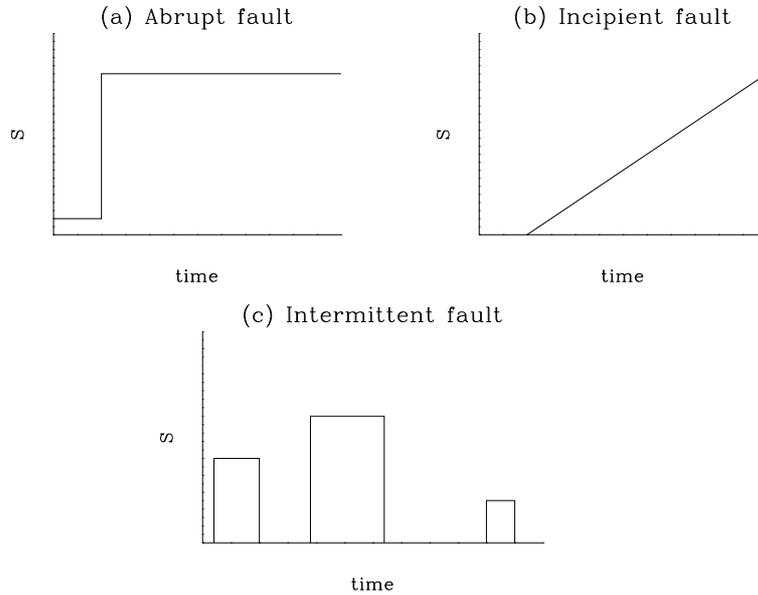


Figure 3.4: Three typical classes of the time-dependency of faults: (a) abrupt fault, (b) incipient fault, (c) intermittent fault.

Detection of faults in ScQL is based on symptoms/residuals that are reflected the changes of observable quantities from their normal behavior. Normally, the consistency check based on analytical redundancy (see Chapter 2) is achieved by comparing the available signals (known inputs and measurements) and a model of the system. A cross-check of these signals is carried out to detect faults. The resulting difference for one signal is referred to as residual signal:

$$r_i(t) = y_i(t) - \hat{y}_i(i), \quad i \in \mathbf{k},$$

where r_i denotes the i^{th} residual, y_i the i^{th} measured system output, \hat{y}_i the estimated i^{th} system output and k the number of residuals. The goal is to generate structured residuals to meet monitoring and diagnosis requirements. More generally, the generation of residuals is performed by the estimation of the system measurements, output vector \mathbf{y} , or of the system parameter vector θ . Then the estimates $\hat{\mathbf{y}}$ and $\hat{\Theta}$ are subtracted from the real observations \mathbf{y} and/or from the system parameters Θ . This leads to the following residual vectors:

$$\mathbf{r}_y(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t) \quad \text{or} \quad \mathbf{r}_\Theta(t) = \Theta_{\text{nom}}(t) - \hat{\Theta}(t)$$

This residual vector $\mathbf{r}_y(t)$ is typical for the observer-based approach, whereas the $\mathbf{r}_\Theta(t)$ corresponds to the parameter estimation approach. Residuals are designed to be equal or converge to zero in the fault-free case ($r_i(t) \approx 0$) and deviate significantly from zero

under occurrence of a fault ($\|r_i(t)\| > \eta_i > 0$, where $\eta_i \in \mathbb{R}$ denotes a threshold). Hence, the residuals represent the fault effects.

In our monitoring system the parameter estimation approach of residual generation (based on different statistics of the science data) is chosen. As for the observer-based approach is concerned it may be apply to the house keeping data, the residuals of which might be derived directly from the output of the satellite. But we will not consider this in our study, and so in the following and in the rest of the thesis only the parameter estimation approach is considered in details.

Residual evaluation and residual structure

A successful residual-based approach requires an appropriate residual evaluation in order to take the following decisions: 1. Is there any fault present?, and 2. If yes, which fault(s) is/are present. Especially the second decision is depending on the fact whether only single faults (one fault at a time) or also multiple (simultaneous) faults are considered. The problem of handling multiple faults lies in the fact that resulting faults effects caused by single faults occur at the same time and hence, they might compensate each other the might overlap in a way that either only one of them or a complete other fault is detected.

Therefore, it is important to obtain the *correct residual structure* for correct residual evaluation. The residuals should be generated in such a way that for each fault a different set of residuals is affected (i.e. the residuals deviate significantly from zero). For multiple faults it should furthermore be guaranteed that the overlapping of the resulting fault effects does not lead to the wrong decision, i.e. missed detection of a fault or a wrong decision about which fault occurred.

A structured residual can be characterized by the following property: *Any residual responds only to a specific subset of faults, and to any fault only a specific subset of residual responds.* Following [30] one can represent a set of p residuals in in two different ways:

- in a geometric way by considering the vector

$$\mathbf{r}(t) = (r_1(t)r_2(t)\dots r_p(t))^T, \quad \text{where } r(t) \in \mathbb{R}^p.$$

- in a Boolean way by defining a *fault code vector* in the following way:

$$\epsilon_i(t) = \begin{cases} 1, & \text{if } \|r_i(t)\| \geq \eta_i; \\ 0, & \text{if } \|r_i(t)\| < \eta_i \end{cases}$$

$$\Rightarrow \boldsymbol{\epsilon}(t) = (\epsilon_1(t)\epsilon_2(t)\dots\epsilon_p(t))^T$$

for $i \in \mathbf{p}$ and the thresholds η_i .

Chapter 3. From Requirements to Diagnostic Solutions

Obviously, the fault code vector $\epsilon(t)$ provides the information whether the i^{th} residual $r_i(t)$ hits a defined threshold η_i or not. In the Boolean notation one can also define a *structure matrix* Φ in the following way, when using the fault vector $\nu(t) = (\nu_1(t)\nu_2(t)\dots\nu_k(t))^T$:

$$\mathbf{r} \leftarrow \Phi \boldsymbol{\nu}$$

where the j^{th} column vector of Φ is defined as: $\Phi^j = \epsilon^{\nu_j}$, where ϵ^{ν_j} describes the code vector ϵ concerning the j^{th} fault ($j \in \mathbf{k}$). Hence, Φ is a $p \times k$ matrix that contains only ones ('1') and zeroes. The operator defined by ' \leftarrow ' must be read as 'is affected by'. The expression $r \leftarrow 0$ is a special case and should be read like the residual r is not affected (by any fault).

To illustrate this notation a simple example is given representing the structured residual where the i^{th} residual $r_i(t)$ is only affected by the i^{th} fault $\nu_i(t)$, and the number of faults k equals the number of residuals p . If $k = p = 3$ then

$$\mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \leftarrow \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\Phi} \underbrace{\begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix}}_{\boldsymbol{\nu}}$$

which can be read as residual r_1 is affected by fault ν_1 , r_2 is affected by fault ν_2 , r_3 is affected by fault ν_3 .

In our approach we will use this simple residual structure in order to generate alarms, i.e. in our case the number of alarms equals the number of residuals. These alarms are produced inside of the monitoring proper. As it was described, we compare the calculated statistics with expected ones and if the findings differ from nominal, i.e. when the residuals exceed the threshold, appropriate alarms rise. This intermediate simple evaluation of the symptoms, that gives the one-to-one relations (from statistics/residuals to alarms), is done to quickly inform an operator that a certain statistics shows that something is wrong with the science data. Then the consequent diagnosis (with many-to-many relation) for fault identifications and isolations will be started in the diagnosis module of ScQL. It may also happen that the ScQL monitoring will not be able to evaluate some problems properly at the real observation phase. In this case the signature of the problem, written down as an alarm into the database, can be retrieved together with appropriate statistics when a more detailed data analysis and diagnosis (First Look) starts.

In the notation given above different kinds of structural residuals can be represented using corresponding structure matrices Φ (see e.g. [30, 17]). In [30] a detailed discussion is given concerning the design of structural residuals. It provides useful definitions that we will also use in this work:

- ◆ **Undetectability in a structure:** A fault ν_j is undetectable in a residual structure if its column ϵ_j in the structure matrix Φ contains only zeros.

- ◆ **Indistinguishability in a structure:** Two faults are indistinguishable in a structure if their respective columns in the structure matrix are identical.

In this thesis the following general description of an efficient residual structure as introduced by [50] will be used:

In the j^{th} fault mode (i.e. when the j^{th} fault occurs; the fault $\nu_j(t) \neq 0, j \in \mathbf{k}$), the residuals $r_i(t)$ for $i \in \Omega_j$ are nonzero, and the residuals $r_\alpha(t)$ for $\alpha \in \mathbf{p} - \Omega_j$ decay asymptotically to zero. The specified family of coding sets $\Omega_j \subseteq \mathbf{p}, j \in \mathbf{k}$, is chosen such that, by knowing which of the residuals $r_i(t)$ are (or decay to) zero and which are not, the fault ν_j can be uniquely identified.

We will use the different coding sets Ω_j to identify the occurring faults. A coding set contains a set of numbers that represents a specific subset of residuals $r_i(t), i \in \mathbf{p}$, i.e. $\Omega_j \subseteq \mathbf{p}, j \in \mathbf{k}$; where k denotes the number of faults and p the number of residuals. In case of single faults the following holds: if the complete set of residuals defined by the coding set Ω_j is affected by an occurring fault it can be said that the occurring fault is the j^{th} fault. For multiple faults extra conditions have to be fulfilled to avoid overlapping or cancellation of faults effects. The effects from multiple faults (e.g. ν_1 and ν_2) might add up in a way that leads to a wrong decision: when $\Omega_1 \cup \Omega_2 = \Omega_3$ the fault ν_3 would be detected instead of ν_1 and ν_2 . The simplest coding set that handles both multiple and single faults would be $p = k$ and $\Omega_j = \{j\}$; in that case the j^{th} fault would only affect the j^{th} residual.

3.3 Summary

The ScQL monitoring focuses on the tracking and diagnosis aspects by means of the model-based simulation and analysis. The model for the ScQL can be considered as a combination of the quantitative model with semi-quantitative elements, that allows to use different abstraction levels to make the simulation and analysis feasible at this stage of the project. The model-based ScQL monitoring is based on the model that contains

- ★ the model of the Galaxy;
- ★ the model of the structure of the DIVA satellite;
- ★ the model of the behavior of DIVA components;
- ★ the model of the scanning strategy of the satellite.

Obviously, the exact and complete 'models' can not be reached. This has different causes, namely: the uncertainty in the structure of the Galaxy, different noise effects,

Chapter 3. From Requirements to Diagnostic Solutions

disturbances and uncertainty in time-varying system parameters. The model uncertainties must be carefully estimated with the help of the "nominal simulation"³ to set up modeled time-varying thresholds $\eta(t)$ for the residual generation and evaluation phases.

³The simulation is called "nominal" if it is performed with the nominal parameters, i.e. in fault-free state.

Chapter 4

Star transit simulation

The initial and the crucial part of our system is the DIVA simulator that generates the science data of DIVA satellite as input to the ScQL engine system. It guides the development of appropriate statistics and residual structures in monitoring and diagnosis part of our system.

The complete model of the system is complicated, and involves a large number of quantities and features, some of which demand special studies. As a complete, realistic simulation of the observations of DIVA is beyond the scope of this investigation, we shall define therefore appropriate simplifications in the model of our simulator and provide the ScQL monitoring with artificial science data which reflect all important features of the DIVA main instrument.

As it has been mentioned above DIVA should have produced a raw data (from the whole CCD mosaic) on board of about 6.5 Tbit per day (95 Mbit per sec, 19 hours of observations) from which 48 Gbit (~ 700 kbit per sec) are transmitted to the ground. A realistic simulation of such data sets, describing various *faulty situation*, for a considerable amount of "mission life-time" is not possible within this study, but it is also not necessary as we shall show below. In fact, in order to develop and evaluate the ScQL monitoring system we can simulate quite an acceptable number of faults by our approach, called star transit simulation.

The major simplifications we make are the production of *star transits (crossings)* in the form of suitable coordinates instead of the simulation of actual pixel values obtained after converting of the ADU of the CCDs. These star crossings provide us with the exact position of the centroid of the stellar image within one pixel in the CCD output data stream. From this knowledge we can simulate the quantities that the instrument software will produce, and which will be transmitted to the ground. The star transits

Chapter 4. Star transit simulation

are developed for the Sky Mapper CCD mosaics and provide the position of the star for along scan and across to scan directions as a function of magnitude by simulating t for DIVA. From the simulated centroids we will be able to simulate the lower-left corner of the star window to be transmitted to ground and any other parameters from *image parameter set* (IPS).

IPS is the main output of the pixel data processing task. IPS record contains parameters of an actual image of an astronomical light source recorded in the pixel values of a window. Such parameters are: centroid position (in pixel coordinates), brightness (amplitude, integrated photo-electrons or ADC units), shape parameters (diameter, duplicity, diffuseness...), diagnostics (disturbed by cosmic ray, badly positioned in the window, truncated by CCD edge, ...). At this stage of the project the complete list of the IPS is still under construction. In our star transit simulator we use the first version at defining image parameter set that is discussed in the following sections of this chapter and summarized in Tables 8.4 and 8.5 in the Appendix.

Moreover, let us ignore technical issues like charge transfer efficiency of the CCDs, possible changes of during the exposure, irregularities of the physical pixel matrix, projection effects from sky to focal plane and etc.

4.1 The Sky

In order to simulate *star transits* through two of DIVA's fields of view one needs a correct representation of the observed environment that will be registered by the detectors the satellite, namely the sky.

The major DIVA targets are stars belonging to the Galaxy (the number of extragalactic objects is negligible at $V < 15$ mag, they should be taken into account at fainter magnitudes [$V > 16$ mag] only). The DIVA satellite scans the stars from inside of the Galaxy and therefore the observed distribution of stars in different directions follows the Galaxy structure.

At different limiting magnitudes, the sky looks differently. At brighter magnitudes, when only nearby stars residing within the galactic disk are observed, the sky is rather represented by an isotropic (independent of the coordinates) stellar density distribution. On the contrary, at faint limiting magnitudes, when the detectors register remote stars (beyond the galactic disk), the *apparent sky* would be strongly anisotropic: the maximum of density objects is then reached in the Galactic plane, and the minimum at the Galactic poles.

In order to build up a correct simulation we define several important entities related to the sky observations. Let us introduce

- ★ *The true sky*: the genuine distribution of stars in spatial coordinates and stellar magnitudes. Thus the true sky can be considered as a sample that is produced according to the adopted model of the Galaxy, and is not subject observation

errors. It is also supposed, that there are no missed objects in the true sky. So, the true sky is a fixed quantity, i.e. neither positions, proper motions nor magnitudes have errors. The true sky is per definition unknown to all applications on-board of the satellite as well as on ground.

- ★ *The observed sky*: this is a sky available to DIVA satellite through its detectors. It depends on the true sky, filtered through the DIVA detectors. The observed parameters of the stars (positions and magnitudes) are different from the true ones producing an imperfectness of the DIVA limits (i.e. signal-to-noise thresholds). As noted before the construction of the more realistic observed sky will be required for the final complete simulation of the space mission. In our investigation, this sky only provides the *star transit* simulation with dependence on the celestial position of the star at a given magnitude, no observation images are produced in the CCDs.
- ★ *The known (catalogued) sky*: this is the present-day knowledge available to the astronomical community and stored in the all-sky catalogues. It is another derivative of the true sky, filtered through the error filters of the different instruments and contaminated with artifacts, non-galactic objects (asteroids or extragalactic objects), it is subject to the data incompleteness and imperfectness of various nature of present-day catalogues.
- ★ *The attitude star catalogue (ASC)*: this is a subset of the catalogued sky, constructed from it according to the DIVA requirements.
- ★ *The simulated sky*: this is a simulation sample of the sky created according to the average stellar coordinate and stellar magnitude distributions. In some sense these distributions are derivatives from the catalogued sky and represent the adopted model of the Galaxy. The simulated sky takes the role of the true sky in the simulation and evaluation process of the ScQL monitoring. It is used in our simulations instead of the *known sky*.

From these sky definitions it is evident that the true sky is used for the simulations of the observations (observed sky) and per definition is unknown to all applications on-board of the satellite as well as on ground. However, the ScQL monitoring system and the data reduction software works with the known sky that is represented here by four other types of the sky: observed, catalogued, ASC and simulated.

Before describing the implementation of the methods of the "sky" generation and the procedure of the star transit simulation itself different coordinate systems are introduced.

4.1.1 Equatorial coordinates; the International Celestial Reference System (ICRS)

The system usually used is based on an extension of the Earth's axis of rotation, hence the name equatorial coordinate system (fig. 4.1 left). If we extend the Earth's axis

Chapter 4. Star transit simulation

outward into space, its intersection with the celestial sphere defines the north and south celestial poles; equidistant between them, and lying directly over the Earth's equator, is the celestial equator. Measurement of "celestial latitude" is given the name declination (DEC), but is otherwise identical to the measurement of latitude on the Earth: the declination at the celestial equator is 0 deg and extends to ± 90 deg at the celestial poles. The east-west measure is called right ascension (RA) rather than "celestial longitude", and differs from geographic longitude in two respects. First, the longitude lines, or hour circles, remain fixed with respect to the sky and do not rotate with the Earth. Second, the right ascension circle is divided into time units of 24 hours rather than in degrees; each hour of angle is equivalent to 15 deg of arc.

The Earth orbits the Sun in a plane called the ecliptic. From our vantage point, however, it appears that the Sun circles us once a year in that same plane; hence, the ecliptic may be alternately defined as "the apparent path of the Sun on the celestial sphere".

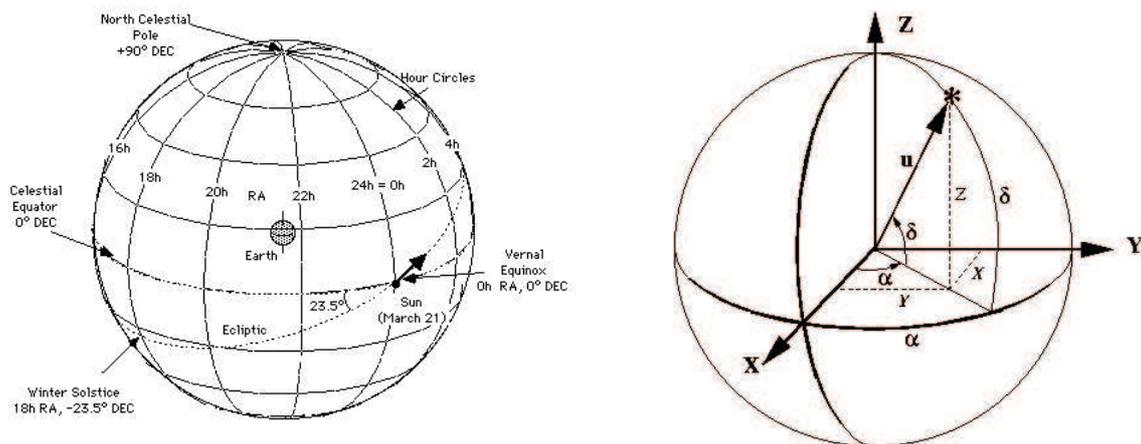


Figure 4.1: Coordinates of the celestial objects are usually given in the equatorial system. Left: celestial sphere and equatorial coordinate system. Right: equatorial and cartesian coordinate systems (* – the star position on the sphere).

The Earth's equator is tilted 23.5 deg from the plane of its orbital motion, or in terms of the celestial sphere, the ecliptic is inclined 23.5 deg from the celestial equator. The ecliptic crosses the equator at two points; the first, called the vernal (spring) equinox, is crossed by the Sun moving from south to north on about March 21st, and sets the moment when spring begins. The second crossing is from north to south, and marks the autumnal equinox six months later. Halfway between these two points, the ecliptic rises to its maximum declination of +23.5 deg (summer solstice), or drops to a minimum declination of -23.5 deg (winter solstice).

Therefore, star positions, or more generally, directions in the astronomical sense, are given in the International Celestial Reference System (ICRS). The spherical longitude

and latitude coordinates (right ascension and declination) are denoted as (α) (δ) (see fig. 4.1 right).

The cartesian unit vectors along the principal axes of the ICRS are denoted \mathbf{X} , \mathbf{Y} , \mathbf{Z} , with \mathbf{Z} pointing towards the celestial north pole, \mathbf{X} towards the spring equinox and $\mathbf{Y} = \mathbf{Z} \times \mathbf{X}$. A general unit vector \mathbf{u} has direction cosines X, Y, Z , such that

$$\mathbf{u} = X\mathbf{X} + Y\mathbf{Y} + Z\mathbf{Z} \quad (4.1)$$

with $X^2 + Y^2 + Z^2 = 1$.

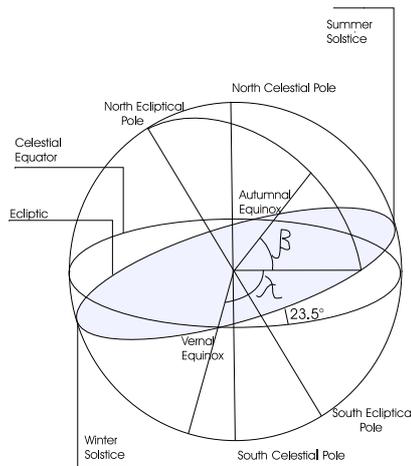


Figure 4.2: Ecliptic coordinate system.

4.1.2 The Ecliptic Coordinate System

In the ecliptic coordinate system (fig. 4.2), the fundamental reference plane is chosen to be the ecliptic, i.e. the orbital plane of the Earth around the Sun. Earth's revolution around the Sun defines an orientation and thus the North and the South Ecliptic Pole.

The ecliptic latitude (β) is defined as the angle between a position and the ecliptic and takes values between -90 and +90 deg, while the ecliptic longitude (λ) is again starting from the vernal equinox and runs from 0 to 360 deg in the same eastward sense as Right Ascension.

4.1.3 Galactic Coordinates

This coordinate system is most useful for considerations of objects beyond the solar system, especially for considerations of objects of our Milky Way galaxy, and sometimes beyond. Here, the galactic plane, or galactic equator, is used as reference plane. This is the great circle of the celestial sphere which best approximates the visible Milky Way.

Chapter 4. Star transit simulation

For historical reasons, the direction from us to the Galactic Center has been selected as zero point for galactic longitude l , and this was counted toward the direction of our Sun's rotational motion which is therefore at $l = 90$ deg. This sense of rotation, however, is opposite to the sense of rotation of our Galaxy, as can be easily checked ! Therefore, the galactic north pole, defined by the galactic coordinate system, coincides with the rotational south pole of our Galaxy, and vice versa.

Galactic latitude b is the angle between a position and the galactic equator and runs from -90 to $+90$ deg. Galactic longitude runs of course from 0 to 360 deg.

4.1.4 Coordinate transformations

Equatorial and galactic coordinates

Given galactic coordinates (b, l) the equatorial coordinates δ (declination) and α (right ascension) can be computed from the formulas:

$$\begin{aligned}\cos \delta \cos (\alpha - 282.25^\circ) &= \cos b \cos (l - 33^\circ) \\ \cos \delta \sin (\alpha - 282.25^\circ) &= \cos b \sin (l - 33^\circ) \cos 62.6^\circ - \sin b \sin 62.6^\circ \\ \sin \delta &= \cos b \sin (l - 33^\circ) \sin 62.6^\circ + \sin b \cos 62.6^\circ\end{aligned}$$

Ecliptic and equatorial coordinates

The obliquity of the ecliptic is essentially constant, $\epsilon = 23.446$ for equinox 1950, respectively $\epsilon = 23.439$ for equinox 2000. The precession of the vernal equinox along the ecliptic is $p_0 = 50.3$ "/year. Hence $\lambda_{2000} = \lambda_{1950} + 0.698^\circ$ The north ecliptic pole is at $\alpha = 18$ h, $\delta = 90^\circ - \epsilon$. The north celestial pole is at $\lambda = 90^\circ$, $\beta = 90^\circ - \epsilon$.

Both are counted eastward from the vernal equinox. Apart from the minimal change in ϵ , the transformation equations then are the same for 1950 and 2000:

- transformation $\alpha, \delta \longrightarrow \lambda, \beta$:

$$\begin{aligned}\sin \beta &= \sin \delta \cos \epsilon - \cos \delta \sin \epsilon \sin \alpha \\ \cos \lambda &= \frac{\cos \alpha \cos \delta}{\cos \beta} \\ \sin \lambda &= \frac{\sin \delta \sin \epsilon - \cos \delta \cos \epsilon \sin \alpha}{\cos \beta}\end{aligned}$$

- transformation $\lambda, \beta \longrightarrow \alpha, \delta$:

$$\begin{aligned}\sin \delta &= \sin \beta \cos \epsilon + \cos \delta \sin \epsilon \sin \alpha \\ \sin \alpha &= \frac{-\sin \beta \sin \epsilon + \cos \beta \cos \epsilon \sin \lambda}{\cos \delta} \\ \cos \alpha &= \frac{\cos \lambda \cos \beta}{\cos \delta}\end{aligned}$$

- transformation $\lambda, \beta \longrightarrow x, y, z$:

$$\begin{aligned}x &= \cos \beta \cos \lambda \\y &= \cos \beta \sin \lambda \\z &= \sin \beta\end{aligned}$$

4.2 The model of Galaxy

We will first consider the simplest model of a distribution of stars as a function of apparent position on the sphere: a homogeneous distribution of stars over the sphere. This simple model of the sky can be used to simulate the properties of attitude star transits (transits of bright stars with known position) because its distribution on the sky should be chosen as homogeneous as possible in order to get an accurate satellite position. (the bright stars are more or less evenly distributed through the sky).

Second, as we intend to provide our star transit generator with star counts to the faint limiting magnitude a more realistic multi-component model of the Galaxy (based on the work of [44, 45]) is incorporated. In this case, the number of stars is derived from a model stellar density distribution computed per unit area on the sky for different magnitude intervals and galactic directions by application of the fundamental equation of stellar statistics.

4.2.1 Stellar distribution and simulation of a map of the sky

We want to simulate N stars (DIVA attitude stars – $N \sim 1.000.000$) uniformly distributed over the sky. There are 41253 square degrees across the entire sky, so it is $N/41253$ stars pro square degree.

Using the ecliptical coordinate system, we arrange the stars in ecliptical parallel circles with an interval about $D = \sqrt{41253/N}$ from each other. Each circle contains the homogeneously distributed stars in its interior with the such a distance that the density of $N/41253$ pro square degree is achieved.

Thus, the map of the sky will be easily produced by the following algorithm:

- put the first star at the North Ecliptic Pole;
- calculate the number of parallel circles as $N_{circ} = NINT(180^\circ / \sqrt{41253/N})$ with the proper distance between each other, i.e. $D_{circ} = 180^\circ / N_{circ}$;
- put the first circle at $\beta = 90^\circ - D_{circ}$ completed with 6 stars at $\lambda = 0^\circ, 60^\circ, 120^\circ, \dots$ etc.;
- put consequently the circles to South with the same interval between each other, (i.e. at $\beta = 90^\circ - 2 \cdot D_{circ}, 90^\circ - 3 \cdot D_{circ}$, etc.), filling properly each circle with the stars at $\lambda = 0^\circ, 2\pi/N_*, 2\pi/N_*, \dots$, where $N_* = NINT(360^\circ \cdot \cos(\beta/D))$;

Chapter 4. Star transit simulation

- and so forth till the South Pole, where the last circle with 6 stars must be created as well as the last star at South Pole;
- transform (λ, β) to (x, y, z) ;
- during this procedure the actual number of stars N_{tot} is calculated.

An output catalogue consists of 4 parameters for each star:

- the star position in cartesian coordinates (x, y, z) that are more convenient for "star transit" simulation, and
- the star identification (ID) number (from 1 to N_{tot}) that is necessary for creating the attitude star catalogue and for the following star identification in our practical approach of star transit simulator.

4.2.2 Simulation of the stellar magnitude distribution

We also have to assign magnitudes to the stars of the simulated sky should also certainly contain magnitudes. In our simulator one can achieve this via:

1. an approximation formula from [19] that holds very nicely between magnitudes $V=3$ and $V=15.3$:

$$\log N_V = -3.72 + 0.45V, \quad N_V = 10^{-3.72}10^{0.45V} \quad (4.2)$$

where N_V is the mean number of stars per square degree which are brighter than V . This means that the brightness distribution function of the stars with respect to V is:

$$D(V) = dN_V/dV = 10^{-3.72}0.45 \ln(10)10^{0.45V}$$

This approximation is suitable enough for DIVA magnitude simulation¹, but not for Gaia. The subsection below explains how this simulation can be done by two different methods;

2. an approximation of predicted star counts down to the 20rd magnitude according to [45]. Allen's formula (4.2) can only describe correctly the brightness distribution down to 16, so the more appropriate distribution will be used for Gaia satellite; to describe the behavior of the brightness distribution down to 20rd magnitude one can also rely on the data from [26].

¹For a spatially homogeneous star distribution without interstellar extinction, N_V would have the same exponential form, but with exponent 0.6 instead of 0.45.

Simulation of stellar magnitude distribution based on Allen's formula [19]

This describes how to simulate V magnitudes — with a predefined distribution function over a predefined interval — can be generated from uniformly distributed $[0, 1]$ random numbers.

- 1 **Method.** The more general problem of creating a set of uncorrelated random variables $\mathbf{y} = \{y_i\}$ with a specific distribution function from a set of uniformly distributed ones $\mathbf{x} = \{x_i\}$ can be stated as follows:

We are looking for a function f such that:

- i) $\mathbf{y} = \{y_i\} = \{f(x_i)\}$ (transformation of x to y)
- ii) $f(0) = y_0$ and $f(1) = y_1$ (transformation into a predefined interval)
- iii) $dN = 1 * dx = H(y) dy$ (predefined distribution function)

This leads to the following differential equation:

$$dy/dx = f'(x) = 1/H(y) = 1/H(f(x)).$$

For the case of our transformation for the V magnitude, H is the function given on page 1, and the target interval is $[3.0, 16]$. The general solution of the equation was found using the 'Maple' computer algebra system. The target interval was reached by utilizing the integration constant of the equation's general solution.

Thus the completed specific solution reads:

$$V(x) = -0.9650988487 \ln(0.1303170580 \cdot 10^{-6} / (x + 0.2917435524 \cdot 10^{-5}))$$

- 2 **Method.** Other way to simulate the V is to use cumulative distribution of $D(V)$ and uniform distribution as follows.

We have a density of the distribution

$$D(V) = C \cdot dN_V/dV = 10^{-3.72} 0.45 \ln(10) 10^{0.45V}$$

- 1) First step is to write a correct form of the density of the distribution, calculating the normalization coefficient:

$$C \int_{-\infty}^{+\infty} D(v) = C \int_3^{15.3} 10^{-3.72} 0.45 \ln(10) 10^{0.45V} dV = 1$$

$$C = 0.0006839$$

So, we can write now the following cumulative distribution:

$$N_{<V} = \begin{cases} 0, & V \leq 3, \\ C \cdot (10^{-3.72} (10^{0.45 \cdot V} - 10^{0.45 \cdot 3})), & 3 < V \leq 15.3, \\ 1, & V > 15.3 \end{cases}$$

Chapter 4. Star transit simulation

2) Let's ξ – uniform distribution random number, then from

$$\xi = C \cdot (10^{-3.72}(10^{0.45 \cdot V} - 10^{0.45 \cdot 3}))$$

one can get

$$V = (1/0.45) \log\left(\frac{\xi}{C} 10^{3.73} + 10^{0.45 \cdot 3}\right)$$

Simulation of stellar magnitude distribution based on the data from [45]

This subsection describes how to simulate the V magnitude distribution with a numerically predefined distribution function by the inverse transform method.

According to [45] the predicted integral stellar density of the Galaxy is presented by the table 4.1. These data are used to create a digital distribution function for interval $V=3\dots 20$, which is presented in fig 4.3(\diamond) along with Allen's distribution function (solid curve). A cubic-spline interpolation and extrapolation [27] has been applied in order to get a good approximation for inverse function.

V [mag]	lg N(V) [$\frac{stars}{deg^2}$]
5.0	-1.31
6.0	-0.821
7.0	-0.342
8.0	0.121
9.0	0.565
10.0	0.993
11.0	1.400
12.0	1.787
13.0	2.155
14.0	2.506
15.0	2.838
16.0	3.149
17.0	3.435
18.0	3.693
19.0	3.936
20.0	4.170

Table 4.1: Predicted integral stellar density of the Galaxy

Inverse transform method (reference) is based on the following simple proposition: let ξ – uniform (0,1) random variable, then for any distribution function F the random variable X defined by

$$X = F^{-1}(\xi)$$

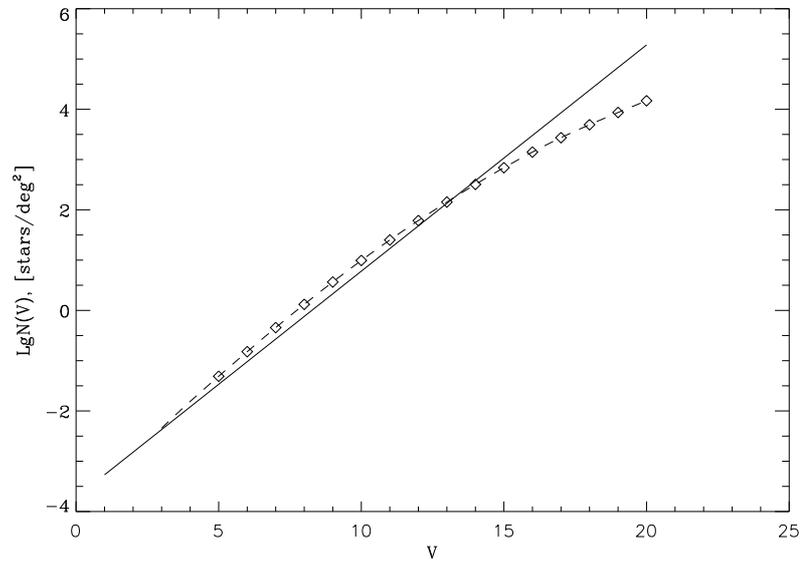


Figure 4.3: The brightness distribution according to [45] (\diamond) and [19] (solid curve).

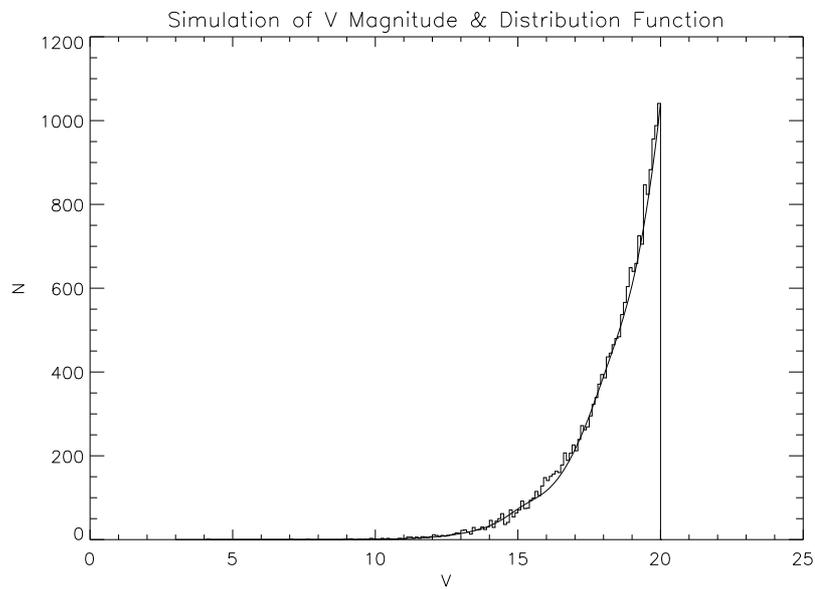


Figure 4.4: Test of the simulation of the stellar magnitude distribution: simulated (histogram) and theoretical (solid line) cumulative brightness distribution functions.

Chapter 4. Star transit simulation

has distribution F . In our case one should calculate the magnitude as

$$V = N_{<V}^{-1}(\xi)$$

If the step of interpolation is rather small then the closest value of digital distribution function can be considered as a true one.

In fig 4.4 the simulated and theoretical cumulative brightness distribution functions are presented. It may be seen that sufficiently good agreement could be achieved with the step of magnitude ~ 0.1

It is worth to note that it would be better to have a gradual decrease of the distribution of detected magnitudes at the lower brightness limit, rather than a sudden cut at 20. This can perhaps be included at a later time. The missing of stars brighter than $V=3$ is irrelevant.

So, we have finally the map of the homogeneously distributed true sky that contains the ID of stars, cartesian coordinates (x,y,z) in ecliptical coordinate system and magnitudes of stars.

4.3 A multi-component model of the Galaxy

One of the required aspects of the DIVA star simulator is its capability to generate the star transits using the realistic Galaxy model rather than a simple homogeneously sphere-distributed stars. This section describes the ingredients of such a model based on the distributions of stellar magnitude, colors and mean parallaxes obtained by [44, 45].

The assumed model produces the Galaxy as a symmetrical system with respect to its rotation axis and to its equatorial plane. Stellar counts are based on a three-component model of the Galaxy consisting of the thin disk (hereafter the disk), thick disk and spheroid populations. Further, the authors of [44, 45] considered several groups of stars in the thin disk, namely the main sequence stars, disk red giants and supergiants of I and II luminosity classes. These five stellar groups have different stellar luminosity functions and spatial distributions.

4.3.1 Stellar distribution: fundamental equation of stellar statistics

In an element of solid angle $d\Omega$, the number of modeled stars A_j with magnitude m in a direction (l, b) are based on the first fundamental equation of stellar statistics for the j -th subsystem (or stellar group):

$$A_j(m, l, b) = d\Omega \int_0^{R_{lim}} D_j(\vec{r}, M_m) F_j(M_m) R^2 dR \quad (4.3)$$

where $D_j(\vec{r}, M_m)$ is the space-density function of the absolute magnitude M_m and the position \vec{r} relative to the Galactic center; $F_j(M_m)$ is the luminosity function. It is assumed that $F_j(M_m)$ does not depend on the position within the Galaxy. R denotes the distance from the Sun, R_{lim} – the maximum distance from the Sun where a star of a given M_m appears with given m magnitude. Density functions $D_j(\vec{r}, M_m)$ of different populations are expressed by Bahcall [40] formulas:

$$D_{disk/thick\ disk}(\vec{r}, M_m) = \exp[-Z/h_Z(M_m) - (X - R_0)/h_X] \quad (4.4)$$

$$D_{spheroid}(\vec{r}, M_m) = (R_s/R_0)^{-7/8} \exp[-10.093(R_s/R_0)^{0.25} + 10.093][1 - 0.08669/(R_s/R_0)^{0.25}] \quad (4.5)$$

where

$$X = (R_0^2 + R^2 \cos b - 2RR_0 \cos b \cos l)^{0.5}$$

is the distance in the Galactic plane from the Galactic center to a star; $Z = R \sin b$ is star vertical distance to the Galactic plane; $r = (X^2 + Z^2)^{0.5}$ and R_0 are distances from the Galactic center to a star and to the Sun, respectively. $R_s = (X^2 + Z^2/k^2)^{0.5}$, where k is a ratio of axes of a spheroid of spatial density distribution; h_X is the disk length scale; h_Z is the disk height scale. As an indicator of the age, h_Z is related to the stellar absolute magnitude differently for main sequence stars and for red giants in the thin disk.

The absolute stellar magnitude M_m is related to apparent magnitude m and extinction via

$$M_m = m - 5 \log R + 5 - A_m(R, b),$$

where $A_m(R, b)$ is the total interstellar extinction at a distance R from the Sun, modeled according to Parenago's formula [39]:

$$A_m = \frac{a_m \cdot h_{Za}}{|\sin b|} \left[1 - \exp\left(\frac{-R \cdot |\sin b|}{h_{Za}}\right) \right]$$

where a_m – the extinction at a distance-unit, h_{Za} – the height scale of an extinction layer. Thus, the total number of stars in the direction with the galactic coordinates (l, b) in unit magnitude interval with mean apparent magnitude m can be presented as

$$A(m, l, b) = \sum_j^5 A_j(m, l, b). \quad (4.6)$$

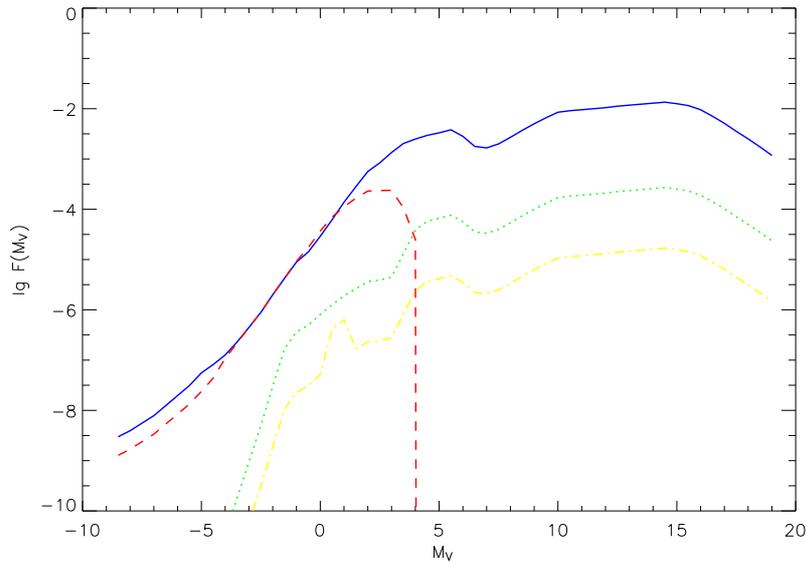


Figure 4.5: The luminosity functions $F_j(M_V)$ in stars/mag/pc³ assumed in the model. The blue solid line and red dashed line are the luminosity functions of the disk main sequence and disk red giants respectively. The luminosity function of the thick disk is presented by the dotted green curve whereas the stars of the spheroid are shown by yellow dot-dashed line.

4.3.2 Luminosity functions

We calculate the distributions given by equation (4.3) with the improved luminosity functions (see fig. 4.3.1) constructed in [44]:

The luminosity function for the stars on (or near) the main sequence of brighter stars up to $M_V = 13$ mag are taken from [58] and [51]; for fainter stars up to $M_V = 19$ mag the results of [38] for the nearby stars are used; the fraction of evolved stars, which have left MS phase (red giants and supergiants) as a function of absolute visual magnitude is taken from [58] and modified taking into account red giant clump, the metal rich equivalent to the horizontal branch of core helium burning stars, at $M_V \approx 0.75$ mag, $(B - V) \approx 1.0$ mag according to data of [37]. The luminosity functions of the thick disk and spheroid and the star number density normalization {thin disk : thick disk : spheroid} as {1 : 0.02 : 0.00125} are taken from [33].

The modeled stellar distributions are calculated with the following parameters:

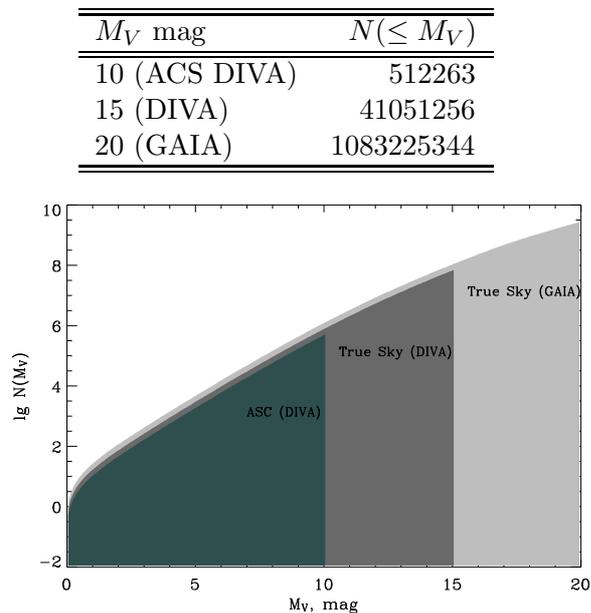


Figure 4.6: This shows modelled integral number of stars distributed through the Galaxy as a function of the magnitude. The shaded regions refer to the total number of stars used to simulate the *true sky* and *attitude star catalogue (ASC)*.

the distance between the Sun and the Galactic center is $R_0 = 8.5$ kpc; the axes ratio of the spatial density distribution for the spheroid is $k = 0.85$, the length scale of the disk subsystems is $h_X = 4$ kpc and the height scale of the thick disk is $h_Z = 1$ kpc. According to [61] and [58] it is assumed that the height scale $h_Z(M)$ rises from 90 to 350 pc at M_V interval [2.3, 5.4] mag for main sequence stars of the disk, whereas for red giants, the height scale changes from 250 to 400 pc at M_V [-0.75, 2.6] mag. The height scale of 90 pc and 100 pc [62] were used for extremely young population of supergiants and interstellar extinction layer, respectively. The extinction at a distance-unit is also taken from [62] as $a_V = 1.6$ mag/kpc.

4.3.3 A sky map simulation

In order to simulate a map of the *true sky* we subdivide the entire sky into $N_{\text{sub-areas}}$ equal sky sub-areas. We then calculate by means of eq. (4.6) differential brightness functions $A_i(M_V)$ in each of these sky sub-areas with the given galactic coordinates $(l, b)_i$ of their centers and limits $(l_{\min}, l_{\max})_i$ & $(b_{\min}, b_{\max})_i$. In each of these sub-areas therefore we can simulate the stars (homogeneously-distributed in the framework of a

Chapter 4. Star transit simulation

given sub-area) with certain magnitudes in the same manner as it was made in the case of the simple homogeneously distributed galaxy model.

We take the following increments in galactic coordinates to construct the grid of the sky: $\Delta l = 30^\circ$ and $\Delta b = 10^\circ$ (for a better approximation near the galaxy plane we take the grid step $\Delta b = 5^\circ$). The number of stars at each sub-areas is defined by a given value of upper limit of star magnitude M_V^{upper} we intend to simulate for the Galaxy map. The modeled integrated number of the stars as a function of M_V is shown on the fig. 4.6.

Finally, for a simulating map of the sky we divided it up into 252 subareas and populated according to the appropriate functions for spatial and brightness distributions. As illustration, two simulated samples of the map of the Galaxy of different limiting magnitudes are presented in the galactic and ecliptic coordinate systems by orthogonal and AITOFF projection on the figures 4.7, 4.8. It can be seen that the features of the disk-like star distribution of our Galaxy increase with regards to a fainter magnitude.

4.4 Attitude Star Catalogue

In principle, the attitude star catalogue will contain positions, magnitudes, proper motions and colors for all attitude stars. Positions and proper motions are parameterized in the equatorial coordinate system, i.e. positions are defined by the two angles (α, δ) , whereas proper motions are given as the time derivatives of these angles.

In our practical applications the ASC is derived from the true sky without proper motions², the positions are also parameterized in cartesian coordinate (x, y, z) representing the equatorial coordinate system on unit celestial sphere. It is assumed that the objects in ASC describe a subset of the true sky in such a way that the "unknown" differences in coordinates and magnitudes between the objects in ASC and the respective objects of the true sky are compatible with the mean errors of the ASC. The simulated errors of the star positions and magnitudes in the ASC are defined by DIVA requirements.

As suggested by [60] the attitude star catalogue should contain ~ 500000 attitude stars. To provide this we generate the ASC from the DIVA *true sky* for the the bright stars $M_V \leq 10$ mag (see fig. 4.6) .

4.5 The Instrument

In the following we will use the notions of "*nominal, true, known*" to distinguish between different status. This notions apply to the whole instrument as well to all its subsystems.

The *nominal instrument* is an instrument with nominal parameters, defined in [11] together with their tolerances. The *nominal instrument* is the instrument that follows the prescribed nominal behavior and therefore is always known.

²For the tasks of ScQL proper motions are not interested, it was presented here for completeness.

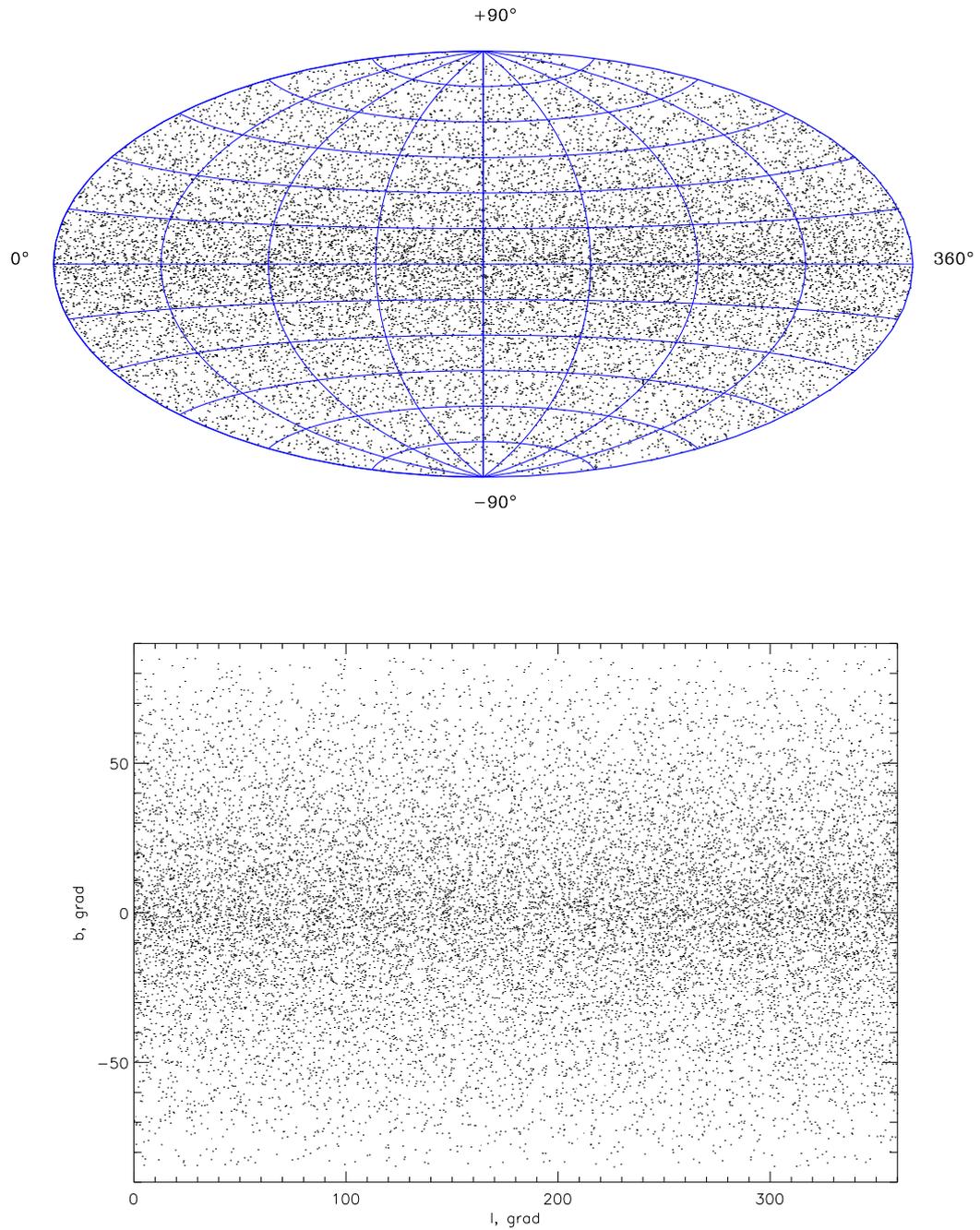


Figure 4.7: The simulated map of the Galaxy for the stars with magnitude up to $M_V = 6.5$ in galactic coordinates (l, b) : lower panel – orthogonal projection, upper panel – AITOFF equal-area projection. It shows that at brighter magnitudes the sky is rather represented by an isotropic stellar density distribution.

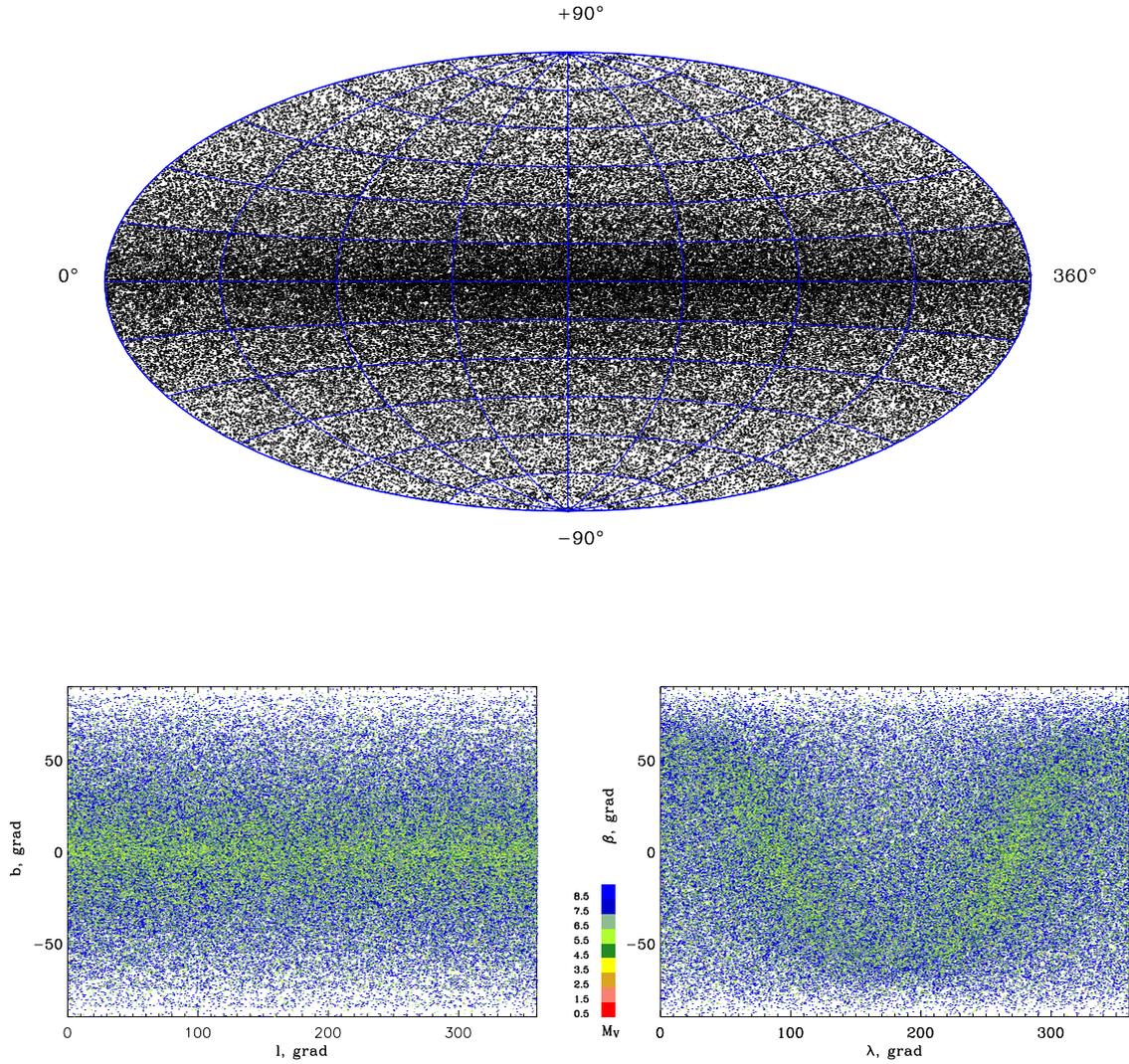


Figure 4.8: A simulated map of the Galaxy for the stars with magnitude up to $M_V = 8.5$ in galactic coordinates (l, b) : orthogonal projection (lower left panel), upper panel – AITOFF equal-area projection. Lower right panel presents the Galaxy map in ecliptic coordinate system (λ, β) . Filled color polygons specify an array of color indexes and star magnitudes M_V . It shows that the features of the disk-like star distribution of our Galaxy increase with regards to a fainter magnitude. Note: an upper limit of the stellar magnitude $M_V = 8.5$ was used in this sample in order to keep the picture of the map more visible. In the DIVA simulator stars are simulated up to $M_V = 15.5$.

The *true instrument* is given by the true values of the parameters of the instrument built according to [11] and is in principle unknown. The *true instrument* can be a nominal one if the parameters are within the tolerances mentioned above. It is not nominal if at least one of the parameters falls outside the tolerance.

The *true parameters* used for the simulation of the observations are unknown to the on-board software. The on-board software tries to find an approximation to the *true parameters* to work with. These parameters are then called *known parameters*. However, there are *known parameters* used in the instrument to perform the observations (and hence the simulation process) that are determined by the on-board software itself (e.g. TDI clock stroke rate). These are *true parameters* and also completely known. It is also important to distinguish the types of parameters "general parameters" and "derived parameters" (see the collection of parameters of DIVA main instruments in Appendix 8.4). Only the derived *true parameters* can be completely known (e.g. TDI clock stroke rate).

The *known instrument* as well as the *true instrument* are in principle functions of time. We also have to distinguish between the *known instrument on board* (KIB) and *known instrument on ground* (KIG). On board the knowledge of the instrument is poorer because of lesser information than on ground. In fact, the data available to the software on board to get knowledge about the instrument is only from the past, whereas the analysis on ground has also informations about the satellite in future.

4.6 DIVA science data and coordinate systems

4.6.1 The CCD pixel stream

Neither equatorial coordinates nor ecliptic coordinates of celestial objects are directly observable by DIVA. Observations are generally performed in coordinate axes attached to the instrument (i.e. the instrument coordinate system). The primary signal of DIVA is the stream of CCD pixels produced by the TDI operation of the detector chips. The primary astrometric observable thus is the location of a stellar image within this stream of pixels. Note that these "pixels" are not physical pixels on a CCD chip, but entities in the output data stream of a CCD. Usually there is no real danger of confusion between these two concepts, but a clear distinction can be made (whenever necessary) by calling them "data pixels" and "physical pixels", respectively.

The identity of a particular (data) pixel in the data stream from DIVA's instrument can be specified by three integer indices (specific values for them will be defined in section 4.6.8):

- k , the running number of the TDI clock stroke at which the pixel was read into the read-out register of the CCD chip.

Chapter 4. Star transit simulation

- m , the running number of the particular CCD column to which the pixel belongs: for the imaging area of the CCD chip, $m = 0, \dots, M - 1$, where M is the total number of (effective) columns for the particular CCD and $m = 0$ is defined to be the pixel at the smallest focal-plane z coordinate. M is 256 pixels in the case of binning (see fig 1.7).
- n , a running number identifying the particular CCD chip to which the pixel belongs: $n = 1, \dots, N$, where N is the total number of CCD chips on the focal plane. The precise numbering of the CCDs is given in Chapter 4.6.7. N is of the order of a few dozen.

In short: Every CCD n produces a "ribbon" of (data) pixels which is "infinitely" long in scan direction (time) and has a width of M pixels in the perpendicular direction. Mathematically, the "ribbon" is a rectangular array, the elements of which are labelled by k and m .

The meaning of "columns" and "lines" on the CCD chip is made clear by the following statement: the TDI mode shifts charges from one line to the next, i.e. the fictitious center of light of a star image stays in the same column during its transit over a CCD chip.

4.6.2 The body-fixed satellite system and field coordinates

We need different coordinate systems to simulate (interpret) DIVA observations. It is useful to express star position in the so-called body-fixed satellite system that is rotating with the satellite. The unit vectors along the principal axes are called \mathbf{x} , \mathbf{y} , \mathbf{z} , see fig. 4.9. The \mathbf{z} axis is the nominal rotation axis of the satellite. The \mathbf{x} axis is in the plane of the two viewing directions (i.e. the two projections of the optical axis of DIVA's main telescope onto the sky), $\gamma/2$ away from each of them, where γ is DIVA's basic angle, i.e. the angle between the two viewing directions. The \mathbf{y} axis is also in the plane of the two viewing directions such that the system \mathbf{x} , \mathbf{y} , \mathbf{z} is right-handed. A general unit vector \mathbf{u} in the satellite system has direction cosines x, y, z , such that

$$\mathbf{u} = x\mathbf{x} + y\mathbf{y} + z\mathbf{z} \tag{4.7}$$

with $x^2 + y^2 + z^2 = 1$. The nominal rotation of DIVA is positive about the positive \mathbf{z} axis. The \mathbf{y} axis thus precedes the \mathbf{x} axis on the sky by 90 degrees.

In order to describe the actual motion of star images across the focal plane of DIVA, it is useful to define two coordinate systems directly related to the viewing directions, i.e. to the two projections of the optical axis of DIVA's main telescope onto the sky. There is one such system per field of view (thus the name field coordinates). The two systems differ from the body-fixed satellite system via a rotation around the z axis by an angle $\pm\gamma/2$. The simple meaning of the field coordinates is shown in fig. 4.9. Their origin is at the projection of the optical axis; the w coordinate is reckoned along the

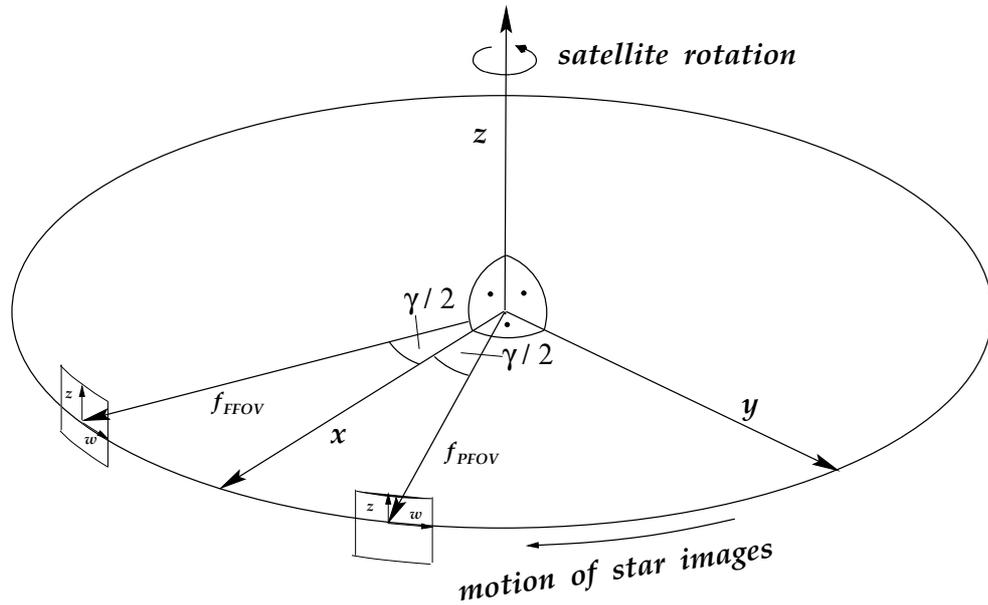


Figure 4.9: The body-fixed satellite system and field coordinates (γ is the angle between two field-of-views).

nominal scan direction; the z coordinate is orthogonal to it. The orientation is such that star images move towards negative values of w , and that celestial objects with positive z coordinate in the satellite system will have positive z field coordinate. The unit vectors along the principal axes of either field coordinate system are denoted \mathbf{f} , \mathbf{w} , \mathbf{z} , with \mathbf{z} being identical to the third axis of the satellite system, and

$$\mathbf{f} = \mathbf{x} \cos \gamma/2 \pm \mathbf{y} \sin \gamma/2 = \mathbf{x} \cos \gamma/2 + f \mathbf{y} \sin \gamma/2 \quad (4.8)$$

$$\mathbf{w} = \mathbf{y} \cos \gamma/2 \mp \mathbf{x} \sin \gamma/2 = \mathbf{y} \cos \gamma/2 - f \mathbf{x} \sin \gamma/2, \quad (4.9)$$

where f is the field-of-view index, which is defined as $f = +1$ for the preceding field of view and $f = -1$ for the following. The field coordinates are direction cosines in the field coordinate system, such that a general unit vector \mathbf{u} is

$$\mathbf{u} = v \mathbf{f} + w \mathbf{w} + z \mathbf{z} \quad (4.10)$$

with $v^2 + w^2 + z^2 = 1$.

4.6.3 Focal-plane coordinates

Focal-plane coordinates are defined to be cartesian coordinates in the focal plane and are used to describe the location of CCDs and other parts on the focal-plane assembly.

Chapter 4. Star transit simulation

Millimeters are chosen as the length unit. There is no need to coin separate notations, since the circumstances will always make clear whether field coordinates or focal-plane coordinates are referred to. In principle, the relation between field coordinates and focal-plane coordinates for a star image should be defined by the detailed optical properties of the DIVA telescope. The field coordinates are reckoned on the celestial unit sphere, the focal-plane coordinates on a plane surface, which for a distortion-free optical system would be a tangent plane to the celestial unit sphere, scaled by the effective focal length of the optics. But to a very good approximation the actual focal-plane coordinates of a point on the focal plane are equal to the field coordinates of the projection of that point onto the sky, multiplied by the effective focal length of the DIVA telescope. Thus we can define the two to be simply identical. This little ‘inexactitude’ is adopted on purpose. Its usefulness will become obvious in a forthcoming document on the astrometric calibration of the DIVA instrument.

Note that this choice of the nominal field-to-focal-plane transformation imposes a tiny (irrelevantly small) barrel-shaped distortion on the ideal DIVA telescope. At the edge of the field of view (i.e. at a field angle of 0.25 degrees) it is slightly less than 10^{-5} . A distortion-free telescope would project a field angle α to a focal-plane coordinate $F \tan \alpha$, with F being the effective focal length. The difference between the sine and tangent of 0.25 degrees is only $4.15 \cdot 10^{-8}$, corresponding to 7 mas or $0.66 \mu\text{m}$ on the focal plane (for $F = 16\text{m}$). This is completely negligible for all practical purposes, since both the distortion of the actual optics and the errors in the placement of the actual hardware parts on the focal plane are much larger.

Since the projection of the optical axis may differ between the dispersed (zero diffraction order) and undispersed field sections and is wavelength-dependent in the first diffraction order, we define the zero point of the focal-plane coordinates to be the projection of the optical axis in the undispersed field.

4.6.4 Pixel coordinates and grid coordinates of a star image

The location of a star image in the stream of CCD pixels is determined from the set of the photometric signal by the centroiding algorithm. The details of that algorithm are of no relevance presently, nor are its properties. The algorithm used in the on-board data processing for attitude determination may (and will!) differ from that used on ground for the scientific data reductions.

Since the centroiding algorithm works on nothing but pixels, the resulting centroid will be in units of pixel indices (Fig. 4.6.4). But, of course, the location of the centroid need not be an integer. Let us denote the centroid position by κ (corresponding to index k) and μ (corresponding to m), see Fig. 4.6.4, and let us call κ and μ the pixel coordinates of the star image. These are the primary astrometric observables of the DIVA instrument.

Further below it will be shown that the actual astrometric information is in the

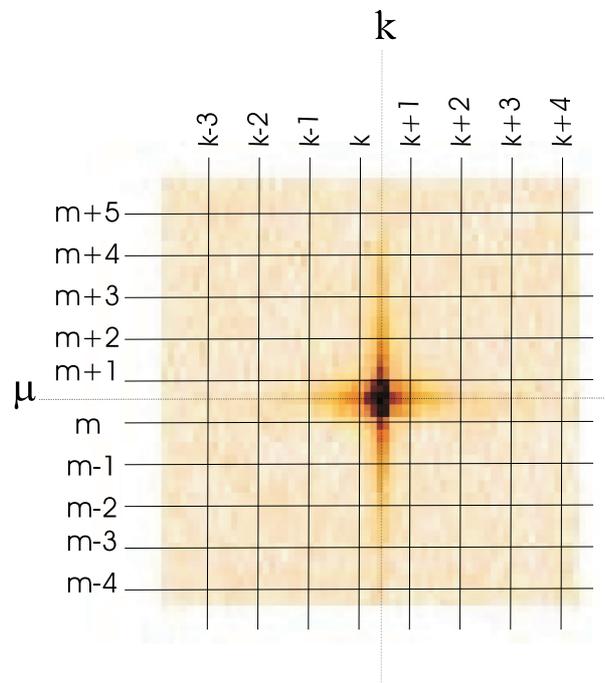


Figure 4.10: Pixel coordinates of SM image.

fractional part of κ , thus we finally define the grid coordinates of a star image as

$$G = \kappa - \text{NINT}(\kappa) \quad (4.11)$$

$$H = \mu \quad (4.12)$$

where NINT is the ‘nearest integer’ function³. G and H are coordinates measured nominally parallel to the field coordinates w and z . The range of possible values is $[-0.5, +0.5]$ for G and roughly $[0, M]$ for H . To be meaningful in the output data flow from the DIVA instrument, any pair of grid coordinates G, H must be supplemented by the integer TDI clock stroke $k = \text{NINT}(\kappa)$ to which it refers, and by the running number n of the CCD chip. In other words: A complete set of grid coordinates consists of four parts, viz. G, H, k, n .

4.6.5 Geometric calibration of the instrument (grid-to-field transformation)

In order to be useful, the astrometric information contained in a star’s image must be transformed from the original grid coordinates G, H to field coordinates w, z (via a

³Equation 4.12 may look a bit silly, giving two names to the same thing. This has been done for the sake of keeping Hipparcos traditions (see [10]).

Chapter 4. Star transit simulation

geometric calibration of the instrument) and finally to equatorial coordinates α, δ (via the attitude matrix). In this sense the geometric calibration comprises the properties of the centroiding algorithm, as well as effects of optical distortions, positioning of chips on the focal plane etc.

The geometric calibration of the DIVA instrument consists of $2N$ pairs of functions $w_{n,f}(H), z_{n,f}(H)$, one for each CCD chip n in each field of view f , giving the field coordinates (w, z) for grid coordinates $(0, H)$. The grid-to-field transformation then has the form

$$w(G, H) = w_{n,f}(H) + Gs_n \cos \alpha_n \quad (4.13)$$

$$z(G, H) = z_{n,f}(H) + Gs_n \sin \alpha_n \quad (4.14)$$

where α_n and s_n are additional calibration parameters. The last terms in Equations 4.13 and 4.14 take care of a possible rotation of the (G, H) frame with respect to the (w, z) frame, caused e.g. by an accidental rotation of the CCD chip on the focal-plane substrate. For a rotation of 3 arcmin the effect may be up to 0.1 mas in z ; in w it is completely negligible ($\cos \alpha_n \simeq 1$). A detailed consideration of applying geometrical calibration is discussed in section 4.8.6, where the prediction algorithm of star transits on SM2 is presented.

4.6.6 Reference great-circle coordinates

The reference great-circle (RGC) coordinates constitute an important coordinate system for the DIVA data reductions. They play no role in the design of the satellite and the on-board data processing. Nevertheless they are defined here for completeness.

A reference great-circle (RGC) is defined by its pole, precisely by its ICRS coordinates α_φ (right ascension) and δ_φ (declination). The RGC coordinates ψ and ζ are longitude and latitude coordinates in a spherical coordinate system determined by this pole. The zero-point of the longitude coordinate ψ is the ascending intersection of the RGC equator with the ICRS equator. The RGC coordinates ψ and ζ are called (RGC) abscissa and (RGC) ordinate, respectively.

4.6.7 Windows

The pixel data stream produced by the DIVA CCDs is much too big to be telemetered to the ground entirely. Therefore it is necessary to select specific parts – so-called windows – of that raw CCD pixel data stream for inclusion into the telemetry data stream sent to the ground.

A window is defined as a rectangular subarray of the pixel array produced by a CCD. As mentioned at the beginning of this chapter we do not simulate the pixel array of a window but only its "lower left corner" (and centroids, see below) that characterizes the window (see Figure 4.11). This "lower left corner" is specified by four integer indices (see also section 4.6.1):

Data structure and "windows"

each CCD produces a long ribbon of data pixels, from which "windows" containing star images are cut out

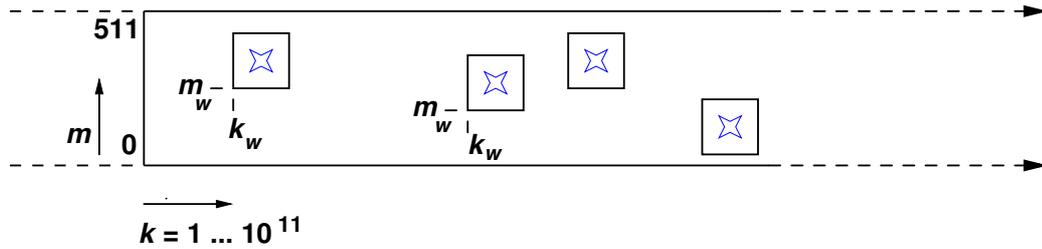


Figure 4.11: Windows in the pixel data stream

- k_w , the number of the TDI clock stroke at which the 'lower left corner' of the window is read into the read-out register of the CCD chip. That means the smallest value of k from all pixels belonging to the window.
- m_w , the number of the particular CCD column of the 'lower left corner' of the window. That means the smallest value of m from all pixels belonging to the window.
- n , the number of the particular CCD chip to which the whole array belongs.
- $type$, characterizes the window type (defined in [12], see Appendix 8.2 for explanation) and with this the size of the window. Also, some information about the observed object enclosed in the window is coded in $type$.

4.6.8 CCD and window numbering: specifics for k , m , n and $type$

Specification for k

The number of the TDI clock stroke will run over the mission from 1 to 10^{11} whole. This would require a 6-Byte integer for each window to be telemetered to the ground. To save telemetry space k is divided into \underline{k} and \overline{k} . The counter \underline{k} is reset at least once at the beginning of the scientific measurements after each perigee. Then \overline{k} counts the number of resets. This \overline{k} does not need to be sent to ground. It can be supplemented on ground. It is only necessary to send \underline{k} to ground. This needs only 4 Bytes per window.

Chapter 4. Star transit simulation

More precisely, the TDI counter \underline{k} is reset whenever the instrument software starts or restarts the TDI clocking. In technical terms of the DIVA System Design and Interface Document [54] this corresponds to the transitions of the instrument software from either the INIT or the IDLE mode to the MEASURE mode. Presumably, such a transition will always occur when the attitude control system (ACS) transits from the fine-pointing mode (FPM) to the nominal-acquisition mode (NAM), i.e. at least after every perigee.

Specification for n

All CCD chips in the Focal plane are named by a number n which in principal is arbitrary. To get a unique order n is set as follows: Starting with SM1 n is increasing in direction of increasing z ($n = 1, \dots, 4$). Then the chips of SM2 ($n = 5, \dots, 8$), SM3 ($n = 9, \dots, 12$), SC1 ($n = 13, \dots, 16$), SC2 ($n = 17, \dots, 20$). This is shown in Figure 4.12.

Specification for m

On each chip the value of m is starting from zero at the imaging column with the smallest z -value and increasing in direction of increasing z . It is also shown in Figure 4.12. The maximum value of m differs between SM, SC on one hand and UV on the other hand. Negative values of m , and values above $M - 1$ will be used to denote underscan pixels.

Specification for $type$

The index $type$ is used to code some information on the window size and the objects contained in the windows according to the observation strategy. For the different detectors (SM, SC, UV) we get different values of $type$. In our simulation we will concentrate on SM CCD mosaic.

4.7 The nominal scanning law

The satellite scans the whole sky on almost great circles, it is optimized for best measurement of parallaxes. There are the following constraints [47]: the angle between the observed fields of view and the Sun should be at least 45° in order to minimize straylight (see fig. 1.5), the inclination of the scans on the ecliptic should be as small as possible because the parallactic effect is parallel to ecliptic (so projection on a scanning circle is larger for small inclinations), two successive scans should overlap in order to avoid unobserved regions (this sets a relation between the scanning speed and the rate of variation of the scanning speed).

In our simulator the scanning law similar to that of Hipparcos was adopted for the DIVA satellite: the sky is scanned with the one satellite rotating in 2 hours, the rotation axis moves slowly on the surface of a cone with its axis pointing towards the sun, the angle between rotation axis and the Sun direction being 45° (see fig. 1.5 and 4.13). One

Physical detector assembly

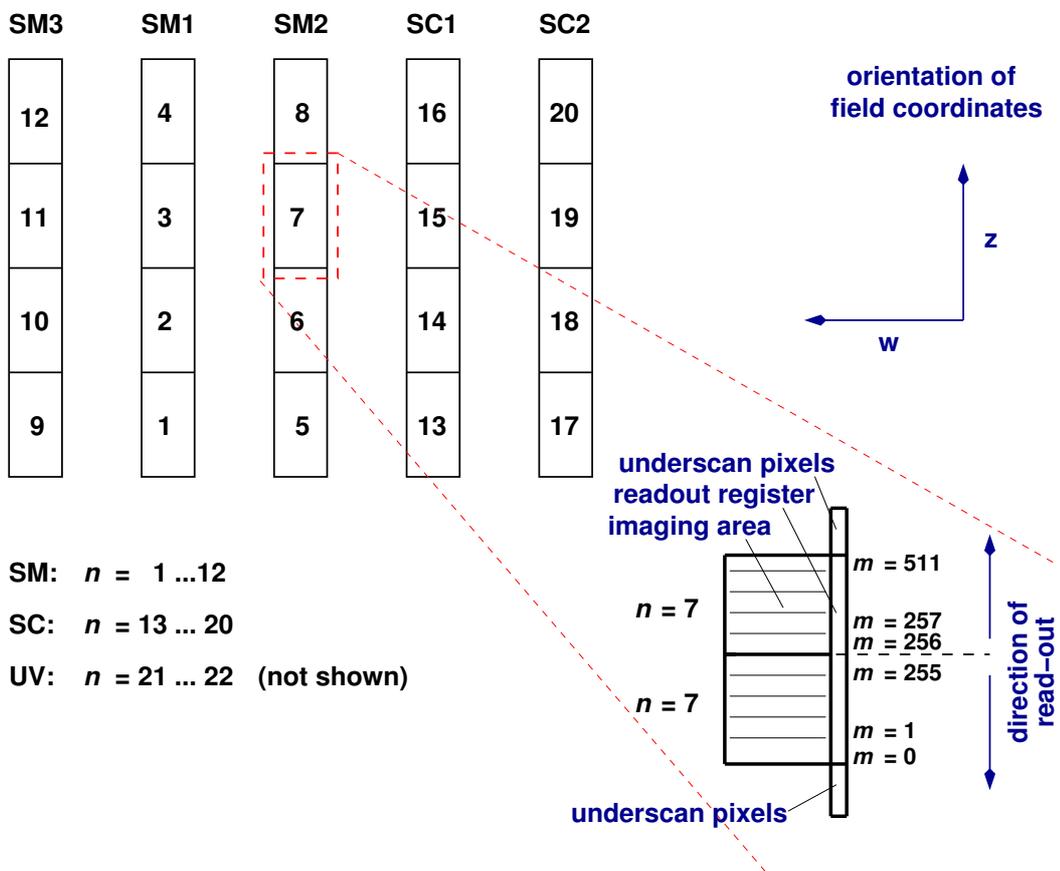


Figure 4.12: The figure shows the numbering of the CCD's by n . The meaning of m is also shown.

Chapter 4. Star transit simulation

revolution of the rotation axis on that cone takes 56 days. With this scanning law, one sky coverage is performed in six months. The details about the nominal scanning law calculation can be found in Appendix 8.3.

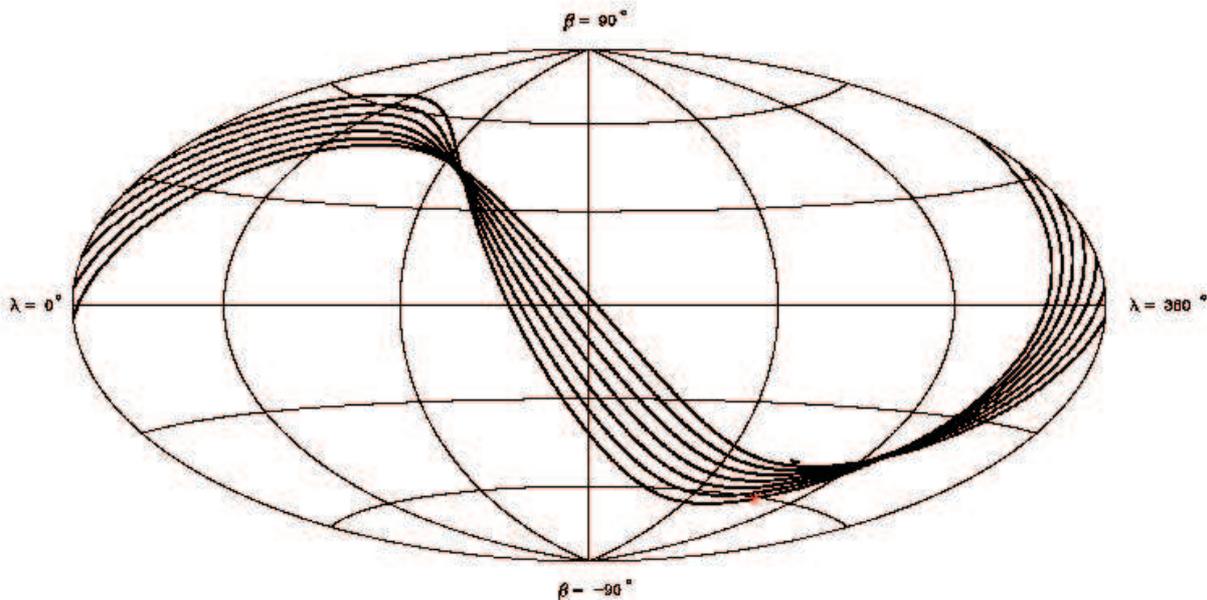


Figure 4.13: The nominal scanning law of DIVA in ecliptical coordinates: we show the part of the sky scanned in 7 consecutive days (★ – coordinates of the begin of observations).

4.8 Star transits and emulation of the work of the on-board software

We discuss here an algorithm of the simulation of star transits as well as of the emulation of the work of on-board software in detail. For the case of simplicity we do not distinguish between the *nominal*, *true* and *known* optical system of the main instrument (see the definitions of the "instrument" in the sec. 4.5). However, as far as focal plane is concerned, we assume in our model that the *known focal plane* is always the *nominal focal plane*, but it differs from the *true focal plane*. Therefore we will be able to simulate, study and take into account a large number of the possible errors of the SM CCD mosaic in the framework of the ScQL monitoring. In the course of the mission the *known focal plan* will be different from the nominal and approach the *true focal plane*.

4.8.1 The true-band sky

In the sections 4.2.1, 4.3 we have simulated the *true sky* (the sky map). As a first step of star transit simulation we determine, the so-called *true-band sky*, for a period of one complete rotation of the satellite (2 hours). For this band we have to find stars according to nominal scanning law from the true sky map. To do this, we

- calculate the attitude matrix of the nominal attitude for the time t ;
- rotate the *true sky* into field coordinates;
- select all stars with $|z_{fc}| \leq z_{\max}$, where $z_{\max} = 3.5^\circ$;
- calculate spherical coordinates for selected stars, i.e. make a transformation of the cartesian vectors to RGC longitude l and latitude b ([9]):

$$l = \arccos \left(\sqrt{\frac{1}{1 + \left(\frac{y}{x}\right)^2}} \right),$$

$$b = \arccos(z)$$

- sorting selected stars according to RGC longitude (the signs of x and y have to be used to select the correct quadrant for l).

4.8.2 Attitude matrix

In principle, the known attitude matrix has to be reconstructed from attitude determination on board but at this stage we simply take it as given at any instant of time. In practice, in our simulation the known and the true attitude matrix are calculated from the nominal scanning law.

4.8.3 Electronics

We generally assume that the true electronics equals *nominal* and *known*, but one can set some chain failures (for true electronics) by a simple qualitative simulation: a particular ADU SM CCD chain is broken (yes/no flag) or the sensitivity of a given ADU SM CCD chain is not appropriate (simulation of the value of the signal-to-noise ratio).

4.8.4 Star transits

Taking into account all above, we can calculate the transits of true sky objects, using "true sky, true attitude, true optical system, true focal plane, true electronics, true TDI clock stroke rate" (see the box, labeled as "Simulator of the star transits", on the simulation flow chart 4.14). At a given time interval $[t_i, t_{i+1}]$ with $\Delta t = 1/724$ sec (1

Chapter 4. Star transit simulation

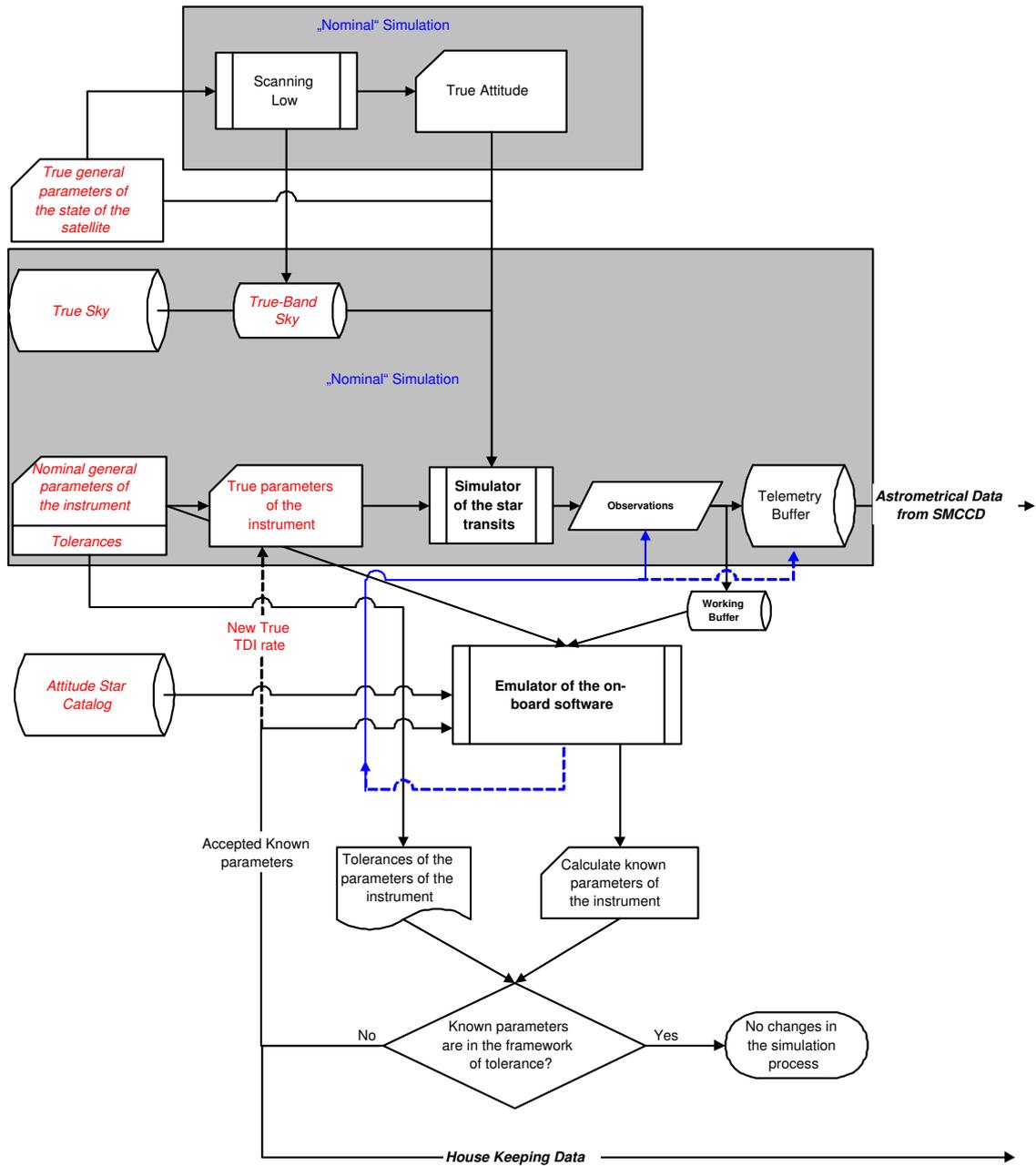


Figure 4.14: Flow chart of the simulation process.

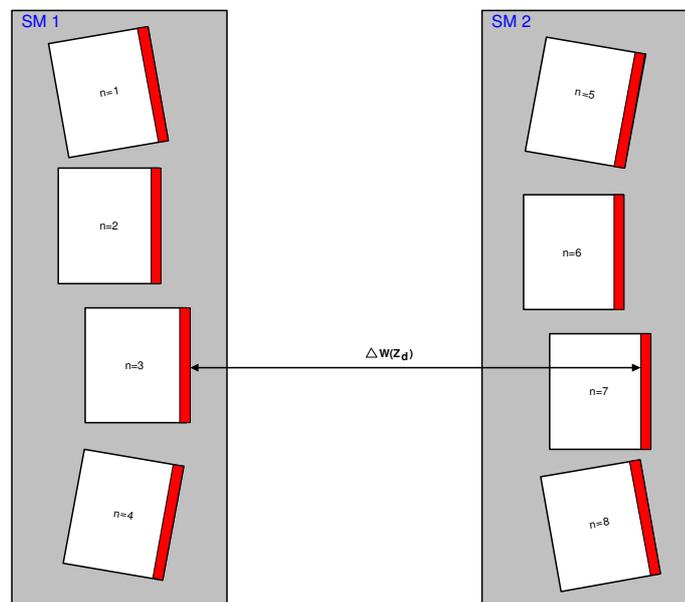


Figure 4.15: Schematic SM1 and SM2 mosaics with readout registers indicated as red lines at the right side of CCDs. Displacements and rotations of the individual CCD chips with the mosaic are exaggerated.

TDI clock stroke) we register the stars which cross read-out chip registers on SM1 and SM2 (see fig. 4.15). So, we registered for the time t_i (or k_i in TDI clock stroke) those stars from FOV1 and FOV2 which fall down on the SM mosaic but with the linear size $\Delta S \cong s_n = 1$ pix in w (scan) direction, corresponding to one TDI clock stroke. From the position of these stars in field coordinates we calculate the centroid (κ, μ) in pixel coordinates. Based on these transits, we can compute the final set of observations that will be transferred to the ground. To get the final observations we emulate the work of the on-board software.

4.8.5 Emulation of on-board software

There are two types of the on-board software that have to be simulated. The first type of on-board software performs the raw-data treatment for particular purposes such as [9]: on-board attitude star catalogue handling, detection and coarse algorithm of star transits for SM1 (partly for SM2), prediction algorithm of SM2 star transits from SM1 detections, on-board algorithm for the background determination, on-board guest observer object handling, on-board magnitude determination and cosmic identification, telemetry buffer handling, data compression algorithm, on-board guest observer object handling etc.

The second type of on-board software is an on-board monitoring software that spe-

Chapter 4. Star transit simulation

cialize in process control. This software should monitor of on-board operations on a routine basis and interfere only when the operations deviates from the prescribed nominal standard [5]. To do this, it performs a real-time analysis of observational data to calculate the parameters of the instrument. If the parameters are not in the framework of their tolerance the intervention must be done. In particular we will consider the TDI clock stroke adjustment algorithm that belongs to monitoring software. If the TDI clock stroke rate does not coincide with the actual satellite rotation, then the on-board monitoring either sets up a new adjusted TDI clock stroke rate (if the discrepancy was not too big) or commands to make a maneuver (the so-called jet-firing).

Detection algorithm on SM1

The detection module decides if the true transit is detected. We can not really simulate this because we do not generate pixel data, but for the needs of ScQL we can simulate a background (as a value) and then detects only stars brighter than one particular magnitude.

Coarse centroiding algorithm on SM1

Coarse centroiding can not be simulated properly either, we should only decide if the coarse centroiding works nominally, i.e. within tolerance $\sigma_{coarse} = 1$, then we distribute the shift of the true transits around centroid according to a normal distribution.

Fine centroiding of attitude stars

To emulate this we have to take the centroids from the true transits and distribute normally the shift of it with $\sigma = 0.14$ pix. (see requirements in [11]).

Window cutting algorithm for non-attitude stars on SM1

We can simulate the nominal case (simply go L_z and L_w pixels back to get left corner of window in pixels coordinate) or non-nominal to set up a wrong value of L_z and L_w .

Window cutting of attitude stars

At this stage of the project we use the same approach to model this as in the window cutting algorithm for non-attitude stars.

Prediction of the star transits on SM2 from SM1

As most windows on SM2 chips are cutted without detection, one has to predict the instant of SM2 crossing for these stars. The detailed explanation of the algorithm of the star transit prediction is given in the section 4.8.6.

Attitude star catalogue handling

According to our Galaxy model we have created not only the map of the true sky but attitude star catalogue also (see 4.3). To handle ASC, one should first of all perform the same steps which were implemented to the true sky map (see 4.8.1). So, now we have to

- calculate the attitude from the nominal scanning law (here *known* is *nominal*);
- select all attitude stars for one rotation of the satellite;
- compute expected RGC coordinates for all selected attitude stars;
- transform of the cartesian vectors to RGC longitude and latitude;
- sorting selected attitude stars to RGC longitude.

To simplify the process of *attitude star identification* we provide all attitude stars with ID number in the true sky map. Thus, during the simulation of star transits we can identify the attitude stars by their ID without calculation of the longitude and latitude of all registered attitude stars.

Star rate determination and TDI clock stroke rate adjustment

If the shifting of the charge on the CCD does not match the actual rotation of the satellite but this mismatch is not yet large enough to initiate the jet-firing then the new TDI clock stroke rate that is synchronized with scan-speed of the satellite must be calculated and accepted as an actual one. The explanation of the two possible algorithms of the TDI clock stroke rate adjustment are discussed in sec. 4.8.7 below.

4.8.6 Prediction of SM2 star transits from SM1 detections

The centroiding algorithm is supposed to provide the location of a centroid (k_d, m_d, n_d) for each image detected on SM1. Additionally, the following calibrational parameters are calculated on the ground (taken as nominal at the beginning) and uplinked to the satellite:

$$\begin{aligned}
 \text{SM1 : } & z_{b,1}^{SM1}, z_{b,2}^{SM1}, z_{b,3}^{SM1}, z_{b,4}^{SM1}, z_{t,1}^{SM1}, z_{t,2}^{SM1}, z_{t,3}^{SM1}, z_{t,4}^{SM1}; \\
 \text{SM2 : } & z_{b,5}^{SM2}, z_{b,6}^{SM2}, z_{b,7}^{SM2}, z_{b,8}^{SM2}, z_{t,5}^{SM2}, z_{t,6}^{SM2}, z_{t,7}^{SM2}, z_{t,8}^{SM2}; \\
 & C_z \text{ and } z_{n,f}(0).
 \end{aligned} \tag{4.15}$$

where $z_{b,i}$ is the "lowest" z coordinate of chip i and $z_{t,i}$ is the "topmost" z coordinate of chip i , respectively (4.15). The superscripts 'SM1', 'SM2' denote the corresponding CCD mosaic to which these parameters belong. C_z is the inverse of the pixel spacing in z -direction and $z_{n,f}(0)$ are calibration parameters.

Chapter 4. Star transit simulation

We use the configuration of the DIVA focal plane that was delivered by Astrium in Preliminary Design Review [54] to compute the values of the nominal location of the CCD chips. The focal plane is characterized by the following general quantities:

Chip size in w direction:	13.824 mm
Chip size in z direction:	27.648 mm
w distance between SM1 & SM2 readout registers:	34 mm
z shift between SM1 & SM2:	1.5 mm
w, z origin:	center of SM2
z gap between chips:	1.5 mm
Total z extent of SM1:	[-56.046, +59.046] mm
Total z extent of SM2:	[-57.564, +57.546] mm
Total z height of a mosaic:	115.092 mm

From these quantities one can calculate the physical location of the individual chips. They are given in table 4.2 in focal-plane coordinates (in mm) (to convert these into field coordinates in (rad), one has to divide by the focal length of the telescope: $F=11.2$ m).

Table 4.2: The geometric arrangement of the nominal DIVA SM CCDs.

SM, chip	z_b (mm)	z_t (mm)
SM1, 1	-56.046	-28.398
SM1, 2	-26.898	+00.750
SM1, 3	+02.250	+29.898
SM1, 4	+31.398	+59.046
SM2, 5	-57.546	-29.898
SM2, 6	-28.398	-00.750
SM2, 7	+00.750	+28.398
SM2, 8	+29.898	+57.546

Computation of field coordinates from pixel coordinates

Because only integer pixel coordinates are interested for the prediction of star transits, grid coordinates for the detections are simply: $G_d = 0$ and $H_d = m_d$ per construction (see 4.6.4). The calibration of the instrument should be completed during the commissioning phase. This calibration includes displacements, rotation and tilts of each CCD chip with respect to the FOV as well as distortions of the CCD chips themselves. This calibration is taken into account by the transformation (see [10]):

$$w_d(G_d, H_d) = w_{n,f}(H_d) \quad z_d(G_d, H_d) = z_{n,f}(H_d), \quad (4.16)$$

where w and z are the field coordinates, f is the field-of-view index and all the other quantities are calibration functions.

Prediction of field coordinates of SM2 transits.

First of all we exclude transit candidates which will not lie on one of the SM2 CCD chips, the following check must be performed – possible SM2 detections at field coordinate z_d must lie within the ranges

$$\begin{aligned} (z_{b,1}^{SM2} + X) \dots (z_{t,1}^{SM2} - X) &\longrightarrow n_p = 5, \\ (z_{b,2}^{SM2} + X) \dots (z_{t,2}^{SM2} - X) &\longrightarrow n_p = 6, \\ (z_{b,3}^{SM2} + X) \dots (z_{t,3}^{SM2} - X) &\longrightarrow n_p = 7, \\ (z_{b,4}^{SM2} + X) \dots (z_{t,3}^{SM2} - X) &\longrightarrow n_p = 8, \end{aligned} \quad (4.17)$$

where X is a margin which guarantees, that complete windows can be extracted. In our simulation we take X to equal one half of the SM-window in z -direction. With known (predicted) CCD chip number n_p , the predicted grid coordinates can be computed

$$H_p = \frac{1}{C_z}(z_d - z_{n_p,f}(0))$$

Per construction the angular distance in scan direction between the detected image and the predicted transit is the angular distance between the w -coordinates of the relevant readout registers [9] (see fig. 4.15), so

$$\Delta w(z_d) = w_{n_{SM2}}(H_p) - w_{n_{SM1}}(H_d).$$

For the cross-scan direction an analogous expressions holds, which vanishes due to the construction of G_p :

$$\Delta z = 0.$$

Prediction of pixel coordinates of the window corner of SM2 transits

With these field coordinate differences, pixel coordinates of window corners for the predicted transits on SM2 for the chip number n_p (see eq. 4.17) can be calculated:

$$\begin{aligned} \underline{k}_{w_p} &= \text{NINT} \left(\frac{\Delta w(z_d)}{s_n} - \frac{L_w}{2} \right), \\ \underline{m}_{w_p} &= \text{NINT} \left(H_p - \frac{L_z}{2} \right), \end{aligned}$$

where s_n is the linear size of one CCD pixel and $L_w = 12$ and $L_z = 7$ are the widths of the normal SM window (scan direction and in cross-scan direction, measured here in number of pixels).

4.8.7 Star rate determination and TDI clock stroke rate adjustment

One of the most important parts of the on-board software is the attitude control system. The attitude control system provides the instrument control software with the estimated attitude and the speed of the satellite. In order to install the correct TDI clock stroke rate, it is mandatory to determine the speed of the satellite better than 0.1 arcsec/sec in scan direction. The science data are going to be used in order to ensure the fulfillment of this requirement.

Let us describe the emulation procedure of the work of the on-board software in the framework of our star transit simulator. It allows to mimic (to a certain extent) the on-board software, performing the star rate determination and thus TDI clock stroke adjustment with the help of the analysis of the science data. For this purpose, one can use

- the pair transits (attitude and/or all stars).

In this case, we can use either attitude stars or all stars in order to compute the TDI clock stroke rate. First of all, we predict the true star transits. Then, the centroid errors (in scan direction) are simulated for κ_{obs} : ± 0.15 pix (fine centroiding) if these are attitude stars and ± 1 pix (coarse centroiding) for all other stars. We compute now the difference between the centroids κ_{obs} of the star transit pairs on SM1 and SM2

$$D_{obs} = \kappa_{obs}(SM2) - \kappa_{obs}(SM1).$$

We keep these values in a buffer⁴. As soon as the buffer is full we calculate the average value of D_{obs} and determine the known TDI clock stroke rate by the following

$$TDI_{known}^{rate} = \frac{TDI_{true}^{rate} \cdot \overline{D}_{obs}}{D_{true}}$$

where D_{true} – the true value of the difference between read-out registers on SM1 and SM2 in TDI clock stroke units. In the nominal case (with nominal TDI clock stroke rate and nominal geometrical calibration) the value of the D_{true} is equal 2518.52.

- the transit of stars with known positions, i.e. attitude star transits. In this case we have to compute the prediction of attitude stars according to the scanning law (through the calculation of the attitude matrix).

As above, we simulate firstly the true star transits from the true sky and calculate the coordinate of its centroids κ_{obs} (scan direction) with errors. The predicted field coordinates of attitude stars are calculated then from the attitude star catalogue,

⁴The size of buffer is to be established.

treated in the same way as the stars of the true sky, but with simulated errors in positions κ_{pred} (the typical error for the attitude catalogue is about 0.05-0.07 arcsec/sec). We keep these observed and predicted centroids in the buffer.

Then, we can calculate the difference for the observed and predicted field coordinates, i.e. ($\Delta\kappa_{obs}$ and $\Delta\kappa_{pred}$), as well as the average values for observed and predicted $\Delta\kappa_{obs/pred}$:

$$\overline{\Delta\kappa_{obs/pred}} = \frac{\sum_{i=1(i \neq j)}^n \Delta\kappa_{ij}}{C_n^2},$$

where $C_n^2 = \frac{1}{2}(n^2 - n)$ – the number of all possible combinations. Finally, we find the known TDI clock stroke rate:

$$TDI_{known}^{rate} = \frac{TDI_{rate}^{true} \overline{\Delta\kappa_{obs}}}{\overline{\Delta\kappa_{pred}}}$$

If TDI_{known}^{rate} deviates from the TDI_{true}^{rate} more than 0.5 lines/sec that corresponds to the difference in the scan rate of ~ 0.1 arcsec/sec (but it is still within the margin of an allowed scan rate, which is 180 arcsec/sec $\pm 2\%$), then the new TDI_{known}^{rate} must be accepted as a true TDI clock stroke rate. If the difference exceeds the threshold then the jet-firing should be initiated first. The actual jet-firing strategy is without scope of this work. It will be optimized during further development of the satellite.

For both of these methods one should take into account the following uncertainties: the errors of the position of stars in the attitude star catalogue, the errors of centroiding algorithm, the errors of known attitude matrix; the errors of TDI clock stroke.

Chapter 4. Star transit simulation

Chapter 5

Monitoring proper and Diagnosis

The task of the ScQL monitoring is to reveal anomalies in the science data and to determine why the DIVA main instrument is not functioning as intended. This chapter describes the monitoring proper and diagnosis parts of the ScQL. The list of the statistics and diagnostics is presented and the explanation of the possible faulty situations is discussed.

5.1 ScQL monitoring dataflow

Figure 5.1 shows the overall data flow foreseen for ScQL monitoring. We suppose that uncompression, formatting and structuring of the incoming telemetry data from the satellite has already been performed and the data stream is ingested into the DIVA database. The initial data treatment is the first step of the scientific data reduction chain and applies certain basic operations to the raw CCD data in order to generate image parameter sets. It comprises all those pre-processes which are needed for a quick assessment of the scientific quality of the science data in the framework of ScQL monitoring, namely, pixel data processing (centroiding) and cross matching (matching and providing the observed star images with a star identifier ID). In our approach we generate the data output of the initial data treatment (pixel data processing) in the framework of the developed star transit simulator. The simulated output from the initial data treatment is then retrieved from the DIVA database by ScQL monitoring proper where the statistics and parameters are estimated, residuals and alarms are generated; the results are presented as tables, plots and histograms for the operator and saved into the database for the following diagnosis. The diagnosis evaluates symptoms, generated by monitoring proper; it retrieves the residuals and appropriate statistics, from the database and provides the operator with the possible interpretation of a problem that has been alarmed in monitoring proper.

All of these main components of the monitoring software are initiated by the ScQL monitoring manager as a chain of processing steps, each of them implemented in the

Chapter 5. Monitoring proper and Diagnosis

form of an individual package. In the previous chapter described in detail the model of the star transit simulator, now we will describe two other main components of our ScQL system – monitoring proper and diagnosis – where all jobs concerning the detection, localization and identification of the abnormal behavior of the satellite are performed.

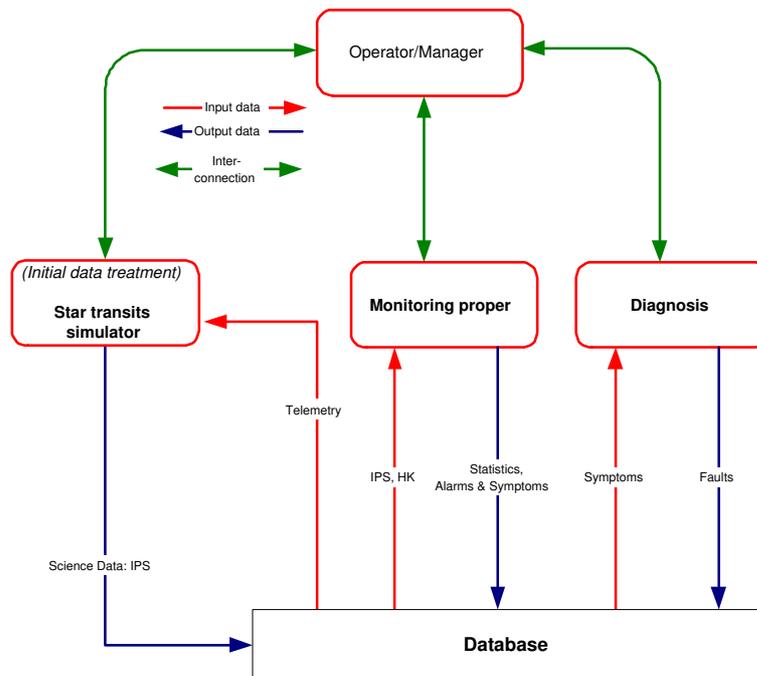


Figure 5.1: Dataflow of the ScQL monitoring (in simulation).

5.2 Monitoring proper

5.2.1 The scheme of the monitoring proper

As it was mentioned the ScQL monitoring should provide the operator with the graphical user interface in order to display estimated statistics and parameters in the terms of plots, tables and histograms.

The ScQL monitoring proper visualization and manipulation tool is based on elements of a powerful Foundation Class Layer (FCL), developed in [64], which helps the programmer to create new data analysis applications. The two primary classes of FCL, Visualizer and the DataForm, were adopted and modified for the task of the monitoring proper. The Visualizer provides a complex widget tool for interactive visualization of large ScQL data structures, and hides the complexity of original IDL¹ widgets and

¹IDL – Interactive Data Language, Research System Inc. See sec. 5.4 where the environments and

graphics behind easy-to-use IDL classes. The DataForm helps with the creation of ScQL monitoring proper user interfaces. In figure 5.2 the structure of the ScQL monitoring proper is presented. The original DataForm class has been extended so that it allows to perform an access to the DIVA database as well as to any external DIVA application which produces specific statistics and performs parameters estimation. An external application is invoked from IDL according to our mixed-language programming specifications for DIVA applications (see sec. 5.4). Finally, we get a flexible software tool that provides sophisticated user interfaces with visualization capabilities (using IDL) and robust numerical algorithms (using C and/or Fortran).

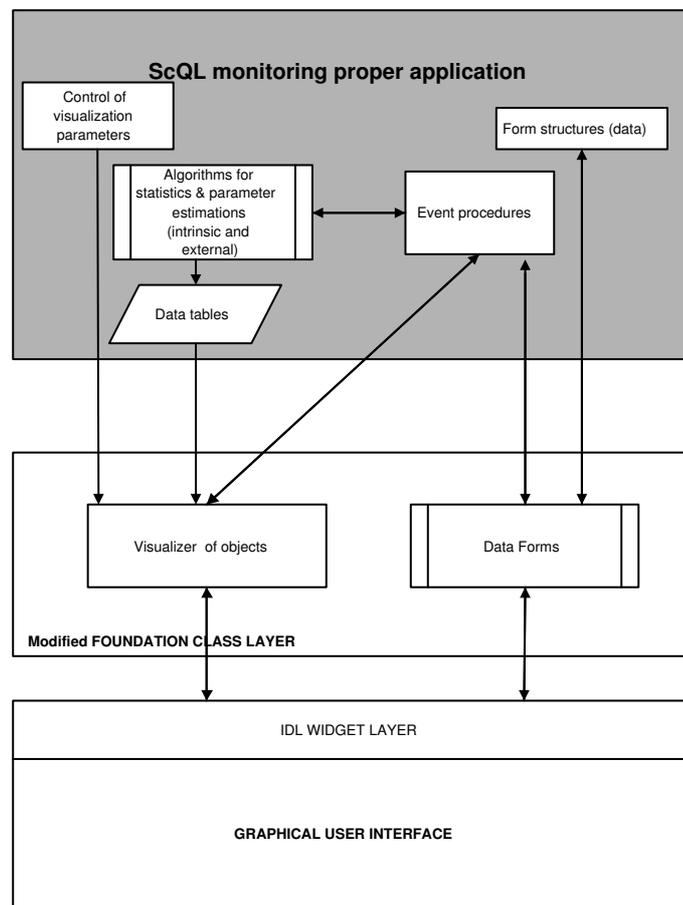


Figure 5.2: Structure of an ScQL monitoring proper employing the MFCL.

In details we will further discuss the main characteristics for the important component of the ScQL monitoring proper where the statistics and parameter estimations will be produced and analyzed.

the concept of the mixed-language programming for the ScQL monitoring are discussed.

5.2.2 Statistics, parameters and residuals.

The list of the items, that must be monitored by the ScQL monitoring proper, includes a wide range of items at various levels of aggregation of the observational data. One should stress, however, that at this stage of the project this list singles out only the major quantities of the statistics and parameters that should be monitored in the framework of the ScQL. It will evolve and grow with the further development of the satellite.

- The ScQL monitoring software shall be capable of displaying the supply voltages downstream of the ADU CCDs, and will rise alarms when the voltages fall outside acceptable limits.
- The ScQL monitoring software shall be capable of displaying the temperatures of the sensors of the focal plane and check whether the thermal status is within the nominal one.
- The ScQL monitoring software shall be capable of displaying and plotting the sets of image processing parameters.
- The ScQL monitoring software shall produce and plot the rates of the number of downloaded windows for the whole CCD mosaic, for a given CCD chip, for a given ADU chain, for a given pixel column block of a given ADU chain and calculate their residuals.

Let N_w – be the number of the downloaded windows (or window rate) from *the observed sky* and let N_{exp} be the number of the windows as expected from *the known sky*.

1. For the time t compute the number of the downloaded windows for the whole SM CCD mosaic and calculate its residual, i.e. the ratio:

$$r_w^{CCDs}(t) = \frac{N_w^{CCDs}(t) - N_{w.exp}^{CCDs}(t)}{N_{w.exp}^{CCDs}(t)}, \quad (5.1)$$

and arise alarm flag $\epsilon_w^{CCDs}(t) = 1$ if

$$\|r_w^{CCDs}(t)\| \geq \eta_w^{CCDs}(t),$$

where $\eta_w^{(*)}(t)$ hereafter denotes the threshold.

2. For the same time t calculate the number of the downloaded windows and its residual for a given SM CCD chip number $n(1 \div 8)$

$$r_w^n(t) = \frac{N_w^n(t) - N_{w.exp}^n(t)}{N_{w.exp}^n(t)},$$

and arise alarm flag $\epsilon_w^n(t) = 1$ if

$$\|r_w^n(t)\| \geq \eta_w^n(t).$$

3. For the time t calculate the number of the downloaded windows and its residual for a given ADU chain (for upper $m=[255 \div 511]$ pixels, for lower $m=[0 \div 254]$ pixels) of the chip number n :

$$r_w^{n(ADU_u/ADU_l)}(t) = \frac{N_w^{n(ADU_u/ADU_l)}(t) - N_{w.exp}^{n(ADU_u/ADU_l)}(t)}{N_{w.exp}^{n(ADU_u/ADU_l)}(t)},$$

and arise alarm flag $\epsilon_w^{n(ADU_u/ADU_l)}(t) = 1$ if

$$||r_w^{n(ADU_u/ADU_l)}(t)|| \geq \eta_w^{n(ADU_u/ADU_l)}(t).$$

4. For the time t compute the number of the downloaded windows $N_w^{n<m_i,m_j>}$ for given column blocks of chip number $n(1 \div 8)$. Calculate then its residual, i.e. compute the following ratio

$$r_w^{n<m_i,m_j>}(t) = \frac{N_w^{n<m_i,m_j>}(t) - N_{w.exp}^{n<m_i,m_j>}(t)}{N_{w.exp}^{n<m_i,m_j>}(t)},$$

where $\langle m_i, m_j \rangle = \langle 0...15 \rangle, \langle 16...31 \rangle, \langle 32...47 \rangle, \langle \dots \rangle, \langle 496...511 \rangle$ possible pixel blocks. Arise alarm flag $\epsilon_w^{n<m_i,m_j>}(t)$ if

$$||r_w^{n<m_i,m_j>}(t)|| \geq \eta_w^{n<m_i,m_j>}(t).$$

From these quantities different plots will be produced in order to control the monitoring and diagnosis:

- time-slice histograms of window rates N_w^{CCDs} , N_w^n , $N_w^{n(ADU_u/ADU_l)}$, $N_w^{n<m_i,m_j>}$, $N_{w.exp}$ vs. \underline{k}_w (in TDI clock stroke or julian time);
- time-slice histogram of the residual r_w^{CCDs} , r_w^n , $r_w^{n(ADU_u/ADU_l)}$, $r_w^{n<m_i,m_j>}$ with the corresponding thresholds η_w^{CCDs} , η_w^n , $\eta_w^{n(ADU_u/ADU_l)}$, $\eta_w^{n<m_i,m_j>}$ vs. \underline{k}_w (in TDI clock stroke or julian time).
- The ScQL monitoring shall count the number of empty windows (windows containing no star image) for the whole CCD mosaic, for a given CCD chip, for a given ADU chain, for a given pixel column block of a given ADU chain.

For a given time t find the numbers of empty windows N_{ew}^{CCDs} , N_{ew}^n , $N_{ew}^{n(ADU_u/ADU_l)}$, $N_{ew}^{n<m_i,m_j>}$ (for the CCDs mosaic, chip, ADU chain, pixel block) and compute the residuals with regard to the numbers of the downloaded windows

$$r_{ew}^{CCDs}(t) = 1 - \frac{N_{ew}^{CCDs}(t)}{N_w^{CCDs}(t)}$$

$$r_{ew}^n(t) = 1 - \frac{N_{ew}^n(t)}{N_w^n(t)}$$

$$r_{ew}^{n(ADU_u/ADU_l)}(t) = 1 - \frac{N_{ew}^{n(ADU_u/ADU_l)}(t)}{N_w^{n(ADU_u/ADU_l)}(t)}$$

$$r_{ew}^{n<m_i,m_j>}(t) = 1 - \frac{N_{ew}^{n<m_i,m_j>}(t)}{N_w^{n<m_i,m_j>}(t)}.$$

Chapter 5. Monitoring proper and Diagnosis

Ideally, these residuals should be zero, because each window must contain an image. In the case of empty windows the real minimum thresholds above which an alarm arises should be derived from the commission phase of the satellite work.

Plot time-slice histogram of the $N_{ew}^{CCDs,n,n(ADU_u/ADU_lADU),n<m_i,m_j>}$.

- The ScQL monitoring software shall produce the accumulated output of the on-ground centroiding algorithm (pixel data processing) from many events (images).

Let C_k^l , C_m^l be the along-scan and across-scan centroid of the star image (i.e. that resulting from the on-ground centroiding algorithm) in l^{th} window. We assume at this stage of the reduction model that every window contains only one star. Multiple object analysis can be incorporated in the future.

From each window containing a star image calculate the centroid of the star image C_k^l , C_m^l by the centroiding algorithm. From these data compute the the average centroid position for the time interval δt

$$C_K = \frac{1}{L} \sum_{\delta t} C_k^l$$

$$C_M = \frac{1}{L} \sum_{\delta t} C_m^l$$

and its uncertainty

$$\Delta C_K = \frac{1}{L} \sqrt{\sum_{\delta t} (\Delta C_k^l)^2}$$

$$\Delta C_M = \frac{1}{L} \sqrt{\sum_{\delta t} (\Delta C_m^l)^2}$$

where ΔC_k^l and ΔC_m^l are the along- and across-scan uncertainties in the centroid position for an individual star image provided by the centroiding algorithm; L is the number of the observations (downloaded windows) in the δt time-interval. Additionally calculate the standard deviations

$$\sigma_{C_k^l}^2 = \frac{1}{L-1} \sqrt{\sum_{\delta t} (C_K - C_k^l)^2}$$

$$\sigma_{C_m^l}^2 = \frac{1}{L-1} \sqrt{\sum_{\delta t} (C_M - C_m^l)^2}$$

of the distribution and its uncertainties.

Arise a flag if $(C_M, \Delta C_M)$ and/or $(C_K, \Delta C_K)$ differ more then k% from their expected values: $(C_K^{exp} = 6.0, \Delta C_K^{exp})$ and $(C_M^{exp} = 3.5, \Delta C_M^{exp})$. The expected values of ΔC_K^{exp}

and ΔC_M^{exp} should be established from the the image simulation and/or mission experience.

The ScQL monitoring software shall be capable to plot a histogram of the average centroids (and standard deviations) of the star images calculated from the downloaded windows as function of time; this histogram can be overplotted with the expected values of the average centroid coordinates and standard deviations.

- The ScQL monitoring software shall calculate:
 - the difference ΔA between the measured amplitude and the expected one (from attitude star catalogue) of the attitude stars for a given ADU;
 - the difference ΔW_k (and ΔW_m) between the measured width and the width as expected from the color of the attitude star;
 - for all background pixels the median

$$M_{kmn} = med(S_{kmn})$$

and its scatter

$$\Delta M_{kmn} = med(S_{kmn} - M_{kmn}).$$

Note: we will not consider the implementation of these items in our prototype as we do not perform the simulation at the pixel level.

- The ScQL monitoring software shall analyse the brightness of the stars.

Analyse the brightness distribution function of observed stars and compare it with the expected one.

- To analyse the brightness distribution the Kolmogorov-Smirnov test is used. We create the observed cumulative distribution by converting the list of data to an unbiased estimator $S_N(V)$ (see [27]). A theoretical brightness distribution $N_{th}(V)$ is derived from the nominal simulation according to the model of the Galaxy (*the known sky*). We use the Kolmogorov-Smirnov D statistic to compare the observed brightness distribution function and expected (theoretical) one. It is defined as the maximum value of the absolute difference between these two cumulative functions:

$$D = \sup_{-\infty < V < \infty} |S_{N_{obs}}(V) - S_{N_{exp}}(V)|$$

The significance level of an observed value D (as a disproof of the null hypothesis that the distributions are the same) is then calculated. The function that enters into the calculation of the significance is written as the following sum ([27]):

$$Q_{KS}(\lambda) = 2 \sum_{j=1}^{\infty} (-1)^{j-1} \exp(-2j^2 \lambda^2)$$

Chapter 5. Monitoring proper and Diagnosis

which is a monotonic function with the limiting values $Q_{KS}(0) = 1$, $Q_{KS}(\infty) = 0$. In terms of this function, the significance level of an observed value of D is given approximately [65]:

$$Prob(D > \text{observed}) = Q_{KS}([\sqrt{n_e} + 0.12 + 0.11/\sqrt{n_e}] D), \quad (5.2)$$

where $n_e = \frac{n_1 n_2}{n_1 + n_2}$ is the effective number of data.

- If the null hypothesis is rejected at the defined significance level, we calculate then the residuals in the form of absolute difference D to determine the problematic brightness interval.

Plot a histograms for the observed and expected brightness distribution of the stars as well as the residuals.

- The ScQL monitoring software shall compute the TDI clock stroke rate.

Due to the rotation of the satellite the stellar image are moving from the left to the right in the focal plane. All CCDs are clocked synchronously with the actual rotation of the satellite. They are operated in time delay integration mode and have an integrated exposure time per CCD mosaic. If the shifting of the charge on the CCD does not match the actual rotation of the satellite but this mismatch is not yet large enough to initiate the jet-firing then the new TDI clock stroke rate that is synchronized with scan-speed of the satellite must have been accepted as an actual one

On-board software with built-in monitoring capabilities performs the same analysis of the science data for the rate determination and TDI clock stroke adjustment that is done on the ground where the work of the on board software should be verified. So, for this purpose we use the same algorithms that were proposed and described in previous chapter 4 in sec. 4.8.7.

However, one should mention that in the case of the on-board software this analyse must be done in real time and only with the knowledge about the past, i.e. with the observations performed up to the certain(current) time, whereas on the ground one can use all available information (from the past and the future from the view point of the event when the satellite rate change occurs) in order to verify more accurate whether the on-board TDI clock stroke rate adjustment algorithm is working correctly. This kind of TDI clock stroke rate verification shall be probably done in the framework of the First Look where more sophisticated data analysis and diagnosis will be performed.

5.3 ScQL diagnosis

The diagnosis task of ScQL monitoring gives an explanation for the faulty behavior the spacecraft. Another words the diagnosis task has to determine what could be wrong with the satellite if it delivers bad observations. In this section we present an initial list of possible explanations of the problems, alarmed/flagged by the ScQL monitoring proper.

Using this list and the list presented for monitoring proper (5.2.2) we will develop the sets of the residual structures for the residual evaluation phase in the framework of the ScQL diagnosis. We would like to stress that the following list should further grow iteratively with the development and evaluation of the concept of the satellite. Using the statistics and estimated parameters, detected by the ScQL monitoring proper (see 5.2.2), we describe possible interpretations of abnormal events. This knowledge will lead us to the residual structures that should be further implemented in the diagnosis engine of our monitoring system. Some faults described in the following list can not be uniquely identified without more detailed simulation on the pixel level of the science data generation. To simplify the presentation of the faults we split the entire group of the faults into two subgroups namely **distinguishable** and **indistinguishable** faults in the framework of our model of the star transits.

5.3.1 Distinguishable faults

Let us start to consider the analysis based on the most trivial diagnostic – ”no windows from a CCD”. At first we define what is meant by the expression ”no windows from a CCD”. The only information, relevant in this context are the along-scan and across-scan pixel coordinates ($\underline{k}_w, \underline{m}_w$) of a window and CCD designation n , or the lack of this information, respectively. If the ScQL data contains no windows with along-scan and across-scan pixel coordinates within a given δt time interval² and a CCD designation for a particular CCD, we call this ”no windows from a CCD”. The total lack of any windows in a CCD can easily be noticed by using the appropriate statistic item (see subsection 5.2.2). In the case of a complete absence of windows in one CCD chip the corresponding $r_w^{CCD}(t)$ would vanish or decrease (this anomaly will be discussed below).

There are several possible causes for this anomaly which will be discussed in the following:

1. Let us assume that **none of the DIVA CCDs delivers any window** to the ground. The reason for this could be:
 - i. The detection algorithm of the instrument processing unit (IPU) fails. Typical detection algorithms determine the background and accept only those pixels as possible pixels belonging to a star image whose signal is sufficiently larger than this background. A global factor (valid for a CCD) is used to determine a threshold above which detections are possible and to distinguish between noise and stars (typical factor is about 3 times the background value).
 - ii. The MMU (Mass Memory Unit) might be full (seldom, e.g. due to missing passages or scanning along the galactic plane). In this case already stored data will be overwritten and therefore missing. Science data probably will be

²The time interval δt should be chosen large enough to provide a reasonable statistics. We use 2–10 minutes time interval for our prototype.

Chapter 5. Monitoring proper and Diagnosis

overwritten earlier than HK data so that QL diagnostics can be performed (with some delay).

- iii. The telemetry is interrupted. That is definitely visible by QL too, i.e. in HK data. ScQL will read the report from QL and decide that's telemetry is interrupted.

2. No windows in a given CCD.

We now want to address the absence of windows in individual CCD. Several causes might be responsible for this anomaly:

- i. The failure of both ADU chains of a given CCD occurred.
- ii. The prediction algorithm of the IPU fails. Basically prediction means calculating firstly the TDI clock stroke k_w at which the corresponding window should be cut out at CCD read-out registers, secondly the across-scan pixel coordinate m_w of the window to be cut-out and thirdly the CCD identifier. To do this the prediction algorithm has to know the instrument calibration data. That are the spatial along-scan offsets between consecutive read-out registers, the across-scan shifting of the CCDs and their rotation against the nominal placing at the focal plane. With the aid of this calibration data and the knowledge of the scan-speed, the algorithm predicts the along-scan coordinate k_w^{pred} , the across-scan coordinate m_w^{pred} and the CCD identifier n^{pred} from the parameters of the detected star. Receiving no windows from a particular CCD during a time interval can be due to three possible failures, all caused by bit shifts in the calibration data (for example, after the passage of a high-level cosmic radiation particle):
 - A. The CCD identifier n^{pred} is out of range. It means in this context either that the CCD identifier is out of range ($n^{pred} > 20$ for the total number of DIVA CCDs, or out of range of the allowed CCD identifier for the considered instrument, in the case of sky mapper (SM1 and SM2) $n^{pred} > 8$ or $n^{pred} < 0$ for sky mapper) or that the CCD identifier on SM1 is not that of a consecutive CCD on SM2.
 - B. The across-scan coordinate m_w^{pred} of the window is out of range. This means that m_w^{pred} is larger (or smaller than 0) than the total number of CCD pixels in that direction, i.e. $m_w^{pred} > 511$ effective pixels.

3. The number of downloaded windows does not match the expected one.

The expected rate of downloaded windows is $N_w^{CCD,n,ADU,<m_i,m_j>}(t)$ (see subsection 5.2.2). Deviations from this expected rate are the fault under consideration here. The deviations can be divided in two regimes:

- i. The rate of windows is larger than the expected one.

A real increase of the window rate would only produce windows with very faint stars. This could occur if the detection algorithm is damaged by the passage of a high-level cosmic radiation particle, so that the threshold above which detection is performed is diminished. These faint stars would be probably excluded from the final ScQL window diagnostics, which should contain relatively bright stars. Nevertheless, this fault needs to be carefully investigated in the framework of First Look.

ii. The rate of windows is smaller than the expected one.

In this item we will describe the possible faults that reduce the rate of cut-out windows compared to expected ones. There are several possible reasons for this fault type:

A. The object detection fails.

Detection is performed on the sky mapper SM1. Only those pixels of the sky mapper data stream are used for the on-board detection whose signal is sufficiently larger than the mean background at that sky region. Typically detection is initiated at a threshold of a factor (3 or so) times the background value. If this factor is shifted by the passage of a cosmic particle, the threshold could be increased so that only brighter stars can be detected. The reduction of the window rate is then homogeneous for SM1 and SM2.

B. A 'loose of connection' from CCDs to MMU.

C. MMU is filled up for some time.

D. Strongly increased background.

A very high background could influence the detection rate because the threshold above which detection sets in would be increased, too. This could in principle decrease the number of windows at the faint end (similar to A.).

E. Overloading of the detection algorithm.

In the vicinity of the galactic plane the number of stars to be detected increases rapidly. Scanning in the galactic plane will overload the on-board detection algorithm so that stars are missing stochastically. Details of this malfunction are beyond the scope of this thesis.

4. **All (some) of the extracted windows are empty.**

At first we define the meaning of the term "empty window". We call a window empty if it contains no star image but only background. (In terms of the star transit simulation we consider the coordinates of the centroid instead of the real image.) There are several possible causes for empty windows:

i. Empty windows in 'detection mode' (on SM1).

Chapter 5. Monitoring proper and Diagnosis

This can be caused either by a faulty window on-board centroiding algorithm or window cutting algorithm. In principle at this stage of the study it is impossible to distinguish between these two cases. However, it is more realistic that the windows are cut out systematically at a false positions (coordinates) in data space than the centroiding algorithm works wrong. The false of \underline{k}_w (or m_w or n) will be then misleadingly predicted on SM2.

ii. Empty windows in 'prediction mode' (on SM2).

A. Damaged instrument calibration data.

As a consequence of changes ("a bit-shift" of a CCD) in the instrument calibration, the predicted pixel coordinates of the windows (\underline{k}_w, m_w) to be cut out might change, so that the window is cut-out at a false area of CCD. A bit-shift change of \underline{k}_w, m_w would result in the events for a certain CCD/CCDs (because each detected window will produce its own \underline{k}_w, m_w).

So, the predicted window pixel edge \underline{k}_w, m_w could be systematically wrong, if the calibration data has changed and the offsets between consecutive CCDs are damaged.

B. The attitude is incorrect.

If the attitude (this is the rotational velocity or the scan-speed in this context) is incorrectly predicted, the adjusted shifting of charges in the CCDs (TDI clock stroke rate) will no longer match the real scan-speed of the satellite. Firstly this will smear out the star image (see below "all images blurred") and secondly this would produce badly predicted windows. This fault arises, if the scan rate is not correctly determined, either by a software fault or by an impact of micrometeorite.

5. **The centroid of the star image in a window does not coincide with the window center.**

To get the most of the data a window it should contain as much signal of the observed star as possible. Therefore the stellar centroid should be located in the center of the window. Of course the on-board detection will have some positional error budget (up to ~ 2 pixels in a single measurement). From this we get an expected mean centroid of the distribution and standard deviation as described in subsection 5.2.2 (monitor diagnostics and plots).

These expected values ($C_K^{exp} = 6.0$ and $C_M^{exp} = 3.5$) can be compared to those resulting from δt -intervals. The result of this comparison may be twofold:

i. The mean centroid of all stars is shifted with respect to the center of the window.

A. Mean centroid shifting in "detection mode" CCDs (SM1).

At the moment only one cause seems to be able to produce this fault type: the detection, centroiding and window cutting algorithms work improperly. This must be proven during the commissioning phase. Therefore, we could only point out at this stage of the study that the detection software works improperly and has to be modified.

- B. Mean centroid shifting in "prediction mode" CCDs (SM2). There seem to be two causes that could produce shifts of the mean centroid of star images in prediction mode windows:

- * (a) Damaged instrument calibration data.

This fault type is nearly an exact copy of ScQL diagnosis 4-ii-A (above). The difference is in the amplitude: whereas the shifts in the predicted k_w completely displace the windows to be cut out in the case of ScQL diagnosis 4-ii-A (so that the star image is no longer situated in it), the shifts under consideration here are much smaller and displace the star image centroids only within the window ($[12 \times 7]$ pixels).

- * (b) The attitude is incorrect.

This fault type is nearly an exact copy of ScQL diagnosis 4-ii-B. The difference is in the amplitude: whereas the shifts in the predicted k_w completely displace the windows to be cut out in the case of ScQL diagnosis task 4-ii-B (so that the star image is no longer situated in it), the shifts under consideration here are much smaller and displace the star image centroids only within the window ($[12 \times 7]$ pixels).

- ii. The standard deviation of the centroid distribution is larger than expected. Of course the standard deviation may be smaller than that expected, but this would imply that DIVA's on-board centroiding or/and prediction accuracy is better than expected. This needs not to be handled.

- A. Increased standard deviation of the centroid distribution in detection mode CCDs (SM1).

The remarks of ScQL debugging 5-i-A hold here, too.

- B. Increased standard deviation of the centroid distribution in prediction mode CCDs (SM2).

If the standard deviation of the centroid distribution of star images of the detection mode CCDs on SM1 is known to match the expectations, the prediction algorithm should not be able to change this standard deviation but could only change the mean of the centroid distribution.

(Or maybe it is possible to connect with the sensitivity of CCDs on SM2?)

6. **Wrong TDI clock stroke rate.** The shifting of charges on the CCD does not match the scan-speed of the satellite. A maladjusted shifting of charge would

Chapter 5. Monitoring proper and Diagnosis

transport the charge at a different speed across the CCD than the star image is transported across the focal plane.

7. Blurred images.

There are a few possible causes that would result in some blurring of star images. Blurred windows are those whose width w is enlarged compared to that expected from the color of the star and whose peak amplitude A is reduced. Using the monitoring ScQL diagnostics ΔA and Δw one would get some Gaussian distribution in case of a nominally working instrument. The δt time interval distribution of real ΔA 's and Δw 's will show deviations from this expected distribution. Some broadening of the distribution will show that the images are blurred on average.

The correct means to perform this diagnostics are the δt -min means of ΔA and Δw , namely $\langle \Delta A \rangle_{\delta t}$ and $\langle \Delta w \rangle_{\delta t}$

Are these parameters within given thresholds³? Deviations of these means from the expected values of the properly working DIVA can be caused by different failures:

- **i.** The optics is defocussed. A special simulation has to be performed in the future for these cases on the pixel levels. This will enhance our understanding of the influence of these two different fault types on the shape of the point spread function. In this approach, especially star colors should be used to judge the focus. Using color-dependent means of ΔA and Δw would help to distinguish defocussed optics from maladjusted shifts of charge in respect of the scan speed of the satellite. The latter would yield color-independent blurred images, contrary to the second fault type.

The investigation of the blurred images in this way is beyond of our study at this stage of the project.

- **ii.** The shifting of the charge (wrong TDI clock stroke rate) is badly adjusted. See item 6 above.

5.3.2 Indistinguishable fault

The given concept of the DIVA main instrument was designed in such a way that the star light from both FOVs are combined on the one focal plane. The ScQL performs the preliminary scientific assessment of the science data at the coarse level that doesn't allow to distinguish to which FOV star image belongs. To do this a special data treatment should be carried out in the framework of pipeline data reduction at the Data Center. However, in the framework of ScQL monitoring we can identify the fault for an individual FOV basing on the attitude stars because their positions are known a priori.

³These thresholds must be determined by future detailed simulation with an image generation

5.3.3 Severity of the different faults

Each of the described faults has a different effect on the over-all behavior of the system. Some faults are more severe than the others and, therefore, may need to be handled with higher priority.

We do not consider the aspect of the severity of the faults in our work as we concentrated mostly on the monitoring component of the ScQL. However, we would like point out some general features about the fault severity in the ScQL diagnosis. It depends on the *type of the failure*, the *kind of the CCD mosaic* as well as the *mode* in which it runs different levels of criticality and consequential different levels of the recovery should be foreseen. Indeed, in some cases the failures with the same type are softened relative to the same level of criticality compared to the mode at which it works: in the case of the preceding detection mode SM, there will be stringent limits to the allowed portion of faulty ADU chains because in this case not only detection mode SM windows will be missing but also the corresponding ones in prediction mode. At the same time, the prediction mode SM mosaic is not as critical as the detection mode SM due to the fact that no prediction are made out from these observations.

5.4 Landscape of the environments

In this section we will take a look at the aspect of the implementation of the ScQL monitoring system in the framework of the DIVA data environments. It is evident that an important component for the correct handling of the DIVA big data sets is its database. So, the smooth integration of the different DIVA software tools, in particular, the ScQL monitoring with its modeling and simulation facilities, into the DIVA database infrastructure is mandatory. For this purpose the concept of the mixed-language programming (MLP), that allows a rapid development of the application, has been studied and successfully applied for the ScQL software.

5.4.1 Database and mixed-language programming

The approach for the DIVA database is discussed in [68]. There we wrote that

”The DIVA satellite produces very big data sets that have to be accessed in different ways and sequences. Most data accesses will be using large quantities of homogeneously structured data (i.e. many identically formatted records) at a time. But, for diagnostic purposes, there will frequently be the wish to have a look at just one or a few (or a few ten thousand) records that have to be selected by whatever peculiarity or common diagnostic property they might have. We shall need a strictly organized versioning, and a strictly organized access administration, and etc...”

Chapter 5. Monitoring proper and Diagnosis

So, the conclusion was that the DIVA data reduction should be done on the basis of a database not on a file system (at least in its operational phase). In [53] the detailed consequences of the programming style and data access modes for a "standard" DIVA application have been carefully investigated in the framework of a relational database management system (DBMS) DB2 Enterprise Extended Edition (DB2 EEE). We propose to use the conception of mixed-language programming in order to combine the strength of various programming languages and to reuse the components that have already been written.

Two types of MLP for DIVA applications has been chosen and successfully tested [53]:

- ◆ **Coarse-Grained MLP**, the application consists of multiple executables. The basic notion is that each executable works on an intermediate chunk of data. Once the work has completed, the next executable may work on the chunk of data. It is supposed that each executable consists of a components written in a single language.

Coarse-Grained MLP approach is as a natural way to organize the pipeline for DIVA data analysis.

- ◆ **Mixed Language Executables (MLX)** [52] is the most common form of MLP to combine components of an application by the direct wrapper.

In the most direct form of MLX, the machine (and compiler) dependent recipe to call a foreign routine is contained in the application code for each foreign routine and for each machine. In spite of some problems with compatibility the direct form of MLX⁴ may be tolerable for final DIVA applications running on a single type of machine (cluster of machines) using routines with simple interface. In the case of the DIVA project one should create a direct wrapper providing a native interface to the foreign routine. The standard DIVA application satisfies a typical scheme, and, therefore, creating a wrapper for each routine will not demand tedious efforts.

MLX is used to combine the components of the DIVA application written by means of the database host language(s) to access the database with the components, which perform calculations and are implemented in other programming language(s)

So, writing a DIVA application we don't restrict ourselves with programming languages but only with the interface structure that one should follow in order to meet the requirements of the DIVA database (see [53] for details). However, for the DIVA pipeline reduction chain the programming language C is strictly mandatory as the C is a DB2 database host language in our case because of the following [53]. The DIVA programs are mostly written in Fortran, but the embedded SQL Fortran is not supported by IBM for Linux shared nothing cluster.

⁴Details about MLX specification can be found either in our work [53] or in the original paper [15].

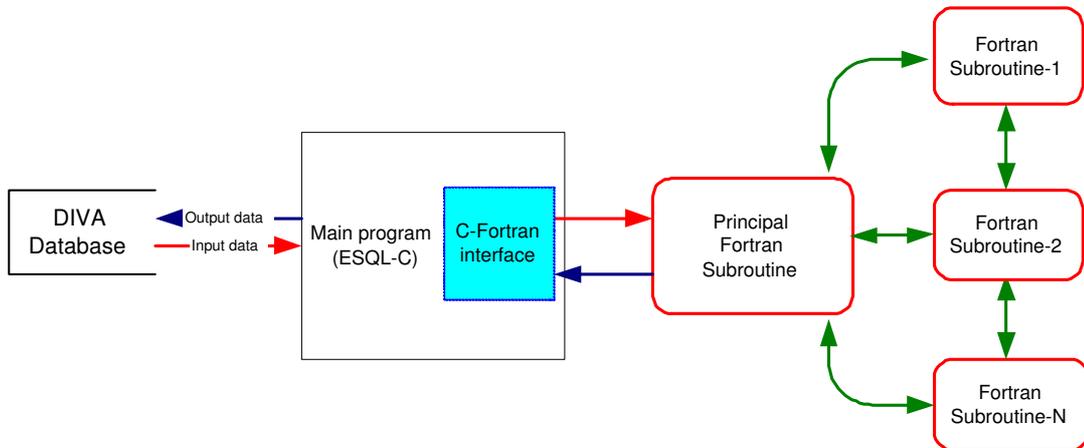


Figure 5.3: The scheme for a typical DIVA application based on mixed language programming (the details see in [53]).

An embedded SQL (ESQL) program consists of code written in an ordinary programming language with the SQL statements inside. Generally, the programming language C is used and mixed with SQL commands in specially marked sections of the program. To build the program, the source code is first passed to the embedded SQL preprocessor, which converts it to an ordinary C program, and afterwards it can be processed by a C compilation tool chain.

By using an MLX approach the program written in Fortran can be easily and effectively combined with ESQL-C, providing therefore an extended program with an access to the DIVA database. Figure 5.3 demonstrates this concept for a typical DIVA application. One should mention that the modern language JAVA could be considered as a host language, however, there is no possibility to combine JAVA and Fortran with a direct wrapper.⁵

5.4.2 ScQL monitoring and MLP

The ScQL monitoring inherits the approach of the MLP to build a flexible software, compatible with the DIVA database. So, we use **ESQL-C** to access the database and **MLX direct C wrapper** to exchange the data with the ScQL monitoring components written in three different languages:

⁵In order to combine Fortran and Java components of the application one should use the C interface as an intermediate wrapper between Fortran and Java. This approach is applied at the time of this writing for Gaia (GDAAS) [3, 66] but can not be considered as an optimal solution at least for "core processes" [29].

Chapter 5. Monitoring proper and Diagnosis

FORTRAN

The programming language Fortran is often referred to as a nearly universal language for scientific numerical calculations. We use Fortran-90/95 (an object oriented version of Fortran) for the star transit simulator and for the ScQL monitoring proper when massive numerical calculations are required.

IDL

The ScQL monitoring proper should have also a graphical user interface to provide the operator with interactive tools and display the statistics and estimated parameters in the form of tables, plots and histograms. For this purpose, IDL RSI(Interactive Data Language, Research System Inc.) is chosen as the visualization & processing package. It provides tools for visualization, data analysis and cross- platform application development [56]. IDL is actively used by the astronomical community around the world, and there exist many lines of code specifically dedicated to astronomical data analysis needs. Another advantage an IDL Virtual Machine (VM) that is available with IDL 6.0 and above: IDL VM is a freely distributed, cross-platform utility for running compiled IDL code and therefore provide IDL users with a simple, no-cost method for distributing IDL applications. An IDL application can easily be tuned into MLP concept, while IDL can be called from external C routines and/or it can itself call an external C function.

In the framework of the ScQL monitoring proper an interactive visualization can be available to any ScQL monitoring proper application, which calculates the statistics and parameters listed above, just by creating a Visualizer object and assigning a science data set to it. The visualizer object creates a complex widget-based GUI for visualizing the data set of the science data and its statistics, and then manages user interactions transparently to the application. Data exchange between the form of the science data and the application is based on the DataForm class and is simplified by mapping the form field to an IDL structure. A DataForm object can be created for any application and filled with entry fields, tables and various standard controls (buttons, droplists, etc.). The mixed-language programming approach makes it possible to incorporate an external application in the DataForm object and use it in (almost) the same way for which the IDL application is supposed. The Visualizer encapsulates the visualization functionality that covers the ScQL monitoring proper requirements for the visualization, namely:

- Provides visualization of the Image Parameter Set (IPS) data and its statistics;
- Allows the user to control visualizations and interactively generate plots and histograms for any column of the IPS datatable;
- Provides an interface to any external application (that calculates the statistics and residuals) and then visualizes its results.

-
- Controls statistics and plotting.
 - Zoom in and out of plot regions using the mouse pointer, track pointer positions in data space.
 - Allows the user to provide a hard copy of selected plots and histograms.

All functions of the ScQL Visualizer object may be controlled both by the user and the application. The application can interact with the Visualizer object to set up specific visualizations and receive user feedback. In fig. 5.4 an example of the ScQL monitoring visualization and manipulation tool is presented. In two panels the rate of the window for the whole SM CCD mosaic and SM CCD chip number 7 as well as the centroid positions in the downloaded windows are shown.

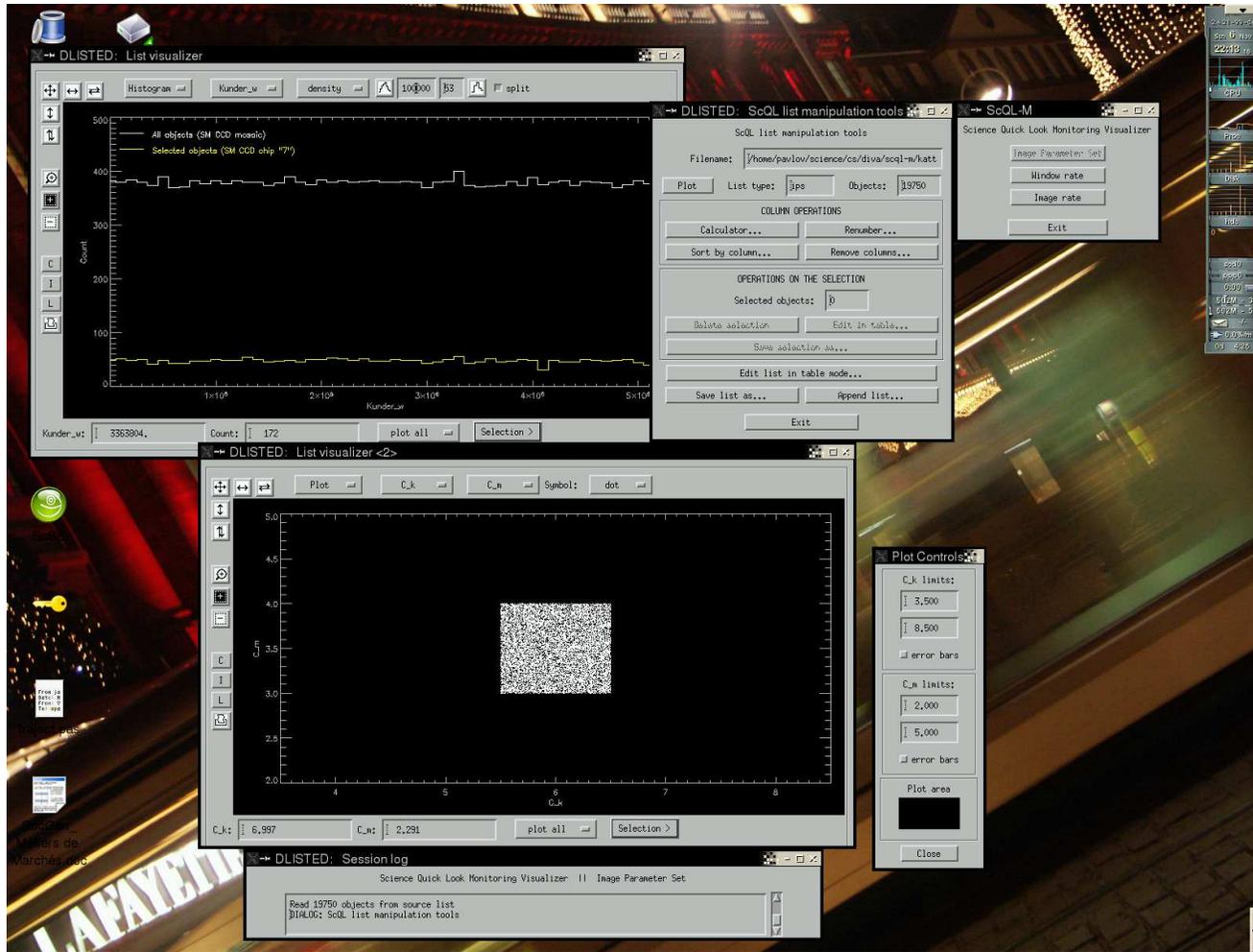


Figure 5.4: The example of the ScQL monitoring Visualizer and DataForms objects. In the top panel the rate of the window for the whole SM CCD mosaic (upper histogram, white solid line) and for SM CCD chip number 7 are presented (lower histogram, yellow solid line); the time is given in TDI clock stroke ($\delta t = 10^5$ TDI). In the bottom panel the centroid positions (C_k , C_m) in downloaded windows in scan- and cross-direction reckoned from the left corner of the window (see Table 8.5 in Appendix 8.2)

CLIPS

CLIPS (C Language Integrated Production System) is used for the ScQL diagnosis task as the product development and delivery expert system shell because it provides the capabilities to develop an effective knowledgeable inference engine [31]. CLIPS is a cohesive tool for handling a wide variety of knowledge with support for rule-based and object-oriented (COOL)⁶ paradigms. We use object-oriented programming of CLIPS that allows us to represent the elements of the diagnosis as objects and modular components in the residual evaluation phase of the ScQL monitoring system. Furthermore, as CLIPS can be embedded within a procedural code, called like a subroutine, and integrated with C and FORTRAN languages, it blends well with our mixed-language programming concept.

⁶COOL [32] – CLIPS Object-Oriented Language provided the fundamental features of any object-oriented language like abstraction, inheritance, and dynamic binding.

Chapter 5. Monitoring proper and Diagnosis

Chapter 6

Science Quick Look prototype evaluation

But in our enthusiasm, we could not resist a radical overhaul of the system, in which all of its major weaknesses have been exposed, analyzed, and replaced with new weaknesses.

– Bruce Leverett,

”Register Allocation in Optimizing Compilers”

After building the prototype of the monitoring system, we evaluate it and show the system’s capability. The objective of this chapter is to determine whether the developed prototype of the ScQL system meets its intended original requirements and goals to guarantee satisfactory performance to the users.

6.1 Basic cycle

Constructing a complex monitoring system such as Science Quick Look, the evaluation of the developed knowledge-based system is a very important stage. Before a knowledge-based system can be deployed, it must be evaluated for accuracy. It is also necessary to set up criteria for the checking of the observation parameters and variables in order to perform verification checks. For this purposes we propose the systematic evaluation of the ScQL monitoring that is based on the cycle: scenario – simulation – monitoring – diagnosis – matching.

The cycle has five main steps, they are shown in figure 6.1:

1. Observations (science data) are simulated from the scenario(s) via the star transit simulator. The faults are presented in the form of specific failure modes (such as an abrupt damage of the SM2 CCD mosaic, prediction mode) and are ordered by time. The scenario is therefore a set of the states of the instrument in the time domain.

Indeed, the simulated science data from the star transit simulator in the case of the simplest Galaxy model can be analyzed with respect to the analytically calculated properties of the model. If the calculated characteristics from the nominal simulated data for the attitude star transits, by using the ScQL monitoring proper, reconciles with the analytical ones, then the developed simulator is working correctly. If the analytical solution does not reconcile the simulated science data, the star transit simulator is not verified and we know that something is wrong in our model.

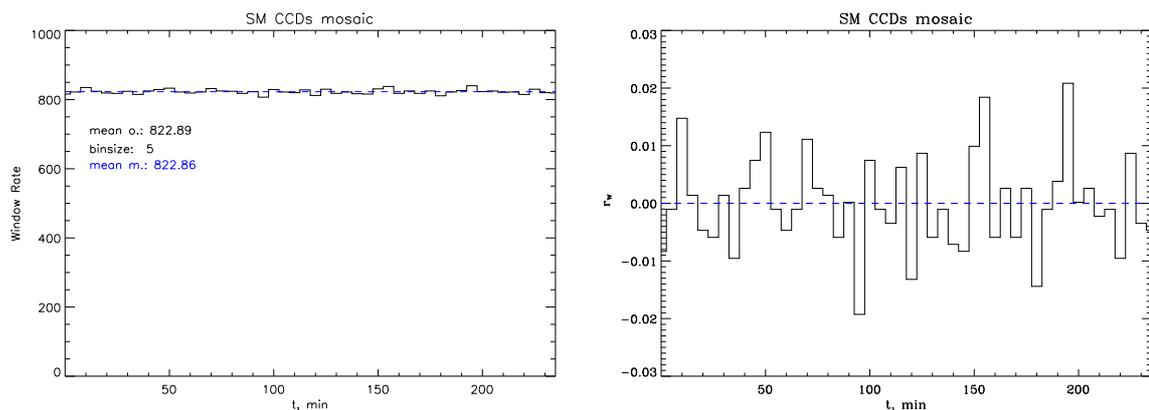


Figure 6.2: Window rate and residuals (in relative units) vs. time on the SM CCD array (CCDs mosaic) for a binsize 5 minutes (solid line histogram). Blue dotted line denotes the analytically mean value. The calculated mean window rate from the simulated observation science data flow (mean $o.$ = 822.89) coincides with the value derived from the analytical model (mean $m.$ = 822.86).

In figures 6.2 – 6.4 we present the number of downloaded windows of the attitude stars N_w and its residuals r_w . The data stream is generated with the nominal parameters of the model of the satellite. The histogram, presented in fig. 6.2, shows the window rate for the whole mosaic for two complete rotations of the DIVA satellite (4 hours) calculated with the binsize δt of 5 minutes. The calculated mean value $o.$ = 822.89 of downloaded windows from this simulated data stream coincides with that value $m.$ = 822.86 that is derived from the analytical calculation. It demonstrates that the numerical model of the star transits works properly, delivering the corrected data flow of the science data (in our study these are data from SM1 and SM2 mosaics). This statistics is an aggregation of the statistics from 8 individual CCD chips (see fig. 6.3). If there are wrong number of the downloaded cut-out windows of the chip n inside a special interval over $m \in [0, 255]$ an ADU chain failure will be responsible for this fault (each CCD chip consists of 2 ADU chains). In the case of the nominal simulation, the statistics on the level of the ADU chain will give the window rate such as on the figure. 6.4, where the downloaded windows from the upper ADU chain of the chip 1 is presented.

In the diagnosis, the statistics for any ADU chain will be taken as an elementary

Chapter 6. Science Quick Look prototype evaluation

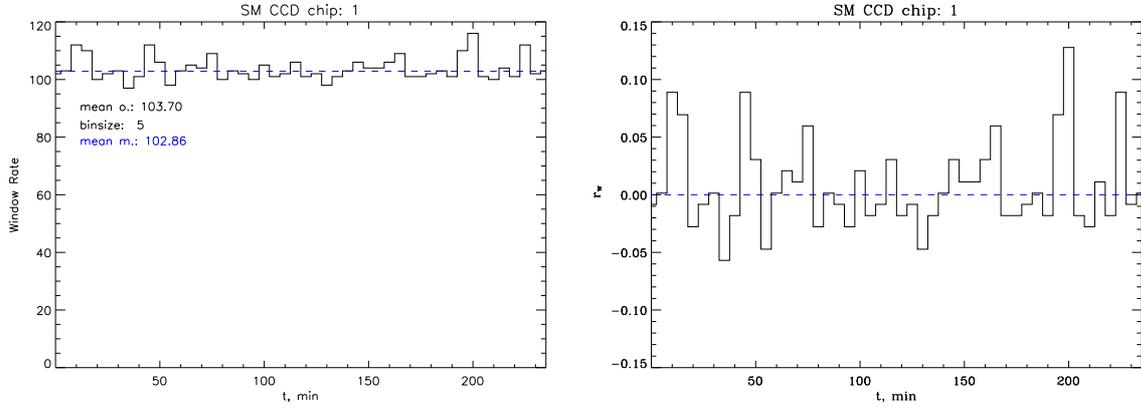


Figure 6.3: Window rate and residuals (in relative units) vs. time on the SM CCD chip for a binsize 5 minutes (solid line histogram). Blue dotted line denotes the analytically mean value. The calculated mean window rate from the simulated observation science data flow (mean $o.$ = 103.70) coincides with the value derived from the analytical model (mean $m.$ = 102.86).

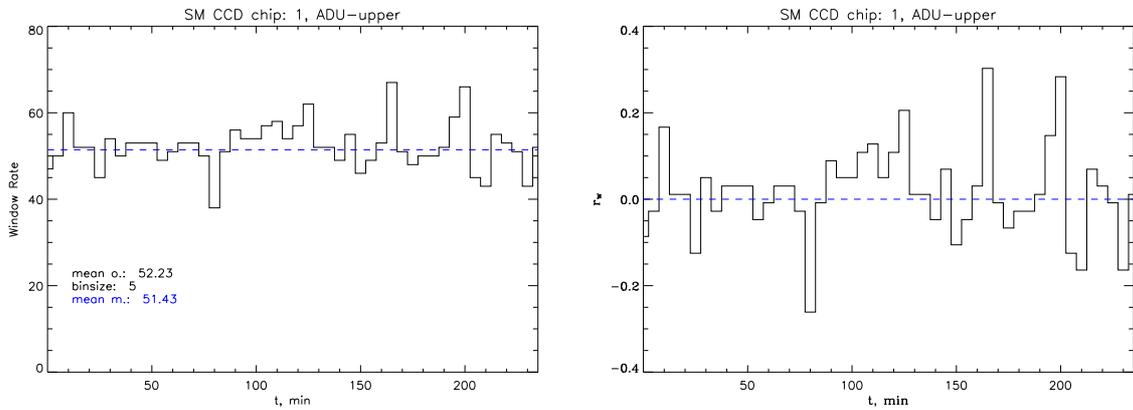


Figure 6.4: Window rate and residuals (in relative units) vs. time on the SM CCD ADU chain for a binsize $\delta t = 5$ minutes (solid line histogram). The blue dotted line denotes the analytically determined mean value. The calculated mean window rate from the simulated observation science data flow (mean $o.$ = 52.23) coincides well with the value derived from the analytical model (mean $m.$ = 51.43).

quantity for all other calculated aggregation states while each ADU chain of the CCD mosaic has its own power supply and can be thus considered as the smallest "active" element of the CCD mosaic. Nevertheless we compute analogous quantities over subsets of the CCD mosaic measurements with the pixel size array smaller than one ADU chain (see in previous chapter section 5.2.2 item 4). Together with analyse of the full mosaic image, that in the case of DIVA must be transmitted each 30 min, this statistics helps to point out where the reason for a deviation from expected values be located on the CCD, i.e. to find out bad pixel regions.

6.2.2 Monitoring of TDI clock stroke rate adjustment

As described in the previous chapters, a maladjusted shifting of charges in the CCDs, transporting the charge at a different speed across the CCD smears out the star images. If the shifting of the charge on the CCDs does not match the actual rotation of the satellite but this mismatch is not yet large enough to initiate the jet-firing then the TDI clock stroke rate should be tuned so that it is synchronized with scan-speed of the satellite. In the star transit simulator we included the emulation of the work of the

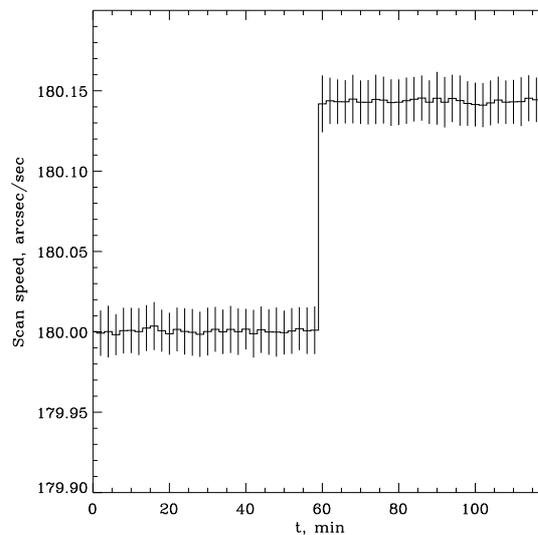


Figure 6.5: The calculated (known) scan speed of the satellite with its uncertainties. In this example an abrupt increase of the true scan speed is simulated on 30 May 2005 at 01:00:00 UTC. The picture shows the determination of this effect by the ScQL software. Error bars refer to mean errors (binsize 2 min.).

on-board software that monitors and adjusts (if necessary) the TDI clock stroke rate, using the transits of attitude stars (see section 4.8.7 of Chapter 4). In its turn the ScQL monitoring on the ground verifies the correctness of these actions, using science and

Chapter 6. Science Quick Look prototype evaluation

house keeping data. On the figure 6.5 the calculated speed of the satellite in the scan

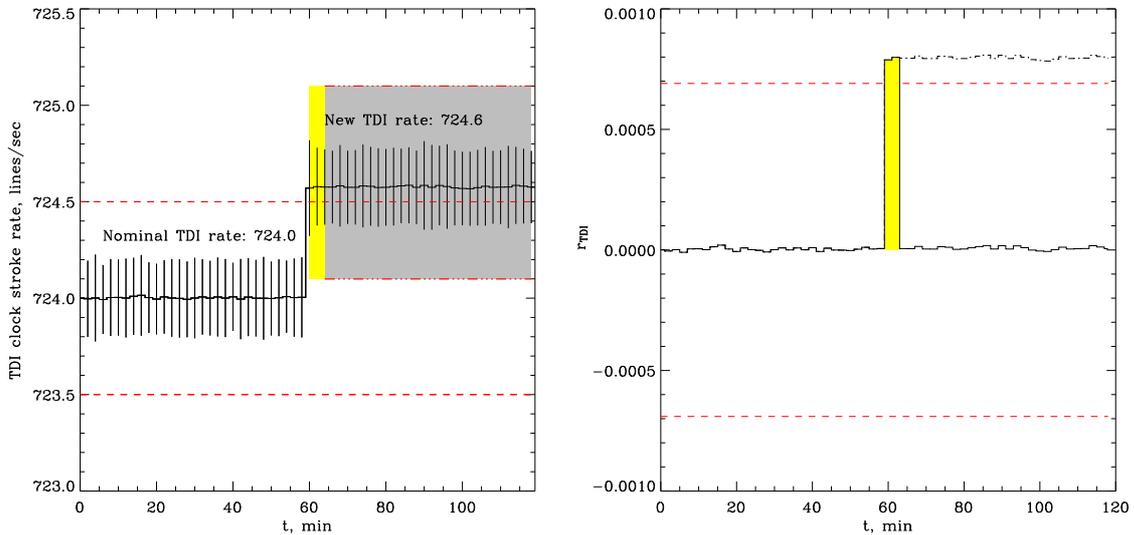


Figure 6.6: Calculated (known) TDI clock stroke rate (left panel) and its residuals (right panel) vs time (binsize 2 min). Left panel: known TDI clock stroke rate with its uncertainties (bars). Yellow area denotes a small time interval (4 min) when the blurred images are delivered. Right panel: solid line histogram is the residuals of TDI clock stroke rate, that confirms the correctness of the work of the on-board software; dot-dashed histogram is the residuals of the TDI clock stroke rate in the case of a permanent maladjusted shifting of CCD charge.

direction is shown. To check the rotation of the satellite we take a time interval δt of 2 minutes in our simulator. This is enough from the statistical point of view because the number of star transits for one million, homogeneously distributed over the sky (attitude stars) is about 330 (from both FOVs). The ScQL monitoring uses the same time interval to prove the calculations of on-board software¹.

In our simulation scenario we introduced an abrupt increase of the scan speed after 60 minutes. As one can see from the histograms on figure 6.6 the on-board software reacted correctly to this event. Firstly, it determines that the true TDI clock stroke rate is not synchronized with the scan speed. On the left panel the calculated (known) TDI clock stroke rate exceeds the threshold (dotted red line). At the next 2 minutes time interval it proves whether the new scan speed is stabilized². If yes (our case), the knew adjusted TDI clock stroke rate is set up as a true one. Thus, due to the changes of the scan-rate the instrument delivers the blurred images during 4 minutes (yellow area) and then it is returned to the nominal status with a new true TDI clock stroke.

¹In the ScQL monitoring the different time interval can be used to calculate the speed of the satellite if necessary.

²It might be reasonable to check whether the speed is stabilized or not during several time intervals

The residuals (solid line on the right panel) of TDI clock stroke rate, calculated in the framework of the ScQL monitoring system, confirms the correctness of the work of the on-board software. Otherwise, if the on-board software doesn't adjust the TDI clock stroke the residuals would be permanent demonstrating of exceedance of the threshold (dot-dashed line) and therefore alarm would be arised by the ScQL monitoring system.

6.3 All star transits

6.3.1 The nominal simulation in the framework of the multi-component Galaxy model. Time-varying thresholds.

The next stage for the evaluation of our monitoring system is to feed it by generating the science data of the observations using a multi-component Galaxy model (*the simulated sky*³). In the framework of this model our star transit generator can provide star counts

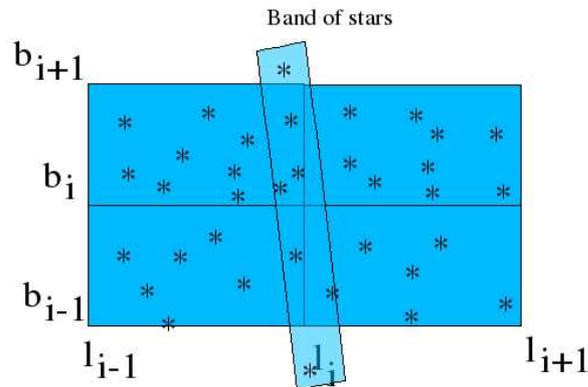


Figure 6.7: An illustration of the simulation of the band sky for one run (sample). This picture shows four sky subareas (the total number of which is 252) and the band of stars for one sample of the sky map. The sky map is simulated according to the multi-component Galaxy model in a given sky band defined by the nominal scanning law. A total number of 50 runs are carried out in order to calculate the expected mean values of the star transit along the band of stars. An average total number of stars in one band is about $1.7 \cdot 10^5$ which corresponds to about $7 \cdot 10^5$ star transits in one complete rotation of the satellite. To simulate 2 hours of observations takes 16 minutes of CPU time on Dell Inspiron 5150, powered by the mobile Intel Pentium 4 processor at 3.06GHz with hyper-threading technology.

³The simulated sky is used in the star transit simulator instead of *the known sky*.

Chapter 6. Science Quick Look prototype evaluation

down to the faint limiting magnitude of the instrument. According to the procedure of the multi-component sky map simulation, described in Chapter 4 section 4.3, the number of star transits is derived using different stellar density which was computed per area unit at the sky for different magnitude intervals. In our practical implementation we present the sky divided into $N_{\text{subareas}} = 252$ subareas. In order to calculate and to evaluate residuals in this case one needs first of all to calculate the expected mean values of the rate of the star transit along the given sky band (the sky band is determined according to nominal scanning law).

To do this we have to run the star transit simulator for a number of samples. The samples are produced by populating scanned subareas (see fig. 6.7) (which are observed by the satellite at a given period of time) according to the appropriate functions modelling the spatial and brightness distributions of the stars on the sky.

As an example we show the window rate of the full SM CCD mosaic calculated from the simulated science data. We take the same stretch of time as we used to demonstrate the crossing of the attitude stars in previous section. But we restrict ourselves to display a total elapsed time of two hours to keep things legible. Figure 6.8 shows how many windows arrive at each two-minute interval. The typical 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. We have four peaks during a full rotation of the satellite in two hours because the main instrument simultaneously observes two fields-of-view⁴ separated by 97.1 degrees. Each dot on the upper panel of figure 6.8 stands for one star transit run (the solid line represents the mean value). A total number of 50 runs with different samples of the star band were carried out in this case. In the lower panel of the same figure the expected mean value of the simulated window rate and its standard deviation are shown. These standard deviations set up the time-thresholds for the following evaluation of the residuals in the case of our multi-component Galaxy model. (Two hours of simulation takes 16 minutes of CPU time on Dell Inspiron 5150, powered by the mobile Intel Pentium 4 processor at 3.06GHz with hyper-threading technology.)

After this, we simulate observations in the fault-free mode of the instrument (see figure 6.9 upper panel). The lower panel of figure 6.9 represents the residuals; the mean window rate is shown as a reference (dotted line at $r_w=0$). We displayed a $1-\sigma$ and a $3-\sigma$ interval (green- and red-dashed line respectively) around the mean value that represents the time-varying alarm thresholds. The solid line is the residual computed by comparing the simulated fault free simulated observation with the expected (predicted mean) value.

⁴The overview of the instrument can be found in Chapter 1 section 1.4.

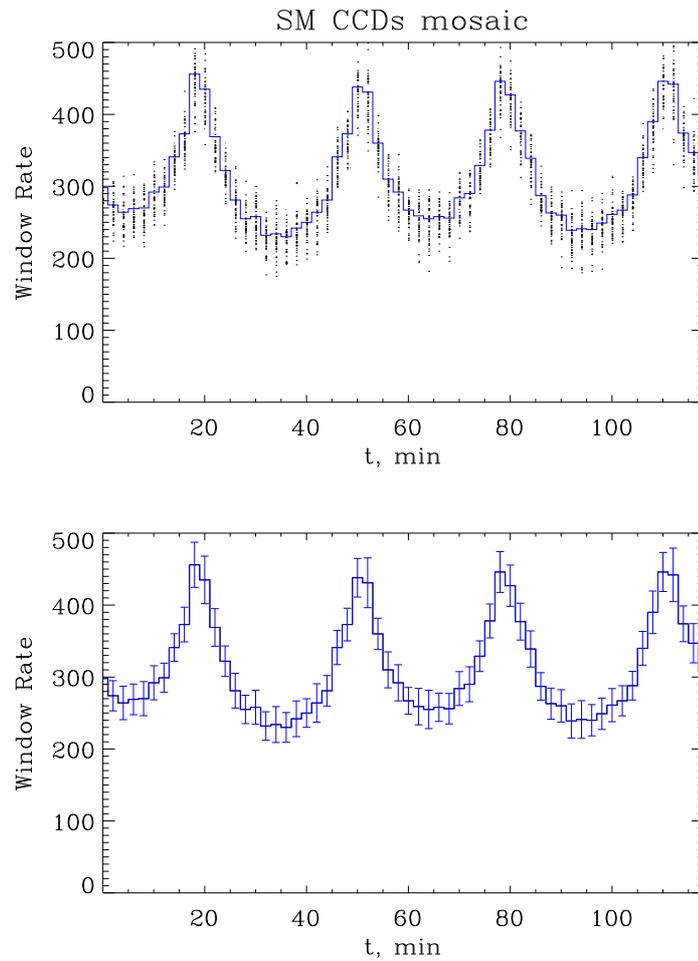


Figure 6.8: Window rate vs. time on the SM CCD array (binsize 2 min). Upper panel: (1) the black dots – nominal window rates based on 50 different samples of the band sky simulation according to the multi-components Galaxy model (see sec. 4.3 of Chapter 4) and (2) the blue solid-line histogram – the calculated predictive (mean) nominal window rate. Lower panel: the predictive window rate and its standard deviation (error-bar).

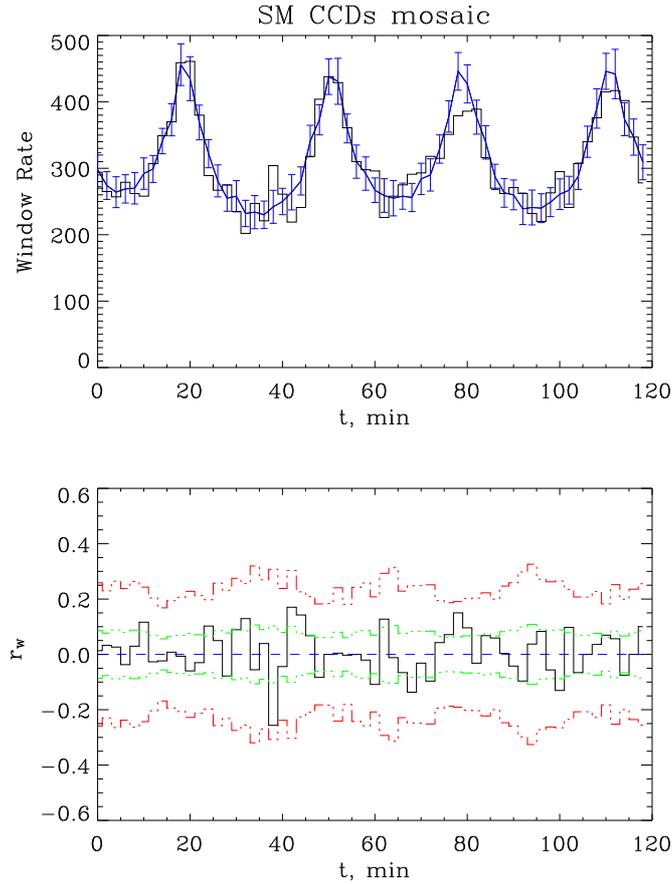


Figure 6.9: **Nominal.** Window rate vs. time on SM CCD array (binsize 2 min). Upper panel: (1) the blue solid-line with error-bars – the predictive window rate and (2) the black solid-line histogram – the simulated nominal window rate. Lower panel: (1) the black solid-line histogram – the residuals r_w^{CCDs} (see f. (5.1) in sec. 5.2.2 of Chapter 5), (2) the green dashed-line histogram and (3) red dashed-line histogram are the time-varying thresholds on the level of 1σ and 3σ correspondingly. All fluctuations of the nominal window rate residuals are in the framework of 3σ .

6.3.2 Monitoring of the window rate: an abrupt, an incipient and an intermittent fault.

We now introduce some simple faults into the simulated science data (an abrupt, an incipient, and an intermittent fault) and watch the reaction of the monitoring system calibrated with the measurements of the nominal star transits.

After approximately 60 minutes of operation, we simulate an abrupt fault (see figure 6.10) decreasing the window rate (introducing the malfunctioning of a part of the CCD array). Figure 6.11 (for intermittent faults between 20 to 50 and 90 to 110 minutes) and figure 6.12 (for an incipient fault after 50 minutes) show a similar picture.

The monitoring system reacted to these events, when the residual signals went over the threshold of 3σ . As a consequence, alarms were raised. For the abrupt failure the four CCD chips (SM1: $n=1$ and $n=2$; SM2: $n=5$ and $n=6$) were introduced to be damaged and the alarm status is defined after 60 minutes. For the intermittent faults (fig. 6.11) the system goes into alarm status for the duration of the faults and reverts back to normal after they are over. In this case the number of the damaged CCD chips is four (SM1: $n=1$ and $n=2$; SM2: $n=5$ and $n=6$) for the time interval between 20 to 50 minutes and three (SM1: $n=1$ and $n=2$; SM2: $n=5$) for the time interval between 90 to 110 minutes. We should stress here that the number of the simultaneously damaged areas of the CCD array which were introduced in the considered failure scenarios, is covered a large area so that the faults could be detectable using only the statistics for the whole CCD mosaic. It was done for the demonstration purpose because (as it has been mentioned earlier) the statistics for the full CCD array is an aggregation of the CCD chips (or ADU chains) and, therefore, is not sensible to detect a fault for the single chip (or ADU chain). Indeed, based only on the statistical thresholds for the full CCD array in the case of the incipient fault (fig. 6.12) the alarm would arise only after 60 minutes whereas a damage of the first chip was introduced after 50 minutes in our simulator. Moreover, it then go back to normal status for a short time, to reach an alarm status only after 70 minutes (as shown in fig. 6.12).

The hierarchical structure of the SM CCD array for the ScQL monitoring

To react to this event promptly the ScQL monitoring calculates the residuals on the level of the individual CCD chips. In the figure 6.13 the residuals of the window rate for chips are plotted for the given case of the incipient fault. It can be seen that this fault is a composition of abrupt failures of four chips of the SM2 CCD. As the residual signal of the window rate from SM2 CCD chip 5 exceeded its threshold, the ScQL monitoring system switched to an alarm mode at the time 00:50:00 (the time of the first damage of the CCD mosaic was introduced in the fault scenario) and the system remained permanently in this alarm status. In the case of the failure on the level of ADU chain appropriate residuals will be generated and evaluated in the same way.

Certainly, this hierarchical logic for the fault detection is somewhat redundant be-

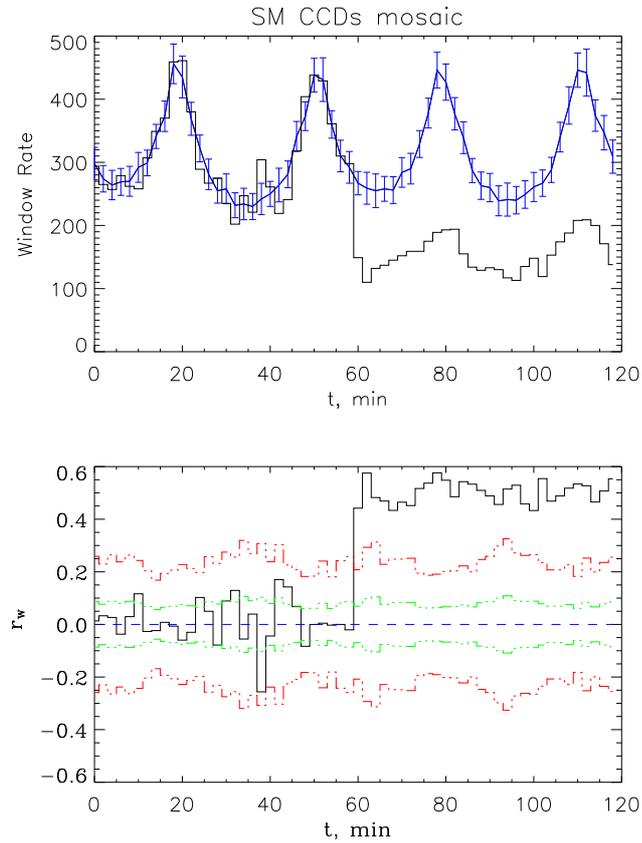


Figure 6.10: **Abrupt fault.** Window rate vs. time on SM CCD array (binsize 2 min). Upper panel: (1) the blue solid-line with error-bars – the predictive rate window, (2) the black solid histogram – the incorrect window rate. Lower panel: (1) the black solid histogram – the residuals r_w^{CCDs} (see f. (5.1) in sec. 5.2.2 of Chapter), (2) the green dashed-line histogram and (3) red dashed-line histogram are time-varying thresholds on the level of 1σ and 3σ correspondingly. The abrupt decrease of the the whole CCD mosaic window rate is simulated as a damage (or shut down) of the four CCD chips (SM1: n=1 and n=2; SM2: n=5 and n=6) on 30 May 2005 starting at 00:10:00 UTC; the considered fault-scenario covers time interval of 2 hours [00:00:00, 02:00:00] UTC.

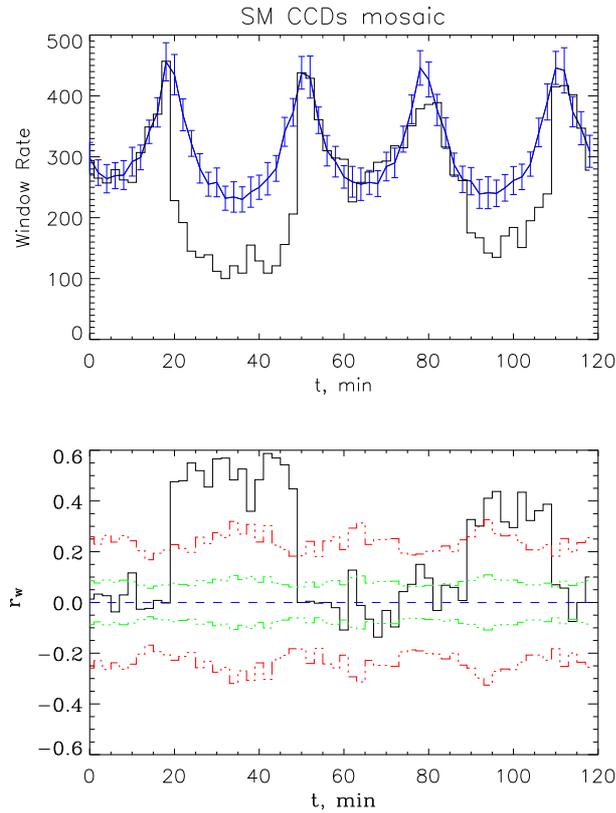


Figure 6.11: **Intermittent fault.** Window rate vs. time on SM CCD array (binsize 2 min). Upper panel: (1) the blue solid-line with error-bars – the predictive rate window, (2) the black solid histogram – the incorrect window rate. Lower panel: (1) the black solid histogram – the residuals r_w^{CCDs} (see f. (5.1) in sec. 5.2.2 of Chapter), (2) green dashed-line histogram and (3) red dashed-line histogram are time-varying thresholds on the level of 1σ and 3σ correspondingly. The intermittent fault of the window rate on whole CCD mosaic is simulated as a the temporarily shutdown of the four (SM1: $n=1$ and $n=2$; SM2: $n=5$ and $n=6$) and three (SM1: $n=1$ and $n=2$; SM2: $n=5$) CCD chips on 30 May 2005 at [00:20:00 - 00:50:00] and [01:30:00, 01:50:00] UTC correspondingly; the considered fault-scenario covers the time interval of 2 hours [00:00:00, 02:00:00] UTC.

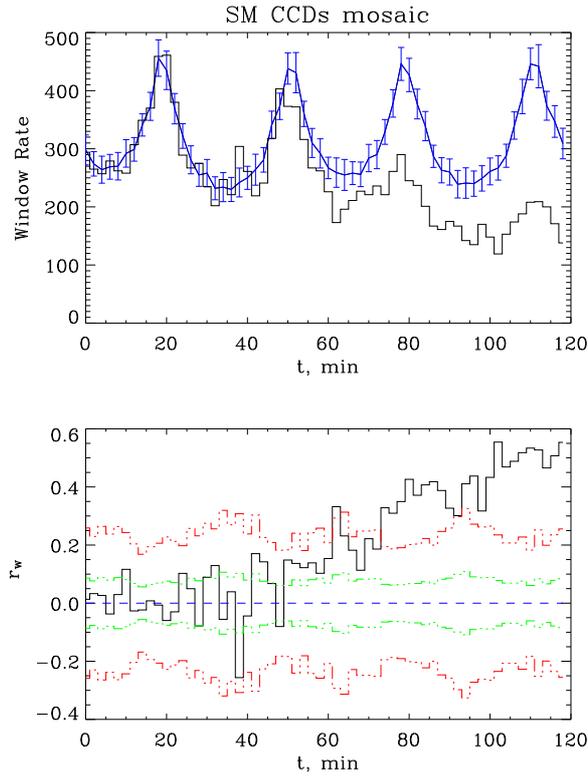


Figure 6.12: **Incipient fault.** Window rate vs. time on SM CCD mosaic (binsize 2 min). Upper panel: (1) the blue solid-line with error-bars – the predictive rate window, (2) the black solid histogram – the incorrect window rate with the degradation of the SM CCD mosaic. Lower panel: (1) the black solid histogram – the residuals r_w^{CCDs} (see f. (5.1) in sec. 5.2.2 of Chapter), (2) green dashed-line histogram and (3) red dashed-line histogram are time-varying thresholds on the level of 1σ and 3σ correspondingly. The decrease of the the whole CCD mosaic window rate is simulated as a consequent damage (or shut down) of the CCD chips on SM2 on 30 May 2005 starting at 00:50:00, 01:00:00, 01:20:00, 01:40:00 UTC; the considered fault-scenario covers the time interval of 2 hours [00:00:00, 02:00:00] UTC.

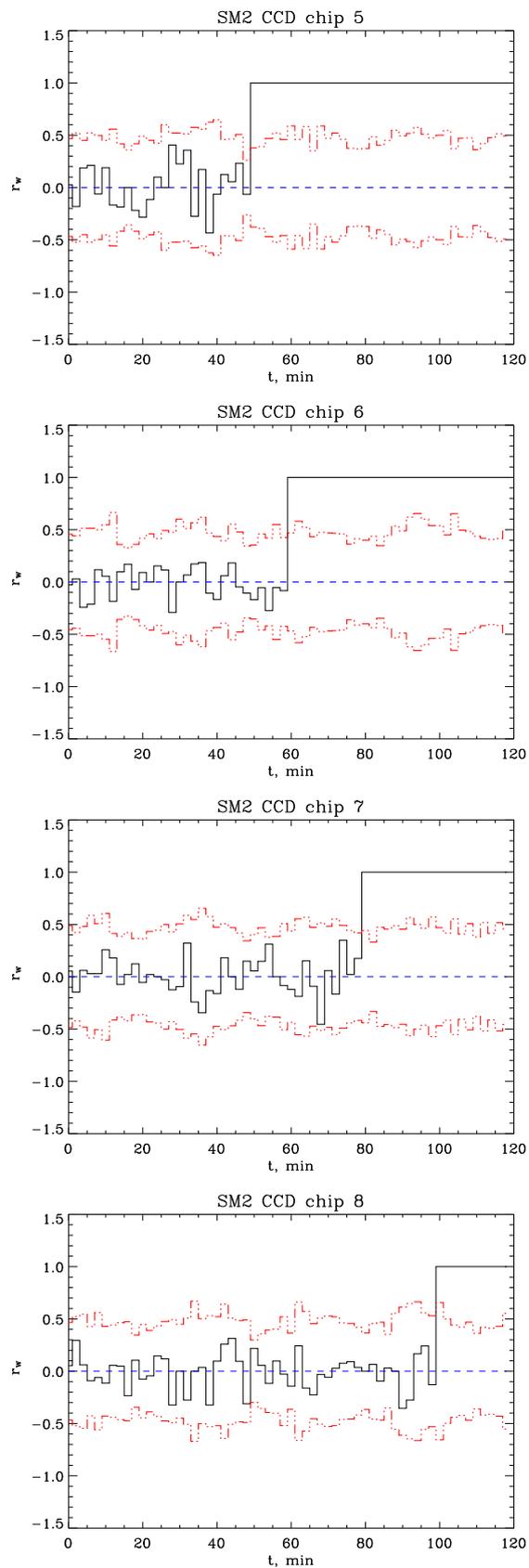


Figure 6.13: The residuals of the window rate on the chip level for the case of the consequent damage (or shut down) of the chips on the SM2 mosaic (see also figure 6.12).

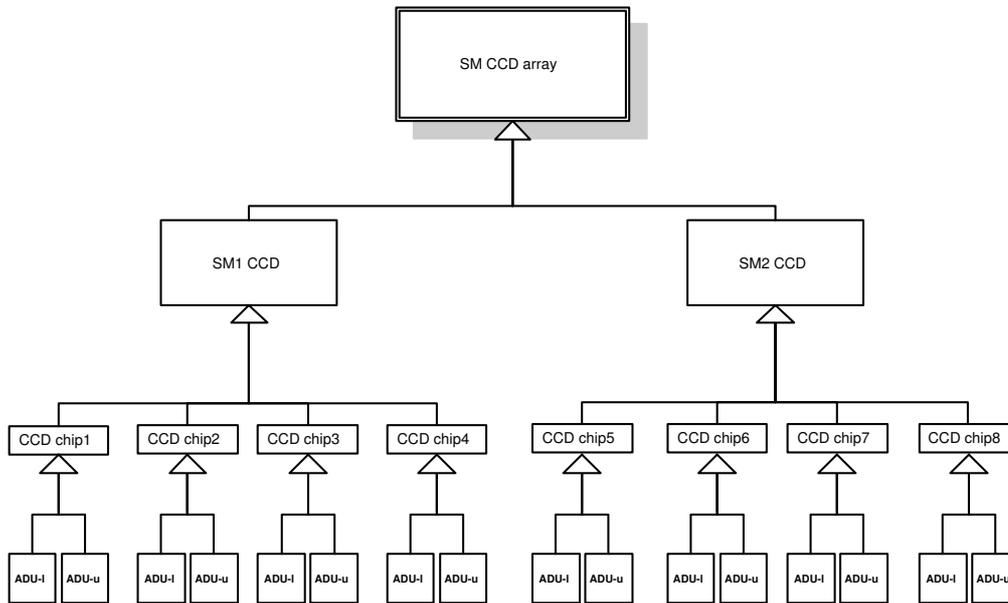


Figure 6.14: Hierarchical structure of the SM CCD array for monitoring of the science data.

cause the complete information for the residual evaluation can be obtained from the statistics generated on the level of ADU chain⁵. However, such hierarchical representation (see fig. 6.14) allows to obtain a more synthesized display of the data and to facilitate the operator-astronomer the understanding of the state of the instrument.

Monitoring of the science data from an individual FOV

To make our hierarchical approach complete we will also calculate separately the residuals for two field-of-views (preceding FOV and following FOV) so that each FOV can be supervised individually.

The given concept of the DIVA main instrument was designed in such a way that the star light from both FOVs, combined on the one focal plane, could not be automatically identified to which FOV it belongs. To do this the data treatment should be carried out in the framework of pipeline data reduction at the Data Center. But, one can calculate it easily for the attitude stars in the framework of the ScQL because their positions are known a priori. However, in the case of Gaia, the viewing directions of both telescopes overlap on a common focal plane with a shift of about 50 mm across-scan so that the star images from each field-of-view can be uniquely identified without any additional

⁵It should be worth to note that the statistics for the small pixel blocks provides us with useful information as for example CCD pixels degradation, but it will rather be used for a supplementary long-term analysis and should not be considered as the monitoring parameters that are crucial for the performance of the mission.

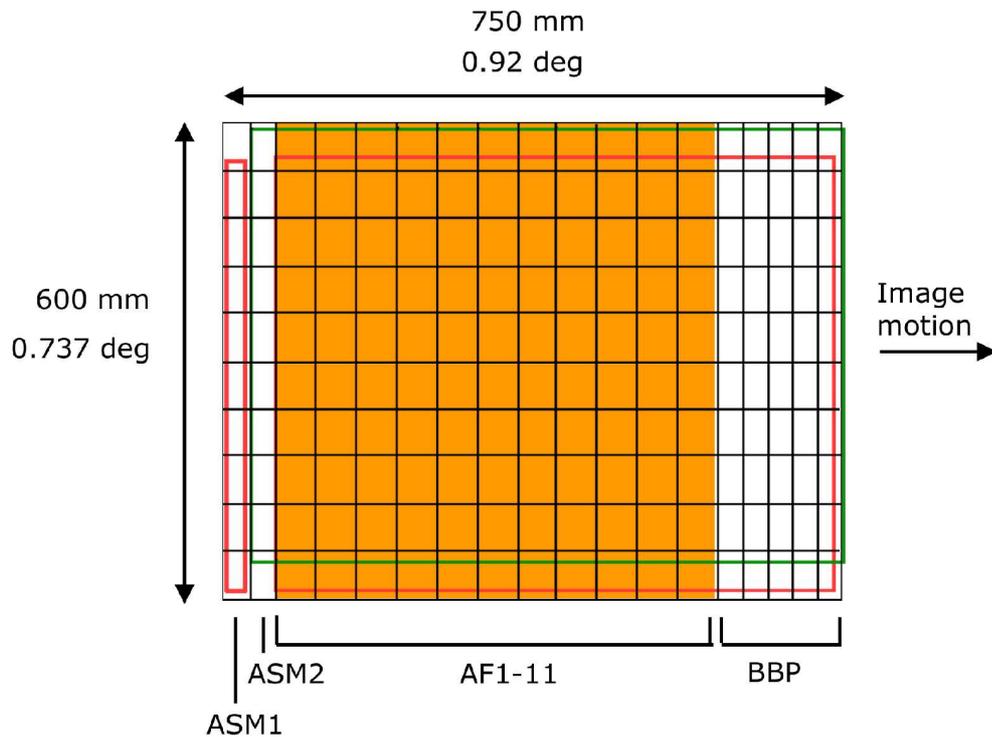


Figure 6.15: Overview of the Gaia Astro focal plane. The two Astro telescopes share the same focal plane: Astro field-1 is the area indicated in red, Astro field-2 is outlined in green. Three functions are assigned to the focal plane system: (i) the Astrometric Sky Mapper, which detects object entering the field of view, and communicates details of the star transit to the subsequent astrometric and broad-band photometric fields; (ii) Astrometric Field (AF), devoted to the astrometric measurements; (iii) the Broad Band Photometer (BBP) which provides multi-color broad-band photometric measurements for each object. The resulting focal plane design consists of a mosaic of 180 CCDs with pixel of $10\mu m$ along scan \times $30\mu m$ across scan size ($44.2 \text{ mas} \times 132.6 \text{ mas}$). From left to right: the first columns are for the sky mappers with the following orange area of the CCDs for astrometry (analogue of DIVA SM CCDs); the last five columns are for broad-band photometry.

object recognition procedure.

Like in the case of DIVA, stars crossing field are detected by the sky mapper (called in terms of Gaia astrometric sky mapper), and then observed in the predicted windowing mode across the main CCD arrays (astrometric field), which is related to DIVA SM2. The sketch of Gaia's focal plane is plotted on the figure 6.15 (see details in [43], [4]). Figure 6.16 illustrates the failure of preceding FOV that causes the homogeneous window rate decreasing in the time interval of 20 minutes (see upper panel). According to our hierarchical monitoring logic, the generated residuals for the preceding FOV (left panel) allows to detect and to identify this fault's occurrence.

6.3.3 Monitoring of the star brightness

Assuming now that we obtain a wrong (small) number of windows from CCD array that neither can be explained by complete chip (or ADU chain) damage nor by the problem with the field-of-views. We introduced the high background during 20 minutes in our star transit simulator for the whole CCD array. So, the derived decreasing of the window rate (or increasing of its residuals, figure 6.17) is also homogeneous in all chips of the CCD mosaic as it was in the case of the failure of the preceding FOV. The brightness of the stars can provide us with additional information to identify this failure. To monitor the brightness of the stars we calculate the cumulative brightness distribution function of observed stars and compare it with the expected one by means of Kolmogorov-Smirnov (K-S) statistic. The constructed cumulative distribution functions are plotted on the figure 6.18 (upper panel) for an interval of one complete rotation of the satellite. The K-S statistic D , and the significance level "prob" (see the formula (5.2) from Chapter 5) for the null hypothesis, that the data sets are drawn from the same distribution, are also presented on at: a small value of the prob shows that the cumulative function of the brightness distribution of the observed data (black line) is significantly different from that of the expected (nominal). On the lower panel of the same figure the absolute value of the difference between two cumulative distribution function helps to point out the interval of the magnitude where it rises above the calculated greatest distance D of the two different nominal data samples (red dotted line). In principle, the most probable scenario that causes a wrong brightness distribution is a high background and/or a bad signal-to-noise ratio of CCDs. It gives the lack of faint stars and affects the tail of the distribution. So, to identify finally the considered failure we calculate the number of the stars of a different brightness interval at the time when the decreasing of the window rate took place. The derived value of the faintest magnitude V_{lim} (see histogram on figure 6.19) of the observed stars at the this time period is too small and correlates with the fact of too small window rate.

Thus, after disproving of the null hypothesis for a standard pre-defined time interval by means of the K-S statistic, that demonstrates a good sensitivity in finding shifts of brightness distribution, the problematic time interval is checked.

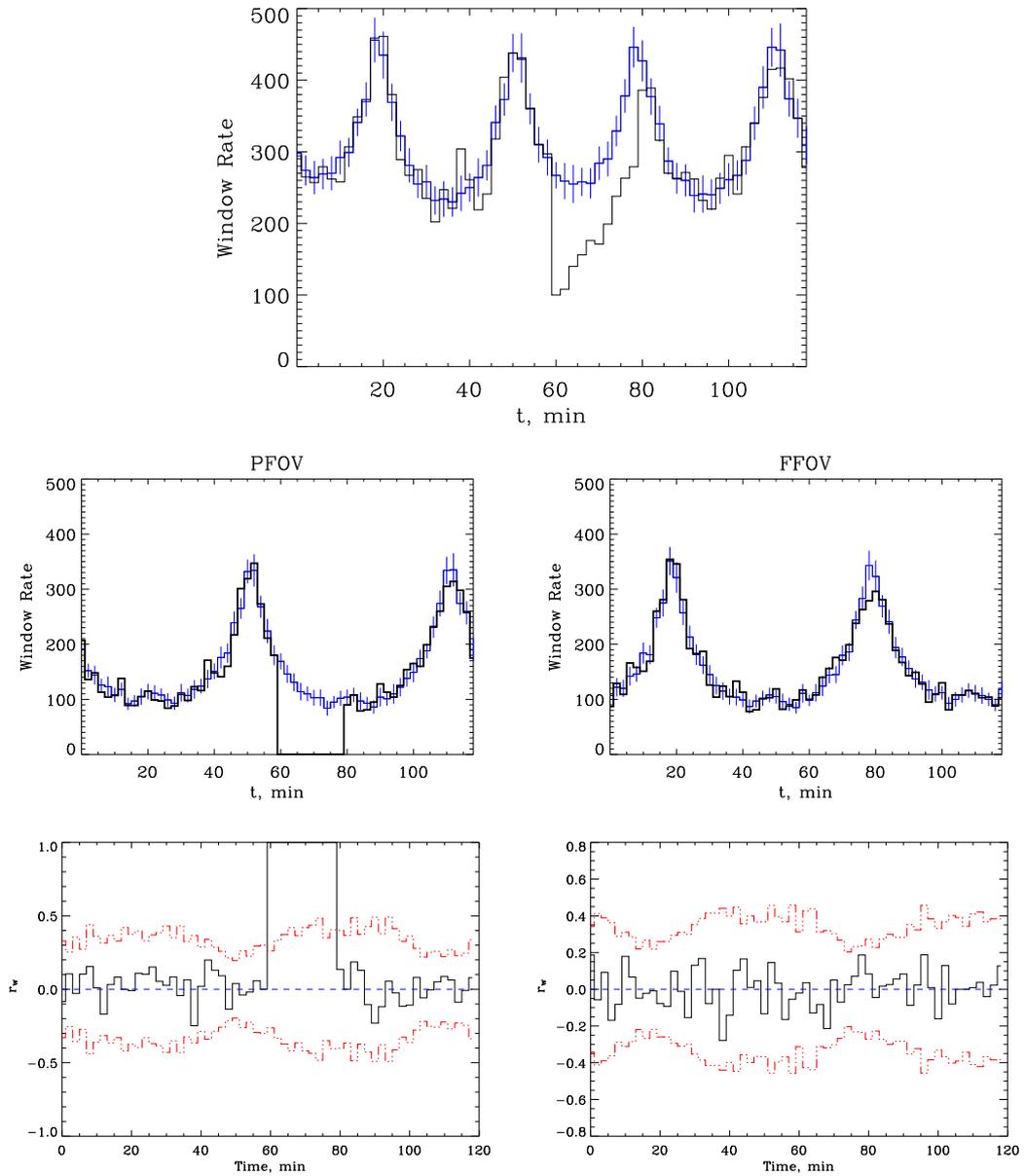


Figure 6.16: The decrease of the window rate on the whole CCD array (upper panel) is due to the temporal problem with the preceding FOV (PFOV) of the main instrument. The residuals of the window rate from the preceding FOV (left panel) together with the residuals of the window rate from the following FOV (FFOV) (that shows no anomaly (right panel)), identify this failure's occurrence. Note: in the case of DIVA this failure can be only identified for attitude stars.

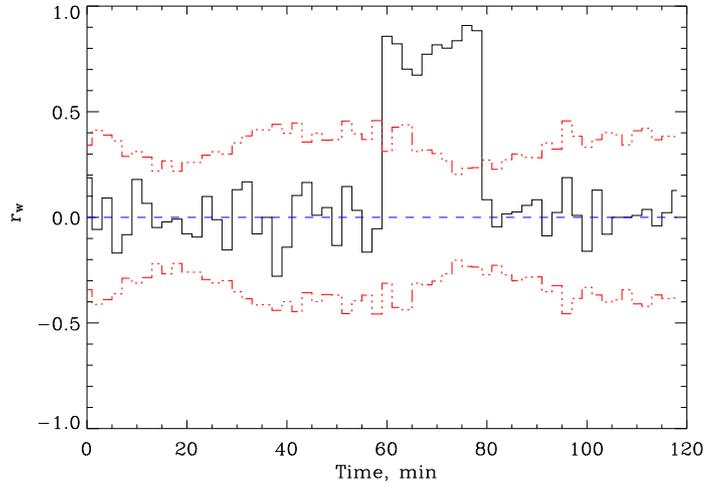


Figure 6.17: The residuals of the window rate of the CCD array. The decrease of the window rate on the whole CCD mosaic is due to the simulation of the high background in time interval [01:00:00, 01:20:00] UTC.

6.3.4 Monitoring of the star centroids

SM CCD rows

Let us examine now some image parameters from downloaded windows. But before, we would like to mention that additionally to the hierarchical representation of the CCD array for the monitoring of the science data we organized CCDs in rows (in pairs) in the context of the ScQL. The reason for this is that stars detected on the SM1 mosaic are predicted on SM2 mosaic, so windows are cut out from one CCD of SM1 are related to windows from other CCD mosaic in the same CCD row. In this sense CCD rows can be considered as a unit for important ScQL monitoring tasks. These related windows in the case of the SM CCDs mosaic create the following chip pairs (see also fig. 4.15 from the Chapter 4): (SM1 chip 1 – SM2 chip 5), (SM1 chip 2 – SM2 chip 6), (SM1 chip 3 – SM2 chip 7), (SM1 chip 4 - SM2 chip 8).

To make the most of the image data of a window it should contain as much signal of the observed star as possible. Thus, the star centroid should be located in the center of the cutted window. On the figure 6.20 the mean centroid of downloaded windows from the nominal two hours simulation are presented. The centroids match the window center with some positional error budget⁶. A bit shift that changes some data in the instrument calibration can provoke that the predicted pixel coordinates of the window to be extracted from SM2 CCD mosaic are changed and therefore the window cutting

⁶The error budget is taken about 1 pix.

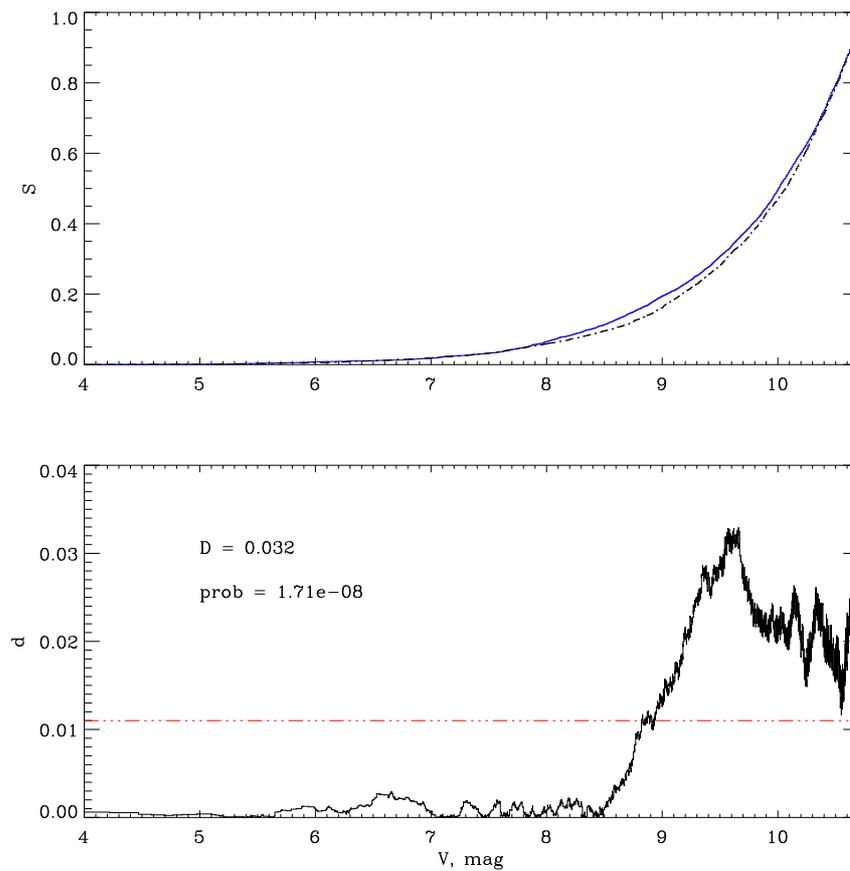


Figure 6.18: Unbiased estimators $S_N(V)$ of the cumulative function for the two brightness distribution are constructed. The statistic D is defined as the maximum value of the absolute difference between two distributions (see text).

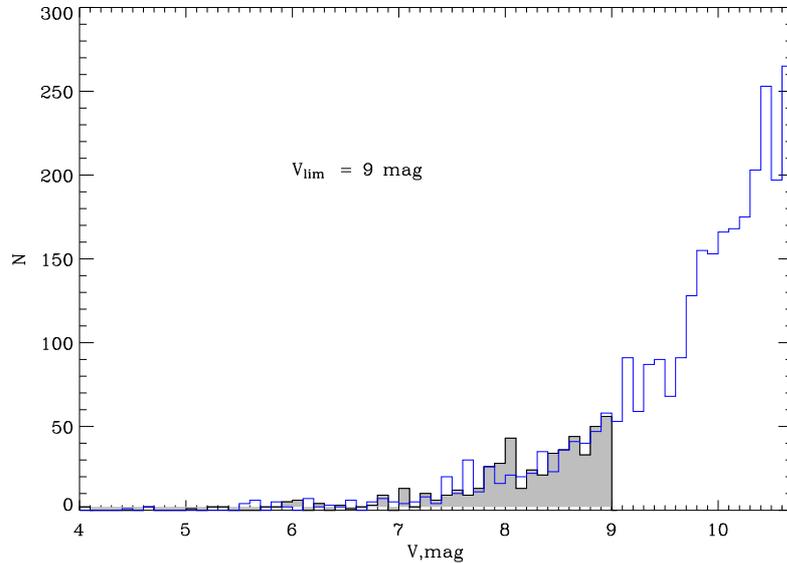


Figure 6.19: The histogram of the brightness distribution for the same time interval as in figure 6.17. The observed star histogram demonstrates an absence of faint stars starting from $V_{\text{lim}} = 9$ mag.

algorithm cuts out the wrong pixel stream area. In the extreme case, the shifts in the predicted \underline{k}_w completely displace the windows to be cut out so that the start image is no longer situated in it, i.e. the empty windows will be extracted.

Identification of dual failure from monitoring of star centroids

As a last failure scenario we will consider a case of dual failure in which two failures, affected the position of the star centroids, occur simultaneously.

Firstly, we suppose that the detection algorithm (or cut-out algorithm)⁷ works improperly and gives a false \underline{k}_w of the windows on the whole SM1 mosaic. If this slight shift is equal 2 pixels, then the predicted window pixel edge \underline{k}_w on the all chips of the SM2 mosaic will be systematically shifted and therefore the along-scan mean centroid C_K of SM2 mosaic should be at about 8 pixel (see figures 6.21 and 6.22). Based on this statistics the ScQL monitoring system generates appropriate residuals and switches to an alarm mode with a diagnostic solution that either the detection or cut-out algorithm for the detection mode on SM1 mosaic works improperly⁸. Furthermore, the system will detect the difference between expected values of the mean centroid position of the images for the pair of chips (1 – 5). The expected position of the along-scan mean

⁷For the moment, we are not able to identify which one of these two algorithm works improperly.

⁸This scenario is a

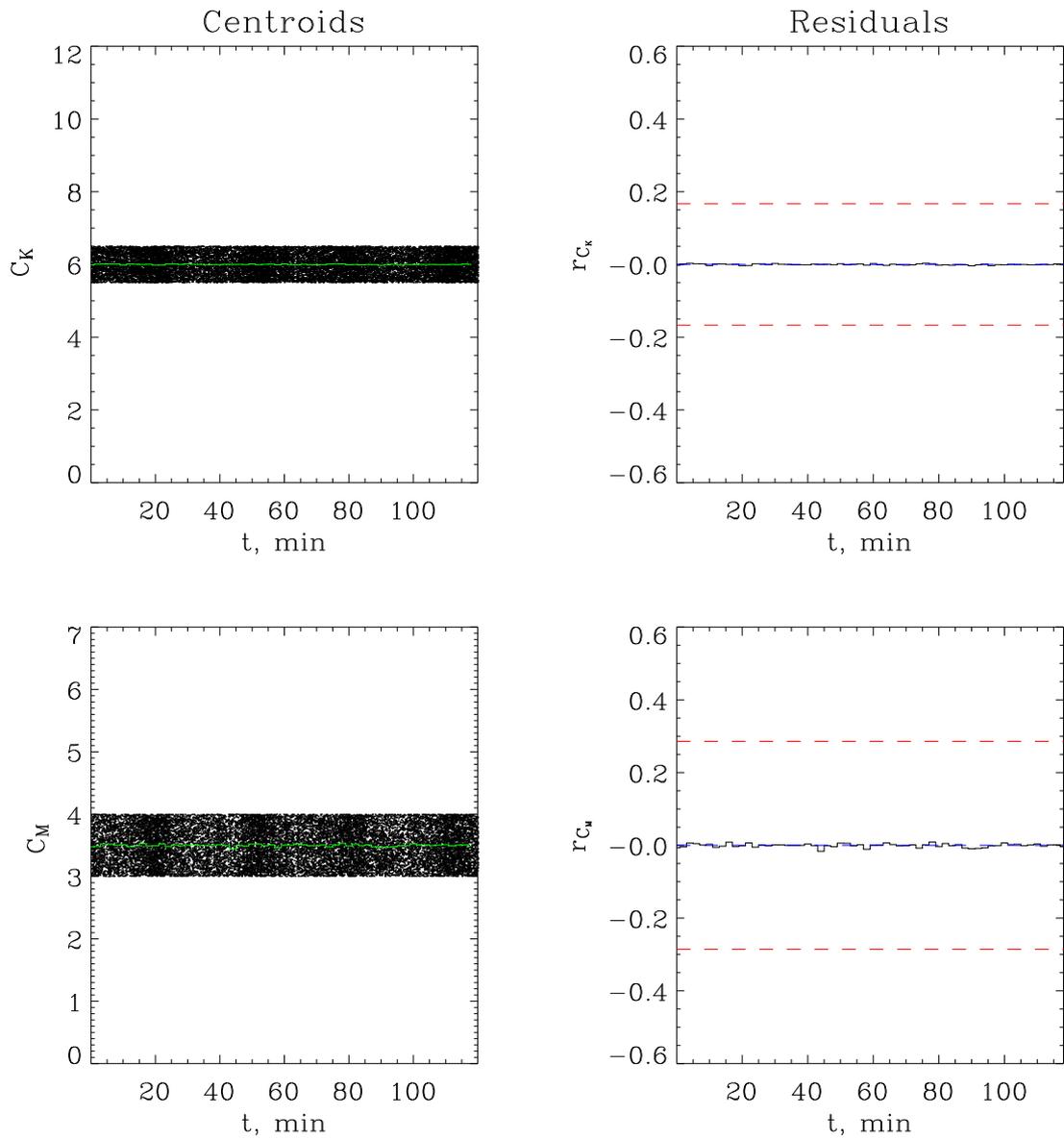


Figure 6.20: Along-scan C_k and across-scan C_m centroids vs. time on SM CCD mosaic for a nominal simulation (dots on the left panels); the histograms (green lines) are the average centroid position C_K and C_L (binsize 2 min). On the right panel are the appropriate residuals (black line histogram) and its thresholds (red dashed line).

Chapter 6. Science Quick Look prototype evaluation

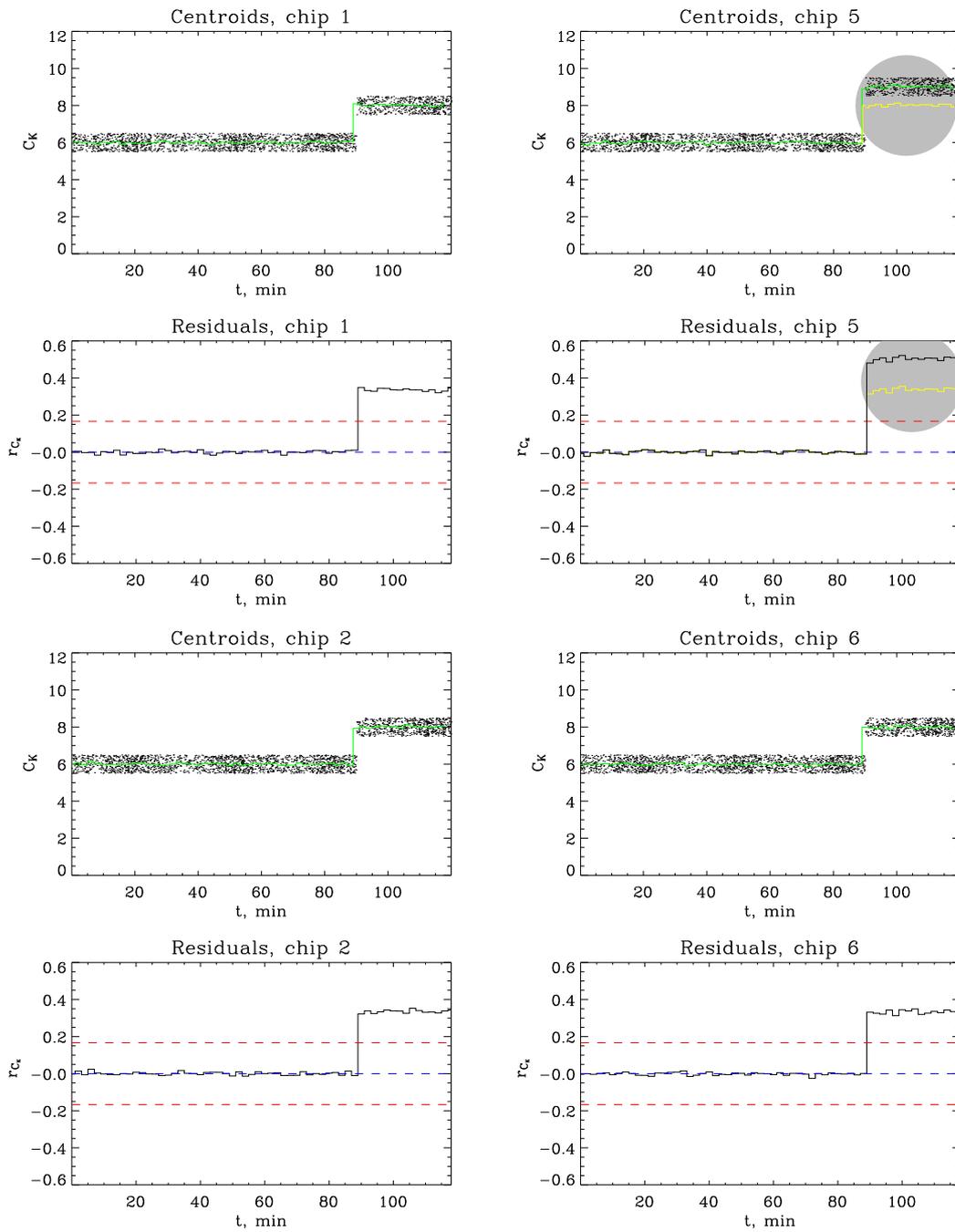


Figure 6.21: The average centroid positions C_K (green line histogram) and its residuals (black line histogram) for the following rows (pairs) of CCD chips: [(1–5), (2–6)] in the case of the dual failure scenario. Dots are along-scan centroids C_k . Yellow line histograms (chip 5) are expected centroid positions C_K (and its residuals) if only the failure of the detection algorithm took place.

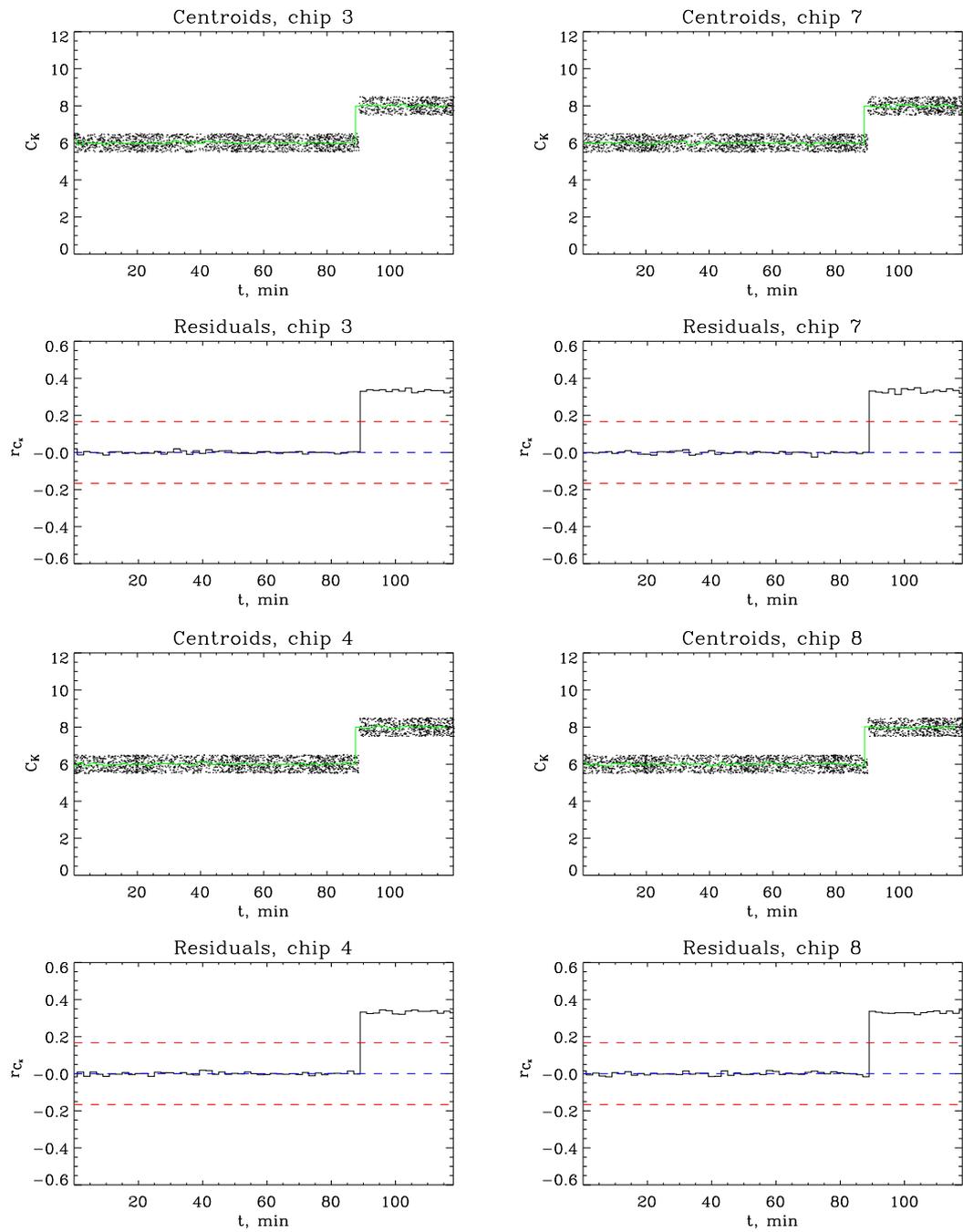


Figure 6.22: The same as on the figure 6.21 but for the other rows of CCD chips: [(3–7), (4–8)].

Chapter 6. Science Quick Look prototype evaluation

centroid of the chip 5 in the case of detected failure of the detection algorithm is shown by yellow line histogram on the figure 6.21. This indicates that the calibration data of the along-scan offsets between the consecutive CCDs is changed of about 1 pixel (see the geometric arrangement of SM CCDs on figure 4.15 from the Chapter 4).

Thus, the result of the failure identification from the diagnosis system will be twofolded:

- *The detection algorithm failed.* The on-board detection (or cut-out) algorithm has been working improperly in the 'detection mode' on the SM1 mosaic since 30-05-2005 00:01:30. Possible intervention is to reload the software. Alert the First Look.
- *The calibration data is damaged or changed.* The along-scan offsets between the chips (1 – 5) has been changed or the calibration data has been corrupted. Possible intervention: upload the calibration data; if the problem persists, the new calibration data should be produced in the framework of the First Look. Alert the First Look.

Note that the incorrect attitude will also produce badly predicted windows, smearing out the star image while the shifting of charges in the CCDs will no longer match the real scan speed. But the centroid positions would not have a systematic shift in this case.

6.4 Summary

In this chapter, we have presented the evaluation of the developed prototype of the ScQL monitoring system. The system was extensively tested, feeding simulated science data generated by the star transit simulator into it. The nominal simulations have been carried out to derive computational solutions of what should be considered as fault-free parameters. The calculated standard deviations set up our modeled time-varying thresholds and confidence level for the ScQL evaluation phase. These time-varying thresholds (for the case of the rate of the downloaded windows/images) follow Poisson's statistics because our knowledge about the sky (in the simulator) is restricted by average stellar coordinate and stellar magnitude distributions. This knowledge, however, will be substantially improved after half a year of observations when the first sphere solution is performed. This will allow us to refine our model and determine thresholds that are as close as possible to the nominal case (fig. 6.23).

According to the proposed evaluation basic cycle, we have simulated different failure situations in the framework of the star transit simulator to verify the reaction of the system. The simple hierarchical data representation seems to be effective and reliable for the purpose of fault identification and provide a synthesized display of the data that facilitate the understanding of the instrument status.

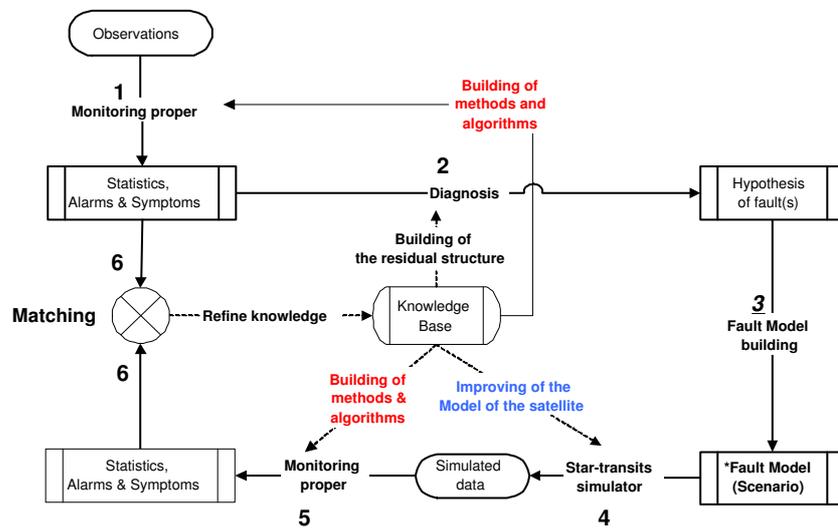


Figure 6.23: Learning diagnostic knowledge cycle.

Chapter 6. Science Quick Look prototype evaluation

Chapter 7

Conclusion and Outlook

This thesis considered different aspects of the development of the monitoring system for rapid assessment of the performance of scanning astrometric satellites. A scanning spacecraft needs a global, coherent, interleaved reduction and calibration of many months of measurements to reach their target levels of precision. It is therefore, on one hand, very difficult to immediately assess the proper functioning of all elements at the required levels of precision and stability. It is, on the other hand, for the same reason of utmost importance to quickly get a handle on the inherent quality of the elementary science measurements. So, monitoring of various kinds is required to continuously check the correct operation of a scanning satellite, both as individual components and as a complete system. To perform the preliminary assessment of the main instrument of the satellite in quasi real time constraints the concept of the Science Quick Look monitoring has been developed in the framework of this thesis. The aim of our study was mostly to explore basic monitoring properties and identify specific features for such a class of space missions but not to produce an operational ScQL monitoring system for a given scanning science astrometry satellite. We have shown that our model of fault detection (monitoring proper) is effective in the prototype of the system supervision of the science data of the DIVA astrometry satellite. It can be easily used in conjunction with fault-identification (diagnosis) for automatically identifying the nature of a failure after detecting its occurrence. The developed concept is quite general and is applicable to other related scanning space mission such as Gaia.

This chapter summarizes the work presented in this thesis. The main results and conclusions are reviewed, directions for further investigations are presented.

7.1 Conclusion

The following number of items can be drawn based on the accomplishments of this thesis.

- ★ In order to guarantee a high level of precision when collecting scientific data during

Chapter 7. Conclusion and Outlook

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 characteristic of a astrometry space mission. As a result of this study we employ a model-based approach, where we do not rely on empirical knowledge about symptoms and faults but on fundamental knowledge of the considered domain. We have explored the modern state of this approach and one thing became clear from the literature that there is no universal method to cope 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 as this is how he or she usually solves scientific problems. There are also shortcomings of the pure symptom-based approaches, the main one being that it is difficult to comprehend why a system arrives at a certain conclusion.

- ★ The design of the model-based ScQL monitoring was driven by careful analysis of the spacecraft domain, paying attention to the specific characteristics of an astrometry space mission. The distinguishing feature of the ScQL monitoring is to perform an indirect control of the work of the components of the main instrument satellite by collecting, analyzing and interpreting the astrometry observations. Thus, at the core of the ScQL model is star transit simulator that mimics the behavior of satellite and simulates its observations. It also guides the development of the two other parts of our system: monitoring proper and diagnosis.
 - 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 (in the form of residual structures) generated by monitoring are related to faults in the system by residual evaluation.

- ★ The complete model of the observations is an extremely challenging task. The first step towards this direction is the model implemented in our star transit simulator. It consists of the model of the Galaxy, the structure and behavior of the main instrument of the satellite and the model of the scanning strategy of the satellite. Our modeling approach achieves a broad coverage of monitoring of the on-board instrument hardware and software by coupling the purely quantitative model with the semi-quantitative approach. Indeed, keeping the description of the work of some components of the instrument at the high level of abstraction permitted to create an extended skeleton of the ScQL monitoring at this stage of the project (when there is a lack of the detailed information of the behavior of some components).

- ★ Maintaining the knowledge for such a complex system was also a challenging task, as small changes in the design may necessitate revisions in a large part of the knowledge base. In our model we introduced strong definitions and parameterizations of the entities related to the model of the observed environment (sky) and instrument. Following this formalism one can modify the ingredients of the model of the system to generate the more detailed star transit simulation and therefore expand the capability and accuracy of the ScQL monitoring itself.
- ★ In the early stages of the DIVA project, we were faced with the need to establish the strategy for the DIVA data reduction software (pipeline) based on a relational database. In order to combine the strength of various programming languages, minimize the efforts and to reuse the components we proposed to use the conception of mixed-language programming. In the ScQL monitoring we successfully applied the MLP approach that allowed us to build a flexible and compatible with the database software.
- ★ Upon analyzing the list of the main quantities of the statistics and parameters which are analyzed at ScQL monitoring, we determined the architecture of the monitoring proper mostly based on the IDL (Interactive Data Language). We adopted and modified the two main parts of Foundation Class Layer (FCL), Visualizer and the Dataform, providing the user interface part and visualizing the output of the star transit simulation and statistics-generating components of ScQL.
- ★ Our work focused on the monitoring part of the 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 (tool) for building expert systems). Applying this tool for the development of our diagnosis part of the system will allow us to perform easily structural testing of implemented rules that are found to be correct in syntax, unique in name and action, complete and consistent. But we do not consider at all the important aspect of the severity of the faults, this certainly should be the task of further studies.
- ★ The proposed design was demonstrated in the framework of the DIVA ScQL prototype with the scenario evaluation procedure. The evaluation was to determine if we succeeded in achieving our objectives to have a tool that assists the operator in quasi-real to deduce the changes of the state of the satellite from the science data, producing appropriate statistics and diagnosis.

The monitoring system was tested extensively by feeding simulated star transit data into it and by generating and evaluating statistics. The evaluating procedure, based on the "scenario – simulation – monitoring – diagnosis – matching" cycle, allows to watch and validate the reaction of the ScQL monitoring in terms of failure scenarios.

Chapter 7. Conclusion and Outlook

The nominal simulations have been carried out to derive computational solutions of what should be considered as a fault-free parameters. The calculated standard deviations set up our modeled time-varying thresholds and therefore the confidence level for the ScQL evaluation phase. These time-varying thresholds (for the case of the rate of the downloaded windows) follow the Poisson's statistics while our knowledge about the sky is restricted by average stellar coordinate and stellar magnitude distributions. This knowledge, however, will be substantially improved after half a year of observations when the first sphere solution is performed.

7.2 Outlook

The results of an evaluation of our system is very promising, so it will be worthwhile to pursue further studies in this area. First of all the diagnosis part of our system must be improved to bring it on par with the monitoring part. Second – as the DIVA project was stopped – we are able to adjust our approach to the next space-based astrometry mission, Gaia, which will be launched in 2012. Indeed, many aspects for the rapid assessment of payload and spacecraft health, developed in this work in the framework of DIVA project, are analogous to those in Gaia due to the fact that the main principle and basic geometry of the measurements are the same.

To build a complete ScQL diagnostic system with the sensitivity it will be extremely valuable to take into account the simulation at CCD pixel level (when the raw data with images are generated) as well as some other technical issues like charge transfer efficiency of the CCD, possible irregularities of the physical pixels, projection effects from the sky to the focal plane, etc.

A successful completion of the ScQL prototype for the DIVA mission provides the basis for our belief that a ScQL monitoring system for the larger project – Gaia – is achievable in terms of the developed concept. Building a ScQL monitoring system for Gaia therefore would become a lot easier if the important steps have already been done in the DIVA project. It is evident, however, that this work has to evolve and grow, along with the concept of the Gaia satellite.

Chapter 8

Appendix

8.1 The contribution of DIVA to astrophysics

The unique combination of homogeneous astrometric and photometric as obtained with DIVA is expected to have a huge scientific impact on many fields of astronomy, and namely:

- calibration of stellar parameters (much better than available now)
- accurately measure numerous double stars
- discovery of nearby faint stars (e.g. brown dwarfs, metal poor subdwarfs, white dwarfs)
- determination of the spatial distribution of stars in the local disk with that, map the spatial structure of extinction
- improve the calibration of RR Lyrae to help better determine the size of the Milky Way to better determine the ages of the galactic globular clusters
- provide accurate proper motions (tangential velocities) of stars to better determine the kinematics in the Milky
- obtain accurate stellar distances perpendicular to the Milky Way disk to get better information on the vertical force K_z
- from the above two to investigate the mass contribution of Dark Matter
- improve the calibration of Cepheid stars to help better determine the distances of other galaxies to better determine the value of the Hubble constant
- obtain an accurate proper motion for both of the Magellanic Clouds

8.2 Window datation and Image Parameter Set

On the long way from raw DIVA telemetry to the final DIVA catalogue, the DIVA data reduction chain produces a lot of intermediate data of many different types [8]. The “atom” of data of a certain type is called a *record*. A record is a set of intimately connected parameters, such as e.g. the individual pixels of a DIVA “window”, along with the time of observation and the number of the CCD chip.

The individual records must be organized into larger structures, which we call *files* and which are described by parameters collected into *file header records*. Records, in turn, are composed of individual physical quantities, counters, flags etc., which we collectively call *data items*.

A specific *data set* (file or record) may by definition contain information from some part of the mission only (e.g. a single “window” or the set of “windows” from a specific mission day), or else data derived from the whole mission (e.g. an object list or one record in an astrometric output catalogue). This depends on the type of data.

The science data for the ScQL expert system should get through initial treatment in order to get image parameter sets. We describe here pre-processed data in more details.

8.2.1 Explanation of the data type descriptions

It gives descriptions (definitions) of the various data types. Every definition is divided into a verbal overview, including an estimate of the data volume, and some tables giving details. The tables describe the data in terms of records and files. This is a purely logical, not a physical description. It should not be mixed up with the technical storage and organization of the data in the central data base. The data base crew is free to avoid duplication of data items, and to organize the data such that the necessary access modes run quickly and smoothly.

In the data type description tables, each line defines one data item. The individual columns of the tables have the following meanings:

abbr.: shorthand designation of the data item (“abbreviation”).

name: full (but still short) name of the data item.

dimension: the array dimension if the data item is an array; a one-dimensional array is specified as e.g. -5:12, where the two numbers -5 and 12 denote the lower and upper limits of the element indices (the length of the array is 18 in this case); multi-dimensional arrays are specified as e.g. -5:12,1:10 (the array is two-dimensional, its total length is 180 in this case). If nothing is written it means a single number.

type: the type of the data item (the array elements); possible values are I*2, I*4, R*4, R*8, Ch*n for integer, real and character (text string) data of different word lengths; word lengths are given in bytes (2, 4 or *n*); data items of type logical or bits are specified as I-type for the moment; with the actually needed number of bits given in some instants.

offset: offset of the data item from the start of a record in an exchange format file (in Bytes).

exch. format: machine-independent ASCII format (in Fortran notation) for the data item, in files used for data exchange between different sites and/or computers

bounds: upper and lower limits for the possible values of the data item; strict bounds must be distinguished from order-of-magnitude indications (the latter being indicated by \sim characters). A * means that there is no limit.

resol.: the numerical resolution needed; if applicable

phys. unit: the physical unit in which the data item is represented; if applicable

description: a more verbose description of the meaning of the data item; usually not a complete definition in full detail.

remark: pointer to remarks, either included at the bottom of the table (indicated by the symbol *), or in some other document (indicated by the document number or title).

opt.: data items that are not necessarily present in a data record (“optional” items) are indicated by ‘yes’; the presence or absence of such items must be indicated by some flag elsewhere in the record (or in the file header, if the presence or absence is characteristic of the whole file rather than the individual records).

8.2.2 Windows

The so-called windows are the central observational results produced by the DIVA instrument(s). There is one window record per on-board-detected or -predicted star image, plus some special windows (see below). The windows have different sizes (see Table 8.3); they are created in varying rates. On average there will be about 200 records/second, but this varies between about 50 and 1500 records/second.

Mean number of windows:

$$\begin{array}{cccccc} 96/s & + & 33/s & + & 14/s & + & 56/s & + & 27/s & = & 226/s \\ \text{SM} & & \text{SC} & & \text{SC}_{zero} & & \text{UV}_l & & \text{UV}_s & & \text{total} \end{array}$$

The database will receive window data as well as all other sorts of GSOC outputs, in bunches. Thus the individual window records must be organized into larger window files (tables), which are described by file header records, see Table 8.1.

The window records are described in Tables 8.2 and 8.3. The individual items contained in the window records fall into two or perhaps three categories, to be briefly explained in the following.

Table 8.1: Windows, Header Contents

abbr.	name	dimension	type	offset	exch. mat	for-	bounds	resol.	phys. unit	description	remark	opt.
	production date		Ch*14					1s	s	start time at which data are written into the DB in the form yyyyymmddmmss in system time of the DB computer (ideally UTC)		
	producer		Ch*10							shorthand for the process creating the DB entry examples may be: routine GSOC delivery, manual replacement of faulty data, manual insertion of missing data		

Table 8.2: Window, Record Contents, Part 1: Window Datation

abbr.	name	dim.	type	offset	exch. format	bounds	resol.	phys. unit	description	remark	opt.
\underline{k}_w			I*4, 26 bit			1,...,~ 64 000 000			running number of TDI clock stroke after the latest reset, at the “lower left corner of the window	TD0201	
\overline{k}_w			I*2, 16 bit			1,...,*			running number of the latest reset of \underline{k}	TD0201	
m_w			I*2, 9 bit			-12,...,523 -12,...,2059			number of CCD column, at the “lower left corner of the window for SM, SC -12,...,-1 and 512,...,523 for underscan region for UV -12,...,-1 and 2048,...,2059 for underscan region	TD0201	
n			I*2, 5 bit			1,...,22 1,...,4 5,...,8 9,...,12 13,...,16 17,...,20 21 22			number of CCD chip for SM1 for SM2 for SM3 for SC1 for SC2 for UV _{short} for UV _{long}	TD0201	
$type$			Ch*1, 8 bit						window type identifier for SM bit 0: normal window bit 1: bright star window bit 2: guest observer object (goo) window bit 3: full-chip window bit 4: attitude star bit 5: rate pair component bit 6: predicted bit 7=0 for goo windows from fFoV bit 7=1 for goo windows from pFoV for SC bit 0 - 6: decimal value =0 → full-chip window decimal value = 6 → zero-order window decimal value =7,...,84 → z-height of the window, normal window bit 7: attitude star window for UV bit 0: normal window bit 1: full-chip window	TD0201	

Table 8.3: Window, Record Contents, Part 2: Pixel Values

abbr.	name	dimension	type	offset	exch. format	bounds	resol.	phys. unit	description	remark	opt.
	window datation		I*4, 3I*2, Ch*1								
<i>pixa</i>	pixel value array	2 dim., var. 0:12, 1:7 0:12, 1:17 1:28, 1:11 1:1024, -12:524 1:1024, -12:524 0:12, 1:7 0:12, 1:z-height -1:4, 1:4 1:256, -12:2059	I*2, 16 bit			0,...,65535	1 ADU	ADU	signal of one window, size depending on <i>n</i> and <i>type</i> for SM normal window including background (index 0) bright star window including background (index 0) guest observer object window full chip window including underscan regions (indizes -12,...,-1 and 513,...,524) for SC full chip window including underscan regions (indizes -12,...,-1 and 513,...,524) zero-order window including background (index 0) normal window including background (index 0), $7 \leq z\text{-height} \leq 84$) for UV normal window including background (index -1 and 0) full chip window including underscan regions (indizes -12,...,-1 and 2048,...,2059)	TD0201 TD0251	
<i>pkey</i>	prediction key		I*4, 3I*2, Ch*1						window datation of that SM window from which the position of this window was predicted. only for predicted SM windows and all SC windows		

8.2.3 Image Parameter Set

Image parameter sets (IPS) are the main output of the pixel data processing task. IPS records are derived from window records. An IPS contains parameters of an actual image of an astronomical source recorded in the pixel values of a window. Such parameters are: centroid position (in pixel coordinates), brightness (amplitude, integrated photo-electrons or ADC units), shape parameters (diameter, duplicity, diffuseness...), diagnostics (disturbed by cosmic ray, badly positioned in the window, truncated by CCD edge, ...). Details are defined by the pixel data processing task. The first version at defining IPS records is given in Tables 8.4 and 8.5.

There can be several images in a window numbered by the index l , and thus several IPS records derived from that window. There will be at least one IPS per window, although it can occasionally happen that there is no detectable image (in this case the IPS will contain dummy parameters).

The PICat number(s) give the preliminary identification(s) of the detected image with an object in the Preliminary Identification Catalogue (PICat). These numbers are produced at the same time as the IPS itself.

Each IPS record must carry the window datation, the image numbering l , the window identification data (if applicable), and the PICat number(s) of the corresponding PICat object(s), again if applicable.

The IPS fall into a large number of categories or types. Each of these types will be accessed by different subsequent processing steps, and in very different ways. This must carefully be taken into account by the data base crew in setting up the organization of the IPS data organization.

8.2.4 Classification according to instrument type

- SM records for normal stars
- SM records for bright stars
- SC records
- SC zero-order records
- full-chip records
- guest observer object records
- UV-long records
- UV-short records

8.2.5 Classification according to object type

- unknown object
- PICat object
- attitude star
- astrometric reference star
- photometric SM reference star
- photometric SC reference star
- photometric UV-long reference star
- photometric UV-short reference star
- guest observer object catalogue star

etc.; all categories of objects will have different usages in the data reduction chain(s). Analogously, images identified with a PICat object will be used differently from the non-identified ones.

8.2.6 Special window types, non-object windows

In addition to the obvious classification of IPS according to instrument type (8.2.4) and object type (8.2.5) there will probably be a few special types of windows which will produce image parameters sets of a completely different nature. One example might be occasional sets of complete CCD lines (the so-called full-chip windows). The pixel data processing task would not derive observations of individual objects from them, but more general things like the instantaneous background level (sky background brightness plus dark current) which may apply to more than one window of the ordinary types (8.2.4). They can also be used to check/optimize on-board image detection. Another possible example is CCD underscan windows from which the CCD bias and read-out noise levels can be derived. Again this applies to a certain stretch of other window data.

The list of such data types has not yet been finalized, nor the frequency of their occurrence or the total amount of data in them. A first sketch of this subject can be found in [12].

8.2.7 Data quantity

SM-type windows:

Tables 8.4 to 8.5 give a realistic draft of the structure and contents of IPS data for SM-type windows.

- a) information repeated from raw data:

datation ($k, m, n, type$): $11 B/window$

maybe identification (see 1.2.3): $2 B/window$ (rarely present)

- b) new information (see Table 8.5):

PICat identification (implying object classification): $4 B/window$

image counter l : $2 B/window$

image parameters proper and flags: $80 B/window$

SC-type windows (details still to be defined):

- a) information repeated from raw data:

as for SM-type records

- b) new information:

PIC identification: $4 B/window$

position-dependent amplitude: $80 \times 2 B/window$

shape parameters: $6 \times 2 B/window$

flags: $2 B/window$

Data quantity

SM-type windows:

repeated: $11 B/window = 0.11 TB/mission$

new: $82 B/window = 0.81 TB/mission$

SC-type windows:

repeated: $11 B/window = 0.02 TB/mission$

new: $174 B/window = 0.29 TB/mission$

total new: $1.23 TB/mission$ per version

In other words, since 5 or 6 versions can be expected, the total amount of IPS data may be as high as 6 to 8 TB (possible mission prolongation not taken into account).

Table 8.4: Image Parameter Set, Header Contents

abbr.	name	dimension	type	offset	exch. format	for-	bounds	resol.	phys. unit	description	remark	opt.
	version number		I*2									
	production date		Ch*14					1s	s	start time at which data are written into the DB in the form yyymmddmmss in system time of the DB computer (ideally UTC)		
	producer		Ch*10							shorthand for the process creating the DB entry examples may be: automatic pipeline processing, automatic reprocessing, manual repeat of a processing run		
	validation		I*2				1,...,5					

Table 8.5: Image Parameter Set, Record Contents

abbr.	name	dimens	type	offset	exch. format	bounds	resol.	phys. unit	description	remark	opt.
	window datation		I*4, 3I*2, Ch*1								
<i>l</i>			I*2, 2 bit			0,...,3			IPS running number		
<i>picat</i>	PICat identifier		I*4						identification number of the star in PICat		
centroid position (pixel coordinates)											
C_k		1:2	R*4			0,...,1		pixel	centroid position in scan direction reckoned from \underline{k}_w , such that $\kappa = \underline{k}_w + C_k$, computed with 2 different methods		
C_m		1:2	R*4			0,...,1		eff. pixel	centroid position in cross-scan direction reckoned from m_w , such that $\mu = m_w + C_m$, computed with 2 different methods		
brightness											
<i>I</i>		1:2	R*4			0,...,*		ADU	amplitude, computed with 2 different methods		
		1:2	R*4					ADU	background		
shape											
		1:2, 1:5	R*4						shape parameters, up to 5, depending on the method, e.g. duplicity, diameter, diffuseness, area		
quality											
		1:2	I*2, 8 bit						diagnostics, up to 8 bit, depending on the method, e.g. disturbed by cosmic ray, badly positioned in the window, truncated by CCD edge		

8.3 Attitude and nominal scanning law

8.3.1 Attitude matrix

The attitude of the satellite is the orientation of the satellite system with respect to the ICRS. It is expressed by the attitude matrix \mathbf{A} , an orthonormal 3×3 matrix. Each line of the matrix contains the equatorial (ICRS) direction cosines of one of the satellite system's principal axes. Conversely, the columns of the matrix contain the direction cosines of the ICRS principal axes expressed in the satellite system. Symbolically:

$$\mathbf{A} = \begin{pmatrix} \mathbf{x}' \\ \mathbf{y}' \\ \mathbf{z}' \end{pmatrix}_{ICRS} = (\mathbf{XYZ})_{sat.syst.} \quad (8.1)$$

where the prime denotes the transpose of a vector.

Multiplying \mathbf{A} with any vector \mathbf{u} given in ICRS coordinates produces the representation of that vector in the satellite system:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{\mathbf{u}} = \mathbf{A} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{\mathbf{u}} \quad (8.2)$$

8.3.2 The nominal scanning law

The following parameters are needed to determine the attitude at the next apogee according to the NSL:

$$\begin{aligned} a_1 &:= -0.16378459 , \\ a_2 &:= -0.01307777 , \\ a_3 &:= 0.00123243 , \\ a_4 &:= 0.00012341 , \\ b_1 &:= 0.08215269 , \\ b_2 &:= 0.99006117 , \\ b_3 &:= 0.04045213 , \\ e &:= 0.016714 , \\ g_1 &:= -0.04414 + 0.017201969d_1 , \\ K &:= 6.4 , \\ L_1 &:= 1.38691 + 0.0172021240d_1 , \\ R &:= 11.25 , \\ \epsilon &:= 0.409092804222328826 , \end{aligned}$$

$$\begin{aligned}
\lambda_1 &:= L_1 + 2e \sin g_1 + \frac{5}{4}e^2 \sin 2g_1, \\
\nu &:= \bar{\nu} + a_1 \cos \bar{\nu} + a_2 \cos 2\bar{\nu} + a_3 \cos 3\bar{\nu} + a_4 \cos 4\bar{\nu} \\
\bar{\nu} &= \nu_0 + K\lambda_1, \\
\xi &:= \frac{\pi}{4}, \\
\Omega &= \Omega_0 + 2\pi R(d_1 - d_0) - \nu \cos \xi + (b_1\bar{\nu} + b_2 \cos \bar{\nu} + b_3 \sin 2\bar{\nu}) \frac{\sin \xi}{K},
\end{aligned}$$

where d_1 is the apogee time in mean solar days from the reference epoch d_0 adopted for the NSL. Since λ_1 may exceed 2π , λ_1 modulo 2π should be used instead of λ_1 . ν_0 and Ω_0 are telecommandable constants fixing the NSL. The attitude according to the NSL is then

$$A_{\text{NSL}}(d_1) = D_3(\Omega) \left[D_2\left(\frac{\pi}{2} - \xi\right) \left[D_1\left(\nu - \frac{\pi}{2}\right) [D_3(\lambda_1) D_1(\epsilon)] \right] \right]$$

with

$$\begin{aligned}
D_1(\phi) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix}, \\
D_2(\phi) &= \begin{pmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{pmatrix}, \\
D_3(\phi) &= \begin{pmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}.
\end{aligned}$$

Multiplying this attitude matrix $A_{\text{NSL}}(d_1)$ with any vector given in the equatorial coordinate system produces the representation of that vector in the body-fixed coordinate system of the satellite. For the back transformation the transposed matrix can be used, which is the inverse matrix of the attitude matrix.

8.4 DIVA main instrument

In the following tables the collection of parameters for DIVA main instrument as well as estimation of date rates are presented:

Table 8.6: DIVA collection of parameters, Part 1 (general parameters)

1a) Common parameters, general			
scan speed	180	"/sec	omega
fields of view	2		N_fov
1b) Common parameters, derived			
rotation period	120	min	$T_{rot}=360*3600"/\omega/60sec$
exposure time for one CCD transit	1.414	sec	$t_{exp}=M_w*scalpx/\omega$
2a) Optics, general			
focal length	11200	mm	F
aperture size in w direction	220	mm	D
aperture size in z direction	110	mm	H
center of preceding aperture in z direction	95	mm	z_prec
center of following aperture in z direction	-95	mm	z_foll
2b) Optics, derived			
gap between apertures in z direction	80	mm	$gap=abs(z_{prec}-z_{foll})-H$
primary mirror, actual size in z direction	300	mm	$H_{prim}=abs(z_{prec}-z_{foll})+H$
primary mirror, max. dist. from opt. axis	186.01	mm	$R_{prim}=\sqrt{(\max(abs(z_{prec}), abs(z_{foll}))+H/2)^2 + (D/2)^2}$
image scale in mm	18.42	"/mm	$scalmm=206265/f$
image scale in pixels	0.249	"/pix	$scalpx=scalmm*pix_w/1000\mu m$
image scale in eff. pixels	0.994	"/eff.pix.	$scafeffpx=scalmm*pix_{eff}/1000\mu m$

Table 8.7: DIVA collection of parameters, Part 2 (general parameters)

3a) CCDs, general			
pixel size in w direction	13.5	mu-m	pix_w
pixel size in z direction	13.5	mu-m	pix_z
on-chip binning factor in z direction	4		binning
CCD chip size in w direction	1024	pixels	M_w
CCD chip size in z direction	2048	pixels	M_z
number of readout channels per chip	2		N_read
3b) CCDs, derived			
CCD chip size in w direction	13.82	mm	size_w=pix_w*M_w/1000mum
CCD chip size in w direction	4.24	arcmin	size_w_arc=size_w/f*180/3,141593*60
CCD chip size in z direction	27.65	mm	size_z=pix_z*M_z/1000mum
CCD chip size in z direction	8.49	arcmin	size_z_arc=size_z/f*180/3,141593*60
effective pixel size in z direction	54	mu-m	pix_eff=pix_z*binning
4a) Focal plane assembly, general			
total number of chips in SM	8	chips	N_SM
number of chips needing image detection	4	chips	N_imadet
total number of chips in SC	8	chips	N_SC
field of view width in z direction, SM	4	chips	fov_sm
field of view width in z direction, SC	4	chips	fov_sc
4b) Focal plane assembly, derived			
total number of chips	16	chips	N=N_SM+N_SC
field of view width in z direction, SM	33.95	arcmin	size_fov_sm
field of view width in z direction, SC	33.95	arcmin	size_fov_sc

Table 8.8: DIVA collection of parameters, Part 3 (data rates)

5) TDI, derived			
TDI clock rate	724.0	lines/sec	$\text{TDI_rate} = \omega / \text{scalpx}$
TDI clock interval	1.381	msec	$\text{TDI_time} = 1 / \text{TDI_rate} * 1000 \text{msec}$
5a) CCD data rates, general			
number of data bits per pixel	16	bit	bitpix
5b) CCD data rates, derived			
CCD data rate, SM	47.45	Mbit/sec	$\text{dat_SM} = N_{\text{SM}} * M_z / \text{binning} * \text{TDI_rate} * \text{bitpix} / 10^6 \text{ Mpix}$
CCD data rate, image detection	23.72	Mbit/s	$\text{dat_imadet} = N_{\text{imadet}} * M_z / \text{binning} * \text{TDI_rate} * \text{bitpix} / 10^6 \text{ Mpix}$
CCD data rate, SC	47.45	Mbit/sec	$\text{dat_SC} = N_{\text{SC}} * M_z / \text{binning} * \text{TDI_rate} * \text{bitpix} / 10^6 \text{ Mpix}$
CCD data rate, total	94.89	Mbit/sec	$\text{dat} = \text{dat_SM} + \text{dat_SC}$
6a) image rates, general			
number of stars, SM, all sky	$35 \cdot 10^6$	stars	stars_SM
max/mean star density, SM SM (Milky Way)	3.3		dense_SM
min/mean star density, SM SM (galactic pole)	0.21		sparse_SM
number of stars, SC, all sky	$12 \cdot 10^6$	stars	stars_SC
max/mean star density, SC SC (Milky Way)	3.0		dense_SC
min/mean star density, SC SC (galactic pole)	0.23		sparse_SC

Table 8.9: DIVA collection of parameters, Part 4 (data rates)

6b) image rates, derived			
area of sky per second, SM	0.113	sq deg	$\text{skysec_SM} = \text{fov} * \text{N_SM} * \text{size_z_arc} / 60 * \text{omega} / 3600$
area of sky per second, image detection	0.057	sq deg	$\text{skysec_imadet} = \text{fov} * \text{N_imadet} * \text{size_w_arc} / 60 * \text{omega} / 3600$
area of sky per second, SC	0.113	sq deg	$\text{skysec_SC} = \text{fov} * \text{N_SC} * \text{size_w_arc} / 60 * \text{omega} / 3600$
stars per second, SM, mean	96.0	stars/sec	$\text{rate_stars_SM} = \text{skysec_SM} * \text{stars_SM} / 41253$
stars per second, SM, max	316.8	stars/sec	$\text{rate_dense_SM} = \text{skysec_SM} * \text{stars_SM} * \text{dense_SM} / 41253$
stars per second, SM, min	20.2	stars/sec	$\text{rate_sparse_SM} = \text{skysec_SM} * \text{stars_SM} * \text{sparse_SM} / 41253$
stars per second, image detection, mean	48.0	stars/sec	$\text{rate_stars_imadet} = \text{skysec_imadet} * \text{stars_SM} / 41253$
stars per second, image detection, max	158.4	stars/sec	$\text{rate_dense_imadet} = \text{skysec_imadet} * \text{stars_imadet} * \text{dense_SM} / 41253$
stars per second, image detection, min	10.1	stars/sec	$\text{rate_sparse_imadet} = \text{skysec_imadet} * \text{stars_imadet} * \text{sparse_SM} / 41253$
stars per second, SC, mean	32.9	stars/sec	$\text{rate_stars_SC} = \text{skysec_SC} * \text{stars_SC} / 41253$
stars per second, SC, max	98.7	stars/sec	$\text{rate_dense_SC} = \text{skysec_SC} * \text{stars_SC} * \text{dense_SM} / 41253$
stars per second, SC, min	7.6	stars/sec	$\text{rate_sparse_SC} = \text{skysec_SC} * \text{stars_SC} * \text{sparse_SM} / 41253$

Table 8.10: DIVA collection of parameters, Part 5 (data rates)

7a) image data rates, general			
window size, SM, in w direction	12	pix	win_w_SM
window size, SM, in z direction	7	eff pix	win_z_SM
window size, SC, in w direction	12	pix	win_w_SC
window size, SC, in z direction	60	eff pix	win_z_SC
7b) image data rates, derived			
SM, mean	129.0	kbit/sec	$\text{dat_stars_SM} = \text{rate_stars_SM} * \text{win_w_SM} * \text{win_z_SM} * \text{bitpix} / 10^3$
SM, max	425.8	kbit/sec	$\text{dat_dense_SM} = \text{dat_stars_SM} * \text{dense_SM}$
SM, min	27.1	kbit/sec	$\text{dat_sparse_SM} = \text{dat_stars_SM} * \text{sparse_SM}$
SC, mean	379.2	kbit/sec	$\text{dat_stars_SC} = \text{rate_stars_SC} * \text{win_w_SC} * \text{win_z_SC} * \text{bitpix} / 10^3$
SC, max	1137.5	kbit/sec	$\text{dat_dense_SC} = \text{dat_stars_SC} * \text{dense_SC}$
SC, min	87.2	kbit/sec	$\text{dat_sparse_SC} = \text{dat_stars_SC} * \text{sparse_SC}$

8.5 Acronym Collection

ACS	Attitude Control System
ADU	Analogue to Digital Unit (corresponded to a half of CCD chip)
AI	Artificial Intelligence
CCD	Charge Coupled Device
DPC	DIVA Data and Processing Center
ESOC	European Space Operations Center
ES	Expert System
FOV	Field-of-View
PFOV	Preceding FOV
FFOV	Following FOV
FL	First Look
FPA	Focal-Plane Assembly
GSOC	German Space Operations Center
HK	House Keeping Data
H/W	Hardware
IDL	Interactive Data Language (Research Systems, Inc.)
IPS	Image Parameter Set
IPU	Instrument Processing Unit
PL	Pipeline
QL	Quick Look
ScQL	Science Quick Look
MMU	Mass Memory Unit
NSL	Nominal Scanning Law
PICat	Preliminary Identification Catalogue
S/W	Software
VC	Virtual Channel

Bibliography

- [1] DIVA. *Web Site: <http://www.ari.uni-heidelberg.de/diva>.*
- [2] European Space Agency. The Hipparcos and Tycho Catalogues. *SP-1200*, 1997.
- [3] S. Ansari and et al. Gaia Spectroscopy, Science and Technology. The Gaia data access and analysis study. volume 298, page 97, 2003.
- [4] Astrium. Gaia System Level Technical Reassessment Study: Final Report. 2002.
- [5] Elias M. Awad. *Building Expert Systems: Principles, Procedures, and Applications*. Addison-Wesley Publishing Company, Inc., 1996.
- [6] U. Bastian. On-board attitude star catalog handling for diva. *DIVA Archive*, 4 2001.
- [7] U. Bastian and M. Biermann. DIVA: Criteria for successful on-board image detection. *DIVA Archive*, September 2001.
- [8] U. Bastian, M. Biermann, and S. Hirte. IDAP: Interface Document for Diva Astrometry and Pipeline. *DIVA Archive*, February 2002.
- [9] U. Bastian, S. Biermann, S. Hirte, and S. Jordan. Algorithms for the DIVA IPU software (V1.4). *DIVA Archive*, March 2001.
- [10] U. Bastian and S. Hirte. Conventions and notations for the basic model of the instrument and of the telemetry measurements. *DIVA Archive*, February 2001.
- [11] U. Bastian, S. Roeser, Kappelmann, Mandel, E. Schilbach, De Boer, R. Scholz, and C.Schalinski. Level 0: Anforderungsdokument (V2.2). *DIVA Archive*, July 2001.
- [12] U. Bastian, E. Schilbach, and M. Biermann. A model observation strategy for DIVA. *DIVA Archive*, November 2001.
- [13] U. Bastian, E. Schilbach, and S. Roeser. DIVA, der 35 000 000-sterne-satellit. eine astrometrisch-photometrische mission des 21. jahrhunderts. *Sterne und Weltraum*, 10:842–851, 1999.
- [14] D.E. Bernard, G.A. Dorais, C. Fry, E.B. Gamble Jr., B. Kanefsky, J. Kurien, W. Millar, N. Muscettola, P.P. Nayak, B. Pell, K.Rajan, N.Rouquette, B. Smith, and B.C. Williams. Design of the remote agent experieiment for the spacecraft autonomy. 1998.
- [15] Burkhard D. Burow. Mixed Language Programming. 18–22 September 1995. "Presented in parallel session 'E.1 Languages' at Computing in High Energy Physics (CHEP95), Rio de Janerio, Brazil".
- [16] S. Cauvin, M.O. Cordier, C. Dousson, P. Laborie, F.Lévy, J.Montmain, M.Porcheron, I.Servet, and L.Travé-Massuyès. Monitoring and alarm interpretation in industrial environments. *AI Communications Journal*, Vol.11(3/41):139–173, 1998.
- [17] J. Chen and R.J. Patton. *Robust model-based fault diagnosis for dynamic systems*. Kluwer academic publishers, Boston, 1999.
- [18] L. Console, D.T. Dupré, and P. Torasso. Abductive reasoning through direct deduction from completed domain models. *Methodologies for intellegent systems*, vol. 4:175 – 182, 1989.
- [19] C.W.Allen. *Astrophysical Quantities*. Athlone Press, 1973.
- [20] R. Davis, B. Buchanan, and E. Shortliffe. Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence*, 8(1):15–45, 1977.
- [21] R. Davis and W. Hamscher. *Exploring Artificial Intelligence*. Morgan Kaufmann Publishers, 1988.

- [22] J. de Kleer. Physics, qualitative. *Artificial Intelligence*, 1992.
- [23] J. de Kleer and B.C.Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.
- [24] J. de Kleer and J. Kurien. Fundamentals of model-based diagnosis.
- [25] D. Dvorak and B. Kuipers. Model-based monitoring of dynamic systems. pages 1238–1243, 1989.
- [26] A. Sozzeti et al. The GAIA astrometric survey of the solar neighborhood and its contribution to the target database for DARWIN/TPF.
- [27] W. H. Press et al. *Numerical Recipes: The Art of Scientific Computing*. N.Y., Cambridge University Press, 1992.
- [28] Hayes-Roth Frederick, A. Waterman Donald, and B. Lenat Douglas, editors. *Building Expert Systems*. West Publishing Company, 1983.
- [29] F. Fugeras and et al. The three dimensional universe with Gaia. Data processing: core tasks. Januar 2005.
- [30] J.J. Getler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker Inc. (1 ed.), 1998.
- [31] J. Giarratano and G. Riley. *CLIPS Reference Manual*. PWS Publishing Company, Boston, 1998.
- [32] J. Giarratano and G. Riley. *CLIPS User's Manual Volume II: Objects*. PWS Publishing Company, Boston, 1998.
- [33] G. Gilmore. New light on faint stars. VI - Structure and evolution of the Galactic spheroid. *Royal Astronomical Society, Monthly Notices (ISSN 0035-8711)*, 207:223–240, 1984.
- [34] S. Hirte. DIVA An-Bord-Auswertung der Attitude-CCD-Daten und Genauigkeitsbilanz. *DIVA Archive*, February 1998.
- [35] S. Hirte. Exact centroiding for attitude determination. *DIVA Archive*, December 2001.
- [36] S. Hirte. Comparison of different methods for the determination of centroids from diva measurements. *DIVA Archive*, August 2003.
- [37] L. Lindengren J. Holmberg, C. Flinn. *ESA Hipparcos Venice'97 Symposium*, 1997.
- [38] H. Jahreis and R. Wielen. *ESA Hipparcos Venice'97 Symposium*, 1997.
- [39] R.M. Soneira J.N. Bahcall. The universe at faint magnitudes. I - Models for the galaxy and the predicted star counts. *Astron. Zhurn.*, 17:3, 1940.
- [40] R.M. Soneira J.N. Bahcall. The universe at faint magnitudes. I - Models for the galaxy and the predicted star counts. *Astrophysical Journal Supplement Series*, 44:73–110, 1980.
- [41] S. Jordan. DIVA: Block-skipping algorithm for acceleration of on-board object detection. *DIVA Archive*, January 2002.
- [42] S. Jordan. DIVA: Possible simple object detection algorithm. *DIVA Archive*, February 2002.
- [43] C. Jordi, C. Babusiaux, D. Katz, J. Portell, and F. Arenou. Gaia: satellite and instrument parameters. *UB-SWG-012*, 2003.
- [44] N. Kharchenko and E. Schilbach. Program Mega: Stellar Counts and Galactic Models. *Baltic Astronomy*, 5:337–356, 1996.
- [45] N. Kharchenko, S.Rybka, A. Yatsenko, and E. Schilbach. Predicted star counts and mean parallaxes down to the 23rd magnitude. *Astron. Nachr.* 318, 3:163–171, 1997.

- [46] Christopher 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.
- [47] J. Kovalevsky. *Modern Astrometry*. Springer-Verlag Berlin Heidelberg New York, 1995.
- [48] B. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(1):289–338, 1986.
- [49] L. Lindegren and M.A.C. Perryman. Gaia: Global astrometric interferometer for astrophysics. *A&AS*, 116:579, 1997.
- [50] M.A. Massoumnia, G.C. Verghese, and A.S. Wilsky. Failure detection and identification. volume 34, pages 316–321, 1989.
- [51] C.A. Murray, M.J Penston, J.J. Binney, and N. Houk. *ESA Hipparcos Venice'97 Symposium*, 1997.
- [52] A. Nathaniel. Interfacing C and FORTRAN. *CERN Computer Newsletter*, (217):9, July – Spetember 1994.
- [53] A.I. Pavlov and A.N. Belikov. Mixed Language Programming in the framework of Diva project. *DIVA Archive*, July 2002.
- [54] W. Pitz and the DIVA Team of Astrium. DIVA System Design and Interface Document (Issue 1). *DIVA Archive*, March 2001.
- [55] R. Reiter. A theory of dianosis from the first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
- [56] Inc. 1994 Research Systems. IDL User's Guide. <http://www.rsinc.com/idl/>.
- [57] S. Roeser. DIVA - beyond HIPPARCOS and towards GAIA. *Reviews in Modern Astronomy – Astronomical Instruments and Methods at the Turn of the 21th Century*, 12:97–106, 1999.
- [58] J.M. Scalo. The stellar initial mass function. *Fundamentals of Cosmic Physics (ISSN 0094-5846)*, 11:1–278, 1986.
- [59] R.C. Schank and R.P. Abelson. *Scripts, Plans, Goals and Understanding*. Erlbaum, Hillsdale, New Jersey, US, 1977.
- [60] S. Scheithauer, S. Theil, and M. Wiegand U. Bastian. Diva post-mission attitude reconstruction: the Great-Circle Reduction. *AN*, 322:197, 2000.
- [61] M. Schmidt. *Astrophysical Journal*, 137:758, 1963.
- [62] A.S. Sharov. *Astron. Zhurn.*, 40:900, 1963.
- [63] J.W. Sheppard and W.R. Simpson. Improving the accuracy of diagnostics provided by fault dictionaries. Number 3, pages 180–186, 1996.
- [64] 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.
- [65] M. A. Stephens. *Journal of the Royal Statistical Society*, 32:115–122, 1970.
- [66] J. Torra and et al. The three dimensional universe with Gaia. Design and overview of the data processing. Januar 2005.
- [67] Hwee Tou Ng. Model-based, multiple-fault diagnosis of time-varying, continous physical devices. *Proceedings of the Sixth IEEE Conference on AI Applications.*, pages 9–15, 1990.
- [68] U.Bastian, M.Biermann, and A.I. Pavlov. Relational data base or file system. *DIVA Archive*, March 2002.
- [69] U.Cortés, I. Roda, C.Mérida-Campos, and E.Rollón. RIMSAT: Deliverable 4.1. Report

on current reasoning engine practice and integration strategies. *University of Girona*, 2003.

- [70] B.C. Williams and P.P. Nayak. A model-based approach to reactive self-configuring systems. pages 971–978, 1999.

This dissertation is submitted to the Lehrstuhl für Praktische Informatik III, Universität Mannheim in fulfillment of the requirements for the Doctor of Natural Sciences.

This thesis was typeset using \LaTeX_ϵ in `report` document class.
Drawing were made in `VISIO`[™] from Microsoft and `CorelDraw`[™] from Corel Corporation.
Graphs were generated in `IDL`[™] from Research Systems Inc.

This research was supported by grant from Klaus Tschira Foundation (KTS).