Data analysis for the LISA Technology Package

M Hewitson[1], **M Armano**[2], **M Benedetti**[3], **J Bogenstahl**[4], **D Bortoluzzi**[5], **P Bosetti**[5], **N Brandt**[6], **A Cavalleri**[7], **G Ciani**[7], **I Cristofolini**[5], **M Cruise**[8], **K Danzmann**[1], **I Diepholz**[1], **R Dolesi**[7], **J Fauste**[2], **L Ferraioli**[7], **D Fertin**[9], **W Fichter**[10], **A García**[1], **C García**[9], **A Grynagier**[10], **F Guzmán**[1], **E Fitzsimons**[4], **G Heinzel**[1], **D Hollington**[11], **J Hough**[4], **M Hueller**[7], **D Hoyland**[8], **O Jennrich**[9], **B Johlander**[9], **C Killow**[4], **A Lobo**[12], **D Mance**[13], **I Mateos**[12], **P W McNamara**[9], **A Monsky**[1], **D Nicolini**[9], **D Nicolodi**[7], **M Nofrarias**[1], **M Perreur-Lloyd**[4], **E Plagnol**[14], **G D Racca**[9], **J Ramos-Castro**[15], **D Robertson**[4], **J Sanjuan**[12], **M O Schulte**[11], **D N A Shaul**[11], **M Smit**[16], **L Stagnaro**[9], **F Steier**[1], **T J Sumner**[11], **N Tateo**[7], **D Tombolato**[7], **G Vischer**[1], **S Vitale**[7], **G Wanner**[1], **H Ward**[4], **S Waschke**[11], **V Wand**[1], **P Wass**[7], **W J Weber**[7], **T Ziegler**[6] and **P Zweifel**[13]

[1] Albert-Einstein-Institut, Max-Planck-Institut für Gravitationsphysik und Universität Hannover, 30167 Hannover, Germany
[2] European Space Agency, ESAC, Villanueva de la Cañada, 28692 Madrid, Spain
[3] Dipartimento di Ingegneria dei Materiali e Tecnologie Industriali, Università di Trento and INFN, Gruppo Collegato di Trento, Mesiano, Trento, Italy
[4] Department of Physics and Astronomy, University of Glasgow, Glasgow, UK
[5] Dipartimento di Ingegneria Meccanica e Strutturale, Università di Trento and INFN, Gruppo Collegato di Trento, Mesiano, Trento, Italy
[6] Astrium GmbH, 88039 Friedrichshafen, Germany
[7] Dipartimento di Fisica, Università di Trento and INFN, Gruppo Collegato di Trento, 38050 Povo, Trento, Italy
[8] Department of Physics and Astronomy, University of Birmingham, Birmingham, UK
[9] European Space Agency, ESTEC, 2200 AG Noordwijk, The Netherlands
[10] Institut für Flugmechanik und Flugregelung, 70569 Stuttgart, Germany
[11] The Blackett Laboratory, Imperial College London, UK
[12] Institut d'Estudis Espacials de Catalunya and Institut de Ciencies de l'Espai, CSIC, 08034 Barcelona, Spain
[13] Institut für Geophysik, Eidgenössische Technische Hochschule Zürich, CH-8093, Zürich, Switzerland
[14] APC UMR7164, Université Paris Diderot, Paris, France
[15] Universitat Politècnica de Catalunya, 08034 Barcelona, Spain
[16] SRON Netherlands Institute for Space Research, 3584 CA Utrecht, The Netherlands

E-mail: martin.hewitson@aei.mpg.de

Abstract
The LISA Technology Package (LTP) on board the LISA Pathfinder mission aims to demonstrate some key concepts for LISA which cannot be tested on ground. The mission consists of a series of preplanned experimental runs. The data analysis for each experiment must be designed in advance of the mission.

During the mission, the analysis must be carried out promptly so that the results can be fed forward into subsequent experiments. As such a robust and flexible data analysis environment needs to be put in place. Since this software is used during mission operations and effects the mission timeline, it must be very robust and tested to a high degree. This paper presents the requirements, design and implementation of the data analysis environment (LTPDA) that will be used for analysing the data from LTP. The use of the analysis software to perform mock data challenges (MDC) is also discussed, and some highlights from the first MDC are presented.

PACS numbers: 95.55.Ym, 04.80.Nn

1. Introduction

The LISA Technology Package [1] (LTP) on board the LISA Pathfinder [2] mission (LPF) aims to demonstrate some key concepts for LISA which cannot be tested on ground. The mission is planned to launch in 2011 with the main goal being to demonstrate the ability to put a test-mass into free fall at a level where any residual acceleration is below 3×10^{-14} m s$^{-2}/\sqrt{\text{Hz}}$ at frequencies around 1 mHz [3]. In order to demonstrate this ability, it is necessary to measure the position of the free falling test-mass. To do this, a second quiet reference test-mass is included and an interferometric readout [4, 5] of the differential displacement of the two masses is performed. Finally, in order to shield the test-masses and to provide infrastructure for the measurement apparatus and control, the two test-masses are enclosed in inertial reference sensors (IRS) [6] which are then placed inside vacuum chambers in a single spacecraft (SC). A more recent discussion of the mission hardware and its status can be found in [7].

In the main science operating mode, the position of the SC relative to the first test-mass is controlled using micro-Newton thrusters attached to the SC. The position of the second test-mass is controlled using capacitive actuators surrounding the test-mass. Details of the drag-free control system can be found in [8].

The mission itself will consist of a series of *runs*, each run consisting of one or more experiments. The runs will be packed into the finite mission time in an as efficient way as possible, in order to maximize the science output of the mission. The complete collection of runs forms the Experiment Master Plan (EMP) of the mission. Many of the experiments will focus on the assessment of the residual differential force noise between the two test-masses under varying conditions. In addition, many other aspects of the experiment, such as thermal effects, interferometer performance and magnetic effects, will be characterized with the hope of fully understanding the contribution of the various noise sources to the test-mass acceleration.

Each experiment of the mission will be analysed and planned prior to the start of the mission. The results of each experiment give information that will be carried forward into the following experiments. In particular, if unexpected results are produced by an experiment, the subsequent experiments may need to be altered to avoid loss of mission time or to avoid problems on the SC. In order to achieve this, the data from each experiment must be promptly analysed in the Science and Technology Operations Centre (STOC). The analysis and design of each experiment, together with confirmation of the robustness and readiness of the data analysis tools, is partly done through the writing of detailed technical notes which define and

analyse the sensitivity of each experiment, as well as by a series of mock data challenges (MDCs) which aim to perform realistic simulations of each experiment. The correct design of each experiment is vital to the mission's success. In some cases, careful analysis of a particular experiment can lead to corrections or adjustments of the mission hardware, which are difficult to envisage on their own. The experiments themselves also define, and are constrained by, the operation of the mission. In the end, a complex interaction between mission design, hardware design, data analysis and experiment planning is required to maximize the success of the mission. Examples of the experiment designs are given in [9, 10].

2. Requirements and mission goals

The need to perform quasi online data analysis in a robust and flexible way has led to a series of requirements on the data analysis software. In particular, the software must be user-friendly, thoroughly tested and able to cope with on-the-fly analysis design. The need to have accountable and reproducible analysis results to avoid information loss between LPF and LISA was also identified. The aim is to avoid having analysis results in the form of plots/figures/documents without full details of the supporting analysis. In addition, in order to ensure that the data and results of the mission have a long life time, they are to be stored to disk in an ASCII file format.

Since contact with the SC is limited to a few hours per day, a limited amount of time will be available between receiving the data from a particular experiment and uploading of telecommands to alter the mission timeline based on the results of that experiment. Hence, the data from the last experiment must be analysed quickly so that any necessary action can be taken to optimize subsequent experiments. To speed up the analysis, multiple scientists will need to access and analyse the data in a concurrent way. As such, a centralized data repository is needed to deliver and receive data analysis products.

An additional constraint on the analysis software is that it should be accessible to scientists who are not experts in programming. As such, a graphical user interface that allows the construction and execution of data analysis pipelines is a required component of the final package.

3. Analysis tools and environment

In order to perform the data analysis for LTP a software tool has been developed to meet the demanding requirements discussed above. This tool is referred to as LTPDA and its design and development are discussed in this section.

3.1. The concept of analysis objects

The requirement to have accountable and reproducible data analysis products led to the concept of analysis objects (AOs). An AO aims to capture much more than just numerical data. For example, an AO may represent a time series, and as such will contain a vector of timestamps and a vector of data values, together with additional information, such as a sample rate and a start time.

Figure 1 shows the contents of an AO. In particular, an AO stores detailed metadata about where the object originates as well as a full processing history for the numerical data held in the object.

Keeping all this information together in this compact form allows analysis results to be viewed, reproduced, reused or inspected at a later date. This concept is developed further to
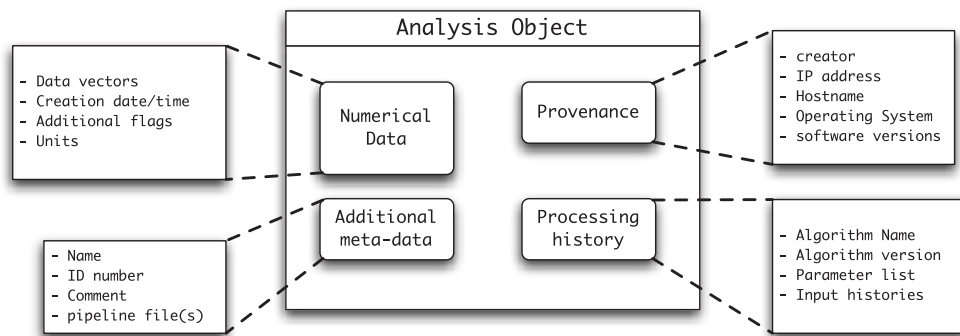
Figure 1. Schematic of the structure of an analysis object. The diagram expands the first layer of an AO to reveal some details of the underlying structure.
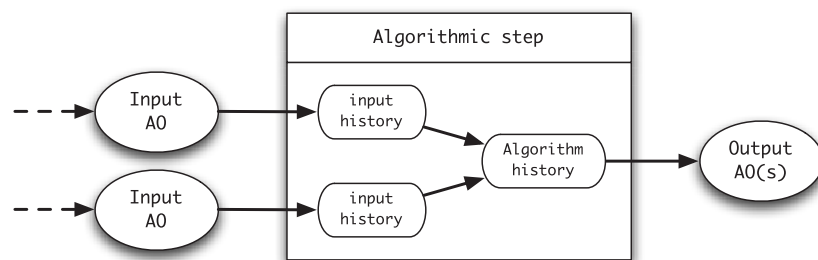

Figure 2. Schematic of the history capturing concept for algorithms working on analysis objects.

include other types of objects (not just numerical data objects); this is discussed in section 3.3. In general, we refer to objects like AOs as LTPDA objects.

3.2. Tracking processing history

The tracking of the full processing history requires that all analysis algorithms properly deal with the history of any input AOs. In addition, each algorithm has to add its own processing history. The general form of such an *history aware* algorithm is shown schematically in figure 2.

When each algorithm is applied to an LTPDA object, it adds an entry in the history tree of that object. Each entry, or *history step*, contains:

- the name of the algorithm,
- the version of the algorithm,
- the configuration parameters for the algorithm,
- the names of the input objects,
- the history trees associated with each input object.

In this way, a processing tree is built up as an object passes through an analysis pipeline.

Once all this information is collected it can be extracted from the object and used, for example, to view the processing history, to rebuild the object following the same processing steps or to reproduce the processing pipeline for alteration. An example of the processing tree of a particular analysis result is shown in figure 3. The diagram is produced by automatically converting the history tree into a DOT file [11] and from there to a PDF.
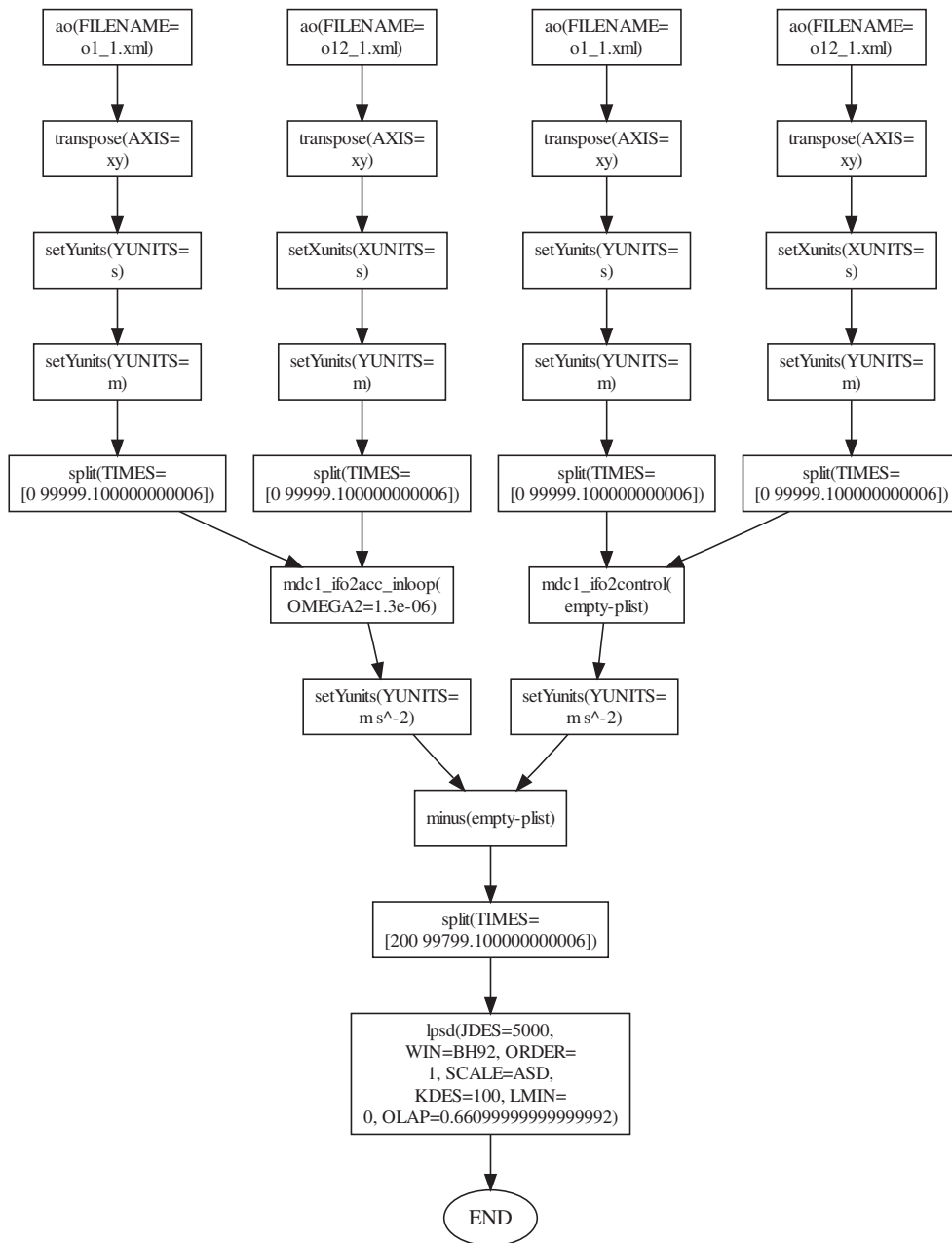
Figure 3. History tree of MDC1 signal processing pipeline. The tree is produced automatically by calling the command `dotview` on the AO history object.

3.3. Implementation

Due to the requirements listed above and the availability of experience within the data analysis development team, MATLAB [12] was chosen as the platform for implementing the data

Table 1. Examples of the main user classes implemented in the LTPDA toolbox.

Class name	Description
ao	Implementation of analysis objects.
tsdata	Store data representing a timeseries.
pzmodel	Implements S-domain pole/zero models.
miir	Implements IIR digital filters.
ssm	Implements statespace model objects.
history	A class to store each history step. The complete history of an analysis is then a linked list of such objects.
plist	A class to store parameter lists. A parameter has a *key name* and a *value*.

analysis software. The concept of AOs, and more generally, LTPDA objects lends itself naturally to an object oriented approach for the software architecture.

The full data analysis software is implemented as a MATLAB toolbox [13]. Some of the main classes of objects that make up the system are shown in table 1. In addition to these classes, many other classes exist to implement, for example, different data types, SI units and spectral windows.

With this infrastructure of classes and class methods (algorithms) in place, the development of a graphical user interface (GUI) to allow for easy assembly and execution of data analysis pipeline is relatively straightforward. The chosen approach is to use Simulink (already part of MATLAB) as a drawing pad where the user composes a data analysis pipeline from a library of available LTPDA blocks. The library is dynamically generated from the contents of the toolbox such that easy maintainability is ensured. The pipeline is then executed by calling the underlying class methods represented by the blocks on the diagram. In this way, the GUI adds no additional processing functionality to the toolbox, but instead provides an additional interface to the standard scripting one.

Having constructed LTPDA objects, they can be saved to disk in an XML format. This satisfies one additional requirement, namely that the data for the mission should be stored in an ASCII form to increase their lifetime.

Development of the complex algorithms required to carry out the analysis needed for LTP is driven by need. For example, the mock data challenges discussed below help guide the algorithm development. Many standard signal-processing algorithms are implemented as a matter of course, making the toolbox more generally useful for the analysis of scientific experiments. Further details of some of the algorithms are given in [14].

3.4. Centralized data access

In order to give concurrent access to analysis results, a data repository was designed with a MySQL [15] database at its core. The LTPDA objects are stored directly in the database. At the same time, much of the metadata is extracted from the objects and stored in various tables in the database to allow for easy searching of analysis products. For the LPF mission, of order 100 raw-data AOs per day is expected. These data will then be analysed and the products stored back in the database. As such we might expect of order 150 AOs per day of the mission. For a 90 day mission, this leads to many thousands of objects in the database. Therefore, the ability to search for previous results is vital since an analysis at the end of the mission may
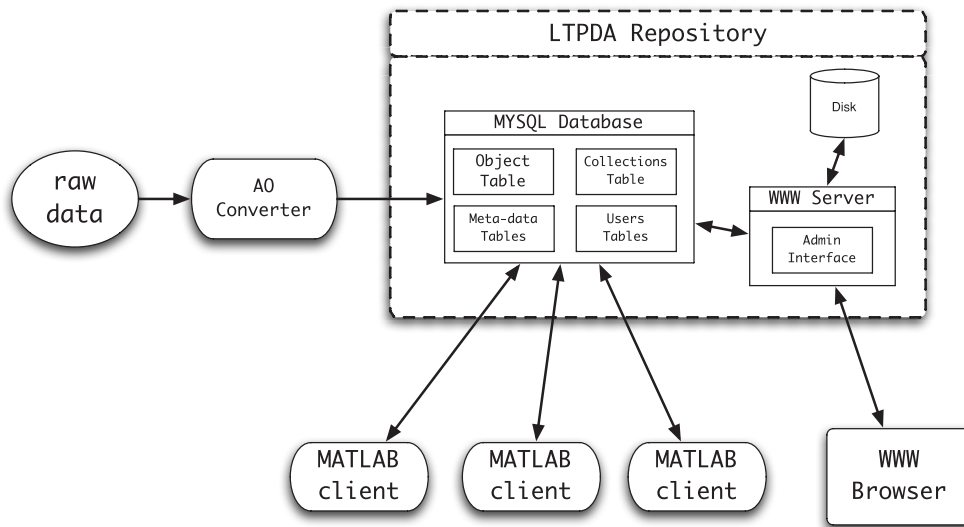
Figure 4. Schematic of an LTPDA repository set-up. The core MySQL database holds all the submitted LTPDA objects as well as detailed metadata about each object. Interaction with the repository can be done directly with the LTPDA toolbox or with a standard web browser.

require input from results generated at the beginning of the mission. A database system is obviously the best choice for this.

Using a well-known database system such as MySQL has the benefit of readily available client interfaces. In particular, it is relatively simple to create a management interface using a combination of standard high-level scripting languages and a web server. Also, various MATLAB interfaces for MySQL exist, allowing easy storage and retrieval of analysis products directly from within MATLAB.

Using a database client/server approach means we can have a central LTPDA repository which can host multiple databases. The user access to each database can be controlled at a fine level.

An overview of how this system will fit into the STOC is shown in figure 4.

The raw data from the SC will be converted into AOs then inserted into the database. The multiple MATLAB clients can then access the raw-data AOs and produce analysis products which can then be uploaded to the same (or a different) database. In addition, a web-based interface has been designed where the objects in the repository can be searched for and downloaded via a web browser.

3.5. Testing

Since the LTPDA package will be used during mission operations to analyse the data coming from the SC, and since the results of any analysis will have a bearing on subsequent experiments in the mission timeline, the software must be tested to a high degree.

The testing of LTPDA is done in four ways. First of all, each elementary algorithm in the toolbox is tested with a suite of unit tests. We test two aspects: that the algorithm can be called as described in its documentation (syntax test) and that the algorithm returns correct results (algorithm test). For release 1.0 of the toolbox, of order 1800 unit tests were performed (and passed).
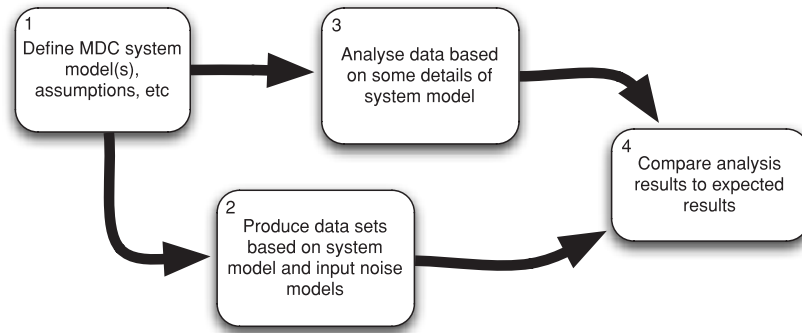
Figure 5. A flow diagram depicting the process involved in LTP MDCs.

The second level of testing involves connecting various algorithms (or blocks) in a series of meaningful ways. Here the aim is to ensure that the various functions in the toolbox are compatible with each other according to the software design requirements.

The final two levels of testing are more focussed on the user. The third level is achieved by using the toolbox to perform mock data challenges (see section 4), exercising the toolbox in realistic scenarios. The fourth level of testing is known as 'acceptance testing'. This is a formal test campaign carried out by ESA on the delivered software. The aim is to ensure that the software can integrate into the STOC and that it is capable of performing its role in the mission.

4. Mock data challenges

Mock data challenges (MDCs) aim to simulate a particular experiment of the mission using simulated data, but the actual analysis tools. As such, they provide a good vehicle for steering the development of the analysis algorithms. They also provide valuable user-level testing in realistic analysis situations. As an obvious byproduct, the scientists involved in the MDCs get useful training and can give feedback on the analysis contained in the EMP. For LTP, we have started a series of MDCs which should converge with the development of the EMP such that each of the main experiments of the mission is represented by one or more MDCs. The general form of an MDC is shown in figure 5.

4.1. The first mock data challenge

The first MDC for LTP was based on a simple 1D model of LTP. The main aim of this MDC was to build up the core algorithms for converting the interferometer outputs to out-of-loop acceleration.

MDC 1 is performed under the assumption that the SC is in the standard science operating mode (M3) where the first test-mass (TM 1) is drag free, the SC follows TM 1 via the micro-Newton thrusters, and test-mass 2 (TM 2) follows the SC via the capacitive actuation in the IRS. Figure 6 shows a model of the SC together with the main forces acting on the system. Figure 7 shows a schematic of the two control loops included in this model. The main parameters of the model are:

ω_1^2—the stiffness of the coupling of TM 1 to the SC.
ω_3^2—the stiffness of the coupling of TM 2 to the SC.
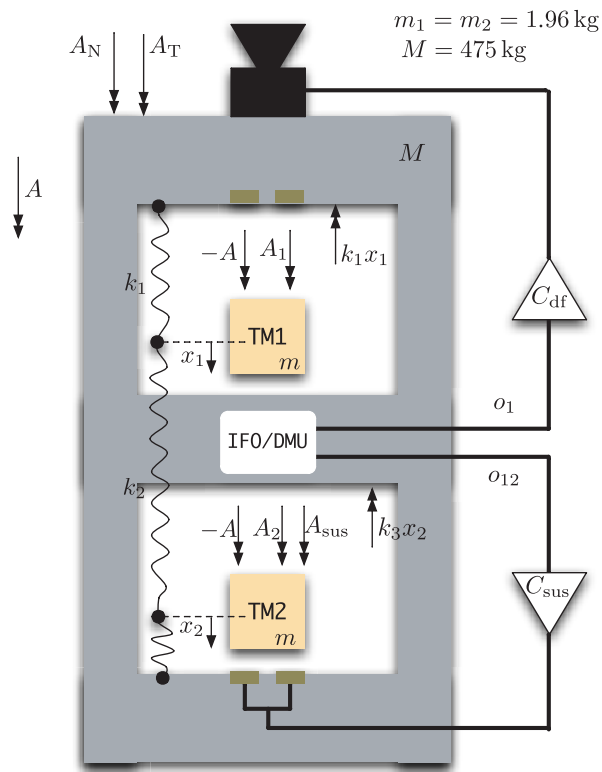
Figure 6. Schematic representation of the spacecraft operating the M3 Science Mode, as simulated by MDC1.
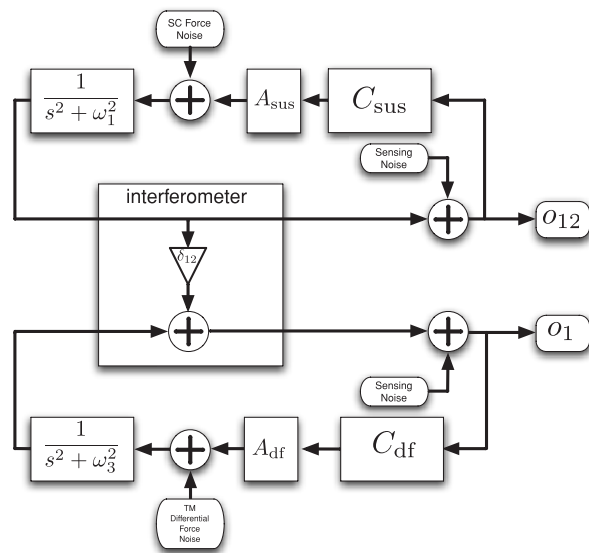

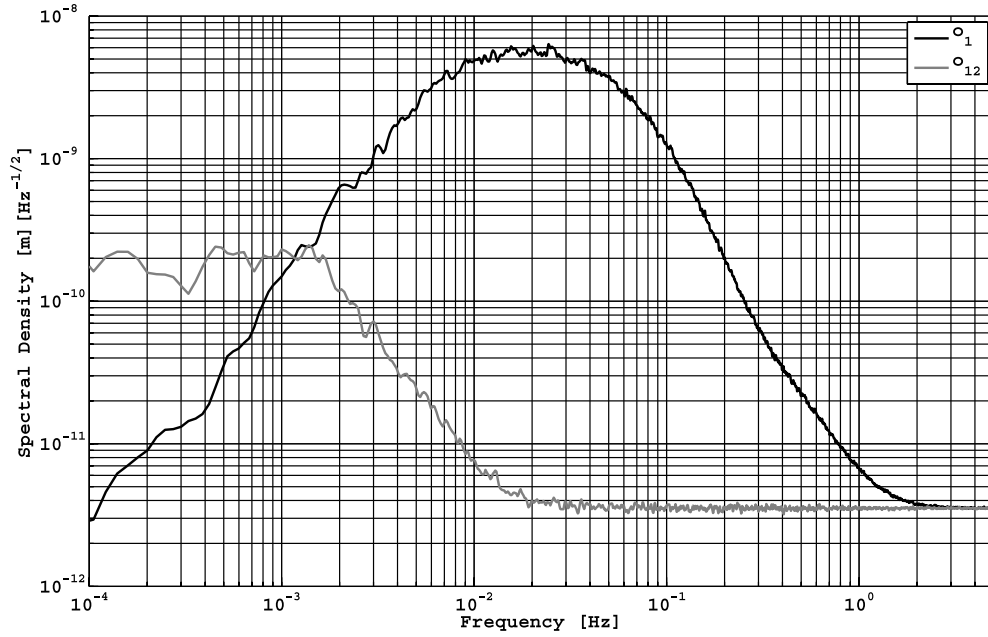**Figure 7.** Schematic representation of the model for MDC1.

Figure 8. Amplitude spectral density (ASD) estimates of the simulated interferometer outputs. The darker curve is the ASD of the output of the interferometer which measures the differential displacement of the SC and TM 1. The lighter curve is the ASD of the output of the second interferometer which measures the differential displacement of the two test-masses.

δ_{12}—the cross coupling of the first channel of the interferometer to the second channel.

C_{sus}—the controller for the x degree of freedom of the capacitive actuators acting on TM 2.

C_{df}—the controller that takes the measurement of the differential displacement of the two test-masses and commands forces to the SC via the micro-Newton thrusters.

The input noises to the model are: force noise on TM 1 due to the noise of the micro-Newton thrusters; internal forces between the two test-masses and the SC due to, for example, stray fluctuating magnetic fields within the SC; readout noise of the two interferometers.

The two outputs of the model correspond to the two interferometer outputs. These are: o_1, the measurement of the position of TM 1 relative to the SC and o_{12}, the measurement of the distance between the two test-masses. The aim of the MDC is to convert these two signals to equivalent out-of-loop acceleration signals, assuming full knowledge of all parameters of the model. Spectral density estimates of the two simulated output signals are shown in figure 8.

Further details of this MDC are contained in [16]. Here we just show the main results and the procedure used to do the calibration to acceleration.

The conversion of the interferometer outputs to acceleration is in principle simple, but technically challenging if one is to avoid numerical artefacts and inaccuracies arising from, for example numerical differentiation or the design of digital filters which are meant to represent continuous physical processes. The first step is to convert the two interferometer outputs to in-loop acceleration terms. Looking at figure 7, we see that this means taking the signal back through the interferometer, and back through the dynamical terms $(s^2 + \omega^2)^{-1}$. The
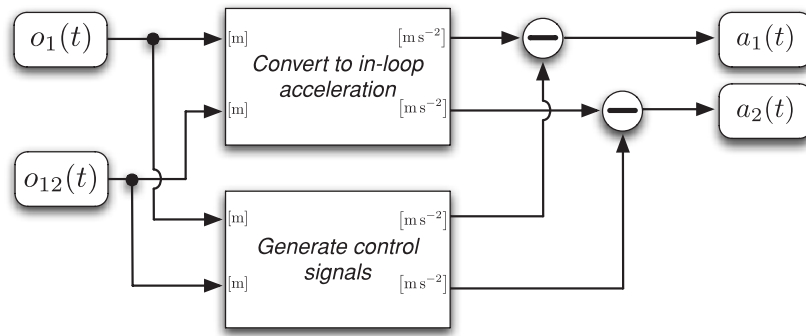
Figure 9. Schematic of the calibration signal processing pipeline.

difficult term here is the double differentiation of the signal to invert the dynamical terms. Various numerical differentiation and filtering techniques were explored during this MDC. Once we have the in-loop acceleration signals, we must account for the commanded forces applied by the two feedback loops. To do that, we must calculate the commanded forces, given the two interferometer outputs. This requires accurate knowledge of the controllers and any actuator delays present in the system. Once we have digital filter representations of these components we can compute the commanded force signals and subtract those from the in-loop acceleration signals. For the calibration of the differential interferometer output, we must also include the effect of the cross coupling from the first interferometer. Performing these steps yields the two out-of-loop acceleration signals $a_1(t)$ and $a_2(t)$. At different frequencies, these two acceleration signals are dominated by difference sources of force noise.

A schematic of the analysis pipeline is shown in figure 9. The detailed processing algorithms are composites of existing LTPDA functions. The result of applying this algorithm to the simulated interferometer outputs gives two timeseries AOs with spectral density estimates as shown in figure 10. Also indicated on the plot are the limiting noise sources at different Fourier frequencies. In the low-frequency region (below a few mHz), the acceleration term a_2 is limited by the differential acceleration noise on the two test-masses and at higher frequencies by sensing noise of the interferometer. In channel a_1, below a few hundred mHz we measure the noise of the thrusters. Again, at higher frequencies the signal is dominated by sensing noise.

4.2. The second mock data challenge

The second LTP MDC aims to develop algorithms suitable for parameter estimation. Roughly speaking, the model of LTP is the same as that used in MDC 1 and the output of the model is again the two interferometer outputs as described above. In this case, however, the analysis team is no longer given the values of some of the parameters used in the model (for example, the overall gains of the two longitudinal control servos and the stiffness of the coupling of the two test-masses to the SC), and hence cannot directly convert the interferometer outputs to out-of-loop acceleration. Instead, additional data sets are provided that include some injected signals from which the parameters should be estimated. The stimulus signals are injected at the input to the loop controllers. The aim of this MDC is to recover values for the parameters together with uncertainties on these parameters. Then,
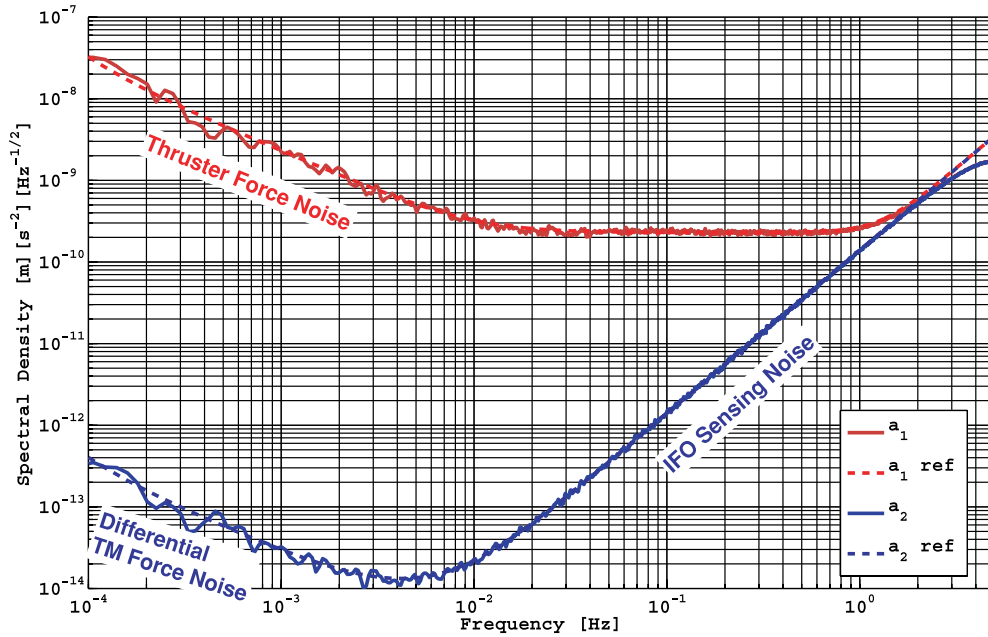
Figure 10. Simulated interferometer outputs calibrated to out-of-loop acceleration.

using these parameter values, reconstruct the output signal and subtract it from the data. This 'cleaned' data can then be calibrated as in MDC 1 to yield out-of-loop acceleration terms.

In order to accomplish this MDC, various new tools for signal extraction and parameter estimation need to be developed and implemented within the LTPDA framework.

5. Summary

This paper presents an overview of the ongoing effort to develop data analysis methods and tools for analysing data from the LTP experiment onboard the LISA Pathfinder mission. A robust and flexible analysis environment has been developed within MATLAB, with a large number of algorithms included to ensure that all the planned experiments of the mission can be analysed. The development of the analysis tools and algorithms goes hand-in-hand with the design of the experiments in the Experiment Master Plan (EMP). In addition, mock data challenges (MDCs) are carried out to test not only the analysis tools, but the analyses themselves. Over time, the MDCs will tend to more and more realistic scenarios, and in the end, should become a full test of the analyses in the EMP.

The first version of the LTPDA toolbox has been released and delivered to ESA, and successfully passed a large set of unit tests, system tests and acceptance tests. In the near future, the second version of the toolbox will be delivered, containing a larger number of algorithms which have arisen from MDCs and the planning of experiments, as well as from the use of the toolbox in analysing laboratory tests of the mission hardware. The third version is planned to be delivered in 2009. This version should contain essentially all tools needed to carry out the mission.

Acknowledgments

References

[1] Anza S *et al* 2005 The LTP experiment on the LISA Pathfinder mission *Class. Quantum Grav.* **22** S125–38
[2] McNamara P, Vitale S and Danzmann K 2008 LISA Pathfinder (on behalf of the LISA Pathfinder science working team) *Class. Quantum Grav.* **25** 114034
[3] Science requirements and top-level architecture definition for the Lisa Technology Package (LTP) on board LISA Pathfinder (SMART-2) *LTPA-UTN-ScRD-Iss003-Rev1*
[4] Heinzel G *et al* 2006 Interferometry for the LISA technology package LTP: an update *J. Phys.: Conf. Ser.* **32** 132–6
[5] Steier F *et al* 2009 The end-to-end testbed of the optical metrology system on-board LISA Pathfinder *Class. Quantum Grav.* **26** 094010
[6] Dolesi R *et al* 2003 Gravitational sensor for LISA and its technology demonstration mission *Class. Quantum Grav.* **20** S99–S108
[7] Armano M *et al* 2009 LISA Pathnder: the experiment and the route to LISA *Class. Quantum Grav.* **26** 094001
[8] Fichter Walter, Gath Peter, Vitale Stefano and Bortoluzzi Daniele 2005 LISA Pathfinder drag-free control and system implications *Class. Quantum Grav.* **22** S139–48
[9] Nofrarias M *et al* 2009 Thermal coupling within LTP dynamics control loop *J. Phys.: Conf. Ser.* **154** 012004
[10] Grynagier A *et al* 2009 The LISA Pathfinder drift mode *Class. Quantum Grav.* **26** 094007
[11] Graphviz—graph visualization software http://www.graphviz.org
[12] MATLAB http://www.mathworks.com
[13] LTPDA: a MATLAB toolbox for accountable and reproducible data analysis http://www.lisa.aei-hannover.de/ltpda
[14] Ferraioli L *et al* 2009 Derivative estimation for assessing low-frequency noise in LISA Pathfinder data reduction *Class. Quantum Grav.* **26** 094013
[15] MySQL http://www.mysql.org
[16] Monsky A *et al* 2009 The first mock data challenge for LTP *Class. Quantum Grav.* **26** 094004