

LISACode

version 2.0

**A scientific simulator for gravitational waves
space-based detector like LISA**

(Laser Interferometer Space Antenna)

User's guide

Antoine Petiteau

April 23, 2012

Contents

1	Introduction	4
2	A shot description of LISACode	5
3	Install	8
3.1	System requirements	8
3.2	For Unix, Linux, Mac OS X	8
3.2.1	Install	8
3.2.2	Install configuration	9
3.2.3	Running test	10
3.3	For Windows	11
3.3.1	Install	11
3.3.2	Test	12
4	Use LISACode	13
4.1	Running of simulator and the two annexe softwares	13
4.2	Running using <i>makeTDI-LISACode.py</i> (required lisatools)	14
5	Configuration file	15
5.1	Generalities	15
5.2	Configuration par fichier ASCII	18
5.2.1	Configuration des temps	19
5.2.2	Configuration de l'interpolation	20
5.2.3	Approximation des retards dans l'application de TDI	20
5.2.4	Configuration des orbites et du temps de parcours	21
5.2.5	Configuration du detecteur	22
5.2.6	Configuration des fichiers de sorties	23
5.2.7	Configuration des ondes gravitationnelles	24
5.2.8	Configuration des bruits	26
5.2.9	Configuration des USOs	28
5.2.10	Configuration de TDI	29
5.3	Configuration par fichier XML	30
6	Problèmes connus	31
6.1	Installation	31
6.1.1	Erreur lors de l'installation	31
6.2	Fichier de configuration	31
6.2.1	Problème dans la lecture du fichier de configuration :	31

7	Support - Contact	31
7.1	At AEI (Albert Einstein Institute)	31
7.2	At APC (AstroParticule et Cosmologie)	31

1 Introduction

This document provides a description of the LISACode software. LISACode is a simulator for LISA like space-based gravitational wave (GW) detectors. Its ambition is to achieve a new degree of sophistication allowing to map, as closely as possible, the impact of the different subsystems on the measurements. It is also a useful tool for generating realistic data including several kind of sources (Massive Black Hole binaries, EMRIs, cosmic string cusp, stochastic background, etc) and for preparing their analysis. It is fully integrated to the Mock LISA Data Challenge. LISACode is not a detailed simulator at the engineering level but rather a tool whose purpose is to bridge the gap between the basic principles of LISA and a future, sophisticated end-to-end simulator. It does not aim to simulate in detail the detector itself but rather it makes use of the response function of its main components, particularly in the way they will affect the noise level of the detector response. It also includes an implementation of the TDI (Time Delay Interferometry) technique which allows to suppress the noise level of the lasers.

The LISACode package contains several executables which corresponds to several tools.

In general, the main inputs and outputs of LISACode basic are time-dependent sequences that describes, for input, the GW strain and noises as a function of time and, for output, the phasemeters response as a function of time or/and their treatment via various TDI combination, in order to remove the laser noise level.

A number of elementary GW signals can be defined , but also it's possible to use the code in conjunction with more sophisticated GW simulators via intermediate data files.

The main outputs of the basic executable, `LISACode2`, are a number of time dependent data signals which will be produced either directly from the phasemeters or from a TDI treatment of these.

The others tools of LISACode package which are derived from the basic executable, are :

- sensitivity computation (`LC2Sensitivity`),
- error estimation for one source based on Fisher Matrix (`LC2FIM`),
- error estimation for a catalogue of sources based on Fisher Matrix (`LC2SNRFIMMulti`),
- simple Monte Carlo Markov Chain for exploring parameters of a particular source (`LC2MCMC`),
- direct power spectral density of TDI (`LC2PSD`),

- SNR (Signal to Noise Ratio) computation (**LC2SNR**),
- generation of polarization components of a GW h_+ and h_\times in Solar System Barycentric reference frame (**LC2GWbary**),
- generation of orbits data (**LC2Orbits**),
- others minor tools.

This document describes, in section 2, the structure of the code. Section 3 will explain how the code can be installed on a variety of platforms. Section 4 describes the use of LISACode. The section 5 describes the configuration file, a list of the main parameters that controls its operations and output.

More scientific information on LISACode are available in article of A.Petiteau [?].

2 A shot description of LISACode

This software is written in C++ for use the modularity of the object programming. The organisation of a LISACode simulation is shown in the figure 1. This organisation maps the main components of the LISA detector as well as its physical inputs.

There are 9 libraries :

- *Outils_Maths* : Objects use like tools the others modules : vector, filters, ...
- *Ondes_Gravit* : Model **gravitational waves (GW)**.
- *Orbitographie* : Model spacecraft orbits.
- *Bruits* : Model noises.
- *USO_Temps* : Model ultra-stable oscillators.
- *Memoire* : Manage memories for output data of each spacecraft.
- *Input_Data* : Read configuration of simulator.
- *Detecteur* : Model LISA detector : Arm response at gravitational waves, phasemeters, general running.
- *TDI* : Apply TDI generator.

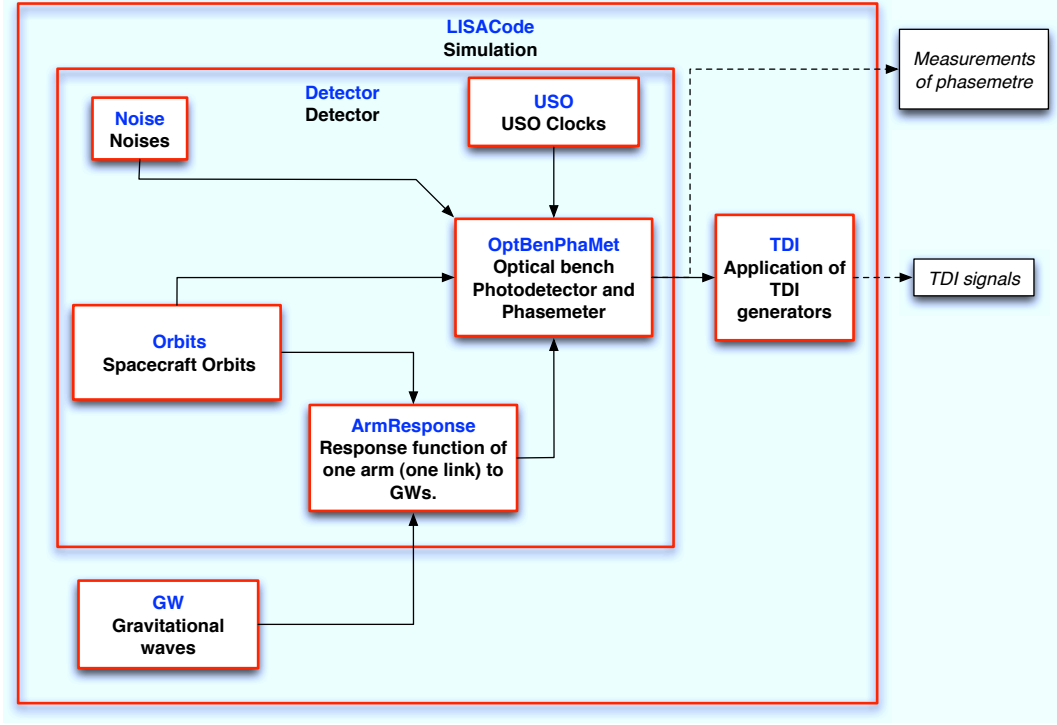


Figure 1: Organisation of a LISACode simulation. The red boxes represent the main modules. The generic name of the groups of classes corresponding to a module is in blue.

The figure 2 shows the organisation of the libraries used by LISACode.

The first input is the GW itself. It can be defined either internally through some models which produce signals from any monochromatic sources, binary system with fixed frequency or binary system computed in Post-Newtonian approximation (1 or 2.5 PN). The GW can also be inputted via a GW time sequence as may be produced by more sophisticated simulator codes.

The orbits of LISA are generated internally via the code. They correspond to realistic orbits that contain both the breathing and rotation modes of LISA as a function of its rotation around the sun. The parameters of these orbits can be adjusted in order to modify the average distance between the satellites (nominally $5 \times 10^9 m$) or be defined in such a way as to keep LISA at a fix and given location. The initial location of the orbits can also be defined at input.

An important ingredient of the LISA response and hence of the codes are the noise inputs of different nature. These include the optical noise due to shot noise and related factors as described in table 1. The inertial mass and the laser noises can also be defined

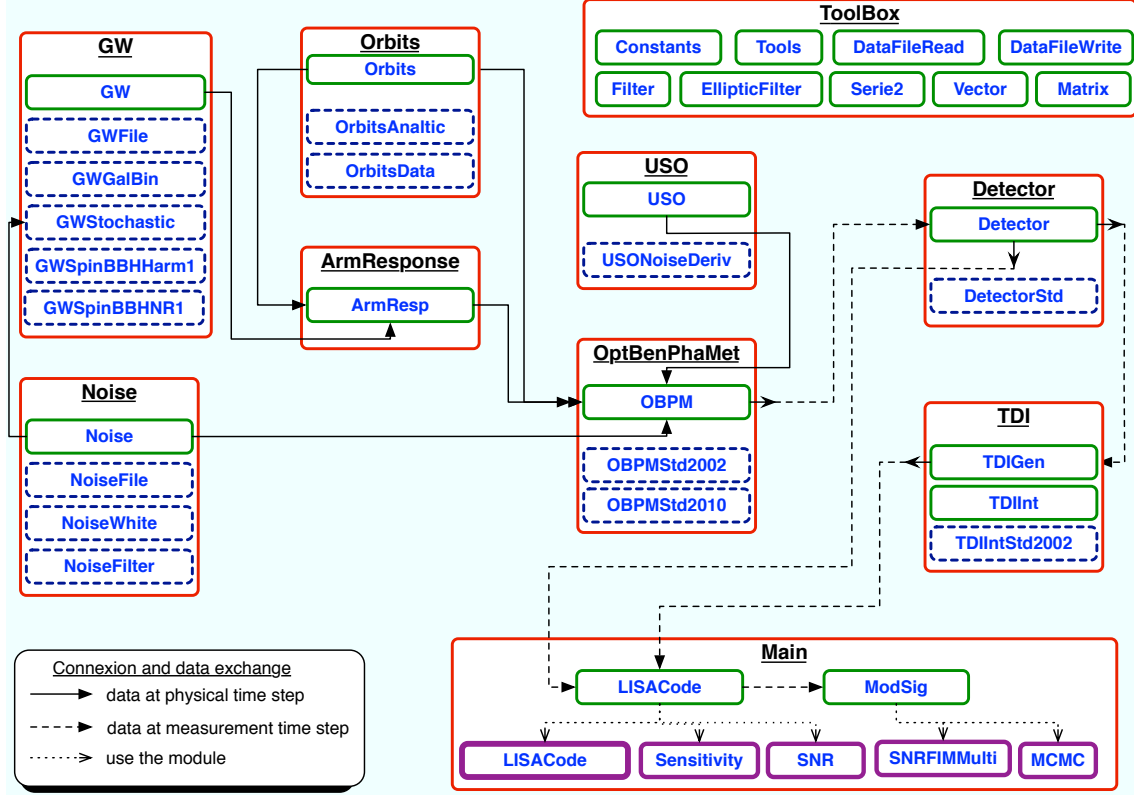


Figure 2: Organisation and dependency of simulator's libraries. The green boxes represent objects, the red boxes the libraries and the pink boxes the executables.

at input. Normally, these noise are defined as bandwidth limited white noise but different shapes of noise can be used.

Using the orbits, the response of LISA to the GW will be calculated and a relative frequency fluctuation (unit in $\frac{\Delta\nu}{\nu}$) will be inputted to the Phasemeter Module. These will be combined with the different noise contribution to produce the primary phasemeter output signal which then will be processed through a elliptic filtering module. This filter is low-pass filter which cuts the frequency at half of measurement frequency for cancel the spectrum aliased. In standard operation, the primary signal will be produced at a 0.5 Hz signal and outputted, after filtering, at a 1 Hz rate.

This signal can be saved on a disk file and/or processed by a TDI module using a variety of TDI combinations that are defined at input. A detailed description of TDI combination can be found in ref [?] and in ref [?].

The above description of the code provides only a brief summary of its capabilities.

Table 1: Error allocation budget based on Table 4.1 of [?]. The fourth column gives the default error included in LISACode. These values can be modified by the user. The frequency dependence of these noises is discussed in 5.2.8

Error source	Error	LISACode	LISACode input ($\frac{\delta\nu}{\nu}$ unit)
Measurement Noise			
Detector shot noise	$11 \times 10^{-12} \text{m.Hz}^{-\frac{1}{2}}$	11	$2.3 \times 10^{-19} \left(\frac{f}{1\text{Hz}} \right) \left(\frac{L}{5 \times 10^9 \text{m}} \right) \sqrt{\frac{1\text{W}}{P}} . \text{Hz}^{-\frac{1}{2}}$
USO	$5 \times 10^{-12} \text{m.Hz}^{-\frac{1}{2}}$	16.5	$3.49 \times 10^{-19} \left(\frac{f}{1\text{Hz}} \right) . \text{Hz}^{-\frac{1}{2}}$
Residual laser phase noise	$5 \times 10^{-12} \text{m.Hz}^{-\frac{1}{2}}$		
Laser beam-pointing instability	$10 \times 10^{-12} \text{m.Hz}^{-\frac{1}{2}}$		
Laser phase measurement and offset lock	$5 \times 10^{-12} \text{m.Hz}^{-\frac{1}{2}}$		
Scattered-light effects	$5 \times 10^{-12} \text{m.Hz}^{-\frac{1}{2}}$		
Other substantial effects	$8.5 \times 10^{-12} \text{m.Hz}^{-\frac{1}{2}}$		
Acceleration Noise			
Inertial Mass noise	$3 \times 10^{-15} \text{m.s}^{-2} . \text{Hz}^{-\frac{1}{2}}$	3	$1.59 \times 10^{-24} \left(\frac{1\text{Hz}}{f} \right) . \text{Hz}^{-\frac{1}{2}}$

3 Install

3.1 System requirements

This software can be installed on UNIX system which have a standard C and C++ compiler.

There is also a Windows executable (tested only for Windows XP).

3.2 For Unix, Linux, Mac OS X

3.2.1 Install

The installation is the same as a standard UNIX software. The following procedure describes the installation steps :

1. Download the package lisacode-1.4.4.tar.gz
(on the web : <http://www.apc.univ-paris7.fr/Downloads/lisa/LISACode/>)
and move the package in the directory (*MyDirectory* in the example) where you want to install LISACode.

2. Uncompress the package :

```
tar xvzf lisacode-1.4.4.tar.gz
```

3. Go in the *LISA_Sim_Tdi* directory :

```
cd MyDirectory/lisacode-1.4.4
```

4. Execute configuration script, *configure*, for make the *makefile* :

```
./configure
```

It's possible to add compilation options and the directory where the executable will be installed (see the part 3.2.2 or the file *INSTALL* for more technicalities on this two points).

5. Execute the *makefile* for compile the code :

```
make
```

6. If you want to install the simulator with your computer's executables or in specified directory, use the command :

```
make install
```

If the software has been installed correctly, there are 3 main executables in the directory *MyDirectory/lisacode-1.4.4/Main/Test*. The 3 main executables are :

- *LISACode* : The main simulator executable which models gravitational wave, processes them in LISA detector and apply TDI on the output data.
- *DnonGW* : Executable which gives relative frequency fluctuations on the beams made by gravitational waves.
- *TDIApply* : Apply TDI generators on phasemeters output data.

If you use *make install* for compile, the 3 executables are also in the directory *bin* located with your computer's executable or in specified directory *MyExe/bin*. There are also many test executable in this directory.

For more useful, copy the executables in your work directory (the directory where you want to make LISA simulation).

3.2.2 Install configuration

Many installation options are detailed in the file *INSTALL* located in *MyDirectory/lisacode-1.4.4*.

IMPORTANT : If the configuration phase failed with the following message :

```
configure: error: fftw3 is required but it's not find! Install or link it first!
```

you must include explicitly the directory which contains the **fftw3** as :

```
./configure CPPFLAGS=-I/pathfftw3/include LDFLAGS=-L/pathfftw3/lib
```

The compilation options should be specified with the configuration. After `./configure`, enter `CXXFLAGS=` followed by compilation options. For example, to optimise the compilation on PowerPCG5, the option is `-O3 -fast` therefore the configuration command is :

```
./configure CXXFLAGS="-O3 -fast -Wno-deprecated"
```

It also possible to specify the path of the directory where the executables will be installed, with the command `make install`. Enter `--prefix=` followed by the directory path. For example, the configuration command may be used on PowerPCG5 with executables in the directory *MyExe* is :

```
./configure CXXFLAGS="-O3 -fast -Wno-deprecated" --prefix="MyExe"
```

3.2.3 Running test

For test the running of LISACode, use configuration file (*ConfigRefBase*) with the following instructions :

1. Go in LISACode directory *MyDirectory/lisacode-1.4*.
2. Launch the simulator : Execute *LISACode* followed by the name of the configuration file *ConfigRefBase* then "123456" (seed of random generator).

If you use *make* for configuration, enter :

```
Main/bin/LISACode ConfigRefBase 123456
```

Else, if you use *make install* for configuration, enter :

```
MyExe/LISACode ConfigRefBase 123456
```

The software run during about 10 secondes or 3 minutes , for make 10 000 s of simulation. The final display is :

```
...
9787 s (remaining time : 00:00:00) #0098 %
X = 1.54307511554625e-18
Y = 2.23380376286915e-18
Z = 2.03070569638376e-18
X2s1 = 3.84102728265182e-18
X2s2 = 7.30253602341876e-18
```

```

X2s3 = 6.49295628725497e-18
P1 = 1.66863806379042e-18
Zeta1 = 9.83725319837678e-19
A = 1.13116170136341e-18
E = 2.24813370548819e-18
T = 8.43890200220657e-19
Alpha = 1.33468352729959e-18
AlphaId = 1.33468352729959e-18
Alphaf2 = 2.66936705459917e-18
Final time : 10001 s

```

```

Header for binary file SCi(NRec=5), Pos(NRec=10) or Delay(NRec=7) :
#TITLE SimDataTitle
#RECORD NRec 10181
#TIME -180 1 10000

```

```

Header for binary file TDI :
#TITLE TDIDataTitle
#RECORD 11 10001
#TIME 0.0 1 10000
Closing ASCII file.
Closing ASCII file.
Closing ASCII file.
Closing ASCII file.
Closing ASCII file.

```

End

3.3 For Windows

There is a LISACode executable for Windows which is tested on Windows XP. But it corresponds to an old version of LISACode.

3.3.1 Install

There isn't really install ; just download the package *LISACode_1-3_for_Windows.zip* at : <http://www.apc.univ-paris7.fr/Downloads/lisa/LISACode/version-1.3/>.
Uncompress the package and run LISACode.exe.

3.3.2 Test

It's possible to test the running of LISACode with the configuration file (*ConfigRefBase*) which is located in the same directory than the executable. Follow this instructions :

1. Open MS-DOS terminal and go to the executable where there are LISACode executable and run the configuration file *ConfigRefBase* (`cd ...`).
2. Run the simulator : *LISACode* followed by the name of the configuration file (*ConfigRefBase*) and "123456" (seed of random generator) .

`LISACode.exe ConfigRefBase 123456`

The result is the same than 3.2.3.

4 Use LISACode

4.1 Running of simulator and the two annexe softwares

The main software which make all the simulation (from gravitational waves to TDI) is called LISACode. For configure the simulation the software read informations in the configuration file. This configuration file may be two type : ASCII or XML. The technicalities of this configuration file is gave in the part ?? . The path of the output files is in the configuration file. For launch a simulation, enter LISACode executable path followed by the configuration file path :

```
./LISACode configuration_file
(if the LISACode executable is in the work directory)
ou
MyExe/LISACode configuration_file
(if the LISACode executable isn't in the work directory)
```

There are two others executables which also read information on configuration file. They make only one parts of the simulation :

- **DnonGW** : This program computes the relative frequency fluctuations on the laser beam which arrive on each phasemeter with only the gravitational response. There are 3 output files which have always the same name :
 - *DnGW_GW.txt* : Time evolution of gravitational waves' polarisation components h_+ and h_\times .
 - *DnGW_Position.txt* : Time evolution of spacecrafts' ecliptic coordinates.
 - *DnGW_TDelay.txt* : Time evolution of flightpath between spacecraft.
 - *DnGW_SigFctTrGW.txt* : Time evolution of relative frequency fluctuation for external laser beams which arrive on each spacecraft.

The time parameters are specified in configuration file.

- **TDIApply** : This program apply TDI generator specified in configuration file, on phasemeter raw data read in files specified in configuration file. The output contains time evolution for each TDI generator. The name of this ouput file is specified in configuration file.

4.2 Running using *makeTDI-LISACode.py* (required lisatools)

If you are familiar with lisatools package software (<http://code.google.com/p/lisatools/>), it is possible to use LISACode in the same way using *makeTDI-LISACode.py*.

- If necessary, install of lisatools package following instruction at : <http://code.google.com/p/lisatools/wiki/Install> .
- Make the XML files which describes the source of gravitational wave. This file must be in XML MLDC format. You can make them manually or using *makesource[...].py* softwares.
- If you want to use the MLDC waveform and not LISACode waveform, you have to make a file which contains the times series of h_+ and h_\times using . For example, for an EMRI which is not yet simulated by LISACode :
`makebarycentric.py EMRI.xml EMRI-barycentric.xml`
 It produces a new XML file associated with a binary file.
- Now, you can run LISACode using *makeTDI-LISACode.py* following by XML source files which can be file with source's parameters only or file pointing toward a binary data file. For example, if you want to include a Black Hole binary computed using LISACode waveform and an EMRI computed using MLDC waveform, use :
`makeTDI-LISACode.py BBH.xml EMRI-barycentric.xml`

Using options given by :

`makeTDI-LISACode.py -h`

you can specify the duration, the time step, the seed, noise or not, orbit (LISAModel), phasemeter output or not and output base file name :

- `-T DURATION, -duration=DURATION` : total time for TDI observables (s) [default 62914560 = 222 * 15]
- `-d TIMESTEP, -timeStep=TIMESTEP` : timestep for TDI observable sampling (s) [default 15]
- `-s SEED, -seed=SEED` : seed for random number generator (int); None time based seed [default None]
- `-N, -noNoise` do not include noise [default False]
- `-I LISAMODEL, -LISA=LISAMODEL` : LISA model: Static, Rigid, Eccentric [default Eccentric]
- `-r, -rawMeasurements` : output raw phase measurements (y's and z's) in addition to TDI X, Y, Z [default False]

- -R RANDOMIZENOISE, `-randomizeNoise=`RANDOMIZENOISE randomize level of noises [e.g., 0.2 for 20% randomization; defaults to 0]
- -l LASERNOISE, `-laserNoise=`LASERNOISE : laser noise level: None, Standard, [numerical value] [e.g., 0.2 for 20% of pm noise at 1 mHz; defaults to None]
- -o OUTPUTFILE, `-outputfile=`OUTPUTFILE : basename for the datasets [defaults to LC]

5 Configuration file

5.1 Generalities

This file contains all the information for simulator configuration. There two types of possible file : ASCII or XML. The XML file can be made with a graphical interface software : *LISA_AutoGUI.jar*. The tables 2 , 3, 4, 5, 6, 7, 8 , 9 and 10 lists the parameters.

Table 2: List of parameters used to configure a simulation whose type is **Time**.

Name	Explanation	Unit	Standard Value
StepPhysic	Time step use for model the continuous process, prior to filtering.	second	0.5
StepMeasure	Time step of output file of Phasemeter signal, after filtering.	second	1
Max	Total duration of calculation.	second	from second to years
DeltaTDIDelay	Possible error on the estimation of the delay.	second	from 0 to 1e-5
StepDisplay	Time step of evolution display (for follow the simulation).	second	from 100 to 1e6

Table 3: List of parameters used to configure a simulation whose type is **PrecisionTDI**.

Name	Explanation	Unit	Standard Value
Interpolation	Type of interpolation performed for TDI		LAG 20
TDIDelayApprox	If ON, delays use in TDI are added else they are imbricated	On/Off	Off

Table 4: List of parameters used to configure a simulation whose type is **Orbits**.

Name	Explanation	Unit	Standard Value
Armlength	Average armlength between LISA satellites	meters	5e9
StartTime	Start time which defines the LISA position on the orbits at beginning of calculation.	seconds (From 0 to year)	0
InitialRotation	Initial phase of LISA triangle configuration (0 : spacecraft 1 at bottom)	radians	0
Move	If On LISA spacecraft move else LISA spacecraft are fixed.	On/Off	On
Order	Order of flightpath calculation : 0 - time computed as from spacecraft positions ,1 - Sagnac effect , 2 - General Relativity effects.	0 / 1 / 2	2

Table 5: List of parameters used to configure a simulation whose type is **Detector**.

Name		Explanation	Unit	Standard Value
LaserPower		Defines the Laser power.	Watt	1
PhaMetFilter		Activates or de-activates Phasemeter Filter.	On/Off	On
PhaMetFilter-Parameters	attenuation	Filter attenuation	dB	180
PhaMetFilter-Parameters	oscillation	Filter oscillations in bandwidth	dB	0.1
PhaMetFilter-Parameters	FactFmes-ForHighFreq	Factor for high transition frequency (high transition frequency over measurement frequency)		0.1
PhaMetFilter-Parameters	FactFmes-ForLowFreq	Factor for low transition frequency (low transition frequency over measurement frequency)		0.3

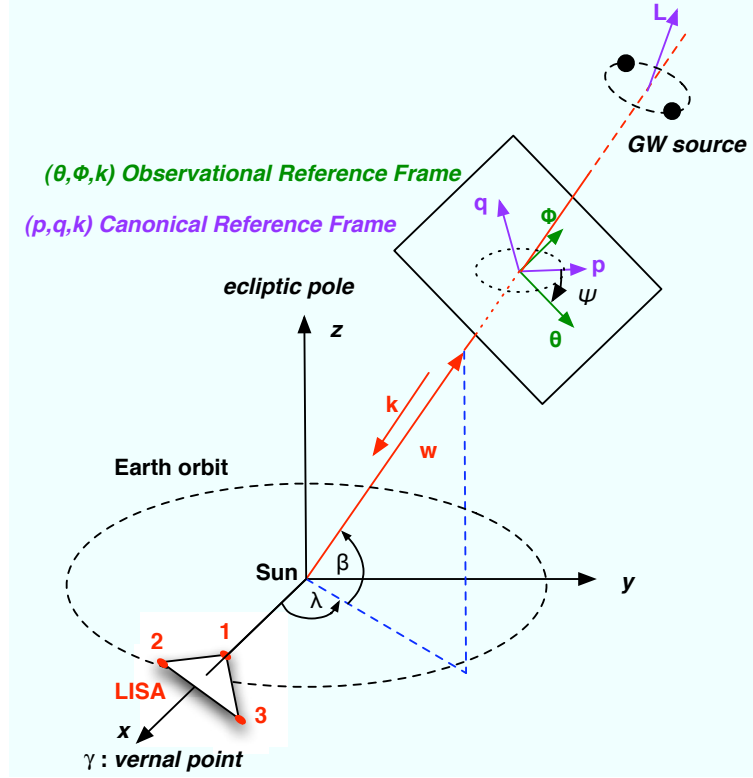


Figure 3: Schematic description of GW parameters and reference frames. The direction to the source is located by the declination β and the ecliptic longitude λ . $(\hat{\theta}, \hat{\phi}, \hat{n})$ is the observational reference frame, i.e. the frame constructed from the direction of the source. In this frame $\hat{\theta}$ is on the meridian. $(\hat{p}, \hat{q}, \hat{n})$ is the canonical reference frame. ψ is the polarization angle, i.e. the angle between the canonical reference frame and the observational reference frame.

5.2 Configuration par fichier ASCII

LISACode can read information of simulation in ASCII configuration file. The syntax of this file obeys certain very strict rules :

- All commands or comments end by a coma dot with a space before(“ ;”).
- Words are separated by space.
- The comments begin by “#” followed by a space.
- The first word of each command is the main key word. It gives information about the command contains. There are 9 main key words which are : Time, Interpolation, TDIDelayApprox , Orbits, Detector, Record, GW, Noise, USO et TDI. This main key word is followed by “:” .
- The command END stops the reading of configuration file.

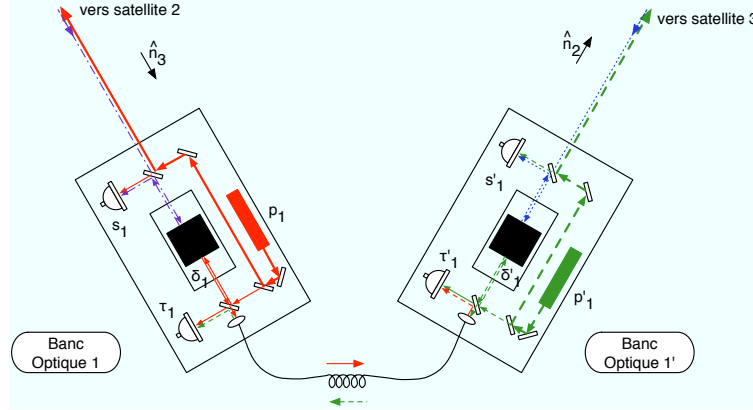


Figure 4: Schematic representation of the two optical benches of spacecraft 1. s_i and s'_i are the phasemeters which measure the interference between the beam from the distant spacecraft and the beam of the local optical bench, i.e. containing the phasemeter. τ_i and τ'_i are the phasemeters which measure the interference between the beam from the other optical bench in the same spacecraft and the beam of the optical bench which contains the phasemeter. p is the laser and δ is the inertial mass(form [?]).

5.2.1 Configuration des temps

The english version si not finished ! Le mot clé principal d'une ligne de configuration d'un paramètre temporel est **Time**. Une liste des parametres est donnee dans la tableau 2. Les différents paramètres temporels et les lignes les configurant sont :

- *Pas de temps physique* : C'est le pas de temps le plus petit de la simulation. Il correspond à la simulation des phénomènes continus.

Ligne de configuration pour un pas de temps physique de 0.5 s :

`Time : StepPhysic 0.5 ;`

- *Pas de temps sur la prise de mesure* : C'est le pas de temps sur la prise des mesures par les phasemètres. Il correspond au pas de temps sur les données de sortie et donc sur les resultats de TDI.

Ligne de configuration pour un pas de temps sur la prise de mesure de 1 s :

`Time : StepMeasure 1 ;`

- *Durée de la simulation* : C'est le temps que dure la simulation.

Ligne de configuration pour une simulation de 10000 s :

`Time : Max 10000 ;`

- *Imprécision de l'information sur les temps de parcours* : Cette incertitude temporelle correspond à l'imprécision sur la connaissance du temps de propagation le long des bras de LISA. C'est une erreur ΔD ajoutée sur les temps de parcours exacts $D_{real} = L_{real}/c$ avant leur utilisation par TDI. Les temps de parcours utilisés dans TDI D_{TDI} sont donc :

$$D_{TDI} = D_{real} + \Delta D \quad (1)$$

Ligne de configuration pour une imprécision sur les retards de 10^{-6} s:

Time : DeltaTDIDelay 1e-6 ;

- *Pas de temps sur l'affichage* : Pas de temps sur l'affichage à l'écran. Il permet à l'utilisateur de suivre l'évolution de la simulation.

Ligne de configuration pour un pas de temps d'affichage de 1000 s:

Time : StepDisplay 1000 ;

5.2.2 Configuration de l'interpolation

Le mot clé principal d'une ligne de configuration de l'interpolation est **Interpolation**. Il existe pour l'instant un seul type d'interpolation : l'interpolation lagrangienne

- *Interpolation dans l'application de TDI* : C'est le type d'interpolation faite sur les données brutes de sorte des phasemetres lors de l'application des retards dans TDI. On interpole pour obtenir la valeur la plus réaliste possible entre 2 mesures. Pour l'interpolation lagrangienne, on précise l'ordre de l'interpolation.

Ligne de configuration pour définir l'interpolation dans l'application de TDI :

Interpolation : LAG 20 ;

5.2.3 Approximation des retards dans l'application de TDI

Il est possible d'accélérer le calcul de TDI en effectuant un calcul moins exact des retards. En effet lorsque l'on calcul le retard global à appliquer sur une mesure, il faut combiner les retards, en interpolant la valeur de chaque retard à partir des précédents. Par exemple le calcul normal d'un "pack TDI" est :

$$D_1 D_2 D_3 s_2 = s_2 \left(t - \left(\frac{L_1(t)}{c} + \frac{L_2 \left(t - \frac{L_1(t)}{c} \right)}{c} + \frac{L_3 \left(t - \left(\frac{L_1(t)}{c} + \frac{L_2 \left(t - \frac{L_1(t)}{c} \right)}{c} \right)}{c} \right) \right) \right) \quad (2)$$

Lorsque l'on simplifie le calcul, les retards sont simplement pris à t et sommés sans interpolation. Soit pour l'exemple précédent :

$$D_1 D_2 D_3 s_2 = s_2 \left(t - \left(\frac{L_1(t)}{c} + \frac{L_2(t)}{c} + \frac{L_3(t)}{c} \right) \right) \quad (3)$$

L'accélération vient du fait que l'on a beaucoup moins d'interpolation à calculer.

- *Delay approximation* : La ligne de configuration pour effectuer un calcul approximatif rapide de TDI est :

TDIDelayApprox : off ;

5.2.4 Configuration des orbites et du temps de parcours

Le mot clé principal d'une ligne de configuration des orbites est **Orbits**. Une liste des paramètres est donnée dans la tableau 4.

- *Longueur des bras* : Spécifie la longueur nominale des bras de LISA.
Ligne de configuration pour définir une longueur de bras nominal de $5 \times 10^9 m$:

Orbits : Armlength 5e9 ;

- *Temps initial des orbites* : Ce paramètre temporel permet de démarrer la simulation avec une position des satellites différentes de la configuration de base. La configuration de base correspond à la position du tableau 11 soit le satellite 1 sur l'axe x et en dessous du plan de l'écliptique et satellite 2 et 3 au dessus avec 2 en $y < 0$ et 3 en $y > 0$.

Ligne de configuration pour définir un temps de démarrage des orbites de $5 \times 10^6 s$ après la configuration initiale :

Orbits : StartTime 5e6 ;

- *Phase initial de rotation du triangle* : Ce paramètre temporel permet de changer la position au temps 0 des satellites (temps courant + temps initial des orbites) par rapport à la configuration de base définie dans la tableau 11. Elle correspond à un angle (en radians) de rotation dans le plan par rapport à la configuration de base. Ligne de configuration pour définir un angle de rotation de $\pi/6$ par rapport à la configuration de base :

Orbits : InitialRotation 0.523598776 ;

- *Mouvement des satellites* : Ce paramètre permet de mettre en mouvement ou non les satellites de LISA. On met **Off** pour fixer les satellites et **On** pour les mettre en mouvement.

Ligne de configuration pour définir une simulation avec LISA mobile :

```
Orbits : Move On ;
```

- *Ordre dans le calcul des temps de propagation* : Ce paramètre spécifie la précision avec lequel est fait le calcul du temps de parcours des photons entre satellites. Le paramètre est a 0 pour un calcul classique considérant uniquement la distance entre satellite. Il est a1 pour un calcul tenant compte de l'effet Sagnac du a la rotation du triangle c'est-à-dire que le long d'un même bras les temps parcours différent. Il est 2 pour un calcul tenant compte des effets de la relativité générale. Si les satellites sont fixes (**Move Off**), on ne peut pas spécifier un autre ordre que 0.

Ligne de configuration pour définir un calcul relativiste des temps de propagation :

```
Orbits : Order 2 ;
```

5.2.5 Configuration du détecteur

Le mot clé principal d'une ligne de la configuration du détecteur est **Detector**. Cette configuration concerne des éléments spécifiques du détecteur comme la puissance laser ou le filtre du phasemètre. Une liste des paramètres est donnée dans la tableau 5.

- *Puissance Laser* : Spécifie la puissance des faisceaux à la sortie des lasers.

Ligne de configuration pour définir une puissance de laser de 1 *Watt* :

```
Detector : LaserPower 1 ;
```

- *Filtre du phasemètre* : Spécifie si on filtre ou non les signaux dans les phasemètres par un filtre passe-bas elliptique s'adaptant au pas de temps physique et au pas de temps de mesure. Ce filtre évite un repliement de spectre lors du sous-échantillonnage. On met **Off** pour rendre le filtre inactif et **On** pour le rendre actif.

Ligne de configuration pour filtrer les signaux dans le phasemètre :

```
Detector : PhaMetFilter On ;
```

- *Paramétrisation du filtre elliptique du phasemètre* : Il y a 4 paramètres pour définir le filtre du phasemètre : l'atténuation en décibels caractérisée par le mot-clé **attenuation**, l'oscillation en bande passante en décibels caractérisée par le mot-clé **oscillation**, la fréquence de coupure haute définie comme un facteur (inférieur à 1) de la fréquence de mesure caractérisée par le mot-clé **FactFmesForHighFreq** et la fréquence

de coupure basse definie comme un facteur (inferieur a 1) de la frequence de mesure caracterisee par le mot-cle **FactFmesForLowFreq**.

Ligne de configuration pour un filtre (filtre par default) d'attenuation de 180 dB , d'oscillation en bande passante de 0.1 dB, de frequence de coupure haute egal a 0.1 fois la frequence de mesure et de frequence de coupure basse egal a 0.3 fois la frequence de mesure :

```
Detector : PhaMetFilterParameters : attenuation 180 oscillation
0.1 FactFmesForHighFreq 0.1 FactFmesForLowFreq 0.3 ;
```

5.2.6 Configuration des fichiers de sorties

Le mot clé principal d'une ligne de configuration d'un fichier de sortie est **Record**. Il existe 4 types de fichiers de sortie. Les sorties pour lesquels aucun fichier n'est spécifié ne sont pas enregistrés mais cela n'empêche pas l'exécution du programme. Une liste des parametres est donnee dans la tableau 6.

- *Fichier de sortie des données d'un satellite* : Fichier dans lequel sont stockées les données en sortie des 4 phasemètres d'un satellite. Ce fichier comporte 5 colonnes (voir la figure 4 pour la position des phasemetres) : temps , phasemetre s_i , phasemetre s'_i , phasemetre τ_i , phasemetre τ'_i .

Ligne de configuration pour définir lenregistrement de la sortie du satellite 1 dans le fichier de nom "*SigPhaMetSC1.txt*" :

```
Record : SignalSC 1 SigPhaMetSC1.txt ;
```

- *Fichier d'enregistrement des temps de parcours* : Fichier dans lequel sont stockes les 6 temps de parcours entre satellites. Ce fichier comporte 7 colonnes : temps , 3 temps de parcours dans le sens direct (L_1/c L_2/c L_3/c) , 3 temps de parcours dans le sens indirect (L'_1/c L'_2/c L'_3/c).

Ligne de configuration pour définir l'enregistrement des temps de parcours dans le fichier de nom "*DelayTDI.txt*" :

```
Record : Delay DelayTDI.txt ;
```

- *Fichier d'enregistrement des positions des satellites* : Fichier dans lequel sont stockes les 3 coordonnées cartésiennes dans le referentiel ecliptique des satellites. Ce fichier comporte 10 colonnes : temps , x_1 , x_2 , x_3 , y_1 , y_2 , y_3 , z_1 , z_2 , z_3 .

Ligne de configuration pour définir l'enregistrement des positions dans le fichier de nom "*SCPos.txt*" :

```
Record : Position SCPos.txt ;
```


- *Fichier d'enregistrement des générateurs TDI* : Fichier dans lequel sont stockées les résultats des générateurs TDI spécifiés (voir 5.2.10). Ce fichier comporte une colonne pour le temps + une colonne par générateur TDI spécifié.

Ligne de configuration pour définir l'enregistrement générateurs TDI dans le fichier de nom : "SignalTDI.txt" :

Record : TDI SignalTDI.txt ;

5.2.7 Configuration des ondes gravitationnelles

Le mot clé principal d'une ligne de configuration d'une onde gravitationnelle est **GW**. Une liste des parametres est donnee dans la tableau 7. More scientific information on LISACode gravitational waves are available in article of A.Petiteau [?]. Il existe 4 types d'onde gravitationnelle possibles dans le simulateur. Chaque paramètre est défini, en précédant la valeur, d'un mot clé correspondant au paramètre. Pour tous les types d'onde, on précise juste après "GW", la direction de la source en coordonnées écliptiques ($\beta \in [-90^\circ, 90^\circ] \rightarrow \{\text{Bet}\}$, $\lambda \in [0^\circ, 360^\circ] \rightarrow \{\text{Lam}\}$) et l'angle de polarisation $\Psi \in [0^\circ, 360^\circ] \rightarrow \{\text{Psi}\}$. Ces 3 angles sont schematise sur la figure 3. Attention ils sont définis en degrés. Par exemple pour une onde dont la source est à $\beta = 27^\circ$ et $\lambda = 297^\circ$ et dont l'angle de polarisation $\psi = 229^\circ$, le début de la ligne de configuration est :

GW : Bet 27.0 , Lam 297.0 , Psi 229.0 : ... ;

- *Onde gravitationnelle monochromatique {Mono}* : Onde gravitationnelle monochromatique dont on définit la fréquence $\{f\}$, l'amplitude de chaque composante de polarisation ($h_{+0} \{\text{hp}\}$ et $h_{\times 0} \{\text{hc}\}$) et la phase initiale (en radians) de chaque composante de polarisation ($\Phi_{+0} \{\text{Phi0hp}\}$ et $\Phi_{\times 0} \{\text{Phi0hc}\}$). L'évolution temporelle des contraintes gravitationnelles de l'onde dans le referentiel propre de l'onde (CFR : Canonical Reference Frame) $h_{CFR+}(t)$ et $h_{CFR\times}(t)$ correspond a :

$$\begin{cases} h_{CFR+}(t) &= h_{0+} \sin(2\pi ft + \phi_{0+}) \\ h_{CFR\times}(t) &= h_{0\times} \sin(2\pi ft + \phi_{0\times}) \end{cases} \quad (4)$$

Ligne de configuration pour définir une onde gravitationnelle monochromatique dont la direction de la source est $\beta = 50^\circ$ et $\lambda = 230^\circ$, l'angle de polarisation $\psi = 15^\circ$, la fréquence $f = 10^{-4}$, l'amplitude des composantes $h_{+0} = 10^{-21}$ et $h_{\times 0} = 0$ et la phase initiale des composantes de polarisation $\Phi_{+0} = 0$ et $\Phi_{\times 0} = 0$:

GW : Bet 50 , Lam 230 , Psi 15 : Mono : f 1e-4 , hp 1e-21 , hc 0.0 , Phi0hp 0.0 , Phi0hc 0.0 ;

- *Onde gravitationnelle d'une binaire de frquence fixe {Binary}* : Onde gravitationnelle issue d'une source binaire de frquence fixe pour laquelle on définit les masses

des deux astres m_1 {M1} et m_2 {M2} en masse solaire, la fréquence orbitale f_{orb} {forb} en Hz, l'angle d'inclinaison i {inc} en degrés, la phase initiale ϕ_0 {Phi0} entre 0 et 2π et la distance entre la source et le détecteur r {r} en kpc. L'évolution temporelle des contraintes gravitationnelles de l'onde dans le référentiel propre de l'onde (CFR : Canonical Reference Frame) $h_{CFR+}(t)$ et $h_{CFR\times}(t)$ correspond à :

$$\begin{cases} h_{CFR+} &= A (1 + \cos^2 i) \cos(4\pi f_{orb}t + \phi_0) \\ h_{CFR\times} &= -2A \cos i \sin(4\pi f_{orb}t + \phi_0) \end{cases} \quad (5)$$

avec

$$m_{tot} = m_1 + m_2 \quad (6)$$

$$R = \left(\frac{Gm_{tot}}{(2\pi f_{orb})^2} \right)^{1/3} \quad (7)$$

$$A = \frac{2G^2 m_1 m_2}{c^4 Rr} \quad (8)$$

Ligne de configuration pour définir une onde gravitationnelle issue d'une binaire de direction $\beta = 37.34^\circ$, $\lambda = 350^\circ$, l'angle de polarisation $\psi = 141.06^\circ$, de masses $M1 = 0.5M_S$ et $M2 = 0.033M_S$, de fréquence $f = 9.72044 \times 10^{-4}$, d'inclinaison $i = 88^\circ$, de phase initiale $\psi_0 = 0^\circ$ et de séparation $r = 0.1$ kpc :

GW : Bet 37.34 , Lam 350 , Psi 141.06 : Binary : M1 0.5 , M2 0.033 , forb 9.72044e-4 , inc 88 , phi0 0 , r 0.1 ;

- *Onde gravitationnelle d'une binaire de fréquence fixe* {PostNewtonBinary} : Onde gravitationnelle issue d'une source binaire calculée dans l'approximation Post-Newtonienne à 1 PN ou 2.5 PN pour laquelle on définit les masses des deux astres m_1 {M1} et m_2 {M2} en masse solaire, le temps de coalescence t_{coal} {tcoal} en seconds, l'angle d'inclinaison i {inc} en degrés, la phase initiale ϕ_0 {Phi0} entre 0 et 2π et la distance entre la source et le détecteur r {r} en kpc. L'ordre Post-Newtonien du calcul est de 1 PN lorsque le paramètre {type} est à 1 ou de 2.5 PN lorsque le paramètre {type} est à 2. Pour le calcul à 2.5 PN, il faut préciser 3 paramètres qui sont : une phase initiale arbitraire {omega0}, la phase à l'entrée du détecteur {taud0} et un paramètre Post-Newtonien {gw} égal à 1 dans la plupart des cas. Le calcul de l'évolution temporelle des contraintes gravitationnelles de l'onde dans le référentiel propre de l'onde (CFR : Canonical Reference Frame) $h_{CFR+}(t)$ et $h_{CFR\times}(t)$ est basé sur l'article [? ? ? ?].

Ligne de configuration pour définir une onde gravitationnelle issue d'une binaire de direction $\beta = 37.34^\circ$, $\lambda = 350^\circ$, l'angle de polarisation $\psi = 141.06^\circ$, de masses $M1 = 0.5M_S$ et $M2 = 0.033M_S$, de fréquence $f = 9.72044 \times 10^{-4}$, d'inclinaison $i = 88^\circ$, de phase initiale $\psi_0 = 0^\circ$ et de séparation $r = 0.1$ kpc :

GW : Bet 37.34 , Lam 352.0 , Psi 141.06 : PostNewtonBinary : M1
1.0e6 , M2 1.0e6 , tcoal 9676800.0 , inc 90 , phase 1.2 , r 1e5 , type
2 , omega0 1.0 , taud0 10.0 , gw 1.0 ;

- *Onde gravitationnelle quelconque (lecture d'un fichier) {File}* : Onde gravitationnelle quelconque dont l'évolution temporelle des composantes de polarisation est lue dans un fichier (3 colonnes : $temps$ h_+ h_\times).
Ligne de configuration pour définir une onde gravitationnelle lue dans le fichier *GWFile.txt* dont la direction de la source est $\beta = 50^\circ$ et $\lambda = 230^\circ$ et l'angle de polarisation $\psi = 15^\circ$.

GW : Bet 50 , Lam 230 , Psi 15 : File : GWFile.txt ;

5.2.8 Configuration des bruits

Le mot clé principal d'une ligne de configuration des bruits est **Noise**. Une liste des parametres est donnee dans la tableau 8. La configuration d'un bruit se fait ensuite en deux étapes : la localisation du bruit puis sa nature. La localisation du bruit se fait par le mot clé correspondant à la situation du bruit : **Laser** pour le bruit issu d'un laser - **Mass** pour le bruit sur une masse inertielle - **Shot** pour le bruit de "shot noise" - **OOPN** pour le bruit des autres bruits de chemin optique (autre que le "shot noise"). Le mot clé de localisation est suivi du numéro du satellite (1, 2 et 3) puis le "sens" du banc : 0 si le banc est dans le "sens direct " ($1 \rightarrow 3 \rightarrow 2 \rightarrow 1$), 1 s'il est dans le "sens indirect" ($1 \rightarrow 2 \rightarrow 3 \rightarrow 1$). On spécifie ensuite la nature du bruit :

- *Bruit blanc {White}*: Modélisation d'un bruit blanc de densité spectrale de puissance donnée.

Ligne de configuration pour définir un bruit blanc de densité spectrale de puissance 10^{-13} (sans dimension car exprimé en variation relative de fréquence) sur le laser du banc optique du satellite 1 en vis-à-vis avec le satellite 2 :

Noise : Laser 1 0 : White 1.0e-13 ;

- *Bruit filtré pour une évolution en $1/f$ de l'amplitude (ou \sqrt{PSD}) {Filter_1of}*: Modélisation d'un bruit dont l'amplitude évolue en $1/f$. La seule valeur à spécifier est le niveau du bruit $A_{\delta\nu/\nu}$ en $Hz^{1/2}$ c'est-à-dire l'amplitude (ou \sqrt{PSD}) tel que $PSD_{\delta\nu/\nu} = A_{\delta\nu/\nu}^2 f^{-2}$. Les coefficients sont ensuite automatiquement calculés par rapport au pas de temps¹.

Par exemple, le bruit d'une masse inertielle est generalement exprime comme un

¹Les coefficients du filtre pour un bruit en $PSD_{\delta\nu/\nu} = A_{\delta\nu/\nu}^2 f^{-2}$ sont : $\alpha_1 = 1$ et $\beta_0 = \beta_1 = A_{\delta\nu/\nu} \pi \Delta t$

bruit d'accélération en $m.s^{-2}.Hz^{-1/2}$: $\sqrt{S_{\Delta a, MI}} = 3 \times 10^{-15} m.s^{-2}.Hz^{-1/2}$. Le calcul du $A_{\delta\nu/\nu, MI}$ est alors :

$$A_{\delta\nu/\nu, MI} = \frac{1}{2\pi c} \sqrt{S_{\Delta a, MI}} = 1.59 \times 10^{-24} Hz^{1/2} \quad (9)$$

Ligne de configuration pour définir un bruit de densité spectrale de puissance $PSD_{\delta\nu/\nu} = (1.59 \times 10^{-24} Hz^{1/2})^2 f^{-2}$ sur les masses inertielles du banc optique du satellite 2 en vis-à-vis avec le satellite 1 :

Noise : Mass 2 1 : Filter_1of 1.59e-24 ;

- *Bruit filtré pour une évolution en f de l'amplitude (ou \sqrt{PSD})* {Filter_f}: Modélisation d'un bruit dont l'amplitude évolue en f . La seule valeur à spécifier est le niveau du bruit $A_{\delta\nu/\nu}$ en $Hz^{-3/2}$ c'est-à-dire l'amplitude (ou \sqrt{PSD}) tel que $PSD_{\delta\nu/\nu} = A_{\delta\nu/\nu}^2 f^2$. Les coefficients sont ensuite automatiquement calculés par rapport au pas de temps². Par exemple, le bruit de chemin optique dans les bancs est généralement exprime comme un bruit en $m.Hz^{-1/2}$: $\sqrt{S_{\Delta a, OOPN}} = 16.7 \times 10^{-12} m.Hz^{-1/2}$. Le calcul du $A_{\delta\nu/\nu, OOPN}$ est alors :

$$A_{\delta\nu/\nu, OOPN} = \frac{2\pi}{c} \sqrt{S_{\Delta a, OOPN}} = 3.49 \times 10^{-19} Hz^{-3/2} \quad (10)$$

Ligne de configuration pour définir un bruit de densité spectrale de puissance $PSD_{\delta\nu/\nu} = (3.49 \times 10^{-19} Hz^{-3/2})^2 f^2$ sur les bruits de chemin optique du banc optique satellite 3 en vis-à-vis avec le satellite 2 :

Noise : OOPN 3 1 : Filter_f 3.49e-19 ;

- *Bruit filtré pour une évolution en f de l'amplitude (ou \sqrt{PSD}), proportionnel à la longueur des bras et inversement proportionnel à la puissance laser* {Filter_fLoSP}: Modélisation d'un bruit dont l'amplitude évolue en f et est proportionnelle à la longueur des bras et inversement proportionnelle à la puissance laser. La seule valeur à spécifier est le niveau du bruit $A_{\delta\nu/\nu}$ en $Hz^{-3/2}$ c'est-à-dire l'amplitude (ou \sqrt{PSD}) tel que $PSD_{\delta\nu/\nu} = A_{\delta\nu/\nu}^2 f^2 \left(\frac{L}{5 \times 10^9 m}\right)^2 \left(\frac{1 W}{P}\right)$. Les coefficients sont ensuite automatiquement calculés par rapport au pas de temps³.

²Les coefficients du filtre pour un bruit en $PSD_{\delta\nu/\nu} = A_{\delta\nu/\nu}^2 f^2$ sont : $\alpha_1 = -1$, $\beta_0 = \frac{A_{\delta\nu/\nu}}{\pi \Delta t}$ et $\beta_1 = -\frac{A_{\delta\nu/\nu}}{\pi \Delta t}$

³Les coefficients du filtre pour un bruit en $PSD_{\delta\nu/\nu} = A_{\delta\nu/\nu}^2 f^2 \left(\frac{L}{5 \times 10^9 m}\right)^2 \left(\frac{1 W}{P}\right)$ sont : $\alpha_1 = -1$, $\beta_0 = \frac{A_{\delta\nu/\nu}}{\pi \Delta t} \left(\frac{L}{5 \times 10^9 m}\right) \sqrt{\left(\frac{1 W}{P}\right)}$ et $\beta_1 = -\frac{A_{\delta\nu/\nu}}{\pi \Delta t} \left(\frac{L}{5 \times 10^9 m}\right) \sqrt{\left(\frac{1 W}{P}\right)}$

Ligne de configuration pour définir un bruit de densité spectrale de puissance $PSD_{\delta\nu/\nu} = (2.30 \times 10^{-19} Hz^{-3/2})^2 f^2 (\frac{L}{5 \times 10^9 m})^2 (\frac{1W}{P})$ sur le "shot noise" du banc optique satellite 3 en vis-à-vis avec le satellite 2 :

Noise : Shot 3 1 : Filter_fLosP 2.30e-19 ;

- *Bruit filtré quelconque*: Modélisation d'un bruit filtré quelconque défini par les coefficients récurrents α {alpha} et directs β {beta} du filtre à réponse impulsionnelle infinie (IIR). On peut également préciser le temps nécessaire la stabilisation du filtre {stablization}. S'il y a plusieurs coefficients d'un même type, il faut les séparer par des virgules.

Ligne de configuration pour définir un bruit filtré dont les coefficients du filtre sont $\alpha = \{1\}$ et $\beta = \{5 \times 10^{-24}, 5 \times 10^{-24}\}$ (bruit d'accélération en f^{-2} sur la densité spectrale de puissance) sur la masse inertielle du banc optique du satellite 2 en vis-à-vis avec le satellite 1 :

Noise : Mass 2 1 : FilterCoef : alpha 1.0 beta 5e-24 , 5e-24 ;

Ligne de configuration pour définir un bruit filtré dont les coefficients du filtre sont $\alpha = \{3.9939806575, -5.9819996369, 3.9820571154, -0.9940381365\}$ et $\beta = \{0.0000510993, -0.0002040002, 0.0003058023, -0.000204002, 0.0000510993\}$ sur la masse inertielle du banc optique du banc optique du satellite 3 en vis-à-vis avec le satellite 2 (attention ce bruit n'a strictement aucune réalité physique et est juste présenté à titre d'exemple de syntaxe):

Noise : Bench 3 1 : FilterCoef : alpha 3.9939806575 , -5.9819996369 , 3.9820571154 , -0.9940381365 beta 0.0000510993 , -0.0002040002 , 0.0003058023 , -0.000204002 , 0.0000510993 stablization 10000 ;

- *Bruit lu dans un fichier*: Modélisation d'un bruit à partir des valeurs d'un fichier (2 colonnes : temps bruit). Le programme charge l'ensemble des valeurs est commence à lire les valeurs partir d'un point quelconque du fichier.

Ligne de configuration pour définir un bruit lu dans le fichier *NoiseOB.txt* et attribué au banc optique du satellite 1 en vis-à-vis avec le satellite 2 :

Noise : Laser 1 0 : File NoiseOB.txt ;

5.2.9 Configuration des USOs

Le mot clé principal d'une ligne de configuration d'un oscillateur ultra-stable est **USO**. Une liste des parametres est donnee dans la tableau 8. Il faut tout d'abord définir le satellite auquel l'USO appartient : pour cela on précède le numéro du satellite du mot clé SC.

On peut ensuite définir 3 paramètres pour un oscillateur ultra-stable (USO) : un offset [offset] en seconde, une dérive dans le temps en seconde par seconde [derivs] et le σ d'un bruit blanc gaussien [noise] en seconde. Ces paramètres sont spécifiés en précédant la valeur par le mot clé correspondant. Si l'on ne définit pas un USO, celui-ci est supposé parfait.

Ligne de configuration pour définir l'oscillateur du satellite 2 avec un offset de 0.006 s, une dérive de $10^{-6} s.s^{-1}$ et un bruit de sigma $10^{-7} s$:

```
USO : SC 2 : offset 0.006 derivs 1e-6 noise 1e-7 ;
```

5.2.10 Configuration de TDI

Le mot clé principal d'une ligne de configuration d'un générateur TDI est TDI. Le mot clé est suivi du nom du générateur puis de sa description. Le générateur est décomposé en un ensemble de "packs" écrits les uns à la suite des autres et séparés par un espace. Un pack correspond à l'application de retards sur la mesure d'un phasemètre ; par exemple $-D_{2'}D_2D_3D_{3'}D_3D_{3'}s_{1'}$ est un pack qui se code par -5236364. Le dernier chiffre correspond à la mesure du phasemètre code selon le tableau 12 . Les autres chiffres aux retards que l'on applique sur la mesure. La correspondance entre un bras et un chiffre du codage est donné par le tableau 13

Ligne de configuration pour définir le générateur TDI Michelson de deuxième génération

```
Z2:      TDI : Z2 3 , 24 , 256 , 2542 , 25416 , 254142 , 2541413 , 25414124
, -6 , -42 , -413 , -4124 , -41253 , -412524 , -4125256 , -41252542 ;
```

Il existe également un certain nombre de generateur dont le pack est predefini dans LISACode. Pour ces generateurs listes dans le tableau 10, le nom du generateur suffi.

Ligne de configuration pour définir le générateur TDI Michelson de deuxième génération

```
Z2:      TDI : X ;
```

5.3 Configuration par fichier XML

LISACode can also be configured by an XML file. The structure of this XML file is the same than its of the Mock LISA Data Challenge (cf. articles [? ? ?] and web site [?].

6 Problèmes connus

6.1 Installation

6.1.1 Erreur lors de l'installation

- Vérifier que le système possède bien une version d'automake 1.8 ou plus récente car dans certaine installtion, il peut être nécessaire d'utiliser automake.

6.2 Fichier de configuration

6.2.1 Problème dans la lecture du fichier de configuration :

- Vérifier qu'il y ait bien un point virgule à la fin de chaque ligne de commande et de commentaire.
- Vérifier qu'il n'y ait pas de ligne vide à la fin de votre fichier de configuration. La dernière ligne doit comporter une commande ou un commentaire. Elle peut également comporter l'instruction END.

7 Support - Contact

7.1 At AEI (Albert Einstein Institute)

Antoine PETITEAU : Antoine.Petiteau[at]aei.mpg.de - +49 (331) 567-7134

7.2 At APC (AstroParticule et Cosmologie)

Eric PLAGNOL : plagnol[at]apc.univ-paris7.fr Gerard AUGER : auger[at]apc.univ-paris7.fr

Table 6: List of parameters used to configure a simulation whose type is **Records**.

Name	Explanation	Standard Value
SignalSC 1	Defines output file name which contains spacecraft 1 phasemeters' raw data.	SC1.txt
SignalSC 2	Defines output file name which contains spacecraft 2 phasemeters' raw data.	SC2.txt
SignalSC 3	Defines output file name which contains spacecraft 3 phasemeters' raw data.	SC3.txt
Delay	Defines output file name which contains 6 flightpath data in seconds.	Delay.txt
Position	Defines output file name which contains ecliptic coordinates of the 3 spacecrafts in meters.	Delay.txt
TDI	Defines output file name which contains data of specified TDI generators.	TDI.txt

Table 7: List of parameters used to configure a simulation whose type is **GW** (see figure 3) **Mono** defines a monochromatic wave, **Binary** defines a binary system with fixed frequency and **PostNewtonBinary** defines a binary system which is computed in Post-Newtonian approximation.

GW Type	Name	Explanation	Unit	Standard Value
All	Bet	Ecliptic latitude (declination) of source direction (from Sun to source)	degrees	from -90 to 90
All	Lam	Ecliptic longitude of source direction (from Sun to source)	degrees	from 0 to 360
All	Psi	Polarisation of the source : angle between the observational reference frame and the canonical reference frame	degrees	from 0 to 360
Mono	f	Frequency of the source	Hertz	
Mono	hp	Amplitude of + component.	none	1e-21
Mono	hc	Amplitude of \times component.	none	1e-21
Mono	Phi0hp	Initial phase of + component.	radians	0 to 2π
Mono	Phi0hc	Initial phase of \times component.	radians	0 to 2π
Binary	M1	Mass of first object	solar mass	
Binary	M2	Mass of second object	solar mass	
Binary	forb	Orbital frequency of binary system.	Hertz	
Binary	inc	Inclination : angle between angular momentum and source's direction	degrees	from -90 to 90
Binary	phi0	Initial phase of binary system.	radians	0 to 2π
Binary	r	Distance between the observator and the source.	KiloParsec	
PostNewtonBinary	M1	Mass of first object	solar mass	
PostNewtonBinary	M2	Mass of second object	solar mass	
PostNewtonBinary	tcoal	Coalescence time (start at 0 at the beginning of the simulation).	seconds	
PostNewtonBinary	inc	Inclination : angle between angular momentum and source's direction	degrees	from -90 to 90
PostNewtonBinary	phase	Initial phase of binary system.	radians	0 to 2π
PostNewtonBinary	r	Distance between the observator and the source.	KiloParsec	
PostNewtonBinary	type	Type of Post-Newtonian calculation : 1 for 1 PN and 2 for 2.5 PN.	1 / 2	2
PostNewtonBinary	omega0	Arbitrary Post-Newtonian phase.	degrees	1
PostNewtonBinary	taud0	Post-Newtonian phase on the detector.		10
PostNewtonBinary	gw	Post-Newtonian factor.		1

Table 8: List of parameters used to configure a simulation whose type is **Noise** (see figure 4 for localization).

Parameter type	Name	Explanation	Unit	Standard Value
Localization	Laser i j	Noise of the laser located in optical bench of the spacecraft i pointing towards the spacecraft i+1 if j = 0 and towards the spacecraft i-1 if j = 1.		
Localization	Mass i j	Noise of the inertial mass located in optical bench i j (i.e. laser i j optical bench).		
Localization	Shot i j	Shot noise of the phasemeters which measure the interference between the beam from the distant spacecraft and the beam of the local optical bench i j (i.e. laser i j optical bench).		
Localization	OOPN i j	Others optical path noise located in optical bench i j (i.e. laser i j optical bench).		
Type	White	White noise at specified level.	$\frac{\Delta\nu}{\nu}.Hz^{-1/2}$ unit	1e-13
Type	Filter_1of	Filter noise corresponds to function 1/f at specified level.	$\frac{\Delta\nu}{\nu}.Hz^{-1/2}$ unit	1.59e-24
Type	Filter_f	Filter noise corresponds to function f at specified level.	$\frac{\Delta\nu}{\nu}.Hz^{-1/2}$ unit	3.49e-19
Type	Filter_fLosP	Filter noise corresponds to function f at specified level proportional to the armlength and inversely proportional to the root square of power.	$\frac{\Delta\nu}{\nu}.Hz^{-1/2}$ unit	2.30e-19
Type	FilterCoef - alpha - beta - stabiliza- tion	Filter noise where the filter coefficient are specified explicitly : Recursive coefficient of filter Direct coefficient of filter Number of data for filter stabilisation	steps	10000

Table 9: List of parameters used to configure a simulation whose type is **USO**. There is one USO by spacecraft specified by **SC** followed by the spacecraft number

Name	Explanation	Unit	Standard Value
offset	Offset of USO compared to current time	seconds	0.006
derivs	Derive of USO compared to current time in second by seconds	seconds	1e-7
noise	Gaussian noise of USO compared to current time with specified σ	seconds	1e-7

Table 10: List of parameters used to configure a simulation whose type is **TDI**. (**SC** mean spacecraft)

Name	Explanation	Pack Value (in LISACode)
<i>Any name</i>	Any TDI generator	to be define after the name
X or Xf or X1s1	Michelson first generation centred on SC1	1 , 35 , 364 , 3653 , -4 , -53 , -521 , -5235
Y or Yf or X1s2	Michelson first generation centred on SC2	2 , 16 , 145 , 1461 , -5 , -61 , -632 , -6316
Z or Zf or X1s3	Michelson first generation centred on SC3	3 , 24 , 256 , 2542 , -6 , -42 , -413 , -4124
X2 or Xf2 or X2s1	Michelson second generation centred on SC1	1 , 35 , 364 , 3653 , 36524 , 365253 , 3652521 , 36525235 , -4 , -53 , -521 , -5235 , -52361 , -523635 , -5236364 , -52363653
Y2 or Yf2 or X2s2	Michelson second generation centred on SC2	2 , 16 , 145 , 1461 , 14635 , 146361 , 1463632 , 14636316 , -5 , -61 , -632 , -6316 , -63142 , -631416 , -6314145 , -63141461
Z2 or Zf2 or X2s3	Michelson second generation centred on SC3	3 , 24 , 256 , 2542 , 25416 , 254142 , 2541413 , 25414124 , -6 , -42 , -413 , -4124 , -41253 , -412524 , -4125256 , -41252542
Alpha or alpha	Basic generator first generation α	-1 , -32 , -133 , 4 , 455 , 56
Beta or beta	Basic generator first generation β	-121 , -2 , -13 , 64 , 5 , 566
Gamma or gamma	Basic generator first generation γ	-21 , -232 , -3 , 464 , 45 , 6
P1	Beacon generator with no beams received by SC1	25 , -63 , -22 , 66 , 642 , -216 , 1463 , -1425
Q1	Beacon generator with no beams received by SC2	36 , -41 , -33 , 44 , 453 , -324 , 2541 , -2536
R1	Beacon generator with no beams received by SC3	14 , -52 , -11 , 55 , 561 , -135 , 3652 , -3614
E1	Monitor generator with no beams emitted by SC1	542 , 56 , -316 , -32 , -144 , 141 , 4 , -1
F1	Monitor generator with no beams emitted by SC2	653 , 64 , -124 , -13 , -255 , 252 , 5 , -2
G1	Monitor generator with no beams emitted by SC3	461 , 45 , -235 , -21 , -366 , 363 , 6 , -3
U1	Relay generator with no beams from SC3 to SC1 and from SC1 to SC2	145 , 1464 , -5 , -64 , 16 , 2 , -6542 , -656
V1	Relay generator with no beams from SC1 to SC2 and from SC2 to SC3	256 , 2545 , -6 , -45 , 24 , 3 , -4653 , -464
W1	Relay generator with no beams from SC2 to SC3 and from SC3	364 , 3656 , -4 , -56 , 35 , 1 , -5461 , -545

Table 11: Orbits basic configuration : Position, X, Y, Z (in meters) of the three spacecraft at different times (in seconds) for a nominal armlength of $5 \cdot 10^6$ km.

Time of year (sec)	0	7865000	15730000	23595000
X_1	148139203885	-2145093689	-151008051907	-5067293342
Y_1	0	149589286122	1447675865	-149546915581
Z_1	-2467045747	35723456	2514822292	84388495
X_2	150301280330	2186867586	-148868371398	-748229640
Y_2	-2478588950	148328221452	-1034018214	-150818789722
Z_2	1287273238	-2121040808	-1224681458	2168940029
X_3	150301280330	2150754223	-148843921432	-760789288
Y_3	1287273238	150818900689	3970806949	-148327868749
Z_3	1286809645	2193080836	-1182122299	-2145580218

Phasemeter	Phasemeter SC	SC en vis-a-vis du banc optique	Code
s_1	1	2	1
s_2	2	3	2
s_3	3	1	3
s'_1	1	3	4
s'_2	2	1	5
s'_3	3	2	6

Table 12: Tableau donnant le codage utilisé dans TDI pour les bras.

Bras	Emetteur \rightarrow Récepteur	Code
1	3 \rightarrow 2	1
2	1 \rightarrow 3	2
3	2 \rightarrow 1	3
1'	2 \rightarrow 3	4
2'	3 \rightarrow 1	5
3'	1 \rightarrow 2	6

Table 13: Tableau donnant le codage utilisé dans TDI pour les phasemetre.