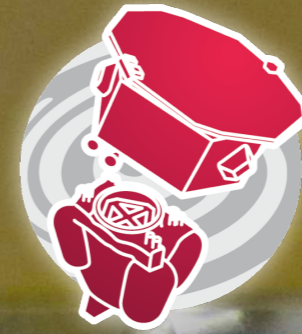




Data Analysis for LISA Pathfinder



lisa pathfinder

One man's gap is another man's experiment

M Hewitson for the LPF Team
Missing Data Workshop, Nice
11th May 2015

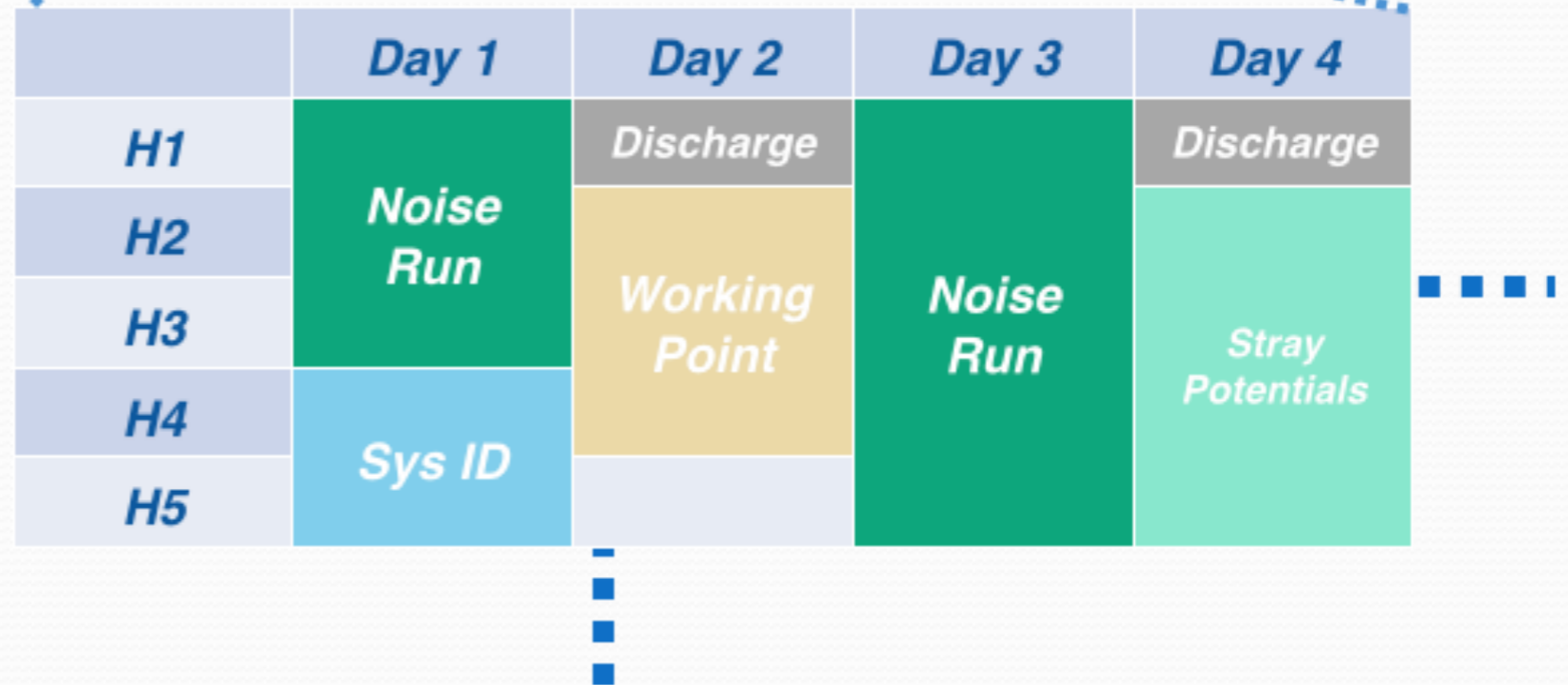
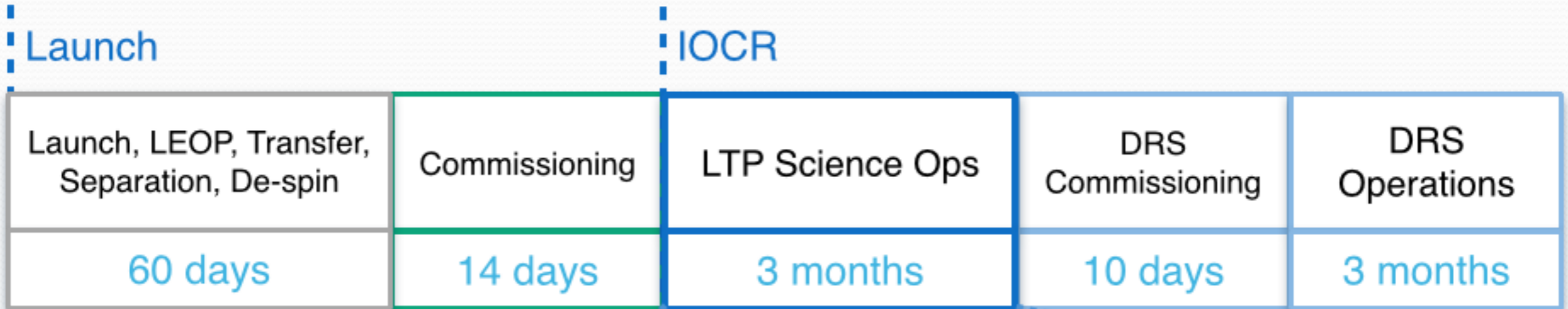


- LPF data analysis overview
- Operations environment
- Examples of gaps or bad data segments
- Dealing with isolated glitches or data gaps
- Free-flight experiment
- Analysis methods for free-flight experiment



- Obtain the best geodesic motion possible
 - quietest differential acceleration of the two TMs
 - $3 \times 10^{-14} \text{ m s}^{-2} / \sqrt{\text{Hz}}$ at 1 mHz
 - ~few pm accuracy position measurement of TM-SC, TM-TM
 - optimisation by changing system parameters
 - determine best configuration by experiments
- Develop a noise model of the system
 - allows the projection of the performance of technologies to LISA

Phases of Operations





- Analysis software and infrastructure
- Defining, simulating and testing experiments
- Analysis pipelines
- End-to-end tests (including hardware wherever possible)
- Interfacing with ESA's STOC and MOC

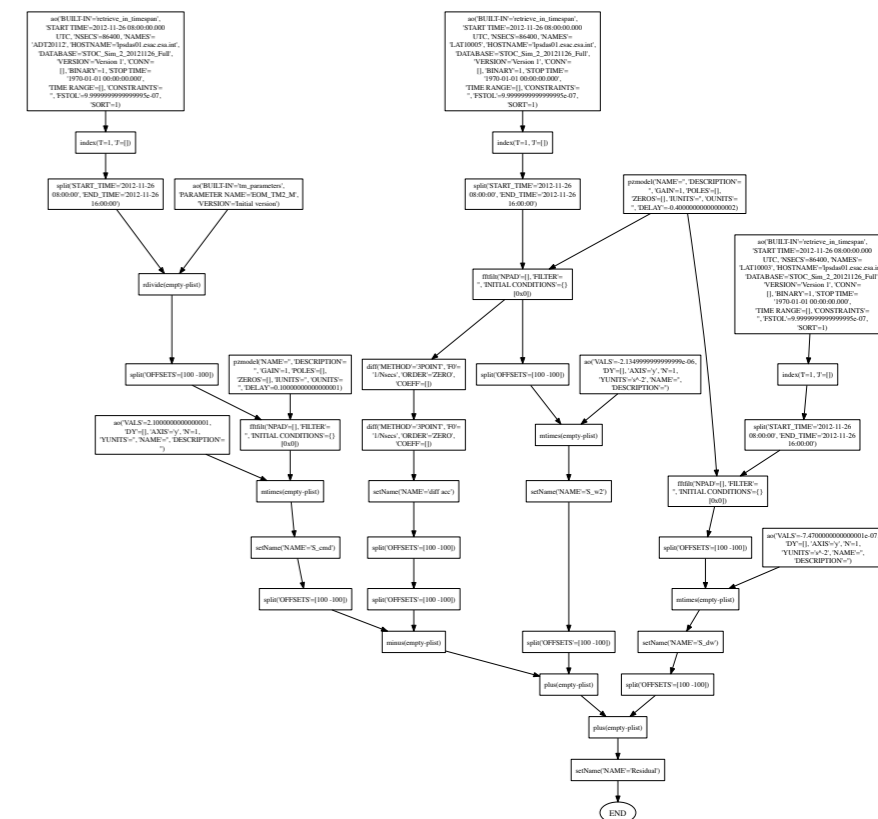


- LTPDA Toolbox

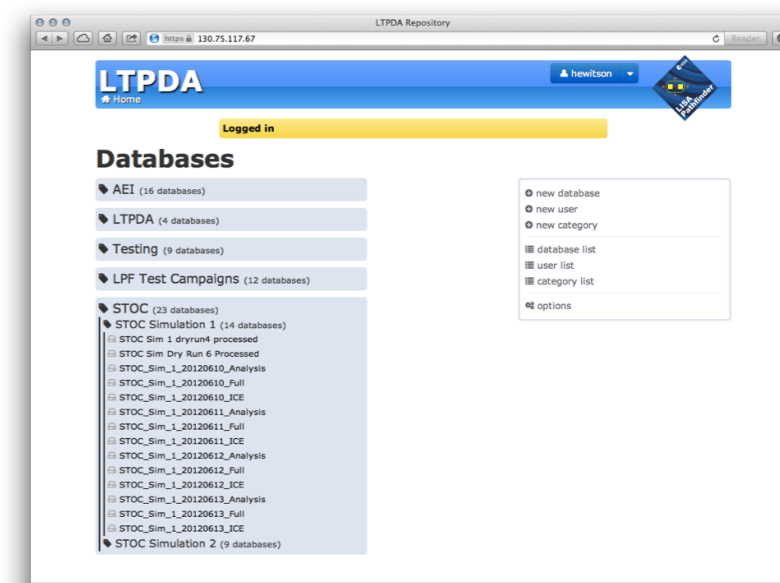
- MATLAB toolbox which implements an object-oriented data analysis environment
- objects track their history so results are traceable and reproducible
- heavily tested and documented
 - ~700 page user manual
 - ~6000 unit tests running every 3 hours
 - multiple system test campaigns
 - formal deliveries to ESA with acceptance tests

- LTPDA Repository

- provides a centralised database structure with web interface for administration and searching
- interface to LTPDA toolbox directly from within MATLAB to submit and retrieve objects
- core client/server system to be used by ESA for LPF mission
- also in heavy daily use in various labs



<http://www.lisa.uni-hannover.de/ltpda/>





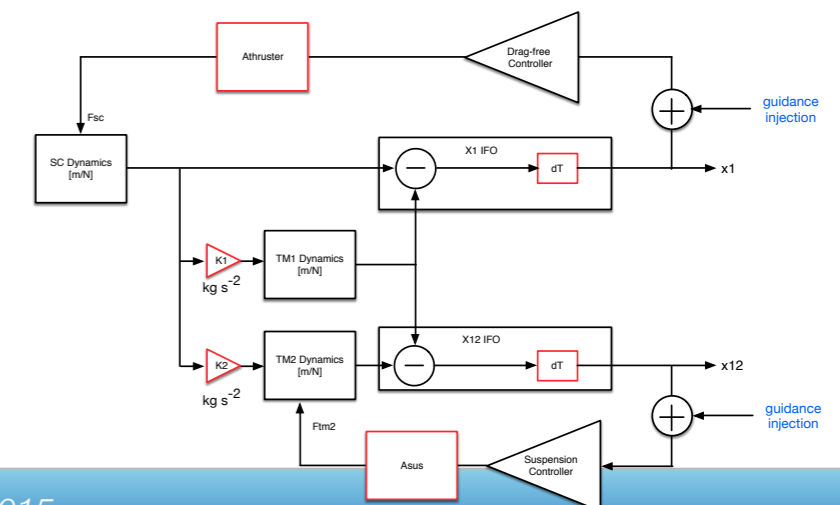
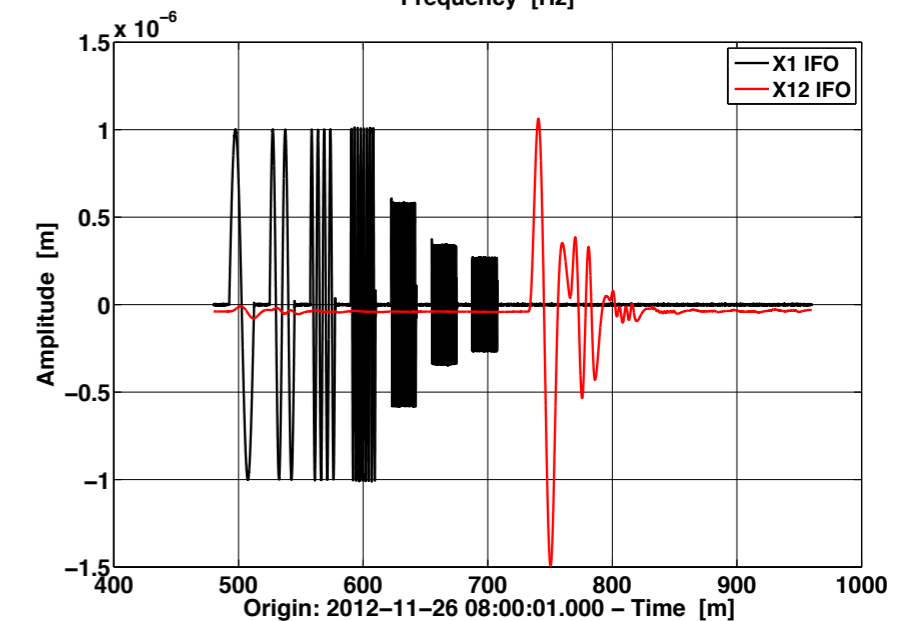
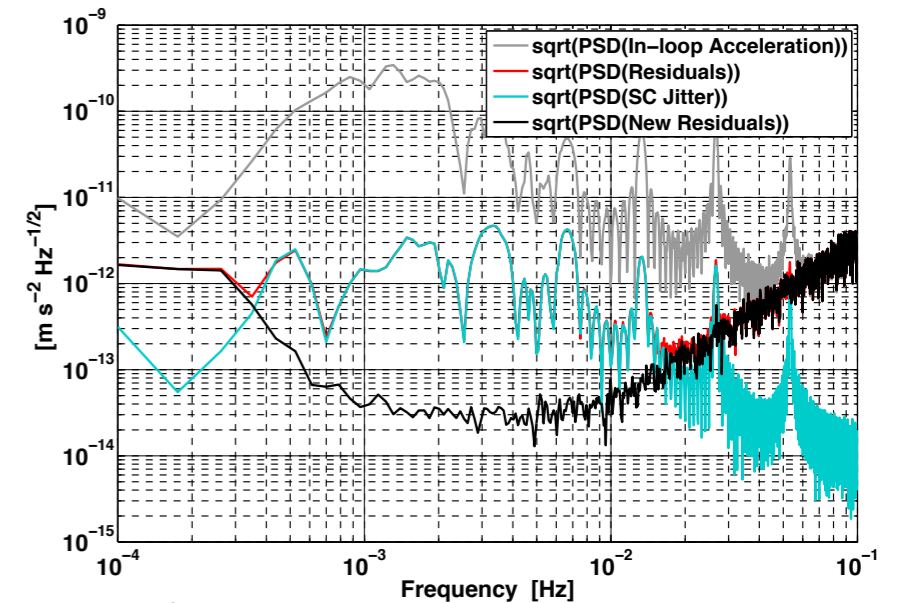
- We have 90 days in the main operational phase for LTP
- We have to plan about 1 week in advance
- We can only change the investigations that will run in about 5 days from now
- We have a large menu of possible investigations we can run
- How do we choose?
- Rough philosophy:
 - low-risk, gentle probing of the system first to gain experience and to understand the state of the system
 - move on to more invasive investigations and begin tuning the system
 - higher risk investigations are planned to be later in the operations

Primary analysis approaches



lisa pathfinder

- Spectral estimation
 - primary science goal, assessment of residual differential acceleration
 - coherence between various signals
 - formation of noise breakdown
- Fitting
 - system identification investigations
 - signal injections
 - dynamical, thermal, magnetic, etc...





Why care about 'missing' data?

- Stops us making spectral analysis using standard (well understood) techniques
 - something more sophisticated is needed
 - careful treatment of the data to avoid gaps/glitches
- Can cause problems during system identification experiments with injected signals
 - glitches: effective reduction of SNR
 - gaps:
 - if small, marginal loss of SNR but added complexity in processing (split into multiple experiments)
 - if large, significant loss of SNR
 - repeat experiment



- Missing data

- telemetry lost between satellite and ground station
 - should be rare (<1% of the time)
- telemetry lost on-board due to errors in data handling
 - should also be rare
- telemetry lost at MOC
 - high redundancy -> rare

- 'Bad' data

- noisy/non-stationary
 - system configuration
 - environmental disturbances
 - ...
- glitches
 - instrumental instabilities
 - timing and clock synchronicity driven effects
 - ...



From either missing data, or unusable data:

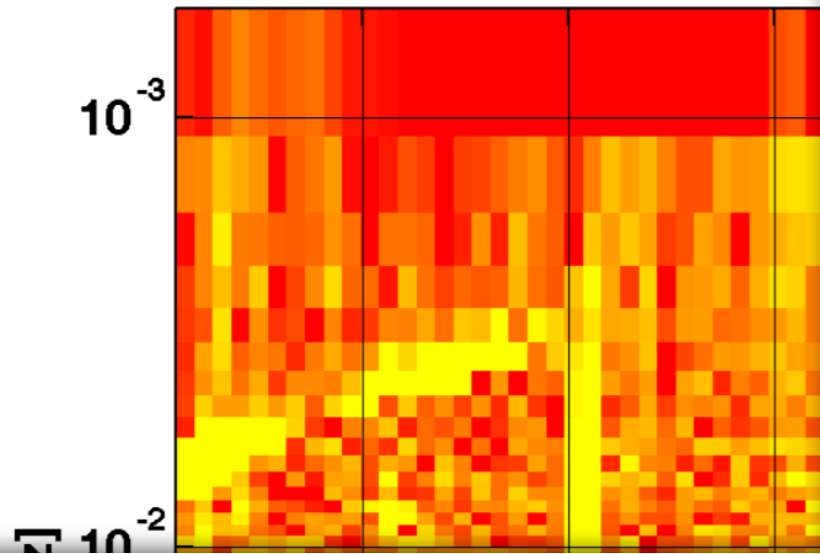
- Ignore
 - analyse segments either side
- Interpolate
 - for simple, large and/or 'slow' signals, this might be ok
 - for tiny gaps (a few samples), maybe even ok for noise only data
- Fill with representative simulated noise
 - with the goal to allow standard tools to be used
- Other techniques we learn about here?!

OMS Test-campaign data

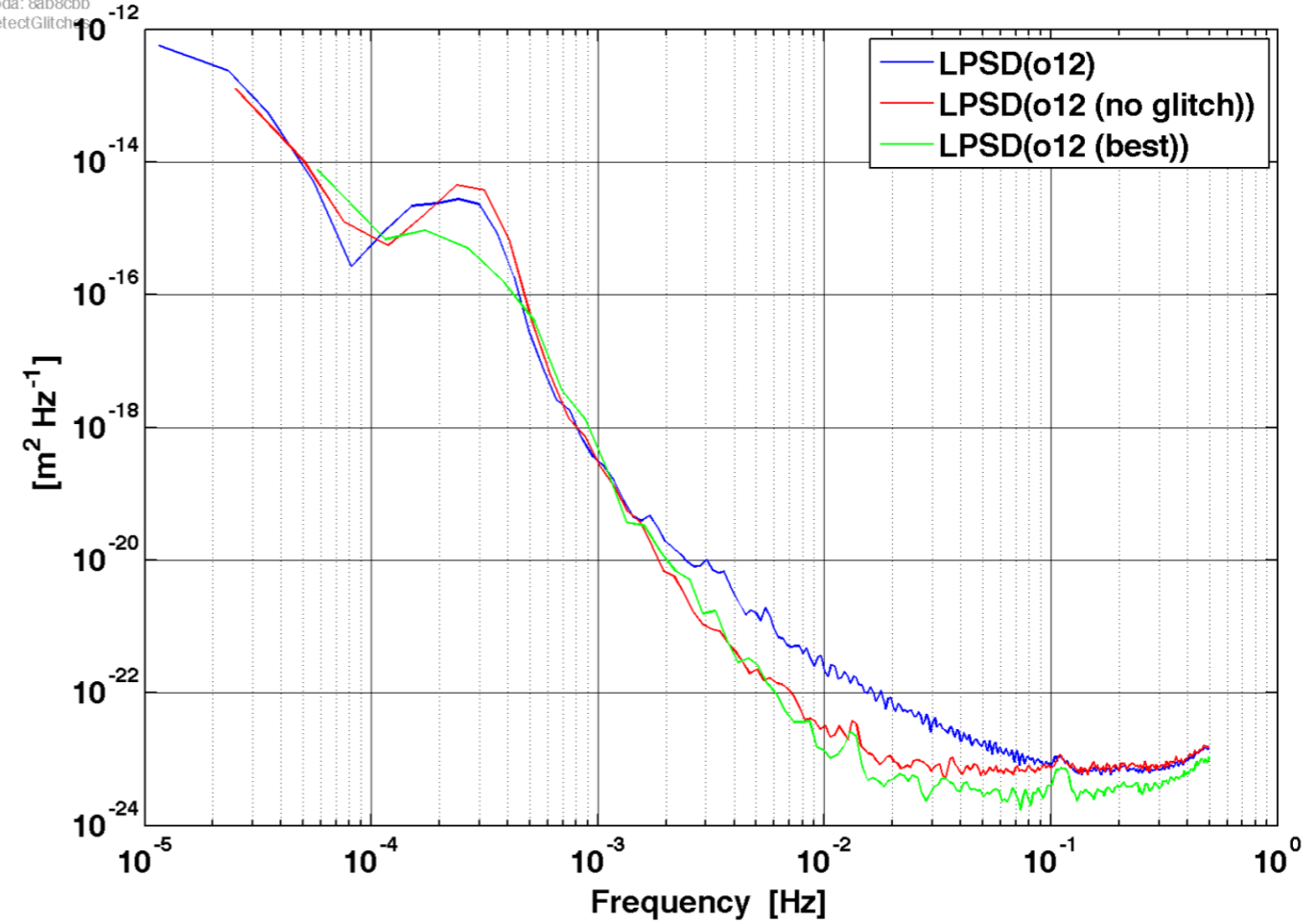


lisa pathfinder

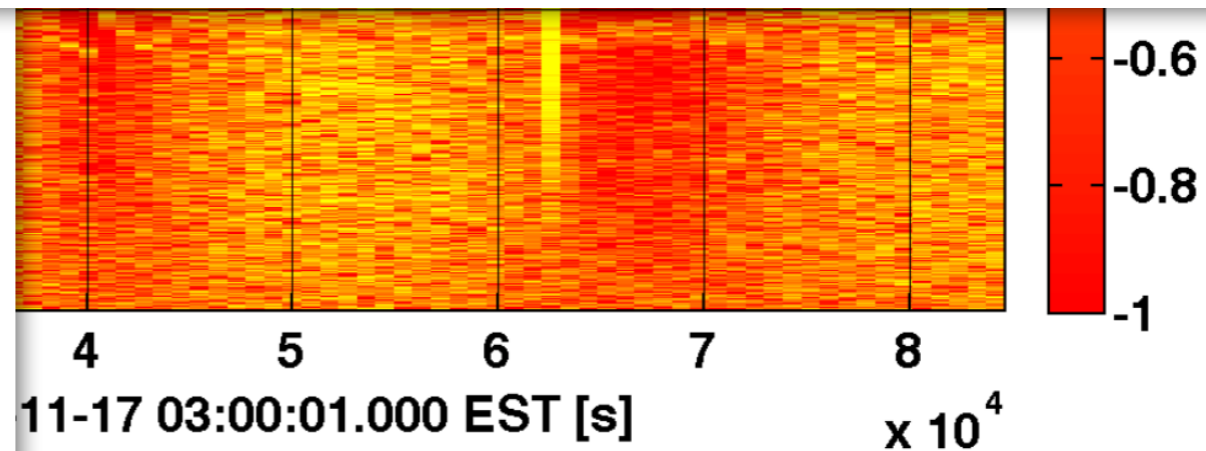
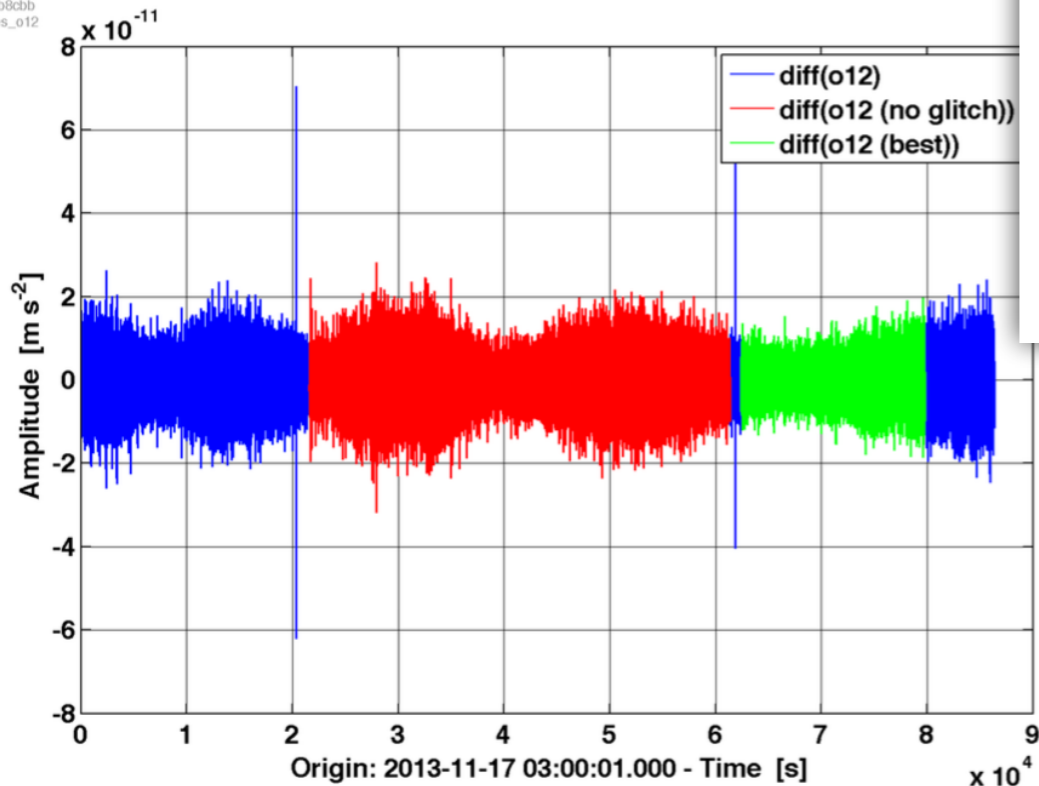
LTPDA 2.8.dev (R2013b)
2013-11-21 11:38:15.809 EST
ltpda: 8ab8cbb
plotSpectrogram



LTPDA 2.8.dev (R2013b)
2013-11-20 15:59:30.864 EST
ltpda: 8ab8cbb
detectGlitch



LTPDA 2.8.dev (R2013b)
2013-11-21 13:39:57.704 EST
ltpda: 8ab8cbb
fillGlitches_o12





- Fill gaps using constrained gaussian noise
 - requires a reasonable model for the spectrum of the underlying noise
- From model we generate a 2-point correlation function

$$C^{jk} = \int_{-\infty}^{+\infty} \frac{1}{2} S_y(f) e^{-2\pi f T(j-k)}.$$

- From this, we have a PDF for the noise process

$$P(\vec{y}) \propto \exp \left[-\frac{1}{2} \sum_{j,k} C_{jk} y^j y^k \right]$$

- Treat data inside and outside the gaps

$$\ln P(\vec{y}) = \text{const} - \frac{1}{2} \left[\sum_{\bar{g}, \bar{g}'} C_{jk} y^j y^k + 2 \sum_{\bar{g}, g} C_{jk} y^j y^k + \sum_{g, g'} C_{jk} y^j y^k \right]$$

Inside gaps to outside gaps

Data in gaps



We want to generate replacement data for the gaps which has the correct spectral properties and correlates ‘properly’ with the data we have.

Recipe:

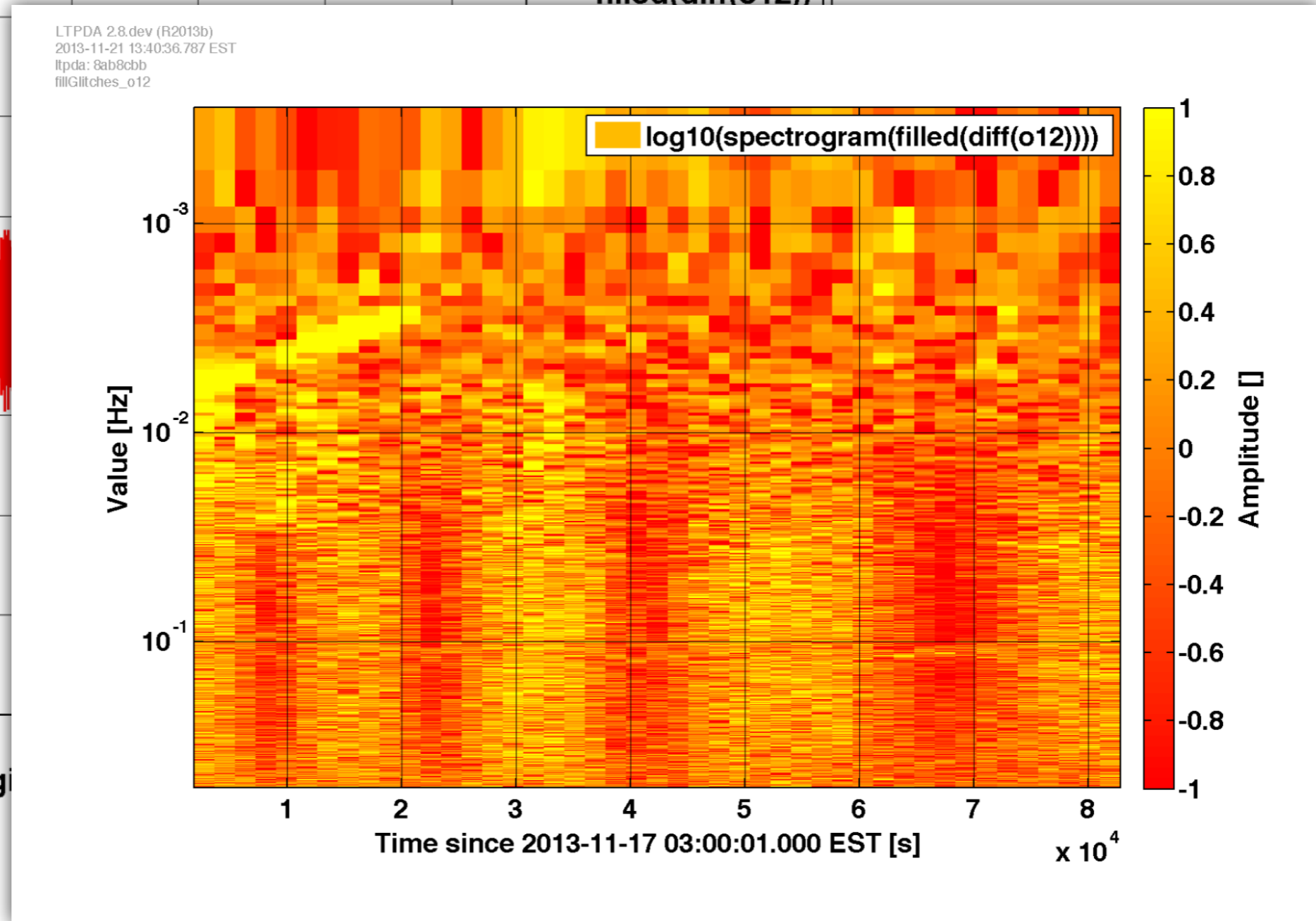
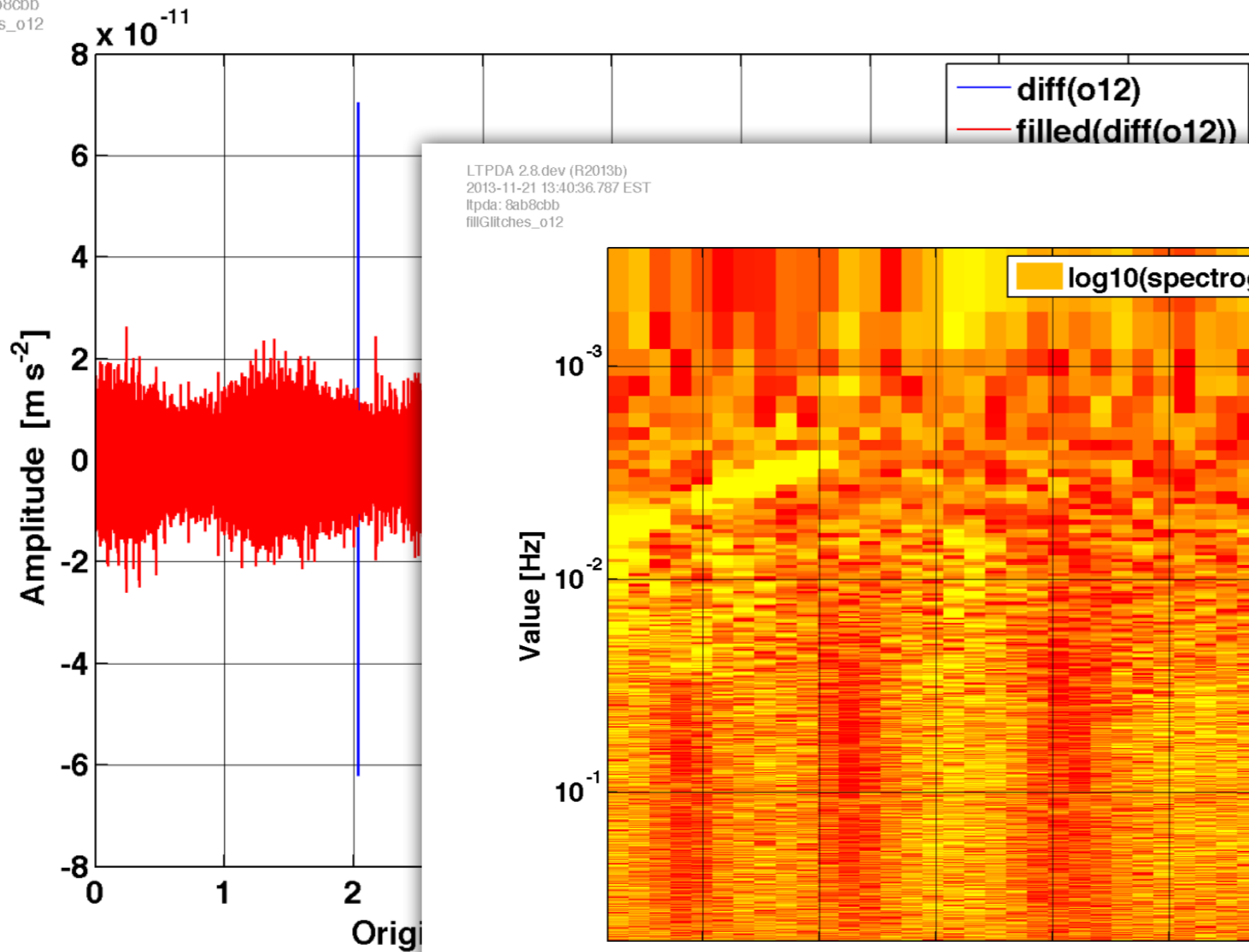
1. Estimate model for the spectral density of the data
2. Compute a 2-point correlation function from that model
3. Compute mean levels inside each gap $\Delta^j \equiv -\sum_k \bar{g}_k C^{jk} \lambda_k$
4. Draw samples from multivariate gaussian distribution with covariance C_{jk} and mean Δ^k .

Replace with constrained gaussian noise



lisa pathfinder

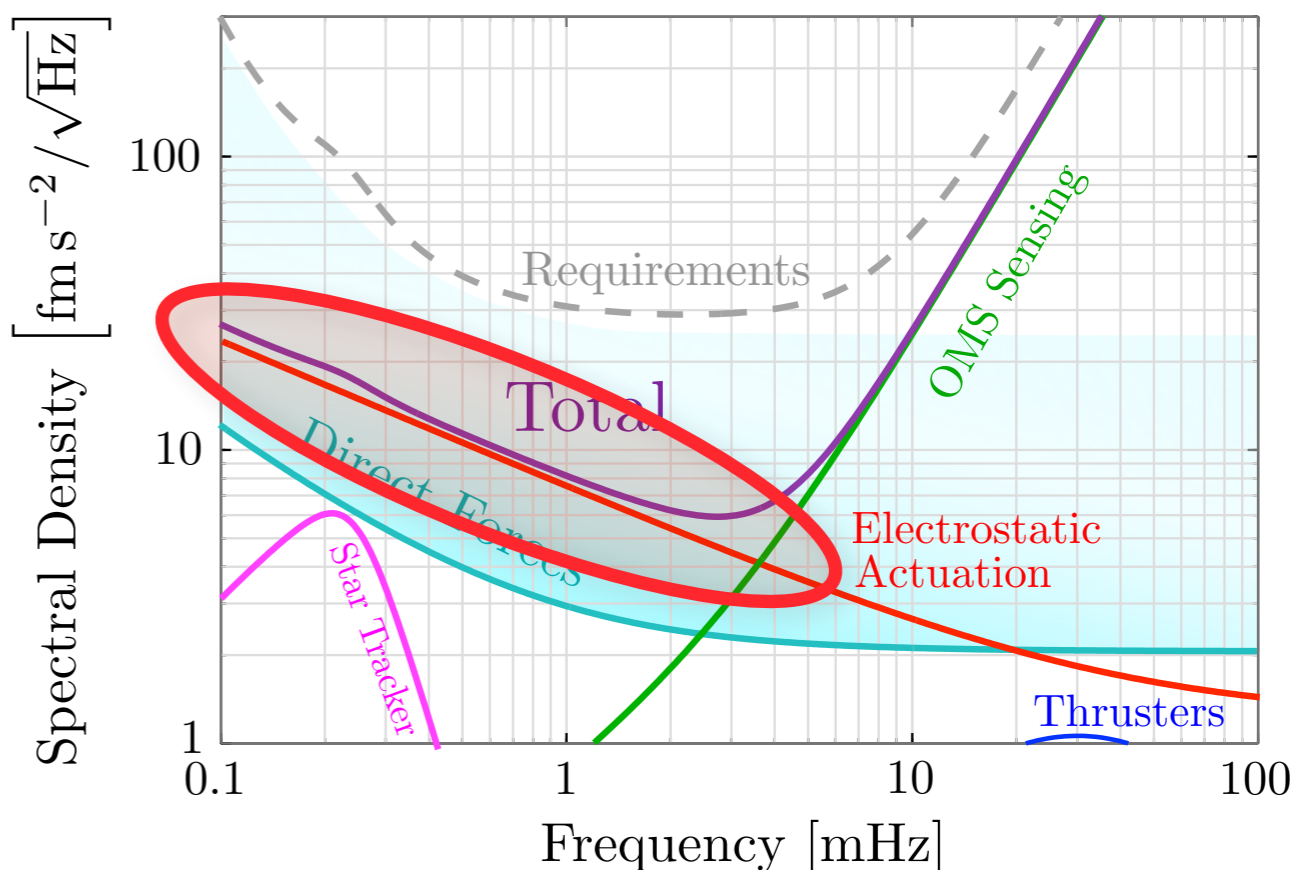
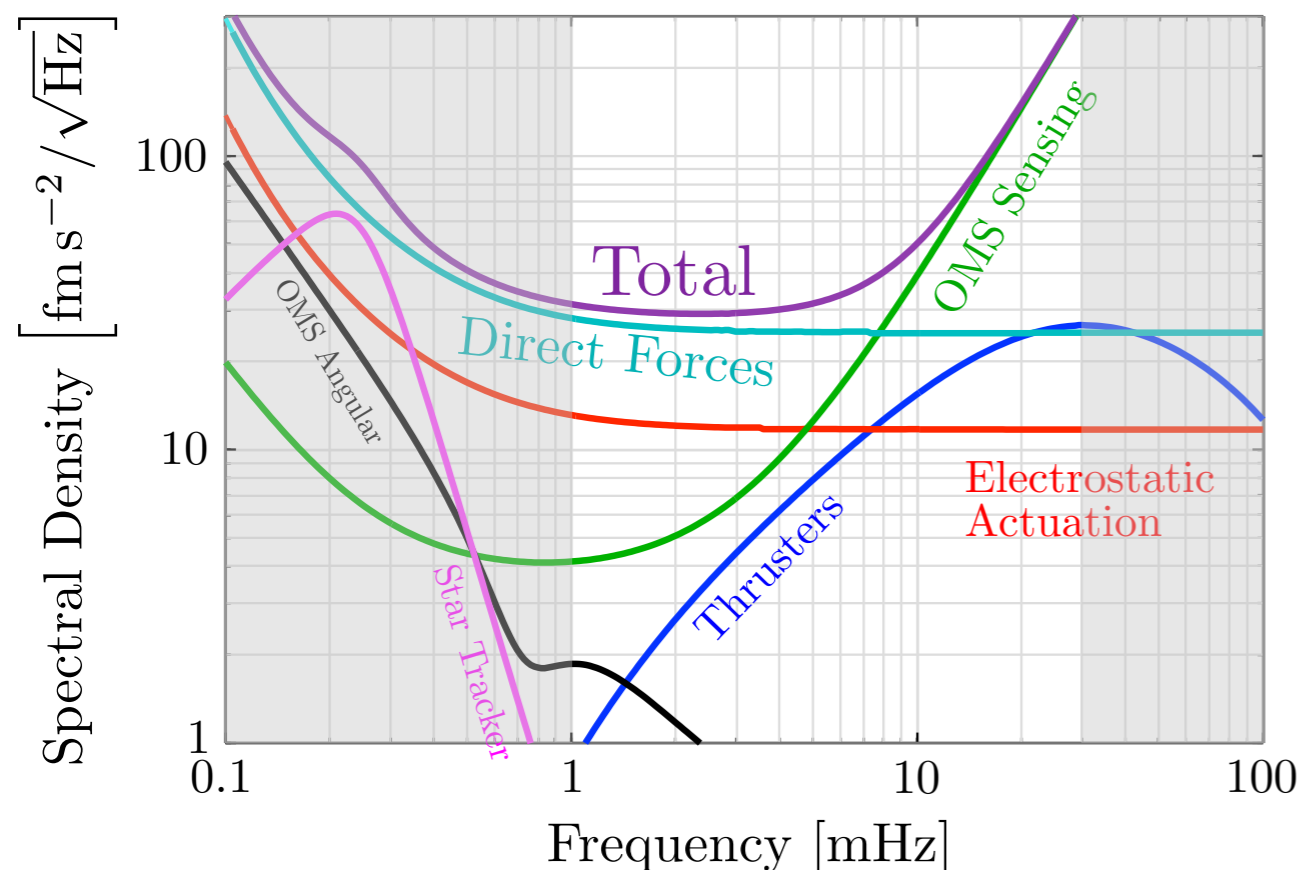
LTPDA 2.8.dev (R2013b)
2013-11-21 13:40:23.465 EST
ltpda: 8ab8cbb
fillGlitches_o12





Another source of 'gaps'

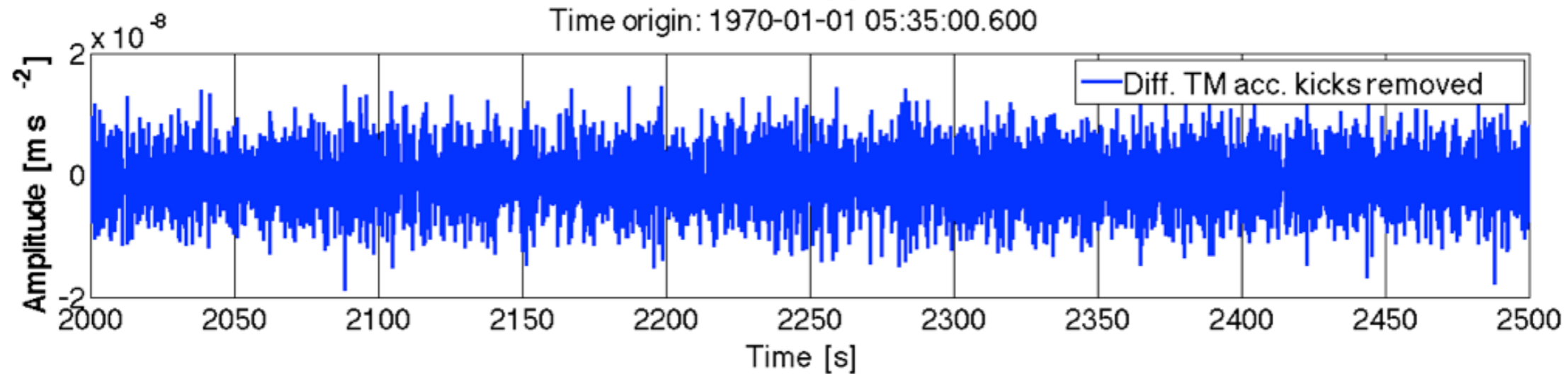
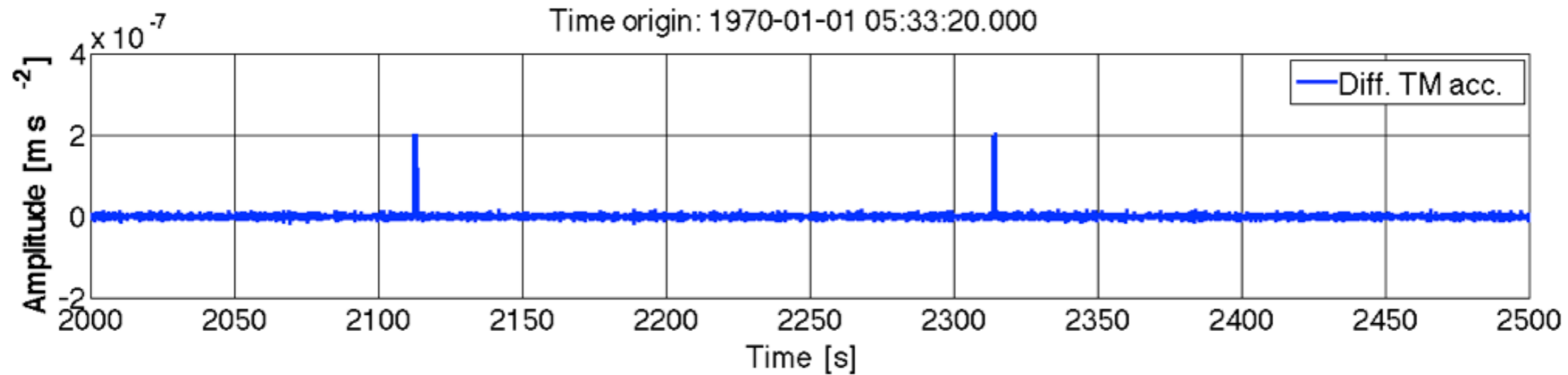
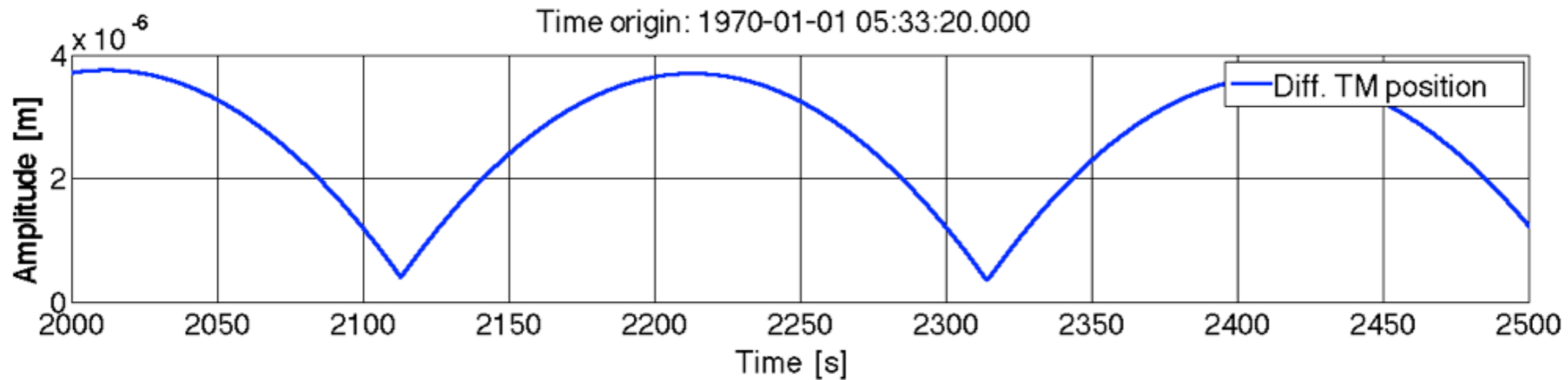
- Ones we create ourselves as a result of an experiment





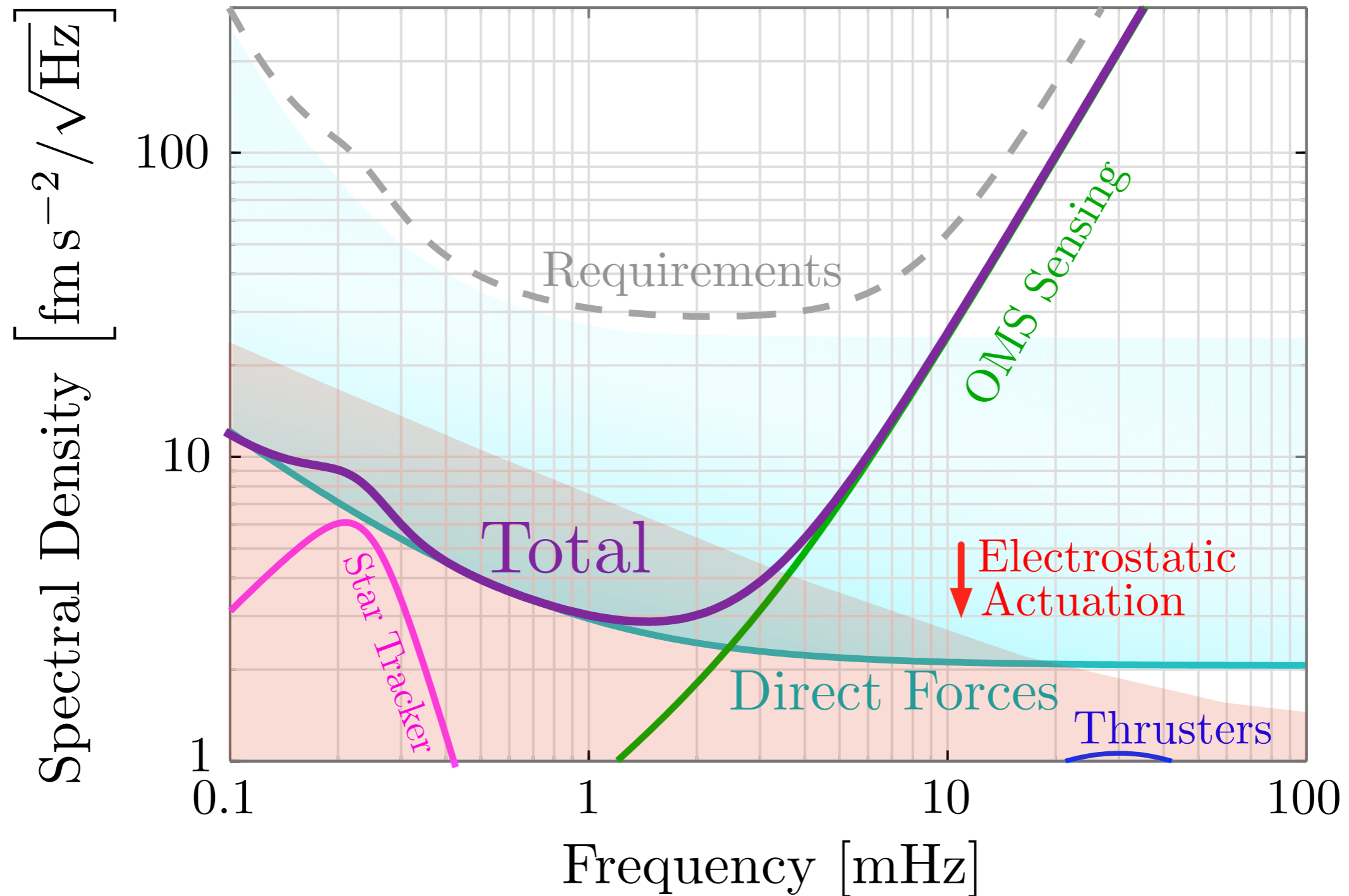
- Capacitive actuation may be limiting in our measurement band
- Do an experiment with the actuation off
 - must be short otherwise the TM will drift too far
 - repeat many times
- Experiment:
 - kick test-mass away
 - turn off actuation
 - let the test-mass drift (in parabola)
 - repeat

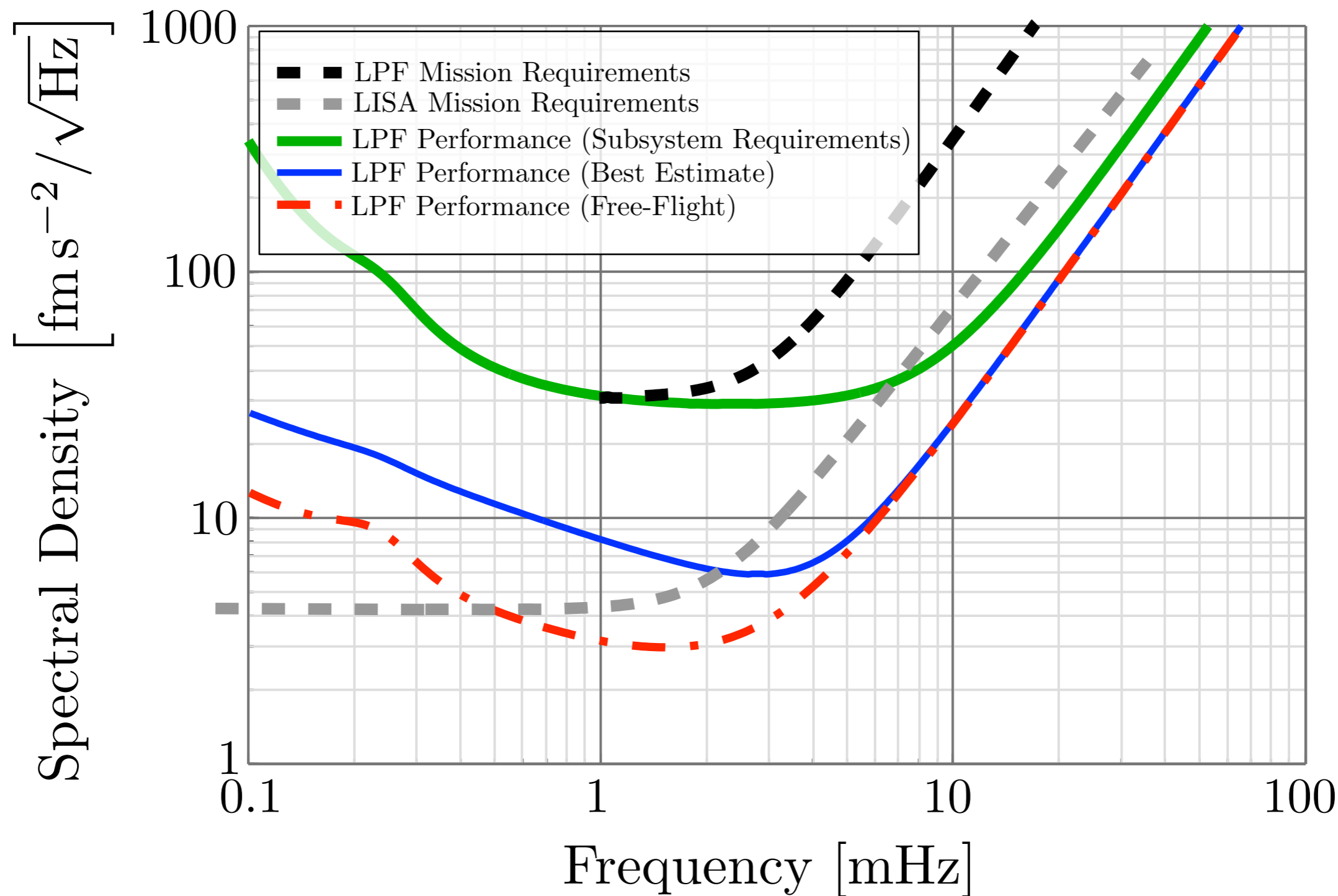
Simulation





- Need to analyse the multiple short (200s) drift segments to estimate the spectrum at 1mHz
- One approach: window the data, and proceed with normal PSD estimate
- Alternative methods include replacing the noise data during the kicks with noise of same spectral content as drift phases to allow standard spectral estimation techniques



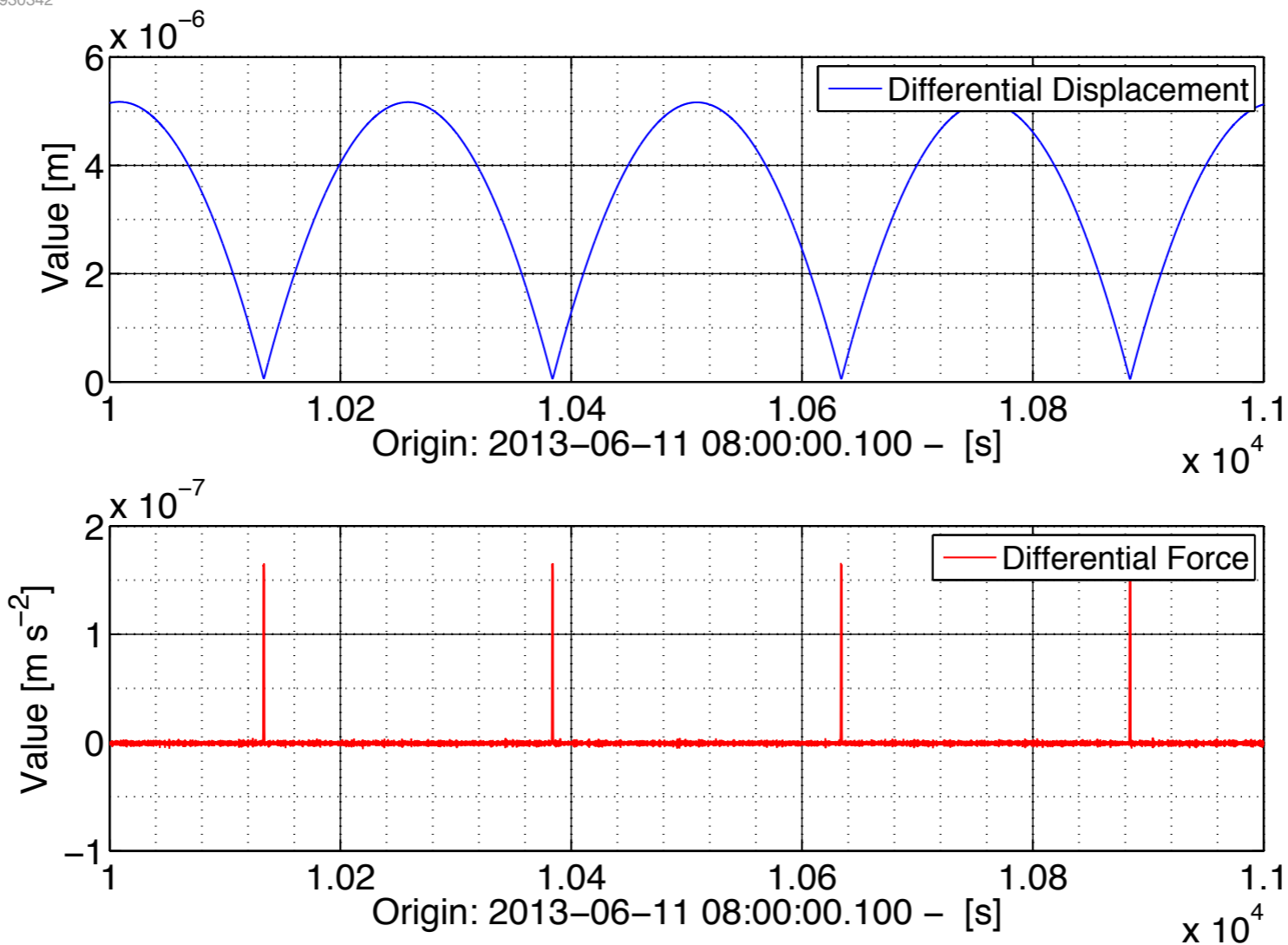


Suppress gaps: “windowing”

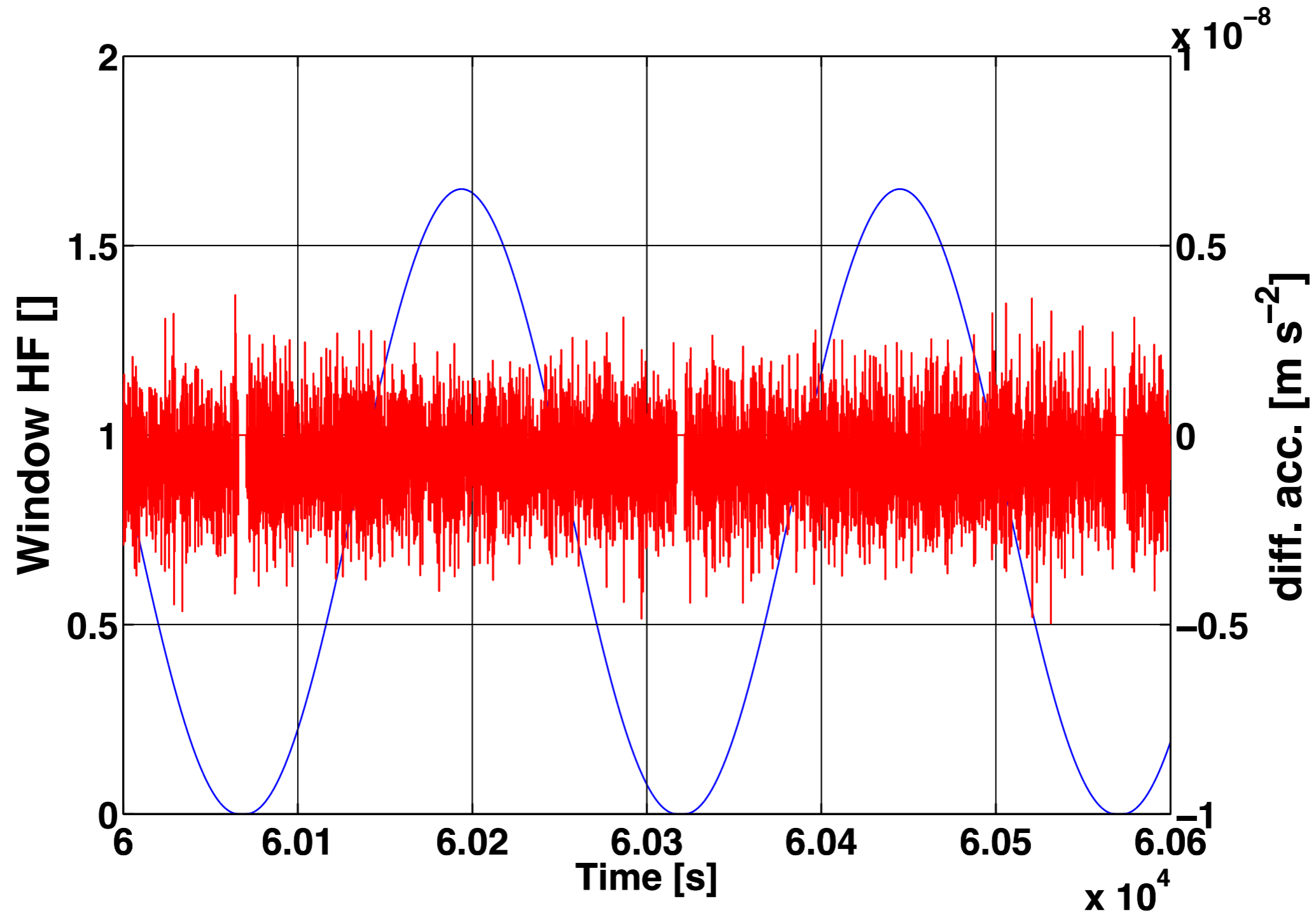


- For regular gaps, we can suppress the contribution by windowing
- Take second derivative of position:

LTPDA 2.7.dev (R2013b Prerelease)
2013-07-01 06:54:25.819
ltpda: b930342
plot



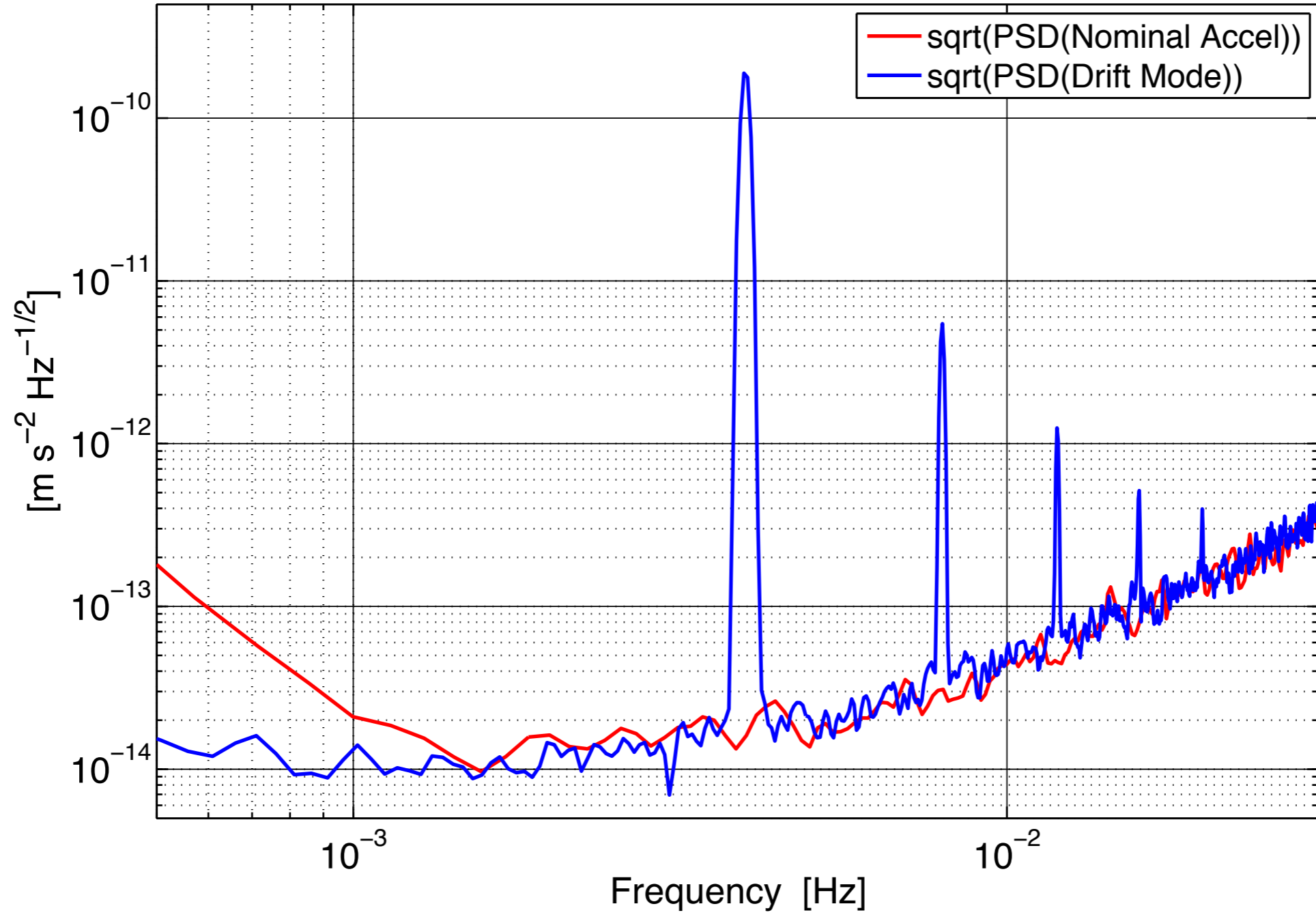
Apply windows



Take spectrum



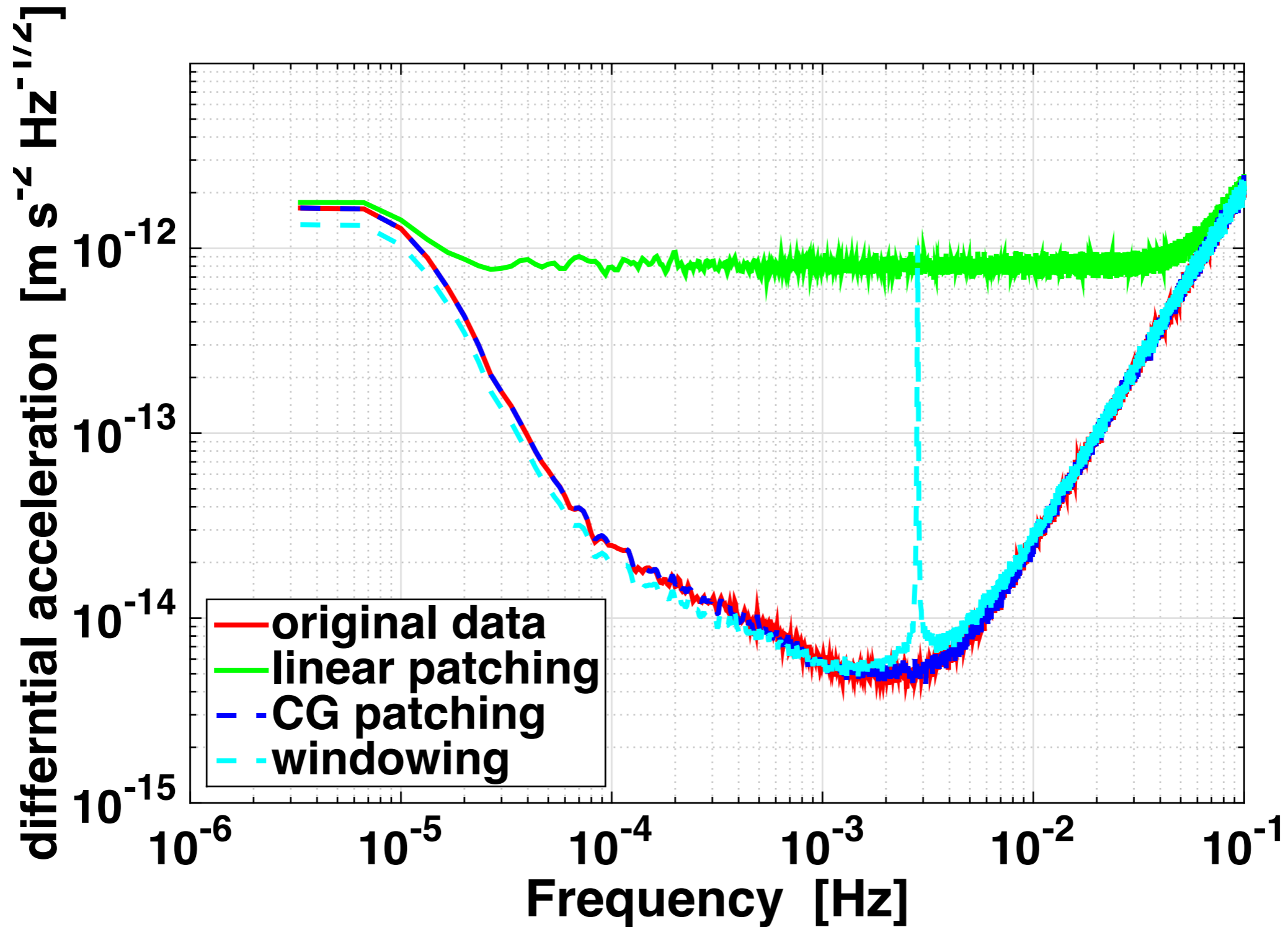
Residual acceleration





- Again, start from a model spectrum
 - fit to data from before free-flight experiment
- Identify each ‘gap’ corresponding to the noisy force kicks
- Generate data which has the correct spectral characteristics
 - correct gaussian noise within the gaps
 - correct correlations on longer timescales

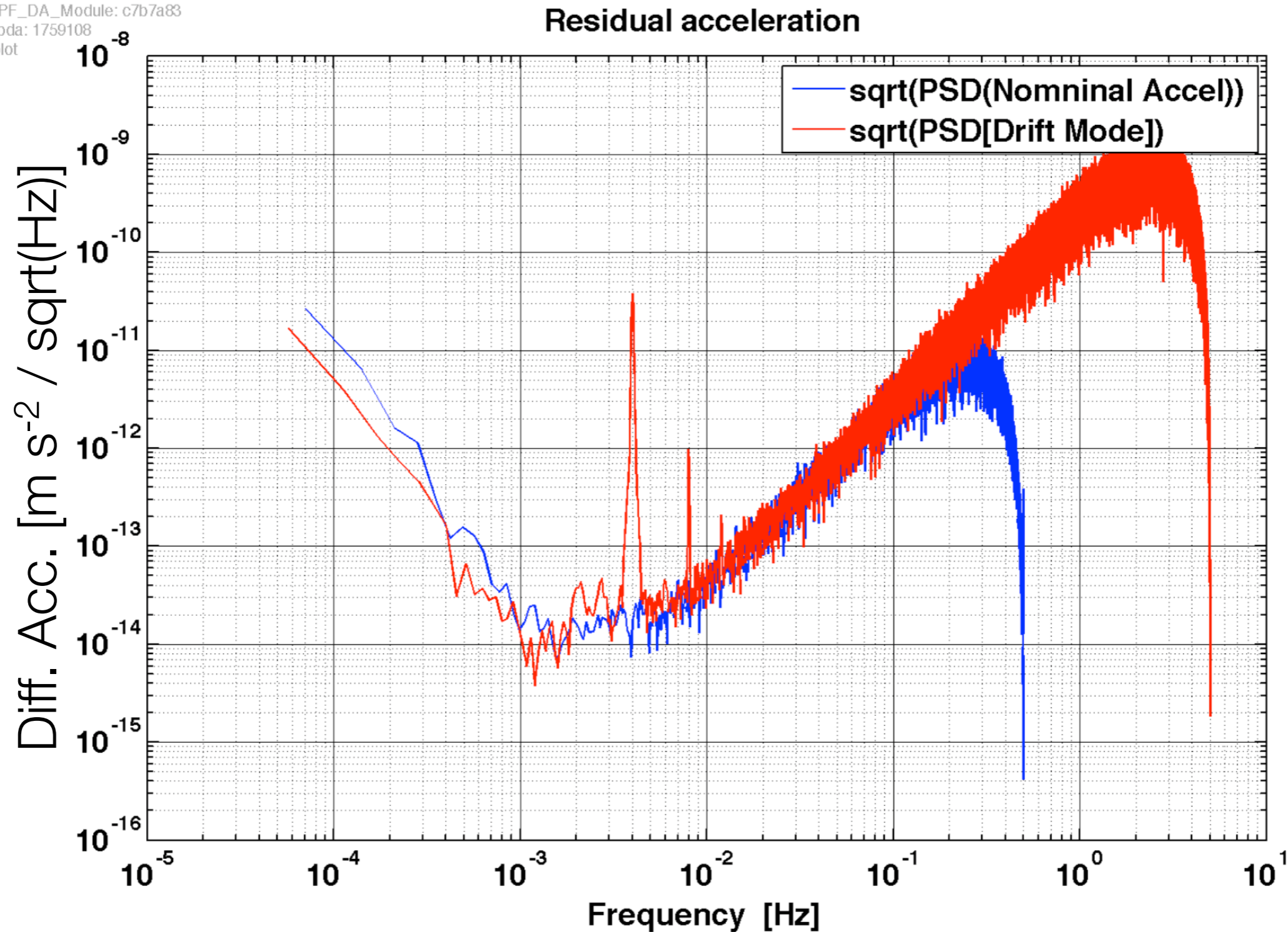
Simulated LPF noise with gaps



Simulated LPF free-flight data



LTPDA 2.6 (R2013a)
2013-06-14 10:00:23.819 UTC
LPF_DA_Module: c7b7a83
ltpda: 1759108
iplot



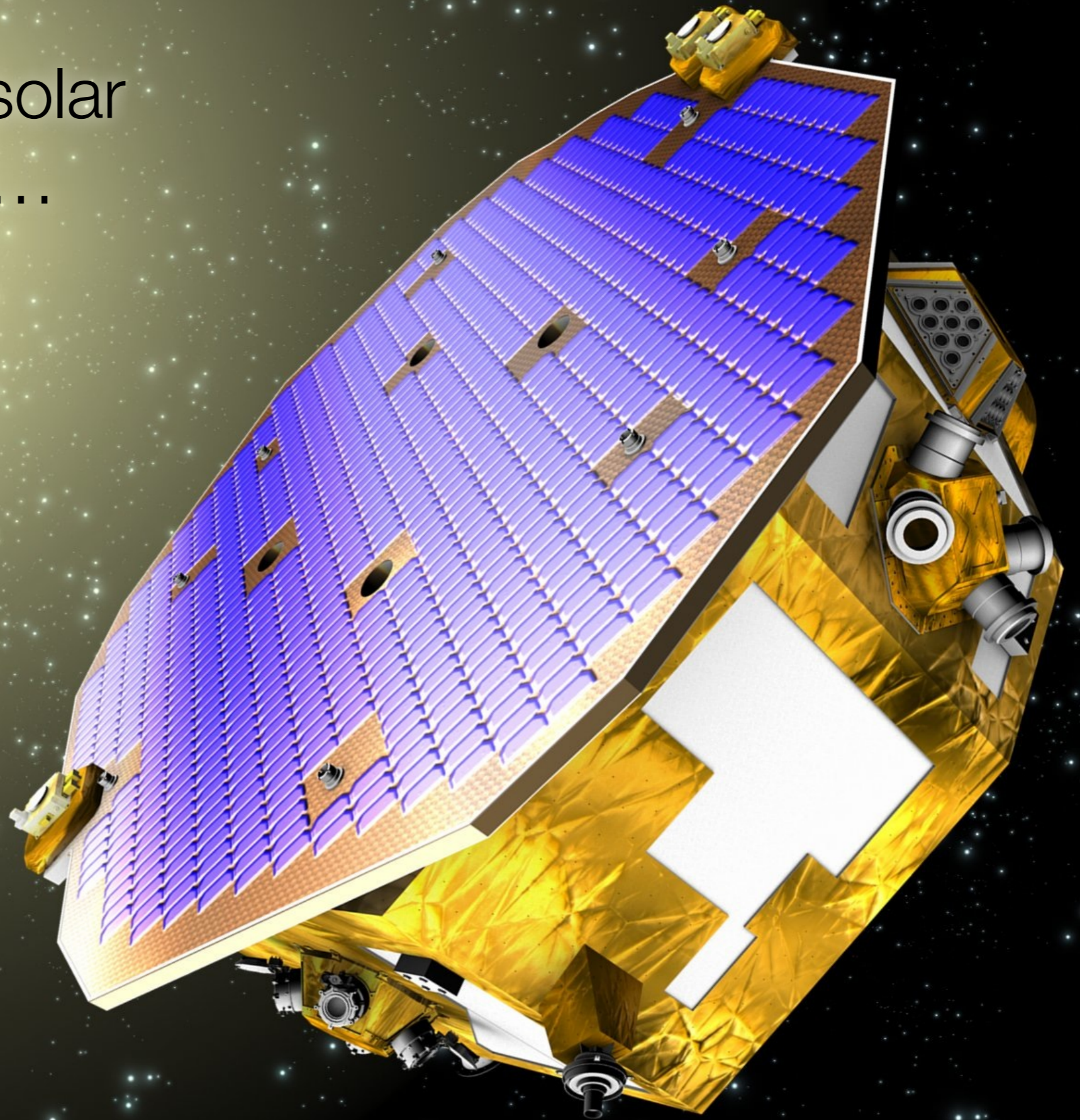


- The devil is in the details
- On few gaps this works well
 - accumulated error is small?
- On more realistic simulated data, or on real data from a torsion pendulum experiment, the results are less good.
 - aliasing issues?
 - incorrect modelling of the deterministic free-flight?
 - something else we didn't think of?



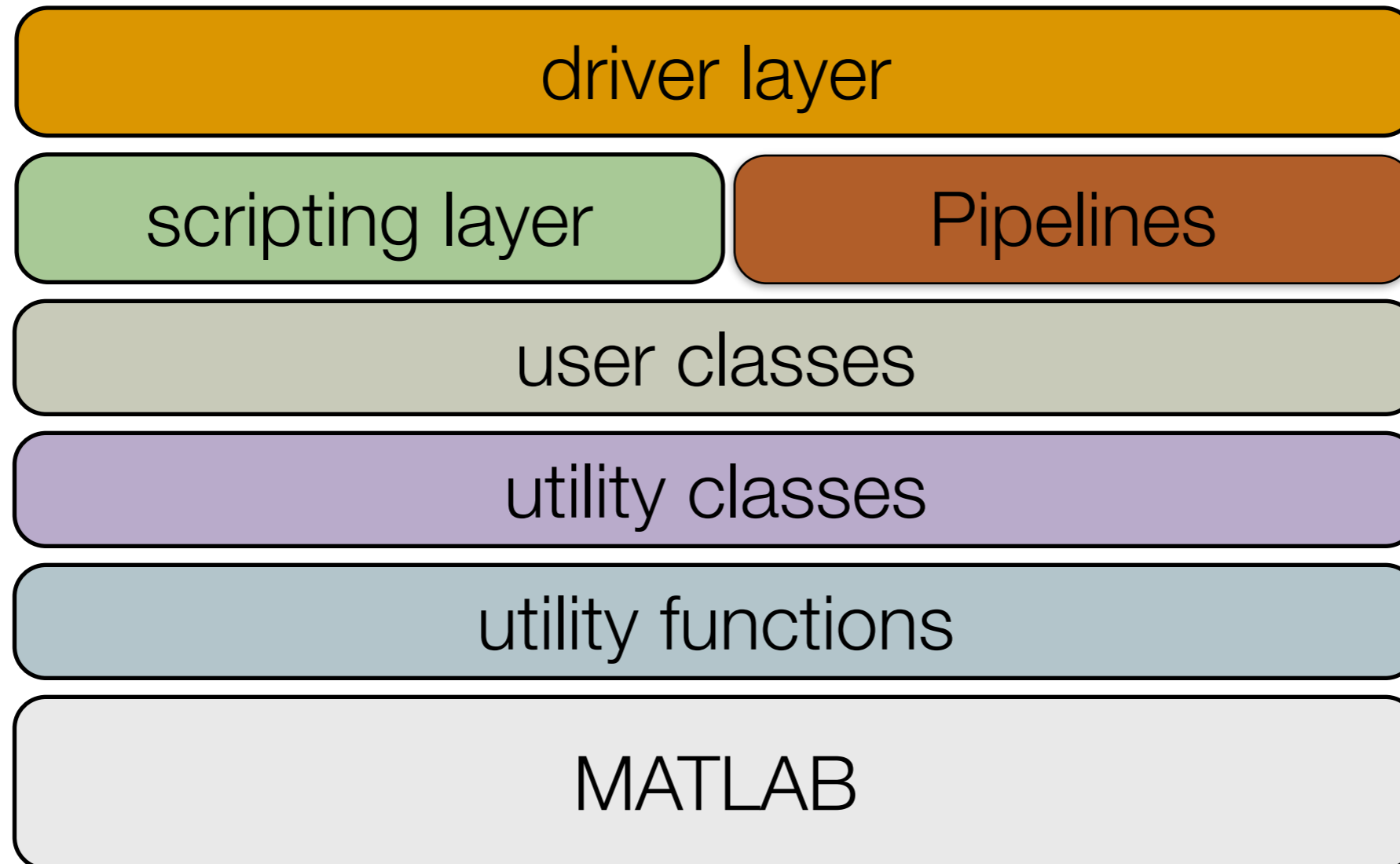
- Gap filling with constrained gaussian noise promises good results
 - details need to be worked out
 - too much ‘tuning’ needed at the moment
 - need to keep error accumulation under control
- It is likely that we will have some gaps and/or periods of bad data in LPF
 - mission lifetime is short, so all data is valuable
 - need to handle these cases

Coming soon to a solar system near you...





- Main requirement:
 - provide a data analysis environment suitable for analysing the experiment data in ‘real-time’
- Additional requirements:
 - toolkit of fairly standard instrument characterisation algorithms
 - results should be reproducible and accountable
 - reduce testing by building on a commercial product
 - ease of use (usable by non-programming experts)
 - simple data access for multiple users





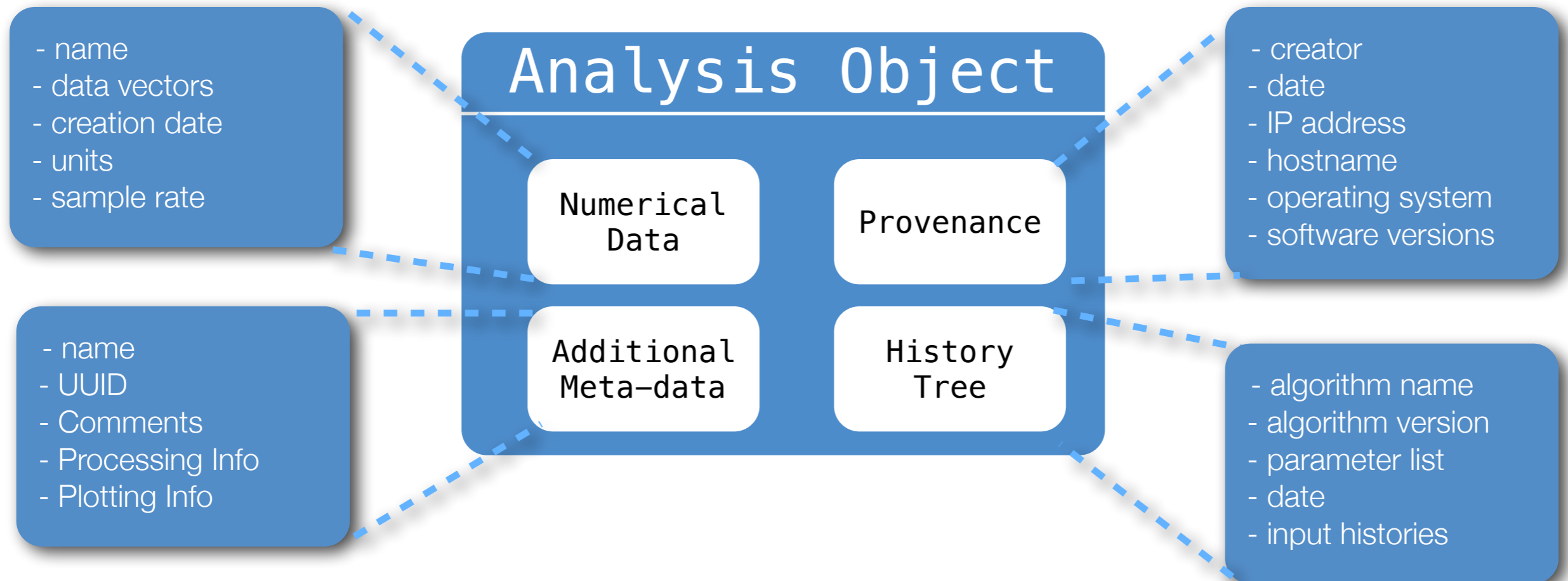
- LTPDA offers an object-oriented data analysis framework
 - we encapsulate (describe) different data analysis concepts with classes
 - users instantiate (build) these classes to get ltpda objects
 - DA algorithms are methods (functions) of these different classes
 - users act on the objects using the methods



Example: Analysis Objects



- We want to encapsulate the concept of an analysis result
 - avoid storing images, text files, documents, etc





```
>> methods ao
```

```
Contents          detrend          hist_gauss       mdc1_ifo2acc_inloop rotate          std
abs               dft              hypot            mdc1_ifo2cont_utn  round          straightLineFit
acos             diag            ifft            mdc1_ifo2control   sDomainFit    string
angle           diff            imag            mdc1_x2acc         save          submit
ao              display        index           mean               scale         sum
asin            dopplercorr     integrate       median            scatterData   sumjoin
atan            downsample     interp          min               search        svd
atan2           dropduplicates interpmisssing  minus            select        svd_fit
bilinfit        dsmean         inv             mode              setDescription t0
bin_data        dx             iplot          mpower            setDx         table
bsubmit         dy            iploty         mrdivide          setDy         tan
buildWhitener1D eig            isprop         mtimes            setFs         tdfit
cat             eq             isvalid        ne                setMdlfile    tfe
char            eqmotion       join            noisege1D          setName       timeaverage
cohere          evaluateModel  lcohere        noisege2D          setPlotinfo   timedomainfit
complex         exp            lcpsd          norm              setProcinfo   times
compute        export         le             normdist          setT0         timeshift
confint        fft            len            nsecs            setUUID       transpose
conj           fftfilt       linSubtract    offset            setX          type
consolidate     filtSubtract  lincom         optSubtraction    setXY         uminus
conv           filter        linedetect     phase             setXunits    unwrap
convert        filtfilt      linfit         plot              setY         update
copy           find          lisovfit       plus              setYunits    upsample
corr           firwhiten     ln             polyfit           setZ         validate
cos            fixfs         log            polynomfit        sign          var
cov            fngen         log10          power             simplifyYunits viewHistory
cpsd           fromProcinfo  lpsd          psd               sin           whiten1D
crbound        fs            lscov         psdconf           sineParams    whiten2D
created        gapfilling    lt             pwelch            smallvector_lincom x
creator        gapfillingoptim ltfe           quasiSweptSine   smallvectorfit  xcorr
csvexport      ge            ltp_ifo2acc   rdivide          smoother      xfit
ctranspose     get           max            real              sort          xunits
curvefit       getdof        mcmc          rebuild           spectrogram   y
delay          gnuplot       md5           removeVal        spikecleaning yunits
delayEstimate  gt            mdc1_cont2act_utn report           split         zDomainFit
demux          heterodyne    mdc1_ifo2acc_fd resample          spsd         zeropad
det            hist          mdc1_ifo2acc_fd_utn rms              sqrt
```

Don't touch me with that AO, I don't know where it's been!



```
ao(FILENAME=mdc2_r2_20...,  
  COLUMNS=[1 2], ROBUST=no,  
  TYPE=tsdata, XUNITS=1xunit,  
  YUNITS=1xunit, COMMENT_CHAR=  
  *, USE_FS=", FILEPATH=")
```

```
setName(NAME=o1_n1)
```

```
setYunits(YUNITS=m)
```

```
setDescription(DESCRIPTION=  
  Noise outp...)
```

```
save(FILENAME=o1_n1.xml)
```

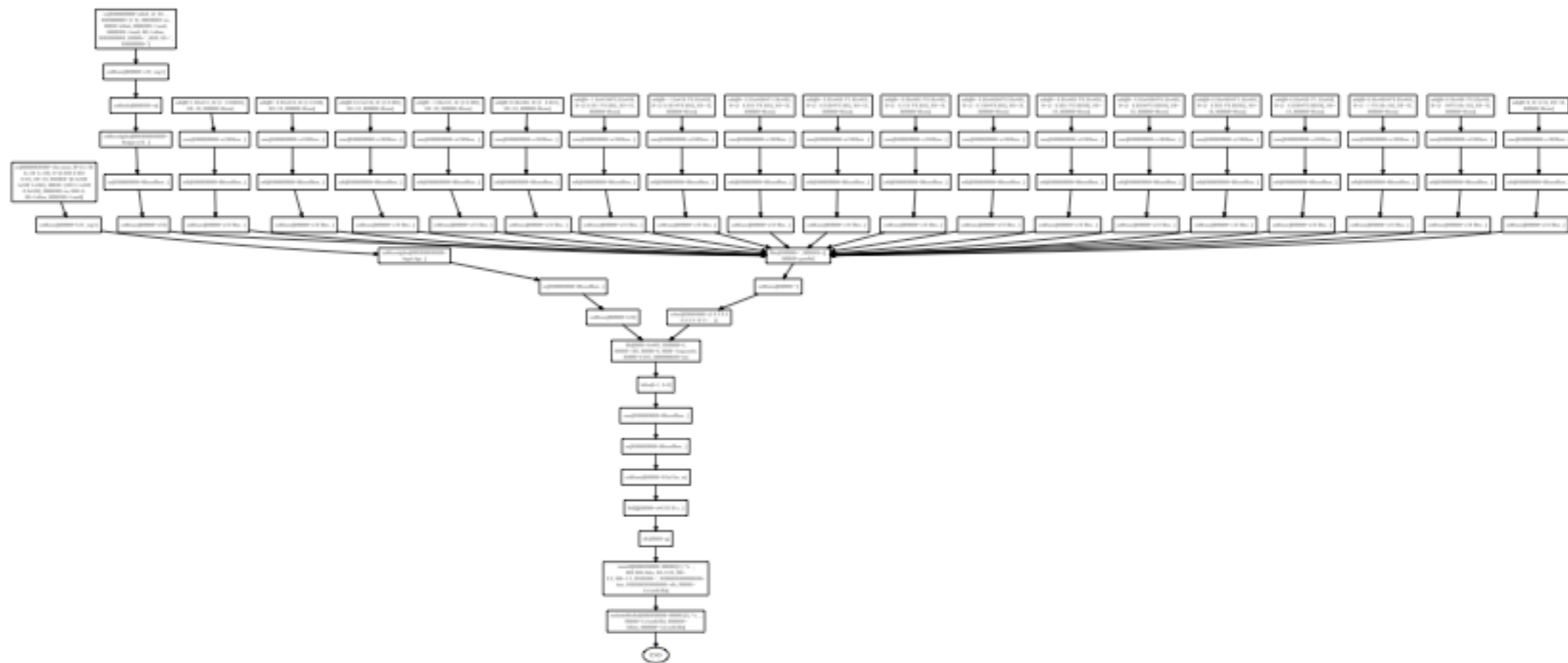
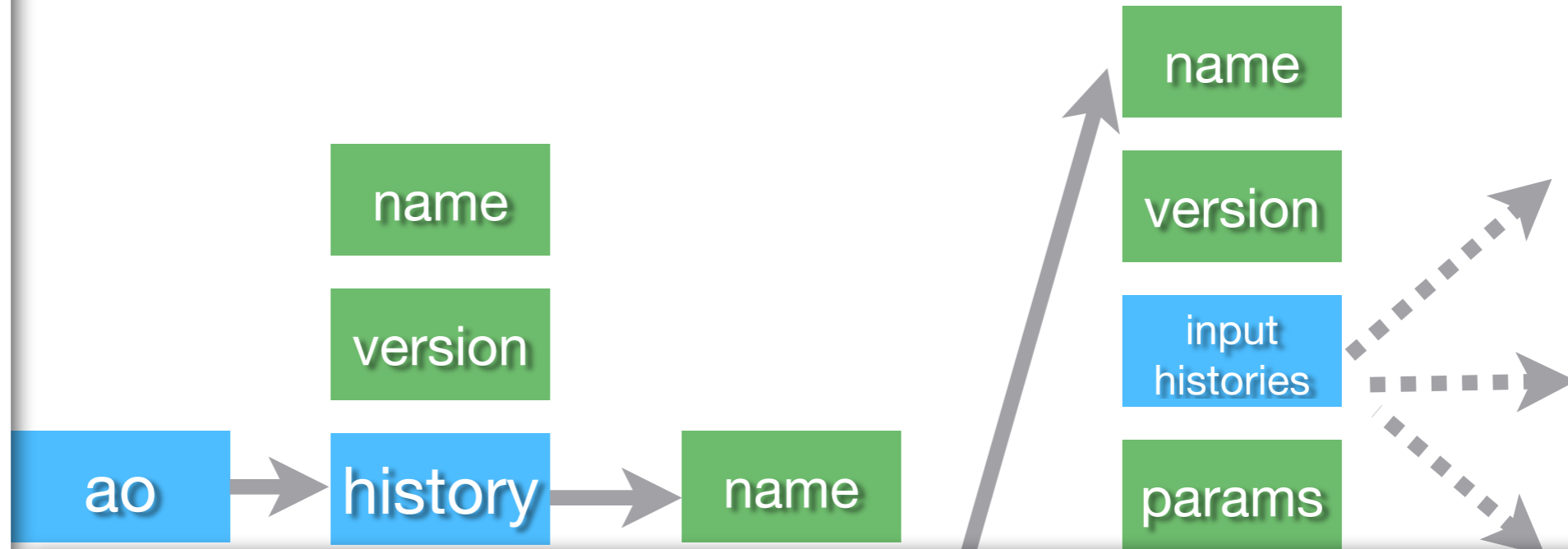
```
ao(FILENAME=/Users/hew...)
```

```
setName(NAME=o1)
```

```
when1D(MODEL=[], MAXITER=  
  50, POLETYPE=2, MINORDER=  
  17, MAXORDER=25, WEIGHTS=2,  
  PLOT=true, DISP=true, RMSEVAR=3,  
  FITTOLERANCE=0.05, KDES=100,  
  JDES=1e+03, LMIN=0, WIN=1xspecwin,  
  OLAP=-1, ORDER=0, SCALE=PSD,  
  RAND_STATE=2.1e+05)
```

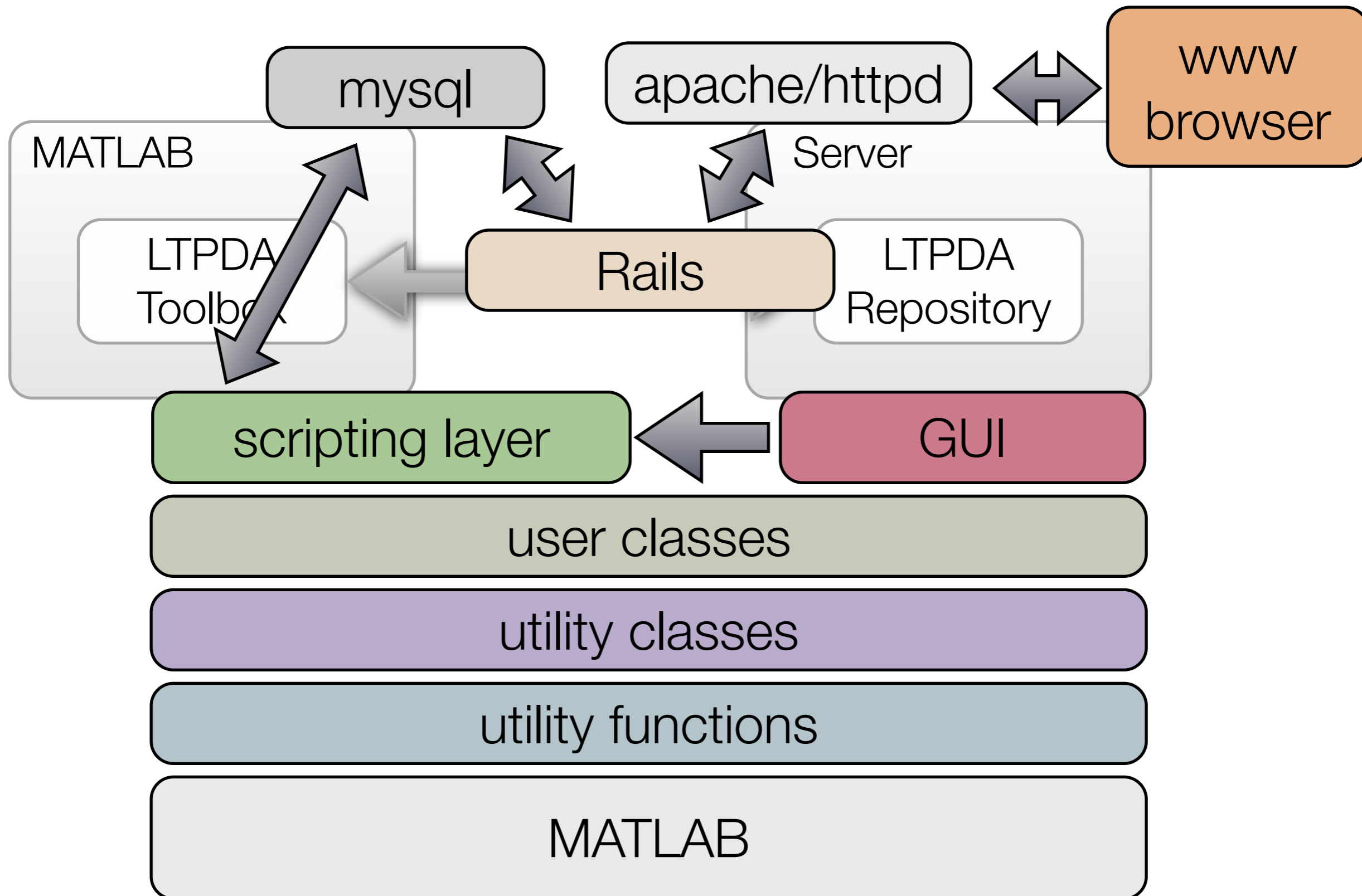
```
lpsd(JDES=1e+03, KDES=100,  
  LMIN=0, WIN=1xspecwin, OLAP=  
  0.661, ORDER=0, SCALE=PSD)
```

END



params

Where can I put my AO?





The screenshot shows the LTPDA Repository web interface. At the top, there's a navigation bar with the LTPDA logo, a home link, a user profile for 'hewitson', and a Lisa Pathfinder logo. Below the navigation bar, a yellow banner indicates 'Logged in'. The main content area is titled 'Databases' and lists several categories: AEI (25 databases), LTPDA (19 databases), Testing (9 databases), LPF Test Campaigns (14 databases), and STOC (92 databases). To the right of the database list is a navigation menu with options: new database, new user, new category, database list, user list, category list, and options. In the bottom left corner, there's a 'Repository Connection' dialog box with fields for hostname (130.75.117.67), username (hewitson), password (masked), and database (ltpda_test). It has 'Get list', 'Cancel', and 'Connect' buttons. In the bottom right, there's a 'Query: 130.75.117.67/ltpda_test' dialog box showing a query editor with a table list (objmeta) and a query: `SELECT obj_type,name,created,ip FROM objmeta ORDER BY id DESC;`. It has 'Done' and 'Execute' buttons.

Manage User Accounts

Manage Databases

Browse/Query Databases

Submit Objects

Retrieve Objects



- Documentation

- user-manual ~700 pages

- ~4500 documented m-files (methods/functions)

Model Report for matrix/m

A built-in model that constructs a model of the MDC3 LT

The control matrix is defined as:

$$H = \begin{pmatrix} (1+dHdF)*HdF & 0 \\ 0 & (1+dHdu)*Hdu \end{pmatrix}$$

Model Versions	Description
LSS v2.0.0	This version is designed to mimic the c The following fit characteristics were ac ● drag free accurate to 4e-9 MSE or s^-2, input is in m ● electrostatic suspension accurate output is in kg m s^-2, input is
STOC Ex 6	This version was used in STOC Exercise The following fit characteristics were ac ● drag free accurate to 1e-8 MSE or s^-2, input is in m ● electrostatic suspension accurate output is in kg m s^-2, input is
MDC3	This version was used in MDC3. Implemented transfer functions are des

Some information of the method matrix/matrix_model	
Class name	matrix
Method name	matrix_model_H
CVS Version	\$Id: matrix_model_H.m,v 1.

LSS v2.0.0
This version is designed to mimic the controllers in LSS V
The following fit characteristics were achieved:
● drag free accurate to 4e-9 MSE on the frequency ra
● electrostatic suspension accurate to 7e-11 MSE on
s^-2, input is in m

Parameter List for version: LSS v2.0.0

Key	Default Value	Options	Description
F	[1e-05;1.04486480405217e-05;1.09174245874697e-05;1.140723270	none	a frequency vector to be used as xvals (Hz)

The screenshot shows a MATLAB help window titled 'LTPDA Toolbox' with the sub-window 'LTPDA Training Session 1'. The left sidebar contains a 'Contents' pane with a tree view including 'Release Notes', 'Installation', 'MATLAB', 'Code Generation from MATLAB', 'Control System Toolbox', 'Fixed-Point Toolbox', and 'LTPDA Toolbox'. Under 'LTPDA Toolbox', there are sub-items like 'Getting Started with the LTPDA', 'User Guide', 'Simulation/modelling', 'Built-in models of LTPDA', 'Generating model noise', 'Parameteric models', 'Statespace models', 'Transfer Function Modellin', 'Signal Pre-processing in LT', 'Signal Processing in LTPDA', 'Graphical User Interfaces in', 'Working with an LTPDA Rej', 'LTPDA Extension Modules', 'Class descriptions', and 'Functions - By Category'. The 'Functions - By Category' section is expanded to show 'LTPDA Training Session 1' with sub-items: 'Topic 1 - The basics of LTI', 'Topic 2 - Pre-processing c', 'Topic 3 - Spectral Analysis', 'Topic 4 - Transfer funcion', and 'Topic 5 - Model fitting'. The main content area displays the title 'LTPDA Training Session 1' and a description: 'This series of help pages constitute the first training session of LTPDA. The various data-packs used throughout the tutorials are available for download on the LTPDA web-site.' It lists five topics: 1. Topic 1 - The basics of LTPDA, 2. Topic 2 - Pre-processing of data, 3. Topic 3 - Spectral Analysis, 4. Topic 4 - Transfer function models and digital filtering, and 5. Topic 5 - Model fitting. Below this, it states: 'In addition, throughout the course of this training session, we will perform a full analysis of some lab data. The inputs to the analysis are two time-series data streams, the first is the recorded output of an interferometer, the second is a recording of the room temperature in the vicinity of the interferometer. Both are recorded with different sample rates and on different sampling grids. The temperature data is unevenly sampled, and may even have missing samples.' It then describes the data manipulation steps: 'During each topic of the training session, the data will be manipulated using the tools introduced in that topic (and previous topics). The aim of the data analysis is to determine the influence of temperature on the interferometer output. In particular the steps will be: 1. Topic 1 Loading and calibrating the raw data. 1. Read in the raw data files and convert them to AOs 2. Plot the two data streams 3. Calibrate the interferometer output to meters (from radians) 4. Calibrate the temperature data to degrees Kelvin from degrees Celcius 5. Save the calibrated data series to XML files, ready for the input to the next topic 2. Topic 2 Pre-processing and data conditioning. 1. Read in the calibrated AOs from XML files 2. Trim the data streams to the same time segments 3. Resample the temperature on to an even sampling grid with no missing samples 4. Resample to the two data streams to a common 1Hz sample rate 5. Interpolate the two data streams on to the same time grid 6. Save the cleaned data to AO XML files 3. Topic 3 Spectral analysis. 1. Load the time-series data from Topics 1 and 2 2. Compare PSDs of the time-series data before and after pre-processing 3. Check the coherence of temperature and IFO output before and after pre-processing.'

analysis object class constructor.

DESCRIPTION: ao analysis object class constructor.
Create an analysis object.

Possible constructors:

- a = ao() - creates an empty analysis object
- a = ao('a1.xml') - creates a new ao by loading a
- a = ao('a1.mat') - creates a new ao by loading th
- a = ao('file.txt') - creates a new ao by loading th
- a = ao('file.dat')
- a = ao('file',pl) (Set: From ASCII File)
- a = ao(data) - creates an ao with a data object
- a = ao(constant) - creates an ao from a constant
- a = ao(specwin) - creates an ao from a specwin object
- a = ao(pzm) - creates an ao from a pole/zero
- a = ao(pzm,nsecs,fs)
- a = ao(smodel) - creates an ao from a symbolic
- a = ao(pest) - creates an ao from a parameter
- a = ao(x,y) - creates an ao with xy data
- a = ao(y, fs) - creates an ao with time-series
- a = ao(x,y,fs) - creates an ao with time-series
- a = ao(x,y,pl) - creates an ao depending from t
- a = ao(plist) - creates an ao from a parameter

Examples

Parameters Description

VERSION: \$Id: ao.m,v 1.348 2011/05/16 07:15:37 hewitson Ex



home

http://www.lisa.aei-hannover.de/ltpda/

LTPDA

a MATLAB® toolbox for accountable and reproducible data analysis

home >

LTPDA is a MATLAB toolbox that uses an object-oriented approach to data analysis. LTPDA Objects are processed through a data analysis pipeline. At each analysis step, a record is kept of exactly what algorithm was applied to which object and with which parameters. In this way, the result of a particular data analysis is one or more objects, each containing the final result as numerical data together with a full processing history of how the result was achieved.

Latest version: V2.4 (requires MATLAB 2010a or above)

LTPDA includes algorithms and objects for

1. pre-processing of time-series data
2. performing spectral analysis of various kinds
3. pe
4. co
5. co
6. an

```
% Create two AOs
a = ao(1:10);
b = ao(2);

% Add them together
c = a+b;

% Set the name
c.setName('const')

% Plot the result
c.iplot
```

In addition, there is a graphical design interface which allows data analysis pipelines to be built using a diagram editor. The resulting diagram is then executed via LTPDA commands.

3

series to

<http://www.lisa.uni-hannover.de/ltpda/>

ining

t labs

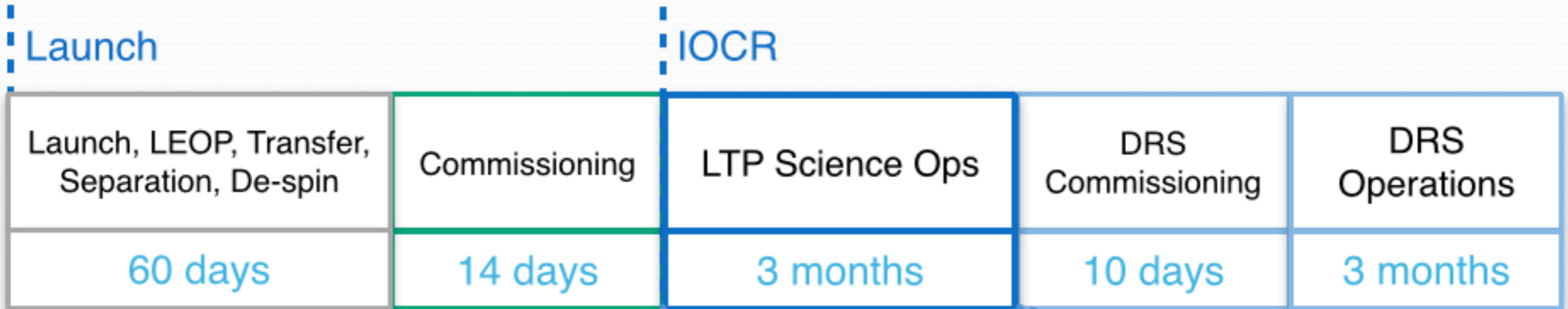
- ### Downloads
- version_2.4
 - version_2.3.1
 - version_2.3
 - version_2.3rc2
 - version_20100225
 - version_2.3rc1
 - version_2.2
 - 20091208
 - version_2.1
 - version_2.1rc3
 - 20090916
 - 20090915
 - version_2.1rc2
 - version_2.1rc1
 - version_2.0.1
 - version_2.0
 - version_2.0rc3
 - version_2.0rc2
 - version_2.0rc1
 - version_1.9.3
 - version_1.9.2
 - version_1.9.1
 - version_1.9.1 beta
 - version_1.0





- An infrastructure in which to develop ‘golden’ analyses for the STOC
- We need at least one pipeline for *every* planned investigation
- Key features:
 - capture blocks of analysis in a controlled way
 - arrange blocks into pipelines
 - maintain flexibility regarding
 - ordering of pipeline steps
 - configuration of each step
 - allow for easy controlled changes during operations

Phases of Operations



	Day 1	Day 2	Day 3	Day 4
H1	Noise Run	Discharge	Noise Run	Discharge
H2		Working Point		Stray Potentials
H3				
H4				
H5				

Week 1: Gentle Probing



- The first two weeks are all about gathering information and gaining experience
- This is our first interaction with the system
- Focus on:
 - noise runs
 - first tests of signal injection (system identification)
 - getting a handle on the charge rate and discharging

CE1	Charge estimate TM1
CE2	Charge estimate TM2
FD1	Fast Discharge TM1
FD2	Fast Discharge TM2

	Hour																								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	Noise run in Sci 1.2																								
2	CE1	CE2	Noise run in Sci 1.2																						
3	CE1	CE2	Sys ID (low amp)									Noise run in Sci 1.2													
4	CE1	CE2	Working point scan (x,y,z), both TMs																						
5	CE1	CE2	Cross-talk investigations, low amplitude																						
6	CE1	CE2	Noise run in Sci 1.2																						
7	Station Keeping											Transition Acc3 -> Sci 1.2								FD1	FD2				



What is a noise run?

- Enter nominal science mode (DFACS mode Sci. 1.2)
 - SC following TM1
 - TM2 following TM1
- Put the system in the 'best' state we know
 - discharged TMs
 - optimal dc compensation voltages
 - best test-mass working point for OMS and GRS
 - ...
- Take data for, e.g., 10 hours



- Understanding the purity of the free-fall we achieve, and what limits it, requires us to assess the residual forces acting on the TMs
 - what's left when we subtract the forces we can account for?
- We compute the relative acceleration of the two TMs based on the observed relative position
- Try to account for the contributions of g_{res} that we know
 - applied control forces
 - couplings due to force gradients

$$g_{\text{res}} = \ddot{x}_{12}$$

$$g_{\text{res}} = \ddot{x}_{12}$$

→ $-g_{\text{control}}$

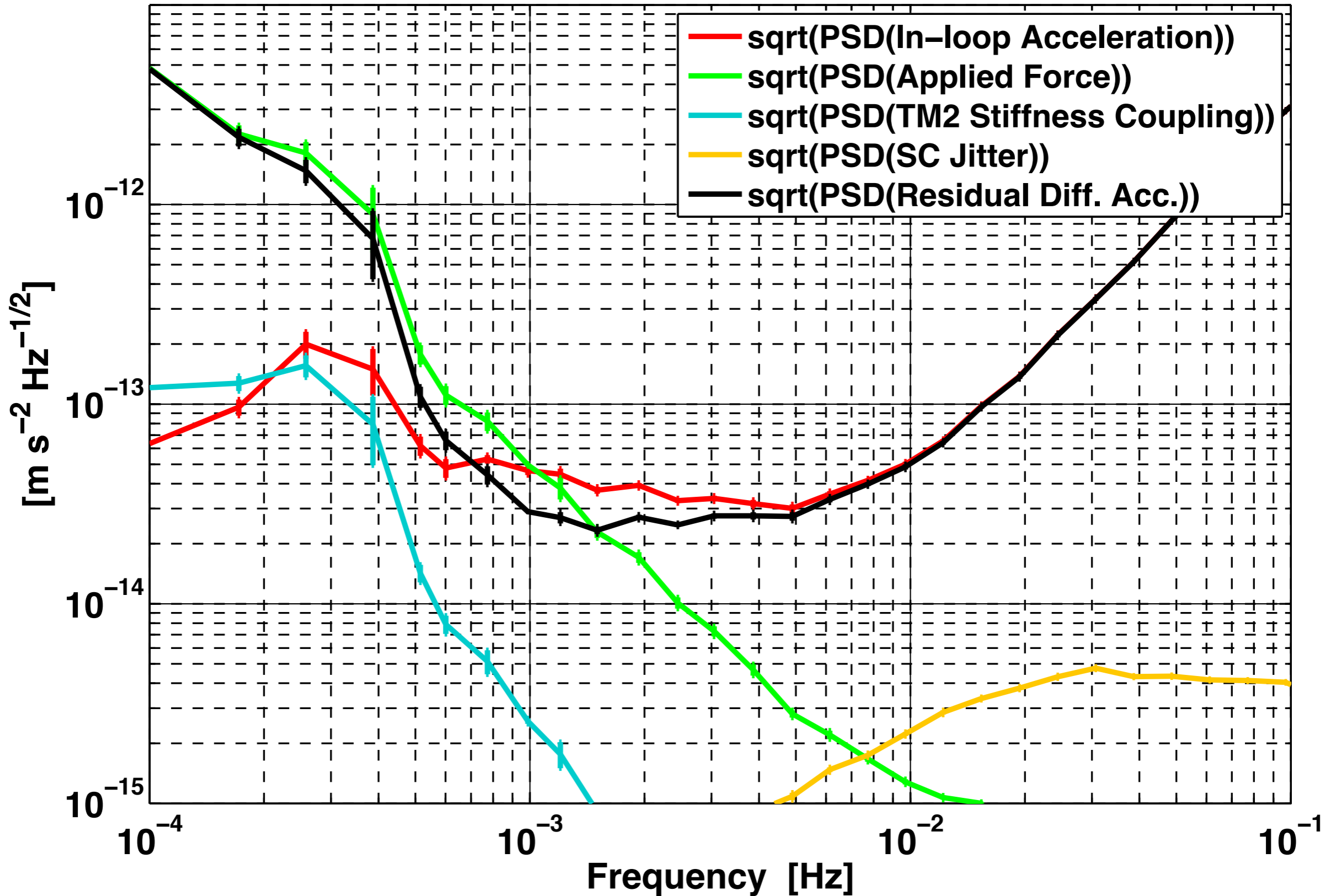
→ $-\omega_{\Delta}^2 x_1 - \omega_2^2 x_{12}$



Analysing the data

1. Download the time-series
2. Assemble the current best estimate of the required system parameters
 - actuator gains, delays, stiffnesses, ...
3. Form linear combination of the time-series
 - with delays, and filtering as necessary
4. Take spectrum of the residuals

The contributions



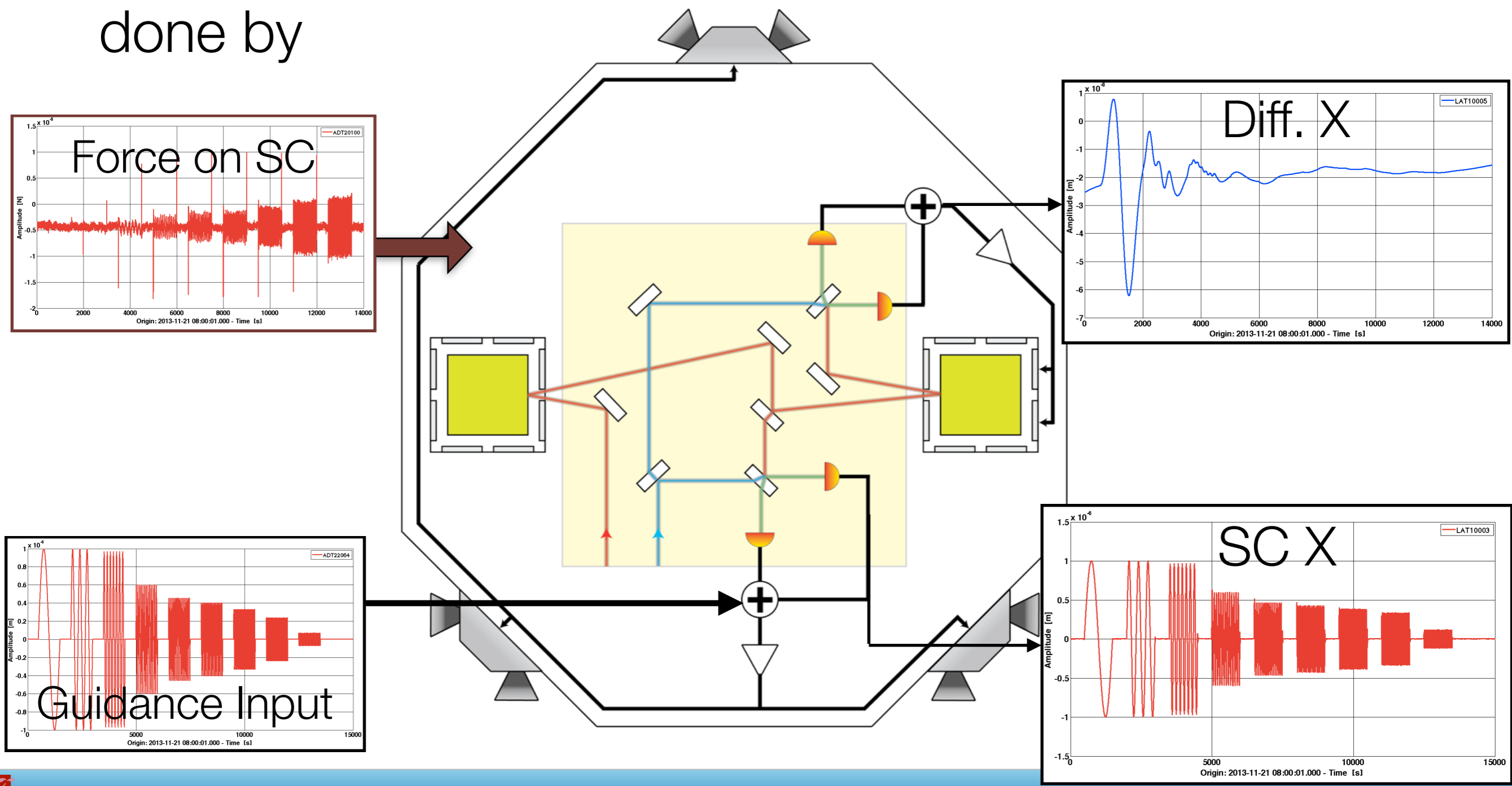


- Estimating our residual acceleration requires knowledge of certain system parameters
 - How do we gain that knowledge?
- At the beginning of operations, this comes from
 - ground measurements
 - system modelling
 - results of industrial commissioning campaign
- How do we improve and update that knowledge?
 - through dedicated investigations

x-axis system identification: part 1



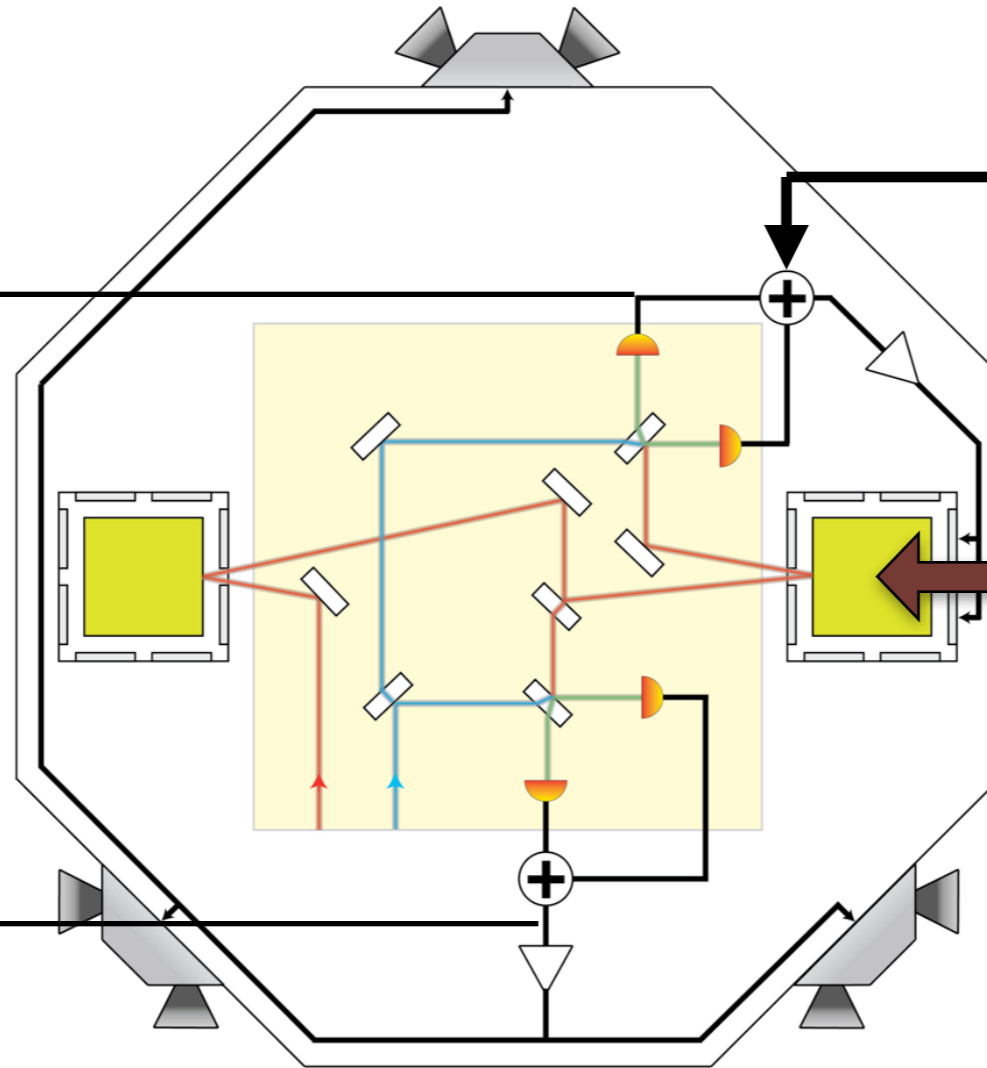
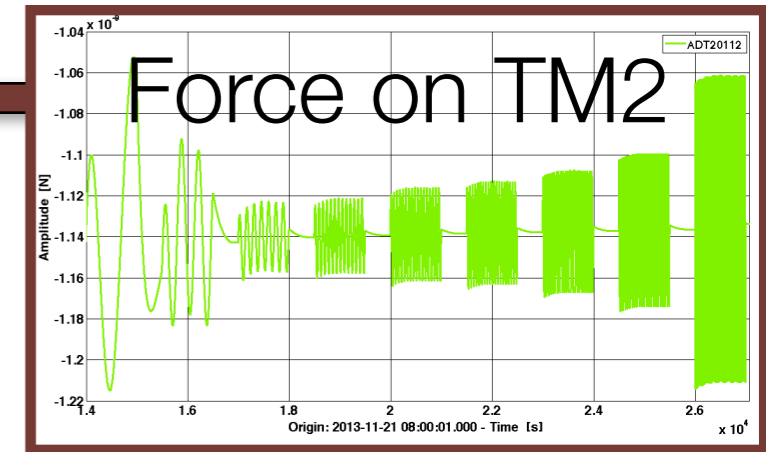
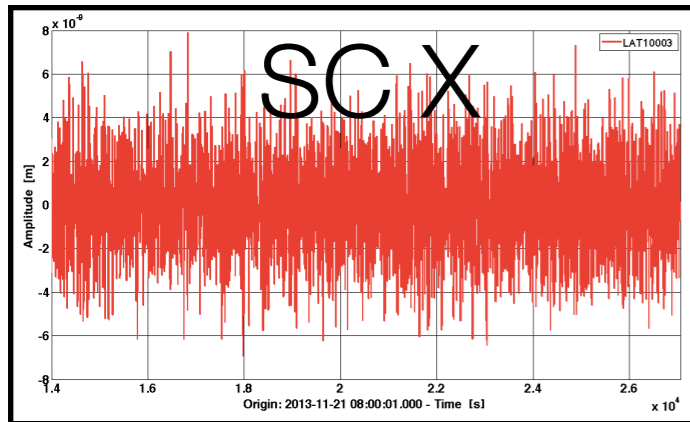
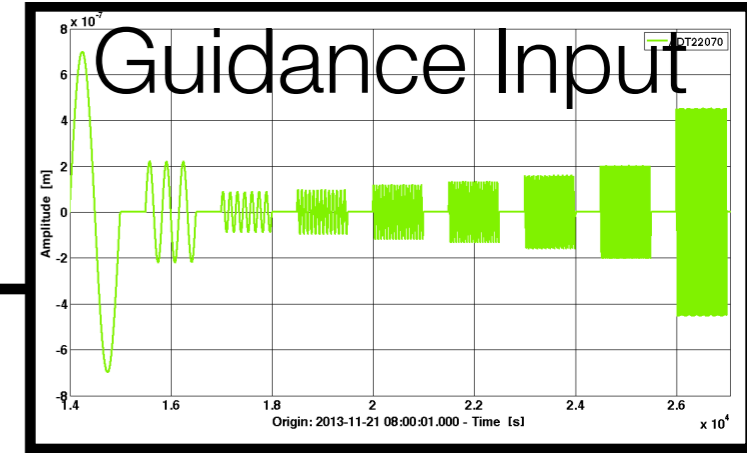
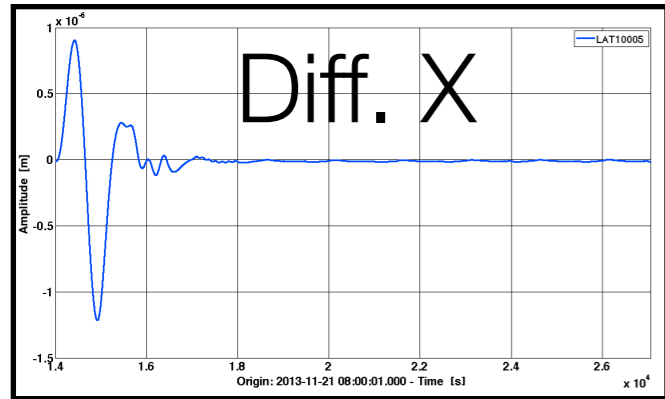
- Goal is to measure the key parameters needed for estimating the residual differential acceleration can be done by



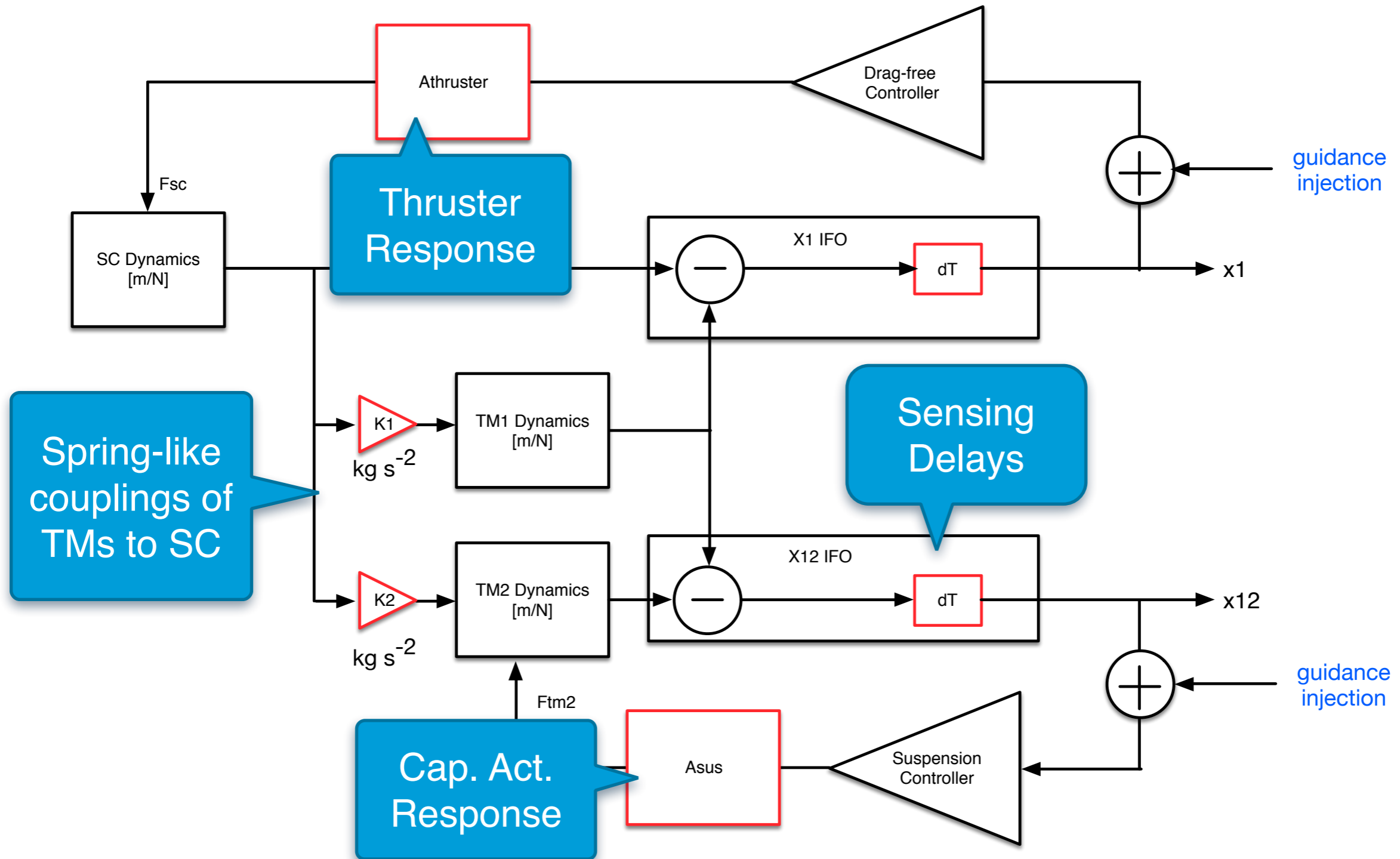
x-axis system identification: part 2



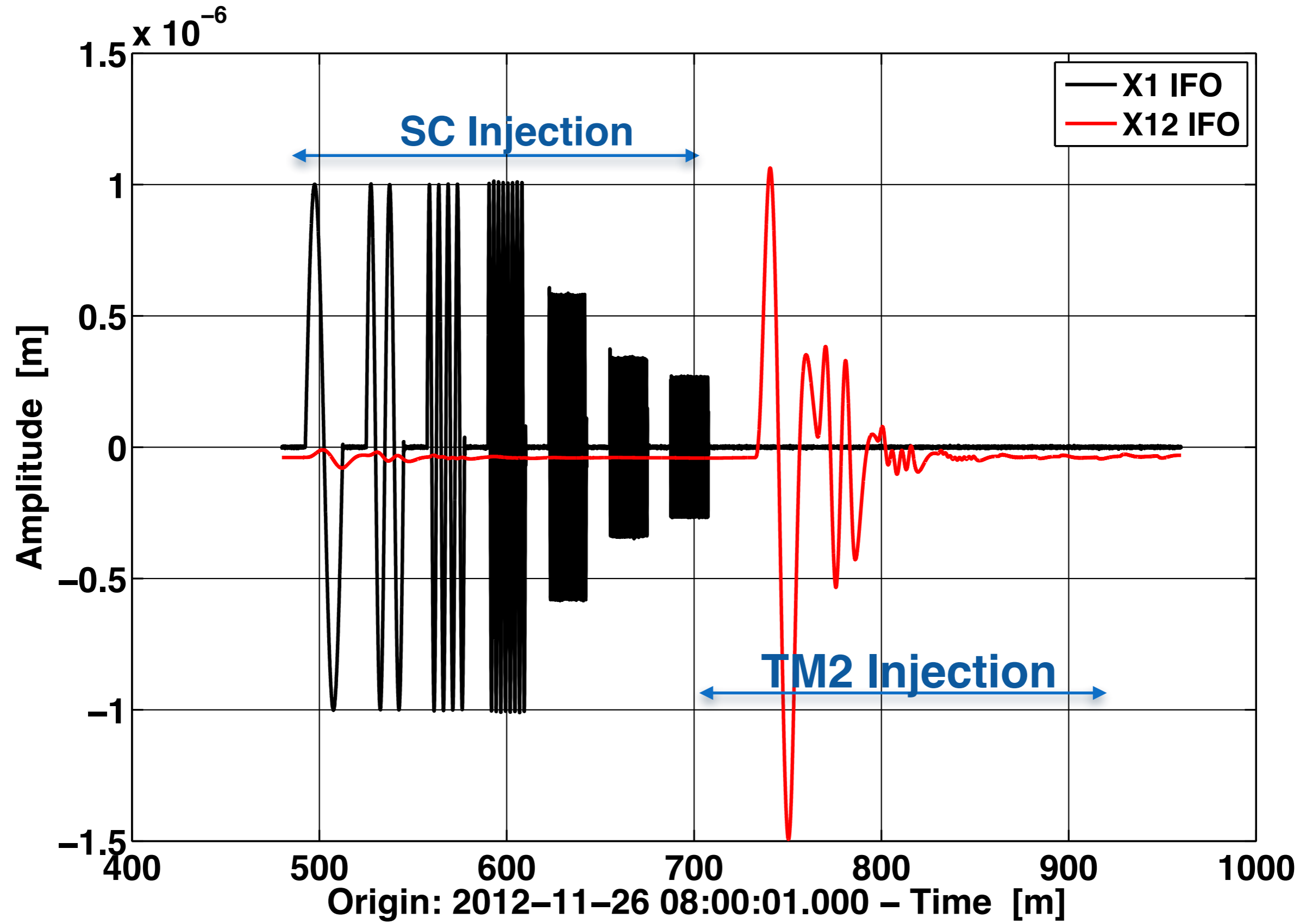
lisa pathfinder



What do we learn from that?

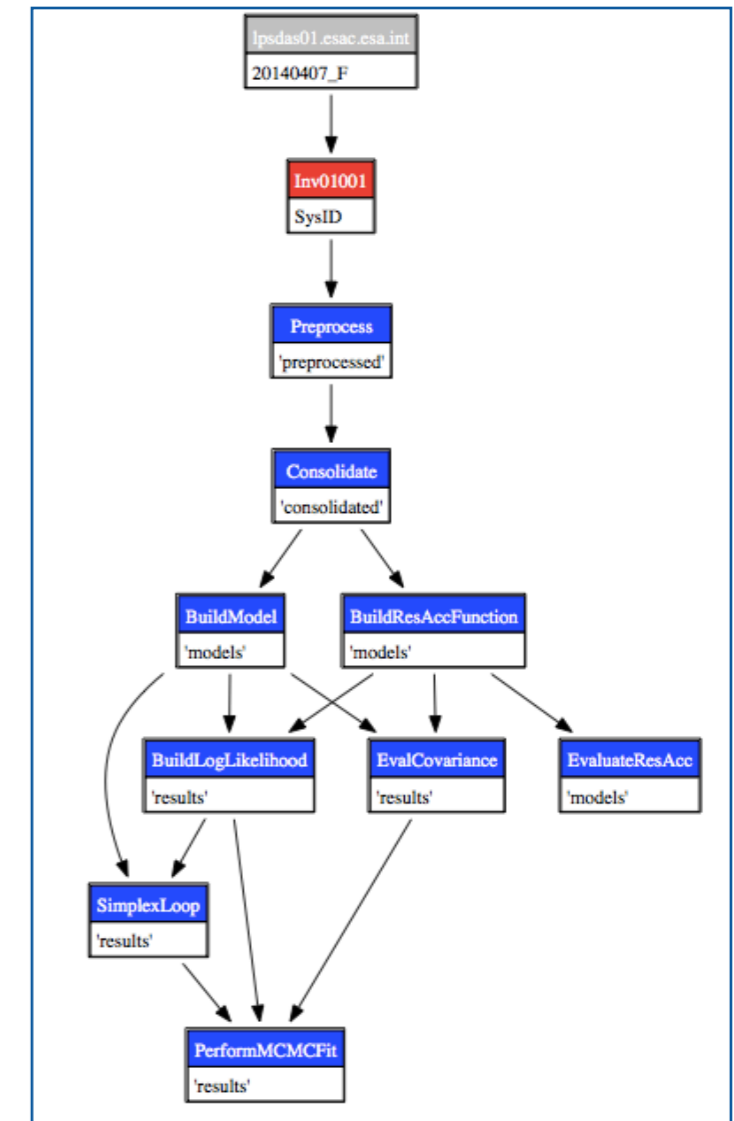


The data





- Follows the same form as for estimating residual differential acceleration
- But now the coefficients in the model are fit so that the linear combination of terms fit the observation
- When a good fit is found, the residuals contain no trace of the injected signals



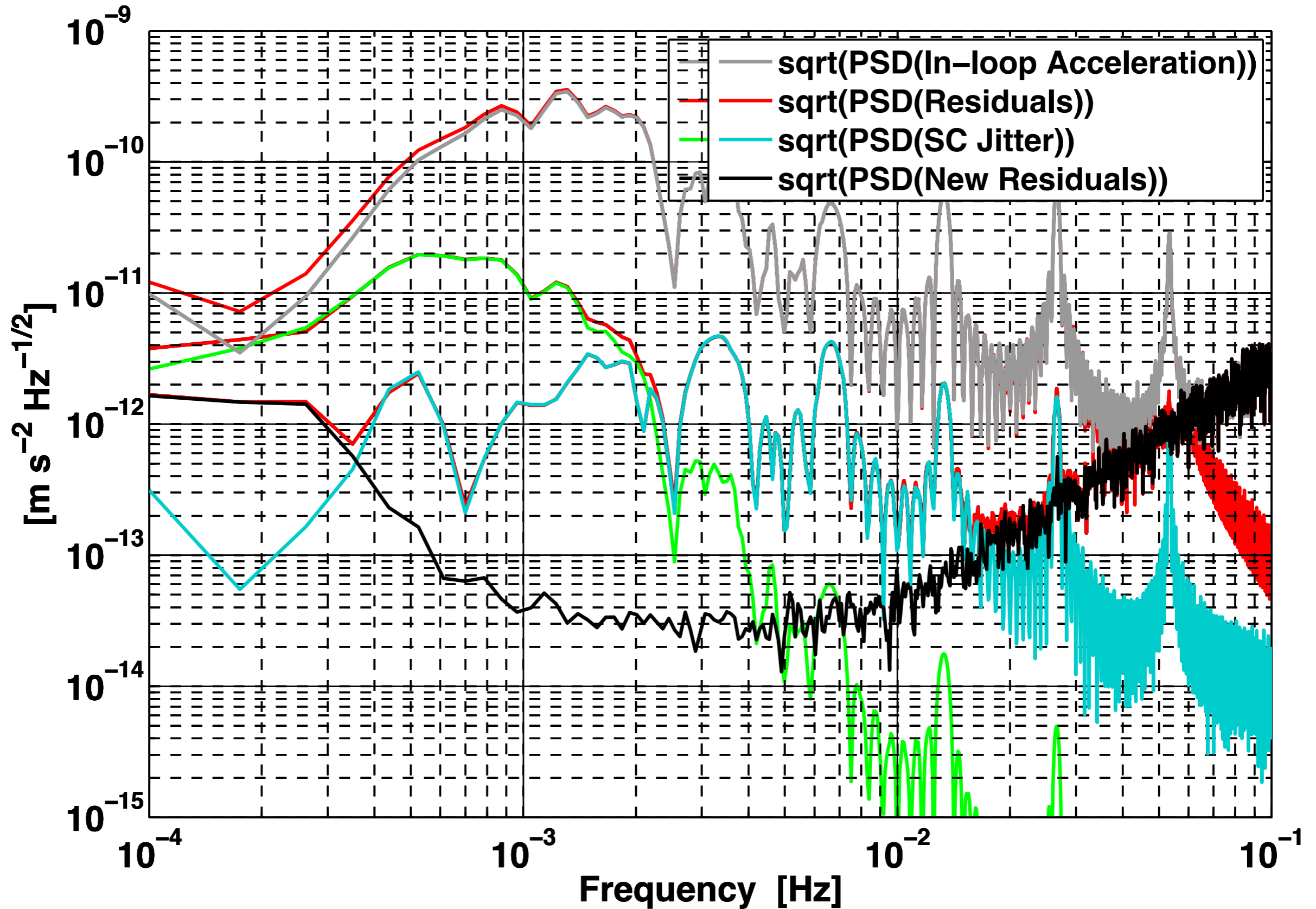
Fit

$$\text{observation} = x_{12}''[k]$$

to

$$\begin{aligned} \text{model} = & -A_{\text{sus}} F(g_{\text{control}}[k], \Delta T) \\ & -(\omega_2^2 - \omega_1^2)x_1[k] \\ & -\omega_2^2 x_{12}[k] \end{aligned}$$

Residuals





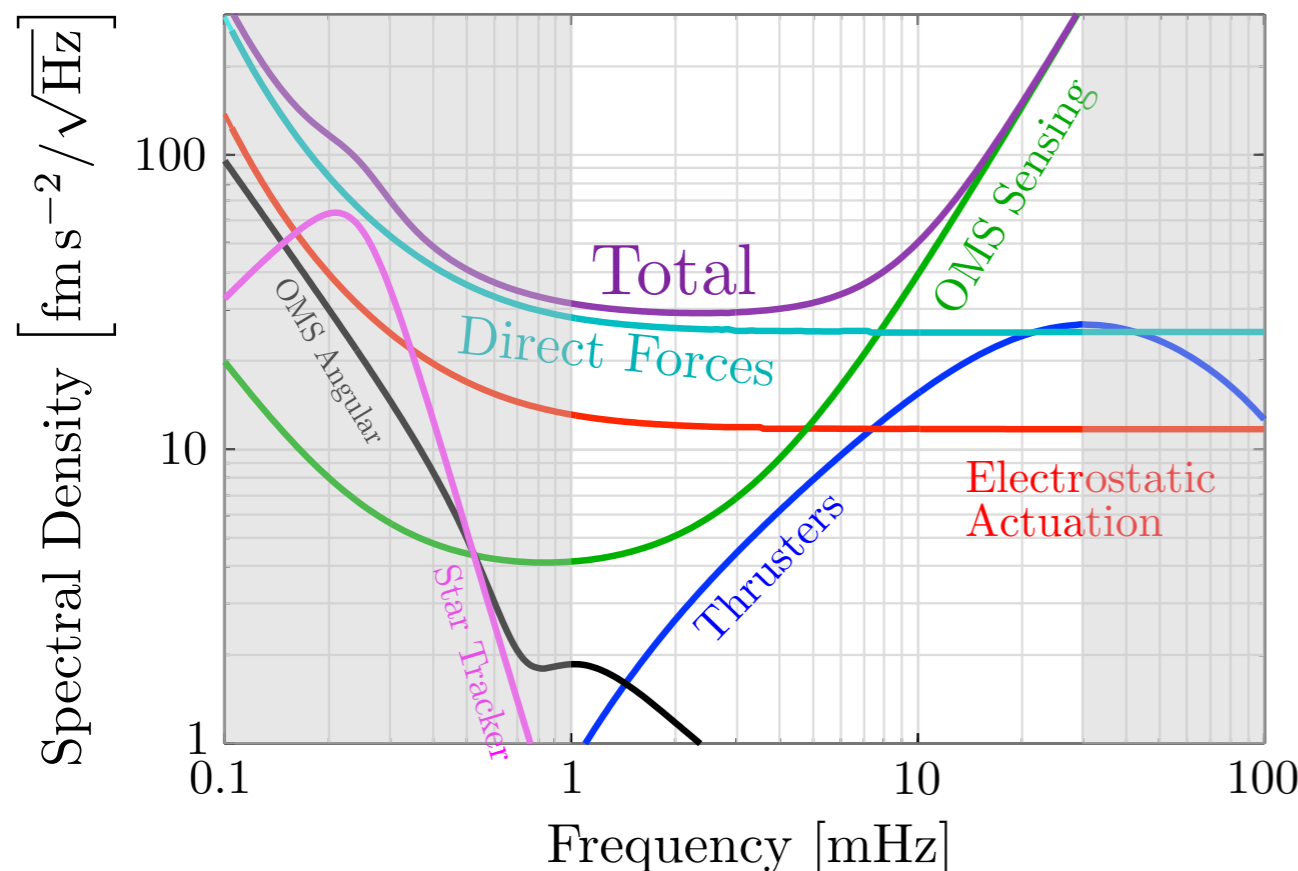
A general scheme

- **Balancing forces:**
 - improves physical modelling and interpretation
 - simplifies the analysis a great deal
- This ‘acceleration’ scheme can be used to account for other contributions
 - cross-talk
 - thermal
 - magnetic
 - free-flight experiments



- How does our observed residual differential acceleration differ from what we expect?
- Why does it differ?
 - this drives the next activities to be performed

Requirements Noise Breakdown



Current Best Estimate

