LIGO-T020014-00-W

THE NOVEMBER 2001 KNOWN PULSAR DEMODULATION MOCK DATA CHALLENGE

Gregory Mendell

LIGO Hanford Observatory

Plan started: November 3, 2001
MDC dates: November 27-30, 2001

Table of Contents

I. Overview

General Remarks

This document plans and implements the November 2001 Known Pulsar Demodulation Mock Data Challenge (KPD MDC). The software tested during this MDC is the knownpulsardemod.so.0.0.0 dynamic shared object (KPD DSO) contribution to lalwrapper, which runs in the LDAS environment. The source code is stored in the lalwrapper CVS archive. The KPD DSO is run by a series of scripts, which initiate LDAS jobs. These scripts send the LDAS jobs arguments for acquiring the data, conditioning the data, and command line arguments for the KPD DSO. The KPD MDC consists of a series of scripts designed to run LDAS jobs that test the KPD DSO. The plan is to run the MDC at LIGO Hanford Observatory (LHO).

The KPD DSO currently serves two purposes. The first is to generate short time baseline discrete Fourier transforms (SFTs) of the gravity wave channel, with appropriate meta data, and archive the SFTs either within or outside of LDAS for future use. The second is to use the LALDemod function to stitch together the SFTs, correcting for amplitude and phase modulation, to produce longer time baseline Fourier transforms (DeFTs) for known pulsars. These DeFTs will be analyzed for the presence of a gravity wave signal. The signal-to-noise ratio and other statistical measures will be calculated, and the distribution and cumulative probability associated with the statistics will be studied. The outcomes will be stored in the directed signal periodic (SNGL_DPERIODIC) database table. Other statistics and the DeFT frequency series or power spectra will go into other database tables in the future. The specifications of the SFTs are given in Appendix A. The method used to generate DeFTs is reviewed briefly in Appendix B.

The primary focus of this MDC will be to test the generation of SFTs. Only basic tests of the current version of the LALDemod function will be performed, using test data only. When the final versions of the LALDemod function is in place another MDC will be held to completely test its use. Separate tests of this function will also be held at the Albert Einstein Institute (AEI) to verify its use in a wide area pulsar search.

Key

Note: * indicates code/task to be added/performed by KPD DSO developer.
Note: ! indicates code/task to be added/performed by others.
Note: # indicates code/task that will NOT be in place for this MDC.

Current Status of the Code

1. Data Acquisition
    a. Can acquire data from an ilwd file in stand-alone mode and data sent to the mpiAPI in LDAS.
    b. Can internally generate test data of the following types:

        i. Type 0 = pure sinusoidal data with phase modulation.
        ii. Type 1 = sinusoidal + gaussian noise (mean = 0, std. dev. 1) with phase modulation.
        iii. Type 2 = input data + sinusoidal data with phase modulation.
        iv. Type 3 = pure sinusoidal data (without phase modulation).
        v. Type 4 = sinusoidal + gaussian noise (mean = 0, std. dev. 1) (without phase modulation).
        vi. # Code to internally generate test data with amplitude modulation will NOT be in place for this MDC.
    c. # After this MDC code will be added to handle data dropouts by padding gaps in input data. (The allowgaps option will be used in the LDAS dataPipeline command, and the dataconAPI fillgaps action will be used to fill the gaps with the mean value of the input data.
    d. Simple input test data exists in ilwd and frame format.
    e. Can acquire stretches of real data, with dropouts and quality information.
    f. # Code to automate the acquisition of known pulsar parameters will NOT be in place for this MDC. For now this information will be hand coded into scripts. If the number of known pulsars that upper limits are set on is small, this may be a sufficient method for doing an actual upper-limits analysis.
2. Data Conditioning
    a. Resample data to 4096 Hz. The resampling will be done by the dataconAPI resample action, which includes anti-alias filtering.
    b. The code handles time delays introduced by the resample action by overlapping each input data segment with part of the previous and next segments, and by locating the segment of interest at the correct GPS start time within the resampled input segment.
    c. # Is any other conditioning needed?
3. SFT generation.
    a. Code generates SFTs on any contiguous input data of duration greater than or equal to that needed to produce 1 SFT.
    b. Code reports progress to master wrapper while generating SFTs.
    c. Code handles quality channel information by padding input data with the mean input value during nonlocked sections of the data.
    d. # Code will handle data dropouts by padding input data with the mean input value during dropouts.
    e. Code outputs SFTs in frame format.
    f. Frame history structure contains meta data about the time series that produces the SFT, including percent of input data that was padded.
    g. Code outputs one or more SFTs per file.
    h. # Code to input SFT data will NOT be in place for this MDC. Currently all SFT data must be computed and held in memory during each LDAS job or a series of LDAS jobs.
4. DeFT generation.

   a. # Since code to input SFT data will NOT be in place for this MDC, and LDAS only allows 3600 seconds of input data per job, the LALDemod function will only be tested on segments of data less than 1 hour long.
   b. The LALDemod function is called to generate DeFTs from the SFTs for a known pulsar's sky position and spin down parameters. This removes phase modulation from the input data, placing the power of the signal in a single frequency bin.
   c. # The LALDemod function currently does not barycenter the arrival times using actual ifo and ephemeris data. This upgrade to the LALDemod code will not be in place before the MDC. Corresponding changes will then have to be made to the KPD DSO.
   d. # The LALDemod function currently does not correct for amplitude modulation due to the changing response as the source sweeps through the detector beam pattern.
   e. # The LALDemod function currently does not exactly match filter for a known pulsar frequency. (That is, it currently does not align a frequency bin with the known pulsar frequency.)

5. Statistical Analysis
   a. The code computes the mean power, ratio of detected power at the known pulsar frequency to the mean power (or closest frequency bin to this frequency), cumulative probability of getting this ratio or larger due to noise only, and signal-to-noise ratio based on simple gaussian statistics.
   b. # LALDemod function needs to compute the F statistic for polarization basis wave forms as described in Jaranowski, Krolak, and Schutz [1].
   c. # More sophisticated analysis of statistics (understanding of actual noise and distributions, etc.) is needed, but will NOT be in place for this MDC.
   d. # Code to compute upper limit on gravitation wave strength from known pulsars will NOT be in place for this MDC.
   e. # Code to compute pulsar parameters if a signal is detected from a known pulsar will NOT be in place for this MDC.
6. Database Output
   a. The code writes output from the DeFT and Statitical Analysis to the Directed Periodic Signal (SGNL_DPERIODIC) data base table.
   b. # The code will save additional statistical information and spectra in the database. (See the Database Definitions Subsection.)
   c. # The code will save the DeFT in a narrow band around the known pulsar frequency. (See the Database Definitions Subsection.)
7. Parallel Computing
   a. Code runs only on the search master.
   b. Code runs a search on one known pulsar per LDAS job.
   c. # Adding of parallel code and tests of MPI communication will NOT be in place for this MDC. Parallel code will be implemented in the future if indicated as useful by this MDC.
8. Scripts
   a. Scripts exist for running KPD DSO as an LDAS job.
   b. Scripts exist to run each test for this MDC.

Input Parameters to KPD DSO

```
BOOLEAN doDemod; Demodulate the SFTs into DeFTs
INT4 mObsCoh; Number of DeFTs to compute from input time
UINT4 gpsStartTime;  GPS start time of data requested
REAL8 deltaT;  time step in seconds = 1/sample rate
UINT4 inputN;  number of input data points
REAL8 tSFT;  duration in seconds of SFTs
INT4 nSFT; number of time domain data points used to compute one SFTs
UINT4 mCohSFTPerCall;  number SFTs to compute each call to ApplySearch
REAL8 templateRA; Right Ascension of template
REAL8 templateDec; Declination of the template
REAL8 templatePhase0; initial phase of template
REAL8 templateFreq0; initial frequency of template
REAL8 templateFreq0Band; Band around f0 to compute DeFT
REAL8 templateFreqDeriv0; initial first deriv of freq
REAL8 templateFreqSecondDeriv0; initial second deriv of freq
REAL8 templateFreqThirdDeriv0; initial third deriv of freq
REAL8 templateFreqFourthDeriv0; initial fourth deriv of freq
REAL8 templateFreqFifthDeriv0; initial fifth deriv of freq
CHAR generateTestData;  Generate test data: y = yes, n = no.
INT2 testType;  Type of test data to generate
REAL8 testRA; Right Ascension of test
REAL8 testDec; Declination of the test
REAL8 testAmplitude;  Amplitude of the test
REAL8 testPhase0; phase of test
REAL8 testFreq0;  frequency of test
REAL8 testFreqDeriv0;  first deriv of freq of test
REAL8 testFreqSecondDeriv0;  second deriv of freq of test
REAL8 testFreqThirdDeriv0;  third deriv of freq of test
REAL8 testFreqFourthDeriv0;  fourth deriv of freq of test
REAL8 testFreqFifthDeriv0;  fifth deriv freq of test
REAL8 testPolarization;  Polarization of test; NOT YET IMPLEMENTED
INT4 testDeletions; If deletions > 1 then gaps of deletions*tSFT occur
CHAR *filterID; Identifies which filter is being used (NOT USED)
CHAR *dataConDesc; Information about data conditioning done
CHAR *IFO; Identifies the interferometer name
CHAR *targetName; Identifies the target of the search
CHAR channel[dbnamelimit]; Name of input channel.
```

Test Data Types

```
0 = sinusoidal test signal only;
1 = 0 + gaussian noise (mean = 0, std dev = 1);
2 = add test signal to input data;
3 = same as 0 but without phase or amplitude modulation due to Earth's
motion;
4 = same as 2 but without phase or amplitude modulation due to Earth's
motion;
```

Database Definitions

```
SNGL_DPERIODIC Database table

ROW COLNAME TYPENAME LENGTH CODEPAGE DEFAULT NULLS REMARKS
1 CREATOR_DB INTEGER 4 0 1 N
2 PROCESS_ID CHARACTER 13 0  N
3 FILTER_ID CHARACTER 13 0  Y
4 IFO CHARACTER 2 819  N
5 START_TIME INTEGER 4 0  N
6 START_TIME_NS INTEGER 4 0  N
7 END_TIME INTEGER 4 0  N
8 END_TIME_NS INTEGER 4 0  N
9 DURATION REAL 4 0  N
10 TARGET_NAME CHARACTER 32 819  Y
11 SKY_RA REAL 8 0  N
12 SKY_DEC REAL 8 0  N
13 FREQUENCY REAL 8 0  N
14 AMPLITUDE REAL 4 0  N
15 PHASE REAL 4 0  N
16 SNR REAL 4 0  Y
17 CONFIDENCE REAL 4 0  Y
18 EVENT_ID CHARACTER 13 0  N
```

Proposed changes to the database

1. STAT_NAME (The name of a statistic computed, e.g., the “F” statistic in Jaranowski, Krolak, and Schutz [1]).
2. STAT_VALUE (The value of the statistic.)
3. STAT_PROB (The cumulative probability associated with the stat value.)

# Need to add code to put other statstitics and DeFT frequency series in the database.

Error Codes and Messages for KPD DSO

Table I.1: Error Codes and Messages

Error Code	Error Message
1	"Null pointer"
2	"mObsCoh = number of DeFTs must be 1 in current code"
3	"Unexpected GPS time interval"
4	"Invalid deltaT"
5	"Unexpected number of input data points"
6	"Invalid tSFT"
7	"Invalid right ascension"
8	"Invalid declination"
9	"Invalid phase"
10	"Invalid frequency +/- 0.5*band (could be negative, outside LIGO band, or to high for sample rate)"

11	"One of the frequency derivatives is possibly too large; frequency will evolve outside allowed band"
12	"generateTestData must be y or n"
13	"Invalid test data type"
14	"Memory allocation error"
15	"No input data was found"
16	"Invalid number of input data points"
17	"Incorrect input data time step"
18	"Incorrect input data start time"
19	"Incorrect input data start time"
20	"Invalid nSFT"
21	"nSFT and tSFT are inconsistent with deltaT"
22	"Invalid mCohSFTPerCall"
23	"Invalid or null dataCondDesc"
24	"Invalid or null IFO"
25	"Invalid or null Target Name"
26	"Invalid input data or inputN for test type 2"
27	"Invalid channel or null channel"
28	"Requested GPS start time results in invalid index to input data"

II. Goals of this MDC

1. Test that KPD DSO responds correctly to error conditions specified by its list of error codes.
2. Test output of SFTs.
    a. Test correctness of SFT output for known input.
        i. Test for impulse, step, and sinusoidal input.
        ii. Test that the KPD DSO run on LDAS produces same output as when run in stand-alone mode.
        iii. Test SFT frame output can be read using tools like FrDump.
    b. Test correctness of meta data stored in an SFT.
    c. # Test that SFT output can be read by AEI code in a future MDC.
3. Test Data Conditioning.
    a. Test that data is resampled correctly.
    b. Test that aliasing does not occur after resampling.
4. Test Data Acquisition, Output, and Storage.
    a. Test time to produce SFTs as function of number of input points.
    b. Test time to acquire real data from disk as a function of duration.
    c. Test disk space needed to store SFTs.
    d. Test that LDAS can send the results to an ftp URL.
    e. Test maximum amount of data that can be acquired by one LDAS job.
5. Test leakage due to round-off error and non-bin-centered frequencies.
    a. Test leakage due to round off error. Probably not an issue, but make sure power for sinusoidal test input with a frequency centered on a SFT frequency bin ends up in one bin.

   b. Test leakage due to non-centering on a bin. Measure how much the ratio of maximum to mean power is lowered for a frequency that is half way between two SFT frequency bins.
6. Test ability to handle data quality information and data dropout.
   a. Test that the –dbqualitychannel dataPipeline command option acquires information from the database about the lock status of the interferometer.
   b. # Test that the –allowgaps dataPipeline command and the dataconAPI fillgaps action allow gaps in the input data and fill them as requested.
7. Test the ability to produce multiple SFTs.
   a. Test output of multiple SFTs in one frame file.
   b. Test output of multiple SFTs.
8. Test output of DeFT generation.
   a. Test correctness of DeFT output for known input. Test on frequencies that are not centered on an SFT bin but are centered on a DeFT bin.
   b. Test that LALDemod corrects for phase modulation for a variety of increasing frequency derivatives.
   c. Test leakage due to round off error? (Actually, I do not think there is an issue here since LALDemod works to double precision. Thus f and f + df are distinguishable in the 0-2048 Hz band for DeFT up to 10,000 years.
   d. # Test loss in signal-to-noise due to approximations made in LALDemod. (Will defer this to the AEI MDC.)
9. Test statistical analysis.
   a. Test correctness of statistical values for known input data.
   b. # Test distributions and statistical values vary correctly when template parameters (sky position and spin down parameters) are varied for known input data.
10. Test database output.
   a. Test that correct values appear in the database table for known input data.
   b. Test that RA, Dec, and Frequency appear in the database table as REAL8.
11. Documentation Test. Test that the input parameters and error codes are documented for the KPD DSO.
12. Test the KPD DSO on real data for a real known pulsar’s parameters. The results will not mean much at this point, but this seems a satisfying way to end a MDC.

III. Tables of MDC Preparation Tasks.

Table III.1: Primary MDC Preparation Tasks

Task	Responsible	Time Estimate	Comments
Write test scripts	Greg	2 Weeks	Test script exists.
Write instructions for running tests and records of results	Greg	1 Week	See this document.
Generate internal test data and prepare files with known output.	Greg	1 day	DSO code can generate sinusoidal test data with gauss-ian noise.
Generate test data in frame file format (and ilwd format).	Greg, Stuart, PULG group, LDAS, previous MDC developers	?	Use ascii2frame for frame input and enable_dump_input in LDAS for ilwd.
Load DSO & Data	Greg	1 hour	Easy

Table III.2: Data Conditioning Tasks

Task	Responsible	Time Estimate	Comments
Identify resample and filter algorithms to use to low pass filter below 2048 Hz and resample to 4096 Hz.	Greg, PULG, LDAS, dataCon Team	Basically 0 if using dataconAPI	Have emailed PULG and dataCon team.
If needed, design filters or get response files	Greg	?	Not needed. Only need DataconAPI resample action.
Decide on use of dataconAPI vs LAL functions	Greg, PULG, dataCon Team	?	Have emailed PULG and dataCon team.
Is above the correct conditioning to do? Should filtering take place before resampling? How to handle introduced time delays?	Greg, PULG, dataCon Team	?	Have emailed PULG and dataCon team for advice. DSO code handles time delays.

Table III.3: DSO Coding Tasks

Task	Responsible	Time Estimate	Comments
Implement code changes to handle data drop outs and quality channel.	Greg and LDAS	2+ days	Ability to do this should be in next LDAS release. Will need help from LDAS to understand the calling syntax.
#Add code to store DeFT output other statistics and DeFT frequency series to the database	Greg (Consult with Peter.)	1 day	Code already outputs to SGNL_PERIODIC. Should just involve copying and modifying existing code.
#Upgrade DSO to use final LALDemod and Barycentering code.	Greg and AEI	1 day	Will consult with AEI.

Table III.4: LAL Coding Tasks (Depending on status of AEI code)

Task	Responsible	Time Estimate	Comments
# LALDemod to use final version of Barycenter code	AEI	?	In place already?
# LALDemod to handle amplitude demodulation	AEI	?	In place already?
# LALDemod to exactly match filter for a known pulsar	Greg and AEI	?	Will need to consult with AEI.
# LALDemod to compute stats based on polarization basis functions	AEI	?	For example, the “F” statistic in Jaranowski, Krolak, and Schutz [1]

Table III.5: LALWRAPPER Coding Tasks

(none)

Table III.6: LDAS Coding Tasks

Task	Responsible	Time Estimate	Comments
DataconAPI will pad gaps in data.	LDAS and Greg	Ready at next release	Will coordinate with LDAS to get this to work in DSO.
Input data structure will contain quality channel	LDAS, Greg, Ed Maros, and John Zweizig.	In current release	Will coordinate with LDAS to get this to work in DSO.
FrameAPI will handle 16 second frames.	LDAS and Greg	Ready at next release	I think almost works now, but not tested.
# LDAS will output each result in separate file.	LDAS, AEI, and Greg	Ready at next release	AEI requested 1 SFT per file. Should discuss if this is efficient.

Table III.7: Database Tasks

Task	Responsible	Time Estimate	Comments
Propose added columns to SGNL_DPERIODIC table	Greg (Will discuss and coordinate with Peter.)	?	Discussed with Peter.
Discuss writing other statistics and spectra.	Greg (Will discuss with Peter)	?	Discussed with Peter.

Table III.8: System Administration Tasks

Task	Responsible	Time Estimate	Comments
Set up as anonymous ftp server on LHO GC host lynx.	LHO SYS ADMIN (Christine And Greg)	2 hours	Should be easy.
Add directory for knowpulsar MDC to CVS archive	UWM and Greg	1 hour	Should be easy.
Add any directories on ldas system?	LDAS and Greg	1 hour	Consult with LDAS

IV. Instructions for running Tests / Records of Test Results

Staffing, Location, Versions

The KPD MDC Coordinator: Gregory Mendell

Location and Date of the MDC: ____.

Support Staff: LDAS, Caltech

LDAS Hardware and Software: Primary location will be LDAS LHO. Some test may be run on the LDAS-DEV system at Caltech and the LDAS LLO systems.

LAL software version used for this MDC: _____________________________________.

LALWRAPPER software version used for this MDC: __________________________.

LDAS software version used for this MDC: ___________________________________.

Date KPD DSO last modified: ____.

KPD MDC Setup

1. Install the version of LAL, LALWRAPPER, and LDAS tagged for this MDC at the sites, as needed.
2. Install the version of the KPD DSO for this MDC at the sites, as needed.
3. Install test data files for the KPD DSO for this MDC at the sites, as needed.
4. Check out from the ldasmdc CVS archive, or obtain a tarball distribution from the KPD MDC coordinator of the KPD MDC test scripts.
5. The primary email account and folder for all KPD MDC jobs will be: ____.
6. The output of all KPD MDC jobs will be stored at this URL: ____.

Notes:

How to run the test scripts

All the KPD MDC test scripts are run from a tcl driver script call RunJob.tclsh. Install this script, the job files (*.job), the knownpulsarimp.schema file, and test output files (ASCII*Output.txt and output_1.SFT.sav) somewhere where tclsh and Internet access is available. To get help on how to use RunJob.tclsh, run the script without any command line options.

The instructions are:

```
[gmendell@vulcan SearchCodeScripts]$ ./RunJob.tclsh

Syntax: ./RunJob.tclsh [-v] [-s <site>] [-db <database>] [-ch channel>]
[-n <name>] [-p <password>] [-e <emailaddress>] -o <logfile> [-a] [-b]
[-run] -job <jobfile>

-v     Print output to stdout.
-s     LDAS site to run job.  Allowed sites are wa, la, mit, uwm, sw,
       dev, and test
-db    LDAS database to use, e.g., lho_test, llo_test. (Set -database
       $DATABASE in job file.)
-ch    Channel name to substitute for $CHANNEL_NAME in job file.
-n     LDAS user name used to start job (required if -run option is
       given).
-p     LDAS password used to start job (required if -run option is
       given).
-e     Email address for messages from LDAS about job.
-o     Print output to specified log file.
-a     Append to an existing log file.  (Otherwise if logfile exists it
       is overwritten.)
-b     Create a backup copy the logfile upon exit. Backup name is
       logfile.timestamp
-run   Run a command as LDAS job.
-job   File that sets up the command to run.

To see the command without running it, leave out the -run option.
If the run option is not given, the command is written to the log file,
but the job is not started.  It is noted in the log file that the job
was not started
```

To save typing in the some of the options, you may hardcode any of these near the top of RunJob.tclsh (after the section where they are initialized as e.g., “NotSet”):

```
set EMAIL_ADDRESS_FOR_LDAS gmendell@ligo-wa.caltech.edu;
set SOCKETURL ldas.ligo-wa.caltech.edu;
set DATABASE lho_test.
set CHANNEL_NAME H2:LSC-AS_Q.
```

However, note that some jobs have these values hard coded in them as well! Check the job file if the values are not set as you expected.

Note that the password is replaced with ***** in the log file.

You will need to obtain a username and password from LDAS or the KPD MDC coordinator to run jobs.

Example run (username and password not given):

```
[gmendell@vulcan SearchCodeScripts]$ ./RunJob.tclsh -v -s la -db
llo_test -ch L1:LSC-AS_Q -n ***** -p ***** -o tst.log -a -run -job
kpdTime16FrameTest.job

Welcome to ./RunJob.tclsh. RESTARTED Wed Nov 21 16:23:12 PST 2001.

LOG_FILE_NAME set to tst.log.
SOURCED JOB FILE: kpdTime16FrameTest.job.
EMAIL_ADDRESS_FOR_LDAS set to gmendell@ligo-wa.caltech.edu.
SOCKETURL set to ldas.ligo-la.caltech.edu.
DATABASE set to llo_test.
CHANNEL_NAME set to L1:LSC-AS_Q.

cmd =  ldasJob { -name ligolab -password ***** -email gmendell@ligo-
wa.caltech.edu } { dataPipeline -returnprotocol file:llo_test-
SFT_Time16FrameTest-690387792-1.gwf -subject { kpdTime16FrameTest } -
resultcomment { kpdTime16FrameTest } -mddapi frame -database llo_test -
dynlib /ldcg_server/dso-test/libldasknownpulsardemod.so.0.0.0 -np 2 -
filterparams (0,1,690387792,2.44140625e-
4,65536,16,65536,1,0.0,0.0,0.0,1,0.5,0.0,0.0,0.0,0.0,0.0,'n',3,0.0,0.0,
2.0,0.0,1,0.0,0.0,0.0,0.0,0.0,0.0,1,"T","NA","T","slice") -
datacondtarget mpi -aliases gw=_ch0 -algorithms { rgw =
resample(gw,1,4); slice(rgw,65546,65536,1); } -frametype R -
interferometers L -times { 690387776-690387823 } -framequery
Adc(L1:LSC-AS_Q) }

Current time = Wed Nov 21 16:23:12 PST 2001
Starting cmd as LDAS job on ldas.ligo-la.caltech.edu.
Opened socket for SOCKETURL = ldas.ligo-la.caltech.edu.
Sent cmd to socket.
Read from socket: {1
Your job is running as: "NORMAL4838"
you will be e-mailed at: "gmendell@ligo-wa.caltech.edu"
when your job is completed, with information
on how to retrieve your results.
}

Closed socket.

Script ./RunJob.tclsh ended normally.
```

How to perform the KPD MDC

Each of the following pages gives instructions for running a test. Run the job indicated using the RunJob.tclsh driver, ftp or scp the results to the KPD MDC output URL given in the KPD MDC Setup section, perform the tasks described, and record the results, bugs, and comments. When finished, ftp or scp all log files to the KPD MDC output URL.

TEST 1 Error Codes

Purpose: Test that KPD DSO responds correctly to the error conditions specified by its list of error codes and error messages.

Tester: _______________. Date & Time: ___________. Tester Location: ____________.

Job Site: _________________________. Job Database: ____________________________.

Job Channel: _____________________. Job Log File: _______________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

Use RunJob.tclsh to run the jobs below. Verify that the jobs return the given error code and message.

Job	Error Code	Error Message	**LDAS Job # Pass/Fail**
KpdBadmObsSFTPerCallTest.job	2	"mObsCoh = number of DeFTs must be 1 in current code"	
KpdBaddeltaTTest.job	4	"Invalid deltaT"	
KpdBadtSFTTest.job	6	"Invalid tSFT"	
KpdBadRATest.job	7	"Invalid right ascension"	
KpdBadDecTest.job	8	"Invalid declination"	
KpdBadFreqTest.job	10	"Invalid frequency +/- 0.5*band (could be negative, outside LIGO band, or too high for sample rate)"	
KpdBadFreqDerivTest.job	11	"One of the frequency derivatives is possibly too large; frequency will evolve outside allowed band"	
KpdBadyOrnTest.job	12	"generateTestData must be y or n"	
KpdBadTestTypeTest.job	13	"Invalid test data type"	
KpdBadGPSTimeTest.job	18	"Incorrect input data start time"	
KpdBadnSFTTest.job	20	"Invalid nSFT"	
KpdBadnSFTtSFTdeltaTTest.job	21	"nSFT and tSFT are inconsistent with deltaT"	

KpdBadmCohSFT PerCallTest.job	22	"Invalid mCohSFTPerCall"	
KpdBadGPSTime Test.job	28	"Requested GPS start time results in invalid index to input data"	

Known faults (list problem report numbers):
____
____
____.

New faults (list problem report numbers):
____
____
____.

Comments:

TEST 2a Correctness of SFT output

Purpose: Test that the SFTs output by the KPD DSO are indeed the DFT of the input data for input data with known results.

Tester: ______________. Date & Time: __________. Tester Location: ___________.

Job Site: ________________________. Job Database: ___________________________.

Job Channel: ____________________. Job Log File: _____________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

Use RunJob.tclsh to run the jobs below, perform the task indicated, and record the results.

1. Impulse tests. These test run on KPDTEST-ImpulseN32I16-600000000-1.gwf, which contains 32 data points sampled at 32 Hz, with an impulse in the 16th data point (index = 15).
   a. Run kpdImpulseTest.job. The output will be an xml file. Ftp the result to the KPD MDC output URL. Append test.2a.1a to the name. Check that the results agree with that in ASCII_ImpN32I16Output.txt.
   LDAS Job #: **Pass/Fail**

   b. Run kpdImpulseFrameTest.job. The output will be a frame file. Ftp the result to the KPD MDC output URL. Append test.2a.1b to the name. Use FrDump to check that the results agree with that in ASCII_ImpN32I16Output.txt.
   LDAS Job #: **Pass/Fail**

   c. Run the KPD DSO in stand-alone mode on the schema file: knownpulsardemod1perCall.schema and data file wrapperImpInput.ilwd. Rename output_1.ilwd to output_1.ilwd.test.2a.1c and ftp this to the KPD MDC output URL. Check that the results agree with that in ASCII_ImpN32I16Output.txt.
   LDAS Job #: **Pass/Fail**

2. Step test. This test runs on KPDTEST-StepN16S7-600000000-1.gwf, which contains 16 data points sampled at 16 Hz, with a step that runs for the first 7 data points (index 0 to 6). Run kpdStepTest.job. Ftp the result to the KPD MDC output URL. Append test.2a.2 to the name. Check that the results agree with that in ASCII_StepN16S7Output.txt.
   LDAS Job #: **Pass/Fail**

3. Sinusoid test. This test runs on KPDTEST-CosN16F2P0-600000000-1.gwf, which contains 16 data points sampled at 16 Hz for a cosine wave with a frequency of 2 Hz. Run kpdCosN16F2P0.job. Ftp the result to the KPD MDC

output URL. Append test.2a.3 to the name. Check that the output consists of all zeros (to single precision), except amplitude = 0.5 in the real part of the third frequency bin, corresponding to the 2 Hz frequency bin.
LDAS Job #: **Pass/Fail**

Known faults (list problem report numbers):
____
____
____.

New faults (list problem report numbers):
____
____
____.

Comments:

TEST 2b Correctness of SFT meta data

Purpose: Test that the SFT meta data output by the KPD DSO is indeed correct for input data with known results.

Tester: ______________. Date & Time: __________. Tester Location: ___________.

Job Site: _______________________. Job Database: __________________________.

Job Channel: ___________________. Job Log File: ____________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

Obtain the results from Test 2a.1b. The output will be a frame file. Use FrDump or /ldas/ldas-0.0/bin/frdump to check each of the following metadata items are set correctly. Use the numbers in ASCII_ImpN32I16Output.txt to verify results than need calculation.

1. The data set name is sft_output.

   Pass/Fail
2. The sampleRate and fShift in the proc data structure are –1 or 0. (These fields are not used in this structure for a frequency series.)

   Pass/Fail
3. ndata = number of data points is 17.

   Pass/Fail
4. The unitY is adc count per root hertz.

   Pass/Fail
5. startX = start frequency = fMin = 0.0.

   Pass/Fail
6. dx = frequency step size = 1, xunit = Hz.

   Pass/Fail
7. The start time in the proc data structure is 600000000 sec, 0 nansec.

   Pass/Fail
8. In the history structure the percent padding is -1.

   Pass/Fail
9. In the history structure the maximum power is correct.

   Pass/Fail
10. In the history structure the frequency associated with the maximum power is correct.

    Pass/Fail
11. In the history structure the mean power is correct.

    Pass/Fail
12. In the history structure the standard deviation of the power is correct.

    Pass/Fail
13. In the history structure the GPS start time is 600000000 sec, 0 nansec.

    Pass/Fail

14. In the history structure the sample rate is 32 Hz.

Pass/Fail

15. In the history structure the number of input data points is 32.

Pass/Fail

Known faults (list problem report numbers):

____

____

____.

New faults (list problem report numbers):

____

____

____.

Comments:

TEST 3 Test Data Conditioning Resample Action

Purpose: Test that the dataconAPI resample action works and that aliasing does not occur after resampling.

Tester: _______________. Date & Time: ___________. Tester Location: ____________.

Job Site: _________________________. Job Database: ____________________________.

Job Channel: _____________________. Job Log File: _______________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

These test run on KPDTEST-CosN192S64F2P0-599999999-3.gwf, KPDTEST-CosN192S64F14P0-599999999-3.gwf, and KPDTEST-CosN192S64F62P0-599999999-3.gwf. These files contain 3 seconds of sinusoidal data for frequencies of 2, 14, and 62 Hz, sampled at 64 Hz. The datconAPI resample action is used to resample this data to 16 Hz. (The test here down samples the data by a factor of 4. During actual runs of the KPD DSO the data will be down sampled from 16834 Hz to 4096 Hz.) The resample action includes anti-alias filtering. This introduces a time delay in the signal. The jobs in this test output the SFT of the resampled data. They test that the KPD DSO code correctly handles the time delay, by seeking the start time of 600000000 seconds in the resampled data, and that aliasing does not occur in the resampled data.

a. Run kpdResamplep1q4CosN192S64F2P0.job. This job runs on the 2 Hz input. Ftp the result to the KPD MDC output URL. Append test.3a to the name. Check that the resulting SFT corresponds to a start time of 600000000 seconds, 0 nanoseconds, has 9 frequencies, with df = 1 Hz, consists of all zeros (to single precision) except the real part of the third frequency bin (the 2 Hz bin) should be 0.5, and was generated from 16 data points sampled at 16 Hz.
LDAS Job #: **Pass/Fail**

b. Run kpdSlicestride4CosN192S64F14P0.job. This job runs on the 14 Hz input, which is aliased into the 2 Hz bin when data is sampled at 16 Hz, unless anti-aliasing filtering is done. This job does not have the dataconAPI do the resample action, but instead uses the slice action to decimate the data by a factor of 4 <u>without</u> anti-alias filtering! Ftp the result to the KPD MDC output URL. Append test.3b to the name. Check that the resulting SFT corresponds to a start time of 600000000 seconds, 0 nanoseconds, has 9 frequencies, with df = 1 Hz, consists of all zeros (to single precision) except the real part of the third frequency bin (the 2 Hz bin) should be 0.5, and was generated from 16 data points sampled at 16 Hz. This test shows that aliasing occurs in the input data is just sliced (decimated).
LDAS Job #: **Pass/Fail**

c. Run kpdResamplep1q4CosN192S64F14P0.job. This job runs on the 14 Hz input. Ftp the result to the KPD MDC output URL. Append test.3c to the name. Check

that the resulting SFT corresponds to a start time of 600000000 seconds, 0 nanoseconds, has 9 frequencies, with df = 1 Hz, consists of zeros (to 0.1%) <u>in every bin</u> and was generated from 16 data points sampled at 16 Hz. This job shows that the 14 Hz signal that was aliased into the 2 Hz bin in test.3b has been removed by the anti-alias filtering of the dataconAPI resample action.
LDAS Job #: **Pass/Fail**

d. Run kpdResamplep1q4CosN192S64F62P0.job. This job runs on the 64 Hz input, which is aliased into the 2 Hz bin when data is sampled at 64 Hz. Ftp the result to the KPD MDC output URL. Append test.3d to the name. Check that the resulting SFT corresponds to a start time of 600000000 seconds, 0 nanoseconds, has 9 frequencies, with df = 1 Hz, consists of all zeros (to 0.1%) except the real part of the third frequency bin (the 2 Hz bin), and was generated from 16 data points sampled at 16 Hz. This test shows that the anti-aliasing filtering done by the dataconAPI resample action cannot removed an aliased signal in the input data. This is reasonable, since the dataconAPI resample action (or any action) cannot distinguish between genuine and aliased signals in the input data, since by definition these are indistinguishable. The dataconAPI resample action can only prevent further aliasing from occurring when it resamples.
LDAS Job #: **Pass/Fail**

Known faults (list problem report numbers):
____
____
____.

New faults (list problem report numbers):
____
____
____.

Comments:

TEST 4 Test Data Acquisition, Output, and Storage

Purpose: Test that the frameAPI can acquire data of at least 2048 seconds. Test the time to acquire the data and generate SFTs as a function of the number of data points. Test the disk space needed to store SFTs as a function of the number of data points. Test that LDAS can send the results to an ftp URL. Test the maximum amount of data that can be acquired by one LDAS job.

Tester: _______________. Date & Time: ___________. Tester Location: ____________.

Job Site: _________________________. Job Database: _____________________________.

Job Channel: _____________________. Job Log File: _______________________________.

Instructions: (See the “How to run the test scripts” section if you need help.)

1. Run the jobs kpdTime16FrameTest.job, kpdTime128FrameTest.job, kpdTime512FrameTest.job, and kpdTime2048FrameTest.job. Ftp the results to the KPD MDC output URL. Append test.4.1.16, test.4.1.128 test.4.1.512, and test.4.1.2048 to the names, respectively. Note that the input data is sampled at 16384 Hz, and then resampled to 4096 Hz in the dataconAPI.

   a. Record the time spent in the frameAPI for each job in this table

N/16384	frameAPI T (seconds)
16	
128	
512	
2048	

   Fit the data to the formual T = a + bN: _________________________.
   LDAS Job #’s: **Pass/Fail**

   b. Record the time spent in the dataconAPI for each job in this table

N/16384	dataconAPI T (seconds)
16	
128	
512	
2048	

   Fit the data to the formual T = a + bN: _________________________.
   LDAS Job #’s: **Pass/Fail**

c. Record the time spent in the mpiAPI for each job in this table

N/4096	mpiAPI T (seconds)
16	
128	
512	
2048	

Fit the data to the formula $T = a + bN + cN\log_2 N$: ____________________.

LDAS Job #'s: **Pass/Fail**

d. Record the total time spent for each job in this table

N/4096	Total T (seconds)
16	
128	
512	
2048	

Fit the data to the formula $T = a + bN + cN\log_2 N$: ____________________.

LDAS Job #'s: **Pass/Fail**

e. Record the size of each output file in this table

N/4096	Total S (Megabytes)
16	
128	
512	
2048	

Fit the data to the formula $S = a + bN$: ____________________.

LDAS Job #'s: **Pass/Fail**

2. The jobs kpdImpulseFTPTest.job and kpdImpulseFTPFrameTest.job test that LDAS can send results to an ftp URL.

a. Run kpdImpulseFTPTest.job. Verify that output the shows up at the URL ftp://lynx.ligo-wa.caltech.edu/pub/incoming/H-SFT_ImpulseFTPTest-600000000-1.xml. Check that the file is not corrupt (it contains the same impulse output as from test 2a.1a).
LDAS Job #: **Pass/Fail**

b. Run kpdImpulseFTPFrameTest.job. Verify that output the shows up at the URL ftp://lynx.ligo-wa.caltech.edu/pub/incoming/H-

SFT_ImpulseFTPFrame-600000000-1.gwf. Check that the file is not corrupt (it contains the same impulse output as from test 2a.1b).
LDAS Job #: **Pass/Fail**

3. Run kpdTime4096FrameTest.job. This job should fail since it asks for 4096 seconds of data (more than 1 hour). Verify that it does fail for this reason, and record the error message that LDAS returns.
____.
LDAS Job #: **Pass/Fail**

Known faults (list problem report numbers):
____
____
____.

New faults (list problem report numbers):
____
____
____.

Comments:

TEST 5 Power Leakage Tests

Purpose: Verify that power is not leaked into nearby bins due to round-off error. Test leakage due to non-centering on a bin. Measure how much the ratio of maximum to mean power is lowered for a frequency that is half way between two SFT frequency bins.

Tester: ______________. Date & Time: __________. Tester Location: ___________.

Job Site: _______________________. Job Database: __________________________.

Job Channel: ___________________. Job Log File: ____________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

1. Run kpdSFTTest3T2048SFT1A2F2000.job. This runs the KPD DSO code on T = 2048 seconds of internally generated 2000 Hz sinusoidal data, sampled at f_s = 4096 Hz with amplitude A = 2. Ftp the result to the KPD MDC output URL. Append test.5.1 to the name. Verify that all the power (to single precision) ends up in the 2000 Hz bin. The power in this bin should be $P = 0.25 A^2T$, the mean power should be $P/(N/2 + 1)$, where $N = f_s T$ = number of input data points, and the standard deviation of the power should be $(A^4T/8f_s)^{1/2}$ for large N. This verifies that power is not leaked due to round-off error.
LDAS Job #: **Pass/Fail**

2. Repeat test 5.1 for kpdSFTTest3T2048SFT1A2F2000PPiby3.job. This runs the KPD DSO code on the same 2000 Hz data as test 5.1, but with the phase shifted by $\pi/3$. Append test5.2 to the result name. Verify that a phase shift gives the same maximum power, frequency associated with maximum power, mean power, and standard deviation of the power (to single precision) as test 5.1.
LDAS Job #: **Pass/Fail**

3. Run kpdSFTTest3T2048SFT1A2F2000-0.5df.job. This code runs on 2048 seconds of internally generated 2000.000244140625 Hz sinusoidal data, sampled at 4096 Hz. This frequency is half way between the 2000 Hz and 2000.00048828125 Hz frequency bins. Ftp the result to the KPD MDC output URL. Append test.5.3 to the name. Measure the ratio of maximum power for job 5.1 and for this job (5.3). The power should be reduced by $4/\pi^2$. Result P_{max}/P_{mean} job 5.1 = ________.; P_{max}/P_{mean} job 5.3 = ________.
LDAS Job #: **Pass/Fail**

Known faults (list problem report numbers):

____

New faults (list problem report numbers):

____

Comments:

TEST 6 Test Data Quality Channel and Dropout Handling

Purpose: Test that the –dbqualitychannel dataPipeline command option acquires information from the database about the lock status of the interferometer. Test that the –allowgaps dataPipeline command and the dataconAPI fillgaps action allow gaps in the input data and fills them as requested.

Tester: ______________. Date & Time: __________. Tester Location: ___________.

Job Site: ________________________. Job Database: ___________________________.

Job Channel: ____________________. Job Log File: _____________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

1. Run kpdQualPassTest.job. Verify that the job finishes successfully. This verifies that the quality channel can be successfully passed to the mpiAPI using the –dbqualitychannel dataPipeline command with the pass option.
   LDAS Job #: **Pass/Fail**

2. Run kpdQualPushTest.job. Verify that the job finishes successfully. This verifies that the quality channel can be successfully pushed to the dataconAPI using the –dbqualitychannel dataPipeline command with the push option.
   LDAS Job #: **Pass/Fail**

3. # Run kpdAllowGapsTime16Test.job at a site where data can be accessed by time. This job requests 48 seconds of data, with a 16 second gap in the middle. Verify that the job finishes successfully. This verifies that the -allowgaps dataPipeline command works. Edit a copy of the job and try adding the dataconAPI fillgaps action to the algorithms section. Run the job again, and verify that the fill gaps action works. Once the jobs finish successfully have LDAS system administration set enable_dump_input to FALSE in LDASwrapper.rsc on the Beowulf server at the site. Rerun the jobs and ftp wrapperInput.ilwd from the job files. Rename this wrapperAllowgapsInput.ilwd and store this in the KPD MDC output URL. Store the job that works at this URL as well. Have LDAS system administration set enable_dump_input to FALSE.
   LDAS Job #: **Pass/Fail**

Known faults (list problem report numbers):

____

New faults (list problem report numbers):

____

Comments:

TEST 7 Test Multiple SFT Output

Purpose: Test output of multiple SFTs in one frame file.

Tester: ______________. Date & Time: __________. Tester Location: ___________.

Job Site: ________________________. Job Database: ___________________________.

Job Channel: ____________________. Job Log File: _____________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

Run kpdImpulseFrame8Test.job. Ftp the result to the KPD MDC output URL. Append test.7.1 to the name. This job outputs a frame file with 8 SFTs, each computed on .125 seconds of 1 second of impulse data sampled as 32 Hz, with the impulse in the 16th data element (index = 15). Verify that the SFTs contain only zeros (to single precision) except for the 4th SFT.

LDAS Job #: **Pass/Fail**

Known faults (list problem report numbers):
____
____
____.

New faults (list problem report numbers):
____
____
____.

Comments:

TEST 8 Test DeFT Output

Purpose: Test correctness of DeFT output for known input. Test on frequencies that are not centered on an SFT bin, but are centered on a DeFT bin. Test that LALDemod corrects for phase modulation for a variety of increasing frequency derivatives. Verify that leakage does not occur due to round-off error.

Tester: ______________. Date & Time: __________. Tester Location: ___________.

Job Site: _______________________. Job Database: __________________________.

Job Channel: ___________________. Job Log File: ____________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

1. Run kpdDemodTest0T16SFT4A2F300.job. This job divides T = 16 seconds of internally generated 300 Hz sinusoidal data with amplitude A = 2 into 4 SFTs, then stitches these together to produce 1 DeFT. Ftp the result to the KPD MDC output URL. Append test.8.1 to the name. Verify that the output (which is an entry for the SGNL_DPERIOD database table) finds the frequency at 300 Hz with output amplitude $0.5AT^{1/2}$.
LDAS Job #: **Pass/Fail**

2. Run kpdDemodTest0T2048SFT2A1F1000-0.5df.job. This job divides T = 2048 seconds of internally generated 1000.00048828125 Hz sinusoidal data with amplitude A = 1 into 2 SFTs, then stitches these together to produce 1 DeFT. This frequency is not centered on a SFT frequency bin, but is centered on a DeFT frequency bin. Ftp the result to the KPD MDC output URL. Append test.8.2 to the name. Verify that the output (which is an entry for the SGNL_DPERIOD database table) finds the frequency at 1000.00048828125 Hz with output amplitude $0.5AT^{1/2}$.
LDAS Job #: **Pass/Fail**

3. Run kpdDemodTest0T2048SFT4A1F300-0.5dfD1e-10.job, kpdDemodTest0T2048SFT4A1F300-0.5dfD1e-8., kpdDemodTest0T2048SFT4A1F300-0.5dfD1e-3., and kpdDemodTest0T2048SFT4A1F300-0.5dfD1e-1. These jobs modulate the frequency of the signal at ever increasing spin down rates (the frequency derivatives range from 1.e-10 to 1.e-1). The LALDemod function should restore the signal to a single frequency bin, to within its approximations. Ftp the result to the KPD MDC output URL. Append test.8.3.1e-10, test.8.3.1e-8, test.8.3.1e-3, test.8.3.1e-1 to the result names respectively. Verify that the output (which is an entry for the SGNL_DPERIOD database table) finds the frequency at 300.00048828125 Hz.
LDAS Job #'s: **Pass/Fail**

4. The above jobs run on data with no noise. Thus the SNR should be undefined for these jobs. Verify that the SNR is undefined for these jobs, or due to round-off error only. If it

is not undefined this either indicates that round-off error or truncation error due to approximations made in the LALDemod function introduce numerical noise into the calculation.

Pass/Fail

Known faults (list problem report numbers):

____

____

____.

New faults (list problem report numbers):

____

____

____.

Comments:

TEST 9 Test Statistical Analysis

Purpose: Test that the KPD DSO produces correct statistical output for known input.

Tester: ______________. Date & Time: __________. Tester Location: ___________.

Job Site: ________________________. Job Database: ___________________________.

Job Channel: ____________________. Job Log File: _____________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

Run the jobs kpdDemodTest1T16SFT4A2F300.job and kpdDemodTest1T2048SFT4A2F300.job. These jobs add zero mean gaussian noise with standard deviation of 1 to a 2cos(2π300t) signal, sampled at 4096 Hz, for T = 16 and 2048 seconds of data respectively. Ftp the result to the KPD MDC output URL. Append test.9.T16 and test.9.T2048 to the output files respectively. Check that the frequency, amplitude, phase, SNR, and confidence are computed correctly for this data. Check that the SNR increases with the square root of T. (The tests described here will require some knowledge of how the KPD DSO works. Consult with the KPD MDC coordinator.)

Pass/Fail

Known faults (list problem report numbers):

____

____

____.

New faults (list problem report numbers):

____

____

____.

Comments:

TEST 10 Test Database Output

Purpose: Test that KPD DSO produces correct output for the database. Verify that this data is ingested into the database correctly.

Tester: ______________. Date & Time: __________. Tester Location: ___________.

Job Site: ________________________. Job Database: ___________________________.

Job Channel: ____________________. Job Log File: _____________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

Use guild to verify that the results from test 9 were entered into the database correctly. Check the SNGL_DPERIODIC, PROCESS, AND PROCESS_PARAMS tables. Check the job file to verify that the correct filter parameters appear in the tables. Check that the RA, Dec., and Frequency are entered as REAL8 numbers.

Pass/Fail

Known faults (list problem report numbers):

____
____
____.

New faults (list problem report numbers):

____
____
____.

Comments:

TEST 11 Documentation Test

Purpose: Test that documentation exists for the KPD DSO. Test the completeness of this documentation.

Tester: _______________. Date & Time: ___________. Tester Location: ____________.

Job Site: _________________________. Job Database: ____________________________.

Job Channel: _____________________. Job Log File: _______________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

Have the KPD MDC coordinator ftp the documentation fo the KPD DSO to the KPD MDC output URL. Check that this documentation gives an overview of the KPD DSO, defines the input filter parameters, and defines the error codes.

Pass/Fail

Known faults (list problem report numbers):

____
____
____.

New faults (list problem report numbers):

____
____
____.

Comments:

TEST 12 Final Test On Real Data using Real Filter Parameters

Purpose: Test the KPD DSO on real data using real filter parameters. This is the last test, and it is just for fun. The KPD DSO is not yet ready to search for actual signals.

Tester: _______________. Date & Time: ___________. Tester Location: ____________.

Job Site: _________________________. Job Database: ____________________________.

Job Channel: _____________________. Job Log File: _______________________________.

Instructions: (See the "How to run the test scripts" section if you need help.)

Copy kpdTime2048FrameTest.job to kpdDemod3600RealTest.job. Edit this job to search of a known pulsar (e.g., the Crab) on 3600 seconds of real data. Ftp the result to the KPD MDC output URL. Append test.12.Real to the output name. The results will not mean anything at this point. Just note if the job succeeds or fails. This should be a satisfying way to end an MDC.

Pass/Fail

Known faults (list problem report numbers):
____
____
____.

New faults (list problem report numbers):
____
____
____.

Comments:

V. Results of the November 2001 KPD MDC

The KPD MDC was held at LHO Nov. 27-30, 2001. The lal and lalwrapper software used for this MDC was tagged PREMDC011125, while the LDAS software used was version 0.0.23pre with a build date of Mon. Nov. 26, 2001. The email account for jobs run during the MDC was gmendell@ligo-wa.caltech.edu. Emails are in the netscape folder LocalMail/LDAS/searchcode/KpdMdcNov01 on host vulcan at LHO, and have been backed up onto the GC network in /home/gmendell/nsmail.tar. Output of KPD MDC jobs are stored at ftp://lynx.ligo-wa.caltech.edu/pub/incoming. All documentation, test data, scripts, logs, and results of the MDC are stored at gravity.phys.uwm.edu in the /usr/local/cvs/ldasmdc cvs archive.

The MDC was very successful, with almost all tests passing. All problems that occurred during the MDC have either been fixed or have been addressed by LDAS, and will be fixed in subsequent releases.

TEST 1 Error Codes

All jobs passed except for error codes 8, 11, and 22. These jobs failed due to bugs in the KPD DSO. These bugs were fixed during the MDC, and the jobs then passed for these error codes. The bug for error code 22 put the DSO into an infinite loop. LDAS allowed the job to run indefinitely since it kept reporting progress (though the progress reported was always 0). This was recorded as LDAS gnats PR 1227.

Problem Reports Issued: LDAS gnats 1227.

TEST 2a Correctness of SFT output

All tests passed.

TEST 2b Correctness of SFT meta data

All tests passed.

TEST 3 Test Data Conditioning Resample Action

All tests passed. Note that it seemed that the datacon resample action can introduce up to a 0.1% error in the SFTs.

Also, unrelated to these test some jobs failed with "mpi::newJobCallback: job stalled in wrapperAPI (no error reported)" or with "mpi::newJobCallback: job stalled. Possible lam error, see logs, wrapper failed to connect". This problem was recorded as LDAS gnats PR 1216.

Problem Reports Issued: LDAS gnats 1216.

TEST 4 Test Data Acquisition, Output, and Storage

All tests passed except 2b, which tested sending frame output from LDAS to an ftp URL. This problem was recorded as LDAS gnats PR 1217. Also, a bug in frame API caused data sequences sent to the dataconAPI to be on sample too long. This problem was recorded as LDAS gnats PR 1206. This was overcome during the MDC by adding an initial slice to the datacon actions.

Problem Reports Issued: LDAS gnats 1206 and 1217.

Results for Test 4.1 and least square fits to the results are below. Note that N is the total number of samples in the times series sent to each API. Initially the data is sampled at 16384 Hz, and then resampled to 4096 Hz in the dataconAPI.

N/16384	frameAPI T (seconds)
16	16.12
128	59.20
512	67.14
2048	147.67

T = 54.3 + 4.44e-2(N/16384)

N/16384	dataconAPI T (seconds)
16	5.94
128	10.36
512	24.53
2048	84.67

T = 5.17 + 3.88e-2(N/16384)

N/4096	mpiAPI T (seconds)
16	20.05
128	23.75
512	32.27
2048	72.96

$T = 20.1488 + 1.76e\text{-}2(N/4096) + 7.41e\text{-}4(N/4096)\log_2(N/4096)$

N/4096	Total T (seconds)
16	103.10
128	106.90
512	140.30
2048	321.71

$T = 93.8 + 2.27e\text{-}5N + 1.81e\text{-}7N\log_2 N$ (keeping 1.81e-7 fixed using mpiAPI data above)

N/4096	Total S (Megabytes)
16	.529158
128	4.99129

512	16.781912
2048	67.112927

S = 0.313862 + 3.26e-2(N/4096)

TEST 5 Power Leakage Tests

All tests passed.

TEST 6 Test Data Quality Channel and Dropout Handling

All tests passed except Test 6.3. This was a test of the –allowgpass dataPipeline LDAS command option, with a request for an intentional gap in the –framequery option. LDAS did not support intentional gaps at the time of this MDC. This problem was recordes at LDAS gnats PR 1219. Initially Test 6.2 failed, when the quality channel was pushed to the dataconAPI. This problem was recorded at LDAS gnats 1218. However, this problem was fixed by LDAS during the MDC, and the test subsequently passed.

Problem Reports Issued: LDAS gnats 1218, 1219.

TEST 7 Test Multiple SFT Output

The one job kept in the MDC for this test passed. Initially, jobs were included to test the –setsingleem and –communicateoutput options, to attempt to produce multiple SFTs in multiple output files from a single LDAS job, with 1 SFT per file. However, the meanings of these options were misunderstood, and it was realized these should not be part of this MDC. The issues are documented in LDAS gnats PR's 1221 and 1222.

Problem Reports Issued: LDAS gnats 1221, 1222.

TEST 8 Test DeFT Output

All tests passed. However, it was known during this MDC that the KPD DSO was not using the correct definition of SNR. (The correctness of the SNR was not a test for this MDC.) This will be corrected before the next KPD MDC.

TEST 9 Test Statistical Analysis

All tests passed. However, see the comments under Test 8 above.

TEST 10 Test Database Output

All tests passed, in that the data expected shows up in the SNGL_DPERIODIC table. However, PARAM_SET and IFO are blank in the process table, and Start_Time and End_Time are 0 in the process table. This may be intended, and no PR's were issued.

TEST 11 Documentation Test

Documentation was added during the MDC, so that this tests now passes.

TEST 12 Final Test On Real Data using Real Filter Parameters

This test passed.

APPENDIX A

SFT SPECIFICATION

SFT Specifications

I. Input Time Series Specifications

1. The Sample Rate

Theoretical pulsar spin rates are in the 0 to 2000 Hz band. Observed pulsar spin rates are in the 0 to 642 Hz band. Thus, theoretically, gravitational radiation is expected in the 0 to 4000 Hz band, plus higher harmonics. Even to cover all the known pulsars requires searching the 0 to 1284 Hz band. For this reason it is probably reasonable to choose:

Sample Rate = fSample = 4096 Hz (Time Step Size = deltaT = 1/4096 Hz = 2.44140625e-4 sec.)

2. The Search band

A sample rate of 4096 gives a Nyquist frequency of 2048 Hz. Thus the search band will be:

Search Band = B = 0 to 2048 Hz.

3. Data Conditioning

(i) The gravity wave channel will be resampled to 4096 Hz.
(ii) Resampling is done using the datacon API resample action will be used. This action includes anti-alias filtering. The knownpulardemod DSO includes code to account for the time delays introduces by this filtering, by inputing a data segment with some small overlap with the previous and next segment, and then finding the slice of the input segment after resampling that corresponds to the desired GPS time and duration (tSFT).

Are any other data conditioning steps needed? Calibration? Line removal? Further filtering?

4. Duration = tSFT

This time must be choosen so that the frequency modulation during tSFT is less than one frequency bin. This is computed by tSFT <= 5.5 X 10^3*sqrt(300 Hz/2048 Hz) s = 2105 s (Schutz and Papa [2]). Choosing the largest power of two less than this upper bound:

Duration = tSFT = 2048 seconds 34.1333... minutes. (Frequency Step Size = deltaF = 1/2048 s = 4.8828125e-4 Hz)

5. Number of Input data points per SFT = inputN

Note that when you request 2048 second you get 2048 - deltaT seconds of data, but fSample*2048 data points. But this is what we want to get a power of two data points. Thus:

inputN = (4096 Hz)*(2048 s) = 8388608 = 2^{23} data points.

6. Input memory requirements per SFT

A. The input data is REAL4 data = 4 bytes per datum.

B. Before resampling the sample rate is 16384 Hz = 4 * 4096 Hz.

C. Before resampling:

(16384 Hz) * 2048 s per SFT * 4 bytes per datum * 1 MB / 2^{20} bytes = 128 MB per SFT.

D. After resampling:

(4096 Hz) * 2048 s per SFT * 4 bytes per datum * 1 MB / 2^{20} bytes = 32 MB per SFT.

7. The Channel. SFT are computed from the gravity wave channel, e.g., H2:LSC-AS_Q.

II. Output SFT frequency series specifications

1. Start Frequency = fMin = 0 Hz.

2. Stop Frequency = Nyquist frequency = fMax = 1024 Hz.

3. Frequency Step Size = Frequency resolution = deltaF = 1/2048 s = 4.8828125e-4 Hz.

4. Output number of data points = nDeltaF:

SFTs are computing using LALForwardRealFFT which outputs inputN/2 + 1 frequency bins, which corresponds to frequencies up to the Nyquist frequency. Thus:

nDeltaF = inputN/2 + 1 = (4096 Hz)*(2048 s)/2 + 1 = 2^{21} + 1 = 4194305.

5. Computation Method

A. SFTs are computing LALForwardRealFFT, which outputs COMPLEX8 data.

B. It should not make a difference whether an approximate or exact realForwardFFTPlan is used.

6. Normalization

SFTs are normalized by multiplying by deltaT/sqrt(tSFT).
Thus SFT units are input data unit per root Hz.

(Hopefully this input data unit can be determined by code, though at the moment it is hard coded as ADC counts for realy data and volts for ifodmo test data in wrapper.ilwd.)

7. Disk Storage Type

SFTs are recast as COMPLEX16 data for use in LALDemod.

8. Disk storage Requirements

For COMPLEX16 data is nDeltaF*16 bytes:

A. Storage per SFT: (4194305)*16 bytes per SFT * 1 MB/ 2^{20} bytes = 64.00001526 MB + overhead per SFT.

B. Storage per Day: (24 hr/day) * (3600 sec/hr) * (1 SFT /2048 Sec) * 64.00001526 MB/ SFT = 2700.000644 MB/day.

C. Seventeen Day E7 Run: 2700.000644 MB/day * 17 day * 1 GB/ 2^{10} MB = 44.8 GB + overhead.

We could save SFTs as COMPLEX8 data as the SFT are only accurate to the precision; the trade off is between disk space and having to recast into COMPLEX16 data to use in LALDemod. (If recasting is not a bottle neck then make no sense to waste disk space. However, if it is a bottleneck, then speed is more important that disk space.)

9. Total Memory requirements per SFT:

A. input 32 MB per SFT + output 64 MB per SFT = 96 MB per SFT + overhead.

B. Maximum number of SFTs in 512 MB memory <= 512 MB/96 MB = 5 SFTs.

C. Total duration of data in 512 MB memory = 5 SFTs * 2048 sec per SFT = 10240 sec = 2.8444... hours.

Clearly, to coherently analyze longer stretches of data requires either reading in data from SFTs or retrieving SFT data from previous LDAS jobs. The later is possible under LDAS, but has not been tested.

10. SFT will be output in frame format. See Section IV below.

11. The SFT will be made available for transfer either via anonymous ftp or scp.

III. Handling Data Drop Out

1. -allowgaps option is used in the LDAS dataPipeline command, and dataconAPI fillgaps action is used is used to fill gaps with mean value of input data.

2. Locked stretches of data must be identified. (-dbqualitychannel option is used in LDAS dataPipeline command. Unlocked segments of the data will be replaced with mean value of the input data.)

Issues of windowing, etc. should be discussed further.

3. The percent of the duration for each SFT that was padded will be stored in the history output structure (see Section IV below). SFT will not be generated if this is 100% (or perhaps above some threshold.)

4. SFT Time Stamps are computed by a static function similar to the "times" function in LALDemoTEST:

VI. SFT frame format

1. Naming Convention:

```
$S-SFT_$C_$N-$GPS-$T.gwf
```

where

A. $S is the site name (e.g., H, L, …).

B. $C indicates the channel the SFT was computed from (e.g., H1, H2, L1).

C. $N is the number of SFTs in the file, if more than 1.

D. $GPS is the GPS start time for the first SFT in the file.

E. $T is the total duration of input data that generated SFTs in the file.

Example: H-SFT_H2-600000000-2048.gwf

This file contains 1 SFTs computed from 2048 seconds of H2:ASC-AS_Q data.

2. Frame Structures

An FrDump of an example SFT file reveals:

```
[gmendell@vulcan SearchCodeOut]$ FrDump -i H2-SFT_TEST1_4-600000000-
1.gwf -d 5 | less
...
History records:
Processed Data:
 Proc:    sft_output sampleRate=-1 fShift=0.0000e+00
comment:multiDimData to ProcData
  Vector:data ndata=9 nDim=1 unitY= unitX=hz  startX=0.00e+00
dx=1.28e+02
   Data(CD) (7.161711e-03,0.000000e+00) (7.879787e-03,1.053996e-03) ...
```

A. sft_output = data set name

B. sampleRate and fShift: IGNORE THESE; WILL BE SET TO -1 OR 0.

C. ndata = nDeltaF = 9

D. unitY should be adc counts per root hertz for this test data. This could not be read by FrDump.

E. startX = Start Frequency = fMin = 0.0

F. dx = deltaF = 128 Hz (Unit is stored as unitX)

E. Data (Note FrDump printed out data as REAL8. Will need to learn how to read data type from frame structure.)

SFT Meta Data is stored in the History Structure. Example ilwd output:

```
            <ilwd name='LDAS_History' size='9'>
                <ilwd name='percent_pad'>
                    <real_4 units='percent'>0.0000000e+00</real_4>
                </ilwd>
                <ilwd name='max_power'>
                    <real_4 units='power'>7.8124991e-03</real_4>
                </ilwd>
                <ilwd name='freq_pwmax'>
                    <real_4 units='hz'>3.8400000e+02</real_4>
                </ilwd>
                <ilwd name='power_mean'>
                    <real_4 units='power'>8.6805545e-04</real_4>
                </ilwd>
                <ilwd name='power_stddev'>
                    <real_4 units='power'>2.4552315e-03</real_4>
                </ilwd>
                <ilwd name='startSec'>
                    <int_4s units='s'>600000000</int_4s>
                </ilwd>
                <ilwd name='startNan'>
                    <int_4s units='ns'>0</int_4s>
                </ilwd>
                <ilwd name='sample_rate'>
                    <real_4 units='hz'>2.0480000e+03</real_4>
                </ilwd>
                <ilwd name='ndata_input'>
                    <int_4s units=' '>16</int_4s>
                </ilwd> </ilwd>
```

The meta data is:

A. Percent Padding.

B. Maximum power is this SFT.

C. The frequency corresponding to this maximum power.

D. The mean power is this SFT.

E. The standard deviation of the power.

F. The start time from which this SFT was generated (StartSec & StartNan).

G. The sample rate of the input data this SFT was generated from.

H. The number of input data points this SFT was generated from.

APPENDIX B

The method used to generate DeFTs is discussed in this appendix. (See Schutz and Papa [2], Williams and Schutz [3], and S. Berukoff and M. A. Papa, [4].)

Given a discrete time series x_a and model of the phase Φ_{ab} one can define the phase demodulate Fourier transform (DeFT) by

$$X_b = \sum_{a=0}^{NM-1} x_a e^{-2\pi i \Phi_{ab}} , \qquad (1)$$

where a is the discrete time index, and b is a discrete frequency index. The number of input data points has been chosen as NM, so facilitate breaking the sum into M short time segments containing N samples each

$$X_b = \sum_{\alpha=0}^{M-1}\sum_{j=0}^{N-1} x_{\alpha j} e^{-2\pi i \Phi_{\alpha j b}} , \qquad (2)$$

where α is the segment index, and j is the short time index. We now replace $x_{\alpha j}$ with

$$x_{\alpha j} = \sum_{k=0}^{N-1} X_{\alpha j}^{SFT} e^{-2\pi i \Phi_{ab}} , \qquad (3)$$

where X^{SFT} is the discrete Fourier transform of the short time segments known as an SFT. The time segments are chosen short enough so that modulation effects due to the Earth's motion and intrinsic source evolution are small, and the source frequency will appear monochromatic in the SFT. Substituting Eq. (3) into (2) and rearranging

$$X_b = \sum_{\alpha=0}^{M-1} Q_{\alpha b} \sum_{k=0}^{N-1} X_{\alpha k}^{SFT} P_{\alpha k b} . \qquad (4)$$

The expressions for P and Q simplify if we expand the phase about the midpoint of each SFT:

$$\Phi_{\alpha j b} = \Phi_{\alpha,1/2,b} + f_{\alpha,1/2,b}(t_{\alpha j} - t_{\alpha,1/2}) . \qquad (5)$$

In this case, P and Q are given by

$$
\begin{aligned}
P_{\alpha kb} &= \frac{\sin u_{\alpha kb}}{u_{\alpha kb}} - i\frac{1-\cos u_{\alpha kb}}{u_{\alpha kb}} \\
Q_{\alpha b} &= e^{iv_{\alpha b}} \\
u_{\alpha kb} &= 2\pi\left(\frac{T}{M} f_{\alpha,1/2,b} - k\right) \\
v_{\alpha b} &= -2\pi\left(\Phi_{\alpha,1/2,b} - \frac{T}{2M} f_{\alpha,1/2,b}\right)
\end{aligned} \qquad (6)
$$

In Eq. (6) T is the total duration of the input time series used to produce a DeFT. Since P is peaked near k* = (T/M)$f_{\alpha 1/2b}$, it has been shown that the sum over k can be limited to approximately k* - 8 to k* + 8. This method is coded in the LALDemod function, written by Berukoff and Papa [4]. This function can also be used for a known pulsar search.

Note that the original computational complexity of computing DeFTs is reduced from O(MN $\times$ number of phase models) to O(MNlog$_2$N) + O(M $\times$ number of phase models). Note that O(MNlog$_2$N) is the complexity of computing the SFTs. But this only has to be done once. Since to conduct an all sky search for pulsars involves a very large number of phase models, the method above would seem to be a substantial reduction in the complexity of the problem. However, this turns out to not be exactly true, since one could demodulate the data stream for a given sky position and set of spin down parameters and then FFT the demodulated data stream. This would give results for all NM frequencies at once, while the DeFT algorithm must be applied for each of NM frequencies. Thus, the DeFT algorithm turns out to be O(M/log$_2$NM) more complex than doing an FFT of demodulated data streams. However, the DeFT algorithm more easily allows the analysis to be split into frequency bands and performed on a parallel cluster of computers.

Finally, note that to include the effects of amplitude modulation (which depends on polarization) one needs to weight the input time series with appropriate beam pattern response functions for the + and $\times$ polarizations, and compute the DeFT for each of these. See Jaranowski, Krolak, and Schutz [1]. Coding is under way at AEI to incorporate this.

References

[1] P. Jaranowski, A. Krolak, and B. F. Schutz, gr-qc/9804014.
[2] B. F. Schutz and M. A. Papa, gr-qc/9905018.
[3] P. R. Williams and B. Schutz, gr-qc/9912029.
[4] S. Berukoff and M. A. Papa, LAL Pulsar Package Documentation, Chapter 4.