First Light CASIMIR/GREAT observing modes and software implementation

J. Stutzki, January 2003, v1.2

1. Introduction

1.1. Philosophy

The two first generation heterodyne instruments on SOFIA, CASIMIR and GREAT, will use a common data acquisition and telescope control software for observing with SOFIA. This document describes the observing modes intended to be used from SOFIA-first light on. These are selected to be absolutely necessary for successful and efficient observing onboard the airborne observatory; at the same time, they are setup to be as simple as possible with regard to the observatory interface. Later on, more observing modes may be added, needing additional complexity at the telescope/observatory interface.

The observing procedures defined below are based on long term experience of observing at ground based radio telescopes and on the KAO observing experience. The observing procedures discussed are widely established at ground based mm- and submm-wave observatories such as CSO, JCMT, HHT and KOSMA. In fact, largely the same software will be used (together with a similar hardware setup) at some of those observatories quite some time before SOFIA flies. In particular, the identical telescope/observatory specific interface (telescope tracking and pointing commands , chopping secondary commands, and telescope/observatory status information) will be used. This ensures that the software is well understood and thoroughly debugged and has (at least some) heritage of successful real time use at an observatory by the time it gets used on SOFIA; potential problems are thus likely to be constrained to the immediate observatory/telescope interaction. This should give a good chance that they can be solved efficiently and without much of any loss of observing time on board SOFIA.

1.2. Software and hardware architecture

The observing software for both instruments will run on a (small) number of Linux-based PCs (and possibly also Linux-based VME-bus computers). It controls on one side the instrument (front-end, backend, calibration system etc.) and on the other hand the SOFIA telescope including the secondary chopper. Quick-look and offline data analysis are a separate software unit (note: this is somewhat different from the original WASP software design, where the data analysis is done online within the data acquisition package), making extensive use of the CLASS package. They access the raw and/or pre-reduced data provided by the observing software in a CFITS format (note: another option is to use the SDFITS format; a final decision on these options has to be made soon).

The interface to the SOFIA telescope itself is through a TCP/IP- Ethernet connection via MCS supported commands. These make the telescope track positions on the sky or scan along predefined tracks; and they control the chopper operation. They also feed back telescope and observatory status information to the observing system. Only the timing of the chopper motion is directly controlled by a hardware TTL connection.

1.3. Modularity

The observing software used by GREAT and CASIMIR is build in a modular way. There are many different ways to define modularity, so that a short description seems appropriate. Here 'modular' means that the various functions to be performed are distributed to relatively small, self-contained tasks, each of which acts under control of a set of input parameters and produces output in the form of further parameters used for input to other tasks or for status information. In the case of the immediate data acquisition tasks, the output contains also the data, generated in some raw format. These tasks internally communicate via the KOSMA_file_io system, i.e. through ASCII parameter files with time-stamps, that also allow for starting and aborting these tasks.

Each task can be run from the operating system command line level and its execution mode is entirely controlled by the KOSMA_file_io ASCII-files. More complex procedures are executed via shell scripts combining sequences of these low level tasks and creating/modifying the appropriate KOSMA_file_io-files. At yet a higher level, these tasks and shell scripts are ultimately controlled via a GUI-based observer interface.

This setup guarantees that every task can be debugged in a straightforward and transparent way, e.g. by changing its input files with any standard text editor. The few tasks that directly control pieces of hardware and thus need the physical presence of this hardware for functioning properly, can be replaced by 'dummy' tasks, that emulate the appropriate interaction with that particular hardware. In this way, the complete system, or any subset of the system, can be debugged in a standalone way without the need to have any of the instrument or observatory hardware attached.

2. Observing Modes

In the following this document largely concentrates on describing the observing modes used by CASIMIR and GREAT at first light, and their detailed implementation. The presentation is oriented toward the software implementation of the defined observing modes and tries to be complete in the sense, that no additional information is necessary to fully specify the sequence of actions to be performed in order to execute an observing mode.

The observing modes are separated into a first, small set of 'fundamental modes', that are defined by a specific data acquisition and telescope/observatory operation mode. From these, a set of 'higher level' observing modes is constructed that allow to perform the full science functionality of the instruments/observatory.

The selection of the fundamental modes to be implemented is based on the need of doing difference measurements on a time scale faster than the typical drift time scale of the instrument. The latter can extend out to some 100 seconds with substantial effort going into proper thermal control of all instrument components; but some 40 seconds or so are achieved more typically. Another important time scale entering into consideration is the typical integration time needed to reach a sufficient signal-to-noise ratio. Even for the brightest lines in the submm-/FIR-band relevant for SOFIA, present state of the art receiver sensitivities require some 10 seconds integration time to reach a S/N of a few to 20. This is of the same order as the drift time scales, implying that typically a few to many differencing cycles are necessary to reach sufficient S/N in a real observation.

Experience with heterodyne observing shows, that signal modulation on time scales of about 1-2 sec is always sufficiently fast; thus, modulating the signal with the secondary chopper (on and off the source) or with frequency-chopping at this rate is sufficient. This data acquisition mode is implemented as **signal-reference-switched sub-scans**.

As frequency chopping is limited to narrow frequency throws (defined by the smoothness of the instrument bandpass), its applicability is limited to sources with relatively narrow lines, typically of widths up to some 10 km/s. Sky-chopping with the secondary is limited to a maximum chop throw on the sky of typically a few arc-minutes (8' in the case of SOFIA) and is thus applicable only for sources with a small spatial extent. The only differencing technique that work for more extended sources and/or sources with wider spectral lines is a slow switch between on- and off-source position by moving the whole telescope. This motivates the implementation of the **total-power sub-scans**.

Total power observing typically requires relatively large overheads for the telescope movements (up to several seconds) and thus rapidly reduces observing efficiency for large on-off distances. If the extended source is to be mapped, the overhead can be reduced drastically by observing many on-source positions within one slow switch cycle together with a longer integration off-source. For optimum signal/noise, the off-source integration has to be *sqrt(n)* times longer than the individual on integration, where a whole cycle contains *n* on-positions. This concept leads to **on-the-fly (OTF) sub-scans** as another fundamental observing mode.

Although other fundamental observing modes may come into discussion, one should note that the above three cover all needs. For example, a sky-chopped OTF mode may at first sight look attractive to implement. With the telescope secondary chopping along the scan direction, one can derive the intensity distribution on the sky by integrating up the differentiated signal, as is regularly done today in broadband bolometer observations. However, comparison of the various time scales discussed above shows, that one can achieve the same and with similar efficiency by combining signal-reference-switched sub-scans at subsequent discrete positions. The issue here is the relatively long total integration time per position needed to achieve sufficient S/N (which is typically at least a few 10 seconds), compared to the overhead time involved in moving the telescope to subsequent near-by positions (which is typically about a second or so). Note that the data transfer time is similarly small, so that the data can be transferred while moving the telescope. In contrast, for bolometer observations of bright, extended sources, the per pixel integration time is on the order of only a second or so, so that the overhead for moving the telescope is a substantial fraction of the time, and on-the-fly observing is more efficient.

Another observing mode widely in use at ground based facitlites is total power raster mapping, where a small number *m* of subsequent discrete map positions is observed in one cycle with a *sqrt(m)*-longer off-source integration. We regard this as unnecessary to implement, as OTF observing is yet more efficient.

Based on these considerations, we do not, at least not for first light, aim at other fundamental observing modes, in particular as they would likely require a more complex synchronization scheme interaction between the telescope, including the chopper, and the data acquisition.

2.1. fundamental modes

2.1.1. total power sub-scan (TP-mode)

In total power mode, the telescope is commanded to stare at a user defined position on the sky. After being commanded to go to that particular position, the telescope acknowledges arrival on that position and regularly feeds back status information confirming that it is still on track within a tolerance specified by the observing software. It keeps tracking that position until it is commanded to do otherwise.

The secondary chopper and all other signal modulation are switched off.

The instrument front-end has been tuned to a specified setting and regularly provides health (and other) status information

The backend is commanded to acquire data for a given length of time. It internally converts this length to the proper number of basic readout cycles (thus specifying the actual sub-scan duration, which may be shorter than the commanded one by a maximum of one readout cycle). It clears the add-up buffer and then integrates subsequent readouts to this buffer. As part of its output it may regularly output the number of readout cycles already integrated (this would serve as progress info). At the end of integration it raises an appropriate flag and returns the added-up buffer as integer-counts raw data.

The data acquisition can be aborted at any time (within a few readout cycles) by issuing an appropriate command parameter to the backend task.

<u>Some additional information:</u> due to the total stability time of typical heterodyne systems of not more than 100 seconds, TP-subscans will typically be of some 10 to 40 seconds duration on the sky. When used in calibration on the instrument internal hot and cold load (see bleow), they will typically be about 5 to 10 seconds, including possibly an equal length TP subscan on some blank sky position (for determining the atmospheric transmission). For spectrometer internal calibration (zero/comb, see below), TP-subscans of 2 seconds duration are typical.

NOTE: due the heterodyne instruments operating at relatively long wavelengths and with correspondingly large beam sizes, gyro tracking is fully sufficient. After an intial setup of the gyro inertial reference trough one of the optical tracking cameras, SOFIA can be pointed blindly for up to several hours within an arcsec tolerance; resetting the gyro drifts is only needed on these long timescales by again using one of the tracking cameras. Therefore, both heterodyne instruments can operate with the solid aluminum tertiary, as long as they use their internal CCD camera to set the gyros to the approriate intertial reference frame (CASIMIR has this implemented, GREAT is thinking about doing so (tbd)).

2.1.2. signal-reference-switched sub-scan (S/R-mode)

In signal-reference mode, the telescope is commanded to track a user defined position on the sky (at least in one of the two sky-chopper positions, if sky-chopping). After being commanded to go to that particular position, the telescope acknowledges arrival on that position and regularly feeds back status information confirming that it is still on track within a tolerance specified by the observing software. It keeps tracking that position until it is commanded to do otherwise.

The secondary chopper or some other signal modulation are switched on, modulating the signal in phase with the S/R-TTL signal distributed throughout the system to the relevant hardware units. For chopping with the telescope secondary, chopper frequencies will typically be around a few Hz down to 0.5 Hz. Other signal modulation (e.g. frequency chop) will be of similar speed.

The instrument front-end has been tuned to a specified setting and regularly provides health (and other) status information

The backend is commanded to acquire data for a given, total length of time. It internally converts this length to the proper number of basic S/R-phases and the number of readouts per S- or R-readout cycles (thus specifying the actual sub-scan duration, which may be shorter than the commanded one). It clears the add-up signal and reference buffer and then integrates subsequent signal and reference readouts to these buffers. As part of its output it may regularly output the number of S/R-readout cycles already integrated (this would serve as progress info). At the end of integration it raises an appropriate flag and returns the added-up signal and reference buffers as integer-counts raw data.

The data acquisition can be aborted at any time (within a few readout cycles) by issuing an appropriate command parameter to the backend task.

2.1.3. on-the-fly sub-scan (OTF-mode)

In on-the-fly mode, the telescope is commanded to track along a great circle on the sky, starting at a user defined position and in a user defined direction and speed at a given start time in the (near) future, and for a given length. After being commanded to do so, the telescope acknowledges arrival on that position with the specified velocity by regularly feeding back status information confirming that it is still on track within a tolerance specified by the observing software. After the specified length of time, the telescope stops tracking at the given speed and keeps tracking the end position of the OTF sub-scan until commanded to do otherwise.

The telescope cannot, of course, reach the commanded position and track across it at the commanded velocity, starting from arbitrary positions on the sky. It would accordingly respond with “not on track”. In order to ensure efficient operation, the observing software has to command the telescope to track a 'pre-OTF' position, which is an appropriate distance in front of the OTF-sub-scan in order to ensure that the telescope can accelerate within its drive system physical torque limits to scan across the start position with the specified velocity at an approximately given time lap in the future. This distance of the 'pre-OTF' position is calculated by a observatory/telescope specific task, that the observing software calls to beforehand. Once tracking the 'pre-OTF' position, the observing software can command the telescope to start the OTF-sub-scan anytime, with the appropriate time lap between issuing the command and the start time specified. This procedure ensures that the telescope can successfully execute the OTF-sub-scan as long as it functions properly.

The secondary chopper and all other signal modulation are switched off.

The instrument front-end has been tuned to a specified setting and regularly provides health (and other) status information

The backend is commanded to acquire data in 'continuous' readout and transfer mode, i.e. starting at the given start time in the (near) future, and integration a specified (small) number of readout cycles into a series of subsequent data buffers, each buffer representing the accumulated signal for a position interval along the OTF-sub-scan. The buffers are cleared at the beginning of the sub-scan. While acquiring data, the backend software has to transfer the previously accumulated buffer to the data reduction pipeline as integer count raw data. As part of its output it may regularly output the number of readout buffers already finished (this would serve as progress info). At the end of integration it raises an appropriate flag to signal the end of the data acquisition.

The data acquisition can be aborted at any time (within a few readout cycles) by issuing an appropriate command parameter to the backend task.

Some additional information: due to the total stability time of typical heterodyne systems of not more than 100 seconds, OTF-subscans will typically be of total duration of order 40 to 60 seconds. In order to get acceptable low level of distortion of the resulting maps from the beam smearing due to the continues motion, on the order of 3 to 4 readout cycles per beam diameter are necessary. In order to keep the data volume and data handling times in reasonable limits, the individual readout cycles are typically selected to be of 1 to 4 second duration, resulting in slew speeds of *(beam diameter)/(readous per beam diameter)/(length of readout cycle)*, which amounts to typical slewing speeds of maximum 20 arcsec/second to minimum 2 arcsec/second for observations of astronomical sources. On SOFIA, such OTF scans would thus be within the fine drive range. When OTF mode is used for a skydip (which on SOFIA implies using the coarse drive with lower positional tolerance, see below), one would typically use readout cycle times of some few seconds, each covering about a few degree in elevation range, and thus would need slew speeds in elevation of about 1 degree/second.

2.2. telescope-control interface

The fundamental observing modes detailed above largely define the parameters that are needed to control the telescope functions and that have to be communicated to the telescope control task via the specific KOSMA_file_io-ASCII parameter file. As the actual telescope position is usually derived from a number of other parameters like the source positions, map offsets, off-source offsets etc., there is still, however, a variety of choices with regard to which parameters one wants to explicitly hand over to the telescope control task. The selection presented below is guided by allowing the observer to easily trace back what position within a given observing project the telescope actually tracks. Thus, one should, for example, explicitly specify map offsets so that the observer immediately sees in a status display of the observing software/telescope task interface that the telescope presently tracks a given offset position relative to the source center position. Such considerations lead to the following list of parameters for the observing software/telescope interface.

NOTE: the present version of this write-up does not at all consider the interface to command the chopper; this will have to be done through the appropriate MCCS commands!

NOTE: the present version of the following table is not fully consistent (e.g. parameter names, chopper control parameters) with the present version of the KOSMA_file_io configuration files tel2obs.cfg and obs2tel.cfg); this has to be updated in the next iteratio.

		parameter	***encoding***	***comment***
from observing software to telescope task:				
info		source name	string*40	for display and archiving
		scan number	integer	"
		sub-scan number	integer	"
		obs-cookie	integer	internal identifier for observation
telescope positioning				
	source position	lambda	floating, degree	
		beta	floating, degree	
		coordinate system of source position	string*10: J2000, B1950, AZ/EL, galactic	
	map position	delta lambda	floating, arcsecs	
		delta beta	floating, arcsecs	
		coordinate system for map positions	string*10: J2000, B1950, AZ/EL, galactic	
		true-angles-flag for lambda	string*1 [Y/N]	apply cosine-correction?
	slew velocities	vellambda	floating, arcsecs/seconds	OTF slew velocity

		parameter	***encoding***	***comment***
		velbeta	floating, arcsecs/seconds	
	offsets	OFFlambda	floating, arcsecs	gives OFF position, correction for chopper offset etc.
		OFFbeta	floating, arcsecs	
		OFF coordinate system	string*10 J2000, B1950, AZ/EL, galactic	different from source coordinate system, e.g. for AZ offsets etc.
		true angle flag for OFFlambda	string*1 [Y/N]	
control parameters				
		tolerance	floating, arcsecs	specifies tracking tolerance acceptable
		starttime	Julian date (note: may be floating seconds from some specicific date are a better choice?)	gives time in (near) future for start of OTF subscan.
		scan duration	seconds	duration of scan (>0 implies OTF mode, =0: ignored)
boresight	focal plane position	x-offset of boresight in focal plane	floating, mm	
		y-offset of boresight in focal plane	floating, mm	

		parameter	***encoding***	***comment***
		focal plane coordinate system	string*20 “fixed_to_telescope” (SOFIA) “fixed_to_horizon” (KOSMA etc.)	
	focal plane rotation	rot_angle_focal_plane	floating, degree	should normally be set to 0 degree, so that third telescope axis (l.o.s.) is set to nominal orientation; will be used with focal plane array (STAR)
from telescope back to observing software:				
status parameters		ontrack	string*1 Y/N	Y if on commanded track within tolerance since last check
		inrange	string*1 Y/N	Y if commanded position is within telescope range
		obs-cookie	integer	return cookie to identify proper observation
		elevation	floating, degree	actual elevation
plus further housekeeping data about telescope attitude and other observatory parameters (to be filled in)				

Notes:

i) to Sean Colgan: I have now allowed for source coordinates and map/OFF-positions to also be in other coordinate systems than just B1950. I think, this is necessary in order to have the flexibility needed for various mapping and OTF-mapping schemes. The coordinate transform that needs to be done inside the telescope control task beyond the KOSMA_file_io-interface.

ii) boresight/focal plane rotation parameters have not been thought through yet, but we are certainly going to need something like this

2.3. higher level observing modes

to be filled in:

2.3.1. frequency calibration

2.3.2. intensity calibration

2.3.3. without sky transmission

2.3.4. with sky transmission

2.3.5. sky-dip

2.3.6. total power scan

2.3.7. double beam-switch scan

2.3.8. OTF-scan

2.3.9. raster mapping

2.3.10. extended mapping