CNRS
Centre National de la Recherche Scientifique

INFN
Instituto Nazionale di Fisica Nucleare

SSFS digital electronics: Design as of September 2016

A. Dalmaz, H. Heitmann, N. Letendre, A. Masserot, B. Mours, E. Pacaud, SS. Petit, L. Rolland, ...

DRAFT

September 21, 2016

VIRGO * A joint CNRS-INFN Project
Project office: Traversa H di via Macerata - I-56021 S. Stefano a Macerata, Cascina (PI)
Secretariat: Telephone (39) 50 752 521 – Fax (39) 50 752 550 – e-mail virgo@pisa.infn.it

Contents

1 Introduction

The Second Stage Frequency Stabilisation (SSFS) loop that was an analog loop in Virgo is being setup as a digital loop in Advanced Virgo since it uses as error signal the output of a digital demodulation board.

This note describes the plan for the SSFS electronics and some intermediate steps in its development and tests.

2 Overview of the SSFS loop

The loops for the laser frequency stabilisation, before and after the SSFS is engaged, are shown in the figures 9.

The SSFS loop takes as error signal the demodulated in-phase signal from a photodiode, either B2 or B4. Loop filters are applied on this error signal to provide the correction signal SSFS_corr sent to the laser via the "rampeauto" analog electronics.

In Advanced Virgo, the B2 and B4 photodiode signals are demodulated digitally. Hence it is necessary to apply digital filters and to use a DAC to convert the correction channel into an analog signal to send it to the rampeauto.

The SSFS loop frequency has a unity gain of the order of 20 kHz. The order of the loop filter is 16 at maximum (i.e. maximum 8 biquad filters)..

The rampeauto electronics board input is a differential voltage input with dynamic[1] ± 12 V. The input connector is a LEMO3 connector.

3 Overview of the SSFS digital electronics

In order to fulfill the requirements for the SSFS digital loop, a dedicated DAQ-Box [1] fast DAC mezzanine has been developped to provide a DAC running up to 1 MHz. Moreover, dedicated firmware is developped to process the digital loop filters in the two on-board DSPs associated to the mezzanine.

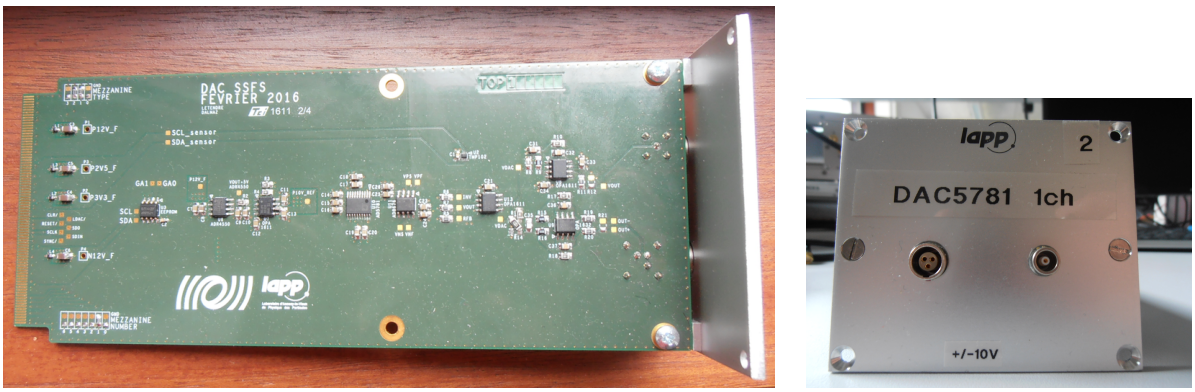
The DAC mezzanine, shown in figure 1, hosts a single 18-bit DAC chip running up to 1 MHz (Analog Devices AD5781). The DAC output is split into two to provide two outputs, one single-ended and one differential. The voltage differential output has a dynamic of ± 10 V.

An analog anti-image filter is present between the DAC output and the mezzanine output: it is a 1st order Butterworth filter with cut-off frequency at 500 kHz??????.

The pipeline from the photodiode RF output to the SSFS error signal is shown in figure 2.

The RF analog signal from the photodiode is first sampled at 400 MHz and digitally demodulated in a digital demodulation mezzanine in a DaqBox located in the air-tank of the

[1] the voltage between GND and both pins IN_+ and IN_- must be kept in the range ± 12 V and the difference $IN_+ - IN_-$ must also be kept in the range ± 12 V.



(a) Top view

(b) Front panel with differential and single-end outputs

Figure 1: Pictures of a DAC5781 mezzanine.

suspended bench. The demodulation process generates two outputs I_raw and Q_raw sampled at 1 MHz. At this level, the demodulation frequency is not exactly equal to the modulation frequency.

In order to keep the loop delay as small as possible, a direct TOLM link[2] has been setup from the suspended bench DaqBox to the SSFS DaqBox located in the Electronic INJ Lab. The two channels sampled at 1 MHz are sent into packets sent at 1 MHz as soon as they are available.

The raw demodulated channels are received in the SSFS DaqBox and pushed as fast as possible to the DSP for processing. First, the "second stage digital demodulation" processes the residual rotation of the signals to correct for the small frequency offset of the first stage and set the demodulation phase. Then, the fully demodulated I_err signal is filtered to apply the SSFS loop digital filter. The filter output $SSFS\_corr$ is sent to the DAC chip to provide the analog SSFS error signal.
All these computations are performed in two on-board DSPs associated to the DAC5781 mezzanine with channels at 1 MHz.
External information is needed for the second stage digital demodulation, the choice of the filter coefficients, the choice of the input photodiode (B2 or B4). A process running on a RTPC will provide this information to the SSFS DaqBox via another TOLM link.

4 DSP algorithm

The two DSPs associated to the DAC5781 mezzanine in the SSFS DaqBox process the second stage digital demodulation and the SSFS loop filters for either B2 or B4 beam input. They

[2] monomode bidirectional optical fiber

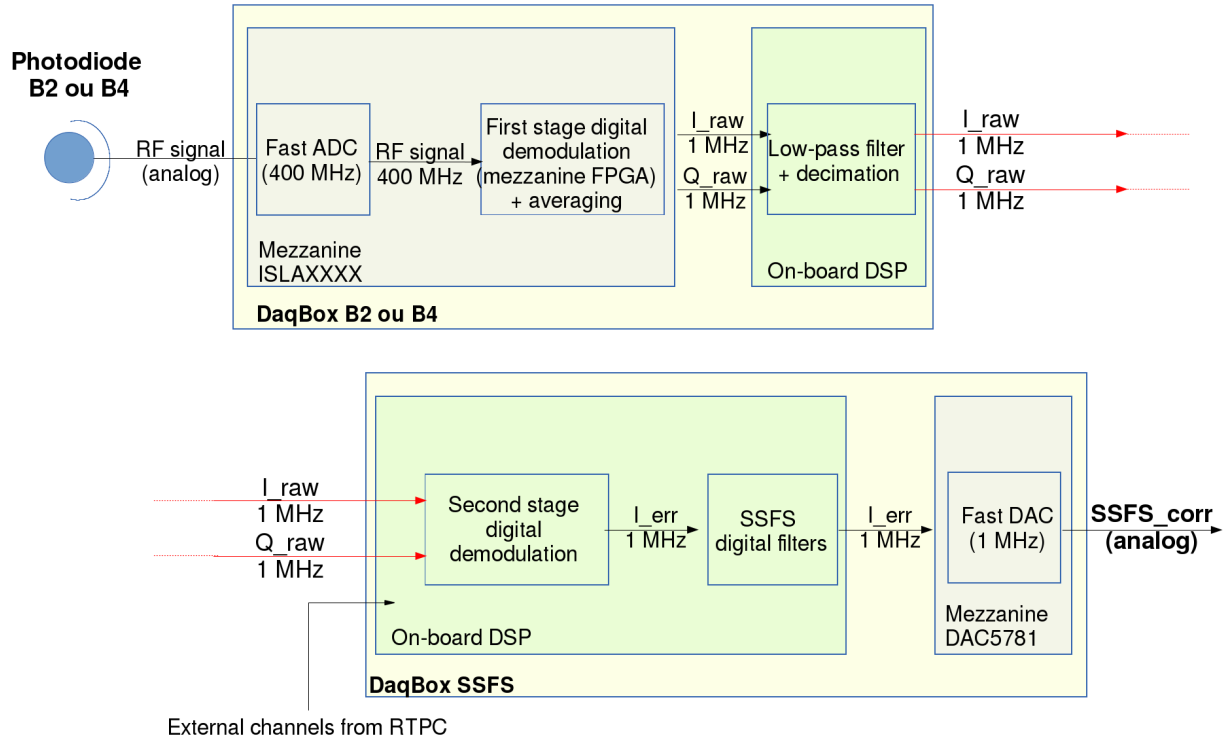


Figure 2: Sketch of the digital SSFS pipeline for a single photodiode.

receive 28 channels in total. Some monitoring channels are sent back to the RTPC. The main output is one single channel, SSFS_corr, sent to the DAC chip for conversion.

The DSPs (ADSP 21469) are running at 450 MHz and the computations are done using 40-bit extended-float precision[3].

4.1 Principle

The algorithm running in the DSPs is shown in the figures 3 and 4.
The possibility to choose between two input beams, B2 or B4, will be provided. Hence the processing will be parallalized using the two DSPs of the fast DAC mezzanine as shown in figure 3. One DSP will make the computation for the B2 beam and the other one will make the computation for the B4 beam. The computations are the same for both but with different input channel. Only internal values that define filter coefficients and external channels are different

[3] float have 32 bits: 1 for sign, 8 for exponant, 23 for mantissa ; extended-float have 40 bits: 1 for sign, 8 for exponant, 31 for mantissa.

between the two pipelines.

In the following paragraphs, the list of input channels is first given ; Then some details about the algorithm itself ; A minimal list of output monitoring channels is then proposed ; the synchronisation of the different input data is finally discussed.

4.2 External input channels from RTPC

It is planned that the SSFS DaqBox receives 28 incoming channels for the DSPs.

The 8 photodiode signals come directly from the DaqBox in the suspended benches as the output of the digital demodulation mezzanine. They are sampled at 1 MHz and sent in packets at 1 MHz, as soon as the data are available to keep the loop delay as low as possible.

The other 20 channels come from a RTPC. The channels are built at 1 MHz but send in packets at a lower frequency to be defined, probably of the order of 50 kHz. It is probably better to send the data into different packets not to delay the incoming photodiode packets at 1 MHz. In any case, different packets will be sent towards the two DSPs to ease their routing in the input FPGA.

- 8 photodiode signals (4 coming from each beam):
  - $B2\_f\_I\_raw$, $B2\_f\_Q\_raw$, $B2\_2f\_I\_raw$, $B2\_2f\_Q\_raw$
  - $B4\_f\_I\_raw$, $B4\_f\_Q\_raw$, $B4\_2f\_I\_raw$, $B4\_2f\_Q\_raw$
- 8 channels with the cosine and sine signals that describe the f and 2f demodulation frequencies and phase (4 for each beams):
  - $B2\_line\_f\_cos$, $B2\_line\_f\_sin$, $B2\_line\_2f\_cos$, $B2\_line\_2f\_sin$
  - $B4\_line\_f\_cos$, $B4\_line\_f\_sin$, $B4\_line\_2f\_cos$, $B4\_line\_2f\_sin$
- 2 channels to apply or not the 2f phase correction (1 for each beam):
  - $B2\_g_{\phi corr}$
  - $B4\_g_{\phi corr}$
- 4 channels that allow to select the filter to be used and the filter gain (2 for each beam). r is a ramp going between 0 and 1 to select the filter. $B2\_gi$ and $B4\_gi$ are the gains of the B2 and B4 filter i:
  - $(1-r) \times \sqrt{B2\_g1}$ and $r \times \sqrt{B2\_g2}$
  - $(1-r) \times \sqrt{B4\_g1}$ and $r \times \sqrt{B4\_g2}$
- 2 channels common to both beam processing but sent twice, in the packet toward DSP1 and also in the packet toward DSP2:

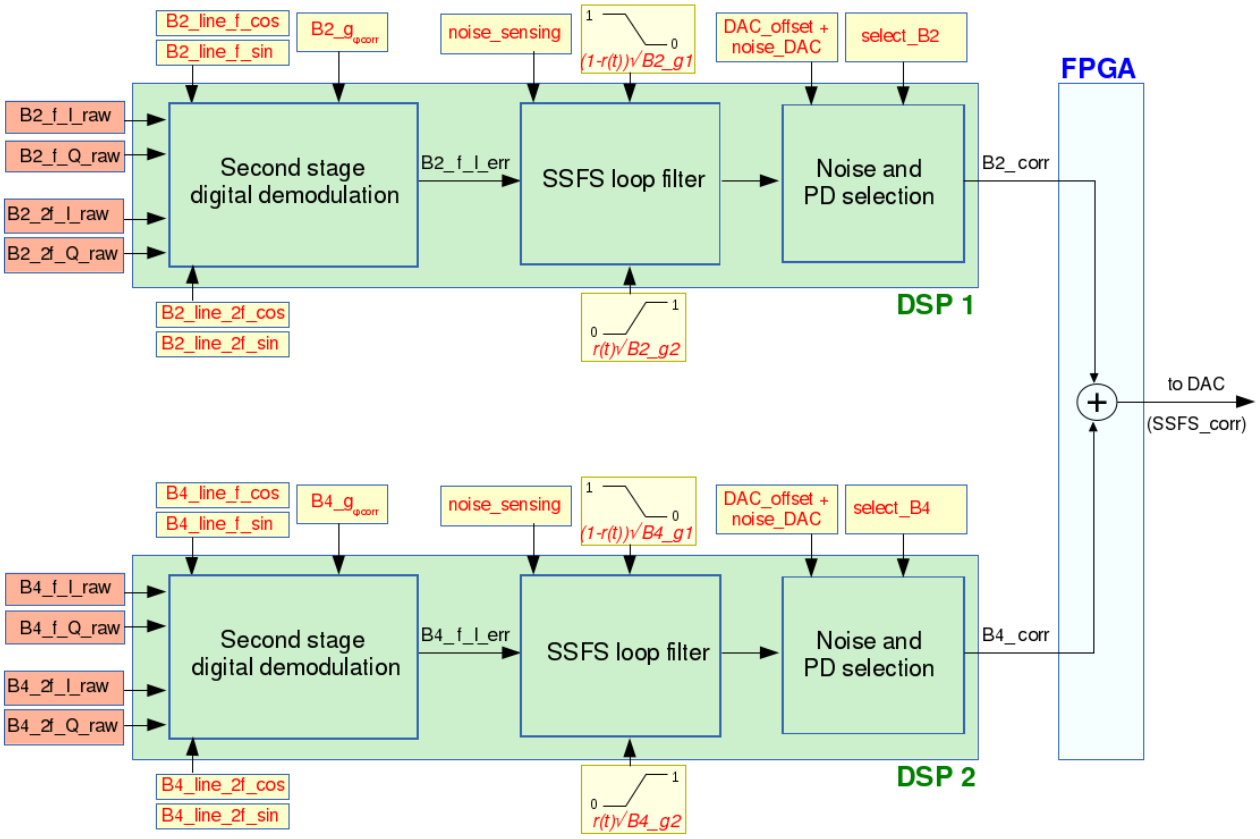


Figure 3: Sketch of the SSFS algorithm and input channels in the two DSPs and FPGA associated to the DAC5781 mezzanine. The input channels are shown in the red and yellow boxes. A DSP process the data for one beam. The main steps are (i) the second stage digital demodulation to extract the SSFS error signal, (ii) the SSFS loop filter themselves and (iii) the possibility to add noise and select which photodiode is used to compute the SSFS correction signal. The output of both the DSP.is sent to the FPGA where they are summed before they are sent to the DAC.

  - $noise\_sensing$ and $noise\_offset\_DAC$

- 2 channels used to select the photodiode used for the SSFS correction:
  - $select\_B2$
  - $select\_B4$

4.3 DSP algorithm

The different steps of the algorithm shown in figure 4 are discussed in this section. It is discussed only for one beam since the same is done in parallel for the other beam on the other DSP.

4.3.1 Second stage digital demodulation

The channels from the photodiodes have been demodulated in the suspended bench DaqBox. A second stage demodulation is needed in order to correct for the demodulation frequency offset, for the demodulation phase and for the gain of the photodiode channels.

The three corrections can be applied multiplying the I and Q photodiode signals by proper sine and cosine channels. The sine channels are sent from a RTPC as $G \times \sin(2\pi\Delta f \times t + \phi)$ (and similar for the cosine) where:

- Δf is the difference between the modulation frequency and the demodulation frequency applied in the demodulation mezzanine in the DaqBox.
- ϕ is the demodulation phase
- G is a gain that takes into account the VGA gain of the photodiode RF channel and the digital averaging process at the output of the mezzanine.

The second stage demodulation is processed for both the channels demodulated at the main frequency f and at $2f$.

4.3.2 Extraction of the phase noise from the $2f$ channels

The $2f$ channel can be used to extract a phase noise. The *atan* operation will be done using linear approximation. It is mandatory that the demodulation phase ϕ of the $2f$ second stage demodulation is properly tuned for the linear approximation to work with enough precision (hence ϕ_{corr} small enough).

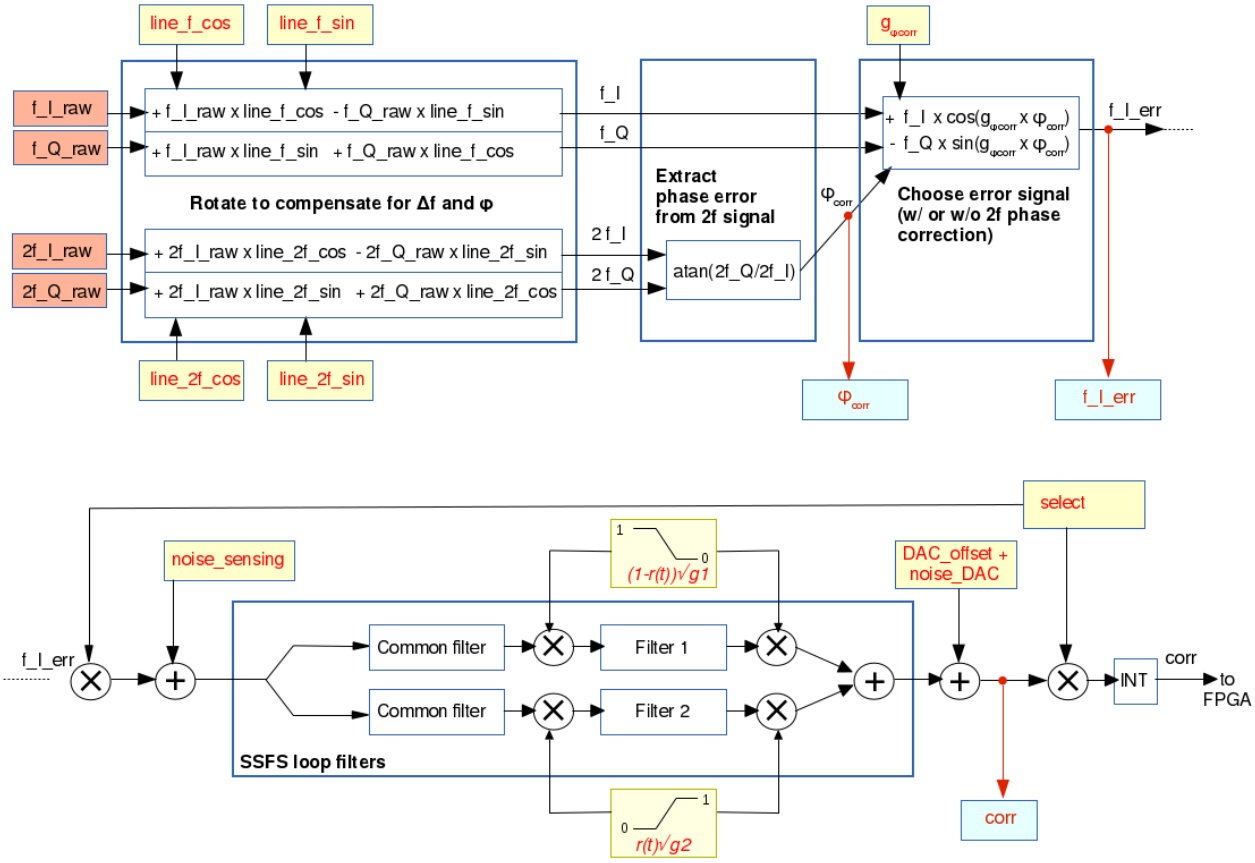


Figure 4: Details of the SSFS algorithm running in one DSP. The input channels are shown in the red and yellow boxes. The output channels are shown in the blue boxes. The first line details the second stage digital demodulation algorithm. The second line details the SSFS loop filtering and the final offset, noise and selection gain applied to the output correction before it is sent to the FPGA.

4.3.3 Correction of the phase noise

The option to slightly rotate the demodulation output at frequency f to correct for the phase noise estimated with the $2f$ signals is provided. A single external channel $g_{\phi corr}$ being equal to 1 or 0 is used to select the application or not of this correction[4].

The rotation needs to compute sine and cosine operations. These operations will also be processed using linear approximations. Again, this requires that the ϕ_{corr} values are small enough.

At this level, the SSFS error signal f_I_err is built. Some "sensing" noise can be added to this error signal. It is provided as an external channel.

4.3.4 Application of the SSFS loop filter

The SSFS error signal is then filtered to build the SSFS correction signal.

It is planned to separate the SSFS loop filter into two filters in series: a common filter which is running continuously and may have slow response, and a second filter (boost filter) which can be modified on-fly. In practice, two boost filters are running in parallel to allow to change the filter parameters on-fly.

The filters are IIR filters whose coefficients are computed taking into account frequency warping. The maximum of order of the filters running along a pipeline (i.e. common filter + boost filter) is 16 in total. The repartition of the biquad between the common and the boost filters needs to be choosen.

The common filter coefficients cannot be modified on-fly. Since the common filter may contain some integrator, it may diverge when used out-of-loop. Hence the error signal is multiplied by the channel $select\_Bi$ before the filter so that the filter has null inputs when it is out-of-loop.
The filter can be reseted by applying 0 input during a long-enough time. In practice, it is more efficient in the 2-core DSP to run two identical "common filters" with the same input and hence same output, as shown in the figure 4.

In order to allow on-fly change of the boost filter coefficients, two filters running in parallel are processed in the DSP: one is applied on the data to compute the correction signal while the other can be modified and prepared before the on-fly transition. The filter coefficients are provided to the DaqBox and DSP as configuration TOLM packets.

The transition from one boost filter to the other one is based on two external channels. Assuming the transition is from filter 1 to filter 2, these two channels are expected to move synchroneously from $\sqrt{g_1}$ to 0 for filter 1 and from 0 to $\sqrt{g_2}$ for filter 2, where g_i is the gain of the filter i. This channel is applied at both the input and output of the filter so that:

[4] In case of phase sign flip, this channel could be set to -1 instead of $+1$?

- the filter gain g_i is properly applied in the overall pipeline,
- the output of the filter that is not used is 0,
- the input of the filter that is not used is also 0, which resets the filter registers after some time.

At this level, the correction signal from a given photodiode pipeline is built.

Commands to modify the filter parameters and reset the filters – The coefficients of the filters are stored in on-board registers. In order to modify the coefficients or reset the filters, some external information must be received. To this aim, it is planned to receive dedicated TOLM configuration packets.
Sending zeros to the filter input is enough to reset the filter after some time in general, but this does not reset the filter in case of bad behavior, for example if *nan* were computed at some time.

Different external information is needed:

- *reset all filters*: resets the common and boost filter registers,
- *load coefficients for common filter*: loads the new coefficients for the common filter and resets the registers of the the common and boost filters.
- *load coefficients for filter 1*: loads the new coefficients for the filter 1 and resets its register.
- *load coefficients for filter 2*: loads the new coefficients for the filter 2 and resets its register.

Some automatic reset of the filter registers are also planned:

- when $select\_Bi$ reaches 0: reset the registers of the all the filters,
- when the channel $((1-r) \times \sqrt{g1})$ reaches 0, reset the registers of filter 1,
- when the channel $(r \times \sqrt{g2})$ reaches 0, reset the registers of filter 2.

4.3.5 Photodiode selection

An external channel ($noise\_offset\_DAC$) can be added to the raw correction signal to correct for the DAC offset if necessary, and add some noise directly to the DAC input for tests.

At this level, two correction signals are computed, one in each DSP. Only one of them must be used in loop. An external selection channel $select\_Bi$, with values between 0 and 1 is used to select the photodiode output: one DSP output is multiplied by $select\_Bi$ while the other one multiplied by $1-select\_Bi$. Sending a ramp signal in the selection channel allows a smooth transition between the two photodiode inputs.

The two extended-float correction signals are then converted to integer Bi_corr corresponding to the DAC dynamic and sent to the mezzanine FPGA on the DaqBox mother board. The sum of both signals is done in the FPGA and is used as input for the DAC chip that converts the error signal to an analog $SSFS\_corr$.

4.4 List of output channels

Some output channels can be sent from the DSPs to the RTPC for monitoring. The channels will be sent at 1 MHz (as they are generated in the DSP processing) in packets at lower frequency, probably of the order of 50 kHz. The data are then low-pass filtered and decimated down to frequencies of the order of 10 kHz in the RTPC. Some samples of data at 1 MHz could be stored in the raw_full data stream for debugging.

The six following channels (three per DSP) will be monitored:

- $B2\_corr$ and $B4\_corr$ since these are the SSFS correction channels. In principle, except during transitions from one photodiode to the other, one channel must be equal to 0 while the other one is the so-called $SSFS\_corr$ channel.
  The channel is probed after the SSFS filters but before the $select\_Bi$ gain is applied so that the filter behavior can be studied offline looking at the transfer function from $noise\_sensing$ to Bi_corr.

- $B2\_f\_I\_err$ and $B4\_f\_I\_err$ to monitor the demodulation process. The transfer functions from these channels to the Bi_corr channels also permit to check the DSP filtering.

- $B2\_\phi corr$ and $B4\_\phi corr$ to monitor the linear approximation of the $2f$ phase noise.

4.5 Synchronisation of input channels

For the second stage demodulation stage to work properly, it is necessary to keep a fixed delay (hence phase) between the I_raw, Q_raw data incoming in packets at 1 MHz and the external $line$ data incoming in packets at lower frequency (to be defined, between 20 kHz and 100 kHz). In order to provide the stability of this delay, the following synchronisation will be setup in the SSFS DaqBox, following the sketch shown in figure 5.

- the data arriving at 1 MHz must be pushed towards the DSPs as soon as they arrive

- the data arriving at a lower frequency must be pushed towards the DSPs at the same rate but in sync with the 1 PPS.

Attention must be paid so that the 1 MHz incoming packets are not delayed by the other input packets (in the MuxDemux stage of the DaqBox) so that they are always sent in-time to the DSP.

5 SSFS_Ctrl: a dedicated RTPC to control/monitor the SSFS input/output channels

A dedicated RTPC is planned to control and monitor the SSFS computations. A direct TOLM link from this RTPC to the SSFS DaqBox is needed in order to exchange 1 MHz packets with data sampled at 1 MHz.

An Acl process (called SSFS_Ctrl) running on this RTPC will provide the 20 external channels that are shown in yellow boxes in the sketch in figure 3. These channels will be generated with sampling frequency of 1 MHz and sent into packets at lower frequency, of few tens of hertz (100 kHz in the example of figure 5).

Most of the channels are constant once the SSFS loop is engaged: 2f correction gain, filter gains and selection, photodiode selection ; Some other channels are noise signals for tests.
The most critical signals are the *line* channels used for the second stage demodulation of the photodiode signals. In particular, they contain the demodulation phase information. Hence the synchronisation between these channels and the 1 MHz packets from the photodiodes must be properly controlled (see section 4.5). Moreover, in order to keep the SSFS filters independent from the gain of the photodiode channel, the sine and cosine values can be multiplied by a proper coefficient. It is needed to know the slope of the photodiode DSP inputs in the SSFS_Ctrl process. It is planned to add in Acl software the possibility to get the slope co-

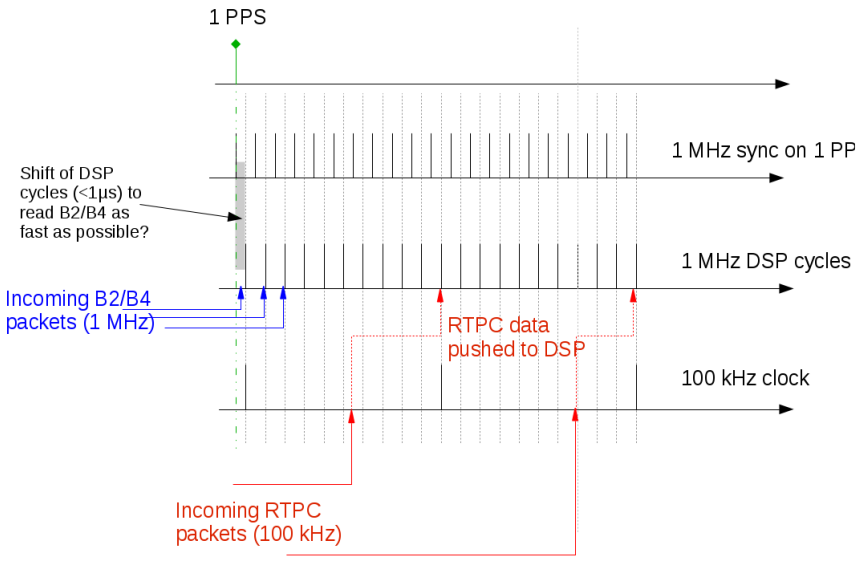


Figure 5: Synchronisation of the incoming packets with the DSP cycles and 1 PPS (example in the case of packets coming from the RTPC at 100 kHz)

efficient of some channels from the dictionary information, even if the data do not reach the RTPC where the Acl process is running[5].

The figure 6 shows the needed TOLM connections and process exchanges for the SSFS loop. It is proposed that the LSC process (or an automation slow process) is the master of the loop activation. Hence the LSC process generates a trigger channel that is used locally and sent to the SSFS_Ctrl and ISYS processes to trigger synchroneous loop filter transitions.

- the photodiode DaqBoxes are connected directly to the SSFS DaqBox for sending 1 MHz data sent in 1 MHz packets.
- the ISYS process receives the data from injection and from the RFC and B2 beams. It sends the TOLM packets to the DSPs for the IB and MC suspensions, and for the beam-pointing control.
- the LSC process receives the data from all the other beams (B4, B1*, B5, B7, B8). It sends the TOLM packets to the DSPs for all the other suspensions (PR, BS, NI, WI, NE, WE, SR, OB).
- the SSFS_Ctrl process sends the TOLM packets to the DSP for the SSFS. The packets will contain in total 20 channels sampled at 1 MHz. The packets will be sent as fast as possible, probably around 50 kHz. To allow such fast exchange, SSFS_Ctrl runs in a decicated RTPC.
- the SSFS loop master (LSC or some automation process) sends a channel (or different channels if needed) to ISYS and SSFS_Ctrl processes to activate/de-activate the SSFS loop, and select the appropriate photodiode, filter, ...
- a low-frequency copy of the $SSFS\_corr$ signal must be sent to ISYS process to control the MC suspension when the SSFS loop is engaged. This channel is sent from SSFS_Ctrl to ISYS. The content of this channel can be computed by two means:
  - the SSFS DSP sends back the $B2\_corr$ and $B4\_corr$ signals at 1 MHz ; they are low-pass filtered and decimated down to 10 kHz in SSFS_Ctrl ; the one being used as $SSFS\_corr$ is then sent to ISYS.
  - an Acl process SSFS_DSP_LF is setup to run the same code as the SSFS DSP but at a lower frequency ($\sim$ 10 kHz) on the RTPC. It generates the low frequency $SSFS\_corr$ signal sent to ISYS. For this process to run, it is needed that the output of the B2 and B4 demodulation mezzanines are sent to this RTPC. Having such a process and comparing its output with the DSP output could be useful to monitor the DSP processing.

[5] In the version v0r14p1, the channel properties are extracted from the channel buffer, hence only for channels read as input of the process and arriving on the RTPC

- we did not see yet any reason to send back channels from SSFS_Ctrl to LSC?

The delay of such a loop can be estimated to less than 6 μs??????as the sum of the hardware and propagation contributions summarized in table 1.

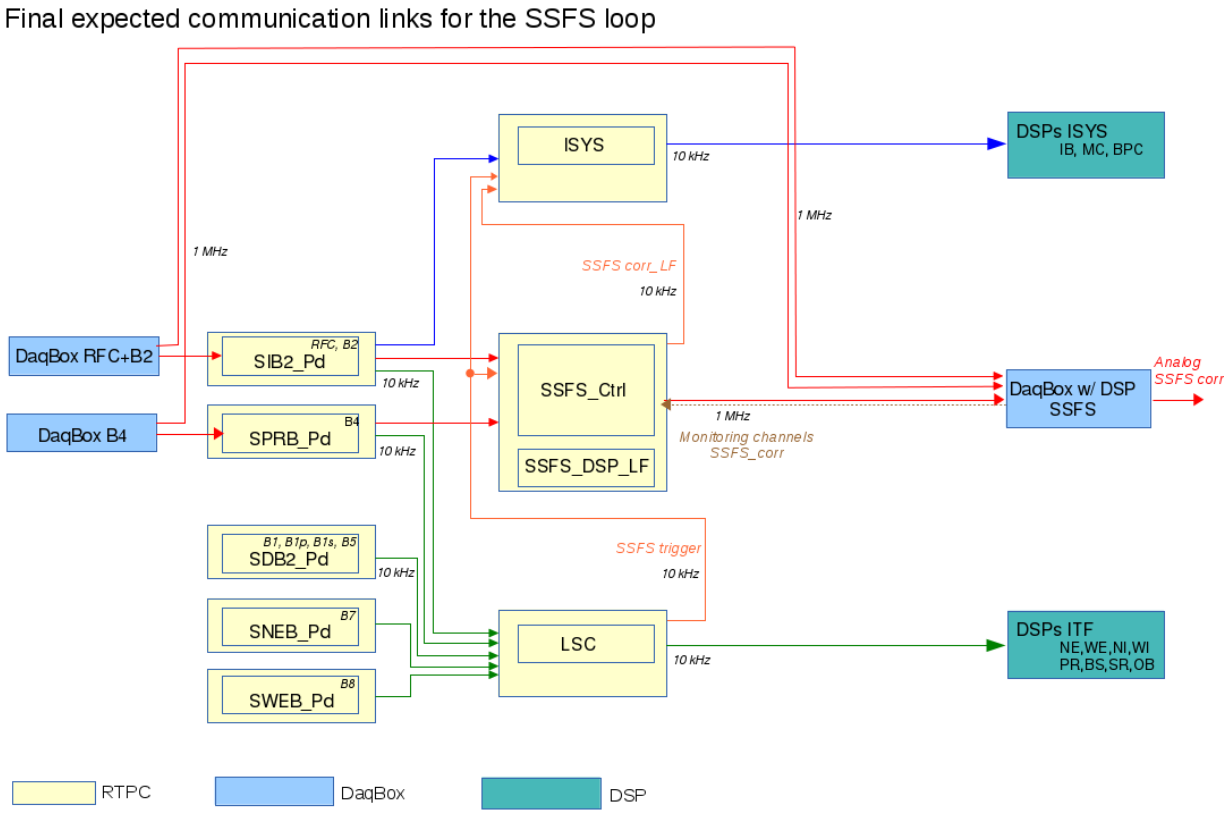


Figure 6: Sketch of the proposed TOLM connections from the photodiode acquired in DaqBoxes to the actuator DSPs for the suspension and the SSFS. Links for ISYS and LSC are shown in blue and green respectively. Emphasis is put on the SSFS whose links are shown in red and orange. Sampling frequencies of the main channels being exchanged are written.

6 Initial test setup with a RTPC emulating the DSP

In order to perform SSFS tests while the SSFS DSP firmware is being prepared (as of today, September 2016), a slightly different setup has been setup. It is shown in figure 7.

The main difference is that an Acl process called SSFS_DSP is run to emulate the DSP, but at a lower frequency. It runs on another decicated RTPC. It must be possible to run this

	Step	Latency (ns)
Photodiode DaqBox	Analog anti-alias filter (ADC 400 MHz)	10
	ADC to FPGA transfer	70
	FPGA demodulation (1 MHz output rate)	1000
	Filtering in the DSP	0 (skipped)
	DaqBox Mux/Demux	500
Propagation	50 m fiber (Pd DaqBox to SSFS DaqBox)	250
SSFS DaqBox	DaqBox Mux/Demux	500
	FPGA and DSP I/O	1600
	DSP computation cyle	1000
	Writing data to DAC5781	360??????
	Analog anti-image filter	500??????
	Total	$\sim$ 5800??????

Table 1: Preliminary estimation of the expected delay due to the electronics in the SSFS loop.

process at least 50 kHz. This RTPC receives the raw data from the photodiodes and the from the process SSFS_Ctrl as if it was the DSP. It also sends back monitoring data to SSFS_Ctrl. After the computation of the DSP algorithm in this process, the generated $SSFS\_corr$ channel is sent at 1 MHz to the SSFS DaqBox and directly applied to the DAC for the conversion to analog signal.

For the tests, we propose:

- to setup the connections as drawn in figure 7.
- to setup the configuration of SSFS_DSP process so that the logic is following exactly the DSP algorithm and the inputs are limited to the planned DSP inputs. This will ensure that the tests can trigger any error in the planned algorithm.
- to setup the SSFS_Ctrl process as close as possible as it will be in the final setup.

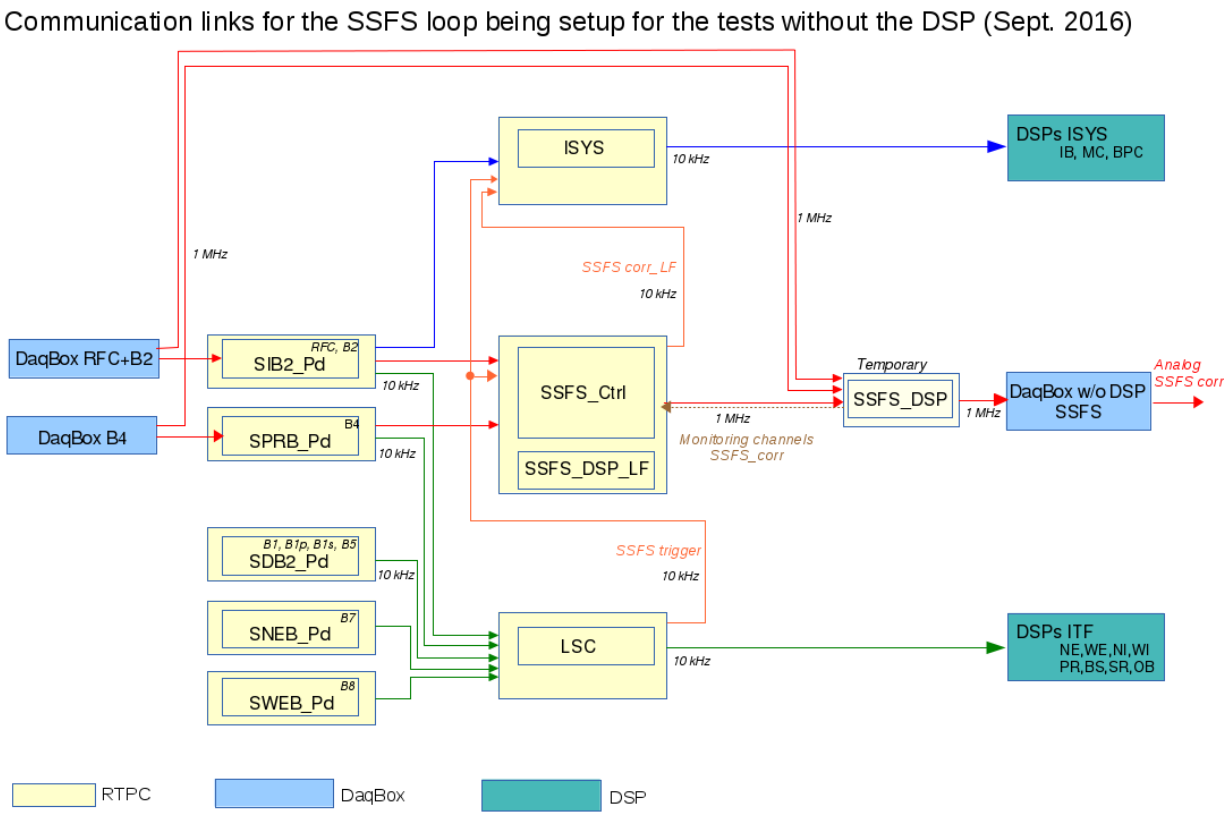


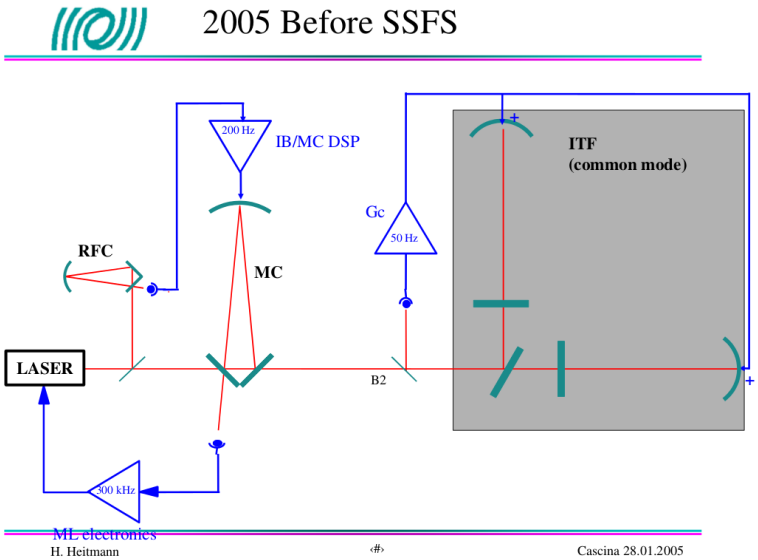
Figure 7: Sketch of the proposed TOLM connections from the photodiode acquired in DaqBoxes to the actuator DSPs for the suspension and the SSFS for the tests while the SSFS DSP is not yet ready to run online. Links for ISYS and LSC are shown in blue and green respectively. Emphasis is put on the SSFS whose links are shown in red and orange. Sampling frequencies of the main channels being exchanged are written.

References

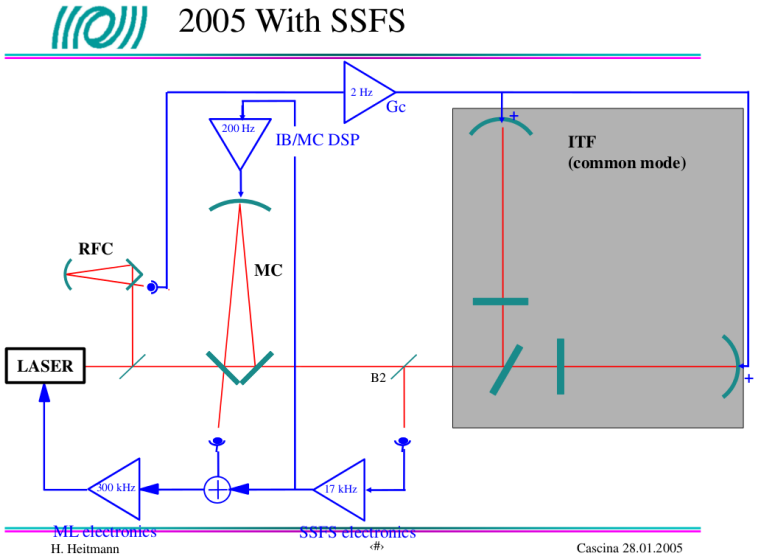
[1] N. Letendre, L. Rolland, *et al.*, "Preliminary DaqBox Specifications," vol. VIR-0108A-12, Mar. 2012.

A Sketch of the ISYS loop in Virgo, w/o and w/ SSFS

The figure 8 shows the frequency stabilisation loops before and after the SSFS loop is engaged as it was setup in initial Virgo in 2005. The SSFS loop was analog.



(a) Virgo in 2005: frequency loop before SSFS loop is engaged

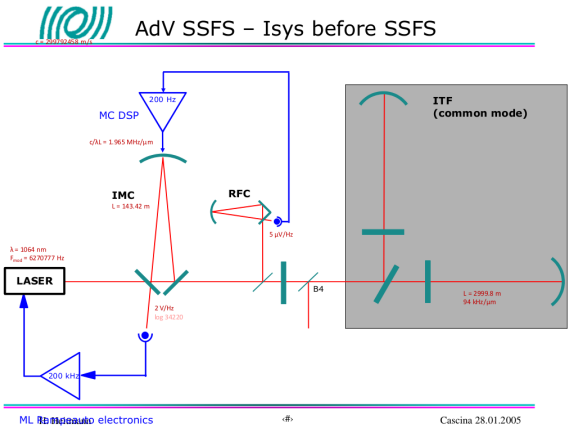


(b) Virgo in 2005: frequency loop with the SSFS loop engaged

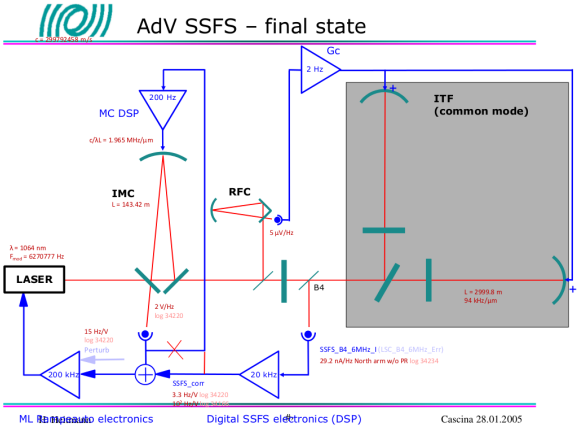
Figure 8: Laser frequency stabilisation control loops setup in Virgo in 2005, before and after the SSFS loop is engaged.

B Sketch of the ISYS loop in Advanced Virgo, w/o and w/ SSFS

The figure 9 shows the frequency stabilisation loops before and after the SSFS loop is engaged as it is being setup in AdvancedVirgo in 2016 The SSFS loop is digital. The SSFS correction signal SSFS_corr cannot be read anymore from the analog electronics (crossed-off blue line), hence it has to be sent digitally to the Sc_MC DSP (new red line).



(a) Advanced Virgo in 2016: frequency loop before SSFS loop is engaged



(b) Advanced Virgo in 2016: frequency loop with the SSFS loop engaged

Figure 9: Laser frequency stabilisation control loops being setup in Virgo in 2016, before and after the SSFS loop is engaged.