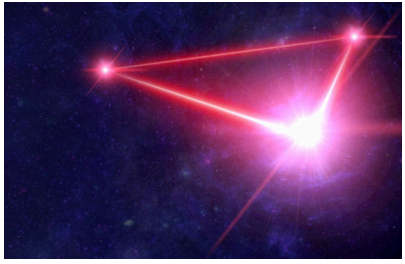
LISA Phasemeter based on MicroTCA as ground-support equipment

Signal distribution and phasemeter software

Christian Darsow-Fromm[1] Johannes Zink[2] Michael Büchler[2] Holger Schlarb[2] Oliver Gerberding[1]

[1]Universität Hamburg [2]DESY

06.12.2023



Gravitational-wave detection

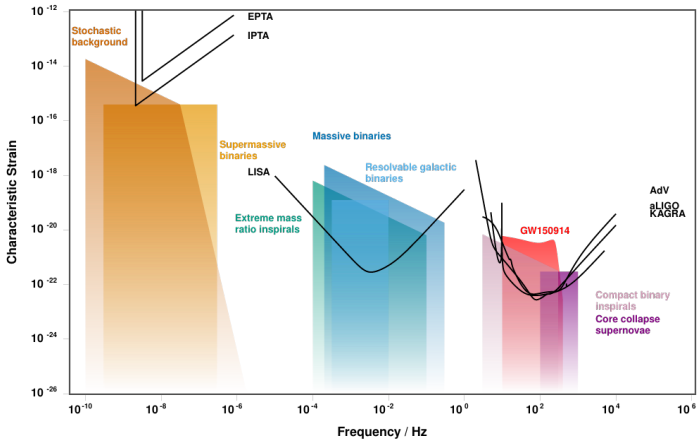


Figure: Gravitational-wave detectors and sources (gwplotter.com, arXiv:1408.0740)

Space-based detector: LISA
Metrology challenge

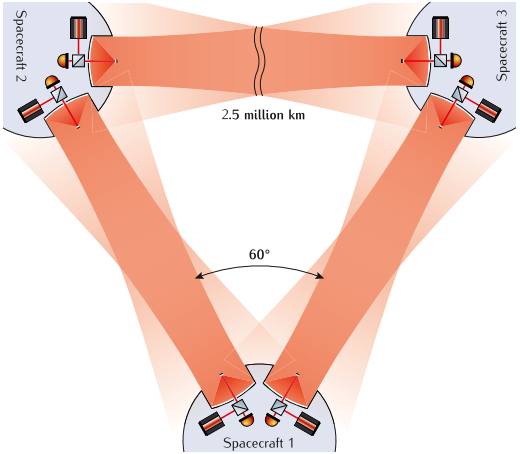


Figure: LISA constellation (source)

Constellation parameters

$$L = 2.5 \cdot 10^9 \text{ m}$$

$$h = \frac{\Delta L}{L} = 4 \cdot 10^{-22}$$

$$\Delta L = 1 \text{ pm}$$

Heterodyne interferometry

$$\Delta v * \max \approx 15 \frac{\text{m}}{\text{s}}$$

$$\Delta f * \text{D} \approx (15 \pm 8) \text{ MHz}$$

$$\Rightarrow \Delta\varphi < 6 \text{ µrad } @ \text{ mHz}$$

LISA
Instrument and signals

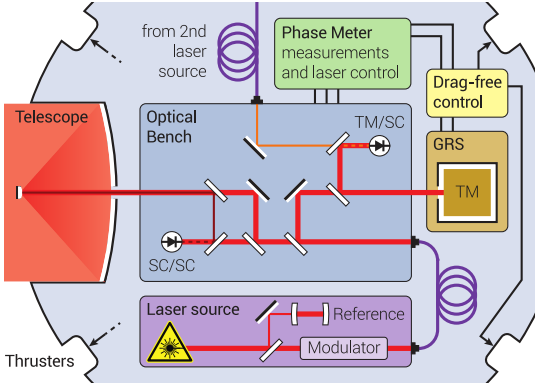
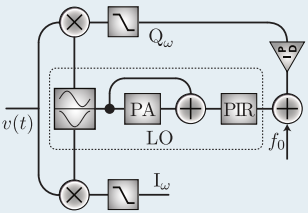
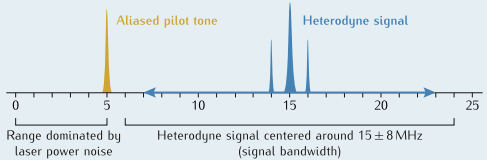


Figure: Optical bench (source)

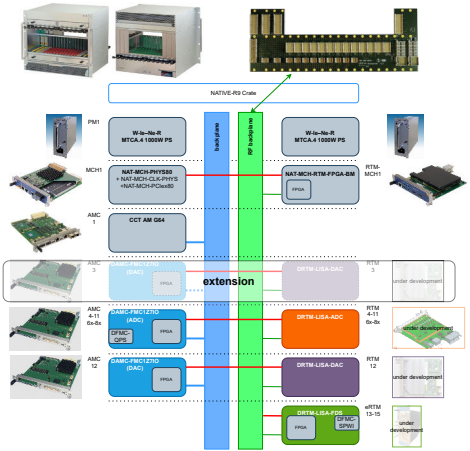
PLL



ADC tones



Ground-support equipment phasemeter
MicroTCA phasemeter design



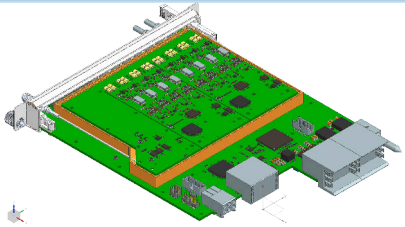
FPGA

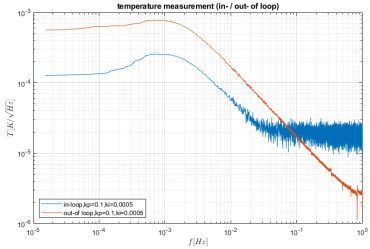
- DAMC-FMC1Z7IO as central AMC module for our fast algorithms and processing

RTMs

- RF Backplane & custom eRTM (DRTM-LISA-FDS) for generation & distribution of clocks and pilot-tone
- Custom RTM for high fidelity phase measurements (DRTM-LISA-ADC)
- Custom RTM for laser control (DRTM-LISA-DAC)

Ground-support equipment phasemeter
Custom DRTM-LISA-ADC





- 8 readout channels
- Modular design to reduce risk
- Pilot-tone for jitter correction distributed from eRTM via RF backplane & Zone 1
- Phase-noise critical parts in the measurement chain are thermally stabilized and sealed
- Active temperature control with local software controller & linear heating circuit reaches $<1\,\mathrm{mK}/\sqrt{\mathrm{Hz}}$ stability in test-boards
- In production, waiting for hardware

Developed by Johannes Zink

Ground-support equipment phasemeter
Data handling stages

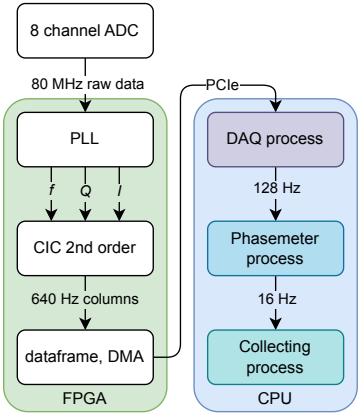


Figure: Data rates and data flow

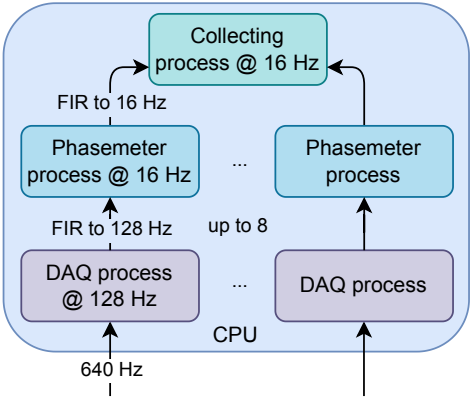


Figure: Software structure

Ground-support equipment phasemeter
Software

Realtime programming

- Parallel realtime programming in Python
- Fine-graded realtime priority necessary for different processes
- Works robustly without additional memory allocations with disabled garbage collector
- Custom ringbuffer written in C++ with `xtensor`, more parts can be moved

Configuration

- Automated YAML configuration to define the state of the phasemeters
- Columns of the FPGA dataframe are defined in `RDL` and exported to `YAML` and `VHDL`

Logging

- Python logging is not thread-safe
- Own extension with custom log messages and collecting process to write to logfile.

One issue

- Sometimes the 128 Hz process needs 15 ms instead of 1 ms for the same calculation and stays in that state
- Then, every part of the process is slower than usual

Ground-support equipment phasemeter
Data transfer

Data transfer

- Library `libudmaio` over PCIe
- xdma or userspace I/O

Parallel ADC bypass

- 80 MHz ADC data can be read out directly
- Second DMA used with completely independent parallel readout

Dataframe definition

- Column definition in `RDL`
- Compiled to `VHDL` with SystemRDL
- Exported to `YAML` for phasemeter software

```
regfile pll {
  reg iq_val { field {} f[64]; };
  iq_val frequency; };
regfile channel {
  pll pll[2];
  reg { field {} f[64];
  } dc; };
```

```yaml
columns:
  timestamp:
    addr: 0
  channel_0:
    dc:
      addr: 72
    pll_0:
      frequency:
        addr: 8
```