AO4ELT7 - June 2023 - Avignon

The RTC for METIS SCAO

Martin Kulas[a], Hugo Coppejans[a], Horst Steuer[a], Thomas Bertram[a], Carlos Correia[b], Philip Neureuther[c], and Florian Briegel[a]

[a]Max Planck Institute for Astronomy, Heidelberg, Germany
[b]Space ODT - Optical Deblurring Technologies Ltd., Porto, Portugal
[c]University of Stuttgart, Germany

ABSTRACT

The Mid-infrared ELT Imager and Spectrograph (METIS) is is one of three first-generation science instruments for the Extremely Large Telescope (ELT) and has recently completed its final design phase. Its Single Conjugate Adaptive Optics (SCAO) system will provide the performance of an extreme adaptive optics system which enables high contrast imaging observations in the thermal/mid-infrared wavelength domain (3 µm – 13.3 µm).

The Real-Time Computer (RTC) is the central component of the SCAO real-time control system. It executes the time critical wavefront control loop as well as associated control tasks by processing the data from the pyramid wavefront sensor and controlling the set of ELT actuators dedicated to adaptive optics. A total of up to 4,866 commands to be computed at a loop rate of up to 1 kHz imposes a number of demanding constraints in terms of memory throughput and computing power on the Hard Real-Time Core (HRTC), which employs GPU acceleration for the bulk of computations.

Several auxiliary functions need to be in place to establish and maintain the quality of the wavefront correction. Among them are the control of the pupil position, the compensation of misregistration and of non-common path aberration, and the adaptation of the temporal control parameters.

The main wavefront control loop has been prototyped to verify timing requirements. A median RTC computation time of 382 µs was achieved for a 300k samples (5 minutes) run. The results are presented in this paper together with the foreseen RTC hardware and the software deployment within the SCAO Control System

Keywords: ELT, METIS, Adaptive Optics, real-time computer, RTC, graphics processing unit, GPU

1. METIS SCAO

The SCAO subsystem of METIS ensures observations near the diffraction limit [1]. It uses the light of a single near-infrared (NIR) source located at or in a limited field around the science target position to measure the phase of the incoming wavefront and applies corrections in real-time by controlling the adaptive mirrors of the ELT.

For further author information: send correspondence to Martin Kulas via e-mail to kulas@mpia.de.

Several distributed entities are involved in this Adaptive Optics (AO) system: in the instrument domain, the SCAO Subsystem consists of the *SCAO Module* and the *Adaptive Optics Control System (AOCS)*. Further entities that are essential for SCAO are located in the telescope domain. Figure 1 shows a simplified block diagram for METIS SCAO.

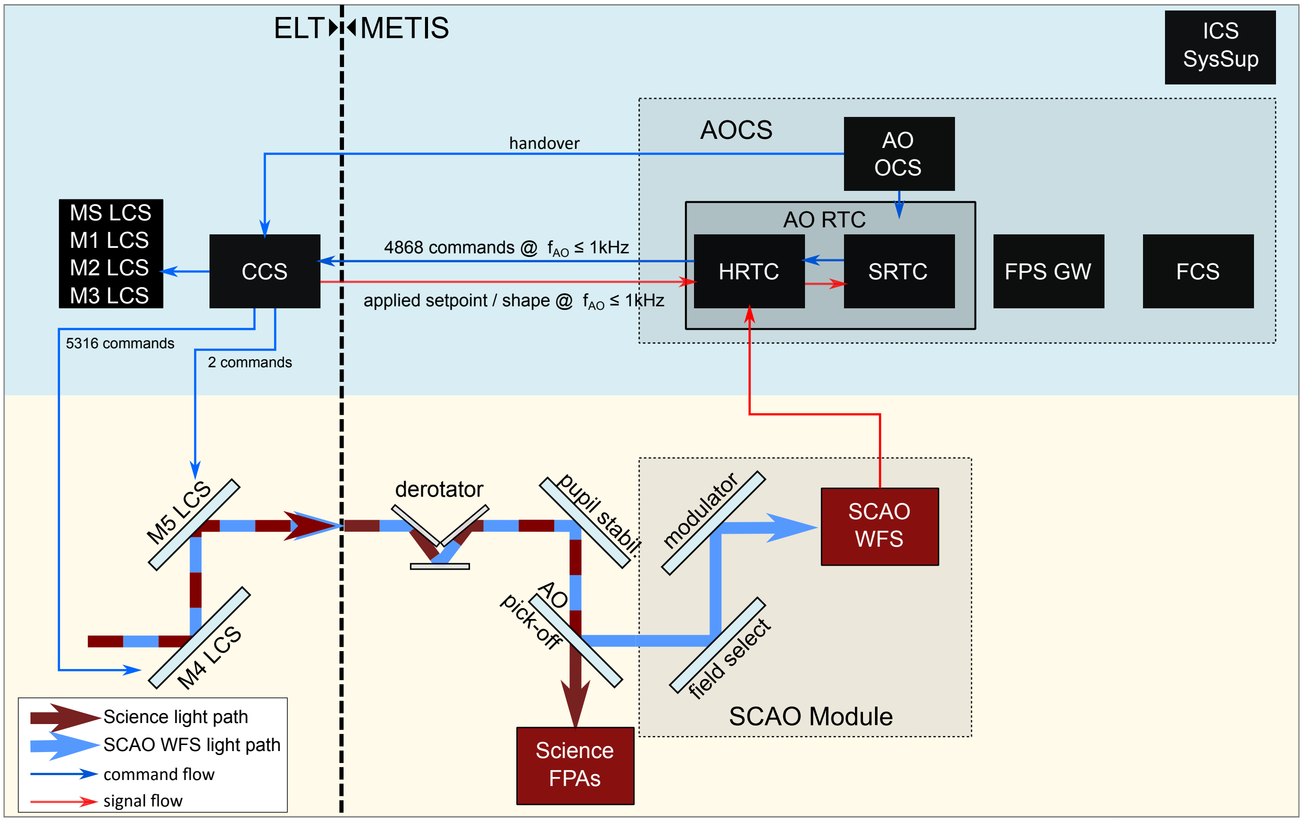

Figure 1: Simplified block diagram of the SCAO system: The Adaptive Optics Control System (AOCS) and the SCAO Module (shaded boxes) are the entities of the SCAO system that belong to the instrument domain. The key entities for the real-time correction of the incoming light of the 'ELT' domain are located on the left side of the figure.

In a closed wavefront control loop, the blue, NIR light is used to measure the instantaneous residual wavefront error by the Wavefront Sensor (WFS). The measurement signal is analyzed by the RTC, and a computed correction is sent to the Central Control System (CCS) to be applied with the ELT quaternary mirror (M4) and ELT tip-tilt field stabilisation mirror (M5) via a Local Control System (LCS). The Focal Plane Sensor Gateway (FPS GW) provides science images to auxiliary AO loops and the Function Control System (FCS) is responsible for controlling all instrument devices, except the detectors.

The *SCAO Module* is located inside the cryostat of METIS. A cold dichroic AO pick-off mirror immediately in front of the SCAO Module is used to separate the near-infrared part of the light, which is used for wavefront sensing. The SCAO Module provides a Pyramid Wavefront Sensor (P-WFS) as well as opto-mechanical actuators for field selection and modulation of the Natural Guide Star (NGS) in the field of view.

The *AOCS* hosts the main wavefront control loop as well as a number of secondary control loops. A key entity of the AOCS is the RTC. Its HRTC is used for the time critical aspects of the wavefront control loop: wavefront sensor signal processing, wavefront reconstruction and the determination of correction commands that are applied with the M4 and M5 mirrors via the CCS. The Soft Real-Time Cluster (SRTC) supervises and optimizes the HRTC operation. Less time critical control tasks are realized outside of the RTC. The AO Observation Coordination System (AO OCS) is the gateway for the METIS ICS to the AOCS. Its task is to coordinate the activities inside of the AOCS.

The environment of the METIS RTC is depicted in figure 2 and it is described in the table below:

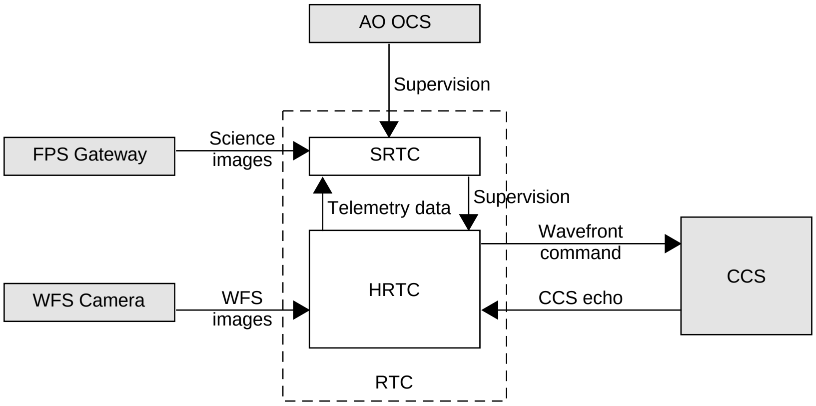

Figure 2: METIS RTC system context.

Name	Description
WFS camera	It delivers WFS images to the HRTC using the network protocol Real-Time MUDPI Stream Protocol (RTMS).
FPS GW	It provides preprocessed science images to the RTC.
CCS	It receives CCS wavefront commands (`Ao:WavefrontCmdStream`) sent from the HRTC and transmits the command echo (`ao:m4_setpoint_applied`). In both cases, the used network protocol is RTMS.
AO OCS	It supervises the SRTC.

2. HRTC

2.1 Problem Size

The HRTC is responsible for performing the Wavefront Control (WFC) loop with tight timing constraints [4]. It computes a wavefront command from each WFS image on every loop cycle. The table below shows the WFC properties for METIS SCAO for the worst case scenario. For example, depending on the results of further analysis, it would be possible that a lower number of modes may be sufficient, which in turn would lead to smaller command matrix size.

Properties	Size
WFS type	P-WFS with four sides
WFS camera frame rate	1 kHz
WFS camera frame size	192 x 192 pixels as int32
Pupil diameter on WFS detector	90 subapertures
Number of subapertures per pupil	6,376
CCS wavefront command vector length	4,868 (includes the measured tip and tilt)
Command matrix size in worst case	4,866 x 12,752 $\rightarrow\approx$249 MB as float32
Temporal controller	PI controller for TT and HO modes
RTC computation time limit for WFC	909 µs

Using the properties above, the required computational demand for each WFS frame is given in the table below:

WFC task	**Computations [MFLOP/cycle]**	**Transferred memory amount/cycle [MB]**
Pixel calibration	0.1	1.3
Wavefront signal computation	0.1	0.7
Wavefront reconstruction (MVM)	124.1	248.3
Temporal control	< 0.1	0.4
Σ	124.4	250.7

The operational intensity of the WFC task is 0.497 [7]. The numbers in the table above show that WFC computing demand is dominated by the Matrix-Vector Multiplication (MVM) in float32.

Based on the computational demand, the performance requirements are derived below:

Performance aspect	**Rate**
WFC computational power	124.4 MFLOP / 909 µs = 137 GFLOP/s
Memory transfer throughput	250.7 MB / 909 µs = 276 GB/s
Pixel reception throughput	148 MB/s
CCS command throughput	22 MB/s
Telemetry throughput	315 MB/s

The most demanding performance aspect is the required memory throughput. For comparison, the standard computer main memory during Final Design Review (FDR) was DDR4-3200 Synchronous Dynamic Random Access Memory (SDRAM) which provides a maximum data rate of approx. 25 GB/s, i.e. eleven memory channels are required to achieve the HRTC memory throughput. On the computer market during FDR, a server-grade CPU contains normally six or eight memory channels, for instance eight memory channels by AMD EPYC3 7003 series.

Nowadays, high-performance CPUs such as AMD EPYC4 7004 series are equipped with up to 12 memory channels for DDR5-4800 SDRAM with about 38 GB/s. In theory, computers with these components are capable to solve the METIS HRTC tasks. Since that hardware was not available before FDR, we developed a METIS HRTC based on Graphics Processing Unit (GPU) accelerators as described in the next section.

2.2 Prototype Hardware

The HRTC hardware is a GPU server which consists of the components listed below:

Component	**Description**
GPU server	Asus Barebone ASUS ESC4000A-E10.
CPU	AMD EPYC 7542 with 32 cores (TDP: 225 W)
Main memory	512 GB DDR4-3200 (data rate: 25.6 GB/s)
GPUs	2x Nvidia A100 for PCIe version 4.0 with 40 GB HBM2 (board power: 250 W)
Power supply units	1+1 redundant 1600 W

The GPU Nvidia A100 has been chosen because it provides high data throughput. We have measured a GPU internal memory throughput of about 1.1 TB/s [*]. The data throughput from the host to the GPU has been measured to be about 26 GB/s. This value is in line with the maximum PCIe 4.0 x16 transmission rate of almost about 32 GB/s.

The power consumption has been measured under high load and regular WFC loop operation via IPMI [†]. High load is generated by running `gpu_burn 3600` [‡] and `stress -c 32 -d 8` [§]. In that case, the computer has consumed around 880 W. In the other load scenario of regular WFC loop operation the power consumption was about 400 W.

---

[*] Measured using CUDA Toolkit utility program : bandwidthTest

[†] The Linux tool `ipmitools` was used to read out the power consumption.

[‡] URL: https://github.com/wilicc/gpu-burn/

[§] URL: https://packages.debian.org/sid/stress

The complete GPU server had a price of about roughly 25 kEUR. The major cost contribution were the GPUs. Both of them cost in total roughly 16 kEUR. For the Manufacturing, Assembly, Integration and Testing (MAIT) phase, we will build the final HRTC with recent hardware such as DDR5 main memory and PCIe 5.0, taking into consideration that the hardware is compatible with the IT Server Standard by the European Southern Observatory (ESO).

2.3 Prototype Software

The HRTC main software structure is a pipeline as depicted in figure 3.

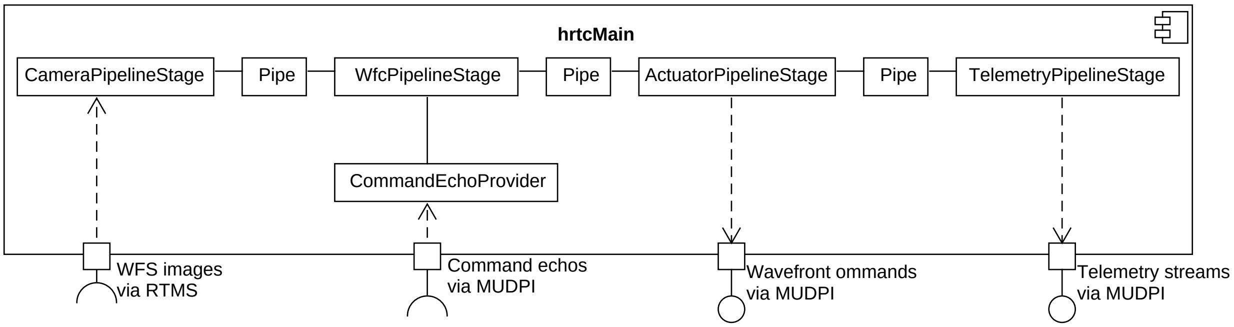

Figure 3: HRTC WFC computing pipeline structure.

The table below describes its building blocks:

Name	**Description**
`CameraPipelineStage`	It receives the WFS images via RTMS and passes it to the `WfcPipelineStage`.
`WfcPipelineStage`	It computes the CCS wavefront commands from the WFS images. The command matrix multiplication is distributed on GPUs. Some parts of the temporal controller are sped up by using the instruction set Advanced Vector Extensions 2 (AVX2).
`ActuatorPipelineStage`	It sends out the CCS wavefront commands. In the moment, the network protocol is MUDPI. In the future, RTMS will be used.
`TelemetryPipelineStage`	It sends out the telemetry records via the network protocol Multicast UDP Interface (MUDPI).
`CommandEchoProvider`	It receives the CCS echo stream via MUDPI and passes it to the `WfcPipelineStage`. In the future, MUDPI will be replaced by RTMS.
`Pipe`	It stores a configurable number of elements.

The current HRTC uses almost exclusively software products that are available in the ELT Development Environment for Linux. The only exceptions are the NVIDIA CUDA Toolkit and ZeroC Ice for the HRTC Control interface including the parameter update. It is foreseen to replace ZeroC Ice by a communication protocol available in the ELT standard such as ZeroMQ or HTTP.

2.4 Performance Experiment

The METIS use cases require that its HRTC computes the CCS wavefront commands quickly and transmits the complete telemetry data. Section 2.1 contains the HRTC performance requirements in numbers. The purpose of the experiment is to evaluate the current HRTC computing power.

2.4.1 Setup

The test setup resembles a possible setup at the ELT which is depicted in figure 4. In the main AO data path, the WFS camera sends out WFS images via RTMS over a 10 Gigabit per second Ethernet (10GE) network to the HRTC. The WFC computing pipeline receives the WFS images, computes CCS wavefront commands and sends them to a CCS simulator which then returns echos of the applied wavefront commands.

The telemetry data from the HRTC is collected by the telemetry sink. A separate computer (HRTC supervisor) performs the parameter update on the HRTC. In order to correlate events, the clocks on all computers are synchronized by signals from a Precision Time Protocol (IEEE-1588) (PTP) time server over the PTP network.

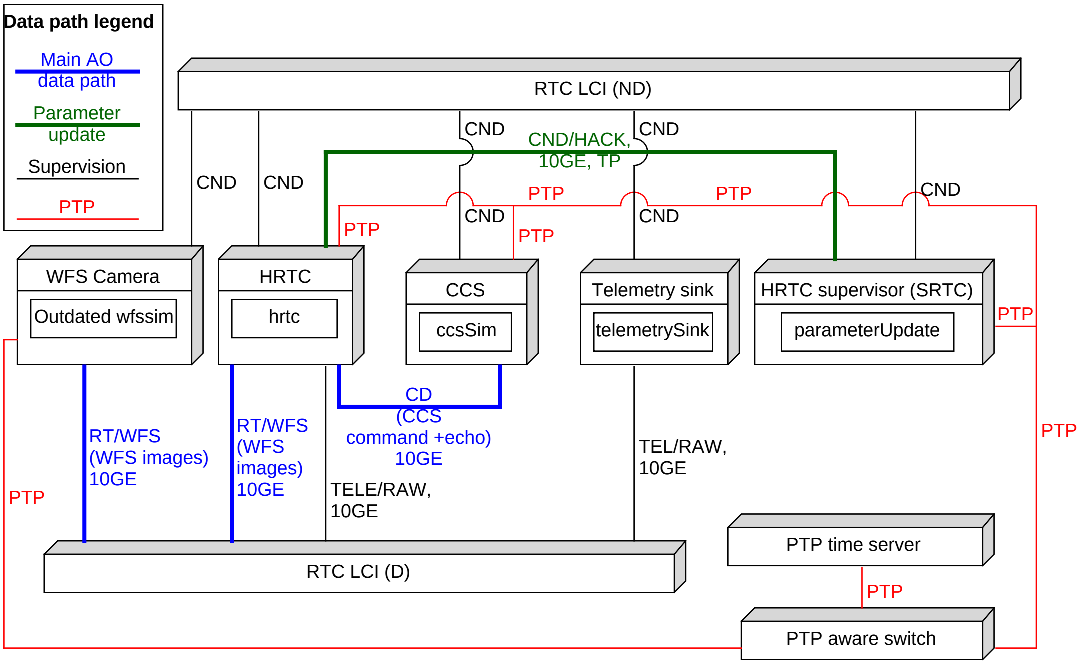

Figure 4: HRTC performance test setup as a deployment diagram. The nodes represent computers and are interconnected by a Ethernet switches that are called RTC Local Communication Infrastructure (LCI). While the RTC LCI (D) carries deterministic traffic, the RTC LCI (ND) carries non-deterministic traffic.
The network is divided into domains according to its traffic. The domains below are used: non-deterministic control network (CND), deterministic control network (CD), real-time network for WFS images (RT/WFS), network for raw telemetry data (TELE/RAW), a time distribution network (PTP) and a temporary domain for the parameter update (CND/HACK).

The nodes are described in the table below:

Node Name	Description
WFS Camera	It runs the not longer supported ESO `wfssim`.
HRTC	It runs the METIS HRTC computing pipeline.
CCS	It runs a minimal CCS simulator that generates echos from the CCS commands.
Telemetry sink	It runs a telemetry sink that captures HRTC WFC telemetry data and dumps timings such as the pipeline timestamps into files for later performance test analysis.

HRTC Supervisor (SRTC)	It runs the program `parameterUpdate` that uploads the Command Matrix (CM).
RTC LCI (ND)	It is a 1GE switch that acts as the non-deterministic RTC LCI.
RTC LCI (D)	It is a 10GE switch that is the deterministic RTC LCI.
PTP time server	It is the PTP grandmaster clock. The hardware is Meinberg LANTIME M1000.
PTP aware switch	It is a PTP aware Ethernet switch. The hardware is Hirschmann MACH104.

The networks CND, CD, RT/WFS, TELE/RAW and PTP are used according to ESO AO RTC common requirements. The network CND/HACK is a temporary solution that allows uploading parameters to the HRTC over a 10GE link since the current RTC LCI(ND) supports only 1GE. It will be replaced soon by a 10GE switch.

The network data streams below are produced during the experiment:

Stream	**Source**	**Sink**	**Throughput [MB/s]**	**Protocol**
WFS image	WFS	HRTC	148	RTMS (future: Multicast)
CCS wavefront command	HRTC	CCS	20	MUDPI (future: RTMS)
CCS echo	CCS	HRTC	20	MUDPI (future: RTMS)
GPU telemetry	HRTC	Tel. sink	<1	MUDPI
Timestamp telemetry	HRTC	Tel. sink	<1	MUDPI
WF error telemetry	HRTC	Tel. sink	20	MUDPI
Parameter update statistics telemetry	HRTC	Tel. sink	<1	MUDPI
CCS command telemetry	HRTC	Tel. sink	20	MUDPI
Subaperture intensities telemetry	HRTC	Tel. sink	26	MUDPI
Calibrated WFS pixel telemetry	HRTC	Tel. sink	148	MUDPI
WFS signal telemetry	HRTC	Tel. sink	52	MUDPI
CCS echo telemetry	HRTC	Tel. sink	20	MUDPI

The METIS HRTC transmits data with a throughput of total 306 MB/s into the network. The majority of that throughput is caused by MUDPI telemetry data with about 286 MB/s. In total, this HRTC receives data with a throughput of approximately 168 MB/s. Here, the WFS image stream dominates that input throughput with roughly 148 MB/s.

The HRTC publishes the WFS pixels as telemetry at WFS frame rate without any decimation because this setting allows assessing the worst case. It is highly likely that a decimation of the WFS pixel telemetry will be implemented in the productive HRTC.

During observation, the SRTC will update parameters on the HRTC on a regular basis. In order to assess the performance degradation during the parameter updates, the HRTC supervisor node runs the program `parameterUpdate` which updates the command matrix on the HRTC every three seconds. Since the command matrix is the largest parameter with a size of about 249 MB (4,866 rows times 12,752 columns in single precision), it will impact the WFC loop performance most.

2.4.2 Results

The experiment lasted for five minutes, i.e. the WFS simulator generated 300,000 images. The most important performance requirement of the METIS HRTC is the RTC computation time for the WFC loop. It must be below 909 µs. The current HRTC measures the RTC computation time by taking time stamps as depicted in the schematic timing diagram in figure 5.

Figure 6 shows the RTC computation time series and figure 7 depicts the histogram. The result is that all 300,000 samples were below the RTC computation time limit of 909 µs. The median was 382 µs and 99.99% of all samples did not exceed 493 µs.

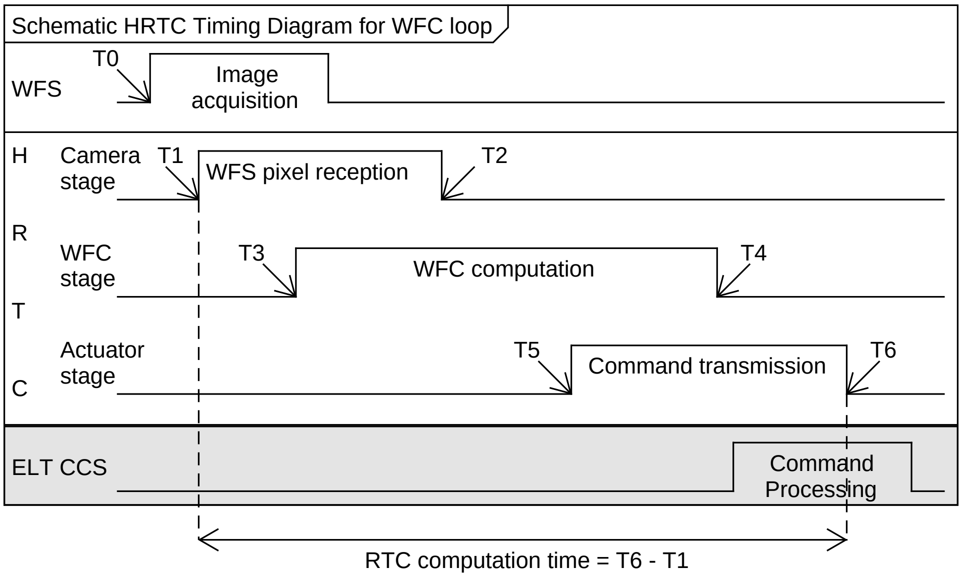

Figure 5: Schematic HRTC timing diagram for the WFC loop. In theory, it is possible that the tasks such as image acquisition, WFS pixel reception and WFC computation overlap. In the current METIS HRTC implementation, these actions occur sequentially because sequential processing is sufficient to meet the METIS performance requirements.
Timestamp T1 denotes the point in time where HRTC receives the first WFS data and timestamp T6 is the point in time where the last command data departs the HRTC.

One hundred parameter updates were conducted during this performance experiment. Every three seconds, the program `parameterUpdate` on the supervisor node updates of the CM on the HRTC. The first update started at frame counter 1,944 and the last update ended at frame counter 299,138.

The parameter update consists of two steps: Firstly, `parameterUpdate` sends the CM to the HRTC which then copies it to a passive parameter bank in the memory of the GPUs. In addition to the passive parameter bank, the HRTC has an active parameter bank in the GPU memory which is used by WFC loop during operation.

After the CM has been uploaded into the GPUs, `parameterUpdate` instructs the HRTC to switch between the two parameter banks. As a result, the former passive parameter bank becomes the active one.

The CM transfer from the supervision node into the GPU memory takes about 1.1 seconds. Of this 1.1 s, the HRTC needs roughly 0.3 s to copy the CM from main memory to GPU memory. Switching between the active and passive parameter banks takes less than 0.1 s from the perspective of `parameterUpdate`.

We assume that the parameter update task causes the peaks of the RTC computation time shown in figure 6 because we did not observe any such peaks in the time series of an comparative experiment without parameter updates. It needs to be investigated which part of command matrix update process impacts the RTC computation time.

The RTC computation time is defined as the time elapsed between the arrival of first WFS data at the HRTC and the departure of the last command data from the HRTC. Since we define that the HRTC network cards are the HRTC boundary, the arrival time of WFS data and departure time of command data must be measured on the network cards in order to compute the correct RTC computation time. However, in the current implementation, the HRTC user space process measures the timestamps for the RTC computation time and not the network cards. The measurement approach distorts the delay of the camera pipeline stage because the WFS camera simulator takes at least 117 µs to transmit the WFS image in a 10GE network which is in contradiction to the mean camera stage delay of about 105 µs in the experiment.

Some drops in the camera stage delays distort the RTC computation time such as frame 178,856. They are caused by fluctuations in processing the WFS camera image packets by both the operating system and the HRTC process. We expected that the distortions will be eliminated when taking timestamps on the network card.

We plan to repeat the experiment with network cards that allow timestamping incoming and outgoing network packets. For that purpose, we recently bought the network cards Intel X550-T2.

Many peaks are visible in the RTC computation time series. The RTC computation time peak at frame counter 182,059 was due to a slow command matrix copy step onto the GPU. The peak at frame 179,956 happened during the command matrix copy onto the GPU as well. Not all peaks are due to matrix copy onto the GPU. For instance, the peak at frame counter 184,855 occurred during reception of the command matrix via ZeroC Ice. The majority of peaks occurred in the WFC pipeline stage. All peaks occurred during the command matrix update process but they seem to occur randomly and not periodic. Although many peaks occurred, all of them were below the RTC computation time limit.

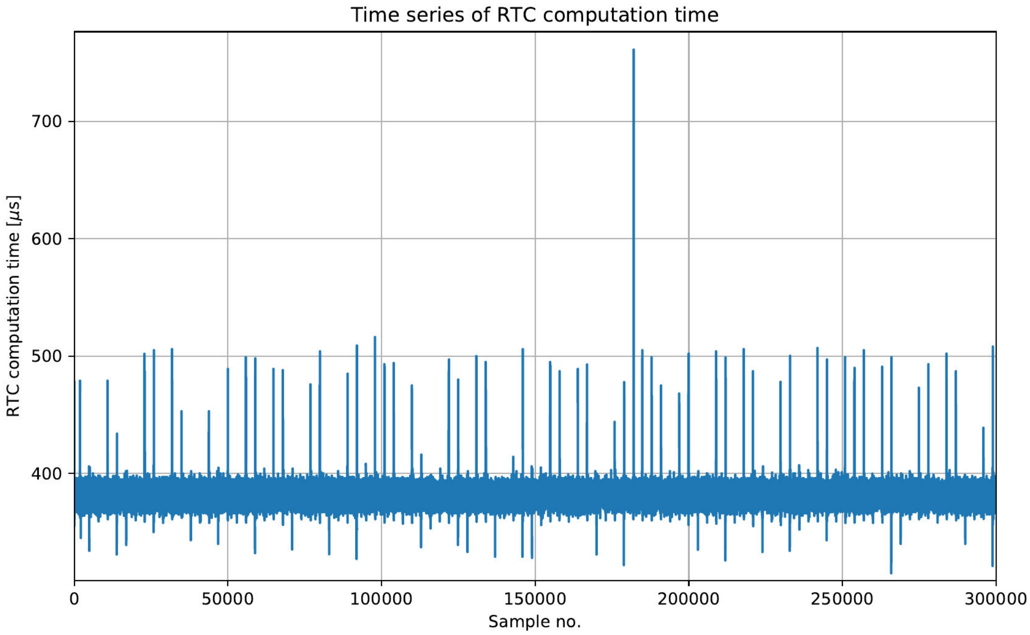

Figure 6: RTC computation time series in the performance experiment. Median: 382µs; mean: 381 µs; standard deviation: 6 µs, min: 315 µs; max: 761 µs; 99.99% percentile: 493 µs.

All 300,000 telemetry records were received by the telemetry sink which is a ordinary desktop computer with the Central Processing Unit (CPU) Intel Core i7-860 from the year 2009 and 8 GB of DDR3-1333 main memory. That fact demonstrates that moderate computer hardware is sufficient to receive and process the telemetry data stream.

The HRTC prototype fulfills our performance requirements. We are confident to use its design in the MAIT phase.

2.5 COMPASS Integration

An additional experiment was conducted to verify the numerical correctness of the HRTC prototype. In order to analyze the expected AO performance and to prove the control concept of METIS SCAO, a modified version of COMputing Platform for Adaptive optics SystemS (COMPASS) [5] was used as described in [4].

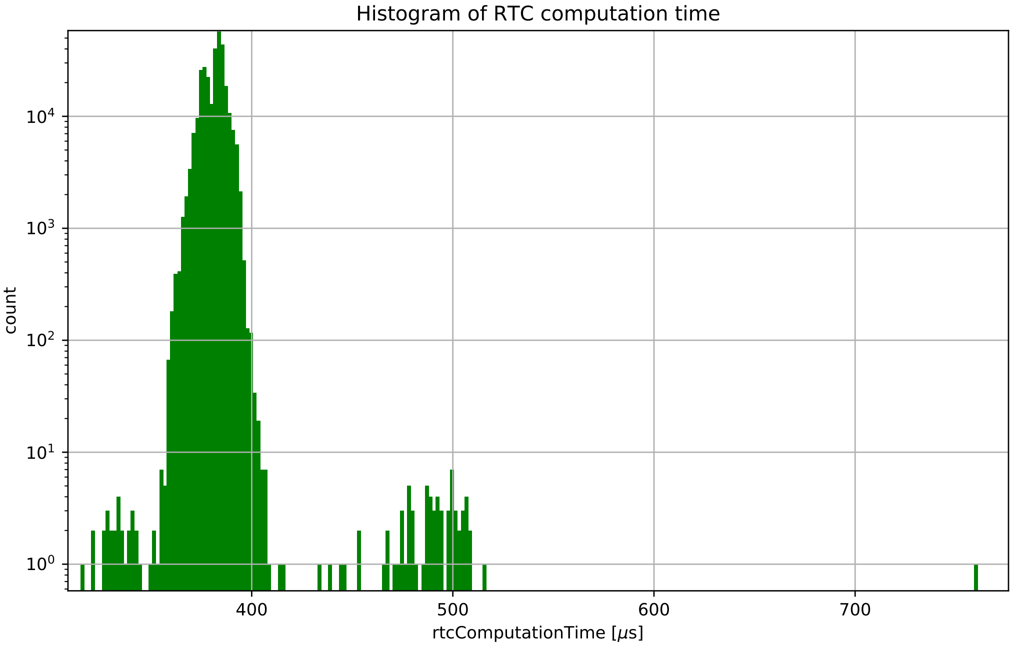

Figure 7: Histogram for RTC computation time in the performance experiment.

For this experiment the HRTC was integrated to run 'in the loop' of the COMPASS simulation, essentially replacing the COMPASS inbuilt RTC functionality as show in figure 8. Thus, COMPASS is responsible for simulating the atmosphere, calculating the forward model of the P-WFS and producing the simulated WFS image. A custom library that allows COMPASS to send and receive both MUDPI and RTMS data was used to send the WFS image to the HRTC prototype, in effect replacing the ESO `wfssim` in the performance experiment.

The HRTC WFC pipeline computes the modal mirror commands and sends them to COMPASS. After COMPASS has received these commands, it applies them to a virtual CCS which translates them to mirror commands for M4 and M5 [3]. Finally, COMPASS returns an echo to the HRTC describing the command applied to M4.

The integration test was successful and simulations that were run with the HRTC 'in the loop' completed with the expected performance regarding the wavefront reconstruction quality [4]. This test verifies that the HRTC WFC stage is numerically correct.

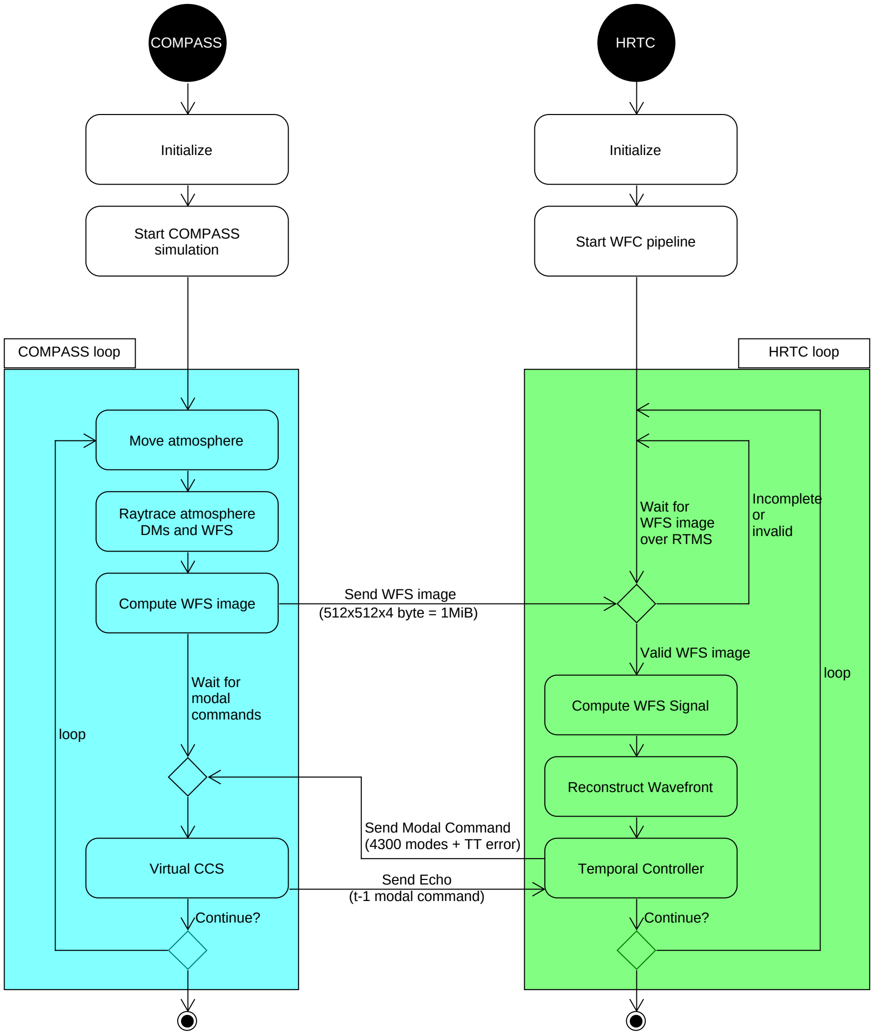

Figure 8: A flow diagram illustration the structure of the COMPASS HRTC integration test.

3. SRTC

The SRTC supervises and optimizes the operation of the HRTC which includes the estimation of system parameters. The most important estimated system parameters are listed below:

Estimated Parameter	**Description**
Mis-registration	Mis-registration between the WFS and M4 is identified by applying the System Parameter Recurrent INvasive Tracking (SPRINT) strategy [6].
Valid subaperture map	A *Subaperture Flux Monitor (SF Monitor)* analyzes the flux across the pupil, as seen by the WFS. Only subapertures that provide a sufficient Signal to Noise Ratio (SNR) are considered. Regions of the pupil that are obscured are excluded from the P-WFS signal and from the CM.
Lateral pupil position	The *Pupil Position Monitor (PP Monitor)* uses the WFS detector images to determine the lateral position of the pupil by applying a matched filter algorithm [2]. It will be used by the Pupil Position Control (PPC) loop, that is an auxiliary control loop realized outside of the RTC.

Position of guide star	Before closing the loop, a the guide star needs to be acquired at the pointing reference position. The *Guide Star Position Monitor (GSP Monitor)* determines the position of the guide star relative to the pointing reference position. It derives the position by analyzing the open-loop global tip/tilt signal obtained with the WFS.

The table below lists the main WFC loop parameters that the SRTC has to optimize regularly:

Optimized Parameter	**Description**
Command matrix	The CM combines the reconstruction of the wavefront from P-WFS signals and the projection into the modal command basis used by the CCS to control the wavefront. The *Command Matrix Optimizer (CM Optimizer)* updates the CM and the corresponding valid subaperture map. New CMs are composed and updated in the WFC loop as required. For observations with minimum zenith distance, the update frequency is ~0.5 Hz.
Differential High Order (DHO)	During high contrast imaging (HCI) observations, the *DHO control loop* updates the P-WFS reference signal in the WFC loop at an update frequency of ~1 Hz.
Modal gain	A classical optimal modal gain optimization is foreseen, which uses the WFC telemetry and minimizes the couple servolag/propagated noise.

The SRTC will be a distributed computer system. Its foreseen computer nodes including their estimated prices, the communication paths and the SRTC environment are depicted in figure 9. We plan to use the ESO IT standard computers in order to facilitate maintenance at the ELT.

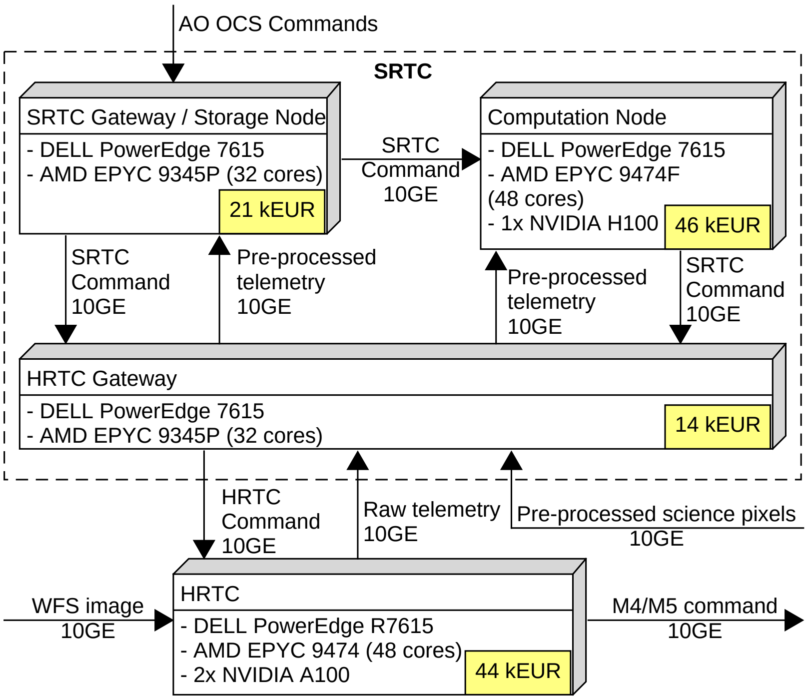

Figure 9: The RTC computer hardware structure and its communication paths inside of METIS AOCS.

The final SRTC will use the RTC toolkit provided by ESO. Initial testing and experimentation with the toolkit will take place over the coming months, and it is expected that the RTC toolkit will be used for the first prototype.

4. CONCLUSION

METIS SCAO including its RTC has passed its FDR in autumn 2022. We have demonstrated that the SCAO HRTC is capable of meeting the performance requirements of the WFC loop in a realistic setup including repeated updates of the command matrix. Using a modified version of COMPASS as a framework for the simulation of the telescope, we have demonstrated that we can close the AO loop using the HRTC prototype. Further development of the HRTC is expected to run in parallel with the SRTC. Once the few problems with the current prototype have been resolved, it can be used as a baseline to determine if any further improvements are warranted or not. The SRTC functionality and its hardware demands have been identified and first prototyping activities have started. Our initial focus will be on the few key components of the SRTC and integrating those with the RTC Toolkit provided by ESO. We are confident that we are prepared for the MAIT phase.

References

[1] T. Bertram et al. “METIS SCAO - implementing AO for ELT”. In: *Adaptive Optics for Extremely Large Telescopes*. Vol. 7. 2023.

[2] Hugo Coppejans et al. “Subpixel pupil tracking on METIS for ELT”. In: *Adaptive Optics for Extremely Large Telescopes*. Vol. 7. 2023.

[3] C. Correia et al. “Optimal modal basis for ELT M4 mirror Force/Position control”. In: *Adaptive Optics for Extremely Large Telescopes*. Vol. 7. 2023.

[4] M. Feldt et al. “Simulating METIS’ SCAO System”. In: *Adaptive Optics for Extremely Large Telescopes*. Vol. 7. 2023.

[5] D. Gratadour et al. “COMPASS: status update and long term development plan”. In: *Adaptive Optics Systems V*. Vol. 9909. SPIE. July 2016, 990971, p. 990971. DOI: `10.1117/12.2232681`.

[6] C. T. Heritier et al. “SPRINT, system parameters recurrent invasive tracking: a fast and least-cost online calibration strategy for adaptive optics”. In: *MNRAS* 504.3 (July 2021), pp. 4274–4290. DOI: `10.1093/mnras/stab1177`. arXiv: `2104.11261 [astro-ph.IM]`.

[7] Georg Ofenbeck et al. “Applying the roofline model”. In: *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. 2014, pp. 76–85. DOI: `10.1109/ISPASS.2014.6844463`.