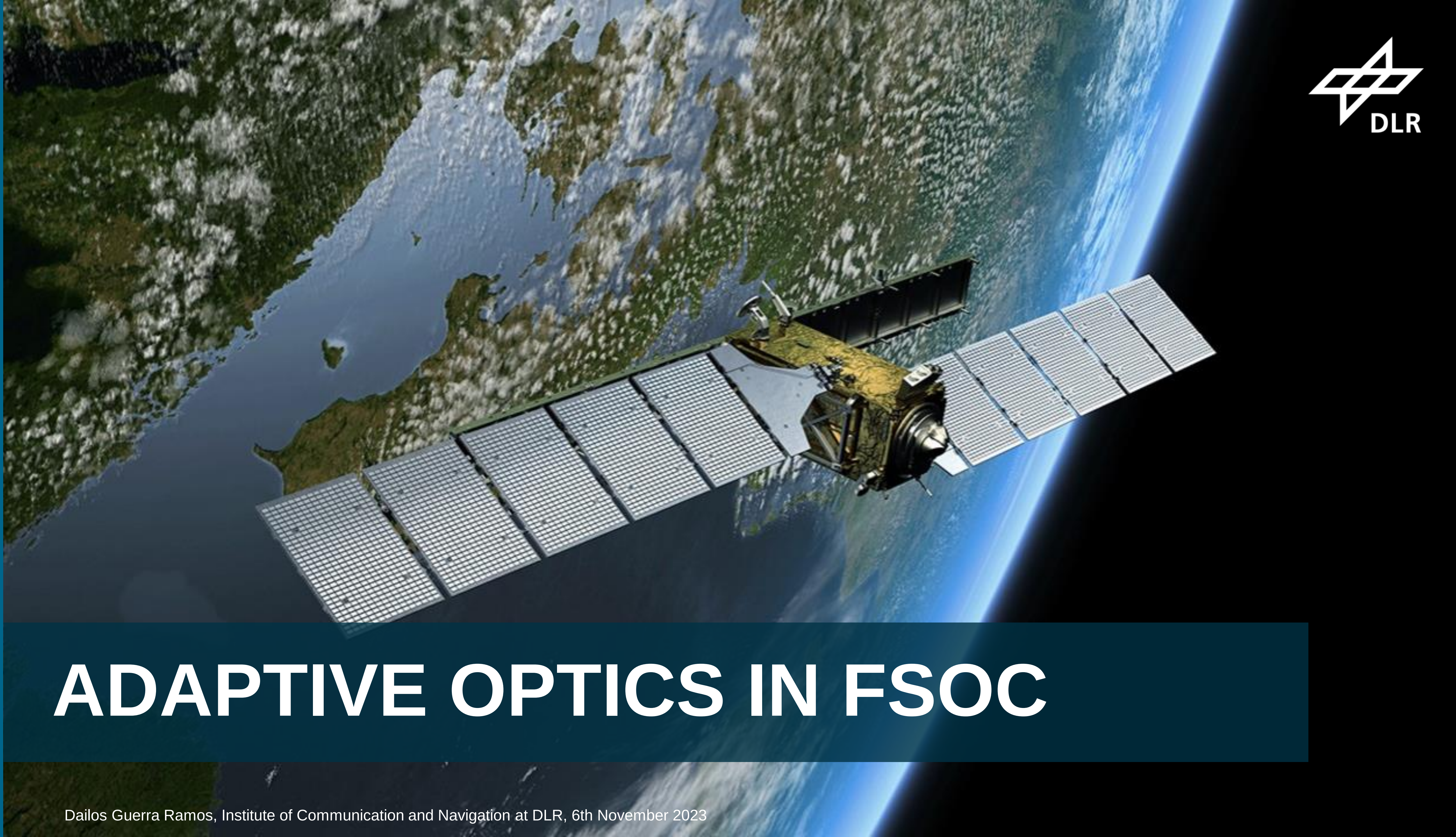
A CPU BASED REAL-TIME CONTROL SYSTEM FOR ADAPTIVE OPTICS IN FREE SPACE OPTICAL COMMUNICATION

A system design and performance evaluation.

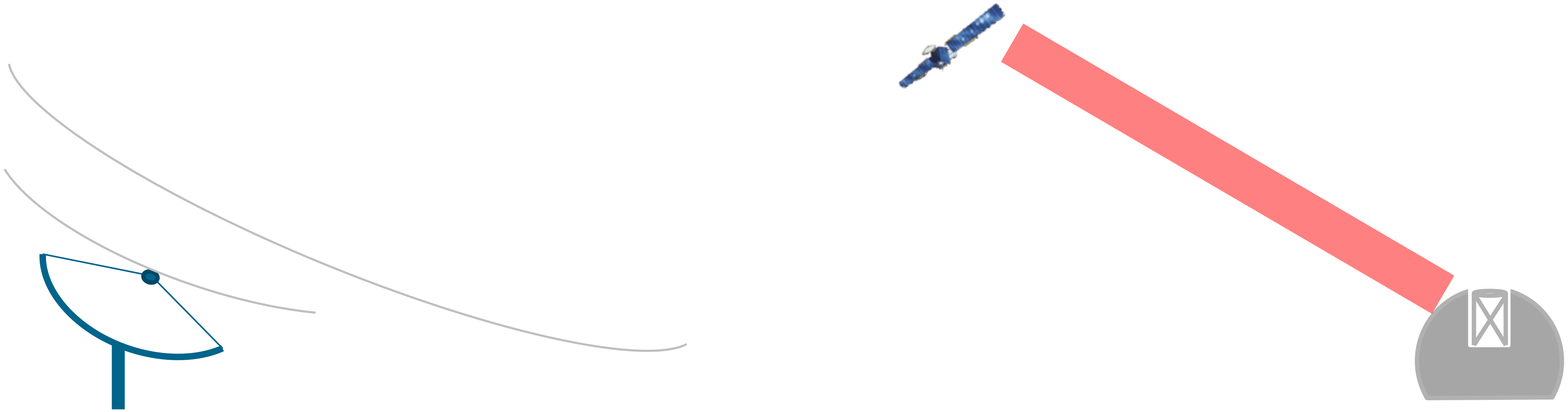
Dailos Guerra Ramos, Institute of Communication and Navigation at DLR, 6th November 2023



ADAPTIVE OPTICS IN FSOC

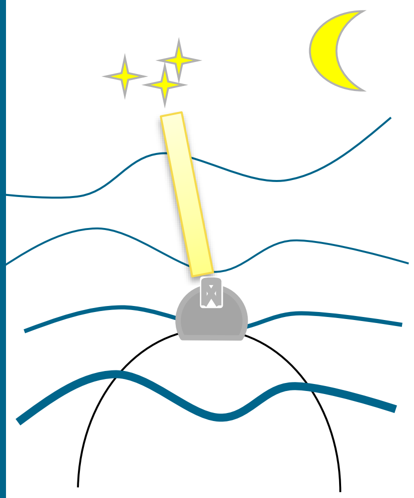
Dailos Guerra Ramos, Institute of Communication and Navigation at DLR, 6th November 2023

Radio vs Free Space Optical Communications

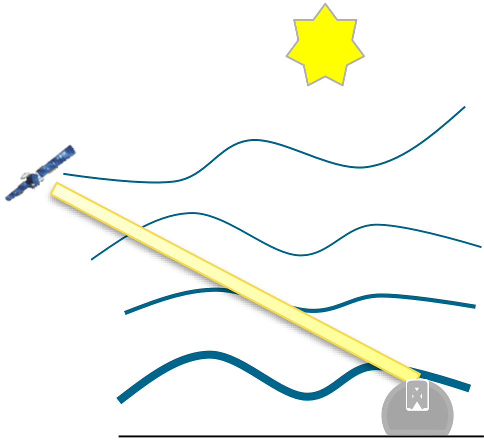


Radio		Optical
Limited	**Potential Bandwidth**	**Very High due to high carrier frequency**
Required	**Spectrum Licensing**	**Not Required**
Wide divergence (broadcast to large area) Classical Encryption schemes	**Security**	**Narrow Divergence (Line of sight)** **Quantum Encryption possible**
Mostly Unaffected	**Robustness to weather**	Cannot be used through cloud/rain
Unaffected	**Robustness to atmospheric turbulence**	Potential link instability

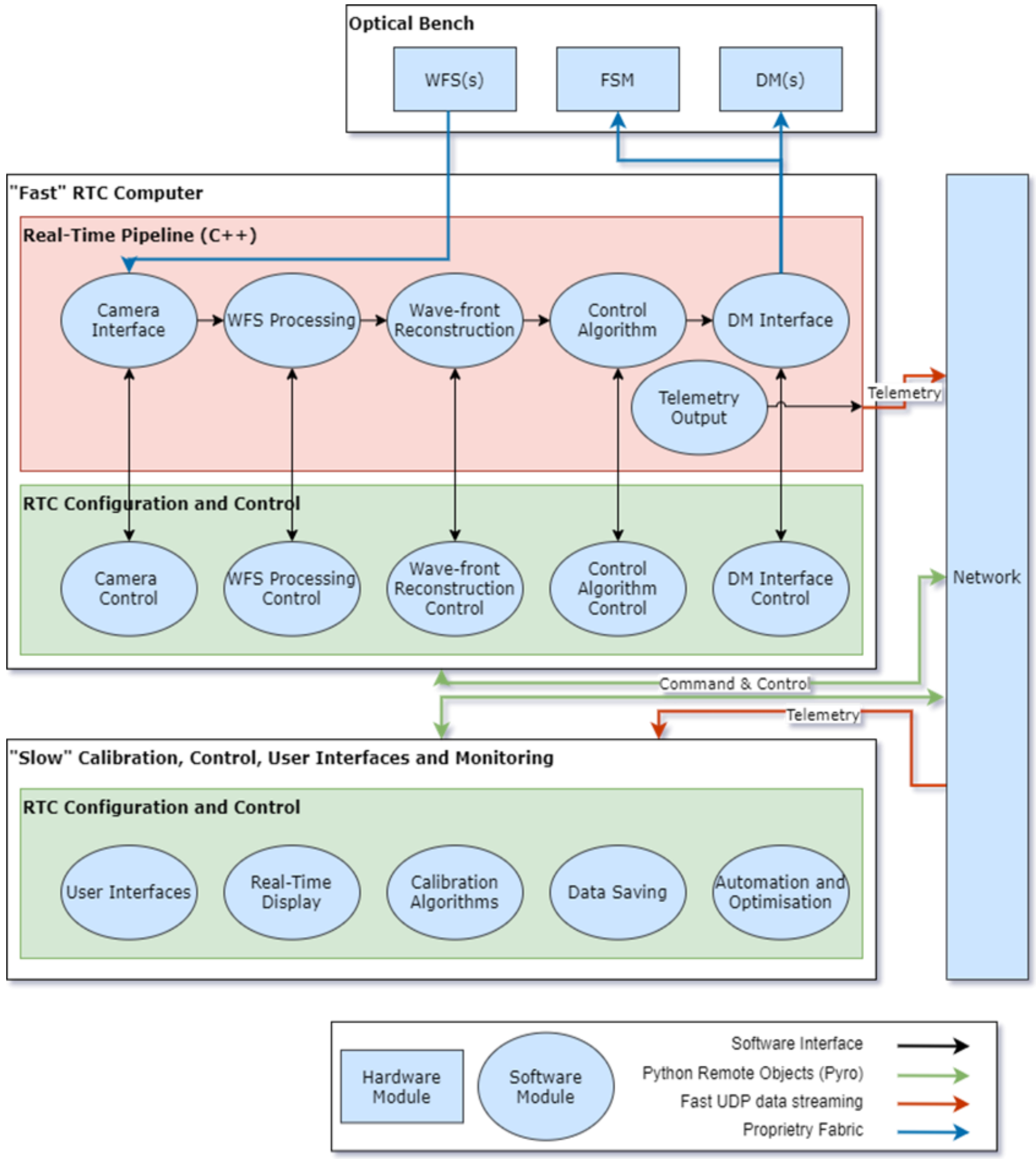
Adaptive Optics for Free Space Optical Communications



Astronomy		FSOC
High altitude above most Turbulence	**Location**	Ground level Potentially in towns & cities
“Weak” **r_0 = 15 – 30cm @ observation wavelength**	**Turbulence Strength**	“Strong” r_0 = 3 – 5 cm @ observation wavelength
Slow	**Turbulence coherence time**	Fast - Increased apparent wind speed due to LEO tracking
500Hz – 1kHz	**Required RTC Iteration Rate**	2kHz – 10kHz
Large (2 – 40m)	**Telescope Size**	**Small** **(0.3 – 1m)**
Large (50000 x 5000)	**AO Problem Size**	**~Small** **1000 x 1000**
Faint	**WFS reference irradiance**	**Bright**
Potentially Large	**Field of View**	**Typically very small**
Visible / Near infrared	**WFS/Science Wavelength**	Near infrared



AO RT Control System



- Operation split over multiple computers
- “Real-Time Pipeline” operates on dedicated RTC computer

- “Slow” Calibration and User interface operation on separate workstation
- Communication between nodes via the network

- Each stage of the Real-Time pipeline represented as stand-alone class in C++
- Python class wrapping each C++ class

- Setup and configuration all performed in Python

PERFORMANCE EVALUATION

EAGLE-1 SATELLITE

- ESA, the European Commission and space companies in Europe.
- Space-based quantum key distribution system @1550nm.
- Low Earth Orbit.
- OGS-OP will be in "In-Orbit Testing" optical ground station.
- Links must be reliable - AO is no longer an experiment but a critical part of the system design

EAGLE-1 AO Parameters

Telescope Diameter	80cm
Minimum elevation angle	20°
Target Fried Parameter	5cm @ 1550nm
DM Actuators	32 x 32
SHWFS sub-apertures	29 x 29, SCAO
Sub-apertures pixels	8 x 8
Iteration rate	~5 kHz

Currently at DLR:

- High-performance COTS CPU-based RTC computer: Dual CPU, AMD EPYC™ 74F3
- Non-strictly real time system (linux)
- Used with AO systems with 100-200 actuators.
- "SCAO" System.

Questions:

- Can our RTC solution handle EAGLE-1?
- How many cores are optimal?
- How stable is the system?

EAGLE-1 AO Hardware

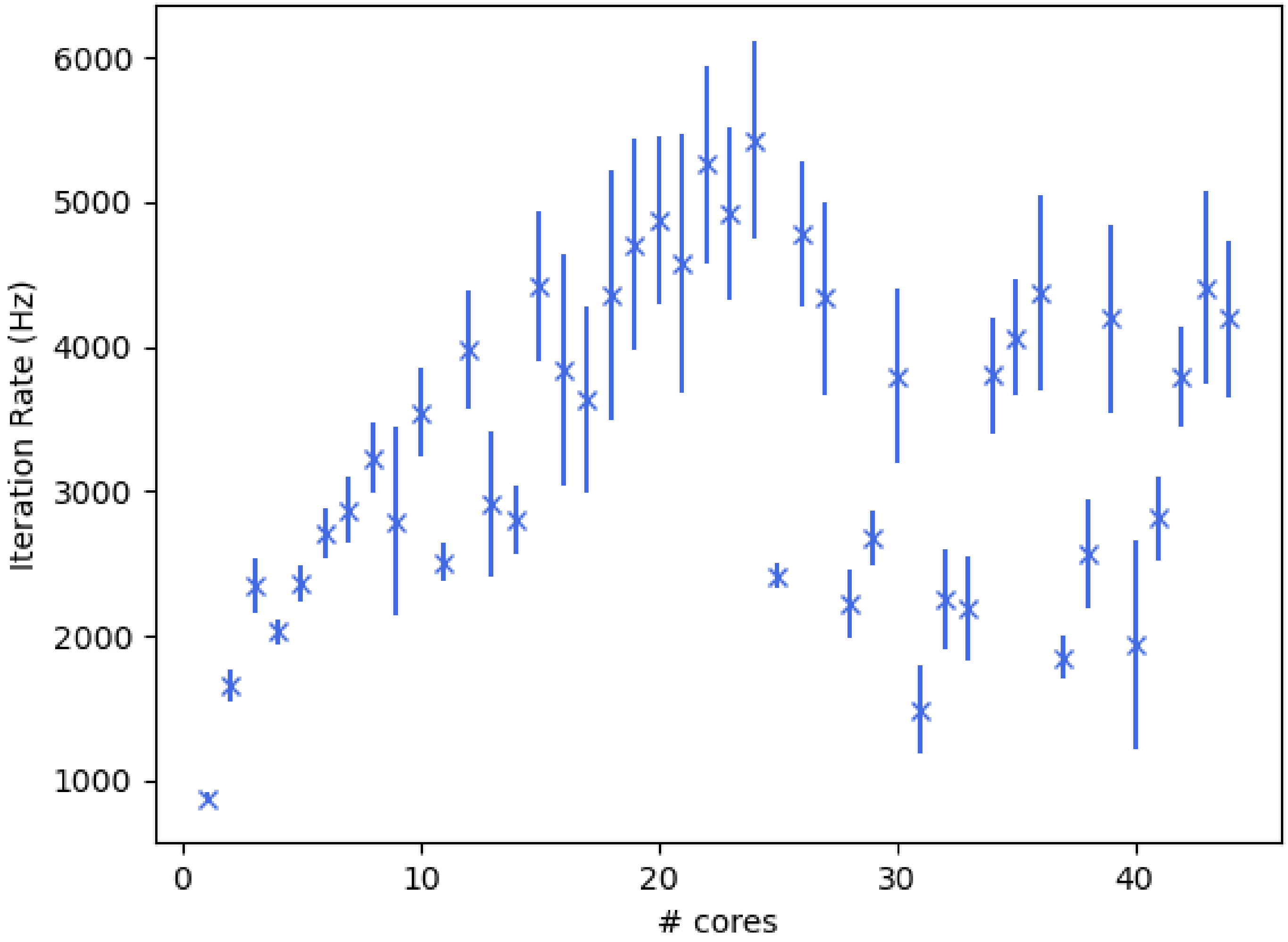
Wavefront sensor:

- Xenics Cheetah 1700.
- Up to 1700Hz full frame (640 x 512)
- ~6kHz @ 256 x 256
- Dual Channel, "CameraLink Medium" (4(!) Cables)
- Image split between channels and must be reconstructed

Deformable mirror:

- ALPAO DM820
- High Speed Ethernet Interface

Iteration rate (mean and std) vs number of cores

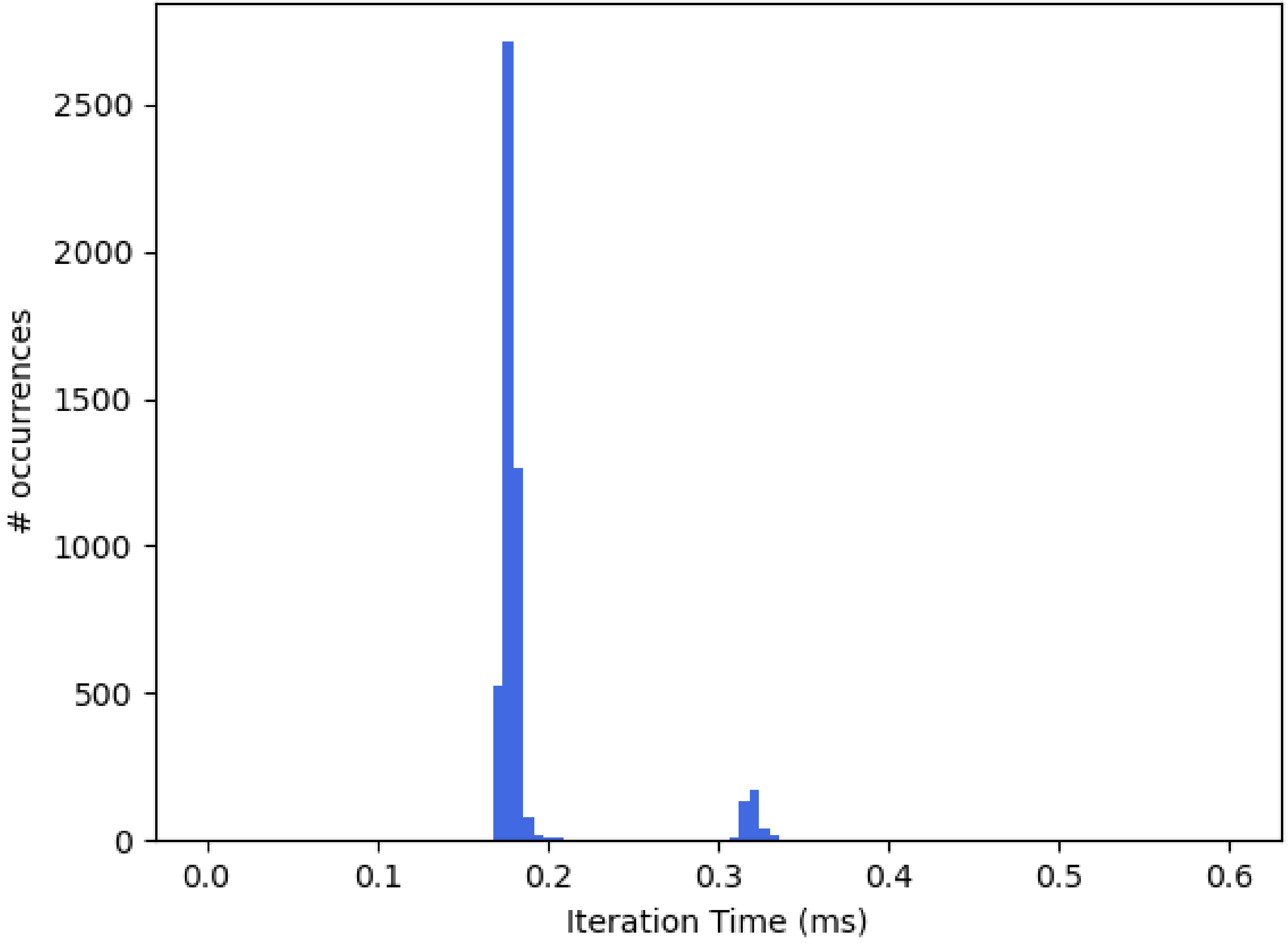


Benchmark:

- Using fake camera that reads from memory.
- Runs of 5000 iterations.
- Timing statistics: mean and standard dev.

- After 24 cores there is not improvement

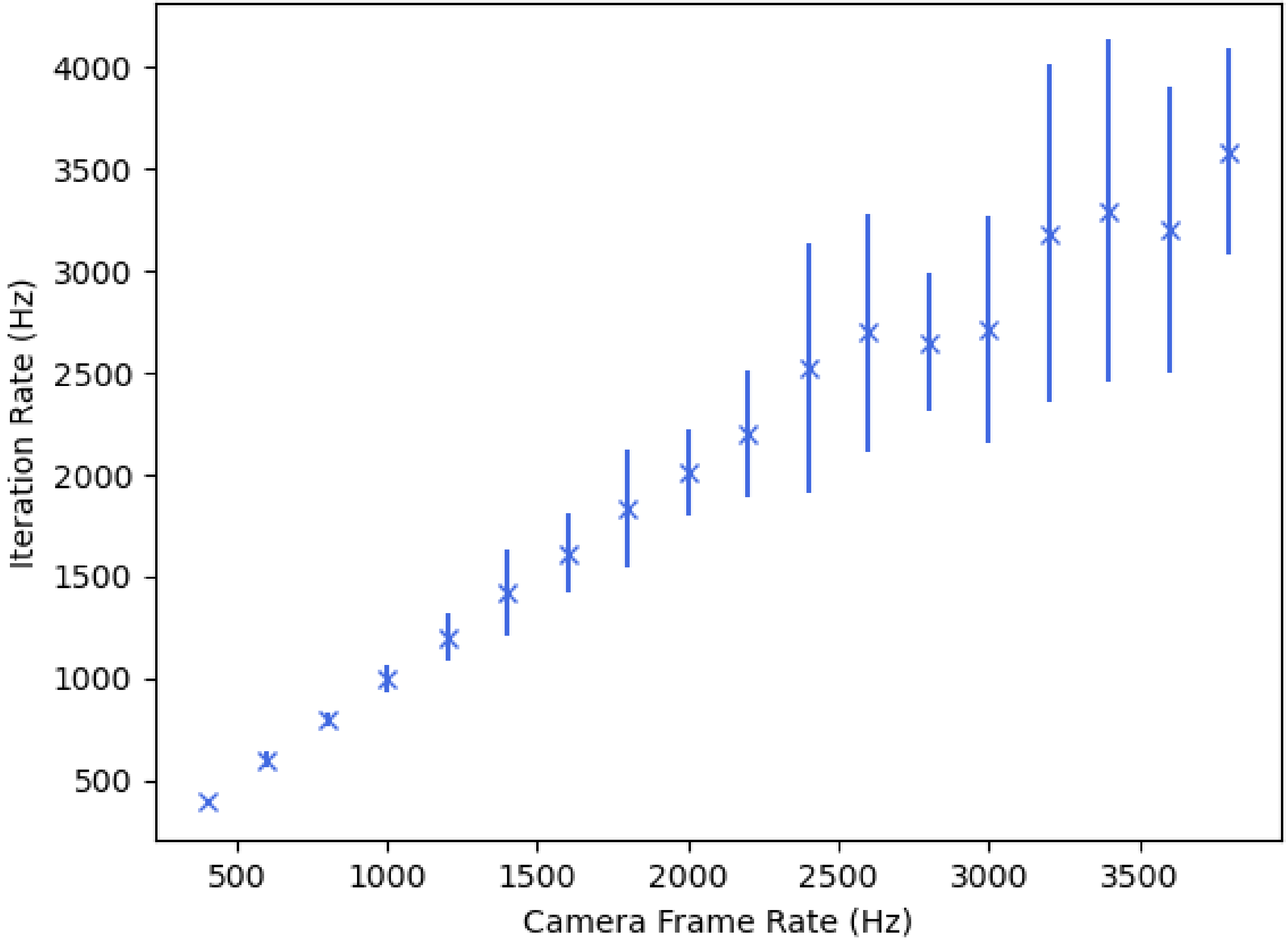
Iteration time histogram with 24 cores



Remarks:

- Most of the ocurrences below 0.2ms.
- Some outliers.

Iteration rate vs camera frame rate



Remarks:

- AO software with Cheetah camera and ALPAO DM820
- At some point the system cannot keep up with camera frame rate.

Conclusions

AO Software fulfills EAGLE-1 requirements

- Reaches the reconstruction rate without the hardware attached
- Further optimization to be done with camera interface

The Eagle-1 satellite is due to launch in 2024 with AO commissioning in 2025