Matrix-free vs. matrix-based real-time control for a MORFEO-like setting

Bernadett Stadler, Daniel Jodlbauer, Andreas Obereder, Stefan Raffetseder, Ronny Ramlau

AO4ELT7 - June 2023 - Avignon

Matrix-free vs. matrix-based real-time control for a MORFEO-like setting

Bernadett Stadler[a], Daniel Jodlbauer[b], Andreas Obereder[b], Stefan Raffetseder[a], and Ronny Ramlau[a,b]

[a]RICAM, Altenberger Strasse 69, 4040 Linz, Austria
[b]Johannes Kepler University, Altenberger Strasse 69, 4040 Linz, Austria

ABSTRACT

The Multiconjugate adaptive Optics Relay For ELT Observations (MORFEO) is one of the key Adaptive Optics (AO) systems on the ELT, aiming for a good wavefront correction over a large field of view. To compensate for the rapidly changing atmospheric distortions, a control algorithm computes in real-time the optimal shapes of the deformable mirrors from the wavefront sensor data. The control problem can be formulated in a matrix-based or a matrix-free way. In this paper we give an overview on existing AO real-time control algorithms and study the differences between matrix-based and matrix-free implementations via numerical simulations in COMPASS. For the numerical experiments we focus on the Finite Element Wavelet Hybrid Algorithm (FEWHA). Originally, this algorithm is formulated in a matrix-free and iterative way, however, FEWHA can be represented as a matrix-vector multiplication as well. We address whether the iterative approach affects the reconstruction quality in comparison to a direct one. Moreover, we study the computational performance of the matrix-free algorithm on a CPU vs. the matrix-based version on a GPU.

Keywords: atmospheric tomography, real-time computing, matrix-free, iterative solver

1. INTRODUCTION

The Extremely Large Telescope (ELT) features several instruments, one of them is the Multiconjugate adaptive Optics Relay For ELT Observations (MORFEO)[13]. MORFEO will not make observations by itself, rather it will enable other instruments, such as the Multi-AO Imaging Camera for Deep Observations (MICADO), to take images with an exceptional quality. Here we focus on the MORFEO Multi Conjugate Adaptive Optics (MCAO) mode, where the data obtained from several wavefront sensors (WFSs) is utilized to estimate the 3D atmospheric wavefront disturbance. The usage of multiple guide stars and deformable mirrors (DMs) together with the 3D atmospheric tomography enables the instrument to correct for multiple directions and a wider field of view (FoV).

Further author information: (Send correspondence to B.S.)
B.S.: E-mail: bernadett.stadler@indmath.uni-linz.ac.at

Developing AO control systems for the ELT is an ambitious and critical task, since a very large amount of data has to be processed in real-time. In order to achieve good results, the implementation of an efficient reconstructor on a high performance computing architecture is important. So far, the standard approach is the Matrix Vector Multiplication (MVM) method, in which a (regularized) generalized inverse of the discretized system operator is precomputed in soft real-time and a matrix-vector multiplication with the vector of sensor measurements is applied in hard real-time. In recent years, several iterative and matrix-free solvers have been proposed, which are fast and benefit from on the fly system updates. There exist various approaches that are dealing with the AO control problem in either a matrix-based or matrix-free way [14, 6, 19, 20, 47, 18, 17, 32, 42, 40, 29, 35, 30, 36, 28]. For the numerical experiments in this paper we focus on the Finite Element Wavelet Hybrid Algorithm (FEWHA)[48, 49, 50]. One of the key features of FEWHA is the dual domain discretization approach, which leads to a sparse representation of the involved matrices and allows an efficient matrix-free representation. However, the algorithm can be represented in a matrix-based way as well. Via numerical simulations we address whether the iterative implementation has an effect on the reconstruction quality in comparison to matrix-based direct solver. Moreover, we study the computational performance of the two control strategies. Real-time implementations for ELT AO systems have been previously studied [1, 25, 13, 5, 46, 11]. Suitable architectures have been evaluated within the Greenflash project[21, 22]. Based on these investigations and together with our previously performed analysis [37, 39] we focus here on a parallel implementation of FEWHA on a multi-core CPU in the matrix-free case and on an NVIDIA Tesla V100 GPU for the matrix-based implementation. The simulations are carried out using the AO software package COMPASS[12].

The paper is organized as follows: We start with a general description of AO control for MCAO systems in Section 2. In Section 3 we give an overview on existing matrix-based and matrix-free real-time control algorithms. Then we focus on FEWHA to compare the quality and computational performance of a matrix-based and matrix-free approach in numerical simulations in Section 4. Finally, in Section 5 we state our conclusion.

2. DEFORMABLE MIRROR CONTROL

Wavefronts that stem from a natural guide star (NGS) or laser guide star (LGS) are measured by one or more WFSs. From the WFS data a control algorithm reconstructs in real-time the optimal shapes of DMs. In order to compensate the rapidly changing atmospheric distortions, its shape has to be updated in less than a millisecond. The structure of this real-time reconstruction depends on the operating mode of the AO system. Here we focus on a MORFEO-like setting. Hence, we assume to have an MCAO system, which aims to achieve a uniformly good correction over the whole field of view (FoV). Moreover, we assume that the system is running in closed loop, i.e., residual wavefronts are measured by the WFSs.

In the following sections we briefly describe the main parts involved for the DM control of MCAO systems: atmospheric tomography, mirror fitting, pseudo open loop computation, and temporal control.

2.1 Atmospheric tomography

In atmospheric tomography we consider a layered model of the atmosphere and assume that all turbulence is located at a finite number of L infinitely thin layers $\phi = (\phi_1, \dots, \phi_L)$, see Figure 1. We aim at reconstructing the turbulent layers, i.e., the refractive index of the turbulent atmosphere, using measurements from WFSs [34]. The atmospheric tomography operator A relates WFS measurements and layers by

$$s = (s_g^x, s_g^y)_{g=1}^G = A\phi, \tag{1}$$

where G is the number of guide stars, and s are the WFS measurements (slopes) in x/y direction.

We assume the usage of a Shack-Hartmann (SH) WFS, which consists of a quadratic array of small lenslets, called subapertures, and a CCD photon detector behind this array. The tomography operator A is then decomposed into a geometric propagation operator P into the direction of the guide star, and an SH operator Γ. For a specific guide star g we obtain

$$s_g = \Gamma_g P_g \phi \quad \text{for } g = 1, \dots, G. \tag{2}$$

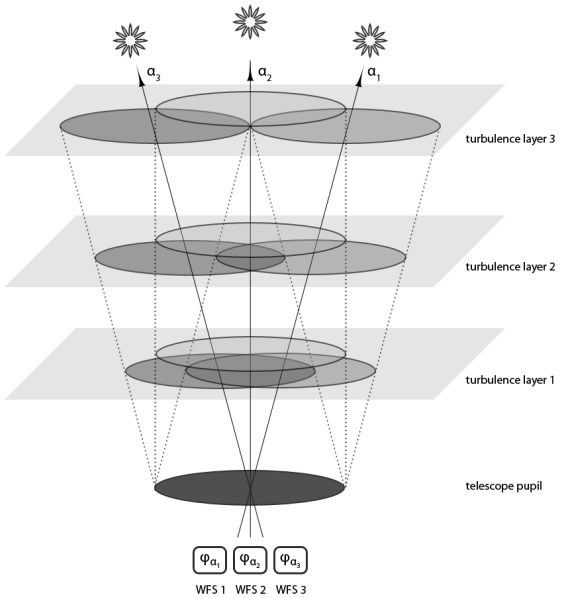


Figure 1: Illustration of turbulent layer reconstruction using 3 WFSs [48].

Within one subaperture Ω_{ij}, $i,j = 1,\dots,N$ the SH measurements are modelled as the average slopes of the wavefront aberration φ. We assume that the incoming wavefront aberration φ is approximated by a continuous piecewise bilinear function φ_{ij}. Then we obtain the SH measurements in a subaperture Ω_{ij} via

$$s_{ij}^x = \frac{(\varphi_{i,j+1} - \varphi_{i,j}) + (\varphi_{i+1,j+1} - \varphi_{i+1,j})}{2}, \tag{3}$$

$$s_{ij}^y = \frac{(\varphi_{i+1,j} - \varphi_{i,j}) + (\varphi_{i+1,j+1} - \varphi_{i,j+1})}{2}. \tag{4}$$

The vectors s^x and s^y are defined as a concatenation of values s_{ij}^x and s_{ij}^y for the set of indices (i,j) that belong to an active subaperture Ω_{ij}. The SH operator Γ maps wavefronts φ to SH-WFS measurements s,

$$s = \begin{pmatrix} s^x \\ s^y \end{pmatrix} = \begin{pmatrix} \Gamma^x \varphi \\ \Gamma^y \varphi \end{pmatrix} = \Gamma\varphi. \tag{5}$$

The wavefront aberrations in direction θ of a natural guide star (NGS) are given by

$$\varphi_\theta(x) = (P_\theta^{NGS}\phi)(x) := \sum_{\ell=1}^{L} \phi_l(x + \theta h_\ell), \tag{6}$$

where ϕ_ℓ is the turbulent layer at altitude h_ℓ for $\ell = 1,\dots,L$. We call P_θ^{NGS} the geometric propagation operator in the direction of the NGS. The point source created by a laser guide star (LGS) is considered to be at a finite height H. Note that we assume sodium LGSs. The incoming wavefront aberrations in the direction θ of an LGS are given by

$$\varphi_\theta(x) = (P_\theta^{LGS}\phi)(x) := \sum_{\ell=1}^{L} \phi_l\left(\left(1 - \frac{h_\ell}{H}\right)x + \theta h_\ell\right), \tag{7}$$

where P_θ^{LGS} is the geometric propagation operator in the direction of the LGS.

The atmospheric tomography problem is a limited angle problem. Mathematically, Equation (1) is ill-posed, i.e., the relation between the solution and the measurements is unstable[4, 23, 24]. Hence, regularization is

required. A common approach is the Bayesian framework, as it allows to incorporate statistical information about turbulence and noise. In this statistical approach we assume Φ to be a random variable corresponding to the turbulence layers. Moreover, we assume additive measurement noise, modeled by the random variable η. The random variables Φ and η are modeled by Gaussian variables with zero mean and covariance matrices C_Φ and C_η, see [33]. The layers are statistically independent, hence, the covariance matrix C_Φ has a block diagonal structure. We assume the noise to be independently distributed in every WFS, which implies a block-diagonal structure of the noise covariance matrix C_η. For this setting the maximum a posteriori (MAP) estimate provides an optimal point estimate for the solution [14], which is given by the solution of the linear system of equations

$$(A^T C_\eta^{-1} A + C_\phi^{-1})\phi = A^T C_\eta^{-1} s, \tag{8}$$

where A^T denotes the transposed tomography operator. We define the tomographic reconstruction operator, which maps WFS measurements onto reconstructed layers, as

$$R := (A^T C_\eta^{-1} A + C_\phi^{-1})^{-1} A^T C_\eta^{-1}. \tag{9}$$

2.2 Mirror fitting

Commonly, piecewise bilinear influence functions are used to model DMs. Those influence functions represent the shape of the DM when a single actuator is pushed. This shape is evaluated at the actuator positions. Note that in the end the DM is controlled via voltages, which are computed from the actuator commands. We refer to mirror fitting for the step in which the DM shape is computed.

From the reconstruction of the turbulent layers $\phi = (\phi_1, \dots, \phi_L)$, described in Section 2.1, the optimal commands $a = (a_1, \dots, a_M)$ of the M DMs have to be computed. In the case of an MCAO system, $M > 1$ DMs are employed for the correction conjugated to different heights $h_1, \dots, h_M$ in the atmosphere. The shapes of the M mirrors are fitted to the reconstructed layers in order to optimize the quality in a large FoV. The standard approach for mirror fitting, see e.g. [10, 28], is to minimize the functional

$$\int_{\mathrm{FoV}} \int_{\Omega_\mathcal{M}} \|(P\phi)(x) - (\bar{P}a)(x)\|^2 dx d\phi, \tag{10}$$

where $\Omega_\mathcal{M}$ denotes the telescope pupil and FoV is a set of directions in the FoV. The operators P and $\bar{P}$ denote the projection through layers and DMs, respectively. For more details on the fitting step we refer to [27, 29].

If there are $L = M$ reconstructed layers exactly located at the altitudes of the DMs (see Figure 2a) the fitting step becomes obvious, i.e., the actuator commands are simply the reconstructed layers $a_j = \phi_j$ for $j = 1, \dots, M$. For more and differently located layers (see Figure 2b) the actuator commands are given by the solution of

$$\bar{P}^T P\phi = \bar{P}^T \bar{P} a, \tag{11}$$

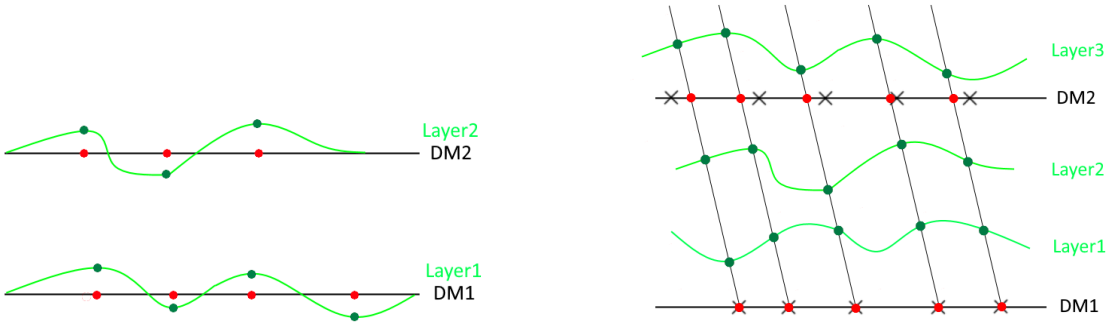
where $P = ((P_{\bar{\theta}_n,\ell})_{n=1}^N)_{\ell=1}^L$ denotes the projection through layer ℓ in direction $\bar{\theta}_n$ and $\bar{P} = (\bar{P}_{\bar{\theta}_n,m})_{n=1}^N)_{m=1}^M$ is projection through DM m in direction $\bar{\theta}_n$. The mirror fitting operator F, which maps reconstructed layers to the commands minimizing (10), can be computed as

$$F = (\bar{P}^T \bar{P})^{-1} \bar{P}^T P. \tag{12}$$

2.3 Pseudo-open loop computation

To enforce stability and robustness of the closed loop DM control, pseudo-open loop computation (POLC) is commonly employed in AO, see e.g. [8]. This approach stabilizes the reconstruction over time. From the original closed loop WFS data so called pseudo-open loop measurements are calculated in a first step by subtracting the down projected mirror updates of the previous time step

$$s = s^{cl} - \Gamma a, \tag{13}$$



(a) **(b)**

Figure 2: Illustration of the mirror fitting step for the reconstructed layers directly located at the altitudes of the DMs (a) and for more and differently located reconstructed layers (b).

where s denote the POL WFS data and s^{cl} the closed loop WFS measurements, i.e., the residual wavefronts. The operator Γ is the SH WFS operator and a is the update vector of actuator commands that are used to deform the mirror.

2.4 Integrator control

There is a certain delay between the time when the measurements are obtained from the WFSs and the time when the wavefronts get corrected by the DMs. Because the atmosphere changes rapidly, the AO system updates the DM shapes based on the current measurements, collected by the WFSs, and the previous DM shapes [26]. We consider a two step delay, see e.g. [48], and denote the previous, the current and the next time step of the loop by the superscript indices $(i-1), (i)$ and $(i+1)$. The new actuator commands $a^{(i+1)}$ are determined from the reconstruction that uses the measurements $s^{(i-1,i)}$ and from the previous actuator commands $a^{(i)}$. Note that because of the delay, the measurements $s^{(i,i+1)}$ are not available at time step $(i+1)$. To indicate that we are in principle applying the wrong correction we use a so called output or loop gain, which combines the actuator commands from the previous time step $a^{(i)}$ and the computed ones a to get the new actuator command vector

$$a^{(i+1)} = (1 - \text{gain}) \cdot a^{(i)} + \text{gain} \cdot a. \tag{14}$$

3. MATRIX-BASED VS. MATRIX-FREE DM CONTROL

Historically, direct matrix-based methods have been the only approaches considered for a long time to control DMs in real-time. Recently, research has been shifted more into the direction of iterative methods implemented in a matrix-free way, mainly motivated by the computational load of ELTs.

3.1 Direct and matrix-based algorithms

The standard procedure to control the DMs, i.e. solve Equation (8) and Equation (11) is called Matrix Vector Multiplication (MVM) method, see e.g. [14]. The actuator commands are computed from the WFS measurements via a matrix-vector multiplication with the reconstruction matrix R defined in Equation (9) and the fitting operator F defined in Equation (12)

$$a = (FR)s. \tag{15}$$

Note that s here refers to the POL measurements as defined in Equation (13). The DM updates are then calculated via Equation (14). There exist different discretization strategies for Equation (15). A typical choice is the basis of Zernike polynomials, see e.g [15]. Moreover, certain factorization techniques, such as Cholesky decomposition, are used to compute and store the inverse in a more efficient way. Altogether, there are different variations of the MVM available.

The calculation of (FR) is often referred to as soft real-time, since the recomputation has to be done only when certain parameters at the telescope or in the atmosphere vary. In particular, the noise level, which changes the

entries of C_η, and the turbulence parameters, that affect C_ϕ, require a recomputation of R. Telescope rotations or misalignment influences the matrix F. The multiplication with the vector of sensor measurements s is done in hard real-time at approximately $500 - 1000$ Hz. The matrix R is dense, thus, the hard real-time computation of the MVM scales at $\mathcal{O}(n^2)$. The dimension n is related to the size of the telescope, in particular, it depends on the number of subapertures of the WFSs and the number of actuators of the DMs. However, the method is well parallelizable and piplineable, and therefore still efficient when high-performance hardware is employed.

3.2 Iterative and matrix-free algorithms

Iterative methods became of interest because they considerably reduce the computational load. Such kind of methods can outperform the MVM method if the number of iterations is small and the left hand side operator has an efficient representation. They rely on a discretization scheme that leads to sparse matrices. These sparse matrices can be efficiently implemented in a matrix-free way. This does not only reduce the computational load and memory consumption, but allows to easily update parameters of the AO system on the fly. Since the left hand side operators of Equation (8) and Equation (11) are symmetric and positive definite, the conjugate gradient (CG) method can be applied. The number of iterations is a crucial indicator for the real-time performance. Preconditioning is an easy and efficient way to reduce the number of iterations. Because of the more complex structure of the iterative algorithms, parallelization is more difficult and pipelining is not possible.

Different preconditioned CG (PCG) methods have been proposed in the literature for the DM control. Early methods, e.g. in [20, 7, 45], employ multigrid preconditioners (MG-PCG) with which it is possible to reduce the computational complexity to $\mathcal{O}(n^{3/2})$. Later on, the research in [44, 47, 18, 17] has been focused on Fourier domain preconditioning (FD-PCG), which leads to methods that scale with $\mathcal{O}(n \log(n))$. The Fourier basis is very efficient in a sparse representation of the underlying operators. Algorithms based on the Fourier transform have been proposed in [43, 16]. The forward as well as the inverse covariance of the noise are sparse. However, the inverse covariance of the layers is usually a dense matrix. To overcome this problem an approximation using a modified turbulence power law in combination with biharmonics has been proposed in [9].

To the best of our knowledge there exist two iterative methods that scale at $\mathcal{O}(n)$. The first one has been proposed in [42, 41, 2, 40] and is called the Fractal Iterative Method (FrIM). In this approach the covariance matrix of turbulence layers is approximated using matrices with hierarchical structure, i.e., $C_\phi \approx KK^T$, where $K = K_1 \cdot \cdots \cdot K_p$ with p denoting the number of scales. The sparse matrices K_i with $1 \leq i \leq p$ are computed by a modified mid-point algorithm. The MAP estimate is formulated as

$$(K^T A^T C_\eta^{-1} A + I)x = K^T A^T C_\eta^{-1} s. \tag{16}$$

The solution is given by the variable transformation $\phi = Kx$. Note that s again denotes the POL measurements, which are computed in a first step. Equation (16) is solved using the PCG method with a diagonal preconditioner. FrIM is referred to as two step approach, i.e., after having solved the MAP estimate the turbulence layers are fitted to the DM shapes. To this end Equation (11) is again solved with the PCG method. The DM command update vector is computed via an integrator control, see Equation (14).

The second iterative method that scales at $\mathcal{O}(n)$ is based on a dual domain discretization into a wavelet and finite element domain called Finite Element Wavelet Hybrid Algorithm (FEWHA) [49, 50, 48, 39, 31, 38]. Compactly supported orthonormal wavelets are used for representing the turbulent layers. In the frequency domain, the wavelet basis allows a completely diagonal representation of C_ϕ. To achieve a sparse representation for the atmospheric tomography operator A, discretization in space is applied using a piecewise bilinear basis of finite elements. The MAP estimate of the tomography problem is then formulated as

$$(W^{-T} \hat{A}^T C_\eta^{-1} \hat{A} W^{-1} + \alpha D)c = W^{-T} \hat{A} T C_\eta^{-1} s, \tag{17}$$

where s are the POL WFS measurements, c denotes the vector of wavelet coefficients, $\hat{A}$ is the tomography operator in the finite element basis and W is the discrete wavelet transform. For tuning purposes a regularization parameter α is introduced, which is optimized via simulations. The solution is given by the variable transform $\phi = W^{-1}c$. Equation (17) is solved using the PCG method. For FEWHA a modified Jacobi preconditioner [49] together with an augmented Krylov subspace method [31] is employed to reduce the number of iterations. Just as

FrIM, FEWHA is a two step approach. After the solution of the MAP estimate is calculated the fitting problem is solved using again the PCG method with a standard Jacobi preconditioner. Special attention has been paid in the matrix-free, parallel implementation of the FEWHA on real-time hardware and its performance analysis, see [39, 37, 38]. The outcome of these studies is that the iterative matrix-free approach performs fast on CPUs, whereas for large matrix-vector multiplications GPUs are better suited.

Note that FrIM as well as FEWHA are also capable of being represented in a matrix-based way. However, in that case an MVM approach is better suited than the iterative PCG. In a matrix-based formulation the sparse structure of the matrices is lost. Thus, an iterative method becomes computationally very demanding, because in every iteration a matrix-vector multiplication with the system matrix has to be applied. For our numerical experiments we stick to FEWHA as control algorithm and evaluate its performance as either matrix-based direct solver or matrix-free iterative solver.

4. NUMERICAL SIMULATIONS

For our numerical simulations we use the software package COMPASS [12], which allows in particular to simulate all critical subcomponents of an AO system in the context of the ELT. The tool takes advantage of the GPU hardware architecture, and thus is able to provide an adequate execution speed for large simulations. FEWHA is implemented in C++, but exposes a Python interface using pybind11*. This allows us to use FEWHA as Python module directly within a COMPASS simulation. The communication and conversion of numpy arrays into C++ arrays is handled automatically by pybind11.

4.1 MORFEO-like system configuration

In our simulations a telescope that gathers light through a primary mirror of 37 m diameter with a central obstruction of approximately 11 % is considered. A 35 layer atmosphere that follows the von Karman statistics is utilized and the quality is evaluated using the Strehl ratio in the K band, i.e., at a wavelength of 2200 nm. The system parameters are summarized in Table 1. An MCAO system is simulated, which uses 3 DMs as defined in Table 2 for the correction. For all numerical simulations the Fried geometry with equidistant actuator spacing for the DMs is assumed [3]. Note that within FEWHA the NGS and the LGS problem are coupled, i.e., no additional tip-tilt mirror is used.

Parameter	Value
Telescope diameter	37 m
Central obstruction	11%
Na-layer height	90 km
Na-layer FWHM	11.4 km
Outer scale L_0	25 m
Field of View	1 arcmin
Simulated duration	1 s
Delay	2 frames
Evaluation criterion	LE Strehl
Evaluation wavelength	K band (2200 nm)

Table 1: System parameters.

Parameter	M4	DM1	DM2
Numb. of actuators	75×75	47×47	37×37
DM altitude	0 km	4 km	12.7 km
Actuator spacing	0.5 m	1 m	1 m

Table 2: DM configuration.

The six high order SH WFSs, which measure the light from the LGSs, are equipped with 74×74 subapertures each. The three low order WFSs, which are used to measure the NGSs aberrations and correct for the tip-tilt uncertainty, are equipped with 2×2 subapertures. The LGSs are positioned in a circle of 90 arcsec diameter and the NGSs in a circle with a diameter of 110 arcsec, see Figure 3. In COMPASS, we compute the measurements from the slope of the incoming wavefronts via a weighted center of gravity (WCoG) algorithm. Details about the parameters can be found in Table 3.

*`https://github.com/pybind/pybind11`

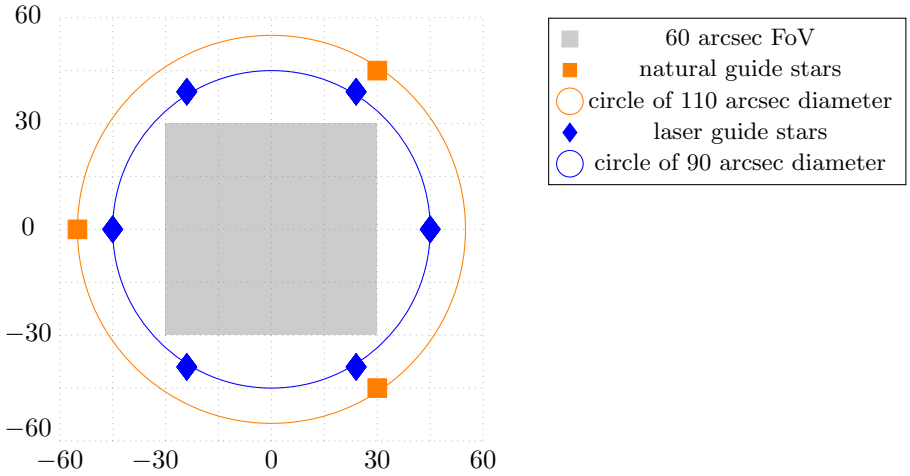


Figure 3: Star asterism of NGSs (orange) in a circle of 110 arcsec diameter and LGSs (blue) in a circle of 90 arcsec diameter. The 60 arcsec FoV is marked in gray.

Parameter	**LGS-WFS**	**NGS-WFS**
Type	SH WFS	SH WFS
Number	6	3
Geometry	74×74 subap.	2×2 subap.
Subaperture size	12×12 pixels	6×6 pixels
Optical throughput	0.23	0.33
FoV per subaperture	16.8 arcsec	1.3 arcsec
GS asterism	90 arcsec diameter	110 arcsec diameter
Wavelength	589 nm	1650 nm
Detector RON	3.0 e^-/pixel/frame	0.5 e^-/pixel/frame
Centroiding algorithm	WCoG	WCoG

Table 3: WFS configuration.

4.2 Reconstruction quality and computational performance

The MORFEO-like setting is simulated once using a direct, matrix-based FEWHA and once with an iterative, matrix-free version. In both cases 9 layers are reconstructed, see Section 2.1. The system is running in closed loop, i.e., the POL measurements are computed in a first step. For the matrix-free implementation the tomography and the mirror fitting problem are solved separately. In mirror fitting 5×5 optimization directions over the FoV are used and only 1 PCG iteration, as this already gives good results. The direct solver applies a Cholesky factorization and the inverse of the system matrix is precomputed. In every loop iteration a matrix-vector product with the vector of WFS measurements is performed. The last step of the algorithm is an integrator control as described in Section 2. The method specific parameters have been optimized via numerical simulations and are listed in Table 4.

Figure 4 shows the Strehl ratio vs. the field off-axis position in arc seconds of the direct, matrix-based implementation (black) and the iterative, matrix-free versions (colored) with 1 to 8 PCG iterations for the tomography problem. Note that in contrast to the direct approach the precomputation of the inverse of the system matrix is not required here. Using more than 8 iterations does not provide quality improvements for the MORFEO-like setting. When using 4 iterations already a very good quality is reached and it only slightly increases with 8 iterations. The Strehl ratio even starts to decrease for more than about 100 PCG iterations. A similar behavior is shown by the direct solver, which obtains a Strehl ratio less than that provided by the PCG

method when using between 2 and about 200 iterations. Although the Strehl ratio for the direct solver is lower, the regularized system in (8) is solved more accurately as the residual is smaller. This seems surprising at first glance, but can have several reasons. We are dealing with a regularized problem, i.e., the computed solution is not the exact solution of the original inverse tomography problem. Stopping the PCG method early acts as an additional regularization. Especially, when certain configurations such as the Cn2 profiles are not close enough to the current physical situation, we can avoid overfitting by stopping the PCG method earlier. Altogether, the number of iterations provides an additional tuning parameter.

Parameter	Value
Tomographic reconstruction	9 layer
Iterations [Tomo., Fitting]	$[1-8, 1]$
Regularization param. α	64
Spot elongation param. α_η	0.2
Loop gain	0.6
Optimization directions	5×5 over FoV

Table 4: Method parameters.

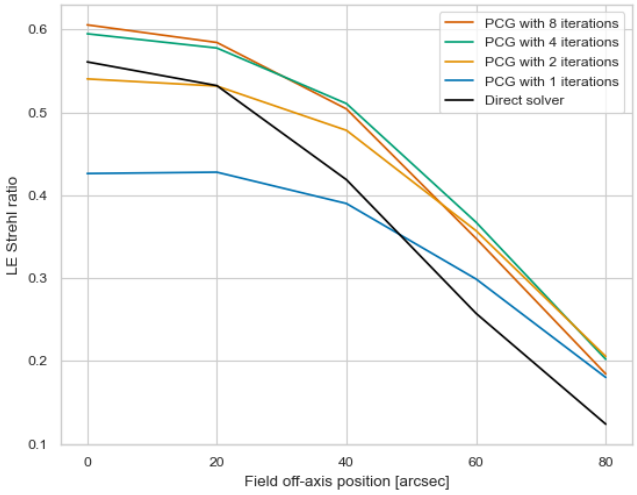


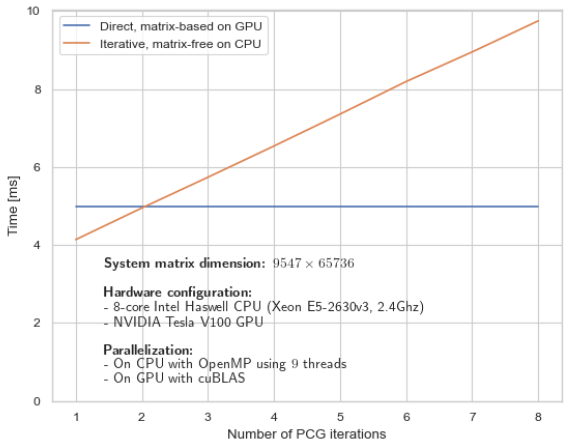
Figure 4: Long exposure (LE) Strehl ratio vs. field off-axis position for FEWHA with a direct solver (black) or a different number of CG iterations for the tomography step (colored).

A parallel version of FEWHA is implemented in a direct, matrix-based way on a GPU and an iterative, matrix-free way on a CPU. Note that only the hard-real-time part is considered here, i.e., not the time to precompute the inverse of the system matrix required for the matrix-based version. The computational performance of a matrix-free CPU and GPU implementation of FEWHA has already been studied in [39] with the outcome that the CPU is better suited in the matrix-free case. For parallelization on CPU OpenMP[†], which is a library supporting multithreading, with 9 threads is used. The number 9 is related to the number of WFSs and layers. For details on the parallelization of the matrix-free version we refer to [39]. On the GPU cuBLAS[‡], which is an implementation of BLAS[§](Basic Linear Algebra Subprograms) employing NVIDIA CUDA, is used. This library makes it easy to perform matrix operations while accessing the computational resources of NVIDIA GPUs.

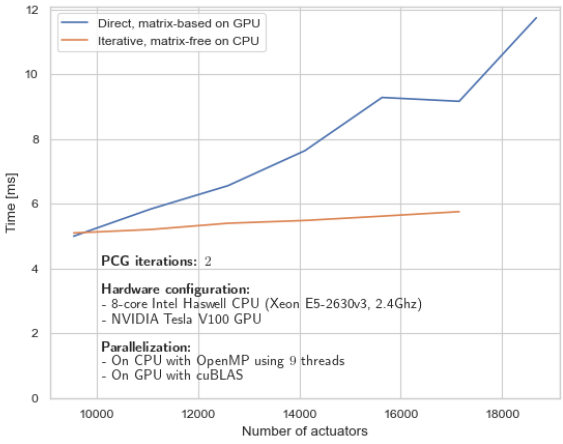
Figure 5a shows the run-time for a matrix-based implementation on GPU (blue) vs. a matrix-free implementation on CPU (orange). The algorithm is running on two 8-core Intel Haswell CPUs (Xeon E5-2630v3, 2.4GHz) and on an NVIDIA Tesla V100 GPU. The system matrix has a size of 9547×65736. For the iterative version the number of PCG iterations is varied. In Table 5 the run-time of different steps of the algorithm for the matrix-free and matrix-based case are listed. Here the number of PCG iterations is fixed to 2. In addition, the data transfer time for the GPU is shown. Note that for the matrix-free implementation the tomography and mirror fitting step are solved separately, while in the matrix-based implementation only one matrix-vector multiplication is performed. We observe that with 2 or less PCG iterations the parallel implementation on CPU outperforms the GPU version. From Table 5 we conclude that a matrix-based implementation is only feasible on a high-performance GPU and the CPU implementation is extremely slow.

[†]`https://www.openmp.org/`
[‡]`https://docs.nvidia.com/cuda/cublas/index.html`
[§]`https://www.netlib.org/blas/`

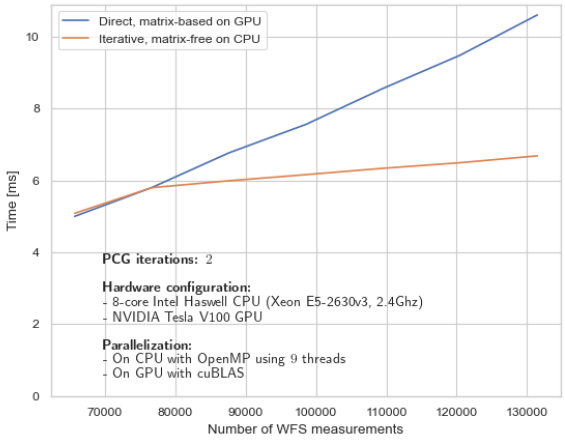
Figure 5b and Figure 5c show the direct GPU implementation (blue) vs. the iterative CPU version (orange) when scaling the number of actuators and WFS measurements, respectively. In this case 2 PCG iterations for the iterative FEWHA and 9 threads for parallelization on CPU are used. Note that the MORFEO-like setting is on the left end of the plots. We observe that for a higher number of actuators or WFS measurements the matrix-free implementation clearly shows its benefits. Furthermore, the parallelization is currently tuned to the MORFEO-like setting, and could still be improved for differently sized configurations. Nonetheless, the run-time for the matrix-free case only slightly increases for larger telescopes, whereas the run-time of the matrix-based implementation grows a lot.



(a) Run-time of FEWHA matrix-free on CPU vs. matrix-based on GPU with cuBLAS for varying number of CG iterations.



(b) Scaling with number of actuators.



(c) Scaling with number of WFS measurements.

Figure 5: Run-time of FEWHA with either a parallel matrix-based implementation on GPU (blue) or a matrix-free implementation on CPU (orange). The matrix-based version uses the cuBLAS library. The matrix-free version is parallelized with OpenMP.

5. CONCLUSION

In this paper we give an overview on different existing direct and iterative DM control algorithms. Direct solvers, such as the MVM method, have been used in the context of DM control since the beginning. They are convenient

	Run-time of parallel implementation of FEWHA		
	Matrix-free CPU	**Matrix-based CPU**	**Matrix-based GPU**
POL computation	0.215 ms	196.045 ms	0.013 ms
Tomography & Fitting	4.546 ms (1.975 + 2.571) ms	200.415 ms	4.988 ms
Data transfer	-	-	0.092 ms
Overall	4.761 ms	396.460 ms	5.010 ms

Table 5: Run-time of different steps for a matrix-based version on CPU and GPU and a matrix-free implementation using 2 PCG iterations to solve the tomography problem on a CPU.

to use, they are easy to implement and their application is well parallelizable and pipelineable. However, they are also very computationally demanding and memory consuming. Hence, fulfilling the real-time requirements for ELTs is only possible with high performance hardware. Mainly because of the computational load of the future ELTs, research recently focused on iterative approaches. Those methods do not require the demanding precomputation of the system matrix. Moreover, they benefit from on the fly system updates. For our numerical experiments we stick to FEWHA, which is a control algorithm based on a discretization using a wavelet and finite element basis. Originally, the method was proposed in an iterative and matrix-free way, however, the method is capable of being represented in a direct and matrix-based manner as well. Note that we expect to get a similar outcome with other iterative approaches such as FrIM. Our numerical simulations in COMPASS show that the iterative approach outperforms the direct one in terms of reconstruction quality when the PCG method is stopped earlier. In this way we omit overfitting to the regularized problem and have an additional tuning parameter at hand. We implement the algorithm in an iterative, matrix-free way on the CPU and in a direct, matrix-based way on the GPU. Note that we have already studied in [39] that for the matrix-free implementation the CPU is better suited than the GPU. For the MORFEO-like test setting the iterative, matrix-free FEWHA on the CPU is faster than the matrix-based GPU version when using 2 or less PCG iterations to solve the tomography problem. When increasing either the number of actuator or the number of WFS measurements the matrix-free implementation shows its benefits. We considered here a 9 layer reconstruction. When reconstructing only 3 layers directly at the altitudes of the DMs and thus omitting the mirror fitting step, the iterative approach becomes very efficient [38]. To conclude, for the MORFEO-like test setting we are on the edge of when an iterative, matrix-free implementation pays off. Considering an increase in either WFS measurements or actuators of future telescopes a iterative, matrix-free DM control algorithms become more viable.

ACKNOWLEDGMENTS

The project has received funding by the Austrian Science Fund (FWF) F6805-N36 (Tomography in Astronomy), the Austrian Research Promotion Agency (FFG) No. FO999888133 (Industrial methods for Adaptive Optics control systems) and the NVIDIA Corporation Academic Hardware Grant Program.

References

[1] Julien Bernard et al. “A GPU based RTC for E-ELT Adaptive optics : Real Time Controller prototype”. In: AO4ELT5, 2017.

[2] E. Brunner, Cl. Béchet, and M. Tallon. “Optimal projection of reconstructed layers onto deformable mirrors with fractal iterative method for AO tomography”. In: *Adaptive Optics Systems III*. Ed. by Brent L. Ellerbroek, Enrico Marchetti, and Jean-Pierre Véran. Vol. 8447. International Society for Optics and Photonics. SPIE, 2012, pp. 1802–1810. DOI: 10.1117/12.926809. URL: https://doi.org/10.1117/12.926809.

[3] M. Cayrel. "E-ELT optomechanics: overview". In: *Ground-based and Airborne Telescopes IV*. Ed. by Larry M. Stepp, Roberto Gilmozzi, and Helen J. Hall. Vol. 8444. International Society for Optics and Photonics. SPIE, 2012, pp. 674–691. DOI: 10.1117/12.925175. URL: https://doi.org/10.1117/12.925175.

[4] M.A. Davison. "The ill-conditioned nature of the limited angle tomography problem". In: *SIAM J. Appl. Math.* 43 (1983), pp. 428–448.

[5] N.A. Dipper et al. "ADAPTIVE OPTICS REAL-TIME CONTROL SYSTEMS FOR THE E-ELT". In: AO4ELT3, 2013.

[6] B. Ellerbroek, L. Gilles, and C.R. Vogel. "A Computationally Efficient Wavefront Reconstructor for Simulation or Multi-Conjugate Adaptive Optics on Giant Telescopes". In: *Proc. SPIE* 4839 (2002).

[7] B. Ellerbroek, L. Gilles, and C.R. Vogel. "Numerical simulations of multiconjugate adaptive optics wavefront reconstruction on giant telescopes". In: *Applied Optics* 42 (2003), pp. 4811–4818.

[8] B. Ellerbroek and C.R. Vogel. "Simulations of closed-loop wavefront reconstruction for multiconjugate adaptive optics on giant telescopes". In: *Proc. SPIE* 5169 (2003), pp. 206–217.

[9] B.L. Ellerbroek. "Efficient computation of minimum-variance wave-front reconstructors with sparse matrix techniques". In: *J. Opt. Soc. Am.* 19.9 (2002), pp. 1803–1816.

[10] B.L. Ellerbroek and C.R. Vogel. "Inverse problems in astronomical adaptive optics". In: *Inverse Problems* 25.6 (2009), p. 063001.

[11] F. Ferreira et al. "Real-time end-to-end AO simulations at ELT scale on multiple GPUs with the COMPASS platform ". In: *Adaptive Optics Systems VI*. Ed. by Laird M. Close, Laura Schreiber, and Dirk Schmidt. Vol. 10703. International Society for Optics and Photonics. SPIE, 2018, pp. 1155–1166. DOI: 10.1117/12.2312593. URL: https://doi.org/10.1117/12.2312593.

[12] Florian Ferreira et al. "COMPASS: An Efficient GPU-based Simulation Software for Adaptive Optics Systems". In: *2018 International Conference on High Performance Computing Simulation (HPCS)*. 2018, pp. 180–187. DOI: 10.1109/HPCS.2018.00043.

[13] I. Foppiano, E. Diolaiti, and P. Ciliegi. *MAORY Adaptive Optics Real-Time Computer User Requirements*. Tech. rep. 2018.

[14] T. Fusco et al. "Optimal wave-front reconstruction strategies for multi conjugate adaptive optics". In: *J. Opt. Soc. Am. A* 18.10 (2001), pp. 2527–2538.

[15] T. Fusco et al. *Optimal wavefront reconstruction strategies for multi conjugate adaptive optics*. J. Opt. Soc. Am. A 18(10):2527–2538. 2001.

[16] D. Gavel. "Tomography for multiconjugate adaptive optics systems using laser guide stars". In: *SPIE Astronomical Telescopes and Instrumentation* 5490 (June 2004), pp. 1356–1373.

[17] L. Gilles and B. Ellerbroek. "Split atmospheric tomography using laser and natural guide stars". In: *J. Opt. Soc. Am.* 25.10 (2008), pp. 2427–35.

[18] L. Gilles, B. Ellerbroek, and C. Vogel. "A comparison of Multigrid V-cycle versus Fourier Domain Preconditioning for Laser Guide Star Atmospheric Tomography". In: *Adaptive Optics: Analysis and Methods/Computational Optical Sensing and Imaging/Information Photonics/Signal Recovery and Synthesis Topical Meetings on CD-ROM, OSA Technical Digest (CD)*. Optical Society of America, 2007.

[19] L. Gilles, B. Ellerbroek, and C.R. Vogel. "Layer-Oriented Multigrid Wavefront Reconstruction Algorithms for Multi-Conjugate Adaptive Optics". In: *Proc. SPIE* 4839 (2002).

[20] L. Gilles, B.L. Ellerbroek, and C.R. Vogel. "Preconditioned conjugate gradient wave-front reconstructors for multiconjugate adaptive optics". In: *Applied Optics* 42.26 (2003), pp. 5233–5250.

[21] D. Gratadour. "Green Flash: Exploiting future and emerging computing technologies for AO RTC at ELT scale". In: *Adaptive Optics Systems V*. Vol. 9909. International Society for Optics and Photonics. AO4ELT5, 2017.

[22] D. Gratadour et al. “Green FLASH: energy efficient real-time control for AO”. In: *Adaptive Optics Systems V*. Ed. by Enrico Marchetti, Laird M. Close, and Jean-Pierre Véran. Vol. 9909. International Society for Optics and Photonics. SPIE, 2016, pp. 1314–1326. DOI: 10.1117/12.2232642. URL: https://doi.org/10.1117/12.2232642.

[23] F. Natterer. *The Mathematics of Computerized Tomography*. Wiley, 1986.

[24] A. Neubauer and R. Ramlau. “A Singular-Value-Type Decomposition for the Atmospheric Tomography Operator”. In: *SIAM Journal on Applied Mathematics* 77.3 (May 2017), pp. 838–853. URL: https://doi.org/10.1137/16M108135X.

[25] Christian Patauner et al. “FPGA based microserver for high performance real-time computing in Adaptive Optics”. In: AO4ELT5, 2017.

[26] Markus Pöttinger, Ronny Ramlau, and Günter Auzinger. “A new temporal control approach for SCAO systems”. In: *Inverse Problems* 36.1 (Dec. 2019), p. 015002. DOI: 10.1088/1361-6420/ab44dc. URL: https://doi.org/10.1088%2F1361-6420%2Fab44dc.

[27] S. Raffetseder. “Optimal Mirror Deformation for Multi-Conjugate Adaptive Optics”. MA thesis. Johannes Kepler University Linz, 2014.

[28] S. Raffetseder, R. Ramlau, and M. Yudytskiy. “Optimal mirror deformation for multi conjugate adaptive optics systems”. In: *Inverse Problems* 32.2 (2016), p. 025009. URL: http://stacks.iop.org/0266-5611/32/i=2/a=025009.

[29] R. Ramlau and M. Rosensteiner. “An efficient solution to the atmospheric turbulence tomography problem using Kaczmarz iteration”. In: *Inverse Problems* 28.9 (2012), p. 095004.

[30] R. Ramlau et al. “Efficient iterative tip/tilt reconstruction for atmospheric tomography”. In: *Inverse Problems in Science and Engineering* 22.8 (2014), pp. 1345–1366. DOI: 10.1080/17415977.2013.873534. eprint: http://dx.doi.org/10.1080/17415977.2013.873534. URL: http://dx.doi.org/10.1080/17415977.2013.873534.

[31] Ronny Ramlau and Bernadett Stadler. “An augmented wavelet reconstructor for atmospheric tomography”. In: *Electron. Trans. Numer. Anal.* 54 (2021), pp. 256–275. DOI: 10.1553/etna_vol54s256.

[32] Clélia Robert et al. “Tomographic wavefront error using multi-LGS constellation sensed with Shack-Hartmann wavefront sensors”. In: *JOSA A* 27.11 (2010), A201–A215.

[33] F. Roddier. *Adaptive Optics in Astronomy*. Cambridge: Cambridge, U.K. ; New York : Cambridge University Press, 1999, p. 411.

[34] M. C. Roggemann and B. Welsh. *Imaging through turbulence*. CRC Press laser and optical science and technology series. CRC Press, 1996.

[35] M. Rosensteiner and R. Ramlau. “The Kaczmarz algorithm for multi-conjugate adaptive optics with laser guide stars”. In: *J. Opt. Soc. Am. A* 30.8 (2013), pp. 1680–1686.

[36] D. Saxenhuber and R. Ramlau. “A Gradient-based method for atmospheric tomography”. In: *Inverse Problems and Imaging* 10.3 (2016), pp. 781–805. DOI: http://dx.doi.org/10.3934/ipi.2016022.

[37] Bernadett Stadler, Roberto Biasi, and Ronny Ramlau. “Feasibility of standard and novel solvers in atmospheric tomography for the ELT”. In: *Proc. AO4ELT6*. 2019.

[38] Bernadett Stadler and Ronny Ramlau. “Performance of an iterative wavelet reconstructor for the Multi-conjugate Adaptive Optics RelaY of the Extremely Large Telescope”. In: *Journal of Astronomical Telescopes, Instruments, and Systems* 8.2 (2022), p. 021503. DOI: 10.1117/1.JATIS.8.2.021503. URL: https://doi.org/10.1117/1.JATIS.8.2.021503.

[39] Bernadett Stadler et al. “Parallel implementation of an iterative solver for atmospheric tomography”. In: *2021 21st International Conference on Computational Science and Its Applications (ICCSA)*. 2021, pp. 123–132. DOI: 10.1109/ICCSA54496.2021.00026.

[40] M. Tallon et al. “Fractal iterative method for fast atmospheric tomography on extremely large telescopes”. In: *Proc. SPIE 7736, Adaptive Optics Systems II*. 2010, pp. 77360X-77360X–10. DOI: 10.1117/12.858042. URL: +%20http://dx.doi.org/10.1117/12.858042.

[41] Michel Tallon et al. “Performances of MCAO on the E-ELT using the Fractal Iterative Method for fast atmospheric tomography”. In: *Adaptive Optics for ELTs II* (2011).

[42] E. Thiébaut and M. Tallon. “Fast minimum variance wavefront reconstruction for extremely large telescopes”. In: *J. Opt. Soc. Am. A* 27 (2010), pp. 1046–1059.

[43] Andrei Tokovinin, Miska Le Louarn, and Marc Sarazin. “Isoplanatism in a multiconjugate adaptive optics system”. In: *JOSA A* 17.10 (2000), pp. 1819–1827.

[44] C.R. Vogel and Q. Yang. “Fast optimal wavefront reconstruction for multi-conjugate adaptive optics using the Fourier domain preconditioned conjugate gradient algorithm”. In: *Optics Express* 14.17 (2006).

[45] C.R. Vogel and Q. Yang. “Multigrid algorithm for least-squares wavefront reconstruction”. In: *Applied Optics* 45.4 (2006), pp. 705–715.

[46] Lianqi Wang and Brent Ellerbroek. “Computer simulations and real-time control of ELT AO systems using graphical processing units”. In: *Adaptive Optics Systems III*. Ed. by Brent L. Ellerbroek, Enrico Marchetti, and Jean-Pierre Véran. Vol. 8447. International Society for Optics and Photonics. SPIE, 2012, pp. 780–790. DOI: 10.1117/12.926500. URL: https://doi.org/10.1117/12.926500.

[47] Q. Yang, C.R. Vogel, and B.L. Ellerbroek. “Fourier domain preconditioned conjugate gradient algorithm for atmospheric tomography”. In: *Applied Optics* 45.21 (2006), pp. 5281–5293.

[48] M. Yudytskiy. “Wavelet methods in adaptive optics”. PhD thesis. Johannes Kepler University Linz, 2014.

[49] M. Yudytskiy, T. Helin, and R. Ramlau. “A frequency dependent preconditioned wavelet method for atmospheric tomography”. In: *Third AO4ELT Conference - Adaptive Optics for Extremely Large Telescopes*. May 2013. DOI: 10.12839/AO4ELT3.13433.

[50] M. Yudytskiy, T. Helin, and R. Ramlau. “Finite element-wavelet hybrid algorithm for atmospheric tomography”. In: *J. Opt. Soc. Am. A* 31.3 (Mar. 2014), pp. 550–560. DOI: 10.1364/JOSAA.31.000550. URL: http://josaa.osa.org/abstract.cfm?URI=josaa-31-3-550.