

## Sparse data-driven wavefront prediction for large-scale adaptive optics

Cerqueira, Paulo; Piscaer, Pieter; Verhaegen, Michel

**DOI**

[10.1364/JOSAA.425668](https://doi.org/10.1364/JOSAA.425668)

**Publication date**

2021

**Document Version**

Final published version

**Published in**

Journal of the Optical Society of America A: Optics and Image Science, and Vision

**Citation (APA)**

Cerqueira, P., Piscaer, P., & Verhaegen, M. (2021). Sparse data-driven wavefront prediction for large-scale adaptive optics. *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, 38(7), 992-1002. <https://doi.org/10.1364/JOSAA.425668>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.



# Sparse data-driven wavefront prediction for large-scale adaptive optics

PAULO CERQUEIRA,\* PIETER PISCAER,  AND MICHEL VERHAEGEN 

Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, The Netherlands

\*Corresponding author: [pcbcerqueira@gmail.com](mailto:pcbcerqueira@gmail.com)

Received 23 March 2021; revised 19 May 2021; accepted 24 May 2021; posted 26 May 2021 (Doc. ID 425668); published 22 June 2021

**This paper presents a computationally efficient wavefront aberration prediction framework for data-driven control in large-scale adaptive optics systems. Our novel prediction algorithm splits prediction into two stages: a high-resolution and a low-resolution stage. For the former, we exploit sparsity structures in the system matrices in a data-driven Kalman filtering algorithm and constrain the identified gain to be likewise sparse; for the latter, we identify a dense Kalman gain and perform corrections to the suboptimal predictions of the former on a smaller grid. This novel prediction framework is able to retain the robustness to measurement noise of the standard Kalman filter in a much more computationally efficient manner, in both its offline and online aspects, while minimally sacrificing performance; its data-driven nature further compensates for modeling errors. As an intermediate result, we present a sparsity-exploiting data-driven Kalman filtering algorithm able to quickly estimate an approximate Kalman gain without solving the Riccati equation.** © 2021 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

<https://doi.org/10.1364/JOSAA.425668>

## 1. INTRODUCTION

Adaptive optics (AO) systems aim to compensate for incoming phase aberrations in optical systems. The incoming light is split between a focal plane camera and a wavefront sensor (WFS). A reconstruction of the aberrated wavefront is developed from the WFS image, which is used by a controller that regulates a deformable mirror (DM) that is able to correct for the phase aberrations. Determining a minimum variance wavefront reconstruction has been extensively studied [1,2]. Due to the existence of modes of the wavefront that are not visible to the sensor and the presence of measurement noise, this is not trivial. Usually, these issues are addressed by including prior information regarding the statistical properties of the aberration and noise.

Due to a delay within the AO control loop, each control input is used to correct an aberration ahead in time, meaning that wavefront prediction techniques are important for effective correction. Extending the minimum variance wavefront reconstruction to the online prediction case, with a dynamic wavefront subject to process and measurement noises, suggests the use of a Kalman filter (KF) in an optimal control framework [3–5]. Because computing the steady-state Kalman gain involves solving the computationally complex discrete algebraic Riccati equation (DARE), straightforward KF implementations are impractical for large-scale applications. As a result, multiple KF-based methods have been proposed to decrease the computational complexity in the large-scale AO context [6–10].

Alongside their large computational complexity, KF-based methods in AO have another important drawback: the computation of the Kalman gain via the DARE requires knowledge of the temporal dynamics of the wavefront and of the statistical properties of the process and measurement noises. In the Kalman filtering literature, data-driven KFs have been proposed to avoid the need for prior knowledge of the statistical properties of both the process and measurement noises [11,12], instead relying on measurement data and the deterministic part of a state-space model; however, practical application of these techniques can be challenging in large-scale systems when limited data are available, and due to their computational complexity. While data-driven Kalman filtering has not previously been brought to AO, the reduction of reliance on prior information has motivated the development of data-driven optimal controllers [13–15] in AO, and the reduction of their computational complexity has led to an increased attention toward the use of structured models, as matrix structures such as sparsity and tensor structures appear in models describing the turbulence dynamics [15–18].

Furthermore, significant work has been dedicated to the design of fast wavefront reconstruction algorithms for online aberration correction [10,19–21], which must accompany the sampling frequency of the system in large-scale AO systems. Among these, the idea of conducting operations on multiple grids of different densities has been explored in methods such as Refs. [20,21] and is explored likewise in the present work. Specifically, we propose a second low-resolution grid designed

to intuitively fit into the discussed modeling and data-driven Kalman filtering frameworks with lessened computational burden.

This paper addresses both the offline and online computational complexities involved in wavefront prediction in a data-driven setting. First, it introduces sparse matrix techniques into data-driven Kalman filtering using AO modeling insights, allowing obtention of the Kalman gain more efficiently than with the DARE and with less reliance on prior knowledge, and serving as a first step for the second contribution. Second, it proposes a framework, the high-low-resolution filter (HLRF), in which wavefront prediction is split into a high-resolution stage and a low-resolution one; by enforcing sparsity in the gain identified for the high-resolution stage and correcting the error thus induced with the low-resolution stage, a computationally efficient *online* prediction framework is created, while reducing the *offline* (identification) time complexity down to linear in the total number of lenslets in the WFS array.

The novelty of the new proposed data-driven framework for predictive AO control is twofold with respect to existing prediction solutions [7–10,19–21]. First, it incorporates information about the process and measurement noises without assuming it is known beforehand, instead deriving it implicitly from measurement data. Second, efficient algorithms, both for the off- and online computations, are derived; in particular, the HLRF sacrifices some of the performance of the KF due to heuristic choices, with simulations showing that it nonetheless achieves high performance in comparison with the full data-driven and DARE-based KFs, with improved prediction times.

This paper is structured as follows: Section 2 provides an introduction to Kalman filtering theory in standard AO systems, including the modeling of the WFS and atmospheric turbulence; the data-driven Kalman filtering algorithm from [22] is overviewed in Section 3; Section 4 discusses an implementation of this algorithm in AO that exploits sparsity structures in the system matrices and introduces the HLRF. The results of both data-driven Kalman filtering and the HLRF are discussed in Section 5. The main conclusions and practical applicability of the methods are addressed in Section 6.

*Notation:* When considering the temporal dynamics, the current discrete time-step is denoted by a subscript  $k$ , i.e.,  $\phi_k$  represents the wavefront at moment  $k$  in time. The pseudo-inverse of a rectangular matrix is denoted by  $A^\dagger$ . The computational complexity, which is considered to be the order of magnitude of the total number of elementary arithmetic operations, is expressed by  $O(\cdot)$ . Rather than including both the number of pixels  $n$  of the predictions and the number  $m$  of lenslets in the WFS array in the complexities,  $m$  will take the place of  $n$ , using the fact that  $n \approx m/2$  in large-scale AO applications. Notation  $w \sim \mathcal{N}(\mu, Q)$  is used to describe a Gaussian random (vector) signal  $w$  with mean  $\mu$  and covariance  $Q$ , and  $E(\cdot)$  denotes the expectation operator. Operator  $\|\cdot\|_F$  represents the Frobenius norm.

## 2. KALMAN FILTERING IN AO

This section provides an introduction to the application of standard KF theory in AO systems. The theory presented in this section can be found in various textbooks; see, for example, [23]

for more information about AO and [24] for more information about system identification and Kalman filtering.

### A. Modeling for AO

This subsection will introduce the components of a standard AO system for astronomy and will discuss the aberration dynamics caused by atmospheric turbulence. A square aperture is considered so as to accommodate the HLRF, as detailed later.

#### 1. WFS

We consider a grid of  $L \times L$  lenslets measuring local wavefront gradients (often called *slopes*)  $y_k \in \mathbb{R}^m$  corresponding to the subaperture of each lenslet (as is the case with Shack–Hartmann or, for example, pyramid sensors). Assuming a square aperture and *Fried geometry*,  $m = 2L^2$ , and the wavefront is sampled on a grid of  $(L+1) \times (L+1)$  *pixels*. Defining the total number of pixels as  $n = (L+1)^2$ , the relation between the sensor signal  $y_k$  and the discretized wavefront  $\phi_k \in \mathbb{R}^n$  is formulated as

$$y_k = G\phi_k + v_k, \quad (1)$$

where  $G \in \mathbb{R}^{m \times n}$  is the *sensor geometry matrix* and  $v_k$  is a measurement noise vector. Usually,  $v_k$  is assumed to be a white Gaussian measurement noise vector with diagonal covariance matrix  $R$ , i.e.,  $v_k \sim \mathcal{N}(0, R)$  [3,8,15].

This sensor configuration is unable to measure two particular modes of the wavefront phase: the piston (spatially constant) and waffle (checkerboard) modes, which is reflected in a rank deficiency of  $G$ , with  $\text{rank}(G) = n - 2$ .

#### 2. Modeling the Turbulence Dynamics

The wavefront aberrations caused by atmospheric turbulence are time-varying and commonly assumed to adhere to Taylor's frozen flow assumption [25], according to which atmospheric turbulence is represented by multiple "frozen" turbulence layers, each moving at its own constant speed in its own direction. A common choice to model the temporal evolution of atmospheric turbulence is the vector auto-regressive (VAR) model of order 1 [3,4,8,26],

$$\phi_{k+1} = A\phi_k + w_k, \quad (2)$$

where  $w_k$  is assumed to be Gaussian white noise  $w_k \sim \mathcal{N}(0, Q)$ , uncorrelated with  $\phi_k$ . Since the speed and direction of the phase screens are typically unknown, a common choice is to assume the state-transition matrix  $A$  is diagonal, ignoring the temporal dynamics of the wavefront [3,4,8]. Although this model may be sufficiently accurate when the movement of the turbulence in between consecutive time steps is negligible, a better  $A$  would represent this movement, especially when the temporal evolution is not negligible. Defining the covariance matrices:  $C_{\phi,j} = E(\phi_{k+j}\phi_k^T)$ ,  $A$  in model (2) is given by

$$A = C_{\phi,1}C_{\phi,0}^{-1}. \quad (3)$$

In practice, covariances  $C_{\phi,0}$  and  $C_{\phi,1}$  are unknown and would have to be estimated using a finite set of wavefront data. Let the wavefront phase data, available from time step 0 to  $N$ , be arranged into "past" and "future" data matrices,

$$\Phi_{0,N} = [\phi_0 \cdots \phi_{N-1}]; \quad \Phi_{1,N} = [\phi_1 \cdots \phi_N].$$

Then, an estimate  $\hat{A}$  of  $A$  can be obtained from a linear least-squares problem,

$$\hat{A} = \underset{A}{\operatorname{argmin}} \|\Phi_{1,N} - A\Phi_{0,N}\|_F^2, \quad (4)$$

which, in the limit as  $N \rightarrow \infty$ , is equivalent to Eq. (3). Another practical complication is that exact wavefront data are unavailable, as they are not directly measurable by sensors. Equation (4) will therefore have to rely on reconstructed wavefront data instead, which introduces additional errors in the model.

### 3. Modeling Errors

On top of the wavefront data set being estimated, a major cause of modeling errors is that the true dynamic system driving the turbulence is more complex than the simple VAR-1 model of Eq. (2). For one, when considering the atmosphere as a stack of multiple dynamic phase screens with each phase screen dynamics exactly represented by a VAR-1 model, their combined influence in the observed wavefront  $\phi_k$  is no longer described by any VAR model in terms of  $\phi_k$ . As a consequence, the VAR-1 model can only approximate the true system dynamics.

The VAR-1 model is additionally only capable of accurately representing movements of the wavefront phase screen that are exact pixel-size shifts in between time steps, given the definition of a pixel as in Section 2.A.1. This issue is addressed in the appendix of [27], which proposes using a lagged version of the VAR-1 model in Eq. (2) such that it more closely represents whole-pixel movement, and running alternating KFs at no additional computational cost.

Both of these modeling errors result in VAR-1 models with nonwhite  $w_k$ , thus violating one of the assumptions made in KF design and negatively impacting its performance. Due to its data-driven nature, the Kalman filtering algorithm presented in this paper is nonetheless capable of compensating for a certain degree of modeling error, as illustrated in Fig. 3. More details on modeling errors are included in [27].

### B. Kalman Filtering

The combination of the VAR-1 turbulence model [Eq. (2)] and the sensor equation [Eq. (1)] yields the following state-space model:

$$\begin{aligned} \phi_{k+1} &= A\phi_k + w_k, \\ y_k &= G\phi_k + v_k, \end{aligned} \quad (5)$$

with  $\phi_k \in \mathbb{R}^n$ , and  $y_k \in \mathbb{R}^m$ . In the context of Kalman filtering, the vector  $\phi_k$  is known as the state vector and  $y_k$  is the output vector. Recall that vectors  $w_k$  and  $v_k$  are zero-mean Gaussian white noise sequences and are here assumed uncorrelated with each other, i.e.,  $E[w_k v_k^T] = 0$ . Process noise  $w_k$  is further uncorrelated with state  $\phi_k$ .

Since the model is linear time-invariant (LTI), the prediction of the steady-state KF is given in observer form by Eq. (6) and in innovation form by Eq. (7) [24],

$$\hat{\phi}_{k+1} = (A - KG)\hat{\phi}_k + Ky_k, \quad (6)$$

$$\hat{\phi}_{k+1} = A\hat{\phi}_k + Ke_k, \quad (7)$$

where  $e_k \sim \mathcal{N}(0, R_e)$  is the prediction-error sequence defined via

$$y_k = G\hat{\phi}_k + e_k. \quad (8)$$

The steady-state Kalman gain  $K$  is obtained from

$$K = APG^T(GPG^T + R)^{-1}, \quad (9)$$

where  $P \in \mathbb{R}^{n \times n}$  is the state-prediction-error covariance matrix, given by the positive-definite solution to the DARE,

$$P = APA^T + Q - APG^T(GPG^T + R)^{-1}GPA^T. \quad (10)$$

The computational complexity of the KF will be split into online and offline complexities. The *online* computational complexity will refer to the computations that have to be performed within each sample time of the KF. In the case of the standard LTI KF of this section, the online computation pertains to Eq. (7), which has a computational complexity of  $\mathcal{O}(m^2)$ . The combined computational complexity of all the other operations will be referred to as the *offline* computational complexity. Finding the solution to the DARE and applying Eq. (9) are, with a complexity of  $\mathcal{O}(m^3)$ , the offline bottlenecks in the computation of the Kalman gain  $K$ .

## 3. DATA-DRIVEN KALMAN FILTERING

The previous section discussed how the use of standard Kalman filters has several difficulties when applied to large-scale AO applications. The data-driven Kalman filtering algorithm by authors Juang and Chen [22], which estimates the Kalman gain directly from measurement data, is here introduced and shortly derived. The use of this data-driven Kalman filtering algorithm instead of the standard DARE-based KF is motivated by the following three properties:

- The data-driven algorithm directly identifies the Kalman gain without solving the DARE nor applying Eq. (9), replacing these with equations that are apt for exploitation of the sparsity of the system matrices in AO.
- It requires only (an estimate of) the deterministic part of the state-space model in Eq. (5) (matrices  $A$  and  $G$ ) and measurement data, implicitly accounting for the stochastic part of the model (matrices  $Q$  and  $R$ ).
- It compensates for a certain degree of modeling error by implicitly considering the modeling errors to be a form of process noise [28] and identifying a corresponding Kalman gain. Given the sources of modeling error discussed in Section 2, it is possible for the identified Kalman gain to outperform the DARE-based gain.

Let matrices  $Y_{i,j,N}$  and  $Y_{i,N}$  be defined as

$$Y_{i,j,N} = \begin{bmatrix} y_i & y_{i+1} & y_{i+2} & \cdots & y_{i+N-1} \\ y_{i+1} & y_{i+2} & & \cdots & y_{i+N} \\ \vdots & & & & \vdots \\ y_{i+j-1} & y_{i+j} & & \cdots & y_{i+j+N-2} \end{bmatrix}, \quad (11)$$

$$Y_{i,N} = [y_i \ y_{i+1} \ y_{i+2} \ \cdots \ y_{i+N-1}] = Y_{i,1,N}$$

where  $Y_{i,j,N} \in \mathbb{R}^{jm \times N}$  is an  $N$ -column block Hankel matrix with  $j$  block rows of output information  $y_k$ . Define  $E_{i,j,N}$  and  $E_{i,N}$  analogously for the prediction errors  $e_k$ . Henceforth, assume data are available from time step 0 up to time step  $N_d$ , for a total data batch length of  $N_d + 1$ . Define as well  $\mathcal{O}_j \in \mathbb{R}^{jm \times n}$  as the extended observability matrix up to power  $j - 1$  of  $A$ ,

$$\mathcal{O}_j = [G^T (GA)^T \dots (GA^{j-1})^T]^T. \quad (12)$$

Consider system (5) and its KF, operating in steady state. Assuming asymptotic stability of the system, it is possible to find an order  $s$  such that  $(A - KG)^k \approx 0$  for any  $k \geq s$ . Using this property, recursively introducing the KF prediction in observer form [Eq. (6)] into itself and combining it with Eq. (8) results in, for an arbitrary time index  $k$ ,

$$\hat{y}_k = G \underbrace{\left[ (A - KG)^{s-1} K \dots (A - KG) K \right]}_{\mathcal{L}} \begin{bmatrix} y_{k-s} \\ \vdots \\ y_{k-2} \\ y_{k-1} \end{bmatrix} + e_k. \quad (13)$$

Let there be a total of  $N_d + 1$  time steps of measurement data from time step 0 to  $N_d$ . Then, stacking the left-hand side of (13) horizontally into matrix  $Y_{s,N}$ ,

$$Y_{s,N} = G\mathcal{L}Y_{0,s,N} + E_{s,N}, \quad (14)$$

where  $N = N_d - s - p + 2$ , so that  $y_{s+p+N-2} = y_{N_d}$ . The first step toward identification of the Kalman gain is estimation of  $G\mathcal{L}$ , which can be done in a linear least-squares setting,

$$\widehat{G\mathcal{L}} = \operatorname{argmin}_{G\mathcal{L}} \| Y_{s,N} - (G\mathcal{L})Y_{0,s,N} \|_{\mathbb{F}}^2. \quad (15)$$

Recalling Eq. (13), note that Eq. (15) is fitting a VAR model of order  $s$  (i.e., a VAR- $s$  model) to measurement data. Define now the Markov parameters of the steady-state KF in observer form [Eq. (16)] and innovation form [Eq. (17)],

$$\mathcal{A}_j = G(A - KG)^{j-1}K, \quad (16)$$

$$\mathcal{B}_j = GA^{j-1}K. \quad (17)$$

With  $G\mathcal{L}$  estimated from Eq. (15), one has obtained estimates of the observer form Markov parameters  $\mathcal{A}_j$  up to  $j = s$ . From these, the next step is to compute the innovation form Markov parameters,

$$\widehat{\mathcal{B}}_j = \widehat{\mathcal{A}}_j + \sum_{i=1}^{j-1} \widehat{\mathcal{B}}_{j-i} \widehat{\mathcal{A}}_i, \quad (18)$$

where  $j \in \{2, 3, \dots, p\}$  and  $p$  is a tuning parameter such that  $2 \leq p \leq s$ . The estimated innovation form Markov parameters, stacked vertically, yield an estimate of  $\mathcal{O}_p K$ ,

$$\widehat{\mathcal{O}_p K} = \begin{bmatrix} \widehat{\mathcal{B}}_1 \\ \vdots \\ \widehat{\mathcal{B}}_p \end{bmatrix} = \begin{bmatrix} \widehat{GK} \\ \vdots \\ \widehat{GA^{p-1}K} \end{bmatrix}, \quad (19)$$

from which an estimate  $\widehat{K}$  of  $K$  can be retrieved in a second linear least-squares problem,

$$\widehat{K} = \operatorname{argmin}_K \| \widehat{\mathcal{O}_p K} - \mathcal{O}_p K \|_{\mathbb{F}}^2. \quad (20)$$

Assuming that  $Y_{0,s,N}$  has full row rank, the solution to Eq. (15) is unique. Further assuming the system is observable and that  $p$  is set large enough,  $\mathcal{O}_p$  has full column rank, and the solution  $\widehat{K}$  to (20) is likewise unique. This estimator for  $K$  is asymptotically unbiased in  $s$ .

The data-driven Kalman filtering procedure of [22] is summarized as follows:

- (1) Gather the data into  $Y_{s,N}$  and  $Y_{0,s,N}$  as in Eqs. (11) and (14).
- (2) Solve the least-squares problem in Eq. (15) for an estimate  $\widehat{G\mathcal{L}}$  of  $G\mathcal{L}$ .
- (3) Compute estimates of the innovation form Markov parameters as per Eq. (18) and build  $\widehat{\mathcal{O}_p K}$  according to Eq. (19).
- (4) Solve the least-squares problem in Eq. (20) for an estimate  $\widehat{K}$  of  $K$ .

#### 4. SPARSE METHODS FOR LARGE-SCALE AO

Solving the optimization problems in Eqs. (15)–(20) for an arbitrary system with dense system matrices has a computational complexity of  $\mathcal{O}(m^3)$ . This section will describe the intuitive reasoning behind sparsity patterns in AO, enabled by the frozen-flow nature of turbulence [17,18] and exploit them in two novel algorithms. The first, in Subsection 4.B, combines the data-driven Kalman filtering of Section 3 with sparse VAR modeling to quickly estimate an approximate Kalman gain, but still in  $\mathcal{O}(m^3)$  time. The second algorithm, in Subsection 4.C, aims to reduce the online computational complexity by directly enforcing sparsity in the identified gain and compensating for the error thus induced with an additional set of fast operations, while additionally reducing the offline identification complexity down to  $\mathcal{O}(m)$ .

##### A. Sparse Turbulence Modeling

The sparsity of the matrix  $A$  can easily be illustrated by recalling the frozen turbulence assumption. Assuming a wind speed of  $v$ , the frozen turbulence assumption implies that, when  $v$  is at least overestimated, and without assuming knowledge of the wind direction, one knows that information travels only within a radius of  $r = \tau v$  in between consecutive time steps,  $\tau$  being the sampling time. Because  $A$  in Eq. (2) establishes this movement within a single sampling period, a sparsity pattern can be set beforehand to directly identify a sparse model.

Consequently, for a single turbulence layer, the sparsity pattern of  $A$  is dictated by the wind speed given in pixels per time step, here denoted by  $\omega$ ,

$$\omega = \frac{v\tau}{\delta}, \quad (21)$$

such that  $\delta$  will be referred to as the *pixel width*, which is defined as the width of a single pixel:  $\delta = D/L$ . The sparsity structure

of  $A$  is then proposed as follows: any entries of  $A$  that relate pixels further than  $\omega$  pixel widths away from each other should be zero. With the adoption of this *a priori* structure for matrix  $A$ , a sparse multibanded matrix is obtained, similar to the structure adopted in [17].

In practice, a single layer may not accurately represent the total atmosphere, and the wind speed  $v$  should be replaced by the dominant wind speed of all the phase screens. The exact wind speed may also be unknown; it is therefore necessary to approximate the sparsity pattern based on the limited information available: if a rough estimate (or overapproximation) of the wind speed is available, this can be used instead in Eq. (21). For example, when  $\delta = 0.05$  m,  $\tau = 0.005$  s and an overestimation  $v \leq 20$  m/s is available, then  $\omega \leq 2$  pixel widths per sampling time.

Let  $\mathcal{S}_A$  denote the set of valid  $A$  matrices with the previously described sparsity pattern. The linear least-squares problem in Eq. (4), reformulated as

$$\hat{A} = \underset{A \in \mathcal{S}_A}{\operatorname{argmin}} \|\Phi_{1,N} - A\Phi_{0,N}\|_F^2 \quad (22)$$

and assuming that there is an average number  $q$  of nonzeros per row of  $A$ , can be solved in  $O(Nq^2m)$  time.

## B. Sparse Data-Driven Kalman Filtering

This section deals with the introduction of sparsity into the data-driven Kalman filtering algorithm of Section 3 and its use in AO. The algorithm is divided into two linear least-squares problems: the first problem fits a VAR model to measurement data, and the second one determines the Kalman gain from these models. These are explored separately ahead.

### 1. Sparse VAR Slope Modeling

In order to obtain efficient solutions to the optimization problem presented in Eq. (15), which fits a VAR- $s$  model to slope data, it is important to obtain a sparse representation of matrix  $GL$ . Consider a VAR- $s$  model for the slopes,

$$y_k = \mathcal{A}_1 y_{k-1} + \mathcal{A}_2 y_{k-2} + \cdots + \mathcal{A}_s y_{k-s} + \xi_k. \quad (23)$$

Following an intuition analogous to that of Section 4.A, a similar radius-based sparsity pattern can be set, leading to a method for sparse VAR- $s$  modeling, as presented in [18]. It should be remarked that an overestimate of  $\omega$  is essential in this case, as it was observed to have a smoothing effect to deal with measurement noise.

In the model of Eq. (23), each  $\mathcal{A}_i$  coefficient establishes a relationship between slope vectors  $i$  time steps apart, so each coefficient should have a sparsity radius  $r_i$  that increases with  $i$ .

Let  $\mathcal{S}_{GL}$  refer to the set of valid  $GL$  matrices with a sparsity pattern fulfilling the description above, and constrain the first least-squares problem of the data-driven Kalman filtering procedure [Eq. (15)],

$$\widehat{GL} = \underset{(GL) \in \mathcal{S}_{GL}}{\operatorname{argmin}} \|Y_{s,N} - (GL)Y_{0,s,N}\|_F^2. \quad (24)$$

Assuming an average of  $q$  nonzero elements per row of  $GL$ , Eq. (24) can be solved in  $O(Nq^2m)$  time. Without exploiting sparsity, solving Eq. (15) would take  $O(Ns^2m^2 + s^2m^3)$  time instead. More details regarding the implementation of Eq. (24) are provided in [27].

Furthermore, the consequently low number of parameters to identify permits, for large-scale application, the use of “tall” data sets, with a number of time-steps of data much smaller than the number  $m$  of lenslets in the WFS array.

### 2. Retrieval of the Kalman Gain

After constructing the sparse estimate  $\widehat{O}_p K$  from Eqs. (18) and (19), which involves only negligible  $O(m)$  sparse matrix products and additions for sensible choices of  $s$  and  $p$ , the remaining final step is to obtain a computationally efficient solution to Eq. (20). However, since the Kalman gain itself is dense, the approach used to formulate the sparse solutions in Eqs. (22) and (24) is not applicable in this case.

Instead, the linear least-squares problem of Eq. (20) has a highly sparse regressor  $O_p$  and regressand  $\widehat{O}_p K$ , which enables the use of dedicated solvers. The sparse rectangular solver built into MATLAB’s \ operator is [29] as of R2020b, and solves Eq. (20) in roughly  $O(m^3)$  time, as is the case without leveraging sparsity or finding the gain with the DARE, but is efficient in practice due to its sparsity-exploiting nature, leading to the results of Fig. 2 that confirm that the entire identification procedure is far faster than the DARE.

Alongside the computational complexity of its computation, which remains the bottleneck for applications of arbitrarily large scale, another major drawback of the dense matrix  $K$  is that it also directly influences the online predictions, given by Eq. (7). Since the online computations involve matrix-vector multiplications with  $K$ , the online computational complexity will scale as  $O(m^2)$ . In order to obtain a scalable online computational complexity, the use of the large and dense matrix  $K$  should be avoided. Section 4.C introduces a new framework, partly based on the presented sparsity-exploiting data-driven KF, that aims to decrease the computational complexity of the online computations and, in addition, decreases the offline computational complexity further.

### 3. Implementation Details and Approximations in AO

Practical issues arise when applying the proposed data-driven Kalman filtering method to AO systems using the approximate VAR-1 turbulence model derived in Section 2. This subsection acknowledges and discusses these issues.

The assumption that  $(A - KG)^s \approx 0$  any small  $s$  generally does not hold for the AO case. Because sparsity is progressively lost with increases in  $s$ ,  $s$  is nonetheless consciously kept low, meaning that the resulting Kalman gain will be approximate. In practice, the presence of the modeling errors (see Section 2) implies that the identified Kalman gain may outperform the DARE-based gain even for low values of  $s$  and that arbitrarily increasing  $s$  does not necessarily mean better performance. The influence of  $s$  on the performance of the KF will be addressed in the simulation results of Section 5.

Furthermore, a near-1 eigenvalue of  $(A - KG)$  corresponding (approximately) to the piston mode leads as well to the appearance of column-wise constant distortions in the identified Kalman gain due to the implicit pseudo-inversion of  $\mathcal{O}_p$  when solving for  $\hat{K}$  in Eq. (20). The column-wise means of the identified Kalman gain should therefore be removed after retrieval with Eq. (20), as should the piston mode of the filter's predictions, lest their mean become unstable and lead to numerical problems.

### C. HLRF

The computational complexity of the standard KF prediction operation [Eq. (7)] is  $\mathcal{O}(m^2)$ . Even with a sparse  $A$ , the denseness of the Kalman gain remains an issue. This section presents an algorithm that aims to have the noise robustness of Kalman filtering, reduce the computation time of the online prediction operation, and improve the time complexity of the offline identification procedure. The algorithm separates prediction into two distinct stages, one in the original array using only sparse matrix operations, and one in an artificially constructed low-resolution array where dense operations are acceptable without significant computational burden. These are as follows:

- The **high-resolution stage** performs prediction in the original, potentially extremely large fine array. In this stage, only sparse operations are acceptable. To achieve this, high sparsity is imposed in the identified gain, resulting in a data-driven sparse observer that is capable of predicting detailed local high-spatial-frequency structures in the wavefront phase, but fails to satisfactorily account for its overarching shape.

- The **low-resolution stage** predicts the remainder of the wavefront in a coarse, low-resolution array using a (dense) data-driven KF. In fact, this stage is effectively predicting the state prediction errors of the sparse observer of the high-resolution stage; these prediction errors comprise low-spatial-frequency modes of the wavefront phase, which should not require an array as dense as the original one for prediction. This low-resolution array is a mathematical construct whose measurements are built from measurement data from the original array using sparse operations likewise, hence demanding no hardware modification.

The algorithm is formulated for square lenslet arrays due to the current construction of the low-resolution stage, which is done by designing each lenslet of the low-resolution array to be a square comprising lenslets of the original high-resolution array. Extension to circular apertures is briefly discussed in Section 6.

#### 1. Sparse Predictor Gain

The HLRF algorithm is based upon setting a radius-based sparsity pattern on the identified gain, analogous to those in Eqs. (22) and (24); denote by  $\mathcal{S}_K$  the set of potential gains  $K^*$  with such a sparsity pattern, and constrain Eq. (20) accordingly,

$$K^* = \operatorname{argmin}_{K \in \mathcal{S}_K} \|\widehat{\mathcal{O}_p K} - \mathcal{O}_p K\|_F^2. \quad (25)$$

Because none of the nonzero patterns of the matrices involved depends on  $m$ , let  $q$  denote the average nonzeros per column of  $K^*$ ,  $\mathcal{O}_p$ , and  $\widehat{\mathcal{O}_p K}$ , without distinction. A dedicated implementation of Eq. (25) can be solved in  $\mathcal{O}(mq^3)$  time, linear in total number of lenslets  $m$ . This, as opposed to the original  $\mathcal{O}(pm^3)$  time of Eq. (20) without exploitation or enforcement of sparsity. Thesis [27] provides further details regarding the implementation of Eq. (25).

Solving Eq. (25) was observed to systematically produce asymptotically stable observers. These, with a tight constraint on the gain, are able to accurately predict high-spatial-frequency modes of the wavefront, but perform poorly for the low-spatial-frequency modes, as shown in Fig. 5 of Section 5, which motivates the use of the low-resolution stage. Note that unlike the dense gain of Section 4.B, the column-wise means of the sparse gain should not be removed, as Eq. (25) will not fit for the distortions, and removal of the means would destroy sparsity.

#### 2. HLRF Formulation

This subsection overviews the HLRF algorithm. The low-resolution lenslet array is of size  $L' \times L'$ ; define  $m' = 2(L')^2$  and  $n' = (L' + 1)^2$ . The exact definitions of the low- and high-resolution sampling arrays will be discussed further in Section 4.C.3. With sparse gain  $K^*$ , the high-resolution stage is given by

$$\hat{\phi}_{k+1}^{[h]} = A\hat{\phi}_k^{[h]} + K^* e_k^{[h]}, \quad (26)$$

where  $\hat{\phi}_k^{[h]} \in \mathbb{R}^{n'}$  is its wavefront prediction for time  $k$  and the measurement prediction errors of the high-resolution stage  $e_k^{[h]} \in \mathbb{R}^{m'}$  are

$$e_k^{[h]} = y_k - G\hat{\phi}_k^{[h]}.$$

Next, for the derivation of the low-resolution stage, define a system (still in the original high-resolution array) whose states are the wavefront prediction errors of the high-resolution stage,

$$\tilde{\phi}_k^{[h]} = \phi_k - \hat{\phi}_k^{[h]}. \quad (27)$$

The dynamics of this wavefront prediction error are given by

$$\begin{aligned} \tilde{\phi}_{k+1}^{[h]} &= (A - K^* G)\tilde{\phi}_k^{[h]} + w_k - K^* v_k \\ e_k^{[h]} &= G\tilde{\phi}_k^{[h]} + v_k \end{aligned} \quad (28)$$

The goal of the low-resolution stage is to predict the wavefront prediction error of the high-resolution stage. As stated previously,  $\tilde{\phi}_k^{[h]}$  is expected to contain mostly low-spatial-frequency modes that should not require an array as dense as the original for sufficiently accurate prediction (and the highest spatial-frequency modes cannot be predicted whatsoever, due to measurement noise). The low-resolution stage therefore involves constructing an artificial low-resolution array within which a KF for Eq. (28) is designed.

Let a superscript prime indicate matrices in the low-resolution array, which are obtained as described below. Using this notation, the wavefront prediction given by the HLRF is computed as follows:

$$\begin{cases} \hat{\phi}_{k+1}^{[h]} = A\hat{\phi}_k^{[h]} + K^*e_k^{[h]} \\ \hat{\phi}_{k+1}^{[l]} = (A - K^*G)'\hat{\phi}_k^{[l]} + K'(\mathcal{H}e_k^{[h]} - G'\hat{\phi}_k^{[l]}) \\ \hat{\phi}_{k+1} = \hat{\phi}_{k+1}^{[h]} + \mathcal{I}(\hat{\phi}_{k+1}^{[l]}) \end{cases}, \quad (29)$$

where  $\hat{\phi}_k^{[l]} \in \mathbb{R}^{n'}$  is the wavefront prediction for time  $k$  of the low-resolution stage. Operator  $\mathcal{I}$  represents linear interpolation from the low-resolution array into the high-resolution array, and sparse matrix  $\mathcal{H}$  converts the measurements on the high-resolution array into measurements on the low-resolution one, as described in Section 4.C.3. Equation (29) is applied in  $O(m)$  time.

In Eq. (29), the Kalman gain  $K' \in \mathbb{R}^{n' \times m'}$  is the low-resolution (dense) Kalman gain, herein identified using the data-driven method of Section 3, considering  $\mathcal{H}e_k$  as measurement data, and  $(A - K^*G)'$  and  $G'$  as system matrices. The state-transition matrix of the low-resolution stage,  $(A - K^*G)'$ , with available wavefront phase data, can be obtained following the procedures of Section 2, after running the measurement data through the high-resolution stage first and subtracting its wavefront predictions from the corresponding wavefront data as per Eq. (27). Matrix  $G' \in \mathbb{R}^{m' \times n'}$  is built exactly like  $G$  but for the dimensions of the low-resolution array.

### 3. Definition of the Low-Resolution Array

Whereas the high-resolution array is defined by the lenslets of the WFS, the low-resolution array is a mathematical construct that is not the result of any hardware within the AO loop. This subsection will define the matrix  $\mathcal{H}$  introduced in Section 4.C.2.

In Eq. (29), the measurement prediction errors  $e_k^{[h]}$  were mapped onto the low-resolution array by matrix  $\mathcal{H}$ . In order to define this mapping, consider the problem of transforming a set of slope measurements  $y_k$  obtained from an  $L \times L$  array of WFS lenslets into an artificial set of measurements  $z_k$  that give the slopes on a lower-dimensional  $L' \times L'$  sampling grid,  $L' \ll L$ . Moreover, assume that  $L'$  is chosen such that  $L_u = L/L'$ , the width of each lenslet of the low-resolution array, henceforth called a *unit*, is an integer value.

Matrix  $\mathcal{H} \in \mathbb{R}^{m' \times m}$  linearly combines the measurements from the original array,

$$z_k = \mathcal{H}y_k, \quad (30)$$

such that, disregarding measurement noise,  $z_k$  is related to the wavefront sampled on a low-resolution array, denoted by  $\psi_k \in \mathbb{R}^{n'}$ , via

$$z_k = G'\psi_k. \quad (31)$$

The low-resolution wavefront  $\psi_k$  can be defined as a selection of entries of  $\phi_k$  that are shared by the low-resolution array. Let matrix  $Z \in \mathbb{R}^{n' \times n}$  of ones and zeros define this selection,

$$\psi_k = Z\phi_k. \quad (32)$$

Combining Eqs. (30)–(32), the problem lies in finding a combination matrix  $\mathcal{H}$  such that

$$G'Z\phi_k = \mathcal{H}G\phi_k \quad (33)$$

for any arbitrary  $\phi_k$ . In the present setting, in which the units are square, and as detailed in Appendix A, it is possible to find an exact solution to Eq. (33) for all  $\phi_k \in \mathbb{R}^n$  with a highly sparse  $\mathcal{H}$  by solving the following least-squares problem:

$$\mathcal{H} = \underset{\mathcal{H} \in \mathcal{S}_{\mathcal{H}}}{\operatorname{argmin}} \| \mathcal{H}G - G'Z \|_{\text{F}}^2, \quad (34)$$

where  $\mathcal{S}_{\mathcal{H}}$  is the set of valid  $\mathcal{H}$  matrices with a sparsity pattern that allows only intraunit combinations, resulting from the implementation in Appendix A with which the results of Section 6 were obtained.

## 5. TUNING AND SIMULATION RESULTS

This section overviews the performance of the proposed algorithms with respect to quality of the predictions and execution times. Prediction quality is measured by the mean squared 2-norm of the **piston-removed** wavefront prediction error, normalized by the mean squared 2-norm of the piston-removed incoming wavefront; this measure of performance is henceforth referred to as normalized mean-squared error (NMSE),

$$\text{NMSE} = \frac{\sum_{k=1}^N \|\hat{\phi}_k - \phi_k\|_2^2}{\sum_{k=1}^N \|\phi_k\|_2^2}, \quad (35)$$

where, to reiterate, both  $\phi_k$  and  $\hat{\phi}_k$  have their piston modes (i.e., their means) removed because offsets in the wavefront phase do not affect image quality, but would alter the NMSE. The NMSE is computed after 500 time steps of simulation to ensure steady-state operation.

As a proof of concept for the proposed algorithm, a numerical simulation study of a standard AO system is performed. A square telescope aperture is used to match the square array of WFS lenslets assumed in the obtention of matrices  $\mathcal{H}$  and  $\mathcal{I}$  in Section 4.C.3. The setting of the analyses is the following:

- The sensor array is an  $L \times L$  square array configured in Fried geometry, with  $L = 60$ . Square arrays were chosen so as to accommodate our HLRF and allow it to be compared to the alternatives; square arrays are adopted throughout for consistency.

- The turbulence parameters are set as follows: the Fried parameter is  $r_0 = 0.1$  m, outer scale is  $L_0 = 25$  m, and the telescope aperture width is  $D = 4$  m. The wind speed is  $\omega = 0.25$  pixels per time step, which, to illustrate, corresponds to  $v \approx 8.2$  m/s assuming a sampling frequency of 500 Hz [see Eq. (21) for details]. The signal-to-noise ratio (SNR) of the slope measurements (of both the simulation and identification data sets) is set to 5 dB.

- As for the data-driven Kalman filtering parameters, the VAR model orders  $s$ , as seen in Eq. (23) and used in Eq. (24), are specified prior to each specific set of results, and  $p = 2$  always, as it was observed to produce the best results; the radial constraints applied to the VAR model of the slopes in Eq. (24) are  $r_1 = r_2 = 1.5$ ,  $r_3 = r_4 = 2$ ,  $r_5 = 2.5$ . In the low-resolution stage, the parameters are always set to ( $s' = 4$ ,  $p' = 2$ ).

- The turbulence dynamics are represented by a single translating phase screen, simulated using the Object-Oriented



MATLAB Adaptive Optics (OOMAO) toolbox [30]. The sensor signals were obtained simply via the application of Eq. (1) to the simulated turbulence, giving us direct control over the SNR of the slope measurements.

- The radius-based constraint of the sparse gain, whose width is denoted  $r_K$ , was here implemented as a square of lenslets around each pixel: a constraint of width zero is such that each pixel accepts only information from the four lenslets immediately adjacent to it; each unit increase in the constraint expands the square one lenslet outward. These implementation details are arbitrary, and another user can define the constraint otherwise.

- The matrix-vector multiplication (MVM) method used herein as a benchmark predicts the wavefront with a regularized least-squares estimator of  $\phi_k$  and a one-step progression via multiplication with  $A$ ,

$$\hat{\phi}_{k+1} = AC_{\phi,0}G^T(GC_{\phi,0}G^T + R)^{-1}y_k, \quad (36)$$

where the underbraced term is computed offline and multiplied with  $y_k$  online. Note that this assumes the covariance matrices of the measurement noise and incoming wavefront are available (and sufficiently well estimated), an assumption that is relaxed by our data-driven methods.

### A. Tuning the Model Order

The model order  $s$  is a vital parameter of our identification procedures. A larger  $s$  yields a VAR model with more time steps of information to smooth out measurement noise, but the added Markov parameters, each with equal or less sparsity than the previous, increase the identification time. Additionally, it has been observed that in the presence of erroneous modeling,  $s$  cannot be increased arbitrarily for better performance.

The influence of  $s$  in the NMSE [Eq. (35)] of both the identified KF and the HLRF is illustrated in Fig. 1. The best performance of the data-driven KF was attained for  $s = 4$ , with slight drops visible for larger  $s$ . As for the HLRF, the tighter the constraint applied to the gain, the sooner performance is maximized with respect to  $s$ ; for example, for  $(r_K = 3)$ , which is used in results below,  $s = 3$  yielded the best performance of the HLRF. Figure 2 shows how the time taken for *offline* identification of the gain (Kalman gain, in the case of 4.B) using the methods from Section 4 increases with  $m$  and  $s$ , and further compares it with that of the DARE.

### B. Performance and Prediction Times

Figure 3 presents a comparison between the NMSE obtained with the DARE-based KF, the MVM method, the identified KF (from Section 4.B), and the HLRF algorithm with varying parameters. Good relative performance of the HLRF is achieved early in the values of its tuning parameters, and both data-driven procedures can outperform the DARE-based KF due to the modeling error induced by a fractional  $\omega$ , as suggested in Section 2. The error-time trade-off is made evident with Fig. 4. Notice that, for instance, the  $(r_K = 3, L' = 10)$  combination yields a better NMSE than the DARE-based KF, while being

an order of magnitude faster than dense Kalman filtering and MVM.

### C. Spatial Frequency Domain Performance

As argued in Section 4, the sparse data-driven observer of the high-resolution stage of the HLRF fails to predict the low-spatial-frequency modes of the incoming wavefront as accurately as a dense data-driven KF does. This motivated the creation of the low-resolution stage and is illustrated in Fig. 5 alongside results for the DARE-based KF and MVM. Although the low frequencies are generally less present in the prediction errors than the high frequencies, the error induced by sparsity is sufficient to bring the NMSE [Eq. (35)] of the sparse observer slightly over that of MVM. Worse models further raise the importance of the low-resolution stage [27].

## 6. CONCLUSIONS AND RECOMMENDATIONS

This section summarizes the main conclusions and addresses the practical applicability of the developed data-driven KF by presenting several recommendations for future research.

### A. Conclusions

This paper has presented a new data-driven wavefront prediction method for real-time application in large-scale AO systems. By exploiting sparsity structures within the system matrices, we first developed a data-driven Kalman filtering method capable of, from measurement data, quickly identifying an approximate Kalman gain, but still in  $O(m^3)$  time.

The novel HLRF was further presented, which splits the wavefront prediction into a low- and a high-resolution stage. With the enforcement of sparsity in all matrices of the high-resolution stage, linearity with respect to  $m$  of both the identification (offline) and prediction (online) computational complexities was obtained, as the low-resolution stage is designed to be computationally light. This algorithm exchanges some of the performance of the KF for lower prediction times. Because the complexities are reduced to linear in  $m$  by shifting

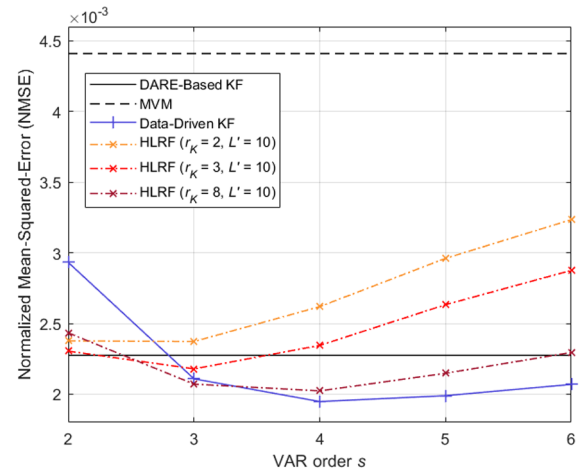
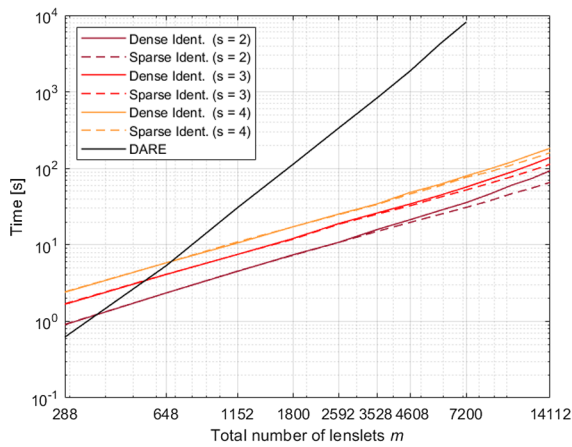
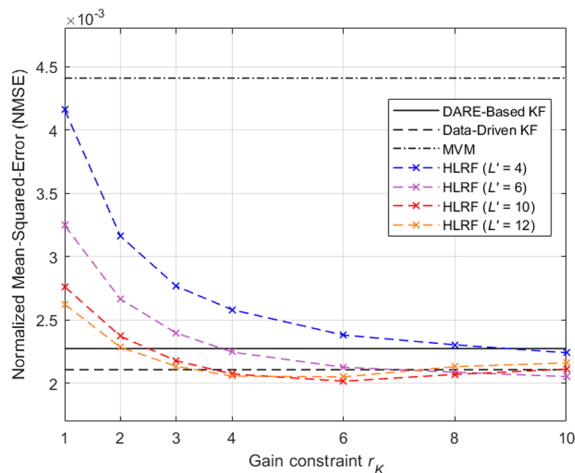


Fig. 1. NMSE [Eq. (35)] of the data-driven KF of Section 4.B (data-driven KF), the HLRF of Section 4.C, the KF obtained with MATLAB’s idare function (DARE-based KF), and MVM [Eq. (36)], for varying model order  $s$  in the identification scheme of Eq. (24).



**Fig. 2.** Average execution times of the *offline* gain identification procedures of Sections 4.B (Dense Ident.) and 4.C.1 (Sparse Ident.), and of MATLAB's idare function (DARE), as a function of the total number  $m$  of lenslets in the sensor array. A data batch length of 5000 was used to identify the gains.

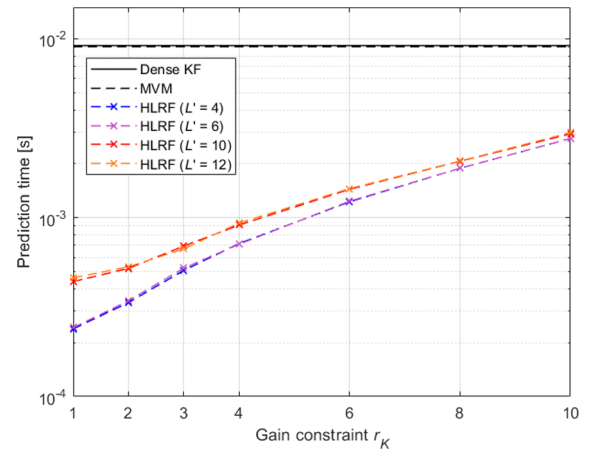


**Fig. 3.** NMSE [Eq. (35)] of the algorithms for varying constraint  $r_K$  of the sparse gain and the width  $L'$  of the low-resolution array. The model order  $s$  used in Eq. (24) was set to  $s = 3$ .

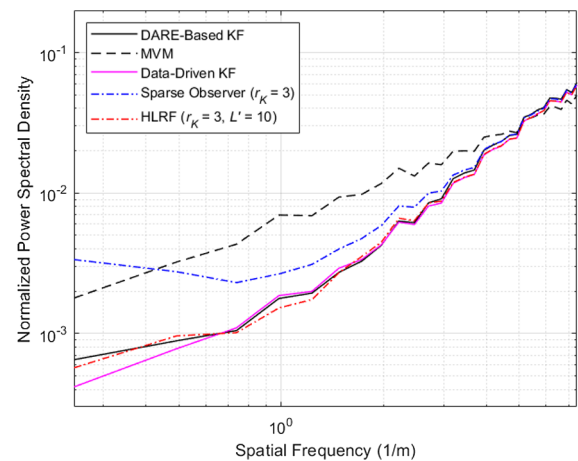
this dependence onto tuning parameters that should take the array size into consideration, it was shown in the example of a  $60 \times 60$  lenslet array that substantial improvements to prediction times demand only minimal sacrifices in performance compared to dense data-driven Kalman filtering.

## B. Applicability and Future Research

Although the results presented in Section 5 are based on simplified numerical simulations only, the HLRF of Section 4.C shows potential as an accurate real-time wavefront prediction method for large-scale AO systems. However, several issues still need to be addressed to make this method practically applicable. Note, however, that the radius-based sparsity of Sections 4.A and 4.B is by default applicable to nonsquare apertures, and that 4.B thus presents an already realistically applicable method for identification of the Kalman gain for AO systems.



**Fig. 4.** Average execution times of the *online* prediction operations of dense Kalman filtering [Eq. (7)], MVM [Eq. (36)], and of the HLRF [Eq. (29)], for varying HLRF parameters  $r_K$  and  $L'$ .



**Fig. 5.** Power spectral density (PSD) of each frequency of the prediction errors for each algorithm, normalized by the PSDs of the incoming uncorrected wavefront.

The main issues that remain to be addressed in the HLRF are the limitations imposed upon the (original) sensor array by the current formulation of the low-resolution array and the corresponding matrix  $\mathcal{H}$ . First, the algorithm is still to be extended to nonsquare lenslet arrays (e.g., corresponding to circular or annular apertures); second, the width of the original array currently must be divisible by that of the low-resolution array, which limits the choices of both widths.

The issue of algorithm extension to nonsquare arrays has been likewise reported in the literature of computationally efficient wavefront estimation methods; see, for example [15], where zero padding is proposed to artificially form a square data array from a nonsquare one. Zero padding can, however, lead to significant estimation errors. Since the low-dimensional array is merely a mathematical construct, future research could find alternative definitions of  $\mathcal{H}$  that enable additional freedom in the shape of the low-dimensional units, which could then accommodate nonsquare arrays and more choices of widths. Furthermore, the results shown herein explored only linear

interpolation from the low-dimensional array to the high-dimensional one. Interpolation methods that better replicate the shape of a wavefront should naturally be explored. The method has so far only been tested in open loop. Due to its low online computational time, the method could be applied in a closed-loop environment, where the predictions pertain to the *residual* wavefront, whose frequency content should differ from that of the uncorrected incoming wavefront. In order to properly assess its closed-loop performance, the algorithm should be paired with a control algorithm that uses its predictions and be tested in a more detailed simulation study, for example using OOMAO [30], or in a laboratory setup, and should be compared to existing fast data-driven optimal control methods such as [15,17].

## APPENDIX A: DETAILED SPARSE CONSTRUCTION OF $\mathcal{H}$

Recall that the low-resolution array consists of larger lenslets, here called *units*, that are  $L_u$  high-resolution lenslets wide. Consider the example of Fig. 6, with  $L = L_u$  and  $L' = 1$ , in order to isolate a single  $2 \times 2$  unit. Let a subscript  $x$  or  $y$  indicate the axis to which a particular slope corresponds. In this example,  $y_{x,k}^{(1)}$  and  $y_{y,k}^{(1)}$  are given by

$$\begin{cases} y_{x,k}^{(1)} = \frac{1}{2} (\phi_k^{(5)} + \phi_k^{(4)} - \phi_k^{(2)} - \phi_k^{(1)}) \\ y_{y,k}^{(1)} = \frac{1}{2} (\phi_k^{(5)} + \phi_k^{(2)} - \phi_k^{(4)} - \phi_k^{(1)}) \end{cases}, \quad (\text{A1})$$

and  $y_{x,k}^{(j)}$  and  $y_{y,k}^{(j)}$ ,  $j \in \{2, 3, 4\}$  are defined similarly. The slopes corresponding to the artificial low-resolution array should be defined likewise. As per Eq. (31),

$$\begin{cases} z_{x,k} = \frac{1}{2} (\phi_k^{(9)} + \phi_k^{(7)} - \phi_k^{(3)} - \phi_k^{(1)}) \\ z_{y,k} = \frac{1}{2} (\phi_k^{(9)} + \phi_k^{(3)} - \phi_k^{(7)} - \phi_k^{(1)}) \end{cases}. \quad (\text{A2})$$

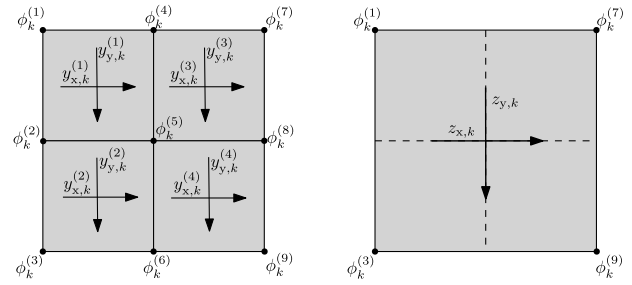
Construction of  $\mathcal{H}$  is based on this intuitive insight. Returning to generality, let  $\mathcal{S}_{y,j}$  denote the indices of  $y_k$  corresponding to the slopes, in the high-resolution array, measured by the lenslets within unit  $j$ ; let  $\mathcal{S}_{\phi,j}$  be analogous for the pixels within unit  $j$ . Denoting a selection of vector entries by a parenthesized superscript, the measurement equation for the lenslets within a single unit of arbitrary dimensions is given as follows:

$$y_k^{(\mathcal{S}_{y,j})} = G_u \phi_k^{(\mathcal{S}_{\phi,j})}, \quad (\text{A3})$$

where  $G_u$  is the same for every unit, and is built like  $G$ , but considering  $L_u$  as the array width. Analogously, let the pixel selection within each unit be given by  $Z_u$ , which merely selects the corner pixels of the unit. As per Eq. (A2), the measurement equation of unit  $j$ , now in the low-resolution array, is simply given by

$$\begin{bmatrix} z_{x,k}^{(j)} \\ z_{y,k}^{(j)} \end{bmatrix} = \underbrace{\begin{bmatrix} -0.5 & -0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \end{bmatrix}}_V Z_u \phi_k^{(\mathcal{S}_{\phi,j})}. \quad (\text{A4})$$

Defining  $\mathcal{H}_u$  as a combination analogous to  $\mathcal{H}$  in Eq. (30), but again within a single unit, it follows from Eqs. (A3) and (A4)



**Fig. 6.** Example of a high-resolution (left) and corresponding low-resolution (right) array for  $L = 2$  and  $L' = 1$ . Note that this pattern also represents a single *unit* of any larger grid with  $L_u = 2$ .

that

$$V Z_u \phi_k^{(\mathcal{S}_{\phi,j})} = \mathcal{H}_u G_u \phi_k^{(\mathcal{S}_{\phi,j})}. \quad (\text{A5})$$

For square units, matrix  $\mathcal{H}_u$  can be drawn from

$$\mathcal{H}_u = \underset{\mathcal{H}_u}{\operatorname{argmin}} \| \mathcal{H}_u G_u - V Z_u \|_F^2, \quad (\text{A6})$$

whose solutions are nonunique due to the presence of the piston and waffle modes in the null space of  $G_u$ . The minimum-norm solution

$$\mathcal{H}_u = V Z_u G_u^\dagger \quad (\text{A7})$$

is such that Eq. (A5) is exact for any arbitrary  $\phi_k^{(\mathcal{S}_{\phi,j})}$ . The entire combination matrix  $\mathcal{H}$  is now built from  $\mathcal{H}_u$ , which is done quickly via repeated insertion of  $\mathcal{H}_u$  into the entries of  $\mathcal{H}$  corresponding to each unit  $j \in \{1, 2, \dots, L'^2\}$ , fulfilling Eq. (33).

In truth, Eq. (A7) takes  $O(m^3)$  time for fixed  $L'$  and  $O(m^{3/2})$  time for  $L' \approx \sqrt{L}$ , but the time effectively taken to solve it is so insignificant regardless (using MATLAB and a sparse  $G_u$ ), even for unrealistically massive units of tens of thousands of lenslets themselves, that it is ignored throughout the paper.

**Funding.** H2020 European Research Council (339681); Seventh Framework Programme (2007-2013).

**Disclosures.** The authors declare no conflicts of interest.

**Data Availability.** Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

## REFERENCES

- B. Ellerbroek and T. Rhoardarmer, "Adaptive wavefront control algorithms for closed loop adaptive optics," *Math. Comput. Modell.* **33**, 145–158 (2001).
- B. L. Ellerbroek, "Efficient computation of minimum-variance wavefront reconstructors with sparse matrix techniques," *J. Opt. Soc. Am. A* **19**, 1803–1816 (2002).
- C. Kulcsár, H.-F. Raynaud, C. Petit, J.-M. Conan, and P. V. De Leseigno, "Optimal control, observers and integrators in adaptive optics," *Opt. Express* **14**, 7464–7476 (2006).
- B. Le Roux, J.-M. Conan, C. Kulcsár, H.-F. Raynaud, L. M. Mugnier, and T. Fusco, "Optimal control law for classical and multiconjugate adaptive optics," *J. Opt. Soc. Am. A* **21**, 1261–1276 (2004).
- G. Sivo, C. Kulcsár, J.-M. Conan, H.-F. Raynaud, É. Gendron, A. Basden, F. Vidal, T. Morris, S. Meimon, C. Petit, D. Gratadour, O.

- Martin, Z. Hubert, A. Sevin, D. Perret, F. Chemla, G. Rousset, N. Dipper, G. Talbot, E. Younger, R. Myers, D. Henry, S. Todd, D. Atkinson, C. Dickson, and A. Longmore, "First on-sky SCAO validation of full LQG control with vibration mitigation on the CANARY pathfinder," *Opt. Express* **22**, 23565–23591 (2014).
6. A. Beghi, A. Cenedese, and A. Masiero, "On the computation of Kalman gain in large adaptive optics systems," in *21st Mediterranean Conference on Control and Automation* (IEEE, 2013), pp. 1374–1379.
  7. C. Correia, J.-M. Conan, C. Kulcsár, H.-F. Raynaud, and C. Petit, "Adapting optimal LQG methods to ELT-sized AO systems," in *1st AO4ELT Conference-Adaptive Optics for Extremely Large Telescopes* (EDP Sciences, 2010), p. 07003.
  8. P. Massioni, H.-F. Raynaud, C. Kulcsár, and J.-M. Conan, "An approximation of the Riccati equation in large-scale systems with application to adaptive optics," *IEEE Trans. Control Syst. Technol.* **23**, 479–487 (2014).
  9. P. Massioni, L. Gilles, and B. Ellerbroek, "Adaptive distributed Kalman filtering with wind estimation for astronomical adaptive optics," *J. Opt. Soc. Am. A* **32**, 2353–2364 (2015).
  10. M. Gray, C. Petit, S. Rodionov, M. Bocquet, L. Bertino, M. Ferrari, and T. Fusco, "Local ensemble transform Kalman filter, a fast non-stationary control law for adaptive optics on ELTs: theoretical aspects and first simulation results," *Opt. Express* **22**, 20894–20913 (2014).
  11. R. Mehra, "Approaches to adaptive filtering," *IEEE Trans. Autom. Control* **17**, 693–698 (1972).
  12. J. Dunič, O. Straka, O. Kost, and J. Havlík, "Noise covariance matrices in state-space models: a survey and comparison of estimation methods. Part I," *Int. J. Adapt. Control Signal Process.* **31**, 1505–1543 (2017).
  13. K. Hinnen, M. Verhaegen, and N. Doelman, "A data-driven  $H_2$  - optimal control approach for adaptive optics," *IEEE Trans. Control Syst. Technol.* **16**, 381–395 (2008).
  14. B. Sinquin, L. Prengere, C. Kulcsár, H.-F. Raynaud, E. Gendron, J. Osborn, A. Basden, J.-M. Conan, N. Bharmal, L. Bardou, L. Staykov, T. Morris, T. Buey, F. Chemla, and M. Cohen, "On-sky results for adaptive optics control with data-driven models on low-order modes," *Mon. Not. R. Astron. Soc.* **498**, 3228–3240 (2020).
  15. B. Sinquin and M. Verhaegen, "Tensor-based predictive control for extremely large-scale single conjugate adaptive optics," *J. Opt. Soc. Am. A* **35**, 1612–1626 (2018).
  16. R. Fraanje, J. Rice, M. Verhaegen, and N. Doelman, "Fast reconstruction and prediction of frozen flow turbulence based on structured Kalman filtering," *J. Opt. Soc. Am. A* **27**, A235–A245 (2010).
  17. C. Yu and M. Verhaegen, "Structured modeling and control of adaptive optics systems," *IEEE Trans. Control Syst. Technol.* **26**, 664–674 (2017).
  18. P. Piscaer, "Sparse VARX model identification for large-scale adaptive optics," Master's thesis (Delft University of Technology, 2016).
  19. M. Rosensteiner, "Wavefront reconstruction for extremely large telescopes via CuRe with domain decomposition," *J. Opt. Soc. Am. A* **29**, 2328–2336 (2012).
  20. L. Gilles, C. R. Vogel, and B. L. Ellerbroek, "Multigrid preconditioned conjugate-gradient method for large-scale wave-front reconstruction," *J. Opt. Soc. Am. A* **19**, 1817–1822 (2002).
  21. D. L. Fried, "Adaptive optics wave function reconstruction and phase unwrapping when branch points are present," *Opt. Commun.* **200**, 43–72 (2001).
  22. J. Juang, C. Chen, and M. Phan, "Estimation of Kalman filter gain from output residuals," *J. Guid. Control Dyn.* **16**, 903–908 (1993).
  23. F. Roddier, *Adaptive Optics in Astronomy* (Cambridge University, 1999).
  24. M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach* (Cambridge University, 2007).
  25. G. I. Taylor, "The spectrum of turbulence," *Proc. R. Soc. London A* **164**, 476–490 (1938).
  26. F. Assémat, R. W. Wilson, and E. Gendron, "Method for simulating infinitely long and non stationary phase screens with optimized memory storage," *Opt. Express* **14**, 988–999 (2006).
  27. P. Cerqueira, "Data-driven filtering for large-scale adaptive optics," Master's thesis (Delft University of Technology, 2020).
  28. A. H. Jazwinski, "Adaptive filtering," *Automatica* **5**, 475–485 (1969).
  29. T. A. Davis, "Algorithm 915, SuiteSparseQR: multifrontal multi-threaded rank-revealing sparse QR factorization," *ACM Trans. Math. Softw.* **38**, 8 (2011).
  30. R. Conan and C. Correia, "Object-oriented Matlab adaptive optics toolbox," *Proc. SPIE* **9148**, 2066–2082 (2014).