Research Article Vol. 39, No. 3 / March 2022 / Journal of the Optical Society of America A 459

Journal of the Optical Society of America A
OPTICS, IMAGE SCIENCE, AND VISION

Dual-update data-driven control of deformable mirrors using Walsh basis functions

ALEKSANDAR HABER[1,*] AND THOMAS BIFANO[2]

[1]Department of Manufacturing and Mechanical Engineering Technology, College of Engineering Technology, Rochester Institute of Technology, 78 Lomb Memorial Drive, Rochester, New York 14623, USA
[2]Mechanical Engineering Department, Boston University, 8 Saint Mary's Street, Boston, Massachusetts 02215, USA
*Corresponding author: aleksandar.haber@gmail.com

Received 9 November 2021; revised 29 January 2022; accepted 29 January 2022; posted 31 January 2022; published 23 February 2022

We develop a novel data-driven method for deformable mirror (DM) control. The developed method updates both the DM model and DM control actions that produce desired mirror surface shapes. The novel method explicitly takes into account actuator constraints and couples a feedback-control algorithm with an algorithm for recursive estimation of DM influence function models. We also explore the possibility of using Walsh basis functions for DM control. By expressing the desired and observed mirror surface shapes as sums of Walsh pattern matrices, we formulate the control problem in the 2D Walsh basis domain. We thoroughly experimentally verify the developed approach on a 140-actuator MEMS DM, developed by Boston Micromachines. Our results show that the novel method produces the root-mean-square surface error in the 14–40 nanometer range. These results can additionally be improved by tuning the control and estimation parameters. The developed approach is also applicable to other DM types such as segmented DMs.

https://doi.org/10.1364/JOSAA.447879

1. INTRODUCTION

Deformable mirrors (DMs) are one of the main components of adaptive optics (AO) systems [1,2]. A typical DM consist of a reflective optical surface that is deformed by a set of actuators. By precisely shaping the surface of a DM, we can compensate for wavefront aberrations in AO systems [3–16].

In this paper, we consider the problem of developing control algorithms for DMs. There are a large number of approaches for DM control. A complete survey of all the methods goes well beyond the length limits of this paper. Consequently, we briefly mention only the most relevant or recent approaches. Most of the DM control approaches are based on the following control paradigm [17–21]. First, a linearized DM model (influence function matrix) is estimated before a correction process. Then, during this process, such a model is either directly inverted or is used in conjunction with an iterative feedback control algorithm to calculate the DM control actions. During the control iterations, only DM control actions (control voltages) are updated on the basis of the observed wavefront aberrations, and the DM model is kept constant. The control approaches that rely upon this control idea might not be able to produce satisfactory wavefront correction performance in at least two scenarios that can be encountered in practice. The first scenario is when the DM behavior changes over time. For example, an environment in which a DM operates might induce temperature fluctuations that can create thermo-mechanical deformations of mirror components as well as other effects that can significantly alter the DM behavior [10,12,13,22–29]. This is especially the case for optical systems operating in space and for optical systems operating with high-power laser sources, where the absorbed heat increases the temperature of optical components. Also, DM temperature can be increased by the heat generated by electrical components used to control the actuators. The second scenario is when DM control actions significantly deviate from points around which a linear DM model is estimated. Significant deviations of control actions from original linearization points excite DM nonlinearities. Namely, most of the existing DM devices and prototypes exhibit nonlinear behavior for sufficiently large magnitudes of control actions that are necessary to correct for wavefront aberrations with significant peak-to-valleys (PVs).

It is a well-known fact in control theory literature that control methods that dynamically update model parameters on the basis of available observations are able to cope with time-varying dynamics, model uncertainties, and system nonlinearities. These methods are referred to as adaptive control methods [30,31]. Even in the case of linear systems, adaptive control methods are more accurate, faster, and, in many cases, more optimal than classical linear feedback algorithms. All these facts motivate the development of an adaptive DM control method in this paper. However, since the word "adaptive" has different meanings in optics and control theory communities, in this paper, we name the developed algorithm as the "dual-update" control algorithm, in order not to confuse an interested reader.

Under the term "dual-update" control algorithm, we understand a control algorithm that is able to update both control actions and a model (influence matrix) during DM operation.

To the best of our knowledge, adaptive control approaches for DM control have received less attention and interest compared with the classical optimal control approaches [16,32]. Thus, the approach proposed in [33] iteratively calibrates a DM model during the correction process. The approach developed in [34] uses a recursive least-squares method to dynamically estimate a DM influence matrix during the correction process. The main limitation of the approaches developed in [33,34] is that they do not explicitly take into account actuator saturation and actuation limits. A reliable control method has to take into account actuator physical constraints and has to be able to deal with actuator saturation. Recently, in [35] we have developed a DM control approach that updates the DM model on the basis of batches of observed wavefront data. This method takes into account actuator saturation. Once the DM model is updated, the correction process is performed in open loop. This method is relatively slow since between the correction steps it requires a relatively large number of wavefront samples to update the model. Apart from these methods, adaptive filters have been used in [36] to predict and control wavefront disturbances.

A widely used approach for DM control is to express the observed wavefront using a Zernike modal basis. In this way, we implicitly perform the spatial model-order reduction of the DM control problem (the control problem is transformed from the spatial domain to the Zernike basis domain). On the other hand, for certain mirror types and for desired wavefront shapes consisting of steep peaks and valleys, Walsh basis functions might be a more suitable option than the Zernike basis functions [37–39]. Apart from this, in the general case, since Walsh basis functions are orthogonal, they can be used instead of Zernike polynomials in classical adaptive optics applications.

In this paper, we develop a novel dual-update DM control approach. The developed approach explicitly takes into account actuator constraints and couples a feedback control algorithm with a recursive estimation of the DM influence matrix. In this way, we are able to dynamically update the DM model and at the same time compute control actions that produce the desired shape. Furthermore, by expressing the desired and observed mirror surface shapes as sums of Walsh pattern matrices, we formulate the control problem in the 2D Walsh basis domain. We experimentally verify the developed approach on a 140 actuator MEMS DM, developed by Boston Micromachines.

The main contributions of this paper are summarized in the sequel. In contrast to other adaptive DM control approaches proposed in the literature, our approach explicitly takes into account actuation limits and saturation. In this way, we can avoid the loss of performance that happens if the actuator control commands are in the saturation range. In addition, we thoroughly experimentally investigate the performance of Walsh basis functions for DM control. This is important since only a handful of articles have explored the possibility of using Walsh basis functions for wavefront reconstruction and control [37–39], and the true potential of using Walsh basis functions for DM control is largely unexplored. It should be emphasized here that, although we have performed experiments on a continuous face sheet DM, the developed control approach is applicable to other DM types. The developed approach is especially suitable for segmented DMs. Finally, we can easily modify the developed control algorithm to use Zernike basis functions instead of the Walsh basis functions.

This paper is organized as follows. In Section 2, we present the procedure for approximating the mirror surface shape as a sum of Walsh pattern matrices. In Section 3, we present the control method. In Section 4, we present the experimental results. In Section 5, we present conclusions and briefly discuss future work.

2. MIRROR DEFORMATION REPRESENTATION USING WALSH PATTERN MATRICES

In this section, we present a simple numerical procedure for approximating the mirror surface shape as a sum of Walsh pattern matrices.

DM surface deformation is usually represented by a matrix. That is, every entry of this matrix is a DM surface deformation at a fixed spatial location. We refer to this matrix as the *mirror deformation matrix*. Let $W \in \mathbb{R}^{n\times n}$ be the mirror deformation matrix, where n is the total size (measured in pixels) along the x and y dimensions of the observed mirror surface. We decompose this matrix as follows:

$$W \approx \sum_{p=1}^{M}\sum_{q=1}^{M} a_{p,q} Z_{p,q}, \tag{1}$$

where $a_{p,q} \in \mathbb{R}$ are coefficients, and $Z_{p,q} \in \mathbb{R}^{n\times n}$ are Walsh pattern matrices with the entries that can either be -1 or 1. The number n should be selected such that $n = 2^V$, where V is a user-selected positive integer. The total number of the Walsh pattern matrices in Eq. (1) is equal to M^2, where $M \leq n$. We form the Walsh pattern matrices by using Walsh basis functions. Here, it should be emphasized that, since this paper presents a proof of concept and due to mathematical simplicity and brevity, we use Walsh basis functions defined over a square domain. The mirror surface shape can also be represented by using Walsh basis functions defined over a circular domain (polar Walsh basis functions) [40]. Consequently, the developed method can easily be used in the case of circular correction domains. However, even without using polar Walsh basis functions, with some modifications, the approach based on square-domain Walsh basis functions can be used for wavefront correction over circular correction domains. This is experimentally demonstrated in Section 4. On the other hand, the control approach developed in Section 3 is practically independent of the type of basis functions for expanding the mirror surface shape. Thus, instead of using Walsh basis functions, we can also use Zernike basis functions in the developed control approach.

In the sequel, we first introduce a procedure for constructing the Walsh pattern matrices $Z_{p,q}$. Then, we introduce a procedure for computing the coefficients $a_{p,q}$. First, we choose the constant V. We select the constant V such that a deformation submatrix with the dimensions of 2^V by 2^V pixels is within the limits of the maximal active mirror surface area that is observable by the used sensor (for more details about the sensor used in our experiments, see Section 4). In our case, we use $V = 8$ or $V = 9$,

giving us the submatrices with the dimensions of 256 by 256, and 512 by 512, respectively (see Section 4 for more details).

Let the entries of the vector $\gamma_i^{(V)} \in \mathbb{R}^n$, $n = 2^V$, represent the values of the Walsh function i of the order V. Here, the index i takes the values from 1 to 2^V. For example, for $V = 2$, the vectors $\gamma_i^{(2)}$, $i = 1, 2, 3, 4$, representing the Walsh basis functions take the following forms:

$$\gamma_1^{(2)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \gamma_2^{(2)} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \quad \gamma_3^{(2)} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix},$$

$$\gamma_4^{(2)} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}. \tag{2}$$

The vectors $\gamma_i^{(V)}$ can easily be constructed by using the MATLAB function hadamard(·). The rows of a matrix returned by this function represent Walsh basis functions. However, the Walsh basis functions that are represented by the rows of this matrix are not arranged in increasing order. Consequently, the rows of the matrix returned by the function hadamard(·) need to be permuted. For more details, see the MATLAB tutorial page [41] explaining the construction process of the Walsh basis functions.

Once the vectors $\gamma_i^{(V)}$ are constructed, for selected V, we use the following procedure to construct the Walsh pattern matrices. First, we need to select the constant M, keeping in mind several competing factors. Generally speaking, we have to make a trade-off between representation accuracy that is increased by increasing M and dimensions of the matrices of the control algorithm that increase with the factor of M^2. Larger matrix dimensions increase the computational and memory complexities of the decomposition process as well as the computational and memory complexities of the control and estimation algorithms that are introduced in Section 3. We have tested the control algorithm for M up to 120, and this value is sufficient for the desired shapes used in our experiments. Our lab computer has 64 GB RAM, and for $M \geq 120$, the MATLAB programming language that is used to control the DM runs out of memory. Larger values of M are possible if the control algorithm is implemented in a more memory-efficient way. One of the possible pathways to decrease the computational and memory complexities is to exploit the structure of the control matrices using approaches similar to the approaches developed in [11,42–50]. However, this requires additional research, implementation, and testing efforts that are left for future research.

Once we have selected V and M, we can proceed with the construction of the pattern matrices. We perform the following steps:

Step 1. We construct the matrices $Z_{1,q} \in \mathbb{R}^{n \times n}$, where $q = 1, 2, \ldots, M$, and $n = 2^V$. The matrix $Z_{1,q}$ is constructed by transposing the vector $\gamma_q^{(V)}$ and by stacking the newly formed row vectors on top of each other n times:

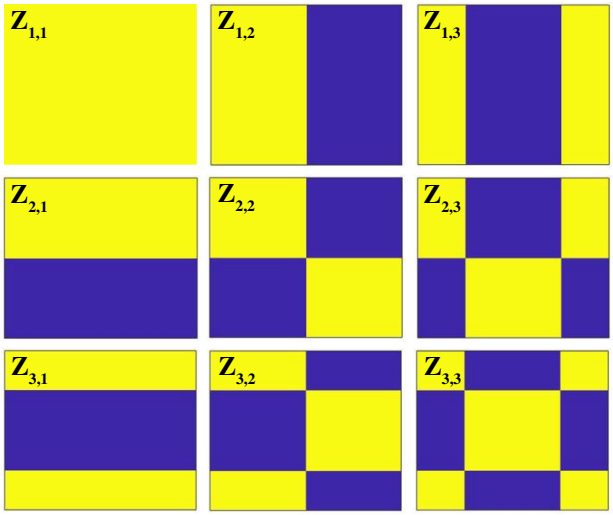

Fig. 1. 2D Walsh pattern matrices. The yellow and blue colors correspond to 1 and −1 values, respectively.

$$Z_{1,q} = \begin{bmatrix} (\gamma_q^{(V)})^T \\ (\gamma_q^{(V)})^T \\ \vdots \\ (\gamma_q^{(V)})^T \end{bmatrix}. \tag{3}$$

Step 2. We construct the matrices $Z_{q,1}$, where $q = 1, 2, \ldots, M$. The matrices $Z_{q,1}$ are constructed by transposing the matrices $Z_{1,q}$ that are formed in Step 1, that is, $Z_{q,1} = Z_{1,q}^T$.

Step 3. We construct the matrices $Z_{p,q}$, for the indices $p \geq 2$ and $q \geq 2$. The matrix $Z_{p,q}$ is calculated as follows:

$$Z_{p,q} = Z_{p,1} \odot Z_{1,q}, \tag{4}$$

where $\odot$ is the matrix element-wise product (Hadamard matrix product).

A few comments about this construction procedure are in order. The matrix $Z_{p,q}$ can be seen as a (p, q) block of a large block matrix. The first block row of this matrix is constructed in Step 1. In Step 2, we construct the first block column, where every block is a transpose of the corresponding matrix in the first block row. Then, in Step 3, we construct the remaining block matrices by simply multiplying the matrices (element-wise multiplication) belonging to the first block column with the matrices belonging to the first block row. The constructed 2D Walsh pattern matrices are shown in Fig. 1 for $M = 3$.

Next, we explain the decomposition process of the matrix W, that is, we explain how to compute the coefficients $a_{p,q}$ in Eq. (1) for known W. First, we vectorize Eq. (1). The vectorization process is done by introducing the vectorization operator vec(·) [51]. Let $X \in \mathbb{R}^{n \times n}$ be an arbitrary matrix with the column vectors denoted by $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^n$. The vectorization operator produces a vector $\mathbf{x} \in \mathbb{R}^{n^2}$ obtained by stacking column vectors $\mathbf{x}_i \in \mathbb{R}^n$ of the matrix X on top of each other. By applying the vectorization operator to Eq. (1), we obtain

$$\mathrm{vec}(W) \approx \sum_{p=1}^{M} \sum_{q=1}^{M} a_{p,q} \mathrm{vec}(Z_{p,q}), \tag{5}$$

$$\mathbf{w} = \sum_{p=1}^{M}\sum_{q=1}^{M} a_{p,q}\mathbf{z}_{p,q}, \tag{6}$$

where $\mathbf{w} = \mathrm{vec}(W)$, $\mathbf{w} \in \mathbb{R}^{n^2}$, and $\mathbf{z}_{p,q} = \mathrm{vec}(Z_{p,q})$, $\mathbf{z}_{p,q} \in \mathbb{R}^{n^2}$. On the other hand, due to the fact that the pattern matrices $Z_{p,q}$ are formed on the basis of the Walsh functions, we have that

$$\mathbf{z}_{l,s}^T\mathbf{z}_{p,q} = \begin{cases} n^2, & l = p \wedge s = q \\ 0, & l \neq p \vee s \neq q \end{cases}. \tag{7}$$

The property in Eq. (7) comes from the fact that the Walsh basis functions form a complete orthogonal set of functions. This property can be used to retrieve the coefficients $a_{p,q}$. Namely, from Eqs. (6) and (7), we have

$$\frac{1}{n^2}\mathbf{z}_{p,q}^T\mathbf{w} = a_{p,q}. \tag{8}$$

The expression in Eq. (8) enables us to construct a projection matrix that produces the coefficients of the expansion in Eq. (1). Let the matrices $\Psi_i \in \mathbb{R}^{M\times n^2}$ and $\Pi \in \mathbb{R}^{M^2\times n^2}$ and the vectors $\mathbf{a}_i \in \mathbb{R}^M$ and $\mathbf{a} \in \mathbb{R}^{M^2}$ be defined by

$$\Pi = \frac{1}{n^2}\begin{bmatrix}\Psi_1\\ \Psi_2\\ \vdots\\ \Psi_M\end{bmatrix}, \quad \Psi_i = \begin{bmatrix}\mathbf{z}_{1,i}^T\\ \mathbf{z}_{2,i}^T\\ \vdots\\ \mathbf{z}_{M,i}^T\end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix}\mathbf{a}_1\\ \mathbf{a}_2\\ \vdots\\ \mathbf{a}_M\end{bmatrix},$$

$$\mathbf{a}_i = \begin{bmatrix}a_{1,i}\\ a_{2,i}\\ \vdots\\ a_{M,i}\end{bmatrix}. \tag{9}$$

The vector $\mathbf{a}$ groups all the coefficients of the expansion in Eq. (1). Then, using this construction, from Eq. (8) we have

$$\mathbf{a} = \Pi\mathbf{w}. \tag{10}$$

The expression in Eq. (10) shows that, to decompose the deformation matrix W as a sum of the Walsh pattern matrices, we just need to vectorize this matrix and multiply the result with the projection matrix Π. That is, to compute the Walsh decomposition coefficients $a_{p,q}$, we need to perform a single vector-matrix multiplication.

3. CONTROL METHOD

In this section, we present the control method. The basic idea of the control method is to update both the DM control actions and the DM influence matrix. To develop the control method, we use the Walsh basis function expansion presented in Section 2.

Let $W_D \in \mathbb{R}^{n\times n}$ be a desired mirror shape that we want to produce, and let $\mathbf{w}_D = \mathrm{vec}(W_D)$, $\mathbf{w}_D \in \mathbb{R}^{n^2}$. Using Eq. (10), we compute the desired set of coefficients as follows:

$$\mathbf{a}_D = \Pi\mathbf{w}_D, \tag{11}$$

where Π is the projection matrix introduced in Eq. (9) and $\mathbf{a}_D \in \mathbb{R}^{M^2}$ is the vector grouping the desired coefficients.

We send control actions to the DM or observe its deformation response at discrete-time instants, denoted by $k \in \mathbb{N}_0$. Let $\mathbf{w}_k \in \mathbb{R}^{n^2}$ be the observed mirror deformation in the vectorized form, that is $\mathbf{w}_k = \mathrm{vec}(W_k)$, where $W_k \in \mathbb{R}^{n\times n}$ is the observed mirror deformation matrix at the discrete-time instant k. By using the projection given in Eq. (10), we have

$$\mathbf{a}_k = \Pi\mathbf{w}_k, \tag{12}$$

where $\mathbf{a}_k \in \mathbb{R}^{M^2}$ are the coefficients. We postulate the following DM model:

$$\mathbf{a}_{k+1} = Q_k\mathbf{g}_k + \mathbf{d}_{k+1}, \tag{13}$$

where $Q_k \in \mathbb{R}^{M^2\times r}$ is the influence matrix, and the vector $\mathbf{g}_k \in \mathbb{R}^r$ is defined by

$$\mathbf{g}_k = \begin{bmatrix} u_{1,k}^{\beta} & u_{2,k}^{\beta} & \dots & u_{r,k}^{\beta}\end{bmatrix}^T, \tag{14}$$

and where $u_{i,k}$, $i = 1, 2, \dots, r$, is the control input applied to the ith actuator at the time instant k, r is the number of DM actuators (in our case $r = 140$), and $\beta = 1.742$ is an estimate of the constant of the exponential dependence between DM control actions and the observed deformation response. This estimate is obtained by using the least-squares approach explained in [35]. For the sequel, we group the control inputs $u_{i,k}$ in the vector $\mathbf{u}_k \in \mathbb{R}^r$. The values of the control input $u_{i,k}$ are in the interval [0, 1], with zero corresponding to no control action and 1 corresponding to the maximal control action applied to the actuator i. The vector $\mathbf{d}_{k+1} \in \mathbb{R}^{M^2}$, that is not known *a priori*, takes into account the measurement noise and unmodeled mirror behavior that are not captured by the model $Q_k\mathbf{g}_k$.

To initialize the control algorithm, we need to obtain an initial estimate of the influence matrix and we also need to compute initial values of control inputs. In the sequel, we explain how to generate these initial values.

A. Initial Estimation of the Influence Matrix and Control Actions

For a positive integer $S \geq r$ (S should be larger than the number of DM actuators), we introduce the following notation for batch data matrices:

$$A_{1:S} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_S\end{bmatrix}, \quad G_{0:S-1} = \begin{bmatrix}\mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_{S-1}\end{bmatrix}, \tag{15}$$

where $A_{1:S} \in \mathbb{R}^{M^2\times S}$ and $G_{0:S-1} \in \mathbb{R}^{r\times S}$. To compute the initial values, we will assume that the influence matrix is constant. Under this assumption, from Eq. (13), we obtain

$$\mathbf{a}_{k+1} = Q^{(0)}\mathbf{g}_k + \mathbf{d}_{k+1}, \tag{16}$$

where $Q^{(0)} \in \mathbb{R}^{M^2\times r}$ is the initial value of the influence matrix Q_k, and this equation is valid for time steps $k = 0, 1, 2, \dots, S-1$. We use the approach presented in [35] to estimate $Q^{(0)}$. First, we generate random input signals $\mathbf{u}_k$, for $k = 0, 1, \dots, S-1$. The entries of the input vector

$\mathbf{u}_k$ are generated from a normal distribution with the mean of 0.5 and the variance of 0.15. If any of the generated entries is larger (smaller) than the upper bound 1 (lower bound 0), then the value of this entry is set to 1 (0). By grouping expressions in Eq. (16) for $k = 0, 1, 2, \ldots, S-1$, we obtain a batch matrix equation. From this equation, we can estimate $Q^{(0)}$ by solving the following multivariable least-squares problem:

$$\min_{Q^{(0)}} \left\| A_{1:S} - Q^{(0)} G_{0:S-1} \right\|_F^2, \tag{17}$$

where $\|\cdot\|_F$ is the Frobenius norm. The solution is given by

$$\hat{Q}^{(0)} = A_{1:S} G_{0:S-1}^T (G_{0:S-1} G_{0:S-1}^T)^{-1}. \tag{18}$$

From Eq. (18), we see the justification of the condition $S \geq r$. Namely, the number of data samples S should satisfy the following condition: $S \geq r$ (r is the number of DM actuators), to ensure that the matrix $G_{0:S-1} G_{0:S-1}^T$ in the solution given by Eq. (18) is invertible (the columns of $G_{0:S-1}$ are linearly independent since the control inputs are randomly selected) and, consequently, that the solution is well-defined. Once the initial value of the influence matrix is determined, we determine the initial control actions by using the following strategy.

The goal of the control algorithm is to produce the desired mirror surface shape, represented by the deformation matrix $W_D \in \mathbb{R}^{n \times n}$. We decompose this matrix to obtain the vector $\mathbf{a}_D \in \mathbb{R}^{M^2}$ consisting of the desired Walsh coefficients. Then, by substituting $Q^{(0)}$ in Eq. (16) by its estimate given by Eq. (18), we can find the initial control actions by solving the following optimization problem:

$$\min_{\mathbf{g}_0} \left\| \mathbf{a}_D - \hat{Q}^{(0)} \mathbf{g}^{(0)} \right\|_2^2, \tag{19}$$

$$\text{subject to: } \mathbf{0} \leq \mathbf{g}^{(0)} \leq \mathbf{1}, \tag{20}$$

where the less-than-equal relation operator $\leq$ is applied element-wise. Let the solution of this optimization problem be denoted by $\hat{\mathbf{g}}^{(0)}$. If we would simply minimize the cost function given by Eq. (19) with respect to $\mathbf{g}^{(0)}$, then most likely the computed control actions would violate the lower limit 0 and upper limit 1 on the control actions. Consequently, the constraint given by Eq. (20) is introduced in order to ensure that the computed control actions are physically realizable. We solve the problem given by Eqs. (19) and (20) by using the MATLAB function lsqlin(). Once the solution of this problem is determined, we can obtain the entries of the initial control input vector $\hat{\mathbf{u}}^{(0)}$ from the entries of $\hat{\mathbf{g}}^{(0)}$. The estimated influence matrix $\hat{Q}^{(0)}$ and the vector $\hat{\mathbf{g}}^{(0)}$ are used to initialize the adaptive control method developed in the sequel.

B. Control Algorithm Development

In the interest of deriving the control algorithm, we have to introduce one simplification related to the unknown vector $\mathbf{d}_k$. To develop the control algorithm, we assume that the vector $\mathbf{d}_k$ does not depend on the control index k, that is

$$\mathbf{d}_k = \mathbf{d} = \text{const}. \tag{21}$$

In experiments, this condition might not hold exactly due to the measurement noise and disturbances. However, this assumption is necessary in order to keep the derivations as simple as possible. As our experiments show, the developed algorithm works well despite the fact that in experiments there might be deviations from the assumption stated in Eq. (21). Even without this assumption, it is possible to derive the control algorithm. However, the mathematical apparatus will become more complex and will involve expectation operators. Taking into account this assumption and the general model given by Eq. (13), we obtain

$$\mathbf{a}_{k+1} = Q_k \mathbf{g}_k + \mathbf{d}. \tag{22}$$

To develop the control algorithm, we define the control error $\varepsilon_k \in \mathbb{R}^{M^2}$ as follows:

$$\varepsilon_k := \mathbf{a}_D - \mathbf{a}_k. \tag{23}$$

By shifting the time index in Eq. (23) and by combining the resulting equation with Eq. (22), we obtain

$$\varepsilon_{k+1} = \mathbf{a}_D - Q_k \mathbf{g}_k - \mathbf{d}. \tag{24}$$

On the other hand, for the time step k, we have

$$\varepsilon_k = \mathbf{a}_D - Q_{k-1} \mathbf{g}_{k-1} - \mathbf{d}. \tag{25}$$

From Eqs. (24) and (25), we have

$$\varepsilon_{k+1} - \varepsilon_k = Q_{k-1} \mathbf{g}_{k-1} - Q_k \mathbf{g}_k, \tag{26}$$

$$\varepsilon_{k+1} = \varepsilon_k + Q_{k-1} \mathbf{g}_{k-1} - Q_k \mathbf{g}_k, \tag{27}$$

$$\varepsilon_{k+1} = \gamma_k - Q_k \mathbf{g}_k, \tag{28}$$

$$\gamma_k = \varepsilon_k + Q_{k-1} \mathbf{g}_{k-1}. \tag{29}$$

At the discrete-time instant k, we can observe the mirror surface deformation and construct the vector $\mathbf{a}_k$. This means that, at the time instant k, the vector ε_k given by Eq. (23) can be constructed. Let us assume that, at the time instants $k-1$ and k, we have computed the estimates of Q_{k-1} and Q_k, which are denoted by $\hat{Q}_{k-1}$ and $\hat{Q}_k$, respectively. The procedure for estimating these influence matrices will be explained later in the text. Also, let us assume that, at the time instant $k-1$, we have computed the vector $\mathbf{g}_{k-1}$ (the vector that is a function of the control actions). Let the computed value of $\mathbf{g}_{k-1}$ be denoted by $\hat{\mathbf{g}}_{k-1}$. Then, by substituting Q_{k-1}, Q_k, and $\mathbf{g}_{k-1}$, by $\hat{Q}_{k-1}$, $\hat{Q}_k$, and $\hat{\mathbf{g}}_{k-1}$ in Eqs. (28) and (29), we define

$$\bar{\varepsilon}_{k+1} = \hat{\gamma}_k - \hat{Q}_k \mathbf{g}_k, \tag{30}$$

$$\hat{\gamma}_k = \varepsilon_k + \hat{Q}_{k-1} \hat{\mathbf{g}}_{k-1}. \tag{31}$$

Our goal is to compute the control actions for the next time step $k+1$. That is, our goal is to compute the vector $\mathbf{g}_k$. First, we introduce the cost function

$$\bar{\varepsilon}_{k+1}^T \bar{\varepsilon}_{k+1} = (\hat{\gamma}_k - \hat{Q}_k \mathbf{g}_k)^T (\hat{\gamma}_k - \hat{Q}_k \mathbf{g}_k). \tag{32}$$

We compute the control inputs by solving the following constrained optimization problem:

$$\min_{\mathbf{g}_k} (\hat{\gamma}_k - \hat{Q}_k \mathbf{g}_k)^T (\hat{\gamma}_k - \hat{Q}_k \mathbf{g}_k), \tag{33}$$

$$\text{subject to } \mathbf{0} \le \mathbf{g}_k \le \mathbf{1}. \tag{34}$$

We solve this problem by using the MATLAB function lsqlin( ). The solution of this optimization problem is denoted by $\hat{\mathbf{g}}_k$.

C. Dual-Update Control Algorithm

To formulate and solve the optimization problem in Eqs. (33) and (34), we need to compute the estimates of the influence matrices $\hat{Q}_{k-1}$ and $\hat{Q}_k$. The vector $\hat{\mathbf{g}}_{k-1}$ is computed by shifting the time index to $k-1$ in Eqs. (33) and (34). We use a recursive identification approach [52] to compute $\hat{Q}_{k-1}$ and $\hat{Q}_k$. To define the recursive identification approach, we need to define the following quantities. By applying the vec(·) operator to Eq. (22), we obtain

$$\mathbf{a}_{k+1} = G_k \mathbf{q}_k + \mathbf{d}, \tag{35}$$

$$G_k = \mathbf{g}_k^T \otimes I, \tag{36}$$

$$\mathbf{q}_k = \text{vec}(Q_k), \tag{37}$$

where $\mathbf{q}_k \in \mathbb{R}^{rM^2}$ is the vector that parametrizes the influence matrix, $G_k \in \mathbb{R}^{M^2 \times rM^2}$, and we have used the following property of the vec(·) operator [46]: $\text{vec}(X_1 X_2 X_3) = (X_3^T \otimes X_1)\text{vec}(X_2)$, for arbitrary matrices X_1, X_2 and X_3. We also define the following quantities related to G_k and $\mathbf{q}_k$:

$$\hat{G}_k = \hat{\mathbf{g}}_k^T \otimes I, \tag{38}$$

$$\hat{\mathbf{q}}_k = \text{vec}(\hat{Q}_k). \tag{39}$$

In the sequel, we formulate the dual-update control algorithm. This algorithm is initialized for $k=1$ with $\hat{\mathbf{q}}_0 = \text{vec}(\hat{Q}^{(0)})$, where $\hat{Q}^{(0)}$ is defined in Eq. (18), and with $\hat{\mathbf{g}}_0 = \hat{\mathbf{g}}^{(0)}$, where $\hat{\mathbf{g}}^{(0)}$ is the solution of the optimization problem defined in Eqs. (19) and (20). To implement the algorithm, we also need an additional matrix $P_k \in \mathbb{R}^{rM^2 \times rM^2}$, which is updated during the control iterations. The initial value of this matrix, for $k=0$, is $P_0 = \delta I$, where δ is a parameter selected by the user. The dual-update control algorithm consists of the following steps that are performed for $k = 1, 2, 3, \ldots$

Step 1. Observe the mirror surface deformation and compute the vector $\mathbf{a}_k$ (Walsh coefficients) by using Eq. (12). On the basis of the computed value $\hat{\mathbf{g}}_{k-1}$, from the previous step $k-1$, form the matrix $\hat{G}_{k-1}$ by using Eq. (38). The following values are available from the previous time step $k-1$: P_{k-1} and $\hat{\mathbf{q}}_{k-1}$. Update the matrix P_k and the vector of influence matrix parameters $\hat{\mathbf{q}}_k$ by performing the following steps:

$$S_{k-1} = (\lambda I + \hat{G}_{k-1} P_{k-1} \hat{G}_{k-1}^T)^{-1}, \tag{40}$$

$$L_k = P_{k-1} \hat{G}_{k-1}^T S_{k-1}, \tag{41}$$

$$P_k = \frac{1}{\lambda} P_{k-1} - \frac{1}{\lambda} L_k \hat{G}_{k-1} P_{k-1}, \tag{42}$$

$$\mathbf{e}_k = \mathbf{a}_k - \hat{G}_{k-1} \hat{\mathbf{q}}_{k-1}, \tag{43}$$

$$\hat{\mathbf{q}}_k = \hat{\mathbf{q}}_{k-1} + L_k \mathbf{e}_k, \tag{44}$$

where $0 < \lambda \le 1$ is a user selected parameter, $\mathbf{e}_k \in \mathbb{R}^{M^2}$, $S_{k-1} \in \mathbb{R}^{M^2 \times M^2}$, and $L_k \in \mathbb{R}^{rM^2 \times M^2}$ is the gain matrix.

Step 2. Form the influence matrix estimates $\hat{Q}_{k-1}$ and $\hat{Q}_k$ by using the computed values $\hat{\mathbf{q}}_{k-1}$ and $\hat{\mathbf{q}}_k$, respectively, by inverting the vectorization operator. Compute ε_k given by Eq. (23) and $\hat{\gamma}_k$ given by Eq. (31). Form and solve the optimization problem defined in Eqs. (33) and (34), by using the MATLAB function lsqlin( ). The solution is given by $\hat{\mathbf{g}}_k$. Using this value, compute the control actions $\hat{\mathbf{u}}_k$. Apply the control actions to the DM, wait for the time step $k+1$, and go to Step 1, where now the time index is shifted to $k+1$.

Several comments about the developed algorithm are in order. The vector $\mathbf{e}_k$ defined in Eq. (43) is called the model error. This vector quantifies the difference between the observed modal response $\mathbf{a}_k$ and the model prediction, given by $\hat{G}_{k-1}\hat{\mathbf{q}}_{k-1}$. Following the guidelines given in ([52], p. 379) and ([31], p. 68), we use $\lambda = 0.98$. However, selecting λ offers other possibilities [31,53]. The matrix P_k is initialized with the parameter $\delta = 0.05$, that is, $P_0 = 0.05I$. Here, we have used a scaled identity matrix (sparse matrix) to initialize P_k, in order to minimize the computational burden. This is necessary since the matrices in Eqs. (40)–(44) are large dimensional. Namely, initialization of P_k as a dense matrix will significantly increase the computational burden. The issue of decreasing the computational complexity of the proposed control algorithm is a future research topic.

4. EXPERIMENTAL RESULTS

In this section, we present the experimental results.

We test the developed approach by using a Boston Micromachines MEMS DM with $r = 140$ actuators. The actuation grid is 12 by 12 with 4 corner actuators that are not active. The DM stroke is about 2 [μm]; the pitch is 400 [μm]. The behavior of this DM type has been analyzed in a number of manuscripts, see, for example, [54,55] and follow-up works. Consequently, due to paper brevity, we do not further summarize other mirror properties.

The produced mirror surface shape is sensed by a partitioned aperture wavefront (PAW) sensor [56–58]. This sensor has a large dynamic range, it is relatively fast, and it operates with uncollimated light sources. It has a relatively high resolution that is only limited by the used camera. In addition, this sensor is speckle-free, robust, and polarization-independent. Further details related to this sensor can be found in [57,58]. The used experimental setup is the same as the experimental setup used to generate the results in our previous publication

[28]. Consequently, in the interest of brevity, we only describe its main components. The light source is an LED (660 [nm], Thorlabs). A system of optical components is used to direct and shape the beam and illuminate the DM surface. A monochrome camera (Blackfly BFS-U3-123S6M-C) is used as a PAW sensing element. The maximal size of the observed image (deformation matrix) is 1001 by 999. The DM and sensor are controlled by using the MATLAB programming anguage.

As explained in Section 2, to decompose the observed deformation as a sum of Walsh pattern matrices, the deformation matrix size should be expressed as a power of 2. On the other hand, the camera of the PAW sensor produces a deformation image size of 1001 by 999 pixels. This image covers an area that is larger than the active DM area. Taking all these things into consideration, we have at least two options for selecting the deformation matrix size. The first option is a 256 by 256 matrix, and the second option is a 512 by 512 matrix. The second option covers the centers of all the actuators (including 4 corner inactive actuators). However, in the second option, a part of deformation caused by the edge actuators will not take part in the defined 512 by 512 deformation matrix. We investigate the performance of the developed method for both options.

A. Results for the 256 by 256 Deformation Matrix

First, we present the control results for the deformation matrix represented by a 256 by 256 image (256 by 256 deformation matrix). This corresponds to $n = 256 = 2^8$, that is, to $V = 8$ (for more details, see Section 2). We generate a "raw" desired mirror surface shape as

$$W_D^{\text{raw}} = a_{1,1}Z_{1,1} + a_{2,2}Z_{2,2} + a_{3,3}Z_{3,3}, \tag{45}$$

with the coefficients of the expansion equal to $a_{1,1} = -1$, $a_{2,2} = -0.1$, and $a_{3,3} = 0.2$. The raw desired surface shape is shown in Fig. 2(a). This desired surface contains vertical (90 degrees) surface changes between the segments of the regular checkerboard pattern shown in Fig. 2(a). On the other hand, the deformation response of a single actuator is a smooth function resembling the Gaussian function. Consequently, the mirror actuators are not able to produce vertical 90-degree deformation changes from one segment to another. Due to this, we need to apply a smoothing low-pass 2D filter to the desired raw surface shape. We choose a Gaussian 2D filter, as shown in Fig. 2(b). The standard deviation of the filter is 30. After applying the Gaussian 2D filter, we offset the resulting deformation by -1 (from every entry of the matrix we subtract -1). This process produces the desired surface shape shown in Fig. 2(c). We compute the decomposition given by the Eq. (1) for $M = 80$. This produces a total of 6400 coefficients, which are shown in Fig. 2(d). These are the desired coefficients we want to produce.

Next, we compute the control actions by using the developed approach. Figure 3 shows the control results. Panels (a) and (b) in Fig. 3 show the desired and best-produced shapes, respectively. Panel (c) in Fig. 3 shows the error (difference between the desired and produced shapes). The root-mean-square (RMS) surface error is 14 [nm]. Finally, Fig. 3(d) shows the desired and best-produced coefficients of the 2D Walsh basis expansion defined in Eq. (1).

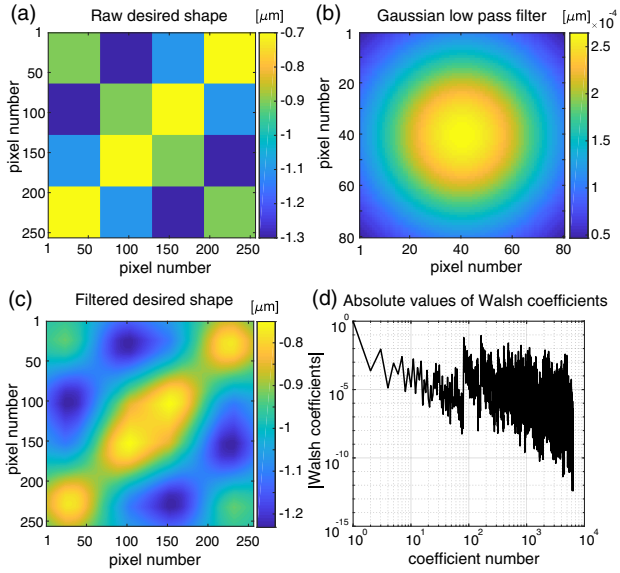

Fig. 2. (a) "Raw" desired shape defined in Eq. (45). (b) Gaussian 2D low-pass filter applied to the "raw" desired shape. (c) Desired shape after applying the filter and offset. (d) Coefficients of the decomposition in Eq. (1) obtained by decomposing the filtered desired shape.

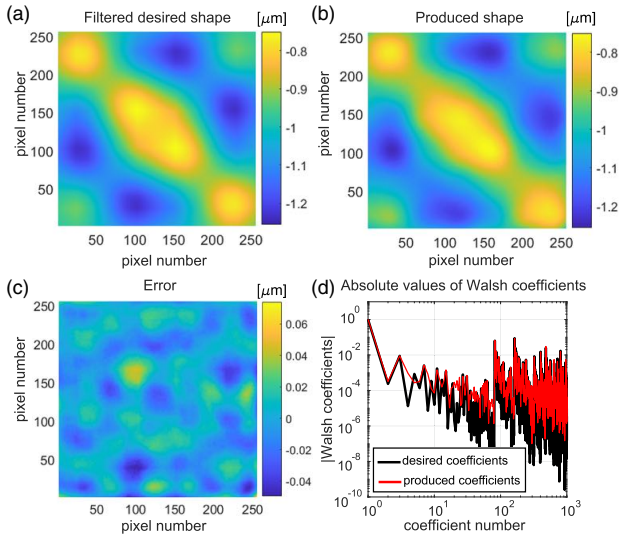

Fig. 3. (a) Desired filtered mirror shape. (b) Best produced mirror shape. (c) Error. (d) Desired and produced coefficients.

Figure 4 shows the surface cross section generated at two horizontal cut planes. Figure 5 shows the convergence of the control error ε_{k+1} defined in Eq. (23) and the model error $\mathbf{e}_{k+1}$ defined in Eq. (43).

The presented results demonstrate the excellent performance of the developed method. The RMS surface error converges to 14.1 [nm] in a relatively small number of iterations. Furthermore, Fig. 4 shows that the method outperforms the least-squares approach for controlling the DM. In our simulations, the least-squares approach is used to generate the initial guess. However, this approach is used in a number of articles as a standalone method to control the DM. That is, the dual-update

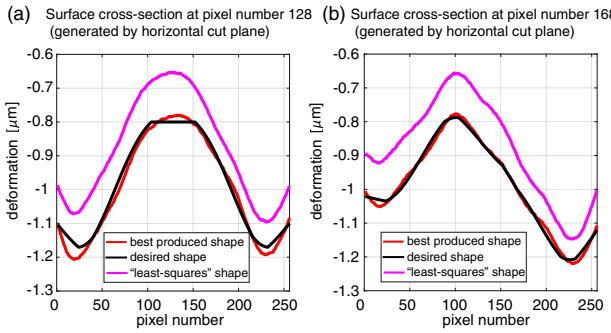

Fig. 4. Surface cross sections at the horizontal cut planes generated at the pixels (a) 128 and (b) 168. The "least-squares" shape corresponds to the initial surface shape produced by the initial control actions generated by solving the optimization problem in Eq. (19) and (20).

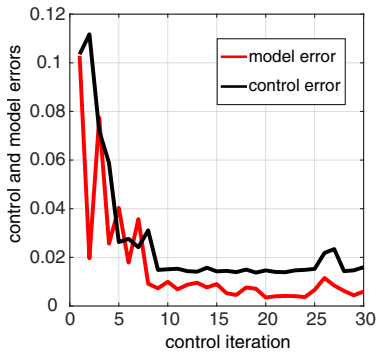

Fig. 5. Convergence of the control method. The graph shows the two norms of the control error ε_{k+1} and model error $\mathbf{e}_{k+1}$ defined in Eqs. (23) and (43), respectively.

control method proposed in this article clearly has a significant advantage over the classical least-squares approaches for DM control.

B. Results for the 512 by 512 Deformation Matrix

Here, we present the results for the observed deformation matrix with the dimension of 512 by 512 pixels. In this case $n = 2^V$, where $V = 9$ (for more details, see Section 2). By defining the deformation matrix in this way, we are able to investigate the influence of the actuation boundaries on the performance of the developed algorithm. We test the following raw desired shapes with spatial frequencies from smaller to larger:

$$W_{D1} = -0.3Z_{1,1} + 0.3Z_{6,6}, \tag{46}$$

$$W_{D2} = -0.3Z_{1,1} + 0.3Z_{7,7}, \tag{47}$$

$$W_{D3} = -0.3Z_{1,1} + 0.3Z_{10,10}. \tag{48}$$

We apply the Gaussian low-pass filter to the desired shapes. The filter has the support of 70 pixels and deviation of 25. Once the filter is applied, on offset of -1 is applied to the filtered shapes to produce the final desired shapes. The final desired shapes are shown in Panel (a) of Figs. 6, 8, and 10. Panel (b) in Figs. 6, 8, and 10 shows the best-produced shapes. Panel (c) in Figs. 6, 8, and 10 shows the surface errors. Panel (d) in Figs. 6, 8, and 10 shows the surface errors over the central mirror part (obtained by cropping the surface error by 90 pixels on all four image sides).

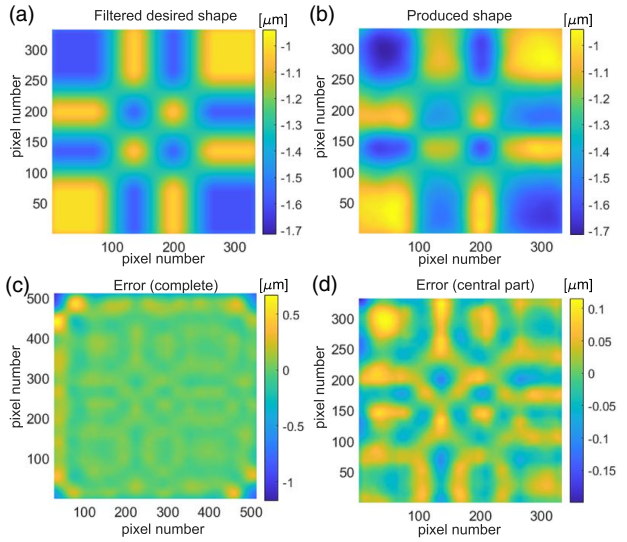

Fig. 6. Control results for the raw desired shape defined in Eq. (46) [note that this shape is filtered and scaled to generate panel (a)]. (a) Filtered and scaled the desired shape. (b) Best produced shape. (c) Surface error. (d) Surface error for the central mirror part [obtained by cropping the surface error shown in panel (c) by 90 pixels on all four image sides].

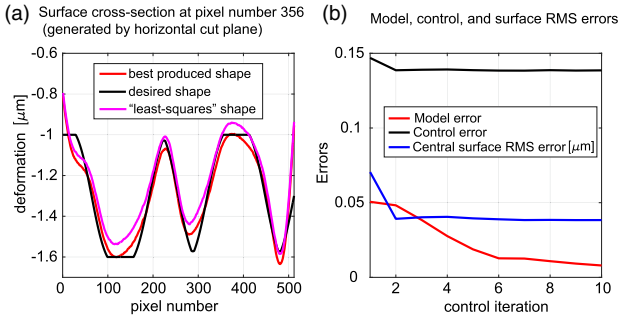

Fig. 7. Control results for the raw desired shape defined in Eq. (46) [note that this shape is filtered and scaled to generate panel (a) in Fig. 6]. (a) Cross-sections of the produced and desired shapes generated by horizontal cut planes. The "least-squares" shape corresponds to the initial surface shape produced by the initial control actions generated by solving the optimization problem in Eqs. (19) and (20). (b) Convergence of the model, control, and RMS surface errors of the developed algorithm. Panel (b) shows the 2-norms of the control error ε_{k+1} and model error $\mathbf{e}_{k+1}$ defined in Eqs. (23) and (43), respectively.

Panel (a) in Figs. 7, 9, and 11 shows the cross-sections of the desired and produced shapes generated by horizontal cut planes. Panels (b) in Figs. 7, 9, and 11 show the convergence of control, model, and RMS surface errors. The above-presented results demonstrate the excellent performance of the developed method. From Panel (b) in Figs. 7, 9, and 11, we can observe that the RMS surface error converges to approximately 40 [nm] in a relatively small number of iterations. These results can additionally be improved by tuning the parameters (parameter λ and

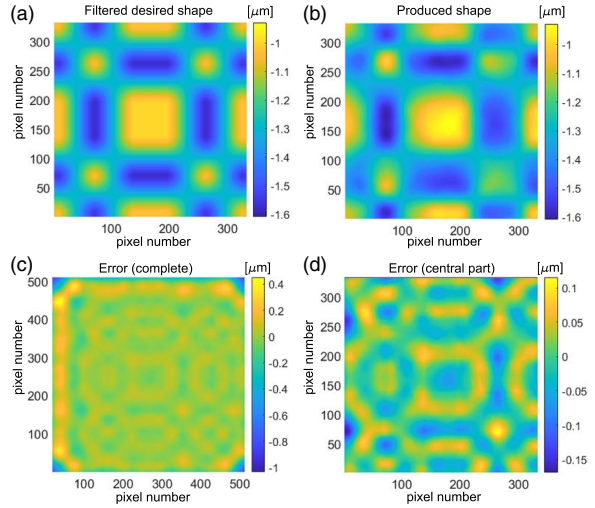

Fig. 8. Control result for the raw desired shape defined in Eq. (47) [note that this shape is filtered and scaled to generate panel (a)]. Captions of panels in this figure correspond to the captions of panels in Fig. 6.

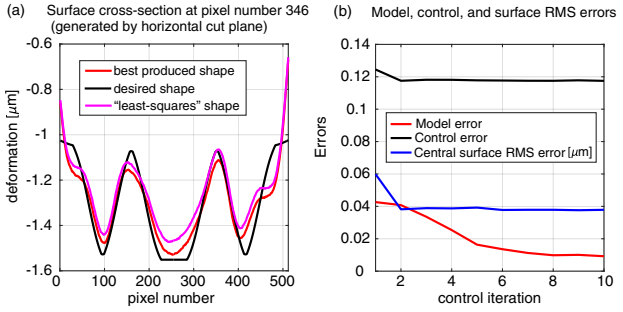

Fig. 9. Control result for the raw desired shape defined in Eq. (47) [note that this shape is filtered and scaled to generate panel (a) in Fig. 8]. Captions of panels in this figure correspond to the captions of panels in Fig. 7.

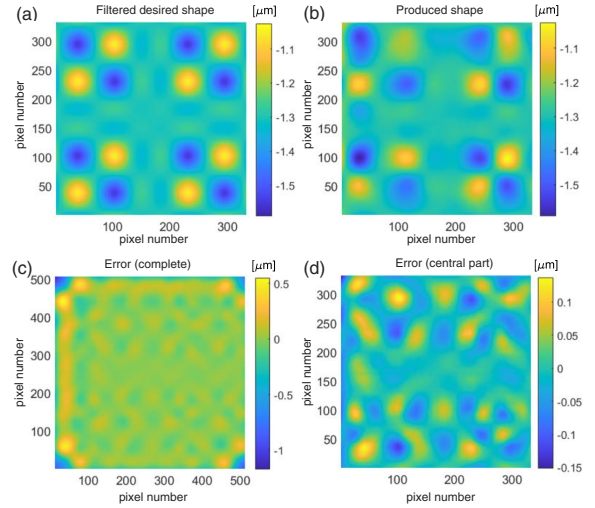

Fig. 10. Control result for the raw desired shape defined in Eq. (48) [note that this shape is filtered and scaled to generate panel (a)]. Captions of panels in this figure correspond to the captions of panels in Fig. 6.

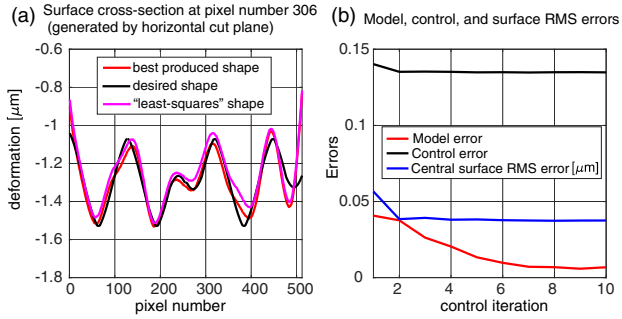

Fig. 11. Control result for the raw desired shape defined in Eq. (48) [note that this shape is filtered and scaled to generate panel (a) in Fig. 10]. Captions of panels in this figure correspond to the captions of panels in Fig. 7.

the matrix P) of the algorithm. The development of methods for optimal tuning of the dual-update algorithm is a future research topic. Furthermore, by comparing the least-squares shapes with the best produced shapes in Panel (a) of the same figures, we can observe that our method has significant advantages over the state-of-the-art least-squares approaches for DM control.

C. Generation of a Zernike Desired Shape

Here, we demonstrate the performance of the developed algorithm for generating the desired shape equal to a Zernike coefficient. We select the desired shape to be equal to an offset and scaled version of Z_6^0. The desired shape is shown in Fig. 12(a). We use the following parameters: $\lambda = 0.98$ and $P = 0.05 \cdot I$. The PV value of the desired shape is 1.7 [μm].

Figures 12 and 13 demonstrate good performance of the developed method. We can observe a central surface RMS error of 21.6 [nm].

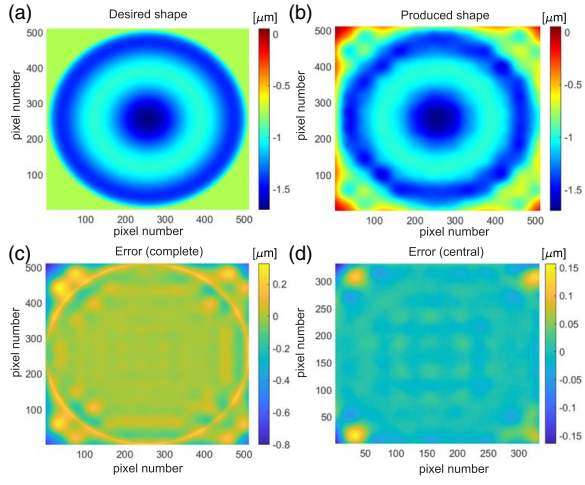

Fig. 12. Control result for the desired shape equal to an offset and scaled version of Z_6^0. (a) Desired shape. (b) Best produced shape. (c) Surface error. (d) Surface error of the central mirror part (obtained by cropping the surface error shown in panel (c) by 90 pixels on all four image sides).

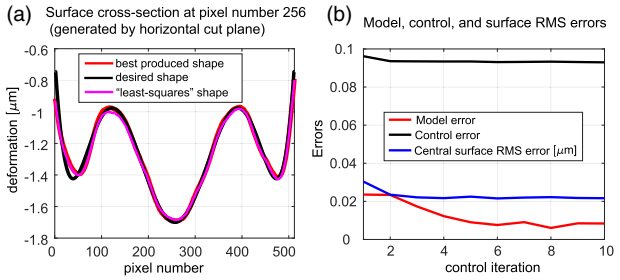

Fig. 13. Control results for the desired shape equal to an offset and scaled version of Z_6^0. Captions of panels in this figure correspond to the captions of panels in Fig. 7.

5. CONCLUSION

Here, we have developed a novel approach for adaptive control of DMs. On the basis of the feedback information provided by the sensor, our method updates the DM influence matrix and the control actions. We have tested the developed algorithm by using a Boston Micromachines DM with 140 actuators. We are able to generate RMS surface errors in the interval of 14–40 [nm]. These experimental results can be further improved by tuning the algorithm parameters. Besides introducing a novel control algorithm, we have also demonstrated the potential of using Walsh basis functions for DM control. Walsh basis functions can also be used as an effective method for the control of segmented DMs. Our approach can straightforwardly be applied to other DM types. In future work, we will focus on improving the performance of the developed approach by optimally tuning the control algorithm parameters and on reducing the computational complexity of the developed approach.

Funding. National Science Foundation (EEC-1647837).

Disclosures. T. B. acknowledges a financial interest in Boston Micromachines Corporation, which produces the commercially available DMs used for the experiments reported in this paper.

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

1. R. Tyson, *Principles of Adaptive Optics* (CRC, 2015).
2. F. Roddier, *Adaptive Optics In Astronomy* (Cambridge, 1999).
3. M. Manetti, M. Morandini, P. Mantegazza, R. Biasi, M. Andrighettoni, and D. Gallieni, "VLT DSM, the control system of the largest deformable secondary mirror ever manufactured," Proc. SPIE **9148**, 91484G (2014).
4. G. Vdovin and P. M. Sarro, "Flexible mirror micromachined in silicon," Appl. Opt. **34**, 2968–2972 (1995).
5. S. K. Ravensbergen, P. C. J. N. Rosielle, and M. Steinbuch, "Deformable mirrors with thermo-mechanical actuators for extreme ultraviolet lithography: design, realization and validation," Precis. Eng. **37**, 353–363 (2013).
6. P.-Y. Madec, "Overview of deformable mirror technologies for adaptive optics and astronomy," Proc. SPIE **8447**, 844705 (2012).
7. A. Polo, A. Haber, S. F. Pereira, M. Verhaegen, and H. P. Urbach, "Linear phase retrieval for real-time adaptive optics," J. Eur. Opt. Soc. Rap. Publ. **8**, 13070 (2013).
8. M. Horenstein, T. Bifano, S. Pappas, J. Perreault, and R. Krishnamoorthy-Mali, "Real time optical correction using electrostatically actuated MEMS devices," J. Electrost. **46**, 91–101 (1999).
9. S. Kuiper, N. Doelman, J. Human, R. Saathof, W. Klop, and M. Maniscalco, "Advances of TNO's electromagnetic deformable mirror development," Proc. SPIE **10706**, 1070619 (2018).
10. A. Haber, A. Polo, S. Ravensbergen, H. P. Urbach, and M. Verhaegen, "Identification of a dynamical model of a thermally actuated deformable mirror," Opt. Lett. **38**, 3061–3064 (2013).
11. A. Haber and M. Verhaegen, "Framework to trade optimality for local processing in large-scale wavefront reconstruction problems," Opt. Lett. **41**, 5162–5165 (2016).
12. A. Haber, A. Polo, I. Maj, S. F. Pereira, H. P. Urbach, and M. Verhaegen, "Predictive control of thermally induced wavefront aberrations," Opt. Express **21**, 21530–21541 (2013).
13. R. Saathof, G. J. M. Schutten, J. W. Spronck, and R. H. M. Schmidt, "Actuation profiles to form Zernike shapes with a thermal active mirror," Opt. Lett. **40**, 205–208 (2015).
14. A. Haber and M. Verhaegen, "Modeling and state-space identification of deformable mirrors," Opt. Express **28**, 4726–4740 (2020).
15. A. Chiuso, R. Muradore, and E. Marchetti, "Dynamic calibration of adaptive optics systems: a system identification approach," IEEE Trans. Control Syst. Technol. **18**, 705–713 (2009).
16. J. Mocci, M. Quintavalla, A. Chiuso, S. Bonora, and R. Muradore, "PI-shaped LQG control design for adaptive optics systems," Control Eng. Pract. **102**, 104528 (2020).
17. A. Haber, A. Polo, C. S. Smith, S. F. Pereira, P. Urbach, and M. Verhaegen, "Iterative learning control of a membrane deformable mirror for optimal wavefront correction," Appl. Opt. **52**, 2363–2373 (2013).
18. E. Fernandez and P. Artal, "Membrane deformable mirror for adaptive optics: performance limits in visual optics," Opt. Express **11**, 1056–1069 (2003).
19. C. Vogel, G. Tyler, Y. Lu, T. Bifano, R. Conan, and C. Blain, "Modeling and parameter estimation for point-actuated continuous-facesheet deformable mirrors," J. Opt. Soc. Am. A **27**, A56–A63 (2010).
20. C. R. Vogel, G. A. Tyler, and Y. Lu, "Modeling, parameter estimation, and open-loop control of MEMS deformable mirrors," Proc. SPIE **7595**, 75950E (2010).
21. A. Polo, A. Haber, S. F. Pereira, M. Verhaegen, and H. P. Urbach, "An innovative and efficient method to control the shape of push-pull membrane deformable mirror," Opt. Express **20**, 27922–27932 (2012).
22. N. Gu, C. Li, Y. Cheng, and C. Rao, "Thermal control for light-weighted primary mirrors of large ground-based solar telescopes," J. Astron. Telesc. Instrum. Syst. **5**, 014005 (2019).
23. C. Blaurock, M. McGinnis, K. Kim, and G. E. Mosier, "Structural-thermal-optical performance (STOP) sensitivity analysis for the James Webb Space Telescope," Proc. SPIE **5867**, 58670V (2005).
24. C. Buleri, M. Kehoe, C. Lukashin, T. Jackson, J. Beckman, A. Curtis, B. Edwards, T. Owen, A. Phenis, and M. Stebbins, "Structural, thermal, and optical performance (STOP) analysis of the NASA ARCSTONE instruments," Proc. SPIE **10925**, 1092503 (2019).
25. Q. Xue, L. Huang, P. Yan, M. Gong, Z. Feng, Y. Qiu, T. Li, and G. Jin, "Research on the particular temperature-induced surface shape of a national ignition facility deformable mirror," Appl. Opt. **52**, 280–287 (2013).
26. M. Kasprzack, B. Canuel, F. Cavalier, R. Day, E. Genin, J. Marque, D. Sentenac, and G. Vajente, "Performance of a thermally deformable mirror for correction of low-order aberrations in laser beams," Appl. Opt. **52**, 2909–2916 (2013).
27. A. Haber, J. E. Draganov, K. Heesh, J. Tesch, and M. Krainak, "Modeling and system identification of transient STOP models of optical systems," Opt. Express **28**, 39250–39265 (2020).
28. A. Haber, J. E. Draganov, K. Heesh, J. Cadena, and M. Krainak, "Modeling, experimental validation, and model order reduction of mirror thermal dynamics," Opt. Express **29**, 24508–24524 (2021).
29. M. Habets, J. Scholten, S. Weiland, and W. Coene, "Multi-mirror adaptive optics for control of thermally induced aberrations in extreme ultraviolet lithography," Proc. SPIE **9776**, 97762D (2016).
30. K. J. Åström and B. Wittenmark, *Adaptive Control* (Courier, 2013).
31. I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive Control: Algorithms, Analysis and Applications* (Springer, 2011).

32. C. Kulcsár, H.-F. Raynaud, C. Petit, and J.-M. Conan, "Minimum variance prediction and control for adaptive optics," Automatica **48**, 1939–1954 (2012).
33. W. Zou and S. A. Burns, "High-accuracy wavefront control for retinal imaging with adaptive-influence-matrix adaptive optics," Opt. Express **17**, 20167–20177 (2009).
34. L. Huang, X. Ma, Q. Bian, T. Li, C. Zhou, and M. Gong, "High-precision system identification method for a deformable mirror in wavefront control," Appl. Opt. **54**, 4313–4317 (2015).
35. A. Haber and T. Bifano, "General approach to precise deformable mirror control," Opt. Express **29**, 33741–33759 (2021).
36. J. Tesch, S. Gibson, and M. Verhaegen, "Receding-horizon adaptive control of aero-optical wavefronts," Opt. Eng. **52**, 071406 (2013).
37. F. Wang, "Utility transforms of optical fields employing deformable mirror," Opt. Lett. **36**, 4383–4385 (2011).
38. F. Wang, "Wavefront sensing through measurements of binary aberration modes," Appl. Opt. **48**, 2865–2870 (2009).
39. F. Wang, "High-contrast imaging via modal convergence of deformable mirror," Astrophys. J. **751**, 83 (2012).
40. L. N. Hazra and A. Guha, "Far-field diffraction properties of radial Walsh filters," J. Opt. Soc. Am. A **3**, 843–846 (1986).
41. MATLAB, "Discrete Walsh-Hadamard transform tutorial," 2021, https://www.mathworks.com/help/signal/ug/discrete-walsh-hadamard-transform.html.
42. P. Massioni, C. Kulcsár, H.-F. Raynaud, and J.-M. Conan, "Fast computation of an optimal controller for large-scale adaptive optics," J. Opt. Soc. Am. A **28**, 2298–2309 (2011).
43. 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 (2015).
44. A. Haber and M. Verhaegen, "Moving horizon estimation for large-scale interconnected systems," IEEE Trans. Autom. Control **58**, 2834–2847 (2013).
45. A. Haber and M. Verhaegen, "Sparsity preserving optimal control of discretized PDE systems," Comput. Method Appl. Math. Eng. **335**, 610–630 (2018).
46. A. Haber and M. Verhaegen, "Sparse solution of the Lyapunov equation for large-scale interconnected systems," Automatica **73**, 256–268 (2016).
47. A. Haber and M. Verhaegen, "Subspace identification of large-scale interconnected systems," IEEE Trans. Autom. Control **59**, 2754–2759 (2014).
48. 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).
49. G. Monchen, B. Sinquin, and M. Verhaegen, "Recursive Kronecker-based vector autoregressive identification for large-scale adaptive optics," IEEE Trans. Control Syst. Technol. **27**, 1677–1684 (2018).
50. P. Cerqueira, P. Piscaer, and M. Verhaegen, "Sparse data-driven wavefront prediction for large-scale adaptive optics," J. Opt. Soc. Am. A **38**, 992–1002 (2021).
51. M. Verhaegen and V. Verdult, *Filtering and System Identification: a Least Squares Approach* (Cambridge, 2007).
52. L. Ljung, *System Identification: Theory for the User* (Prentice Hall, 1999).
53. L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification* (MIT, 1983).
54. A. Diouf, A. P. Legendre, J. B. Stewart, T. G. Bifano, and Y. Lu, "Open-loop shape control for continuous microelectromechanical system deformable mirror," Appl. Opt. **49**, G148–G154 (2010).
55. J. B. Stewart, A. Diouf, Y. Zhou, and T. G. Bifano, "Open-loop control of a MEMS deformable mirror for large-amplitude wavefront control," J. Opt. Soc. Am. A **24**, 3827–3833 (2007).
56. R. Barankov and J. Mertz, "Single-exposure surface profilometry using partitioned aperture wavefront imaging," Opt. Lett. **38**, 3961–3964 (2013).
57. A. B. Parthasarathy, K. K. Chu, T. N. Ford, and J. Mertz, "Quantitative phase imaging using a partitioned detection aperture," Opt. Lett. **37**, 4062–4064 (2012).
58. J. Li, D. R. Beaulieu, H. Paudel, R. Barankov, T. G. Bifano, and J. Mertz, "Conjugate adaptive optics in widefield microscopy with an extended-source wavefront sensor," Optica **2**, 682–688 (2015).