Determinant-Based Fast Greedy Sensor Selection Algorithm

YUJI SAITO [ID][1], **TAKU NONOMURA** [ID][1], **KEIGO YAMADA** [ID][1], **KUMI NAKAI** [ID][1], **TAKAYUKI NAGATA** [ID][1], **KEISUKE ASAI** [1], **YASUO SASAKI** [2], **AND DAISUKE TSUBAKINO** [2], **(Member, IEEE)**

[1] Department of Aerospace Engineering, Tohoku University, Sendai 980-8579, Japan
[2] Department of Aerospace Engineering, Nagoya University, Nagoya 464-8603, Japan

Corresponding author: Yuji Saito (yuji.saito@tohoku.ac.jp)

ABSTRACT In this paper, the sparse sensor placement problem for least-squares estimation is considered, and the previous novel approach of the sparse sensor selection algorithm is extended. The maximization of the determinant of the matrix which appears in pseudo-inverse matrix operations is employed as an objective function of the problem in the present extended approach. The procedure for the maximization of the determinant of the corresponding matrix is proved to be mathematically the same as that of the previously proposed QR method when the number of sensors is less than that of state variables (undersampling). On the other hand, the authors have developed a new algorithm for when the number of sensors is greater than that of state variables (oversampling). Then, a unified formulation of the two algorithms is derived, and the lower bound of the objective function given by this algorithm is shown using the monotone submodularity of the objective function. The effectiveness of the proposed algorithm on the problem using real datasets is demonstrated by comparing with the results of other algorithms. The numerical results show that the proposed algorithm improves the estimation error by approximately 10% compared with the conventional methods in the oversampling case, where the estimation error is defined as the ratio of the difference between the reconstructed data and the full observation data to the full observation. For the NOAA-SST sensor problem, which has more than ten thousand sensor candidate points, the proposed algorithm selects the sensor positions in few seconds, which required several hours with the other algorithms in the oversampling case on a 3.40 GHz computer.

INDEX TERMS Optimization, sparse sensor selection, greedy algorithms.

I. INTRODUCTION

Reduced-order modeling for fluid analysis has been gathering a lot of attention. With regards to reduced-order modeling, a proper orthogonal decomposition (POD) [1], [2] is one of the effective methods for decomposing high-dimensional data into several significant modes of flow fields. Here, POD means a data-driven method which gives the most significant and relevant structure in the data and exactly corresponds to principal component analysis and Karhunen-Loève (KL) decomposition, where the decomposed modes are orthogonal to each other. The POD analysis for a discrete data matrix can be carried out by applying singular value decomposition, as is often the case in the engineering fields. There are several

The associate editor coordinating the review of this manuscript and approving it for publication was Anton Kos [ID].

advanced data-driven methods: dynamic mode decomposition [3], [4]; empirical mode decomposition; and others that include efforts by the authors [5], [6]. This research is only based on the POD which is the most basic data-driven method for reduced-order modeling. If the data, such as flow fields, can be effectively expressed by a limited number of POD modes, limited sensors placed at appropriate positions will give the approximated full state information. Such effective observation might be one of the keys for flow control and flow prediction. This idea has been adopted by Manohar *et al.* [7], and the sparse-sensor-placement algorithm has been developed and discussed. The idea is expressed by the following equation:

$$y = HUx = Cx. \tag{1}$$

Here, $y \in \mathbb{R}^p$, $x \in \mathbb{R}^r$, $H \in \mathbb{R}^{p \times n}$ and $U \in \mathbb{R}^{n \times r}$ are the observation vector, the POD mode amplitude, the sparse sensor location matrix, and the spatial POD modes, respectively. In addition, p, n, and r are the number of sensor locations, the degrees of freedom of the spatial POD modes, and the number of POD modes, respectively. The problem associated with the above equation is considered to be a sensor selection problem when U and strength x are assumed to be a sensor-candidate matrix and the latent state variables, respectively. A graphical representation of the equation is shown in Fig. 1, where the element corresponding to the sensor location is unity and the others are 0 in each row of H.
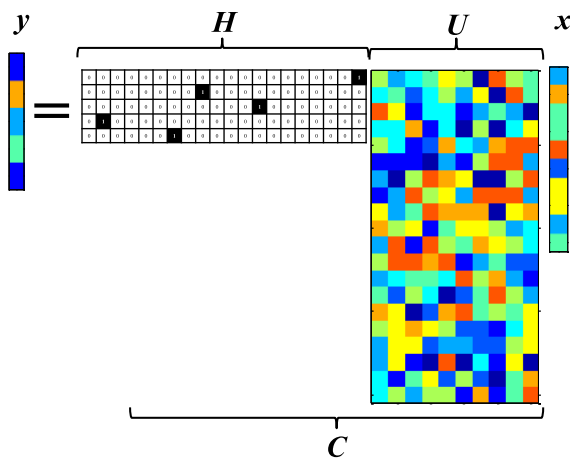

FIGURE 1. Graphical representation for sensor matrix H of (1).

The sparse-sensor-placement problem is determining the optimal sensor placement of the limited p sensors from n sensor candidate points in this paper. Optimal sensor placement is an important challenge in the design, prediction, estimation, and control of high-dimensional systems. High-dimensional states can often leverage a latent low-dimensional representation, and this inherent compressibility enables sparse sensing. For example, in the applications of aerospace engineering, such as launch vehicles and satellites, optimal sensor placement is an important subject in performance prediction, control of the system, fault diagnostics and prognostics, etc., because of the limitations of installation, cost, and downlink capacity for transferring measurement data. Therefore, the study of optimal sensor placement is important in this field. Such sparse point sensors should be selected considering the POD modes. Although compressed sensing can recover a wider class of signals, the benefits of exploiting known patterns in data with optimized sensing is utilized, similar to as in a previous study [7]. Drastic reductions in the required number of sensors and improved reconstruction can be expected in this case. In a series of studies, Clark *et al.* extended this idea and developed an optimization method for placing sensors under a cost constraint [8]. Manohar *et al.* developed a sensor optimization method using balanced truncation for linear systems [9]. Saito *et al.* extended the greedy method to vector sensor

problems in the context of a fluid dynamic measurement application [10]. Thus far, this sensor selection problem has been solved by convex approximation and a greedy algorithm, where the greedy algorithm was shown to be much faster than the convex approximation algorithm [7].

A previous study [7] introduced a greedy algorithm based on the QR-discrete-empirical-interpolation method (QDEIM) [11], [12]. Here, QR stands for QR factorization with column pivoting matrix decomposition. They extended the greedy algorithm for the least-squares problem when the number of sensors is greater than that of state variables. Both convex approximation and greedy methods work reasonably well for the sensor selection problem. Joshi and Boyd [13] defined the objective function for the sensor selection problem, which corresponds to minimization of the volume of the confidence ellipsoid. Joshi and Boyd [13] proposed a convex approximation method for this objective function. On the other hand, the proposed convex approximation method suffers from a long computational time. Recently, Nonomura *et al.* extended this convex approximation method to a randomized subspace Newton convex approximation method for sparse sensor selection [14], while Nagata *et al.* proposed convex/nonconvex methods of the sensor selection problems using the alternating direction method of multipliers (ADMM) [15]. Although those proposed new convex methods reduce the complexity from that of the conventional convex approximation method, e.g. ($\mathcal{O}(mnr^2)$) in the ADMM method compared with ($\mathcal{O}(mn^3)$) in the conventional convex approximation method, they are still slower than the greedy method ($\mathcal{O}(pnr^2)$) because those convex methods require m iterative calculations that are usually greater than the number of sensors (p). Therefore, this paper focuses on the greedy algorithm because the convex approximation method is not computationally efficient for high-dimensional data. Manohar *et al.* [7] have proposed a QR method that provides an approximate greedy solution for the optimization, which is known to be a submatrix volume maximization, and improved the computational time of the problem, relative to the convex approximation method, by using the QR method. However, the QR method does not seem to connect straightforwardly with the problem $y = Cx$ and lacks a mathematical proof for its implementation in [7]. Particularly when the number of sensors is greater than that of state variables, the theoretical background for calculation is unclear for the greedy algorithm, and the complexity of the QR method is not much different from that of the convex approximation method [7] (See Appendix A for a detailed explanation). Therefore, the excellent idea of using the QR method is extended in this study. Peherstorfer *et al.* proposed an oversampling-point selection method in the framework of a DEIM-based reduced order model; the method is based on lower bounds of the smallest eigenvalues of certain structured matrix updates in the case of the number of sensors being greater than that of state variables [16]. They showed that their proposal is almost the best choice among the existing oversampling-point selection methods [7], [11], [17]–[19] reported in DEIM studies.
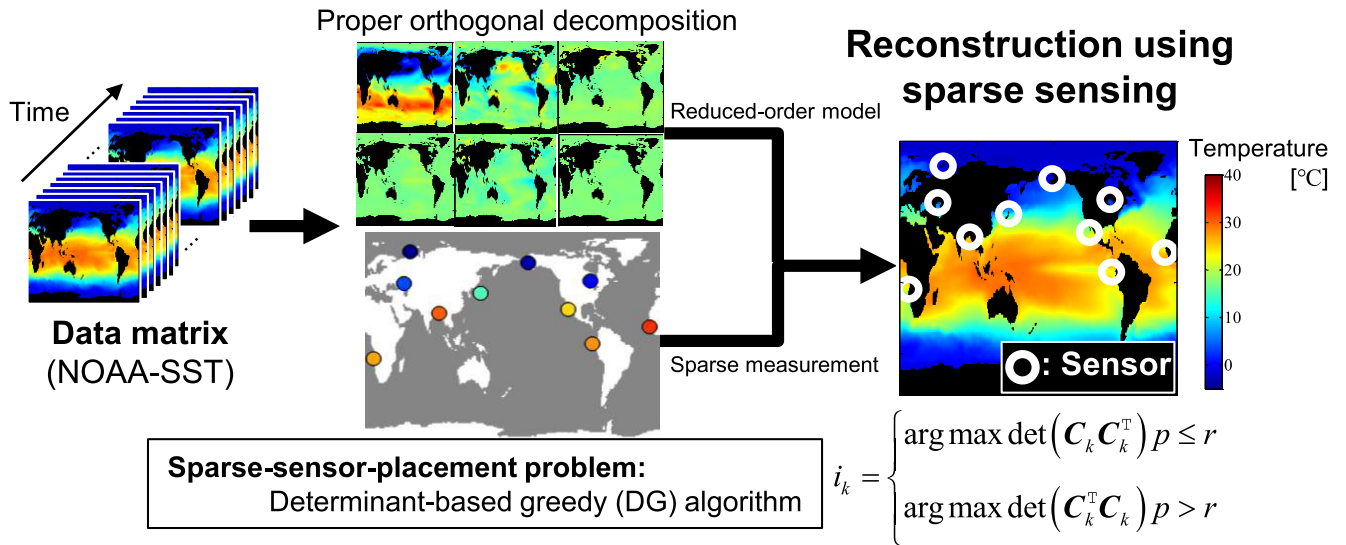
FIGURE 2. Conceptual flow chart of this paper (NOAA-SST sensor problem).

This paper introduces an improved formulation and proves its monotone submodularity and presents its performance. Fig. 2 summarizes the concept of this paper the NOAA-SST sensor problem. The main novelties and contributions of this paper are as follows:

- The main contribution of this paper is extending the previous DEIM studies and the QR method to the cases of undersampled and oversampled sensor placement, where the number of sensors is less and greater than the number of latent state variables, respectively.
- The procedure for the maximization of the determinant of the corresponding matrix is mathematically proved to be the same as that of the QR method when the number of sensors is less than that of state variables (see Theorem 1).
- A unified formulation is derived, and the lower bound of the objective function is shown using monotone submodularity.
- A new hybrid algorithm is proposed in this paper, based on the abovementioned results. The effectiveness of the proposed new hybrid algorithm is demonstrated by comparing with the results of other algorithms, through the problems using the randomly generated data and the real datasets of the flowfield around an airfoil and the global climate.
- The demonstrations show that the proposed algorithm is faster than the other methods and obtains one of the highest determinants and almost the smallest level of estimation errors.

The results for the methods proposed by Joshi and Boyd [13] and Manohar *et al.* [7] will be compared to our results. The results for the method proposed by Peherstorfer *et al.* [16] are not compared in the main discussion of this paper because the objective function of [16] is different from ours. Instead, the superiority of the present algorithm to the

method proposed by Peherstorfer *et al.* [16] is demonstrated in Appendix C by applying the algorithms to a practical example.

II. FORMULATIONS AND ALGORITHMS
A. PREVIOUS GREEDY ALGORITHM [7]
In the greedy algorithm based on QR decomposition for the scalar measurement problem, the ith sensor is chosen as follows:

$$i_k = \underset{i \in \mathcal{S} \setminus \mathcal{S}_k}{arg\ max} \|\boldsymbol{v}_i\|_2^2, \qquad (2)$$

where i_k is the index of the kth selected sensor. Here, we initialize $\boldsymbol{v}_i = [\ V_{i,1}\ V_{i,2}\ \dots\ V_{i,r}\ ]$ and $\{V_{ij}\} = \boldsymbol{V} = \boldsymbol{U}$ and $\{V_{ij}\} = \boldsymbol{V} = \boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}$ for the $p = r$ and $p > r$ sensor conditions, respectively, as done in [7]. Additionally, the set of indices for the sensor candidates and the subset of indices for the determined sensors are expressed as $\mathcal{S}\ (= \{1, \dots, n\})$ and $\mathcal{S}_k\ (= \{i_1, \dots, i_k\}, \mathcal{S}_k \subset \mathcal{S})$, respectively. Given index i, matrix $\boldsymbol{V}$ is pivoted and QR decomposed using the newly selected $\boldsymbol{v}_i$. Then, the next sensor is chosen for the remaining matrix. The algorithms for $p = r$ and $p > r$ sensors are QDEIM and optimized sparse sensor placement as an extension of QDEIM, respectively. The scalar measurement problem is treated in this study, and the extension to the vector measurement problem is left for the future study. It should be noted that Saito *et al.* extended the greedy algorithm based on QR decomposition to the vector measurement problem [10].

The optimization considered is maximization of the determinant of the $\boldsymbol{C}$ matrix to stably solve for the vector $\boldsymbol{x}$ in the case of $p = r$. The selection of a sensor position is based on maximizing the norm of the corresponding row vector of the sensor-candidate matrix. On the other hand, a calculation algorithm for the condition $p > r$ has already been proposed in [7] (which corresponds to the use of $\boldsymbol{V} = \boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}$ instead

of $V = U$), but the validity of the algorithm was not well clarified in [7]. Appendix A concisely explains the work of Manohar *et al.* [7].

B. PROPOSED DETERMINANT-BASED GREEDY ALGORITHM

Let C_k denote the kth sensor-candidate matrix as follows:

$$C_k = \left[u_{i_1}^T \ u_{i_2}^T \ \dots \ u_{i_{k-1}}^T \ u_{i_k}^T \right]^T, \tag{3}$$

where i_k and u_{i_k} are the index of the kth selected sensor and the corresponding row vector of the sensor-candidate matrix, respectively. The following algorithm can be derived straightforwardly by considering the pseudo-inverse matrix of C. The greedy sensor selection problem is defined such that the locations of the first-to-$(k - 1)$th sensors already selected are known. Equation (1) can be solved as $\hat{x} = C^\dagger y$ and divided into two cases:

$$\hat{x} = \begin{cases} C^T \left(C C^T \right)^{-1} y, & p \leq r, \\ \left(C^T C \right)^{-1} C^T y, & p > r. \end{cases} \tag{4}$$

Here, C is assumed to be a full-column-rank or full-row-rank matrix. Our sensor selection automatically produces a full-rank observation matrix because of the procedure of the maximization of the determinant. This assumption is reasonable because the performance of a full-rank observation matrix is clearly better than that of a rank-deficient matrix. Hence, the discussion for a full-column-rank or full-row-rank matrix only is sufficient for the algorithm developed in this study. Here, U is considered to be a full-column-rank matrix, and the existence of any unobservable subspace is not considered for simplicity.

The optimization could also be divided into the two cases of maximizing the determinants of $C C^T$ and $C^T C$ for $p \leq r$ and $p > r$, respectively. The maximization of the determinant of $C C^T$ in the case of $p \leq r$ is justified by being connected to the maximization of the determinant of $C^T C$, as indicated in (14) later in this paper. Here, the maximization of the determinant of $C^T C$ in the case of $p > r$ is justified as follows. The uniform independent Gaussian noises $\mathcal{N}(0, \sigma^2 I)$ are assumed to be imposed on the observation, and the estimated parameters $\hat{x}$ can be obtained as

$$\hat{x} = \left(C^T C \right)^{-1} C^T (y + H v) = x + \left(C^T C \right)^{-1} C^T H v. \tag{5}$$

The covariance matrix of the estimation error is expressed as follows:

$$\begin{aligned} E &\left[(x - \hat{x}) (x - \hat{x})^T \right] \\ &= E \left[\left(C^T C \right)^{-1} C^T H v v^T H^T C \left(C^T C \right)^{-1} \right] \\ &= \sigma^2 E \left[\left(C^T C \right)^{-1} C^T C \left(C^T C \right)^{-1} \right] \\ &= \sigma^2 \left(C^T C \right)^{-1}, \end{aligned} \tag{6}$$

where $E \left[v v^T \right] = \sigma^2 I$ and $H H^T = I$. Therefore, the maximizations of the determinants of $C C^T$ and $C^T C$ are equivalent to minimizing the determinant of the error covariance matrix, resulting in minimizing the volume of the confidence ellipsoid of the regression estimates of the linear model parameters.

The first-to-$(k - 1)$th sensors are already selected and it becomes the selecting problem of only the kth sensor in the greedy algorithm. The problem in the case of $k \leq r$ has already been addressed by Manohar *et al.* [7]. Their algorithm selects the sensor position based on maximizing the norm of the corresponding row vector of the sensor-candidate matrix U applying the QR decomposition, as explained in [7], where they note that the QR decomposition provides an approximate greedy solution for the maximization of the determinant of C. The equivalence of the QR pivoting decision to the greedy heuristic maximizing the determinant was previously stated in [8] without proof. Therefore, the mathematical background for the maximization of the determinant of $C C^T$ will be described in the next section. The problem in the case of $k > r$ has also been addressed in previous studies [13], [20], and the proposed objective in this study (the maximization of the determinant of $C^T C$) corresponds to designing an experiment to minimize the volume of the resulting confidence ellipsoid [20].

The step-by-step maximization of the determinant of $C_k C_k^T$ is considered using the greedy method in the case of $k \leq r$. Here, the objective is to maximize the determinant of the matrix appearing in pseudo-inverse matrix operations, leading to a minimization of confidence intervals. The matrix can be expanded as follows (see, e.g., [21] for a detailed derivation):

$$\begin{aligned} \det(C_k C_k^T) &= \det \left(\begin{bmatrix} C_{k-1} \\ u_i \end{bmatrix} \begin{bmatrix} C_{k-1}^T & u_i^T \end{bmatrix} \right) \\ &= \det \left(\begin{bmatrix} C_{k-1} C_{k-1}^T & C_{k-1} u_i^T \\ u_i C_{k-1}^T & u_i u_i^T \end{bmatrix} \right) \\ &= u_i \left(I - C_{k-1}^T \left(C_{k-1} C_{k-1}^T \right)^{-1} C_{k-1} \right) u_i^T \\ &\quad \times \det \left(C_{k-1} C_{k-1}^T \right), \end{aligned} \tag{7}$$

and, therefore,

$$\begin{aligned} i_k &= \arg\max_{i \in \mathcal{S} \setminus \mathcal{S}_k} \det \left(C_k C_k^T \right) \\ &= \arg\max_{i \in \mathcal{S} \setminus \mathcal{S}_k} u_i \left(I - C_{k-1}^T \left(C_{k-1} C_{k-1}^T \right)^{-1} C_{k-1} \right) u_i^T. \end{aligned} \tag{8}$$

In the kth step of sensor selection of the QR (or Gram-Schmidt) method and the present method, the following equality is obtained:

$$v_i v_i^T = u_i \left(I - C_{k-1}^T \left(C_{k-1} C_{k-1}^T \right)^{-1} C_{k-1} \right) u_i^T, \tag{9}$$

and, therefore, the two algorithms are equivalent.

Theorem 1: The QR method is mathematically equivalent to the DG method in the case of $k \le r$.

This is because $I - C_{k-1}^{\mathrm{T}} \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1} C_{k-1} = P_{C_{k-1}}^{\perp}$ is the projection matrix to the orthogonal complement of the already selected sensor vector space, and therefore the absolute value of the sensor-candidate matrix in the orthogonal complement space $u_i \left(I - C_{k-1}^{\mathrm{T}} \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1} C_{k-1} \right) u_i^{\mathrm{T}}$ exactly corresponds to the absolute value of v_i, which is the sensor-candidate vector with the sensor components already selected in the kth step subtracted. See Appendix B for a detailed proof.

The following equation illustrates a somewhat efficient way to recursively calculate $\left(C_k C_k^{\mathrm{T}} \right)^{-1}$ when the new sensor is found and the step number k is incremented:

$$\left(C_k C_k^{\mathrm{T}} \right)^{-1} = \left(\begin{bmatrix} C_{k-1} \\ u_{i_k} \end{bmatrix} \begin{bmatrix} C_{k-1}^{\mathrm{T}} & u_{i_k}^{\mathrm{T}} \end{bmatrix} \right)^{-1} = \begin{pmatrix} A & b^{\mathrm{T}} \\ b & d \end{pmatrix},$$

(10)

where

$$A = \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1}$$
$$\times \left\{ I + \frac{C_{k-1} u_{i_k}^{\mathrm{T}} u_{i_k} C_{k-1}^{\mathrm{T}} \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1}}{u_{i_k} \left(I - C_{k-1}^{\mathrm{T}} \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1} C_{k-1} \right) u_{i_k}^{\mathrm{T}}} \right\},$$

$$b = \frac{-u_{i_k} C_{k-1}^{\mathrm{T}} \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1}}{u_{i_k} \left(I - C_{k-1}^{\mathrm{T}} \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1} C_{k-1} \right) u_{i_k}^{\mathrm{T}}},$$

$$d = \frac{1}{u_{i_k} \left(I - C_{k-1}^{\mathrm{T}} \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1} C_{k-1} \right) u_{i_k}^{\mathrm{T}}}.$$

However, the overall computational speed of the previous QR method is faster than that of the greedy algorithm using this recursive equation. The previous QR method is therefore chosen for the case of $k < r$ except for the validation in this paper. Although, the algorithm cannot be used for practical applications in a straightforward manner, it is presented herein as Alg. 1.

The maximization of the determinant of $C^{\mathrm{T}} C$ is simply considered in the case of $k \ge r$ (see, e.g., [21] for a detailed derivation), which is commonly known as the D-optimal criterion in design of experiments.

$$\det \left(C_k^{\mathrm{T}} C_k \right)$$
$$= \det \left(\begin{bmatrix} C_{k-1}^{\mathrm{T}} & u_i^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} C_{k-1} \\ u_i \end{bmatrix} \right)$$
$$= \left(1 + u_i \left(C_{k-1}^{\mathrm{T}} C_{k-1} \right)^{-1} u_i^{\mathrm{T}} \right) \det \left(C_{k-1}^{\mathrm{T}} C_{k-1} \right). \quad (11)$$

Therefore,

$$i_k = \arg \max_{i \in \mathcal{S} \setminus \mathcal{S}_k} \det \left(C_{k-1}^{\mathrm{T}} C_{k-1} + u_i^{\mathrm{T}} u_i \right)$$
$$= \arg \max_{i \in \mathcal{S} \setminus \mathcal{S}_k} \left(1 + u_i \left(C_{k-1}^{\mathrm{T}} C_{k-1} \right)^{-1} u_i^{\mathrm{T}} \right). \quad (12)$$

Algorithm 1 Determinant-Based Greedy Algorithm for Sparse Sensor Placement for $k \le r$ (corresponding to the Previous QR Algorithm for $p = r$)

Set sensor-candidate matrix U.
$u_i = \begin{bmatrix} U_{i,1} & U_{i,2} & \dots & U_{i,r} \end{bmatrix}$
$i \leftarrow \arg \max_{i \in \mathcal{S}} \left(1 + u_i u^{\mathrm{T}} \right)$
$H_{1,i} = 1$
$C_1 = u_i$
$\begin{bmatrix} U_{i,1} & U_{i,2} & \dots & U_{i,r} \end{bmatrix} \leftarrow [0, 0, \dots, 0]$
for $k = 2, \dots, r$ **do**
$\quad u_i = \begin{bmatrix} U_{i,1} & U_{i,2} & \dots & U_{i,r} \end{bmatrix}$
$\quad i \leftarrow \arg \max_{i \in \mathcal{S} \setminus \mathcal{S}_k} u_i \left(I - C_{k-1}^{\mathrm{T}} \left(C_{k-1} C_{k-1}^{\mathrm{T}} \right)^{-1} C_{k-1} \right) u_i^{\mathrm{T}}$
$\quad$ Update $\left(C_k C_k^{\mathrm{T}} \right)^{-1}$ using Eq. (10)
$\quad H_{k,i} = 1$
$\quad C_k = \begin{bmatrix} u_{i_1}^{\mathrm{T}} & u_{i_2}^{\mathrm{T}} & \dots & u_{i_{k-1}}^{\mathrm{T}} & u_{i_k}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$
$\quad \begin{bmatrix} U_{i,1} & U_{i,2} & \dots & U_{i,r} \end{bmatrix} \leftarrow [0, 0, \dots, 0]$
end for

The complexity of this procedure is only $O(r^2)$ for one component of a candidate sensor vector when $\left(C_{k-1}^{\mathrm{T}} C_{k-1} \right)^{-1}$ is known, and the complexity when searching all components of the vector becomes $O(nr^2)$. When a new sensor is found and the step number k is incremented, $\left(C_k^{\mathrm{T}} C_k \right)^{-1}$ can be computed recursively using a matrix inversion lemma as follows:

$$\left(C_k^{\mathrm{T}} C_k \right)^{-1}$$
$$= \left(C_{k-1}^{\mathrm{T}} C_{k-1} + u_i^{\mathrm{T}} u_i \right)^{-1}$$
$$= \left(C_{k-1}^{\mathrm{T}} C_{k-1} \right)^{-1}$$
$$\times \left(I - u_i^{\mathrm{T}} \left(1 + u_i \left(C_{k-1}^{\mathrm{T}} C_{k-1} \right)^{-1} u_i^{\mathrm{T}} \right)^{-1} \right.$$
$$\left. \times u_i \left(C_{k-1}^{\mathrm{T}} C_{k-1} \right)^{-1} \right). \quad (13)$$

The determinant-based greedy algorithm for sparse sensor placement for $k > r$ is summarized as Alg. 2.

C. UNIFIED EXPRESSIONS FOR PROPOSED ALGORITHM

Here, we introduce unified expressions for the case in which the number of sensors is less than that of modes and that in which the number of sensors is greater than or equal to that of modes. The maximization of the term $\det \left(C^{\mathrm{T}} C + \epsilon I \right)$ is considered in this case, where ϵ is a sufficiently small number. This objective function approximately corresponds to $\det \left(C^{\mathrm{T}} C \right)$ when the number of sensors is greater than or equal to that of the modes. On the other hand, when the number of sensors p is less than that of the modes r, it becomes

$$\det \left(C^{\mathrm{T}} C + \epsilon I \right) = \epsilon^r \det \left(I + \frac{1}{\epsilon} C^{\mathrm{T}} C \right)$$

Algorithm 2 Determinant-Based Greedy Algorithm for Sparse Sensor Placement for $k > r$

Set sensor-candidate matrix U.
Obtain H_o using the previous QR algorithm (or Alg. 1) for the number of sensors $p_o = r$.
Set $\left(C_{p_o}^T C_{p_o}\right)^{-1} = \left(U^T H_o^T H_o U\right)^{-1}$
for i of the sensor position chosen by the conventional greedy algorithm **do**
 Set $U_i = \begin{bmatrix} U_{i,1} & U_{i,2} & \ldots & U_{i,r} \end{bmatrix} = [0, 0, \ldots, 0]$
end for
for $k = p_o + 1, \ldots, p$ **do**
 $U_i = \begin{bmatrix} U_{i,1} & U_{i,2} & \ldots & U_{i,r} \end{bmatrix}$
 $i \leftarrow \underset{i}{\arg\max} \left(1 + u_i \left(C_{k-1}^T C_{k-1}\right)^{-1} u_i \right)$
 $\left(C_k^T C_k\right)^{-1}$
  $\leftarrow \left(C_{k-1}^T C_{k-1}\right)^{-1}$
  $\left(I - u_i^T \left(1 + u_i \left(C_{k-1}^T C_{k-1}\right)^{-1} u_i^T \right)^{-1} \right.$
  $\left. u_i \left(C_{k-1}^T C_{k-1}\right)^{-1} \right)$
 $H_{k,i} = 1$
 $C_k = [u_{i_1}^T \; u_{i_2}^T \; \ldots \; u_{i_{k-1}}^T \; u_{i_k}^T]^T$
 $[U_{i,1} \; U_{i,2} \; \ldots \; U_{i,r}] \leftarrow [0, 0, \ldots, 0]$
end for

$$= \epsilon^r \det\left(I + \frac{1}{\epsilon} C C^T \right)$$
$$= \epsilon^{r-p} \det\left(C C^T + \epsilon I \right), \quad (14)$$

and then the objective corresponds to the maximization of $\det\left(C C^T + \epsilon I\right) \sim \det\left(C C^T\right)$ as we did in the previous subsection. Therefore, the approximated objective function is defined to be $\det(C^T C + \epsilon I)$ which asymptotically approaches the maximum of the determinant in the limit that ϵ goes to 0. Although this function with sufficiently small ϵ should not be implemented in the computational code because it may include large round-off error when ϵ is sufficiently small to neglect its effects, this unified function will be used for the proof of monotone submodularity and discussion of the approximation rate in the next section.

III. SUBMODULARITY AND APPROXIMATION RATE

It is well known that an explicit approximation rate of the greedy algorithm is available if the objective function is monotone and submodular [22]. In this section, we show that a slightly modified version of our unified objective function is certainly monotone and submodular. By using the approximation rate for this function, we derive a lower bound for the proposed algorithm.

Let $S \subset \{1, 2, \ldots, n\}$ be a set of labels of selected sensors and C_S be the corresponding sensor matrix. Namely, if $S = \{i_1, i_2, \ldots, i_k\}$, then C_S is given by

$$C_S = \begin{bmatrix} u_{i_1}^T & u_{i_2}^T & \cdots & u_{i_k}^T \end{bmatrix}^T. \quad (15)$$

Define a function $f : 2^{\{1,2,\ldots,n\}} \to \mathbb{R}$ by

$$f(S) = \log \det(C_S^T C_S + \epsilon I) - r \log \epsilon$$
$$= \log \det\left(\frac{1}{\epsilon} C_S^T C_S + I \right) \quad (16)$$

for each $S \subset \{1, 2, \ldots, n\}$. An offset term $-r \log \epsilon$ is added so that the value of f for the empty set can be regarded as $f(\emptyset) = \log \det I = 0$.

In what follows, we first show that f is a submodular function. Namely, for any $S, T \subset \{1, 2, \ldots, n\}$ with $S \subset T$ and $i \in \{1, 2, \ldots, n\} \setminus T$, the function f satisfies

$$f\left(S \cup \{i\}\right) - f(S) \geq f\left(T \cup \{i\}\right) - f(T). \quad (17)$$

The monotonicity of f is next proved.

Proposition 1: The function $f : 2^{\{1,2,\ldots,n\}} \to \mathbb{R}$ defined by (16) is submodular.

Proof: Take arbitrary $S, T \subset \{1, 2, \ldots, n\}$ such that $S \subset T$. For simplicity of notation, let us set $\Psi_\epsilon := (C_S^T C_S + \epsilon I)^{-1}$. It follows from the definition of f that, for any $i \in \{1, 2, \ldots, n\} \setminus T$,

$$f(S \cup \{i\}) - f(S)$$
$$= \log \det(C_S^T C_S + u_i^T u_i + \epsilon I) - \log \det(C_S^T C_S + \epsilon I)$$
$$= \log \det\left((C_S^T C_S + u_i^T u_i + \epsilon I) \Psi_\epsilon \right)$$
$$= \log \det\left(I + u_i^T u_i \Psi_\epsilon \right)$$
$$= \log\left(1 + u_i \Psi_\epsilon u_i^T \right). \quad (18)$$

Due to the positive definiteness of Ψ_ϵ, the value of (18) is positive for any $u_i \neq 0$.

We next evaluate $f(T \cup \{i\}) - f(T)$. Since $S \subset T$, there exists a permutation matrix P such that

$$P C_T = \begin{bmatrix} C_S \\ C_{T \setminus S} \end{bmatrix}. \quad (19)$$

Hence, we have

$$C_T^T C_T = C_T^T P^T P C_T = C_S^T C_S + C_{T \setminus S}^T C_{T \setminus S}, \quad (20)$$

where the fact $P^T P = I$ has been used. Direct computation together with (20) gives

$$f(T \cup \{i\}) - f(T)$$
$$= \log \det(C_T^T C_T + u_i^T u_i + \epsilon I) - \log \det(C_T^T C_T + \epsilon I)$$
$$= \log\left(1 + u_i(C_T^T C_T + \epsilon I)^{-1} u_i^T \right)$$
$$= \log\left(1 + u_i(C_S^T C_S + C_{T \setminus S}^T C_{T \setminus S} + \epsilon I)^{-1} u_i^T \right). \quad (21)$$

The matrix in parentheses can be calculated as

$$\left(C_S^T C_S + C_{T \setminus S}^T C_{T \setminus S} + \epsilon I\right)^{-1}$$
$$= \Psi_\epsilon \left(I + C_{T \setminus S}^T C_{T \setminus S} \Psi_\epsilon \right)^{-1}$$
$$= \Psi_\epsilon \left(I - C_{T \setminus S}^T \left(I + C_{T \setminus S} \Psi_\epsilon C_{T \setminus S}^T \right)^{-1} C_{T \setminus S} \Psi_\epsilon \right)$$

$$= \boldsymbol{\Psi}_\epsilon - \boldsymbol{\Psi}_\epsilon \boldsymbol{C}_{T \setminus S}^{\mathrm{T}} \left(\boldsymbol{I} + \boldsymbol{C}_{T \setminus S} \boldsymbol{\Psi}_\epsilon \boldsymbol{C}_{T \setminus S}^{\mathrm{T}} \right)^{-1} \boldsymbol{C}_{T \setminus S} \boldsymbol{\Psi}_\epsilon$$

$$=: \boldsymbol{\Psi}_\epsilon - \overline{\boldsymbol{\Psi}}_\epsilon \tag{22}$$

Note that $\overline{\boldsymbol{\Psi}}_\epsilon$ is a positive semidefinite matrix. Substituting the above relation into (21) yields

$$f(T \cup \{i\}) - f(T) = \log\left(1 + \boldsymbol{u}_i \boldsymbol{\Psi}_\epsilon \boldsymbol{u}_i^{\mathrm{T}} - \boldsymbol{u}_i \overline{\boldsymbol{\Psi}}_\epsilon \boldsymbol{u}_i^{\mathrm{T}} \right). \tag{23}$$

Since $\overline{\boldsymbol{\Psi}}_\epsilon$ is positive semidefinite, the monotonicity of the logarithm gives

$$\log\left(1 + \boldsymbol{u}_i \boldsymbol{\Psi}_\epsilon \boldsymbol{u}_i^{\mathrm{T}} \right) \geq \log\left(1 + \boldsymbol{u}_i \boldsymbol{\Psi}_\epsilon \boldsymbol{u}_i^{\mathrm{T}} - \boldsymbol{u}_i \overline{\boldsymbol{\Psi}}_\epsilon \boldsymbol{u}_i^{\mathrm{T}} \right), \tag{24}$$

which immediately implies (17). This completes the proof. ☐

The next result guarantees the monotonicity of f.

Proposition 2: The function $f : 2^{\{1,2,\ldots,n\}} \to \mathbb{R}$ defined by (16) is monotone.

Proof: For given $S, T \subset \{1, 2, \ldots, n\}$ with $S \subset T$, it is clear that (18) holds also for any $i \in T \setminus S$. Hence, $f(S \cup \{i\}) - f(S) \geq 0$. Similarly, we can show that $f(S \cup \{i\} \cup \{j\}) - f(S) \geq 0$ for $j \in T \setminus (S \cup \{i\})$. Repeated application of this argument yields $f(S) \leq f(T)$, which is the desired conclusion. ☐

We have seen that f is a monotone submodular function. For optimization problems with respect to monotone submodular functions, Nemhauser *et al.* have proved that the following inequality holds [22]:

$$f(S_{\mathrm{greedy}}) \geq \left(1 - \left(1 - \frac{1}{k} \right) \right)^k f(S_{\mathrm{opt}})$$

$$\geq \left(1 - \frac{1}{e} \right) f(S_{\mathrm{opt}}), \tag{25}$$

where S_{opt} is an optimal solution and S_{greedy} is the solution obtained by the greedy algorithm. Applying this inequality to f defined by (16) gives

$$\left(1 - \frac{1}{e} \right) \left(\log \det \left(\boldsymbol{C}_{S_{\mathrm{opt}}}^{\mathrm{T}} \boldsymbol{C}_{S_{\mathrm{opt}}} + \epsilon \boldsymbol{I} \right) - r \log \epsilon \right)$$

$$\leq \log \det \left(\boldsymbol{C}_{S_{\mathrm{greedy}}}^{\mathrm{T}} \boldsymbol{C}_{S_{\mathrm{greedy}}} + \epsilon \boldsymbol{I} \right) - r \log \epsilon. \tag{26}$$

Several properties of the logarithm lead to

$$\epsilon^{\frac{r}{e}} \left(\det \left(\boldsymbol{C}_{S_{\mathrm{opt}}}^{\mathrm{T}} \boldsymbol{C}_{S_{\mathrm{opt}}} + \epsilon \boldsymbol{I} \right) \right)^{1 - \frac{1}{e}} \leq \det \left(\boldsymbol{C}_{S_{\mathrm{greedy}}}^{\mathrm{T}} \boldsymbol{C}_{S_{\mathrm{greedy}}} + \epsilon \boldsymbol{I} \right). \tag{27}$$

This inequality provides the lower bound of the greedy approximation for our problem.

Parameter ϵ was originally assumed to be sufficiently small. However, the coefficient $\epsilon^{r/e}$ on the right-hand side of (27) tends to 0 as ϵ goes to 0. Hence, a finite ϵ should be appropriately adopted to obtain a practical lower bound for the unified greedy algorithm proposed in this paper.

IV. RESULTS AND DISCUSSION

The numerical experiments are conducted and the proposed methods are validated. Hereinafter, three different implementations of the greedy methods are compared: the QR, DG, and QD methods listed in Table 1. Here, the QR method is the greedy method proposed in [7], the DG method is the greedy method for the pure maximization of the determinant, and the QD method is the greedy method, one part of which is replaced by the QR method which is mathematically proved to be equivalent to the greedy method for the pure maximization of the determinant. Table 2 explains the computational complexities of each method for selecting p sensors. The QD method is included because the QR implementation for the previously proposed greedy optimization is much faster than the DG method, and the two implementations are demonstrated to give us the same numerical optimized solution as each other except for round-off errors in practical situations. The random selection and convex approximation methods [13] are evaluated as the references, in addition to the greedy methods.

TABLE 1. Greedy sensor selection methods investigated in this study.

Name	Method	Implementation
QR [7]	$p \leq r$: QR for $\boldsymbol{C}$	Alg. 3
	$p > r$: QR for $\boldsymbol{CC}^{\mathrm{T}}$	Alg. 3
DG (present)[a]	$p \leq r$: arg max det $\left(\boldsymbol{CC}^{\mathrm{T}} \right)$	Alg. 1
	$p > r$: arg max det $\left(\boldsymbol{C}^{\mathrm{T}} \boldsymbol{C} \right)$	Alg. 2
QD (present, recommended)[a]	$p \leq r$: QR for $\boldsymbol{C}$	Alg. 3
	$p > r$: arg max det $\left(\boldsymbol{C}^{\mathrm{T}} \boldsymbol{C} \right)$	Alg. 2

TABLE 2. Computational complexities of each method for selecting the pth sensor.

Name	Computational complexity
QR [7]	$p \leq r$: $O\left(nr^2 \right)$
	$p > r$: $O\left(n^3 \right)$
DG (present)	$p \leq r$: $O\left(pnr^2 \right)$
	$p > r$: $O\left(pnr^2 \right)$
QD (present, recommended)	$p \leq r$: $O\left(pnr \right)$
	$p > r$: $O\left(pnr^2 \right)$
convex [13]	$O\left(n^3 \right)$ per iterations
	*usually converges around 100 iterations.

A. RANDOM SENSOR PROBLEM

The random sensor-candidate matrices, $\boldsymbol{U} \in \mathbb{R}^{2000 \times r}$, are set according to the Gaussian distribution of $\mathcal{N}(0, 1)$ with $n = 2000$ and $r = 10$ for the components in this validation. The random selection, convex approximation, QR, DG, and QD methods listed in Table 1 are evaluated. The determinants of $\boldsymbol{CC}^{\mathrm{T}} = \boldsymbol{HUU}^{\mathrm{T}} \boldsymbol{H}^{\mathrm{T}}$ ($p \leq r$) and $\boldsymbol{C}^{\mathrm{T}} \boldsymbol{C} = \boldsymbol{U}^{\mathrm{T}} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{HU}$ ($p > r$) are calculated using both sensor-candidate matrices after selecting sensors. The random sensor problem is solved

TABLE 3. Computational environments.

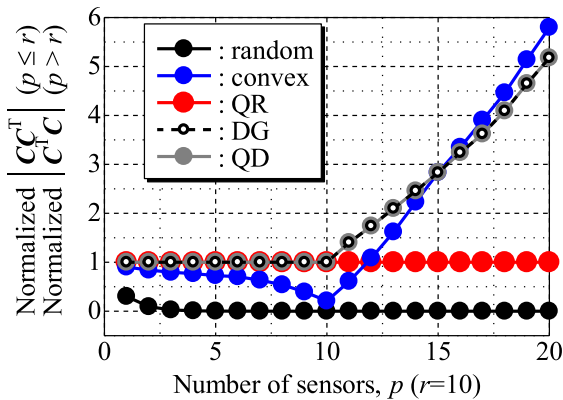|	
Processor information	Intel(R) Core(TM) i7-6800K@ 3.40 GHz
Random access memory	128 GB
System type	64 bit operating system x64 base processor
Program code	Matlab R2013a [23]
Operating system	Windows 10 Pro Version:1890

FIGURE 3. Normalized determinant of CC^T ($p \leq r$) or $C^T C$ ($p > r$) against the number of sensors for the number of POD modes $r = 10$ in the random sensor problem.

under the computational environments listed in Table 3. The results are averages over 1000 random samples. Fig. 3 shows the relationship between the determinant of CC^T ($p \leq r$) or $C^T C$ ($p > r$) and the number of sensors p for the number of POD modes $r = 10$ in the random sensor problem, obtained by the random selection (solid black line with closed circles), convex approximation (solid blue line with closed circles), QR (solid red line with closed circles), DG (dotted black line with open circles), and QD (solid gray line with closed circles) methods. All plots are normalized by the determinant of CC^T ($p \leq r$) or $C^T C$ ($p > r$) for the QR method. This normalization shows a clear trend of each method compared with that of the conventional QR method. There exist pathological matrices for which the determinant does not increase, namely, the Kahan matrices [24]. However, we did not experience such failures, and all the determinants of CC^T ($p \leq r$) or $C^T C$ ($p > r$) increase in the random problem. The normalized determinants of CC^T of both the DG and QD methods are the same as that of the QR method in the case of $p \leq r$, which confirms what is shown in Appendix B. On the other hand, the normalized determinants of $C^T C$ of both the DG and QD methods increase as the number of sensors increases in the case of $p > r$. The DG method in the case of $p > r$ selects the sensor positions using the step-by-step maximization of the determinant of $C^T C$. On the other hand, the QR method conducts the full

QR decomposition for UU^T and does not seem to connect straightforwardly with the maximization of the determinant of $C^T C$. For example, the determinants of $C^T C$ of both the DG and QD methods are five times higher than that of the QR method in the case of $p = 20$ for the random sensor problem. Although the computational complexity or time of the convex approximation method is much larger than those of other calculation methods, the convex approximation method obtains the highest normalized determinant of $C^T C$ for $p > 15$ among these methods.

Fig. 4 shows the relationship between the computational time and the number of sensors p for the number of POD modes $r = 10$ in the random sensor problem. As indicated in Table 2, the computational complexity of the convex approximation algorithm is the largest among these calculation methods. The computational complexity of the QR method is close to that of the convex approximation method because the QR method is implemented in Matlab using the native Matlab "qr" subroutine and returns the desired list of all sensors. The QR and QD methods for $p \leq r$ give the same time because the same algorithm is used in this case. The computational time of the DG method gradually increases as the number of sensors increases because each sensor is chosen using the step-by-step maximization of the determinant of $C_k C_k^T$ or $C_k^T C_k$. On the other hand, the computational time for the QR method is almost constant for $1 \leq p \leq 10$ and $11 \leq p \leq 20$, because it takes a long time to conduct the full QR decomposition. The computational time of the QR method has a large difference at $p = 11$ because the dimension of the matrix processed by the QR decomposition increases between the computation for $U \in \mathbb{R}^{n \times r}$ and that for $UU^T \in \mathbb{R}^{n \times n}$. The trend of the computational time for the QD method is similar to that for the QR method, but the computational time of the QD method is much shorter than that of the QR method in the case of $p > r$. For example, the computational time of the QD method is approximately 10 times smaller than that of the QR and convex methods in
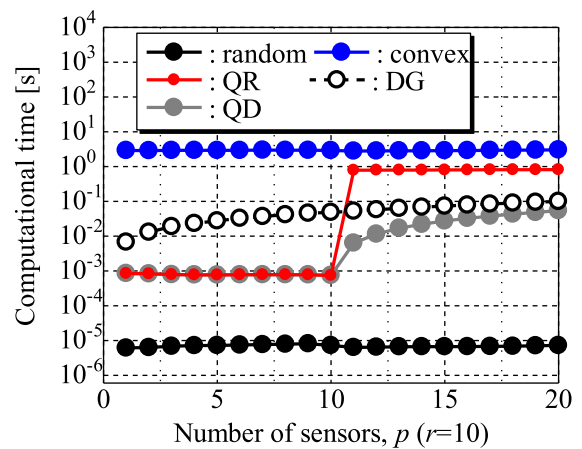
FIGURE 4. Computational time against the number of sensors for the number of POD modes $r = 10$ in the random sensor problem.

the case of $p = 20$ for the random sensor problem. Therefore, the QD method proposed in this paper is more effective for sparse sensor placement than the convex algorithm because the determinants of $\boldsymbol{CC}^T$ ($p \leq r$) and $\boldsymbol{C}^T\boldsymbol{C}$ ($p > r$) are larger and the computational time of the QD method is much shorter than those of the other calculation methods in the case of $1 \leq p \leq 20$ (the number of POD modes $r = 10$ for the random sensor problem).

B. PIV SENSOR PROBLEM

Real-time particle-image-velocimetry (PIV) measurement of the flowfield is required for the feedback control of a high-speed flowfield in laboratory experiments. The flowfield is calculated from the cross-correlation coefficient for each interrogation window of the particle image in the PIV measurement, but the number of windows that can be processed in a short time is limited because of the high computational costs of the calculation of cross-correlation coefficients. We have been developing a sparse-processing-particle-image-velocimetry (SPPIV)-measurement system [25]. The key point of this SPPIV-measurement system is that the amount of processing data is reduced and the flowfield is estimated by a limited number of sparsely located selected windows located, which allows the real-time PIV measurement of the flowfield. The development of an appropriate sensor selection method is required for the highly accurate SPPIV, and the framework of this study could prove useful for that task.

A time-resolved dataset of the flowfield around an airfoil was acquired by a PIV measurement previously [26] (Fig. 5) and is used for the reconstruction problem as a demonstration of a key technology used in SPPIV. Although the latent variable estimation and corresponding sensor selection methods might be better to be extended to the vector measurement [10], [25] owing to the nature of the PIV measurement, this problem is simplified to be one with the scalar measurement and is used for the verification of the proposed algorithm in this study. Therefore, the lateral components of the velocities are only employed in this study, unlike the previous study in which the lateral and vertical components of the velocities are

simultaneously treated [10], [25]. These PIV data with noise were obtained through real wind-tunnel experiments.

Here, the PIV data are briefly explained, and refer to [26] for a detailed explanation. The wind tunnel testing was conducted in the Tohoku-University Basic Aerodynamic Research Wind Tunnel (T-BART) with a closed test section having a 300 mm × 300 mm cross-section. The airfoil of the test model had an NACA0015 profile, with the chord length and span width being 100 mm and 300 mm, respectively. The freestream velocity U_∞ and the angle of attack for the airfoil α were set to be 10 m/s and 16 degrees, respectively. The chord Reynolds number was 6.4×10^4. Time-resolved PIV measurement was conducted with a double-pulse laser (LDY-300PIV, Litron). The time between pulses, the sampling rate, the particle image resolution, and the total number of image pairs were 100 μs, 5000 Hz, 1024 × 1024 pixels, and 500, respectively. The tracer particles were 50% aqueous solution of glycerin with an estimated diameter of a few micrometers. The particle images were acquired by using the double-pulse laser and a high-speed camera (SA-X2, Photron), which were synchronized to each other.

The sparse sensor problem for the reconstruction of the lateral-velocity components of velocities measured by PIV is solved using the same computational environments in Table 3 as used in the random sensor problem. Fig. 6 shows the relationship between the estimation error and the number of POD modes, where the estimation error is defined as the ratio of the difference between the reconstructed data and the full observation data to the full observation. The estimation error e is defined as follows:

$$e = \frac{1}{N} \sum_{j=1}^{N} \frac{\|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j\|_2^2}{\|\boldsymbol{x}_j\|_2^2} \qquad (28)$$

where $\hat{\boldsymbol{x}}$ is the estimated data vector by using a sparse sensor measurement and competed as follows: $\hat{\boldsymbol{x}} = \boldsymbol{UC}^\dagger\boldsymbol{y}$. Here, the subscript j denotes the quantity of jth time step and N
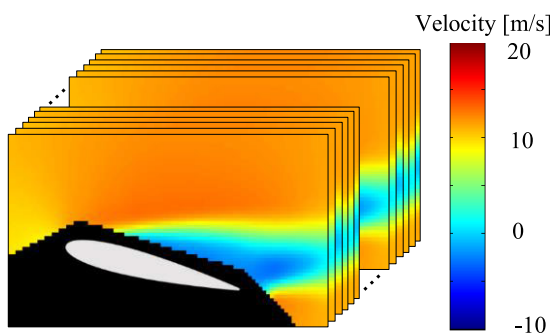

FIGURE 5. Dataset of PIV. Here, only the lateral components of the velocities are used for this problem.
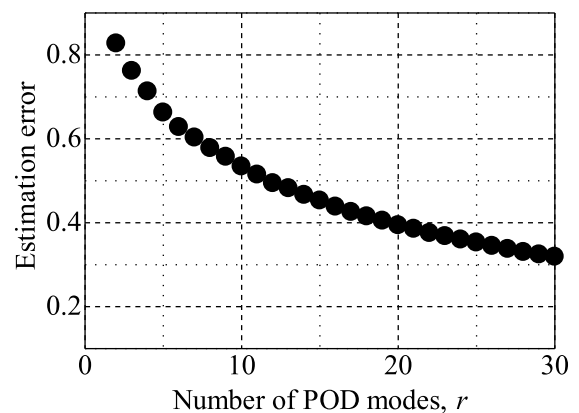

FIGURE 6. Estimation error against the number of POD modes for the PIV sensor problem. Estimation error here refers to the error from projecting the full state onto *r* POD modes. Here, only the lateral components of the velocities are used for this problem.

represents the number of tested data. In the full observation, the estimated data vector is computed as follows: $\hat{x} = U_r U_r^T x$. The POD modes are truncated and an $r = 10$ low-dimensional representation is obtained. Fig. 6 shows that the minimum estimation error is 0.54 for $r = 10$. We consider the results of the sensor selection problem (the number of sensors $p = 1$–20) solved by the methods: the random selection, convex approximation [13], QR [7], DG, and QD methods. Figs. 7 and 8 show the sensor positions and the reconstructed flowfield of a single snapshot from the sensors selected by the random selection, convex approximation, QR, and QD methods in the cases of, respectively, $r = p = 10$ and $r = 10, p = 12$. Note that all sensors of the QD method are located at the same positions as those of the DG method in all cases. Fig. 7 shows that the sensor position of the DG method is the same as that of the QR method in the case of $p \leq r$, which confirms what is explained in Appendix B. On the other hand, Fig. 8 shows that the greedy added sensor positions of the QR and QD methods differ from each other in the case of $p > r$. The sensor positions of the QR and DG
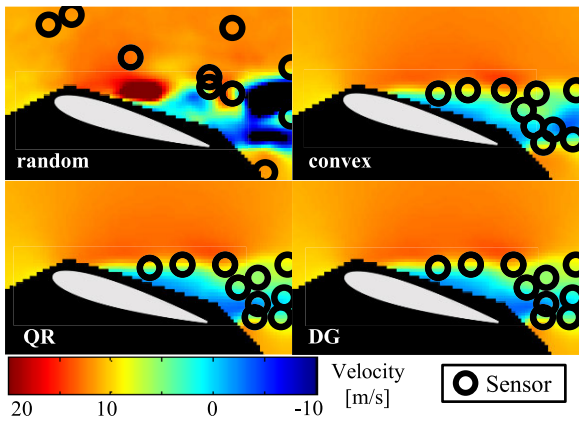
methods in Fig. 8 are the same as the first 12 sensor positions of the QR and DG methods in the case of $p > 12$ because the QR and DG methods are both greedy algorithms. The quality of the sensors is evaluated by considering the estimation error. First, the estimation error in the case in which the training and validation data are the same as each other is discussed.

Fig. 9 shows the relationship between the estimation error and the number of sensors for the number of POD modes $r = 10$ in the PIV problem. The estimation errors of the QR, DG, and QD methods decrease as the number of sensors increases, but the error of the convex approximation method increases in the case of $p = 7$–10. Although the estimation error of the random selection is averaged over 1000 trials, all the values are outside the range shown in the vertical axis of the figure. The estimation errors of the QR, DG, and QD methods coincide in the case of $p \leq r$ because they give the same set of sensor locations. The estimation errors of the DG and QD methods remain the same as each other in the case of $p > r$, as in the random sensor problem. Although the estimation error of the QR method also decreases as the number of sensors increases, it provides inferior sensor positions compared to the DG or QD method.

The estimation error for the case in which the training and validation data are different is discussed, which seems to be a more practical situation. A K-fold cross-validation ($K = 5$) is conducted and the performances of algorithms are compared. The same PIV data set is used for the cross-validation. The 500 snapshots are partitioned into five segments. Each of the five data segments is used as the validation data with the remaining data used for training and the estimation error is evaluated. Fig. 10 shows the results of this K-fold cross-validation ($K = 5$) for a varying number of sensors p. The error bars represent the standard deviations of the five calculations, but the error bars of the random selection and convex approximation methods are omitted for easy understanding of the figure. Fig. 10 shows the same

FIGURE 7. Sensor positions and reconstructed flowfield of a single snapshot from the sensors for $r = p = 10$ in the PIV sensor problem. Here, only the lateral components of the velocities are used for this problem.
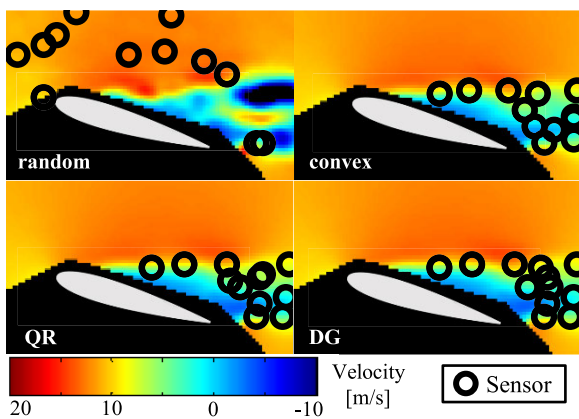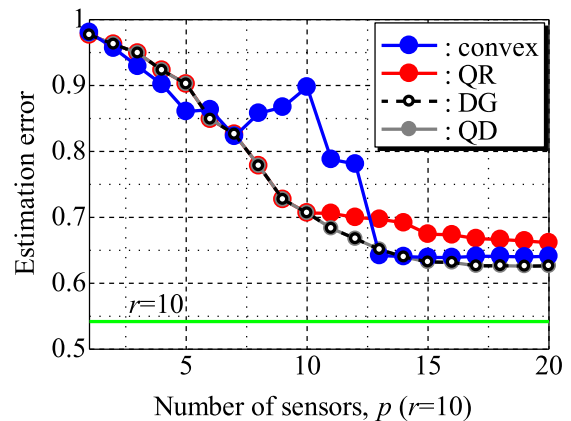

FIGURE 8. Sensor positions and reconstructed flowfield of a single snapshot for the sensors $r = 10, p = 12$ in the PIV sensor problem. Here, only the lateral components of the velocities are used for this problem.

FIGURE 9. Estimation error against the number of sensors for the number of POD modes $r = 10$ in the PIV sensor problem when the validation data are the same as the training data. Here, only the lateral components of the velocities are used for this problem.
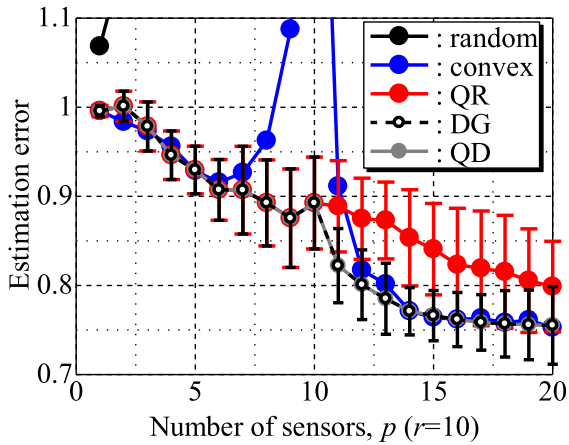
FIGURE 10. *K*-fold cross-validation results of estimation error against the number of sensors for the number of POD modes *r* = 10 in the PIV sensor problem. Here, only the lateral components of the velocities are used for this problem.
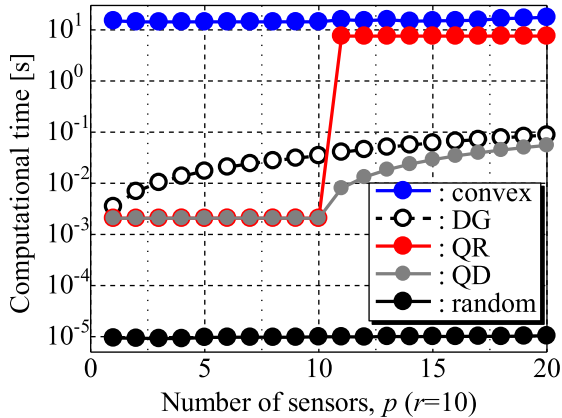

FIGURE 11. Computational time against the number of sensors for the number of POD modes *r* = 10 in the PIV sensor problem. Here, only the lateral components of the velocities are used for this problem.

trend as that shown in Fig. 9, which generalizes that the characteristics of the sensors selected by each algorithm do not change when we look at more practical situations in the PIV problem.

Fig. 11 shows the computational time of each method for selecting sensors for the number of sensors $p = 1$–20 in the PIV problem. Although the values of the random selection, DG, QR, and QD methods plotted in Fig. 11 are averaged over 1000 calculations, those of the convex approximation method are averaged over 200 calculations due to the huge computational cost. The trend in Fig. 11 is the same as that in Fig. 4. These results show that the proposed method, referred to as the QD method herein, has the best properties in terms of shorter computational time and a more favorable sensor selection when the numbers of sensors and POD modes are $p = 1$–20 and $r = 10$, respectively.

C. NOAA-SST SENSOR PROBLEM

The data set that we finally adopt is the NOAA OISST (NOAA-SST) V2 mean sea surface temperature set, comprising weekly global sea surface temperature measurements in the years between 1990 and 2000. The data are publicly available online [27]. There are a total of 520 snapshots on a 360×180 spatial grid (Fig. 12). The sparse sensor problem for NOAA-SST is solved using the same computational environments in Table 3 as those used in the problems of the random sensor and PIV. Fig. 13 shows the relationship between the estimation error and the number of POD modes. In the case of NOAA-SST, the POD modes are truncated and the $r = 10$ low-dimensional representation is obtained. The minimum estimation error is 0.30 for $r = 10$ as Fig. 13 shows. We consider the results of the sensor selection problem (the number of sensors $p = 10$–20) solved by each method: the random selection, convex approximation [13], QR [7], DG, and QD methods.

Figs. 14 and 15 show the sensor positions selected by the random selection, convex approximation, QR, and QD methods in the cases of, respectively, $r = p = 10$ and $r = 10$, $p = 15$. All sensors of the QD method are located at the same
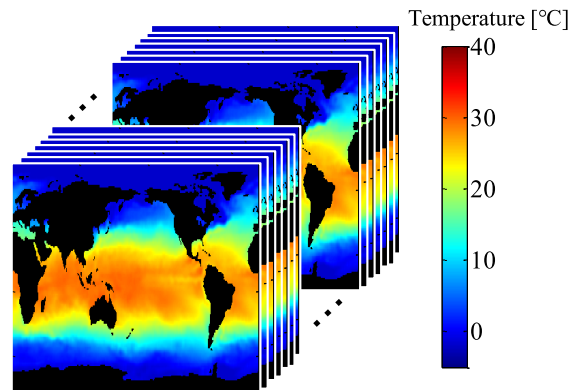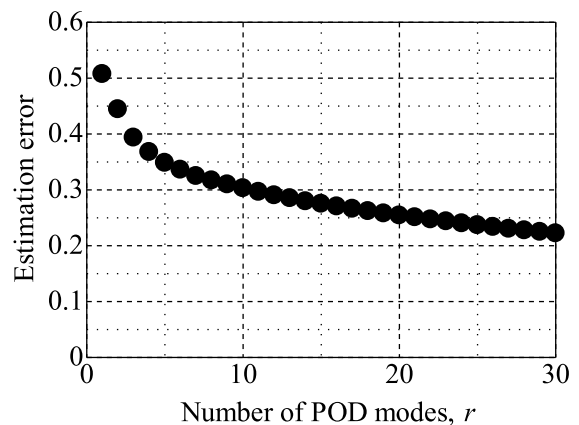

FIGURE 12. Dataset of NOAA-SST.

FIGURE 13. Estimation error against the number of POD modes for the NOAA-SST sensor problem.
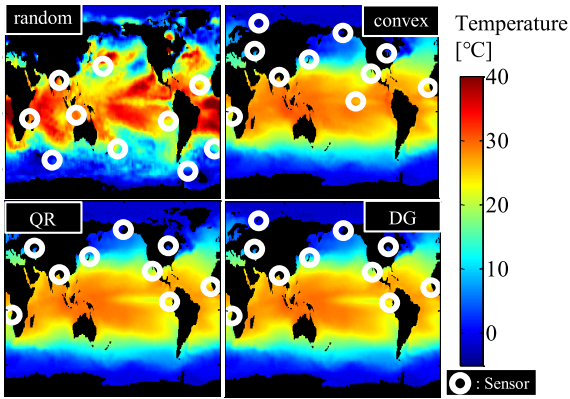
FIGURE 14. Sensor positions and reconstructed temperature field of a single snapshot from the sensors for $r = p = 10$ in the NOAA-SST sensor problem.
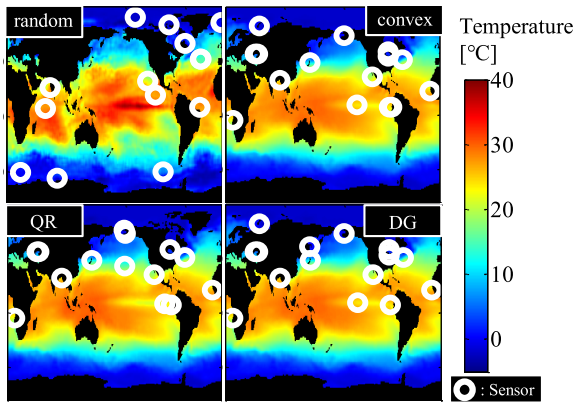

FIGURE 15. Sensor positions and reconstructed temperature field of a single snapshot from the sensors for $r = 10$, $p = 15$ in the NOAA-SST sensor problem.
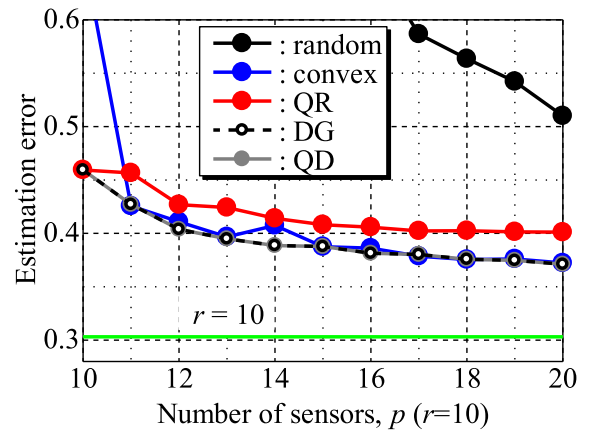

FIGURE 16. Estimation error against the number of sensors for the number of POD modes $r = 10$ in the NOAA-SST sensor problem when the validation data are the same as the training data.
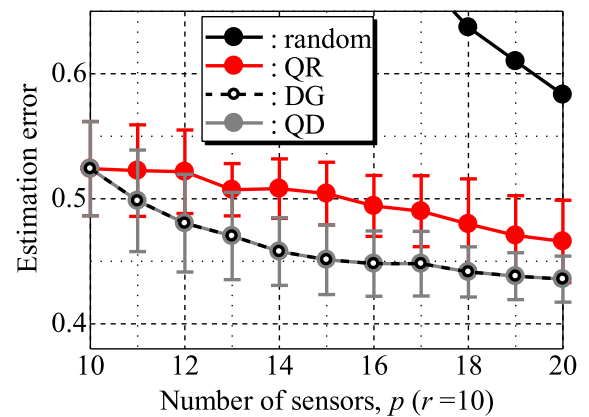

FIGURE 17. K-fold cross-validation results of estimation error against the number of sensors for the number of POD modes $r = 10$ in the NOAA-SST sensor problem.

positions as those of the DG method in all cases, as in the problems of the random sensor and PIV.

The quality of the sensors is evaluated by considering the estimation error. First, the estimation error in the case in which the training and validation data are the same as each other is discussed. Fig. 16 shows the relationship between the estimation error and the number of sensors for the number of POD modes $r = 10$ in the NOAA-SST problem. All estimation errors decrease as the number of sensors increases. The estimation errors of the QR, DG, and QD methods coincide in the case of $p = 10$ because they give the same set of sensor locations. The estimation errors of the DG and QD methods remain the same as each other in the case of $p > r$, as in the random sensor problem. Although the estimation error of the QR method also decreases as the number of sensors increases, it provides inferior sensor positions compared to the DG or QD method.

Next, the estimation errors for the case in which the training and validation data are different are considered, which seems to be a more practical situation. A K-fold cross-validation ($K = 5$) is conducted and the performances

of algorithms are compared. The same NOAA-SST data set is used for the cross-validation. The 520 snapshots are partitioned into five segments. Each of the five data segments is used as the validation data with the remaining data used for training and the estimation error is evaluated. Fig. 17 shows the results of this K-fold cross-validation ($K = 5$) for a varying number of sensors p. In the figure, error bars represent the standard deviations of the five calculations. The convex method is not included, because its computational time is very long in the NOAA-SST problem. Fig. 17 shows the same trend as that shown in Fig. 16, which generalizes that the characteristics of the sensors selected by each algorithm do not change when more practical situations are considered. Therefore, both the DG and QD methods in the case of $p > r$ are more accurate sensor selection methods than the QR method, in which the QR decomposition of UU^{T} is performed.

Fig. 18 shows the computational time of each method for selecting sensors for the number of sensors $p = 10$–20.
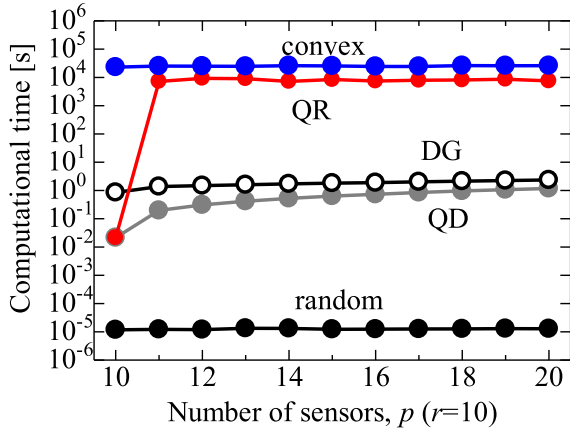
FIGURE 18. Computational time against the number of sensors for the number of POD modes $r = 10$ in the NOAA-SST sensor problem.

The computational time of the QR method has a large difference at $p = 11$ because the dimension of the matrix processed by the QR decomposition increases between the computation for $U \in \mathbb{R}^{n \times r}$ and that for $UU^{\mathrm{T}} \in \mathbb{R}^{n \times n}$. This trend is the same as that of the problems of the random sensor and PIV. On the other hand, the computational time of the DG method gradually increases as the number of sensors increases because each sensor is chosen using the step-by-step maximization of the determinant of $C_k^{\mathrm{T}} C_k$ in the case of $p > r$. Therefore, the complexities of the QR and DG/QD methods are $\mathcal{O}(n^3)$ and $\mathcal{O}(pnr^2)$ in the case of $p > r$, respectively. Hence, for the high-dimensional data such as the NOAA-SST sensor problem, the computational time of the QD method is much shorter than that of the QR method in the case of $p > r$. Although the trend in Fig. 18 is the same as that in Figs. 4 and 11, note that the computational time of the convex approximation method is approximately three hours, whereas that of the QD method, which gives a lower estimation error, is just a few seconds. These results show that the proposed method, referred to as the QD method herein, has the best properties in terms of shorter computational time and a more favorable sensor selection when the numbers of sensors and POD modes are $p = 10$–20 and $r = 10$, respectively. Based on the proposed determinant-based greedy algorithm, we are currently working on implementations that take noise into consideration [28] and reevaluate the objective functions [29].

V. CONCLUSION
Optimal sensor placement is an important challenge in the design, prediction, estimation, and control of high-dimensional systems. In this study, the sparse sensor placement problem is considered for least-squares estimation. First, the objective function of the problem is redefined for the maximization of the determinant of the matrix appearing in pseudo-inverse matrix operations, leading to the maximization of the corresponding confidence intervals. The procedure for the maximization of the determinant of the corresponding

matrix is proved to be mathematically equivalent to that of the previous QR method when the number of sensors is less than that of state variables. On the other hand, for the case that the number of sensors is greater than that of state variables, we developed a new algorithm then, a unified formulation is derived, and the lower bound of the objective function given by this algorithm is shown using its monotone submodularity. In the proposed algorithm, optimal sensors are obtained by the QR method until the number of sensors is equal to that of state variables, beyond which, new sensors are calculated by the proposed determinant-based greedy method which is accelerated by both a determinant formula and a matrix inversion lemma. The effectiveness of this algorithm on datasets related to the flowfield around an airfoil and the global climate is demonstrated in comparison to other algorithms. The computational time of the proposed extended determinant-based greedy algorithm is shown to be smaller than that of other methods with one of the smallest estimation errors. One weakness of the proposed algorithm is its robustness to noise, but it has been addressed and improved in [28]. The further improvement of the greedy method will be the subject of challenging future research. We have been developing a sparse-processing-particle-image-velocimetry (SPPIV)-measurement system [25]. In the SPPIV-measurement system, the amount of processing data is reduced and the flowfield is estimated by a limited number of sparsely located interrogation windows, leading to the real-time PIV measurement of the flowfield. The development of an appropriate sensor selection method is required for the highly accurate SPPIV, and the proposed algorithm combined with the vector extension [10] could be useful for that task. In addition to the SPPIV-measurement system, the proposed method, which can significantly reduce the computational time of selecting the sensor position compared to the previous studies, is suitable for application to particularly high-dimensional data such as combustion data.

APPENDIX A
SUMMARY OF PREVIOUS GREEDY ALGORITHM AND QR DECOMPOSITION IN THE OVERSAMPLING CASE
The QR algorithm is presented as in Alg. 3. The optimization is to maximize the determinant of the matrix C to stably solve for the x vector in the case of $p = r$. On the other hand, the validity of the algorithm in the oversampling case ($p > r$) was not well clarified in [7]. Here, the QR method in oversampling is considered. The random sensor-candidate matrices, $U \in \mathbb{R}^{1000 \times r}$, were set, where the component of the matrices is given by the Gaussian distribution $\mathcal{N}(0, 1)$ and $r = 10$. The QR decomposition is carried out as follows:

$$QR = UU^{\mathrm{T}}, \tag{29}$$

where $Q \in \mathbb{R}^{1000 \times 1000}$ is an orthogonal matrix (its columns are orthogonal unit vectors) and $R \in \mathbb{R}^{1000 \times r}$ is an upper triangular matrix. Fig. 19 shows the residual error, e, against the number of ranks, r'. Here, the residual error expresses the

Algorithm 3 QR Algorithm for Sparse Sensor Placement

Set sensor-candidate matrix U.
Set number of sensors $p \geq r$.
if $p = r$ **then**
   $V = U$
else if $p > r$ **then**
   $V = UU^T$
end if
for $k = 1, \ldots, p$ **do**
   $v_i = [\, V_{i1} \;\; V_{i2} \;\ldots\; V_{ir} \,]$
   $i \leftarrow \underset{i \in \mathcal{S} \setminus \mathcal{S}_k}{arg\ max} \|v_i\|_2^2$
   $w_k \leftarrow v_i$
   $V \leftarrow V - V w_k^T w_k / \|w_k\|_2^2$
   $H_{k,i} = 1$
end for

following equation:

$$e = \frac{\|QR - Q'R'\|_F}{\|QR\|_F},\qquad (30)$$

where Q' and R' denote the matrices that consist of the first r' columns of Q and the first r' rows of R, respectively, and the subscript F expresses the Frobenius norm. Fig. 19 shows that the reconstruction with the QR decomposition of $r' = r$ is almost the same as the original matrix UU^T. This is because the rank of UU^T is mathematically r and the residual error of $r' \geq r = 10$ is the computationally generated machine epsilon $\left(O\left(10^{-16}\right)\right)$ in double-precision decimals. Therefore, the QR decomposition works only for the residual of the machine epsilon level in the oversampling case. However, the QR method selects worse and better sensor positions than the present method and the random method, respectively, as explained in Appendix C. This is because the QR method selects the sensor positions where the residual of the machine epsilon level is large (the sensor position where the value of $U^T U$ is possibly large). As a result, it accidentally selects
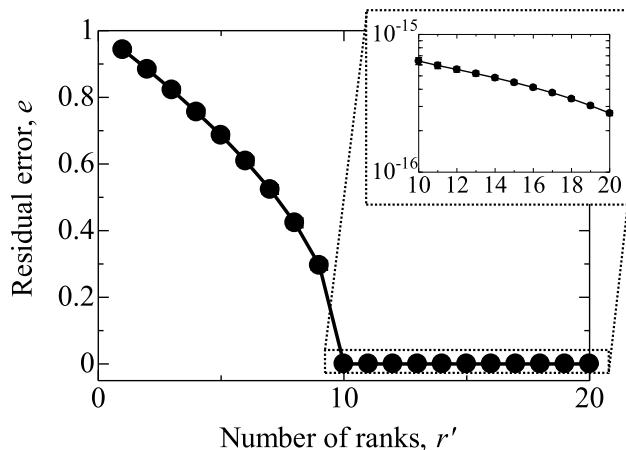

FIGURE 19. Residual error against the number of ranks at the implementation of QR decomposition for UU^T.

somehow reasonable sensor positions rather than randomized positions, but it is not optimized for the sensor selection problem in the oversampling case.

APPENDIX B
EQUIVALENCE OF OPERATIONS IN THE QR AND PRESENT METHODS FOR $k \leq r$

The proof of Theorem 1 is given as follows.

Proof: We start by deriving a mathematical expression of the QR method. The first step is to find a row vector of U with the largest norm. Suppose that $w_1 \in \mathbb{R}^{1 \times r}$ is such a vector and that it is in the i_1th row of U. Namely, $w_1 = u_{i_1}$. In the next step, we subtract components parallel to w_1 from every row vector of U. This operation is represented as

$$U\left(I - \frac{w_1^T w_1}{\|w_1\|^2}\right) =: U P_1^{QR}. \qquad (31)$$

Note that the matrix $P_1^{QR} \in \mathbb{R}^{r \times r}$ satisfies $P_1^{QR} = \left(P_1^{QR}\right)^T$ and $(P_1^{QR})^2 = P_1^{QR}$. Hence, P_1^{QR} is a symmetric projection matrix. We then seek a row vector $w_2 \in \mathbb{R}^{1 \times r}$ of $U P_1^{QR}$ with the largest norm. Let us suppose that $w_2 = v_{i_2}$, which is the i_2th row vector of $U P_1^{QR}$. Clearly, w_2 satisfies $w_2 = u_{i_2} P_1^{QR}$ and $w_2 w_1^T = 0$. As in the previous steps, we consider the matrix

$$U P_1^{QR}\left(I - \frac{w_2^T w_2}{\|w_2\|^2}\right) =: U P_2^{QR} \qquad (32)$$

and find a row vector of $U P_2^{QR}$ with the largest norm. Since $w_2 w_1^T = 0$, we have $w_2^T w_2 P_1^{QR} = P_1^{QR} w_2^T w_2$. This yields

$$P_2^{QR\,T} = \left(I - \frac{w_2^T w_2}{\|w_2\|^2}\right) P_1^{QR} = P_1^{QR}\left(I - \frac{w_2^T w_2}{\|w_2\|^2}\right) = P_2^{QR},$$

$$P_2^{QR\,2} = P_1^{QR}\left(I - \frac{w_2^T w_2}{\|w_2\|^2}\right) P_1^{QR}\left(I - \frac{w_2^T w_2}{\|w_2\|^2}\right)$$

$$= P_1^{QR}\left(I - \frac{w_2^T w_2}{\|w_2\|^2}\right) = P_2^{QR}. \qquad (33)$$

Thus, P_2^{QR} is also a symmetric projection. In this way, we can define a sequence of symmetric projection matrices $\{P_k^{QR}\}_{k=0}^r$ by

$$P_k^{QR} := P_{k-1}^{QR}\left(I - \frac{w_k^T w_k}{\|w_k\|^2}\right), \quad k = 1, 2, \ldots, r, \quad (34)$$

where $P_0^{QR} = I$ and $w_k \in \mathbb{R}^{1 \times r}$ is the vector selected in the kth step of the QR method. Suppose that w_k is the i_kth row vector, v_{i_k}, of $U P_{k-1}^{QR}$. Then, we have $w_k = v_{i_k} = u_{i_k} P_{k-1}^{QR}$.

Now, notice that the squared 2-norm of $w_k = v_{i_k}$ is given by $\|w_k\|^2 = u_{i_k} P_{k-1}^{QR} u_{i_k}^T$. Hence, w_k can be characterized as

$$w_k = v_{i_k} = u_{i_k} P_{k-1}^{QR},$$
$$i_k = \underset{i}{arg\ max}\ u_i P_{k-1}^{QR} u_i^T. \qquad (35)$$

In the DG method explained in Alg. 1, we choose a row vector of U that maximizes a quadratic form involving matrix C. This fact motivates us to relate this matrix to P_k^{QR}.

In the remainder of the proof, we show that $P_k^{QR} = I - C_k^T (C_k C_k^T)^{-1} C_k$, where

$$C_k = \begin{bmatrix} u_{i_1}^T & u_{i_2}^T & \cdots & u_{i_k}^T \end{bmatrix}^T. \tag{36}$$

Since $w_1 = u_{i_1}$, we have $C_1 = u_{i_1}$. It immediately follows that

$$I - C_1^T(C_1 C_1^T)^{-1} C_1 = I - \frac{u_{i_1}^T u_{i_1}}{u_{i_1} u_{i_1}^T} = I - \frac{w_1^T w_1}{\|w_1\|^2} = P_1^{QR}. \tag{37}$$

To prove our claim by induction, we assume that $P_{k-1}^{QR} = I - C_{k-1}^T (C_{k-1} C_{k-1}^T)^{-1} C_{k-1}$. Under this assumption, we represent the matrix $I - C_k^T (C_k C_k^T)^{-1} C_k$ in terms of P_{k-1}^{QR}. The second term can be written as

$$C_k^T \left(C_k C_k^T \right)^{-1} C_k$$
$$= \begin{bmatrix} C_{k-1}^T & u_{i_k}^T \end{bmatrix} \begin{bmatrix} C_{k-1} C_{k-1}^T & C_{k-1} u_{i_k}^T \\ u_{i_k} C_{k-1}^T & u_{i_k} u_{i_k}^T \end{bmatrix}^{-1} \begin{bmatrix} C_{k-1} \\ u_{i_k} \end{bmatrix}. \tag{38}$$

Let $\delta \in \mathbb{R}$ be the Schur complement of the block square matrix in the middle of the right-hand side of (38). That is,

$$\delta = u_{i_k} u_{i_k}^T - u_{i_k} C_{k-1}^T (C_{k-1} C_{k-1}^T)^{-1} C_{k-1} u_{i_k}^T$$
$$= u_{i_k} P_{k-1}^{QR} u_{i_k}^T$$
$$= \|w_k\|^2. \tag{39}$$

Then, by a property of the inverses of block matrices,

$$\begin{bmatrix} C_{k-1} C_{k-1}^T & C_{k-1} u_{i_k}^T \\ u_{i_k} C_{k-1}^T & u_{i_k} u_{i_k}^T \end{bmatrix}^{-1}$$
$$= \begin{bmatrix} Q_{k-1}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\delta} \begin{bmatrix} Q_{k-1}^{-1} C_{k-1} u_{i_k}^T \\ -1 \end{bmatrix} \begin{bmatrix} u_{i_k} C_{k-1}^T Q_{k-1}^{-1} & -1 \end{bmatrix}, \tag{40}$$

where $Q_{k-1} := C_{k-1} C_{k-1}^T$. Substituting this result into (38) gives

$$C_k^T \left(C_k C_k^T \right)^{-1} C_k = I - P_{k-1}^{QR} + \frac{1}{\delta} P_{k-1}^{QR} u_{i_k}^T u_{i_k} P_{k-1}^{QR}. \tag{41}$$

This relation readily shows that

$$I - C_k^T \left(C_k C_k^T \right)^{-1} C_k = P_{k-1}^{QR} - \frac{1}{\delta} P_{k-1}^{QR} u_{i_k}^T u_{i_k} P_{k-1}^{QR}$$
$$= P_{k-1}^{QR} - \frac{1}{\delta} \left(P_{k-1}^{QR} \right)^2 u_{i_k}^T u_{i_k} P_{k-1}^{QR}$$
$$= P_{k-1}^{QR} \left(I - \frac{1}{\delta} P_{k-1}^{QR} u_{i_k}^T u_{i_k} P_{k-1}^{QR} \right)$$
$$= P_{k-1}^{QR} \left(I - \frac{w_k^T w_k}{\|w_k\|^2} \right)$$
$$= P_k^{QR}. \tag{42}$$

Thus, mathematical induction guarantees that $P_k^{QR} = I - C_k^T \left(C_k C_k^T \right)^{-1} C_k$ for any $k \in \{1, 2, \ldots, r\}$. This relation means that, for each $k \in \{1, 2, \ldots, r\}$, w_k is obtained by maximizing the quadratic form

$$u_i \left(I - C_{k-1}^T (C_{k-1} C_{k-1}^T)^{-1} C_{k-1} \right) u_i^T \tag{43}$$

with respect to i. This is what is described in Alg. 1. Hence, the QR method coincides with our DG method for $k \leq r$. $\square$

APPENDIX C
COMPARISON TO PREVIOUS STUDIES IN OVERSAMPLING CASE

Previous studies have introduced several methods for the sensor (sampling point) selection problem under oversampling ($p > r$). In the QR method [7], sensor selection is conducted by maximizing the determinant of the matrix CC^T in the case of $p > r$. The calculation algorithm for the condition $p > r$ has already been proposed in [7] (which corresponds to the use of $V = UU^T$), but the validity of the algorithm was not well clarified in [7]. The "GappyPOD+R" method uses the QR and random selection methods for $p \leq r$ and $p > r$, respectively. Comparison between the QR, GappyPOD+R, and random selection methods will help us to evaluate the performance of the QR method in the oversampling case. This "GappyPOD+R" method is exactly the same as the "GappyPOD+R" method in [16], [30].

Peherstorfer *et al.* proposed a sensor selection method which is based on the lower bounds of the smallest eigenvalues of certain structured matrix updates in the oversampling case, which is referred to as the "GappyPOD+E" method [16], [30]. It should be noted that this second method performed the best among the oversampling-point selection methods that they compared [16].

Fig. 20 shows the relationship between the estimation error and the number of sensors for the number of POD modes $r = 10$ in the NOAA-SST problem for the oversampling case ($p > r$) when the training and validation data are the same
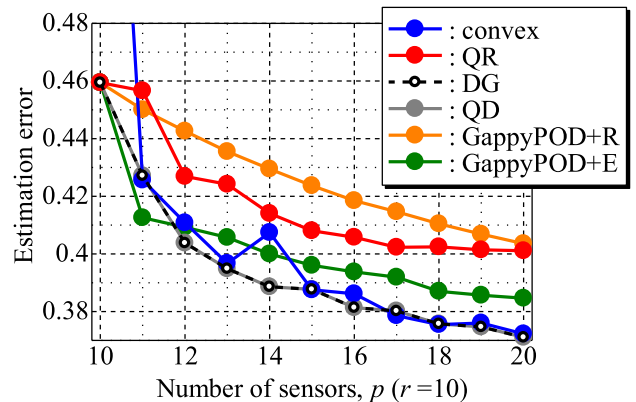

FIGURE 20. Estimation error against the number of sensors for the number of POD modes *r* = 10 in the NOAA-SST problem.

as each other. The estimation error of the GappyPOD+R method (solid orange line with closed circles) lies between those of the QR and random selection methods (see Fig. 16). The QR method provides better sensor selection than the random selection method although the former has a huge computational cost in the oversampling case, as shown in Fig. 18. The estimation error of the GappyPOD+E method (solid green line with closed circles) is less than that of the QR method and is the smallest of all the methods in the case of $p = 11$. However, the method proposed in this study, the DG method, performs better sensor selection in the oversampling case except for $p = 11$ than does the GappayPOD+E method, which is almost the best method ever proposed among oversampling point selection methods in the DEIM framework.

REFERENCES

[1] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annu. Rev. Fluid Mech.*, vol. 25, no. 1, pp. 539–575, Jan. 1993.

[2] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, "Modal analysis of fluid flows: An overview," *AIAA J.*, vol. 55, no. 12, pp. 4013–4041, Dec. 2017.

[3] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech.*, vol. 656, pp. 5–28, Aug. 2010.

[4] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode 6 Decomposition: Data-Driven Modeling of Complex Systems*, vol. 149. Philadelphia, PA, USA: SIAM, 2016.

[5] T. Nonomura, H. Shibata, and R. Takaki, "Dynamic mode decomposition using a Kalman filter for parameter estimation," *AIP Adv.*, vol. 8, no. 10, Oct. 2018, Art. no. 105106.

[6] T. Nonomura, H. Shibata, and R. Takaki, "Extended-Kalman-filter-based dynamic mode decomposition for simultaneous system identification and denoising," *PLoS ONE*, vol. 14, no. 2, pp. 1–46, 2019.

[7] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns," *IEEE Control Syst.*, vol. 38, no. 3, pp. 63–86, Jun. 2018.

[8] E. Clark, T. Askham, S. L. Brunton, and J. N. Kutz, "Greedy sensor placement with cost constraints," *IEEE Sensors J.*, vol. 19, no. 7, pp. 2642–2656, Apr. 2019.

[9] K. Manohar, J. N. Kutz, and S. L. Brunton, "Optimal sensor and actuator selection using balanced model reduction," 2018, *arXiv:1812.01574*. [Online]. Available: http://arxiv.org/abs/1812.01574

[10] Y. Saito, T. Nonomura, K. Nankai, K. Yamada, K. Asai, Y. Sasaki, and D. Tsubakino, "Data-driven vector-measurement-sensor selection based on greedy algorithm," *IEEE Sensors Lett.*, vol. 4, no. 7, Jul. 2020, Art. no. 7002604.

[11] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM J. Sci. Comput.*, vol. 32, no. 5, pp. 2737–2764, Jan. 2010.

[12] Z. Drmač and S. Gugercin, "A new selection operator for the discrete empirical interpolation method–improved a priori error bound and extensions," *SIAM J. Sci. Comput.*, vol. 38, no. 2, pp. A631–A648, Jan. 2016.

[13] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[14] T. Nonomura, S. Ono, K. Nakai, and Y. Saito, "Randomized subspace Newton convex method applied to data-driven sensor selection problem," *IEEE Signal Process. Lett.*, vol. 28, pp. 284–288, 2021.

[15] T. Nagata, T. Nonomura, K. Nakai, K. Yamada, Y. Saito, and S. Ono, "Data-driven sparse sensor selection based on A-optimal design of experiment with ADMM," *IEEE Sensors J.*, early access, Apr. 19, 2021, doi: 10.1109/JSEN.2021.3073978.

[16] B. Peherstorfer, Z. Drmač, and S. Gugercin, "Stability of discrete empirical interpolation and gappy proper orthogonal decomposition with randomized and deterministic sampling points," *SIAM J. Sci. Comput.*, vol. 42, no. 5, pp. A2837–A2864, Jan. 2020.

[17] P. Astrid, S. Weiland, K. Willcox, and T. Backx, "Missing point estimation in models described by proper orthogonal decomposition," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2237–2251, Nov. 2008.

[18] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations," *Comp. Rendus Mathematique*, vol. 339, no. 9, pp. 667–672, Nov. 2004.

[19] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, "The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows," *J. Comput. Phys.*, vol. 242, pp. 623–647, Jun. 2013.

[20] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[22] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions," *Math. Program.*, vol. 14, no. 1, pp. 265–294, Dec. 1978.

[23] Y. Saito. *Determinant-Based-Fast-Greedy-Sensor-Selection-Algorithm*. Accessed: Mar. 2021. [Online]. Available: https://github.com/Aerodynamics-Lab/Determinant-based-Fast-Greedy-Sensor-Selection-Algorithm

[24] M. Gu and S. C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing QR factorization," *SIAM J. Sci. Comput.*, vol. 17, no. 4, pp. 848–869, Jul. 1996.

[25] N. Kanda, K. Nakai, Y. Saito, T. Nonomura, and K. Asai, "Feasibility study on real-time observation of flow velocity field by sparse processing particle image velocimetry," *Trans. Jpn. Soc. Aeronaut. Space Sci.*, to be published.

[26] K. Nankai, Y. Ozawa, T. Nonomura, and K. Asai, "Linear reduced-order model based on PIV data of flow field around airfoil," *Trans. Jpn. Soc. Aeronaut. Space Sci.*, vol. 62, no. 4, pp. 227–235, 2019.

[27] NOAA/OAR/ESRL. *NOAA Optimal Interpolation (OI) Sea Surface Temperature (SST) v2*. Accessed: Jul. 2019. [Online]. Available: https://www.esrl.noaa.gov/psd/data/gridded/data.noaa.oisst.v2.html

[28] K. Yamada, Y. Saito, K. Nankai, T. Nonomura, K. Asai, and D. Tsubakino, "Fast greedy optimization of sensor selection in measurement with correlated noise," *Mech. Syst. Signal Process.*, vol. 158, Sep. 2021, Art. no. 107619.

[29] K. Nakai, K. Yamada, T. Nagata, Y. Saito, and T. Nonomura, "Effect of objective function on data-driven greedy sparse sensor optimization," *IEEE Access*, vol. 9, pp. 46731–46743, 2021.

[30] E. Clark, S. L. Brunton, and J. N. Kutz, "Multi-fidelity sensor selection: Greedy algorithms to place cheap and expensive sensors with cost constraints," *IEEE Sensors J.*, vol. 21, no. 1, pp. 600–611, Jul. 2020.

YUJI SAITO received the B.S. degree in mechanical engineering, and the Ph.D. degree in mechanical space engineering from Hokkaido University, Sapporo, Japan, in 2014 and 2018, respectively. He is currently an Assistant Professor of aerospace engineering with Tohoku University, Sendai, Japan.
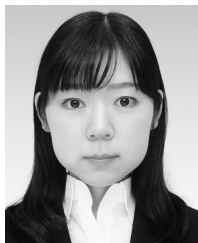
TAKU NONOMURA received the B.S. degree in mechanical and aerospace engineering from Nagoya University, Nagoya, Japan, in 2003, and the Ph.D. degree in aerospace engineering from The University of Tokyo, Tokyo, Japan, in 2008. He is currently an Associate Professor of aerospace engineering with Tohoku University, Sendai, Japan.

KEIGO YAMADA received the B.S. degree in physics from Tohoku University, Sendai, Japan, in 2019, where he is currently pursuing the M.S. degree.

KEISUKE ASAI received the bachelor's degree in aeronautical engineering from Kyoto University, Japan, in 1980, and the Ph.D. degree from The University of Tokyo, in 1995. From 1980 to 2003, he worked with the National Aerospace Laboratory (now, Aeronautical Directorate of JAXA). He is currently a Professor of aerospace engineering with Tohoku University, Sendai, Japan.

KUMI NAKAI received the Ph.D. degree in mechanical systems engineering from the Tokyo University of Agriculture and Technology, Japan, in 2020. From 2017 to 2020, she was a Research Fellow of the Japan Society for the Promotion of Science (JSPS), Tokyo University of Agriculture and Technology, Japan. Since 2020, she has been a Postdoctoral Researcher with Tohoku University, Japan. Her research interests include data-driven science, discharge plasma dynamics, fluid dynamics, and flow control utilizing atmospheric pressure plasma.

YASUO SASAKI received the B.S. degree from Nagoya University, Nagoya, Japan, in 2017, where he is currently pursuing the Ph.D. degree in engineering.

TAKAYUKI NAGATA received the B.S. and M.S. degrees in mechanical and aerospace engineering from Tokai University, Japan, in 2015 and 2017, respectively, and the Ph.D. degree in aerospace engineering from Tohoku University, Japan, in 2020. From 2018 to 2020, he was a Research Fellow of the Japan Society for the Promotion of Science (JSPS), Tohoku University. He is currently a Postdoctoral Researcher with Tohoku University.

DAISUKE TSUBAKINO (Member, IEEE) received the B.S. degree from Nagoya University, Nagoya, Japan, in 2005, and the Ph.D. degree in information science and technology from The University of Tokyo, Tokyo, Japan, in 2011. He is currently a Lecturer with Nagoya University.

• • •