2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Optimal sensor selection via proximal optimization algorithms

Armin Zare and Mihailo R. Jovanović

***Abstract*— We consider the problem of optimal sensor selection in large-scale dynamical systems. To address the combinatorial aspect of this problem, we use a suitable convex surrogate for complexity. The resulting non-convex optimization problem fits nicely into a sparsity-promoting framework for the selection of sensors in order to gracefully degrade performance relative to the optimal Kalman filter that uses all available sensors. Furthermore, a standard change of variables can be used to cast this problem as a semidefinite program (SDP). For large-scale problems, we propose a customized proximal gradient method that scales better than standard SDP solvers. While structural features complicate the use of the proximal Newton method, we investigate alternative second-order extensions using the forward-backward quasi-Newton method.**

***Index Terms*— Convex optimization, proximal methods, sensor selection, semidefinite programming, sparsity-promoting estimation and control, quasi-Newton methods.**

I. Introduction

An important challenge in the active control of distributed parameter systems is the judicious placement of sensors and actuators. Due to structural constraints, this problem is often posed as the selection of a subset of sensor and actuators from a pre-specified configuration in order to achieve a certain level of performance in terms of observability and controllability. While the best performance is typically achieved when using all available sensors and actuators, this option may be infeasible from a computational or economical standpoint. In many applications, it is inevitable that we sacrifice performance in order to obtain a minimal set of sensors and actuators. Selecting an optimal set of sensors and actuators, however, can become a nontrivial task that involves an intractable combinatorial search.

The sensor and actuator selection problem has recently received attention, mainly due to the emergence of large-scale applications, for example in power networks, systems biology, and industrial cyber-physical systems. To address the combinatorial aspect of this problem, current work has mostly focused on employing heuristics based on two popular approaches: combinatorial greedy algorithms and convex relaxation. Combinatorial methods mainly rely on notions of submodularity/supermodularity in an attempt to provide near-optimality guarantees [1]–[3]. On the other hand, convex or non-convex methods have also been proposed for this problem, which do not utilize greedy algorithms [4]–[10]. For linear time-invariant (LTI) systems, we focus on the latter approach and consider the problem of selecting a subset of sensors in order to gracefully degrade performance relative to the optimal Kalman filter that utilizes all available sensors. Actuator selection for LTI systems can be similarly addressed via the dual formulation of the sensor selection problem.

In [11], [12], a sparsity-promoting framework was introduced to effectively obtain block-sparse structured feedback/observer gains as well as select actuators/sensors. However, this framework addresses a more general class of problems and does not exploit a certain hidden convexity of sensor/actuator selection. For the design of optimal row-sparse feedback gains, a convex characterization was proposed in [8]. Based on this formulation, a customized optimization algorithm was proposed in [9] for large-scale sensor/actuator selection. This algorithm used the Alternating Direction Method of Multiplier (ADMM) and exploited the problem structure to gain computational efficiency relative to standard semidefinite programming (SDP) solvers. However, the ADMM algorithm involves subproblems that are difficult to solve and is thus not well-suited for large-scale systems. Herein, we further exploit the problem structure and propose a customized proximal gradient (PG) algorithm that scales with the third power of the state-space dimension. We also address the problem of ill-conditioning by employing a second-order algorithm based on the forward-backward quasi-Newton method [13], which has been shown to improve accuracy and practical convergence properties.

Our presentation is organized as follows. In Section II, we state the sensor selection problem and introduce the change of variables which leads to the convex reformulation. In Section III, we present customized algorithms for solving this problem. In Section IV, we present results of numerical experiments. Finally, we conclude with a summary of results and future directions in Section V.

II. Problem formulation

Consider the linear time-invariant system

$$\begin{aligned} \dot{x} &= A\,x + B\,d \\ y &= C\,x + \eta \end{aligned}$$

where $x(t) \in \mathbb{C}^n$ is the state vector, $y(t) \in \mathbb{C}^p$ is the output, $d(t)$ and $\eta(t)$ are zero-mean white stochastic processes with covariances Ω_d and Ω_η, respectively, $B \in \mathbb{C}^{n\times m}$ is the input matrix, $C \in \mathbb{C}^{p\times n}$ is the output matrix with $n \leq p$, and (A, C) is an observable pair. The observer

$$\begin{aligned} \dot{\hat{x}} &= A\,\hat{x} + L\,(y - \hat{y}) \\ &= A\,\hat{x} + L\,C\,(x - \hat{x}) + L\,\eta \end{aligned}$$

provides an estimate $\hat{x}$ of the state x from noisy measurements y, and L is the observer gain. When $A - L\,C$ is

Financial support from the National Science Foundation under Award CMMI 1739243 and the Air Force Office of Scientific Research under Award FA9550-16-1-0009 is gratefully acknowledged.

A. Zare and M. R. Jovanović are with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089. E-mails: armin.zare@usc.edu, mihailo@usc.edu.

Hurwitz, the zero-mean estimate of x is given by $\hat{x}$, and the estimation error $\tilde{x} := x - \hat{x}$ follows the the dynamics

$$\dot{\tilde{x}} \;=\; (A \,-\, L\,C)\,\tilde{x} \;+\; B\,d \;-\; L\,\eta. \tag{1}$$

The Kalman filter gain L minimizes the $\mathcal{H}_2$ norm of $\tilde{x}$, i.e., variance amplification from process and measurement noise to estimation error, and can be obtained by solving

$$\begin{array}{ll} \underset{L,\,X}{\text{minimize}} & \text{trace}\,(X\,B\,\Omega_d\,B^* \;+\; X\,L\,\Omega_\eta\,L^*) \\ \text{subject to} & (A-L\,C)^*X \;+\; X(A-L\,C) \;+\; I \;=\; 0 \\ & X \;\succ\; 0. \end{array} \tag{2}$$

Here, A, B, Ω_d, and Ω_η are problem data, while $X = X^* \in \mathbb{C}^{n\times n}$ and $L \in \mathbb{C}^{n\times p}$ are optimization variables. The optimal solution to this problem can be obtained by solving the observer Riccati equation arising from the corresponding KKT conditions. In general, the optimal gain matrix L has no particular structure (i.e., L has no zero entries) and therefore all available measurements are used.

When the ith column of L is identically equal to zero, the ith measurement is not used. Therefore, designing a Kalman filter which uses a subset of the available sensors is equivalent to designing a column-sparse Kalman gain matrix L, which can be sought by promoting column-sparsity via the following regularized optimization problem

$$\begin{array}{ll} \underset{L,\,X}{\text{minimize}} & \text{trace}\,(V_d X \;+\; X\,L\,\Omega_\eta\,L^*) \;+\; \gamma \displaystyle\sum_{i\,=\,1}^{n} w_i \|L\,\mathrm{e}_i\|_2 \\ \text{subject to} & (A-L\,C)^*X \;+\; X(A-L\,C) \;+\; I \;=\; 0 \\ & X \;\succ\; 0. \end{array} \tag{3}$$

Here, $\gamma > 0$ specifies the importance of sparsity, w_i are nonzero weights, e_i is the ith unit vector in $\mathbb{R}^p$, and $V_d := B\,\Omega_d B^*$.

Remark 1: The dual formulation of the sensor selection problem (3) *addresses the actuator selection problem; see [9] for additional details.*

Since X is positive definite, the standard change of variables $Y := XL$ and the equivalence between the column-sparsity of L and the column-sparsity of Y [8] can be utilized to bring problem (3) into the following form

$$\begin{array}{ll} \underset{X,\,Y}{\text{minimize}} & \text{trace}\left(V_d X + X^{-1} Y\,\Omega_\eta Y^*\right) + \gamma \displaystyle\sum_{i\,=\,1}^{n} w_i \|Y\mathrm{e}_i\|_2 \\ \text{subject to} & A^*X \;+\; X\,A \;-\; Y\,C \;-\; C^*Y^* \;+\; I \;=\; 0 \\ & X \;\succ\; 0, \end{array} \tag{4}$$

which is SDP representable [14]. After solving this problem, the optimal gain matrix can be recovered as $L = X^{-1}Y$. The convexity of (4) follows from the convexity of its objective function and the convexity of the constraint set [15]. For large-size problems that cannot be handled by general-purpose SDP solvers, we next invoke proximal methods to develop first- and second-order algorithms.

Remark 2: The non-convex characterization (3) *can be directly addressed by the ADMM algorithm of [12] or a proximal gradient algorithm similar to what we present in Section III. However, the uncovered sparsity patterns can from what is achieved from solving the convex problem* (4)*.*

III. Customized algorithms

The objective function in problem (4) is composed of a differentiable and a non-differentiable component, which limits the utility of standard gradient descent. Problems of this form appear in many application fields, e.g., control, system identification, machine learning, and statistics. The proximal gradient (PG) method generalizes gradient descent to such composite minimization problems [16] and is most effective when the proximal operator associated with the non-differentiable component is easy to evaluate [17]. The global convergence rate of the PG method is known to be sublinear with a convergence rate of $1/k$ for convex problems. Similar to other first-order methods, PG is suitable for computing solutions with small to medium precision and suffers from issues that arise from ill-conditioning.

Second-order generalizations of PG, namely the proximal Newton method [18], [19], have also been proposed as a remedy to the aforementioned issues. However, the proximal mapping that is used in these methods corresponds to the generalized norm that is based on the Hessian of the smooth part of the objective function. As we show next, this distorts the structure of problem (4) and complicates the computation of the proximal operator. Instead, we employ a second-order algorithm based on the forward-backward quasi-Newton method, which has been shown to improve the accuracy and practical convergence of PG [13].

For notational compactness, we write the affine constraint in optimization problem (4) as

$$\mathcal{A}(X) \;-\; \mathcal{B}(Y) + I \;=\; 0 \tag{5}$$

with linear operators $\mathcal{A}$: $\mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ and $\mathcal{B}$: $\mathbb{C}^{n\times p} \to \mathbb{C}^{n\times n}$ defined as

$$\begin{aligned} \mathcal{A}(X) \;&:=\; A^*\,X \;+\; X\,A, \\ \mathcal{B}(Y) \;&:=\; Y\,C \;+\; C^*Y^*. \end{aligned}$$

In order to bring problem (4) into a form that is accessible to the standard PG method, we eliminate the variable X. For any Y, there is a unique X that solves Eq. (5) if and only if $\lambda_i + \bar{\lambda}_j \neq 0$ for all i and j, where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of the matrix A [20]. Herein, we assume that $\lambda_i + \bar{\lambda}_j \neq 0$ which allows us to express the variable X as a linear function of Y,

$$X(Y) \;=\; \mathcal{A}^{-1}\,(\mathcal{B}(Y) \;-\; I)\,, \tag{6}$$

and restate problem (4) as

$$\begin{array}{ll} \underset{Y}{\text{minimize}} & f(Y) \;+\; \gamma\,g(Y) \\ \text{subject to} & X(Y) \;\succ\; 0. \end{array} \tag{7}$$

Here,

$$f(Y) \;:=\; \mathrm{trace}\left(V_d\, X(Y) \;+\; X^{-1}(Y)\, Y\, \Omega_\eta\, Y^*\right) \quad (8a)$$

$$g(Y) \;:=\; \sum_{i=1}^{n} w_i \,\|Y\,\mathrm{e}_i\|_2. \quad (8b)$$

Optimization problem (7) is convex because it is equivalent to (4) constrained to the affine subspace defined by (6).

A. Proximal Newton-type methods

Proximal Newton-type methods consider a local quadratic approximation to model the curvature of function $f(Y)$ at each iteration k, i.e.,

$$Y^{k+1} \;:=\; \underset{Y}{\mathrm{argmin}}\; f(Y^k) \;+\; \left\langle \nabla f(Y^k), Y - Y^k \right\rangle \;+\; \frac{1}{2}\,\|Y \;-\; Y^k\|^2_{H_k} \;+\; \gamma\, g(Y),$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product $\langle M_1, M_2 \rangle := \mathrm{trace}(M_1^* M_2)$, and $\|\cdot\|_H$ is the generalized norm $\|V\|_H := \mathrm{trace}(V^* H(V))$. In the case of vector optimization, H is a symmetric positive-definite matrix that approximates the Hessian of f. When the variables are matrices, however, the Hessian denotes a linear operator $H\colon \mathbb{C}^{n\times p} \to \mathbb{C}^{n\times p}$.

A proximal Newton-type algorithm for solving problem (7) takes the following form:

$$Y^{k+1} \;:=\; \mathbf{prox}^{H_k}_{\beta_k g}\left(Y^k \;-\; \alpha_k\, H_k^{-1} \nabla f(Y^k)\right), \quad (9)$$

where $\beta_k = \alpha_k \gamma$, $\alpha_k > 0$ is the step-size, and $\mathbf{prox}^H_{\beta g}(\cdot)$ is the *scaled* proximal operator associated with the nonsmooth function g

$$\mathbf{prox}^H_{\beta g}(V) \;:=\; \underset{Y}{\mathrm{argmin}}\; g(Y) \;+\; \frac{1}{2\beta}\,\|Y \;-\; V\|^2_H. \quad (10)$$

This method has been successfully used for generalized sparse linear modeling [21] and sparse inverse covariance estimation [22]. In general, unless H_k has a very particular structure (e.g., diagonal or banded), computing the proximal mapping in (9) can become significantly involved and requires an inner iterative procedure. Moreover, for large problems, computing the Hessian is prohibitive. Nevertheless, there have been efforts to utilize quasi-Newton schemes to approximate the Hessian [19], [23].

In the sensor selection problem (7), the Hessian of the function f can be expressed analytically. However, the complexity of computing the corresponding scaled proximal mapping prohibits the efficient implementation of the Proximal Newton method for sensor selection. We next present the PG method, which is a special case of (9) when H_k is the identity operator. This allows for the separation of problem (10) over various columns of Y, and subsequently leads to a *closed-form expression* for the update (9).

B. Proximal gradient method

The PG method for solving problem (7) is given by

$$Y^{k+1} \;:=\; \mathbf{prox}_{\beta_k g}\left(Y^k \;-\; \alpha_k\, \nabla f(Y^k)\right), \quad (11)$$

where Y^k denotes the kth iterate of the algorithm, $\beta_k = \alpha_k \gamma$, $\alpha_k > 0$ is the step-size, and $\mathbf{prox}_{\beta g}(\cdot)$ is the proximal operator

$$\mathbf{prox}_{\beta g}(V) \;:=\; \underset{Y}{\mathrm{argmin}}\; g(Y) \;+\; \frac{1}{2\beta}\,\|Y \;-\; V\|^2_F, \quad (12)$$

with $\|\cdot\|_F$ being the Frobenius norm. This proximal operator is determined by the soft-thresholding operator which acts on the rows of matrix V,

$$\mathcal{S}_\beta(V\,\mathrm{e}_i) \;=\; \begin{cases} (1 \;-\; \beta/\|V\,\mathrm{e}_i\|_2)\, V\,\mathrm{e}_i, & \|V\,\mathrm{e}_i\|_2 \;>\; \beta\, w_i \\ 0, & \|V\,\mathrm{e}_i\|_2 \;\leq\; \beta\, w_i. \end{cases} \quad (13)$$

Note that step (11) results from a local quadratic approximation of f at iteration k, i.e.,

$$Y^{k+1} \;:=\; \underset{Y}{\mathrm{argmin}}\; f(Y^k) \;+\; \left\langle \nabla f(Y^k), Y - Y^k \right\rangle \;+\; \frac{1}{2\,\alpha_k}\,\|Y \;-\; Y^k\|^2_F \;+\; \gamma\, g(Y), \quad (14)$$

followed by a completion of squares that brings the problem into the form of (12). From (14) it is clear that PG is a special case of proximal Newton-type methods when H_k is the scaled identity operator, $H_k = (1/\alpha_k) I$. The expression for the gradient of $f(Y)$ used in (11) is provided in Appendix A.

At each iteration of PG, we determine the step-size α_k via an adaptive Barzilai-Borwein step-size selection [24] to ensure sufficient descent and positive definiteness of $X(Y^{k+1})$; see [16, Theorem 3.1] for details. We terminate the algorithm when the fixed-point residual of the proximal mapping

$$R_{\alpha_k}(Y^k) \;:=\; \frac{1}{\alpha_k}\left(Y^k \;-\; \mathbf{prox}_{\beta_k g}(Y^k - \alpha_k\, \nabla f(Y^k))\right) \quad (15)$$

is small enough.

We next follow the developments of [13] to solve problem (7) using the forward-backward quasi-Newton method. This method allows us to incorporate second-order information in the computation of our updates.

C. Forward-backward quasi-Newton method

The forward backward method of [13] can be viewed as a (variable-metric) gradient-based method for minimizing the forward-backward envelope (FBE). The FBE serves as a continuously differentiable, exact penalty function for the original nonsmooth optimization problem.

The FBE of the objective function in problem (7)

$$\varphi(Y) \;=\; f(Y) \;+\; \gamma\, g(Y)$$

is given by

$$\begin{aligned} \varphi_\alpha(Y) \;:=\; &\inf_V\; f(Y) \;+\; \langle \nabla f(Y), V - Y \rangle \;+\; \frac{1}{2\,\alpha}\|V \;-\; Y\|^2_F \;+\; \gamma\, g(V) \\ \;=\; &\; f(Y) \;-\; \frac{\alpha}{2}\|\nabla f(Y)\|^2_F \;+\; M_{\beta g}\left(Y - \alpha\, \nabla f(Y)\right), \end{aligned}$$

where $\beta = \alpha\gamma$, and $M_{\beta g}$ is the Moreau envelope associated

with the proximal operator of function g [17], i.e.,

$$M_{\beta g}(Y) \;:=\; \inf_{V}\; g(V) \;+\; \frac{1}{2\,\beta}\|Y \,-\, V\|_F^2.$$

The Moreau envelope is a continuously differentiable function, even though g is not, and its gradient is given by

$$\nabla M_{\beta g}(Y) \;=\; \frac{1}{\alpha}\left(Y \,-\, \mathbf{prox}_{\beta g}(Y)\right).$$

The FBE φ_α characterizes a partially linearized variant of the objective function φ on the manifold that corresponding to explicit minimization over the variable in the nonsmooth term g. The FBE is also continuously differentiable and its gradient is given by

$$\nabla\varphi_\alpha(Y) \;=\; R_\alpha(Y) \;-\; \alpha\,\nabla^2 f(Y, R_\alpha(Y)), \tag{16}$$

where $R_\alpha(Y)$ is defined in Eq. (15).

The MINFBE algorithm [13, Algorithm 1] for solving problem (7) follows a sequence of iterations in which the FBE is minimized,

$$W^{k+1} \;:=\; Y^k \;+\; \tau_k\,\Delta Y^k \tag{17a}$$

$$Y^{k+1} \;:=\; \mathbf{prox}_{\beta_k g}\left(W^{k+1} \;+\; \alpha_k\,\nabla f(W^{k+1})\right) \tag{17b}$$

where ΔY^k is chosen such that

$$\left\langle \Delta Y^k, \nabla\varphi_{\alpha_k}(Y^k)\right\rangle \;\leq\; 0.$$

In (17a) and (17b), the step-size α_k is initialized via an adaptive Barzilai-Borwein step-size selection technique and τ_k is initialized with 1. These step-sizes are then adjusted through backtracking to satisfy

$$\varphi_{\alpha_k}(W^k) \;\leq\; \varphi_{\alpha_k}(Y^k) \tag{18a}$$

$$f(Y^{k+1}) \;>\; f(Y^k) \,-\, \alpha_k\left\langle \nabla f(Y^k), R_{\alpha_k}(Y^k)\right\rangle \,+\, \frac{(1-\zeta)\alpha_k}{2}\|R_{\alpha_k}(Y^k)\|_F^2, \tag{18b}$$

respectively, and to maintain $X(Y^{k+1}) \succ 0$. The algorithm terminates when the fixed-point residual $R_{\alpha_k}(Y^k)$ is small enough. When f is convex, this method enjoys the same sublinear converge rate as the proximal gradient method; see [13] for additional details.

Since the Moreau envelope is continuously differentiable, gradient-based methods can be used to minimize the FBE. Herein, we employ the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method highlighted in [25, Algorithm 7.4] to estimate the Hessian inverse of φ_α and to compute the descent direction as $\Delta Y^k = -H_k^{-1}(\nabla\varphi_{\alpha_k})$. Our computational experiments show that this approach results in satisfactory performance for modest values of m (between 5-20). The expression for the Hessian $\nabla^2 f(Y, \tilde{Y})$ that is necessary for computing the gradient of the FBE in Eq. (16) is provided in Appendix B.

While BFGS is guaranteed to converge for convex functions with Lipschitz continuous gradients [26], L-BFGS requires strong convexity for guaranteed convergence [13]. Furthermore, the L-BFGS method can converge slowly for ill-conditioned problems in which the Hessian contains a wide distribution of eigenvalues [25]. An alternative approach would be to consider Hessian-free Newton methods in which the descent direction $-H_k^{-1}(\nabla\varphi_{\alpha_k})$ are computed directly via conjugate-gradient-based schemes. Implementation of such techniques is a subject of ongoing work.

D. Computational complexity

Computation of the gradient of f involves computation of the matrix X from Y based on (6), a matrix inversion, and solution to the Lyapunov equation, which each take $O(n^3)$ operations, as well as an $O(n^2p)$ matrix-matrix multiplication. In addition, the proximal operator for the function g amounts to computing the 2-norm of all p columns of a matrix with n rows, which takes $O(np)$ operations. These steps are embedded within an iterative backtracking procedure for selecting the step-size α. Thus, if the step-size selection takes q_1 inner iterations the total computation cost for a single iteration of the PG algorithm is $O(q_1 n^3)$. On the other hand, the forward-backward quasi-Newton method MINFBE involves an additional iterative backtracking procedure for selecting the step-size τ that can take q_2 inner iterations and computation of the Hessian of f that involves 2 solutions to the Lyapunov equation, each requiring $O(n^3)$ operations. Therefore, implementation of MINFBE takes $O(q_1 q_2 n^3)$ operations per iteration. In contrast, the worst-case complexity of standard SDP solvers is $O(n^6)$.

Remark 3: As aforementioned, if the spectrum of A has mirroring eigenvalues, i.e., $\lambda_i + \bar{\lambda}_j = 0$ for some i and j, the variable X cannot be uniquely expresses as a linear function of Y. While dualizing the relation between X and Y would avoid this issue, it would come at the cost of slower convergence resulting from higher computational complexity, e.g., $O(n^5)$ per iteration in the case of the ADMM algorithm of [9]. Section IV includes a comparison with this algorithm.

IV. An example

We use a mass-spring-damper (MSD) system with N masses to compare the performance of various optimization algorithms for sensor selection. After setting all masses, spring and damping constants to unity, the dynamics of the position and velocity of masses are governed by the following state-space representation

$$\dot{x} \;=\; A\,x \;+\; B\,d$$

where the state vector $x = [\,p^T \;\; v^T\,]^T$ contains the positions and velocity of masses, d represents a zero-mean unit variance white process. The input and state matrices are

$$A \;=\; \begin{bmatrix} O & I \\ -T & -I \end{bmatrix}, \quad B \;=\; \begin{bmatrix} 0 \\ I \end{bmatrix}$$

where O and I are zero and identity matrices of suitable size, and T is a Toeplitz matrix with 2 on the main diagonal and -1 on the first super- and sub-diagonals. Here, possible sensor measurements are the position and velocity of masses.

For $C = I$, $V_d = I$, and $\Omega_\eta = 10I$, we solve the sensor selection problem (4) for $N = 10, 20, 30, 40$ and

TABLE I

COMPARISON OF DIFFERENT ALGORITHMS (IN SECONDS) FOR DIFFERENT NUMBER OF STATES n AND $\gamma = 10$.

n	CVX	PG	MINFBE	ADMM
20	3	0.1	1.29	7.7
40	12	0.8	43	49.7
60	83.4	3.1	81.8	126.8
80	295	11.2	124.6	216.5
100	971	41.2	274.6	456.9

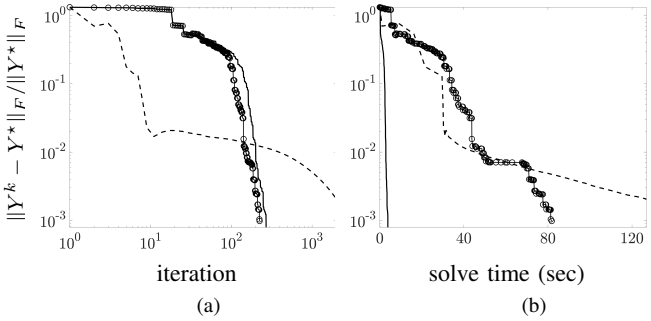

Fig. 1. Convergence curves showing performance of PG ($-$), MINFBE ($\bigcirc$), and ADMM ($--$) versus (a) the number of outer iterations; and (b) solve times for $N = 30$ masses and $\gamma = 10$. Here, $Y^\star$ is the optimal value for Y computed using CVX.

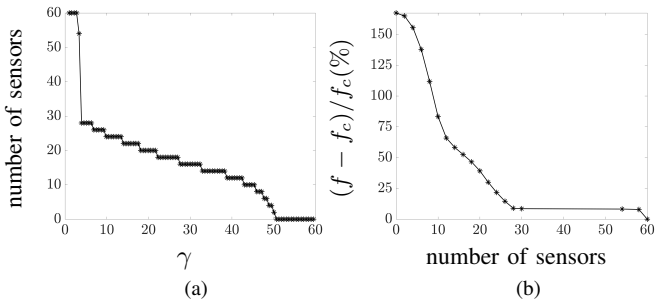

Fig. 2. (a) Number of retained sensors as a function of the sparsity-promoting parameter γ; and (b) performance loss vs the number of retained sensors for the MSD with $N = 30$ masses. Here, $f_c(Y)$ denotes the performance index in Eq. (8a) when all sensors are available ($\gamma = 0$).

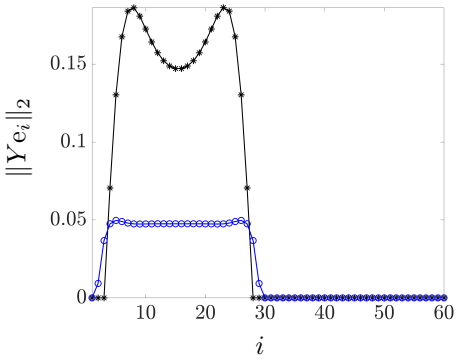

Fig. 3. Column norms of the solution of problems (4) ($*$) and (3) ($\bigcirc$) for the MSD system with $N = 30$ masses and $\gamma = 10$. The index i denotes the column number.

50 masses ($n = 2N$ states) and for various values of the regularization parameter γ. For $\gamma = 10$, Table I compares the proposed PG and MINFBE algorithms against CVX [27] and the ADMM of [9]. The algorithms were initialized with $Y^0 = X_c L_c$, where L_c and X_c solve the observer Riccati equation specifying the optimal Kalman filter that uses all output measurements. This choice guarantees that $X(Y^0) \succ 0$. All algorithms were implemented in MATLAB and executed on a 2.9 GHz Intel Core i5 processor with 16 GB RAM. They terminate when an iterate achieves a certain distance from optimality, i.e., $\|X^k - X^\star\|_F/\|X^\star\|_F < \epsilon$ and $\|Y^k - Y^\star\|_F/\|Y^\star\|_F < \epsilon$. The choice of $\epsilon = 10^{-3}$ guarantees that the value of the objective is within 0.01% of optimality. Clearly PG outperforms all other methods. Figure 1 shows convergence curves of PG, MINFBE, and ADMM for $N = 30$ masses and $\gamma = 10$.

From the convergence plots of Fig. 1 it is evident that compared to PG, MINFBE takes fewer iterations but significantly more computational time to reach the same level of accuracy. Even though the MINFBE method converges, its poor performance can be an indication that the descent direction computed via BFGS or L-BFGS are not reasonable approximations of the Newton descent direction.

For $N = 30$ masses, Fig. 2(a) shows the number of retained sensors as a function of γ and Fig. 2(b) shows the percentage of performance degradation as a function of the number of retained sensors. In problem (4), as γ increases sensors are dropped and the performance degrades relative to when all sensors are made available ($\gamma = 0$). For example Fig. 2(b) shows that by dropping 2 velocity sensors the performance degrades by 7.9% and that by dropping the remaining 28 velocity sensors the performance degradation only increases by another 0.6%. On the other hand, this figure illustrates the importance of position measurements; dropping half of the sensors dedicated to the position of masses results in a 54% performance loss. Finally, for $N = 30$ masses and $\gamma = 10$, Fig. 3 illustrates how the solution to the convex formulation of the sensor selection problem (4) can differ from that of the non-convex formulation (3).

V. CONCLUDING REMARKS

We provide customized algorithms for the convex characterization of the sensor selection problem in LTI systems. By exploiting the problem structure and implicitly handling the structural constraint on observability gramians and filter gains, we bring the problem into a form that is accessible to the proximal gradient method. We also employ the forward-backward quasi-Newton method, which utilizes second-order information to improve performance. Our numerical experiments demonstrate that the proximal gradient algorithm

outperforms the forward-backward quasi-Newton method, as well as a previously developed splitting method based on ADMM. Due to the importance of addressing the issue of ill-conditioning in the broader context of nonsmooth composite optimization (especially in the statistical modeling of complex dynamical systems [28]) our ongoing effort is directed towards improving the performance of the proposed second-order method via direct computation of the Newton direction. An alternative approach would also be to consider second-order primal-dual methods [29].

APPENDIX

A. Gradient of $f(Y)$ in Eq. (11)

To find $\nabla f(Y)$ in (11), we expand $f(Y + \epsilon \tilde{Y})$ around Y for the variation $\epsilon\tilde{Y}$, and collect terms of $O(\epsilon)$. We also account for the variation of X as a result of the variation of Y from

$$(X + \epsilon \tilde{X})^{-1} \;=\; X^{-1} \;-\; \epsilon\, X^{-1} \tilde{X}\, X^{-1} \;+\; O(\epsilon^2),$$

and the linear dependence of $\tilde{X}$ on $\tilde{Y}$, i.e.,

$$\tilde{X} \;=\; \mathcal{A}^{-1}(\mathcal{B}(\tilde{Y})).$$

Thus, at the kth iteration, the gradient of f with respect to Y can be computed as is given by

$$\nabla f(Y^k) \;=\; 2\, X^{-1} Y^k\, \Omega_\eta \;-\; 2\,(W_2 \;-\; W_1)\, C^*,$$

where W_1 and W_2 are solutions to the following Lyapunov equations

$$\begin{aligned} A\, W_1 \;+\; W_1 A^* \;+\; X^{-1} Y^k\, \Omega_\eta\, Y^{k*} X^{-1} \;&=\; 0 \\ A\, W_2 \;+\; W_2 A^* \;+\; V_d \;&=\; 0 \end{aligned}$$

Here, X^{-1} denotes the inverse of $X(Y^k)$.

B. Hessian of $f(Y)$ in Eq. (16)

To find $\nabla^2 f(Y, \tilde{Y})$ in (16), we expand $\nabla f(Y + \epsilon \tilde{Y})$ around Y for the variation $\tilde{Y}$, and collect terms of $O(\epsilon)$. At the kth iteration we have

$$\nabla^2 f(Y^k, \tilde{Y}) \;=\; 2\,(X^{-1}\tilde{Y} - X^{-1} W_3 X^{-1} Y^k)\,\Omega_\eta + 2\,\tilde{W}_1 C^*,$$

where $\tilde{W}_1$ and W_3 are solutions to the following Lyapunov equations

$$0 \;=\; A^* W_3 \;+\; W_3 A \;-\; \tilde{Y}\, C \;-\; C^* \tilde{Y}^*$$

$$\begin{aligned} 0 \;=\;& A\,\tilde{W}_1 \;+\; \tilde{W}_1 A^* \;+\; X^{-1}\tilde{Y}^k\, \Omega_\eta\, Y^{k*} X^{-1} \;+ \\ & X^{-1} Y^k\, \Omega_\eta\, \tilde{Y}^{k*} X^{-1} - X^{-1} W_3\, X^{-1} Y^k\, \Omega_\eta\, Y^{k*} X^{-1} \\ & - X^{-1} Y^k\, \Omega_\eta\, Y^{k*} X^{-1} W_3\, X^{-1} \end{aligned}$$

Here, X^{-1} denotes the inverse of $X(Y^k)$.

REFERENCES

[1] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 91–101, 2016.

[2] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 67–78, 2016.

[3] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202–210, 2017.

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

[5] V. Roy, S. P. Chepuri, and G. Leus, "Sparsity-enforcing sensor selection for DOA estimation," in *2013 IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2013, pp. 340–343.

[6] S. Kondoh, C. Yatomi, and K. Inoue, "The positioning of sensors and actuators in the vibration control of flexible systems," *JSME Int. J., Ser. III*, vol. 33, no. 2, pp. 145–152, 1990.

[7] K. Hiramoto, H. Doki, and G. Obinata, "Optimal sensor/actuator placement for active vibration control using explicit solution of algebraic Riccati equation," *J. Sound Vib.*, vol. 229, no. 5, pp. 1057–1075, 2000.

[8] B. Polyak, M. Khlebnikov, and P. Shcherbakov, "An LMI approach to structured sparse feedback design in linear control systems," in *Proceedings of the 2013 European Control Conference*, 2013, pp. 833–838.

[9] N. K. Dhingra, M. R. Jovanović, and Z. Q. Luo, "An ADMM algorithm for optimal sensor and actuator selection," in *Proceedings of the 53rd IEEE Conference on Decision and Control*, 2014, pp. 4039–4044.

[10] U. Münz, M. Pfister, and P. Wolfrum, "Sensor and actuator placement for linear systems based on $\mathcal{H}_2$ and $\mathcal{H}_\infty$ optimization," *IEEE Trans. Automat. Control*, vol. 59, no. 11, pp. 2984–2989, 2014.

[11] F. Lin, M. Fardad, and M. R. Jovanović, "Sparse feedback synthesis via the alternating direction method of multipliers," in *Proceedings of the 2012 American Control Conference*, 2012, pp. 4765–4770.

[12] F. Lin, M. Fardad, and M. R. Jovanović, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Trans. Automat. Control*, vol. 58, no. 9, pp. 2426–2431, September 2013.

[13] L. Stella, A. Themelis, and P. Patrinos, "Forward-backward quasi-Newton methods for nonsmooth optimization problems," *Comput. Optim. Appl.*, vol. 67, no. 3, pp. 443–487, 2017.

[14] G. E. Dullerud and F. Paganini, *A course in robust control theory: a convex approach*. New York: Springer-Verlag, 2000.

[15] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[16] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[17] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.

[18] S. Becker and J. Fadili, "A quasi-newton proximal splitting method," in *NIPS*, 2012, pp. 2618–2626.

[19] J. D. Lee, Y. Sun, and M. A. Saunders, "Proximal Newton-type methods for minimizing composite functions," *SIAM J. Optim.*, vol. 24, no. 3, pp. 1420–1443, 2014.

[20] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012.

[21] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, p. 1, 2010.

[22] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar, "Sparse inverse covariance matrix estimation using quadratic approximation," in *NIPS*, 2011, pp. 2330–2338.

[23] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Mathematical Programming*, vol. 117, no. 1-2, pp. 387–423, 2009.

[24] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, no. 1, pp. 141–148, 1988.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[26] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2013.

[27] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[28] A. Zare, M. R. Jovanović, and T. T. Georgiou, "Perturbation of system dynamics and the covariance completion problem," in *Proceedings of the 55th IEEE Conference on Decision and Control*, 2016, pp. 7036–7041.

[29] N. K. Dhingra, S. Z. Khong, and M. R. Jovanović, "A second order primal-dual method for nonsmooth convex composite optimization," *IEEE Trans. Automat. Control*, 2017, submitted; also arXiv:1709.01610.