53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

An ADMM algorithm for optimal sensor and actuator selection

Neil K. Dhingra, Mihailo R. Jovanović, and Zhi-Quan Luo

Abstract— **We consider the problem of the optimal selection of a subset of available sensors or actuators in large-scale dynamical systems. By replacing a combinatorial penalty on the number of sensors or actuators with a convex sparsity-promoting term, we cast this problem as a semidefinite program. The solution of the resulting convex optimization problem is used to select sensors (actuators) in order to gracefully degrade performance relative to the optimal Kalman filter (Linear Quadratic Regulator) that uses all available sensing (actuating) capabilities. We employ the alternating direction method of multipliers to develop a customized algorithm that is well-suited for large-scale problems. Our algorithm scales better than standard SDP solvers with respect to both the state dimension and the number of available sensors or actuators.**

Index Terms— **Actuator and sensor selection, alternating direction method of multipliers, convex optimization, semidefinite programming, sparsity-promoting estimation and control.**

I. Introduction

In traditional applications, controller or observer design deals with the problem of how to use a pre-specified configuration of sensors and actuators in order to attain the desired objective. In general, the best performance is achieved by using *all* of the available sensors or actuators. However, this option may be computationally or economically infeasible. We thus consider the problem of selecting a subset of available sensors or actuators in order to gracefully degrade performance relative to the setup where all of them are used.

Typically, sensor/actuator selection and placement is performed by a designer with expert knowledge of the system. However, in large-scale applications and systems with complex interactions, it can be difficult to do this effectively. For linear time-invariant dynamical systems, we develop a framework and an efficient algorithm to systematically choose sensors and actuators via convex optimization.

Our starting point is a formulation with an abundance of potential sensors or actuators. This setup can encode information about different types or different placements of sensing and actuating capabilities. We consider the problem of selecting subsets of available options from this full model. Applications of this formulation range from placement of Phasor Measurement Units (PMUs) in power systems, to placement of sensors and actuators along an aircraft wing, to the distribution of GPS units in a formation of multi-vehicle systems.

The problem of interest is a difficult combinatorial optimization problem. Although there is a wide body of previous work in this area, most of the available literature either uses heuristic methods or does not consider dynamical models. In [1], the authors provide a convex sensor selection formulation for a problem with linear measurements. The authors of [2] select sparse subsets of sensors to minimize the Cramer-Rao bound of a class of nonlinear measurement models. The placement of PMUs in power systems was formulated as a variation of the optimal experiment design in [3]. Actuator selection via genetic algorithms was explored in [4]. A non-convex formulation of the joint sensor and actuator placement was provided in [5], [6] and it was recently applied to the linearized Ginzburg-Landau equation [7]. The leader selection problem in consensus networks can be seen as a type of a structured joint actuator and sensor selection problem which admits a convex relaxation [8] and even an analytical solution for one or two leaders [9]. However, this formulation does not extend naturally to broader classes of problems.

The sparsity-promoting framework introduced in [10]–[12] can be used to obtain block-sparse structured feedback and observer gains and select actuators or sensors. Indeed, algorithms developed in [12] have been used in [13] for the sensor selection in a target tracking problem. However, these algorithms have been developed for general structured control/estimation problems and they do not exploit the hidden convexity of the actuator/sensor selection problem.

In [14], the authors introduced a convex semidefinite programming (SDP) characterization of the problem formulation considered in [12] for enhancing certain forms of sparsity in the feedback gain. Although sensor and actuator selection falls into the class of problems considered by [14], generic SDP solvers cannot handle large-scale applications. Since we are interested in high-dimensional systems with many sensors/actuators, we use the alternating direction method of multipliers (ADMM) [15] to develop a customized solver that is well-suited for large problems. In contrast to standard SDP solvers, whose computational complexity scales unfavorably with the number of states/sensors/actuators, the worst case per-iteration complexity of our method scales only with the number of states. Furthermore, our algorithm performs much better than standard SDP solvers in numerical experiments.

The rest of the paper is organized as follows. In Section II, we state the actuator and sensor selection problems and introduce their convex formulations. In Section III,

Financial support from the National Science Foundation under award ECCS-1407958, the NASA Harriet G. Jenkins Predoctoral Fellowship Program, and the MnDRIVE Graduate Scholars Program is gratefully acknowledged.

N. K. Dhingra, M. R. Jovanović, and Z. Q. Luo are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455. E-mails: dhin0008@umn.edu, mihailo@umn.edu, luozq@umn.edu.

we develop a customized ADMM algorithm for actuator selection. In Section IV, we provide an example to illustrate the utility of our algorithm and in Section V we summarize our developments.

II. Problem formulation

A. Actuator selection

Consider the standard state-space system,

$$\begin{aligned} \dot{x} &= A\,x \;+\; B_1\,d \;+\; B_2\,u \\ z &= \begin{bmatrix} Q^{1/2} \\ 0 \end{bmatrix} x \;+\; \begin{bmatrix} 0 \\ R^{1/2} \end{bmatrix} u \end{aligned}$$

where d is a zero-mean white stochastic process with covariance V_d and the pair (A, B_2) is controllable. The optimal $\mathcal{H}_2$ controller minimizes the steady-state variance,

$$\lim_{t\,\to\,\infty} \; \mathcal{E}\left(x^T(t)\,Q\,x(t) \;+\; u^T(t)\,R\,u(t)\right)$$

where $\mathcal{E}$ is the expectation operator, $Q = Q^T \succeq 0$ specifies a weight on the system states, and $R = R^T \succ 0$ specifies the penalty on the control input. The global optimal controller for this problem is a state feedback law of the form $u = -Kx$. Although this controller is readily computed by solving the corresponding algebraic Riccati equation, it typically uses all input channels and thus all available actuators.

We are interested in designing an optimal controller which uses a subset of the available actuators. This will be accomplished by augmenting the $\mathcal{H}_2$ performance index with a term that promotes row-sparsity of the feedback gain matrix K. The resulting problem can be cast as a semidefinite program and thus solved efficiently for small problems.

1) SDP Formulation: Under the state feedback control law, $u = -Kx$, the closed-loop system is given by

$$\begin{aligned} \dot{x} &= (A \;-\; B_2K)\,x \;+\; B_1\,d \\ z &= \begin{bmatrix} Q^{1/2} \\ -R^{1/2}K \end{bmatrix} x. \end{aligned} \tag{1}$$

The $\mathcal{H}_2$ norm of system (1) is determined by

$$J(K) \;=\; \mathrm{trace}\left(Q\,X \;+\; K^T R\,K\,X\right)$$

where $X = X^T \succ 0$ is the controllability gramian of the closed loop system,

$$(A \;-\; B_2K)\,X \;+\; X\,(A \;-\; B_2K)^T \;+\; B_1\,V_d\,B_1^T \;=\; 0.$$

Since X is positive definite and therefore invertible, the standard change of coordinates $Y := KX$ can be used to express $J(K)$ in terms of X and Y [16],

$$J(X,Y) \;=\; \mathrm{trace}\left(Q\,X \;+\; X^{-1}\,Y^T R\,Y\right)$$

and to bring the optimal $\mathcal{H}_2$ problem into the following form

$$\begin{aligned} \underset{X,\,Y}{\text{minimize}} \quad & J(X,Y) \\ \text{subject to} \quad & AX + XA^T - B_2Y - Y^TB_2^T + V = 0 \\ & X \succ 0 \end{aligned} \tag{2}$$

with $V := B_1V_dB_1^T$. By taking the Schur complement of $X^{-1}Y^TRY$ this problem can be expressed as an SDP [16]. Finally, the optimal feedback gain can be recovered by $K = YX^{-1}$. In what follows, we use this SDP characterization to introduce the actuator selection problem.

2) Sparsity Structure: When the ith row of K is identically equal to zero, the ith control input is not used. Therefore, obtaining a control law which uses only a subset of available actuators can be achieved by promoting row-sparsity of K. Our developments are facilitated by the equivalence between the row-sparsity of K and Y; the ith row of K is equal to zero if and only if the ith row of $Y = KX$ is equal to zero [14].

Drawing on the group-sparsity paradigm [17], we augment (2) with a sparsity-promoting penalty on the rows of Y,

$$\begin{aligned} \text{minimize} \quad & J(X,Y) \;+\; \gamma \sum_{i\,=\,1}^{m} w_i \,\|\mathrm{e}_i^T Y\|_2 \\ \text{subject to} \quad & AX + XA^T - B_2Y - Y^TB_2^T + V = 0 \\ & X \succ 0. \end{aligned} \tag{3}$$

The parameter $\gamma > 0$ specifies the importance of sparsity relative to $\mathcal{H}_2$ performance, w_i are nonzero weights, and e_i is the ith unit vector in $\mathbb{R}^m$.

This problem can still be cast as an SDP and standard solvers can be used to find its solution. However, since generic SDP solvers do not exploit the structure of (3), they do not scale gracefully with problem dimension.

B. Sensor selection

The sensor selection problem can be approached in a similar manner. Consider a linear time-invariant system,

$$\begin{aligned} \dot{x} &= A_s\,x \;+\; B_1\,d \\ y &= C\,x \;+\; \eta \end{aligned}$$

where d and η are zero-mean white stochastic processes with covariances V_d and V_η, respectively, and (A_s, C) is an observable pair. The observer,

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

estimates the state x from the noisy measurements y using a linear injection term with an observer gain L. For the Hurwitz matrix $A_s - LC$, the zero-mean estimate of x is given by $\hat{x}$ and the dynamics of the estimation error $\tilde{x} := x - \hat{x}$ are governed by

$$\dot{\tilde{x}} \;=\; (A_s \;-\; L\,C)\,\tilde{x} \;+\; B_1\,d \;-\; L\,\eta. \tag{4}$$

The variance amplification from the noise sources d and η to the estimation error $\tilde{x}$ is determined by

$$J_o(L) \;=\; \mathrm{trace}\,(X_o\,B_1\,V_d\,B_1^T \;+\; X_o\,L\,V_\eta\,L^T) \tag{5}$$

where X_o is the observability gramian of the error system (4),

$$(A_s \;-\; L\,C)^T\,X_o \;+\; X_o\,(A_s \;-\; L\,C) \;+\; I \;=\; 0.$$

The Kalman filter gain L, resulting from the observer Riccati equation, provides an optimal observer with the smallest variance amplification.

Our objective is to design a Kalman filter which uses a subset of the available sensors. This can be achieved by enhancing column-sparsity of the observer gain L. Since the change of coordinates $Z := X_o L$ preserves column sparsity of L, we formulate the sensor selection problem as

$$\begin{array}{ll} \underset{X_o,\, Z}{\text{minimize}} & J_o(X_o, Z) \;+\; \gamma \displaystyle\sum_{i=1}^{r} w_i \, \|Z \, \mathrm{e}_i\|_2 \\ \text{subject to} & A_s^T X_o + X_o A_s - C^T Z^T - Z C + I = 0 \\ & X_o \succ 0 \end{array} \tag{6}$$

where

$$J_o(X_o, Z) \;=\; \mathrm{trace}\left(X_o \, B_1 \, V_d \, B_1^T \;+\; X_o^{-1} \, Z \, V_\eta \, Z^T\right).$$

We note that the sensor selection problem (6) can be obtained from the actuator selection problem (3) by setting the problem data in (3) to

$$\begin{array}{lllll} A \;=\; A_s^T, & B_2 \;=\; C^T, & Q \;=\; B_1 \, V_d \, B_1^T \\ V \;=\; I, & R \;=\; V_\eta \end{array} \tag{7}$$

and recovering the variables $X_o = X$ and $Z = Y^T$.

III. Customized algorithm

We next develop an efficient algorithm for solving the actuator selection problem (3); the solution to the sensor selection problem (6) can be obtained by mapping it to (3) via (7). The challenges in solving the optimization problem (3) arise from

- the positive definite constraint;
- the linear Lyapunov-like constraint;
- the non-smoothness of the sparsity-promoting term.

To ensure positive definiteness of X, we use projected descent techniques to optimize over the positive definite cone. We dualize the linear constraint and split (3) into two simpler subproblems over X and Y via the alternating direction method of multipliers (ADMM). This splitting separates the objective function component associated with the positive definite constraint (the subproblem over X) from the non-differentiable component (the subproblem over Y). Since the subproblem over Y is efficiently solvable, our method scales well with the number of sensors or actuators.

We employ a projected version of Newton's method to solve the X-subproblem. The Newton search direction is obtained via a conjugate gradient algorithm. The Y-subproblem is solved with a proximal method.

A. Alternating direction method of multipliers

To employ ADMM, we first form the augmented Lagrangian corresponding to the optimization problem (3),

$$\mathcal{L}_\rho(X, Y, \Lambda) \;:=\; J(X, Y) \;+\; \gamma \, g(Y) \;+\; \langle \Lambda, h(X, Y) \rangle \;+\; (\rho/2) \, \|h(X, Y)\|_F^2$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product between two matrices, $h(X, Y)$ is the linear Lyapunov-like constraint

$$h(X, Y) \;:=\; AX \;+\; XA^T \;-\; B_2 Y \;-\; Y^T B_2^T \;+\; V$$

and $g(Y)$ denotes the sparsity-promoting term,

$$g(Y) \;:=\; \sum_{i=1}^{m} w_i \, \|\mathrm{e}_i^T Y\|_2.$$

Relative to the standard Lagrangian, $\mathcal{L}_\rho$ contains an additional quadratic penalty on the violation of the Lyapunov constraint. The positive parameter ρ specifies how close X and Y are to satisfying this linear constraint at each iteration.

The ADMM iteration uses the update sequence [15]

$$\begin{array}{rcl} X_{k+1} & = & \underset{X}{\arg\min} \;\; \mathcal{L}_\rho(X, \, Y_k, \, \Lambda_k) \\ Y_{k+1} & = & \underset{Y}{\arg\min} \;\; \mathcal{L}_\rho(X_{k+1}, \, Y, \, \Lambda_k) \\ \Lambda_{k+1} & = & \Lambda_k \;+\; \rho \, h(X_{k+1}, \, Y_{k+1}) \end{array}$$

to find the optimal solution to the original problem. The stopping criteria depend on the primal residual, which quantifies how well X_k and Y_k satisfy the linear constraint, and the dual residual, which quantifies the difference between Y_k and Y_{k-1}. We refer the reader to [15] for details.

B. X-minimization step

The X-minimization step is equivalent to solving,

$$\begin{array}{ll} \underset{X}{\text{minimize}} & J(X, Y_k) \;+\; \dfrac{\rho}{2} \, \|h(X, Y_k) \;+\; \dfrac{1}{\rho} \, \Lambda_k\|_F^2 \\ \text{subject to} & X \;\succ\; 0. \end{array} \tag{8}$$

We use a projected version of Newton's method to solve a sequence of quadratic approximations to (8).

1) Newton's method: At each iteration, Newton's method does a line search from $\bar{X}$ in the direction Δ_n which minimizes the quadratic approximation of (8),

$$\Delta_n \;:=\; \underset{\Delta}{\arg\min} \;\; \frac{1}{2} \, \langle H_{\bar{X}}(\Delta), \Delta \rangle \;+\; \left\langle G(\bar{X}), \Delta \right\rangle \tag{9}$$

where $G(\bar{X})$ is the gradient of $\mathcal{L}_\rho$ with respect to X evaluated at $\bar{X}$, and $H_{\bar{X}}(\Delta)$ is a linear function of Δ that contains information about the Hessian of $\mathcal{L}_\rho$; see equations (12) and (13) in the appendix. Solving the linear equation

$$H_{\bar{X}}(\Delta) \;+\; G(\bar{X}) \;=\; 0$$

yields the Newton direction Δ_n which is computed using the conjugate gradient method [18].

2) Projection: The set $\mathcal{C}$ approximates the positive definite cone $X \succ 0$ with $X \succeq \epsilon I$. Once the Newton direction is determined, the step

$$X \;=\; \mathcal{P}_{\mathcal{C}} \left(\bar{X} \;+\; \frac{1}{\beta} \, \Delta_n \right)$$

is taken, where $\mathcal{P}_{\mathcal{C}}$ is the projection on the set $\mathcal{C}$ and the step size $1/\beta$ is chosen using an Armijo backtracking search.

To project the symmetric matrix M onto $\mathcal{C}$, its eigenvalues

are projected onto the set $\lambda_i \geq \epsilon$. From the eigenvalue decomposition $M = U\,\mathrm{diag}(\lambda)\,U^T$, where λ is a vector of the eigenvalues and U is a matrix of the corresponding eigenvectors, the projection is $\mathcal{P}_{\mathcal{C}}(M) = U\,\mathrm{diag}\left(\max\left(\lambda, \epsilon\right)\right)U^T$.

While Netwon's method reduces the number of required steps, computing the search directions can be prohibitively expensive for large-scale systems. To deal with this issue, we are currently exploring efficient implementation of projected gradient descent methods, quasi-Newton methods, and the use of log-barrier functions. Furthermore, in practice it is not necessary to solve the X-minimization step exactly and we can terminate after a few Newton or gradient steps [15].

C. Y-minimization step

The Y-minimization step is equivalent to solving,

$$\begin{array}{ll}\underset{Y}{\text{minimize}} & \gamma\, g(Y) \;+\; \mathrm{trace}\left(X_{k+1}^{-1} Y^T R Y\right) \;+ \\ & (\rho/2)\,\|h(X_{k+1}, Y) \;+\; (1/\rho)\,\Lambda_k\|_F^2. \end{array} \tag{10}$$

The objective function is the sum of a quadratic term and a separable sum of ℓ_2 norms: a problem form commonly referred to as group LASSO. This problem has been extensively studied in recent years and there are a variety of techniques for computing its solution. We employ an efficient proximal method known as Iterative Soft-Thresholding (ISTA) [19]. At each point $\bar{Y}$, the smooth part of the objective function is linearized and a proximal term is added,

$$\underset{Y}{\text{minimize}} \quad \gamma\, g(Y) \;+\; \left\langle F(\bar{Y}), Y\right\rangle \;+\; \frac{\beta}{2}\,\|Y \;-\; \bar{Y}\|_F^2$$

where, $F(\bar{Y})$ is the gradient of the smooth part of (10) evaluated at $\bar{Y}$; see equation (14) in the appendix. This approximation has an analytic minimizer; the update of the ith row of the matrix Y is given by

$$\mathrm{e}_i^T Y \;=\; \mathcal{S}_\alpha\left(\mathrm{e}_i^T \bar{Y} \;-\; \frac{1}{\beta}\,\mathrm{e}_i^T F(\bar{Y})\right)$$

where $\alpha := \gamma w_i/\beta$ and $\mathcal{S}$ is the block shrinkage operator which acts on each row of Y as

$$\mathcal{S}_\alpha(\mathrm{e}_i^T Y) \;=\; \begin{cases} \left(1 \;-\; \alpha/\|\mathrm{e}_i^T Y\|_2\right)\mathrm{e}_i^T Y, & \|\mathrm{e}_i^T Y\|_2 \;>\; \alpha \\ 0, & \|\mathrm{e}_i^T Y\|_2 \;\leq\; \alpha. \end{cases}$$

D. Iterative reweighting

Inspired by [20], we employ an iterative reweighing scheme to select the weights w_i in the sparsity-promoting term $\sum_i w_i\|a_i\|_2$ to obtain sparser structures at lower values of γ. The authors in [20] noted that if $w_i = 1/\|a_i\|_2$, then there is an exact correspondence between the weighted ℓ_1 norm and the cardinality function. However, implementing such weights requires *a priori* knowledge of the values $\|a_i\|_2$ at the optimal a. Consequently, we implement a reweighting scheme in which we run the algorithm multiple times for each value of γ and update the weights as,

$$w_i^{j+1} \;=\; \frac{1}{\|\,\mathrm{e}_i^T Y\,\|_2 \;+\; \epsilon} \tag{11}$$

where $\epsilon > 0$ ensures that the update is always well-defined.

IV. An example

We use a simple mass-spring system to illustrate the utility of our algorithm in the sensor selection problem. This system has a clear intuitive interpretation and its dimension can be easily scaled while retaining the problem structure. Consider a series of masses connected by linear springs. With unit masses, unit spring constants and no friction, the dynamics of each mass are described by

$$\ddot{p}_i \;=\; -(p_i \;-\; p_{i+1}) \;-\; (p_i \;-\; p_{i-1}) \;+\; d_i$$

where p_i is the position of the ith mass. When the first and last masses are affixed to rigid bodies, the aggregate dynamics are given by

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} \;=\; \begin{bmatrix} 0 & I \\ -T & 0 \end{bmatrix}\begin{bmatrix} p \\ v \end{bmatrix} \;+\; \begin{bmatrix} 0 \\ I \end{bmatrix} d$$

where p, v, and d are the position, velocity and disturbance vectors, and T is a Toeplitz matrix with 2 on the main diagonal and -1 on the first super- and sub-diagonals. The possible sensor outputs are the position and velocity vectors.

A. Algorithm speed and computational complexity

The complexity of solving the sensor selection SDP with interior point methods is $O((n + r)^6)$, where n is the dimension of the state-space and r is the number of sensors. In our algorithm, the greatest cost is incurred by computing the Newton direction in the X-minimization step. Since $X \in \mathbb{R}^{n\times n}$, the worst case complexity of computing Newton direction is $O(n^6)$. This is because each conjugate gradient step takes $O(n^3)$ operations and, in general, n^2 conjugate gradient steps are required to obtain convergence. In well-conditioned problems, the conjugate gradient method achieves high accuracy much faster which significantly reduces computational complexity. The Y-minimization step has a computational cost of only $O(nr)$, so the overall cost per ADMM iteration is $O(n^6)$ unless $r \geq n^5$.

Figure 1 shows the time required by ADMM and by CVX [21], [22] to solve (6) with $\gamma = 100$ for mass-spring systems of increasing sizes. Our algorithm scales favorably here and it scales much better when only the number of sensors is varied. Figure 2 shows scaling with *just* the number of outputs. As the number of sensors increases, CVX's computation time increases while the computation time of ADMM barely changes.

B. Sensor selection

We consider a system with 20 masses (40 states) and potential position and velocity measurements for each mass. As γ increases, sparser observer structures are uncovered at the cost of compromising quality of estimation. The tradeoff between the number of sensors and the variance of the estimation error is shown in Figure 3.

Figures 4 and 5 show the position and velocity sensor topologies identified by the ADMM algorithm as γ is increased. To save space, only novel topologies are shown. The selected sensor configurations have symmetric topology,

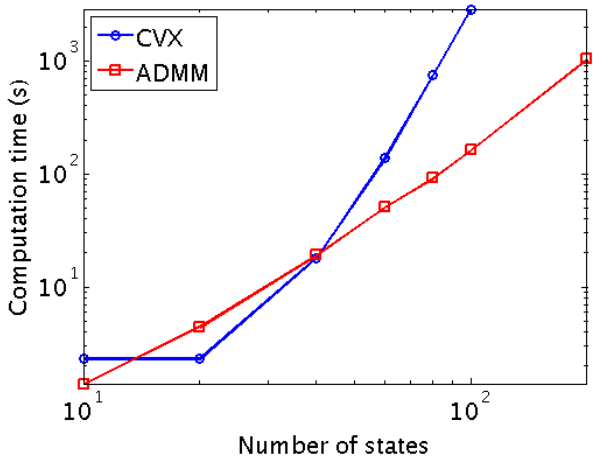

Fig. 1. Scaling of computation time with the number of states for CVX and for ADMM for mass-spring system with $\gamma = 100$ and position and velocity outputs. Empirically, we observe that CVX scales roughly with n^6 while ADMM scales roughly with n^3.

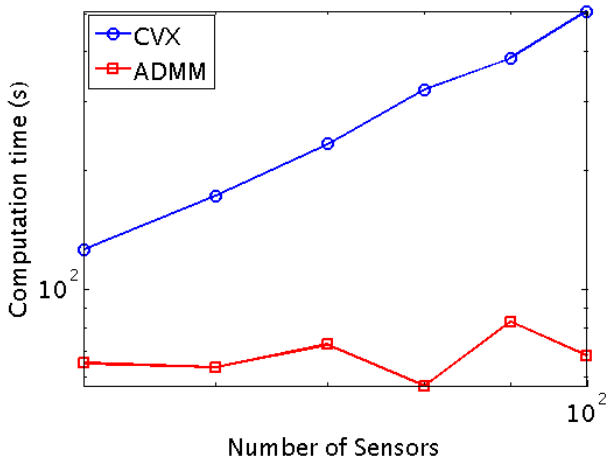

Fig. 2. Scaling of computation time with the number of sensors for CVX and for ADMM for mass-spring system with $\gamma = 100$ and $n = 50$. Outputs are random linear combinations of the states.

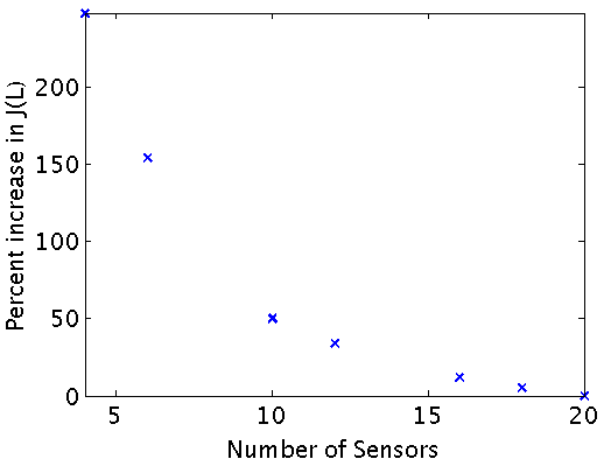

Fig. 3. Percent increase in $J(L)$ in terms of the number of sensors.

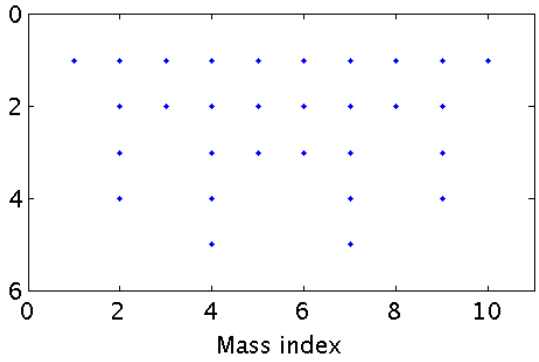

Fig. 4. Retained position sensors as γ increases. A blue dot indicates that the position of the corresponding mass is being measured. The top row shows the densest sensor topology, and the bottom row shows the sparsest.

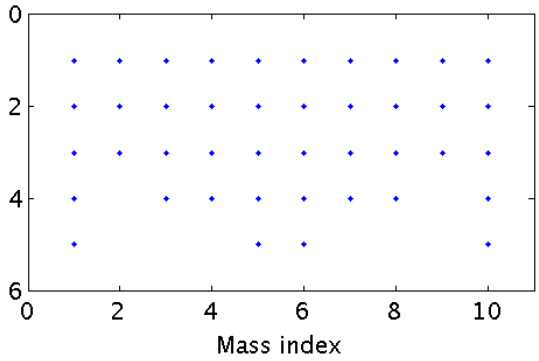

Fig. 5. Retained velocity sensors as γ increases. A blue dot indicates that the velocity of the corresponding mass is being measured. The top row shows the densest sensor topology, and the bottom row shows the sparsest.

which is expected for this example. Notably, velocity measurements are, in general, more important but the locations of the most important position and velocity sensors differ.

C. Iterative reweighting

Figure 6 illustrates the utility of iterative reweighting. When constant sparsity-promoting weights are used, large values of γ are required to identify sparse structures. Here, we run our ADMM algorithm 3 times for each value of γ, updating the weights using (11) and retaining them as we increase γ.

V. Concluding remarks

We have provided a convex characterization of sensor and actuator selection problems in linear time-invariant dynamical systems. An efficient customized ADMM algorithm that relies on iterative reweighting has been developed. The speed

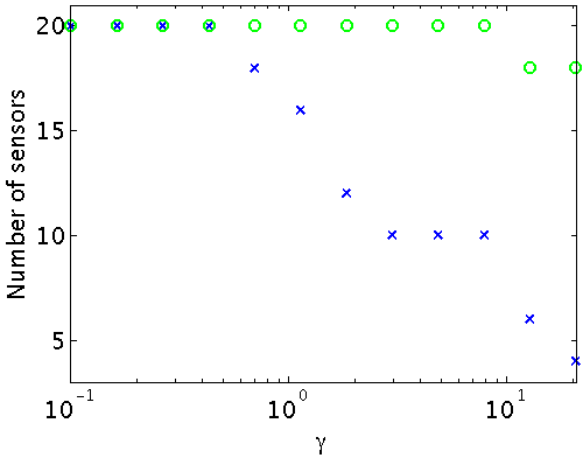

Fig. 6. Number of sensors versus γ for a scheme which uses iterative reweighting and for the scheme which uses constant weights. Iterative reweighting promotes sparser structures earlier.

and utility of our algorithm have been demonstrated using a mass-spring example.

The most expensive step of our algorithm involves computation of Newton direction via the conjugate gradient method. In order to improve computational complexity, we are currently exploring the development of projected gradient and quasi-Newton methods.

REFERENCES

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

[2] 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.

[3] V. Kekatos, G. B. Giannakis, and B. Wollenberg, "Optimal placement of phasor measurement units via convex relaxation," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1521–1530, 2012.

[4] J. L. Rogers, "A parallel approach to optimum actuator selection with a genetic algorithm," in *AIAA Guidance, Navigation, and Control Conference*, 2000, pp. 14–17.

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

[6] 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.

[7] K. K. Chen and C. W. Rowley, "$\mathcal{H}_2$ optimal actuator and sensor placement in the linearised complex Ginzburg-Landau system," *J. Fluid Mech.*, vol. 681, pp. 241–260, 2011.

[8] F. Lin, M. Fardad, and M. R. Jovanović, "Algorithms for leader selection in stochastically forced consensus networks," *IEEE Trans. Automat. Control*, vol. 59, no. 7, pp. 1789–1802, July 2014.

[9] K. Fitch and N. E. Leonard, "Information centrality and optimal leader selection in noisy networks," in *Proceedings of the 52nd IEEE Conference on Decision and Control*, 2013, pp. 7510–7515.

[10] M. Fardad, F. Lin, and M. R. Jovanović, "Sparsity-promoting optimal control for a class of distributed systems," in *Proceedings of the 2011 American Control Conference*, San Francisco, CA, 2011, pp. 2050–2055.

[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*, Montréal, Canada, 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] E. Masazade, M. Fardad, and P. K. Varshney, "Sparsity-promoting extended Kalman filtering for target tracking in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 19, pp. 845–848, 2012.

[14] 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.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–124, 2011.

[16] G. E. Dullerud and F. Paganini, *A course in robust control theory*. Springer, 2000.

[17] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[18] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

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

[20] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.

[21] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," http://cvxr.com/cvx, Sept. 2013.

[22] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

APPENDIX

A. X-minimization

The gradient of $\mathcal{L}_\rho$ with respect to X is given by

$$G(X) := Q - X^{-1}Y^T R Y X^{-1} + A\big(\Lambda + \rho\, h(X,Y)\big) + \big(\Lambda + \rho\, h(X,Y)\big)A^T. \tag{12}$$

The quadratic term used in (9) is,

$$\frac{1}{2}\,\mathrm{vec}(\Delta)^T(\nabla^2\mathcal{L}_\rho)\,\mathrm{vec}(\Delta)$$

where $\nabla^2\mathcal{L}_\rho$ is the Hessian. This can be more conveniently expressed as, $(1/2)\,\langle H(\Delta),\Delta\rangle$ where,

$$H_X(\Delta) := H_{1,X}(\Delta) + \rho\, H_2(\Delta). \tag{13}$$

The first term in (13) comes from the performance index,

$$H_{1,X}(\Delta) := G_{1,X}\,\Delta\,X^{-1} + X^{-1}\,\Delta\,G_{1,X}$$
$$G_{1,X} := X^{-1}Y^T R Y X^{-1}$$

and the second term comes from the constraint penalties,

$$H_2(\Delta) := A^T A\,\Delta + \Delta\,A^T A + A\,\Delta\,A + A^T\Delta\,A^T.$$

B. Y-minimization

The smooth part of the objective function in (10) is everything but $g(Y)$. Its gradient with respect to Y is

$$F(Y) := 2RYX^{-1} + 2\rho\left(B_2^T Y^T B_2^T + B_2^T B_2 Y\right) - 2B_2^T\left(\Lambda + \rho(AX + XA^T + V)\right). \tag{14}$$