NOVA

University of Newcastle Research Online

nova.newcastle.edu.au

Kong, He; Goodwin, Graham C.; Seron, Maria M. "A cost-effective sparse communication strategy for networked linear control systems: an SVD-based approach". Originally published in International Journal of Robust and Nonlinear Control Vol. 25, Issue 14, p. 2223-2240 (2015)

Available from:

<http://dx.doi.org/10.1002/rnc.3188>

This is the peer reviewed version of the following article: Kong, He; Goodwin, Graham C.; Seron, Maria M. "A cost-effective sparse communication strategy for networked linear control systems: an SVD-based approach", International Journal of Robust and Nonlinear Control Vol. 25, Issue 14, p. 2223-2240 (2015), which has been published in final form at <http://dx.doi.org/10.1002/rnc.3188>.

This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Self-Archiving

Accessed from: <http://hdl.handle.net/1959.13/1317574>

A Cost-effective Communication Strategy for Networked Linear Control Systems: An SVD-based Approach

He Kong*, Graham C. Goodwin, and Maria M. Seron

March 26, 2014

Abstract

In networked control, there is often an incentive to communicate only what is absolutely necessary to achieve the desired performance goals. This is true of both the downlink (between a control base station and actuators) and the uplink (between the sensors and base station). Here we present a strategy aimed at this problem based on a singular value decomposition (SVD) of the Hessian of the quadratic performance index generally considered in Model Predictive Control (MPC). The singular vectors are employed to generate an orthonormal basis function expansion of the unconstrained solution to the finite horizon optimal control problem. These are pre-loaded into each actuator and each sensor. On the downlink, the actuators are informed, in real-time, about which basis functions they should use. On the uplink, after a “burn in period”, the sensors need only communicate when their response departs from that pre-calculated for the given basis functions. We show that this strategy facilitates communication in both the downlink and uplink in a cost-effective fashion. We also show that the strategy can be modified so that input constraints are satisfied. We illustrate the proposed results by applying them to a simulation of the cross direction (CD) control of a paper machine. Potential extensions and other applications of the proposed strategy are also discussed.

1 Introduction

The topic of sparse estimation has been extensively studied in several areas of signal processing, e.g., image processing (see, for example, [1] and the references therein), compressive sensing (see, for example, [2] and the references therein), etc. Sparse modeling is typically achieved by utilizing some form of regularization. In many cases, an ℓ_1 regularization is used since it has been shown that, under certain conditions, this leads to an optimal sparse representation (see, for example, [3] and the references therein).

Recent research in the control literature has sought to extend these ideas to control problems. For example, in [4], after the concept of an ℓ_0 -system gain is introduced for discrete-time linear time-invariant (LTI) systems, a system theoretic explanation is presented regarding the sparse closed-loop response of ℓ_1 -optimal controlled systems. In [5], several techniques are proposed to design feedback systems that sparsify the state in order to perfectly reconstruct it using compressive sensing algorithms. The effect of such sparsification schemes on the stability and regulation of the plant have also been studied. The idea of ℓ_1 regularization has also been examined in recent MPC literature (see for example, [6]-[9]). For example, [6] presents an algorithm with sufficient accuracy and moderate complexity solving optimization problems with an ℓ_1 regularized least-squares cost function subject to recursive equality constraints. In [7], by adopting an ℓ_1 regularization, for rate limited networks, a design technique of sparse packets with

*The authors are with the Centre for Complex Dynamic Systems and Control, School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan NSW 2308, Australia. Email: he.kong@uon.edu.au (He Kong), graham.goodwin@newcastle.edu.au (Graham Goodwin), maria.seron@newcastle.edu.au (Maria Seron).

asymptotic stability of the control loop in the presence of bounded packet dropouts has been presented. The results of [7] have been complemented and enhanced in [8] where a detailed technical analysis of the scheme and proofs of results have been presented. Reference [9] provides a new paradigm for MPC policies which feature an ℓ_1 -regularised least squares loss function. In that work, a discrete-time dual-mode MPC is formulated, and its stability is proven by application of standard MPC arguments.

Our motivation in the current paper is similar to that in the papers mentioned above, namely, we seek a sparse solution to a control problem. We depart from the literature cited above since we do not deploy ℓ_1 regularization. Instead, we focus on the parameterization of the solution space. Our approach is motivated, in part, by the observation that the choice of parameterization deployed in all optimization problems affects the ease of their solutions. There are several factors relevant to the choice, e.g., (i) the connection to the final implementation space (in control this is usually the individual control moves); (ii) the similarity of the chosen functions to “typical” solutions (in control these are often exponential functions of time); and (iii) the degree of orthogonality of the functions in terms of their impact on the desired objective. Here we emphasize point (iii) since this facilitates the solution of related sparsity questions. In particular, we deploy the vectors appearing in an SVD of the Hessian used in MPC as basis functions. We believe that the proposed strategy has several potential advantages including (i) it provides a simple, but near optimal, solution to the convex input constrained control problem; (ii) it allows robustness to be automatically achieved; (iii) it provides a sparse mechanism for minimizing downlink and uplink communication. The SVD decomposes the quadratic cost function into a sum of quadratic terms where each term is associated with one singular vector. Hence, no coupling exists between the impact of different terms. This makes it simple to answer questions such as: (i) what is the maximum combination of vectors necessary to achieve a given goal; (ii) what vectors play little (or no) role; (iii) if we decide to add just one extra vector, which choice would have the greatest impact on the cost function; (iv) if we want to eliminate just one vector, which choice would have the least impact on the cost function.

The results in the current paper are related to earlier work described in [10]. In [10], a sub-optimal receding horizon control strategy for input constrained linear systems was presented. The strategy was based on an SVD of the Hessian of the quadratic performance index generally considered in MPC. Moreover, in [10], it was shown that, for sufficiently long prediction horizons, the singular values are arbitrarily close to the magnitude of the energy density spectrum of the system seen by the performance index. The singular vectors are employed to generate a basis function expansion of the unconstrained solution to the finite horizon optimal control problem. At each sampling time, a feasible control sequence is determined by selecting a variable subset of the basis representation.

Core aspects of the technique proposed here differ from the method of [10]. For example, in [10], the main idea is to successively eliminate those basis vectors which contribute least (in a certain sense) to the solution until a near optimal and feasible solution to the constrained optimization is obtained. Here we successively add those basis vectors which contribute most (in a certain sense) to the solution until a near optimal and feasible solution to the problem is found. Thus, we shift from feasibility to feasibility combined with sparsity. An advantage of the method proposed here, compared to the results in [6]–[9], is that the current method avoids solving the associated optimization problem. Moreover, the method proposed here results in a sparse communication technique, since a small amount of information is transmitted from which the actuators reconstruct the actual input. Thus, the resulting control signal is not necessarily sparse. The method helps to reduce communication burden while providing guarantees of stability and a prespecified degree of sub-optimality. This paper extends the preliminary results presented in the conference paper [11]. The remainder of the paper is organized as follows. In Section 2, some preliminaries on MPC are presented. Section 3 describes the SVD mechanism. Section 4 presents stability analysis. Implementation of the proposed sparse control strategy for networked control systems is discussed in Section 5 (we emphasize the downlink between the control base station and the plant). A simulation example is presented in Section 6. Section 7 concludes the paper.

Notation: The notation used in this paper is standard. A^T is used to denote the transpose of matrix A . $\mathbf{R}$ and $\mathbf{R}_{\geq 0}$ denote the real and non-negative real number sets, respectively. $\mathbf{R}^n$ stands for the n -dimensional Euclidean space. The notation $M > 0$ (≥ 0) means that M is real symmetric and positive definite (semi-definite). I_n stands for identity matrices of n dimensions. 0_n denotes matrices of n dimensions whose

elements are all 0. r_m^k stands for a row vector of dimension m whose k -th element is 1 while the rest are 0. $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ denote the minimum and maximum eigenvalues of a matrix M . Vectors and matrices, if their dimensions are not explicitly stated, are assumed to be compatible for the associated algebraic operations. $\mathbb{Z}^+[i, j]$ denotes the set of positive integers between i and j . The Euclidean norm is denoted by $|\cdot|$. $\mathcal{N}$ denotes the set of non-negative integers and $\mathcal{N}^+$ denotes the set of positive integers.

2 Fundamentals of MPC

Here we review standard material on MPC, see, e.g., [12]-[13]. We consider an LTI discrete-time system described in state-space form

$$\begin{cases} x_{k+1} = Ax_k + Bu_k, \\ y_k = Cx_k, \end{cases}, \quad (1)$$

where $x_k \in \mathbf{R}^n$, $u_k \in \mathbf{R}^m$, and $y_k \in \mathbf{R}^p$ are the state vector, the input vector, and the output vector, respectively. We assume that (A, B) and (A, C) are stabilizable and detectable, respectively. We also assume that the system is subject to input constraints $u_{k+t} \in \mathbf{U}$, for $t = 0, 1, \dots$. We are interested in a quadratic performance index of the following form:

$$J_N(x_k, \mathbf{u}) = \sum_{t=0}^{N-1} [x_{k+t|k}^T Q x_{k+t|k} + u_{k+t|k}^T R u_{k+t|k}] + x_{k+N|k}^T P x_{k+N|k} \quad (2)$$

where $\mathbf{u}$ contains the control moves u_{k+t} defined over the finite horizon $t \in [0, N-1]$ with $N \in \mathbb{Z}^+$ and $N \geq 1$, that is,

$$\mathbf{u} = [u_{k|k}^T, u_{k+1|k}^T, \dots, u_{k+N-1|k}^T]^T. \quad (3)$$

The weighting matrices satisfy $Q \geq 0, R > 0$ whilst the terminal weighting matrix P is taken to be the unique symmetric positive definite solution to the discrete time algebraic Riccati equation (ARE)

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A. \quad (4)$$

The vector $x_{k+t|k}$ in (2) represents the prediction of the state vector of the system at time $k+t$ given the information available at time k and based on the model in (1), i.e.,

$$\begin{cases} x_{k+t+1|k} = Ax_{k+t|k} + Bu_{k+t|k}, & k = 0, 1, \dots, \\ x_{k|k} = x_k \end{cases}. \quad (5)$$

MPC solves, at each time step, the following finite horizon optimal control problem

$$\mathcal{P}_N(x) : \begin{cases} \mathbf{u}^o(x) = \arg \min J_N(x, \mathbf{u}) \\ \text{subject to (5), } u_{k+t|k} \in \mathbf{U}, \text{ for } t = 0, \dots, N-1 \end{cases}, \quad (6)$$

where $x = x_k$ is the state at the current time. Only the first move (that is, the first m elements) of the optimal control vector $\mathbf{u}^o(x)$ is applied to the system (1). The optimization procedure is repeated at each time step when a new state measurement becomes available.

3 SVD-based sparse strategy

In this section, we review the SVD approach to MPC and use it to develop a sparse communication strategy for utilizations in networked control systems (NCS). The strategy proposed here is close in spirit to those described in [14]-[20]. However, our strategy is based on the inherent structure of the cost function as viewed via an SVD. Our goal is to achieve sparse communication of the control moves.

3.1 SVD of the optimization problem

Here we summarize the idea of using SVD in an optimization problem as originally presented in [10]. Using matrix notation, $J_N(x_k, \mathbf{u})$ in (2) can be expressed as follows:

$$J_N(x_k, \mathbf{u}) = \mathbf{u}^T \mathbf{H} \mathbf{u} + 2\mathbf{u}^T \mathbf{F} x_k + c \quad (7)$$

where $c = x_k^T (\Lambda^T \mathbf{Q} \Lambda + Q) x_k$ is a constant (independent of $\mathbf{u}$),

$$\begin{cases} \mathbf{H} = \mathbf{R} + \Gamma^T \mathbf{Q} \Gamma, \quad \mathbf{R} = \text{diag}(R, \dots, R), \\ \mathbf{F} = \Gamma^T \mathbf{Q} \Lambda, \quad \mathbf{Q} = \text{diag}(Q, \dots, Q, P), \end{cases} \quad (8)$$

with

$$\begin{cases} \mathbf{Q} = \text{diag}(Q, \dots, Q, P), \\ \Lambda = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \Gamma = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix} \end{cases} \quad (9)$$

Matrices Λ and Γ define the relationship between the vector of control moves $\mathbf{u}$ and the vector, $\mathbf{x} = [x_{k+1|k}^T, \dots, x_{k+N|k}^T]^T$, which contains the future time evolution of the state vector of the system (1), i.e., $\mathbf{x} = \Lambda x_k + \Gamma \mathbf{u}$. The unconstrained solution to the finite horizon optimal control problem $\mathcal{P}_N(x)$ for $x = x_k$ is readily seen to be

$$\mathbf{u}_{uc}^o(x) = -\mathbf{H}^{-1} \mathbf{F} x_k. \quad (10)$$

In the constrained case, the solution is more complex. Specifically, as pointed out in [21] and [22], an explicit solution for $\mathbf{u}^o(x)$ in (6) consists of a partition of the state space into a finite (but possibly large) number of regions in which the corresponding control law is an affine function of the states.

In this paper, we explore an alternative approach. We first seek to determine which components of the vector of control moves $\mathbf{u}$ have the *largest* impact on the magnitude of the performance index $J_N(x_k, \mathbf{u})$. Note that this is determined by the structure of $J_N(x_k, \mathbf{u})$ in terms of the control vector $\mathbf{u}$. To clarify this, we consider an SVD of the Hessian $\mathbf{H}$ of $J_N(x_k, \mathbf{u})$,

$$\mathbf{H} = \mathbf{V} \mathbf{S} \mathbf{V}^T. \quad (11)$$

The matrix $\mathbf{S} \in \mathbf{R}^{Nm \times Nm}$ is diagonal, with positive diagonal entries $\{\sigma_1, \sigma_2, \dots, \sigma_{Nm}\}$ arranged in decreasing order. The scalars $\{\sigma_i\}_{i=1}^{Nm}$ are the singular values of $\mathbf{H}$. The columns of the matrix $\mathbf{V}$ contain the right singular vectors of $\mathbf{H}$. Moreover, in this case, left and right singular vectors are identical since $\mathbf{H}$ is symmetric. The fact that the singular vector matrix $\mathbf{V}$ is orthonormal, i.e., $\mathbf{V} \mathbf{V}^T = \mathbf{V}^T \mathbf{V} = \mathbf{I}_{Nm}$ implies that columns of $\mathbf{V}$ provide a set of orthogonal basis vectors spanning $\mathbf{R}^{Nm}$. As a result, the vector of control moves $\mathbf{u}$ in (3) can be expressed as a linear combination of the singular vectors of $\mathbf{H}$, i.e.,

$$\mathbf{u} = \mathbf{V} \tilde{\mathbf{u}} = \sum_{i=1}^{Nm} v_i \tilde{u}_i, \quad (12)$$

where v_i are the columns of $\mathbf{V}$ and $\tilde{u}_i$ are the entries of the vector $\tilde{\mathbf{u}}$, $i \in \mathbb{Z}^+[1, Nm]$. Using the change of variable defined in (12), we can recast the performance index (7) in terms of the new vector of control moves $\tilde{\mathbf{u}}$

$$\tilde{J}_N(x_k, \tilde{\mathbf{u}}) \triangleq J_N(x_k, \mathbf{V} \tilde{\mathbf{u}}) = \tilde{\mathbf{u}}^T \mathbf{S} \tilde{\mathbf{u}} + 2\tilde{\mathbf{u}}^T \mathbf{V}^T \mathbf{F} x_k + c.$$

Completing squares around the unconstrained optimal solution $\tilde{\mathbf{u}}_{uc}^o(x_k) = -\mathbf{S}^{-1} \mathbf{V}^T \mathbf{F} x_k$, we finally obtain

$$\begin{aligned} \tilde{J}_N(x_k, \tilde{\mathbf{u}}) &= \tilde{J}_{N,uc}^o + [\tilde{\mathbf{u}} - \tilde{\mathbf{u}}_{uc}^o(x_k)]^T \mathbf{S} [\tilde{\mathbf{u}} - \tilde{\mathbf{u}}_{uc}^o(x_k)] \\ &= \tilde{J}_{N,uc}^o + \sum_{i=1}^{Nm} \sigma_i (\tilde{u}_i - \tilde{u}_{i,uc}^o)^2 \end{aligned} \quad (13)$$

where $\tilde{u}_{i,uc}^o$ is the i -th entry of vector $\tilde{\mathbf{u}}_{uc}^o$, and $\tilde{J}_{N,uc}^o$ is the unconstrained optimal cost.

Remark 1 We see from (13) that each element of the basis function expansion contributes in an additive and independent way to the cost. As explored in [10], there exists a direct connection between the singular values of the matrix Γ and the singular values of $\mathbf{H}$. To clarify, consider the SVD of Γ , i.e., $\Gamma = U\bar{S}V^T$, and for simplicity consider $R = 0$ and $Q = I_n$. Then, from the definition of $\mathbf{H}$ in (11), we obtain $\mathbf{H} = \Gamma^T \Gamma = V\bar{S}^2 V^T$, since $U^T U = U U^T = I_{Nn}$. For the connection between the singular values of $\mathbf{H}$ with long prediction horizons N and the open loop properties of the system, one refer to [10].

3.2 SVD-based sparse algorithm: the unconstrained case

From the last subsection, we know that the impact of each singular vector in the cost function (13) is independent. Therefore, we can readily ask which singular vectors have the greatest impact on the cost function. We immediately have the following result.

Proposition 1 Say at step s we have agreed to include $\tilde{\mathbf{u}}_{j,uc}^o$ for $j \in N_s$ where $N_s \subset \mathbb{Z}^+[1, Nm]$. If we then decide to add $\tilde{\mathbf{u}}_{l,uc}^o$ for some $l \in C_s \triangleq \{z \mid z \in \mathbb{Z}^+[1, Nm] \text{ and } z \notin N_s\}$, it follows that the greatest impact on cost reduction is to choose l corresponding to

$$\max_{l \in C_s} \sigma_l(\tilde{\mathbf{u}}_{l,uc}^o)^2.$$

Proof. Using the singular vectors for the set N_s , the cost becomes

$$J_s^o = \tilde{J}_{N,uc}^o + \sum_{j \notin N_s} \sigma_j(\tilde{\mathbf{u}}_{j,uc}^o)^2.$$

Adding the singular vector for l , leads to

$$\begin{aligned} J_{s+1}^o &= \tilde{J}_{N,uc}^o + \sum_{j \notin N_{s+1}} \sigma_j(\tilde{\mathbf{u}}_{j,uc}^o)^2 \\ &= J_s^o - \sigma_l(\tilde{\mathbf{u}}_{l,uc}^o)^2. \end{aligned}$$

The result follows immediately. ■

Therefore, we see that the best choice for step $s+1$ is determined by a combination of the gain associated with the singular vector, i.e., σ_l , and some consideration for what we actually need to do, i.e., $(\tilde{\mathbf{u}}_{l,uc}^o)^2$. Based on this result, an unconstrained SVD-based sparse RHC strategy is described by the following algorithm.

Algorithm 1 (Unconstrained SVD-based sparse RHC strategy)

1. At time k , given the current state $x = x_k$, calculate the unconstrained optimal solution to $\mathcal{P}_N(x)$, i.e., $\tilde{\mathbf{u}}_{uc}^o(x_k) = -S^{-1}V^T \mathbf{F}x_k$;
2. Order the solution in decreasing order of values of $\sigma_i(\tilde{\mathbf{u}}_{i,uc}^o)^2$, $i \in \mathbb{Z}^+[1, Nm]$, with corresponding indices $o_1, o_2, \dots$;
3. Starting from o_1 and proceeding to o_{Nm} , find the smallest $s \in \mathbb{Z}^+[1, Nm]$, such that the resulting cost J_s^o satisfies $\tilde{J}_{N,uc}^o \leq J_s^o \leq \lambda \tilde{J}_{N,uc}^o$, where $\lambda > 1$ defines a prespecified degree of sub-optimality and $\tilde{J}_{N,uc}^o$ is the unconstrained optimal cost.
4. Compute the unconstrained SVD-based sparse RHC strategy

$$\begin{aligned} \mathbf{u}_{svd}(x_k) &= V\tilde{\mathbf{u}}_{svd}^o(x_k) \\ &= V M_s \begin{bmatrix} \tilde{\mathbf{u}}_{o_1}^o(x_k) & \cdots & \tilde{\mathbf{u}}_{o_s}^o(x_k) & 0 & \cdots & 0 \end{bmatrix}^T, \end{aligned}$$

where M_s is an index reordering matrix such that each element $\tilde{\mathbf{u}}_{o_j}^o(x_k)$ is placed in the o_j -th entry of the resulting vector $\tilde{\mathbf{u}}_{svd}^o(x_k)$, and zero is placed everywhere else.

5. Communicate the indices $o_1, \dots, o_s$ and the corresponding values of $\tilde{\mathbf{u}}_{svd}^o(x_k)$ to all actuators;
6. Each actuator constructs its own control move at the current sampling instant, i.e., for $i = 1, \dots, m$, $u_k^i = r_m^i D \mathbf{u}_{svd}(x_k)$, where

$$D = \begin{bmatrix} I_m & 0_{m \times (N-1)m} \end{bmatrix}; \quad (14)$$

7. Set $k = k + 1$ and return to step 1.

Remark 2 The algorithm performs in a greedy way since we order the solutions in decreasing values of $\sigma_i(\tilde{\mathbf{u}}_{i,uc}^o)^2$, $i \in \mathbb{Z}^+[1, Nm]$.

3.3 Dealing with convex input constraints

The basic algorithm as described in Subsection 3.2 does not explicitly include the consideration of constraints. However, for systems subject to input constraints (see, for example, [10], [23], and [24]), the SVD decomposition suggests a greedy strategy for obtaining a feasible constrained solution. The idea is to proceed as in the last subsection but check each time after a new singular vector is added that the new solution remains feasible. Thus, one adds components, starting from the one associated with the largest value of $\sigma_l(\tilde{\mathbf{u}}_{l,uc}^o)^2$, until the resulting control vector exceeds the allowable bounds. Then one scales the final step to just meet the bounds. More precisely, as in the last subsection, say at step s we have agreed to include $\tilde{\mathbf{u}}_{j,uc}^o$ for $j \in N_s$ where $N_s \subset \mathbb{Z}^+[1, Nm]$. For step s , we have the cost J_s^o . If we then decide to add $\tilde{\mathbf{u}}_{l,uc}^o$ for some $l \in C_s$ at step $s + 1$, we then have $J_{s+1}^o = J_s^o - \sigma_l(\tilde{\mathbf{u}}_{l,uc}^o)^2$. Once adding a particular singular vector leads to infeasibility, then we need to scale the control move associated with the largest value of $\sigma_l(\tilde{\mathbf{u}}_{l,uc}^o)^2$, until the resulting control vector becomes feasible. For this purpose, we define

$$\mathbf{u}_{svd} = \sum_{i=1}^s v_i \tilde{\mathbf{u}}_{i,uc}^o + v_{s+1} \alpha \tilde{\mathbf{u}}_{s+1,uc}^o \quad (15)$$

for a given scalar $\alpha \in [0, 1)$. Clearly, the unconstrained solution (12) with $\mathbf{u} = \mathbf{u}_{uc}^o(x)$ will be recovered if we set $s = Nm$ and $\alpha = 0$. The proposed input constrained SVD-based sparse RHC strategy is described in the following algorithm.

Algorithm 2 (Input constrained SVD-based sparse RHC strategy)

1. At time k , given the current state $x = x_k$, calculate the unconstrained optimal solution to $\mathcal{P}_N(x)$, i.e., $\tilde{\mathbf{u}}_{uc}^o(x_k) = -S^{-1}V^T \mathbf{F}x_k$;
2. Order the solution in decreasing order of the values of $\sigma_i(\tilde{\mathbf{u}}_{i,uc}^o)^2$, $i \in \mathbb{Z}^+[1, Nm]$, with corresponding indices $o_1, o_2, \dots$;
3. Find the smallest $\gamma = s + \alpha$, where $s \in \mathbb{Z}^+[1, Nm]$ and $\alpha \in [0, 1)$, such that

$$\begin{aligned} \mathbf{u}_{svd}(x_k) &= V \tilde{\mathbf{u}}_{svd}^o(x_k) \\ &= VM_{s+1} \begin{bmatrix} \tilde{\mathbf{u}}_{o_1}^o(x_k) & \dots & \tilde{\mathbf{u}}_{o_s}^o(x_k) & \alpha \tilde{\mathbf{u}}_{o_{s+1}}^o(x_k) & 0 & \dots & 0 \end{bmatrix}^T \end{aligned}$$

lies on the boundary of the constraint set, and the resulting cost J_s^o satisfies

$$\tilde{J}_{N,uc}^o \leq J_s^o \leq \lambda \tilde{J}_{N,uc}^o,$$

where $\lambda > 1$ defines the degree of sub-optimality and $\tilde{J}_{N,uc}^o$ is the unconstrained optimal cost. The matrix M_{s+1} is such that each element $\tilde{\mathbf{u}}_{o_j}^o(x_k)$, for $j = 1, \dots, s$, is placed as the o_j -th entry of the resulting vector $\tilde{\mathbf{u}}_{svd}^o(x_k)$, the element $\alpha \tilde{\mathbf{u}}_{o_{s+1}}^o(x_k)$ is placed as the o_{s+1} -th entry of the resulting vector $\tilde{\mathbf{u}}_{svd}^o(x_k)$, and zero is placed everywhere else.

4. Communicate the indices $o_1, \dots, o_{s+1}$ and the corresponding values of $\tilde{\mathbf{u}}_{svd}^o(x_k)$ to all actuators;
5. Each actuator constructs its own control move at the current sampling instant, i.e., for $i = 1, \dots, m$, $u_k^i = r_m^i D \mathbf{u}_{svd}(x_k)$, where D is as in (14);
6. Set $k = k + 1$ and return to Step 1.

Remark 3 Note that Algorithm 1 and Algorithm 2 differ from the Orthogonal Matching Pursuit (OMP) algorithm of [7] in several respects. For example, both Algorithm 1 and Algorithm 2 perform in a greedy way since we order the solutions in decreasing values of $\sigma_i(\tilde{\mathbf{u}}_{i,uc}^o)^2$, $i \in \mathbb{Z}^+[1, Nm]$. As a result, with the smallest amount of control information to be sent, input constraint satisfaction and sub-optimality of the method are guaranteed and a sparse description of the control input is obtained. On the other hand, the OMP algorithm of [7] only considers the case without constraints, and belongs to the general class of regularization methods as those in [6] and [9].

Remark 4 It should be noted that the value of λ in Algorithm 1 and Algorithm 2 needs to be selected carefully, since at each sampling instant, one expects a larger number of singular vectors to be needed, for a smaller value of λ . Moreover, it would be desirable to have a time-varying λ . However, we have not investigated the optimal way to obtain a time-varying λ at this stage.

Remark 5 Some remarks regarding feasibility and the issue of sub-optimality are made below. It might be reasonable to give priority to the former rather than the latter. Therefore, in practice, for Step 3, we suggest that one first check if the SVD-based sparse strategy is feasible. If yes, we proceed to increase the control action to achieve better performance until it becomes infeasible. If not, the SVD-based sparse strategy has already achieved the best performance (not necessarily the pre-specified measure of sub-optimality) with the least amount of information. Also, for open-loop stable systems, it might be generally reasonable to apply no control action if the states of the system enter a certain area which is close enough to the origin. Therefore, for open-loop stable systems, Algorithm 1 and Algorithm 2 can be readily modified so that the SVD-based sparse control communication strategy is only applied when the states of the system are outside of a region close to the origin. After the states reach that region, no information is needed to be sent as the system will eventually converge to the origin with no control action. We will illustrate these points via examples in Section 6.

4 Stability Analysis

4.1 Perturbation analysis

Closed-loop exponential stability can be established using a Lyapunov argument similar to that presented in [10]. Here we give details and make appropriate embellishments to cover the algorithms of Section 3. A key observation is that the control provided by the algorithms is a perturbation of the unconstrained optimal control problem defined by the cost function (2). Note that constraints are automatically included by virtue of the selection procedure.

Denote the matrices

$$K = \mathbf{H}^{-1}\mathbf{F}, \quad (16)$$

$$K_\gamma = V D_\gamma S^{-1} V^T \mathbf{F}, \quad (17)$$

where $\mathbf{H}, \mathbf{F}, V, S$ are defined in (8) and (11), respectively; $\gamma = s + \alpha$, with $s \in \mathbb{Z}[0, Nm]$ and $\alpha \in [0, 1)$; D_γ is a diagonal matrix with 1 in the appropriate places in its s diagonal entries, α in any other of its remaining diagonal entries, and zeros in the rest of its diagonal entries. Denote $E_\gamma = K - K_\gamma$. It is now evident that using D and K as in (14) and (16), respectively, we have

$$DK = K_o = (R + B^T P B)^{-1} B^T P A, \quad (18)$$

where P is the symmetric positive definite solution to the ARE (4). Then the closed-loop system obtained by applying the SVD-based sparse RHC strategy to system (1) can be described in perturbed form as

$$x_{k+1} = Ax_k - BDK_\gamma x_k = A_o x_k + BG_\gamma x_k, \quad (19)$$

where A_o is the unperturbed optimal closed-loop system matrix and $BG_\gamma x_k$ represents the perturbation, i.e.,

$$A_o = A - BK_o, \quad G_\gamma = D(K - K_\gamma) \triangleq DE_\gamma. \quad (20)$$

Before presenting the main stability result of the SVD-based sparse strategy for the case with convex input constraints, we firstly present the following lemma and corollary, which contain results related to Lemma 5.1 and Corollary 5.2 of [10] and can be proved in a similar fashion. Here we include the proof of the next lemma for the sake of completeness.

Lemma 1 *Consider two values $\gamma_1 = s_1 + \alpha$, $\gamma_2 = s_2 + \alpha$ with $s_1 \geq s_2$ such that $\gamma_1 \geq \gamma_2$. Denote $s_t = s_1 - s_2$. If D_{γ_1} (defined in the text immediately) shares all the non-zero diagonal entries of D_{γ_2} in the same entries of itself, and has 1 in s_t entries on the diagonal and 0 in the remaining diagonal entries, then, for any given vector x of a compatible dimension, we have*

$$|E_{\gamma_1} x| \leq |E_{\gamma_2} x|.$$

Proof. Note that $\mathbf{u}_{uc}^o(x_k) = -Kx_k$, $\mathbf{u}_{svd}(x_k) = -K_\gamma x_k$. Substituting (11) into (16), we obtain $K = VS^{-1}V^T\mathbf{F}$. Therefore, it holds that

$$E_\gamma = V\Delta_\gamma S^{-1}V^T\mathbf{F}, \quad (21)$$

where

$$\Delta_\gamma = I - D_\gamma. \quad (22)$$

Next, denote $d_\gamma = E_\gamma x$ for any given state vector $x \in \mathbf{R}^n$. Then we have $|d_\gamma| = |V\Delta_\gamma S^{-1}V^T\mathbf{F}x|$. Since V is orthonormal, the above equation reduces to $|d_\gamma| = |\Delta_\gamma S^{-1}V^T\mathbf{F}x|$. It can be seen that as the value of $\gamma = s + \alpha$ increases, there are fewer components of the vector $S^{-1}V^T\mathbf{F}x$ that are included when constructing d_γ . Therefore, given $s_1, s_2 \in \mathbb{Z}[0, Nm]$ with $s_1 \geq s_2$, if D_{γ_1} and D_{γ_2} satisfy the assumptions made above, it can be concluded that $|d_{\gamma_1}| \leq |d_{\gamma_2}|$, i.e.,

$$|E_{\gamma_1} x| \leq |E_{\gamma_2} x|. \quad (23)$$

This completes the proof. ■

The following corollary is immediate.

Corollary 1 *Consider K and K_γ as in (16) and (17), and as before, denote $E_\gamma = K - K_\gamma$, we have*

$$0 \leq |E_\gamma x| \leq |Kx|, \quad (24)$$

for any given vector x of a compatible dimension, $\gamma = s + \alpha$, with $s \in \mathbb{Z}[0, Nm]$ and $\alpha \in [0, 1)$.

Remark 6 *Though the results in Lemma 1 and Corollary 1 are very similar to those of Lemma 5.1 and Corollary 5.2 in [10], there is a key difference between them. That is the way to construct D_γ . In Lemma 1 and Corollary 1, D_γ is formulated based on cost reduction of the cost function (13) while in Lemma 5.1 and Corollary 5.2 of [10], D_γ is constructed by discarding the components of $\tilde{\mathbf{u}}_{uc}^o(x_k)$, starting from the one associated with the smallest singular value.*

We next present the main stability result of the SVD-based sparse strategy.

Theorem 1 *Consider $\delta > 0$ and suppose $\gamma = s + \alpha = \gamma^*$ at Step 3 of Algorithm 2 can be selected such that*

$$|E_{\gamma^*}|^2 < \frac{\lambda_{\min}(Q) - \delta \lambda_{\max}(P)}{|B^T P B + R|}. \quad (25)$$

Then the closed-loop system resulting from the application of Algorithm 2 is exponentially stable.

Proof. Consider the Lyapunov function candidate $\mathcal{V}(x_k) = x_k^T P x_k$ with P being the solution to the ARE (4). Along the closed-loop trajectories of (19), we have

$$\begin{aligned} \mathcal{V}(x_{k+1}) - \mathcal{V}(x_k) &= x_{k+1}^T P x_{k+1} - x_k^T P x_k \\ &= x_k^T [(A_o + B G_\gamma)^T P (A_o + B G_\gamma) - P] x_k \\ &= x_k^T [-Q - K_o^T R K_o + G_\gamma^T B^T P B G_\gamma + G_\gamma^T B^T P A_o + A_o^T P B G_\gamma] x_k. \end{aligned} \quad (26)$$

Moreover, it can be verified that

$$B^T P A_o = R K_o. \quad (27)$$

By substituting (27) into (26), we obtain

$$\begin{aligned} \mathcal{V}(x_{k+1}) - \mathcal{V}(x_k) &= x_k^T [-Q - (K_o - G_\gamma)^T R (K_o - G_\gamma) + G_\gamma^T (B^T P B + R) G_\gamma] x_k \\ &\leq -x_k^T [Q - G_\gamma^T (B^T P B + R) G_\gamma] x_k \\ &= -x_k^T [Q - E_\gamma^T D^T (B^T P B + R) D E_\gamma] x_k \end{aligned} \quad (28)$$

where D is as in (14), E_γ is defined in (21), with matrix Δ_γ in (22) above is a diagonal matrix with zeros in the diagonal entries corresponding to the indices $o_1, \dots, o_s$, the value $1 - \alpha$ in the o_{j+1} -th diagonal entry, and ones in the remaining diagonal entries. With E_γ chosen as in (25), we have that, given $\delta > 0$, the following inequality holds

$$x_k^T [Q - E_\gamma^T D^T (B^T P B + R) D E_\gamma] x_k \geq \delta x_k^T P x_k. \quad (29)$$

Hence

$$\mathcal{V}(x_{k+1}) - \mathcal{V}(x_k) \leq -\delta x_k^T P x_k$$

and exponential stability follows. ■

Remark 7 It can be seen from the results given above that the sufficient condition (25) which guarantees stability is expressed in terms of a norm requirement on E_γ . In the above proof, the results of Lemma 1 have been applied so that when one increases γ , more singular vectors are added to the previously selected ones until E_γ has the desired property (25). That is to say, one increases the number of the active singular vectors until $|E_\gamma|$ is lower than the desired bound. This will potentially reduce the communication burden in practice. Some discussion about the issues of feasibility and sub-optimality has been made in Remark 5. Here we make further comments regarding the issues of feasibility and the norm requirement in the stability guarantee in (25). Note that the norm requirement for stability can be conflicting with the issues of feasibility or sub-optimality in the sense that the number of active singular vectors to satisfy (25) might be different from the number necessary to meet the feasibility or sub-optimality requirement. However, in practice, it makes sense to emphasize feasibility rather than sub-optimality, as long as stability is guaranteed. Therefore, in the remainder of the paper, we assume that the norm requirement on E_γ in (25) is always feasible so that stability can be guaranteed. After the norm requirement in (25) is met, we increase the number of active singular vectors until the resulting control action is no longer feasible.

4.2 Trajectory-based Cost Reduction

Here we present an alternative approach to the stability analysis. The approach here is in the spirit of standard proofs used in MPC, see pp. 155-158 of [25]. The core idea is to select the control input so that the cost function reduces at each sampling instant. The following theorem presents a condition on E_γ for two sequential sampling instants which guarantees that the cost is decreasing and hence the closed-loop system is asymptotically stable.

Theorem 2 Suppose that at the current sampling instant, the matrix $E_\gamma = E_\gamma^a$ has been chosen to obtain a feasible SVD-based control

$$\mathbf{u}_{svd}^a(x_k) = -K_\gamma^a x_k = -(K - E_\gamma^a) x_k \in \mathbf{U}^N \triangleq \mathbf{U} \times \dots \times \mathbf{U}$$

for $\mathcal{P}_N(x_k)$ (see (6)). At the next sampling instant, if $E_\gamma = E_\gamma^b$ can be chosen such that

$$\mathbf{u}_{svd}^b(x_k) = -K_\gamma^b x_k = -(K - E_\gamma^b)x_k \in \mathbf{U}^N \triangleq \mathbf{U} \times \dots \times \mathbf{U}$$

and

$$P_1 + P_2 + P_3 < 0, \quad (30)$$

where

$$P_1 = (A_o + BDE_\gamma^a)^T (K - E_\gamma^b)^T \mathbf{H} (K - E_\gamma^b) (A_o + BDE_\gamma^a) - (K - E_\gamma^a)^T \mathbf{H} (K - E_\gamma^a), \quad (31)$$

$$P_2 = 2(K - E_\gamma^a)^T \mathbf{F} - 2(A_o + BDE_\gamma^a)^T (K - E_\gamma^b)^T \mathbf{F} (A_o + BDE_\gamma^a), \quad (32)$$

$$P_3 = (A_o + BDE_\gamma^a)^T (\Lambda^T \mathbf{Q} \Lambda + Q) (A_o + BDE_\gamma^a) - (\Lambda^T \mathbf{Q} \Lambda + Q), \quad (33)$$

with $\mathbf{H}, \mathbf{F}, \mathbf{Q}, \Lambda, D, A_o, K$ defined in (8), (9), (14), (20) and (16), respectively, then the value of the cost (2) decreases along the trajectory and hence the closed-loop system is asymptotically stable.

Proof. From the earlier analysis, it can be noted that if the current state is x_k , the control

$$\mathbf{u}_{svd}^a(x_k) = -K_\gamma^a x_k = -(K - E_\gamma^a)x_k \in \mathbf{U}^N = \mathbf{U} \times \dots \times \mathbf{U}$$

is a feasible control sequence for $\mathcal{P}_N(x_k)$. After the first element of $\mathbf{u}_{svd}^a(x_k)$ is applied to the system, we have the closed-loop system

$$x_{k+1} = Ax_k - BDK_\gamma^a x_k = (A_o + BDE_\gamma^a)x_k. \quad (34)$$

The resulting cost (2) is

$$J_N^a(x_k, \mathbf{u}_{svd}^a(x_k)) = \mathbf{u}_{svd}^{aT} \mathbf{H} \mathbf{u}_{svd}^a + 2\mathbf{u}_{svd}^{aT} \mathbf{F} x_k + x_k^T (\Lambda^T \mathbf{Q} \Lambda + Q) x_k.$$

Moreover, the initial state for the next sampling instant is x_{k+1} as in (34). Assume that, at this sampling instant, we select E_γ^b . Then we have

$$\begin{aligned} \mathbf{u}_{svd}^b(x_{k+1}) &= -K_\gamma^b x_{k+1} = -(K - E_\gamma^b)x_{k+1} \\ &= -(K - E_\gamma^b)(A_o + BDE_\gamma^a)x_k \in \mathbf{U}^N = \mathbf{U} \times \dots \times \mathbf{U}. \end{aligned}$$

Hence, the next value of the cost (2) can be shown (after some algebra) to be

$$J_N^b(x_{k+1}, \mathbf{u}_{svd}^b(x_{k+1})) = \mathbf{u}_{svd}^{bT} \mathbf{H} \mathbf{u}_{svd}^b + 2\mathbf{u}_{svd}^{bT} \mathbf{F} x_{k+1} + x_{k+1}^T (\Lambda^T \mathbf{Q} \Lambda + Q) x_{k+1}.$$

To obtain closed-loop stability [25], we only need to prove that

$$J_N^b(x_{k+1}, \mathbf{u}_{svd}^b(x_{k+1})) - J_N^a(x_k, \mathbf{u}_{svd}^a(x_k)) < 0.$$

It can be verified that the above inequality reduces to

$$x_k^T (P_1 + P_2 + P_3) x_k < 0,$$

where P_1, P_2, P_3 are defined in (31)-(33). This completes the proof. ■

Remark 8 The result in Theorem 2 presents an easily verified sufficient condition to guarantee closed-loop asymptotic stability. It is different from the condition in Theorem 1 in that it is not explicitly expressed in terms of a norm requirement on E_γ , but is, instead, based on adjusting E_γ as the trajectory evolves to guarantee satisfaction of the usual condition for MPC stability.

Based on the above analysis, we have the following modified SVD-based sparse strategy.

Algorithm 3 (Modified SVD-based sparse RHC strategy with input constraints)

1. At time k , given the current state $x = x_k$, calculate the unconstrained optimal solution to $\mathcal{P}_N(x)$, i.e., $\tilde{\mathbf{u}}_{uc}^o(x_k) = -S^{-1}V^T\mathbf{F}x_k$;
2. Order the solution in decreasing order of the values of $\sigma_i(\tilde{\mathbf{u}}_{i,uc}^o)^2$, $i \in \mathbb{Z}^+[1, Nm]$, with corresponding indices $o_1, o_2, \dots$;
3. Find the smallest $\gamma = s + \alpha$, where $s \in \mathbb{Z}^+[1, Nm]$ and $\alpha \in [0, 1)$, formulate $\mathbf{u}_{svd}(x_k)$ as in (15) so that the resulting cost J_s^o satisfies

$$J_s^o \leq J_{s-1}^o,$$

where J_{s-1}^o defines the cost of the previous sampling instant, and $\mathbf{u}_{svd}(x_k)$ lies on the boundary of the constraint set.

4. Communicate the indices $o_1, \dots, o_{s+1}$ and the corresponding values of $\tilde{\mathbf{u}}_{svd}^o(x_k)$ to all actuators;
5. Each actuator construct its own control move at the current sampling instant, i.e., for $i = 1, \dots, m$, $u_k^i = r_m^i D \mathbf{u}_{svd}(x_k)$, where D is the same as in Algorithm 2.
6. Set $k = k + 1$ and return to Step 1.

Remark 9 As the above algorithm requires storing the cost of the previous sampling instant, in order to initialize the algorithm in practice, we suggest at the first few sampling instants, one apply Algorithm 2 and then switch to Algorithm 3.

5 Realization of the Sparse Communication Strategy in Networked Control

In this section, we present some remarks on applying the proposed sparse communication strategy in the context of networked control.

5.1 Downlink communications

The principle is that the base station determines which singular values should be used (i.e., σ_j for $j \in N_s \subset \mathbb{Z}^+[1, Nm]$). Also, the size of the associated control (i.e., $\tilde{\mathbf{u}}_{\sigma_j}^o$) has been determined. The base station then transmits the pairs $(j, \tilde{\mathbf{u}}_{\sigma_j}^o)$ for $j \in N_s$. Each actuator is then able to reconstruct the required input sequence. One has several options to implement the control. Common choices in practice could include the following: (i) Implement just the first step of the control sequence as is usual in MPC; (ii) Implement a longer string of inputs until further notification of a change in control is received.

5.2 Uplink communications

The above discussion has focused on minimizing downlink transmissions by only notifying the actuators of the indices of the chosen singular vectors and the real scalars giving the corresponding projected input. The same concept can be used in the uplink but we have not, as yet, fully explored the idea. As an initial idea, consider the case of stable linear systems. In this case, there will be an initial transient due to the unknown global initial state of the system. At each step, the base station broadcasts control decisions to all the sensors and actuators. Each sensor can predict the response since it knows the actions (corresponding to the chosen singular vectors) of the actuators. After a “burn in period”, the response will converge to a predictable response as seen at each sensor since each knows the singular values that have been used by all actuators. Thus, the only time that a sensor needs to communicate back to the base station is when its observed response differs significantly from the predicted response. This would correspond to a disturbance which might, for example, be seen by a sub-set of the sensors. They then need to signal their

observed values back to the base station (The base station can assume that all silent sensors are sitting at their predicted values). The base station then needs to estimate a new global state to use in the control calculations and signal to all sensors that a new state estimate is being used. The sensors should then wait another “burn in period” before going silent.

6 Illustrative Example

In this section, we illustrate the proposed method with a simulation of a typical industrial problem, i.e., CD control of a paper machines. A paper machine is a specified type of web forming process. In a general set-up, these processes are characterized by raw material entering one end of the machine and a thin web or film being produced at the other end of the machine. The raw material is fed into the machine continuously or semi-continuously. Examples include sheet and film processes. These are effectively two dimensional spatially-distributed processes [26]. Several properties of the sheet material vary in the direction in which the sheet runs, known as the machine direction (MD), and in the direction across the sheet, known as the CD. The main objective of the control applied to sheet and film processes is to maintain both the MD and CD profiles of the sheet as flat as possible, in spite of multiple disturbances. The CD control problem for a paper machine is a multi-variable high dimensional problem with tight input constraints (see for example, [27], [28], [29] and the references therein). Indeed, it is not uncommon to have several hundred actuator-sensor pairs.

In order to design effective control algorithms for web forming processes, one requires a mathematical model of the process. When building a mathematical model of the system, it is important to consider the trade-off between the effort needed to obtain a detailed model of such complex systems and the benefits that can be extracted from it. Fortunately, for most web-forming processes, it is possible to build a fairly accurate model by considering the following two simplifying assumptions: (1) The effect of one single actuator is almost instantaneous in the CD direction; (2) The dynamic response in the machine direction is the same for every actuator-sensor pair. These assumptions are usually well justified in practice. Based on these two assumptions, we adopt the model as formulated in [30] and [31]. Assume that the system has 100 input-output pairs. Suppose that the system is resting in a steady state condition having a flat paper density profile and all the actuators at their default positions. Next, assume that a unit step is applied to one actuator. Based on the first simplifying assumption, one expects to observe a sudden change of the paper density profile across the entire CD of the web. One can then measure this change and arrange all the measures in one single column vector as follows:

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_{100} \end{bmatrix}.$$

It is reasonable to normalize the measurement of the profile density so that it becomes 1 at the sensor node whose corresponding actuator node is subject to the unit step; and the measurement of the profile density typically decays in an exponential way along the CD in both directions. Moreover, it is also justifiable to hypothesize that, if a unit step experiment were to be repeated for another actuator, then the resulting CD profiles should be a simple shifted version (centered at the position of the actuator screw number) of the previous profile. Note that there are important practical issues associated with what happens at the edges, but for simplicity, we will ignore these effects.

In a similar fashion, we could repeat the unit step experiment for each of the 100 actuators in the example, and record the corresponding values of the vector y . We can interpret the values of y as the vector gain between a certain actuator and the resulting CD profile. However, to quantify the net effect of all the actuators movements on the resulting CD profile, we need to combine the information collected in all the unit step experiments. To do this, we will make use of matrix notation. We define $\bar{u}$ to be a column vector

containing the value of the position of each screw on the slice lip:

$$\bar{u} = \begin{bmatrix} \bar{u}_1 \\ \vdots \\ \bar{u}_{100} \end{bmatrix}.$$

The net effect on the CD profile at each position is the sum of the partial effects generated by each actuator. If we define $\bar{y}$ to be a vector containing this net effect, we have that

$$\bar{y} = \bar{B}\bar{u},$$

where $\bar{B}$ is a matrix whose columns are the measurements collected after each unit step experiment. Based on the above analysis, we know that the diagonal entries of matrix $\bar{B}$ are all 1. We now return to the issue of the machine direction dynamics and the second assumption introduced previously. Denote $y(t)$ a column vector that contains the measurements of the paper density at a downstream position from the array of actuators. When a unit step is applied to one of the actuators, the CD profile changes according to the entries in the corresponding column of $\bar{B}$. However, the density of the paper on the moving forming fabric will change only gradually, until it settles at a certain steady state value. This is an indication of the dynamic response of the system in the machine direction (MD).

Several authors have shown that a simple first order model with time delay can be used to model the machine direction dynamics (see, for example, [32]). In particular, the following transfer function is suggested $h(s) = \frac{g_0 e^{-s\beta}}{\tau s + 1}$, where g_0 is the transfer function's d.c. gain, τ is the time constant and β is the time delay. The time delay $e^{-s\beta}$ accounts for the physical separation that exists between the actuators and the sensors in a typical web forming process application. We now use the assumption that the dynamic response in the machine direction is the same for every actuator-sensor pair. We can then write

$$y(s) = \begin{bmatrix} \frac{g_0 e^{-s\beta}}{\tau s + 1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{g_0 e^{-s\beta}}{\tau s + 1} \end{bmatrix} \bar{B}u(s).$$

For the purpose of illustration, we take $g_0 = 3$, $\tau = 5$. We will not explicitly include the delay since this only introduces a time shift in the results. Therefore, the final design model is

$$y(s) = G(s)u(s), \tag{35}$$

where

$$G(s) = \begin{bmatrix} \frac{3}{5s+1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{3}{5s+1} \end{bmatrix} \bar{B}.$$

Note that the model (35) is a continuous-time open loop stable system. The system has 100 inputs, 100 outputs and 100 states. Assume that $G(s)$ is discretized with a sampling period $T = 0.2s$. We use MPC with a prediction horizon of 10, and take $Q = I_{100}$, $R = 50I_{100}$, and set P to be the solution of the discrete time ARE (4).

6.1 The unconstrained case

Here we assume that there are no constraints on the system and that it is required to regulate the state of the system to the origin. All initial states are taken to be 10. As the system is open-loop stable, according to Remark 5, we assume that when the states of the system enter a ball around the origin, the SVD-based sparse strategy is no longer applied. Using $\lambda = 1.2$ and applying the unconstrained SVD-based sparse RHC strategy of Section 3 yields the trajectories shown in Figure 1. The red dashed line, the green dotted and dashed line and the black solid line represent, at each time instant, the first element of the system

Figure 1: The trajectory of the state and the output of the plant and the number of active singular vectors for the unconstrained case with $\lambda = 1.2$

output, the first element of the state, and the number of the active singular vectors. It can be seen from Figure 1 that the proposed strategy gives good transient performance and that a maximum of 40 singular vectors is needed.

Moreover, when we reduce or increase the value of λ , it is expected that the number of active singular vectors should become larger and smaller, respectively. This is verified in Figure 2, where the green dashed line, the black solid line, and the red dotted dashed line correspond to the number of active singular vectors for $\lambda = 1.12$, $\lambda = 1.2$ and $\lambda = 1.28$, respectively.

Figure 2: The number of the active singular vectors for different values of λ (the green dashed line, the black solid line and the red dotted dashed line for $\lambda = 1.12$, $\lambda = 1.2$, $\lambda = 1.28$, respectively)

6.2 The constrained case

Next, we assume that the system has the following input constraints $\mathbf{U}_1 \triangleq \{u \in \mathbf{R}^{100} \mid u \in [-8, 8] \times \cdots \times [-8, 8]\}$. We choose $\lambda = 1.4$ to compare the unconstrained case and the constrained case. As above, we assume that when the state enters a suitable ball, the SVD-based sparse strategy is no longer applied. We denote N_a the number of active singular vectors for these two cases. We compare the number of active singular vectors at each sampling instant, as shown in Figure 3, where the block solid and dashed lines represent the number of active singular vectors for the unconstrained case and the constrained case, respectively. Note that less singular vectors are used in the constrained case due to the requirement of constraint satisfaction.

Figure 3: The number of the active singular vectors for the unconstrained (the block solid line) and the constrained cases (the black dashed line) for $\lambda = 1.4$

It can be noted from Figure 3 that the system enters the region where the algorithm is stopped earlier for the unconstrained case. This is as anticipated since when there is no limit on the control effort, the system converges more quickly. Moreover, when we choose $\lambda = 1.4$ and apply a stricter input constraint $\mathbf{U}_2 \triangleq \{u \in \mathbf{R}^{100} \mid u \in [-1, 1] \times \cdots \times [-1, 1]\}$, the number of active singular vectors becomes even smaller as illustrated in Figure 4, where the red dotted dashed line and the black dashed line represent, respectively, the number of active singular vectors for the strict constraint case and the moderate constraint case.

Figure 4: The number of the active singular vectors with different constraints for $\lambda = 1.4$ (the red dotted dashed line and the black dashed line for the case of $\mathbf{U}_2$ and $\mathbf{U}_1$, respectively)

Next we seek to illustrate the results in Theorem 2. Since the system is open-loop stable, a reasonable way to illustrate the points of Theorem 2 is to put a limit on the number of singular vectors to be sent. We vary this number from 0 to 10 and illustrate the change of the cost in Figure 5. For simplicity, we assume that there is no input constraints present. Initial conditions are taken to be a smooth cosine wave with 10 in the middle and 0s on the edges. This kind of initial condition represent a typical form of disturbance found in paper machine control. In Figure 5, the black dashed line represents the case when no control action is applied, the red solid lines from top to bottom stand for the case when the number of active singular vectors varies from 1 to 10, respectively, and the blue dotted and dashed line represents the case when all singular vectors are active. Note that there are a total of 1,000 singular vectors yet half of the total allowable performance improvement is obtained by using just 10 of them. It can be noticed from Figure 5 that as we increase the number of singular vectors, the cost reduction becomes smaller and smaller. Moreover, with 10 active singular vectors, the cost has been reduced significantly. It can be estimated that the cost reduction when applying the SVD-based approach, compared to the case when no

control action is applied, is between 30%-50%. Note that this has been achieved with only a small amount of communication burden.

Figure 5: Evolution of the cost with different numbers of active singular vectors (the black dashed line, the red solid lines from top to bottom, and the blue dotted and dashed line respectively represent the case when no control action is applied, the cases when the number of active singular vectors varies from 1 to 10, and the case when all singular vectors are active)

7 Conclusion and Discussions

We have proposed a sparse communication strategy for NCS based on a sub-optimal solution to the finite horizon optimal control problem solved at each time step. The sub-optimal solution is constructed from an SVD basis function expansion of the unconstrained solution to the optimization problem. At each sampling time, the proposed strategy determines a feasible control sequence by selecting a variable subset of the basis representation. The proposed method provides a well-defined trade-off between system performance and the required control effort. We have shown that convex input constraints can be incorporated into the sparse communication strategy for networked linear control systems. We have also established a detailed stability proof and illustrated the proposed method via a numerical simulation. Planned future work includes the extension to tracking problem, study of the finite alphabet constraint case and investigation of applications to power electronics problems.

References

- [1] M. Elad, M. A. T. Figueiredo, and Y. Ma, On the role of sparse and redundant representations in image processing, *Proceedings of the IEEE*, Vol. 98, No. 6, pp. 972 - 982, 2010.
- [2] E. J. Candes, J. K. Romberg, and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. on Information Theory*, Vol. 52, No. 2, pp. 489-509, 2006.
- [3] D. L. Donoho, Compressed Sensing, *IEEE Trans. on Information Theory*, Vol. 52, No. 4, pp. 1289-1306, 2006.
- [4] S. Schuler, C. Ebenbauer, and F. Allgöwer, ℓ_0 -system gain and ℓ_1 -optimal control, *Proceedings of the 18th IFAC World Congress*, pp. 9230-9235, Milano, 2011.
- [5] S. Bhattacharya and T. Basar, Sparsity based feedback design: a new paradigm in opportunistic sensing, *Proceedings of the American Control Conference*, pp. 3704-3709, San Francisco, CA, 2011.

- [6] M. Annergren, A. Hansson, and B. Wahlberg, An ADMM algorithm for solving ℓ_1 regularized MPC, *Proceedings of the 51st IEEE Conference on Decision and Control*, pp. 4486-4491, Maui, Hawaii, 2012.
- [7] M. Nagahara, D. E. Quevedo, and J. Østergaard, Packetized predictive control for rate-limited networks via sparse representation, *Proceedings of the 51st IEEE Conference on Decision and Control*, pp. 1362-1367, Maui, Hawaii, 2012.
- [8] M. Nagahara, D. E. Quevedo, and J. Østergaard, Sparse packetized predictive control for networked control over erasure channels, *IEEE Trans. on Automatic Control*, Vol. 59, No. 7, July 2014 (to appear).
- [9] M. Gallieri and J. M. Maciejowski, The ℓ_{asso} MPC: smart regulation of over-actuated systems, *Proceedings of the American Control Conference*, pp. 1217-1222, Montreal, Canada, 2012.
- [10] O. J. Rojas, G. C. Goodwin, M. M. Seron, and A. Feuer, An SVD based strategy for receding horizon control of input constrained linear systems, *International Journal of Robust & Nonlinear Control*, Vol. 14, No. 13-14, pp. 1207-1226, 2004.
- [11] H. Kong, G. C. Goodwin, and M. M. Seron, Sparse networked control of input constrained linear systems, *Proceedings of the 52nd IEEE Conference on Decision and Control*, pp. 6615-6620, Florence, Italy, 2013.
- [12] G. C. Goodwin, M. M. Seron, and J. D. Doná, *Constrained control and estimation: an optimization approach*, New York: Springer, 2005.
- [13] H. Kong, G. C. Goodwin, and M. M. Seron, Predictive metamorphic control, *Automatica*, Vol. 49, No. 12, pp. 3670-3676, 2013.
- [14] G. C. Goodwin, D. E. Quevedo, and E. I. Silva, Architectures and coder design for networked control systems, *Automatica*, Vol. 44, No. 1, pp. 248-257, 2008.
- [15] W. P. Heath, A. G. Wills and J. A. G. Akkermans, A sufficient condition for the stability of optimizing controllers with saturating actuators, *International Journal of Robust & Nonlinear Control*, Vol. 15, No. 12, pp. 515-529, 2005.
- [16] R. Yang, P. Shi, G. P. Liu, and H. Gao, Network-based feedback control for systems with mixed delays based on quantization and dropout compensation, *Automatica*, Vol. 47, No. 12, pp. 2805-2809, 2011.
- [17] L. Zhang, H. Gao, and O. Kaynak, Network-induced constraints in networked control systems—a survey, *IEEE Trans. on Industrial Informatics*, Vol. 9, No. 1, pp. 403-416, 2013.
- [18] G. C. Goodwin, H. Haimovich, D. E. Quevedo, and J. S. Welsh, A moving horizon approach to networked control system design, *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, pp. 1427-1445, 2004.
- [19] H. Kong, G. C. Goodwin, and M. M. Seron, A revisit to inverse optimality of linear systems, *International Journal of Control*, Vol. 85, No. 10, pp. 1506-1514, 2012.
- [20] M. G. Cea and G. C. Goodwin, Stabilization of systems over bit-rate-constrained networked control architectures, *IEEE Trans. on Industrial Informatics*, Vol. 9, No. 1, pp. 357-364, 2013.
- [21] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica*, Vol. 38, No. 1, pp. 3-20, 2002.
- [22] M. M. Seron, G. C. Goodwin, J. A. Dona, Characterisation of receding horizon control for constrained linear systems, *Asian Journal of Control*, Vol. 5, No. 2, pp. 271-286, 2003.
- [23] M. M. Seron, G. C. Goodwin, and S. F. Graebe, Control system design issues for unstable linear systems with saturated inputs, *IEE Proceedings-Control Theory and Applications*, Vol. 142, No. 4, pp. 335-344, 1995.

- [24] B. Zhou, G. R. Duan, and H. Kong, Global stabilization of linear systems with bounded controls using state-dependent saturation functions, *Journal of Systems Science & Complexity*, Vol. 24, No. 3, pp. 477-490, 2011.
- [25] J. B. Rawlings and D. Q. Mayne, Model predictive control: theory and design, LLC, Madison: Nob Hill Publishing, 2009.
- [26] P. E. Wellstead and W. P. Heath, Two-dimensional control systems: application to the CD and MD control problem. A new theory considers the wet end as a 2-D dynamic process, *Pulp & Paper Canada*, Vol. 95, No. 2, pp. 48-51, 1994.
- [27] G. E. Stewart, D. M. Gorinevsky, and G. A. Dumont, Feedback controller design for a spatially distributed system: the paper machine problem, *IEEE Trans. on Control Systems Technology*, Vol. 11, No. 5, pp. 612-628, 2003.
- [28] O. J. Rojas, G. C. Goodwin, and G. V. Johnston, Spatial frequency anti-windup strategy for cross directional control problems, *IEE Proceedings-Control Theory and Applications*, Vol. 149, No. 5, pp. 414-422, 2002.
- [29] W. P. Heath, Orthogonal functions for cross directional control of web forming processes, *Automatica*, Vol. 32, No. 2, pp. 183-198, 1996.
- [30] G. C. Goodwin, S. F. Graebe, M. E. Salgado, Control System Design, Prentice-Hall, 2001.
- [31] G. C. Goodwin, Virtual laboratories for control system design (Paper Machine Package), <http://csd-new.newcastle.edu.au/>
- [32] A. P. Featherstone, J. G. VanAntwerp and R. D. Braatz, Identification and control of sheet and film processes, Springer Verlag, London, 2000.