A Low Rank Approach to Minimize Sensor-to-Actuator Communication in Finite Horizon Output Feedback

Antoine Aspeel[1], Jakob Nylof[2], Jing Shuang (Lisa) Li[1] and Necmiye Ozay[1]

Abstract— **Many modern controllers are composed of different components that communicate in real-time over some network with limited resources. In this work, we are interested in designing a controller that can be implemented with a minimum number of sensor-to-actuator messages, while satisfying safety constraints over a finite horizon. For finite horizon problems, a linear time-varying controller with memory can be represented as a block-lower-triangular matrix. We show that the rank of this matrix exactly captures the minimum number of messages needed to be sent from the sensors to actuators to implement such a controller. Moreover, we introduce a novel matrix factorization called *causal factorization* that gives the required implementation. Finally, we show that the rank of the controller is the same as the rank of the Youla parameter, enabling the Youla parametrization (or analogous parametrizations) to be used to design the controller, which reduces the overall design problem into a rank minimization one over a convex set. Finally, convex relaxations for rank are used to demonstrate that our approach leads to 20-50% less messages on a simulation than a benchmark method.**

I. Introduction

In a growing number of real-world applications (e.g., wireless sensor networks), controllers are implemented using distributed components (i.e., sensors and actuators) which must coordinate via limited resources. This work particularly concerns scenarios where sensors and actuators are not collocated, as in smart building heating systems, drone control in a motion capture arena, or on factory floors. In these cases, sensors and actuators may send messages to one another over a communication network with channel bandwidth constraints [1]. One resource-minimizing approach is to transmit fewer messages along the communication network to reduce the burden on the network. In this work, we consider the problem of designing a controller that can be implemented with a minimum number of sensor-to-actuator messages, while satisfying safety constraints.

1) Related works: The use of a communication network between distributed components to control a dynamic system induces numerous technical challenges (e.g., packet losses, communication delays and quantization due to limited bandwidth). The field of *networked control systems* has emerged to tackle these challenges. In this field, a common objective is to minimize the use of sensors and actuators [2], [3], [4], which gave rise to *self-* and *event-triggered controllers* [5]. Self-triggered controllers decide when to take the next action while computing the current action; event-triggered controllers trigger a control action only when the state (or state estimate) satisfies a certain condition. A related work [6] proposed a procedure to approximate a pre-designed linear stabilizing controller with a sensor- and an actuator-side controller that exchange messages only at given time instances to allow resource-aware implementations. However, the question of minimizing the number of messages is not as well characterized in the literature. In the context of static state-feedback quadratic control, a related recent work [7] proposed the use of a low-rank gain matrix to reduce the energy used in broadcast communication between agents over a wireless network. However, our work differs substantially from [7] as we consider the time-varying, output feedback, and safety-constrained setting. In particular, the time-varying formulation allows us to optimize "when" and "what" to communicate, where we show that rank indeed is the correct metric to minimize sensor-to-actuator communication.

In the related field of *distributed control*, several physically interconnected (i.e., dynamically coupled) subsystems equipped with local controllers communicate with each other via a communication network [8]. In this context, we might want to constrain which subsystems communicate with which others. This translates into sparsity constraints on the controller, which generally leads to NP-Hard problems [9]. While sparsity constraints are simple linear constraints on the controller gains, most problems are not convex in the controller gains directly, even in the centralized setting. This has led to the study of several parametrizations, like *Youla or Q-parametrization* [10] or *system-level synthesis* (SLS) [11], where the control design problem is rendered convex after a suitable nonlinear change of variables. Then, the complexity of the distributed control design problem depends on whether the constraints remain convex after the change of variables. Indeed, any convex constraint on the controller gain translates into a convex constraint over the Youla parameter if and only if it is *quadratically invariant* (QI) [12], [13]. However, not all sparsity constraints are QI [11, Section 3.5], limiting the use of Youla parametrization for distributed control. To circumvent these limitations, SLS was developed [11], [14] — a key feature of this approach is that it admits natural controller implementations in terms of the SLS parameters.

An alternative to having a fixed sparsity pattern constraining the communication architecture is to co-optimize the placement of sensors, actuators, and communication links by promoting sparsity [15], [16]. Convex relaxations of the sparsity maximization problem are proposed to minimize

*This work is funded by the ONR grant N00014-21-1-2431 (CLEVR-AI) and NSF Grant CNS-1837680.

[1]A. A., J. S. L., and N. O. are with the Electrical Engineering and Computer Science Department, Univ. of Michigan, Ann Arbor, MI antoinas,jslisali,necmiye@umich.edu.

[2]J. N. is with KTH Royal Institute of Technology, Stockholm, Sweden, and he was a visiting scholar at the Univ. of Michigan jnylof@kth.se.

the number of sensors, actuators, and communication links. Though very closely related to the problem we consider, this framework does not explicitly consider the problem of minimizing messages — in the aforementioned works, if a communication link is placed, it is assumed to be free of limitations (e.g., bandwidth). Also related are the works on sensor/actuator scheduling to maintain observability/controllability while promoting sparsity [17].

2) Contributions: We consider a linear dynamical system subject to safety constraints over a finite horizon and address the problem of synthesizing an output feedback control law with memory while *minimizing the number of messages sent from the sensors to the actuators*. In this finite-horizon setting, linear memoryful controllers can be represented by a block-lower-triangular matrix $\mathbf{K}$.

Our three main contributions are as follows. First, we prove that minimizing the number of messages can be done by minimizing the rank of $\mathbf{K}$. Second, we introduce the *causal factorization* of a matrix, and provide an algorithm to compute it. We prove that a minimum-message controller implementation can be obtained by computing the causal factorization of $\mathbf{K}$. Third, we prove that the rank of the controller is always equal to the rank of the Youla parameter, enabling the use of both Youla and SLS parametrization to design the controller. Finally, we use convex heuristics for rank minimization to demonstrate numerically that our method leads to fewer messages than sparsity-based approaches.

3) Notation: $\operatorname{rank} A$ and $\operatorname{Im} A$ denote the rank and image of the matrix A, respectively. $A_{i,:}$ is the i-th row of A and $A_{l:k,:}$ the submatrix of A formed by the rows $\{i \mid l \leq i \leq k\}$ (similarly for columns). $\operatorname{blkdiag}(A_1, \dots, A_n)$ denotes a block-diagonal matrix with diagonal blocks $A_1, \dots, A_n$. A matrix $A \in \mathbb{R}^{Tm \times Tn}$ is (m, n)-block-lower-triangular if $A_{tm+1:(t+1)m, \tau n+1:(\tau+1)n} = 0$ for all $0 \leq t < \tau \leq T-1$. $\mathcal{X} \times \mathcal{Y}$ is the Cartesian product between sets $\mathcal{X}$ and $\mathcal{Y}$, and $\mathcal{X}^n$ is the n-th Cartesian power of the set $\mathcal{X}$.

II. Problem statement

Consider a linear time-varying discrete time system

$$x_{t+1} = A_t x_t + B_t u_t + w_t, \qquad y_t = C_t x_t + v_t \tag{1}$$

where $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$, $w_t \in \mathbb{R}^{n_x}$, $y_t \in \mathbb{R}^{n_y}$, and $v_t \in \mathbb{R}^{n_y}$ are state, actuation, process noise, sensor measurement, and measurement noise, respectively. The finite horizon $t = 0, \dots, T$ is considered.

We aim to minimize sensor-to-actuator messages while constraining the system trajectories for any disturbances in a bounded set. In particular, we consider polyhedra $\mathcal{X}_t$, $\mathcal{W}_t \subset \mathbb{R}^{n_x}$, $\mathcal{V}_t \subset \mathbb{R}^{n_y}$, $\mathcal{U}_t \subset \mathbb{R}^{n_u}$ and define the following constraint.

Constraint 1 (Safety): For all $w_t \in \mathcal{W}_t$, $v_t \in \mathcal{V}_t$ for $t = 0, \dots, T-1$ and for all $x_0 \in \mathcal{X}_0$, it holds that $u_t \in \mathcal{U}_t$ for $t = 0, \dots, T-1$, and $x_t \in \mathcal{X}_t$ for $t = 1, \dots, T$.

Before formalizing the problem of message minimization, we consider a motivating example.

Example 1 (Minimizing messages): Let (1) be a SISO system ($n_y = n_u = 1$), $T = 4$ and let

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}, \ \mathbf{u} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}, \ \mathbf{K} = \begin{bmatrix} 5 & & & \\ 10 & 0 & & \\ 0 & 3 & 4 & \\ 15 & 6 & 8 & 0 \end{bmatrix}.$$

To implement the controller $\mathbf{u} = \mathbf{K}\mathbf{y}$, one can observe that $\mathbf{K}_{4,4} = 0$, and conclude that only 3 sensor-to-actuator messages are needed (y_3 is not used). However, we show that this controller can be implemented with only 2 such messages. First, note that $\mathbf{K} = \mathbf{D}\mathbf{E} := \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 0 & 1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 3 & 4 & 0 \end{bmatrix}$. This shows that the measurements can be encoded by the matrix $\mathbf{E}$ before the transmission and then be decoded using matrix $\mathbf{D}$ (see Fig. 1). On the sensor side, the message $m_1 = \mathbf{E}_{1,:}\mathbf{y} = 5y_0$ can be sent at time $t_1 = 0$, and the message $m_2 = \mathbf{E}_{2,:}\mathbf{y} = 3y_1 + 4y_2$ can be sent at time $t_2 = 2$. On the actuator side, the inputs can be recovered as $u_0 = 1m_1$, $u_1 = 2m_1$, $u_2 = 1m_2$, and $u_3 = 3m_1 + 2m_2$. Crucially, the causality is respected: each message is transmitted (i) after the encoded measurements have been measured, but (ii) before being used by the actuator. This allows sending only two messages. ▲

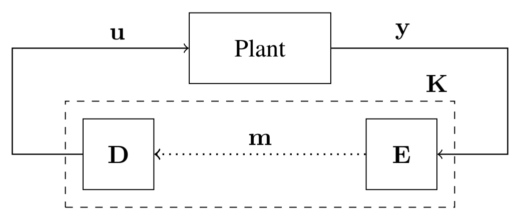


Fig. 1: Block representation of the encoder-decoder structure of the controller.

Inspired by the previous example, we define the controller with the following encoder-decoder structure: for $0 \leq t_1 \leq t_2 \leq \dots \leq t_r \leq T$, let

$$m_k = \sum_{\tau \leq t_k} e_{(k,\tau)}^\top y_\tau \text{ and } u_t = \sum_{k \text{ s.t. } t_k \leq t} d_{(t,k)} m_k \tag{2}$$

with $e_{(k,\tau)} \in \mathbb{R}^{n_y}$, $d_{(t,k)} \in \mathbb{R}^{n_u}$, and each $m_k \in \mathbb{R}$ is a message sent from the sensor to the actuator at time t_k (each message is a real number). For the SISO system in Example 1, $e_{(k,\tau)}^\top = \mathbf{E}_{k,\tau+1} \in \mathbb{R}$ and $d_{(t,k)} = \mathbf{D}_{t+1,k} \in \mathbb{R}$.

Remark 1: For simplicity, we consider the messages m_k sent from sensors to actuators to be real numbers, which can in theory require infinite bandwidth. In practice, the messages exchanged must be quantized, e.g., implemented in fixed-point arithmetic. This can be handled by including rounding errors in the noise terms in (1). A more indepth study of this issue to obtain information-theoretic bounds as in [18], [19], and references therein, is left for future work.⧫

The problem of minimizing the number r of sensor-to-actuator messages can be written as follows.

Problem 1: Find the minimal r such that there exist $\{t_k\}_{k=1}^r$, $\{e_{(k,\tau)}\}_{k=1,\dots,r}^{\tau=0,\dots,t_k}$ and $\{d_{(t,k)}\}_{k=1,\dots,r}^{t=0,\dots,t_k}$ satisfying (1), (2) and Constraint 1.

Remark 2: Enforcing $d_{(t,l)} = d_{(t_k,l)}$ for all $l \leq k$ is equivalent to enforcing zero order hold constraints $u_t = u_{t_k}$ for all t such that $t_k \leq t < t_{k+1}$. As such, optimizing over the $d_{(t,l)}$ can be interpreted as optimizing the (linear) holding mechanism, i.e., what control input should be used when there is no new measurement. ♦

III. Method

The controller structure (2) can be rewritten compactly as $\mathbf{m} = \mathbf{E}\mathbf{y}$, $\mathbf{u} = \mathbf{D}\mathbf{m}$ by defining $\mathbf{u} \coloneqq \begin{bmatrix} u_0^\top & \dots & u_T^\top \end{bmatrix}^\top$, $\mathbf{y} \coloneqq \begin{bmatrix} y_0^\top & \dots & y_T^\top \end{bmatrix}^\top$, $\mathbf{m} \coloneqq \begin{bmatrix} m_1^\top & \dots & m_r^\top \end{bmatrix}^\top$ and

$$\mathbf{D} \coloneqq \begin{bmatrix} d_{(0,1)} & \cdots & d_{(0,r)} \\ \vdots & & \vdots \\ d_{(T,1)} & \cdots & d_{(T,r)} \end{bmatrix}, \quad \mathbf{E} \coloneqq \begin{bmatrix} e_{(1,0)}^\top & \cdots & e_{(1,T)}^\top \\ \vdots & & \vdots \\ e_{(r,0)}^\top & \cdots & e_{(r,T)}^\top \end{bmatrix},$$

with

$$e_{(k,\tau)} = 0 \text{ when } \tau > t_k, \text{ and } d_{(t,k)} = 0 \text{ when } t_k > t. \quad (3)$$

Matrices $\mathbf{E}$ and $\mathbf{D}$ can be interpreted as encoders and decoders, respectively. Note that because of (3) they satisfy the following causality constraints.

Constraint 2 (Causal encoder/decoder): $\mathbf{E}_{k,\tau n_y + j} = \mathbf{D}_{t n_u + i,k} = 0$ for all $k = 1, \dots, r$, for all $\tau > t_k > t$, for all $i = 1, \dots, n_u$, and for all $j = 1, \dots, n_y$.

Constraint 2 (or equivalently, conditions (3)) states that a message transmitted at time t_k can neither (i) encode measurements received after t_k, nor (ii) be used before t_k. Importantly, any matrices satisfying Constraint 2 define some $\{e_{(k,\tau)}\}_{k=1,\dots,r}^{\tau=0,\dots,t_k}$ and $\{d_{(t,k)}\}_{k=1,\dots,r}^{t=0,\dots,t_k}$ in (2). Consequently, one can optimize over the set of matrices $\mathbf{D}$ and $\mathbf{E}$ satisfying Constraint 2 instead of optimizing over $e_{(k,\tau)}$ and $d_{(t,k)}$.

A. Reformulation as a controller with memory

The encoder-decoder structure (2) of the controller can be written as a linear time-varying output feedback controller with memory

$$u_t = \sum_{\tau \leq t} K_{(t,\tau)} y_\tau, \quad (4)$$

with $K_{(t,\tau)} = \sum_{k \text{ s.t. } \tau \leq t_k \leq t} d_{(t,k)} e_{(k,\tau)}^\top$. We use the following notation

$$\mathbf{K} \coloneqq \begin{bmatrix} K_{(0,0)} & & & \\ K_{(1,0)} & K_{(1,1)} & & \\ \vdots & \ddots & \ddots & \\ K_{(T,0)} & \dots & K_{(T,T-1)} & K_{(T,T)} \end{bmatrix}. \quad (5)$$

The controller structure can be written compactly as $\mathbf{u} = \mathbf{K}\mathbf{y}$. It follows that $\mathbf{K} = \mathbf{D}\mathbf{E}$ is (n_u, n_y)-block-lower-triangular.

We aim to solve Problem 1 by optimizing over $\mathbf{K}$ instead of $(\mathbf{D}, \mathbf{E})$ to avoid the bilinear term $\mathbf{D}\mathbf{E}$. To do so, we need to know in which cases and how matrices $(\mathbf{D}, \mathbf{E})$ can be recovered from $\mathbf{K}$. This is captured by the notion of *causal factorization*.

Definition 1 (Causal factorization): Let $\mathbf{K} \in \mathbb{R}^{(T+1)n_u \times (T+1)n_y}$ be a (n_u, n_y)-block-lower-triangular matrix. A pair of matrices $(\mathbf{D}, \mathbf{E}) \in \mathbb{R}^{(T+1)n_u \times r} \times \mathbb{R}^{r \times (T+1)n_y}$ is a *causal factorization of* $\mathbf{K}$ *with band* r if $\mathbf{K} = \mathbf{D}\mathbf{E}$ and there exist integers $0 \leq t_1 \leq t_2 \leq \cdots \leq t_r \leq T$ such that Constraint 2 (causality) holds.

Note that because $\mathbf{K} = \mathbf{D}\mathbf{E}$, there is no causal factorization with band smaller than $\operatorname{rank} \mathbf{K}$. We now state our first main result.

Theorem 1: Any (n_u, n_y)-block-lower-triangular matrix $\mathbf{K} \in \mathbb{R}^{(T+1)n_u \times (T+1)n_y}$ admits a causal factorization $(\mathbf{D}, \mathbf{E})$ with band equal to $\operatorname{rank} \mathbf{K}$. In addition, Algorithm 1 returns such a causal factorization.

Before proving Theorem 1, let us discuss the intuition of Algorithm 1. The matrix $\mathbf{E}$ is constructed as follows: a row of $\mathbf{K}$ is added to $\mathbf{E}$ only if it is linearly independent of the rows preceding it. In this way, when a line from $\mathbf{K}$ is not added to $\mathbf{E}$, it can be reconstructed (linearly) from the preceding lines in $\mathbf{E}$ (this is how $\mathbf{D}$ is constructed).

Proof: It is enough to prove the second claim of Theorem 1. To this end, first, note that for all $l = 1, \dots, (T+1)n_u$:

$$\mathbf{K}_{l,:} \in \operatorname{Im} \mathbf{K}_{1:l,:} = \operatorname{Im} \begin{bmatrix} \mathbf{K}_{c_1,:} \\ \mathbf{K}_{c_2,:} \\ \vdots \\ \mathbf{K}_{c_{r_l},:} \end{bmatrix} = \operatorname{Im} \mathbf{E}_{1:r_l,:},$$

where the first equality follows from the definition of c_k (line 6), and the second equality follows from the definition of $\mathbf{E}$ (line 7). Consequently, the system of linear equations in line 10 always has a solution, and $\mathbf{K} = \mathbf{D}\mathbf{E}$.

For $k = 1, \dots, r$, let t_k be such that $c_k = t_k n_u + i_k$ for some $i_k \in \{1, \dots, m\}$. Then, $0 \leq t_1 \leq \cdots \leq t_r \leq T$.

Let $k \in \{1, \dots, r\}$, $\tau > t_k$ and $j \in \{1, \dots, n_y\}$. We need to prove that $\mathbf{E}_{k,\tau n_y + j} = 0$. By definition of $\mathbf{E}$ (line 7), $\mathbf{E}_{k,\tau n_y + j} = \mathbf{K}_{c_k,\tau n_y + j}$. But by definition of t_k, $c_k \leq t_k n_u + n_u < \tau n_u + 1$. It follows from the block-lower-triangularity of $\mathbf{K}$ that $\mathbf{E}_{k,\tau n_u + j} = 0$.

Let $k \in \{1, \dots, r\}$, $t < t_k$ and $i \in \{1, \dots, n_u\}$. We need to prove that $\mathbf{D}_{t n_u + i,k} = 0$. Note that $t n_u + i \leq (t+1) n_u \leq t_k n_u < t_k n_u + i_k = c_k$. Then, by definition of c_k (line 6), $r_{t n_u + i} < r_{c_k} = k$. It follows from the definition of $\mathbf{D}$ (see line 11 with $l = t n_u + i$) that $\mathbf{D}_{t n_u + i,k} = 0$. ■

Algorithm 1 Causal factorization

Require: $\mathbf{K} \in \mathbb{R}^{(T+1)n_u \times (T+1)n_y}$, (n_u, n_y)-block-lower-triangular.

```
1:  for l = 1, ..., (T+1)n_u do
2:      r_l := rank K_{1:l,:}
3:  end for
4:  r := r_{(T+1)n_u}                              ▷ r = rank K
5:  for k = 1, ..., r do                           ▷ Compute E
6:      c_k := min{l | r_l = k}
7:      E_{k,:} = K_{c_k,:}
8:  end for
9:  for l = 1, ..., (T+1)n_u do                    ▷ Compute D
10:     Find D_{l,1:r_l} such that K_{l,:} = D_{l,1:r_l} E_{1:r_l,:}
11:     D_{l,r_l+1:r} := 0
12: end for
13: return (D, E)
```

Causal factorization is not unique. Indeed, for any invertible diagonal matrix Λ, the pairs $(\mathbf{D}, \mathbf{E})$ and $(\mathbf{D}\Lambda^{-1}, \Lambda\mathbf{E})$ have the same band and factorize the same matrix. The factorization in Example 1 is the one computed using Algorithm 1.

It follows from Theorem 1 that Problem 1 can be formulated as a rank minimization problem over $\mathbf{K}$.

Corollary 1: Optimal r, $\{t_k\}_{k=1}^r$, $\{e_{(k,\tau)}\}_{k=1,\dots,r}^{\tau=0,\dots,t_k}$ and $\{d_{(t,k)}\}_{k=1,\dots,r}^{t=0,\dots,t_k}$ for Problem 1 can be obtained by finding an optimal $\mathbf{K}^*$ for

$$\min_{\mathbf{K}} \ \operatorname{rank} \ \mathbf{K} \text{ s.t. (1), (4), (5), Constraint 1,} \tag{6}$$

and computing a causal factorization of $\mathbf{K}^*$ with band equal to $\operatorname{rank}\mathbf{K}^*$.

Proof: Let $r \in \mathbb{N}$ and $\{t_k\}_{k=1}^r$. As already noted in Section III, there exist vectors $d_{(t,k)}$ and $e_{(k,\tau)}$ satisfying (2) if and only if there exist matrices $(\mathbf{D}, \mathbf{E}) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{r\times m}$ satisfying $\mathbf{u} = \mathbf{DEy}$ and Constraint 2. Such matrices exist if and only if there exist a block-lower-triangular $\mathbf{K}$ such that $\mathbf{u} = \mathbf{Ky}$ that admits a causal factorization with band r (this follows from the definition of causal factorization). Because the band of a causal factorization can not be smaller than the rank, and by Theorem 1, $\mathbf{K}$ admits a causal factorization with band r if and only if $\operatorname{rank}\mathbf{K} \leq r$. Consequently, minimizing r is equivalent to minimizing $\operatorname{rank}\mathbf{K}$. Finally, note that $\mathbf{u} = \mathbf{Ky}$ and the block-lower-triangularity of $\mathbf{K}$ are equivalent to (4) and (5). This concludes the proof. ■

Remark 3 (Relation to sparsity): The *regularization for design* framework [16] proposes a method to optimize (and minimize) placement of sensors, actuators, and communication links using sparsity-based approaches. This method was originally developed for static infinite-horizon controllers, but can be adapted directly to time-varying controllers over a finite horizon. In a time-varying controller, regularization for design can be used to optimize not only sensor/actuator placements but also sensor/actuator usage. For example, minimizing the use of actuators can be written $\min \sum_l \|\mathbf{K}_{l,:}\|_0$. Indeed, $\mathbf{K}_{tm+i,:} = 0$ implies $u_i(t) = 0$ and actuator i is not used at time t. Though closely related to the problem at hand, this sparsity-based approach does not explicitly consider message minimization.

B. System level synthesis (SLS)

To handle the safety Constraint 1 (which involves *for all* quantifiers), we follow [20] and use SLS to rewrite it linearly in terms of SLS parameters (note that [20] considers state feedback). Then, we prove that rank minimization of $\mathbf{K}$ has a natural reformulation in terms of SLS (or Youla) parameters, allowing us to rewrite (6) as a rank minimization problem subject to linear constraints.

We make the following definitions: $\mathbf{x} := \begin{bmatrix} x_0^\top & \dots & x_T^\top \end{bmatrix}^\top$, $\mathbf{w} := \begin{bmatrix} x_0^\top & w_0^\top & \dots & w_{T-1}^\top \end{bmatrix}^\top$, $\mathbf{v} := \begin{bmatrix} v_0^\top & \dots & v_T^\top \end{bmatrix}^\top$, the matrix Z is the block-downshift operator (identity matrices on the first sub-diagonal and zeros elsewhere) and $\mathcal{A} := \operatorname{blkdiag}(A_0, \dots, A_{T-1}, 0)$, $\mathcal{B} := \operatorname{blkdiag}(B_0, \dots, B_{T-1}, 0)$, $\mathcal{C} := \operatorname{blkdiag}(C_0, \dots, C_T)$.

We have that (1) and (2) can be written $\mathbf{x} = Z\mathcal{A}\mathbf{x} + Z\mathcal{B}\mathbf{u} + \mathbf{w}$, $\mathbf{y} = \mathcal{C}\mathbf{x} + \mathbf{v}$, $\mathbf{u} = \mathbf{Ky}$ and equivalently

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_{xx} & \mathbf{\Phi}_{xy} \\ \mathbf{\Phi}_{ux} & \mathbf{\Phi}_{uy} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix}, \tag{7}$$

with $\mathbf{\Phi}_{xx} = (I - Z\mathcal{A} - Z\mathcal{B}\mathbf{K}\mathcal{C})^{-1}$, $\mathbf{\Phi}_{xy} = \mathbf{\Phi}_{xx} Z\mathcal{B}\mathbf{K}$, $\mathbf{\Phi}_{ux} = \mathbf{K}\mathcal{C}\mathbf{\Phi}_{xx}$ and $\mathbf{\Phi}_{uy} = \mathbf{K} + \mathbf{K}\mathcal{C}\mathbf{\Phi}_{xx}Z\mathcal{B}\mathbf{K}$. The following proposition is the basis for finite horizon output feedback SLS and gives a condition under which all block-lower triangular controllers $\mathbf{K}$ can be parameterized by block-lower triangular system response $\{\mathbf{\Phi}_{xx}, \mathbf{\Phi}_{xy}, \mathbf{\Phi}_{ux}, \mathbf{\Phi}_{uy}\}$ (and vice versa).

Proposition 1 (Adapted from [21, Lemma 1]): Over the horizon $t = 0, \dots, T$, the system dynamics (1) with controller (2), the following are true:

1) the affine subspace defined by

$$\begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_{xx} & \mathbf{\Phi}_{xy} \\ \mathbf{\Phi}_{ux} & \mathbf{\Phi}_{uy} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{8a}$$

$$\begin{bmatrix} \mathbf{\Phi}_{xx} & \mathbf{\Phi}_{xy} \\ \mathbf{\Phi}_{ux} & \mathbf{\Phi}_{uy} \end{bmatrix} \begin{bmatrix} I - Z\mathcal{A} \\ -\mathcal{C} \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{8b}$$

parameterizes all possible system responses (7).

2) for any block-lower-triangular matrices $\{\mathbf{\Phi}_{xx}, \mathbf{\Phi}_{xy}, \mathbf{\Phi}_{ux}, \mathbf{\Phi}_{uy}\}$ satisfying (8), the controller $\mathbf{K} = \mathbf{\Phi}_{uy} - \mathbf{\Phi}_{ux}\mathbf{\Phi}_{xx}^{-1}\mathbf{\Phi}_{xy}$ achieves the desired system response (7).

The safety Constraint 1 can be handled linearly thanks to SLS. Indeed, it can be written $\tilde{\mathbf{\Phi}}\mathcal{N} \subseteq \mathcal{S}$, where $\mathcal{N} := \mathcal{X}_0 \times \bigtimes_{t=0}^{T-1} \mathcal{W}_t \times \bigtimes_{t=0}^{T-1} \mathcal{V}_t$, $\mathcal{S} := \bigtimes_{t=1}^{T} \mathcal{X}_t \times \bigtimes_{t=0}^{T-1} \mathcal{U}_t$, and

$$\tilde{\mathbf{\Phi}} := \begin{bmatrix} (\mathbf{\Phi}_{xx})_{n_x+1:(T+1)n_x,:} & (\mathbf{\Phi}_{xy})_{n_x+1:(T+1)n_x,:} \\ (\mathbf{\Phi}_{ux})_{1:Tn_u,:} & (\mathbf{\Phi}_{uy})_{1:Tn_u,:} \end{bmatrix}.$$

By Farkas' lemma, this is equivalent to the existence of a matrix Λ such that

$$\Lambda \geq 0, \quad \Lambda H_{\mathcal{N}} = H_{\mathcal{S}}\tilde{\mathbf{\Phi}}, \quad \Lambda h_{\mathcal{N}} \leq h_{\mathcal{S}}, \tag{9}$$

where we used the H-representation $\mathcal{P} = \{p | H_{\mathcal{P}} p \leq h_{\mathcal{P}}\}$ for $\mathcal{P} \in \{\mathcal{N}, \mathcal{S}\}$. Importantly, these constraints are linear in Λ and the SLS parameters.[1]

To use SLS to solve problem (6), we also need to relate the rank of $\mathbf{K}$ to the SLS parameters. That is our second main contribution.

Theorem 2: For any block-lower-triangular matrices $\{\mathbf{\Phi}_{xx}, \mathbf{\Phi}_{xy}, \mathbf{\Phi}_{ux}, \mathbf{\Phi}_{uy}\}$ satisfying (8), $\operatorname{rank}\mathbf{\Phi}_{uy} = \operatorname{rank}\mathbf{K}$, where $\mathbf{K} := \mathbf{\Phi}_{uy} - \mathbf{\Phi}_{ux}\mathbf{\Phi}_{xx}^{-1}\mathbf{\Phi}_{xy}$.

Proof: Assuming (8), it follows from the second statement in Proposition 1 that equation (7) holds. Then, one can write $\mathbf{\Phi}_{uy} = (I + \mathbf{\Phi}_{ux}Z\mathcal{B})\mathbf{K}$. Because $\mathbf{\Phi}_{ux}$, Z and $\mathcal{B}$ are block-lower-triangular, strictly block-lower-triangular and block diagonal, respectively, $\mathbf{\Phi}_{ux}Z\mathcal{B}$ is strictly block-lower diagonal and $(I + \mathbf{\Phi}_{ux}Z\mathcal{B})$ is invertible. Consequently, $\operatorname{rank}\mathbf{K} = \operatorname{rank}\mathbf{\Phi}_{uy}$. ■

Remark 4: The matrix $\mathbf{\Phi}_{uy}$ is the Youla parameter (see [10, Equation (11)]), so this theorem states that the controller

[1] Polytope containment constraints can also be handled using Youla parametrization (see e.g., [2, Lemmas 3 and 4]).

and the Youla parameter have the same rank. We note that the constraint $\mathbf{K} \in \{\mathbf{K} \mid \operatorname{rank} \mathbf{K} \leq r\}$ is QI[2] for any plant, while $\mathbf{K} \in \{\mathbf{K} \mid \operatorname{rank} \mathbf{K} = r\}$ is not.

Theorem 2 suggests that the minimal number of sensor-to-actuator messages does not depend on the parametrization — in particular, message minimization can be done using the standard, Youla, or SLS parametrization. SLS may be preferable as it allows us to consider extra constraints that are not QI; in Section III-D, we show that SLS can handle cases where (i) several sensors do not share their measurements and (ii) several actuators do not share the messages they receive (whereas Youla parametrization does not allow this). ⧫

It follows from Proposition 1, the discussion on polytope containment, and Theorem 2 that (6) can be rewritten as a rank minimization problem subject to linear constraints.

Corollary 2: An optimal solution $\mathbf{K}^*$ to Problem (6) is given by an optimal solution $\{\mathbf{\Phi}_{xx}^*, \mathbf{\Phi}_{xy}^*, \mathbf{\Phi}_{ux}^*, \mathbf{\Phi}_{uy}^*\}$ of

$$\begin{aligned} \min_{\mathbf{\Phi}_{xx},\mathbf{\Phi}_{xy},\mathbf{\Phi}_{ux},\mathbf{\Phi}_{uy},\Lambda} \quad & \operatorname{rank}\ \mathbf{\Phi}_{uy} \\ \text{s.t.} \quad & (8), (9) \text{ and } \mathbf{\Phi}_{xx},\ \mathbf{\Phi}_{xy}, \mathbf{\Phi}_{ux},\ \mathbf{\Phi}_{uy} \\ & \text{are } (n_u, n_y)\text{-block-lower-triangular,} \end{aligned} \tag{10}$$

with $\mathbf{K}^* = \mathbf{\Phi}_{uy}^* - \mathbf{\Phi}_{ux}^* (\mathbf{\Phi}_{xx}^*)^{-1} \mathbf{\Phi}_{xy}^*$.

Remark 5 (Relation to sparsity continued): Maximizing sparsity typically corresponds to minimizing the use of sensors/actuators. For example, minimizing the use of actuators reduces to maximizing the row sparsity of $\mathbf{K}$. Since row-sparsity is QI, $\min \sum_l \|\mathbf{K}_{l,:}\|_0 = \min \sum_l \|[\mathbf{\Phi}_{uy}]_{l,:}\|_0$.⧫

C. Numerical considerations

Problem (10) is a rank minimization problem which is NP-hard. To solve it approximately, we use the *reweighted nuclear norm heuristic* described in [22, Section III] with regularization parameter $\delta = 0.01$.

Once $\{\mathbf{\Phi}_{xx}^*, \mathbf{\Phi}_{xy}^*, \mathbf{\Phi}_{ux}^*, \mathbf{\Phi}_{uy}^*\}$ have been obtained, we compute the corresponding gain matrix $\mathbf{K}^* = \mathbf{\Phi}_{uy}^* - \mathbf{\Phi}_{ux}^* (\mathbf{\Phi}_{xx}^*)^{-1} \mathbf{\Phi}_{xy}^*$ (see Proposition 1). Then, its causal factorization $(\mathbf{D}_\epsilon, \mathbf{E}_\epsilon)$ is computed using Algorithm 1 (the ranks in line 2 of the algorithm are computed with respect to a tolerance $\epsilon > 0$). To handle the factorization error $\mathbf{K}^* \approx \mathbf{K}_\epsilon := \mathbf{D}_\epsilon \mathbf{E}_\epsilon$ due to $\epsilon \neq 0$, the feasibility of $\mathbf{K}_\epsilon$ is then checked by computing the corresponding $\{\mathbf{\Phi}_{xx}^\epsilon, \mathbf{\Phi}_{xy}^\epsilon, \mathbf{\Phi}_{ux}^\epsilon, \mathbf{\Phi}_{uy}^\epsilon\}$ (see equations following (7)).

Remark 6 (Relation to sparsity continued): Like minimizing rank, maximizing row sparsity is NP-hard. However, the 0-norm can be replaced by the 2-norm, leading to the (convex) actuator norm considered in [16]. Alternatively, the row sparsity can be optimized using a reweighting heuristic [23]. From a computational point of view, minimizing the nuclear norm is a semidefinite program, while minimizing the sensor/actuator norm is a second order cone program (which can be done more efficiently). Another notable difference between sparsity and rank optimization is as follows: when maximizing sparsity, the controller can be re-optimized after the sparsity pattern has been fixed, since imposing a sparsity pattern is a linear constraint. On the contrary, the controller can not be re-optimized after the minimum rank has been found because rank constraints are not convex. ⧫

D. Multiple sensors and actuators

Above, we assumed that there is only one sensor and one actuator (both of which make vector measurements/actuations). The case where several sensors do not share their measurements and several actuators do not share the messages they receive can be handled thanks to SLS (but can not be handled with Youla parametrization). For the sake of explanation, assume that each sensor j makes a scalar measurement $y_j(t) \in \mathbb{R}$ and that each actuator i actuates a scalar input $u_i(t) \in \mathbb{R}$. Using, for example, the controller implementation in [11, Fig. 11(c)], the following sparsity constraints can be used to prevent communication between sensors: $[\mathcal{C}\mathbf{\Phi}_{xy}]_{tn+j_1,\tau n+j_2} = 0$, for all t, τ and $j_1 \neq j_2$. Then, the number of messages sent from sensor j to actuator i is given by $\operatorname{rank} \mathbf{\Phi}_{uy}^{(i,j)}$, where $[\mathbf{\Phi}_{uy}^{(i,j)}]_{t,\tau} := [\mathbf{\Phi}_{uy}]_{tm+i,\tau n+j}$. Finally, minimizing the total number of messages is solving $\min \sum_i \sum_j \operatorname{rank} \mathbf{\Phi}_{uy}^{(i,j)}$. A causal factorization can be computed for each $\mathbf{\Phi}_{uy}^{(i,j)}$ independently.

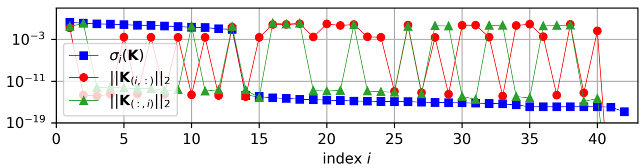
IV. Numerical demonstrations

To illustrate our method,[3] we consider a drone represented by the two-dimensional double integrator dynamics $\ddot{p}^x = u^x$, $\ddot{p}^y = u^y$, where (p^x, p^y) represent the (x, y)-position of a drone subject to a force (u^x, u^y). The state of the system is defined as $x = \begin{bmatrix} p^x & p^y & \dot{p}^x & \dot{p}^y \end{bmatrix}^\top$. The dynamics is exactly discretized with unit discretization step and $T = 20$. A process noise $w_t \in [-0.05, 0.05]^4$ is considered. The position $\begin{bmatrix} p^x & p^y \end{bmatrix}^\top$ is measured with some additive noise $v_t \in [-0.05, 0.05]^2$. The initial state is $x_0 \in [-8, -6]^2 \times \{0\}^2$. The input constraints are $u_t \in [-2, 2]$. The time-varying safety constraints over the state are $x_{10} \in [5, 9] \times [-9, -5] \times [-2, 2]^2$, $x_{20} \in [5, 9]^2 \times [-1, 1]^2$, and $x_t \in [-10, 10]^2 \times [-2, 2]^2$ at all other time steps.

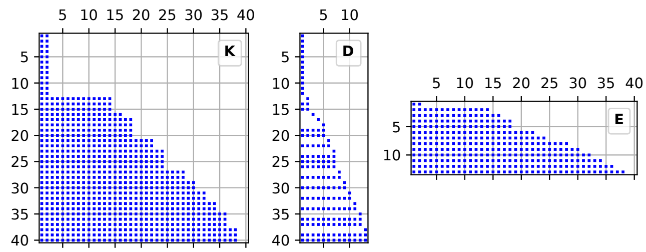
We compare our method to the sensor/actuator norms (see Remarks 3, 5, 6) using the same reweighting heuristic [23]. Fig. 2a shows the singular values (resp. the 2-norm of columns/rows) of $\mathbf{K}$. The large gaps indicate a small truncation error. Minimizing the rank took 205 seconds, while maximizing the column and row sparsity took 33 and 27 seconds, respectively. Fig. 2b presents the sparsity of the gain obtained from rank minimization and its causal factorization (computed in 0.01 seconds). Fig. 2c shows the sparsity of the gain obtained by actuator and sensor norm minimization. Our method requires 13 messages (the band of the factorization), while sensor-/actuator-norms require 16 and 26 messages, respectively (the number of non-zero columns/rows). Finally, Fig. 2d presents trajectories using the controller synthesized with our method. As expected, the trajectories satisfy the constraints.

[2] A set Ω is QI with respect to a plant if $\mathbf{K}P_{22}\mathbf{K} \in \Omega$ for all $\mathbf{K} \in \Omega$, where the matrix P_{22} depends on the plant.

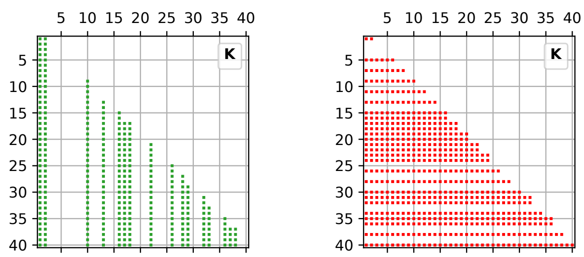
[3] The code that generates the figures and implements our algorithm is available at: `https://github.com/aaspeel/lowRankControl`. The reported computation times are obtained using a laptop with a Quad-Core Intel i7 CPU and 16 GB of RAM.



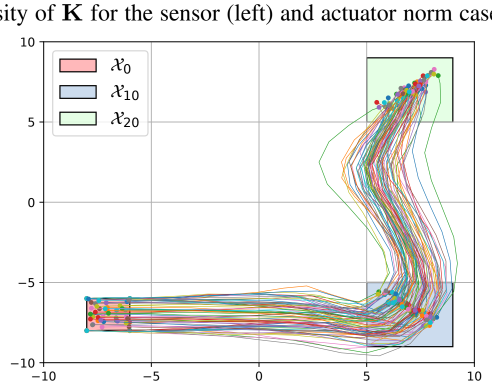
(a) Ordered singular values of $\mathbf{K}$ for the nuclear norm case (squares). Column/row 2-norms of $\mathbf{K}$ for the sensor/actuator norm case (triangles/circles) after 8 iterations of the reweighting heuristics.



(b) Sparsity of $\mathbf{K}$ and its causal factorization $(\mathbf{D}, \mathbf{E})$ for the nuclear norm case.



(c) Sparsity of $\mathbf{K}$ for the sensor (left) and actuator norm case (right).



(d) (p_t^x, p_t^y) for the nuclear norm case (40 trajectories were generated from uniformly sampled noises and another 40 from noises over its vertices). The dots indicate the positions at times $t = 0$, 10 and 20.

Fig. 2: Comparison of the solutions for the nuclear, sensor and actuator norm optimization problems.

V. Conclusion

We addressed the problem of synthesizing a controller that can be implemented with a minimum number of sensor-to-actuator messages. Our results show that for this objective, when considering linear time-varying controllers with memory, controller implementation benefits from having some computation on the sensor side (encoder) and some on the actuator side (decoder) as opposed to being fully collocated with the sensor or actuator. Our method relies on first minimizing the rank of the Youla parameter that we showed to be the same as the rank of the controller (with minimum rank giving the least number of message transmissions) and then computing the causal factorization of the controller. Future work will investigate infinite horizon counterparts of this method.

References

[1] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzen, "How does control timing affect performance? Analysis and simulation of timing using jitterbug and truetime," *IEEE Contr. Systems Magazine*, vol. 23, no. 3, pp. 16–30, 2003.

[2] A. Aspeel, K. Rutledge, R. M. Jungers, B. Macq, and N. Özay, "Optimal control for linear networked control systems with information transmission constraints," in *2021 60th IEEE Conf. on Decision and Contr. (CDC)*. IEEE, 2021, pp. 1960–1967.

[3] M. H. Balaghi, D. J. Antunes, and W. M. Heemels, "An l 2-consistent data transmission sequence for linear systems," in *2019 IEEE 58th Conf. on Decision and Contr. (CDC)*. IEEE, 2019, pp. 2622–2627.

[4] Y. Long, S. Liu, and L. Xie, "Stochastic channel allocation for networked control systems," *IEEE Contr. Systems Letters*, vol. 1, no. 1, pp. 176–181, 2017.

[5] W. P. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *2012 IEEE 51st IEEE Conf. on Decision and Contr. (CDC)*. IEEE, 2012, pp. 3270–3285.

[6] M. Braksmayer and L. Mirkin, "Redesign of stabilizing discrete-time controllers to accommodate intermittent sampling," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2633–2638, 2017.

[7] M. Cho, A. Abdallah, and M. Rasouli, "Low-rank LQR optimal control design for controlling distributed multi-agent systems," in *2023 European Contr. Conf. (ECC)*, 2023, pp. 1–6.

[8] G. Antonelli, "Interconnected dynamic systems: An overview on distributed control," *IEEE Contr. Systems Magazine*, vol. 33, no. 1, pp. 76–88, 2013.

[9] Y. Wang, J. A. Lopez, and M. Sznaier, "Convex optimization approaches to information structured decentralized control," *IEEE Trans. on Autom. Contr.*, vol. 63, no. 10, pp. 3393–3403, 2018.

[10] J. Skaf and S. P. Boyd, "Design of affine controllers via convex optimization," *IEEE Trans. on Autom. Contr.*, vol. 55, no. 11, pp. 2476–2487, 2010.

[11] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, "System level synthesis," *Annual Reviews in Contr.*, vol. 47, pp. 364–393, 2019.

[12] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Trans. on Autom. Contr.*, vol. 50, no. 12, pp. 1984–1996, 2005.

[13] L. Lessard and S. Lall, "Quadratic invariance is necessary and sufficient for convexity," in *Proc. of the 2011 American Contr. Conf.* IEEE, 2011, pp. 5360–5362.

[14] C. Amo Alonso, J. S. Li, J. Anderson, and N. Matni, "Distributed and localized model predictive control. part i: Synthesis and implementation," *IEEE Trans. on Contr. of Network Systems*, 2022.

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

[16] N. Matni and V. Chandrasekaran, "Regularization for design," *IEEE Trans. on Autom. Contr.*, vol. 61, no. 12, pp. 3991–4006, 2016.

[17] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Trans. on Autom. Contr.*, vol. 66, no. 4, pp. 1686–1701, 2020.

[18] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Trans. on Autom. Contr.*, vol. 49, no. 7, pp. 1056–1068, 2004.

[19] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: An overview," *Proc. of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[20] Y. Chen and J. Anderson, "System level synthesis with state and input constraints," in *2019 IEEE 58th Conf. on Decision and Contr. (CDC)*. IEEE, 2019, pp. 5258–5263.

[21] S. M. Hassaan and S. Z. Yong, "System-level recurrent state estimators for affine systems subject to data losses modeled by automata," in *2022 IEEE 61st Conf. on Decision and Contr. (CDC)*. IEEE, 2022, pp. 4118–4124.

[22] K. Mohan and M. Fazel, "Reweighted nuclear norm minimization with application to system identification," in *Proc. of the 2010 American Contr. Conf.* IEEE, 2010, pp. 2953–2959.

[23] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization," *J. of Fourier analysis and applications*, vol. 14, pp. 877–905, 2008.