Co-Design of Delays and Sparse Controllers for Bandwidth-Constrained Cyber-Physical Systems

Nandini Negi1,2 and Aranya Chakrabortty1,3

1Electrical & Computer Engineering, North Carolina State University
Email: 2nnegi@ncsu.edu, 3aranya.chakrabortty@ncsu.edu

Abstract—We address the problem of sparsity-promoting optimal control of cyber-physical systems with feedback delays. The delays are categorized into two classes - namely, intra-layer delay, and inter-layer delay between the cyber and the physical layers. Our objective is to minimize the $H_2$-norm of the closed-loop system by designing an optimal combination of these two delays along with a sparse state-feedback controller, while respecting a given bandwidth constraint. We propose a two-loop optimization algorithm for this. The inner loop, based on alternating directions method of multipliers (ADMM), handles the conflicting directions of decreasing $H_2$-norm and increasing sparsity of the controller. The outer loop comprises of semidefinite program (SDP)-based relaxations of non-convex inequalities necessary for stable co-design of the delays with the controller. We illustrate this algorithm using simulations that highlight various aspects of how delays and sparsity impact the stability and $H_2$-performance of a LTI system.

I. INTRODUCTION

In recent years, sparsity-promoting optimal control has emerged as a key tool for enabling economical control of large-scale cyber-physical systems (CPSs). These designs use optimization algorithms such as ADMM [1], LASSO [2], GraSP [3], and PALM [4]. An extension of these results to LTI systems with communication delays has been reported in [5]. Since most real-world CPSs operate under stringent constraints for bandwidth, stability and closed-loop performance in the presence of delays are important requirements for these controllers [6]. Accordingly, the algorithm in [5] derives convex relaxations of bilinear matrix inequalities to design a sparse controller, while guaranteeing closed-loop stability under a constant delay.

In this paper, we extend the design in [5] one step further by considering the delays themselves as design variables. Our formulation is motivated by modern CPS communication technologies such as software-defined networking (SDN) and cloud computing that offer flexibility to network operators in choosing delays in communication links. We consider two kinds of delays - namely (1) inter-layer delay that arises in the local-area network (LAN) connecting the sensors in the physical layer to the computational units in the cyber layer, and (2) intra-layer delay that arises in the SDN connecting the computational units spread across the cyber-layer. Our goal is to co-design these two delays with a sparse feedback controller so that the $H_2$-norm of the closed-loop system is minimized, while ensuring that both delays are greater than or equal to their individual lower bounds that arise from the cost of the network bandwidth. The main contribution of this paper is to develop a hierarchical optimization algorithm that provides a guided solution for this co-design. The outer loop designs the two delays and finds a corresponding stabilizing controller by sequentially relaxing the non-linear matrix equations required for the co-design. The inner loop sparsifies this controller while minimizing the closed-loop $H_2$-norm. Our results show that depending on the plant dynamics, the relative magnitudes of the two delays for achieving the optimal $H_2$-norm can be notably different.

Note that our problem is fundamentally different from the conventional bandwidth allocation and delay assignment problems commonly addressed in the networking literature [7], where the utility functions to be optimized are static objectives. Our goal, in contrast, is to design a bandwidth allocation mechanism that minimizes the $H_2$-norm of a CPS over a sparse state-feedback controller. We illustrate the effectiveness of our algorithm using simulations that highlight the impacts of delays and sparsity on $H_2$-performance. The proofs of all lemmas, theorems and propositions are listed in [8, Sec. 6], unless stated otherwise.

II. PROBLEM FORMULATION

A. State Feedback with Communication Delays

Consider a LTI system with the following dynamics:

$$\dot{x}(t) = Ax(t) + Bu(t) + B_w w(t),$$

(1)

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control, and $w \in \mathbb{R}^r$ is the exogenous input, with the corresponding matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $B_w \in \mathbb{R}^{n \times r}$. We design a state-feedback controller, ideally represented as $u(t) = -K x(t)$. However, due to limited bandwidth availability, the controller includes finite delays in the feedback. The CPS model that we consider is described as follows.

- There are $p$ sensors and actuators in the physical layer and $x(t)$ is correspondingly divided into $p$ non-overlapping parts $\bar{x}_1(t), \ldots, \bar{x}_p(t)$, where $\bar{x}_i$ is measured by the $i$-th sensor.
- There are $p$ computing units or control nodes located in a virtual cloud network. The $i$-th sub-state $\bar{x}_i(t)$ is transmitted to the $i$-th control node through LAN with delay $\tau_{c}/2$.
- Inside the cloud, also referred to as the cyber layer, the control nodes share their individual sub-states $\bar{x}_i(t)$ with each other over an SDN with delay $\tau_c$. Each control node
calculates a portion of the control input vector denoted as
\( \bar{u}_i(t) \in \mathbb{R}^{m_i} \), where \( \sum_{i=1}^{p} m_i = m \).

- The calculated control inputs are transmitted back to the
  physical layer with \( \tau_d/2 \) delay.

A schematic of this CPS with \( n = 3, m = 2 \) and \( p = 2 \) is
shown in Fig. 1. Denoting \( \tau_o = \tau_d + \tau_c \), the control input
may be expressed as:

\[
u(t) = -\left( \frac{K}{K_d} x(t - \tau_d) - \left( \frac{K}{K_o} x(t - \tau_o) \right) \right),
\]

where \( K \) represents Hadamard product and \( K_d, K_o \in \mathbb{R}^{m \times n} \)
are binary matrices such that

\[
I_d(i, j) = \begin{cases} 1, & \text{if } i = q, x_j \in \bar{x}_q, \\ 0, & \text{otherwise} \end{cases},
\]

and \( I_o \) is the complement of \( I_d \). For the system shown in
Fig. 1, \( I_d \) and \( I_o \) are:

\[
I_d = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad
I_o = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.
\]

The closed-loop system of (1)-(2) can be written as:

\[
\dot{x}(t) = Ax(t) - BK_dx(t - \tau_d) - BK_ox(t - \tau_o) + B_u w(t),
\]

\[
z(t) = Cx(t) + Du(t) = \begin{bmatrix} Q^{1/2} \\ 0 \end{bmatrix}, \quad
D = \begin{bmatrix} 0 \\ R^{1/2} \end{bmatrix},
\]

where \( z(t) \) is the measurable output, \( Q \geq 0 \) and \( R > 0 \). We
make the standard assumption that \( (A, B) \) and \( (A, Q^{1/2}) \)
are stabilizable and detectable, respectively [1, Sec. II].

---

**B. Problem Setup**

Our goal is to design a \( K \) that minimizes the \( H_2 \)-norm of the
transfer function from \( w(t) \) to \( z(t) \) for the time-delayed
system (5). In general, the \( H_2 \)-performance of (5) will be
worse than that of the delay-free system [9, Sec. 5.6.1].

Therefore, the trivial solution would be \( \tau_d = \tau_o = 0 \), which
will require infinite bandwidth.

Let the combined bandwidth of links connecting the physical
to the cloud be \( W_{cp} \) and that of SDN links inside the cloud be
\( W_{cc} \). Then, the total cost for renting bandwidth can be written as:

\[
S = m_{cp} W_{cp} + m_{cc} W_{cc},
\]

where \( m_{cp} \) and \( m_{cc} \) are the respective dollar costs for renting
LAN and SDN links. \( W_{cp} \) and \( W_{cc} \) are divided into the total
number of links as described below.

- The uplink for carrying \( \bar{u}_i(t) \) back to the actuator is not
  needed if the \( i \)-th block row of \( K \) is entirely 0. Similarly, if
  the \( i \)-th block column of \( K \) is 0, then \( \bar{x}_i \) is no longer required
  for calculating any control input, and the corresponding
downlink becomes redundant. The uplinks and downlinks together
constitute the LAN links. Thus, \( W_{cp} \) is effectively divided into
the number of non-zero block rows and columns of \( K \) denoted by
\( N_{row}(K) \) and \( N_{col}(K) \), respectively.

- \( W_{cc} \) is divided into the number of non-zero off-diagonal
  blocks of \( K \) denoted by \( N_{off}(K) \).

Accordingly, we can write the bandwidth constraint as:

\[
S = 2m_{cp} \left( \frac{N_{row}(K) + N_{col}(K)}{\tau_d} \right) + m_{cc} \left( \frac{N_{off}(K)}{\tau_o - \tau_d} \right) \leq S_b,
\]

where \( S_b > 0 \) is a mandatory budget that is imposed to
prevent infinite bandwidth. To minimize the cost of renting
the links and bandwidth, we wish to reduce the number of
both LAN and SDN links by promoting sparsity in \( K \). Our
design objectives, therefore, are listed as:

\[ \textbf{P1:} \quad \text{Design } \tau_d, \tau_o \text{ and } K \text{ such that} \]

- \( H_2 \)-norm of the closed-loop transfer function of (5)
  from \( w(t) \) to \( z(t) \), denoted as \( J \), is minimized.

- The bandwidth cost \( S \) satisfies (7) for some given budget
  \( S_b \), which is large enough for the problem to be feasible.

- Sparsity of \( K \) is promoted.

Given \( S_b \), \( \textbf{P1} \) can be mathematically stated as:

\[
\begin{align*}
\text{minimize}_{K, \tau_d, \tau_o} & \quad J(K, \tau_d, \tau_o) + g(K), \\
\text{subject to} & \quad K \text{ stabilizes (5) for } \tau_o, \tau_d \text{ delays}, \\
& \quad S(\tau_d, \tau_o, K) \leq S_b,
\end{align*}
\]

where \( S \) is given by (7), and \( g(K) \) is a sparsity-promoting
function which will be introduced in Sec. IV-A. The closed-
form expression of \( J \) is derived next.

---

**C. \( H_2 \) norm for the Delayed System**

The delayed system (5) is infinite dimensional. In order to
obtain a linear, finite dimensional LTI approximation of (5),
we use the method of spectral discretization given in [10].

Since \( \tau_o > \tau_d \) in (5), following [10], we divide \([-\tau_o, 0]\) into
a grid of \( N \) scaled and shifted Chebyshev extremal points

\[
\theta_{k+1} = \tau_o \frac{\cos \left( \frac{(N-k-1)\pi}{N-1} \right) - 1}{\tau_o}, \quad k = 0, \ldots, N-1,
\]

such that \( \theta_1 = -\tau_o \) and \( \theta_N = 0 \). The choice of \( N \) is guided
by [10, Sec. 4]. Let \( v(\theta) = x(t + \theta) \) denote the \( \theta \)-shifted
state vector. The extended state \( \eta \) and the closed-loop state
matrix \( A_{cl} \) can then be written as:

\[
\eta = [v^T(\theta_1), \ldots, v^T(\theta_N)]^T, \quad
l_j(\theta) = \sum_{m=1}^{N} \sum_{m \neq j} \theta_m a_{mj} v_n, \\
A_{cl} = \begin{cases} \dot{v}_n(l_j(\theta)) & j = 1, \ldots, N, \quad i = 1, \ldots, N-1 \\
-l_N(\tau_o)BK_dA, & j = N, \quad i = N \\
-l_i(\tau_o)BK_d - BK_o, & j = 1, \quad i = N \\
-l_j(\tau_o)BK_d & j = 2, \ldots, N-1, \quad i = N, \end{cases}
\]
where \( K_d = K \circ T_d, K_o = K \circ T_o \). We can write \( A_{cl} \) as:
\[
A_{cl} = \tilde{A} - BK_oN_o^T - BK_dN_d^T,
\]
(11)
\[\mathcal{B} = MB, M = [0, \ldots, 0, I_n]^T, N_o = [I_n, 0, \ldots, 0]^T,\]
(12)
where \( \tilde{A} \) is independent of \( K_o \) and \( K_d \), and the explicit expressions for \( \tilde{A} \) and \( N_d \) in terms of \( \tau_o \) and \( \tau_d \) will be derived shortly in the next section. The linear approximation of the closed-loop system (5) becomes:
\[
\dot{y}(t) = A_{cl}y(t) + B_wu(t),
\]
(13a)
\[
z(t) = Cy(t), \quad C = \begin{bmatrix} Q^{1/2}M^T & -R^{1/2}(K_dN_d^T + K_oN_o^T) \end{bmatrix},
\]
(13b)
where \( B_w = MB_w \). The algebraic Riccati equations (AREs) and the closed-loop \( \mathcal{H}_2 \)-norm \( J \) can be written as:
\[
A_{cl}^*P + PA_{cl} = -C^TC = -(\hat{Q} + \hat{C}^TR\hat{C}),
\]
(14)
\[
A_{cl}L + LA_{cl}^* = -BB^T,
\]
(15)
\[
J(K, \tau_d, \tau_o) = \text{Tr}(B^TPB) = \text{Tr}(CLC^T).
\]
(16)
where \( \hat{Q} = MQM^T \) and \( \hat{C} = (K_dN_d^T + K_oN_o^T) \).

### III. Derivation of the Gradient of \( \mathcal{H}_2 \) Norm

Our goal is to design \( (K, \tau_d, \tau_o) \) to minimize \( J \). However, from (14)-(16), we see that \( J \) is a function of \( \tilde{A} \) and \( N_o \), besides \( K \). To compute the gradient of \( J \) w.r.t. \( (K, \tau_d, \tau_o) \), it is important to express \( \tilde{A} \) and \( N_o \) in terms of these three design variables. We begin this section with these derivations.

#### A. \( \mathcal{H}_2 \) Performance and Design Variables

Recall that the closed-loop state matrix \( A_{cl} = \tilde{A} - B(K_oN_o^T + K_dN_d^T) \). In the next two lemmas, we express \( A_{cl} \) as a function of \( \tau_o, K \) and the delay ratio \( c = \tau_d/\tau_o \).

**Lemma 1:** \( \tilde{A} = A_o/\tau_o + \tilde{A} \) is a function of \( \tau_o \), and can be written as:
\[
\tilde{A} = \frac{1}{\tau_o}A + \tilde{A}, \quad \tilde{A} = \text{diag}(0, A),
\]
(17)
where \( A \) is a constant matrix for constant \( N \).

**Lemma 2:** \( N_o \) is a function of the ratio \( c = \tau_d/\tau_o \in [0, 1] \), and can be written as:
\[
N_o(c) = (\Gamma(c)) \otimes I_n, \quad \Gamma(c) = [c^{N-1} c^{N-2} \ldots c^2 c^1]^T,
\]
(18)
where \( \Gamma \in \mathbb{R}^{N \times N} \) is a constant matrix for constant \( N \).

Lemmas 1 and 2 show that for fixed \( N \) and \( J \) for the system in (13) can be written as a function of \( \tau_o \) and \( c \). Henceforth, all of our analysis for minimizing \( J \) will be carried out using \( \tau_o \) and \( c \), instead of \( \tau_o \) and \( \tau_d \). This change of variables is invertible, and therefore, there is no loss of generality.

#### B. Gradient of \( \mathcal{H}_2 \) Norm

In order to minimize \( J \), we next derive the gradient of \( J \). We define a set \( K := \{(K, \tau_o, c) : \text{Re}(\lambda(A_{cl})) < 0\} \), i.e., the set of \( K \) that guarantee closed-loop stability of (13). Given this definition, we first prove the existence of a unique solution of (14) and differentiability of \( P \), followed by the derivation of \( \nabla J \). For the rest of the paper, the \( A'(B) \) notation represents differentiability of \( A \) depending on \( B \).

**Lemma 3:** Let \( (K, \tau_o, c) \in K \). Then, there exists a unique solution \( P(K, \tau_o, c) \) of (14). Moreover, \( P \) is differentiable with respect to the variables \( \tau_o \), \( c \) and \( K \) on \( K \). Specifically, \( P'(\tau_o) \) is a non-linear function of \( c \in [0, 1] \) through \( N_o(c) \) as shown in Lemma 2, and therefore, the exact expression of \( N_o(c) \) cannot be used while forming the SDP relaxations. To circumvent this problem, we divide \([0, 1]\) into \( k_c \)
sub-intervals \([c_1, c_2], [c_2, c_3], \ldots, [c_k, c_{k+1}]\) with each sub-interval small enough to allow \(N_d(c)\) to be approximated as an affine function \(\hat{N}_d(c)\). Let each sub-interval \([c_i, c_{i+1}]\) have an associated \(\chi^{(i)} \in \mathbb{R}^{N \times 2}\) as the vector of affine coefficients. The approximated function is written as:

\[
\hat{N}_d(c) = \left(\chi^{(i)}c + 1\right)^T \otimes I_n, \; c \in [c_i, c_{i+1}], \; i = 1, \ldots, k_c.
\] (25)

The coefficients \(\chi^{(i)}\) can be computed from a linear curve fitting on (18). Larger the number of sub-intervals \(k_c\), lower is the approximation error \(\|\hat{N}_d - N_d\|\). For our simulations in Sec. V, we have used \(k_c = 10\). We next present the SDP relaxation for the co-design of \((K, c)\).

**Theorem 3:** Consider a known tuple \((K^*, \tau^*, e^*) \in \mathcal{K}\) with \(e^* \in [c_i, c_{i+1}]\) for some \(i \in \{1, \ldots, k_c\}\) satisfying (15) with a known \(L^*\) for closed-loop state matrix \(A_{\mathcal{K}}^+(K^*, \tau^*, e^*)\). Let \(c = e^* + \Delta c, \; K = K^* + \Delta K, \; L = L^* + \Delta L\) and \(\alpha \in \mathbb{R}\) be a solution of the following SDP:

\[
\begin{aligned}
\phi_0 + \phi_1 + B B^T + \alpha I & \\ c_1 \leq c & \leq c_{i+1}, \quad \|\Delta L\| \leq \beta, \\
\alpha & \geq 2\beta B\Delta K + \beta \Delta K_d N_d] \|c^*\| + 2\beta \Delta \| B \Delta K \| + \\
d\alpha \geq 2\beta \Delta K_d N_d T^*(B \Delta K)^T + 2\beta \| B \Delta K_d \| L^*, (26c)
\end{aligned}
\]

where \(\alpha, \Delta K, \Delta L, \) and \(\Delta c\) are the design variables. Then, \((K^*, \tau^*, c)\) is a stabilizing tuple for (13). In (26), \(\Delta N_d = \hat{N}_d(c) - N_d(c^*)\), \(\phi_0 = A_{\mathcal{K}} L^* + L A_{\mathcal{K}}^T\), \(\phi_1 = A_1 L^* + L A_1^T\), \(A_1 = -B(K^*) N_d T^*(c^*) + \Delta K_d N_d T^*(c^*) + \Delta K_d \), and \(\beta\) is chosen constant, and \(\mathcal{S} \geq [\|N_d(c)\|, \|\cdot\|].\)

Starting from a known stabilizing tuple \((K^*, \tau^*, e^*)\), Theorems 2 and 3 enable us to co-design new stabilizing pairs \((K, \tau_0)\) and \((K, c)\), respectively. Next, we integrate the bandwidth cost constraint (7) with the SDPs in (24) and (26).

**D. Incorporating Bandwidth Constraints**

We impose the bandwidth cost constraint (7) as part of \(\mathbf{P_1}\), which can be rewritten as:

\[
S = 2m_c p + \left(\frac{N_{row}(K) + N_{coll}(K)}{c_{\tau_0}}\right) + m_c e \left(\frac{N_{off}(K)}{\tau - c_{\tau_0}}\right) \leq S_h. \quad (27)
\]

Recall that \(S\) is the total bandwidth cost and \(S_h\) is the known upper bound imposed on it. When (27) is imposed on SDPs (24) and (26), we obtain an alternative form of (27), which is stated in the next proposition.

**Proposition 1:** Consider a known tuple \((K^*, \tau_0, e^*) \in \mathcal{K}\) with an associated bandwidth cost \(S^* \leq S_h\). Denoting \(n_{\mathcal{K}} = N_{row}(K^*) + N_{coll}(K^*)\) and \(n_{\mathcal{E}} = N_{off}(K^*)\), the following statements are true.

1) Keeping \(\tau_0 = \tau_0^*\), let \(e^*\) be perturbed to \(c\) resulting in a cost \(S\). Then, \(\delta S(c) = S - S^*\) is a convex function of \(c\):

\[
\delta S(c) = (S^* + \tau_0^*) e^* + \left(m_c n_{\mathcal{E}} - 2m_p n_{\mathcal{K}} + S^* + \tau_0^* \right) c + 2m_p n_{\mathcal{K}}.
\]

2) Keeping \(c = e^*\), let \(\tau_0^*\) be perturbed to \(\tau_0\) resulting in a new bandwidth cost \(S\). Then, \(\delta S(\tau_0) = S - S^*\) is an affine function of \(\tau_0\):

\[
\delta S(\tau_0) = \frac{1}{S^*} \left(\frac{2m_p n_{\mathcal{K}}}{e^*} + \frac{m_c n_{\mathcal{E}}}{(1 - e^*)}\right) - \tau_0.
\]

Both the constraints \(\delta S(c) \leq 0\) and \(\delta S(\tau_0) \leq 0\) individually imply \(S \leq S_h\).

Since \(\delta S(\tau_0)\) and \(\delta S(c)\) are each convex w.r.t their respective arguments, we can easily incorporate them in the co-design SDPs of Theorems 2 and 3 to satisfy the bandwidth constraint in (27). Note that since \(K\) is co-designed with either \(\tau_0\) or \(c\), the true bandwidth cost \(S\) depends on \(K\) as well through \((N_{row}(K) + N_{coll}(K))\) and \(N_{off}(K)\). If \(N_{row}(K) \leq N_{row}(K^*)\), \(N_{coll}(K) \leq N_{coll}(K^*)\) and \(N_{off}(K) \leq N_{off}(K^*)\), one can easily verify that \(\delta S(c) = 0\) and \(\delta S(\tau_0) \leq 0\) in (28)-(29) hold, and the true bandwidth costs always satisfy (27). We ensure this fact by imposing a two-loop structure in our design algorithm.

**IV. PROBLEM SETUP IN TWO-LOOP ADMM FORM**

The \(l_2\)-norm \(J\), in general, increases with increasing sparsity of \(K\), while the bandwidth cost \(S\) reduces. Due to these trade-offs between the objectives and the constraints, \(\mathbf{P_1}\) is a prime candidate to be reformulated as a two-loop ADMM optimization. The outer-loop co-designs \((K, \tau_0)\) and \((K, c)\) using (24)-(26) under the bandwidth constraints (28)-(29). The inner-loop sparsifies \(K\) while minimizing \(J\). We describe the inner and outer loops in Sec. IV-A and IV-B respectively, followed by the main algorithm in Sec. IV-C.

**A. Inner ADMM Loop**

Throughout the inner ADMM loop, we hold both \(\tau_0\) and \(c\) as constants. The mathematical program of the inner loop denoted as \(\mathbf{P_{1_{in}}}\) is written as follows:

\[
\begin{align*}
\mathbf{P_{1_{in}}}: \; & \text{minimize } J(K) + \lambda g(F), \; \text{subject to } K = F, \quad (30) \\
& \text{subject to } K, F \end{align*}
\]

where \(\lambda\) is a regularization parameter and \(g(F) = \|W \circ F\|_{l_1}\) is the weighted \(l_1\) norm function which is used to induce sparsity in \(F\). The weight matrix \(W\) for \(g(F)\) is updated iteratively through a series of reweighting steps from the solution of the previous iteration as [11]:

\[
W_{ij} = 1/|F_{ij}| + \epsilon, \quad 0 < \epsilon \ll 1. \quad (31)
\]

The augmented Lagrangian for \(\mathbf{P_{1_{in}}}\) is

\[
\mathcal{L}_{p} = J(K) + \lambda g(F) + \text{Tr}(\Theta^T(K - F)) + \frac{\rho}{2}\|K - F\|_F^2, \quad (32)
\]

where \(\rho > 0\) and \(\Theta\) is the dual variable. ADMM involves solving each objective separately while simultaneously projecting onto the other’s solution set. As shown in [1], [12], (32) is used to derive a sequence of iterative steps \(K\)-min, \(F\)-min and \(\Theta\)-min by completing the squares w.r.t each variable:

\[
\begin{align*}
K_{k+1} &= \argmin_{K} \frac{J(K)}{K} + \frac{\rho}{2}\|K - U_k\|^2_F, \quad (33a) \\
F_{k+1} &= \argmin_{F} \frac{J(F)}{F} + \frac{\rho}{2}\|F - V_{k}\|^2_F, \quad (33b) \\
\Theta_{k+1} &= \Theta_k + \rho(K_{k+1} - F_{k+1}), \quad (33c)
\end{align*}
\]

where \(U_k = F_k - \frac{1}{\rho} \Theta_k\) and \(V_k = K_{k+1} + \frac{1}{\rho} \Theta_k\). We next present methods to solve \(K\)-min and \(F\)-min.

900
1) **K-min Step:** Setting $\nabla \Phi_1(K) = 0$ and using Theorem 1, we get the following condition for optimality:

$$[(GLN_d) \circ \mathcal{I}_d + (GLN_o) \circ \mathcal{I}_o] + \frac{\rho}{2}(K - U) = 0, \quad (34)$$

where $G = R(K_d N_d^T + K_o N_o^T) - B^T P$ and $U = U_k$ for the $(k+1)$-th iteration of the ADMM loop. $P$ and $L$ are the solutions of AREs (14) and (15), respectively. K-min begins with a stabilizing $K$, solves (14) and (15) for $P$ and $L$, and then solves (34) to obtain a new gain $\hat{K}$ as follows:

$$K = \text{Reshape} \left( \left( \tilde{V}_d \circ T_d + \tilde{V}_o \circ T_o + \rho I \right)^{-1} \mu, \left[ n, n \right] \right), \quad (35)$$

where $T_d = (T_d \circ \tilde{V}_d^T + T_d \circ \tilde{V}_d^T)$, $T_o = (T_o \circ \tilde{V}_o^T + T_o \circ \tilde{V}_d^T)$, $T_{do} = 2(N_d^T L N_o \circ R)$, $T_{oo} = 2(N_o^T L N_o \circ R)$, and $\mu = \vec{c}(2B^T PL N_d \circ \mathcal{I}_d + 2B^T PL N_o \circ \mathcal{I}_o + \rho U)$.

The notation $B = \text{Reshape}(A, [p, q])$ is used for an operator that reshapes $A \in \mathbb{R}^{m \times n}$ in row-traversing order to another matrix $B \in \mathbb{R}^{p \times q}$, provided $pq = mn$. We use $\vec{c}$ to represent the vectorization operator and $1^T \in \mathbb{R}^{n^2}$ to represent a vector of all ones. For details of the above derivation, see [8, Sec. 6.8]. It can be shown that $K = K - \hat{K}$ is the descent direction for $\Phi_1$ [13, See Lemma 4.1]. The Armijo-Goldstein line search method can then be used to determine a step size $s$ to ensure that $(K + s\hat{K})$ stabilizes (13) for the $\tau_o$ and $c$ that are held constant for the inner-ADMM loop. The iterative process continues till $\nabla \Phi_1(K) \approx 0$.

2) **F-min Step:** The solution of the F-min step is well-known in the literature [12, Sec. 4.4.3] as:

$$F_{ij} = \begin{cases} (1 - \frac{a_{ij}}{|V_{ij}|})V_{ij}, & \text{if } |V_{ij}| > a_{ij}, \\ 0, & \text{otherwise}, \end{cases} \quad (36)$$

where $a_{ij} = \frac{1}{\rho} |V_{ij}|$. Note that large values of $\lambda$ will induce more sparsity, and therefore may lead to a sudden increase in $J$. Therefore, $\lambda$ must be increased in small steps. The regularization path, for example, can be logarithmically spaced from $0.01\lambda_{max}$ to $0.95\lambda_{max}$, where $\lambda_{max}$ is ideally the critical value of $\lambda$ above which the solution of $\text{P}_1$ is $K = F = 0$ [12]. In our simulations, $\lambda_{max} = 10^5$.

### B. Outer Loop

The outer-loop of our algorithm designs $\tau_o$ and $c$ with bandwidth constraint (27) and updates the weight matrix $W$ for minimizing the weighted $l_1$ norm in (31). Co-design of $K$ in this loop is necessary to ensure stability, as $\tau_o$ and $c$ change. Let $K^* = F^*$ and $\Theta^*$ be the output of the last converged inner loop with $U^* = K^* - \frac{1}{\rho} \Theta^*$. Programs $\text{P}_{1o1}$ and $\text{P}_{1o2}$ directly design $(K, \tau_o)$ and $(K, c)$, respectively, in sequence as follows:

$$\text{P}_{1o2} : \ \text{minimize} \ \hat{J}(K, c) + \frac{\rho}{2} \| K - U^* \|^2_F, \quad (37c)$$

subject to $\delta S(\tau_o) \leq 0$, SDP in Eq. (26), \quad (37d)

where $\hat{J}(K, \tau_o) = \text{Tr}(B^T P B)$, $\hat{J}(K, c) = \text{Tr}(LC^T C^*)$, $C^* = (K^* \circ \mathcal{I}_d) N_d^T (c^*) + (K^* \circ \mathcal{I}_o) N_o^T (c^*)$, $N_{off}(K^*)$, $\tau_o^*$ and $c^*$ from (27).

### C. Main Algorithm

- Using $\text{P}_{1o1}$, we first co-design a stabilizing pair $(\hat{K}, \tau_o)$ from an initial tuple $(K^*, \tau_o^*, c^*) \in \mathcal{K}$. The two are designed together as the initial $K^*$ may not be stabilizing for $\tau_o$ satisfying the bandwidth constraint (29).
- We then use the solution of $\text{P}_{1o1}$, i.e., $(\hat{K}, \tau_o, c^*) \in \mathcal{K}$ as the initial point for $\text{P}_{1o2}$ to find a new pair $(K, c)$. Using Proposition 1, let $c_{\text{min}}$ be the minimizer of $\delta S(c)$. If $\hat{K}$ is stabilizing for $c_{\text{min}}$, then instead of co-designing $(K, c)$, we can set $c = c_{\text{min}}$ and $K = \hat{K}$, and then use a procedure similar to K-min to minimize $J(K)$, starting from $\hat{K}$.
- The inner-loop begins with $(K, \tau_o, c) \in \mathcal{K}$. $K$ is updated in the direction of decreasing $J$ and increasing sparsity while $\tau_o$ and $c$ remain constant.
- Following [1, Sec. III-D] and [12, Sec. 3.4.1], $\rho$ in (33) is chosen to be sufficiently large to ensure the convergence of the inner ADMM loop. Since $J$ is nonconvex, convergence of this loop, in general, is not guaranteed, as is commonly seen in the sparsity promoting literature [1]. However, large values of $\rho$ have been shown to facilitate convergence. We use $\rho = 100$ for our simulations. The stopping criterion for the inner loop is Line 8 of Algo. 1 follows [12, Sec. 3.3.1].

### V. Simulations Results

We next validate Algorithm 1 by comparing it to an algorithm that consists of only the inner ADMM loop, referred to as the constant-delay algorithm. Both algorithms start from $(K^*, \tau_o^*, c^*) \in \mathcal{K}$. The delays $\tau_d^*$ and $\tau_d^*$ are kept constant throughout the constant-delay algorithm and the number of
Fig. 2. Case B-I (a), (b), (c), (d) show \( J, S, \tau_o \) and \( c \) vs \( N_z \) where \( N_z \) is the number of zero elements of \( K \). ‘DD’ and ‘W/O DD’ indicate Algorithm I and constant-delay algorithms respectively.

Fig. 3. Case B-II (a), (b), (c), (d) show \( J, S, \tau_o \) and \( c \) vs \( N_z(K) \) where \( N_z \) is the number of zero elements of \( K \). ‘DD’ and ‘W/O DD’ indicate Algorithm I and constant-delay algorithms respectively.

The priority later shifts to decreasing \( c \) as sparsity increases. The shifting priority of one delay over another highlights the implicit relationship between \( K, \tau_o \) and \( \tau_d \).

**Case B-II:** We consider another randomly generated \( A \in \mathbb{R}^{10 \times 10} \) with \( (c^*, \tau_o^*) = (0.24, 0.089) \), \( m_{cp} = 21 \) and \( m_{cc} = 31 \). The initial conditions result in \( c_{min} = 0.448 > c^* \) from (28). However, \((K^*, c_{min})\) is an unstable tuple, and therefore, we rely on \( \text{P1}_o \) to co-design \((K, c)\). Fig. 3 (c), (d) show that as sparsity increases, Algorithm I continuously increases \( c \) and decreases \( \tau_o \) to maintain optimality of \( J \). As shown in Proposition 1, a decrease in \( \tau_o \) increases the bandwidth cost \( S \). However, since \( c \) moves towards \( c_{min} \), \( S \) in Fig. 3 (b) remains comparable to that of the constant-delay algorithm despite the continuous decrease in \( \tau_o \). Fig. 3 (a), (b) show that as a trade-off for slightly higher \( S \) from Algorithm I, we obtain a lower \( J \) as compared to the constant-delay algorithm for all the sparsity levels.

**VI. CONCLUSION**

This paper presented a co-design for network delays and sparse controllers to improve the \( \mathcal{H}_2 \)-performance of time-delayed systems. The challenges of co-design arising from the implicit functional relationships between the delays, the sparse controller, and the \( \mathcal{H}_2 \)-norm are overcome by developing a hierarchical algorithm. Numerical simulations show the effectiveness of the design, while bringing out interesting observations about these implicit relationships.

**REFERENCES**

[1] F. Lin, M. Fardad, and M. R. Jovanović, “Design of optimal sparse feedback gains via the alternating direction method of multipliers,” *IEEE Trans. Autom. Control*, vol. 58, no. 9, 2013.

[2] M. Wytock and J. Z. Kolter, “A fast algorithm for sparse controller design,” *arXiv preprint arXiv:1312.4892*, 2013.

[3] F. Lin, A. Chakrabortty, and A. Duel-Hallen, “Game-theoretic multi-agent control and network cost allocation under communication constraints,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, 2017.

[4] F. Lin and V. Adetola, “Sparse output feedback synthesis via proximal alternating linearization method,” *arXiv preprint arXiv:1706.08191*, 2017.

[5] N. Negi and A. Chakrabortty, “Sparse optimal control of LTI systems under sparsity-dependent delays,” in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 2669–2674.

[6] P. Naghshtabrizi, J. P. Hespanha, and A. R. Teel, “Stability of delay impulsive systems with application to networked control systems,” *II Meas Control*, vol. 32, no. 5, pp. 511–528, 2010.

[7] F. P. Kelly, A. K. Maulloo, and D. K. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[8] N. Negi and A. Chakraborty, “Co-design of delays and sparse controllers for bandwidth-constrained cyber-physical systems,” *arXiv preprint arXiv:2003.05888*, 2020.

[9] K. Gu, J. Chen, and V. L. Kharitonov, *Stability of time-delay systems*. Springer Science & Business Media, 2003.

[10] J. Vanbiervliet, W. Michiels, and E. Jarlebring, “Using spectral discretisation for the optimal \( h_2 \) design of time-delay systems,” *International Journal of Control*, vol. 84, no. 2, pp. 226–241, 2011.

[11] E. J. Candès, M. B. Wakin, and S. P. Boyd, “Enhancing sparsity by reweighted \( \ell_1 \) minimization,” *Journal of Fourier analysis and applications*, vol. 14, no. 5, pp. 877–905, 2008.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach.*, vol. 3, no. 1, 2011.

[13] T. Rautert and E. W. Sachs, “Computational design of optimal output feedback controllers,” *SIAM J Optim.*, vol. 7, no. 3, 1997.