2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Scheduling and control of networked systems: a sparsity approach

Anubhab Dasgupta and Atreyee Kundu

Abstract— **This paper deals with the design of scheduling logic and control logic for networked control systems (NCSs) with limited communication resources. We consider an NCS with N plants that communicate with remotely located controllers over a shared band-limited communication network. Due to a limited capacity of the network, at most M $(< N)$ plants can exchange information with their controllers at any instant of time and the remaining at least $N - M$ plants operate in open-loop. We present an algorithm that co-designs an allocation scheme of the communication network among the plants (scheduling logic) and the control inputs for the plants (control logic). Given a non-zero initial state for each plant, a scheduling and control logic obtained from our algorithm ensures that the states are steered to zero in the given time horizon for all the plants. We also provide sufficient conditions on the plant dynamics, capacity of the communication network and the given time horizon that lead to a numerically tractable implementation of the proposed algorithm. We employ a feasibility problem with sparsity constraints as the key tool in our design. A numerical example is presented to demonstrate the proposed results.**

I. Introduction

Networked Control Systems (NCSs) are spatially distributed systems in which the communication between plants and their controllers occurs through a shared digital communication network [4, Section 1]. They are an integral part of modern day Cyber-Physical Systems (CPS) and Internet-of-Things (IoT) applications.

NCSs with a large number of plants often suffer from limited communication resources. Indeed, the bandwidth of shared communication networks is typically limited. The scenario in which the number of plants sharing a communication network is higher than the capacity of the network is called *medium access constraint*. This scenario motivates the need to allocate the communication network to each plant in a manner so that good qualitative and quantitative properties of the plants are preserved. The task of efficient allocation of a shared communication network is commonly referred to as a *scheduling problem* and the corresponding allocation scheme is called a *scheduling logic*. In this paper we study algorithmic design of scheduling logic and its corresponding control logic for NCSs.

The existing classes of scheduling logic can be classified broadly into two categories: static (also called open-loop) and dynamic (also called closed-loop). In case of the former, a finite length allocation scheme of the network is determined offline prior to its application, while in case of the latter, the allocation of the shared network is determined based on some information about the plant (e.g., states, outputs, access status of sensors and actuators, etc.). While static logic are easier to implement and guarantee activation of each sensor and actuator, dynamic scheduling logic are often more suitable to ensure optimal performance of the plants under communication uncertainties.In this paper we will restrict our attention to static scheduling logic.

Various tools and techniques for the design of scheduling logic for NCSs have been used in the literature. For NCSs with continuous-time linear plants, a static scheduling logic that preserves stability of all plants are characterized using common Lyapunov functions in [5] and piecewise Lyapunov-like functions with average dwell time switching in [12]. A more general case of co-designing a static scheduling logic and control action is addressed using combinatorial optimization with periodic control theory in [17] and Linear Matrix Inequalities (LMIs) optimization with average dwell time technique in [2]. The discrete-time setting has also attracted considerable research attention. The authors of [20] characterize static switching logic that ensures reachability and observability of the plants under limited communication, and design an observer-based feedback controller for these logic. The corresponding techniques were later extended to the case of constant transmission delays [6] and Linear Quadratic Gaussian (LQG) control problem [7]. We employ multiple Lyapunov-like functions and graph theory to design stability preserving periodic scheduling logic in [10], Lie algebra for the design of stability preserving periodic scheduling logic under jamming attacks in the network in [9] and matrix inequalities for stability preserving periodic scheduling logic under data losses in the network in [11]. Recently in [19] we have presented a probabilistic algorithm for the design of stability preserving static scheduling logic. Event-triggered scheduling logics that preserve stability of all plants under communication delays are proposed in [1]. In [13] the authors propose a mechanism to allocate network resources by finding optimal node that minimizes a certain cost function in every network time instant. The design of dynamic scheduling logic for stability of each plant under both communication uncertainties and computational limitations is studied in [18]. In [3] a class of distributed control-aware random network access logics for the sensors such that all control loops are stabilizable, is presented. A dynamic scheduling logic based on predictions of both control performance and channel quality at run-time, is proposed in [14]. In this paper we present a sparsity driven algorithm for the design of scheduling and control logic for NCSs.

Anubhab Dasgupta is with the Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, West Bengal 721302, India. Email: anubhab.dasgupta@kgpian.iitkgp.ac.in. Atreyee Kundu is with the Department of Electrical Engineering, Indian Institute of Technology Kharagpur, West Bengal 721302, India. Email: atreyee@ee.iitkgp.ac.in.

We consider an NCS consisting of multiple discrete-time linear plants whose feedback loops are closed through a shared communication network. Due to a limited communication capacity of the network, only a few plants can exchange information with their controllers at any instant of time. Consequently, the remaining plants operate in open-loop. Given a non-zero initial state for each plant and a time horizon, we design a scheduling logic and a control logic that together steer the given initial state of each plant to zero in the given time horizon. We operate under the assumption that each plant requires access to the shared communication network at least once for its state to reach zero in the given time.

The primary tool for our design is a feasibility problem with sparse constraints (ℓ_0-constraints). Its solution is a control logic that for each plant steers the given non-zero initial state to zero in the given time horizon while obeying the constraint that not more than a certain number of plants receive a non-zero control input at any instant of time. Once such a control logic is obtained, we allow the plants with non-zero control inputs an access to the shared communication network. A scheduling logic is designed accordingly. The presence of an ℓ_0-constraint in our design of a control logic, however, requires us to solve a non-convex optimization problem. We present sufficient conditions on the plant dynamics, the capacity of the communication network and the given time horizon such that the non-convex optimization problem under consideration admits solutions. We also discuss algorithms to construct these solutions numerically. To the best of our knowledge, our algorithm introduces a new tool for the design of scheduling and control logic for NCSs to the literature.

The remainder of this paper is organized as follows: In Section II we formulate the problem under consideration. Our results appear in Section III. A numerical example is presented in Section IV. We conclude in Section V with a brief discussion of future research direction. Owing to space limitation, we omit proofs of our results. They will be presented in a longer journal version.

Notation. $\mathbb{R}$ is the set of real numbers and $\mathbb{N}$ is the set of natural numbers. For $v \in \mathbb{R}^n$, $\|v\|_0$ denotes its ℓ_0-norm, i.e., the number of non-zero elements in v. We use $\|v\|$ to denote the Euclidean norm of v. We let 0_n denote a n-dimensional zero vector. We extend this notation to also represent an $m(> n)$-dimensional vector with $m-n$-many non-zero entries and n-many zero entries as $\begin{pmatrix} a_1 \\ \vdots \\ a_{m-n} \\ 0_n \end{pmatrix}$ (resp., $\begin{pmatrix} 0_n \\ a_1 \\ \vdots \\ a_{m-n} \end{pmatrix}$).

II. Problem statement

We consider an NCS with N plants whose dynamics are given by

$$x_i(t+1) = A_i x_i(t) + b_i u_i(t), \;\; x_i(0) = x_i^0, \qquad (1)$$

$t = 0, 1, \dots, T-1$, where $x_i(t) \in \mathbb{R}^{d_i}$ and $u_i(t) \in \mathbb{R}$ are the vector of states and scalar control input of the i-th plant at time t, respectively, $A_i \in \mathbb{R}^{d_i \times d_i}$, $b_i \in \mathbb{R}^{d_i}$ are constants, $i = 1, 2, \dots, N$, and the time horizon $T \in \mathbb{N}$ is given.

The controllers are remotely located and each plant communicates with its controller through a shared communication network. We consider that the shared network has a limited communication capacity in the sense that at any time instant, at most M plants $(0 < M < N)$ can access the network. Consequently, at least $N - M$ plants operate in open loop (i.e., with $u_i(t) = 0$) at every time instant.

Assumption 1: The communication network is otherwise ideal in the sense that exchange of information between plants and their controllers is not affected by communication uncertainties.

Let $\mathcal{S}$ be the set of all subsets of $\{1, 2, \dots, N\}$ containing at most M distinct elements. We call a function $\gamma_T : \{0, 1, \dots, T-1\} \to \mathcal{S}$, that specifies at every time instant $t = 0, 1, \dots, T-1$, at most M plants of the NCS which have access to the shared network at that time, as a *scheduling logic*. We define the function $\upsilon_T : \{0, 1, \dots, T-1\} \to \mathbb{R}^N$ as $\upsilon_T(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \end{pmatrix}$. It specifies, at every time instant $t = 0, 1, \dots, T-1$, the control inputs $u_i(t) \in \mathbb{R}$ for the plants $i = 1, 2, \dots, N$, and is called a *control logic*.

Remark 1: In view of our definition of an open-loop operation of plants, for a fixed $\gamma_T(t)$, we need $u_i(t) = 0$ for all $i \in \{1, 2, \dots, N\}$ such that $i \notin \gamma_T(t)$. Further, in our setting, a closed-loop operation of a plant i with $u_i(t) = 0$ is equivalent to an open-loop operation of plant i. Thus, for a fixed $\upsilon_T(t)$, excluding all plants $i \in \{1, 2, \dots, N\}$ with $u_i(t) = 0$ from $\gamma_T(t)$ is no loss of generality.

In the sequel we will refer to a pair, (γ_T, υ_T), as a *scheduling and control logic* for the NCS under consideration. We will solve the following problem:

Problem 1: Given the plant dynamics, (A_i, b_i), $i = 1, 2, \dots, N$, the initial states, $x_i(0) = \xi_i \neq 0_{d_i}$, $i = 1, 2, \dots, N$, the capacity of the communication network, M, and the time horizon, T, design a scheduling and control logic, (γ_T, υ_T), under which $x_i(T) = 0_{d_i}$ for each plant $i = 1, 2, \dots, N$.

Notice that if there are at least $N - M$ plants $i \in \{1, 2, \dots, N\}$ such that $x_i(0) = \xi_i$ can be steered to $x_i(T) = 0_{d_i}$ in open-loop (i.e., there exists $\tau \in \{1, 2, \dots, T\}$ such that $A_i^\tau \xi_i = 0_{d_i}$), then the remaining at most M plants can be allowed to communicate with their remotely located controllers at every time $t = 0, 1, \dots, T-1$. In this scenario Problem 1 can be addressed solely by studying reachability of the remaining plants. We will study Problem 1 at a general level and assume that for each plant $i = 1, 2, \dots, N$, at least one non-zero control input is required to steer ξ_i to 0_{d_i} in T time units. Consequently, we require $T \geqslant \lceil \frac{N}{M} \rceil$. Indeed, with $T < \lceil \frac{N}{M} \rceil$, there is at least one plant that does not have access to the communication network at any time instant $t \in \{0, 1, \dots, T-1\}$.

We will solve Problem 1 in an *offline* manner in the sense that we will compute $\gamma_T(t)$ and $\upsilon_T(t)$, $t = 0, 1, \ldots, T-1$ at one go prior to their application to the NCS. We will employ an optimization problem that yields a control logic, υ_T, with a pre-specified sparsity to accommodate the communication capacity, M, of the network. We will design a scheduling logic, γ_T, by utilizing the sparse nature of the obtained υ_T.

III. MAIN RESULTS

A. Design of a scheduling and control logic, (γ_T, υ_T)

We first present an algorithm to design a scheduling and control logic, (γ_T, υ_T). We will then show that a (γ_T, υ_T) obtained from our algorithm steers $x_i(0) = \xi_i$ to $x_i(T) = 0_{d_i}$ for each plant $i = 1, 2, \ldots, N$.

Algorithm 1 Construction of a scheduling and control logic, (γ_T, υ_T)

Input: The plant dynamics, (A_i, b_i), $i = 1, 2, \ldots, N$, the capacity of the communication network, M, the initial states, $x_i(0) = \xi_i$, $i = 1, 2, \ldots, N$, and time horizon, T.

Output: A scheduling and control logic, (γ_T, υ_T).

1: Solve the following feasibility problem for υ_T:

$$\underset{\upsilon_T(0),\ldots,\upsilon_T(T-1)}{\text{minimize}} \quad 1 \tag{2}$$

$$\text{subject to} \begin{cases} x_i(t+1) = A_i x_i(t) + b_i u_i(t), \\ \qquad t = 0, 1, \ldots, T-1, \\ \qquad i = 1, 2, \ldots, N, \\ x_i(0) = \xi_i,\ x_i(T) = 0_{d_i}, \\ \qquad i = 1, 2, \ldots, N, \\ \|\upsilon_T(t)\|_0 \leqslant M,\ t = 0, 1, \ldots, T-1. \end{cases}$$

2: If there exists a solution υ_T to the feasibility problem (2), then go to Step 3. Otherwise, terminate the algorithm.

3: Construct $\gamma_T(t)$ as the set containing the elements of the set $\{1, 2, \ldots, N\}$ such that $u_i(t) \neq 0$, (i.e., the i-th component of $\upsilon_T(t)$ is non-zero), $t = 0, 1, \ldots, T-1$.

Algorithm 1 involves two steps: First, it solves the feasibility problem (2) to compute the control logic, υ_T. The sequences $\left(\left(u_i(t)\right)_{t=0}^{T-1}\right)_{i=1}^{N}$ have the following properties: (a) for each plant $i = 1, 2, \ldots, N$, $\left(u_i(t)\right)_{t=0}^{T-1}$ steers the given initial state ξ_i to 0_{d_i} in (at most) T units of time, and (b) at every time $t = 0, 1, \ldots, T-1$, $u_i(t) \neq 0$ at most for M-many plants $i \in \{1, 2, \ldots, N\}$. Second, it assigns the plants $i \in \{1, 2, \ldots, N\}$ with $u_i(t) \neq 0$ to $\gamma_T(t)$, i.e., they are allowed an access to the shared communication network at time t, $t = 0, 1, \ldots, T-1$. These plants receive their non-zero control inputs at time t and the plants $i \notin \gamma_T(t)$ operate in open-loop with $u_i(t) = 0$, $t = 0, 1, \ldots, T-1$. The feasibility problem (2) is a sparse optimization problem in the sense that it involves a sparse constraint (ℓ_0-constraint). Indeed, we employ an upper bound on the number of non-zero elements in $\upsilon_T(t)$ and thus a lower bound on the number of zero elements in $\upsilon_T(t)$. This accommodates the communication capacity, M, in our design of scheduling logic, γ_T and control logic, υ_T.

The following result asserts that for each plant $i = 1, 2, \ldots, N$, the state $x_i(0) = \xi_i$ is steered to $x_i(T) = 0_{d_i}$ under a scheduling and control logic, (γ_T, υ_T), obtained from our algorithm.

Proposition 1: Suppose that the plant dynamics, (A_i, b_i), $i = 1, 2, \ldots, N$, the capacity of the communication network, M, the initial states, $x_i(0) = \xi_i$, $i = 1, 2, \ldots, N$, and the time horizon, T, are given. Let (γ_T, υ_T) be a scheduling and control logic obtained from Algorithm 1. Then for each plant $i = 1, 2, \ldots, N$, $x_i(T) = 0_{d_i}$ under (γ_T, υ_T).

A scheduling and control logic, (γ_T, υ_T), obtained from Algorithm 1 is our solution to Problem 1.

Remark 2: A scheduling and control logic, (γ_T, υ_T), obtained from Algorithm 1 is *static* or *open-loop* in the sense that it is designed offline and does not adapt to faults in the plants and/or other components in the NCS that may occur during operation.

Remark 3: Recently in [8] the authors presented a sparsity based approach (joint ℓ_0 and L^0-optimization) for scheduling of a continuous-time linear networked system towards achieving a certain controllability metric. In this paper we use sparse optimization (ℓ_0-optimization) for the design of scheduling and control logic for NCSs with multiple plants and a limited communication capacity.

Notice that the key component in our design of (γ_T, υ_T) is obtaining a solution to the feasibility problem (2). However, solving (2) numerically is not a straightforward task. Indeed, since ℓ_0-norm is not a convex function, all constraints in (2) are not convex and the definition of a convex optimization problem is violated. In the remainder of this section we address solvability of the feasibility problem (2).

B. Solution to the feasibility problem (2)

Let $U_{\xi_i} = \left\{\left(u_i(t)\right)_{t=0}^{T-1} \,\middle|\, x_i(0) = \xi_i \text{ is steered to } x_i(T) = 0_{d_i} \text{ under } u_i(t),\ t = 0, 1, \ldots, T-1\right\}$. We observe that existence of a solution to (2) is equivalent to:

Proposition 2: Suppose that $U_{\xi_i} \neq \emptyset$ for all $i = 1, 2, \ldots, N$ and there exist elements $\left(\overline{u}_i(t)\right)_{t=0}^{T-1} \in U_{\xi_i}$, $i = 1, 2, \ldots, N$ such that $\left\|\begin{pmatrix}\overline{u}_1(t)\\ \overline{u}_2(t)\\ \vdots\\ \overline{u}_N(t)\end{pmatrix}\right\|_0 \leqslant M$ for all $t = 0, 1, \ldots, T-1$. Then there exists a solution to the feasibility problem (2).

The above Proposition is immediate. In particular, $\upsilon_T(t) = \begin{pmatrix}u_1(t)\\ u_2(t)\\ \vdots\\ u_N(t)\end{pmatrix} = \begin{pmatrix}\overline{u}_1(t)\\ \overline{u}_2(t)\\ \vdots\\ \overline{u}_N(t)\end{pmatrix}$, $t = 0, 1, \ldots, T-1$. Constructing of all elements of U_{ξ_i}, $i = 1, 2, \ldots, N$ and finding suitable combinations of $\left(\left(\overline{u}_i(t)\right)_{t=0}^{T-1}\right)_{i=1}^{N}$ can be performed by employing an exhaustive search technique. Notice that we require $x_i(T) = A_i^T \xi_i + \Phi_i \begin{pmatrix} \overline{u}_i(0) \\ \overline{u}_i(1) \\ \vdots \\ \overline{u}_i(T-1) \end{pmatrix} = 0_{d_i}$, where $\Phi_i = \begin{pmatrix} A_i^{T-1} b_i & A_i^{T-2} b_i & \dots & A_i b_i & b_i \end{pmatrix}$. We present a set of sufficient conditions on the plant dynamics, (A_i, b_i), $i = 1, 2, \dots, N$, the capacity of the communication network, M, and the time horizon, T, such that (2) admits a solution and an exhaustive search to compute the same is not needed. We also discuss algorithmic construction of the solution.

Let $\Psi_i = \begin{pmatrix} A_i^{d_i-1} b_i & A_i^{d_i-2} b_i & \cdots & A_i b_i & b_i \end{pmatrix}$, $i = 1, 2, \dots, N$.

Definition 1: We call a plant i *reachable* if the pair $(A_i, b_i) \in \mathbb{R}^{d_i \times d_i} \times \mathbb{R}^{d_i}$ satisfies $\text{rank}(\Psi_i) = d_i$.

Our first result for the existence of a solution to the feasibility problem (2) is the following:

Proposition 3: Suppose that

C1) Each plant $i = 1, 2, \dots, N$ is reachable.

C2) There exist $\hat{d}_j \in \mathbb{N}$, $\mathcal{P}_j \subseteq \{1, 2, \dots, N\}$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$ such that
  a) $|\mathcal{P}_j| \leqslant M$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$,
  b) $\mathcal{P}_\ell \cap \mathcal{P}_m = \emptyset$ for all $\ell, m = 1, 2, \dots, \lceil \frac{N}{M} \rceil$, $\ell \neq m$,
  c) $\bigcup_{j=1}^{\lceil \frac{N}{M} \rceil} \mathcal{P}_j = \{1, 2, \dots, N\}$,
  d) $\hat{d}_j > d_i$ for all $i \in \mathcal{P}_j$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$, and
  e) $\sum_{j=1}^{\lceil \frac{N}{M} \rceil} \hat{d}_j \leqslant T$.

Then a control logic, υ_T, obtained from Algorithm 2 is a solution to the feasibility problem (2).

Proposition 3 asserts that if all the plants are reachable and there exist positive integers, $\hat{d}_j$ and sets, $\mathcal{P}_j \subseteq \{1, 2, \dots, N\}$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$, that satisfy certain properties, then there exists a solution to the feasibility problem (2) and this solution can be computed by employing Algorithm 2. We split the set of all plants $\{1, 2, \dots, N\}$ into disjoint subsets, $\mathcal{P}_j$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$ and use $\hat{d}_j$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$ as the time duration for steering the state of the elements in $\mathcal{P}_j$ to zero with at most d_i-many non-zero control inputs. The cardinality of each $\mathcal{P}_j$ ensures that no more than M plants have non-zero control inputs at any time instant t and the upper bound on the sum of $\hat{d}_j$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$ ensures that the given initial states of all plants are steered to zero in the given time horizon T.

Our second result towards ensuring existence of a solution to the feasibility problem (2) is the following:

Proposition 4: Suppose that

D1) Each plant $i = 1, 2, \dots, N$ is reachable.

D2) There exist $\mathbb{N} \ni \hat{d}_i > d_i$ for all $i = 1, 2, \dots, N$ and $\mathcal{Q}_j \subseteq \{1, 2, \dots, N\}$, $j = 1, 2, \dots, p$ with $p \leqslant M$ such that
  a) $\mathcal{Q}_m \cap \mathcal{Q}_n = \emptyset$ for all $m, n = 1, 2, \dots, p$, $m \neq n$,
  b) $\bigcup_{j=1}^{p} \mathcal{Q}_j = \{1, 2, \dots, N\}$, and
  c) $\sum_{i \in \mathcal{Q}_j} \hat{d}_i \leqslant T$ for all $\mathcal{Q}_j$, $j = 1, 2, \dots, p$.

Then a control logic, υ_T, obtained from Algorithm 3 is a solution to the feasibility problem (2).

Algorithm 2 Construction of a control logic, υ_T, when conditions C1)-C2) are satisfied

Input: The plant dynamics, (A_i, b_i), $i = 1, 2, \dots, N$, the capacity of the communication network, M, the initial states, $x_i(0) = \xi_i$, $i = 1, 2, \dots, N$, time horizon, T, the numbers, $\hat{d}_j$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$ and the sets, $\mathcal{P}_j$, $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$.

Output: A control logic, υ_T.

1: **for** $j = 1, 2, \dots, \lceil \frac{N}{M} \rceil$ **do**
2: Compute $\tilde{d}_j = \sum_{k=1}^{j-1} \hat{d}_k$.
3: **for** each $i \in \mathcal{P}_j$ **do**
4: Set

$$\begin{pmatrix} u_i(\tilde{d}_j + 0) \\ \vdots \\ u_i(\tilde{d}_j + \hat{d}_j - 1) \end{pmatrix} = \begin{pmatrix} 0_{\hat{d}_j - d_i} \\ -\Psi_i^{-1} A_i^{\hat{d}_j} A_i^{\tilde{d}_j} \xi_i \end{pmatrix} \quad (3)$$

and $u_i(\tau) = 0$ for all $\tau \in \{0, 1, \dots, T-1\} \setminus \{\tilde{d}_j + 0, \dots, \tilde{d}_j + \hat{d}_j - 1\}$.
5: **end for**
6: **end for**
7: Output $\upsilon_T(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \end{pmatrix}$, $t = 0, 1, \dots, T-1$.

Proposition 4 asserts that if all the plants are reachable and there exist positive integers $\hat{d}_i$, $i = 1, 2, \dots, N$ and sets $\mathcal{Q}_j \subseteq \{1, 2, \dots, N\}$, $j = 1, 2, \dots, p$ with $p \leqslant M$ that satisfy certain properties, then there exists a solution to the feasibility problem (2) and this solution can be computed by employing Algorithm 3. For each plant $i = 1, 2, \dots, N$, we use $\hat{d}_i$ as the time duration for steering the state of the plant to 0_{d_i} with at most d_i-many non-zero control inputs. We split the set of all plants $\{1, 2, \dots, N\}$ into at most M disjoint subsets, $\mathcal{Q}_j$, $j = 1, 2, \dots, p \, (\leqslant M)$ such that the sum of $\hat{d}_i$ for all elements i in any $\mathcal{Q}_j$ does not exceed T. The number of $\mathcal{Q}_j$, $j = 1, 2, \dots, p$ in use ensures that not more than M plants have non-zero control inputs at any time instant t and the upper bound on the sum of $\hat{d}_i$, $i \in \mathcal{Q}_j$, $j \in \{1, 2, \dots, p\}$ ensures that the given initial states of all plants are steered to zero in the given time horizon T.

We now summarize our sufficient conditions for the existence of a solution to the feasibility problem (2). Consider a matrix, $\mathcal{M}$, with M rows and T columns. Suppose that we want to fill in the entries of $\mathcal{M}$ with elements from the set $\{1, 2, \dots, N\}$ obeying that each element must appear for a pre-specified number of consecutive instances $\hat{d}_j$, $j \in \{1, 2, \dots, \lceil \frac{N}{M} \rceil\}$ (resp., $\hat{d}_i$, $i \in \{1, 2, \dots, N\}$). Algorithm 2

Algorithm 3 Construction of a control logic, υ_T when conditions D1)-D2) are satisfied

Input: The plant dynamics, (A_i, b_i), $i = 1, 2, \dots, N$, the initial states, $x_i(0) = \xi_i$, $i = 1, 2, \dots, N$, time horizon, T, the integers, $\hat{d}_i$, $i = 1, 2, \dots, N$ and the sets, $\mathcal{Q}_j$, $j = 1, 2, \dots, p$.

Output: A control logic, υ_T.

Let $\mathcal{Q}_j(k)$ denote the k-th element of the set $\mathcal{Q}_j$, $k = 1, 2, \dots, |\mathcal{Q}_j|$.

for each $j = 1, 2, \dots, p$ **do**
  for each $k = 1, 2, \dots, |\mathcal{Q}_j|$ **do**
    Compute $\tilde{d}_{\mathcal{Q}_j(k)} = \sum_{\ell=1}^{k-1} \hat{d}_{\mathcal{Q}_j(\ell)}$.
    Set

$$\begin{pmatrix} u_{\mathcal{Q}_j(k)}\big(\tilde{d}_{\mathcal{Q}_j(k)} + 0\big) \\ \vdots \\ u_{\mathcal{Q}_j(k)}\big(\tilde{d}_{\mathcal{Q}_j(k)} + \hat{d}_{\mathcal{Q}_j(k)} - 1\big) \end{pmatrix} = \begin{pmatrix} 0_{\hat{d}_{\mathcal{Q}_j(k)} - d_{\mathcal{Q}_j(k)}} \\ -\Psi^{-1}_{\mathcal{Q}_j(k)} A^{\hat{d}_{\mathcal{Q}_j(k)}}_{\mathcal{Q}_j(k)} A^{\tilde{d}_{\mathcal{Q}_j(k)}}_{\mathcal{Q}_j(k)} \xi_{\mathcal{Q}_j(k)} \end{pmatrix} \quad (4)$$

    and $u_{\mathcal{Q}_j(k)}(\tau) = 0$ for all $\tau \in \{0, 1, \dots, T-1\} \setminus \big\{\tilde{d}_{\mathcal{Q}_j(k)} + 0, \dots, \tilde{d}_{\mathcal{Q}_j(k)} + \hat{d}_{\mathcal{Q}_j(k)} - 1\big\}$.
  end for
end for
Output $\upsilon_T(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \end{pmatrix}$, $t = 0, 1, \dots, T-1$.

splits the columns into segments of length $\hat{d}_1$, $\hat{d}_2$, ..., $\hat{d}_{\lceil \frac{N}{M} \rceil}$ and fill in the elements from $\mathcal{P}_1$, $\mathcal{P}_2$, ..., $\mathcal{P}_{\lceil \frac{N}{M} \rceil}$, respectively, while Algorithm 3 assigns elements from $\mathcal{Q}_1$, $\mathcal{Q}_2$, ..., $\mathcal{Q}_p$ for the corresponding $\hat{d}_i$, $i = 1, 2, \dots, N$ duration of time along the rows $1, 2, \dots, p\ (\leqslant M)$, respectively.

Propositions 3 and 4 are algebraically related as follows: If $|\mathcal{Q}_j| = \lceil \frac{N}{M} \rceil$ for all $j = 1, 2, \dots, p$, then we can pick $\mathcal{P}_k = \{\mathcal{Q}_1(k), \mathcal{Q}_2(k), \dots, \mathcal{Q}_p(k)\}$ and $\hat{d}_k = \max\{\hat{d}_{\mathcal{Q}_1(k)}, \hat{d}_{\mathcal{Q}_2(k)}, \dots, \hat{d}_{\mathcal{Q}_p(k)}\}$, $k = 1, 2, \dots, \lceil \frac{N}{M} \rceil$. Given a time horizon, T, it is clear that Proposition 4 can cater to a bigger N compared to Proposition 3 as the choice of $\hat{d}_i$ is specific to i and not a maximal value over a set of i's which can be of different dimension, d_i. In other words, Proposition 3 is more useful when $d_i \neq d$ for all $i = 1, 2, \dots, N$. Consider, for example, an NCS with $N = 4$ and $M = 2$. Let $d_1 = 1$, $d_2 = 2$, $d_3 = 3$ and $d_4 = 4$. Let $T = 7$. We have that Proposition 3 holds with $\hat{d}_1 = 2$, $\hat{d}_2 = 3$, $\hat{d}_3 = 4$, $\hat{d}_4 = 5$ and $\mathcal{Q}_1 = \{1, 4\}$, $\mathcal{Q}_2 = \{2, 3\}$. However, for any choice of disjoint $\mathcal{P}_1$, $\mathcal{P}_2 \subseteq \{1, 2, 3, 4\}$, we cannot choose $\hat{d}_1$, $\hat{d}_2$ such that $\hat{d}_1 + \hat{d}_2 \leqslant 7$.

Remark 4: Suppose that a solution, υ_T, to the feasibility problem (2) is computed by employing Algorithm 2 (resp., Algorithm 3). Notice that for each plant $i \in \{1, 2, \dots, N\}$, we have at most d_i-many non-zero control inputs in a $\hat{d}_j$, $j \in \{1, 2, \dots, \lceil \frac{N}{M} \rceil\}$ (resp., $\hat{d}_i$) duration of time. The remaining at least $\hat{d}_j - d_i$ (resp., $\hat{d}_i - d_i$) elements are 0. By construction of the control logic, υ_T, non-zero elements appear for at most M-many plants at any time instant. Consequently, for a scheduling logic, γ_T, there exist time instants $\tau \in \{0, 1, \dots, T-1\}$ such that $\gamma_T(\tau) = \emptyset$, i.e., no plant has access to the shared communication network. This feature is specific to our method of solving (2) used in Algorithms 2 and 3.

Remark 5: Analysis and design of a sparsest control sequence for linear systems (also known as a maximum hands-off control) has attracted a considerable research attention in the recent past, see e.g., [16], [15]. Such a sequence $\big(u(t)\big)_{t=0}^{T-1}$ achieves a desired control objective with a minimum number of non-zero elements and can be designed by minimizing the ℓ_0-norm of $\begin{pmatrix} u(0) \\ \vdots \\ u(T-1) \end{pmatrix}$. Our design of a scheduling and control logic, (γ_T, υ_T), presented in this paper is similar in spirit to maximum hands-off control. Indeed, instead of minimizing the ℓ_0-norm of control sequence for a specific plant, i.e., $\begin{pmatrix} u_i(0) \\ \vdots \\ u_i(T-1) \end{pmatrix}$, we minimize the ℓ_0-norm of the t-th element of the control sequence of each plant, i.e., $\begin{pmatrix} u_1(t) \\ \vdots \\ u_N(t) \end{pmatrix}$.

Remark 6: A widely used technique to solve a non-convex optimization problem is to solve its convex relaxation and find sufficient conditions on the parameters of the optimization problem that ensure that a solution of the original problem coincides with a solution to the convex relaxation. We identify the problem of analysis and design of a convex relaxation of the feasibility problem (2) (á la [15, Theorem 1]) as a topic of further investigation.

IV. Numerical experiment

We consider an NCS with number of plants, $N = 6$ and capacity of the shared communication network, $M = 2$. The plants are a discretised version of a linearized inverted pendulum system presented in [17, Section 4] with sampling time 0.05. We have $A_i = \begin{pmatrix} 1.0123 & 0.0502 \\ 0.4920 & 1.0123 \end{pmatrix}$, $b_i = \begin{pmatrix} 0.0123 \\ 0.4920 \end{pmatrix}$, $i = 1, 2, \dots, 6$. Let the time horizon, $T = 10$ and the initial states, $\xi_i = \begin{pmatrix} -0.2 \\ 0.1 \end{pmatrix}$, $i = 1, 2, \dots, 6$.

We note that all the plants are reachable. Thus, condition C1) holds. Further, condition C2) holds with $\hat{d}_1 = \hat{d}_2 = \hat{d}_3 = 3$ and $\mathcal{P}_1 = \{1, 2\}$, $\mathcal{P}_2 = \{3, 4\}$, $\mathcal{P}_3 = \{5, 6\}$. We employ Algorithm 2 and obtain the control logic $\upsilon_T(0) = \upsilon_T(3) = \upsilon_T(6) = \upsilon_T(9) = 0_6$, $\upsilon_T(1) = \begin{pmatrix} 8.2321 \\ 8.2321 \\ 0_4 \end{pmatrix}$, $\upsilon_T(2) =$

$$\begin{pmatrix} -8.0409 \\ -8.0409 \\ 0_4 \end{pmatrix}, \upsilon_T(4) = \begin{pmatrix} 0_2 \\ 10.0633 \\ 10.0633 \\ 0_2 \end{pmatrix}, \upsilon_T(5) = \begin{pmatrix} 0_2 \\ -9.2317 \\ -9.2317 \\ 0_2 \end{pmatrix},$$

$\upsilon_T(7) = \begin{pmatrix} 0_4 \\ 14.1547 \\ 14.1547 \end{pmatrix}$, $\upsilon_T(8) = \begin{pmatrix} 0_4 \\ -12.4959 \\ -12.4959 \end{pmatrix}$. The corresponding scheduling logic, γ_T, is: $\gamma_T(0) = \gamma_T(3) = \gamma_T(6) = \gamma_T(9) = \emptyset$, $\gamma_T(1) = \gamma_T(2) = \{1, 2\}$, $\gamma_T(4) = \gamma_T(5) = \{3, 4\}$, $\gamma_T(7) = \gamma_T(8) = \{5, 6\}$.

The state trajectory, $\left(\|x_i(t)\|\right)_{t=0}^{T-1}$, $i = 1, 2, \ldots, N$ under (γ_T, υ_T) is illustrated in Figure 1. It is observed that ξ_i is steered to 0_{d_i} for all plants $i = 1, 2, \ldots, N$ in the NCS.

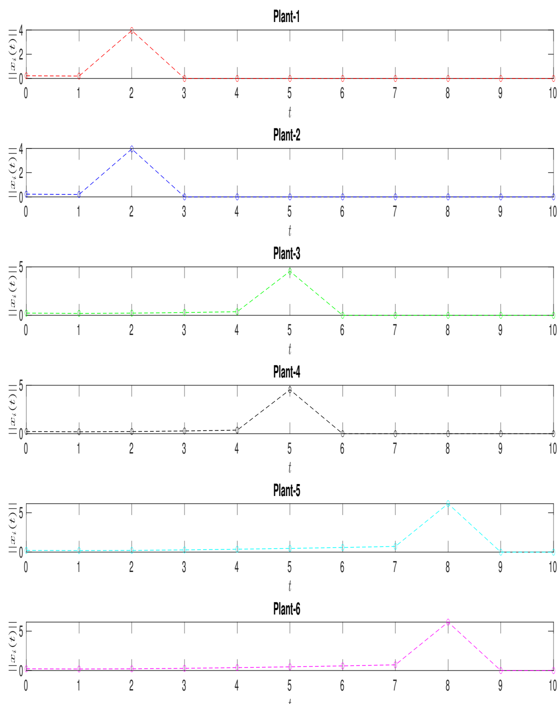

Fig. 1. $\left(\|x(t)\|\right)_{t=0}^{T-1}$ versus t for all plants $i = 1, 2, \ldots, 6$

We next note that condition D1) holds. Further, condition D2) holds with $\hat{d}_i = 3$, $i = 1, 2, \ldots, 6$ and $\mathcal{Q}_1 = \{1, 2, 3\}$, $\mathcal{Q}_2 = \{4, 5, 6\}$. Notice that we have the exact same scenario as for conditions C1)-C2) with $\hat{d}_j = 3$, $j = 1, 2, 3$ and $\mathcal{P}_1 = \mathcal{Q}_1$, $\mathcal{P}_2 = \mathcal{Q}_2$.

Remark 7: Notice that the non-zero entries of υ_T in our example above are of high magnitude. In practice, applicable range of control inputs may be limited based on inherent restrictions of components/instruments. We identify incorporating such constraints as a future direction of our research.

V. Concluding remarks

In this paper we presented an algorithm for the design of scheduling and control logic for NCSs under limited but ideal communication. Our algorithm involves a feasibility problem with sparsity constraint (ℓ_0-constraint) to compute the control logic, and the corresponding scheduling logic is designed using plants with non-zero control inputs. The use of sparse optimization techniques for the design of scheduling and control logic under communication limitations *and* uncertainties is currently under investigation and will be reported elsewhere.

References

[1] S. Al-Areqi, D. Görges, and S. Liu, *Event-based control and scheduling codesign: stochastic and robust approaches*, IEEE Transactions on Automatic Control, 60 (2015), pp. 1291–1303.

[2] S.-L. Dai, H. Lin, and S. S. Ge, *Scheduling-and-control codesign for a collection of networked control systems with uncertain delays*, IEEE Transactions on Control Systems Technology, 18 (2010), pp. 66–78.

[3] K. Gatsis, A. Ribeiro, and G. Pappas, *Control-aware random access communication*. Proceedings of the 7th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), 2016, Vienna, Austria, pp. 1-9.

[4] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, *A survey of recent results in networked control systems*, Proc. of IEEE Special Issue on Technology of Networked Control Systems, 95 (2007), pp. 138–162.

[5] D. Hristu-Varsakelis, *Feedback control systems as users of a shared network: Communication sequences that guarantee stability*. Proceedings of the 40th IEEE Conference on Decision and Control, 2001, Orlando, Florida, USA, pp. 3631-3636.

[6] D. Hristu-Varsakelis, *Short-period communication and the role of zero-order holding in networked control systems*, IEEE Transactions on Automatic Control, 53 (2008), pp. 1285–1290.

[7] D. Hristu-Varsakelis and L. Zhang, *LQG control of networked control systems with access constraints and delays*, International Journal of Control, 81 (2008), pp. 1266–1280.

[8] T. Ikeda and K. Kashima, *Sparse control node scheduling in networked systems based on approximate controllability metrics*, IEEE Trans. Control Netw. Syst., 9 (2022), pp. 1166–1177.

[9] A. Kundu, *Scheduling networked control systems under jamming attacks*, Proceedings of the 59th IEEE Conference on Decision and Control, (2020), p. 3298–3303.

[10] A. Kundu and D. E. Quevedo, *Stabilizing scheduling policies for networked control systems*, IEEE Transactions on Control of Network Systems, 7 (2020), pp. 163–175.

[11] ———, *On periodic scheduling and control for networked systems under random data loss*, IEEE Transactions on Control of Network Systems, 8 (2021), pp. 1788–1798.

[12] H. Lin, G. Zhai, L. Fang, and P. J. Antsaklis, *Stability and h-inf performance preserving scheduling policy for networked control systems*. Proc. of the 16th IFAC World Congress, 2005, Prague, Czech Republic.

[13] M. Lješnjanin, D. E. Quevedo, and D. Nešić, *Packetized MPC with dynamic scheduling constraints and bounded packet dropouts*, Automatica J. IFAC, 50 (2014), pp. 784–797.

[14] Y. Ma, J. Guo, Y. Wang, A. Chakrabarty, H. Ahn, P. Orlik, and C. Lu, *Optimal dynamic scheduling of wireless networked control systems*. Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, 2019, Montreal, Canada, pp. 77-86.

[15] M. Nagahara, J. Ostergaard, and D. E. Quevedo, *Discrete-time hands-off control by sparse optimization*, EURASIP Journal on Advances in Signal Processing, 76 (2016).

[16] M. Nagahara, D. E. Quevedo, and D. Nešić, *Maximum hands-off control: a paradigm of control effort minimization*, IEEE Trans. Automat. Control, 61 (2016), pp. 735–747.

[17] H. Rehbinder and M. Sanfridson, *Scheduling of a limited communication channel for optimal control*, Automatica, 40 (2004), pp. 491—-500.

[18] I. Saha, S. Baruah, and R. Majumdar, *Dynamic scheduling for networked control systems*. Proceedings of the 18th ACM International Conference on Hybrid Systems: Computation & Control, 2015, Seattle, Washington, pp. 98-107.

[19] M. Singh and A. Kundu, *A scheduling algorithm for networked control systems*, Proceedings of the IFAC Conference on Networked Systems (NecSys), (2021), pp. 37—-42.

[20] L. Zhang and D. Hristu-Varsakelis, *Communication and control co-design for networked control systems*, Automatica J. IFAC, 42 (2006), pp. 953–958.