Measurement Scheduling for Control Invariance in Networked Control Systems

Alessandro Colombo, Masoud Bahraini, and Paolo Falcone

***Abstract*— We discuss a new reachability problem for networked controlled system where a master node —the controller— broadcasts commands to a set of slave nodes, which must take turn to relay back state measurements. This problem finds applications in some robotics and intelligent transportation systems setups. Constraints on communication demand a coupled design of the controller and the measurement schedule. We prove that the problem is formally equivalent to the Pinwheel Problem from scheduling theory, and building on this result we provide conditions for schedulability and reachability. The results are illustrated in three numerical examples.**

I. Introduction

The problem of designing a feedback law to keep a system's state within a compact admissible region in the presence of uncertainties has been studied for over forty years [1], in countless different scenarios and with different modelling assumptions. The tasks involved in the computation of the sets reached by the state, under control and disturbance inputs, are generally known as *reachability* problems. Often, reachability is used to provide guarantees that a system's state can be indefinitely kept inside an admissible set or away from a bad set, in a framework known as *safety verification*. Problems of reachability and safety verification have found use in model predictive control design [2], control for coordination and collision avoidance of multiagent systems [3], [4], differential games [5], among others.

More recently, the evolution of embedded communication and computation devices and the arrival of the Internet of Things has moved the focus of much control theoretic research towards distributed versions of the reachability and verification problems [6]–[8]. In this setting, the designer faces the additional challenge of handling a control system where sensing, decision, and actuation are implemented at physically separate nodes of a network, which interact with each other through a communication link with non-negligible physical limitations [9]–[11].

An interesting subset of this broad family of problems, which has received little attention so far, regards a scenario where a central decision node is in charge of keeping the state of a set of independent subsystems within given admissible sets, receiving measurement data and broadcasting control commands through a common communication channel, as in Fig. 1. This is a model, for instance, of remotely sensed and actuated robotic systems based on CAN communication or, as we will discuss in our application example, of remote multiagent control setups for intelligent transportation systems field testing. This scenario shares some similarities with event driven control [11]. However, in our case the core problem is to guarantee invariance of the admissible sets despite the communication constraints, rather than to ensure stability while minimizing communication. As we will see, this shift in focus brings about a corresponding shift in the set of available tools.

A. Colombo is with DEIB, Politecnico di Milano, Italy. alessandro.colombo@polimi.it

M. Baharaini and P. Falcone are with the department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. masoudb/falcone@chalmers.se. This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation

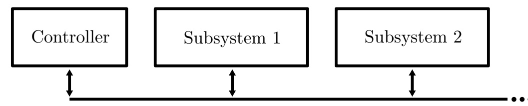

Fig. 1. A Controller and n subsystems share a common channel to exchange control and measurement data.

In this paper we target a reachability and safety verification problem, in discrete time, for the above-described networked control system. This is, to the best of our knowledge, the first attempt at the formalization and solution of this particular subclass of networked reachability problems. With respect to a standard reachability problem, the limitations of the communication channel imply that the controller can measure only a subset of nodes at any given time. Thus, a suitable *measurement schedule* must be designed concurrently with the control law in order to ensure the proper performance. We formalize a general model for the control problem class, and prove that the measurement schedule design problem is formally equivalent to the *Pinwheel Problem* from scheduling theory [12]. This gives us a powerful set of tools to co-design scheduling and control algorithms, and to provide guarantees on permanent schedulability, as a function of the dynamics of each of the systems' nodes.

We introduce the mathematical model and problem formulation in Sec. II-A. Then, the main theoretical results are discussed in Sec. III. An application of these results to the challenge of coordinating a set of remotely actuated agents is discussed in Sec. IV and V, where our algorithms are used to design a tracking feedback with guaranteed error bounds for all remotely controlled agents, and to compute the (approximate) minimum tracking error bound compatible with a given set of agents and communication constraints.

II. Centralized controller with communication constraints.

A. Model and notation

Consider a set of q discrete-time time-invariant linear systems

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + F_i v_i(t), \qquad (1)$$

where $x_i \in \mathbb{R}^{n_i}$ is the state, $u_i \in \mathbb{R}^{m_i}$ the control input, $v_i \in \mathbb{R}^{p_i}$ an additive bounded noise within the polytope $\mathcal{V}_i$, defined as

$$\mathcal{V}_i := \{v_i \in \mathbb{R}^{p_i} : E_i v_i \le f_i\},$$

for some matrix E_i and vector f_i of suitable dimension.

For each system i we define an *admissible set* $\mathcal{A}_i \subset \mathbb{R}^{n_i}$, which describes the set of states within which the state x_i should be kept. The q systems (1) describes the dynamics of a set of decoupled subsystems, which may represent independent agents in a multiagent system, or dynamically decoupled components of a larger plant. We write

$$x(t+1) = Ax(t) + Bu(t) + Fv(t) \qquad (2)$$

the model of the q subsystems together, and call $\mathcal{A} := \mathcal{A}_1 \times \mathcal{A}_2 \times \dots$ the admissible set of (2), obtained as the Cartesian product of the q admissible sets $\mathcal{A}_i$. Letting $n := \sum n_i$, $m := \sum m_i$, $p := \sum p_i$, the above system has $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $v \in \mathbb{R}^p$, while A, B, and F are block-diagonal.

Let us assume that (2) is controlled by a central controller, which stabilizes the system through a full-state static feedback law $u(k) = -Kx(k)$, and assume, for the moment, that the full state $x(k)$ is known to the central controller at each time step. We thus have the following closed-loop system

$$x(t+1) = (A - BK)x(t) + Fv(t). \qquad (3)$$

The above equation will be modified later to account for limitations on the state information available to the controller.

B. Reachability and Invariance Properties

We say that $\mathcal{S} \subset \mathbb{R}^n$ is *robust invariant* for the system (3), if

$$\forall\, x(t) \in \mathcal{S} \text{ and } \forall\, v \in \mathcal{V}, x(t+1) \in \mathcal{S}.$$

Let $\{\mathcal{S}\}$ be the set of all robust invariant sets of the above system. We call the *maximal robust invariant set* and denote it by $\mathcal{S}_\infty$ the maximal, with respect to set inclusion, among all the robust invariant sets that are subsets of $\mathcal{A}$:

$$\mathcal{S}_\infty := \max\{\mathcal{S} \in \{\mathcal{S}\} : \mathcal{S} \in \mathcal{A}\}.$$

A robust invariant set $\mathcal{S}$ as defined above is guaranteed to contain a forward-time trajectory of system (3), provided $x(0) \in \mathcal{S}$, regardless of the disturbance v.

For the autonomous system $x(t+1) = f(x(t), v(t))$ affected by disturbance $v(t)$, we define the 1-*step reachable set* as the set of states that can be reached in *one step* from the set of initial states $\mathcal{O}$:

$$\mathrm{Reach}_1^f(\mathcal{O}) := \{x \in \mathbb{R}^n : \exists x_0 \in \mathcal{O},\ \exists v \in \mathcal{V},\ \text{such that } x = f(x_0, v)\}. \qquad (4)$$

The extension of (4) to a t-step reachable set is straightforward. Numerical tools for the calculation of $\mathcal{S}_\infty$ and $\mathrm{Reach}_t^f(\mathcal{O})$ can be found in [13].

C. Problem formulation

We assume that communication between the central controller and the subsystems is routed through a common channel (as in Fig. 1), for instance a wireless network or a CAN link. This poses constraints on the amount of information that can be exchanged between central controller and subsystems at each time step. In particular, even though more challenging assumptions are in principle possible, we assume that the central controller can broadcast the full control vector $u(t)$ to all subsystems at each time step, while only a single subsystem can communicate its state to the central controller at any given time step. To encode this constraint, we can proceed as follows. Consider the q square $n \times n$ matrices

$$\tilde{C}_1 := \begin{pmatrix} I_{11} & & & 0 \\ & \mathbf{0}_{22} & & \\ & & \mathbf{0}_{33} & \\ 0 & & & \ddots \end{pmatrix},\ \tilde{C}_2 := \begin{pmatrix} \mathbf{0}_{11} & & & 0 \\ & I_{22} & & \\ & & \mathbf{0}_{33} & \\ 0 & & & \ddots \end{pmatrix}, \dots,$$

where the diagonal blocks I_{ii} are identities of dimension n_i while $\mathbf{0}_{ii}$ are zero square matrices of dimension n_i, and consider q more matrices $\hat{C}_i := I - \tilde{C}_i$. Call $\mathcal{C}$ the set of n matrices $C_i := (\tilde{C}_i | \hat{C}_i) \in \{0,1\}^{n \times 2n},\ i = 1, \dots, n$. Each row of C_i sums to 1, and the unit element of each row identifies whether the corresponding element of $\hat{x}$ is updated with a new measure of x, or must be estimated by the central controller without use of new information.

The closed-loop dynamics (3) with communication constraints can then written using the matrices in $\mathcal{C}$ as

$$\begin{aligned} x(t+1) &= Ax(t) - BK\hat{x}(t) + Fv(t), \\ \hat{x}^-(t+1) &= (A - BK)\hat{x}(t), \\ \hat{x}(t+1) &= C_{\delta(t)} \begin{pmatrix} x(t+1) \\ \hat{x}^-(t+1) \end{pmatrix},\ C_{\delta(t)} \in \mathcal{C}, \end{aligned} \qquad (5)$$

where $\delta \in \{1, 2, \dots, n\}$ is a scheduling signal.

The considered setting describes a master-slave control broadcast network and constrained communication between slaves and master. We postpone to Sec. IV and V a more thorough discussion of practical problems that fall within this framework. At this stage, we can instead identify two theoretical issues worth of investigation

Problem 1 (Schedulability)**.** Given a controlled system (5), an admissible set $\mathcal{A}$, along with a set of measurement matrices $\mathcal{C}$, determine whether, for all $x(0) \in \mathcal{S}_\infty$, there exists a measurement schedule $\delta(t)$ which guarantees $x(t) \in \mathcal{S}_\infty,\ \forall\, t > 0$.

We will say that system (5) *is schedulable* if Problem 1 has positive answer.

Problem 2 (Measurement schedule design)**.** Given the controlled, *schedulable* system (5), an admissible set $\mathcal{A}$, a set of measurement matrices $\mathcal{C}$, and an initial condition $x(0) \in \mathcal{S}_\infty$, determine a measurement schedule $\delta(t)$ which guarantees $x(t) \in \mathcal{S}_\infty,\ \forall\, t > 0$.

III. Main Results

To address Problems 1 and 2, we exploit the similarity of their structure with that of the Pinwheel Problem, from the scheduling literature [12], [14].

Notice first of all that Problem 1 is a *decision problem*, since its solution is a *yes* or *no* answer. An *instance* I of Problem 1 is the full set of data (model and initial conditions) necessary to define the problem. We say that Problem 1 *accepts* an instance I if its answer to I is *yes*. Finally, two decision problems are *equivalent* if there exists a polynomial-time mapping of instances of one to the other, and one accepts an instance I if and only if the other accepts it.

We now define the Pinwheel Problem, and prove its equivalence with Problem 1.

Consider a subsystem i, and let us assume that x_i is measured at time 0, so that $\hat{x}_i(0) = x_i(0)$, and that no further measure is available for $t > 0$. Then, the evolution of x_i under (5) can then be written as

$$x_i(t+1) = A_i x(t) - BK\left(A_i - B_i K_i\right)^t x(0) + Fv(t),$$
$$:= \hat{f}_i(x(t), v(t)) \quad (6)$$

Using $\hat{f}$ from the above equation, and the corresponding reach operator $\mathrm{Reach}_t^{\hat{f}}(\cdot)$, we can compute an upper bound to the time distance between two measurements of x_i needed to guarantee that the state can be kept within $\mathcal{A}_i$ despite the growing estimation error. This is done in Algorithm 1.

Algorithm 1 Computation of α_i

```
for all i ∈ 1,...,q do
    compute S_∞i.
    define α_i := max{t : Reach_t^{f̂_i}(S_∞i) ⊆ S_∞i}.
end for
return {α_1,...,α_q}
```

We have:

Lemma 1. If $x_i(0) \in \mathcal{S}_{\infty i}$, and x_i evolves according to (6),

$$x_i(t) \in \mathcal{S}_{\infty i},\ \forall\, t \in \{1, \ldots, \alpha_i\},$$

while

$$\exists\, x_i(0) \in \mathcal{S}_{\infty i}, \{v(0), \ldots, v(\alpha_i)\} : x_i(\alpha_i + 1) \notin \mathcal{S}_{\infty i}$$

Proof. The statement is a consequence of the definition of reachable set. □

Consider now the following:

Pinwheel Problem (From [14])**.** Given the set of integers $\alpha_1, \ldots, \alpha_q$, determine the existence of an infinite sequence over the symbols $1, \ldots, q$ such that there is at least one symbol i within any subsequence of α_i consecutive symbols.

In the light of Lemma 1, it is a simple exercise to prove the following

Theorem 1. *Problem 1 and the Pinwheel Problem are equivalent, with Algorithm 1 as the mapping of instances of Problem 1 to instances of the Pinwheel Problem.*

We can thus employ results formulated in the scheduling literature for the Pinwheel Problem to address the solutions of our Problems 1 and 2. Let us begin with the following result:

Theorem 2 (Theorem 2.1 in [12])**.** *All instances of the Pinwheel Problem that admit a schedule admit a cyclic schedule, i.e., a schedule whose symbols repeat periodically.*

Hence, if (5) is schedulable, then it is schedulable by a cyclic schedule. We will denote

$$\boldsymbol{\delta} := \{\delta(1), \ldots, \delta(t)\}.$$

a full cycle of such a cyclic schedule. We will use this fact in the design of solution algorithms to Problem 2. Conditions for schedulability have been formulated in terms of the *density* of the problem instance, which is defined as

$$\rho(I) := \sum_i \frac{1}{\alpha_i}.$$

While an upper bound to the density guaranteeing schedulability is not known, we have the following sufficient condition for schedulability:

Lemma 2 (From [15])**.** All instances I of the Pinwheel Problem with $\rho(I) \leq 0.7$ are schedulable.

Notice that it is easy to construct schedulable instances with $\rho = 1$ (for example two subsystems with $\alpha_1 = \alpha_2 = 2$), and non-schedulable instances with $0.7 < \rho < 1$ (for example three subsystem with $\alpha_1 = 2, \alpha_2 = 3, \alpha_3 = 7$, as we discuss in the examples). We now have a solution to Problem 1:

Theorem 3 (Solution to Problem 1)**.** *Given an instance I of Problem 1, a necessary condition for schedulability is $\rho(I) \leq 1$; a sufficient condition for schedulability is $\rho(I) \leq 0.7$.*

Proof. Using Theorem 1, schedulability of an instance of Problem 1 corresponds to schedulability of a corresponding Pinwheel Problem. The necessary condition simply follows from the fact that the terms $\frac{1}{\alpha_i}$ in the density function correspond to the minimum fraction of time instants that should be allocated to measurement of subsystem i out of any $m = \prod_i \alpha_i$ subsequent time instants. Their sum must thus be smaller than 1 for an instance to be schedulable. The sufficient condition follows from Lemma 2. □

According to Lemma 2 and Theorem 3, with appropriate algorithms we can design a cyclic schedule for any instance of Problem 1 with density $\rho \leq 0.7$. Examples of such algorithms are found, for example, in [15]. For the sake of simplicity, we report in Algorithm 2 a variation of the algorithm *SimpleGreedy*, from [12], which applies to instances with density $\rho \leq 0.5$.

Algorithm 2 Measurement scheduling algorithm (**input**: $\alpha_1, \dots, \alpha_q$, **output**: δ)

define $\beta_i := 2^j$, where $j := \max_{j\in\mathbb{N}} : 2^j \le \alpha_i$
order β_i so that $\beta_i \le \beta_{i+1}$
define $m := \prod_i \beta_i$
define a sequence of empty slots indexed 0 through $2m-1$
for $i := 1$ **to** q **do**
  $j :=$ smallest index of an empty slot
  repeat
    put i into slot j
    $j := j + \beta_i$
  until $j \ge 2m$
end for
Delete all empty slots
Assign to each slot j a vector $c_j := (c_{j,1}, \dots, c_{j,q})$, where $c_{j,l}$ denotes the number of slots since the last occurrence of l
Locate indices s and t that have been assigned identical vectors
return $\boldsymbol{\delta} :=$ the contents of slots s through $t-1$

Lemma 3 (Theorem 3.1 and Corollary 3.2 in [12])**.** Algorithm 2 computes a cyclic schedule for any instance I of the Pinwheel Problem with $\rho(I) \le 0.5$.

Algorithm 2 finally provides a solution to Problem 2:

Theorem 4 (Solution of Problem 2)**.** *For all instances with $\rho(I) \le 0.5$, a measurement schedule computed through Algorithm 2 ensures $x \in \mathcal{A}$, $\forall\, t \ge 0$.*

Proof. The statement follows from Theorem 1 and Lemma 3. □

IV. Trajectory Tracking

In Sec. III, we have formulated the problem of finding admissible measurement schedules, which keep the state trajectories of the system (5) within an invariant set. Such formulation is next extended to the problem of *tracking a reference trajectory without exceeding a given error bound*. Let x_r be the reference state, generated as solution of the equation

$$x_r(t+1) = Ax_r(t) + Bu_r(t),$$

where the input u_r is designed to give the desired reference. Similarly to (5), we can define the following closed-loop dynamics under the measurement schedule $\delta(t)$:

$$\begin{aligned} x(t+1) &= Ax(t) - BK(\hat{x}(t) - x_r(t)) + Bu_r(t) + Fv(t), \\ x_r(t+1) &= Ax_r(t) + Bu_r(t), \\ \hat{x}^-(t+1) &= (A-BK)\hat{x}(t) \\ \hat{x}(t+1) &= C_{\delta(t)} \begin{pmatrix} x(t+1) \\ \hat{x}^-(t+1) \end{pmatrix}, \; C_{\delta(t)} \in \mathcal{C}. \end{aligned} \tag{7}$$

The system (7) can be rewritten in terms of the actual tracking error $\theta = x - x_r$ and estimated tracking error $\hat{\theta} = \hat{x} - x_r$, as follows

$$\begin{aligned} \theta(t+1) &= A\theta(t) - BK\hat{\theta}(t) + Fv(t), \\ \hat{\theta}^-(t+1) &= (A-BK)\hat{\theta}(t) \\ \hat{\theta}(t+1) &= C_{\delta(t)} \begin{pmatrix} \theta(t+1) \\ \hat{\theta}^-(t+1) \end{pmatrix}, \; C_{\delta(t)} \in \mathcal{C}. \end{aligned} \tag{8}$$

The above equation is formally equivalent to (5). Hence, we can use the results of Sec. III to find $\delta(t)$ (i.e., schedule the communication) such that the error θ remains bounded within a given box.

Let $\Theta \in \mathbb{R}^n$ be the admissible set of the tracking error θ. We assume that Θ contains the vector 0. As discussed in Sec. II-B, we can calculate a maximal robust invariant set $\mathcal{S}_\infty$ with respect to the admissible set Θ. Given $\mathcal{S}_\infty$ and the corresponding integers $\{\alpha_1, \dots, \alpha_q\}$ obtained from Algorithm 1, we can use Algorithm 2 to derive a cyclic measurement schedule ensuring

$$\theta(0) \in \Theta \Rightarrow \theta(t) \in \Theta, \,\forall\, t > 0.$$

In the reference tracking problem for the system (7) with constrained communication, the set Θ bounds the tracking error. It is then natural to question *what is the smallest set Θ for which schedulability can be guaranteed, given the communication constraints*. The solution of such a problem provides a measure of the performance of the controlled system, while tracking an arbitrary trajectory, provided the initial tracking error is sufficiently small.

Define for each subsystem i, an admissible error set

$$\Theta_i(\gamma_i) := \{\theta_i \in \mathbb{R}^n : M_i\theta_i \le \gamma_i \mathbf{1}\}, \tag{9}$$

where $\mathbf{1}$ is the vector with unit elements, M_i is a suitable matrix, and $\gamma_i \in \mathbb{R}_+$. We can formulate a third control problem:

Problem 3 (Minimization of Θ)**.** Given a controlled system (7) with error dynamics (8), along with a set of measurement matrices $\mathcal{C}$, solve

$$\min_{\gamma_1,\dots,\gamma_q} \quad (\gamma_1, \dots, \gamma_q)\,, \tag{10a}$$

$$s.t.$$

$$\sum_i \frac{1}{\alpha_i(\gamma_i)} \le 0.7, \tag{10b}$$

Note that the problem (10) is a multi-objective optimization problem, for which multiple Pareto optimal solutions may exist.

V. Numerical results

Next, results from numerical simulations are shown for three examples, where the communication scheduling approaches proposed in Sec. III-IV are applied. In particular Example 1, discussing a simple case with three decoupled systems, illustrates the necessary condition for schedulability given by Theorem 3; examples 2 and 3 discus a trajectory tracking problem for a multiagent model.

Example 1 (Necessary condition for schedulability (Theorem 3))**.** Consider a set of three 1-dimensional systems

$$x_i(k+1) = x_i(k) + u_i(k) + v_i(k), \tag{11}$$

where the states, inputs, and disturbances are constrained within the sets $\mathcal{X}_i = \{-1 \le x_i \le 1\}, \mathcal{U}_i = \{-1 \le u_i \le 1\}, \mathcal{V}_i = \{-\tilde{v}_i \le v_i \le \tilde{v}_i\}$, $i = 1,\ 2,\ 3$, and $\tilde{v}_1 = 0.4,\ \tilde{v}_2 = 0.25,\ \tilde{v}_3 = 0.12$. With the state-feedback control

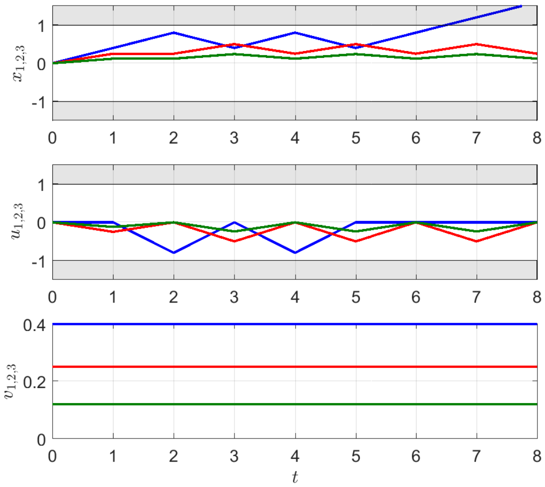

Fig. 2. State, input, and disturbance trajectories for the three subsystems in Example 1. The shaded bands identify state values out of the admissible set, and the input values out of bound.

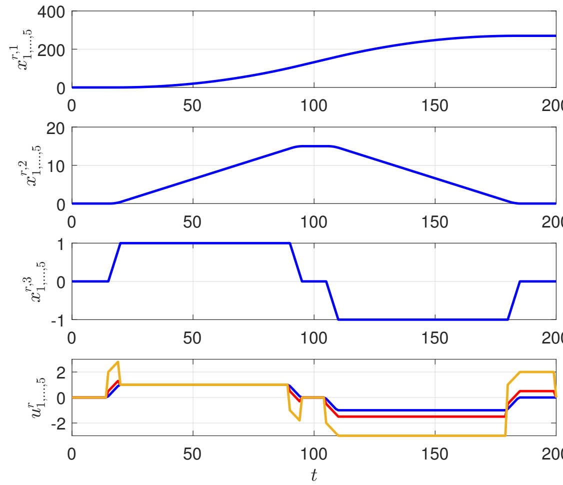

Fig. 3. Reference trajectories for the five vehicles.

law $u_i(k) = -x_i(k)$, it is straightforward to show that $S_{\infty i} = \mathcal{X}_i, \forall i$, and $\alpha_1 = 2$, $\alpha_2 = 3$, and $\alpha_3 = 7$. The *density* for this problem is $0.7 < \rho = \frac{41}{42} < 1$ and, according to Theorem 3, the existence of an admissible measurement schedule for the system (11) it is not guaranteed. Indeed, subsystem 1 (with $\alpha = 2$) needs communicating every 2 steps, and subsystem 2 (with $\alpha = 3$) must therefore fill in all remaining communication slots, leaving no slots available for subsystem 3, regardless of its α-value.

The effect of the lack of schedulability on the behavior of the three closed-loop subsystems is shown in Fig. 2, where the schedule $\delta(t) = \{1, 2, 1, 2, 1, 2, 3, 2, \ldots\}$ has been adopted, and the disturbances are set to their upper bounds. We can see that the state of subsystem 1 leaves the admissible set at $t = 7$, when the state of subsystem 3 is measured instead of that of 1.

Example 2 (Reference tracking for remote-controlled vehicles)**.** Consider now the case of five remotely controlled vehicles, described by the models

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + F_i v_i(t), \forall\, i$$

where

$$A_i = \begin{bmatrix} 1 & h & 0 \\ 0 & 1 & h \\ 0 & 0 & 1 - \frac{h}{\tau_i} \end{bmatrix},\ B_i = \begin{bmatrix} 0 \\ 0 \\ \frac{h}{\tau_i} \end{bmatrix},\ F_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

and $\tau_1 = 0.1$, $\tau_2 = \tau_3 = 0.5$, $\tau_4 = \tau_5 = 2$, $h = 0.2$.

The longitudinal motion of the five vehicles must track the five reference trajectories in Fig. 3 within prescribed error bounds, to realize a specified traffic scenario. Such situations occur, for instance, when setting up full-scale test scenarios for driver-assist systems.

The reference state trajectories are generated by the following dynamical model

$$x_i^r(t+1) = A_i x_i^r(t) + B_i u_i^r(t), \forall\, i$$

while the tracking inputs are defined as

$$u_i(t) = \underbrace{K_i\left(\hat{x}_i(t) - x_i^r(t)\right)}_{u_i^{fb}(t)} + \underbrace{B_i u_i^r}_{u_i^{ff}(t)}, \forall\, i,$$

where $u_i^{ff}(t)$ are feed-forward terms and the gains of the feedback terms $u_i^{fb}(t)$ are

$$\begin{aligned} K_1 &= \begin{bmatrix} 12.5000 & 7.5000 & 0.5000 \end{bmatrix}, \\ K_2 = K_3 &= \begin{bmatrix} 62.4999 & 37.5000 & 6.5000 \end{bmatrix}, \\ K_4 = K_5 &= \begin{bmatrix} 249.9997 & 149.9999 & 29.0000 \end{bmatrix}. \end{aligned}$$

The feedback terms $u_i^{fb}(t)$ are constrained to belong to the sets $\mathcal{U}_i = \{-4 \le u_i^{fb} \le 1.5\}, \forall\, i$, while the disturbances are assumed to be bounded within the sets $\mathcal{V}_i = \{-\tilde{v}_i \le v_i \le \tilde{v}_i\}$ with $\tilde{v}_1 = 0.06$, $, \tilde{v}_2 = \tilde{v}_3 = 0.0015$, $\tilde{v}_4 = \tilde{v}_5 = 0.0005$.

By defining the tracking errors as $\theta_i = x_i - x_i^r$ and their estimates as $\hat{\theta}_i = \hat{x}_i - x_i^r$, the errors dynamics can be derived as in (7). For each system, the tracking errors should be kept within the following bounds $\Theta_i = \left\{ \begin{bmatrix} -1 \\ -0.1 \end{bmatrix} \le \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \theta_i \le \begin{bmatrix} 1 \\ 0.1 \end{bmatrix} \right\}$, $\forall\, i$.

Algorithm 1 calculates the following $\alpha_1 = 4$, $\alpha_2 = \alpha_3 = 17$, $\alpha_4 = \alpha_5 = 19$, while Algorithm 2 calculates the following cyclic schedule, of which we report one full cycle:

$$\boldsymbol{\delta} = \{1, 2, 3, 4, 1, 5, 1, 1\}. \tag{12}$$

Note that feasible schedules with shorter cycle length would be possible: Algorithm 2 does not necessarily return a *minimal* schedule.

The tracking errors for the above schedule, along with the corresponding feedback control actions and the disturbances, are reported in Fig. 4, and are compared with those obtained with a simple round-robin schedule in Fig. 5. While the measurement schedule (12) keeps the position and velocity errors of all systems within the prescribed bounds, with the

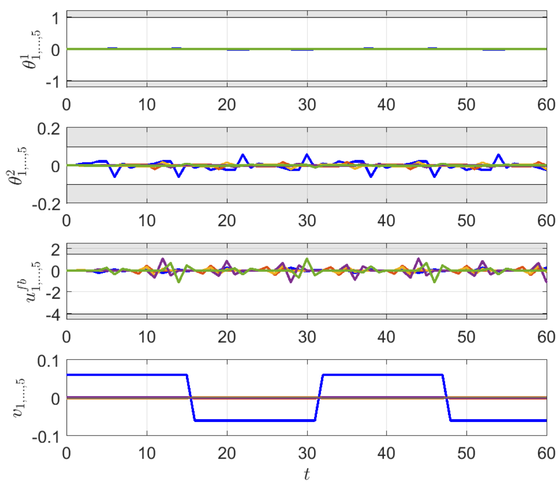

Fig. 4. The five colours identify state (θ_i^1, θ_i^2) and feedback input u_i^{fb} trajectories of the five vehicles, with measurement schedule (12). The shaded bands identify state values out of the admissible set, and input values out of bound. Notice that all subsystems remain within their admissible sets.

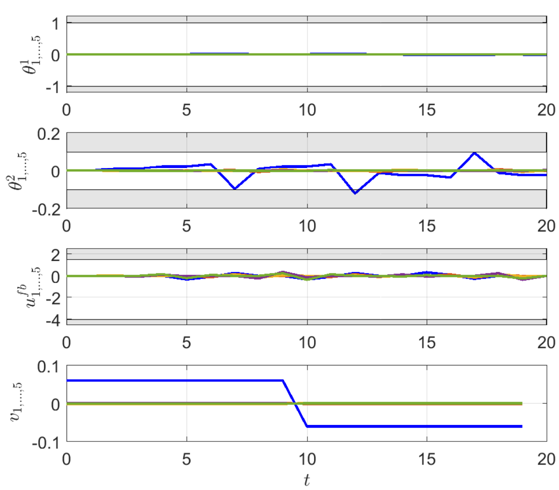

Fig. 5. Trajectories of the five vehicles with a round-robin measurement schedule, colour coding as in Fig. 4. Notice that the blue subsystem's θ^2 state leaves the admissible set at $t = 12$, due to inappropriate scheduling.

round-robin schedule, the velocity tracking error of one of the subsystems exceeds the bounds.

Example 3 (Minimization of the bounds on the tracking errors)**.** In this example, we solve Problem 3 for the problem considered in Example 2. Let us define the geometry of the admissible set with the matrix

$$M_i = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \ \forall i, \tag{13}$$

in (9). The solution of (10) leads to $\gamma_1 = \gamma_2 = \ldots = \gamma_5 = 0.0600$, with $\mathcal{S}_\infty = \mathcal{A}$. Given (13), this corresponds to the following error bounds:

$$-0.6 \le \theta_i^1 \le 0.6, \ \forall\, i,$$
$$-0.06 \le \theta_i^2 \le 0.06, \ \forall\, i,$$

and $\alpha_1 = 3,\ \alpha_2 = \alpha_3 = 13,\ \alpha_4 = \alpha_5 = 16,\ \rho(I) = 0.6122$. A cyclic measurement schedule satisfying the constraints is

$$\boldsymbol{\delta} = \{1, 2, 3, 1, 4, 5\}.$$

Note that the calculated γ_1 provide significantly smaller bounds than the ones used in Example 2.

VI. Conclusions

We have discussed a new class of reachability and verification problems for networked controlled systems. We have linked the reachability problem to the Pinwheel Problem, and we have used results from the scheduling literature to solve the control and measurement schedule design. Our hypotheses on the network structure and communication constraints encompass some examples of practical interest, but are nonetheless quite restrictive. More challenging theoretical questions arise if we allow a richer communication structure or a more complex network topology; in these cases the requirements and constraints on communication between network nodes will likely map onto more complex structures than the Pinwheel Problem. Our results can provide some intuition as to how these scenarios can be tackled.

References

[1] D. P. Bertsekas, "Infinite-time reachability of state-space regions by using feedback control," *IEEE Trans. Autom. Control*, vol. 17, pp. 604–613, 1972.

[2] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[3] M. Prandini and J. Hu, "Application of reachability analysis for stochastic hybrid systems to aircraft conflict prediction," *IEEE Trans. Autom. Control*, vol. 54, pp. 913–917, 2009.

[4] A. Colombo and D. Del Vecchio, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Trans. Autom. Control*, vol. 60, pp. 1515–1527, 2015.

[5] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, pp. 947–957, 2005.

[6] S. M. Loos, A. Platzer, and L. Nistor, "Adaptive cruise control: Hybrid, distributed, and now formally verified," in *Proceedings of the 17th International Symposium on Formal Methods*, 2011, pp. 42–56.

[7] A. Colombo, G. Rodrigues De Campos, and F. Della Rossa, "Control of a city road network: Distributed exact verification of traffic safety," *IEEE Trans. Autom. Control*, vol. 62, pp. 4933–4948, 2017.

[8] A. Colombo and F. Della Rossa, "Composition of least restrictive controllers, with application to collision avoidance in multiagent systems," *IEEE Trans. Autom. Control*, vol. Accepted, 2018.

[9] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009, electronically available at http://coordinationbook.info.

[10] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, pp. 1291–1297, 2011.

[11] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conference on Decision and Control*, 2012, pp. 3270–3285.

[12] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel, "The pinwheel: A real-time scheduling problem," in *Proceedings of the Hawaii International Conference on System Science*, 1989, pp. 693–702.

[13] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[14] C.-C. Han, K.-J. Lin, and C.-J. Hou, "Distance-constrained scheduling and its applications to real-time systems," *Transactions on Computers*, vol. 45, pp. 814–826, 1996.

[15] D. Chen and A. Mok, "The pinwheel: A real-time scheduling problem," in *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Champan & Hall, 2004, ch. 27.