Optimal Control Design for Perturbed Constrained Networked Control Systems

Masoud Bahraini, Mario Zanon, Alessandro Colombo, and Paolo Falcone

***Abstract*—This paper focuses on an optimal control design problem for a class of perturbed networked control systems where a number of systems, subject to state and input constraints, share a communication network with limited bandwidth. We first formulate an optimal control design problem with a constant feedback gain in order to minimize the communication demand for each system while guaranteeing satisfaction of state and input constraints; we show that this optimization problem is very hard to solve. Then, we formulate the same optimal control design problem with a non-constant feedback gain; we argue that this problem is less difficult and results in a lower, or equal, communication demand in comparison to the design with the constant feedback gain. We illustrate and compare these optimal control designs by a simple example.**

***Index Terms*—Networked control systems, Control over communications, Constrained control, Robust control, Predictive control for linear systems.**

I. Introduction

A Networked control system (NCS) is a system whose feedback is closed through a communication channel. NCSs have several advantages over traditional systems. On one hand, NCSs eliminate wiring, thus reducing the complexity and cost of connected systems. In addition, modification and upgrade of NCSs can be performed easily since no major change in the structure is needed. As a result, NCSs have a wide range of applications. On the other hand, communication networks introduce some challenging issues such as limited bandwidth, delays, and packet dropouts. These issues degrade the system performance and may cause instability [1].

Typically, a medium access control (MAC) mechanism is designed to share communication resources. There are two types of MAC mechanisms: (a) random access, in which the systems gain access to the medium randomly, and (b) scheduling, in which access is assigned according to a deterministic rule [2], [3]. Unlike random access mechanisms, scheduling mechanisms allow one to give performance guarantees. This makes them more suitable for safety critical NCSs. Furthermore, scheduling mechanisms may perform better than random access mechanisms when the effects of the communication network, such as delays and packet dropouts, are explicitly considered [4].

M. Baharaini and P. Falcone are with the department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. masoudb/falcone@chalmers.se.

M. Zanon is with IMT School for Advanced Studies Lucca, Italy. mario.zanon@imtlucca.it

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

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.

Control and scheduling codesign has gained attention in NCSs. For instance, this codesign was formulated as a mixed-integer quadratic optimization problem in [5]. A generalization of this work was proposed in [6] in which output feedback was considered. However, this work did not consider any uncertainty in either the communication link or systems. Similarly, the authors in [7] considered a finite horizon linear quadratic cost function and formulated a mixed integer quadratic problem to design the feedback policy and communication schedule in presence of network constraints. However, none of these works consider state and input constraints in the codesign.

In [8] and [9], the authors considered perturbed linear systems with state and input constraints. The goal was to design an MPC policy that reduces the communication demand while guaranteeing constraints satisfaction. However, this MPC policy was dependent on a static feedback gain which deteriorates the optimal solution. In [10], the same authors tried to address the communication aspect of the problem by considering a mixed-integer optimization problem. However, this communication scheduling problem is hard to solve at each time instant and there is no guarantee for the existence of a feasible solution when the bandwidth is limited.

We proposed scheduling strategies for constrained NCSs in [11], [12] which guarantee robust invariance of linear time invariant systems, i.e., guarantee robust satisfaction of the state and input constraints, in a shared communication medium scenario. While schedule designs in these papers take the control policy into account, the control design does not use any information regarding the scheduling policy. In this paper, we aim at improving our previous results by introducing information about the schedule design procedure in the controller design procedure. To that end, we formulate an optimal feedback design problem to minimize the communication demand for each system. We show that this design problem results in a very difficult optimization problem for linear state feedback; nevertheless, this problem becomes a standard quadratic optimization problem when the constant feedback gain consideration is dropped and the optimization is defined over the input sequence in a finite horizon.

The rest of the paper is organized as follows. In Section II, necessary definitions and results are recalled. Section III addresses optimal design of static linear state feedback and model predictive control (MPC). Section IV discusses which invariant set to choose for each system to minimize the communication demand. Section V provides a numerical example to illustrate the advantages of the proposed methods. Finally, this paper is concluded and several future extensions are suggested in Section VI.

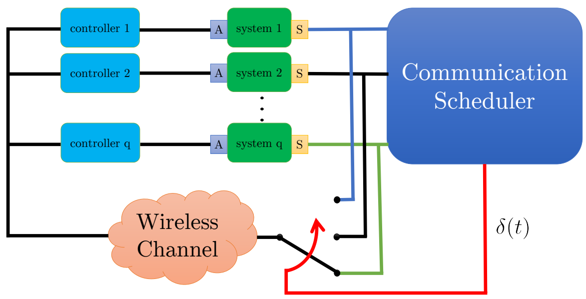

Fig. 1. Structure of the networked control system

II. Preliminaries

In this paper, we consider the multi-agent setting shown in Fig. 1, where the local control loops receive the sensor measurements through a shared wireless channel, with limited communication capacity. The central scheduler is in charge of scheduling the measurement updates for each local loop to guarantee that local state and input constraints are not violated. Here, we assume that such scheduling problem is solved with the tools proposed in [11], [12], which determine a measurement update schedule, based on the concept of safe time intervals $\alpha_1, \ldots, \alpha_q$, of the networked systems. For each system, α_i is the maximum number of time instants between two consecutive measurement updates, such that constraint satisfaction can be guaranteed. Computation of α_i is described in detail in Section II-A. Intuitively, this quantity encodes how unstable and perturbed a system is. Therefore, a schedule is feasible in this setting if and only if it ensures that every system i receives a measurement update at least once every α_i time instants. In case the local control loops require too frequent communication updates, i.e., α_i for $i = 1, \ldots, q$ are too small, the scheduling problem may be infeasible. In order to characterize feasibility of the scheduling problem for a given set of the safe time intervals, it is convenient to introduce the following *density function*.

Definition 1 (Density Function [13])**.** The *density function* $\rho(\alpha_1, \ldots, \alpha_q)$ is defined as

$$\rho(\alpha_1, \ldots, \alpha_q) := \sum_{i=1}^{q} \frac{1}{\alpha_i}, \tag{1}$$

where α_i is the safe time interval for system i.

Lemma 1. (Scheduling feasibility [14]) Inequalities

$$\rho(\alpha) \leq m_c, \quad \rho(\alpha) \leq 0.5 m_c \tag{2}$$

are, respectively, a necessary and a sufficient condition for the existence of a feasible schedule for an instance of the scheduling problem where m_c is the number of systems that can communicate through the channel simultaneously.

The control policy has an impact on the schedulability. A control policy that results in greater safe time intervals for the systems, i.e., lower communication demands, increases the schedulability chance by lowering the density, see Lemma 1. Note that a low density is desirable for a number of reasons, including ease of schedule design and robustness against packet losses [12].

A. Computation of the Safe Time Interval

Since the considered NCS has dynamically decoupled systems, their communication demands are independent. Therefore, in the following we focus on a single linear, perturbed system subject to constraints

$$x(t+1) = Ax(t) + Bu(t) + Ev(t), \tag{3a}$$
$$x \in \mathcal{X}, \quad u \in \mathcal{U}, \quad v \in \mathcal{V}, \tag{3b}$$

with

$$\mathcal{X} := \{x \in \mathbb{R}^n : A_x x \leq b_x\}, \tag{4a}$$
$$\mathcal{U} := \{u \in \mathbb{R}^m : A_u u \leq b_u\}, \tag{4b}$$
$$\mathcal{V} := \{v \in \mathbb{R}^p : A_v v \leq b_v\}, \tag{4c}$$

where x, u, and v are the system's state, input, and disturbance, respectively. The *admissible sets* $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{V}$ define the state and the input constraints as well as the disturbance bounds. In (4), the matrices A_x, A_u, and A_v and the vectors b_x, b_u, and b_v are constant and used to define the admissible sets, which are assumed to be convex, compact, and contain the origin in their interiors. Furthermore, we assume that the pair (A, B) is controllable.

In order to formally define the safe time interval α, we first introduce the following definitions.

Definition 2 (Robust Positively Invariant Set)**.** Consider (3) with a given law $u(t) := \kappa(x(t))$. Set $\mathcal{S} \subseteq \mathcal{X}$ is called a robust positive invariant (RPI) set for system (3) if

$$x(t) \in \mathcal{S} \implies x(t+1) \in \mathcal{S},\ \kappa(x(t)) \in \mathcal{U},\ \forall v(t) \in \mathcal{V}. \tag{5}$$

Definition 3 (Maximal RPI Set)**.** The maximal robust positively invariant (MRPI) set $\mathcal{S}_\infty$ is an RPI set which satisfies

$$\mathcal{S} \subseteq \mathcal{S}_\infty,\ \forall \mathcal{S}, \tag{6}$$

where $\mathcal{S}$ is an RPI set for the system.

Definition 4 (Robust Control Invariant Set)**.** Set $\mathcal{C} \subseteq \mathcal{X}$ is called a robust control invariant (RCI) set for system (3) if

$$x(t) \in \mathcal{C} \implies \exists u \in \mathcal{U} \text{ s.t. } x(t+1) \in \mathcal{C},\ \forall v(t) \in \mathcal{V}. \tag{7}$$

Definition 5 (Maximal RCI set)**.** The maximal robust control invariant (MRCI) set $\mathcal{C}_\infty$ is an RCI set which satisfies

$$\mathcal{C} \subseteq \mathcal{C}_\infty,\ \forall \mathcal{C}, \tag{8}$$

where $\mathcal{C}$ is an RCI set for the system.

Conditions guaranteeing that invariant sets exist system (3)-(4) have been studied in viability theory and reachability analysis. An extensive literature review can be found in [15] and references therein. Note that in general the described system may not have any invariant set. We refer to [15] for the necessary and sufficient conditions for the existence of the robust positively/control invariant set for the system.

The robust invariance of either $\mathcal{C}$ or $\mathcal{S}$ has been defined assuming that the actual states are always accessible by the controller. In our case, the states are measured at $t = 0$, while they are predicted for $t > 0$ according to

$$\hat{x}(t) = \begin{cases} x(t), & t = 0, \\ A\hat{x}(t-1) + Bu(t-1), & t > 0. \end{cases} \tag{9}$$

In this case neither the robust control invariance nor the robust positive invariance hold since only the predicted state is accessible. Hence, in order to analyze the evolution of (3) w.r.t. to admissible states and inputs set, we use reachability analysis. Define the function F as

$$\begin{aligned} x(t) &= A^t x(0) + \sum_{i=0}^{t-1} A^{t-i-1} \left(Bu(i) + Ev(i) \right) \\ &=: F(t, x(0), \mathbf{u}, \mathbf{v}), \end{aligned} \tag{10}$$

where $\mathbf{u} := (u(0), u(1), \ldots, u(t-1))$ and $\mathbf{v} := (v(0), v(1), \ldots, v(t-1))$. In case the control policy is given, we substitute $u(i) = \kappa(\hat{x}(i))$ in (10). Using F, the safe time interval α can be defined, which characterizes the evolution of the system within a specific set, under intermittent state measurements.

Definition 6 (Safe Time Interval)**.** The *safe time interval* α is defined as

$$\begin{aligned} \alpha := \max_t \{ t : \forall x(0) \in \mathcal{O},\ \exists u(0), \ldots, u(t-1) \in \mathcal{U} \text{ s.t.} \\ F(t, x(0), \mathbf{u}, \mathbf{v}) \in \mathcal{O},\ \forall v(0), \ldots, v(t-1) \in \mathcal{V} \}, \end{aligned} \tag{11}$$

where $\mathcal{O}$ is either $\mathcal{C}_\infty$ or $\mathcal{S}_\infty$, and

$$\bar{u}(i) := \begin{cases} \kappa(\hat{x}(i)) & \text{if } \mathcal{O} = \mathcal{S}_\infty, \\ u(i) & \text{if } \mathcal{O} = \mathcal{C}_\infty. \end{cases} \tag{12}$$

Note that the set $\mathcal{O}$ in Definition 6 can be any RPI set, in case the feedback policy is given, or any RCI set, in case the input function is not fixed. In the following section, we only consider the maximal invariant sets, i.e., $\mathcal{O} = \mathcal{S}_\infty$ or $\mathcal{O} = \mathcal{C}_\infty$. This decision is motivated in Section IV.

Based on Definition 6, α is the maximum number of consecutive time instants where the system stays in $\mathcal{O}$ when it starts therein and receives no additional state measurements. Therefore, a schedule that guarantees each system i receives a measurement update at least once during each α_i consecutive time instants, guarantees preservation of invariance for all network's systems by construction. Preservation of the invariance for all systems guarantees recursive satisfaction of the constraints [11].

III. Schedule-Aware Controller Design

In this section, we provide our main results. In the first subsection, we formulate an optimization problem to find an optimal static state feedback that maximizes the safe time interval for system (3). Unfortunately, this leads to an offline optimization problem that is very hard to solve. In the second subsection, maximization of the safe time interval is formulated w.r.t. $u(t), u(t+1), \ldots$, as in MPC.

A. Linear State Feedback

In this subsection, we assume $u(t) = -K\hat{x}(t)$ and maximize the safe time interval α w.r.t. K. To that end, we first address the computation of the MRPI set $\mathcal{S}_\infty(K)$ and define $\mathcal{O} := \mathcal{S}_\infty(K)$ in (11). Then, we formulate an optimization problem to maximize $\alpha(K)$ w.r.t. K. Note that in this case, the MRPI set is used since the controller structure is fixed.

Given K, one can define the admissible set $\mathcal{A}$ for system (3) as

$$\mathcal{A} := \{ x \in \mathbb{R}^n : Hx \le g \},\ H := \begin{bmatrix} A_x \\ -A_u K \end{bmatrix},\ g := \begin{bmatrix} b_x \\ b_u \end{bmatrix}. \tag{13}$$

The set $\mathcal{S}_\infty(K)$ can be computed as in [16], i.e.,

$$\mathcal{S}_\infty(K) = \left\{ x : HA_{\mathrm{c}}^k x \le g_k,\ 0 \le k \le n^* \right\}, \tag{14}$$

where $A_{\mathrm{c}} := A - BK$, $g_0 := g$ and

$$g_k := g - \max_v \left(H \sum_{j=1}^{k} A_{\mathrm{c}}^{j-1} Ev(j) \right) \text{ s.t. } v(j) \in \mathcal{V}, \tag{15}$$

for any $k > 0$, where the maximization is done componentwise and n^* is a positive integer such that

$$\mathcal{S}_\infty(K) \subseteq \{ x :\ HA_{\mathrm{c}}^n x \le g_n \},\ \forall n \ge n^*. \tag{16}$$

Since n^* is not known *a priori*, one can use a large positive number instead to make sure (16) holds for each K. Also note that (15) is a parametric optimization problem since K is unknown.

One can find the maximum safe time interval by solving

$$\max_K \ \alpha \tag{17a}$$

$$\text{s.t. } \mathcal{S}_\infty(K) \subseteq \bigcap_{j=1}^{\alpha} \mathcal{S}_j(K), \tag{17b}$$

where $\alpha \in \mathbb{N}$, $K \in \mathbb{R}^{m \times n}$, and

$$\mathcal{S}_j(K) := \{ x :\ HA_{\mathrm{c}}^{k+j} x \le g_k - \tilde{g}_{k,j},\ 0 \le k \le n^* \}, \tag{18}$$

$$\tilde{g}_{k,j} = \max_v \left(HA_{\mathrm{c}}^k \sum_{i=0}^{j-1} A^i Ev(i) \right) \text{ s.t. } v(i) \in \mathcal{V}. \tag{19}$$

Similar to (15), (19) consists of an elementwise parametric maximization since K is unknown.

Formulation (17) describes a mixed-integer optimization problem with nonlinear inequalities which also includes parametric optimization for computation of (15) and (19). These make this optimization problem very hard to solve.

Remark 1. While (17) is very hard to solve, the numerical computation of $\mathcal{S}_\infty$ and α for a given K is rather simple. Therefore, it is possible to use an evolutionary algorithm, such as Genetic Algorithm, to find the optimal K by evaluating the cost function for different K and evolving toward the optimal solution.

B. Unstructured Controller

In this subsection, we drop the constant feedback gain considered in Subsection III-A and maximize the safe time interval for system (3) w.r.t. $(u(t), u(t+1), \ldots)$. To that end, we set $\mathcal{O} = \mathcal{C}_\infty$ since the control law is not given *a priori*. In this case, we compute the maximum achievable α before designing the control law. Then, given α, we design an MPC law for the system such that the closed loop system has this maximum safe time interval.

The safe time interval α, defined in (11), depends on either $\mathcal{S}_\infty$ or $\mathcal{C}_\infty$. While $\mathcal{S}_\infty$ depends on the control law, $\mathcal{C}_\infty$ only depends on the control input constraints, see [16]. Therefore, in case $\mathcal{O} := \mathcal{C}_\infty$, one can find the maximum achievable α before designing the feedback policy, see Algorithm 1, where we use the Minkowski sum $\oplus$ and difference $\ominus$, defined as

$$P \oplus Q := \{p + q : \ p \in P, \ q \in Q\}, \tag{20}$$

$$P \ominus Q := \{z : \ z \oplus Q \subseteq P\}. \tag{21}$$

Algorithm 1 Maximum achievable α for $\mathcal{O} = \mathcal{C}_\infty$

$\mathcal{X}_0 \leftarrow \mathcal{C}_\infty$ and $k = 0$
while $\mathcal{X}_k = \mathcal{C}_\infty$ **do**
  $k = k + 1$
  $\mathcal{X}_k = \left\{x \in \mathcal{C}_\infty : \exists u(0), \ldots, u(k-1) \in \mathcal{U} \text{ s.t. } A^k x + \sum_{i=0}^{k-1} A^{k-1-i} B u(i) \oplus \left(\bigoplus_{i=0}^{k-1} A^i E \mathcal{V}\right) \in \mathcal{C}_\infty\right\}$
end while
$\alpha = k - 1$
return α

Lemma 2. Assume $\mathcal{O} := \mathcal{C}_\infty$ in (11). Then, Algorithm 1 returns the maximum achievable α.

Proof. Assume $\tilde{\alpha}$ is the solution of (11), i.e., the maximum achievable safe time interval for set $\mathcal{C}_\infty$. Using Definition 6, one can argue that $\mathcal{X}_k = \mathcal{C}_\infty$ holds for all $k \leq \tilde{\alpha}$ in Algorithm 1. As a result, the value returned by the algorithm satisfies $\alpha \geq \tilde{\alpha}$ by construction. However, if the algorithm would return $\alpha > \tilde{\alpha}$, one could conclude

$$\forall x(0) \in \mathcal{C}_\infty, \ \exists u(0), \ldots, u(t-1) \in \mathcal{U} \ \textit{s.t.}$$
$$F(x(0), u, v) \in \mathcal{O}, \ \forall v(0), \ldots, v(t-1) \in \mathcal{V},$$

which contradicts the assumption that $\tilde{\alpha}$ is the maximum achievable safe time interval. □

Note that Algorithm 1 only returns the maximum achievable α without specifying any corresponding feedback policy. In order to design a feedback policy that results in the maximum safe time interval, consider

$$\min_{\bar{x}, u} \sum_{k=0}^{\alpha-1} \left(\bar{x}_k^\top Q \bar{x}_k + u_k^\top R u_k\right) + \bar{x}_\alpha^\top P_f \bar{x}_\alpha \tag{22a}$$

$$\text{s.t. } \bar{x}_0 = x_0 \in \mathcal{C}_\infty, \tag{22b}$$

$$\bar{x}_{k+1} = A x_k + B u_k, \tag{22c}$$

$$u_k \in \mathcal{U}, \tag{22d}$$

$$\bar{x}_\alpha \in \bar{\mathcal{X}}_f, \tag{22e}$$

where Q, R, and P_f are positive definite matrices with appropriate sizes and $\bar{\mathcal{X}}_f := \mathcal{C}_\infty \ominus \left(\bigoplus_{i=0}^{\alpha-1} A^i E \mathcal{V}\right)$.

Remark 2. The optimization problem (22) is an MPC with restricted constraints, as in [17], i.e., $\bar{\mathcal{X}}_f$, is tightened for the nominal state $\bar{x}_\alpha$. This strategy guarantees robust constraint satisfaction for the actual states, i.e., $x(\alpha) \in \mathcal{C}_\infty$.

Lemma 3. Assume that $x(0)$ is measured and α is the maximum achievable safe time interval. In addition, assume that $x(t)$ is measured at least once every α consecutive time instants. Then, Algorithm 2 returns a control policy which guarantees robust satisfaction of $x(t) \in \mathcal{C}_\infty$.

Proof. Since $x(0)$ is measured and α is returned by Algorithm 1, optimization problem (22) is feasible initially by construction. This implies $\bar{x}(\alpha) \in \bar{\mathcal{X}}_f$ and consequently $x(\alpha) \in \mathcal{C}_\infty$ when $u_0^*, \ldots, u_{\alpha-1}^*$ is applied to the system in open loop. This implies that $x(t) \in \mathcal{C}_\infty$ also holds for all $1 \leq t \leq \alpha - 1$. Indeed, if $x(t) \notin \mathcal{C}_\infty$, then $\nexists\, u(t) \in \mathcal{U}, \ldots, u(\alpha-1) \in \mathcal{U}$ such that $x(\alpha) \in \mathcal{C}_\infty$, $\forall v(t) \in \mathcal{V}, \ldots, v(\alpha-1) \in \mathcal{V}$. □

Algorithm 2 MPC policy implementation

if $x(t)$ is measured **then**
  $x_0 = x(t)$ and solve (22)
  $u(t) \leftarrow u_0^*, \ldots, u(t+\alpha-1) \leftarrow u_{\alpha-1}^*$
end if
apply $u(t)$ to the system

IV. Relationship of α with Invariant Sets

In the previous section, we assumed that using either $\mathcal{O} = \mathcal{S}_\infty$ or $\mathcal{O} = \mathcal{C}_\infty$ is the most appropriate choice for computing α. Unfortunately, though this seems reasonable, it is not easy to prove that this maximizes α. In this section, we first provide a conservative upper bound on α which can be used to empirically assess the quality of the safe time interval computed using $\mathcal{S}_\infty$ or $\mathcal{C}_\infty$. Afterwards, we prove that in some specific cases, the use of larger invariant sets does not decrease the safe time interval.

A. Safe time interval upper bound

One can find a conservative upper bound to the safe time interval which is independent of the control law and the choice of the invariant set. We note that Algorithm 1 only returns the maximum α when $\mathcal{O} = \mathcal{C}_\infty$; while this algorithm returns the global maximum in some cases, we have no proof that this α is the global maximum regardless of the choice for the robust invariant set $\mathcal{O}$. The suggested upper bound for α may be used, for instance, to decide the minimum bandwidth needed for the network. This upper bound is defined in Lemma 4.

Lemma 4. Consider the safe time interval α as defined in (11) and assume that the admissible sets are symmetric w.r.t. the origin. Then, $\bar{\alpha} \geq \alpha$ holds where

$$\bar{\alpha} := \max_{\alpha} \left\{\alpha : \ \bigoplus_{i=0}^{\alpha-1} A^{\alpha-1-i} E \mathcal{V} \subseteq \mathcal{X}\right\}. \tag{23}$$

Proof. Using (4a) and (10), one can conclude that

$$A_x \left(A^{\alpha} x(0) + \sum_{i=0}^{\alpha-1} A^{\alpha-i-1} \left(Ev(i) + Bu(i) \right) \right) \leq b_x. \quad (24)$$

Equation (24) holds for any $x(0) \in \mathcal{O}$ where $\mathcal{O}$ is any RPI or RCI set for system (3). Although set $\mathcal{O}$ is not known *a priori*, it always contains the origin. Therefore, (24) holds for $x(0) = 0$, i.e.,

$$A_x \sum_{i=0}^{\alpha-1} A^{\alpha-i-1} Ev(i) \leq b_x, \quad (25)$$

where the control inputs are zero since the initial state is zero. In fact, in the linear state feedback case, $\hat{x}(t) = 0$ for all $t \geq 0$ which implies $u(t) = -K\hat{x}(t) = 0$. Similarly, $x(0) = 0$ in (22) results in $u(0) = \ldots = u(\alpha - 1) = 0$ as the optimal solution of the problem the symmetry assumption. □

B. Robust invariant set choice

In this subsection we justify the maximal invariant set choice, i.e., either $\mathcal{O} = \mathcal{S}_\infty$ or $\mathcal{O} = \mathcal{C}_\infty$, for maximization of α, which is defined in (17). This choice is made for the following reasons.

First, $\mathcal{O}$ is a set of initial states for which recursive constraints satisfaction can be guaranteed. Typically, a larger set for the initial states is an advantage and this set is the largest when it is the maximal invariant set, i.e., either $\mathcal{O} = \mathcal{S}_\infty$ or $\mathcal{O} = \mathcal{C}_\infty$. Second, we speculate that

$$\mathcal{O}_1 \subseteq \mathcal{O}_2 \implies \alpha(\mathcal{O}_1) \leq \alpha(\mathcal{O}_2), \quad (26)$$

where $\mathcal{O}_1$ and $\mathcal{O}_2$ are arbitrary robust invariant sets and $\alpha(\mathcal{O}_i)$ is the safe time interval α when $\mathcal{O} = \mathcal{O}_i$, see (11). This speculation is suggested based on the following remark and lemmas.

Remark 3 (System open-loop evolution)**.** Consider $\alpha(\mathcal{O})$ as defined in (11). When α is large enough, one can argue $A^\alpha x(0) + \sum_{i=0}^{\alpha-1} A^i Bu(i) \approx 0$ since the pair (A, B) is controllable and $x(0) \in \mathcal{O}$. This results in

$$\alpha(\mathcal{S}) \approx \max\{t : \bigoplus_{i=0}^{t-1} A^i E\mathcal{V} \subseteq \mathcal{S}\}, \quad (27)$$

which in turn implies $\alpha(\mathcal{O}_1) \leq \alpha(\mathcal{O}_2)$ if $\mathcal{O}_1 \subseteq \mathcal{O}_2$.

Using Remark 3, one can speculate that a larger invariant set results in a bigger safe time interval. Although this remark only holds for a large enough α, we next show that this speculation is valid at least in two special cases.

Lemma 5. Inequality $\alpha(\mathcal{O}_1) \leq \alpha(\mathcal{O}_2)$ holds if

$$\mathcal{O}_2 = \gamma\mathcal{O}_1, \ \mathcal{U}_2 = \gamma\mathcal{U}_1, \ \gamma \geq 1, \quad (28)$$

where $\mathcal{O}_i$ is a robust invariant set for the system and $\mathcal{U}_i$ is the admissible set for the input used in $\alpha(\mathcal{O}_i)$.

Proof. By definition,

$$x(0) \in \mathcal{O}_1 \implies \exists u(0), \ldots, u(\alpha_1) \in \mathcal{U}_1 \text{ s.t. } x(\alpha_1) \in \mathcal{O}_1, \quad (29)$$

holds for all $v(i) \in \mathcal{V}$ where $\alpha_1 = \alpha(\mathcal{O}_1)$ and $x(\alpha_1)$ is defined in (10). This entails

$$\gamma \left(A^{\alpha_1} x(0) + \sum_{i=0}^{\alpha_1-1} A^{t-i-1} \left(Bu(i) + Ev(i) \right) \right) \in \gamma\mathcal{O}_1, \quad (30)$$

which implies $\alpha(\mathcal{O}_1) \leq \alpha(\mathcal{O}_2)$ by definition. □

Lemma 6. Assume that $\Delta\mathcal{O}$ and $\Delta\mathcal{U}$ are compact sets which contain the origin in their interiors and

$$x \in \Delta\mathcal{O} \implies \exists u \in \Delta\mathcal{U} \text{ s.t. } Ax + Bu \in \Delta\mathcal{O}. \quad (31)$$

Then, inequality $\alpha(\mathcal{O}_1) \leq \alpha(\mathcal{O}_2)$ holds if

$$\mathcal{O}_2 = \mathcal{O}_1 \oplus \Delta\mathcal{O}, \ \mathcal{U}_2 = \mathcal{U}_1 \oplus \Delta\mathcal{U} \quad (32)$$

where $\mathcal{O}_i$ is a robust invariant set for the system and $\mathcal{U}_i$ is the admissible set for the input used in $\alpha(\mathcal{O}_i)$.

Proof. By definition,

$$x(0) \in \mathcal{O}_1 \implies \exists u(0), \ldots, u(\alpha_1) \in \mathcal{U}_1 \text{ s.t. } x(\alpha_1) \in \mathcal{O}_1, \quad (33)$$

holds for all $v(i) \in \mathcal{V}$ where $\alpha_1 = \alpha(\mathcal{O}_1)$ and $x(\alpha_1)$ is defined in (10). This implies

$$A^{\alpha_1} x(0) + \sum_{i=0}^{\alpha_1-1} A^{t-i-1} \left(Bu(i) + Ev(i) \right) \in \mathcal{O}_1. \quad (34)$$

Furthermore, (31) implies for any $\delta x(0) \in \Delta\mathcal{O}$, $\delta u(i) \in \Delta\mathcal{U}$ exists such that

$$A^{\alpha_1} \delta x(0) + \sum_{i=0}^{\alpha_1-1} A^{t-i-1} B\delta u(i) \in \Delta\mathcal{O}. \quad (35)$$

Equations (34) and (35) imply

$$A^{\alpha_1} \tilde{x}(0) + \sum_{i=0}^{\alpha_1-1} A^{t-i-1} \left(B\tilde{u}(i) + Ev(i) \right) \in \mathcal{O}_2, \quad (36)$$

where $\tilde{x}(0) = x(0) + \delta x(0)$ is an arbitrary point in $\mathcal{O}_2$ and $\tilde{u}(i) = u(i) + \delta u(i) \in \mathcal{U}_2$. Equation (36) implies $\alpha(\mathcal{O}_1) \leq \alpha(\mathcal{O}_2)$ by definition. □

Lemmas 5 and 6 provide cases in which the safe time interval is bigger when the robust invariant set is larger.

V. Numerical Results

In this section, we provide an example to compare the safe time interval for different control strategies. We consider a model of the longitudinal dynamics of a vehicle as a discrete-time system (3) with the following parameters:

$$A = \begin{bmatrix} 1 & 0.2 & 0.02 \\ 0 & 1 & 0.2 \\ 0 & 0 & 0 \end{bmatrix}, \ B = E = \begin{bmatrix} 0.0013 \\ 0.02 \\ 0.2 \end{bmatrix}. \quad (37)$$

The scalar input is the vehicle's jerk, and both states and inputs are constrained for avoiding collisions and actuator saturation, so that the state and input admissible sets are

$$\mathcal{X} = \left\{ [x_1 \ x_2 \ x_3]^\top : |x_1| \leq 1, \ |x_2| \leq 3, \ |x_3| \leq 4 \right\}. \quad (38)$$

and $\mathcal{U} = \{u : \ |u| \leq 1\}$, respectively. The input is subject to an additive noise whose admissible set is described by

$\mathcal{V}(\bar{v}) = \{v : |v| \leq \bar{v}\}$, which we vary in the simulation for comparison. This for instance can describe an intersection scenario in which a set of such vehicles must be simultaneously controlled to track received trajectories from a coordinator in order to safely and efficiently cross the intersection.

We compare the following control laws.

- *LQR*: The controller gain K is chosen as the solution of the LQR problem with cost gains $Q = \text{diag}([1, 1, 1]),\ R = 1$.
- *Optimal gain*: The controller gain K is chosen as the solution of the optimization problem (17).
- *MPC*: The controller is calculated by solving an MPC problem as in Algorithm 1.

The safe time intervals for these cases and their upper bounds, calculated according to Lemma 4, are presented in Table I.

TABLE I
COMPARISON OF SAFE TIME INTERVALS FOR DIFFERENT CONTROL LAWS

$\bar{v}$	0.9	0.7	0.3	0.1
LQR	0	0	2	3
Optimal gain	1	5	7	27
MPC	4	8	19	38
$\bar{\alpha}$	14	16	24	41

For the LQR controller, the safe time intervals are zero for the first two instances of the problem, i.e., the RPI sets do not exist. Table I indicates that one can increase the safe time interval significantly compared to a case in which the controller gain is not designed to maximize the safe time interval. The table also shows that the safe time intervals using optimal MPC are better than those using optimal static state feedback. The invariant sets for these three controllers in case of $\bar{v} = 0.3$ are displayed in Fig. 2 together with their projections. As expected, maximal invariant set for the MPC case is the largest set. Note that the maximal invariant set in the optimal gain case is larger than the one in the LQR case; this observation is consistent with the conjecture (26) since the safe time interval for the optimal gain case is greater than the one in the LQR case.

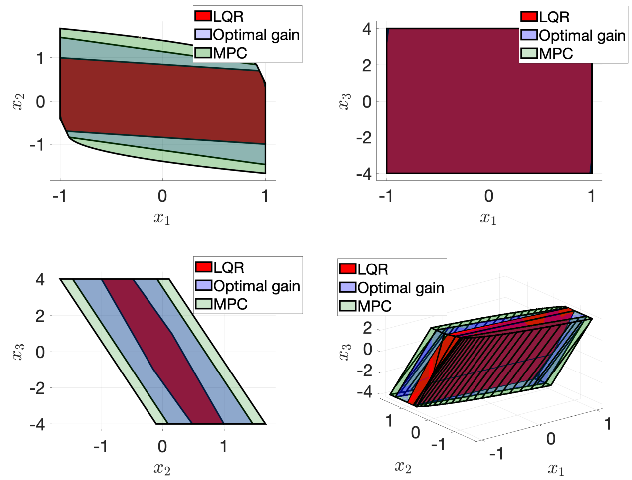

Fig. 2. Comparison of the maximal invariant sets for different control laws

VI. CONCLUSIONS

In this paper, we formulated optimal control laws that minimize the communication demand for perturbed constrained linear systems in a networked control system. We showed that this optimization problem is very hard when the feedback gain is constant. However, this optimization problem becomes a simple quadratic programming and the system has the lowest achievable communication demand when MPC is used.

As future studies, we will consider an additional network between controllers and actuators in the NCS.

REFERENCES

[1] X.-M. Zhang, Q.-L. Han, and X. Yu, "Survey on recent advances in networked control systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1740–1752, 2015.

[2] X. Liu and A. Goldsmith, "Wireless medium access control in networked control systems," in *Proceedings of the 2004 American Control Conference*, vol. 4. IEEE, 2004, pp. 3605–3610.

[3] M. Zanon, T. Charalambous, H. Wymeersch, and P. Falcone, "Optimal scheduling of downlink communication for a multi-agent system with a central observation post," *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 37–42, 2017.

[4] M. Xia, V. Gupta, and P. J. Antsaklis, "Networked state estimation over a shared communication medium," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1729–1741, 2016.

[5] M. S. Branicky, S. M. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 2. IEEE, 2002, pp. 1211–1217.

[6] L. Zhang and D. Hristu-Varsakelis, "Communication and control co-design for networked control systems," *Automatica*, vol. 42, no. 6, pp. 953–958, 2006.

[7] M. M. B. Gaid, A. Cela, and Y. Hamam, "Optimal integrated control and scheduling of networked control systems with communication constraints: Application to a car suspension system," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 776–787, 2006.

[8] F. D. Brunner, M. Heemels, and F. Allgöwer, "Robust self-triggered mpc for constrained linear systems: A tube-based approach," *Automatica*, vol. 72, pp. 73–83, 2016.

[9] F. D. Brunner, M. Lazar, and F. Allgöwer, "Stabilizing model predictive control: on the enlargement of the terminal set," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 15, pp. 2646–2670, 2015.

[10] F. D. Brunner, T. Gommans, W. Heemels, and F. Allgöwer, "Communication scheduling in robust self-triggered mpc for linear discrete-time systems," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 132–137, 2015.

[11] A. Colombo, M. Bahraini, and P. Falcone, "Measurement scheduling for control invariance in networked control systems," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3361–3366.

[12] M. Bahraini, M. Zanon, A. Colombo, and P. Falcone, "Receding-horizon robust online communication scheduling for constrained networked control systems," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2969–2974.

[13] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel, "The pinwheel: A real-time scheduling problem," in *Proceedings of the 22nd Hawaii International Conference of System Science*, 1989, pp. 693–702.

[14] A. Bar-Noy and R. E. Ladner, "Windows scheduling problems for broadcast systems," *SIAM Journal on Computing*, vol. 32, no. 4, pp. 1091–1113, 2003.

[15] S. Rakovic, "Robust control of constrained discrete time systems: Characterization and implementation," Ph.D. dissertation, University of London, 2005.

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

[17] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: predictive control with restricted constraints," *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.