Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-13 (2022) 25–30

Scheduling of over-actuated networked control systems

J. Kreiss * R. Postoyan * D. Nešić ** W.P.M.H. Heemels ***

* *Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France.*
** *Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, VIC 3010, Australia.*
*** *Control Systems Technology Section, Department of Mechanical Engineering, Eindhoven University of Technology, Netherlands.*

Abstract: We investigate the scenario where an over-actuated plant is controlled over a network. We concentrate on the effect of two network-induced phenomena: varying transmission intervals and scheduling, in the sense that only one of the actuators receives new data at each transmission instant. We present an emulation-based solution for the controller design, i.e., the controller is first designed to stabilize the origin of the plant ignoring the network and second, the packet-based network is taken into account and conditions on the maximum allowable transmission interval (MATI) and the scheduling protocol are given to preserve the stability of the closed-loop system. Our results are tailored to over-actuated plants leading to significant improvements compared to applying off-the-shelf results available in the literature. In particular, a new model is derived and new conditions on the scheduling protocol are given, which lead to a two-measure stability property for the networked control system. We illustrate how new classes of scheduling protocols can be derived by exploiting over-actuation, which leads to larger MATI bounds compared to applying existing results, as shown on a numerical example.

Keywords: Networked control systems, over-actuated systems, hybrid systems, stability of hybrid systems, Lyapunov methods.

1. INTRODUCTION

Communication networks are increasingly used in control applications to connect a plant with its sensors control unit, in which case we talk of *networked control systems* (NCS). NCS offer many advantages over classical dedicated point-to-point connections in terms of ease of implementation and maintenance, lower cost etc. The use of wireless networks is even unavoidable in some applications, such as cooperative driving, see, e.g., (Dolk et al., 2017). However, networks also inevitably generate communication imperfections including (aperiodic) transmissions, scheduling, quantization, delays and packet losses, which may seriously deteriorate the desired closed-loop behaviour. This motivates the development of control theoretical tools dedicated to NCS and various methods are now available in the literature, see, e.g., (Heemels et al., 2010; Hetel et al., 2017; Nešić and Teel, 2004; Park et al., 2017; Peters et al., 2016; Walsh et al., 2001).

In this work, we investigate over-actuated plants, which communicate with their controller over a packet-based digital network subject to aperiodic transmissions and scheduling, like power grids, see, e.g., (Kreiss et al., 2021; Singh et al., 2014). Over-actuation means that inputs are available in the system to make it resilient to failures, to enhance performance, to optimize control power or to deal with attacks see, e.g., (Cristofaro et al., 2018; Johansen and Fossen, 2013; Kreiss et al., 2021; Trip et al., 2019). As a result, we have more control inputs at our disposal than needed to achieve the control goal. Existing techniques for NCS such as those in e.g., (Carnevale et al., 2007; Donkers et al., 2011; Liu et al., 2012; Nešić and Teel, 2004), apply to over-actuated systems but do not exploit their special features and consequently impose stronger conditions than needed in particular on the scheduling protocol. Consider the round-robin (RR) protocol for instance, which grants access to the network to each actuator node in a cyclic pre-determined fashion. RR would visit all nodes one after the other even though this may not be needed as several actuators may have the same effect on the plant dynamics. This demonstrates that there is a need for networked control tools, which take into account the features of over-actuated systems. To concentrate on the challenges due to over-actuation, we consider a set-up where the controller is directly connected to the sensors and sends the control input to the plant actuators over a digital channel where each actuator is associated to a network node, see Fig. 1. As a result, only one of the actuators nodes receives new data at each transmission instant.

In this paper, we consider a general framework based on nonlinear plant models, which are over-actuated with respect to static controllers. We proceed by emulation in

[1] Work supported by HANDY project ANR-18-CE40-0010-02, the France-Australia collaboration project IRP-ARS CNRS, and the Australian Research Council under the Discovery Project DP200101303.

10.1016/j.ifacol.2022.07.230

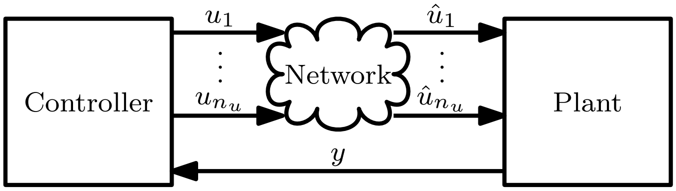

Fig. 1. The NCS set-up where the number of nodes n_u corresponds to the number of actuators.

the sense that we first design a stabilizing output-feedback controller for the closed-loop system in the absence of the network. At this stage, any relevant control design technique in the literature can be applied to synthesize the feedback law. The controller will be set-valued because of over-actuation, which is a first difference with respect to the related NCS literature. In the second design stage, we take into account the packet-based communication network and present a new hybrid model of the overall system, for which a jump corresponds to a transmission. Afterwards, we present the conditions we impose on the original closed-loop system as in (Carnevale et al., 2007), the scheduling rule and the *maximum allowable transmission interval* (MATI) based on which we ensure that the NCS satisfies a uniform global asymptotic stability property. Contrary to e.g., (Carnevale et al., 2007; Heemels et al., 2010; Hertneck and Allgöwer, 2020; Nešić and Teel, 2004), the protocol is required to satisfy a two-measure stability property (Cai et al., 2007; Teel and Praly, 2000) because of over-actuation, as opposed to standard uniform global exponential/asymptotic stability. This leads to a new two-measure stability property for the NCS.

While uniformly globally exponentially stable (UGES) protocols as considered in, e.g., (Carnevale et al., 2007; Donkers et al., 2011; Liu et al., 2012; Nešić and Teel, 2004), can be used to ensure the condition we impose on the protocol, this notion needs to be revisited and modified when dealing with over-actuated plants simply because all the actuator nodes do not have to be used. In fact, we show how over-actuation can be exploited to develop new, tailored scheduling protocols in this context. These new protocols exploit the redundancy of the inputs of the plant and can lead to larger MATI bounds as we show on a numerical example. We believe that these protocols are one of the main contributions of this work and that many more could be derived in the future. Note that the proofs are omitted for space reasons.

2. PROBLEM STATEMENT AND MODELLING

2.1 Plant and effective control

Consider the plant model

$$\dot{x}_p = f_p(x_p, Bu), \quad y = h_p(x_p), \tag{1}$$

where $x_p \in \mathbb{R}^{n_{x_p}}$ is the state, $u \in \mathbb{R}^{n_u}$ is the control input, $y \in \mathbb{R}^{n_y}$ is the output, $B \in \mathbb{R}^{n_B \times n_u}$ is a real, non-zero matrix, and $n_{x_p}, n_u, n_y, n_B \in \mathbb{Z}_{>0}$. We introduce the *effective* control input $v := Bu \in \mathbb{R}^{n_v}$ as in (Johansen and Fossen, 2013), where $n_v := n_B$. In that way, (1) becomes

$$\dot{x}_p = f_p(x_p, v), \quad y = h_p(x_p). \tag{2}$$

We proceed by emulation and we thus first design the controller ignoring the network. We assume for this purpose that we know an effective static output-feedback controller in the absence of network of the form

$$v = g_{\text{eff}}(y), \tag{3}$$

which robustly stabilizes the origin of system (2) in a sense made precise in the sequel. Note that we consider only static output controllers to concentrate on the issues stemming from over-actuation. Appropriate generalisations to dynamic controllers are left for future work.

2.2 Network-free over-actuated plant

System (1) is assumed to be over-actuated in the sense that it has redundant inputs, which is formalized by the next assumption, see also (Harkegard and Glad, 2005; Zaccarian, 2009).

Assumption 1. $\dim \operatorname{Ker}\{B\} > 0$. □

Remark 2. Systems satisfying Assumption 1 are usually referred as static over-actuated systems, see e.g. (Zaccarian, 2009). □

Given (3), we can then derive a control allocation policy to specify the value of u in (1) such that $Bu = g_{\text{eff}}(y)$ holds, see, e.g., (Bodson, 2002; Johansen and Fossen, 2013; Oppenheimer et al., 2006). Hence, the control input u applied to system (6) satisfies

$$u \in G_c(y) := \{w \in \mathbb{R}^{n_u} : Bw = g_{\text{eff}}(y)\}. \tag{4}$$

Note that G_c is indeed a set-valued map in view of Assumption 1.

2.3 Network set-up

We investigate the scenario where the controller is implemented over a network. To concentrate on the issues related to over-actuation, we focus on the case where only control input signals are communicated over the network and assume for simplicity, that the sensors are directly connected to the controller, see Fig. 1. Moreover, each component u_i, $i \in \{1, \ldots, n_u\}$, of u is associated to a given node. We therefore have n_u (actuator) nodes.

Transmissions occur at times t_i, $i \in \mathbb{Z}_{\geqslant 0}$, which verify, for all $i \in \mathbb{Z}_{\geqslant 0}$,

$$t_{i+1} - t_i \in [\varepsilon, T], \tag{5}$$

where $0 < \varepsilon \leqslant T$, ε being the *minimum allowable inter-transmission interval* and T the *maximum allowable inter-transmission interval* (MATI). At each t_i, $i \in \mathbb{Z}_{\geqslant 0}$, a single actuator node receives a packet sent by the controller. As a result, system (1) can be written as

$$\dot{x}_p = f_p(x_p, B\hat{u}), \quad y = h_p(x_p), \tag{6}$$

between two successive transmission instants, where $\hat{u} := (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_{n_u})$ is the network-induced input. At each transmission instant t_i, for $i \in \mathbb{Z}_{\geqslant 0}$, a single component $\hat{u}_j$, $j \in \{1, \ldots, n_u\}$, of $\hat{u}$ is updated using the newly received data, i.e.,

$$\hat{u}_j(t_i^+) = h_{j,\text{rec}}(u_j(t_i), \hat{u}_j(t_i), \eta(t_i)), \tag{7}$$

where $\eta \in \mathcal{S} \subseteq \mathbb{R}^{n_\eta}$ with $n_\eta \in \mathbb{Z}_{\geqslant 0}$, is a vector of variables, which may be needed to implement the scheduling protocol. Examples of variables η include transmission counters and toggle variables, see, for instance, Section 4 and (Nešić and Teel, 2004, Remark 2). We write the dynamics of the η-system using the general form

$$\dot{\eta} = f_\eta(x_p, \hat{u}, \eta), \tag{8}$$

between two successive transmission instants, which is assumed to be forward complete (Angeli and Sontag, 1999), and, at each transmission instant t_i, $i \in \mathbb{Z}_{\geqslant 0}$,

$$\eta(t_i^+) = g_\eta(\eta(t_i)). \tag{9}$$

Examples of systems (8)-(9) will be provided in Section 4.

We do not necessarily enforce $\hat{u}_j(t_i^+)$ to be equal to $u_j(t_i)$ in (7), we use instead a general term $h_{j,\text{rec}}(u_j(t_i), \hat{u}_j(t_i), \eta(t_i))$. We plan to show in future work how this degree of flexibility can be exploited for over-actuated plants. Note that the index "rec" in $h_{j,\text{rec}}$ stands for "received".

Because u_j depends on y through the set-valued map G_c, see (4), and since we are dealing with an output-feedback controller, see (3), which can be view as a function of x_p in view of (1), we can rewrite (7) as

$$\hat{u}_j(t_i^+) \in H_{j,\text{rec}}(x_p(t_i), \hat{u}_j(t_i), \eta(t_i)). \tag{10}$$

On the other hand, at time t_i, the other $n_u - 1$ nodes do not receive any new data, therefore, for $k \in \{1, \ldots, n_u\} \backslash \{j\}$,

$$\hat{u}_k(t_i^+) \in H_{k,\text{hold}}(x_p(t_i), \hat{u}_k(t_i), \eta(t_i)). \tag{11}$$

Several options can be envisioned for set-valued map $H_{k,\text{hold}}$ among which $H_{k,\text{hold}}(x_p(t_i), \hat{u}_k(t_i), \eta(t_i)) = \{\hat{u}_k(t_i)\}$ meaning that $\hat{u}_k$ remains constant when u_k is not transmitted over the network, or $H_{k,\text{hold}}(x_p(t_i), \hat{u}_k(t_i), \eta(t_i)) = \{0\}$ corresponding to a zeroing strategy in, e.s., (Schenato, 2009). Note that we consider a set-valued map in (11) for the sake of generality and to be consistent with (10).

Between two successive transmissions, we use a general holding function $\hat{f}_u$ to generate $\hat{u}$, i.e., for any $i \in \mathbb{Z}_{\geqslant 0}$ and $t \in (t_i, t_{i+1})$,

$$\dot{\hat{u}} = \hat{f}_u(x_p, \hat{u}, \eta). \tag{12}$$

Zero-order-hold devices correspond to $\hat{f}_u = 0$, but other strategies can be implemented as long as these lead to model (12) and that the latter is forward complete.

2.4 Hybrid model

We are almost ready to present the overall model of the NCS. Before that, we introduce the clock variable τ to measure the time since the last transmission as in e.g., (Carnevale et al., 2007), whose dynamics is

$$\begin{aligned} \dot{\tau} &= 1 \quad \text{when } \tau \in [0, T] \\ \tau^+ &= 0 \quad \text{when } \tau \in [\varepsilon, T]. \end{aligned} \tag{13}$$

As a result, the overall system is given by

$$\left.\begin{aligned} \dot{x}_p &= f_p(x_p, B\hat{u}) \\ \dot{\hat{u}} &= \hat{f}_u(x_p, \hat{u}, \eta) \\ \dot{\eta} &= f_\eta(x_p, \hat{u}, \eta) \\ \dot{\tau} &= 1 \end{aligned}\right\} \tau \in [0, T]$$
$$\left.\begin{aligned} x_p^+ &= x_p \\ \hat{u}^+ &\in (\mathbb{I} - \Delta(x_p, \hat{u}, \eta))\, H_{\text{rec}}(x_p, \hat{u}, \eta) \\ &\quad + \Delta(x_p, \hat{u}, \eta) H_{\text{hold}}(x_p, \hat{u}, \eta) \\ \eta^+ &= g_\eta(\eta) \\ \tau^+ &= 0 \end{aligned}\right\} \tau \in [\varepsilon, T], \tag{14}$$

where $\Delta(x_p, \hat{u}, \eta)$ is a $n_u \times n_u$ diagonal matrix, whose diagonal components are all 1 except one of them, which is zero, $H_{\text{rec}} := (H_{1,\text{rec}}, \ldots, H_{n_u,\text{rec}})$ and $H_{\text{hold}} := (H_{1,\text{hold}}, \ldots, H_{n_u,\text{hold}})$. The location of the 0 diagonal component of Δ depends on the scheduling protocol; this is explained in more detail in Section 4.

For the sake of convenience, we define $q := (x_p, \hat{u}, \eta, \tau) \in \mathcal{Q} := \mathbb{R}^{n_{x_p}} \times \mathbb{R}^{n_u} \times \mathcal{S} \times [0, T]$ and write system (14) as

$$\begin{aligned} \dot{q} &= f(q) \quad \text{when } \tau \in [0, T] \\ q^+ &\in G(q) \quad \text{when } \tau \in [\varepsilon, T], \end{aligned} \tag{15}$$

where the expressions of f and G follow from (14).

2.5 Objective

Our main objective is to give conditions on the original closed-loop system (1)-(4), the scheduling protocol and the MATI T in (5) to ensure a global asymptotic stability property for system (14). We specifically aim at exploiting the fact that system (1) is over-actuated to propose tailored scheduling rules, leading to possibly improved guarantees compared to directly applying the existing results (Carnevale et al., 2007; Heemels et al., 2010).

3. STABILITY RESULTS

To proceed with the stability analysis of (14), we follow a similar approach as in (Carnevale et al., 2007; Nešić and Teel, 2004) and we interpret system (14) as the feedback interconnection of the physical system, whose variable is x_p, and the network-induced system, whose associated variables are $(\hat{u}, \eta, \tau)$. In the related literature, it is customary to introduce the network-induced error on the control input u, i.e., $e_u := \hat{u} - u$ with $u \in G_c(y)$ in our case, to model the perturbative term due to the network, which impacts the physical system. Since plant (1) is over-actuated, another relevant quantity to consider is the network-induced error on the effective control input v, i.e., $e_v := Be_u$. This is an important difference with the existing works developed for general plant models ignoring over-actuation. Indeed, because e_v is the actual network-induced error affecting the closed-loop dynamics, the stability property established below in Theorem 4 differs from existing ones. Moreover, this allows to envision new protocols, as will be explained in Section 4. Now that we have defined e_u, we can state the next assumption.

Assumption 3. There exist $\underline{\alpha}_V, \overline{\alpha}_V, \alpha_V \in \mathcal{K}_\infty$, $V : \mathbb{R}^{n_{x_p}} \to \mathbb{R}_{\geqslant 0}$ continuously differentiable, $W : \mathcal{Q} \to \mathbb{R}_{\geqslant 0}$ locally Lipschitz [2], $H \in \mathbb{R}^{n_{x_p}} \to \mathbb{R}$ continuous, $0 < \underline{a}_W \leqslant \overline{a}_W$, $L, \gamma \geqslant 0$ and $\rho \in)0, 1)$ such that the following hold.

(i-a) For any $x_p \in \mathbb{R}^{n_{x_p}}$, $\underline{\alpha}_V(|x_p|) \leqslant V(x_p) \leqslant \overline{\alpha}_V(|x_p|)$.
(i-b) For any $q \in \mathcal{Q}$, $\langle \nabla V(x_p), f_p(x_p, B\hat{u}) \rangle \leqslant -\alpha_V(|x_p|) - H^2(x_p) + \gamma^2 W^2(q)$.
(ii-a) For any $q \in \mathcal{Q}$ and $u \in G_c(y)$, $\underline{a}_W |Be_u| \leqslant W(q) \leqslant \overline{a}_W |e_u|$.
(ii-b) For almost all $q \in \mathcal{Q}$, $\langle \nabla W(q), f(q) \rangle \leqslant LW(q) + H(x_p)$.
(ii-c) For any $q \in \mathcal{Q}$ with $\tau \in [\varepsilon, T]$ and $g \in G(q)$, $W(g) \leqslant \rho W(q)$. □

Item (i) of Assumption 3 implies that the x_p-system is $\mathcal{L}_2$-stable from $W(q)$ to $H(x)$, like in, e.g., (Carnevale et al., 2007; Heemels et al., 2010; Hertneck and Allgöwer, 2020; Nešić and Teel, 2004). This property is typically ensured at the first step of emulation when designing the effective

[2] When η involves variables defined on (a subset of) $\mathbb{Z}_{\geqslant 0}$, which are constant during flows, like a transmission counter or a toggle variable for instance, Lipschitz properties of W with respect to these variables are (obviously) not required.

control (4). It is, for instance, verified when plant (1) and controller (3) are linear and the origin of the corresponding closed-loop system is uniformly globally exponentially stable (UGES) for which the results are new as far as we know. On the other hand, item (ii) of Assumption 3 is related to the scheduling protocol. In particular, items (ii-a) and (ii-c) imply that the scheduling protocol satisfies a UGES property, which differs from those considered in, e.g., (Carnevale et al., 2007; Heemels et al., 2010; Hertneck and Allgöwer, 2020; Nešić and Teel, 2004), because the lower bound in item (ii-a) of Assumption 3 involves $|Be_u|$ and not $|e_u|$. This is due to the over-actuation of plant (1). We exploit this fact in Section 4 to propose novel scheduling protocols tailored to this class of systems in the sequel. On the other hand, item (ii-b) is verified when the gradient of W is bounded (almost everywhere) by a constant and f satisfies a linear growth condition, e.g. (Nešić and Teel, 2004, Section 5).

The next theorem ensures a uniform global asymptotic stability property provided Assumptions 1-3 hold and the MATI T is suitably selected.

Theorem 4. Consider system (15) and suppose the following hold.

(i) Assumptions 1-3 are verified.
(ii) $T < \mathcal{T}(\gamma, L, \rho)$ and $\varepsilon \in (0, T]$ where

$$\mathcal{T}(\gamma, L, \rho) := \begin{cases} \frac{1}{Lr}\arctan\left(\frac{r(1-\rho)}{2\frac{\rho}{1+\rho}\left(\frac{\gamma}{L}-1\right)+1+\rho}\right) & \gamma > L \\ \frac{1}{Lr}\text{arctanh}\left(\frac{r(1-\rho)}{2\frac{\rho}{1+\rho}\left(\frac{\gamma}{L}-1\right)+1+\rho}\right) & \gamma < L \\ \frac{1}{L}\frac{1-\rho}{1+\rho} & \gamma = L \end{cases}$$

with $r := \sqrt{\left|\left(\frac{\gamma}{L}\right)^2 - 1\right|}$.

There exists $\beta \in \mathcal{KL}$ such that for any solution q and $e_u(0,0) = \hat{u}(0,0) - u_0$ with $u_0 \in G_c(y(0,0))$,

$$|(x_p(t,j), Be_u(t,j))| \leqslant \beta(|(x_p(0,0), e_u(0,0))|, t+j) \quad (16)$$

for all $(t, j) \in \operatorname{dom} q$ and, if q is maximal, it is complete. □

The main difference between Theorem 4 and the related results of the literature is that the e_u-component of the solutions to (14) is not guaranteed to converge to 0 as time grows. Indeed, it may be the case that the scheduling protocol ignores certain nodes, while still ensuring a desirable property for the x_p-system, as exemplified in the next section. As a result, we guarantee instead a uniform global asymptotic stability with respect to two measures (Cai et al., 2007; Teel and Praly, 2000).

4. SCHEDULING PROTOCOLS

While standard UGES protocols such as RR, which, again, grants access to the network to each node in a cyclic pre-determined fashion, or maximum-error-first try-once-discard (TOD), which selects the nodes with the largest network-induced error, ensure the satisfaction of items (ii-a) and (ii-c) of Assumption 3, these may not be the best options for over-actuated plant (1). Indeed, redundant control signals may then be communicated over the network, which are not relevant for control. In this section, we provide examples of scheduling protocols that exploit over-actuation to avoid this possible shortcoming of these off-the-shelf protocols while ensuring the satisfaction of items (ii-a) and (ii-c) of Assumption 3. We emphasize that only a sample of possible new protocols are presented here.

4.1 UGES protocol over a selection of n_s nodes

Because system (1) is over-actuated and each actuator is associated to one node, the scheduling protocol does not need to visit all n_u nodes to ensure the satisfaction of items (ii-a) and (ii-c) of Assumption 3. Indeed, only a subset of $n_s \leqslant n_u$ nodes is a priori needed for this purpose. We can then implement a UGES protocol over these n_s nodes. We formalize this idea below.

For $n, m \in \mathbb{N}$, denote by $\{0,1\}^{n\times m}$ the set of matrices with n lines and m columns where each element is either 0 or 1. The selected n_s actuators correspond to the input $u_s \in \mathbb{R}^{n_s}$ where $u_s = Su$ and $S \in \{0,1\}^{n_s \times n_u}$ is defined to select the corresponding n_s components of u_s from u. On the other hand, the $n_u - n_s$ remaining inputs are denoted by $u_r \in \mathbb{R}^{n_u - n_s}$ and $u_r = Ru$ with $R \in \{0,1\}^{(n_u-n_s)\times n_u}$. We thus write $(u_s, u_r) = [S^\top\, R^\top]^\top u$. Matrix $[S^\top\, R^\top]^\top$ is a permutation matrix and is thus orthogonal. As a consequence, the transposed matrix $\begin{bmatrix} S^\top & R^\top \end{bmatrix}$ corresponds to its inverse and $u = S^\top u_s + R^\top u_r$.

Of course, the n_s actuators cannot be selected arbitrarily and must be able to generate the desired effective control input v, i.e.,

$$\forall u \in \mathbb{R}^{n_u},\ \exists u_s \text{ s.t. } BS^\top u_s = Bu = v. \quad (17)$$

Condition (17) is equivalent to

$$\text{Im}\{BS^\top\} = \text{Im}\{B\} \quad (18)$$

and implies $n_s \geqslant \text{rank}\{B\}$. Hence, the n_s nodes have to be chosen such that (18) holds, i.e., to span the column space of B. In view of (18), we can define $g_s : \mathbb{R}^{n_y} \to \mathbb{R}^{n_s}$ such that

$$u_s = g_s(y) \text{ and } BS^\top g_s(y) = g_{\text{eff}}(y), \quad (19)$$

with g_{eff} from (3). On the other hand, we set $u_r = 0$ as the remaining inputs are not needed to generate the desired effective control input. As a consequence, (3) holds.

Remark 5. When $n_s = \text{rank}\{B\}$, there exists a unique function g_s such that (19) holds. This is due to the left-invertibility of $BS^\top$ when $n_s = \text{rank}\{B\}$. □

We consider update laws at jumps for $\hat{u}$ of the form

$$\hat{u}_s^+ = u_s + h_s(\eta, e_s), \quad \hat{u}_r^+ = 0, \quad (20)$$

where $e_s := \hat{u}_s - u_s$ with u_s given by (19) and $h_s : \mathcal{S} \times \mathbb{R}^{n_s} \to \mathbb{R}^{n_s}$. We assume that h_s ensures the next assumption.

Assumption 6. The protocol equation $e_s^+ = h_s(\eta, e_s)$ is UGES, i.e., there exist $W : \mathcal{S} \times \mathbb{R}^{n_s} \to \mathbb{R}_{\geqslant 0}$, $\rho_s \in (0,1)$ and $\underline{a}_{W,s}, \overline{a}_{W,s} > 0$ such that for all $\eta \in \mathcal{S}$ and $e_s \in \mathbb{R}^{n_s}$,

$$\begin{aligned} \underline{a}_{W,s}|e_s| \leqslant W(\eta, e_s) \leqslant \overline{a}_{W,s}|e_s| \\ W(g_\eta(\eta), h_s(\eta, e_s)) \leqslant \rho_s W(\eta, e_s). \end{aligned} \quad (21)$$

□

Assumption 6 means that the protocol, which schedules network access among the n_s selected nodes, is UGES as defined in (Nešić and Teel, 2004, Definition 7). Examples include RR and TOD protocols, which both satisfy Assumption 6 with:

- (RR) $n_\eta = 1$, $\mathcal{Q} = \mathbb{Z}_{\geqslant 0}$, $f_\eta = 0$, $g_\eta(\eta) = \eta + 1$, $\underline{a}_{W,s} = 1$, $\overline{a}_{W,s} = \sqrt{n_s}$ and $\rho_s = \sqrt{(n_s-1)/n_s}$;

- (TOD) no variable η ($n_\eta = 0$), $\underline{a}_{W,s} = 1$, $\overline{a}_{W,s} = 1$ and $\rho_s = \sqrt{(n_s - 1)/n_s}$.

Assumption 6 implies the satisfaction of items (ii-a) and (ii-c) of Assumption 3 as formalized next.

Proposition 7. Under Assumptions 1 and 6, items (ii-a) and (ii-c) of Assumption 3 hold with W as in Assumption 6, $\underline{a}_W = \underline{a}_{W,s}/|BS^\top|$, $\overline{a}_W = \overline{a}_{W,s}$, and $\rho = \rho_s$. □

The use of UGES protocol over the n_s nodes instead of a UGES protocol over the original n_u nodes leads to smaller constant ρ_s in general, as exemplified by the RR and the TOD protocols above where $\rho_s = \sqrt{(n_s - 1)/n_s}$ instead of $\rho = \sqrt{(n_u - 1)/n_u}$ in the standard case. This may help enlarging the MATI bound in item (ii) of Theorem 4 as the map $\mathcal{T}$ is decreasing with ρ. However, the choice of W also impacts the value of L in item (ii-b) of Assumption 3 and of γ in item (i-b) of Assumption 3 via the definition of function H, which raises the question of how to choose the n_s nodes. Still, the examples in Section 5 show that significant improvements are obtained in this case.

Remark 8. When the Lyapunov function W in Assumption 6 has a uniformly bounded gradient (as for RR and TOD), so is the gradient of W in Proposition 7. Then, item (ii-b) of Assumption 3 follows when f satisfies a linear growth condition as mentioned after Assumption 3. □

To conclude this subsection, we would like to point out that when plant (1) is over-actuated to distribute the control effort among all the actuators, the protocols above are not adapted as $n_u - n_s$ actuators, those corresponding to u_r, are never used. In this case, an alternative is to implement a switched protocol, which schedules transmissions among different groups of selected nodes at each mode. The protocol can then update all actuators signals over a given period of times, while still exploiting input redundancy during each mode. Assuming each protocol is UGES, like in Assumption 6, and that the corresponding Lyapunov function does not grow accross jumps due to the other scheduling rule, we can prove that items (ii-a) and (ii-c) of Assumption 3 is satisfied. The corresponding model and analysis are omitted for space reasons. This will be considered in future work.

4.2 A new maximum-error-first try-once-discard protocol

We have already mentioned that the TOD protocol is a suitable candidate for our purpose as it satisfies items (ii-a) and (ii-c) of Assumption 3, but does not exploit over-actuation. Recall that TOD grants access to the node i for which $|\hat{u}_i - u_i|$ is the largest among all the nodes (Nešić and Teel, 2004; Walsh et al., 2002). The underlying idea being that the network-induced error of this node is the biggest and thus must be sent in priority to the actuators. This leads to the largest possible decrease of $|e_u|$ after a jump. When dealing with over-actuated systems, this rationale may no longer be suitable. Indeed, what matters here is the network-induced error on v, i.e., $e_v = Be_u$, not on u, as already mentioned. We should therefore give priority to the node, which leads to the smallest error $|e_v|$ after a transmission. We present in this section a new protocol called TOD$_{\text{eff}}$ inspired by this idea.

	RR		TOD		TOD$_{\text{eff}}$	
	$\mathcal{T}$	MATI	$\mathcal{T}$	MATI	$\mathcal{T}$	MATI
u_{s1}	0.169	0.260	0.169	0.260	0.169	0.260
u_{s2}	0.015	**0.203**	**0.029**	0.200	0.010	0.202
u_{s3}	0.009	0.132	**0.017**	0.136	0.006	**0.137**
u	0.001	0.115	0.004	0.045	**0.011**	**0.122**

Table 1. Values of $\mathcal{T}$ and of the estimated MATI in Section 5

We select an output-feedback law $g : \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ such that $u = g(y)$ and $Bg(y) = g_{\text{eff}}(y)$. We propose TOD-effective (TOD$_{\text{eff}}$) protocol, which gives the update law for $\hat{u}$ at a transmission

$$\hat{u}^+ \in u + \Psi(i)e_u, \tag{22}$$

where $\Psi(i) := \text{diag}\{\mathbf{1}_{i-1}, 0, \mathbf{1}_{n_u-i}\}$ and i satisfying

$$i \in \text{argmax}_{j\in\{1,\ldots,n_u\}}|B(\mathbb{I} - \Psi(j))e_u|. \tag{23}$$

The term $B\Psi(j)e_u$ corresponds to the value of Be_u after a transmission of node $j \in \{1, \ldots, n_u\}$. Hence, by defining i as in (23), TOD$_{\text{eff}}$ selects the node leading to the largest difference (decrease) of Be_u after one transmission, thereby respecting the rationale of TOD protocol. Note that no variables η are needed here.

The next proposition states that TOD$_{\text{eff}}$ protocol verifies items (ii-a) and (ii-c) of Assumption 3.

Proposition 9. TOD$_{\text{eff}}$ protocol in (22) satisfies items (ii-a) and (ii-c) of Assumption 3 with $W(e_u) = \sqrt{\sum_{i=1}^{n_u} |B(\mathbb{I} - \Psi(i))e_u|^2}$ for any $e_u \in \mathbb{R}^{n_u}$, $\underline{\alpha}_W = 1$, $\overline{\alpha}_W = \sqrt{\sum_{i=1}^{n_u} |B(\mathbb{I} - \Psi(i))|^2}$, and $\rho = \sqrt{(n_u - 1)/n_u}$. □

Remark 10. TOD$_{\text{eff}}$ is presented considering the n_u nodes, but it can also be applied to a selection of n_s nodes as in Section 4.1, see Section 5 for an example. Note also that the partial derivatives of W with respect to e_u in Proposition 9 are uniformly bounded almost everywhere, which is useful to establish item (ii-b) in Assumption 3. □

Remark 11. Constant ρ in Proposition 9 is the same as for classical TOD protocol. However, since $\overline{a}_W$ and L are different, the resulting MATI bound may be larger as it will be shown in Section 5. □

5. NUMERICAL CASE STUDY

We present a numerical example where the plant model and the controller are linear. Note that even for this case the presented results are new. In particular, we consider the scalar system given by $\dot{x}_p = Ax_p + B_vBu$, $y = x_p$ where $A = 5$, $B_v = 1$ and $B = [4\ 3\ 2\ 1]$. We thus have $n_u = 4$ nodes. We have that $\dim(\text{Ker}\{B\}) = 3$ and Assumption 1 holds. Controller (3) is designed as $v = K_vx_p$ with $K_v = -5.9$. We investigate different scheduling policies and compare the corresponding MATI bounds $\mathcal{T}$, see Table 5. We also give an estimation of the MATI based on simulations (see the columns named MATI). The estimation is based on performing a hundred tests, where the initial condition is randomly taken in the set $[-20, 20]$. The largest value of the transmission interval for which the hundred tests remains stable becomes the estimation of the MATI. We start with the "blind" approach, which ignores over-actuation. In particular, we consider RR and TOD protocols over the 4 nodes. We also consider the proposed TOD$_{\text{eff}}$ protocol over the 4 nodes (see the line labelled u). The state feedback control law is given in this case by $u = Kx_p$ where $K = [-0.18\ -0.14\ -0.09\ -4.57]^\top$ that verifies $BK = K_v$. The simulations show that TOD_{eff} leads to a larger estimated MATI compared to the classical TOD. Then we consider successively a selection of $n_{s1} = 1$, $n_{s2} = 2$ and $n_{s3} = 3$ nodes and implement a RR, a TOD or a TOD_{eff} protocol over them. We take for this purpose $u_{s1} = u_1$, $u_{s2} = (u_1, u_2)$ and $u_{s3} = (u_1, u_2, u_3)$, which ensure the satisfaction of (18) with $S_i = \left[\mathbb{I}_i\ \mathbf{0}_{\mathbf{i}\times(\mathbf{4-i})}\right]$, $i \in \{1, 2, 3\}$. In this case, the feedback control law is given by $u_{s_i} = K_{s_i} x_p$ where $K_{s_i} = S_i B^\top (B S_i^\top S_i B^\top)^{-1} K_v$ verifies $B S_i^\top K_{s_i} = K_v$. We observe in Table 5 that when the number of input selected decreases, both $\mathcal{T}$ and the estimated MATI considerably increase. We note that none of the protocols can be considered as superior over the others in terms of MATIs. However, we also see that, when there are four nodes and thus when many nodes are redundant, the new TOD_{eff} provides both the largest values for $\mathcal{T}$ and the estimated MATI. Finally, note that for u_{s1}, since there is only one node, all protocols logically give the same values for $\mathcal{T}$ and the estimated MATI.

6. FUTURE WORK

This works opens the door to various relevant extensions among which the case where the number of actuators differs from the number of nodes, and the scenario where the network is also used to connect the controller to the sensors and the system is "over-sensed", in the sense that there are more sensors than needed to know the plant output y. Also, since over-actuation is often motivated by fault tolerance or robustness to attacks considerations, we plan in future work to take these phenomena explicitly into account in the model and in the analysis to come up with exploitable conditions on the closed-loop system and the scheduling protocol adapted to these contexts.

REFERENCES

Angeli, D. and Sontag, E. (1999). Forward completeness, unboundedness observability, and their Lyapunov characterizations. *Systems & Control Letters*, 38, 209–217.

Bodson, M. (2002). Evaluation of optimization methods for control allocation. *J. of Guidance, Control, and Dynamics*, 25(4), 703–711.

Cai, C., Teel, A., and Goebel, R. (2007). Smooth Lyapunov functions for hybrid systems - Part I: existence is equivalent to robustness. *IEEE Transactions on Automatic Control*, 52(7), 1264–1277.

Carnevale, D., Teel, A., and Nešić, D. (2007). A Lyapunov proof of an improved maximum allowable transfer interval for networked control systems. *IEEE Trans. on Automatic Control*, 52(5), 892–897.

Cristofaro, A., Galeani, S., and Corradini, M.L. (2018). A saturated dynamic input allocation policy for preventing undetectable attacks in cyber-physical systems. In *2018 European Control Conference (ECC)*, 845–850.

Dolk, V., Ploeg, J., and Heemels, W. (2017). Event-triggered control for string-stable vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems*, 18(12), 3486–3500.

Donkers, M., Heemels, W., van de Wouw, N., and Hetel, L. (2011). Stability analysis of networked control systems using a switched linear systems approach. *IEEE Transactions on Automatic Control*, 56(9), 2101–2115.

Harkegard, O. and Glad, S.T. (2005). Resolving actuator redundancy-optimal control vs. control allocation. *Automatica*, 41(1), 137–144.

Heemels, W., Teel, A., van de Wouw, N., and Nešić, D. (2010). Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *IEEE Trans. on Aut. Control*, 55(8), 1781–1796.

Hertneck, M. and Allgöwer, F. (2020). A simple approach to increase the maximum allowable transmission interval. *arXiv preprint arXiv:2006.03268*.

Hetel, L., Fiter, C., Omran, H., Seuret, A., Fridman, E., Richard, J.P., and Niculescu, S. (2017). Recent developments on the stability of systems with aperiodic sampling: An overview. *Automatica*, 76, 309–335.

Johansen, T. and Fossen, T. (2013). Control allocation-a survey. *Automatica*, 49(5), 1087–1103.

Kreiss, J., Bodson, M., Delpoux, R., Gauthier, J.Y., Trégouët, J.F., and Lin-Shi, X. (2021). Optimal control allocation for the parallel interconnection of buck converters. *Control Engineering Practice*, 109, 104727.

Liu, K., Fridman, E., and Hetel, L. (2012). Stability and L2-gain analysis of networked control systems under Round-Robin scheduling: a time-delay approach. *Systems & control letters*, 61(5), 666–675.

Nešić, D. and Teel, A. (2004). Input-output stability properties of networked control systems. *IEEE Trans. on Aut. Control*, 49, 1650–1667.

Oppenheimer, M., Doman, D., and Bolender, M. (2006). Control allocation for over-actuated systems. In *Med. Conf. on Control and Automation, Ancona, Italy*, 1–6.

Park, P., Ergen, S., Fischione, C., Lu, C., and Johansson, K. (2017). Wireless network design for control systems: A survey. *IEEE Comm. Surveys & Tutorials*, 20(2), 978–1013.

Peters, E., Quevedo, D., and Fu, M. (2016). Controller and scheduler codesign for feedback control over IEEE 802.15.4 networks. *IEEE Transactions on Control Systems Technology*, 24(6), 2016–2030.

Schenato, L. (2009). To zero or to hold control inputs with lossy links? *IEEE Transactions on Automatic Control*, 54(5), 1093–1099.

Singh, A.K., Singh, R., and Pal, B.C. (2014). Stability analysis of networked control in smart grids. *IEEE Transactions on Smart Grid*, 6(1), 381–390.

Teel, A. and Praly, L. (2000). A smooth Lyapunov function from a class-$\mathcal{KL}$ estimate involving two positive semidefinite functions. *ESAIM: Control, Optimisation and Calculus of Variations*, 5(1), 313–367.

Trip, S., Scholten, T., and De Persis, C. (2019). Optimal regulation of flow networks with transient constraints. *Automatica*, 104, 141–153.

Walsh, G.C., Ye, H., and Bushnell, G. (2002). Stability analysis of networked control systems. *IEEE Transactions Control Systems Technology*, 10(3), 438–446.

Walsh, G., Beldiman, O., and Bushnell, L. (2001). Asymptotic behavior of nonlinear networked control systems. *IEEE Trans. on Aut. Control*, 46, 1093–1097.

Zaccarian, L. (2009). Dynamic allocation for input redundant control systems. *Automatica*, 45(6), 1431–1438.