Finite-Gain $\mathcal{L}_2$ Stability in Distributed Event-Triggered Networked Control Systems with Data Dropouts

Xiaofeng Wang and Michael D. Lemmon

***Abstract*— This paper studies event-triggered networked control systems. Event-triggering has an agent broadcasts its sampled state to its neighbors only when its local error signal exceeds a given threshold. We provide a sufficient condition for the existence of "local" events such that the resulting networked control systems is $\mathcal{L}_2$ stable. By "local", we mean that each agent's events are only associated with its own state information. Based on this condition, we propose a distributed scheme for each agent to design its local events, using linear matrix inequalities. This scheme applies to linear systems and the resulting event-triggered system is finite-gain $\mathcal{L}_2$ stable. Moreover, we consider data dropouts in networked control systems and propose a distributed method that enables each agent to locally identify the maximal allowable number of its successive data dropouts without loss of the system stability.**

I. Introduction

Networked Control Systems (NCS) have received a lot of attention these days. In such a system, numerous subsystems (also called "agents") that are physically coupled together exchange information through a real-time communication network. Specific examples of NCS include electrical power grids and transportation networks. The networking of control effort in NCS can be advantageous in terms of lower system costs due to streamlined installation and maintenance costs.

The introduction of communication networks, however, raises new challenges. With a real-time network, the communication media is customarily accessed in a mutually exclusive manner. In other words, only one agent can transmit its information at a time. Moreover, data has to be transmitted in a discrete-time manner instead of continuous-time. Therefore, one important issue in the implementation of such NCS is to identify the transmission decision logics that can provide guarantees on overall system performance.

Early work analyzing scheduling of real-time network traffic was presented in [1]. However, the impact of communication constraints on system performance was not been addressed in these works. [2] noticed the harmful effect of the communication delay on the system stability and considered the one packet transmission problem, where all of the system outputs were packaged into a single packet. As a result, agents in the network do not have to compete for channel access. One packet transmission strategies, however, use a supervisor to summarize all subsystem data into this single packet. As a result such schemes may be impractical for large-scale systems.

Asynchronous transmissions were considered in [3]. This work derived bounds on the maximum admissible time interval (MATI) that a message can be delayed while still maintaining closed loop system stability. It led to scheduling methods [4] that were able to assure the MATI was not violated. Further work was done in [5], [6] to tighten bounds on the MATI. All this work confined its attention to control area network buses where centralized computers are used to coordinate the information transmission with some protocols.

However, in all of the aforementioned work, the computation of the bounds on the MATI and the execution of the protocols must be done in a highly centralized manner, which is impractical in large-scaled systems as we mentioned before. Moreover, because the MATI is computed before the system is deployed, it must ensure adequate behavior over a wide range of possible input disturbances. As a result, the MATI may be conservative. Consequently, the bandwidth of the network will be higher than necessary to ensure that the MATI is not violated. These limitations suggest a great need for distributed approaches to address this timing issue in a way that enables the NCS to use network bandwidth in an extremely frugal manner.

Recently, decentralized event-triggering feedback schemes were proposed in [7] and [8] for asymptotic stability of linear and nonlinear systems, respectively, where an agent broadcasts its sampled state to its neighbors only when its local error signal exceeds a given threshold. Most recently, an implementation of event-triggering in sensor-network was introduced in [9]. As mentioned in [10], [11], event-triggering can dynamically adjust the task periods according to the variation in system states. This makes it possible to reduce the frequency with which subsystems communicate and therefore save network bandwidth.

This paper studies event-triggered NCS with finite-gain $\mathcal{L}_2$ stability. We first provide a sufficient condition for the existence of "local" events such that the resulting event-triggered NCS is $\mathcal{L}_2$ stable. By "local", we mean that each agent's events are only associated with its own state information. Based on this condition, we propose a distributed scheme for each agent to design its local events, using linear matrix inequalities (LMI). This scheme applies to linear systems and the resulting NCS is finite-gain $\mathcal{L}_2$ stable.

Another contribution of this paper is its consideration of data dropouts that always happen in real-time network, but were not considered in [7], [8], [9]. Unlike the prior work that modelled data dropouts as stochastic processes using a centralized approach [12], [13], we propose a distributed method that enables each agent to locally identify the

Both authors are with the department of electrical engineering, University of Notre Dame, Notre Dame, IN 46556; e-mail: xwang13,lemmon@nd.edu.
The authors gratefully acknowledge the partial financial support of the National Science Foundation (NSF CNS-0720457)

maximal allowable number of its successive data dropouts (MANSD) with the guarantee of NCS's $\mathcal{L}_2$ stability. We use an example to illustrate the method to compute MANSD as well as the distributed event design procedure.

The paper is organized as follows: section II formulates the problem; a sufficient condition for the existence of local events is presented in section III; a distributed event design scheme is proposed in section IV; Section V discusses the data dropouts in event-triggered NCS; Simulation results are presented in section VI; Section VII draws the conclusions.

II. Problem Formulation

Consider a distributed NCS containing N subsystems (also called "agents"). Let $\mathcal{N} = \{1, 2, \cdots, N\}$. $Z_i \in \mathcal{N}$ denotes the set of agents that agent i can get information from; $D_i \subset \mathcal{N}$ denotes the set of agents that directly drive agent i's dynamics; $U_i \in \mathcal{N}$ denotes the set of agents that can receive agent i's broadcasted information; $S_i \in \mathcal{N}$ denotes the set of agents who are directly driven by agent i. Here $i \notin Z_i \cup D_i \cup U_i \cup S_i$.

The state equation of the ith agent is

$$\begin{array}{l}\dot{x}_i(t) = A_{ii}x_i(t) + B_i u_i(t) + \sum_{j\in D_i} A_{ij}x_j(t) + C_i w_i(t)\\ u_i(t) = K_{ii}x_i(t) + \sum_{j\in Z_i} K_{ij}x_j(t),\end{array} \tag{1}$$

where $x_i : \mathbb{R} \to \mathbb{R}^n$ is the state trajectory of agent i, $u_i : \mathbb{R} \to \mathbb{R}^m$ is a control input and $w_i : \mathbb{R} \to \mathbb{R}^l$ is an exogenous disturbance function in $\mathcal{L}_2$ space. To simplify the analysis, we assume that the states/inputs/disturbances of the agents have the same dimension. The results in this paper can be easily extended to the case where the dimensions of agents' states/inputs/disturbances are different from each other.

This paper considers a real-time implementation of this distributed NCS. In such a system, agent i can only detect its own state, x_i. If the local error signal exceeds some given threshold, which can be detected by hardware detectors, agent i will sample and broadcast its state information to its neighbors through a communication network. Agent i's control, u_i, at time t is computed based on its neighbors' latest broadcast states (also called "measured states") at time t, denoted as $\hat{x}_j(t) \in \mathbb{R}^n$. We define the local error $e_i : \mathbb{R} \to \mathbb{R}^n$ by $e_i(t) = x_i(t) - \hat{x}_i(t)$ and $e = (e_1^T, \cdots, e_N^T)^T$. We also assume that there is no delay between broadcasting the sampled state and applying the updated control inputs to the subplants. The control signal used by agent i is held constant by a zero-order hold (ZOH) until one of its neighbors makes another broadcast. This means that agent i has the following state equation,

$$\begin{array}{l}\dot{x}_i(t) = A_{ii}x_i(t) + B_i u_i(t) + \sum_{j\in D_i} A_{ij}x_j(t) + C_i w_i(t)\\ u_i(t) = K_{ii}\hat{x}_i(t) + \sum_{j\in Z_i} K_{ij}\hat{x}_j(t).\end{array} \tag{2}$$

Therefore, the state equation of the entire NCS is

$$\begin{array}{rcl}\dot{x}(t) &=& Ax(t) + Bu(t) + Cw(t)\\ u(t) &=& K\hat{x}(t).\end{array} \tag{3}$$

In equation (3), $x = (x_1^T, \cdots, x_N^T)^T$, $u = (u_1^T, \cdots, u_N^T)^T$, $w = (w_1^T, \cdots, w_N^T)^T$, and

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1N}\\ \cdots & \cdots & \cdots\\ A_{N1} & \cdots & A_{NN}\end{bmatrix}, B = \begin{bmatrix} B_1 & \cdots & 0\\ \cdots & \cdots & \cdots\\ 0 & \cdots & B_N\end{bmatrix},$$
$$K = \begin{bmatrix} K_{11} & \cdots & K_{1N}\\ \cdots & \cdots & \cdots\\ K_{N1} & \cdots & K_{NN}\end{bmatrix}, C = \begin{bmatrix} C_1 & \cdots & 0\\ \cdots & \cdots & \cdots\\ 0 & \cdots & C_N\end{bmatrix},$$

where $A_{ij} = \mathbf{0}$ if $j \notin D_i$ and $K_{ij} = \mathbf{0}$ if $j \notin Z_i$.

Definition 2.1: The system in equation (3) is said to be finite-gain $\mathcal{L}_2$ stable from w to x with an induced gain less than γ if there exist non-negative constants γ and ξ such that

$$\left(\int_0^\infty \|x(t)\|_2^2 dt\right)^{\frac{1}{2}} \le \gamma \left(\int_0^\infty \|w(t)\|_2^2 dt\right)^{\frac{1}{2}} + \xi \tag{4}$$

for any $w \in \mathcal{L}_2$.

The objective of this paper is to develop distributed event-triggering schemes to identify the broadcast release time such that the NCS defined in equation (2) is finite-gain $\mathcal{L}_2$ stable.

III. Local Event Design

In this section, we propose a centralized approach to design local events for agents that are used to trigger the broadcast. Linear Matrix Inequalities (LMI) are used to identify the parameters in those events. The resulting event-triggered NCS is finite-gain $\mathcal{L}_2$ stable, as shown in theorem 3.1. We use $|S| \in \mathbb{N}$ to denote the number of the elements in a given set S, $\|\cdot\|_2$ to denote 2-norm of a vector, and $\|\cdot\|$ to denote the matrix norm.

Theorem 3.1: Consider the NCS in equation (3). Assume that there exist positive-definite matrices $P, Q \in \mathbb{R}^{nN\times nN}$ and $W_i, M_i \in \mathbb{R}^{n\times n}$, $i = 1, 2, \cdots, N$ such that:

$$P(A+BK) + (A+BK)^T P + \frac{1}{\gamma^2}PCC^T P \le -Q \tag{5}$$

$$Q - PBKM^{-1}K^T B^T P \ge W \tag{6}$$

$$P, Q, M_i, W_i > 0, \tag{7}$$

hold, where $M = \text{diag}\{M_j\}_{j\in\mathcal{N}}$ and $W = \text{diag}\{W_j\}_{j\in\mathcal{N}}$. If for any $i \in \mathcal{N}$, the inequality

$$-\rho_i x_i^T(t)W_i x_i(t) + e_i^T(t)M_i e_i(t) \le 0, \tag{8}$$

holds for all $t \ge 0$ with some $\rho_i \in (0, 1)$, then the NCS is finite-gain $\mathcal{L}_2$ stable with an induced gain less than $\frac{\gamma}{\sqrt{\min_i\{(1-\rho_i)\lambda_{\min}(W_i)\}}}$.

Proof: Note that, when agent i broadcasts its state, equation (8) will be trivially satisfied since $e_i(t) = 0$ at that time. The well-posedness of equation (8) is established.

Consider $\dot{V}$ with $V(x) = x^T Px$ at time t. For notational convenience, we use x, $\hat{x}$, e, w, x_i, $\hat{x}_i$, e_i to denote $x(t)$, $\hat{x}(t)$, $e(t)$, $w(t)$, $x_i(t)$, $\hat{x}_i(t)$, $e_i(t)$, respectively.

$$\begin{aligned}\dot{V} &= x^T(PA + A^T P)x + 2x^T PBK\hat{x} + 2x^T PCw\\ &\le x^T(P(A+BK) + (A+BK)^T P + \frac{1}{\gamma^2}PCC^T P)x\\ &\quad -2x^T PBKe + \gamma^2\|w\|_2^2\\ &\le -x^T Qx - 2x^T PBKe + \gamma^2\|w\|_2^2\\ &\le -x^T(Q - PBKM^{-1}K^T B^T P)x + e^T Me + \gamma^2\|w\|_2^2\end{aligned}$$

Combining equation (6) and the preceding inequality yields

$$\begin{aligned}\dot{V} &\leq -x^T W x + e^T M e + \gamma^2 \|w\|_2^2 \\ &= -\sum_{i\in\mathcal{N}} x_i^T W_i x_i + \sum_{i\in\mathcal{N}} e_i^T M_i e_i + \gamma^2 \|w\|_2^2.\end{aligned}$$

Applying equation (8) into the preceding equation yields

$$\begin{aligned}\dot{V} &\leq -\sum_{i\in\mathcal{N}} (1-\rho_i) x_i^T W_i x_i + \gamma^2 \|w\|_2^2 \\ &\leq -\min_i \{(1-\rho_i)\lambda_{\min}(W_i)\} \|x\|_2^2 + \gamma^2 \|w\|_2^2\end{aligned}$$

for any $t \geq 0$, which is sufficient to show that the NCS in equation (2) is finite-gain $\mathcal{L}_2$ stable with an induced gain less than $\frac{\gamma}{\sqrt{\min_i\{(1-\rho_i)\lambda_{\min}(W_i)\}}}$. ■

Remark 3.2: Equation (5), (6) can be rewritten as

$$\left[\begin{array}{cc} -P(A+BK)-(A+BK)^TP-Q & PC \\ C^TP & \gamma^2 I_{lN\times lN}\end{array}\right] \geq 0 \tag{9}$$

$$\left[\begin{array}{cc} Q-W & PBK \\ K^TB^TP & M\end{array}\right] \geq 0, \tag{10}$$

respectively. Therefore, equation (7), (9), (10) form linear matrix inequalities (LMI), which can be used to compute the desired matrices.

Remark 3.3: We use the violation of the inequality in equation (8) to trigger agent i's broadcast. Notice that this inequality is only associated with $x_i(t)$ and $e_i(t)$. The positive-definiteness of W_i is to ensure the satisfaction of inequality (8) when broadcasts happen, as shown in the beginning of the proof. Otherwise, it may lead to continuous broadcasts of agent i, which is impractical in reality.

Theorem 3.1 shows that we need to find W_i, M_i to construct the local events. It follows immediately that the matrices $\{W_j\}_{j\in\mathcal{N}}$ and $\{M_j\}_{j\in\mathcal{N}}$ required in theorem 3.1 always exist, provided that equation (5) holds. This is stated in corollary 3.4.

Corollary 3.4: Consider the NCS in equation (2). If equation (5) holds, then there always exist positive definite matrices $W_i, M_i \in \mathbb{R}^{n\times n}$, $i \in \mathcal{N}$ satisfying equation (6).

Proof: Because $Q > 0$, there must exist a positive constant $\varepsilon \in (0, \lambda_{\min}(Q))$. It is easy to verify that

$$W_i = \varepsilon I_{n\times n} \text{ and } M_i = \frac{\|PBK\|^2}{\lambda_{\min}(Q)-\varepsilon} I_{n\times n}$$

satisfy equation (6), (7). ■

Theorem 3.1 presents a method to design local events and corollary 3.4 shows the existence of these events. As mentioned in remark 3.2, equation (5), (6), (7) in theorem 3.1 can be posed as LMI. However, directly solving this LMI for an admissible solution is a highly centralized approach. It may not be suitable for large-scale systems. In fact, notice that in order to design its local event, agent i just needs to find the matrices W_i and M_i. This allows the possibility to decentralize the design procedure. In the following section, we will further discuss the distributed design scheme.

IV. Distributed Design Scheme

In this section, we propose a distributed event design scheme for NCS. In this approach, each agent is associated with a local LMI problem. The feasibility of these local LMI implies the feasibility of the centralized LMI in equation (7), (9), (10), since the solutions to local LMI can be used to construct the solution to the centralized LMI.

Let take a look at agent i. Assume that

$$\begin{aligned}&Z_i \cup \{i\} = \{i_1, i_2, \cdots, i_{q_i}\} \subseteq \mathcal{N}, \\ &Z_i \cup D_i \cup \{i\} = \{i_1, \cdots, i_{q_i}, i_{q_i+1}, \cdots, i_{s_i}\} \subseteq \mathcal{N}.\end{aligned}$$

It is easy to see that $q_i = |Z_i| + 1$ and $s_i = |Z_i \cup D_i| + 1$. Without loss of the generality, we assume $i_1 = i$.

For notational convenience, we define four matrices $A_i \in \mathbb{R}^{n\times Ns_i}$, $K_i \in \mathbb{R}^{m\times ns_i}$, and $\tilde{K}_i \in \mathbb{R}^{m\times nq_i}$, $H_i \in \mathbb{R}^{ns_i\times l}$ by

$$\begin{aligned}A_i &= (A_{i,i_1}, A_{i,i_2}, \cdots, A_{i,i_{s_i}}) \in \mathbb{R}^{n\times ns_i}, \\ K_i &= (K_{i,i_1}, K_{i,i_1}, \cdots, K_{i,i_{s_i}}) \in \mathbb{R}^{m\times ns_i}, \\ \tilde{K}_i &= (K_{i,i_1}, K_{i,i_1}, \cdots, K_{i,i_{q_i}}) \in \mathbb{R}^{m\times nq_i}, \\ H_i &= \left[\begin{array}{c} P_iC_i \\ \mathbf{0}\end{array}\right] \in \mathbb{R}^{ns_i\times l},\end{aligned}$$

and two functions $F_i : \mathbb{R}^{n\times n} \to \mathbb{R}^{ns_i\times ns_i}$ and $G : \mathbb{R}^{n\times n} \times \mathbb{R} \to \mathbb{R}^{ns_i\times ns_i}$ by

$$F_i(P_i) = \left[\begin{array}{c} P_i(A_i + B_iK_i) \\ \mathbf{0}\end{array}\right] \in \mathbb{R}^{ns_i\times ns_i},$$

$$G_i(Q_i;\beta) = \left[\begin{array}{cccc} Q_i & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\beta I & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & -\beta I\end{array}\right] \in \mathbb{R}^{ns_i\times ns_i}.$$

With these matrices and functions, we can define the local LMI problem associated with agent i:

Problem 4.1 (Local LMI): Given constants $\delta, \beta > 0$, find $P_i, Q_i, W_i \in \mathbb{R}^{n\times n}$ and $\gamma_i \in \mathbb{R}$ such that

$$\left[\begin{array}{cc} -F_i(P_i) - F_i^T(P_i) - G_i(Q_i;\beta) & H_i \\ H_i^T & \gamma_i I_{l\times l}\end{array}\right] \geq 0, \tag{11}$$

$$\left[\begin{array}{cc} Q_i - |S_i \cup U_i|\beta I_{n\times n} - W_i & P_iB_i\tilde{K}_i \\ \tilde{K}_i^T B_i^T P_i & \delta I_{nq_i\times nq_i}\end{array}\right] \geq 0, \tag{12}$$

$$P_i, W_i > 0, \gamma_i > 0. \tag{13}$$

Here is the distributed event design procedure for NCS.

Distributed Event Design Procedure (DEDP)

```
1   Select $\delta, \beta > 0$;
2   For subsystem $i$,
    (1)   Solve problem 4.1 for $P_i$, $W_i$, $Q_i$ and $\gamma_i$;
    (2)   Design the broadcast-triggering event:
              $-\rho_i x_i^T(t) W_i x_i(t) + \delta(|U_i| + 1)\|e_i(t)\|_2^2 = 0$
          for some $\rho_i \in (0, 1)$.
```

Remark 4.2: In DEDP, two parameters, δ and β, are preselected and all agents share the same δ and β. We will discuss how to select these two parameters later (in corollary 4.6 and 4.7). A more general setup is to pre-select a group of parameters $\{\delta_i\}_{i=1}^N$ and $\{\beta_i\}_{i=1}^N$. For any $i \in \mathcal{N}$, agent i is associated with a pair of (δ_i, β_i). In that case, the definitions of F_i and G_i as well as equation (12) have to be changed slightly. So does the structure of local events. These changes will not affect the main results in this paper. To outline the main idea, we just use two parameters.

The following theorem shows that using DEDP, the resulting event-triggered NCS is $\mathcal{L}_2$ stable.

Theorem 4.3: Consider the NCS in equation (2). Assume that for any $i \in \mathcal{N}$, the local LMI in problem 4.1 is feasible and P_i, Q_i, $W_i \in \mathbb{R}^{n\times n}$, and $\gamma_i \in \mathbb{R}$ are the solutions. If for any $i \in \mathcal{N}$, the inequality

$$-\rho_i x_i^T(t) W_i x_i(t) + \delta(|U_i|+1)\|e_i(t)\|_2^2 \leq 0, \tag{14}$$

holds for all $t \geq 0$ with some $\rho_i \in (0,1)$, then the NCS is finite-gain $\mathcal{L}_2$ stable with an induced gain less than $\frac{\max_i\{\sqrt{\gamma_i}\}}{\sqrt{\min_i\{(1-\rho_i)\lambda_{\min}(W_i)\}}}$.

Proof: Notice that the inequality still holds when we expand the matrices in equation (11) into $nN \times nN$ dimension by appropriately adding zero. Summing both sides of the expanded matrix inequalities yields the satisfaction of equation (9) and therefore equation (5) with

$$P = \text{diag}\{P_i\}_{i=1}^N, \quad \gamma = \max_i\{\sqrt{\gamma_i}\}$$
$$Q = \text{diag}\{Q_i - |S_i \cup U_i|\beta I_{n\times n}\}_{i=1}^N$$

where $Q_i - |S_i \cup U_i|\beta I_{n\times n} > 0$ holds due to equation (12). Similarly, we canshow the satisfaction of equation (6) with

$$W = \text{diag}\{W_i\}_{i=1}^N, \text{ and } M = \text{diag}\{\delta(|U_i|+1)I_{n\times n}\}_{i=1}^N.$$

Since the hypotheses in theorem 3.1 are satisfied, we conclude that the NCS is finite-gain $\mathcal{L}_2$ stable with an induced gain less than $\frac{\max_i\{\sqrt{\gamma_i}\}}{\sqrt{\min_i\{(1-\rho_i)\lambda_{\min}(W_i)\}}}$. ■

Remark 4.4: Since the two parameters δ and β are preselected, the local problem associated with agent i only requires the information on agent i's system dynamics. To design the local events, agents do not have to know other agents' information. So the design scheme is distributed.

Remark 4.5: The dimensions of the matrices in the left-hand side of LMI in equation (11) and (12) are $(ns_i + l) \times (ns_i + l)$ and $(nq_i + n) \times (nq_i + n)$, respectively. They are much smaller than the dimensions of the matrices in the left-hand side of LMI in equation in (9) and (10), which are $(nN + lN) \times (nN + lN)$ and $2nN \times 2nN$, respectively. Even when $s_i = q_i = N$, which means that each agent is coupled with all other agents, the local LMI still has a smaller scale. One thing worth mentioning is that although the dimension of the local LMI is smaller, conservativeness is introduced since matrices P and Q are restricted to be block diagonal.

As shown in theorem 4.3, the feasibility of local LMI determines the existence of local events. To ensure the feasibility, the selection of δ, β is very important. The following corollary provides a sufficient condition for the feasibility of local LMI.

Corollary 4.6: Consider the NCS in equation (2). For any $i \in \mathcal{N}$, if there exist positive-definite matrices $P_i \in \mathbb{R}^{n\times n}$ such that

$$F_i(P_i) + F_i^T(P_i) + G_i(|S_i \cup U_i|\beta I_{n\times n}; \beta) < 0 \tag{15}$$

then there always exists a positive constant $\delta^* \in \mathbb{R}^+$, such that for any $\delta \geq \delta^*$, the LMI in problem 4.1 is feasible.

Proof: Equation (15) implies that there exists a positive definite matrix $Q_i \in \mathbb{R}^{n\times n}$ such that

$$F_i(P_i) + F_i^T(P_i) + G_i(Q_i; \beta) < 0 \tag{16}$$
$$Q_i - |S_i \cup U_i|\beta I_{n\times n} > 0 \tag{17}$$

Since equation (16) holds, we know that there always exists a positive constant $\gamma_i^* \in \mathbb{R}^+$ such that for all $\gamma_i \geq \gamma_i^*$, equation (11) holds.

Equation (17) implies that there exists a positive definite matrix $W_i \in \mathbb{R}^{n\times n}$ such that

$$Q_i - |S_i \cup U_i|\beta I_{n\times n} - W_i > 0 \tag{18}$$

which suggests that there always exists a positive constant $\delta^* \in \mathbb{R}^+$ such that for all $\delta \geq \delta^*$, equation (12) holds. ■

Corollary 4.6 suggests that δ must be large enough to guarantee the feasibility of the local LMI, provided equation (15) holds. We still need to know how to select β. In the following corollary, we show that the satisfaction of equation (15) is independent of the selection of β.

Corollary 4.7: If there exist a positive-definite matrix $P_i \in \mathbb{R}^{n\times n}$ and a positive constant $\beta \in \mathbb{R}$ such that equation (15) holds, then for any $\hat{\beta} > 0$, the pair $\frac{\hat{\beta}}{\beta}P_i$ and $\hat{\beta}$ also satisfies equation (15).

Proof: This can be easily proven by the definitions of F_i and G_i. ■

Remark 4.8: Corollary 4.7 means that the existence of P_i satisfying equation (15) is independent of the value of β. As to the existence of events, we just need to arbitrarily pick a positive constant β in the first step of DEDP.

V. Data Dropouts

In the previous sections, we did not consider data dropouts. In other words, whenever a broadcast release is triggered, the agent will sample and transmit its local state to its neighbors successfully. Data dropouts, however, frequently happen in NCS. In this section, we study the NCS with data dropouts. In particular, we consider the networks in which the agent will not be notified when transmission fails. An assumption is that data dropouts only happen when the sampled states are sent to the controllers through the network.

In such a system, when the hardware detector located at agent i detects the occurrence of the local event, the local state will be sampled and ready to be transmitted to its neighbors through the channel. At the same time, the event will be automatically updated from k to $k+1$ with the newly sampled state. Once the transmission fails, the controllers will not receive the sampled state and, therefore, the control inputs will not be updated. Notice that in this case, the local event is updated, but the control inputs are not.

In the following discussion, we provide a distributed approach that enables each agent to locally determine the *maximal allowable number of successive data dropouts* (MANSD) of that agent with the guarantee of stability of the NCS. The idea is to have events happen earlier than the violation of the inequality in equation (14) so that even if some data is lost, equation (14) can still be satisfied.

Before we introduce the results, we need to define two different types of broadcast release: the triggered release, r_j^i , and the successful release, b_k^i. r_j^i is the time when the jth broadcast of agent i is released, but not necessarily transmitted successfully. b_k^i is the time when the kth successful broadcast of agent i is released. Obviously, $\{b_k^i\}_{k=1}^\infty$ is a subsequence of $\{r_j^i\}_{j=1}^\infty$. Notice that $\hat{x}_i(t) = x_i(b_k^i)$ for all $t \in [b_k^i, b_{k+1}^i)$. For notational convenience, we define $\hat{e}_i^j : \mathbb{R} \to \mathbb{R}^n$ as $\hat{e}_i^j(t) = x_i(t) - x_i(r_j^i)$ for $t \in [r_j^i, r_{j+1}^i)$.

Theorem 5.1: Consider the NCS in equation (2). Assume that for any $i \in \mathcal{N}$, the local LMI in problem 4.1 is feasible. If for any $i \in \mathcal{N}$, the next broadcast release time, r_{j+1}^i, is triggered by the violation of

$$-\rho_i\|x_i(t)\|_2 + \sigma_i\|\hat{e}_i^j(t)\|_2 \le 0, \tag{19}$$

for some $\rho_i \in (0,1)$, where

$$\sigma_i = \sqrt{\frac{\delta(|U_i|+1)}{\lambda_{\min}(W_i)}} \tag{20}$$

and the largest number of successive data dropouts, $d_i \in \mathbb{Z}$, satisfies

$$d_i < \log_{\left(1+\frac{\rho_i}{\sigma_i}\right)}\left(1+\frac{1}{\sigma_i}\right) - 1 \tag{21}$$

then the NCS is still finite-gain $\mathcal{L}_2$ stable.

Proof: For notational convenience, we assume $b_k^i = r_0^i < r_1^i < \cdots < r_{d_i}^i < r_{d_i+1}^i = b_{k+1}^i$ over $[b_k^i, b_{k+1}^i)$.

Notice that for any $t \in [b_k^i, b_{k+1}^i)$, there must be $j \in \{0, 1, \cdots, d_i\}$ so that $t \in [r_j^i, r_{j+1}^i)$. Consider $\|e_i(t)\|_2$ over the time interval $[r_j^i, r_{j+1}^i)$.

$$\begin{aligned}
&\|e_i(t)\|_2 = \|x_i(t) - x_i(b_k^i)\|_2 \\
&\le \textstyle\sum_{p=0}^{j-1} \|x_i(r_{p+1}^i) - x_i(r_p^i)\|_2 + \|x_i(t) - x_i(r_j^i)\|_2 \\
&= \textstyle\sum_{p=0}^{j-1} \|\hat{e}_i^p(r_{p+1}^i)\|_2 + \|\hat{e}_i^j(t)\|_2
\end{aligned}$$

holds for $\forall t \in [r_j^i, r_{j+1}^i)$.

Applying equation (19) into the preceding equation yields

$$\|e_i(t)\|_2 \le \textstyle\sum_{p=0}^{j-1} \frac{\rho_i}{\sigma_i}\|x_i(r_{p+1}^i)\|_2 + \frac{\rho_i}{\sigma_i}\|x_i(t)\|_2 \tag{22}$$

for all $t \in [r_j^i, r_{j+1}^i)$ and all $j = 0, 1, \cdots, d_i$.

For any $t \in [r_j^i, r_{j+1}^i)$, $\|\hat{e}_i^j(t)\|_2 = \|x_i(t) - x_i(r_j^i)\|_2 \le \frac{\rho_i}{\sigma_i}\|x_i(t)\|_2$ holds. Therefore, we have

$$\|x_i(r_j^i)\|_2 \le \left(1 + \frac{\rho_i}{\sigma_i}\right)\|x_i(t)\|_2. \tag{23}$$

Similarly, we have

$$\|x_i(r_{p+1}^i)\|_2 \le \left(1 + \frac{\rho_i}{\sigma_i}\right)\|x_i(r_{p+2}^i)\|_2 \tag{24}$$

for $p = 0, 1, 2, \cdots, j-2$. Equation (23), (24) imply that

$$\|x_i(r_{p+1}^i)\|_2 \le \left(1 + \frac{\rho_i}{\sigma_i}\right)^{j-p}\|x_i(t)\|_2 \tag{25}$$

for $p = 0, \cdots, j-1$. Applying equation (25) into (22) yields

$$\begin{aligned}
&\|e_i(t)\|_2 \le \textstyle\sum_{p=0}^{j-1} \frac{\rho_i}{\sigma_i}\left(1 + \frac{\rho_i}{\sigma_i}\right)^{j-p}\|x_i(t)\|_2 + \frac{\rho_i}{\sigma_i}\|x_i(t)\|_2 \\
&\le \left(\left(1 + \frac{\rho_i}{\sigma_i}\right)^{d_i+1} - 1\right)\|x_i(t)\|_2
\end{aligned} \tag{26}$$

for all $t \in [r_j^i, r_{j+1}^i)$. Since t is arbitrarily selected over $[b_k^i, b_{k+1}^i)$, equation (26) holds for all $t \in [b_k^i, b_{k+1}^i)$.

By equation (21), we know that $\left(1 + \frac{\rho_i}{\sigma_i}\right)^{d_i+1} - 1 < \frac{1}{\sigma_i}$. So there must be a positive constant $\kappa_i \in (0,1)$ such that $\left(1 + \frac{\rho_i}{\sigma_i}\right)^{d_i+1} - 1 \le \frac{\kappa_i}{\sigma_i} < \frac{1}{\sigma_i}$ holds. Combining this inequality with equation (26) yields $\|e_i(t)\|_2 \le \frac{\kappa_i}{\sigma_i}\|x_i(t)\|_2$ with $\kappa_i \in (0,1)$ for all $t \in [b_k^i, b_{k+1}^i)$. Since the hypotheses in theorem 4.3 are satisfied, we conclude that the NCS is finite-gain $\mathcal{L}_2$ stable with an induced gain less than $\frac{\max_i\{\sqrt{\gamma_i}\}}{\sqrt{\min_i\{(1-\kappa_i)\lambda_{\min}(W_i)\}}}$. ∎

Remark 5.2: By equation (21), we know that each agent's MANSD can be identified locally, depending on the selection of ρ_i. If agent i wants its MANSD to be large, ρ_i must be small. In general, however, small ρ_i will result in short broadcast periods $(r_{j+1}^i - r_j^i)$. Therefore, there is a tradeoff between the MANSD and the broadcast period.

VI. Simulations

This section presents simulation results demonstrating the decentralized event-triggering scheme. The systemunder study is a collection of carts, which are coupled together by springs (figure 1). The ith subsystem state is the vector $x_i = \left[\begin{array}{cc} y_i & \dot{y}_i \end{array}\right]^T$ where y_i is the ith cart's position. We assume that at the equilibrium, all springs are unstretched.

The state equation for the ith cart is equation (2), where

$$A_{ii} = \left[\begin{array}{cc} 0 & 1 \\ -\mu_i k & 0 \end{array}\right], B_i = \left[\begin{array}{c} 0 \\ 1 \end{array}\right], \tag{27}$$

$$A_{ij} = \left[\begin{array}{cc} 0 & 0 \\ \nu_{ij} k & 0 \end{array}\right], C_i = \left[\begin{array}{c} 1 \\ 1 \end{array}\right]. \tag{28}$$

In the preceding equation, we have $k = 5$ is the spring constant, $\mu_1 = \mu_N = 1$ and $\mu_i = 2$ for $i = 2, \cdots, N-1$. Also $\nu_{ij} = 1$ for $i \notin \{1, N\}$ and $j \in \{i-1, i+1\}$ and $\nu_{12} = \nu_{N,N-1} = 1$. Otherwise, $\nu_{ij} = 0$.

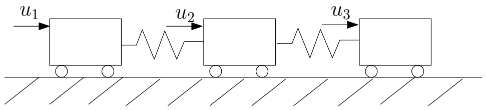

Fig. 1. Three carts coupled by springs

The control input of subsystem i is

$$u_i = K_{ii}\hat{x}_i + K_{i,i-1}\hat{x}_{i-1} + K_{i,i+1}\hat{x}_{i+1},$$

where $K_{11} = K_{NN} = \left[\begin{array}{cc} -4 & -6 \end{array}\right]$, $K_{ii} = \left[\begin{array}{cc} 1 & -6 \end{array}\right]$ for $i = 2, \cdots, N-1$, and $K_{i,i-1} = K_{i,i+1} = \left[\begin{array}{cc} -5 & 0 \end{array}\right]$ except that $K_{10} = K_{N,N+1} = 0$.

We considered the case with $N = 3$. We set $\delta = 100$ and $\beta = 1$. Local LMIs were solved using MATLAB toolbox. With $\rho_i = 0.2$ for $i = 1, 2, 3$, the triggering events are

$$\begin{aligned}
&-0.2\|x_i(t)\|_2 + 2.31\|\hat{e}_i^j(t)\|_2 = 0, \text{ for } i = 1, 3 \\
&-0.2\|x_i(t)\|_2 + 3.04\|\hat{e}_i^j(t)\|_2 = 0, \text{ for } i = 2
\end{aligned}$$

according to equation (19) and the MANSDs for agents are all 3 according to equation (21). In the following simulation, we assume that the number of successive data dropouts in each agent is equal to that agent's MANSD.

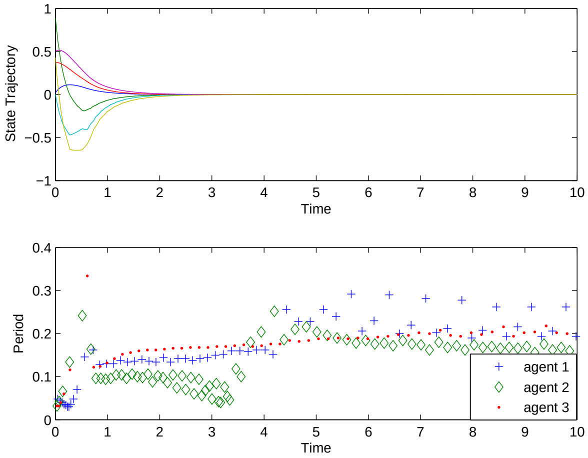

Fig. 2. State trajectory and successful broadcast periods for an event-triggered NCS

We ran the event-triggered NCS for 10 seconds without disturbances. The initial state x_0 was randomly generated satisfying $\|x_0\|_\infty \leq 1$. The numbers of triggered broadcasts by agent 1, 2, 3 are 244, 303, 231, respectively. The numbers of successful broadcasts by agent 1, 2, 3 are 61, 75, 57, respectively. From the top plot of figure 2, we can see that the system is asymptotically stable. The successful broadcast periods $(b^i_{k+1} - b^i_k)$ of the agents are shown in the bottom plot of figure 2 that vary in a wide range. It demonstrates the ability of event-triggering in adjusting sampling periods in response to variations in the system's states.

We then ran the system for 40 seconds with $N = 10$ and added an external disturbance into agent 1, where $|w_1(t)| \leq 0.1$ for $t \in [3,7]$ and $w(t) = 0$ otherwise. The successful broadcast periods of agent 1 (cross), 7 (diamond), 10 (dot) are plotted in Figure 3. We see from the figure that agent 1's broadcast periods become short when the disturbance comes in during $t \in [3,7]$. It is because event-triggering can adjust the agent's broadcast periods in response to variations in the system's external inputs. Although no disturbance directly comes into agent 7, its periods are also reduced several seconds after the disturbance comes into agent 1. So are the periods of agent 10. This is because the effect of the disturbance into agent 1 is passed to each agent, from 1 to 10. Agent 2 is affected by the changes in agent 1; agent 3 is affected by the changes in agent 2, and so on. The spatial distance causes a time delay. Therefore the periods of agents are largely reduced during different time intervals.

VII. Conclusions

This paper studies event-triggered NCS. We provide a sufficient condition for the existence of local events such that the resulting event-triggered NCS is $\mathcal{L}_2$ stable. Based on this condition, we propose a distributed scheme for each agent to design its local events, using linear matrix inequalities. This scheme applies to linear continuous-time systems and the resulting event-triggered system is finite-gain $\mathcal{L}_2$ stable. Moreover, we consider data dropouts in NCS and propose a distributed method that enables each agent to locally identify the maximal allowable number of its successive data dropouts without loss of the system stability.

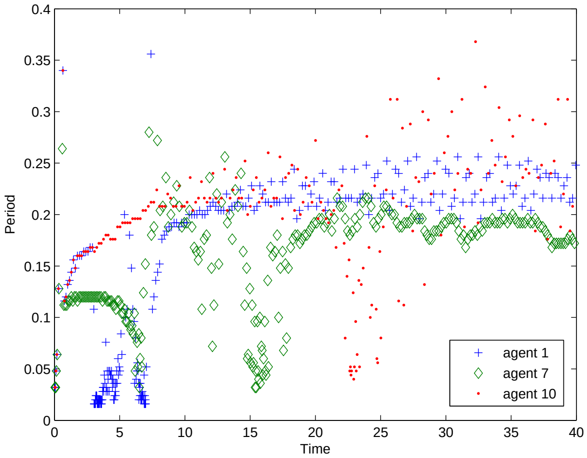

Fig. 3. Successful broadcast periods versus time in an event-triggered NCS with disturbances

References

[1] K. G. Shin, "Real-time communications in a computer-controlled workcell," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 1, pp. 105–113, 1991.

[2] W. S. Wong and R. W. Brockett, "Systems with finite communication bandwidth constraints – Part II: Stabilization with limited information feedback," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 1049–1053, 1999.

[3] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, 2002.

[4] W. Zhang, M. Branicky, and S. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, pp. 84–99, 2001.

[5] D. Nesic and A. R. Teel, "Input-output stability properties of networked control systems," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1650–1667, 2004.

[6] D. Carnevale, A. R. Teel, and D. Nesic, "Further results on stability of networked control systems: a lyapunov approach," *IEEE Transactions on Automatic Control*, vol. 52, pp. 892–897, 2007.

[7] X. Wang and M. Lemmon, "Event-triggered broadcasting across distributed networked control systems," in *American Control Conference*, 2008.

[8] ——, "Decentralized event-triggering broadcast over networked systems," in *Hybrid Systems: Computation and Control*, 2008.

[9] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," 2008, to appear in *Proceedings of the 47th Conference on Decision and Control*.

[10] X. Wang and M. Lemmon, "Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability," to appear in *IEEE Transactions on Automatic Control*.

[11] P. Tabuada, "Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks," *Automatic Control, IEEE Transactions on*, vol. 52, no. 9, pp. 1680–1685, 2007.

[12] Q. Ling and M. D. Lemmon, "Soft real-time scheduling of networked control systems with dropouts governed by a Markov chain," in *American Control Conference*, 2003.

[13] P. A. Kawka and A. G. Alleyne, "Stability and performance of packet-based feedback control over a Markov channel," in *American Control Conference*, 2006.