Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

DECENTRALIZED STABILIZATION OF NETWORKED SYSTEMS UNDER DATA-RATE CONSTRAINTS [1]

Alexey S. Matveev * **Andrey V. Savkin** **

* *Department of Mathematics and Mechanics, Saint Petersburg University, Saint Petersburg, 198504, Russia (Tel: 7-812-4284148; e-mail: alexeymatveev@hotmail.com)*
** *School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, 2052, Australia (Tel: 612-93856359; e-mail: a.savkin@unsw.edu.au)*

Abstract: The paper deals with an advanced networking scenario involving a noisy linear plant with multiple actuators and sensors connected via a complex communication network with varying topology. The network contains a large number of spatially distributed nodes equipped with CPU's that process and transmit data. Processing algorithms need to be designed for some nodes, whereas they are fixed for other nodes. During transmission, data may incur delays, be lost or corrupted. The objective is to stabilize the plant. A necessary and sufficient condition for stabilizability is given in terms of the so-called rate (capacity) domain of a communication network. It is shown that the problem of networked stabilization of noisy plants is ultimately reduced to a hard long-standing one of the information theory: calculating the capacity domains of communication networks.

Keywords: Networked Systems; Control over networks; Distributed control.

1. INTRODUCTION

The field of control under communication constraints has attracted much attention due to the emergence of numerous applications where bandwidth communication constraints become a real concern. This motivates the development of a new chapter of control theory that deals with networked control systems and combines together the control and communication issues, taking into account all the limitations on communication between sensors, controllers, and actuators.

In determining the minimum data rates required for stabilization, fundamental advances have been made for simplest networks with single "sensor-controller" and "controller-actuator" channels (see e.g., (Baillieul, 1999), (Wong and Brockett, 1999), (Nair and Evans, 2004) and the literature therein). Other references on this subject include (Petersen and Savkin, 2001), (Savkin, 2006), (Matveev and Savkin, 2007). However many modern control systems are implemented in a decentralized fashion, which results in a less trivial topology with multiple spatially distributed sensors, controllers, and actuators communicating over a serial digital network. This paper attempts to extend the above fundamental results on this situation.

The case of multiple sensors and controllers, where each sensor is linked with every controller by a perfect channel with time-varying finite data rate, was studied in (Nair *et al.*, 2004) for real-diagonalizable systems. Separate necessary and sufficient conditions for stabilizability were obtained. In general, they are not tight and become tight if the system is stabilizable by every actuator and detectable by every sensor. In (Matveev and Savkin, 2005*b*), tight conditions for stabilizability of noise-less linear multiple sensor systems were ob-

[1] Partially supported by the Australian Research Council and the Russian Foundation for Basic Research grant 06-08-01386.

tained in the case of delayed and lossy channels and not necessarily diagonalizable systems. In (LaScala and Evans, 2005), minimum data rates required for reconstruction of the track estimates of the state of a Gauss-Markov process to a given level of accuracy are presented. The paper (Matveev and Savkin, 2005*a*) describes the stabilizability region for linear noise-less plants with multiple sensors and controllers connected via a general topology network.

This paper extends these results on the case of noisy sensors, uncertain plants with bounded additive disturbance, and networks with after-effects. We also improve the result of (Matveev and Savkin, 2005*a*) by showing that not a novel concept of capacity domain introduced there but one closer to the classic concept underlies the description of the stabilizability domain.

We consider a general finite-capacity network with spatially distributed communicating elements. Every element has a computing capability and converts the incoming information steams into outgoing data flows. Some elements (*sensors*) can also partially observe an outer unstable process. Some other elements (*actuators*) are able to directly affect the process. The remaining elements act as intermediate *controllers* taking part in converting the sensor data into controls in a decentralized and distributed fashion. The data processing algorithms (DPA) at the elements should be designed to make the closed-loop system stable.

The network is given: it is indicated between which elements data can be communicated, at which rates, and in which way. Transmitted messages may incur delays, be lost and corrupted, interfere, and collide. So the transmission result may depend on the packets dispatched from many elements. The network topology is arbitrary and may be dynamically altered by authorized elements. A simple example is a switch of a communication channel from service (connection) of one pair of elements to service of another pair.

Arbitrary restrictions are imposed on DPA of the controllers. They may be due to limitations on the memory size, variety of executable operations, CPU speed etc. Not excluded is the case of a single feasible DPA. An example is a data storage and channel dynamically served and switched, respectively, in accordance with given protocols. We yet assume that any causal DPA can be implemented at the sensors and actuators.

The outer discrete-time linear process is affected by a bounded additive disturbance; there are bounded noises in the sensors. The objective is to find conditions under which the elements can be equipped with feasible DPA so that the closed-loop system is stable.

We give an evidence that this problem is reduced to the standard problem of the information sciences: finding the capacity domain. This domain describes the rates at which data can be sent from an element (network node) or set of nodes to another node or set of nodes. It is shown that stabiizability holds if and only if a certain vector characterizing the rate of instability of the outer process belongs to the interior of the above domain and its closure, respectively. Design of the stabilizing algorithm is ultimately reduced to invention of the block code transmitting data at the rates matching the entries of this vector.

The capacity domain of the primal network cannot be put in use here, and an imaginary network should be employed. It results from the original one by including artificial nodes and infinite capacity channels. First, included are channels from every actuator to all sensors influenced by it. These channels express the view of the control loop as a communication facility. The channels of the second kind broadcast messages from artificial nodes, each associated with an unstable mode of the process, to all sensors detecting the mode. Third, included are additive interference channels delivering data to another set of artificial nodes, each associated with an unstable mode: the channel collects data from all actuators that affect the mode. For these artificial nodes, DPA are limited to projecting the received real signal onto the integer grid. The employed capacity domain answers the question: How much data can be transmitted between every two artificial nodes associated with a common unstable mode?

The body of the paper is organized as follows. Sections 2 and 3 present the network model and the problem statement, respectively. Section 4 describes the auxiliary network. Section 5 recalls the notion of the network capacity domain. In Sections 6 and 7, we state the assumptions about the network and the main result, respectively. Section 8 offers illustrative examples.

2. GENERAL MODEL OF THE NETWORK

We consider a network with the set of nodes (elements) $\mathfrak{H}$ partitioned into subsets $\mathfrak{H}_s, \mathfrak{H}_a$, and $\mathfrak{H}_c$ of sensors, actuators, and controllers, respectively. The element h receives signals from some other elements and if $h \in \mathfrak{H}_s$, has an access to a measurement $y_h \in \mathbb{R}^{n_{y,h}}$ of an outer process. These signals constitute the *inner input* $i_h \in \mathfrak{I}_h$; the *entire input* $\widehat{i}_h := i_h$ if $h \notin \mathfrak{H}_s$ and $\widehat{i}_h = [i_h, y_h]$ otherwise. The element h also emits signals to some other elements, these signals constitute its *inner output* $o_h \in \mathfrak{O}_h$. Any actuator $h \in \mathfrak{H}_a$ also produces a control $u_h \in \mathbb{R}^{n_{u,h}}$. The *entire output* $\widehat{o}_h := o_h$ if $h \notin \mathfrak{H}_a$ and $\widehat{o}_h = [o_h, u_h]$ otherwise. The alphabets $\mathfrak{I}_h$ and $\mathfrak{O}_h$ are given. For example, if i_h arrives over a perfect channel with capacity $\mathfrak{c}$, then $\mathfrak{I}_h$ is the channel alphabet of the size $2^{\mathfrak{c}}$. If data o_h are emitted into k channels with respective alphabets $\mathfrak{C}_1, \dots, \mathfrak{C}_k$, then $\mathfrak{O}_h = \mathfrak{C}_1 \times \cdots \times \mathfrak{C}_k$.

Now we introduce four united ensembles of the outer and inner inputs and outputs, respectively:

$$\boldsymbol{Y} := \{y_h\}_{h \in \mathfrak{H}_s}, \; \boldsymbol{U} := \{u_h\}_{h \in \mathfrak{H}_a}, \; \boldsymbol{S} := \{s_h\}_{h \in \mathfrak{H}},$$

$S = O, I$. The roles of the network and control strategy (endowing every h with a DPA) are as follows

$$\boldsymbol{O} \xrightarrow{\text{network}} \boldsymbol{I}, \qquad [\boldsymbol{I}, \boldsymbol{Y}] \xrightarrow{\text{control strategy}} [\boldsymbol{O}, \boldsymbol{U}]. \quad (1)$$

The first transformation is governed the *network state* $\boldsymbol{N}$ from a given alphabet $\mathfrak{N}$. The computing capability of any element h is specified by 1) the given *memory alphabet* $\mathfrak{M}_h$ with elements m_h representing the memory content and 2) the set $\mathfrak{A}_h$ of admissible DPA:

$$\mathcal{A}_h \equiv [\mathcal{O}_h(\cdot), \mathcal{M}_h(\cdot), m_h^0] \in \mathfrak{A}_h. \quad (2)$$

Here m_h^0 is the initial memory content and

$$o_h(t) = \mathcal{O}_h[m_h(t), \widehat{i}_h(t), t], \qquad m_h(t+1) = \mathcal{M}_h[m_h(t), \widehat{i}_h(t), t].$$

By putting $\boldsymbol{M} := \{m_h\}_{h\in\mathfrak{H}}$ and

$$\text{(control strategy)} \sim \boldsymbol{\mathcal{A}} := \{\mathcal{A}_h\}_{h\in\mathfrak{H}},$$

relations (1) can be specified as follows:

$$\boldsymbol{O} \xrightarrow[\boldsymbol{N}]{\text{network}} \boldsymbol{I}, \quad [\boldsymbol{I}, \boldsymbol{Y}] \xrightarrow[\boldsymbol{M}]{\boldsymbol{\mathcal{A}}} [\boldsymbol{O}, \boldsymbol{U}],$$
$$\boldsymbol{O} \xrightarrow[\boldsymbol{N}]{\text{network}} \boldsymbol{N}^+, \quad [\boldsymbol{I}, \boldsymbol{Y}] \xrightarrow[\boldsymbol{M}]{\boldsymbol{\mathcal{A}}} \boldsymbol{M}^+,$$

where $^+$ denotes the value at the next time instant. For deterministic networks, these relations take the form

$$\begin{aligned} \boldsymbol{I}^+ &= \mathcal{I}[\boldsymbol{O}(t), \boldsymbol{N}(t)], \boldsymbol{N}^+ = \mathcal{N}[\boldsymbol{O}(t), \boldsymbol{N}(t)], \\ \boldsymbol{I}(0) &= \boldsymbol{I}_0, \boldsymbol{N}(0) = \boldsymbol{N}_0, \boldsymbol{M}(0) = \mathcal{M}_{\boldsymbol{\mathcal{A}}}^0, \\ \boldsymbol{O}(t) &= \mathcal{O}[\boldsymbol{I}(t), \boldsymbol{Y}(t), \boldsymbol{M}(t), \boldsymbol{\mathcal{A}}, t], \\ \boldsymbol{U}(t) &= \mathcal{U}[\boldsymbol{I}(t), \boldsymbol{Y}(t), \boldsymbol{M}(t), \boldsymbol{\mathcal{A}}, t], \\ \boldsymbol{M}^+ &= \mathcal{M}[\boldsymbol{I}(t), \boldsymbol{Y}(t), \boldsymbol{M}(t), \boldsymbol{\mathcal{A}}, t], \boldsymbol{\mathcal{A}} \in \mathfrak{A}, \end{aligned} \quad (3)$$

where the fifth equation results from (2).

Formally, the network is a facility transforming the time sequence of outer inputs $\boldsymbol{Y}(0), \boldsymbol{Y}(1), \boldsymbol{Y}(2), \dots$ into the sequence of outer outputs $\boldsymbol{U}(0), \boldsymbol{U}(1), \boldsymbol{U}(2), \dots$ in accordance with (3), where the *data processing strategy* (DPS) $\boldsymbol{\mathcal{A}} \in \mathfrak{A}$ is a control parameter. All functions denoted by capital script letters, the set $\mathfrak{A}$, the network initial state $\boldsymbol{N}_0$ and $\boldsymbol{I}_0$, and the sets within which the variables from (3) range are given.

Examples The state $\boldsymbol{N}$ may represent the memories of the channels and the network mode. For instance, the memory of the d-delayed channel $n(t) := [e(t-1), \dots, e(t-d+1)]$ and the messages r and e at its receiving and transmitting ends are related by

$$r(t+1) = \mathcal{R}[n(t), e(t)] \quad (:= e[t-d+1]),$$
$$n(t+1) = \mathcal{N}[n(t), e(t)] \quad (:= [e(t), \dots, e(t-d+2)]).$$

The mode may determine the current topology of the network (e.g., which elements are linked by switching channels) and parameters of the communication medium (e.g., capacity, level of data distortion, etc.). For example, there may be a choice between larger channel alphabets with more message corruption and smaller alphabets with lesser corruption.

For the element h with b-bit memory, $\mathfrak{M}_h$ has the size 2^b. For CPU with limited computational power, $\big(\mathcal{O}_h[m, \widehat{i}, t], \mathcal{M}_h[m, \widehat{i}, t]\big) \in \mathfrak{R}_h(m, \widehat{i}) \ \forall m, \widehat{i}, t$, where $\mathfrak{R}(m, \widehat{i})$ is the known set of all possible results that may be obtained by means of this CPU for the unit time from m and $\widehat{i}$. In the case of unlimited memory and the computational power, $\mathfrak{M}_h$ is the set of all finite sequences of inputs $S = [\widehat{i}^0, \widehat{i}^1, \dots, \widehat{i}^k]$ including the "empty" one $\circledast$. The set $\mathfrak{A}_h$ consists of $\mathcal{A}_h$ for which $\circledast$ is the initial state and the function $\mathcal{M}_h(\cdot)$ of $S, \widehat{i}$ acts by adding $\widehat{i}$ to S from the right (and possible dropout of several entries of S). If any entry outside the group of k concluding ones should be dropped necessarily, we deal with an element with the memory of length k.

For more detailed discussion and further examples, we refer the reader to the forthcoming monograph (Matveev and Savkin, 2008, Ch.9).

3. STATEMENT OF THE STABILIZATION PROBLEM

We consider linear discrete-time multiple sensor and actuator systems of the form:

$$x^+ = Ax(t) + \sum_{i=1}^{l} B_i u_i(t) + \xi(t); \ x(0) = x^0; \quad (4)$$

$$y_j(t) = C_j x(t) + \chi_j(t), \qquad j = 1, \dots, k. \quad (5)$$

Here $x \in \mathbb{R}^{n_x}$ is the state, $u_i \in \mathbb{R}^{n_{u,i}}$ is the output of the ith actuator, $y_j \in \mathbb{R}^{n_{y,j}}$ is the observation performed by the jth sensor, and $\xi(t), \chi_j(t), x^0$ are bounded

$$\|\xi(t)\| \le D, \ \|\chi_j(t)\| \le D_j^y \, \forall t, j, \ \|x^0\| \le D_x \quad (6)$$

exogenous disturbance, noise in the jth sensor, and initial state, respectively. The system is unstable: there is an eigenvalue λ of A with $|\lambda| \ge 1$.

The objective is to stabilize the plant by means of multiple decentralized controllers. Along with sensors and actuators, they are spatially distributed and constitute the set of nodes of a given network **NW** described in the previous section. So DPS $\boldsymbol{\mathcal{A}}$, sequences of noises $\{\xi(t)\}, \{\chi_j(t)\}$, and initial state x^0 uniquely determine a process in the closed-loop system.

Definition 1. A DPS $\boldsymbol{\mathcal{A}}$ is said to *stabilize* the system if it keeps the stabilization error bounded:

$$\sup_t \sup_{\{\xi(\theta)\}, \{\chi_j(\theta)\}, x^0} \|x(t)\| < \infty, \quad (7)$$

and to *regularly stabilize* it, if (7) also holds with $u_i(t)$ substituted in place of $x(t)$ for any i. In (7) the second sup is over noises and initial states satisfying (6).

Which networks fit to stabilize a given unstable linear plant under a proper design of the control strategy?

The main result of the paper is that this question is reducible to the following standard question studied in the traditional information sciences.

Q) *How much data may be communicated from the input to output nodes across the network that results from a certain extension of the original one?*

All limitations on communication and DPA are inherited by this extension. So **Q)** concerns communication under the restrictions and with the features (e.g., channel switching) inherent in the primal network.

4. EXTENDED NETWORK

This network is introduced via three steps.

Control Based Extension (CBE). As is shown in e.g., (Matveev and Savkin, 2006), as much information as desired may be communicated via the plant from every actuator to any sensor capable of detecting its actions. We first explicitly express this possibility by formal adding several new channels to the network. After this, the network is studied out of connection with the plant.

Let L_i^{+c} and L_j^{-o} be the subspaces of states that are controllable and non-observable by the ith actuator and jth sensor, respectively. The set of actuator-sensor pairs communicating via the plant is

$$\mathrm{CVP} := \left\{(i,j) : L_i^{+c} \not\subset L_j^{-o}\right\}.$$

For every such pair, let us link the ith actuator to the jth sensor by the noise-less channel with the infinite alphabet ($\mathbb{R}$ for the definiteness). Since control u_i influences y_j with the delay $d_{i\to j} := \min\{d \geq 0 : C_j A^d B_i \neq 0\} + 1$, this channel is taken to be $d_{i\to j}$-delayed. Insertion of all these channels gives rise to the *control based extension* **CBE(NW)** of **NW**.

To simplify the matters, we assume that the unstable $|\lambda| \geq 1$ eigenvalues of A are distinct. A proper linear change of the variables shapes the state x into

$$x = \Big(\underbrace{x_{-s},\ldots,x_0}_{\sim M_{st}(A)},\underbrace{x_1,\ldots,x_{g_1}}_{\sim M_1},\underbrace{\ldots}_{\sim M_2},\ldots,\underbrace{\ldots x_{n^+}}_{\sim M_g}\Big), \tag{8}$$

where $M_{st}(A)$ is the stable subspace of A and M_ν is the invariant subspace related to either a real unstable eigenvalue λ_ν or a couple of conjugate complex ones $\lambda_\nu \neq \overline{\lambda}_\nu$. With a slight abuse of terminology, $x_\alpha, \alpha \geq 1$ are called *unstable modes*. Any such mode is associated with unique $M_\alpha = M_\nu$ and $|\lambda_\alpha| := |\lambda_\nu|$.

We assume that unlike the controllers, any causal DPA can be implemented at the sensors and actuators (see Assumption 1 further). Then the network can accept exterior inputs y_h at the sensor nodes and produce exterior outputs u_h at the actuator ones in any form.

Mode-wise Prefix. Now we change the scheme of data injection into **CBE(NW)**. We introduce new outer data sources, each associated with a particular unstable mode x_α and hosted by a new artificial input node N_α^{in}. This node accepts any causal DPA and instantaneously broadcasts data via a perfect infinite alphabet channel to all sensors j that observe this mode $M_\alpha \cap L_j^{-o} = \{0\}$. The ensemble of these new channels and nodes $\mathbf{PREF_{mw}}$ is called the *mode-wise prefix*; the resultant network is denoted by $\mathbf{PREF_{mw}} \boxplus \mathbf{CBE(NW)}$.

Mode-wise Suffix is used to change the scheme of data emission. The new output nodes N_α^{out} are still associated with unstable modes x_α and attached to $\mathbf{PREF_{mw}} \boxplus \mathbf{CBE(NW)}$ via real additive interference channels. The channel to N_α^{out} collects real signals from all actuators i that control the mode $M_\alpha \subset L_i^{+c}$ and instantaneously delivers the sum of signals to N_α^{out}. DPA at N_α^{out} is limited to the projection of the received real signal into the nearest integer ($i + 1/2 \mapsto i$ for integer i). The *mode-wise suffix with quantization* $\mathbf{SUFF^q_{mw}}$ is the ensemble of all these channels and nodes. The resultant network is denoted by

$$\mathbf{PREF_{mw}} \boxplus \mathbf{CBE(NW)} \boxplus \mathbf{SUFF^q_{mw}}. \tag{9}$$

We stress that $N_\alpha^{\text{in}} \neq N_\alpha^{\text{out}}$. This is motivated by the double role of every unstable mode: it is simultaneously an object of observation and control.

5. NETWORK CAPACITY DOMAIN

Answers to the question **Q)** are traditionally given in terms of the so-called capacity domain. Now we recall this notion with respect to the network (9).

Let every node N_α^{in} host an informant $\mathfrak{I}_\alpha$ producing a message $\mathfrak{y}_\alpha$, which serves as the outer input for N_α^{in}. This message should be transmitted to N_α^{out}, where it appears in the form of the outer output $\mathfrak{u}_\alpha$ of this node. The transmission is arranged by choosing a DPS for the network. It includes a DPS $\boldsymbol{A} \in \mathfrak{A}$ for the original network and DPA at the artificial nodes.

A *networked block code* with block length T is a DPS that acts during the time interval $[0 : T]$, serves informants producing constant message sequences $\mathfrak{y}_\alpha(t) \equiv \mathfrak{y}_\alpha \in [1 : F_\alpha]$ $\forall t$ and generates the outputs in the matching form $\mathfrak{u}_\alpha(t) \in [1 : F_\alpha] \cup \{\circledast\}$ ($\circledast$ means "no decision".) The *rate vector* of this code is

$$\boldsymbol{r}_{\text{code}} := \left(\frac{\log_2 F_1}{T+1}, \ldots, \frac{\log_2 F_{n^+}}{T+1}\right). \tag{10}$$

A networked block code is *errorless* if at the terminal time it correctly recognizes the messages from all informants $\mathfrak{u}_\alpha(T) = \mathfrak{y}_\alpha$ $\forall \alpha$ irrespective of which messages $\mathfrak{y}_\alpha \in [1 : F_\alpha]$ were dispatched. A vector $\boldsymbol{r} \in \mathbb{R}^{n^+}$ is called the *achievable rate vector* if for arbitrarily large $\overline{T}$ and small $\epsilon > 0$, there exists an errorless networked block code with block length $T \geq \overline{T}$ whose rate vector (10) approaches $\boldsymbol{r}$ with accuracy ϵ, i.e., $\|\boldsymbol{r}_{\text{code}} - \boldsymbol{r}\| < \epsilon$. The *capacity (rate) domain* **CD** is the set of all achievable rate vectors.

6. FORMAL ASSUMPTIONS

To get substantial results, we need more assumptions about the network. By the first of them, we restrict attention to sensors and actuators with unlimited memories and computational powers, whereas arbitrary restrictions may be imposed on the controllers.

Let $\boldsymbol{O}_c, \boldsymbol{I}_c, \boldsymbol{M}_c$ stand for the ensembles of outputs, inputs, and memories of all controllers, respectively.

Assumption 1. The following statements hold:

A) DPA part of equations (3) disintegrate into separate equations concerning each sensor, each actuator and the rest of the network, respectively:

$$\widehat{o}_h(t) = \mathcal{O}_h[\widehat{i}_h(t), m_h(t), \boldsymbol{\mathcal{A}}, t],$$
$$m_h^+ = \mathcal{M}_h[\widehat{i}_h(t), m_h(t), \boldsymbol{\mathcal{A}}, t],$$
$$m_h(0) = \mathcal{M}_h^0(\boldsymbol{\mathcal{A}}) \qquad \forall h \in \mathfrak{H}_s \cup \mathfrak{H}_a; \quad (11)$$

$$\left.\begin{array}{r} \boldsymbol{O}_c(t) = \mathcal{O}_c[\boldsymbol{I}_c(t), \boldsymbol{M}_c(t), \boldsymbol{\mathcal{A}}, t] \\ \boldsymbol{M}_c^+ = \mathcal{M}_c[\boldsymbol{I}_c(t), \boldsymbol{M}_c(t), \boldsymbol{\mathcal{A}}, t] \end{array}\right| \begin{array}{l} \boldsymbol{M}_c(0) = \\ = \mathcal{M}_0^c(\boldsymbol{\mathcal{A}}) \end{array}. \quad (12)$$

B) The right hand sides in (11) independently range over all functions (of the arguments from (11) except for $\boldsymbol{\mathcal{A}}$) as $\boldsymbol{\mathcal{A}}$ runs over $\mathfrak{A}$ even if the run of $\boldsymbol{\mathcal{A}}$ is such that equations (12) are kept unchanged.

C) For any $h \in \mathfrak{H}_s \cup \mathfrak{H}_a$, the memory alphabet $\mathfrak{M}_h$ is infinite ($\mathfrak{M}_h = \mathbb{R}$ for the definiteness).

The next assumption means that the network is stationary and can be reset to the initial state.

Assumption 2. For any $\boldsymbol{\mathcal{A}} \in \mathfrak{A}$ and time T, there exists a DPS $\boldsymbol{\mathcal{A}}_{\text{res}} \in \mathfrak{A}$ that is identical to $\boldsymbol{\mathcal{A}}$ until $t = T$ and resets the network to the initial state $\boldsymbol{N}(T_*) = \boldsymbol{N}_0, \boldsymbol{I}(T_*) = \boldsymbol{I}_0, \boldsymbol{M}(T_*) = \mathcal{M}^0(\boldsymbol{\mathcal{A}})$ at a time $T_* > T$ such that $T_* - T \leq \delta T_{\max}$, where $\delta T_{\max}$ does not depend on $\boldsymbol{\mathcal{A}}$. Moreover, the memory content $\boldsymbol{M}$ can be driven to the state $\mathcal{M}^0(\boldsymbol{\mathcal{A}}_1)$ initial for any other strategy $\boldsymbol{\mathcal{A}}_1 \in \mathfrak{A}$ that is equivalent to $\boldsymbol{\mathcal{A}}$ modulo a given finite partition (i.e., $\exists \nu : \boldsymbol{\mathcal{A}}_1, \boldsymbol{\mathcal{A}} \in \mathfrak{M}_\nu$) of the memory alphabet $\{\boldsymbol{M}\} = \mathfrak{M} = \mathfrak{M}_1 \cup \ldots \cup \mathfrak{M}_Q$.

Finally, DPS can be concatenated and periodically extended from a finite time interval.

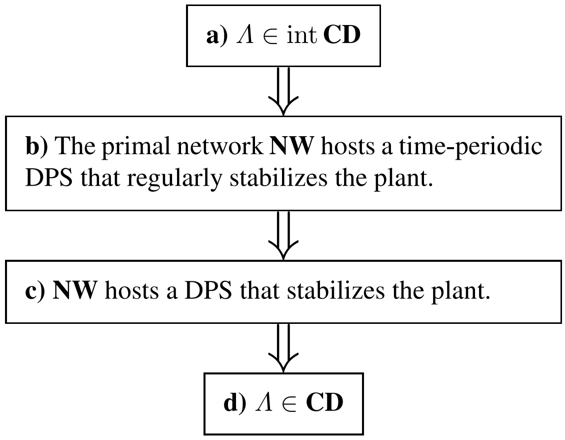
Assumption 3. For any $\boldsymbol{\mathcal{A}}_i \in \mathfrak{A}, i = 1, 2$ and time $T = 0, 1, \ldots$ such that the network driven by $\boldsymbol{\mathcal{A}}_1$ arrives at time T at the state initial for $\boldsymbol{\mathcal{A}}_2$, there exists a DPS $\boldsymbol{\mathcal{A}} \in \mathfrak{A}$ that is identical to $\boldsymbol{\mathcal{A}}_1$ and $\boldsymbol{\mathcal{A}}_2$ on the time interval $[0 : T-1]$ and afterwards, respectively.

Assumption 4. For any $\boldsymbol{\mathcal{A}} \in \mathfrak{A}$ and time $\tau = 1, 2, \ldots$ such that the network returns to the initial state at time τ, there exists a τ-periodic (in time) DPS $\boldsymbol{\mathcal{A}}_{\text{per}} \in \mathfrak{A}$ that is identical to $\boldsymbol{\mathcal{A}}$ on the time interval $[0 : \tau - 1]$.

7. CRITERION FOR STABILIZABILITY

The symbol $\operatorname{int} B$ stands for the interior of the set B.

Theorem 1. We consider the capacity domain **CD** of the extended network (9) and the representation (8), associate any unstable mode $x_\alpha, \alpha \geq 1$ with the related eigenvalue modulus $|\lambda_\alpha|$, and denote $\Lambda := \operatorname{col}\Big(\log_2 |\lambda_1|, \ldots, \log_2 |\lambda_{n^+}|\Big)$. Then



The proof and the description of a stabilizing DPS will be given in (Matveev and Savkin, 2008, Ch.9).

Theorem 1 is true not only if the unstable eigenvalues are distinct. It remains valid under more general assumptions adopted in (Matveev and Savkin, 2005*b*).

The basic inclusion $\Lambda \in \mathbf{CD}$ can often be simplified by passing to vector and domain of a lesser dimension. Some general facts underlying this operation are presented in (Matveev and Savkin, 2008, Ch.9).

8. TWO SENSORS AND ACTUATORS

Now we offer examples illustrating a possible final form of the criterion obtained by means of Theorem 1. In these examples, the rate domains were calculated via showing that the communication network at hand can be interpreted as a fluid transportation facility.

We consider the linear plant (4), (5) with two sensors S1, S2 and actuators A1, A2 ($l = k = 2$), which are directly linked by noise-less channels with given finite capacities. The absence of the channel from Sj to Ai is modeled by annihilating its capacity $c_{ji} := 0$. The plant is detectable and stabilizable by the entire sets of the sensors and actuators, respectively.

Actuators with Non-Intersecting Zones of Influence. We first assume that the actuators affect no common unstable mode. Then the unstable subspace

$$M_{unst}(A) = M_{11} \oplus M_{\mathbf{b}1} \oplus M_{21} \oplus M_{12} \oplus M_{\mathbf{b}2} \oplus M_{22},$$

- where $M_{ji}, j \neq \mathbf{b}$ is the subspace of states controllable by Ai, observable by the jth sensor, and non-observable by the companion sensor;
- $M_{\mathbf{b}i}$ is the subspace of states controllable by the ith actuator and observable by both sensors.

We denote $\det A|_{\{0\}} := 1$ and $\boldsymbol{e}_{\nu i} := \log_2 |\det A|_{M_{\nu i}}|$, where $\nu = 1, 2, \mathbf{b}, i = 1, 2$.

For 2×2-matrices $M = (m_{ij})$, we put:

- $[M]^{\leftrightarrow} := m_{11} + m_{12}$;
- $[M]_{\leftrightarrow} := m_{21} + m_{22}$;
- ${}_{\updownarrow}[M] := m_{11} + m_{21}$;
- $[M]_{\updownarrow} := m_{12} + m_{22}$;
- $\Sigma_{ij}^{-}(M) := \sum_{(i',j') \neq (i,j)} m_{i',j'}$.

Finally, we introduce the matrices

$$E := \begin{pmatrix} \boldsymbol{e}_{11} & \boldsymbol{e}_{12} \\ \boldsymbol{e}_{21} & \boldsymbol{e}_{22} \end{pmatrix}, \qquad C := \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}.$$

Proposition 1. Suppose that any sensor is able to detect the actions of every actuator: $L_i^{+c} \not\subset L_j^{-o} \; \forall i, j = 1, 2$. If the plant is stabilizable, then

$$\begin{gathered} [E]^{\leftrightarrow} \leq [C]^{\leftrightarrow}, \quad [E]_{\leftrightarrow} \leq [C]_{\leftrightarrow} \\ {}_{\updownarrow}[E + \mathbf{diag}(\boldsymbol{e}_{\mathbf{b}1}, \boldsymbol{e}_{\mathbf{b}2})] \leq_{\updownarrow} [C], \\ [E + \mathbf{diag}(\boldsymbol{e}_{\mathbf{b}1}, \boldsymbol{e}_{\mathbf{b}2})]_{\updownarrow} \leq [C]_{\updownarrow} \\ \Sigma_{ij}^{-}(E) + \boldsymbol{e}_{\mathbf{b}j'} \leq \Sigma_{ij}^{-}(C) \qquad \forall i, j = 1, 2, \end{gathered} \tag{13}$$

where $j' := 1$ if $j = 2$ and $j' = 2$ if $j = 1$.

Conversely, if relations (13) hold with the strict inequality signs, the plant is regularly stabilizable.

Two Independent Agents. The zones of influence of the actuators are still disjoint but unlike Proposition 1, every sensor is affected by a single actuator. By changing enumeration of the sensors, one can assume that Si detects the actions of only Ai. Then $M_{unst}(A) = M_1 \oplus M_2$, where M_i is the subspace of states controllable by Ai, observable by Si, and non-observable by Si'. The situation can be interpreted as if there are two independent agents with the state spaces M_1 and M_2, respectively, each equipped with its own actuator and sensor measuring the state of the owner. It may seem that then the cross channels S1 $\mapsto$ A2 and S2 $\mapsto$ A1 are useless, and the conditions for stabilizability come to $\log_2 |\det A|_{M_i}| \leq c_{ii} \forall i$. The next proposition shows that this is not the case, and the system can be stabilized even if $c_{11} = c_{22} = 0$.

Proposition 2. If the plant is stabilizable, then

$$\begin{gathered} \log_2 |\det A|_{M_i}| \leq c_{ii} + \min\{c_{12}, c_{21}\} \; i = 1, 2, \\ \log_2 |\det A|_{M_{unst}(A)}| \leq c_{11} + c_{22} + \min\{c_{12}, c_{21}\}. \end{gathered}$$

Conversely, if these relations hold with the strict inequality signs, the plant is regularly stabilizable.

Actuators with Identical Zones of Influence, i.e., the plant is stabilizable by any actuator. Then

$$M_{unst}(A) = M_1 \oplus M_{\mathbf{b}} \oplus M_2,$$

where $M_j, j \neq \mathbf{b}$ is the subspace of states observable by Sj and non-observable by Sj', and $M_{\mathbf{b}}$ is the set of states observable by both sensors.

Proposition 3. If the plant is stabilizable, then

$$\begin{gathered} \log_2 |\det A|_{M_i}| \leq c_{i1} + c_{i2} \quad i = 1, 2, \\ \log_2 |\det A|_{M_{unst}(A)}| \leq c_{11} + c_{22} + c_{12} + c_{21}. \end{gathered}$$

Conversely, if these relations hold with the strict inequality signs, the plant is regularly stabilizable.

REFERENCES

Baillieul, J. (1999). Feedback designs for controlling device arrays with communication channel bandwidth constraints. In: *Proc. of ARO Workshop on Smart Structures*. Penns. Uni., Univ. Park, P.A.

LaScala, B. F. and R. J. Evans (2005). Minimum necessary data rates for accurate track fusion. In: *Proc. of the 44th IEEE CDC and ECC 2005*. Seville, Spain. pp. 6966–6971.

Matveev, A. S. and A. V. Savkin (2005*a*). Decentralized stabilization of linear systems via limited capacity communication networks. In: *Proc. of the 44th IEEE CDC and ECC 2005*. Seville, Spain. pp. 1155–1161.

Matveev, A. S. and A. V. Savkin (2005*b*). Multi-rate stabilization of linear multiple sensor systems via limited capacity communication channels. *SIAM J. on Control and Optimization* **44**(2), 584–618.

Matveev, A. S. and A. V. Savkin (2006). On a problem related to application of digital networked communication technology to stabilization of noisy plants over noisy channels. In: *Proc.of the IEEE Conf. on Cont. Appl.*. Munich. pp. 2072–2077.

Matveev, A. S. and A. V. Savkin (2007). An analogue of Shannon information theory for detection and stabilization via noisy discrete communication channels. *SIAM J. on Control and Optimization* **46**(4), 1323–1361.

Matveev, A. S. and A. V. Savkin (2008). *Estimation and Control over Communication Networks*. Birkhauser. Boston.

Nair, G. N. and R. J. Evans (2004). Stabilisability of stochastic linear systems with finite feedback data rates. *SIAM Journal on Control and Optimization* **43**(2), 413–436.

Nair, G. N., R. J. Evans and P. E. Caines (2004). Stabilizing decentralized linear systems under data rate constraints. In: *Proc. of the 43rd IEEE CDC*. Atlantis, Bahamas. pp. 3992–3997.

Petersen, I. R. and A. V. Savkin (2001). Multi-rate stabilization of multivariable discrete-time linear systems via a limited capacity comunication channel. In: *Proc. of the 40th IEEE CDC*. Orlando, Florida, USA. pp. 304–309.

Savkin, A. V. (2006). Analysis and synthesis of networked control systems: topological entropy, observability, robustness, and optimal control. *Automatica* **42**(1), 51–62.

Wong, W. S. and R. W. Brockett (1999). Systems with finite communication bandwidth constraints: Stabilization with limited information feedback. *IEEE Trans. on Autom. Control* **44**, 1049–1053.