Systems & Control Letters 94 (2016) 77–83

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Stabilization by controller networks

Shinsaku Izumi [a,*], Shun-ichi Azuma [b], Toshiharu Sugie [b]

[a] *Faculty of Computer Science and Systems Engineering, Okayama Prefectural University, 111 Kuboki, Soja, Okayama 719-1197, Japan*

[b] *Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyoku, Kyoto 606-8501, Japan*

ARTICLE INFO

Article history:
Received 14 July 2015
Received in revised form
26 May 2016
Accepted 27 May 2016
Available online 17 June 2016

Keywords:
Networked control systems
Distributed control
Cooperative control
Consensus protocols
Controller networks

ABSTRACT

This paper addresses a design method of controller networks, *i.e.*, networked controllers which cooperatively determine the control inputs by exchanging the information with their neighbors. The problem considered here is to design the controllers stabilizing the resulting feedback system for an unknown network topology. As a solution to the problem, we propose controllers such that the entire network acts as a state feedback controller through a consensus protocol, and derive gain conditions to stabilize the resulting feedback system. This enables us to obtain a controller network that is robust against uncertainties of the network topology.

1. Introduction

Wireless control systems have drawn increasing attention in recent years. The reason lies in advantages of wireless networks over wired ones. In fact, wireless technologies reduce costs and time which would be necessary for installation and maintenance of the network cables.

A typical setup of the wireless control system is composed of sensors, a controller, actuators, and communication nodes, and the control is performed over the multi-hop network. That is, the controller receives the signals from the sensors through the communication nodes, and sends the control signals to the actuators through those. From a theoretical point of view, this type of system can be categorized into the so-called networked control system by regarding the network of the communication nodes as a communication channel. For the networked control systems, a number of studies have been conducted so far, *e.g.*, estimation [1,2] and stabilization [3,4].

Meanwhile, we consider here the system in Fig. 1, composed of sensor nodes, controller nodes, and actuator nodes. In this system, each node can not only forward the information but also process it. The nodes cooperatively determine the control inputs by exchanging the information with their neighbors. The network is called here the *controller network*. The motivation for considering the controller network is that it makes the resulting control system robust against failures. In fact, the system with the multi-hop network does not work if the controller fails, but even if some controller nodes fail in the system in Fig. 1 the others compensate for the failures.

The idea of this framework has been originally proposed in [5]. The authors have proposed a design method of the controller network to stabilize a given plant. However, the method has two drawbacks. First, it is assumed there that the network topology is fixed, but the topology often changes in wireless networks in a real situation. For example, wireless interference [6] causes link failures, which results in the topology changes. If such a topology change occurs in the framework, the resulting feedback system may be unstable. Second, the method is not scalable. In fact, it discriminately deals with all the nodes, which implies that we have to implement custom algorithms for each node. Hence, it is difficult to apply the controller network to large-scale systems such as traffic systems.

Thus, in this paper, we establish a framework of robust and scalable controller networks. The control system considered here is shown in Fig. 2, which is composed of a single-input plant, sensor nodes, controller nodes, and an actuator node. For this system, we suppose that the network topology belongs to a prespecified set but the detail (which element it is) is *unknown*, and assume that all the controller nodes are the *same*. Then, our problem is to design all the nodes stabilizing the resulting feedback system. Since the detailed information on the network topology is not exploited for controller design, the solution stabilizes the feedback system even if the topology changes. Furthermore, since all the controller

* Corresponding author.
E-mail address: izumi@cse.oka-pu.ac.jp (S. Izumi).

http://dx.doi.org/10.1016/j.sysconle.2016.05.014

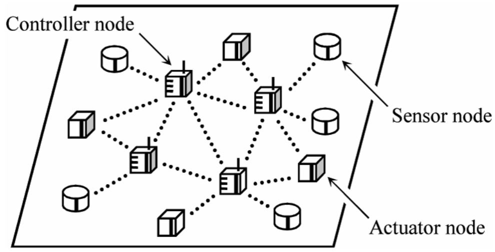

Fig. 1. Control system with controller network. The dotted lines represent the network structure.

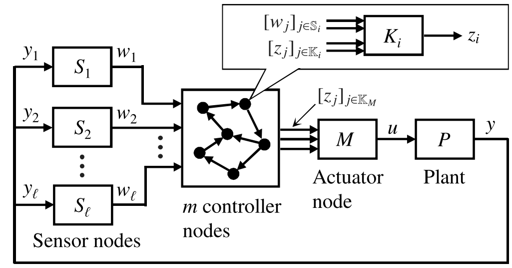

Fig. 2. Feedback system Σ.

nodes are assumed to be the same, we indiscriminately deal with them, which makes the solution scalable. Meanwhile, due to these constraints, the problem is more challenging than that in [5].

For the issue, this paper makes two contributions. First, we present a solution to the above design problem. It is given as the nodes such that the entire network acts as a state feedback controller by a consensus protocol. We then prove that the resulting feedback system is stable if the gains are appropriately chosen. The key idea is to introduce a *parameterized* coordinate transformation and characterize the parameter from the viewpoint of the stability. If a fixed coordinate transformation is applied in the same way, we can only show the stability for some specific plants. However, by introducing the parameterized coordinate transformation and reducing the stabilization problem to that of finding a range of the parameter, we can prove the stability for all controllable plants. Second, we clarify the relation between the stabilizing gain and the network topology. As a result, it is shown that the large gain will be needed for the stabilization when the network among the controller nodes is sparse. This implies that we should design the gain only for the sparsest network in the prespecified set, which substantially reduces time to develop the controller networks.

This paper is based on the conference version [7], and contains full explanations and proofs omitted there.

Notation. Let $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{C}$ be the real number field, the set of positive real numbers, and the complex number field, respectively. We denote by 1_n the $n \times 1$ vector whose elements are one. For the vectors $x := [x_1 \ \ x_2 \ \ \cdots \ \ x_n]^\top \in \mathbb{C}^n$ and $y := [y_1 \ \ y_2 \ \ \cdots \ \ y_n]^\top \in \mathbb{C}^n$, let $x \bullet y := [x_1y_1 \ \ x_2y_2 \ \ \cdots \ \ x_ny_n]^\top \in \mathbb{C}^n$. Also, let $x^{(-1)}$ be the elementwise inverse of x, *i.e.*, $x^{(-1)} := [x_1^{-1} \ \ x_2^{-1} \ \ \cdots \ \ x_n^{-1}]^\top \in \mathbb{C}^n$, and let $\mathrm{diag}(x)$ be the diagonal matrix whose i-th diagonal element is x_i. The $n \times m$ zero matrix and the $n \times n$ identity matrix are expressed by $0_{n\times m}$ and I_n, respectively. For the matrix $M \in \mathbb{C}^{n\times n}$, $\lambda_i(M) \in \mathbb{C}$ represents the eigenvalue with the i-th smallest modulus, and $v_i(M) \in \mathbb{C}^n$ expresses the eigenvector for $\lambda_i(M)$. Furthermore, $\lambda(M) \in \mathbb{C}^n$ and $V(M) \in \mathbb{C}^{n\times n}$ are defined as $\lambda(M) := [\lambda_1(M) \ \ \lambda_2(M) \ \ \cdots \ \ \lambda_n(M)]^\top$and $V(M) := [v_1(M) \ \ v_2(M) \ \ \cdots \ \ v_n(M)]$, respectively. For the matrix M, $[M]_{ij}$ denotes the (i,j)-element of M, and $\|M\|$ represents the induced ∞-norm. We use $|\mathbb{S}|$ to express the cardinality of the set $\mathbb{S}$. For the numbers $x_1, x_2, \ldots, x_n \in \mathbb{R}$ and the set $\mathbb{I} := \{i_1, i_2, \ldots, i_m\} \subseteq \{1, 2, \ldots, n\}$, let $[x_i]_{i\in\mathbb{I}} := [x_{i_1} \ \ x_{i_2} \ \ \cdots \ \ x_{i_m}]^\top \in \mathbb{R}^m$. Finally, for the graph Laplacian L of a directed graph, the following properties should be noted [8].

(L1) The graph Laplacian L has a zero eigenvalue, and the corresponding eigenvector is the vector whose elements are one, *i.e.*, $L1_n = 0_{n\times 1}$.

(L2) If the graph is bidirectional, *i.e.*, there exists an edge from node j to node i if there exists an edge from node i to node j, then L is positive-semidefinite. Moreover, the eigenvalues are nonnegative real numbers.

(L3) If the graph is strongly connected, then the zero eigenvalue of L is isolated, *i.e.*, $\lambda_2(L) \neq 0$.

2. Problem formulation

2.1. System description

Consider the feedback system Σ in Fig. 2, which is composed of a single-input ℓ-output plant, ℓ sensor nodes, m controller nodes, and one actuator node.

The plant P is a continuous-time linear system

$$P : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) \end{cases} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}$ is the input, $y(t) \in \mathbb{R}^\ell$ is the output, and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, and $C \in \mathbb{R}^{\ell\times n}$ are constant matrices. The initial state is given as $x(0) = x_0 \in \mathbb{R}^n$.

The sensor node S_i ($i \in \{1, 2, \ldots, \ell\}$) is of the form

$$S_i : w_i(t) = \sigma_i(y_i(t)) \tag{2}$$

where $y_i(t) \in \mathbb{R}$ is the input, that is, the i-th element of $y(t)$, $w_i(t) \in \mathbb{R}$ is the output, and $\sigma_i : \mathbb{R} \to \mathbb{R}$ is a function.

The controller node K_i ($i \in \{1, 2, \ldots, m\}$) is given by

$$K_i : \begin{cases} \dot{\xi}_i(t) = \kappa_1(\xi_i(t), [w_j(t)]_{j\in\mathbb{S}_i}, [z_j(t)]_{j\in\mathbb{K}_i}), \\ z_i(t) = \kappa_2(\xi_i(t), [w_j(t)]_{j\in\mathbb{S}_i}, [z_j(t)]_{j\in\mathbb{K}_i}) \end{cases} \tag{3}$$

where $\xi_i(t) \in \mathbb{R}$ is the state, $[w_j(t)]_{j\in\mathbb{S}_i} \in \mathbb{R}^{|\mathbb{S}_i|}$ and $[z_j(t)]_{j\in\mathbb{K}_i} \in \mathbb{R}^{|\mathbb{K}_i|}$ are the inputs, $z_i(t) \in \mathbb{R}$ is the output, and $\kappa_1, \kappa_2 : \mathbb{R} \times \mathbb{R}^{|\mathbb{S}_i|} \times \mathbb{R}^{|\mathbb{K}_i|} \to \mathbb{R}$ are functions. The set $\mathbb{S}_i \subseteq \{1, 2, \ldots, \ell\}$ is the index set of the neighbor sensor nodes, *i.e.*, the sensor nodes whose outputs are available to K_i. Similarly, $\mathbb{K}_i \subseteq \{1, 2, \ldots, m\}\backslash\{i\}$ is the index set of the neighbor controller nodes. The functions κ_1 and κ_2 and the initial state $\xi_i(0)$ are assumed to be the *same* for K_i ($i = 1, 2, \ldots, m$). This implies that the controller nodes are handled in an indiscriminate manner, which makes the system scalable. We further assume $\xi_i(0) = 0$.

The actuator node M is given by

$$M : u(t) = \mu([z_j(t)]_{j\in\mathbb{K}_M}) \tag{4}$$

where $[z_j(t)]_{j\in\mathbb{K}_M} \in \mathbb{R}^{|\mathbb{K}_M|}$ is the input, $u(t) \in \mathbb{R}$ is the output, and $\mu : \mathbb{R}^{|\mathbb{K}_M|} \to \mathbb{R}$ is a function. The set $\mathbb{K}_M \subseteq \{1, 2, \ldots, m\}$ is the index set of the neighbor controller nodes.

For simplicity of notation, $w(t) \in \mathbb{R}^\ell$, $\xi(t) \in \mathbb{R}^m$, and $z(t) \in \mathbb{R}^m$ are defined as $w(t) := [w_1(t) \ \ w_2(t) \ \ \cdots \ \ w_\ell(t)]^\top$, $\xi(t) := [\xi_1(t) \ \ \xi_2(t) \ \ \cdots \ \ \xi_m(t)]^\top$, and $z(t) := [z_1(t) \ \ z_2(t) \ \ \cdots \ \ z_m(t)]^\top$, respectively. The first one is the collective output of the sensor nodes, and the second and last ones are the collective state and output of the controller nodes, respectively.

The idea of the feedback system Σ is as follows. The sensor node S_i sends the signal $w_i(t)$ to the neighbor controller nodes based on the measurement $y_i(t)$. The controller nodes K_i ($i = 1, 2, \ldots, m$) communicate with their neighbor ones, and share compressed information on all the sensor signals and the controller outputs, *i.e.*, $w(t) \in \mathbb{R}^\ell$ and $z(t) \in \mathbb{R}^m$, as the scalar states $\xi_i(t)$ ($i = 1, 2, \ldots, m$). The actuator node M determines the control input $u(t)$ according to the shared information.