ELSEVIER

Systems & Control Letters 37 (1999) 193–205

SYSTEMS & CONTROL LETTERS

www.elsevier.com/locate/sysconle

Limited communication control ☆

Dimitris Hristu *, Kristi Morgansen

Harvard University, Division of Engineering and Applied Sciences, 29 Oxford St., Cambridge, MA 02138, USA

Received 8 July 1998; received in revised form 6 January 1999; accepted 23 February 1999

Abstract

This work in limited communication control is directed towards bringing together classical control theory and communication theoretical issues that are of practical importance in the design of control systems. It is common to "decouple" the communication aspects from the underlying dynamics of a system, as this simplifies the analysis and generally works well for classical models. However, in situations where a single decision maker controls many subsystems over a communication channel of a finite capacity, the computation of control signals and their transmission to each system may take significant amounts of time. To address such cases, we consider a class of discrete-time models that jointly optimize over control and communication goals. Real-world examples where these models play a role include remotely controlled unmanned autonomous vehicles (UAVs), planetary rovers, arrays of microactuators and power control in mobile communication.

Keywords: Discrete time linear systems; Simulated annealing; Control with limited attention

1. Introduction

One current research topic receiving a great deal of attention involves the control and coordination of multiple semi-autonomous subsystems. Examples include fleets of unmanned autonomous vehicles (UAVs), remote planetary exploration with multiple coordinated robots, and the control of microactuator arrays. In such systems, simultaneous communication with all subsystems may not be possible because of physical or performance constraints. In the case of UAVs, stealth requirements may prevent continuous communication between subsystems and the central controller. For planetary rovers, their distance from earth and their limited on-board power lead to sparse communication patterns. In addition, when the number of subsystems to be controlled exceeds a threshold, as in the case of large arrays of microactuators, communication at real-time rates may not be possible. Such examples of control systems are a departure from the usual "continuous communication" model, and the controller must divide its attention between the subsystems.

The problem with which this paper is concerned is that of stabilizing multiple coupled linear systems under output feedback when we only have the capability of communicating (read the state and/or set the input)

☆ This work was funded by Draper Laboratories IR&D Grant no. DLH505345, NSF Grant no. EEC 94-02384 and ARO Grant no. DAAG 55-97-1-0144.

* Corresponding author.
E-mail address: hristu@hrl.harvard.edu (D. Hristu)

0167-6911/99/$ - see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0167-6911(99)00020-1

with one subsystem at a time and when the overall communication rate is limited. Related problems in hybrid system theory can be found in [3,6]. Previous work in [5,16,17] has considered the problems of distributed computation and of system control and estimation with limited bandwidth. Some work specifically discussing systems with communication constraints can be found in [1,4,7,11,13,15]. The work in [7] upon which this paper is based, addressed the effects of communicating sequentially with each of a set of linear systems and described a method for stabilizing all systems in the set. In particular, this last work showed how the stabilization problem for multiple periodic linear systems is equivalent to searching for stable elements in an affine subspace. That subspace is spanned by a set of (constant) matrices derived from the systems to be controlled. Problems of this type do not have useful analytic solutions in general. Work by Blondel and Tsitsiklis [2] has shown that a discretized version of this problem is NP-hard (see [9] for a discussion of NP-hard problems).

In Section 3 we present an algorithm for obtaining the matrices that span the affine subspace of interest. We call this method the "extensification" algorithm for reasons that will become clear in the development and demonstrate its use in Section 4. Section 5 discusses a simulated annealing algorithm for finding elements in the subspace that correspond to stabilizing feedback gains for the overall system, if such gains exist. Some simulation results obtained with these algorithms are presented in Section 6.

2. Problem statement

We start from a finite collection of linear systems that may be coupled together through their dynamics and feedback. A single controller makes a choice every Δt units of time about which of the systems to communicate with. The sequence of communication choices effectively determines how much attention is given to each system. Because only one system receives updated control information at each step, each system receives a finite amount of attention. The pattern of communication choices defines a *communication sequence* which we will assume to be periodic. Given the collection of systems and a communication sequence, we want to find a set of constant feedback gains for each subsystem in order to stabilize all systems in the collection. The solutions we have developed are based on gradient descent and simulated annealing methods and represent a novel approach to the problem.

The work in [7] suggests combining the time-dependent dynamics of each subsystem with the communication sequence to arrive at what we call the "extensive form" of the entire collection. This process results in a single higher-dimensional time-invariant linear system of the form

$$Z(k+1) = \left(\hat{A}_0 + \sum_{1}^{m} \gamma_i \hat{A}_i\right) Z(k). \tag{1}$$

The dimension of this system is $N \cdot p^2$ where $N = \sum_i \dim(x_i)$, p is the length of the communication period and $\gamma_i \in \mathbb{R}$ determine the feedback gains.

Under this formulation, the stabilization problem considered in this paper is equivalent to the following:

Let $\{\hat{A}_i\} \in \mathbb{R}^{q\times q}$ *be a set of* $(m+1)$ *matrices which form the basis of an affine subspace,* $\mathscr{S} \subset \mathbb{R}^{q\times q}$. *Find* $\gamma_i \in \mathbb{R}$, *such that*

$$\hat{A} = \hat{A}_0 + \sum_{i=1}^{m} \gamma_i \hat{A}_i \tag{2}$$

is a stable element of $\mathscr{S}$.

As mentioned above, it is known that the discretized version of this problem is NP-hard.

3. Extensive form of multiple discrete-time systems

Consider the task of simultaneously controlling the set of n subsystems

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^{n} A_{ij} x_j(k) + \sum_{j=1}^{n} B_{ij} u_j(k), \quad i = 1,\dots,n, \\ y_i(k) &= C_i x_i(k) \end{aligned} \tag{3}$$

with a decision maker which cannot communicate with all subsystems at once. In particular, consider a periodic communication sequence s with a period of length p where at each step in the sequence data transmission occurs only for a single subsystem. For now, we will assume that both transmission and reception of data occur simultaneously and do not require two separate steps in the communication sequence. Generalizing to the case where data transmission and reception require separate steps is straightforward but requires longer communication sequences.

The internal dynamics of each subsystem and the dynamic coupling between subsystems evolve independently of any communication delays. The control inputs however, are determined by state information which will not always be up to date. Under the assumption of linear output feedback, the controls $u_i(k)$ are given by

$$u_i(k) = \sum_{j=1}^{n} \Gamma_{ij} C_j x_j(k - d(k,s)), \tag{4}$$

where $d(k,s)$ is a delay that depends on the communication sequence s and the current step k. As a result, the closed-loop update equations for each subsystem are time-varying and are given by

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^{n} A_{ij} x_j(k) + \sum_{l=1}^{n} B_{il} \sum_{j=1}^{n} \Gamma_{lj} C_j x_j(k - d(k,s)) \\ &= \sum_{j=1}^{n} A_{ij} x_j(k) + \sum_{j=1}^{n} F_{ij} x_j(k - d(k,s)), \end{aligned} \tag{5}$$

where

$$F_{ij} = \sum_{l=1}^{n} B_{il} \Gamma_{lj} C_j. \tag{6}$$

We want to find values for the feedback gains Γ_{ij} such that all subsystems are stable. For conceptual and numerical reasons, we will combine all time-varying subsystems into a single time-invariant system

$$Z(k+1) = (\hat{A}_0 + \hat{A}(\Gamma_{ij}))Z(k) \tag{7}$$

which we will refer to as the extensive form. In this extensive form, the matrix $\hat{A}_0$ comes from the drift dynamics. The effects of feedback will be expressed in the gain-dependent matrix $\hat{A}$.

3.1. Drift dynamics

The first step in creating the extensive form is to define "buffer" states, $\hat{x}_i(k)$, that contain past state values for each step in the communication sequence. For the ith subsystem we have

$$\hat{x}_i(k) = \begin{bmatrix} x_i(k-p+1) \\ \vdots \\ x_i(k-1) \\ x_i(k) \end{bmatrix}. \tag{8}$$

We can then write the drift dynamics for the ith subsystem as a system of dimension $\dim(x_i)\cdot p$:

$$\hat{x}_i(k+1) = (J_p \otimes I)\hat{x}_i(k) + \sum_{j=1}^{n} (E_{n,n} \otimes A_{ij})\hat{x}_j(k) = \sum_{j=1}^{n} \tilde{A}_{ij}\hat{x}_j(k), \tag{9}$$

where J_p is the $p\times p$ Jordan matrix composed of all zeros except for ones on the superdiagonal, $E_{i,j}$ is a matrix of zeros with a one in the (i,j)th position, and $\otimes$ is the Kronecker product.[1]

3.2. Feedback dynamics

To determine how the gain matrices Γ_{ij} enter the dynamics of the buffered subsystems, we will define an integer matrix, T, that contains an update sequence for each subsystem. The matrix T will have n rows (one for each subsystem) and p columns. The (i,j)th entry of T will denote the number of steps from the jth step of the communication sequence until the next communication with the ith subsystem. The feedback dynamics for subsystem i at the qth step in the communication sequence can then be expressed as

$$\hat{x}_i(k+1) = \sum_{j=1}^{n} (E_{n,T_{jq}} \otimes F_{ij})\hat{x}_j(k) = \sum_{j=1}^{n} \tilde{F}_{ijq}\hat{x}_j(k), \tag{10}$$

where

$$\tilde{F}_{ijq} = E_{n,T_{jq}} \otimes F_{ij} \tag{11}$$

and T_{jq} is the (j,q)th entry of T.

3.3. Combined drift and feedback

Now, at each sequence step, $q=1,\dots,p$, we can write the evolution of all subsystems as

$$\tilde{x}_q(k+1) = M_q\tilde{x}_q(k), \tag{12}$$

where

$$\tilde{x}_q(k) = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix}, \tag{13}$$

$$M_q = \begin{bmatrix} \tilde{A}_{11} & \cdots & \tilde{A}_{1n} \\ & \vdots & \\ \tilde{A}_{n1} & \cdots & \tilde{A}_{nn} \end{bmatrix} + \begin{bmatrix} \tilde{F}_{11q} & \cdots & \tilde{F}_{1nq} \\ & \vdots & \\ \tilde{F}_{n1q} & \cdots & \tilde{F}_{nnq} \end{bmatrix} = \tilde{A} + \tilde{F}_q. \tag{14}$$

Notice that because the drift dynamics evolve independent of any communication delays, $\tilde{A}$ is constant. Because the communication sequence is periodic, so is the matrix $\tilde{F}_q$. As a result, M_q in Eq. (12) is p-periodic in q.

[1] The Kronecker product of two matrices, $A\in\mathbb{R}^{m\times n}$ and B, is defined by

$$A\otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ & \ddots & \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

We can now rewrite the periodic discrete-time linear system in Eq. (12) as an equivalent higher-dimensional time-invariant system (see, e.g. [8]). If we define

$$Z(k)=\begin{bmatrix}\tilde{x}_1\\ \vdots\\ \tilde{x}_p\end{bmatrix}, \tag{15}$$

then the time-invariant system given by

$$Z(k+1)=\begin{bmatrix}0 & \cdots & 0 & 0 & M_1\\ M_2 & 0 & \cdots & 0 & 0\\ 0 & M_3 & 0 & \cdots & 0\\ & & \vdots & & \\ 0 & \cdots & 0 & M_p & 0\end{bmatrix}Z(k) \tag{16}$$

captures the closed-loop behavior of all the original subsystems (see [12] for an alternative technique). By construction, stability of the extensive form is equivalent to stability of the original system. Each of the M_i are affine in the Γ_{ij}, therefore we can extract a basis for the affine subspace described in Section 2 and rewrite Eq. (16) as

$$Z(k+1)=\left(\hat{A}_0+\sum_{i=1}^{m}\gamma_i\hat{A}_i\right)Z(k)=\hat{A}Z(k), \tag{17}$$

where the scalars γ_i determine the feedback gain matrices Γ_{ij}. It should be noted that the matrices $\hat{A}_i$ are sparse by construction.

If we take the pth power of the matrix $\hat{A}$, the result is

$$\hat{A}^p=\begin{bmatrix}M_1M_pM_{p-1}\ldots M_2 & 0 & 0 & \cdots & 0\\ 0 & M_2M_1M_p\ldots M_3 & 0 & \cdots & 0\\ & & \ddots & & \\ 0 & 0 & \cdots & 0 & M_pM_{p-1}\ldots M_1\end{bmatrix}. \tag{18}$$

This matrix shows that after a full communication period, the stability properties of the extensive form can be captured in a lower-dimensional space by considering

$$z(k+1)=\hat{M}z(k), \tag{19}$$

where $\hat{M}=M_pM_{p-1}\ldots M_1$. In this lower-dimensional space, the number of gains is the same as in the extensive form, however the gains enter the matrix $\hat{M}$ nonlinearly. If the number of gains is m, then the number of gain matrix coefficients for the extensive form is $m+1$ while in the lower-dimensional space the number of gain matrix coefficients is $\sum_{n=0}^{p}\binom{m+n-1}{n}$. Thus, even though the lower-dimensional space has dimension $p\cdot N$ (while the extensive form has dimension $p^2\cdot N$), the advantage of shorter algorithm run times is quickly lost against the memory demands of the large number of gain coefficient matrices. As an example, consider a system composed of two coupled single-input second-order subsystems with two outputs each. Under output feedback, there will be a total of eight feedback gains to be selected. Assuming a communication sequence of length four, the extensive form will consist of 8 matrices of dimension 64 as opposed to 495 matrices of dimension 16 for the low-dimensional approach. Assuming each matrix entry occupies two bytes, the memory requirements are 64 and 247.5 Kb, respectively. This disparity makes the latter method unwieldy as the number of states and the length of the communication sequence increase.

4. An extensification example

Consider the example of controlling the two subsystems

$$\begin{aligned} x_1(k+1) &= A_{11}x_1(k) + A_{12}x_2(k) + B_{11}u_1(k) + B_{12}u_2(k), \\ y_1(k) &= C_1x_1(k), \\ x_2(k+1) &= A_{21}x_1(k) + A_{22}x_2(k) + B_{21}u_1(k) + B_{22}u_2(k), \\ y_2(k) &= C_2x_2(k) \end{aligned} \tag{20}$$

with a periodic communication sequence $[1,2,2]$, where a control value is sent to the first subsystem and then two control values are sent to the second subsystem. We assume that outputs are received at the same time that control values are sent and that the control values that are sent reflect the new output values. The closed-loop dynamics are then

$$\begin{aligned} x_1(k+1) &= \sum_{i=1}^{2} A_{1i}x_i(k) + \sum_{i=1}^{2} B_{1j} \sum_{i=1}^{2} \Gamma_{jl}C_lx_l(k) \\ &= \sum_{i=1}^{2} A_{1i}x_i(k) + \sum_{i=1}^{2} F_{1j}x_j(k), \end{aligned} \tag{21}$$

$$\begin{aligned} x_2(k+1) &= \sum_{i=1}^{2} A_{2i}x_i(k) + \sum_{i=1}^{2} B_{1j} \sum_{i=1}^{2} \Gamma_{jl}C_lx_l(k) \\ &= \sum_{i=1}^{2} A_{2i}x_i(k) + \sum_{i=1}^{2} F_{2j}x_j(k). \end{aligned} \tag{22}$$

With the given three-step communication cycle, the system equations will evolve according to

$$\begin{aligned} x_1(k+1) &= \sum_{i=1}^{2} A_{1i}x_i(k) + F_{11}x_1(k) + F_{12}x_2(k-1) \\ x_1(k+2) &= \sum_{i=1}^{2} A_{1i}x_i(k+1) + F_{11}x_1(k) + F_{12}x_2(k+1) \\ x_1(k+3) &= \sum_{i=1}^{2} A_{1i}x_i(k+2) + F_{11}x_1(k) + F_{12}x_2(k+2) \\ x_1(k+4) &= \sum_{i=1}^{2} A_{1i}x_i(k+3) + F_{11}x_1(k+3) + F_{12}x_2(k+2) \\ &\vdots \\ x_2(k+1) &= \sum_{i=1}^{2} A_{2i}x_i(k) + F_{21}x_1(k) + F_{22}x_2(k-1) \\ x_2(k+2) &= \sum_{i=1}^{2} A_{2i}x_i(k+1) + F_{21}x_1(k) + F_{22}x_2(k+1) \end{aligned} \tag{23}$$

$$x_2(k+3) = \sum_{i=1}^{2} A_{2i}x_i(k+2) + F_{21}x_1(k) + F_{22}x_2(k+2)$$

$$x_2(k+4) = \sum_{i=1}^{2} A_{2i}x_i(k+3) + F_{21}x_1(k+3) + F_{22}x_2(k+2)$$

$$\vdots$$

In this case, the two subsystems x_1 and x_2 are coupled and their dynamics have a periodicity of three steps. We then define the buffered states

$$\hat{x}_1(k) = \begin{bmatrix} x_1(k-2) \\ x_1(k-1) \\ x_1(k) \end{bmatrix}, \qquad \hat{x}_2(k) = \begin{bmatrix} x_2(k-2) \\ x_2(k-1) \\ x_2(k) \end{bmatrix}. \tag{24}$$

The corresponding drift dynamics are

$$\begin{aligned} \hat{x}_1(k+1) &= \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & A_{11} \end{bmatrix} \hat{x}_1(k) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & A_{12} \end{bmatrix} \hat{x}_2(k), \\ \hat{x}_2(k+1) &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & A_{21} \end{bmatrix} \hat{x}_1(k) + \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & A_{22} \end{bmatrix} \hat{x}_2(k). \end{aligned} \tag{25}$$

Let $\tilde{x} = [\hat{x}_1^{\mathrm{T}}, \hat{x}_2^{\mathrm{T}}]^{\mathrm{T}}$. The combined drift and feedback dynamics at each of the three steps of the communication sequence are

$$\tilde{x}_q(k+1) = M_q \tilde{x}_q(k), \tag{26}$$

where

$$M_1 = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ F_{11} & 0 & A_{11} & 0 & 0 & F_{12}+A_{12} \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ F_{21} & 0 & A_{21} & 0 & 0 & F_{22}+A_{22} \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & F_{11}+A_{11} & 0 & F_{12} & A_{12} \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & F_{21}+A_{21} & 0 & F_{22} & A_{22} \end{bmatrix},$$

$$M_3 = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & F_{11} & A_{11} & 0 & 0 & F_{12}+A_{12} \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & F_{21} & A_{21} & 0 & 0 & F_{22}+A_{22} \end{bmatrix}, \tag{27}$$

$$F_{ij} = \sum_{l=1}^{2} B_{il} \Gamma_{lj} C_j. \tag{28}$$

The state vector of the overall system is then $Z = [\tilde{x}_1^{\mathrm{T}}, \tilde{x}_2^{\mathrm{T}}, \tilde{x}_3^{\mathrm{T}}]^{\mathrm{T}}$, and the extensive form is

$$Z(k+1) = \begin{bmatrix} 0 & 0 & M_1 \\ M_2 & 0 & 0 \\ 0 & M_3 & 0 \end{bmatrix} Z(k). \tag{29}$$

From the last equation, we can then extract the matrices that correspond to the internal dynamics and to each element of the feedback gains Γ_{ij} to arrive at

$$Z(k+1) = \left(\hat{A}_0 + \sum_i \gamma_i \hat{A}_i\right) Z(k). \tag{30}$$

5. Finding a set of stabilizing gains

After our collection of linear systems has been put in the extensive form as described in Section 3, the problem of stabilizing the systems becomes equivalent to finding a stable element of the affine subspace

$$\hat{A} = \hat{A}_0 + \sum_{i=1}^{m} \gamma_i \hat{A}_i. \tag{31}$$

Because there may be many choices for the gains γ_i that correspond to stable elements in the subspace, we will require that the eigenvalues of $\hat{A} = \hat{A}_0 + \sum_i \gamma_i \hat{A}_i$ are enclosed in a circle with the smallest possible radius. This suggests minimizing the spectral radius of the closed-loop system

$$\eta = \|\lambda_{\max}(\hat{A})\|. \tag{32}$$

In general, we expect (and observe in numerical simulations) many local minima; this makes gradient descent methods ineffective. Instead, we have developed a modified descent algorithm that uses simulated annealing. Because the extensive form of our systems essentially expresses their dynamic behavior for each step in the communication sequence, having the spectral radius of the extensive system be less than unity is equivalent to stable behavior across all systems. Our algorithm numerically computes the gradient $\partial\eta/\partial\gamma_i$ and then lets the gains γ_i flow along that gradient, adding a white-noise term $\mathrm{d}w$ with a gain $g(t)$ that decays to zero:

$$\mathrm{d}\gamma_i = \frac{\partial \eta}{\partial \gamma_i}\mathrm{d}t + g(t)\,\mathrm{d}w. \tag{33}$$

The "cooling schedule" $g(t)$ should go to zero as $t \to \infty$, but it should do so at a slow enough rate for the spectral radius to reach the global minimum.

6. Simulated annealing results

We have simulated a set of two coupled second-order linear subsystems with inputs $u, v \in \mathbb{R}$, states $x, z \in \mathbb{R}^2$ and outputs $y, w \in \mathbb{R}$. The input–output relationship is described by

$$\begin{bmatrix} y(k+1) \\ w(k+1) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \begin{bmatrix} u(k) \\ v(k) \end{bmatrix}, \tag{34}$$

where

$$G_{11}(z) = \frac{(-0.8+z)(-0.33+z)(-0.05+z)}{(-2.67+z)(-1+z)(-0.33+z)(-0.33+z)},$$

$$G_{12}(z) = \frac{2(-1.64+z)(-0.33+z)(-0.28+z)}{(-2.67+z)(-1+z)(-0.33+z)(-0.33+z)},$$

$$G_{21}(z) = \frac{0.95(-2.64+z)(-1.12+z)}{(-2.67+z)(-1+z)(-0.33+z)(-0.33+z)},$$

$$G_{22}(z) = \frac{1.95(-2.66+z)(-0.71+z)}{(-2.67+z)(-1+z)(-0.33+z)(-0.33+z)}. \tag{35}$$

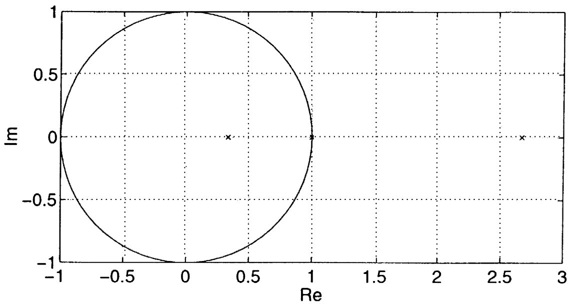

Fig. 1. Eiegenvalues of original systems, $\|\lambda_{\max}\| = 2.67$.

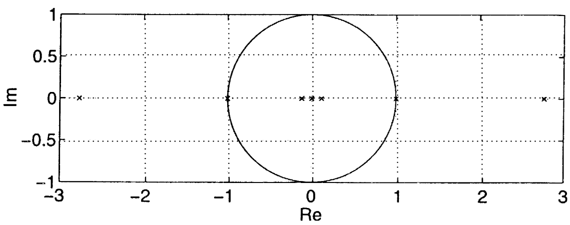

Fig. 2. Uncontrolled system eigenvalues for $s = [1,2]$, $\|\lambda_{\max}\| = 2.77$.

Equivalently, a state-space representation is

$$\begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 1/5 & 0 & 0 \\ 0 & 11/4 & 0 & 1/5 \\ 1 & 1/5 & 1/3 & 3/4 \\ 0 & -1 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} u(k) \\ v(k) \end{bmatrix},$$

$$\begin{bmatrix} y(k+1) \\ w(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix}.$$

It should be noted that the open loop system is unstable. Its eigenvalues are shown in Fig. 1.

Also, note that if the coupling between the two subsystems is removed, the first subsystem (with state x) has a spectral radius of 2.75, as opposed to only 0.25 for the second subsystem. This relationship suggests that a communication sequence that directs more of the controller's attention to the first subsystem may lead to better performance. In the following, we investigate the performance of our algorithm using two different communication sequences. Both cases were run with the same cooling schedule, shown in Fig. 4. The extensification and simulated annealing algorithms were implemented using Matlab$^{\text{TM}}$ and in C++.

6.1. Control with uniform attention

If the controller is to pay equal attention to each of the two subsystems, then the communication sequence may be taken to be [1,2]. After the extensive form is computed, the eigenvalues of the uncontrolled extensified system are those shown in Fig. 2. Performing simulated annealing on the feedback gains that are available reduces the spectral radius, but not enough to stabilize the system. The resulting final eigenvalues of the extensified system are shown in Fig. 3. As expected, the closed loop behavior of the overall system is unstable. A plot of the cooling schedule is shown in Fig. 4. The evolution of the spectral radius for the overall system is shown in Fig. 5. A representative plot of the state evolution is given in Fig. 6.

The simulated annealing algorithm was stopped after 2000 steps, followed by a deterministic gradient descent. The cooling schedule was a $\frac{1}{\log(n)}$ decay until the noise gain $g(t)$ reached a level of 0.2 and then

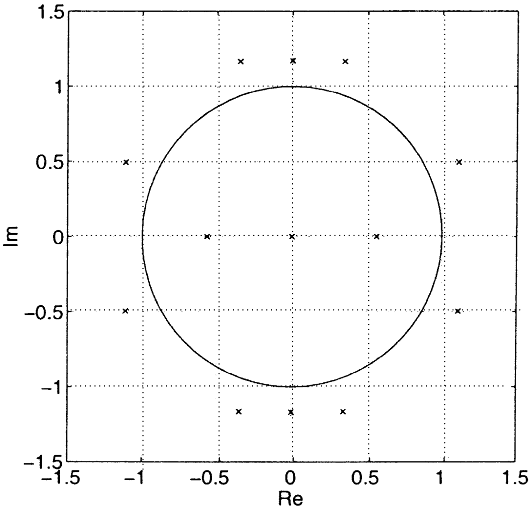

Fig. 3. Closed loop eigenvalues for $s = [1,2]$, $\|\lambda_{\max}\| = 1.2$.

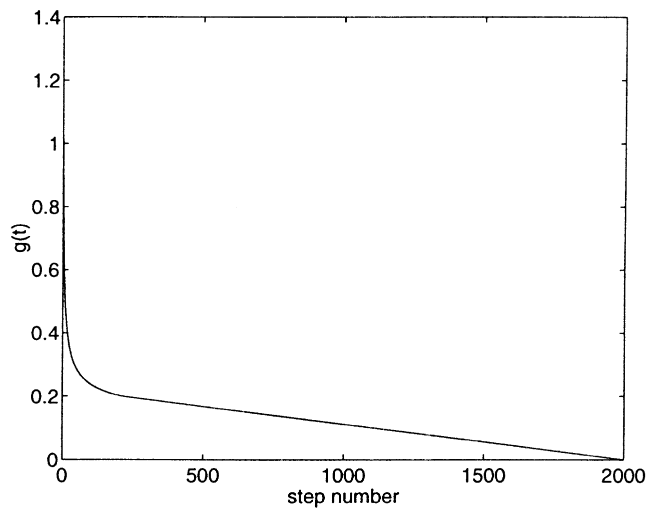

Fig. 4. Cooling schedule.

proceeded linearly to zero. These parameters were chosen after some numerical experimentation. Choosing “good” cooling schedules for the problem studied in this paper remains a subject for further research.

6.2. Towards optimal communication

Motivated by the relative sizes of the eigenvalues for the x and z subsystems, we also investigated a period-four communication sequence [1,1,1,2], that devotes three cycles to the first subsystem for every one cycle allocated to the second subsystem. The cooling schedule was the same as in the uniform communication case.

In this case, simulated annealing stabilized the closed-loop system, reducing the spectral radius to 0.817. The final closed-loop eigenvalues are shown in Fig. 7. In addition, the evolution of the spectral radius of

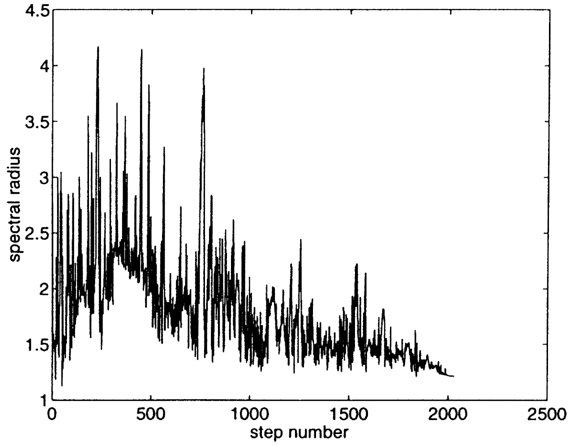

Fig. 5. Evolution of spectral radius.

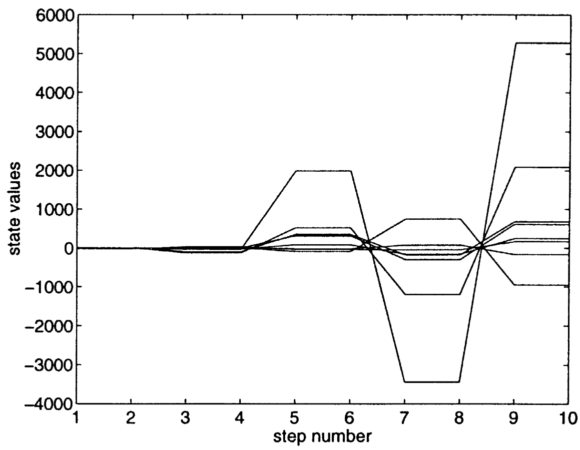

Fig. 6. State evolution with $s = [1, 2]$.

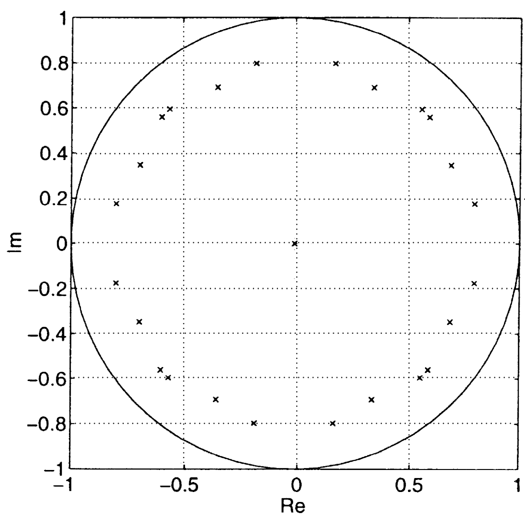

Fig. 7. Closed-loop eigenvalues for $s = [1, 1, 1, 2]$, $\lambda_{\max} = 0.817$.

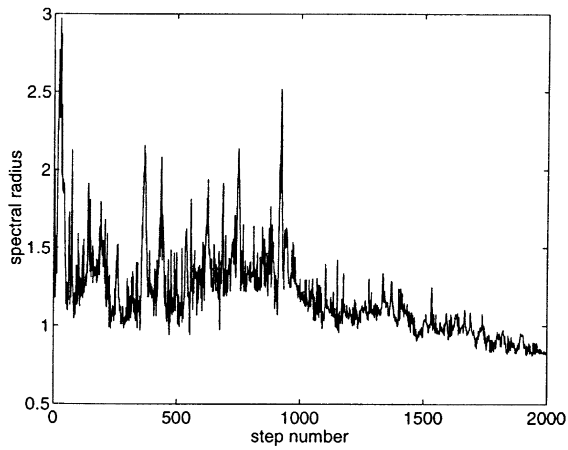

Fig. 8. Evolution of spectral radius.

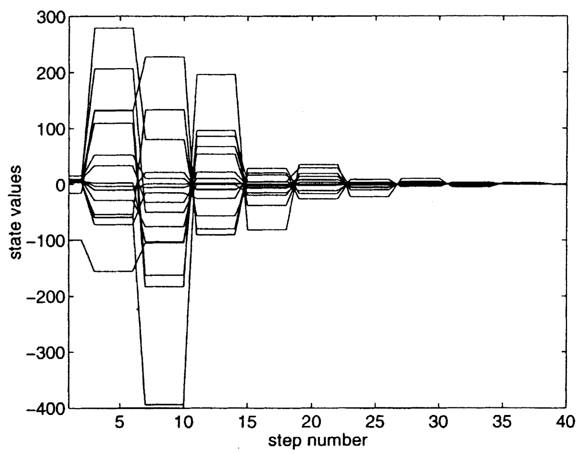

Fig. 9. State evolution with $s = [1, 1, 1, 2]$.

the system is shown in Fig. 8. Finally, a plot of the state evolution starting from a randomly selected initial conditions is given in Fig. 9.

6.3. Algorithm performance

The extensification and search for stabilizing gains for the system of Section 6.1 took approximately 5 min to complete on a dual 400 MHz Pentium-II machine running a compiled executable version of the original Matlab^{TM} code. The computations involved manipulating eight 16×16 matrices. The example in Section 6.2 took approximately 30 min to complete using the same hardware. In this case, because of the longer communication sequence, the extensive form was the sum of eight 64×64 matrices.

For larger systems and longer communication periods, the dimension of the extensive form and the time it takes to perform a single iteration of the simulated annealing algorithm grow significantly. On the platform discussed, for eight total states across all subsystems and a communication sequence of length four, the algorithm requires sixteen 128×128 matrices and takes 90 sec to complete a single iteration. If the length of the communication sequence is increased to 5, the dimension of the extensive form is 200, and one iteration requires approximately 2 min. The running time is expected to scale like $N^2 \cdot p^4$.

7. Conclusions and future work

In this paper we have presented a solution to a class of problems in feedback stabilization of multiple coupled control systems with limited communication. Our approach is novel in the sense that it brings together communication and control theoretical issues that arise in this class of problems. Potential applications for our work include remotely controlled systems, mobile communications, UAVs and other areas where communication with the system(s) of interest is limited. Our algorithm automatically computes the "extensive form" associated with a collection of linear systems, and proceeds to find a set of stabilizing feedback gains using simulated annealing. Both steps have been implemented using Matlab^{TM} and converted into C++ and stand-alone PC applications.

The performance of our algorithm depends on the choice of an appropriate cooling schedule in order for the spectral radius of the overall system to approach its global minimum. In the results presented here, the cooling schedule was chosen based on numerical experimentation with the particular problem we were trying to solve. An investigation of strategies for choosing "good" cooling schedules which give optimal performance for the general problem is in order. A starting point can be the examination of previous work on the subject (see [10,14]) and a study of its applicability in our case. Further numerical experimentation is needed to determine the relationship between the number of subsystems (and their sizes) and the completion times for the simulated annealing algorithm. Shorter run times could also be achieved by taking advantage of the sparseness of the extensive form.

Acknowledgements

The authors would like to thank Prof. R.W. Brockett for his helpful discussions on the subject.

References

[1] R. Ahlswede, I. Csiszár, Hypothesis testing with communication constraints, IEEE Trans. Inform. Theory 32 (1986) 533–542.
[2] V. Blondell, J. Tsitsiklis, NP hardness of some linear control design problems, SIAM J. Control Optim. 35 (6) (1997) 2118–2127.
[3] M.S. Branicky, V.S. Borkar, S.K. Mitter, A unified framework for hybrid control: model and optimal control theory, IEEE Trans. Automat. Control 43 (1998) 31–45.
[4] V.S. Borkar, S.K. Mitter, LQG control with communication constraints, Technical Report LIDS-P-2326, MIT, 1995.
[5] R.W. Brockett, On the computer control of movement, in: Proc. 1988 IEEE Conf. on Robotics and Automation, April 1988, pp. 534–540.
[6] R.W. Brockett, Hybrid Models for Motion Control Systems, Birkhäuser, Boston, 1993, pp. 29–54.
[7] R.W. Brockett, Stabilization of motor networks, in: Proc. 34th IEEE Conf. on Decision and Control, IEEE, New York, 1995.
[8] J.H. Davis, Stability conditions derived from spectral theory: discrete systems with periodic feedback, SIAM J. Control 10 (1) (1972) 1–13.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[10] B. Hajek, Cooling schedules for optimal annealing, Math. Oper. Res. 13 (2) (1988) 311–329.
[11] P.T. Kabamba, S. Hara, Worst-case analysis and design of sampled-data control systems, IEEE Trans. Automat. Control 38 (1993) 1337–1357.
[12] M. Kuijper, How to construct a periodically time-varying system of minimal lag: an algorithm based on twisting, Automatica, in press.
[13] S.R. Kulkarni, P.J. Ramadge, Bit rate bounds for state tracking with finite rate communication, in: Proc 35th IEEE CDC, 1996.
[14] C.B. Gellat, Jr., S. Kirkpatrick, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[15] P. Voulgaris, Control of asynchronous sampled data systems, IEEE Trans. Automat. Control. 39 (1994) 1451–1455.
[16] W.S. Wong, R.W. Brockett, Systems with finite communication bandwidth constraints – part i: state estimation problems, IEEE Trans. Automat. Control 42 (9) (1997) 1294–1299.
[17] W.S. Wong, R.W. Brockett, Systems with finite bandwidth constraints – ii: state estimation problems, IEEE Trans. Automat. Control, in press.