WP15 5:50

Proceedings of the 34th
Conference on Decision & Control
New Orleans, LA - December 1995

Stabilization of Motor Networks

Roger Brockett*
Division of Applied Sciences
Harvard University, Cambridge MA 02138
brockett@hrl.harvard.edu

Abstract

In some cases the most important factor limiting the performance of a distributed control system is not the availability of computational power but rather the availablity of time on a shared communication network for communication between the sensors, the control computer and the actuators. In this paper we develop a mathematical model describing a class of multivariable control problems of this type and present an algorithm which can be used to investigate the existence of stabilizing control laws in the presence of communication constraints. Our model assumes that the communication facilities are to be time-shared according to a pattern which is repeated periodically. The designer has the problem of picking the pattern such that effective control laws can be implemented within the constraints it imposes. If the systems being controlled are linear, there is an affine family of possible closed-loop transition matrices sassociated with each communication pattern. The selection and implementation of a particular control law which is supported by the given pattern then defines the performance. This approach allows us to put the problem of designing the communication pattern in a form that can be investigated using mathematical programming techniques. In particular one can evaluate the advantages and disadvantages associated with allocating more communication resources to some control loops and less to others.

1 Introduction

The decreasing cost associated with adding microprocessors and communication capability to sensors and actuators has made control problems involving networks of sensors, actuators and computers quite common. A recent IEEE Spectrum article [1] surveys some of these developments. In particular, networks of electric motors operating under the control of a host computer can provide effective solutions to motion control problems. However, there exists almost no tools of analysis appropriate for establishing limitations on the performance of such systems. Results that describe the extent to which systems can be controlled, or give nontrivial conditions underwhich they can be stabilized, are not available. Currently existing device networks consist of sensors and actuators linked together by a communication network. The devices on the network can communicate with each other, through the host computer, but in many situations there is only one communication channel. Contention for this channel can cause significant delays. Given this fact, it is of interest to investigate the best use of the channel, given particular control objectives.

We consider a class of actuator/sensor devices acting under the instructions of a single control computer. We assume that the individual devices incorporate the ability to receive and store several types of instructions. The simplest type instructs the device to report the value stored in one of its registers. This might be the only instruction supported by a sensing device. If the device is an actuator, typical instruction sets would include the instruction to adjust its set-point, and if the actuator has a local control system, instructions to adjust its feedback gains. Electric motors, packaged with encoders, control computers, and communication ports, supporting such instructions are now available. (These ideas are presented in a more formal way in [2].)

Of course the sensor/actuator devices operate in sampled data mode. Their local clock rate is usually much faster than the highest frequency available for communication so that we have a multi-rate sampling situation. To simplify matters we ignore the errors caused by finite word length representation of real numbers and proceed as if the sensors and actuators generate and act on real numbers. Matters of this type are considered in [4].

The evolution equation of the physical devices can be lumped together with the equations of any local controllers. We assume that at each sampling instant there is a state x consisting of the state of the individual physical devices together with the values of all past measurements currently in memory. There is also a set of linear functionals, $\langle c_i(t), x(t)\rangle$, $i = 1, 2, ..., f$ indicating those functionals of $x(t)$ that are available at time t for use by the control computer. We can think of the $c_i(t)$ as describing the *availability* of information from the sensors. We also have a set of vectors $b_i(t), b_2(t), ..., b_g$ which enter into the overall evolution equation

$$x(t+h) = Ax(t) + \sum_{i=1}^{g} b_i(t)u_i(t)$$

and serve to describe which control actions are available for use at time t. We can think of these as describing

*This work was supported in part by the National Science Foundation under Engineering Research Center Program, NSF D CDR-8803012, by the US Army Research Office under grant DAAL03-92-G-0115(Center for Intelligent Control Systems), and by ONR N0014-92-1887

the extent to which the desired control policy is *deliverable.* We postulate that each communication event, such as the request to send the value of a variable associated with one of the subsystems or the actual transmission of the data requested, or the instruction to establish a new set-point for the i^{th} unit, takes a certain multiple of a number τ. We have in mind that τ is much larger than the sampling times associated with the individual sensors and actuators. We call τ the *communication period.*

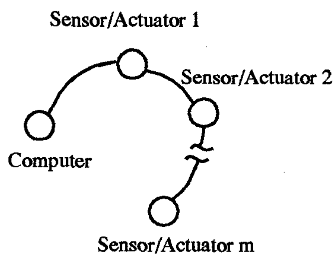

Figure 1: The daisy chain configuration.

Communication capacity will, of course, depend on the topology of the network. We are particularly interested in the familiar "daisy chain" configuration. In this case only one communication operation can be carried out at a time, leading to a limitation on the availability of data and the deliverability of instructions. In this situation it is important to determine which variables have the most relevance and hence to determine to which variables one should allocate the most communication time. The next section discusses this in more detail.

2 Communication Sequencing

The overall evolution equation appearing in the previous section includes the equations of motion for the individual devices. These are modeled by discrete-time systems of the form

$$x_i(t+h) = A_i x_i(t) + B_i u_i(t) \ ; \ y_i(t) = C_i x_i(t)$$

The sampling rate for the individual systems may vary from system to system. However, we are interested in cases for which even the least of these is much higher than the communication period. Under this hypothesis these systems may be regarded as continuous-time systems as viewed over the network. We postulate the existence of two types of control. One is local and fast the other is global but slower and subject to constraints arising from the limited capacity to communicate. We model the effects of the first type control by saying that we can instruct a local controller to use a matrix $K_i(t)$ so as to alter the local systems consistent with

$$x_i(t+h) = \hat{A}_i x_i(t) + b_i u_i(t)$$

$$y_i(t) = C_i x_i(t)$$

where $\hat{A}_i = A_i - G_i K_i(t) H_i$. The second type of control implements feedback which couples the output of one subsystem to the input of another. This leads to a set of coupled equations of the form

$$x_i(t+h) = (\hat{A}_i x_i(t) + b_i(k_{ij}(t) y_j(t - l_{ij}(t)) + u_i(t)$$

$$y_i(t) = c_i x_i(t)$$

with the l_{ij} representing the delays associated with requesting and delivering sensor information. Eliminating the explicit dependence on y, this leads to a delay-difference equation for x,

$$x_i(t+h) = (\hat{A}_i x_i(t) + b_i(k_{ij}(t) c_j x(t - l_{ij}(t)) + u_i(t)$$

where, as described above, the l_{ij} account for the fact that present values of the physical variables sensed at location i stored by device j may be the most recent data available.

Of course there exist a variety of strategies for making use of the communication channel. If we apply feedback from one unit to another then there will be cross coupling terms. The network can be used to implement sparse coupling and, in this way, obtain relatively high speed interaction or it can be used to implement a fully coupled feedback system at lower speed. Some variables can be given less access to the channel and others more, or they can all be treated equally. The limited bandwidth implies that the sampling density available to any particular feedback path is subjected to an overall constraint that limits the sum of the sampling densities. It is unnecessarily restrictive to assume that all the cross-communication channels operate at the same frequency. We will refer to the choice of channel access as *communication sequencing.* If only a single channel is available, one can think of dividing the time axis into multiples of τ and allocating a communication task to each interval as suggested by figure 2. A special case, one that is both useful in practice and theoretically interesting, arises when the communication sequence is assumed to be periodic in the sense that the access pattern is repeated periodically. In this case if the uncontrolled systems are time-invariant the controlled system will be time-varying with a periodic time variation whose period is the length of the access pattern. This has the effect of making the $l_{ij}(t)$, appearing in the previous equation, periodic with this period. Put differently, we assume that there is fixed allocation of the communication channels whereby the output of the i^{th} system is communicated to the input of the j^{th} system every T_{ij} units of time and that initially it arrives there with a delay of $\delta_{ij} h$ units of time.

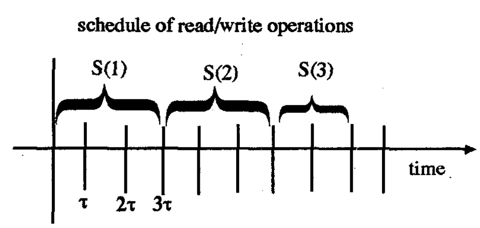

Figure 1: The Allocating time slots to communication tasks.

A delay difference equation with periodic delays can always be replaced by an ordinary autonomous equation on a higher dimensional space. For example, the difference equation

$$x(j+1) = ax(j) + bx(j - j mod 3)$$

implies the set of equations

$$x(1) = ax(0) + bx(0)$$

$$x(2) = ax(1) + bx(0)$$

$$x(3) = ax(2) + bx(0)$$

$$x(4) = ax(3) + bx(3)$$

$$\cdots$$

These can be organized as an autonomous three dimensional system

$$\begin{bmatrix} x_1(3(i+1)) \\ x_2(3(i+1)+1) \\ x_3(3(i+1)+2) \end{bmatrix} = \begin{bmatrix} a+b & 0 & 0 \\ b & a & 0 \\ b & 0 & a \end{bmatrix} \begin{bmatrix} x_1(3i) \\ x_2(3i+1) \\ x_3(3i+2) \end{bmatrix}$$

The dimension of the autonomous system will equal the product of the dimension of the original state vector and the least period of the periodic variation. By analogy with the use of the term in mathematical programming, we will will refer to this as the *extensive form* of the equations. We denote the extensive form transition matrix by $\mathcal{A}$. Notice that the stability of the original system can be easily related to the stability of the transition matrix of the extensive form; the original system is stable if and only if $\mathcal{A}$ has all it eigenvalues inside the unit circle.

Assuming now that the communication sequence is periodic of period $p\tau$, we may find an autonomous description of the dynamics in extensive form. If we restrict attention to the case in which the feedback control laws are linear, how do the feedback gains enter into the transition matrix associated with the extensive form? Because of the communication overhead associated with changing a set-point, a set-point once fixed can not be changed for a few periods. This means that a scalar describing the feedback between the i^{th} output and the j^{th} input will show up in a number of entries in the transition matrix associated with the extensive form. In fact, by analogy with the ideas of Floquet theory, the eigenvalues of $\mathcal{A}$ describe the growth or decay, of the solutions over the period $p\tau$.

We may summarize this development in the following way.

Remark 1: For a given linear system with an n-dimensional state vector, each choice of communication sequence of period $p\tau$, defines an affine subspace of the space of all real $pn \times pn$ matrices. Each matrix in this space corresponds to a specific choice of feedback gains and the mapping from the space of gains to the affine subspace is affine,

$$k \mapsto \mathcal{A}_0 + \sum k_i \mathcal{A}_i$$

The eigenvalues of the matrices in this subspace determine the rate of growth or decay of the closed-loop system.

On the basis of this analysis we see that questions which arise in the design of distributed networks of actuators and sensors depend on the selection of a subspace of the space of n by n matrices and that stability depends on finding an element in that subspace having eigenvalues in a suitable region. The subspace is characterized by the communication pattern, i.e. by the choice of the temporal pattern describing the sequence in which sensor information is communicated to the actuators.

The Communication Sequence Problem: Find a period p and a communication sequence of period p such that the affine subspace of the set of all $pn \times pn$ matrices associated with the communication sequence contains a matrix all of whose eigenvalues lie in a region Γ contained in the complex plane.

3 Enhancing Stability

The solution of the problem identified at the end of the last section depends on being able to determine if a given affine subspace of the space of square matrices of a given size contains a matrix whose eigenvalues are in a certain region.

The Affine Subspace Stability Problem: Given a region Γ of the complex plane, and a set of square matrices $\{A_0, A_1, \cdots, A_r\}$, determine whether or not the there exists a matrix of the form

$$A = A_0 + \sum_1^r \alpha_i A_i$$

with A having all its eigenvalues inside the region Γ.

Although the Routh-Hurwitz conditions and their various modifications and refinements will, in principle, allow one to check if a given matrix has its zeroes inside a half-plane or a disk, it is usually not practical to use such tests when attempting to determine if a parametrized family contains a such a matrix unless the size of the matrix is small. The design of algorithms for solving problems of this type is challenging, in part, because when described in terms of the matrix coefficients they lack convexity. For example, it can happen that A and B both have their eigenvalues in the left half-plane but that $(1/2)(A+B)$ does not. This is in contrast with the conditions for a matrix to be nonnegative definite. The conditions on the coefficients of a symmetric matrix that imply, and are implied by, nonnegative definiteness are, of course, nonlinear. Even so, the set of positive definite matrices, thought of as a subset of the vector space consisting of the symmetric matrices, form a convex cone. If H_1 and H_2 are nonnegative definite then so is $\alpha H_1 + (1-\alpha)H_2$ for all α between zero and one. This fact makes the design of algorithms for extremization of linear functionals subject to the constraint that the matrix is nonnegative definite easier than might be supposed from examining the usual algebraic test for nonnegative definite. This has been investigated in various general settings by Nesterov and Nimerovsky [6] and applied by Boyd et al.[7] in the context of automatic control.

Fortunately, results on nonnegative definiteness can give information about stability. As has been observed before, one can rephrase the fact that there exists a quadratic Liapunov function for an asymptotically stable linear system as: A square matrix A has all its eigenvalues in the circle of radius r centered at the origin if and only if the linear family of matrices

$$\mathcal{Q} = \{Q | Q = \begin{bmatrix} R & 0 \\ 0 & R - r^{-2}A^TRA \end{bmatrix} \; ; \; R = R^T\}$$

contains a positive definite matrix. Moreover, if there is a positive definite matrix of the given form, say one corresponding to the choice $R = R^*$ then $trR^* > 0$ and the unity trace matrix $(trR^*)^{-1}R^*$ is also a solution. Thus we can also say that a square matrix A has all its eigenvalues in the circle of radius r centered at the origin if and only if the affine family of matrices

$$\tilde{\mathcal{Q}} = \{Q | Q = \begin{bmatrix} R & 0 \\ 0 & R - r^{-2}A^TRA \end{bmatrix} \; ; \; R = R^T \; ; \; trR = 1\}$$

contains a positive definite matrix. Conversely, if A does not satisfy the eigenvalue condition then the distance between $\tilde{\mathcal{Q}}$ and the cone of nonnegative definite matrices is strictly positive.

Let $\mathcal{S}(n)$ denote the space of $n \times n$ symmetric matrices regarded as an inner product space with $\langle S_1, S_2 \rangle = trS_1S_2$. Let $\mathcal{P}(n) \subset \mathcal{S}(n)$ denote the set of nonnegative definite matrices, let $\tilde{\mathcal{S}}(n)$ denote the subset of $\mathcal{S}(n)$ consisting of those symmetric matrices of trace 1. Finally, if Q is an arbitrary symmetric matrix we let $d(Q, \mathcal{P})$ denote the distance between the matrix Q and the nearest nonnegative matrix, as measured by the norm associated with the given inner product. It follows from general principles that if $\mathcal{C}$ is a closed convex set of symmetric matrices then there there is an element of $\mathcal{C}$ that is closest to $\mathcal{P}$. Algorithms for finding the matrix in $\mathcal{C}$ closest to $\mathcal{P}$ are described in [6].

Define

$$\mathcal{Q}(R, k) = \{Q | Q = \begin{bmatrix} R & 0 \\ 0 & R - A^T(k)RA(k) \end{bmatrix}\}$$

Theorem: Given an affine subspace of a space of square matrices characterized by

$$\mathcal{A} = \{A : A(k) = A_0 + \sum k_i A_i\} \; ; \; A_0 \neq 0$$

and given $R^0 \in \tilde{\mathcal{S}}(n)$ and k^0, define R^i and k^i by the iteration whereby $R^{i+1} \in \tilde{\mathcal{S}}(n)$ minimizes the distance between

$$\mathcal{Q} = \{Q | Q = \begin{bmatrix} R & 0 \\ 0 & R - r^{-2}A^T(k^i)RA(k^i) \end{bmatrix}\}$$

and $\mathcal{P}(n)$ and k^{i+1} minimizes the distance between

$$\mathcal{Q} = \{Q | Q = \begin{bmatrix} R^{i+1} & 0 \\ 0 & R^{i+1} - r^{-2}A^T(k)R^{i+1}A(k) \end{bmatrix}\}$$

and $\mathcal{P}(n)$. Then if $\mathcal{A}$ does not contain a matrix with all its eigenvalues in the right half-plane, the sequence (R^i, k^i) converges to a local minimum of the distance between the bilinear family consisting of matrices of the form

$$\Phi = \{Q : Q = \begin{bmatrix} R & 0 \\ 0 & -RA(k) - A^T(k)R \end{bmatrix}; R + R^T\}$$

and $\mathcal{P}(n)$.

Proof: The set Φ is not necessarily convex. However, it is convex in k for fixed R and convex in R for fixed k. Thus the minima referred to in the theorem statement make sense. Clearly each step of the iteration brings Q closer to the cone of positive definite matrices or, at worst, does not alter the distance. The sequence of distances is bounded from below and therefore it has a limit. If the limit is zero then we can proceed to find a suitable Q and k if the limit is positive then we can only assert that the algorithm found a local minimum which was positive. We cannot rule out the possibility that a different starting point might lead to a different limit, and that this second local minimum might be in the cone of positive semidefinite matrices.

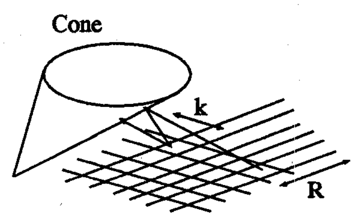

Figure 3: Illustrating the alternating steps described in the theorem.

The above algorithm depends on being able to solve the problem of identifying the symmetric matrix in a given subspace that comes closest to the nonnegative definite cone. That is, given a subspace of the set of symmetric matrices, $\mathcal{Q} = Span\{Q_1, Q_2, ...Q_k\}$, give an algorithm for determining if there is a positive definite matrix in their linear span.

Positive definite matrices have positive traces; if a linear combination Q of a set of symmetric matrices $\{Q_1, Q_2, ..., Q_k\}$, contains a positive definite matrix Q then

$$trQ = \sum \alpha_i trQ_i$$

Thus the numbers $tr(Q_1)$, $tr(Q_2)$,...,$tr(Q_k)$ cannot all vanish. Moreover, if the space contains a positive definite matrix it contains one having trace one. This follows immediately from the fact that $H > 0$ implies $\alpha H > 0$ for all positive α.

The approach to such problems recommended by Nesterov and Nimerovsky [5] as well as Boyd et al. [6], involves the use of a penalty function in the form

$$\phi(\alpha, Q) = \ln\det(\alpha I + Q)$$

clearly this function is positive when α is large and goes to minus infinity as $(\alpha I + Q)$ approaches the boundary of the cone of positive definite matrices. Thus if one minimizes $\gamma\alpha - \phi(\alpha, Q)$ with respect to real α and Q in a subspace, then it is intuitive that the for γ approaching infinity, α^* will approach the the smallest eigenvalue in the subspace of symmetric matrices under consideration.

A Penalty Function Algorithm: Let

$$\mathcal{Q} = span\{Q_1, Q_2, ..., Q_k\}$$

be a subspace of the space of real symmetric matrices. Consider the problem of finding the minimum value of

α such that $\alpha I - Q$ is positive definite for $Q \in \mathcal{Q}$ and $trQ = 1$. To do this we select a sequence of positive real numbers, $\gamma^0, \gamma^1, \gamma^2, ...$, monotone increasing and unbounded, and then minimize $\gamma^i\alpha + \ln\det(\alpha I + Q)$ starting with α sufficiently large. For fixed γ the gradient flow is just

$$\dot{Q} = -\sum Q_i tr Q_i (I\gamma\alpha - Q)$$
$$\dot{\alpha} = -\gamma tr(I\gamma\alpha + Q)^{-1}$$

provided that the Q's are orthonormalized, i.e.

$$\langle Q_I, Q_j \rangle = \delta_{ij}$$

4 References

1. Reza Raji, "Smart Networks for Control," IEEE Spectrum, June, 1994, pp 49-55.
2. R. W. Brockett, "On the Computer Control of Movement," *Proceedings of the 1988 IEEE Conference on Robotics and Automation*, New York, April 1988, pp. 534-540.
3. R .W. Brockett, "Hybrid Models for Motion Control Systems," in *Perspectives in Control*, (H. Trentelman and J.C. Willems, Eds.), Birkhäuser, Boston, 1993, pp. 29-54.
4. Wing Wong and R. W. Brockett, "Stabilization with Limited Information Feedback," (submitted for publication)
5. R. W. Brockett, "Language Driven Hybrid Systems," *Proceedings of the 33rd IEEE Conference on Decision and Control*, 1994, pp. 4210-4214.
6. Yuri Nesterov and Arkadii Nemirovskii,, *Interior-Point Polynomial Algorithms in Convex Programming,* SIAM, Philadelpha, PA, 1994.
7. Stephen. Boyd, Laurent El Ghaoui, Eric Feron and Venkataramanan Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory,* SIAM, Philadelpha, PA, 1994.