

 Open access • Proceedings Article • DOI:10.1109/CDC.1977.271553

A sampled data stackelberg coordination scheme for the multicontroller problem

— [Source link](#) 

Peter M Walsh, Jose B. Cruz

Institutions: University of Illinois at Urbana–Champaign

Published on: 01 Dec 1977 - Conference on Decision and Control

Topics: Stackelberg competition

Related papers:

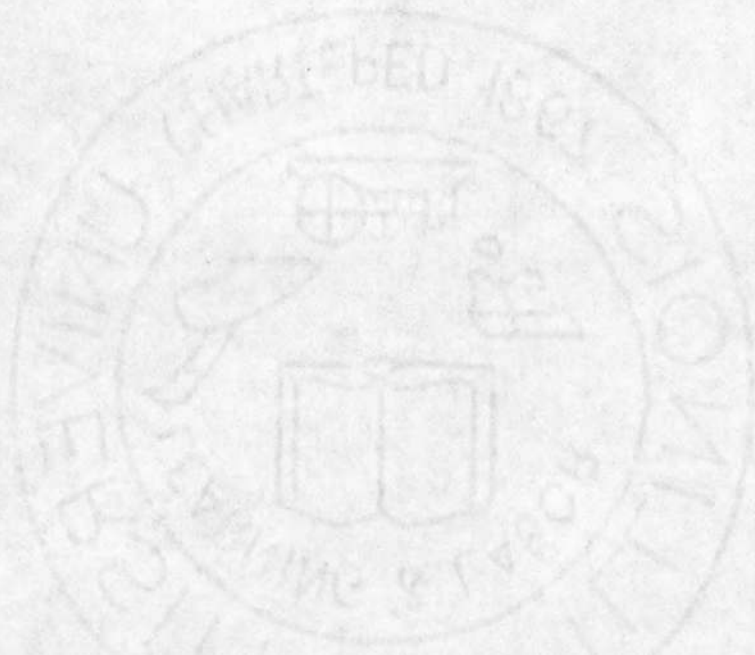
- [Model predictive sampled-data redesign for nonlinear systems](#)
- [Linear quadratic optimal flow control in multi-source connection-oriented communication networks](#)
- [Suboptimal control of linear systems with state and control inequality constraints](#)
- [Dynamic output feedback robust model predictive control with guaranteed quadratic boundedness](#)
- [Stabilization of Nonlinear Systems Under Variable Sampling: A Fuzzy Control Approach](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/a-sampled-data-stackelberg-coordination-scheme-for-the-23kb2ks0tt>

A SAMPLED DATA STACKELBERG COORDINATION SCHEME
FOR THE MULTICONTROLLER PROBLEM*

P. M. Walsh and J. B. Cruz, Jr.
Decision and Control Laboratory
Coordinated Science Laboratory and
Department of Electrical Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801



* This work was supported in part by the National Science Foundation under Grant ENG-74-20091, by the Energy Research and Development Administration under Contract U.S. ERDA EX-76-C-01-2088, Electric Energy Systems Division, and by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259.

A SAMPLED DATA STACKELBERG COORDINATION SCHEME
FOR THE MULTICONTROLLER PROBLEM

by

P. M. Walsh and J. B. Cruz, Jr.

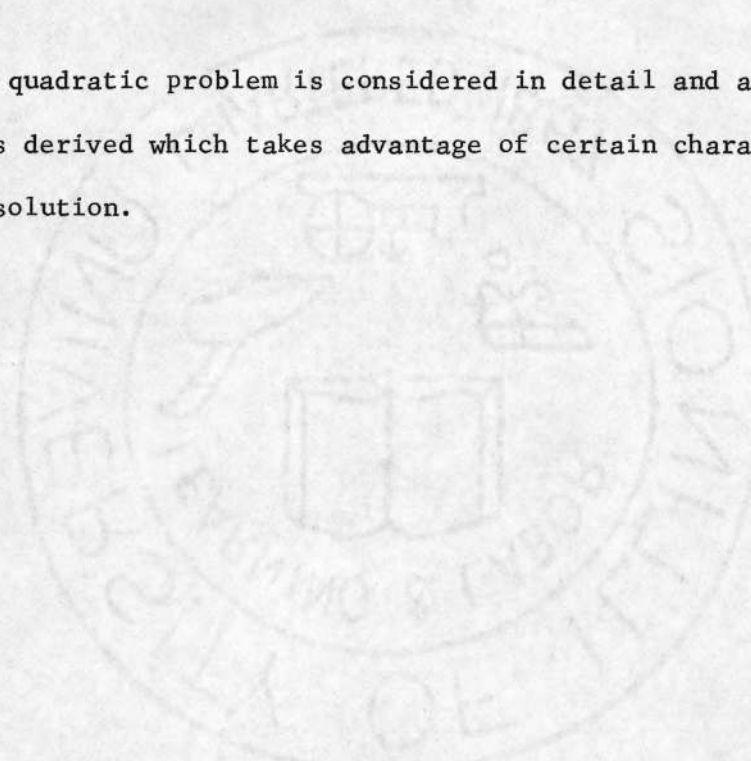
This work was supported in part by the National Science Foundation under Grant ENG-74-20091, in part by the Energy Research and Development Administration under Contract U.S. ERDA EX-76-C-01-2088, Electric Energy Systems Division, and in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259.

Abstract

In this paper we consider a multicontroller problem in which each control agent has a different objective function. The actions of the controllers are coordinated by the influence of one controller who plays according to a Stackelberg strategy relative to the remaining players. We investigate this formulation when the information flow is restricted by constraining it to be in the form of sampled data acquisition.

The computational advantage of the sampled data formulation is quite significant. Of equal importance is the relationship among the sampled data, closed loop and open loop solutions. The existence of and solution for the closed loop Stackelberg solution for the continuous time game are, at present, unresolved problems. The primary motivation for considering the sampled data formulation is to obtain a solution which maintains the computational simplicity of the tractable open loop solution while gaining the responsiveness of a state feedback solution, avoiding the complications of the purely closed loop formulation.

The linear quadratic problem is considered in detail and an efficient solution algorithm is derived which takes advantage of certain characteristics of the sampled data solution.



1. Introduction

In this paper, we consider the problem of formulating a hierarchical control structure for a multicontroller problem using the differential game concept of a Stackelberg strategy. It is assumed that in general each agent has a different objective function and that one agent, the coordinator and Stackelberg leader, has an overall objective function.

There have been numerous investigations recently into the usefulness and characteristics of the Stackelberg strategy applied to dynamic systems [1-11]. In particular, the use of the Stackelberg strategy for the coordination of many agents has been considered in [4] and [11].

A form of periodic coordination has been considered by Chong and Athans [12] in which the vertical communication in the hierarchy is constrained to be periodic. Their basic assumptions are different from those of this paper and subsequently the nature of the solutions are quite dissimilar.

With a Stackelberg strategy, we assume it is known that one player, the coordinator and Stackelberg leader, will determine his controls before any of the other players (followers or lower level decisionmakers). The lower level decisionmakers then perform their optimizations subject to their knowledge of the coordinator's decision, that is, they are reacting to his decisions. The followers act simultaneously and we consider the case when they play a Nash strategy among themselves. The leader performs his optimization subject to the expected reactions of the followers. The leader's ability to make decisions first, taking into account the reactions of the lower level decisionmakers, enables him, to a degree, to impose his criterion onto the other controllers.

Further discussion of the Nash and Stackelberg strategies for dynamic games can be found in the references.

In order to see the motivation and significance of the sampled data formulation it is necessary to appreciate two particular aspects of the continuous time Stackelberg problem.

First, unlike the classic single agent, linear quadratic control problem, or even certain multicontroller problems, the necessary conditions derived by the variational technique for the linear quadratic, continuous time, closed loop Stackelberg problem result in a non-linear control, the existence of which is not assured [8]. This problem has yet to be fully resolved.

A second anomaly of the Stackelberg solution for general dynamic games is that the principle of optimality does not, in general, hold. The imposition of the principle of optimality for discrete time games has been considered in [8] while the procedure for doing this for continuous time games has yet to be resolved.

With these aspects of the continuous time Stackelberg problem in mind, the significance of the sampled data formulation is apparent. That is, the resultant control laws are piecewise continuous linear time varying functions of the measurements for the linear quadratic case and, as we have formulated it, the principle of optimality holds at the sampling times.

Recent work on the Stackelberg strategy for continuous time dynamic systems has concentrated primarily on the open loop solution [4] and on the linearly constrained closed loop solution [6]. For the linear quadratic case, the open loop solution is a linear function of the initial condition and the solution in [6] is linear by construction but the principle of optimality does

not, in general, hold. The linear form of the sampled data solution is a direct result of this information constraint and is not due to any structural (linear) constraint being imposed on the form of the solution.

By considering the sampled data formulation we have been able to obtain a responsive state feedback solution, which is tractable, has a very simple form for implementation, and for which the principle of optimality holds at the sampling times. Of equal importance is the existence of an efficient algorithm for the calculation of this solution. Thus an important contribution of this paper is the derivation of a computationally efficient technique for the solution of the linear quadratic case.

Our objectives in deriving the solution algorithm have been to (i) minimize the on-line computations and (ii) to take advantage of the nature of the sampled data solution to greatly reduce the horizon over which integrations must be performed, thereby reducing off-line computations as well. These objectives are obtained as a result of employing a form of invariant imbedding [13].

In Section 2 we formulate the problem and present necessary conditions for the solution. The linear quadratic case will be considered in Section 3 and techniques for the solution of the linear quadratic case will be discussed in Section 4. Section 5 summarizes the results.

2. Sampled Data Formulation

Consider the system

$$\dot{x} = f(x, u_i; i = 0, 1, \dots, m), \quad x(t_0) = x_0, \quad (1)$$

$u_i \in R^{r_i}$, $x \in R^n$, where r_i is the dimension of the i th control vector. Each lower level control, u_i , for $i=1, \dots, m$, is chosen to reduce as much as possible the scalar index

$$J_i = K_{if}(x(t_f)) + \int_{t_0}^{t_f} L_i(x, u_j; j=0, 1, \dots, m) dt. \quad (2)$$

The coordinator's control, u_0 , is chosen to reduce as much as possible the scalar index

$$J_0 = K_{of}(x(t_f)) + \int_{t_0}^{t_f} L_0(x, u_i; i=0, 1, \dots, m) dt. \quad (3)$$

The terminal time, t_f , is fixed.

The information is assumed to be in the form of sampled data acquisition, that is, measurements are taken at r discrete instances in time $\{t_i \in [t_0, t_f), i=0, 1, \dots, r-1\}$. The controls will be functions of time and the latest state measurement, i.e., $u_i = u_i(t, x_j)$ for $t_j \leq t < t_{j+1}$, for all i , where $x_j \triangleq x(t_j)$.

At each sample time, t_j , the leader will calculate and announce $u_0(t, x_j)$ for $t \in [t_j, t_f)$. This control is chosen to minimize the leader's performance index under the assumption that the followers will in turn be minimizing their respective performance indices subject to the announced leader's control. The controllers are not simply solving repeated open loop solutions, but rather, at each time, t_j , the controls are calculated based on the assumption that future measurements will be available at t_k , $k=j+1, \dots, r-1$.

The necessary conditions needed to find the $u_i(x_j, t)$ for $t \in [t_j, t_{j+1})$ are found by the variational method. Contrary to the single controller case

or even certain multicontroller strategies, the Stackelberg controls found by the variational method do not in general satisfy the principle of optimality [8]. For this sampled data formulation, the principle of optimality does hold at the sampling times t_j , $j=0,1,\dots,r-1$. The controls $u_i(x_j,t) \forall i$ and for $t \in [t_j, t_{j+1})$ are calculated taking into account that similar optimizations are to be performed at future sample times to find $u_i(x_k,t)$, $t \in [t_k, t_{k+1})$, $j < k \leq r-1$. The dependence of the $u_i(x_j,t)$, $t \in [t_j, t_{j+1})$ on the future controls will be imbedded in the boundary conditions at t_{j+1} .

In order to establish appropriate boundary conditions we will need expressions for the costs to go at the sampling times. Let the optimum costs to go at time t_j be denoted by $V_i^*(x(t_j), t_j)$, $i=0,1,\dots,m$. Then for the interval $[t_j, t_{j+1})$

$$V_i^*(x_j, t_j) = \min_{u_i} \{ V_i^*(x_{j+1}, t_{j+1}) + \int_{t_j}^{t_{j+1}} L_i(x, u_k; k=0,1,\dots,m) dt \} \quad (4)$$

where

$$V_i^*(x(t_f), t_f) = K_{if}(x(t_f)), \quad i = 0,1,\dots,m \quad (5)$$

and where the minimization with respect to u_i in (4) is subject to the system constraint and to the minimizations being performed by the other controllers according to the strategy outlined in the preceding paragraphs. Note that the optimizations of the future periods are imbedded in the term $V_i^*(x_{j+1}, t_{j+1})$. Also notice that at sample time t_j , all controls from t_j through t_f will, in principle, be calculated and that they are independent of any control action prior to t_j . So, by construction, the principle of optimality does hold at the sample times.

The necessary conditions are an extension of those derived in [11].

The necessary conditions for the followers on $[t_j, t_{j+1})$ for $i=1, \dots, m$ are

$$\dot{x} = f(x, u_i; i=0, 1, \dots, m), \quad x(t_j) = x_j \quad (6)$$

$$\dot{p}_i = -\frac{\partial H'_i}{\partial x}, \quad p_i(t_{j+1})' = \frac{\partial V_i^*(x(t_{j+1}), t_{j+1})}{\partial x(t_{j+1})} \quad (7)$$

$$0 = \frac{\partial H'_i}{\partial u_i} \quad (8)$$

where

$$H_i(x, p_i, u_k; k=0, 1, \dots, m) = L_i(x, u_k; k=0, 1, \dots, m) + p_i' f(x, u_k; k=0, 1, \dots, m). \quad (9)$$

The necessary conditions for the leader on $[t_j, t_{j+1})$ are

$$\dot{\lambda} = -\frac{\partial H'_0}{\partial x}, \quad \lambda(t_{j+1})' = \frac{\partial V_0^*(x(t_{j+1}), t_{j+1})}{\partial x(t_{j+1})} - \sum_{k=1}^m \gamma_k'(t_{j+1}^-) \frac{\partial^2 V_k^*(x(t_{j+1}), t_{j+1})}{\partial x(t_{j+1})^2} \quad (10)$$

$$\dot{\gamma}_i = -\frac{\partial H'_0}{\partial p_i}, \quad \gamma_i(t_j^+) = 0, \quad i = 1, \dots, m \quad (11)$$

where $\gamma_i(t_j^-) = \lim_{t \rightarrow t_j^-} \gamma_i(t)$ for γ_i defined on the $(j-1)$ st interval $[t_{j-1}, t_j)$ and $\gamma_i(t_j^+) = \gamma_i(t_j)$ defined on the j th interval $[t_j, t_{j+1})$.

$$\frac{\partial H_0}{\partial u_0} = 0 \quad (12)$$

$$\frac{\partial H_0}{\partial u_i} = 0, \quad i = 1, \dots, m \quad (13)$$

where

$$H_0(x, \lambda, p_i, \gamma_i, \beta_i; i=1, 2, \dots, m, u_j; j=0, 1, \dots, m) = L_0(x, u_i; i=0, 1, \dots, m) + \lambda' f(x, u_i; i=0, 1, \dots, m) + \sum_{k=1}^m \left\{ \gamma_k' \left(-\frac{\partial H_k}{\partial x} \right)' + \beta_k' \left(\frac{\partial H_k}{\partial u_k} \right)' \right\}. \quad (14)$$

Equation (13) and the constraints appended under the summation sign in (14) are due to the leader taking into account the reactions of the lower level decisionmakers.

3. The Linear Quadratic Case

Assume the system is linear

$$\dot{x} = Ax + \sum_{i=0}^m B_i u_i \quad (15)$$

$$x(t_0) = x_0 \quad (16)$$

and the criteria quadratic

$$J_i = \frac{1}{2} x' K_{if} x \Big|_{t=t_f} + \frac{1}{2} \int_{t_0}^{t_f} (x' Q_i x + \sum_{j=0}^m u_j' R_{i,j} u_j) dt. \quad (17)$$

The necessary conditions for the lower level controllers for $t \in [t_j, t_{j+1})$ and $i = 1, \dots, m$ are

$$\dot{p}_i = -Q_i x - A' p_i, \quad p_i'(t_{j+1}) = \frac{\partial V_i^*(x(t_{j+1}), t_{j+1})}{\partial x(t_{j+1})} \quad (18)$$

$$u_i = -R_{i,i}^{-1} B_i' p_i. \quad (19)$$

The necessary conditions for the leader are

$$\dot{\lambda} = -Q_0 x - A' \lambda + \sum_{i=1}^m Q_i \gamma_i \quad (20)$$

$$\lambda'(t_{j+1}) = \frac{\partial V_0^*(x(t_{j+1}), t_{j+1})}{\partial x(t_{j+1})} - \sum_{i=1}^m \gamma_i'(t_{j+1}^-) \frac{\partial^2 V_i^*(x(t_{j+1}), t_{j+1})}{\partial x(t_{j+1})^2}$$

$$\dot{\gamma}_i = A \gamma_i - S_{o,i} p_i + S_i \lambda, \quad \gamma_i(t_j^+) = 0 \quad (21)$$

$$u_0 = -R_{o,o}^{-1} B_o' \lambda \quad (22)$$

where

$$S_i \triangleq B_i R_{i,i}^{-1} B_i'$$

$$S_{j,i} \triangleq B_i R_{i,i}^{-1} R_{j,i} R_{i,i}^{-1} B_i'$$

During each interval, the state will evolve according to

$$\dot{x} = Ax - \sum_{i=1}^m S_i p_i - S_o \lambda \quad (23)$$

for $t \in [t_j, t_{j+1})$ where $x(t_j)$ is determined in the previous interval.

If the state measurements are made at r discrete instances in time, we are faced with an $(r+1)$ -point boundary value problem. At this stage, there are two alternate approaches we can take to the problem. The first and standard approach starts by assuming an explicit functional dependence of the costates on the state. This results in a set of coupled matrix Riccati equations which must be solved repeatedly at each sample time. A general algorithm for the efficient solution of these equations for each new set of boundary conditions will be outlined in the next section. We will also consider an even more efficient approach utilizing invariant imbedding [13,14].

It is based on an assumption of the functional dependence of the state and costates on one another and of their explicit dependence on their respective boundary conditions. This result will be shown in detail.

4. Solution of the Linear Quadratic Problem

The first approach to dealing with the $r+1$ point boundary value problem starts by assuming that the costates depend on the states by affine functions. The affine dependence, rather than simply linear, is necessary so that the lower level decisionmakers will be able to calculate their controls as functions of the leader's announced control, i.e., their computations will be coupled to the leader's sequentially, not simultaneously.

Differential equations can be found for the coefficients of these functions and for the associated costs to go. If m is the number of controllers, the problem can be reduced to that of solving m coupled matrix Riccati equations and m matrix Lyapunov equations at each sample time, all with boundary conditions at a common time. The same set of equations are resolved at each sample time with only a change in the boundary conditions. A sampled data Nash formulation has been considered by Simaan and Cruz [9] and a computational technique for the solution of the resultant Riccati equations has also been obtained [10]. We have obtained a generalization of [10] in which the solutions of the Riccati equations are expressed in terms of a preliminary solution due to a specific set of boundary conditions and a correction term dependent on the actual boundary conditions. An algorithm is found for finding these correction terms requiring the solution of m uncoupled matrix Riccati equations, thus providing substantial improvement over a brute force solution of the coupled equations. We will not

present the details of this technique but rather will describe an alternate approach which is even more attractive.

The Second Approach: Invariant Imbedding

The ultimate goal when deriving the solution technique is to minimize the amount of computations required by taking advantage of the fact that the equations to be solved are the same in each sample interval and only the boundary conditions change.

The derivations performed in the remainder of this section will proceed as outlined below. First we define more compact notation, grouping the state and costates according to their boundary conditions. We then assume an explicit functional dependence of the costates on the state and on the costates' boundary conditions. Due to this assumption, the solutions of the resultant equations are independent of the changing costates' boundary conditions and it is because of this independence that we are able to obtain the computational savings. The cost to go equations are derived since they are needed to generate the appropriate boundary conditions to plug into the solution functions. A functional dependence of the costs to go on their boundary conditions is also assumed and finally the boundary conditions for each interval are established in terms of those in the adjacent interval. The details of the derivation follow.

Rather than making the standard assumption of a functional dependence of the costates on the state alone as in the first approach, we will make a different assumption. Notice that on the interval $[t_j, t_{j+1})$, the costates $p_{1i}, \forall i$, equations (18) and λ , equation (20), have boundary conditions at t_{j+1} . The costates $\gamma_i, \forall i$, equations (21) and the state x , equation (23),

have boundary conditions at t_j . For convenience of notation, let us group the state and costate vectors according to boundary conditions as follows

$$\begin{aligned} y_1 &\stackrel{\Delta}{=} x \\ y_2 &\stackrel{\Delta}{=} (\gamma_1' : \gamma_2' : \dots : \gamma_m')' \\ y_3 &\stackrel{\Delta}{=} (\lambda' : p_1' : p_2' : \dots : p_m')'. \end{aligned}$$

Now equations (18), (20), (21) and (23) can be expressed as

$$\frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 & A_{13} \\ 0 & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (27)$$

where the A_{ij} of (27) are appropriate concatenations of the Q, A and S matrices of (18), (20), (21) and (23). In each interval $[t_j, t_{j+1})$, the vectors y_1 and y_2 have boundary conditions at t_j and the vector y_3 has boundary conditions at t_{j+1} .

$$y_2(t_j^+) = 0 \quad (28)$$

$$y_3(t_{j+1}^-) = \begin{bmatrix} \lambda \\ \text{---} \\ p_1 \\ \vdots \\ p_m \end{bmatrix} = \begin{bmatrix} \frac{\partial V_0'}{\partial y_1} - \sum_{i=1}^m \frac{\partial^2 V_i}{\partial y_1^2} \cdot \gamma_i \\ \text{---} \\ \frac{\partial V_1'}{\partial y_1} \\ \vdots \\ \frac{\partial V_m'}{\partial y_1} \end{bmatrix} \Big|_{t_{j+1}^-} \quad (29)$$

where $y_2(t_j^+) = y_2(t_j)$ defined on the interval $[t_j, t_{j+1})$ and $y_3(t_{j+1}^-) = \lim_{t \rightarrow t_{j+1}^-} y_3(t)$ for $y_3(t)$ defined on the interval $[t_j, t_{j+1})$.

It is in the next step where we deviate from the standard approach. We will make assumptions of the functional dependence of the costates on the state and on the costates' boundary conditions. In so doing we will be able to solve for these functions independent of the costates' boundary conditions.

For $t \in [t_j, t_{j+1})$ assume¹

$$y_2(t) = F_1(t)y_1(t) + F_2(t)y_2(t_j) + F_3(t)y_3(t_{j+1}) \quad (30)$$

and

$$y_3(t) = G_1(t)y_1(t) + G_2(t)y_2(t) + G_3(t)y_3(t_{j+1}). \quad (31)$$

By differentiation of (30) and (31) and by substitution of (27), we find²

$$\dot{G}_1 = A_{31} + A_{33}G_1 - G_1A_{11} - G_1A_{13}G_1 - G_2A_{23}G_1, \quad G_1(t_{j+1}) = 0 \quad (32)$$

$$\dot{G}_2 = A_{32} + A_{33}G_2 - G_1A_{13}G_2 - G_2A_{22} - G_2A_{23}G_2, \quad G_2(t_{j+1}) = 0 \quad (33)$$

$$\dot{G}_3 = (A_{33} - G_1A_{13} - G_2A_{23})G_3, \quad G_3(t_{j+1}) = I \quad (34)$$

$$\dot{F}_1 = (A_{22} + A_{23}G_2)F_1 - F_1(A_{11} + A_{13}G_1) - F_1A_{13}G_2F_1 + A_{23}G_1, \quad F_1(t_j) = 0 \quad (35)$$

$$\dot{F}_2 = A_{22}F_2 + A_{23}G_2F_2 - F_1A_{13}G_2F_2, \quad F_2(t_j) = I \quad (36)$$

$$\dot{F}_3 = (A_{22} + A_{23}G_2 - F_1A_{13}G_2)F_3 + A_{23}G_3 - F_1A_{13}G_3, \quad F_3(t_j) = 0. \quad (37)$$

¹The dependence of $y_3(t)$ on $y_2(t)$ instead of $y_2(t_j)$ results in simplified computations.

²All matrices are evaluated at time t unless indicated otherwise.

Since $y_2(t_j) = 0$ and by substituting (30) into (31) we have

$$y_2(t) = F_1(t)y_1(t) + F_3(t)y_3(t_{j+1}) \quad (38)$$

$$y_3(t) = \bar{G}_1(t)y_1(t) + \bar{G}_3(t)y_3(t_{j+1}) \quad (39)$$

where $\bar{G}_1 = G_1 + G_2F_1$ and $\bar{G}_3 = G_3 + G_2F_3$.

For $t \in [t_j, t_{j+1})$ assume

$$y_1(t) = H_1(t)y_1(t_j) + H_3(t)y_3(t_{j+1}) \quad (40)$$

by differentiation of (40) and substitution of (27) and (39) we find

$$\dot{H}_1 = (A_{11} + A_{13}\bar{G}_1)H_1, \quad H_1(t_j) = I \quad (41)$$

$$\dot{H}_3 = (A_{11} + A_{13}\bar{G}_1)H_3 + A_{13}\bar{G}_3, \quad H_3(t_j) = 0. \quad (42)$$

If the system (15) and the criteria functions (17) are time invariant and if the sampling rate is constant, that is if $(t_{j+1} - t_j) = T = \text{constant}$ for all j , the equations (32) through (37), (41) and (42) will be the same for each interval. Then, since their boundary conditions are invariant, these equations will have to be solved only once and the same solution will be valid for every interval $[t_j, t_{j+1})$, $j = 0, 1, \dots, r-1$.

Boundary Conditions and Cost To Go Equations

The boundary conditions for the costate equations on the j th interval $[t_j, t_{j+1})$ are known in terms of the costs to go at the end of the interval,

(7) and (10). Therefore, for the purpose of obtaining the costates' boundary conditions, we must first derive the cost to go equations. First, substituting (19) and (22) for the controls and with the form of the solution for y_3 as in (39), recalling that $y_3 \stackrel{\Delta}{=} (\lambda' \ ; \ p_1' \ ; \ \dots \ ; \ p_m')'$, the integrands L_i of the criterion functions can be written

$$L_i = \frac{1}{2} \{ x' Q_i x + \sum_{j=0}^m u'_{j} R_{ij} u \} = \frac{1}{2} \{ y_1' Q_i y_1 + y_3' \hat{S}_i y_3 \}$$

and for $t \in [t_j, t_{j+1})$,

$$L_i = \frac{1}{2} \{ y_1' \bar{S}_{i1} y_1 + y_3(t_{j+1})' \bar{S}_{i2} y_3(t_{j+1}) \} + y_1' \bar{S}_{i3} y_3(t_{j+1}) \quad (43)$$

where all variables are evaluated at time t unless indicated otherwise, and where

$$\bar{S}_{i1} = Q_i + \bar{G}_1' \hat{S}_i \bar{G}_1$$

$$\bar{S}_{i2} = \bar{G}_3' \hat{S}_i \bar{G}_3$$

$$\bar{S}_{i3} = \bar{G}_1' \hat{S}_i \bar{G}_3$$

and

$$\hat{S}_i \stackrel{\Delta}{=} \begin{bmatrix} S_{i0} & & & & 0 \\ & S_{i1} & & & \\ & & \dots & & \\ & & & \dots & \\ 0 & & & & S_{im} \end{bmatrix}.$$

Due to the assumed explicit dependence of the costates, $y_3(t)$ on their boundary conditions in each interval, we must make a similar assumption for the form of the cost to go equations so that they will also be independent

of the changing boundary conditions. That is, for the interval $t \in [t_j, t_{j+1})$ we define the function

$$V_i(y_1(t), t) \stackrel{\Delta}{=} \frac{1}{2} \{ y_1(t)' C_{i1}(t) y_1(t) + y_3(t_{j+1}^-)' C_{i2}(t) y_3(t_{j+1}^-) \} + y_1(t)' C_{i3}(t) y_3(t_{j+1}^-). \quad (44)$$

When evaluated at t_j , with the controls in the interval $[t_j, t_f)$ being the optimal controls defined according to (4), this function is then the optimum cost to go, denoted $V_i^*(y_1(t_j), t_j)$. By (44) we see that on the interval $[t_j, t_{j+1})$, the cost to go is not only quadratic in y_1 , but also has a quadratic term in $y_3(t_{j+1}^-)$ and a cross term in $y_1(t)$ and $y_3(t_{j+1}^-)$.

From the relationship between the costs to go (44) and the integrands of the criteria functions (43), the differential equations of the coefficient matrices in (44) are found to be

$$\dot{C}_{i1} = -\bar{S}_{i1} - C_{i1} \bar{A}_{11} - \bar{A}_{11}' C_{i1} \quad (45)$$

$$\dot{C}_{i2} = -\bar{S}_{i2} - 2\bar{A}_{13}' C_{i3} \quad (46)$$

$$\dot{C}_{i3} = -\bar{S}_{i3} - C_{i1} \bar{A}_{13} - \bar{A}_{11}' C_{i3} \quad (47)$$

where $\bar{A}_{11} = (A_{11} + A_{13} \bar{G}_1)$ and $\bar{A}_{13} = A_{13} \bar{G}_3$.

Boundary Conditions

The boundary conditions for the last interval, that is, at the terminal time, t_f , are

$$\begin{aligned} C_{i1}(t_f) &= K_{if} \\ C_{i2}(t_f) &= 0 \\ C_{i3}(t_f) &= 0. \end{aligned} \quad (48)$$

We must also establish appropriate boundary conditions for the remaining intervals. The costs to go must be continuous and therefore

$$V_i(y_1(t_j^-), t_j^-) = V_i(y_1(t_j^+), t_j^+). \quad (49)$$

Since the cost to go equations are integrated backwards, we are trying to establish the $C_{ik}(t_j^-)$ in terms of the $C_{ik}(t_j^+)$ at each j , for each i , and for all k , $k = 1, 2, 3$.

Let us choose

$$C_{i2}(t_j^-) = 0 \quad (50)$$

$$C_{i3}(t_j^-) = 0 \quad (51)$$

for all j and for all i . So now we must simply find $C_{i1}(t_j^-)$ in terms of the $C_{ik}(t_j^+)$ for $k = 1, 2$, and 3 .

Due to their interrelatedness, we must simultaneously consider solving for the boundary conditions $y_3(t_j^-)$ from (18), (20) and (44) and solving for the $C_{i1}(t_j^-)$ in terms of the $C_{ik}(t_j^+)$, $k = 1, 2, 3$, from (49).

To minimize the required computations, it is advantageous if $y_3(t)$ is broken up

$$y_3 \stackrel{\Delta}{=} \begin{bmatrix} 1 \\ y_3 \\ \text{-----} \\ 2 \\ y_3 \end{bmatrix} \stackrel{\Delta}{=} \begin{bmatrix} \lambda \\ \text{-----} \\ p_1 \\ \vdots \\ p_m \end{bmatrix}. \quad (52)$$

The derivation of the boundary conditions for the j th interval $[t_j, t_{j+1})$ proceeds as follows. From (29), (44), (50) and (51)

$$y_3^2(t_{j+1}^-) = \begin{bmatrix} C_{11} \\ C_{21} \\ \vdots \\ C_{m1} \end{bmatrix} \bigg|_{t_{j+1}^-} y_1(t_{j+1}) \quad (53)$$

and

$$\begin{aligned} y_3^1(t_{j+1}^-) &= [C_{01}y_1 - \bar{C}(F_1y_1 + F_3y_3)] \bigg|_{t_{j+1}^-} \\ &= [C_{01}y_1 - \bar{C}(F_1y_1 + F_3^1y_3^1 + F_3^2y_3^2)] \bigg|_{t_{j+1}^-} \end{aligned} \quad (54)$$

where $\bar{C} = [C'_{11} \ : \ C'_{21} \ : \ \dots \ : \ C'_{m1}]$ and where F_3 is broken up into $F_3 = [F_3^1 \ : \ F_3^2]$ with F_3^1 and F_3^2 having dimensions which correspond to y_3^1 and y_3^2 . By substituting (53) into (54), equation (54) becomes

$$y_3^1(t_{j+1}^-) = [C_{01}y_1 - \bar{C}(F_1y_1 + F_3^1y_3^1 + F_3^2\bar{C}'y_1)] \bigg|_{t_{j+1}^-}$$

so

$$y_3^1(t_{j+1}^-) = [(I + \bar{C}F_3^1)^{-1}(C_{01} - \bar{C}(F_1 + F_3^2\bar{C}'))y_1] \bigg|_{t_{j+1}^-} \quad (55)$$

Combining (53) and (55) defines D_{j+1}

$$y_3(t_{j+1}^-) = D_{j+1}y_1(t_{j+1}) \quad (56)$$

where

$$D_{j+1} = \begin{bmatrix} (I + \bar{C}F_3^1)^{-1}(C_{01} - \bar{C}(F_1 + F_3^2\bar{C}')) \\ \hline \bar{C}' \end{bmatrix} \bigg|_{t_{j+1}^-} \quad (57)$$

By breaking up y_3 as in (52) we need only invert a matrix of dimension n , the system dimension, to obtain D_{j+1} . Otherwise we would have had to invert a matrix of dimension $n \cdot (m+1)$.

To find the $C_{i1}(t_{j+1}^-)$ we also need a relationship between $y_3(t_{j+1}^-)$ and $y_1(t_j)$. That is, from (40) and (56) we can find

$$y_3(t_{j+1}^-) = E_j y_1(t_j) \quad (58)$$

where

$$E_j = D_{j+1} (I - H_3(t_{j+1}^-) D_{j+1})^{-1} H_1(t_{j+1}^-). \quad (59)$$

So, from (44), (49), (50), (51) and (58)

$$C_{i1}(t_j^-) = C_{i1}(t_j^+) + E_j' C_{i2}(t_j^+) E_j + 2C_{i3}(t_j^+) E_j. \quad (60)$$

We now have all of the required boundary conditions. The cost to go boundary conditions are (48), (50), (51) and (60) and the costate boundary conditions are (56) or (58).

Solution of the Cost to Go Equations

In each interval, we do not need the cost to go for all $t \in [t_j, t_{j+1})$ but rather we only need the value at the initial boundary, i.e., we only need to solve for the $C_{ik}(t_j)$ in terms of the $C_{i1}(t_{j+1}^-)$.

The cost to go equations, (45) through (47), are the same for each interval and only the boundary conditions change. In order to avoid resolving these equations in each interval, we will assume a functional dependence of the cost to go matrices on their boundary conditions, similar to the technique used on the costates. Since the cost to go equations are linear, we can find

such a functional dependence. It will be independent of the changing boundary conditions and can therefore be presolved. The solution of the function will be valid for each interval.

For notational convenience, we will "stack" the columns of the cost to go matrices so that the matrix equations (45) through (47) can be written as vector equations. Let \tilde{c}_{ik} be the vector corresponding to the matrix C_{ik} . Define \tilde{c}_i as

$$\tilde{c}_i \triangleq \begin{bmatrix} \tilde{c}_{i1} \\ \tilde{c}_{i2} \\ \tilde{c}_{i3} \end{bmatrix}. \quad (61)$$

Then (45) through (47) can be rewritten as

$$\dot{\tilde{c}}_i = \tilde{A}_i \tilde{c}_i + \tilde{b}_i \quad (62)$$

where the matrix \tilde{A}_i and the vector \tilde{b}_i are known from the coefficient matrices of (45) through (47). We can now solve for the functional dependence of the solution of (62) in the j th interval on the boundary condition $\tilde{c}_i(t_{j+1}^-)$. Actually, since $\tilde{c}_{i2}(t_{j+1}^-) = 0$ and $\tilde{c}_{i3}(t_{j+1}^-) = 0$, we need only assume dependence of the solution on $\tilde{c}_{i1}(t_{j+1}^-)$, i.e., for $t \in [t_j, t_{j+1})$ assume

$$\tilde{c}_i(t) = M_i(t) \tilde{c}_{i1}(t_{j+1}^-) + d_i(t). \quad (63)$$

From (62) and (63) it follows that

$$\dot{M}_i = \tilde{A}_i M_i, \quad M_i(t_{j+1}^-) = \begin{bmatrix} I \\ \text{---} \\ 0 \end{bmatrix} \quad (64)$$

$$\dot{d}_i = \tilde{A}_i d_i + \tilde{b}_i, \quad d_i(t_{j+1}^-) = 0 \quad (65)$$

where the dimension of the identity matrix in $M_i(t_{j+1}^-)$ is the same as the dimension of \tilde{c}_{i1} .

If the system is time invariant and if the sampling rate is constant then (64) and (65) need be solved only once over one sampling interval. In fact, only the value of $M_i(t_j^+)$ and $d_i(t_j^+)$ need be stored since we only need $\tilde{c}_i(t_j^+)$ in terms of $\tilde{c}_{i1}(t_{j+1}^-)$. That is

$$\tilde{c}_i(t_j^+) = M_i(t_j^+) \tilde{c}_{i1}(t_{j+1}^-) + d_i(t_j^+) \quad (66)$$

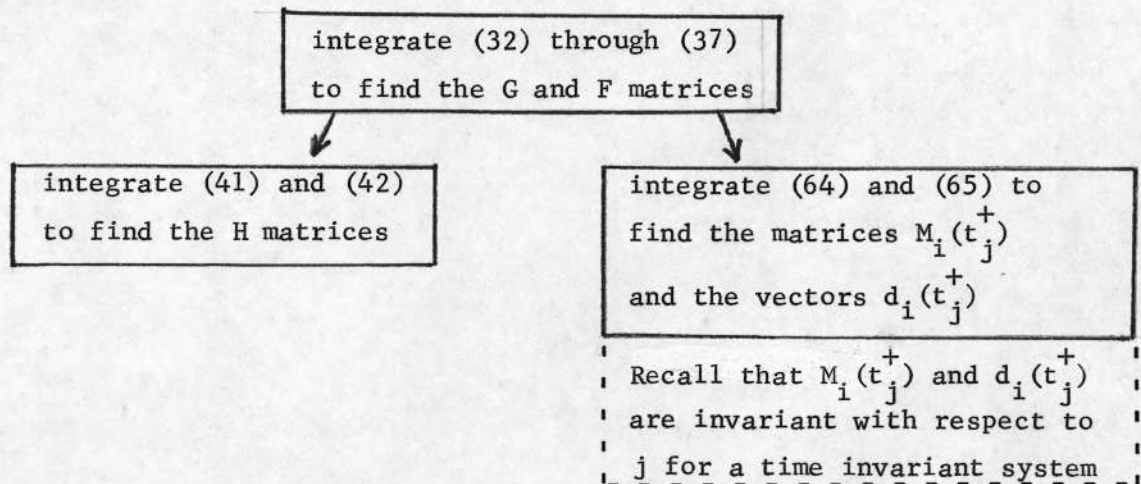
where $M_i(t_j^+)$ and $d_i(t_j^+)$ are the same for all j .

Due to the relationship (63), we will not have to solve the cost to go equations (45) through (47) repeatedly for each sample interval but need only plug into (66).

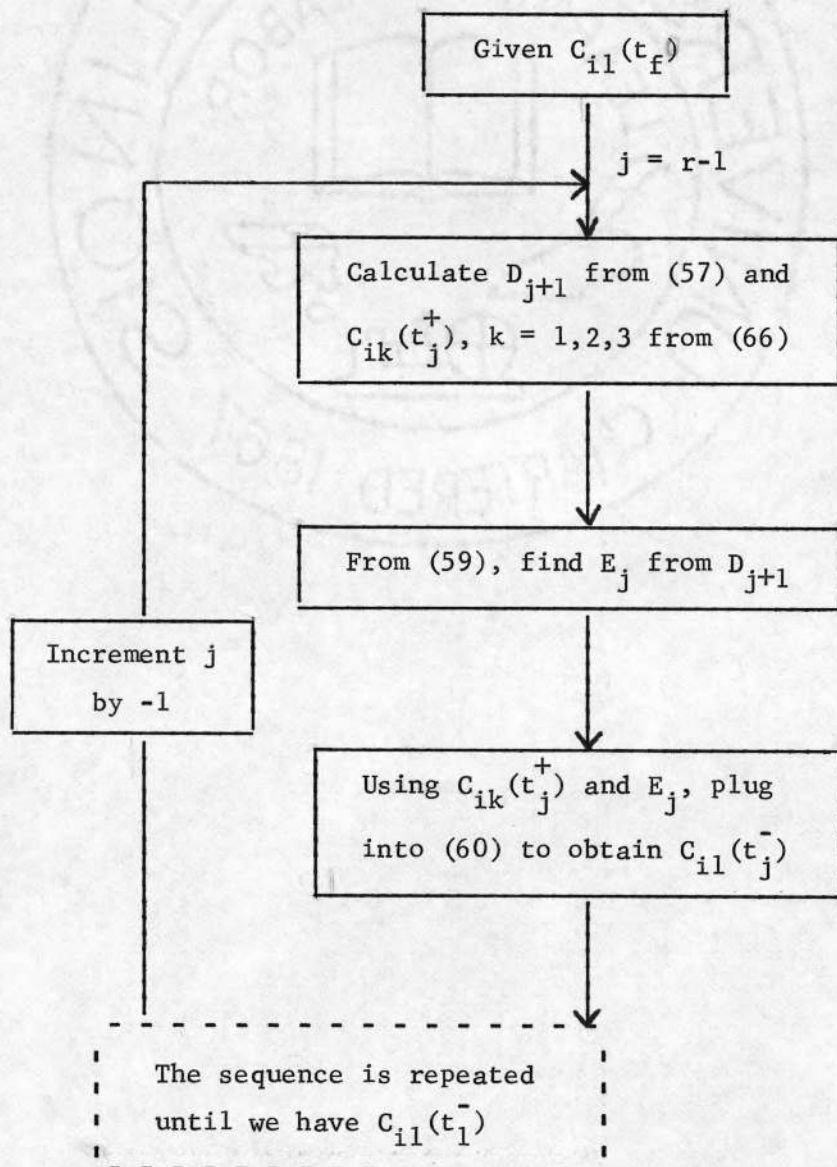
Summary of Algorithm

We will now summarize the required calculations in the following flow chart. The major steps and reference to the related equations are given in the order in which they must be computed.

All integrations are performed over only one sample interval if the system is time invariant.



Going backwards from $j = r-1$ to $j = 1$, beginning with the known $C_{i1}(t_f)$ from (48), the following calculations must be done for each j in order to obtain the boundary conditions for each interval.



Implementation

The controls can now be implemented forward in time. They are found by (19), (22), the definitions of y_3 , i.e., $y_3 = (\lambda' \vdots p_1' \vdots \dots \vdots p_m')'$, and $y_1 = x$, and the evolution of $y_3(t)$ in each interval, $t \in [t_j, t_{j+1})$ given by

$$y_3(t) = P(t)y_1(t_j) \quad (67)$$

where

$$P(t) = [\bar{G}_1(t)(H_1(t) + H_3(t)E_j) + \bar{G}_3(t)E_j] \quad (68)$$

which is derived from (39), (40) and (58).

If $P(t)$ is broken up as

$$P(t) = \begin{bmatrix} P_0(t) \\ P_1(t) \\ \vdots \\ P_m(t) \end{bmatrix}$$

where each block $P_i(t)$ is n by n , then the i th control during the j th interval is

$$u_i(t) = -R_{ii}^{-1}B_i'P_i(t)x(t_j).$$

As outlined above, there are a number of equations to be integrated, some of which are of large dimension. These integrations, however, are done once only and are performed over a period equal to the length of only one sample interval. Thus, as the number of samples taken increases, the computational burden is reduced. Computationally the only limiting factor which prohibits us from allowing the length of the sample intervals to become

arbitrarily small is the corresponding increase in the number of matrix inversions which must be performed at the sampling times in order to generate the required boundary conditions for each interval. That is, as the period of integration becomes smaller, these matrix inversions will tend to become the dominant computational burden. The matrix inversions present another difficulty since, in general, we are unable to guarantee their existence.

Comparison of Techniques

The first technique discussed at the beginning of this section is a method for converting the problem of repeatedly solving m coupled matrix Riccati equations to that of solving m uncoupled matrix Riccati equations providing significant computational savings. These equations, however, must still be solved repeatedly for each sample interval with only a change in the boundary conditions.

The second approach, which we have derived in detail, requires a set of linear and Riccati equations to be solved once only over a horizon which is the length of only one sample interval. The computational advantage of this second technique is due to the fact that the integrations are performed over only one sample interval which is, in general, considerably shorter than the time horizon of the original problem.

5. Conclusions

In this paper a sampled data Stackelberg strategy has been considered. The advantages of the sampled data formulation can be seen by considering certain characteristics of the continuous time Stackelberg problem. The linear quadratic, continuous time, closed loop Stackelberg problem results in a solution, if it exists, in which the controls are non-linear functions of the

state. Furthermore, the Stackelberg solution for general dynamic games does not, in general, satisfy the principle of optimality. The principle of optimality can be imposed for discrete time games but the procedure for doing this for continuous time games has not been established.

The sampled data solution results in linear control laws for the linear quadratic case and the solution is constructed so that the principle of optimality holds at the sample times. The advantage of linear control laws is that they are quite simple to implement. The principle of optimality is particularly advantageous in a game formulation in that we might not otherwise be able to insure against any player deviating from his predetermined controls.

In deriving the sampled data solution we have been able to obtain considerable computational savings. That is, rather than performing integrations over the entire time horizon of the original problem, we are able to imbed the subproblems of each sample interval into a more general formulation, the solution of which requires integrations over a period equal to the length of only one sample interval. The computational technique, an application of invariant imbedding developed for the particular case of a Stackelberg strategy and the type of boundary conditions peculiar to it, is quite useful for many problems, in particular for a variety of sampled data formulations.

References

- [1] C. I. Chen and J. B. Cruz, Jr., "Stackelberg Solution for Two Person Games with Biased Information Patterns," IEEE Trans. Automatic Control, Vol. AC-17, No. 6, 1972, pp. 791-798.
- [2] M. Simaan and J. B. Cruz, Jr., "On the Stackelberg Strategy in Nonzero-Sum Games," J. Opt. Theory Appl., Vol. 11, No. 5, 1973, pp. 533-555.
- [3] D. Castanon and M. Athans, "On Stochastic Dynamic Stackelberg Strategies," Automatica, Vol. 12, 1976, pp. 177-183.
- [4] M. Simaan, "Stackelberg Optimization of Two-Level Systems," Proc. 1977 JACC, June 1977, San Francisco, pp. 1113-1118.
- [5] J. B. Cruz, Jr., "Stackelberg Strategies for Multilevel Systems," in Directions in Large Scale Systems, Y. C. Ho and S. K. Mitter, editors, Plenum Publishing Corp., New York, 1976, pp. 139-147.
- [6] J. Medanic, "Closed-Loop Stackelberg Strategies in Linear-Quadratic Problems," Proc. 1977 JACC, June 1977, San Francisco, pp. 1324-1329.
- [7] J. B. Cruz, Jr., "Survey of Nash and Stackelberg Equilibrium Strategies in Dynamic Games," Annals of Economic and Social Measurement, Vol. 4, No. 2, 1975, pp. 339-344.
- [8] M. Simaan and J. B. Cruz, Jr., "Additional Aspects of the Stackelberg Strategy in Nonzero-Sum Games," J. Opt. Theory Appl., Vol. 11, No. 6, June 1973, pp. 613-626.
- [9] M. Simaan and J. B. Cruz, Jr., "Sampled Data Nash Controls in Nonzero-Sum Differential Games," Int. J. of Control, Vol. 17, 1973, pp. 1201-1209.
- [10] M. Simaan and J. B. Cruz, Jr., "On the Solution of the Open Loop Nash Riccati Equations in Linear Quadratic Differential Games," Int. J. of Control, Vol. 18, 1973, pp. 57-63.
- [11] M. Simaan and J. B. Cruz, Jr., "A Stackelberg Strategy for Games with Many Players," IEEE Trans. Automatic Control, Vol. AC-18, No. 3, 1973, pp. 322-324.
- [12] C. Y. Chong and M. Athans, "On the Periodic Coordination of Linear Stochastic Systems," Automatica, Vol. 12, 1976, pp. 321-335.
- [13] R. Bellman and G. M. Wing, An Introduction to Invariant Imbedding, Wiley-Interscience, New York, 1975.
- [14] E. D. Denman, Coupled Modes in Plasmas, Elastic Media and Parametric Amplifiers, American Elsevier, New York, 1971.