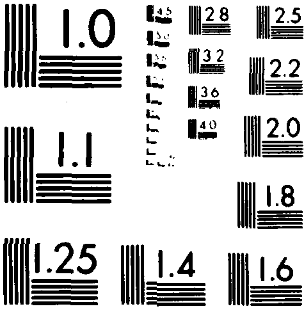
AD-A085 763   ILLINOIS UNIV AT URBANA-CHAMPAIGN DECISION AND CONTROL LAB   F/G 12/2
HIERARCHICAL COMPUTATION OF GAINS FOR THE DECENTRALIZFD LINEAR --ETC(U)
JAN 80   J V PIETRAS   N00014-79-C-0424
UNCLASSIFIED   DC-35   NL

1 of 1
AD
A085763

END
DATE
FILMED
7-80
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A085 763	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 HIERARCHICAL COMPUTATION OF GAINS FOR THE DECENTRALIZED LINEAR QUADRATIC REGULATOR PROBLEM		5. TYPE OF REPORT & PERIOD COVERED Technical Rep[redacted]t
		6. PERFORMING ORG. REPORT NUMBER R-871(DC-35); UILU-80-2203
7. AUTHOR(s) 10 JOHN VICTOR/PIETRAS	15	8. ~~CONTRACT OR GRANT NUMBER(s)~~ N00014-79-C-0424
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, Illinois 61801		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12 54
11. CONTROLLING OFFICE NAME AND ADDRESS Joint Services Electronics Program	11	12. ~~REPORT DATE~~ Jan[redacted]80
		13. ~~NUMBER OF PAGES~~ 45
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 14 DC-35, UILU-80-2203		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the ~~abstract entered in Block 20,~~ if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Decentralized contrel
Hierarchical control
Decomposition
Optimization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The subject of this report is the application of a hierarchical computation structure to the computation of fixed decentralized state feedback gains for the regulation of linear systems with fixed (but arbitrary) dimension. This category of control problems encompasses many practical examples. In addition to those cases where the system is naturally linear, this framework extends to those nonlinear problems that can be characterized (via linearization) as linear systems driven by white noise.

The decomposition technique that is utilized is that of interaction

410715

20. ABSTRACT (continued)

prediction. As will be shown, this particular method has the desirable characteristic of minimal computations at the supremal level. Because of the limitation of information available for feeding back under the decentralized control structure, the detemrination of the optimal feedback gains is dependent upon the (expected) values of the initial conditions. Thus the control which is derived is open loop. Although extensions to the basic methodology exist for forming closed loop controllers, the approach used here is totally off-line calculation of the closed loop optimal gains.

A

UILU-80-2203

HIERARCHICAL COMPUTATION OF GAINS FOR THE DECENTRALIZED LINEAR QUADRATIC REGULATOR PROBLEM

by

John Victor Pietras

This work was supported in part by the Joint Services Electronics Program (U. S. Army, U. S. Navy, and U. S. Air Force) under Contract N00014-79-C-0424

HIERARCHICAL COMPUTATION OF GAINS FOR THE DECENTRALIZED LINEAR QUADRATIC REGULATOR PROBLEM

BY

JOHN VICTOR PIETRAS

B.S., Case Western Reserve University, 1976

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1980

Urbana, Illinois

ACKNOWLEDGEMENT

I would like to express my gratitude to Professor Douglas P. Looze, whose guidance and support made the research for this thesis a pleasant and rewarding experience.

TABLE OF CONTENTS

TABLES

FIGURES

1. INTRODUCTION

The study of large scale systems and the design of control structures for them has promoted the development of a number of methodologies for the classification of such systems and the structures by which they can be controlled. Two such control structures are those of hierarchical control and decentralized control. Although these two structures are often thought to imply one another, they in fact deal with different aspects of the problem of controlling large scale systems.

Hierarchical control theory ([1]-[4]) is founded on the decomposition techniques of mathematical programming ([5]). It is concerned with the partitioning of the calculations required for the computation of the system control. This control is then computed iteratively by coordination at a global level of the subproblems formed by the partitioning of the calculations. The motivation for hierarchical control theory is the opportunity for savings in computer time and space given by decomposition techniques.

Decentralized control theory ([6]-[8]) is concerned not with how the control is computed, but rather with how the control utilizes the information present in the state of the system. Specifically, a decentralized control structure restricts the states which can be fed back to form the various inputs to the system. Although such a feedback law is suboptimal in comparison with full state feedback, its desirability is seen in the elimination of various information links, which in a physically distributed system can be prohibitively expensive.

Unfortunately, decentralized control laws are usually more difficult to compute than centralized (full state feedback) laws. Thus the application of a hierarchical structure, with its promise of reduced computation cost, to the calculation of a decentralized control is very appealing. Furthermore, the two control structures utilize localized dynamics: hierarchical control in terms of subproblem solution, and decentralized control in terms of which states are available for control formulation. The decomposition of the system for hierarchical purposes can be done in accordance with the partition of the state defined by the decentralized feedback information flow pattern.

The subject of this report is the application of a hierarchical computation structure to the computation of fixed decentralized state feedback gains for the regulation of linear systems with fixed (but arbitrary) dimension. This category of control problem encompasses many practical examples. In addition to those cases where the system is naturally linear, this framework extends to those nonlinear problems that can be characterized (via linearization) as linear systems driven by white noise ([9],[10]).

The decomposition technique that is utilized is that of interaction prediction. As will be shown, this particular method has the desirable characteristic of minimal computations at the supremal level. Because of the limitation of information available for feeding back under the decentralized control structure, the determination of the optimal feedback gains is dependent upon the (expected) values of the initial conditions. Thus the control which is derived is open loop. Although extensions to the basic methodology exist for forming closed loop controllers (see, for example [11]), the approach used here is a totally off-line calculation of the closed loop optimal gains.

Chapter 2 presents the formulation of the problem under consideration, first in terms of optimization of a linear dynamic system with respect to a linearly quadratic (LQ) cost function, and then as an equivalent static optimization problem. Chapter 3 formulates the interaction prediction algorithm as a partitioning of the standard mathematical programming problem. In Chapter 4, this prediction algorithm is applied to the problem under consideration. The various computational aspects of the prediction principle as applied to this problem are presented in the form of a comparison with the non-decomposed (full order) solution algorithm in Chapter 5. Chapter 6 introduces a demonstration model and presents results and interpretation of tests with said model, plus discussion on application to control of power systems. Finally, Chapter 7 summarizes the major results of the report.

2. PROBLEM FORMULATION

Consider the N linear, time-invariant, interconnected systems

$$\dot{x}_i = A_{ii}x_i + \sum_{i\neq j} A_{ij}x_j + B_i u_i + \omega_i, \quad i = 1,N$$
$$x_i \in R^{n_i}, \; u_i \in R^{m_i}, \; \omega_i \in R^{n_i}, \tag{2.1}$$

where $\omega(t)$ is a zero mean white noise process with spectral density Ξ. The objective is to choose the N controls u_i in order to minimize the quadratic, time-averaged cost functional:

$$J = \lim_{t_f \to \infty} \frac{1}{t_f} E\{\sum_i \int_o^{t_f} [x_i^T Q_i x_i + u_i^T R_i u_i] dt\} \tag{2.2}$$

where $Q_i = Q_i^T \geq 0$ and $R_i = R_i^T > 0$. These N controls are to be formed by the linear decentralized state feedback control law

$$u_i = -G_i x_i \tag{2.3}$$

Define

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, \; u = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix}, \; \omega_i = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_N \end{bmatrix}$$

$$A = [A_{ij}], \; B = \text{diag}[B_i], \; G = \text{diag}[G_i], \; Q = \text{diag}[Q_i],$$
$$R = \text{diag}[R_i], \; \Xi = \text{diag}[\Xi_{ij}]$$

Then the overall problem (2.1)-(2.3) can be rewritten as

$$\dot{x} = Ax + Bu + \omega \tag{2.4}$$

$$x \in R^n, \; u \in R^m, \; \omega \in R^n, \; n = \sum_{i=1}^N n_i, \; m = \sum_{i=1}^N m_i$$
$$J = E\{\lim_{t_f \to \infty} \frac{1}{t_f} \int_o^{t_f} [x^T Q x + u^T R u] dt\} \tag{2.5}$$

$$u = -Gx \tag{2.6}$$

The adoption of the control law (2.6) results in a closed loop feedback matrix parameterized by G, $A_c(G)$. Furthermore, the cost J also becomes parameterized by G.

Since the integrand of (2.5) is non-negative, the Fubini Theorem implies that the integrand and expectation can be interchanged. Exploiting the linearity of the trace, integration, and expectation operators:

$$\begin{aligned} J(G) &= \lim_{t_f \to \infty} \frac{1}{t_f} \int_o^{t_f} E\{x^T(t)(Q + G^T RG)x(t)\}dt \\ &= \lim_{t_f \to \infty} \frac{1}{t_f} \int_o^{t_f} E\{tr[(Q + G^T RG)x(t)x^T(t)]\}dt \\ &= tr\{(Q + G^T RG)[\lim_{t_f \to \infty} \frac{1}{t_f} \int_o^{t_f} E\{x(t)x^T(t)\}dt]\} \end{aligned} \tag{2.7}$$

Define $P(t) \overset{\Delta}{=} E\{x(t)x^T(t)\}$. Then

$$\lim_{t_f \to \infty} \frac{1}{t_f} \int_o^{t_f} P(t)dt = P \tag{2.8}$$

For all G such that A - BG is asymptotically stable.

The problem (2.4)-(2.6) has thus been transformed into the equivalent static minimization problem:

$$\min_{G \in \mathcal{S}} J(G) = tr\{(Q + G^T RG)P\} \tag{2.9}$$

subject to

$$(A-BG)P + P(A-BG)^T + \Xi = 0 \tag{2.10}$$

where

$$\mathcal{S} \overset{\Delta}{=} \{G : (A-BG) \text{ is asymptotically stable}\}$$

3. INTERACTION PREDICTION

As was stated in the introduction, a hierarchical control structure presents several theoretical advantages over a full-system control structure, some of which are possible distribution of processing load and reduction of the overall computational effort. Several decomposition algorithms exist for which such a hierarchy can be implemented. Among these are the goal coordination, primal,and interaction prediction algorithms. The interaction prediction method will be used to decompose (2.9)-(2.10) into a three level hierarchy. The following is a brief description of this algorithm.

Consider the minimization problem

$$\min_{\substack{u\in U\\ z\in Z}} J(u,z) \tag{3.1}$$

subject to

$$z = g(u) \tag{3.2}$$

$$h(u,z) = 0 \tag{3.3}$$

where U, $Y \subset U$ and Z are Banach spaces and $J: U \times Z \rightarrow R$, $g: U \rightarrow Z$ and $h: U \times Z \rightarrow Y$ are twice continuously Frechet differential functions with $\mathcal{R}\{h_u\} = Y$.

The Lagrangian for the problem (3.1)-(3.3) is given by

$$L(u,z,p,\lambda) = J(u,z) + \langle p, z-g(u)\rangle + \langle \lambda, h(u,z)\rangle \tag{3.4}$$

The stationarity condition, which must be satisfied at the (local) optimal solution, is given by the set of equations:

$$L_u(u,z,p,\lambda) = J_u(u,z) - g_u(u)p + h_u(u,z)\lambda = 0 \qquad (3.5)$$

$$L_z(u,z,p,\lambda) = p + J_z(u,z) - h_z(u,z)\lambda = 0 \qquad (3.6)$$

$$L_p(u,z,p,\lambda) = z - g(u) = 0 \qquad (3.7)$$

$$L_\lambda(u,z,p,\lambda) = h(u,z) = 0 \qquad (3.8)$$

Any optimization technique applied to the problem (3.1)-(3.3) must satisfy the stationarity condition (3.5)-(3.8) at the optimum. The distinction between the various decompositions is the way in which the equations of the stationarity condition are partioned among the supremal and infimal levels of the hierarchy [12]. For interaction prediction, the supremal problem involves enforcing the constraints (3.6) and (3.7):

$$f_s(p,z,u,\lambda) = \begin{bmatrix} L_p \\ L_z \end{bmatrix} = \begin{bmatrix} z - g(u) \\ p + J_z(u,z) + h_z(u,z)\lambda \end{bmatrix} \qquad (3.9)$$

with p and z designated the supremal variables. The infimal enforces (3.5) and (3.8):

$$f_I(p,z,u,\lambda) = \begin{bmatrix} L_u \\ L_\lambda \end{bmatrix} = \begin{bmatrix} J_u(u,z) - g_u(u)p + h_u(u,z)\lambda \\ h(u,z) \end{bmatrix} \qquad (3.10)$$

with u and λ the infimal variables. The problems f_s and f_I are solved in an iterative fashion, with each step solving the iteration equations:

$$F_s(p^{k+1},z^{k+1},u^k,\lambda^k) = 0 \qquad (3.11)$$

$$F_I(p^{k+1},z^{k+1},u^{k+1},\lambda^{k+1}) = 0 \qquad (3.12)$$

4. APPLICATION OF THE INTERACTION PREDICTION ALGORITHM TO THE DECENTRALIZED CONTROL PROBLEM

The purpose of this chapter is to present the interaction prediction decomposition as it applies to the decentralized control problem. The approach will be to reformulate the static minimization problem of Chapter 2 in terms of the standard mathematical programming problem of Chapter 3. Using this approach, the supremal and infimal equations follow directly.

In order to put (2.9)-(2.10) into the form of (3.1)-(3.3), the open loop system matrix is split into block diagonal and off-diagonal matrices:

$$A_d \triangleq [A_{ii}] \qquad i = 1,N \tag{4.1}$$

$$A_o = A - A_d \tag{4.2}$$

Introduce the interaction matrix Z:

$$Z \triangleq A_o P + PA_o^T \tag{4.3}$$

then the constraint (2.8) becomes

$$Z = A_o P + PA_o^T \tag{4.4}$$

$$(A_d - BG)P + P(A_d - BG)^T + \Xi + Z = 0 \tag{4.5}$$

the problem defined by (2.7) subject to (4.4)-(4.5) is then exactly of the form of equations (3.1)-(3.3). The Lagrangian for this problem is:

$$L(G,P,Z,K,\Lambda) = \mathrm{tr}\{[Q + G^T RG]P + K[(A_d - BG)P + P(A_d - BG)^T + \Xi + Z] + \Lambda[A_o P + PA_o^T - Z]\} \tag{4.6}$$

Where Λ and K serve as Lagrange multiplier matrices for (4.4) and (4.5), respectively.

The stationarity condition must, of course, be satisfied. The resulting five stationarity equations are:

$$L_G(G,P,Z,K,\Lambda) = RGP_d - B^T(KP)_d = 0 \tag{4.7}$$

$$\begin{aligned} L_P(G,P,Z,K,\Lambda) &= Q + G^TRG + (A_d - BG)^TK + K(A_d - BG) + A_o^T\Lambda \\ &\quad + \Lambda A_o = 0 \end{aligned} \tag{4.8}$$

$$L_Z(G,P,Z,K,\Lambda) = K - \Lambda = 0 \tag{4.9}$$

$$L_K(G,P,Z,K,\Lambda) = (A_d - BG)P + P(A_d - BG)^T + \Xi + Z = 0 \tag{4.10}$$

$$L_\Lambda(G,P,Z,K,\Lambda) = A_oP + PA_o^T - Z = 0 \tag{4.11}$$

Equation (4.7) follows from Theorem 3 in [13]. (4.8)-(4.11) are straightforward calculations.

Following the procedure described in Chapter 3, equations (4.9) and (4.11) are associated with the supremal level, and (4.6), (4.8) and (4.10) are assigned to the infimal.

The resultant hierarchical algorithm is based on the iteration of the equations:

$$\text{Supremal} \begin{cases} Z_{k+1} = A_oP_k + P_kA_o^T & (4.12) \\ \Lambda_{k+1} = K_k & (4.13) \end{cases}$$

$$\text{Infimal} \begin{cases} RG_kP_{dk} - B^T(K_kP_k)_d = \nabla_G J & (4.14) \\ (A_d - BG_k)P_k + P_k(A_d - BG_k)^T + W_k = 0 & (4.15) \\ (A_d - BG_k)^TK_k + K_k(A_d - BG_k) + G_k^TRG_k + V_k = 0 & (4.16) \end{cases}$$

where

$$W_k = \Xi + Z_k \tag{4.17}$$

and

$$V_k = Q + A_o^T \Lambda_k + \Lambda_k A_o^T \tag{4.18}$$

are formed at the supremal level. The subscript k is the iteration index.

It should be noted that for this realization of the decomposition, the gradient equation (4.14) couples the subsystems, and as such might seem inappropriately placed in the infimal. However, because $A_{cd} = A_d - BG$ is a block diagonal matrix, the Lyapunov equations (4.15) and (4.16) can be decomposed into a number of smaller Lyapunov and Sylvester equations corresponding to the subsystems of the infimal level. Thus computations of P and K are executed in a decentralized manner. The solution for P and K presents the greatest computational complexity of the problem, and this load is distributed over the infimal subproblems. Only the multiplication $(KP)_d$ is done in a centralized manner.

It is this decomposition of the Lyapunov equations that in fact is the motivation for the use of the decomposition. Because the number of multiplications involved in the solution of an n-dimension Lyapunov equation is on the order of n^3, reduction of the system dimension by decomposition reduces computational load. Of course, the coupling of the problem (represented by the off-diagonal elements of A) must be taken into account. This is accomplished at the supremal level of the hierarchy. Considerations of computational efficiency are the subject of Chapter 5.

Figure 4.1 is a block diagram of the hierarchical structure of the algorithm. It has the form of a three level hierarchy with the bottom

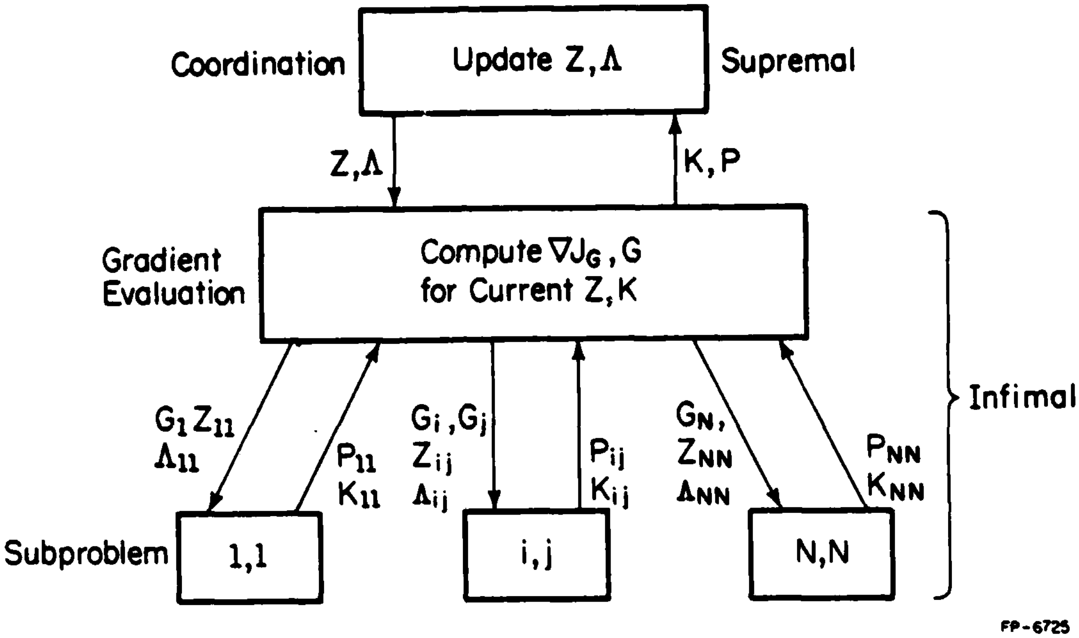


Figure 4.1. Hierarchical Structure of the Algorithm for the Computation of Decentralized Gains.

level solving for the submatrices P_{ij} and K_{ij} in a decoupled manner.

5. COMPUTATIONAL EFFICIENCY OF DECOMPOSITION

In this Chapter a comparison is made of the computational requirements of the decomposition approach with those of the full-order solution technique. The measure of computation is taken to be the number of double precision floating point (DPFP) multiplications performed in vector and matrix operations.[1] Counts are limited to vectors and matrices because these grow as the order of the system, and it is the trend toward more efficient computation as n gets large that is of interest.

The method of optimization employed involves an accelerated gradient search which requires the evaluation of the performance index for a test feedback gain matrix, and the gradient of the index at the test gain matrix. The calculations required to evaluate the performance index (hereafter designated CALCF) for a given gain matrix G can be separated into three distinct computations:

a) Formation of the closed loop matrix

$$\bar{A} = A - BG;$$

b) Solution of the Lyapunov equation

$$\bar{A}P + P\bar{A}^T + W = 0; \quad \text{and}$$

c) Evaluation of the cost function

$$\text{cost} = 1/2\,\text{tr}\{\bar{Q}P\},$$

where $\bar{Q} = G'RG + V$

[1] The DPFP multiplication is the most time-consuming single operation performed, making it a good index of computational complexity. Furthermore, the number of DPFP additions associated with these matrix operations is nearly the number of the multiplications. Thus the multiplication count is a very close approximation to the addition count.

Similarly, the calculations required to derive the gradient (CALCG) of the cost at G can be separated into the two computations:

d) Solution of the Lyapunov equation

$$\bar{A}^T K + K\bar{A} + \bar{Q} = 0; \quad \text{and}$$

e) Evaluation of the gradient

$$GRAD = RGP_d - B^T(KP)_d$$

Subchapter 5.1 contains an analysis of the computations required to solve the Lyapunov equations, first for the full order method and then for the decomposition approach the other computations are then presented in Subchapter 5.2. Finally, a comparison of the efficiency of the two techniques, and the resulting implications on system structure, are presented in Subchapter 5.3.

5.1. Lyapunov Equations

5.1.1. Full Order Evaluation (Table 5.1)

Both CALCF and CALCG require the solution for X of Lyapunov equations of the form

$$RX + XR^T + T = 0,$$

where R, X, and T are n x n matrices. The solution of this equation employs the technique of Bartels and Stewart [14]. This metnod first involves the reduction of R^T to Real Schur Form (RSF), and then employs a back substitution to produce the solution. This two-step approach allows for reduced computation when the system matrix R remains fixed and only the driving matrix T changes. This feature is exploited in this application. Each of

Group	Eqn. Ref. No.	Operation	Multiplication Count
CALCF	1	Closed Loop System Matrix	$\sum_{i=1}^{N} n_i^2 m_i$
		$A_c = A - BG$	
	2	Lyapunov Equation	
		$A_c P + PA_c^T + \Xi = 0$	
	(a)	RSF of A_c	$(2 + 4\sigma)n^3$ [see footnote 2]
	(b)	Back Substitution for P	$7/2\ n^3$
	3	Closed Loop Weighting Matrix	
		$\overline{Q} = G^T RG + Q$ For	$\sum_{i=1}^{N} m_i n_i (2m_i + n_i + 1)$
		Cost $= tr(\overline{Q}P)/2$	
CALCG	4	Back Substiution for K in	$7/2\ n^3$
		$A_c^T K + KA_c + \overline{Q} = 0$	
	5	Evaluation of the Gradient	
		$GRAD = RGP_d - B^T (KP)_d$	$\sum_{i=1}^{N} n_i (m_i^2 + 2m_i n_i + nn_i)$
Gradient Search	6	For Each Call on CALCF or CALCG	$O[(\sum_{i=1}^{N} (n_i m_i))^2]$

Table 5.1. Computation Counts for Full-Order Evaluation.

[2] σ is the Average Number of QR Steps Required to make a Subdiagonal Element Negligible. It is usually Overestimated by $\sigma = 1.5$.

the two steps, RSF reduction and back substitution, requires on the order of n^3 multiplications.

In CALCF, the equation to be solved is

$$A_cP + PA_c^T + \Xi = 0$$

Because A_c is newly evaluated for each call on CALCF, it must first be reduced to RSF before back substitution (entries 2a and 2b in Table 5.1). In CALCG, the Lyapunov equation that must be solved involves the transpose of the system matrix of CALCF:

$$A_c^TK + KA_c + \bar{Q} = 0$$

The RSF of a matrix can be produced from the RSF of its transpose by simple interchange of rows and columns ([14],[15]). Furthermore, CALCG is only performed for a particular closed loop system A_c after the cost for A_c is found in CALCF. Thus the RSF of A_c need be found only once for both CALCF and CALCG, leaving only the back substitution step to be performed in CALCG (entry 4).

5.1.2. Decomposition Method (Table 5.2)

In the decomposition approach, the nth order Lyapunov equations are solved by decomposition into subequations corresponding to the N subsystems. Because only the block diagonal elements of the closed loop system matrix, A_{cd}, are used, the subsystem Lyapunov equations decouple. Thus in CALCF, the diagonal blocks of P are found via:

$$A_{cd_i}P_{ii} + P_{ii}A_{cd_i}^T + W_{ii} = 0, \quad i = j$$

The off-diagonal blocks of P must be found via the more general Sylvester

Function Group	Eqn. Ref. No.	Operation	Multiplication Count
CALCF	1	Diagonal Closed Loop System Matrix $A_{cd} = A_d - BG$	$\sum_{i=1}^{N} n_i^2 m_i$
	2	Lyapunov Equation Solved Via N(N+1)/2 Subequations:	
	(a)	RSF Reduction of Diagonal Blocks	$(2+4\sigma)\sum_{i=1}^{N} n_i^3$
	(b)	Back Substitution for Diagonal P	$7/2 \sum_{i=1}^{N} n_i^3$
	(c)	Back Substitution for Off-Diagonal P	$5/2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (n_i^2 n_j + n_i n_j^2)$
	3	Closed Loop Weighting Matrix $\overline{Q} = G^T RG + V$ For Cost = tr($\overline{Q}$P)/2	$1/2 \sum_{i=1}^{N} m_i n_i (2m_i + n_i + 1)$
CALCG	4	Back Substitution for K in $A^T_{cd_i} K_{ij} + K_{ij} A_{cd_j} + \overline{Q} = 0$	
	(a)	Diagonal Blocks of K	$7/2 \sum_{i=1}^{N} n_i^3$
	(b)	Off-Diagonal Blocks of K	$5/2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (n_i^2 n_j + n_i n_j^2)$

Table 5.2. Computation Counts for Decomposition Evaluation.

(continued on next page)

Function Group	Eqn. Ref. No.	Operation	Multiplication Count
	5	Evaluation of the Gradient	
	(Total)	$\text{Grad} = RGP_d - B^T(KP)_d$	$\sum_{i=1}^{N} n_i(m_i^2 + 2m_i n_i + nn_i)$
Gradient Search	6	For Each Call on <u>CALCF</u> or <u>CALCG</u>	$O[(\sum_{i=1}^{N}(n_i m_i))^2]$
Supremal Coordination	7	$Z = A_o P + PA_o^T$	$N * \sum_{i=1}^{N}[n_i^2 * (n-n_i)]$
Loop	8	$V = A_o^T \Lambda + \Lambda A_o + Q$	$N * \sum_{i=1}^{N}[n_i^2 * (n-n_i)]$

Table 5.2. continued

equation:

$$A_{cd_i}P_{ij} + P_{ij}A^T_{cd_j} + W_{ij} = 0, \qquad i \neq j$$

Because of the symmetry of P and W, it is only necessary to solve the upper or lower triangle of P. Thus P is found with the solution of (N(N+1))/2 reduced order Lyapunov/Sylvester equations.

The simple relationship between the RSF's of a matrix and its transpose are exploited here to further reduce computation. The reduction of A_{cd_i} (i=1,N) to RSF is performed only for the Lyapunov solution of the N diagonal blocks of P. The (N(N-1))/2 Sylvester equations make use of the RSF's computed on the diagonal, requiring computations for back substitution only [entries 2a-2c, Table 5.2].

As in the full order approach, the RSF's used in CALCG for the solution of the Lyapunov/Sylvester equations

$$A^T_{cd_i}K_{ij} + K_{ij}A_{cd_j} + \overline{Q}_{ij} = 0$$

are those computed in CALCF requiring only the back substitution for all (N(N+1))/2 equations [entries 4a & 4b].

5.2. Auxiliary Computations

In addition to the solution of the Lyapunov (and, Sylvester) equations, DPFP multiplications are performed elsewhere in CALCF and CALCG, in the gradient optimization, and, for the decomposition case, in the supremal coordination. These additional computations are outlined in the sequel.

5.2.1. CALCF and CALCG

Unlike the techniqu3s employed for the solution of the Lyapunov equations, the form of the other computations in CALCF and CALCG are identical for both full order and decomposition algorithms. Therefore, these computations are outlined only once, but referenced to the summaries of computations for both algorithms.

In CALCF the closed loop system matrix $\overline{A} = A-BG$ must be found. This involves the calculatior. of the matrix product B*G[entry 1, Table 5.1 & 5.2]. Also in CALCF, the closed loop weighting matrix $\overline{Q} = G^TRG + V$ must be found for the cost. This requires the quadratic matrix multiplication G^T*R*G[entries 2].

CALCG requires a string of matrix multiplications to produce the matrix $GRAD = RGP_d - B^T(KP)_d$ [Entry 5]. This is performed via a set of matrix multiplications:

$$RG = R*G$$

$$RGPD = RG*P_d$$

$$KPD = (K*P)_d$$

$$BTKPD = B^T*KPD$$

In the implementation of the two algorithms, the block diagonal 3tructure of the problem is exploited whenever possible. For example, the product BG of the two block diagonal matrices B and G is formed along the diagonal blocks

$$BG_i = B_i*G_i, \qquad i = 1,N$$

In order to eliminate multiplications of off-diagonal zeroes.

5.2.2. Gradient Search

Because the trajectory of optimization via gradient search is strongly dependent upon the structure of the problem, as well as upon the initial guess, it is impossible to specify a priori the computations required in the execution of a gradient search. However, for each call on CALCF or CALCG, calculations on the order of M^2 (designated $0(M^2)$), where M is the number of parameters being optimized, are performed. M is determined by the product of the dimensions of the submatrices of the input matrix B [entries 6]. The significance of these multiplications with respect to overall computational effort is discussed in Subchapter 5.3.

5.2.3. Supremal Coordination

The hierarchical structure of the decomposition introduces some computations at the supremal level, in the formation of the matrices Z and V [entries 7 & 8, Table 5.2].

Because A_o is an off-diagonal matrix (i.e., the diagonal blocks are zero), the multiplication by known zeros can be avoided for computational efficiency. This multiplication count is based on no knowledge of the density of the off-diagonal submatrices. However, the class of system for which this decomposition technique is being proposed is often characterized by sparse off-diagonal blocks. If this sparsity is utilized at the supremal level, the actual multiplications at the supremal level will be significantly less than those presented in entries 7 & 8.

5.3. Comparison of Techniques

As formulated here, the two techniques for optimization (full order vs. decomposition) differ only in the solution of the Lyapunov

equations and the coordination at the supremal level.
The relationship between these two factors and its effect on the computational efficiency is the topic of this Subchapter.

In general, the decomposition technique is attractive if the system has the structure of an aggregation of many low-ordered, weakly-coupled subsystems. Consider a system in which N subsystem, each of order n_s and with m_s inputs, are interconnected. Then $n = Nn_s$ and $m = Nm_s$. We examine the effect on computation as N increases.

Table 5.3 contains the formulae for the total number of multiplications for CALGF & CALCG for the full order and decomposition algorithms. These formulae are derivel for the structure under consideration from the general formulae presented in Table 5.1 and 5.2 (entries 1-5), with $\sigma = 1.5$. Because of the fixed subsystem dimension and input number, the computational savings are seen in the smaller power of N which appears in the decomposition formulae. As N gets very large the full order equations are dominated by the terms $\frac{23N^3}{2} n_s^3$ for CALCF and $\frac{7N^3}{2} n_s^2$ for CALCG. This becomes increasingly large in comparision with the dominant terms of the decomposition evaluation, $\frac{5N^2}{2} n_s^3$ for CALCF and $\frac{7N^2}{2} n_s^3$ for CALCG.[3]

This saving is, of course, offset by the computational cost of coordination. For each reevaluation of the supremal variables, the gradient

[3] In the decomposition algorithm, the Lyapunov computations in CALCG are kept small enough to be on the order of those for the evaluation of the quadratic term $G^T RG$.

Entry	Function Description	Multiplication Count Formula
1	Multiplies Per Full Order Evaluation of CALCF	$N(n_s^2 m_s + 1/2(m_s n_s)(2m_s + n_s + 1)) + \frac{23}{2} N^3 n_s^3$
2	Multiplies Per Full Order Evaluation of CALCG	$N(n_s m_s^2 + 2m_s n_s^2 + Nn_s^3) + \frac{7}{2} N^3 n_s^3$
3	Multiplies Per Evaluation of CALCF Via Decomposition	$N(n_s^2 m_s + 1/2(m_s n_s)(2m_s + n_s + 1)) + N(\frac{23}{2} + \frac{5}{2}(N-1))n_s^3$
4	Multiplies Per Evaluation of CALCG Via Decomposition	$N(n_s^2 m_s + 2m_s n_s^2 + Nn_s^3) + N(7/2 + 5/2(N-1))n_s^3$

$$n_i = n_s, \quad m_i = m_s; \; i=1,N \qquad n = Nn_s, \; m = Nm_s$$

$$\sigma = 1.5$$

Table 5.3. Comparison of Multiplications for Full Order Evaluation in CALCF and CALCG Vs Those for Evaluation via Decomposition in the Case of Equidimensioned Subsystems with Equal Numbers of Inputs.

optimization process must be reinitialized and re-executed. Obviously, the rate of convergence at the supremal level is a major factor in evaluating the overall computational advantage of the decomposition. Unfortunately, it is not possible to determine the convergence at the supremal level from dimensioning information, because the convergence is a function of the coupling between the various subsystems. As the coupling becomes stronger, the convergence slows or even diverges. As the convergence rate is reduced, more calls are made to CALCF and CALCG, reducing the effect of the favorable ratio of multiplication counts. Roughly, the convergence rate must be such that (for large N) the total number of calls to CALCF and CALCG are less than N times those call counts for the full-order solution of the problem before a savings is realized.

The preceding was a presentation in simplest terms of the relationship between the reduced computational cost at the infimal level and the resultant costs for coordination. The multiplications performed in the gradient search were not considered. The significance of these computations depends upon the number of inputs to the subsystems. For the postulated aggregate of equally-dimensioned subsystems, the number of multiplications performed per call to CALCF or CALCG (in both the decomposition and full order algorithms) is $0[(N n_s m_s)^2]$. In the lower limiting case, where $m_s = 1$, $0(N^2 n_s^2)$ is significant with respect to the decomposition's multiplications of CALCF and CALCG proper. This significance increases as the number of inputs to the subsystems increases. Specifically, if $m_s > \sqrt{n_s}$, the multiplications in the gradient search are greater than those of the Lyapunov equations. Thus in actuality the rate of convergence must be such that the total number of calls to CALCF and CALCG are significantly less than N times those call counts for the full-order solution.

6. IMPLEMENTATION AND EXAMPLE

Consider a demonstration system consisting of A loop of double integrators. Assume that the state of each integrator is available locally for each subsystem. Furthermore, assume coupling between the subsystems is a uniform value α. Then the open loop state space equation is given by

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \cdots & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ \alpha & 0 & & & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & & 0 \\ 0 & & & 1 \end{bmatrix} u + w \qquad (6.1)$$

where $n = 2N$ and $m = N$. The object is to find N local feedback vectors G_i in order to minimize a cost function of the form (2.2). For the series of experiments on this demonstration system, the weighting matrices $Q = I$, $R = I$, were used. The matrix Ξ is taken to be block diagonal, with the blocks alternating between $\begin{bmatrix} 1 & 2 \\ 2 & 10 \end{bmatrix}$ and $\begin{bmatrix} 5 & -6 \\ -6 & 8 \end{bmatrix}$. The block diagram for this system is given in Figure 6.1.

Implementation of the algorithm presented in (4.12)-(4.16) has the basic structure of solving the subproblems until the gradient of G is "sufficiently close" to zero (as indicated by the Euclidean norm of the gradient $< \varepsilon_I$) for a given Z and Λ. The process terminates when the new values for the matrices Z and Λ are sufficiently close to their values in the previous iteration (as indicated by $[\sum_{i=1}^{n} \sum_{j=1}^{n} (\Lambda_k - \Lambda_{k-1})^2]^{1/2} < \varepsilon_S$). This basic framework can be modified to provide increased efficiency of the algorithm. Commentary on these modifications is made where appropriate

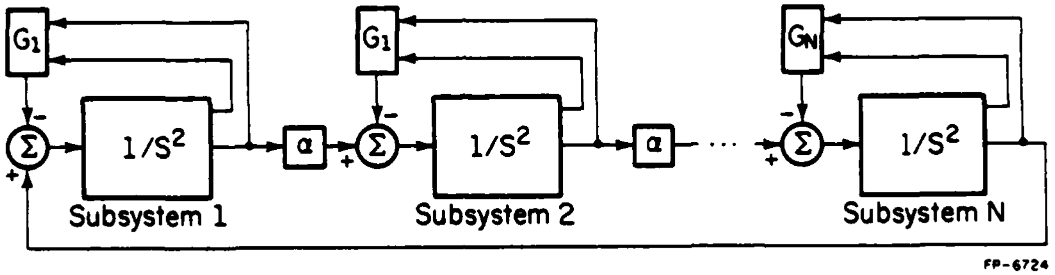


Figure 6.1. Block Diagram of Chained Double Integrators with Local State Feedback.

in the following discussion of results for the demonstration system.

6.1. Effects of Coupling

The first area of investigation is that of the effect of coupling on the total number of iterations required. Table 6.1 lists these quantities for the parameter α ranging from 0.3 to 0.9 coupling, for systems of 2,4 and 8 subsystems. Noteworthy is the closeness in the number of iterations for 2,4, and 8 subsystems for a given value of α. This strongly suggests that it is the system structure, and not its dimension, which controls the convergence at the supremal level. In another series of tests, only one of the couplings was varied in the 8th order (4 subsystem) system while the rest were held to 0.3. When the varying coupling exceeded 0.3, the iteration count followed the varying coupling. The resulting iteration counts were very close to those when all of the couplings were uniformly at the higher value, indicating that the iteration count is dominated by the strongest coupling between subsystems.

The rate of increase of iterations vs. coupling is also seen to be extremely linear. Care should be taken in attempting to generalize this characteristic, however, because of the simplicity of the coupling structure and uniformity in α for this particular example.

6.2. Calls To CALCF/CALCG Vs Iteration Number

An indication of the rate of convergence of the optimization is the increase in speed at which the infimal problem is solved for subsequent values of the infimal variables. This speed is inversely proportional to the number of calls to CALCF and CALCG per iteration. Before presentation

Coupling α	N = 2	N = 4	N = 8
0.3	13	13	13
0.5	21	19	19
0.7	27	24	25
0.9	34	29	30

$\varepsilon_s = \varepsilon_I = 10^{-6}$ Initial Guess $G_i = [1\ 1.732]$

Table 6.1. Iterations of Supremal Loop for Varying Coupling.

of the results, however, a few comments on the interpretation of these counts is in order. For the case where Ξ is block-diagonal and the P and K matrices are initialized to zero, the first iteration solves for the N submatrices G_i as though the subsystems were completely decoupled. Therefore the solution for this G via gradient search is highly inefficient. The more direct approach would be the solution of N corresponding Riccati equations. This also implies that the only variation in calls to CALCF & CALCG due to different initial guesses for the G matrix occur in the first iteration, since the second iteration always starts with the optimal G for the decoupled system. Thus for the second iteration onward, the path of optimization is the same regardless of the initial guess for G (assuming that the Hessian at the decoupled optimal G is sufficiently well-formed). The first iteration counts for CALCF and CALCG therefore say very little about the optimization via this decomposition technique. In fact, because these initial calls are used to solve a problem that is of a different nature from the rest of the optimization process, they bias the data and impede insight into the workings of the algorithm. Therefore these intial counts are disregarded in the following analysis.

When these adjustment for first iteration counts are made (Table 6.2) total CALCG is seen to vary quite linearly with CALCF, with an offset which grows with the order of the system. This linearity can be attributed to the accerelerated gradient search employed, which almost always assures a reduction of the cost for each call on CALCF. This being the case, only calls to CALCF will be considered henceforth, with the understanding that the calls to CALCG behave accordingly.

Coupling	N = 2		N = 4		N = 8	
α	F	G	F	G	F	G
0.3	42	42	55	51	73	63
0.5	64	64	85	76	109	95
0.7	87	87	100	97	122	114
0.9	113	112	127	120	143	135

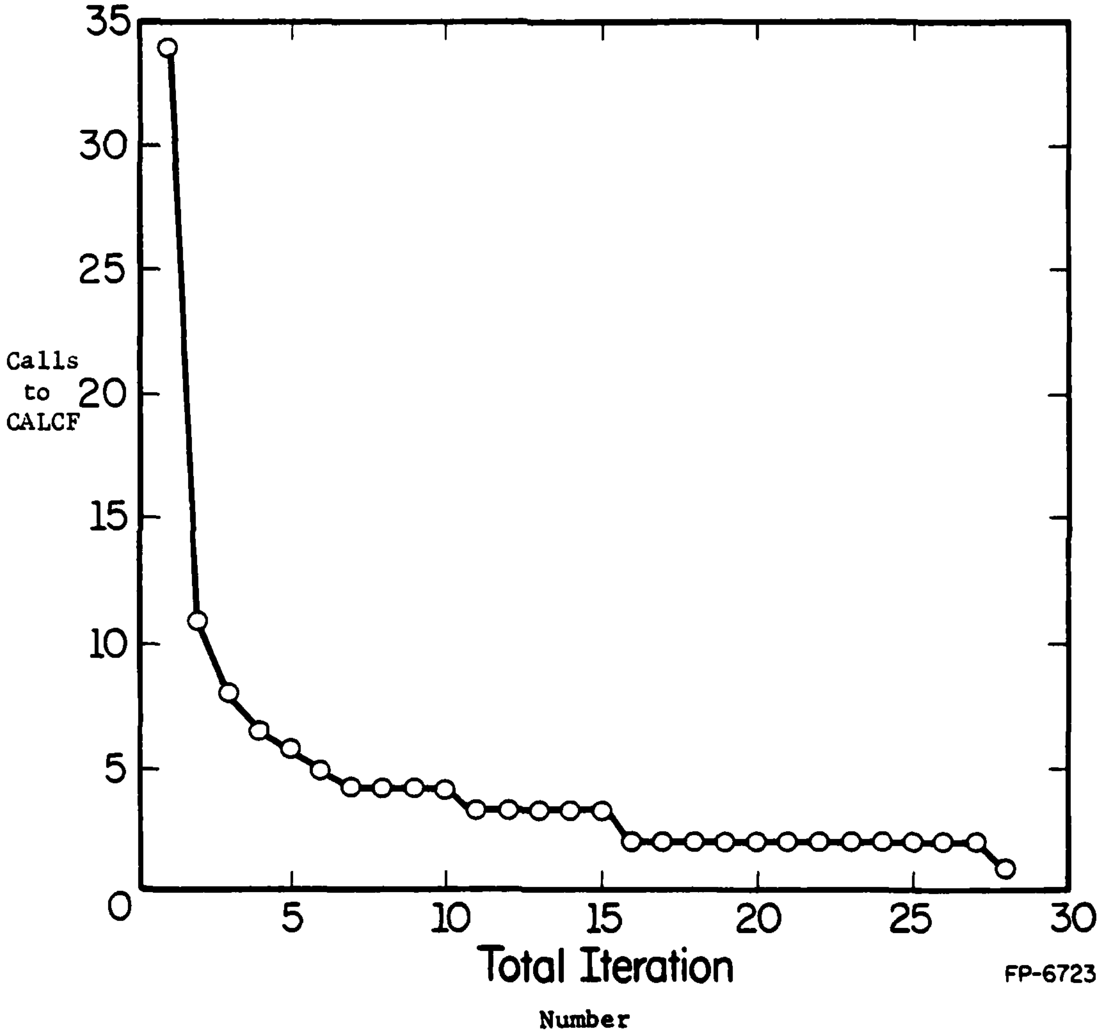
F = Total Calls to CALCF   G = Total Calls to CALCG

Table 6.2. Adjusted CALCF/CALCG Vs Coupling.

Figure 6.2 is a plot of the calls to CALCF vs iteration number for N = 4 and α = 0.9. The extremely rapid drop is attributable to both the convergence of the supremal variables and to the increasing accuracy of the approximation to the Hessian being used by the gradient search algorithm. This rapid reduction in calls to CALCF (and CALCG) is characteristic of all of the examples run. The long tail of 2 calls/ iteration is a function of the supremal convergence bound ε_S. In the particular example of Figure 6.2, the problem is essentially solved after 15 iterations, with the remaining iterations enforcing the $\varepsilon_S = 10^{-6}$ condition.

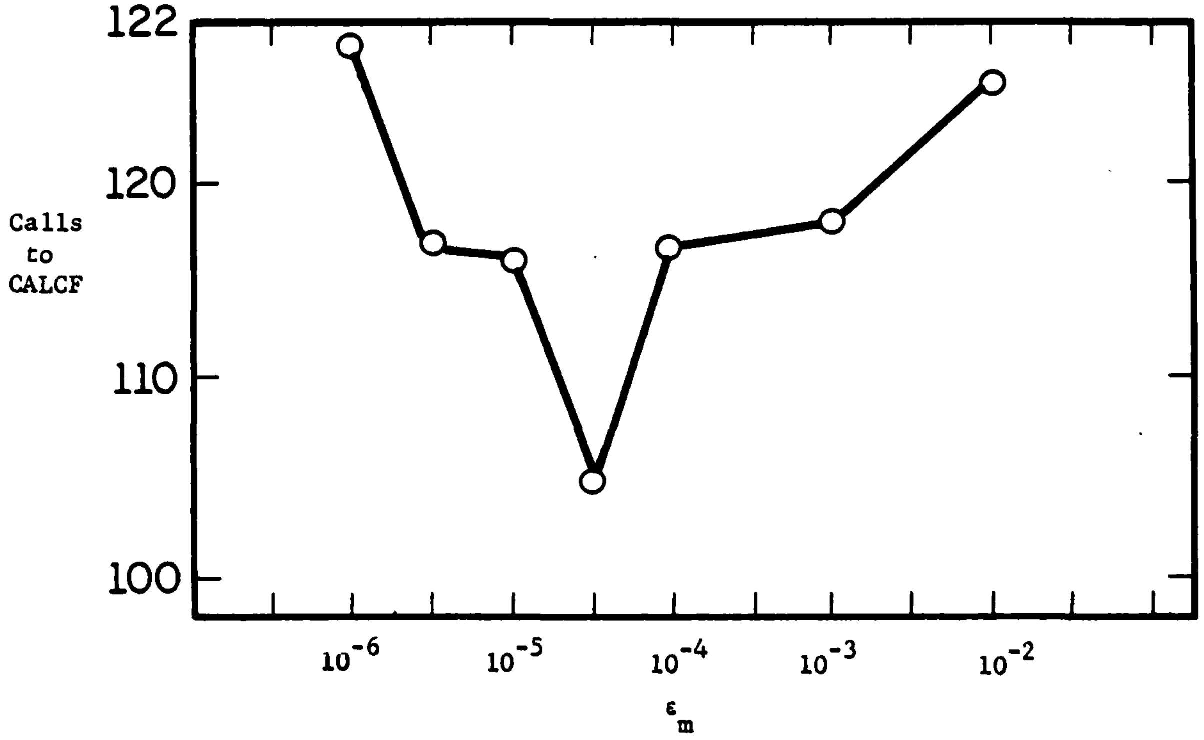
6.3. Relaxation of ε_I

During the first few iterations of the supremal, the supremal variables Z and Λ are at some distance from their final (optimal) values, and so the gain matrix G being found for these iterations is also distant from its optimal value. Under this condition, it is not necessary to force the satisfaction of the infimal stationarity condition to the final desired accuracy. Initial relaxation of ε_I can be incorporated by specifying a minimum accuracy for the infimal (min ε_m) which is strengthened to its final value ε_I as the supremal variables converge. With appropriate choice of min ε_m, this modification has resulted in a reduction of up to 25% in calls to CALCF and CALCG, with the savings increasing as the coupling is stronger. This relationship to coupling follows from the fact that the stronger the coupling, the more distance between initial and final values of the supremal variables. Figure 6.3 is a plot of total calls to CALCF as a



N = 4 $\alpha = 0.9$ $\varepsilon_s = \varepsilon_I = 10^{-6}$

Figure 6.2. Calls to CALCF Vs Iteration Number.



$N = 4$ $\alpha = 0.9$ $\varepsilon_I = \varepsilon_s = 10^{-6}$

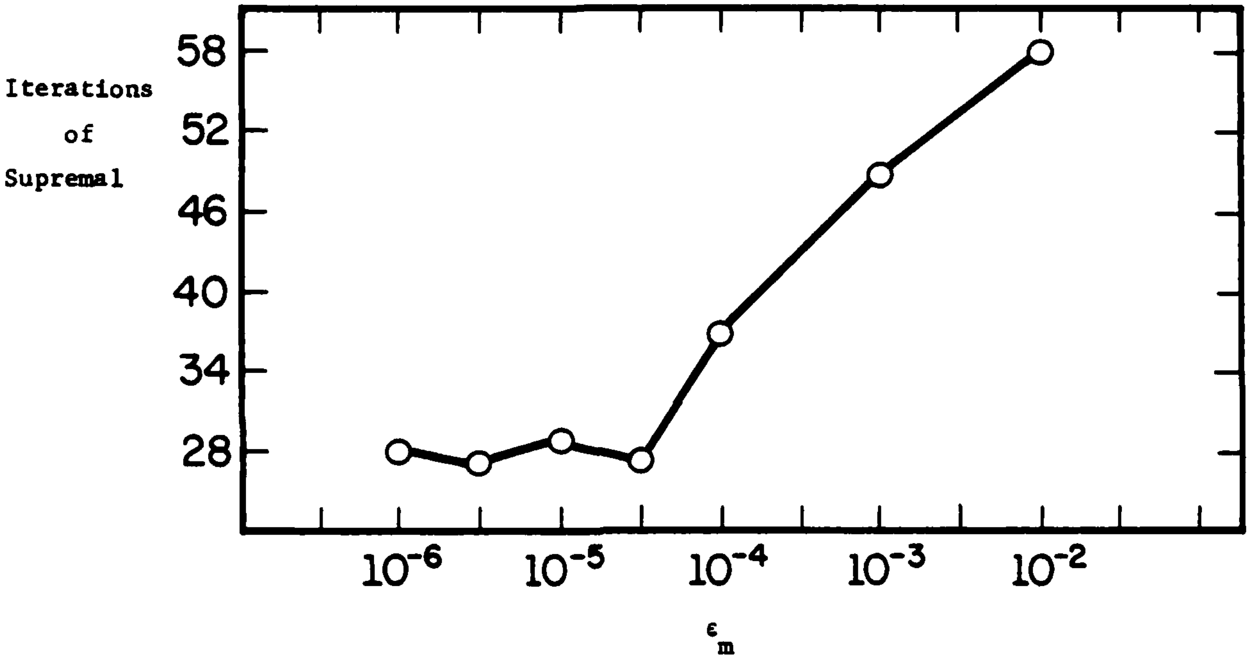
Figure 6.3. Total Calls to CALCF Vs ε_m.

function of ε_m for $N = 4$ and $\alpha = 0.9$. For this example, the minimum appears to be narrowly defined near 5×10^{-4}, although any value between 10^{-6} and 5×10^{-4} results in some saving. For $\varepsilon_m > 5 \times 10^{-4}$ the call count begins to rise again. This rise in call count is accompanied by a rise in iteration count, as seen in Figure 6.4. This combination results from over-relaxation of the infimal stationarity condition, which invalidates the convergence properties of the strictly - formulated iteration equations (3.11)-(3.21). Although the call count in this region is still less than that of the unrelaxed trajectory, the added iterations put an unnecessary burden on the supremal. This is especially undesirable if the problem is being solved in a truly distributed manner, where a poor choice of ε_m could result in undue interprocessor communication.

A comparison of the data for ε_m ranging from unrelaxed to critically relaxed (for this example, $10^{-6} \leq \varepsilon_m \leq 5 \times 10^{-4}$) shows that the reduction in calls to CALCF is evenly distributed among the first few iterations, which is the expected and desired result. This even reduction per iteration continues for the overrelaxed case, but the information being passed between infimal and supremal is inaccurate , thus requiring more iterations (and thus more total calls to CALCF and CALCG) before convergence is achieved.

6.4. Strong Coupling

The problem formulation of (2.1)-(2.2) emphasizes the coupled-subsystem nature of the class of systems being analyzed. Inherent in this classification is the notion that the subsystems are in some sense weakly coupled. If the coupling becomes too strong, the domain of convergence



$N = 4$ $\alpha = 0.9$ $\varepsilon_I = \varepsilon_s = 10^{-6}$

Figure 6.4. Total Iterations of Supremal Vs ε_m.

of the optimal P and Λ matrices becomes small enough that arbitrary initial values (i.e., $P = K = 0$) will not be attracted. In such cases either of two techniques can be applied.

The first technique is that of "walking-up" the solution. In this method the coupling is weakened sufficiently for a solution to be found. In effect, for the original problem with system matrix $A = A_d + A_o$, a new system is defined with system matrix $A_w = A_d + \beta A_o$, where $0 \leq \beta < 1$. This new problem is solved completely for the optimal P_w, K_w and G_w. The problem is then reinitialized with a stronger value for β, using P_w, K_w and G_w as starting values. This process continues until $\beta = 1$.

For the demonstratjion system with $N = 2$, the variables converge unaided for coupling up to a maximum value of $\alpha = 1.6$. By walking up the system in several steps, the solution was found for $\alpha = 2.7$, with a total accumulation of 505 calls to CALCF and 475 calls to CALCG in 152 iterations of the supremal loop.

The second technique applicable to the problem is that of relaxation ([16]). With this method (4.12)-(4.13) become

$$Z_{k+1} = t(A_oP_k + P_kA_o^T) + (1-t)Z_k \qquad (6.2)$$

$$\Lambda_{k+1} = tK_k + (1-t)\Lambda_k \qquad (6.3)$$

where t (the relaxation parameter) is initialized to 1 and reduced when incapability of convergence is detected. As convergence is reached for the relaxed Z & Λ, t is increased to a maximum value of 1.

Applying the relaxation equations (6.2)-(6.3) to the aforementioned problem, a solution was found for $\alpha = 2.7$ after 61 iterations involving 193 calls to CALCF and 183 calls to CALCG.

Comparison of these results indicated that when no prior knowledge about the domain of convergence is assumed, the relaxation method is preferable. The (eventual) success of the walking-up technique, however, indicates that some knowledge as to the neighborhood of the optimal P & K is useful in speeding up the optimization process. Table 6.3 provides an example of the advantage of starting 'near' the solution. The ability to start near the solution is a characteristic of some real-world systems. For example, the coupling between power plants in a large power grid is a function of the phase-angle between the plants. If the phase angles shift slowly enough, the controls can be recalculated on the basis of previous information, rather than from start.

6.5. Comparison of Computations for Evaluation Via Decomposition Vs Full Order Evaluation

As has been stated previously, the impetus for applying decomposition techniques to this optimization problem is the expected reduction of computations as the system becomes large (Chapter 5). Table 6.4 is a tabulation of the total counts of multiplications for $\alpha = 0.5$ systems of 2, 4 and 8 subsystems.

6.6. Comments on Experimental Results

Experimentation with the chained double integrator shows promise for the application of this solution algorithm. Of course, it is often not possible to ascertain how much of the performance is determined by the method of solution chosen and how much by the particular example chosen to demonstrate that method. The findings presented in this chapter have been chosen because they appear to represent the characteristicts of the

Counts for Solution with P,K,& G for α =	Iterations	Total Calls To CALCF	Total Calls To CALCG
0	51	180	180
1.4	39	109	104

N = 2 α = 1.5

Table 6.3. Savings in Computations for "Walking-Up" From "Near" Values of P, K, & G.

		2 Subsystems		4 Subsystems		8 Subsystems	
		Decomp	Full	Decomp	Full	Decomp	Full
1)	Total Calls to CALCF	64	19	85	27	109	25
2)	CALCF Mult Factor(1)-(3)[4]	252	754	704	5984	2208	47456
3)	(1) x (2)	16128	26330	59840	161568	240672	1186400
4)	Total Calls to CALCG	64	16	76	20	95	21
5)	CALCG Mult Factor	148	276	520	1960	1936	14928
6)	(4) x (5)	9472	4416	39520	39200	183920	313488
7)	Mults in Gradsearch	20704	7588	93040	31984	442736	116592
8)	Iterations of Supremal	21	-	19	-	19	-
9)	IT. Mult Factor (7)-(8)	64	-	768	-	7168	-
10)	(8) x (9)	1344	-	14592	-	136192	-
11)	Total=(3) + (6) + (7)+(10)	47648	38334	206992	232752	1003520	1616480
12)	<u>Decomp Total</u> Full Total	1.25		.889		.62	

α = Ø.5

Table 6.4. Comparison of Total Decomposition Counts Vs Full Order Counts for N = 2,4 & 8.

[4] Refers to the corresponding entries in Tables 5.1-5.2.

decomposition algorithm. Many other interesting relationships have not been presented because they seem to be too dependent upon the double integrator, single, uniform coupling structure.

This decomposition technique is not applicable to all decentralized control problems. As has been shown in this chapter, convergence of the algorithm is closely related to the degree of weakness of the coupling between the subsystems. When this coupling becomes sufficiently strong, the supremal variables diverge with successive iterations. In such cases the formation of the supremal variables can be relaxed. In theory ([12]), this relaxation will always result in the eventual convergence of the supremal variables. In practice, however, the relaxation may move the spectral radius, γ, of the iteration operator very near to 1. For $\gamma \approx 1$ the rate of convergence may be so low as to make this technique impractical (see [16],[13]).

One system structure which proved to be nonamenable to this decomposition technique is that of a network of electrical power generating plants. Control of such power systems has attracted much academic and industrial interest ([17]-[20]). In such a network, the subsystems are the various power plants in the network, and the coupling is introduced via the tie lines over which the plants pool power to meet shifting demand. This coupling is a function of the phase difference in the output of the plants. Minimization of the phase angle between the power plants maximizes the coupling between them. Unfortunately, the more efficient a power network is in pooling its collective output, the less likely that a hierarchical control structure can be applied to it.

An attempt was made to compute the decentralized state feedback gain matrix for the three interconnected power plant example presented in

[18]. The overall system matrix was partitioned into three subsystems. The coupling, which represented 10% of the tie line capacity with a 45° phase angle between the subsystems (pairwise) , proved too strong for supremal convergence. Relaxation of the supremal was tried, but abandoned when the supremal variables were still diverging for $t = .001$. An attempt to "walk-up" the problem by solving the problem completely for successively stronger coupling was also terminated when the coupling at only 0.23 of its final strength produced almost no perceptible convergence in successive iterations of the supremal variables.

7. SUMMARY

This report has presented the application of the interaction prediction decomposition to the hierarchical computation of optimal gains for decentralized linear systems with quadratic cost functions. For systems which have the structure of a large aggregation of many low-ordered, weakly coupled subsystems, this approach has been shown to actually reduce total computation costs when compared to centralized computation of the gains. Furthermore, the decomposition of the system structure for purposes of computation corresponds directly to the decomposition for purposes of control. This correspondence facilitates implementation of both computation and control in a truly distributed, decentralized manner.

REFERENCES

[1] M. D. Mesarovic, D. Macko, and Y. Takahaka, Theory of Hierarchical, Multilevel Systems, Wiley, New York, 1970.

[2] D. A. Wismer, Optimization Methods for Large Scale Systems with Applications, McGraw-Hill, 1971.

[3] W. Findeison, Lectures on Hierarchical Control Systems,Center for Control Sciences, University of Minnesota, October 1976.

[4] N. Smith and A. P. Sage, "An Introduction to Hierarchical Systems Theory," Comp. and Elec. Engr., Vol. 1, No. 1, 1973.

[5] L. S. Lasdon, Optimization Theory for Large Systems, McMillan, New York, 1970.

[6] H. S. Witsenhausen, "Separation of Estimation and Control for Discrete Time Systems," Proc. IEEE,59, No. 11, November 1971.

[7] Y. C. Ho and K. C. Chu, "Team Decision Theory and Information Structures in Optimal Control Problems, Part I," IEEE Trans. A.C., AC-17, No. 1, February 1972.

[8] N. R. Sandell, Jr., "Decomposition vs. Decentralization in Large-Scale Systems Theory," 1976 IEEE CDC, Clearwater, Florida, December 1976.

[9] I. Lefkowitz, "Multilevel Approach Applied to Control Systems Design," ASME J. of Basic Engr., Vol. 88, pp. 392-398, 1966.

[10] M. Athans, "The Role and Use of the Stochastic Control Linear Quadratic Gaussian Problem in Control System Design," IEEE Trans. A.C., AC-16, No. 6, December 1971.

[11] M. G. Singh, M. F. Hassan, and A. Titli, "Multilevel Feedback Control for Interconnected Dynamical Systems Using the Prediction Principle," IEEE Trans. Systems, Man and Cybernetics, SMC-6, No. 4, 1976.

[12] G. Cohen, "On an Algorithm of Decentralized Optimal Control," J. Math. Analysis and Applications, 59, No. 2, June 1977.

[13] D. P. Looze, Hierarchical Control and Decomposition of Decentralized Linear Stochastic Systems, Ph.D. Thesis, MIT, September 1978.

[14] R. H. Bartels and G. W. Stewart, "Solution of the Matrix Equation $Ax+XB=C$; Communications of the ACM, 15, No. 9, September 1972.

[15] D. L. Kleinman and P. K. Rao, "Extensions to the Bartels-Stewart Algorithm for Linear Matrix Equations," IEEE Trans. A.C., AC-23, No. 1, February 1978.

[16] J. M. Ortega, and W. C. Rheinbolt, Interactive Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[17] O. I. Elgard, Electric Energy Systems Theory: An Introduction, McGraw-Hill, New York, 1971.

[18] M. Calovic,"Dynamic State Space Models of Electric Power Systems," Departments of Electrical and Mechanical Engineering, University of Illinois, 1971.

[19] E. J. Davidson, and N. K. Tripathi, "The Optimal Decentralized Control of a Large Power System: Load and Frequency Control," IEEE Trans. A.C., AC-23, No. 2, April 1979.

[20] S. A. Arafeh, "Hierarchical Control of Power Distribution Systems," IEEE Trans. A.C., AC-23, No. 2, April 1978.