(65)

H_2, FIXED ARCHITECTURE, CONTROL DESIGN FOR LARGE SCALE SYSTEMS

by

MATHIEU MERCADAL

Ingénieur de l'Ecole Centrale des Arts et Manufactures, Paris (1985)

S.M. in Aeronautics and Astronautics MIT, Cambridge (1987)

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1990

Signature of Author ______________________________
Department of Aeronautics and Astronautics
May 4, 1990

Certified by ______________________________
Prof. Wallace E. Vander Velde
Chairman, Thesis Supervisor

Certified by ______________________________
Prof. Steven R. Hall

Certified by ______________________________
Prof. Lena Valavani

Accepted by ______________________________
Prof. Harold Y. Wachman
Chairman, Departmental Graduate Committee

H_2, FIXED ARCHITECTURE CONTROL DESIGN FOR LARGE SCALE SYSTEMS

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the
degree of Doctor of Philosophy
at the Massachusetts Institute of Technology

by

MATHIEU MERCADAL

May, 4, 1990

ABSTRACT

The H_2, fixed architecture, control problem is a classic LQG problem whose solution is constrained to be a linear time invariant compensator with a decentralized processing structure. The compensator can be made of p independent subcontrollers, each of which has a fixed order and connects selected sensors to selected actuators. The H_2, fixed architecture, control problem allows the design of simplified feedback systems needed to control large scale systems. Its solution becomes more complicated, however, as more constraints are introduced. This work derives the necessary conditions for optimality for the problem and studies their properties. It is found that the filter and control problems couple when the architecture constraints are introduced, and that the different subcontrollers must be coordinated in order to achieve global system performance. The problem requires the simultaneous solution of highly coupled matrix equations. The use of homotopy is investigated as a numerical tool, and its convergence properties studied. It is found that the general constrained problem may have multiple stabilizing solutions, and that these solutions may be local minima or saddle points for the quadratic cost. The nature of the solution is not invariant when the parameters of the system are changed. Bifurcations occur, and a solution may continuously transform into a nonstabilizing compensator. Using a modified homotopy procedure, fixed architecture compensators are derived for models of large flexible structures to help understand the properties of the constrained solutions and compare them to the corresponding unconstrained ones.

Thesis Supervisor: Wallace E. Vander Velde
Title: Professor of Aeronautics and Astronautics

Thesis Supervisor: Steven R. Hall
Title: Boeing Assistant Professor

Thesis Supervisor: Lena Valavani
Title: Associate Professor

ACKNOWLEDGMENTS

As the endeavor comes to an end and you find yourself caught in the exhilaration of graduation, you tend to forget less pleasant moments where things were not going that well. In those moments, the help, direction and support of many have been invaluable. Let these persons be acknowledged here.

My sincere gratitude goes to my advisor, Professor Vander Velde, under the supervision of which I have had the pleasure to work during my entire stay at MIT as a student. Our very regular meetings have helped make my understanding of the subject more coherent and his insistence in finding some sensible interpretation to the problem has led to a somewhat more appealing dissertation. My thanks go to the other members of my committee as well, Professor Lena Valavani and Professor Steve Hall for their encouragement and support.

A special mention goes to Doctor Dennis Bernstein of Harris Corporation. Doctor Bernstein has shown a lot of interest in my work which is closely related to his own. His enthousiasm has led me to study the problem much harder, resulting in a much better understanding of the subject.

I have been lucky enough to meet many interesting persons as a student, and I consider them close friends. I would like to extend my deepest thanks to Norman Wereley and to his wife, Jong-On Hahm. I could hardly understand Norm when I first arrived, and I blamed it on my poor understanding of the English language. I later found that other persons feel that he does mumble a little bit. Norm and I drank gallons of coffee together, (by the cup, large for him, small for myself), talked a lot of nonsense, said a couple of intelligent things too, played softball, barbecued. The list goes on. Norm has shown a lot a patience with me, even after I rearranged his office and put his desk sideways. I still think it works better that way.

Many thanks also to Dave Vos who let me play with his unicycle; Darryll Pines, my involuntary officemate, who still does not understand why French people behave the way they do; Siegfried Zerweckh, whose office I borrowed when he went on to build experimental airplanes; Jim Alexander, Karl Pendergast, Jean Louis Grevet, Thierry De Rycke. They all have been a great source of support and good times.

I cannot forget the time I have spent with the Von Flotow family, Andy, who let me get dirty in some of his numerous home improvement projects, Lucia his wife, and his three kids, Bungy (a.k.a. Andreas junior, or the third?), the elder, Fritz (short for Friedrich), and the latest addition Claudia. I have spent many good hours playing with Bungy and reading stories to him. This experience has been an invaluable practice of the English language for me, and also a source of wonderful moments. Thank you all for that.

Finally, my dearest thanks go to my family, and more particularly, to my father Georges and my mother Danièle. Their unconditional support has been the principal reason of my success. They gave me courage many times when I was feeling down, they gave me confidence in periods of doubts, they were always a phonecall away to offer their help. To them, my eternal gratitude.

This work was wholly supported by NASA Langley Research Center under NASA Grant NAG-1-126 administered by Doctor Raymond Montgomery.

TABLE OF CONTENT

TABLE OF FIGURES

INTRODUCTION

1.1 ON THE CONTROL OF LARGE SCALE SYSTEMS

1.1.1 General

Some physical systems necessitate the use of large dimensional state vectors to describe their complex dynamics. Economic and ecologic models, power and communication networks, are examples of such systems and have motivated the early developments in the study of large scale systems. Another example of such systems are structures, which must theoretically be modeled with infinite dimensional state vectors. Structural modes in a system can be usually ignored if they are at sufficiently high frequencies and are well damped. This is not the case for large, flexible, space structures, where a large number of structural modes have to be actively controlled to reduce vibration levels. The controller cannot, on the other hand, become too complex because of implementability, reliability and robustness considerations. The required level of complexity in wiring, centralized data collecting, and centralized processing may not be achieved with the existing flight qualified hardware. It may result in an unacceptable loss of performance due to time delays and the necessity to dramatically reduce the sampling frequency. The complexity of the system makes it prone to more failures. Since vibration suppression on flexible space structures is the principal motivation of this work, Linear Time Invariant (LTI) systems are considered.

The complexity of the large scale problems influences their study, the control design procedures and the implementation of the control law. The size of the model must be chosen in accordance with the numerical tools at hand, limiting right away the level of details that one can reach. The accuracy of the model deteriorates, in any case, as it tries to encompass too many effects, and a larger model may be highly unreliable. The control synthesis techniques that require a complete model are also limited in the size of problems they can reliably handle. Finally, the controller must remain simple enough in order to be implementable. The goal of all the methods developed to handle large scale problems is to simplify the model, the design method, or the structure of the compensator [San78]. The trade-off in all design methods is made between system performance, control feedback complexity, and design procedure complexity. Better performance usually requires more complex control schemes. An optimal controller may be, however, more difficult to obtain, and suboptimal designs may be preferred since they can be usually found in a simpler way.

Large scale systems can usually be seen as a large number of simpler interconnected subsystems. This is a result of the modeling process in many cases, since one can reliably identify the local dynamics and the local interactions only. Large systems are, thus, generally built from the bottom up. Finite element methods, for example, model large structures by breaking them down into smaller rigid elements, and they also require a model of the local interaction properties. The resulting assembled model has dynamics that approximate realistically the original structure. Power and communication networks are also usually modeled as the interconnection of nodes which are characterized by their own fast internal dynamics and the way they interact with neighboring nodes. The nature of the connections and of the coupling are paramount properties that shape the behavior of the overall system [Kok81, Chw82]. The next section shows how they may influence the choice of a control architecture.

1.1.2 Decentralized versus Hierarchic Control Architectures

[Kok81, Chw82] distinguish two types of large scale systems in power networks. The characterization can be extended to other large interconnected systems. Each type suggests a specific control architecture that provides simpler feedback systems with high performance [San78].

The first class of systems is constituted by weakly coupled systems. When the coupling between subsystems is weak, the internal dynamics of one subsystem are only slightly modified when the other subsystems are connected. Hence, the global dynamics of the interconnected system can be approximated by studying the internal dynamics of each isolated subsystem.

The architecture of such systems suggests the use of a decentralized control structure. Each subsystem is provided with its own independent controller that uses only local information and has only local control authority. The control structure is much simpler since one does not need to take into account the overall complexity of the system. The coupling should not change the performance of the closed loop system more than it affects the open loop system. For weakly coupled systems, a locally decentralized control architecture appears therefore to be the natural way to simplify the compensator.

The second class of systems recognized in [Kok81, Chw82] is made of weakly connected systems. The dynamic matrix of an interconnected LTI system is assembled using the subsystem dynamic matrices and the interaction matrices. The diagonal blockmatrices represent the internal dynamics of each subsystem, and the off-diagonal blocks represent the interconnection. A weakly connected system is such that the size of the off-diagonal elements are comparable to the smallest terms of the internal dynamic matrices. The overall dynamics will have two timescales. Steady disturbances and steady commands will be propagated throughout the entire system and the connections will make the subsystems act in a coherent manner, more like a group. Rapidly varying disturbances and commands introduced locally, on the other hand, will be filtered by the weak connections and will result in a fast, mostly local, incoherent response of the subsystems. Coherence and incoherence should be understood, here, as the possibility or the impossibility to determine and to control the actions of remote subsystems using local information and local control with a bandwidth comparable to that of the actions in question.

A natural control architecture appears to be, in the case of weakly connected systems, a multilevel, or hierarchic, control structure. Fast local controllers can handle the fast, incoherent dynamics of the system, while an upper level controller handles the slow coherent motion. At the lower level, controllers gather information locally and receive directives from the upper level. The information is also condensed and sent to the upper level. The upper level receives the aggregate information from the lower level and estimates the interaction between the subsystems. In return, it calculates directives to send to the lower levels. This ensures some degree of cooperation between the subcontrollers and increases the overall performance of the system.

Weak coupling and weak connections are hard to identify in most cases. Such properties are asymptotic and, if in the limit the design procedure yielding a decentralized or a hierarchic control is simplified, this will not be the case in general. One must therefore find a general design method that can generate simplified control structures in a systematic way. Optimality based methods and, especially, Linear Quadratic (LQ) methodologies, have been very successful at generating complex multiloop controllers in an integrated fashion. The computation of the controller is automated and the mathematical details of the procedure are hidden from the designer. The design is therefore performed at the system level and deals with control issues only.

1.1.3 On Linear Quadratic Control Design Methodologies

Optimal LQ control for linear systems was not originally stated as a feedback control problem. The LQ control problem consists of driving the states of a system from an arbitrary initial condition X_0 back to zero in a prescribed amount of time t_f, while minimizing the integral over time of a quadratic cost functional involving the states of the system and the inputs required for control. Such a cost index is an energy measure for the closed loop system and is therefore an H_2 norm. The resolution of the finite time problem is a differential two point boundary value problem that can be solved even if the system is linear, but time varying. The control law is an open loop scheme since the solution to the problem is the time history of the control to be applied. As t_f goes to infinity, and if the system is time invariant, the solution becomes, however, a static feedback law and is known as the Linear Quadratic Regulator (LQR), [Kwa72b]. The feedback solution has many desirable properties: the first one is that the closed loop system is guaranteed to be stable under stabilizability and detectability assumptions [Kwa72b]. The control yields a guaranteed phase margin of 60 degrees and has a gain margin extending from one half to infinity. The properties of the LQR in terms of classical feedback theory are therefore excellent, even though the method was not derived for such a purpose. The calculation of the gains requires solving a matrix Riccati equation for which very accurate and reliable software has been developed. Hence, the computation of the solution is generally not an issue.

The LQR solution is rather impractical since it requires the knowledge of all the states of the system. Extending the Kalman-Bucy filter ideas to control, the LQG methodology generalizes the LQR to the case where only a limited number of measurements are available. The problem is set in a stochastic framework. White noise enters the plant and corrupts the measurement. The cost to be minimized is quadratic, even though it has to be averaged to comply with the probabilistic approach. The main property of the problem is that it separates into an LQR problem and an optimal filtering problem: this is known as the separation principle [Kwa72b]. The filter and the control problem can be solved independently without influencing one another. The closed loop system is, again, guaranteed to be stable under detectability and stabilizability assumptions. The optimal control and filter gains are found by solving two independent Riccati equations. The procedure is, therefore, still very easy from a numerical viewpoint. The compensator has become an LTI dynamic compensator of finite order. Its dynamics require as many poles as the plant. The LQG methodology produces truly multiloop dynamic output feedback compensators and constitute a very interesting design procedure. The price paid by implementing a filter is that there is no more guarantee of gain and phase margins [Doy79].

The LQG solution possesses many asymptotic properties which can be used to obtain feedback performance stated in terms of classical control theory criteria. The LQG/LTR (Loop Transfer Recovery) methodology is based on these asymptotic properties [Ath86]. The design goals are stated in terms of sensitivity, disturbance rejection, command following and crossover frequency. The LQG is only a tool and has lost its optimality significance. The cost and the perturbations have become generic parameters that are used to obtain frequency domain properties. Classical control design techniques, such as the use of integrators to fight steady state disturbance and obtain zero tracking error, can be incorporated. Frequency shaping of the noise and the cost allows one to design, for example, notch filters, and to tailor the sensitivity properties in chosen frequency ranges [Gup80].

In summary, LQ methodologies provide a very flexible design tool that produces multiloop designs as easily as single loop designs in an integrated fashion. They guarantee closed loop stability, and frequency domain properties can be obtained by a proper selection of the quadratic cost and of the disturbance characteristics. Very fast and very reliable software has been developed to solve the problem numerically. The design can be easily iterated by changing the parameter of the optimization problem. Even though the optimization procedure looks at enhancing the nominal performance of the plant, a compensator with satisfactory robustness characteristics can be found in most cases. The current LQ designs may not, however, be applicable to the control of large scale systems since their use would produce centralized compensators of very large order.

1.1.4 Defining General Architecture Constraints

A generic way for constraining the control architecture must be defined in order to modify the LQ optimization problem. Locally decentralized control schemes for weakly coupled systems as well as multilevel schemes for weakly connected systems should obey these general architecture constraints. The choice made in this present work is to allow for a decentralized processing structure. The following will define in more details what a decentralized processing structure is, and we will try to motivate such a choice.

The decentralized processing structure consists of dividing and distributing the processing of the data and the control law to several smaller processors, or subcontrollers. The different processors are not allowed to communicate. Their control authority and the information they receive may also be limited. Each subcontroller may be connected to a smaller number of selected sensors and, similarly, it may be connected to a smaller number of selected actuators. The complexity of each subcontroller can be reduced and the order of the transfer function realized by a given processor may be fixed by the designer. Many control architectures follow these general constraints. In particular, the full order centralized compensator consists only of one full order subcontroller; the reduced order centralized compensator consists of one subcontroller with a number of poles; locally decentralized compensators defined in Section 1.1.2 are made of many subcontrollers which use local sensors and local actuators.

Decentralized processing does not mean, however, that all the fixed architecture compensators in that class will have the characteristics of being locally decentralized. Compensators with hierarchic characters can be generated as well. Consider, for example, a flexible beam with many sensors and many actuators. Assume that some subcontrollers are required to use the signals coming from closely located sensors and are connected to actuators located in the same region. These local subcontrollers will have high control authority on the local dynamics, but poor control authority on the global dynamics of the beam since they require the beam to propagate information to and from remotely located points of the structure. On the other hand, aggregate information can be obtained by merging local sensor information at different location on the beam and sending it to one subcontroller. This subcontroller can also have high control authority on the global modes of the structure if it has access to actuators spread throughout the entire beam. Again, local actuators can be aggregated so that the subcontroller can only have a limited control resolution at the local level. Such a control structure can result in the "local" subcontrollers having a higher bandwidth than the "global" subcontroller. The actual implementation of such a scheme may have a multilevel aspect: local computers will have a direct authority on the local sensors and the local actuators. They gather and merge the information to send to the global controller and, in return, obtain the aggregate inputs that can be added to the local control inputs.

The decentralized processing structure appears, therefore, to be a very general structure that can generate many different control architectures that will be simpler, and yet have high performances. The H_2, fixed architecture control problem consists of defining an LQ problem and looking for the optimal solution that belongs to the set of compensators having a required control structure. The centralized full order compensator as well as the centralized reduced order compensator can be viewed as special cases of decentralized processing. Hence, the constrained architecture that is chosen generalizes the compensator structures that has been already used. The H_2 fixed architecture control design problem is therefore a generalization of existing LQ design methodologies. This method should produce controllers of adequate complexity while retaining some of the properties of the more classic LQ designs.

1.2 RESEARCH OBJECTIVES AND CONTRIBUTION

As it has already been stated, the purpose of this work is to generalize the LQ design methodology in order to produce H_2 optimal fixed architecture compensators which will satisfy the implementability requirements for large scale systems. Constrained techniques have already appeared in the development of the LQ methodologies [Lev70, Joh70]. They have been principally focused on reducing the order of the compensator. Reduced order, frequency domain oriented, designs have been recently considered [Cal89]. More general architectures have been considered as well [Wen80, Ber87b]. Most of the work done on constrained LQ techniques has consisted of deriving the optimality conditions and using some general purpose algorithm to solve the problem. Very few general properties of the method have been found until [Hyl84] which uncovered some of the structure of the reduced order problem. Simultaneously, homotopy, or continuation, algorithms have been proposed to solve the reduced order problem [Ric87, Ric89]. The claim is that such procedures are very good at solving complex, coupled matrix equations. Convergence appears to be better than existing procedures, but little has been proved theoretically.

The contribution of this work is to extend the understanding of the reduced order problem to the fixed architecture case. A structured set of optimality conditions is derived. It clearly shows the effect of the order and architecture constraints on the solution: the separation principle does not hold anymore; the control and filter Riccati equations that appear in the unconstrained LQG problem are modified and become coupled; some of the coupling comes from the reduction of the order, as shown in [Hyl84]. This work also shows that the multiple subcontrollers need to be coordinated since the overall control system must optimize a global cost index. One must, therefore, solve simultaneously the filter and control problem, find the optimal coupling and the optimal coordination between subcontrollers.

The second contribution of this work is to develop a homotopy algorithm for solving the fixed architecture problem, to investigate its convergence properties, and to refine the procedure to make it more reliable and deal with singularities. A general understanding of the nature and the number of solutions to constrained LQG problems is gained in the process, and it is shown, in particular, that the homotopy algorithms do not have global convergence properties and must therefore allow for noncontinuous behavior of the solution at some critical points. The study shows that the optimality conditions have many solutions, some being stabilizing, some being nonstabilizing, and some being local maxima, minima or saddle points. The number of minima, saddle points and stabilizing solutions is problem dependent and is not constant when the problem parameters are changed. The problem loses the central property of having a unique stabilizing solution as soon as constraints are introduced.

Finally, a third contribution is the derivation of some realistic design examples. The examples help test the numerical procedure. They also uncover some of the properties of the constrained designs and relate them to the properties of corresponding unconstrained designs. Finally, the examples provide a partial understanding on how to select the control architecture.

1.3 THESIS OUTLINE

This document is organized in seven chapters, including the present introduction. Chapter 2 is devoted to reviewing different approaches that have been proposed for controlling large scale systems. Many approaches try to utilize the special properties of the system they try to control and result in simplified design procedures. Other approaches try to simplify complex controllers obtained through unconstrained optimization. A constrained optimization procedure will be more complicated, but it will produce better designs. The review helps put the present work into perspective.

The optimality conditions for the H_2 fixed architecture control problem are derived in Chapter 3. The analytic form of the Hessian is also derived. The Hessian is the matrix of second derivatives, and it plays a central role in the derivation of a continuous homotopy algorithm. It also allows the determination of the type of solutions that are obtained, minimum, maximum or saddle point, and it characterizes critical points whose role is preeminent in homotopy based techniques.

Chapter 4 is devoted to the study of the properties of the optimality conditions. Structured conditions are developed to show how the LQG problem is modified when order and architecture constraints are introduced. The chapter also shows that the problem is under-determined: the optimal solution defines the compensator transfer function and the cost functional is invariant when the state space realization of the controller is changed. Minimal and reduced sets of parameters and equations are studied in the chapter.

A continuous homotopy algorithm is developed in Chapter 5 and its convergence properties studied. The numerical problems that follow from the under-determination of the state space realization of the controller are resolved. The number and the nature of the solutions to the fixed architecture control problem are investigated, and the reliability of the algorithm is improved by allowing jumps in the solutions when critical points are encountered.

Chapter 6 contains a variety of design examples aimed at testing the numerical procedure and understanding some of the properties of the constrained compensators. Two large flexible structures are more particularly investigated.

Chapter 7 ends this document with some conclusions and recommendations for future work. Five appendices can be found at the end of the document. They contain technical proofs that have been omitted in the text and details of the numerical problems that are treated in the various chapters, as well as details of the solutions obtained numerically.

OVERVIEW OF DESIGN APPROACHES TO THE CONTROL OF LARGE SCALE SYSTEMS

2.1 OVERVIEW

The field of large scale systems has generated many different control approaches [San78]. The complexity of large scale systems requires that simplifications be made in many cases. These simplifications concern either the modeling of the system, the structure and complexity of the controller, or the design procedure itself. Different methods try to use different properties of the systems to generate simpler control laws. The simplifications they perform have adverse effects on the performance of the control system.

Model simplification and design simplification procedures are shown to work well when the system has specific properties, such as being weakly coupled or weakly connected [Kok81, Chw82]. These properties ensure that the simplified designs produced by these methods are near optimal solutions.

When no such asymptotic properties exist, one must use direct constrained optimization techniques. The constraints that are applied to the problem ensure that the controller that is generated meets some implementability requirements. The multiplication of the architecture constraints make the problem harder to solve, but it yields controllers which perform better.

Higher control layers have been considered to recover some of the performance lost with simpler design techniques. Multilevel, or hierarchic, control methods separate the control into many levels. Higher levels try to coordinate the local controllers to increase the global performance of the system. The structure of the controller gains in complexity, but the overall design procedure is simpler. Again, asymptotic properties are required. The following sections discuss in detail these various control design approaches found in the literature.

2.2 SIMPLIFICATION METHODS

2.2.1 Aggregation Method

Model simplification methods were first to appear and were aimed at making the study of large systems possible. Aggregation techniques, [Aok68, Chi71, Sir79], appeared in the field of economy. Economic systems consist of a large number of agents which act independently and have essentially similar dynamics. Those agents can be individuals in the economy and the dynamics describe the way they spend, invest, or save their income. As long as the dynamics are similar and there is no interaction between the agents, only one average individual is necessary to describe the behavior of the whole, and the different agents can be aggregated into one single state, thus reducing tremendously the analysis of economic equilibria. Some theoretical justifications have been brought to the original idea. The principle of aggregation has been shown to be a particular form of contraction [Ike80b]. Its goal is, in fact, to find a reduced order dynamic system which matches at all time, and for any initial conditions, the projection of the overall state vector. That is, the trajectory of the aggregate system, for initial conditions being the projection of the entire initial state vector, will be the projection of the trajectory of the complete system, when both are driven by the same inputs. The choice of the projection is the objective of the aggregation procedure. In the case of redundant states and redundant equations, as it is the case when independent agents are acting in a similar fashion, the averaging over the agents is the same as starting with the initial average and propagating it using the common dynamics of the different agents. It was shown in [Sir79] that in order to have an exact match between the projected trajectory and that of the simplified model, one has to select modes of the original system and project their eigenstructure onto the reduced subspace. The choice of a good aggregate model consists, therefore, of selecting the predominant modes of the original system. The choice in [Sir79] is based on the size of the modal residuals of the plant transfer function.

Control strategies have been tried using aggregated models for general linear time invariant systems [Sir79]. Considering the Linear Quadratic Regulator (LQR) for control purposes, suboptimal control laws can be derived using the simplified model. The idea is to solve the LQR problem for the reduced order system using an aggregated cost functional which is as close as possible to the cost functional chosen for the complete system. The implementation of the reduced order control law to the complete system will yield a stable system whose poles are the modes not retained in the aggregation and the closed loop poles of the reduced order system. The modes not retained in the aggregate model do not change since the corresponding states are not contained in the aggregate state vector which is fed back. A lower bound for the optimal cost that one would obtain by designing the optimal regulator problem for the overall system can be evaluated, yielding a measure of suboptimality [Sin78, Ike84]. LQR is a full state feedback scheme which is very unrealistic for large systems. Direct output feedback, and modified Linear Quadratic Gaussian (LQG) techniques using a simplified observer are also investigated in [Sir79]. Spillover results from feeding back the states which are not modeled but which are present in the measurement and corrupt the filter. Stability cannot always be guaranteed and the near optimality of the design cannot be estimated in that case. Global stability of the complete system can be guaranteed sometimes. The coupling via the measurements and the feedback law between the modes retained in the simplified model, and those which were not, must remain in that case within some bounds which depend on the closed loop dynamics of the aggregate system.

2.2.2 Other Forms of Model Reduction

[Hyl85] contains a thorough review and an enlightening discussion on order reduction techniques. The paper recognizes two kinds of approaches, some which are optimality based approaches and others which rely on system theoretic arguments. Optimality based methods involve the minimization of the norm in operator space of the difference between the complete and the approximate model. The norm considered in [Wil70] is the weighted covariance of the steady state output error between the outputs of the original and the reduced order system, with both systems being driven by the same white noise. The norm is therefore an H_2 norm. Necessary conditions for optimality can be obtained and solved as a parameter optimization problem. The contribution of [Hyl85], which looks at the same problem, is to uncover the structure of the solution. It shows that one must find two positive semidefinite matrices, called pseudogrammians, that satisfy modified versions of the Lyapunov equations that yield the controllability and the observability grammian for the original system [Hyl85]. The pseudogrammians are rank deficient, reflecting the fact that the order of the approximate model is smaller than that of the original system. The nullspaces of the pseudogrammians are governed by a projection operator which has to obey optimality conditions as well. The projection selects the part of the state space that is retained in the reduced order model. It takes into account simultaneously the three geometries of the problem, the eigenstructure of the system, the geometry induced by the control matrix and the geometry induced by the measurement matrix. Such an approach is very different from selecting modes.

Other norms have been used in order to evaluate the performance of the simplified model. [Glo84] uses the Hankel norm [Fra87] of the error between the outputs of the complete system and of those of the reduced order model, as both systems are subjected to the same inputs. The choice of the Hankel norm makes the model reduction problem tractable and solvable, and it also minimizes an upper bound on the infinity norm of the error between the impulse responses of the two models. The direct minimization of the infinity norm of the difference between the impulse responses makes the solution of the problem much more difficult to find since it is a constrained model matching problem for which no simple resolution method exists as yet. The optimal reduced order model is shown in [Glo84] to match the highest Hankel singular values of the original system. The infinity norm of the difference between the frequency responses is bounded from above by the sum of the smallest Hankel of the system which have not been matched.

The method of Skelton [Ske80] is guided by optimality consideration as well. The method does not however solve any optimization problem, but it selects states based on their contribution to a quadratic cost. This method sacrifices optimality for simplicity in the solution procedure. Nevertheless, the idea of cost component ranking remains very important and has some interesting applications. Cost component ranking is indeed used in [Hyl85] in order to sort the various solutions to the optimal problem. When incorporated to the numerical software, it helps the solution converge toward the global minimum [Hyl85].

A second type of approach is based on system theoretic arguments [Mor81]. The goal of the method is to eliminate subsystems which contribute little to the impulse response of the system. The method considers the difference between the weighted impulse response of the complete and the approximate systems. The error is therefore totally similar to that of the quadratic based optimality method of [Wil70]. Instead of performing the optimization, however, the method of [Mor81] considers a state space representation of the original system such that the controllability grammian is equal to the observability grammian and both are in diagonal form. Such a state space realization is called a balanced realization. The representation gives symmetric roles to the control matrix and to the output matrix of the system. The states that correspond to the largest eigenvalues of the balanced grammians are then selected to form the reduced order system. When the original system is composed of weakly coupled systems, the method produces a near extremal solution for the quadratic cost. There is no guarantee however that this near extremal point is the global minimum, or even just a local one. The method can indeed be compared to the cost component ranking approach since it tends to break up the cost into the sum of many contributions. The choice of the states in the balancing techniques does not take into account however the value of the contribution to the cost but a quantity which is similarity invariant and which, therefore, ignores scaling which is a central part of any cost functional.

Model reduction techniques work well when the system can be represented as two weakly coupled subsystems, like weakly connected systems, and when its dynamics separate into one slow and one fast part which remain lightly coupled. Any type of control design can be attempted on the reduced order model. One must, of course, keep in mind the existence of the extra dynamics and the problem of spillover that might drive the unmodeled part of the system unstable. Techniques similar to that of [Sir79] can be used in order to guarantee some level of stability. The model reduction techniques that try to minimize the coupling yield potentially larger margins of stability, allowing for better control performance. The performance of designs obtained with a simplified system remains, however, intrinsically limited since there is no mechanism to reduce the potentially negative effects of the part of the dynamics that have been ignored in the design. The procedure will be successful only if the ignored dynamics have an asymptotically small effect on the dynamics of the system in the control bandwidth or, conversely, if the control bandwidth is kept low. Model reduction may therefore be considered for analyzing the systems but may be a poor approach to designing simplified controllers.

2.3 PERTURBATION TECHNIQUES

2.3.1 Forewords

Perturbation techniques are based upon asymptotic properties of the systems to which they apply [San78]. A distinction is made between singular and nonsingular perturbations, for both types lead to very different developments. Nonsingular perturbation theory applies to composite systems constituted of weakly coupled subsystems. As the coupling vanishes, the system becomes a set of independent subsystems. As long as it remains within certain bounds, the coupling can be ignored and the control can be designed for each individual parts. Singular perturbation theory applies to systems with slow and fast timescales. As the fast dynamics become infinitely fast, the corresponding fast states can be condensed out and the resulting system is made of a slow global dynamics. If the slow time scale is infinitely slow, the fast modes can be controlled about the quasi-steady state set by the slow dynamics. Such an approach leads naturally to hierarchic control structures.

2.3.2 Nonsingular Perturbations

The nonsingular perturbations occur in the case of weak coupling [San78, Kok81, Chw82]. The overall system is composed of interconnected dynamic subsystems. Each subsystem has a proper set of sensors and actuators and a proper dynamics. The coupling is such that the dynamics of each individual subsystem is only slightly perturbed when the other systems are connected. In that case, one intuitive approach is to neglect the interaction and consider each subsystem as isolated. Local controllers can then be designed and the closed loop characteristics of each subsystem should be only slightly changed when the other subsystems are connected as long as the coupling remains asymptotically small.

The determination of the magnitude of the coupling is, of course, a difficult task, and one main area of research has been to determine bounds below which the composite system is guaranteed to be stable. A set of interconnected systems that remains stable as a whole for any value of the coupling, as long as the coupling stays within a predetermined class, is called connectively stable [Sil73, Sil76, Sil78, Sin78]. The property is intimately tied to the choice of coupling that is allowed. One would like to find control systems that maximize the class of coupling for which the system remains stable in order to give the system more robustness. To that effect, [Sil73] proposes the following design procedure: first, solve for each isolated subsystem the LQR problem with guaranteed degree of stability. The quadratic cost functional is the integral over time of the quantity $e^{2\alpha t}(X^T QX + U^T RU)$, where X is the state vector, U the input vector, Q and R are weighting matrices. This guarantees the closed loop poles of the isolated subsystems to have a real part below $-\alpha$. Second, adjust the parameter α so that the system is connectively stable. The property translates into an algebraic criterion involving the internal dynamics of the subsystems and the coupling [Sin78]. Roughly speaking, the system will be connectively stable if the local dynamics is much faster than those of the outer loops, whose bandwidths are tied to the strength of the connections. The increase in α makes the local dynamics faster and allows for larger stability margins. Such margins are computed in [Sil73, Sil76, Sil78]. The computation of the bound as well as the derivation of the connective stability criterion in [Sin78] involve the use of vector Lyapunov techniques, [Sil78]. Such techniques consist of building suitable Lyapunov functions for each independent subsystem. The resulting global Lyapunov function is the sum of the local Lyapunov functions and, whereas it is difficult to prove that the time derivative of the overall function is negative when the size of the coupling is not exactly known, an upper bound for that derivative can be found using the local functions. This bound will be guaranteed to be negative as long as some simpler inequality conditions are met by the local Lyapunov functions and which involve the local dynamics as well as some simple upper bound on the coupling. Vector Lyapunov methods provide, therefore, a simpler sufficiency test for connective stability. [Ike80a] generalizes the study of connective stability to time varying systems.

The design obtained by ignoring the coupling results in a decentralized control scheme where each subsystem is controlled by its local actuators using local state variables. It yields very good robustness characteristics, since the system remains stable for a large class of structural changes. This approach can be qualified as noncooperative since the system is broken down into subsystems which are made as independent as possible. Hence, the dynamics of the system, and especially the coupling existing between the subsystems, is not fully used by the local controllers which only have a limited knowledge of the overall structure. The subsystems do not cooperate and neither do the controllers. This implies relying on higher control gains, and it does not consider the fact that the coupling may actually be beneficial. The two beam example of [Ber87b] shown in Chapter 5 illustrates this phenomenon: the coupling is introduced between two beams via a increasingly stiffer spring. Each beam has its own controller, and the feedback implemented is the optimal decentralized controller. As the spring stiffness is increased, but remains small, the optimal cost decreases. The coupling can, therefore, have a beneficial effect (Chapter 5, Table 5.5).

A hierarchic control scheme can also be derived using the connective stability philosophy [Sin78, Sin80]. Indeed, one can try to actively reduce the size of the coupling between subsystems. Such a task must be performed at an upper level since the interaction results in a global effect. The approach of [Sin78, Sin80] is to design local LQ regulator loops for each isolated subsystem. The perturbation entering each subsystem in the form of coupling is reduced by a global controller which tries to reduce the interaction as much as possible. In the best case, the design decouples the subsystems via the global controller, and then implements local optimal regulators for each subsystem. Such a controller is of course suboptimal. Bounds on suboptimality are computed in [Sin78]. The optimal performance being considered there is however the one obtained with zero coupling. The approach presents some advantages in the simplicity of the design: finding the gains that decouple the subsystems is nothing more than an algebraic manipulation; the remaining task is to solve a number of reduced order Riccati equations for the subsystems considered as isolated, with order much smaller than that of the complete system. The robustness is improved and one does not even require the connection to be linear to ensure the connective stability. Its drawbacks are the same as with the decentralized structure. The noncooperation goes even further since some control effort is spent to fight the coupling.

The cooperation between local controllers can be improved by including some part of the coupling in the design, [Sil79, Hod86]. The idea is to make the subsystems overlap: the system state variables are partitioned into subsets which define the state vectors for the subsystems; an overlapping partition will allow for one state variable to be shared by the state vectors of two or more subsystems. The dynamics of such a variable will therefore be taken into account by many different local controllers. Based on the results of [Ike84] on system expansion and system contraction, it is shown in [Hod86] that the problem considered is similar to that of [Sil73] and the design procedure is in fact the same: for each isolated subsystem, the LQR problem with guaranteed degree of stability is solved. The bound for α is less conservative when an overlapping decomposition is used [Oth86]. [Ike80b] shows that more freedom exists to build vector Lyapunov functions with an overlapping decomposition, thus succeeding in proving stability more often than when the vector Lyapunov functions are based on a disjoint decomposition of the system.

2.3.3 Nyquist Array Method and Diagonal Dominance

The Nyquist Array Method is a frequency domain method that can be included in the category of nonsingular perturbation techniques. It can be regarded as an attempt to generalize to Multi-Input Multi-Output (MIMO) systems design techniques developed for Single-Input Single-Output (SISO) systems, and which are based on the Nyquist or the inverse Nyquist diagram [Ros74]. The Nyquist stability criterion [Daz81] for SISO system is primarily an analysis tool: given a system, the Nyquist contour will tell whether or not the closed loop system is stable. The Nyquist contour contains, however, much more information, and it allows one to understand in more details how stability can be achieved. Bode design techniques have been derived to that effect and the Nyquist contour has led to the development of synthesis tools. For a MIMO system, the Nyquist contour plots the determinant of the return difference matrix [Mac89]. The return difference matrix is the loop transfer function (plant and compensator in series) plus the identity matrix. The determinant is a complicated function that makes it impossible, in general, to understand how loops interact and influence stability. By diagonalizing the matrix at every frequency, one obtains l eigenvalues, functions of frequency, where l is either the number of inputs or outputs, depending on where the loop is broken. Each eigenvalue can be plotted in the Nyquist plane, and it is shown in [Ros74] that the number of encirclements of the critical point by the product of the eigenvalues is equal to the sum of the encirclements of that point by each of the eigenvalues. Hence, upon diagonalization, the stability conditions can be checked by studying the phase and gain properties of each eigenvalue taken as a SISO system. The Nyquist Array Method refers to the splitting of the MIMO Nyquist test into a set of simpler SISO Nyquist tests that can be obtained, for example, by diagonalization of the return difference matrix. Simpler procedure can, however be found.

Diagonalizing the matrix transfer function at every frequency is impractical and the Nyquist Array, or an approximation of it, cannot be obtained without some simplifying assumptions. The notion of diagonal dominance provides a simple measure of how close a matrix is to a diagonal operator. A matrix is row (column) diagonally dominant if the norm of each diagonal element is greater than the sum of the norms of the off-diagonal elements located on the corresponding row (column). The eigenvalues of a matrix are contained in the Gershgorin circles, [Ros74, Mac89], which are circles centered on the diagonal elements of the matrix and whose radii are the sum of the norms of the off-diagonal elements. Considering the return difference matrix, and varying the frequency, the corresponding Gershgorin circles will describe bands, and each Gershgorin band will contain the Nyquist contour of an eigenvalue of the return difference matrix. The diagonal dominance property ensures that the bands will not contain the origin. Consequently, the number of encirclements of the origin by the eigenvalues of the return difference matrix is equal to the number of encirclements of the origin by the centers of the circles, which are also the diagonal entries of the return difference matrix. A simple sufficient condition for stability can be therefore derived, which does not involve the eigenvalue decomposition, or the inversion, of the return difference matrix. The first part of the computation consists of checking for diagonal dominance of the return difference matrix. This is equivalent to checking the diagonal dominance of the loop transfer matrix, since the difference between the two matrices is the identity. The next step consists of applying the SISO stability criterion to the diagonal entries of the return difference matrix using the origin as the critical point. This is also equivalent to applying the criterion to the diagonal entries of the loop transfer matrix using -1 as the critical point. The inverse Nyquist Array criterion consists of plotting the inverse of the diagonal entries. This sometimes results in better graphical appearances for the contours, but it is exactly similar in terms of interpreting the plot [Mac89].

The first step of the design procedure presented in [Ros74] is to tailor the matrix transfer function. Starting from a physical input output matrix transfer function, one uses pre and postcompensation as well as recombination of the physical inputs and outputs to obtain some matrix transfer function as diagonally dominant as possible. The inputs to the new system are thus the inputs to the precompensator and the outputs are those of the postcompensator. Inner loops can be closed to modify the input output characteristics of the plant. The whole purpose of these operations is to minimize the sum of the norms of the off-diagonal elements of the rows (or the columns) of the matrix transfer function defined between the new inputs and the new outputs to enforce diagonal dominance. A set of feedback gains is then chosen so that the Nyquist stability criteria are satisfied. The method can handle nonlinearity since the Popov circle criterion can be extended to the MIMO case the same way the Nyquist criterion was. The control law is connectively stable, meaning that the actual values of the off-diagonal elements of the closed loop matrix transfer function are not important as long as the matrix satisfies the diagonal dominance property. The main drawback of the method is that there is no really straightforward way to achieve diagonal dominance. Computer aided tools have been developed to help obtain diagonal dominance [Mac89]. Pseudo-diagonalization, [Ros74, Mac89], consists of trying to make the plant transfer function diagonal using compensation, and it is very similar in essence to the idea of [Sin78] to use a global controller to decouple the subsystems constituting the overall system, and the same restrictions apply. Performance, disturbance rejection, control effort and compensator bandwidth are also difficult to understand, especially if a lot of pre and postfiltering has been used. The procedure generates potentially conservative design since it is based on a sufficiency test.

The procedure of [Oth86] is similar to that of [Ros74] but has relaxed dominance conditions. The property is called quasi-block diagonal dominance. A diagonally dominant matrix always satisfies the quasi-block diagonal dominance criterion but the reverse is not true. The methodology presented in [Oht86] includes the possibility to decompose the matrix transfer function into overlapping blocks. The restrictions about the noncooperation of the connectively stable decentralized control applied for the methodology of [Sil73]. But, again, benefits are to be expected by making an overlapping decomposition of the system [Sil79]. In that case, the system input vector as well as the output vector are partitioned into subsets of inputs and subsets of outputs. The reason for expecting better performance with an overlapping decomposition is similar whether the approach a frequency domain or a time domain approach: local controllers are built using more structural information.

The local LQG/LTR design methodology presented in [Ift87] uses block diagonal dominance properties even though the problem is presented in a stability robustness setting. The overall state vector is partitioned into possibly overlapping subsets to define the subsystems. For every subsystem, the coupling with the rest of the system is translated in terms of a multiplicative error which is then bounded by some upper bound function of frequency, $e(\omega)$. A standard LQG/LTR procedure is then applied to each subsystem, where $e(\omega)$ is used for the stability robustness test [Ath86]. The procedure guarantees stability of the overall system since the gains have been chosen in such a way that the outer loops cannot destabilize the local subsystems.

2.3.4 Singular Perturbation Methods: the Multi-Timescale Approach

Singular perturbation theory applies to systems which have well separated spectra [Sak84]. In that case, systems separate into a distinct slow and fast part. When the time constants of the slow and the fast system are well separated, simplifications occur. A global slow system can be built by assuming the fast dynamics to be infinitely fast and considering the corresponding dynamic equations to be algebraic relations between state variables. This produces a reduced order aggregate model that describes the slow behavior of the system. A control system can be derived based on the reduced order model. The resulting control will have a low bandwidth. Considering the fast dynamics again, a fast behavior will be observed on top of the slow dynamics. A fast part can be added to the control in order to cancel the fast dynamic effects relative to the slow behavior. [Chw76] applies the singular perturbation techniques to derive a near optimal two timescale LQR solution in the deterministic case while [Ten77] treats the same problem in the stochastic case. Such composite controllers are, of course, suboptimal. The degree of suboptimality is estimated in the deterministic case in [Chw76]. In [Ten77], it is shown that, as the perturbation tends toward zero, the suboptimal closed loop system tends asymptotically toward the optimum. The advantage of the multi-timescale techniques is that they simplify the design procedure by breaking it into two simpler steps, one for the slow part of the control, one for the fast part, and only reduced order models need to be considered in each case. The control that results from this procedure is naturally hierarchic: The slow modes are controlled with a reduced order controller and with a relatively small bandwidth. The state of the overall system is extrapolated from the reduced order model and the lower controller tries to reduce the fast errors between the desired trajectory which is the result of the extrapolation and the actual trajectory. The two timescale case can be extended to a multi-timescale case (with more than two timescales) to get more resolution, as shown in [Ozg79]. The design method can be used iteratively to design controllers operating with different bandwidths. This should improve the degree of suboptimality, as more structural information is used to derive the control system. More recent developments have considered multi-timescale LQG controllers (based on multi-timescale state estimators) and multi-timescale filters [Kha84, Kha87]. Decentralized multi-timescale compensators have also been investigated [Kha80]. The approach has also received a frequency domain treatment in [Lus85], leading to a multi-bandwidth design procedure.

Restrictions apply to the use of multi-timescale design techniques. The closed loop system must be multi-timescale with bandwidths similar to those of the open loop system. This is not, however, a very limiting restriction in the case of a large flexible structure since the amount of control one can get from the actuators is usually limited, and very high gains are not conceivable. The second problem is to evaluate how suboptimal the design is. This is highly dependent on the choice of the fast and the slow system and on the bandwidth separation. The intermediate dynamics can potentially be driven unstable and will generally result in poor overall performances. One really needs asymptotic separation of the bandwidths to implement the method successfully.

2.4 SIMPLIFIED COMPENSATOR DESIGN

2.4.1 General

The complexity and size of the problem may dictate the use of simplified compensator structures, even though the plant does not have any properties leading naturally to a simpler design. This is the case when the coupling between subsystems is not weak enough, or when the system does not have two clearly separated time scales. Nonclassical information pattern in the feedback loop is a common way of simplifying the controller structure [Chg71]. The information pattern is called nonclassical when the control law that drives a given actuator is based on a limited knowledge of the outputs of the system and a limited knowledge of the actions of the remaining actuators. In the decentralized control case, the control inputs driving a given subsystem are functions of the outputs of that subsystem only. The simplification of the controller structure results in a tremendous complication of the control design procedure. As reported in [San78, Sin78, Sin80], the optimal LQR compensator with nonclassical information pattern is nonlinear. Nonlinear feedback is not a practical solution, and a more common approach is to consider linear feedback laws with constrained information structures. Assuming the feedback to be linear does not, however, solve all the problems since the separation principle does not hold anymore when the information structure is constrained [Chg71, Hyl84]. The optimization process is therefore intrinsically more difficult.

2.4.2 Stabilization and Pole Placement

The existence of a centralized stabilizing compensator is guaranteed if the system under consideration is both detectable and stabilizable. The order of such a compensator has a lower bound as shown in [Bra70]. The compensator is, however, centralized: the information coming from all the sensors is simultaneously processed to generate the input commands for all the actuators. Stabilizability and detectability do not guarantee that there exists a compensator with the given constrained architecture that will stabilize the plant. The notion of fixed poles generalizes the notion of observability and controllability for LTI systems with fixed architecture controllers [Wan73]. For a given feedback architecture, the fixed poles are the poles that do not move when the control loop is closed. When the feedback is centralized, the fixed poles are just the uncontrollable and the unobservable poles. One method to determine the fixed poles is to close the control loops with the required architecture using direct output feedback with randomly selected gains. The fixed poles will always be left unchanged. Hence, they have probability one to be detected with such a procedure [Wan73].

A system with stable fixed poles can be stabilized by dynamic output feedback with the chosen architecture. When the orders of the subcontrollers are chosen appropriately, the poles of the closed loop system which are not open loop fixed poles can be freely assigned [Wan73]. These results permit to extend the robust servomechanism problem [Dav76a] to the robust decentralized servomechanism problem [Dav76b]. The robustness is defined as the property for the control system to remain asymptotically stable and regulate with zero steady state error in the presence of steady disturbances and steady structural error.

[Wes84] specializes the decentralized servomechanism problem to large space structures. The results were derived assuming sets of dual sensors and actuators. Under these conditions, it is shown that the decentralized robust servomechanism has a solution if and only if the centralized robust servomechanism has a solution, in other words, if and only if the rigid body modes are controllable and observable. An other very interesting result is that it is possible to design a decentralized controller for which the unmodeled higher order modes will not be destabilized.

The study of [Cor76] gives another complete set of conditions for stability and pole placement using decentralized control. The approach is to determine conditions under which a system made of interconnected subsystems can be made controllable and observable from the inputs and outputs of one particular subsystem. Loops are closed around the other subsystems in order to modify the coupling and make the entire system controllable and observable from the actuators and sensors of the selected subsystem. Once the controllability and observability conditions are met for the selected set of sensors and actuators, dynamic compensation can be used to place the closed loop poles.

All the existence theorems proving that pole placement is possible under certain conditions are very important from a theoretical point of view, but they have very little applicability when design is concerned: the performance of the closed loop system is indeed hard to translate in terms of eigenstructure specifications. The order of the design may also be quite high. Performance oriented, or optimality based techniques are better suited for design purposes.

2.4.3 Optimality Based Simplified Compensator Design Techniques

One common approach to designing compensators is to define the performance of the system in terms of a cost index and try to find the compensator minimizing that cost. Linear Quadratic techniques have yielded very powerful MIMO design tools and the solution procedure has become very efficient. The control they yield is centralized and the order of the compensator is equal to that of the plant and larger is frequency shaping of the cost is used [Gup80]. The resulting controller may be too complex if the plant itself is very complex. The next sensible step is to find the best compensator that satisfies the implementation requirements.

[Chg71] considers the design of a near optimal decentralized LQG design by replicating the centralized LQG solution: each compensator runs an unbiased estimator of the plant state vector and the controls generated by one subcontroller are linear combinations of these state estimates. The determination of the various gains involve solving coupled modified Riccati equations. The only advantage obtained with the scheme of [Chg71] is that the information pattern is somewhat simplified. The overall control must however maintain several full state estimators and cannot be realistically implemented if the order of the plant is large.

A more general approach consists of fixing the structure of the compensator as well as its order such that it represents an acceptable level of complexity, and solve the constrained LQR and LQG problems as a parameter optimization problem. Reduced order optimal H_2 compensators have appeared early in the literature, [Lev70, Joh70], following the development of the unconstrained quadratic methods. They have raised the interest of many [And71, Bas75, Men75, Ly82, Kab83, Hyl84, Ly85, Moe85, Kra88, Cal89]. Most of the work has been centered on finding reduced order compensators. In [Wen80, Ber87b], however, the information pattern is specified as well. [Wen80] contains the most general control structure, whereas [Ber87] studies locally decentralized controllers (no overlapping information allowed). First order necessary conditions for optimality can be easily derived. Solving them is a very difficult optimization problem. Few theoretical results have been found to explain the nature of the solution and the properties of the controllers one can obtained through these direct methods. Only with the more recent efforts of Hyland and Bernstein has one tried to explain the structure of the problem and shown how it is closely related to the full order LQG problem [Hyl84]. When the compensator is full order, the classical LQG problem reduces to solving two uncoupled Riccati equations of order equal to that of the plant. When the order of the compensator is smaller than that of the plant, it is shown in [Hyl84] that the solution to the optimal problem consists of solving two full order modified Riccati equations coupled by two modified Lyapunov equations via a projection operator whose rank is equal to the order of the compensator. The projection tries to determine the best subspace in the plant state space where control should be performed. The determination of the projection is an integral part of the optimization process [Hyl84]. Attempts have been made to use the structure of the equations and the projection in order to develop better algorithms for solving the constrained LQG problem [Hyl83, Ric89]. These numerical techniques do not generalized, however, to the decentralized control case. The results can also be derived for discrete time systems [Ber86c, Ber86d]. The optimal fixed order compensator problem can be stated for infinite dimensional plants as well [Ber86a]. The optimality conditions can be transformed into two modified Riccati equations and two modified Lyapunov equations, all coupled through the optimal projection. Instead of matrices, however, these equations involve infinite dimensional linear operators. Due to the infinite dimension of the state space, one needs to call upon properties of linear operators in Hilbert spaces. The proofs are consequently more involved, and this result of theoretical importance has little application since a numerical solution requires the discretization of the problem. Nevertheless, it ensures that, by taking a large but finite dimensional approximation of the plant, and by solving the optimal projection equations for this model, one will find a compensator that tends asymptotically to the optimal solution as the order of the model is increased. The projection method has also been extended to the filtering problem [Ber85]. A more detailed review of the direct quadratic optimization approaches will be made in the following chapters.

The numerical difficulties associated with the direct solution of the optimization problem have been a deterrence to many, and a simpler approach has been sought through indirect design methods. A large order compensator is a large scale system, and an approximate model can be derived for it using techniques similar to those used for simplifying the plant. Like in the model reduction case, many different approaches have been studied for reducing the compensator [Enn84, Liu86, Opd90]. The rationale behind designing a full order compensator first, and reducing it next, is that the higher modes of the plant are taken into account in the design process unlike in the case when the controller is based on a simplified model of the plant only. As the order of the compensator is increased, the optimal performance should be recovered and, by choosing a reduced order approximation as close as possible to the complete controller, one should limit the performance degradation to a minimum.

Indirect procedures are much easier to implement that the direct ones, but they lead to nonstabilizing compensators in many cases and the closed loop performance may often be unsatisfactory [Liu86]. As pointed out in [Ric87, Hyl90], designs obtained through direct methods always yield better performance and always stabilize the plant provided that there exists a stabilizing compensator in the class of compensators having the required order. The indirect methods cannot been generalized to the case of constrained information pattern.

2.5 MULTILEVEL TECHNIQUES

2.5.1 General

The multilevel, or hierarchic, architecture appears as a natural way to control complex systems made of a large number of coupled subsystems. Hierarchies seem to be the preferred way of evolution for societies. Hierarchic organizations maximize the welfare of the group by making its constituting elements cooperate. Furthermore, the seemingly complex control structure breaks the processing down in such a way that each decision maker (i.e. controller) needs not have a complete understanding of the global system in all its details but only some partial knowledge of it [Chg76]. At the subsystem level, local controllers operate using local information and information supplied by a global controller. They supply in return the global controller with partial and condensed information about the local sensor outputs as well as the local actions they are taking. The global controller has perfect structural information about the system, and knows in particular how the subsystems interact. Given the information received from the subsystems, the global controller sends directives to each local controller so that more cooperation occurs within the system. Each subcontroller, be it at the local level or at the global level, operates with partial and simplified information, limiting the complexity of the control task for each decision maker. Such an architecture is very elegant, but the design procedure must take into account the entire model in order to distribute the tasks between the subcontrollers. The constraints on the information pattern will complicate tremendously the design procedure.

2.5.2 Periodic Coordination

As argued by Chong and Athans in [Chg76], if the global controller can receive all the local information, the optimal solution will be for the global controller to cancel out the local actions and superimpose the centralized optimal solution. As a result, [Chg76] considers that the global controller operates at a smaller rate than the local controllers. Such a structure is called periodic coordination, since the directives arrive at the local level periodically every l time steps, where l is the ratio between the global controller sampling time and the local controller sampling time. Interconnected systems are considered in [Chg76] and the optimal Linear Quadratic solution with periodic coordination is studied. The control structure is as follows: local controllers drive local actuators based upon local information. The local control law would be LQ optimal if there were no coupling between the subsystems. At the upper level, the interaction between the subsystems is estimated, based on a priori information and past measurements. The update of the estimate of the interactions is done periodically every l steps. Two kinds of periodic control are developed in [Chg76]. The first one is qualified as open loop, meaning that the coordinating parameters are computed based on past information and without expecting future information. Thus, the estimate tries to minimize the mean error due to the interaction for all future times as if no more updates were able to refine the estimate. The second one is qualified as closed loop, meaning that future measurements are expected. In that case, the estimate tries to minimize the mean error due to coupling for the next l steps only, knowing that the estimate will be refined later on. The closed loop scheme is more complex to solve and its resolution does not decouple at the subsystem level. It should yield, however, a better solution. The method appears to be an elegant design method. Still, even if optimality is reached, little is known about stability.

2.5.3 Goal Coordination and Interaction Prediction Methods

The problem considered throughout [Sin80] is the time varying LQR problem for interconnected systems. The computation of the optimal control sequence requires the knowledge of the full state variables. In order to simplify the computation, [Sin80] considers each individual subsystem separately. For one particular subsystem, the interaction of the other subsystems is a sequence of vectors which are linear combinations of the states of the other subsystems. These vectors can be defined as new variables. Additional variables must also be defined in that case. They are called coordination variables and are in fact Lagrange multipliers that are introduced in order to relate the interaction variables to the states of the subsystems from where the coupling arise. In the Goal Coordination Method, [Sin75, Sin80], also referred to as the Interaction Balance Method, the optimal control sequence is solved at two levels. At the lower, or subsystem level, one computes the optimal control sequence as if the subsystems were isolated. The coordination variables are used at the local level as parameters for the local minimization problem that generates the local control sequence. At the upper level, the coordination variables are updated in order to optimize the overall cost of the interconnected system. The updating process is truly a minimization algorithm. The gradient of the cost relative to the coordination variables is computed at the subsystem level and is used in the upper level in the optimization procedure. The optimum is found recursively by first assuming a value for the coordination variables, then by computing the gradient of the cost at the lower level, solving only reduced order minimization problems. A different scheme attributed to Takahara is referred to as the Interaction Prediction Method [Sin75, Sin80]. The method uses both the aforementioned coordination variables as well as the coupling variables to define the coordination vector between the local and the global problem. The computation is carried out like in the Goal Coordination Method by assuming a value for the coordination vector at the upper level and by computing the gradient of the cost at the lower level. Convergence properties are enhanced when the coupling variables are not eliminated in favor of the coordination variables.

Both multilevel techniques require the iterative computation of a minimum at each time step. A high rate of convergence is reported using either method for fairly complicated systems. Both methods are so-called infeasible methods [Sin75, Sin80] since the coupling variables introduced in the problem are equal to the real coupling only when the solution is reached. The main drawback of such methods is that suboptimal control sequences cannot be obtained by relaxing the accuracy on the determination of the minimum at each time step. Such a control sequence could very well destabilize the plant and does not satisfy any of the problem constraints. Therefore, the expected reduction in computation time due to the breaking down of the large minimization problem into simpler reduced order problems may very well be overestimated because of the need to reach accurately a minimum at each time step. The complexity of the implementation is not addressed either. The time varying problem is solved as an open loop problem. The feedback problem can also be solved. The same procedure is used, but the control at the subsystem level is a function of the local states and the coupling variables. The gains are computed in a recursive manner, using the coordination technique. The main advantage is that they require the resolution of only reduced order Riccati equations, whose calculation grows much faster than linearly with the order. The control that comes out of the procedure is a centralized full state feedback LQR, and is therefore not suitable for implementation.

2.5.4 Hierarchic Control with Distributed Sensors and Actuators

Hardware and implementation considerations have led to the development of last class of controllers reviewed in this chapter. [Hal90, How90] have considered structures with distributed sensors and actuators. The premise that such sensors and actuators can be built has been suggested by the advances in piezoelectric materials. The deformations of a piezoelectric layer transforms the local strain into a voltage which can be measured. Similarly, a voltage applied locally will produce a force on the structure. Hence, distributed action on the structure can be obtained. Hierarchic control appears, in that case, to be the only approach that can utilize the unique possibilities offered by distributed sensing and actuating capabilities while producing a control structure of acceptable complexity. The method becomes optimal if there is a frequency gap in the spectrum of the structure and if the higher modes do not propagate and can be controlled optimally with local control only. The method is therefore related to that of [Ozg79], even if the latter is not developed directly in terms of a multilevel control. In [Hal90, How90], the upper level controller receives condensed local information from which it estimates the global motion of the system. Such a global motion is made of the slow modes of the structure. The upper level controller fights low frequency perturbations that affect the entire system in a very coherent way. The global level sends back to the local controllers the global shape of the system. The local controller will then take out high frequency perturbations that affect the global modes. High frequency perturbations have a tendency to be localized and be less coherent over the structure, which is why they can be eliminated by simpler local controllers. Fairly simple proportional plus derivative feedback on the local displacement variables is used at the lower level. The upper level also coordinates the lower level controllers and makes sure they do not excite the lower modes by eliminating the slow coherent residual effects generated by the local control laws. The control input at each point is the sum of the coherent part coming from the upper level and the local part that has been cleaned of its residual coherent part. The drawbacks of the method are the same as those pointed in section 2.3.4: good control of the slow and fast modes is achieved, but intermediate modes may be affected adversely. The method will work better if the structure has well separated slow and fast modes. Nevertheless, the method does take into account more of the physics of the problem and yields a control law that is implementable with the type of technology envisioned in [Hal90].

OPTIMALITY CONDITIONS FOR THE H_2 FIXED ARCHITECTURE CONTROL

3.1 INTRODUCTION

The control design methodology presented in this work is a generalization of the well known LQG design methodology. The LQG methodology has been extensively studied and applied to design Multi-Input Multi-Output feedback regulators. It constitutes a good design tool for the following reasons:

– the optimal solution is a linear time invariant feedback system with a rational transfer function. Its dynamics happen to have the same order as the plant. The closed loop system is asymptotically stable under detectability and stabilizability assumptions.

– the design is a truly Multi-Input Multi-Output feedback system and all sensors and actuators are included at once in the design procedure.

– the design parameters provided by the designer have physical meanings which leads to an insightful iteration of the design. These parameters include a model of the disturbance entering the plant, the definition of the outputs that must be regulated and a scaling of their respective importance in the overall performance of the system. It also includes a scaling of the amount of energy one can require from each actuator and information about the amount of noise that corrupts the measurements of each sensor. All these design parameters can be related to physical data in terms of noise intensities or energies.

– finally, the solution to the problem is unique. One must solve two Riccati equations to find the solution and there now exists very reliable algorithms for solving such equations. The LQG methodology is therefore very appealing and easy to use for designing stabilizing multivariable feedback laws.

The method suffers, however, many drawbacks and has little practical applicability to the control of large flexible structures for example. One main drawback is that it may result in feedback systems of very high orders and high complexity. The data processing capabilities of existing flight qualified computers will be rapidly exceeded. Furthermore, the wiring might be too complex and the testing impossible. Finally, such complex control systems are prone to more failures. A second drawback is that these high order designs may not be robust since nowhere in the optimization procedure is it stated that the model may be incorrect. In order to design an LQG compensator for a large structure, one can include more vibrational modes in the model in order to encompass potential spillover problems. The LQG solution must then have more modes as well, and the design of the controller will rely on modes which are increasingly more poorly modeled. The result may be a compensator finely tuned to the wrong model and which in reality misinterprets the information it receives, thus potentially driving the closed loop system unstable.

The Optimal H_2 Fixed Architecture Control Design approach is a direct attempt to resolve the problem of controller complexity while it tries to retain some of the best features of the LQG design methodology. The idea of constraining the order or the structure of the LQG solution has appeared repeatedly in the literature. The reduced order compensator problem has received most of the attention [Joh70, Men75, Kab83, Hyl84]. Some schemes for constraining the architecture have, however, also been proposed in [Wen80], and in a less general way in [Ber87b]. The fixed architecture control design problem consists of setting the control problem as the optimization of some H_2 norm of the closed loop system similar in every way to the H_2 norm considered in the unconstrained LQG problem. The difference is that the solution is required to have a given architecture which is specified in advance. Typically, the feedback system is made up of p independent processors which cannot communicate between each other. Each processor has also limited memory and can only realize a transfer function which has a fixed number of poles. Finally, each processor is connected to some selected sensors and actuators. The choice of the number of processors, number of poles and the selection of the sensors and the actuators is a trade-off between the simplicity of the feedback system and its performance. The choice of the architecture will also influence the difficulty of numerically finding a solution. The fixed architecture design is a suboptimal solution when one considers the unconstrained problem. Hence, the solution will not be as highly tuned to the problem as the overall optimal solution. This may ensure better robustness properties, even though there is not a guarantee since the robustness requirements are not included in the optimization procedure. Some attempts have been made to generalize the LQ methodologies to deal with robustness issues, [Ber86d, Ber87a, Che88], but it is out of the scope of this research whose principal focus is to understand the effects of imposing architecture constraints on the controller.

The H_2 optimization problem is stated in the first part of the chapter. A Lagrangian is defined for the problem and the architecture constraints are incorporated. Tools necessary for the derivation of the optimality conditions are presented in the following section. These are rules of calculus that apply to matrix spaces and are used to differentiate the Lagrangian of the problem.

The LQG methodology generates dynamic compensators which are strictly proper. Hence, the transfer functions always roll off at high frequencies. The method cannot handle static output feedback since it assumes that each sensor is corrupted by white noise. The direct feedback of white noise into the system would make the cost infinite. This forces the sensor outputs to be filtered. The LQR methodology, on the other hand, does not consider the measurements to be corrupted by noise and it can be seen as a static output feedback scheme, where it is assumed that all the states can be independently measured. The generalization of the method is a fixed architecture, static output feedback scheme. The H_2, Fixed Architecture, Static Output Feedback problem is derived in the chapter for the sake of completeness. The first and second order optimality conditions are given in Section 3.4.

Finally, in order to motivate the use of Optimal Fixed Architecture Control, some examples of possible control architectures are presented in Section 3.5, such as decentralized, hierarchic or fixed dynamics compensation.

3.2 PROBLEM STATEMENT

3.2.1 The H_2 Optimal Control Problem

Consider the n-dimensional linear time invariant plant with m inputs and l outputs:

$$\dot{X} = AX + B_1u_1 + B_2u_2 + \cdots + B_mu_m + w$$

$$\begin{aligned} y_1 &= C_1X + v_1 \\ y_2 &= C_2X + v_2 \\ &\vdots \\ y_l &= C_lX + v_l \end{aligned}$$

where $A \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times 1}$, $C_i \in \mathbb{R}^{1\times n}$. $w \in \mathbb{R}^n$ is a white process noise vector whose covariance is a symmetric positive semidefinite matrix $V \in \mathbb{R}^{n\times n}$. Each measurement y_i is corrupted by a white measurement noise v_i whose variance is $V_i \in \mathbb{R}$. Gathering the input signals $u_1, u_2,\cdots, u_m$ as well as the measurement signals $y_1, y_2,\cdots, y_l$ into two vectors u and y, the plant model becomes:

$$\begin{aligned} \dot{X} &= AX + Bu + v \\ y &= CX + v_c \end{aligned}$$

where:

$$B = \left[B_1\, B_2 \cdots B_m\right],\ C = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_l \end{bmatrix},\ v_c = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_l \end{bmatrix},\ E\{v_cv_c^T\} = V_c\delta(t)$$

In order to modify the closed loop characteristics of the system, one wishes to implement an LTI feedback loop. The control law has the generic form:

$$\dot{X}_c = A_c X_c + Ky$$
$$u = G X_c$$

where X_c is the compensator state vector, $A_c \in \mathbb{R}^{n_c \times n_c}$ is the compensator dynamics; $K \in \mathbb{R}^{n_c \times l}$ corresponds to the filter gains; $G \in \mathbb{R}^{m \times n_c}$ corresponds to the control gains.

The performance of the closed loop system is established by looking at a quadratic cost J that penalizes both plant states and control effort:

$$J = \lim_{t \to \infty} \frac{1}{2t} E\{ \int_0^t X(\tau)^T R X(\tau) + u(\tau)^T R_c u(\tau) d\tau \},$$

where $R \in \mathbb{R}^{n \times n}$ is symmetric positive semidefinite, $R_c \in \mathbb{R}^{m \times m}$ is symmetric positive definite, $E\{\cdot\}$ is the expectation operator. The same cost J is obtained when one considers the plant to be subjected to deterministic disturbances, by integrating the energy $X(\tau)^T R X(\tau) + u(\tau)^T R_c u(\tau)$ over an infinite period of time as the perturbations w(t) and v(t) (formerly process and measurement noise) are two vectors of impulses equal respectively to $w(t) = \sqrt{V}\delta(t)$ and $v(t) = \sqrt{V_c}\delta(t)$. $\sqrt{X}$ denotes the square root of the symmetric positive matrix X. J is therefore the square of an H_2 norm defined for the closed loop, and the cost functional is a general quadratic cost that can have several interpretations. Following the stochastic interpretation of the problem, J can equivalently be written as:

$$J = \lim_{t \to \infty} \frac{1}{2} E\{ X(t)^T R X(t) + u(t)^T R_c u(t) \}$$

One wishes to find A_c, G and K that minimize the cost J. If no other constraints are imposed on the control loop and if n_c, the number of poles in the compensator, is free, the problem is the standard LQG problem.

3.2.2 Constrained Control Architecture

LQG designs are not always satisfactory and cannot be implemented in many cases. To make the control simpler the following constraints are introduced: the processing of the control law is distributed among p smaller processors. Each processor has a limited memory or, in other words, each processor has a limited number of integrators to realize its transfer function. The order of the i^{th} processor is denoted n_i and is fixed by the designer. The information flow and the control authority are also limited. Each processor is connected to a selected set of sensors and actuators. The set $\mathcal{U}_i$ contains m_i elements which are the indices of the m_i actuators that are connected to the i^{th} processor. The number m_i and the indices in $\mathcal{U}_i$ are specified by the designer. The set $\mathcal{Y}_i$ contains l_i indices which are the indices of the l_i sensors connected to the i^{th} processor. The number l_i and the indices in $\mathcal{Y}_i$ are also specified by the designer. The overall feedback loop will therefore be described as follows:

1) Processor i is described by its own state vector X_i which is n_i dimensional.
2) The global compensator is the aggregation of the X_i:

$$X_c = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}$$

3) The matrix A_c is block diagonal:

$$A_c = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_p \end{bmatrix} \tag{3.2.1}$$

where $A_i \in \mathbb{R}^{n_i \times n_i}$ describes the internal dynamics of processor i.

4) G and K can be block partitioned in, respectively, m times p row vectors and p times l column vectors:

$$G = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1p} \\ G_{21} & G_{22} & \cdots & G_{2p} \\ \vdots & \vdots & & \vdots \\ G_{m1} & G_{m2} & \cdots & G_{mp} \end{bmatrix}, \quad G_{ij} \in \mathbb{R}^{1 \times n_j}$$

$$K = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1l} \\ K_{21} & K_{22} & \cdots & K_{2l} \\ \vdots & \vdots & & \vdots \\ K_{p1} & K_{p2} & \cdots & K_{pl} \end{bmatrix}, \quad K_{ij} \in \mathbb{R}^{n_i \times 1} \tag{3.2.2}$$

where $G_{ij} = 0_{1 \times n_j}$ if $i \notin \mathcal{U}_j$ (the i^{th} control input u_i does not use X_j) and where $K_{ij} = 0_{n_i \times 1}$ if $j \notin \mathcal{Y}_i$, (X_i is not driven by the output signal y_j).

Define the following matrices Π_i and π_i^k:

$$\Pi_i = [0_{n_i \times n_1} \cdots 0_{n_i \times n_{i-1}} I_{n_i} 0_{n_i \times n_{i+1}} \cdots 0_{n_i \times n_p}] \in \mathbb{R}^{n_i \times n_c}$$

$$\pi_i^k = [0 \cdots 0 \underset{i \text{ element}}{1} 0 \cdots 0] \in \mathbb{R}^{1 \times k} \tag{3.2.3}$$

Consider a matrix having n_c rows partitioned in p blocks of $n_1, n_2, \cdots, n_p$ rows. Premultiplying by Π_i isolates the i^{th} set of n_i rows. Considering the transposed matrix and a partitioning of its columns, postmultiplying by Π_i^T corresponds to isolating the i^{th} set of n_i columns. Premultiplying by π_i^k provides the i^{th} row of a matrix, postmultiplying by its transposed provides the i^{th} column. Using the matrices defined above, the architecture conditions can be expressed more simply as:

$$\pi_i^m G \Pi_j^T = 0_{1 \times n_j} \quad \Leftrightarrow \quad i \notin \mathcal{U}_j$$
$$\Pi_i K \pi_j^{l\,T} = 0_{n_i \times 1} \quad \Leftrightarrow \quad j \notin \mathcal{Y}_i$$

The i^{th} subcontroller has the following dynamics:

$$\dot{X}_i = A_i X_i + \sum_{j \in \mathcal{Y}_i} K_{ij} y_j$$

The k^{th} control input is a linear combination of the state vectors of the subcontrollers to which it is connected or, in other words, it is the linear combination of the X_i such that $k \in \mathcal{U}_i$:

$$u_k = \sum_{i:k\in\mathcal{U}_i} G_{ki}X_i$$

Take, for example, a plant with 3 inputs and 3 outputs and assume that the feedback loop is made of two second order processors such that:

1) measurements 1 and 2 are available to processor 1: $\mathcal{Y}_1 = \{1, 2\}$
2) measurements 2 and 3 are available to processor 2: $\mathcal{Y}_2 = \{2, 3\}$
3) actuators 1 and 2 are driven by processor 1: $\mathcal{U}_1 = \{1, 2\}$
4) actuators 2 and 3 are driven by processor 2: $\mathcal{U}_2 = \{2, 3\}$

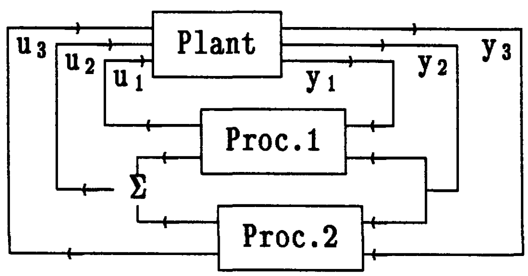

Figure 3.1: Example of Feedback Architecture

The feedback architecture is shown in figure 3.1. The corresponding K, A_c and G have the following form:

$$A_c = \begin{bmatrix} X & X & 0 & 0 \\ X & X & 0 & 0 \\ 0 & 0 & X & X \\ 0 & 0 & X & X \end{bmatrix}, \quad K = \begin{bmatrix} X & X & 0 \\ X & X & 0 \\ 0 & X & X \\ 0 & X & X \end{bmatrix}$$

$$G = \begin{bmatrix} X & X & 0 & 0 \\ X & X & X & X \\ 0 & 0 & X & X \end{bmatrix}$$

where an X denotes a free entry.

3.2.3 Lagrangian Formulation and Problem Statement

The cost functional J is a scalar quantity and it is therefore equal to its trace. Hence:

$$J = \mathrm{Tr}(\,J\,)$$

$$= \frac{1}{2}\,\mathrm{Tr}\left[\,\lim_{t\to\infty} E\{\,X^T R X + u^T R_c u\,\}\right]$$

$$= \frac{1}{2}\,\mathrm{Tr}\left[R\,E\{\,\lim_{t\to\infty} XX^T\} + R_c E\{\,\lim_{t\to\infty} uu^T\}\right]$$

Defining

$$X_{cl} = \begin{bmatrix} X \\ X_c \end{bmatrix},$$ closed loop state vector, dimension $\tilde{n} = n + n_c$

$$Q = \lim_{t\to\infty} E\{X_{cl}X_{cl}^T\},\ \in \mathbb{R}^{\tilde{n}\times\tilde{n}},$$ steady state, closed loop covariance matrix,

$$A_{cl} = \begin{bmatrix} A & BG \\ KC & A_c \end{bmatrix},\ \in \mathbb{R}^{\tilde{n}\times\tilde{n}},$$ closed loop dynamics,

$$R_{cl} = \begin{bmatrix} R & 0 \\ 0 & G^T R_c G \end{bmatrix},\ \in \mathbb{R}^{\tilde{n}\times\tilde{n}},$$ symmetric positive,

$$V_{cl} = \begin{bmatrix} V & 0 \\ 0 & KV_c K^T \end{bmatrix},\ \in \mathbb{R}^{\tilde{n}\times\tilde{n}},$$ symmetric positive.

the cost J becomes:

$$J = \frac{1}{2}\,\mathrm{Tr}QR_{cl} \tag{3.2.4}$$

where Q has to satisfy the steady state filter Lyapunov equation:

$$0_{\tilde{n}\times\tilde{n}} = A_{cl}Q + QA_{cl}^{T} + V_{cl} \tag{3.2.5}$$

Eq.(3.2.4) is a much simpler expression of the cost than the integral formulation. One must, however, include the constraint of Q satisfying Eq.(3.2.5) in the problem. The triplet (G, A_c, K) is a vector in the product space $\mathbb{R}^{m\times n_c} \times \mathbb{R}^{n_c\times n_c} \times \mathbb{R}^{n_c\times 1}$. Adding two vectors in such a space is to add the corresponding matrices, the multiplication by a scalar is to multiply each matrix. One can verify by inspection all the properties of a linear space. Thus, the cost J is a functional on a vector space. The dependence of J on the control parameters arises directly from R_{cl} and indirectly from Q. It is not possible to solve for Q in closed form by solving Eq.(3.2.5). The alternative is, therefore, to consider Q as a variable and use Lagrange multipliers. Eq.(3.2.5) appears as a set of $\tilde{n}^2$ constraints on Q, G, A_c and K. Denoting by the matrix $E = A_{cl}Q + QA_{cl}^{T} + V_{cl}$, we define $1/2\ P_{ji}$ to be the Lagrange multiplier associated with E_{ij}. They correspond to the sensitivity of the cost to variations in the intensity of the disturbance affecting the closed loop system. The Lagrangian becomes:

$$L = J + \frac{1}{2}\sum_{i=1}^{\tilde{n}}\ \sum_{j=1}^{\tilde{n}} P_{ji}E_{ij},$$

Regrouping the element P_{ij} into an $\tilde{n}\times\tilde{n}$ matrix P, and recognizing in the double summation the trace of a product, the Lagrangian becomes:

$$L = \frac{1}{2}\,\mathrm{Tr}(QR_{cl} + P(A_{cl}Q + QA_{cl}^T + V_{cl}))$$

$$= \frac{1}{2}\,\mathrm{Tr}(QR_{cl} + PV_{cl} + PQA_{cl}^T + A_{cl}QP) \tag{3.2.6}$$

We can now summarize the H_2, Fixed Architecture Optimal Control Problem:

Problem 3.1: *Given the* LTI *plant with* m *inputs and* l *outputs:*

$$\dot{X} = AX + Bu + w$$
$$y = CX + v_c$$

where w *is a white noise vector with* $E\{ww^T\} = V\delta(t)$, v_c *is white noise vector with* $E\{v_cv_c^T\} = V_c\delta(t)$, *and given the controller architecture specified by:*

p	*number of subcontrollers*
n_i	*maximum order of subcontroller* i
$\mathcal{Y}_i$	*set of indices of sensors connected to subcontroller* i
$\mathcal{U}_i$	*set of indices of actuators connected to subcontroller* i

find $G \in \mathbb{R}^{m \times n_c}$, $A_c \in \mathbb{R}^{n_c \times n_c}$, $K \in \mathbb{R}^{n_c \times l}$ *such that*

$$A_c = \mathrm{blockdiag}(A_1, A_2, \cdots, A_p), \qquad A_i \in \mathbb{R}^{n_i \times n_i}$$

$$G = [G_{ij}]_{\substack{i=1,\cdots,m\\ j=1,\cdots,p}}, \quad G_{ij} \in \mathbb{R}^{1 \times n_j}, \qquad G_{ij} = 0_{1 \times n_j} \text{ if } i \notin \mathcal{U}_j$$

$$K = [K_{ij}]_{\substack{i=1,\cdots,p\\ j=1,\cdots,l}}, \quad K_{ij} \in \mathbb{R}^{n_i \times 1}, \qquad K_{ij} = 0_{n_i \times 1} \text{ if } j \notin \mathcal{Y}_i$$

and find P, $Q \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$ *positive semidefinite, where* Q *is the closed loop covariance matrix, and* P *is the sensitivity of the cost to changes in disturbance intensities, to minimize the quadratic functional:*

$$L = \frac{1}{2}\,\mathrm{Tr}(QR_{cl} + PV_{cl} + PQA_{cl}^T + A_{cl}QP)$$

where A_{cl}, R_{cl} *and* V_{cl} *are as before. The solution is a quintuplet* (G, A_c, K, P, Q) *in the product space* $\mathbb{R}^{m \times n_c} \times \mathbb{R}^{n_c \times n_c} \times \mathbb{R}^{n_c \times l} \times \mathbb{R}^{\tilde{n} \times \tilde{n}} \times \mathbb{R}^{\tilde{n} \times \tilde{n}}$. *The compensator, whose state space representation is*

$$\begin{aligned} \dot{X}_c &= A_c X_c + Ky \\ u &= G X_c \end{aligned}$$

will minimize the quadratic cost J *of* Eq.(3.2.4) ∎

As stated in [Hyl84], the cost is a positive quantity when there exists a stabilizing compensator. The optimization problem occurs on an open set and the Lagrangian is differentiable on this open set. Hence, the minimum of the cost will be obtained for a set of parameters that make the Lagrangian stationary [Kir70]. The necessary conditions for optimality are derived in the next section.

3.3 DERIVATION OF THE NECESSARY CONDITIONS FOR OPTIMALITY

3.3.1 Calculus in Matrix Space

3.3.1.1 General on Matrix Spaces

For any integers r and s, the space of r by s matrices $\mathbb{R}^{r \times s}$ is a vector space. Indeed, one can add two matrices, multiply them by a scalar and verify that all the properties of a linear space are satisfied. The matrices $E^{r,s}_{k_{ij}}$ form a canonical basis of $\mathbb{R}^{r \times s}$, where $E^{r,s}_{k_{ij}}$ are defined by:

$$E^{r,s}_{k_{ij}} = \begin{array}{cl} \begin{bmatrix} 0 \cdots 0 & 0 & 0 \cdots 0 \\ \vdots & \vdots & \vdots \\ 0 \cdots 0 & 0 & 0 \cdots 0 \\ 0 \cdots 0 & 1 & 0 \cdots 0 \\ 0 \cdots 0 & 0 & 0 \cdots 0 \\ \vdots & \vdots & \vdots \\ 0 \cdots 0 & 0 & 0 \cdots 0 \end{bmatrix} & \begin{array}{l} \\ \\ \\ \text{i row,} \\ \\ \\ \\ \end{array} \\ \text{j column} & \end{array} \qquad (3.3.1)$$

where $\qquad k_{ij} = r\,(i-1) + j, \quad 1 \leq i \leq r, 1 \leq j \leq s,$

The matrices $E^{r,s}_{k_{ij}}$ have a single index, $k_{ij} \in \mathbb{N}$, which is related to the location where the matrix has its nonzero element on the i^{th} row and j^{th} column. The relation that yields k_{ij} as a function of the pair (i, j) consists simply of counting the elements of the matrix row by row, and it transforms the doubly indexed sequence (i, j) into the simple sequence k_{ij}. For any $M \in \mathbb{R}^{r\times s}$, the following linear combination holds:

$$M = \sum_{k_{ij}=1}^{r\cdot s} M_{ij}\, E^{r,s}_{k_{ij}}$$

where M_{ij} is the element of M located on the i^{th} row and j^{th} column. We will denote by $E_{r,s} = \{E^{r,s}_1, E^{r,s}_2, \cdots, E^{r,s}_{r\cdot s}\}$ the canonical basis on $\mathbb{R}^{r\times s}$. If the elements of the matrix M are arranged in a r×s column vector m using a single index, then M is uniquely represented by the vector m which is the vector of components of M on the basis E. More generally, any vector is uniquely represented by the vector of its components on a basis which needs not be canonical. This leads to the following definition:

Definition 3.1: *Let* S *be a subspace of* $\mathbb{R}^{r\times s}$, *generated by a family of linearly independent matrices* $E_S = \{ E_1, E_2, \cdots, E_n\}$, $E_j \in \mathbb{R}^{r\times s}$, $S = \text{span}\{E_1, E_2, \cdots, E_n\}$. *Then, for any* $M \in S$, *there exists a unique n-dimensional vector*

$$m = [m_1, m_2, \cdots, m_n]^T$$

such that:
$$M = \sum_{j=1}^{n} m_j E_j$$

We define the law $*$ *to relate any matrix* M *to the vector of its components* m *on the basis* E_S *in which* M *lies as follows, and write:*

$$m = M * E_S$$
$$M = E_S * m$$

■

When $\mathbf{S}$ corresponds to the entire space and when one uses the canonical basis E defined before, the operation $M{*}E$ corresponds to taking each row of the matrix M, transposing it and stacking it into a column vector.

The quintuplet (G, A_c, K, P, Q) is a vector in the product space $\mathbf{S} = \mathbb{R}^{m \times n_c} \times \mathbb{R}^{n_c \times n_c} \times \mathbb{R}^{n_c \times 1} \times \mathbb{R}^{\tilde{n} \times \tilde{n}} \times \mathbb{R}^{\tilde{n} \times \tilde{n}}$ and constitutes the variable of the optimization problem. The control architecture defines subspaces on $\mathbb{R}^{m \times n_c}$, $\mathbb{R}^{n_c \times n_c}$ and $\mathbb{R}^{n_c \times 1}$. Indeed, specifying the architecture consists of requiring that certain entries of G, A_c and K be zero: clearly, if G_1 and G_2 in $\mathbb{R}^{m \times n_c}$ have common zero entries, so will $\lambda_1 G_1 + \lambda_2 G_2$. G lies therefore in a subspace S_G of $\mathbb{R}^{m \times n_c}$, and using similar arguments, A_c and K lie respectively in S_{A_c} and S_K, subspaces of $\mathbb{R}^{n_c \times n_c}$ and $\mathbb{R}^{n_c \times 1}$. One can define three bases,

$$
\begin{aligned}
E_G &= \{E^g{}_1, E^g{}_2, \cdots, E^g{}_{n_g}\},\ E^g{}_i \in \mathbb{R}^{m \times n_c}, \\
E_{A_c} &= \{E^a{}_1, E^a{}_2, \cdots, E^a{}_{n_a}\},\ E^a{}_i \in \mathbb{R}^{n_c \times n_c}, \\
E_K &= \{E^k{}_1, E^k{}_2, \cdots, E^k{}_{n_k}\},\ E^k{}_i \in \mathbb{R}^{n_c \times 1},
\end{aligned}
$$

for the three subspaces S_G, S_{A_c} and S_K. G, A_c and K will then be uniquely represented by three column vectors

$$
\begin{aligned}
g &= G{*}E_G \\
a_c &= A_c{*}E_{A_c} \\
k &= K{*}E_K
\end{aligned}
$$

The most obvious basis vectors to consider for spanning S_G, S_{A_c} and S_K are canonical basis vectors of $\mathbb{R}^{m \times n_c}$, $\mathbb{R}^{n_c \times n_c}$ and $\mathbb{R}^{n_c \times 1}$ respectively. Because the architecture only imposes zero entries in the different matrices, E_G, E_{A_c} and E_K can be formed by retaining the canonical matrices that have a 1 at a location i,j corresponding to a free entry in G, A_c or K:

$$E^{m,n_c}_{k_{ij}} \in E_G \iff G_{ij} \text{ is free}$$

$$E^{n_c,n_c}_{k_{ij}} \in E_{A_c} \iff A_{c_{ij}} \text{ is free}$$

$$E^{n_c,l}_{k_{ij}} \in E_K \iff K_{ij} \text{ is free}$$

g, a_c and k are then built by stacking up in a column vector the free entries of the matrices G, A_c and K. The reverse operation consists of placing the free entries of G, A_c and K which are stored in a more compact form in g, a_c and k at their correct locations and completing the matrices with zeros. Take, for example, the control architecture of Section 3.2.2. E_G is made of the eight following matrices:

$$E^g_1 = \begin{bmatrix}1\,0\,0\,0\\0\,0\,0\,0\\0\,0\,0\,0\end{bmatrix}, E^g_2 = \begin{bmatrix}0\,1\,0\,0\\0\,0\,0\,0\\0\,0\,0\,0\end{bmatrix}, E^g_3 = \begin{bmatrix}0\,0\,0\,0\\1\,0\,0\,0\\0\,0\,0\,0\end{bmatrix}, E^g_4 = \begin{bmatrix}0\,0\,0\,0\\0\,1\,0\,0\\0\,0\,0\,0\end{bmatrix}$$

$$E^g_5 = \begin{bmatrix}0\,0\,0\,0\\0\,0\,1\,0\\0\,0\,0\,0\end{bmatrix}, E^g_6 = \begin{bmatrix}0\,0\,0\,0\\0\,0\,0\,1\\0\,0\,0\,0\end{bmatrix}, E^g_7 = \begin{bmatrix}0\,0\,0\,0\\0\,0\,0\,0\\0\,0\,1\,0\end{bmatrix}, E^g_8 = \begin{bmatrix}0\,0\,0\,0\\0\,0\,0\,0\\0\,0\,0\,1\end{bmatrix}$$

and:

$$G = \begin{bmatrix}G_{11} & G_{12} & 0 & 0\\ G_{21} & G_{22} & G_{23} & G_{24}\\ 0 & 0 & G_{33} & G_{34}\end{bmatrix} = E_G * g,$$

$$g = [G_{11}\ G_{12}\ G_{21}\ G_{22}\ G_{23}\ G_{24}\ G_{33}\ G_{34}]$$

3.3.1.2 Representation of the Differential of a Function in Matrix Space

The differential of a function F mapping a vector space V into a vector space W at a point $x_0 \in V$ is a linear operator from V to W. Consider that V and W are two matrix spaces, equal to $\mathbb{R}^{m\times n}$, and $\mathbb{R}^{l\times p}$ respectively. The function F from V to W maps an m×n matrix into an l×p matrix and, for every $M \in \mathbb{R}^{m\times n}$, F(M) is a matrix in $\mathbb{R}^{l\times p}$. F can therefore be split into l times p functionals F_{ij} from $R^{m\times n}$ to $\mathbb{R}$:

$$F(M) = \begin{bmatrix} F_{11}(M) & F_{12}(M) & \cdots & F_{1p}(M)\\ F_{21}(M) & F_{22}(M) & \cdots & F_{2p}(M)\\ \vdots & \vdots & & \vdots\\ F_{l1}(M) & F_{l2}(M) & \cdots & F_{lp}(M)\end{bmatrix}$$

The differential of F_{ij} taken at a point M_0, if it exists, is a linear form that maps $\mathbb{R}^{m \times n}$ into the real line $\mathbb{R}$. Hence, the differential of F_{ij} at M_0 is given by m times n coefficients that uniquely define a linear operator from $\mathbb{R}^{m \times n}$ to $\mathbb{R}$. These m×n coefficients can be regrouped into a m×n matrix, $F_{ij_M}(M_0) \in \mathbb{R}^{m \times n}$, that uniquely represents the differential of F_{ij} at M_0 [Ath68]. The matrix $F_{ij_M}(M_0)$ is, so far, only a convenient way to represent the differential of F_{ij}, but it can also be used to calculate the first order variation of F_{ij} about $F_{ij}(M_0)$ when the Trace operator is introduced. Perturbing M_0 by εM_1, one gets

$$F_{ij}(M_0 + \varepsilon M_1) = F_{ij}(M_0) + \varepsilon \mathrm{Tr} F_{ij_M}(M_0)^T M_1 + O(\varepsilon^2)$$

The Trace operator appears naturally in this context, since the bilinear operator

$$<M,N> = \mathrm{Tr}(M^T N) \tag{3.3.1}$$

defines an inner product on the matrix space $\mathbb{R}^{m \times n}$ [Ath68]. The inner product confers a Hilbert space structure to $\mathbb{R}^{m \times n}$ and, for any linear form f from $\mathbb{R}^{m \times n}$ to $\mathbb{R}$, including the differential of F_{ij}, there exists a matrix $F \in \mathbb{R}^{m \times n}$ that uniquely represents f which can be written as:

$$\begin{aligned} f(M) &= <F,M> \\ &= \mathrm{Tr} F^T M \end{aligned}$$

Extending the notation, one can define the differential of the entire matrix operator F at M_0 in the form of a matrix:

Definition 3.2: *Let* $F \in \mathbb{R}^{l \times p}$*, be a matrix whose elements* F_{ij} *are differentiable functionals on a matrix space* $\mathbb{R}^{m \times n}$*. Denoting by* $F_{ij_M} \in \mathbb{R}^{m \times n}$ *the matrix representing the differential of* F_{ij} *with respect to* M *at* M_0*, the matrix* F_M,

$\mathbb{R}^{(l\times m)\times(p\times n)}$, *defined in block form as:*

$$F_M = \begin{bmatrix} F_{11_M} & F_{12_M} & \cdots & F_{1p_M} \\ F_{21_M} & F_{22_M} & \cdots & F_{2p_M} \\ \vdots & \vdots & & \vdots \\ F_{l1_M} & F_{l2_M} & \cdots & F_{lp_M} \end{bmatrix},$$

uniquely defines the differential of F *at* M_0. *The operator* $M_1 \to F_M \cdot M_1$, $M_1 \in \mathbb{R}^{m\times n}$, *defined as:*

$$F_M \cdot M_1 = \begin{bmatrix} \mathrm{Tr} F_{11_M}{}^T M_1 & \mathrm{Tr} F_{12_M}{}^T M_1 & \cdots & \mathrm{Tr} F_{1p_M}{}^T M_1 \\ \mathrm{Tr} F_{21_M}{}^T M_1 & \mathrm{Tr} F_{22_M}{}^T M_1 & \cdots & \mathrm{Tr} F_{2p_M}{}^T M_1 \\ \vdots & \vdots & & \vdots \\ \mathrm{Tr} F_{l1_M}{}^T M_1 & \mathrm{Tr} F_{l2_M}{}^T M_1 & \cdots & \mathrm{Tr} F_{lp_M}{}^T M_1 \end{bmatrix}$$

is a linear map from $\mathbb{R}^{m\times n}$ *to* $\mathbb{R}^{l\times p}$. *To first order in* ε, *the value of* F *is:*

$$F(M_0+\varepsilon M_1) = F(M_0) + \varepsilon F_M \cdot M_1 + O(\varepsilon^2)$$ ∎

Definition 3.2 gives a representation of the differential of a matrix with respect to a matrix as well as a means to evaluate the differential for any perturbation. The differentiation rules, and especially the chain rule can be simply written using the notation: consider G: $\mathbb{R}^{m\times n} \to \mathbb{R}^{l\times p}$, and F: $\mathbb{R}^{l\times p} \to \mathbb{R}^{r\times s}$, are two differentiable matrix functions. Then H: $\mathbb{R}^{m\times n} \to \mathbb{R}^{r\times s}$, defined as,

$$H(M) = F(G(M)) \in \mathbb{R}^{r\times s},\ M \in \mathbb{R}^{m\times n},\ G(M) \in \mathbb{R}^{l\times p}$$

is differentiable with respect to M and, for any $M_1 \in \mathbb{R}^{m\times n}$, $H_M \cdot M_1$ is given by:

$$H_M \cdot M_1 = F_G \cdot (G_M \cdot M_1),\ G_M \cdot M_1 \in \mathbb{R}^{l\times p},$$

This defines the composed operator:

$$H_M = F_G \cdot G_M$$

3.3.1.3 Differentiation of a Quadratic Functional

Consider the following quadratic functional f on $\mathbb{R}^{l\times p}$ defined by

$$f(\mathbf{M}) = \mathrm{Tr}(\mathbf{M}^T\mathbf{R}\mathbf{M} + \mathbf{S}^T\mathbf{M} + \mathbf{M}^T\mathbf{S} + \mathbf{T})$$

$M, S \in \mathbb{R}^{l\times p}$, $R \in \mathbb{R}^{l\times l}$, $T \in \mathbb{R}^{p\times p}$. Subtracting $f(M_0)$ from $f(M_0+\varepsilon M_1)$, one gets:

$$\begin{aligned} f(\mathbf{M}_0+\varepsilon\mathbf{M}_1) - f(\mathbf{M}_0) &= \mathrm{Tr}\Big[(\mathbf{M}_0+\varepsilon\mathbf{M}_1)^T\mathbf{R}(\mathbf{M}_0+\varepsilon\mathbf{M}_1) + \mathbf{S}^T(\mathbf{M}_0+\varepsilon\mathbf{M}_1) + \\ &\quad + (\mathbf{M}_0+\varepsilon\mathbf{M}_1)^T\mathbf{S} + \mathbf{T})\Big] - \mathrm{Tr}(\mathbf{M}_0^T\mathbf{R}\mathbf{M}_0 + \mathbf{S}^T\mathbf{M}_0 + \mathbf{M}_0^T\mathbf{S} + \mathbf{T}) \\ &= 2\varepsilon\mathrm{Tr}(\mathbf{R}\mathbf{M}_0 + \mathbf{S})^T\mathbf{M}_1 + \varepsilon^2\mathrm{Tr}(\mathbf{M}_1^T\mathbf{R}\mathbf{M}_1) \end{aligned}$$

Thus: $\quad f_{\mathbf{M}}(\mathbf{M}_0) = 2(\mathbf{R}\mathbf{M}_0 + \mathbf{S})$

3.3.2 Variation of the Lagrangian with Respect to P, Q, A_c, G and K

The form of X_{cl} induces the following partitioning of P, Q and defining M = PQ:

$$P = \begin{bmatrix} P_{00} & P_{0c} \\ P_{c0} & P_{cc} \end{bmatrix}, \; Q = \begin{bmatrix} Q_{00} & Q_{0c} \\ Q_{c0} & Q_{cc} \end{bmatrix}, \; M = \begin{bmatrix} M_{00} & M_{0c} \\ M_{c0} & M_{cc} \end{bmatrix} = PQ \qquad (3.3.2)$$

where P_{00}, Q_{00}, $M_{00} \in \mathbb{R}^{n\times n}$, P_{0c}, Q_{0c}, $M_{0c} \in \mathbb{R}^{n\times n_c}$, P_{c0}, Q_{c0}, $M_{c0} \in \mathbb{R}^{n_c\times n}$, P_{cc}, Q_{cc}, $M_{cc} \in \mathbb{R}^{n_c\times n_c}$. Furthermore,

$$\begin{aligned} M_{00} &= P_{00}Q_{00} + P_{0c}Q_{c0} \\ M_{0c} &= P_{00}Q_{0c} + P_{0c}Q_{cc} \\ M_{c0} &= P_{c0}Q_{00} + P_{cc}Q_{c0} \\ M_{cc} &= P_{cc}Q_{cc} + P_{c0}Q_{0c} \end{aligned}$$

one can expand the Lagrangian in three different ways:

$$L = \frac{1}{2}\,\mathrm{Tr}P(A_{cl}Q + QA_{cl}^T + V_{cl}) + \frac{1}{2}\mathrm{Tr}QR_{cl} \tag{3.3.3}$$

$$L = \frac{1}{2}\mathrm{Tr}Q(A_{cl}^TP + PA_{cl} + R_{cl}) + \frac{1}{2}\mathrm{Tr}PV_{cl} \tag{3.3.4}$$

$$\begin{aligned} L = & \frac{1}{2}\mathrm{Tr}(G^TR_cGQ_{cc} + 2G^TB^TM_{0c}) + \\ & + \frac{1}{2}\mathrm{Tr}(P_{cc}KV_cK^T + 2M_{c0}C^TK^T) + \\ & + \mathrm{Tr}A_c^TM_{cc} + \frac{1}{2}\mathrm{Tr}RQ_{00} + \frac{1}{2}\mathrm{Tr}P_{00}V + \mathrm{Tr}AM_{00} \end{aligned} \tag{3.3.5}$$

The variation of the Lagrangian with respect to P, Q, G, A_c, and K can now be obtained straightforwardly using one of the expressions for L given above and the differentiation rules of a quadratic functional given in Section 3.3.1.3. When the derivatives with respect to one of the matrices is taken, the remaining ones are considered fixed parameters. V_{cl} and R_{cl} do not depend on P and Q, and furthermore they are symmetric. L_P is directly obtained from Eq.(3.3.3). L_Q follows from Eq.(3.3.4). Eq.(3.3.5) splits the Lagrangian into a sum of different parts, each of which depends only on A_c, G or K. The derivation of L_{A_c}, L_G and L_K is then obvious. The algebra yields:

$$L_P = \frac{1}{2}(A_{cl}Q + QA_{cl}^T + V_{cl}) \tag{3.3.6}$$

$$L_Q = \frac{1}{2}(A_{cl}^TP + PA_{cl} + R_{cl}) \tag{3.3.7}$$

$$L_{A_c} = M_{cc} \tag{3.3.8}$$

$$L_G = R_cGQ_{cc} + B^TM_{0c} \tag{3.3.9}$$

$$L_K = P_{cc}KV_c + M_{c0}C^T \tag{3.3.10}$$

If the parameters G, A_c, K , P and Q are modified by the quantities εG_1, εA_{c1}, εK_1, εP_1 and εQ_1, the Lagrangian is, to first order:

$$L(G+\varepsilon G_1, A_c+\varepsilon A_{c1}, K+\varepsilon K_1, P+\varepsilon P_1, Q+\varepsilon Q_1) =$$
$$L(G,A_c,K,P,Q) + \varepsilon \mathrm{Tr} L_{A_c}^T A_{c1} + \varepsilon \mathrm{Tr} L_G^T G_1 +$$
$$+ \varepsilon \mathrm{Tr} L_K^T K_1 + \varepsilon \mathrm{Tr} L_P^T P_1 + \varepsilon \mathrm{Tr} L_Q^T Q_1 + O(\varepsilon^2) \tag{3.3.11}$$

L_G is an $m \times n_c$ matrix that can be block partitioned like G, Eq.(3.2.2)

$$L_G = \begin{bmatrix} L_{G_{11}} & L_{G_{12}} & \cdots & L_{G_{1p}} \\ L_{G_{21}} & L_{G_{22}} & \cdots & L_{G_{2p}} \\ \vdots & \vdots & & \vdots \\ L_{G_{m1}} & L_{G_{m2}} & \cdots & L_{G_{mp}} \end{bmatrix}, \; L_{G_{ij}} \in \mathbb{R}^{1 \times n_j} \tag{3.3.12}$$

Similarly, L_K is an $n_c \times l$ matrix that can be block partitioned like K, Eq.(3.2.2)

$$L_K = \begin{bmatrix} L_{K_{11}} & L_{K_{12}} & \cdots & L_{K_{1l}} \\ L_{K_{21}} & L_{K_{22}} & \cdots & L_{K_{2l}} \\ \vdots & \vdots & & \vdots \\ L_{K_{p1}} & L_{K_{p2}} & \cdots & L_{K_{pl}} \end{bmatrix}, \; L_{K_{ij}} \in \mathbb{R}^{n_i \times 1} \tag{3.3.13}$$

3.3.3 First Order Necessary Conditions for Optimality

The independent variables of the problem are the entries P_{ij} of P, the entries Q_{ij} of Q as well as the the entries g_i of g, a_i of a_c and k_i of k. The partial derivatives of P, Q, G, A_c and K with respect to those variables are respectively:

$$\begin{aligned} P_{P_{ij}} &= E^{\tilde{n},\tilde{n}}_{k_{ij}}, \\ Q_{Q_{ij}} &= E^{\tilde{n},\tilde{n}}_{k_{ij}}, \\ G_{g_i} &= E^{g}{}_i, \quad E^{g}{}_i \in \mathcal{E}_G, \\ A_{c_{a_i}} &= E^{a}{}_i, \quad E^{a}{}_i \in \mathcal{E}_{A_c} \\ K_{k_i} &= E^{k}{}_i, \quad E^{k}{}_i \in \mathcal{E}_K \end{aligned}$$

Because the problem is stated on an open set [Hyl84], the first order necessary conditions for optimality require that the Lagrangian be stationary for all admissible perturbations [Kir70]. Hence, the derivative of the Lagrangian with respect to all free variables is zero:

$$
\begin{aligned}
L_{P_{ij}} &= L_P \cdot P_{ij} &= 0 \\
L_{Q_{ij}} &= L_Q \cdot Q_{ij} &= 0 \\
L_{g_i} &= L_G \cdot G_{g_i} &= 0 \\
L_{a_i} &= L_{A_c} \cdot A_{c\,a_i} &= 0 \\
L_{k_i} &= L_K \cdot K_{k_i} &= 0
\end{aligned}
$$

The stationarity conditions become:

$$0 = \mathrm{Tr}(L_P{}^T E^{\tilde{n},\tilde{n}}_{k\,ij}), \quad i, j = 1, \cdots, \tilde{n} \tag{3.3.14}$$

$$0 = \mathrm{Tr}(L_Q{}^T E^{\tilde{n},\tilde{n}}_{k\,ij}), \quad i, j = 1, \cdots, \tilde{n} \tag{3.3.15}$$

$$0 = \mathrm{Tr}(L_{A_c}{}^T E^{a}{}_i), \quad i = 1, \cdots, n_a \tag{3.3.16}$$

$$0 = \mathrm{Tr}(L_G{}^T E^{g}{}_i), \quad i = 1, \cdots, n_g \tag{3.3.17}$$

$$0 = \mathrm{Tr}(L_K{}^T E^{k}{}_i), \quad i = 1, \cdots, n_k \tag{3.3.18}$$

Eq.(3.3.14) just states that all entries in L_P must be zero, and similarly, Eq.(3.3.15) states that L_Q must be zero. Eqs.(3.3.15-17) imply that the entries of L_G, L_{A_c} and L_K corresponding to free entries of G, A_c and K must be zero. A_c is a block diagonal matrix. Eq.(3.3.16) thus states that the diagonal blocks of L_{A_c} must be zero. Using the block partitioned forms of L_G and L_K, Eqs.(3.3.17-18) become:

$$
\begin{aligned}
0_{1\times n_j} &= L_{G_{ij}} \\
&= \pi^m_i \, L_G \, \Pi_j{}^T, \text{ for } i \in \mathcal{U}_j \\
0_{n_i\times 1} &= L_{K_{ij}} \\
&= \Pi_i \, L_K \, \pi^l_j{}^T, \text{ for } j \in \mathcal{Y}_i
\end{aligned}
$$

Using the developed forms of L_P, L_Q, L_{A_c}, L_G and L_K, Eqs.(3.3.6-10), the optimality conditions become:

Proposition 3.1: *The matrices* P, Q, G, Ac *and* K *form a stationary solution of Problem* 3.1 *if the following conditions hold:*

$$0_{\tilde{n}\times\tilde{n}} = \mathbf{A}_{cl}^T\mathbf{P} + \mathbf{P}\mathbf{A}_{cl} + \mathbf{R}_{cl} \tag{3.3.19}$$

$$0_{\tilde{n}\times\tilde{n}} = \mathbf{A}_{cl}\mathbf{Q} + \mathbf{Q}\mathbf{A}_{cl}^T + \mathbf{V}_{cl} \tag{3.3.20}$$

$$0_{n_i\times n_i} = \mathbf{M}_{ii} = \Pi_i\mathbf{M}_{cc}\Pi_i^T, \ 1\leq i \leq p \tag{3.3.21}$$

$$0_{1\times n_j} = \pi_i^m\mathbf{R}_c G\mathbf{Q}_{cc}\Pi_j^T + B_i^T\mathbf{M}_{0j}, \ i \in \mathcal{U}_j \tag{3.3.22}$$

$$0_{n_i\times 1} = \Pi_i P_{cc} K V_c \pi_j^{l\,T} + \mathbf{M}_{i0}C_j^T, \ j \in \mathcal{Y}_i \tag{3.3.23}$$

where P_{00}, Q_{00}, M_{00}, P_{0c}, Q_{0c}, M_{0c}, P_{c0}, Q_{c0}, M_{c0}, P_{cc}, Q_{cc} *and* M_{cc} *are defined in Eq.*(3.3.2), *and where,*

$$P_{cc} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1p} \\ P_{21} & P_{22} & \cdots & P_{2p} \\ \vdots & \vdots & & \vdots \\ P_{p1} & P_{p2} & \cdots & P_{pp} \end{bmatrix}, \quad P_{c0} = \begin{bmatrix} P_{10} \\ P_{20} \\ \vdots \\ P_{p0} \end{bmatrix}$$

$$P_{0c} = \begin{bmatrix} P_{01} & P_{02} & \cdots & P_{0p} \end{bmatrix}$$

$$\mathbf{Q}_{cc} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} & \cdots & \mathbf{Q}_{1p} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} & \cdots & \mathbf{Q}_{2p} \\ \vdots & \vdots & & \vdots \\ \mathbf{Q}_{p1} & \mathbf{Q}_{p2} & \cdots & \mathbf{Q}_{pp} \end{bmatrix}, \quad \mathbf{Q}_{c0} = \begin{bmatrix} \mathbf{Q}_{10} \\ \mathbf{Q}_{20} \\ \vdots \\ \mathbf{Q}_{p0} \end{bmatrix}$$

$$\mathbf{Q}_{0c} = \begin{bmatrix} \mathbf{Q}_{01} & \mathbf{Q}_{02} & \cdots & \mathbf{Q}_{0p} \end{bmatrix}$$

P_{ij}, $Q_{ij} \in \mathbb{R}^{n_i\times n_j}$, P_{0i}, $Q_{0i} \in \mathbb{R}^{n\times n_i}$, P_{i0}, $Q_{i0} \in \mathbb{R}^{n_i\times n}$. M_{ij} *is given by:*

$$\mathbf{M}_{ij} = \sum_{k=0}^{p} P_{ik}\mathbf{Q}_{kj}, \qquad \text{for } i = 0,\cdots, p, \ j = 0, \ \cdots, \ p$$

■

When the matrices P and Q satisfy Eqs.(3.3.19,20), they become, respectively, the observability and the controllability grammian of the closed loop system, where the inputs to the closed loop system are the process noise and the measurement noise and where the outputs are the controlled variables and the control inputs. The matrix M becomes the Hankel matrix of the closed loop system [Glo84, Fra87]. The eigenvalues of the matrix indicates the transmission properties between the inputs and the outputs of the system. Hence, Eqs.(3.3.21-23) try to reduce, in some sense, the eigenvalues of M, either directly, Eq.(3.3.21), or indirectly by selecting the proper dynamics for the controller and the proper interaction between the controller and the plant. The architecture constraints reduce the freedom that one has to shape M.

3.3.4 Second Order Necessary Conditions for Optimality

The second order necessary conditions state that the matrix of second derivatives of the Lagrangian with respect to the free variables of the problem must form a positive matrix. This matrix, also known as the Hessian, is always a symmetric matrix. One can compute it by differentiating the first derivatives of Eqs.(3.3.6-10). The matrices P and Q can be seen as intermediate variables. If the closed loop dynamics are strictly stable, then the two Lyapunov equations of Eqs.(3.3.19,20) have unique solutions that yield P and Q as functions of G, A_c and K. In order to obtain a Hessian of smaller size we will consider that the free variables of the problem are the vectors g, a_c and k that are grouped in a vector ξ:

$$\xi = \begin{bmatrix} a_c \\ g \\ k \end{bmatrix}, \tag{3.3.24}$$

the Hessian matrix is:

$$L_{\xi\xi} = [\, L_{\xi_i \xi_j} \,]_{\substack{1 \le i \le n_\xi \\ 1 \le j \le n_\xi}}, \quad n_\xi = n_{a_c} + n_g + n_k,$$

or, in partitioned form, using the partitioning of ξ:

$$L_{\xi\xi} = \begin{bmatrix} L_{a_c a_c} & L_{a_c g} & L_{a_c k} \\ L_{g a_c} & L_{gg} & L_{gk} \\ L_{k a_c} & L_{kg} & L_{kk} \end{bmatrix} \tag{3.3.25}$$

Because P and Q have been eliminated, one must use the chain rule in the differentiation of L_g, L_{a_c} and L_k in order to account for the fact that P and Q satisfy Eqs.(3.3.19,20). L_{a_i}, the derivative of L with respect to a_i, is given Eq.(3.3.15) by:

$$L_{a_i} = \mathrm{Tr}(L_{A_c}^T E^{a}{}_i)$$

L_{A_c}, given by Eq.(3.3.6) is a matrix function of P and Q. Hence, the variation of L_{A_c} with respect to ξ_j is:

$$L_{A_c \xi_j} = L_{A_c P} \cdot P_{\xi_j} + L_{A_c Q} \cdot Q_{\xi_j},$$

where the product composition rule is defined in definition 3.2. Hence, from Eq.(3.3.15):

$$L_{a_i \xi_j} = \mathrm{Tr} E^{a}{}_i^T (L_{A_c P} \cdot P_{\xi_j} + L_{A_c Q} \cdot Q_{\xi_j}) \tag{3.3.26}$$

Similarly, denoting by H the matrices G or K, by L_H, L_G or L_K, and by h the vectors g or k, the variation of L_H with respect to ξ_j is:

$$L_{H\xi_j} = L_{H\xi_j} + L_{HP} \cdot P_{\xi_j} + L_{HQ} \cdot Q_{\xi_j}$$

Eqs.(3.3.17,18) yield:

$$L_{h_i \xi_j} = \mathrm{tr} E^{h}{}_i^T (L_{H\xi_j} + L_{HP} \cdot P_{\xi_j} + L_{HQ} \cdot Q_{\xi_j}) \tag{3.3.27}$$

The matrices P_{ξ_j} and Q_{ξ_j} are found by differentiating Eqs.(3.3.19,20) with respect to ξ_j. P_{ξ_j} and Q_{ξ_j} satisfy:

$$0_{\tilde{n} \times \tilde{n}} = A_{cl}^T P_{\xi_j} + P_{\xi_j} A_{cl} + A_{cl}{}_{\xi_j}^T P + P A_{cl}{}_{\xi_j} + R_{cl}{}_{\xi_j}$$

$$0_{\tilde{n} \times \tilde{n}} = A_{cl} Q_{\xi_j} + Q_{\xi_j} A_{cl}^T + A_{cl}{}_{\xi_j} Q + Q A_{cl}{}_{\xi_j}^T + V_{cl}{}_{\xi_j}$$

where:

$$A_{cl\xi_j} = \begin{cases} \begin{bmatrix} 0_{n\times n} & 0_{n\times n_c} \\ 0_{n_c\times n} & E^{a}{}_i \end{bmatrix} & \xi_j = a_i \\ \begin{bmatrix} 0_{n\times n} & BE^{g}{}_i \\ 0_{n_c\times n} & 0_{n_c\times n_c} \end{bmatrix} & \xi_j = g_i \\ \begin{bmatrix} 0_{n\times n} & 0_{n\times n_c} \\ E^{k}{}_i C & 0_{n_c\times n_c} \end{bmatrix} & \xi_j = k_i \end{cases} \tag{3.3.28}$$

$$R_{cl\xi_j} = \begin{bmatrix} 0_{n\times n} & 0_{n\times n_c} \\ 0_{n_c\times n} & G^T R_c E^{g}{}_i + E^{g}{}_i{}^T R_c G \end{bmatrix},\ \xi_j = g_i,\ 0_{\tilde{n}\times\tilde{n}} \text{ otherwise,} \tag{3.3.29}$$

$$V_{cl\xi_j} = \begin{bmatrix} 0_{n\times n} & 0_{n\times n_c} \\ 0_{n_c\times n} & KV_c E^{k}{}_i{}^T + E^{k}{}_i V_c K^T \end{bmatrix},\ \xi_j = k_i,\ 0_{\tilde{n}\times\tilde{n}} \text{ otherwise.} \tag{3.3.30}$$

Defining:

$$M_{\xi_j} = P_{\xi_j} Q + P Q_{\xi_j}$$

and partitioning P_{ξ_j}, Q_{ξ_j} and M_{ξ_j}, according to Eqs.(3.3.2) as:

$$P_{\xi_j} = \begin{bmatrix} P^1_{00} & P^1_{0c} \\ P^1_{c0} & P^1_{cc} \end{bmatrix}, Q_{\xi_j} = \begin{bmatrix} Q^1_{00} & Q^1_{0c} \\ Q^1_{c0} & Q^1_{cc} \end{bmatrix}, M_{\xi_j} = \begin{bmatrix} M^1_{00} & M^1_{0c} \\ M^1_{c0} & M^1_{cc} \end{bmatrix}$$

$L_{A_c\xi_j}$, $L_{G\xi_j}$ and $L_{K\xi_j}$ are:

$$L_{A_c\xi_j} = M^1_{cc} \tag{3.3.31}$$

$$L_{G\xi_j} = R_c E^{g}{}_i Q_{cc}\delta(g_i,\xi_j) + R_c G Q^1_{cc} + B^T M^1_{0c} \tag{3.3.32}$$

$$L_{K\xi_j} = P_{cc} E^{k}{}_i V_c \delta(k_i,\xi_j) + P^1_{cc} K V_c + M^1_{c0} C^T \tag{3.3.33}$$

where $\delta(h_i,\xi_j) = 1$ if $h_i = \xi_j$, 0 otherwise.

Proposition 3.2: *A solution to Problem 3.1 makes the cost stationary. It corresponds to a local minimum is the Hessian* $L_{\xi\xi}$ *is positive. The condition is sufficient if the Hessian is not singular.* ■

Proof

This result is a standard theorem that ensures that the cost can only increase when the stationary point is submitted to small perturbations, thus making such a point a minimum locally [Kir70].

3.4 THE FIXED ARCHITECTURE STATIC OUTPUT FEEDBACK PROBLEM

3.4.1 Problem Statement

The compensators that have been looked into so far are dynamic compensators whose transfer functions roll off at high frequency. Static Output Feedback on the other hand yields an all pass transfer function. Consider the n-dimensional LTI plant with m inputs and l noiseless outputs:

$$\dot{X} = AX + Bu + w$$
$$y = CX$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$, $w \in \mathbb{R}^n$ white noise, covariance V. We look for a feedback law

$$u = -Fy,\ F \in \mathbb{R}^{m\times l}$$

that minimizes the following quadratic cost:

$$J = \lim_{t\to\infty} \frac{1}{2} E\{X^T R X + u^T R_c u\}$$

Because the control u is directly proportional to the output y, one cannot allow for white noise to corrupt the measurement: this would make the variance of the control infinite along with the value of the cost J. Colored noise however can be accommodated by incorporating the noise dynamics into the plant: the output of the augmented system becomes a noiseless signal as required. As in Section 3.2.3, the cost can be written as:

$$J = \frac{1}{2} \mathrm{Tr}\; Q R_{cl} \tag{3.4.1}$$

where

$$Q = \lim_{t\to\infty} E\{XX^T\}$$

is the closed loop steady state covariance matrix that satisfies the filter Lyapunov equation:

$$0_{n\times n} = A_{cl} Q + Q A_{cl}^T + V_{cl} \tag{3.4.2}$$

and where

$$A_{cl} = A - BFC$$
$$V_{cl} = V$$
$$R_{cl} = R + C^T F^T R_c F C.$$

As in Section 3.2.3, one can define a Lagrangian for the problem in order to incorporate Eq.(3.4.2) and account for the dependence of Q on the feedback law. The Lagrangian is:

$$L = \frac{1}{2} \mathrm{Tr}(Q R_{cl} + P V_{cl} + A_{cl} Q P + P Q A_{cl}^T) \tag{3.4.3}$$

It is possible to restrict the authority of the sensors and the actuators by insisting that only certain sensor-actuator pairings be retained in the feedback loop. Define $\mathcal{G}$ as the set of all pairs (i,j) such that sensor i is connected to actuator j. F_{ij} will be nonzero if and only if (i,j) belongs to $\mathcal{G}$. As in Section 3.2.3, one can define a basis for the subspace S_F of $\mathbb{R}^{m\times l}$ in which F must lie. S_F is spanned by the basis matrices $E^f{}_i$ such that:

$$E^f{}_i = E^{m,l}_{k\,ij} \quad \Leftrightarrow \quad F_{ij} \text{ is free}$$

Denoting by $\mathcal{E}_F = \{E^f{}_1, E^f{}_2, \cdots, E^f{}_{n_f}\}$, F can be written in the form:

$$F = \mathcal{E}_F * f$$

where $$f = F * \mathcal{E}_F = [f_1, f_2, \cdots, f_{n_f}]^T$$

3.4.2 Necessary Conditions for Optimality

The stationarity conditions are obtained using the differentiation rules of Section 3.3.1. The variation of the Lagrangian is:

$$L_P = \frac{1}{2}(A_{cl}Q + QA_{cl}{}^T + V) \tag{3.4.4}$$

$$L_Q = \frac{1}{2}(A_{cl}{}^T P + PA_{cl} + R_{cl}) \tag{3.4.5}$$

$$L_F = R_c F C Q C^T - B^T P Q C^T \tag{3.4.6}$$

The first order optimality conditions become:

$$0_{n\times n} = A_{cl}Q + QA_{cl}{}^T + V \tag{3.4.7}$$

$$0_{n\times n} = A_{cl}{}^T P + PA_{cl} + R_{cl} \tag{3.4.8}$$

$$0 = \mathrm{Tr} E^f{}_i{}^T (R_c F C Q C^T - B^T P Q C^T),\ i = 1, \cdots, n_f$$

$$= \pi^m_i (R_c F C Q C^T - B^T P Q C^T) \pi^l_j{}^T,\ (\,i,\ j) \in \mathcal{G} \tag{3.4.9}$$

P and Q can be eliminated from Eqs.(3.4.7,8). The steps for deriving the Hessian L_{ff} are similar to Eqs.(3.3.24-33). In this case, P_{f_i} and Q_{f_i} satisfy respectively:

$$0_{n\times n} = A_{cl}Q_{f_i} + Q_{f_i}A_{cl}{}^T + A_{cl\,f_i}Q + QA_{cl\,f_i}{}^T$$

$$0_{n\times n} = A_{cl}{}^T P_{f_i} + P_{f_i}A_{cl} + A_{cl\,f_i}{}^T P + PA_{cl\,f_i} + R_{cl\,f_i}$$

where

$$A_{cl_{f_i}} = -BE^{f}{}_iC$$
$$R_{cl_{f_i}} = C^TF^TR_cE^{f}{}_iC + C^TE^{f}{}_i{}^TR_cFC$$

L_{Ff_i} is given by:

$$L_{Ff_i} = R_cE^{f}{}_iCQC^T + R_cFCQ_{f_i}C^T - B^T(P_{f_i}Q + PQ_{f_i})C^T$$

and finally:

$$L_{f_jf_i} = TrE^{f}{}_j{}^TL_{Ff_i}$$

The second order necessary conditions for optimality require that L_{ff} be positive.

3.5 EXAMPLES OF CONTROL ARCHITECTURE

This section is aimed at motivating the use of a fixed architecture control structure. Indeed, the optimality conditions are rendered more complex by the introduction of constraints in the feedback loop and one can wonder what are the benefits one can expect from such compensators. The following subsections present some potentially useful control structures.

3.5.1 Fixed Order Controller

This constitutes the simplest constrained control structure: the number of poles in the feedback loop is limited to n_c. There is a single processor connected to every sensor and every actuator. Such an architecture corresponds to the fixed order LQG problem. If n_c is equal to n, one gets the usual LQG problem. For large scale systems, the order of the model can be quite high and there might be a limit to the order of the compensator one can realize. Instead of reducing the order of the plant to match that of the compensator, one can solve the reduced order LQG problem in order to get the best LQG compensator of order n_c for the plant. The optimality conditions of Proposition 3.1 can be put in the classical form of two uncoupled Riccati equations when n_c is equal to n, [Kwa72b], or can be transformed as an Optimal Projection Equation problem if n_c is strictly less than n [Hyl84], as it will be shown in Chapter 4.

3.5.2 Decentralized Fixed Order Controller

The control law is implemented on p separate processors, each of which is limited in the number of poles it can realize. Each processor receives information from its own set of sensors and control its own set of actuators. No sensor and no actuator can be shared by any two loops. The compensator is then a set of p totally independent loops. With no loss of generality, it can be assumed that the first l_1 sensors are attributed to compensator 1, the next l_2 to compensator 2, etc., and that compensator 1 drives the first m_1 actuators, compensator 2 drives the next m_2 etc. G and K are then block diagonal:

$$G = \begin{bmatrix} G_1 & 0 & \cdots & 0 \\ 0 & G_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & G_p \end{bmatrix}, \; G_i \in \mathbb{R}^{m_i \times n_i}$$

$$K = \begin{bmatrix} K_1 & 0 & \cdots & 0 \\ 0 & K_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & K_p \end{bmatrix}, \; K_i \in \mathbb{R}^{n_i \times l_i}$$

The compensator transfer function T_C is then also block diagonal:

$$T_C(s) = \begin{bmatrix} G_1(sI-A_1)^{-1}K_1 & 0 & \cdots & 0 \\ 0 & G_2(sI-A_2)^{-1}K_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & G_p(sI-A_p)^{-1}K_p \end{bmatrix}$$

Such a structure can be used for systems made of very weakly coupled subsystems: on an aircraft for example, the handling qualities and the engine operations, even though coupled, are very distinct subsystems that can be controlled separately. On a large structure, one might want to geographically distribute the computation as well as the information flow. If the structure of the plant consists of various modules, it is conceivable that each module has its own set of sensors, actuators and computation capability and that, because of wiring and interface complexity, one desires to keep the control loops of each module independent.

3.5.3 Overlapping Controller

The purpose is still to limit the information flow in the feedback loop and to give the processors limited information and limited authority. Here, each processor has its own sensors and actuators but is allowed to receive information from the sensors of neighboring processors. G is therefore still block diagonal, but K has a tridiagonal block structure:

$$G = \begin{bmatrix} G_{11} & 0 & 0 & 0 \cdots 0 \\ 0 & G_{22} & 0 & 0 \cdots 0 \\ 0 & 0 & G_{33} & 0 \cdots 0 \\ \vdots & \vdots & & \vdots \end{bmatrix}, \ G_{ij} \in \mathbb{R}^{m_i \times n_j},$$

$$K = \begin{bmatrix} K_{11} & K_{12} & 0 & 0 \cdots 0 \\ K_{21} & K_{22} & K_{23} & 0 \cdots 0 \\ 0 & K_{32} & K_{33} & K_{34} \cdots 0 \\ \vdots & \vdots & & \vdots \end{bmatrix}, \ K_{ij} \in \mathbb{R}^{n_i \times l_j}$$

The compensator transfer function T_C has the same tridiagonal structure as K inthat case:

$$T_{ii}(s) = G_{ii}(sI - A_i)^{-1} K_{ii}$$

$$T_{ii+1}(s) = G_{ii}(sI - A_i)^{-1} K_{ii+1}$$

$$T_{ii-1}(s) = G_{ii}(sI - A_i)^{-1} K_{ii-1}$$

Such an architecture can also appear on a large flexible structure that possesses many distributed sensors and actuators. The dynamics of the structure at one point depend on the inertia and external forces occurring at that point as well as the dynamics of the neighboring points. Consequently, it makes sense to obtain information about the local dynamics of neighboring elements in order to attenuate the propagation of vibrations in the structure.

3.5.4 Hierarchic Controller

One way to implement a hierarchic control scheme is to let one of the processors have more authority than the remaining ones. This means that one of the processor can receive information from all the sensors and that it will drive all the actuators. G and K are:

$$G = \begin{bmatrix} G_1 & 0 & \cdots & 0 & G_{1p} \\ 0 & G_2 & \cdots & 0 & G_{2p} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & G_{p-1} & G_{p-1p} \end{bmatrix}$$

$$K = \begin{bmatrix} K_1 & 0 & \cdots & 0 \\ 0 & K_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & K_{p-1} \\ K_{p1} & K_{p2} & & K_{pp-1} \end{bmatrix}$$

Processor number p plays the role of an upper level or global controller. Assuming that this processor is of reduced order, and assuming that its dynamics are much slower than that of the processors $i = 1, \cdots, p-1$, the compensator transfer function $T_C(s)$ will be approximately block diagonal at high frequency, as in the case of the decentralized architecture, corresponding to high bandwidth, local control; at low frequency, T_C will be a full matrix with interconnection between every sensor and every actuator, corresponding to global, low bandwidth control.

3.5.5 Fixed Dynamics Controller and Frequency Weighted Cost

Assume the dynamic of the controller is the realization of band-pass filters applied to the outputs of the sensors and that these filters have been designed before starting the optimization procedure. A_c, and K are therefore fixed, and G is restricted to be: $G = FG_0$, where G_0 is chosen such that the triplet (G_0, A_c, K) realizes the transfer functions of the filters. F is an m x 1 matrix of free parameters. If one augments the plant with the dynamics of the various filters, the problem becomes a Fixed Architecture Static Output Feedback problem. Other design techniques used to obtain frequency domain results using the LQG methodology can be used here as well. Integrators can be used in order to remove steady state tracking error. The cost can be shaped as well [Gup80]. The procedure is to filter the variables of interest or filter the process noise to make it colored. The plant must then be augmented with the states of the filter. Any given fixed architecture LQG problem can then be solved using the modified plant. The frequency shaping of the cost has the same effect on the full order LQG problem as on any fixed architecture problem.

3.6 CONCLUSION

First and second order optimality conditions for two types of H_2 Optimal Fixed Architecture Control problems have been derived in this chapter. The first type of problem is a constrained LQG problem which results in multiloop dynamic compensation. The loops are built around parallel processors which dynamically connect selected sets of sensors to selected sets of actuators. The second type of problem is a constrained LQR problem that yields a static output feedback compensator. The use of both types of controllers renders possible the design of compensators such as fixed order, decentralized, overlapping, hierarchic or fixed dynamics compensators.

INVESTIGATION OF THE PROPERTIES OF THE OPTIMALITY CONDITIONS

4.1 INTRODUCTION

The H_2 fixed architecture control problem being a generalization of the classical LQG problem, one would hope for some properties of the unconstrained problem to extend to the fixed architecture case. One well known property of the unconstrained LQG problem is the separation principle: its resolution separates into an optimal full state feedback and an optimal filtering problem. Two Riccati equations, the Control and the Filter Algebraic Riccati Equations (CARE and FARE) are solved to find the control gains and the filter gains. These equations do not appear immediately, however, when the unconstrained LQG problem is set as the optimization of the cost over the class of centralized full order compensators, which is a class of compensators that contains the overall optimal solution. The CARE and the FARE are hidden in the optimality conditions Eqs.(3.3.19-23) that hold for this problem. One must, therefore, transform Eqs.(3.3.19-23) in order to eliminate the matrices G, A_c and K from the problem and perform some block transformations on the matrices P and Q, which are 2n-dimensional, in order to obtain two Riccati equations. Similarly, the Optimal Projection Equations (OPE) that hold in the case of the reduced order compensator problem, [Hyl84], are also hidden in Eqs.(3.3.19-23). The OPE show that the reduced order compensator problem consists of two coupled control and filtering problems where the coupling takes the form of an oblique projection (that is not necessarily orthogonal) [Hyl84]. The control gains, filter gains, and the compensator dynamics depend on the four n-dimensional nonnegative symmetric matrices solutions to the OPE.

The first part of the chapter will be devoted to transforming the optimality conditions derived in Chapter 3 in order to unveil the structure of the Fixed Architecture Control Problem. The structured optimality conditions become, in particular, the usual CARE and FARE when the compensator is centralized and full order, and they become the OPE in the case of the reduced order compensator problem. What happens, in that case, is that the control gains, the filter gains, and the compensator dynamics can be eliminated from the problem and can be expressed as functions of the matrices P and Q as well as the parameters of the problem. When the values of G, A_c and K are substituted into the equations yielding P and Q, one obtains sets of equations which only depend on P and Q. These equations may be more complex, but they depend on a smaller number of variables. It will be shown in this chapter that G, A_c and K can also be found as functions of P and Q, but that one cannot, however, obtain an analytic expression for G, A_c and K as a function of P and Q, in the case of the fixed architecture control problem. Hence, one cannot derive equations yielding P and Q that involve P and Q only, and one must always resort to the use G, A_c and K as intermediate variables.

The remainder of the chapter focuses on the study of the properties of the optimality conditions in their original form, since the structured conditions do not bring any simplification. It will be shown that Eqs.(3.3.19-23) do not define a well-posed problem. The optimality conditions define the compensator transfer function, but they do not select any specific realization for that transfer function. Hence, if (G, A_c, K) is a particular state space realization of the compensator transfer function that satisfies the optimality conditions, so will any realization obtained through a similarity transformation. The solution to the problem is a class of equivalence, where two triplets (G, A_c, K) are said to be equivalent if they realize the same transfer function. The most general problem is, therefore, singular. The singularity can be removed or reduced in a number of ways. All approaches consist of trying to constrain the realization of the compensator. They are of very pratical importance since numerical methods converge faster when the solution is an isolated point in parameter space.

The last part of the chapter is devoted to showing that the problem can have solutions which are saddle points. Such an occurrence is very important since it shows that the problem is not convex and that the Eqs.(3.3.19-23) may have more than one solution.

The chapter begins with mathematical preliminaries. They will focus on the matrix Lyapunov equation which plays a predominant role in the H_2 problem. A theorem about existence and uniqueness of the solution to a linear system of matrix equations will also be proven.

4.2 MATHEMATICAL PRELIMINARIES

4.2.1 The Matrix Equation AX – XB = C

Proposition 4.1: *Given the matrices* $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times m}$, X, $C \in \mathbb{R}^{n\times m}$, *the matrix equation*

$$AX - XB = C$$

is a linear equation in X. *It has a unique solution if and only if* A *and* B *have no common eigenvalues. If* A *and* B *share an eigenvalue* λ, *and denoting by* m_a *the multiplicity of* λ *as a characteristic value of* A, *and* m_b *the multiplicity of* λ *as a characteristic value of* B, *the equation will admit no solution if there exists* $w \in \mathrm{Ker}(\lambda I - A^T)^{m_a}$ *and* $v \in \mathrm{Ker}(\lambda I - B)^{m_b}$ *such that:* $w^T C v \neq 0$. *The equation has an infinite number of solutions otherwise.* ■

Proof: The proof can be found in Gantmacher, Chapter VIII [Gan59]. If each eigenvalue has multiplicity 1, w is a left eigenvector for A and v is a right eigenvector for B. In the more general case, one has to consider a Jordan decomposition, and the multiplicity of the eigenvalue has to be taken into account. □

Proposition 4.2: *The Lyapunov equation:*

$$0_{n \times n} = AX + XA^T + V \qquad (4.2.1)$$

respectively:

$$0_{n \times n} = A^TX + XA + R \qquad (4.2.2)$$

where A, X, V, R $\in \mathbb{R}^{n \times n}$, *has a unique solution if and only if* A *does not have any eigenvalue on the imaginary axis. If* V (*respectively* R) *is symmetric and positive semidefinite, and if* (A, $V^{1/2}$) *is stabilizable,* (*respectively* ($R^{1/2}$, A) *is detectable*) *the following statements are equivalent:*

i) A *is asymptotically stable*

ii) X *symmetric, and* X ≥ 0 ∎

Proof: The uniqueness of the solution comes directly from Proposition 4.1 for B $= -A^T$. The proof of the second part of the proposition can be found in [Kai80] pp 178-179. The present formulation is more general, however, since it does not insist on X being positive definite. Assume X is singular, and let v be a singular vector of X. Premultiplying Eq.(4.2.1) by v^T and postmultiplying by v yields:

$$0 = v^TVv$$

V being positive semidefinite, this is is equivalent to:

$$0_n = V^{1/2}v \qquad (4.2.3)$$

Postmultiplying Eq.(4.2.1) by v only, one gets:

$$0_n = XA^Tv,$$

implying that the nullspace of X is invariant under A: there must be an eigenvector of A^T, w, associated with an eigenvalue λ , which is in the nullspace of X. w satisfies Eq.(4.2.3), which implies that the corresponding mode is uncontrollable. The pair (A,$V^{1/2}$) being stabilizable, λ is strictly negative. Even if X is only semidefinite, A is asymptotically stable. □

4.2.2 Solution to a Linear System of Matrix Equations

Theorem 4.1: *Let* C, D, X, R *be matrices in* $\mathbb{R}^{m \times m}$, $\mathbb{R}^{n \times n}$, $\mathbb{R}^{m \times n}$ *and* $\mathbb{R}^{m \times n}$ *respectively,* C *and* D *symmetric, positive semidefinite. Let* $E_x = \{ E_1, E_2, \cdots, E_p\}$ *be a family of independent matrices on* $\mathbb{R}^{m \times n}$. *Consider the system of equations:*

$$X = E_x * x,\ x \in \mathbb{R}^p$$
$$\mathrm{Tr} E_j^T C X D = \mathrm{Tr} E_j^T R, \quad j = 1, \cdots, p \tag{4.2.4}$$

where $*$ *follows Definition* 3.1, *and* E_x *defines the subspace where* X *lies. Such a system is a linear system. It has a unique solution if* C *and* D *are definite. It has an infinite number of solutions, otherwise, if:*

$$\mathrm{Ker}(C) \subset \mathrm{Ker}(R^T)$$
$$\mathrm{Ker}(D) \subset \mathrm{Ker}(R) \tag{4.2.5}$$

■

Proof: The proof is deferred to Appendix A.

4.2.3 Generalized Inverses

Definition 4.1: *Given* $X \in \mathbb{R}^{n \times n}$, *with the following singular value decomposition:*

$$X = U \begin{bmatrix} \Sigma & 0_{m \times r} \\ 0_{r \times m} & 0_{r \times r} \end{bmatrix} V^H$$

where $U, V \in \mathbb{C}^{n \times n}$, *unitary,* $\Sigma \in \mathbb{R}^{m \times m}$, *diagonal, with strictly positive diagonal elements,* $r = n - m$; *Define* $X^\dagger$:

$$X^\dagger = V \begin{bmatrix} \Sigma^{-1} & 0_{m \times r} \\ 0_{r \times m} & 0_{r \times r} \end{bmatrix} U^H$$

$X^\dagger$ *is the Moore-Penrose generalized inverse of* X [Cam79]. $X^\dagger$ *satisfies the* *following properties:*

$$XX^{\dagger}X = X$$
$$(XX^{\dagger}) = (XX^{\dagger})^{H}$$
$$(X^{\dagger}X) = (X^{\dagger}X)^{H}$$
$$X^{\dagger}XX^{\dagger} = X^{\dagger}$$

$X^{\dagger}X$ *is an orthogonal projector parallel to* Ker(X), *the nullspace of* X. *When* X *is symmetric,* $XX^{\dagger} = X^{\dagger}X$. ∎

The properties given in the definition can be obtained simply by inspection using the singular value decomposition. □

Definition 4.2: *Consider* $X \in \mathbb{R}^{n \times n}$, *such that* X *and* X^2 *have the same range.* X *has then a Jordan decomposition of the following form:*

$$X = L \begin{bmatrix} J_1 0 & \cdots 0 & 0 \\ 0\ J_2 & \cdots 0 & 0 \\ \vdots & \ddots & \vdots \\ 0\ 0 & \cdots J_r & 0 \\ 0\ 0 & \cdots 0 & 0 \end{bmatrix} L^{-1},$$

where the J_i *are nonsingular. Define* $X^{\#}$ *as:*

$$X^{\#} = L \begin{bmatrix} J_1^{-1} & 0 & \cdots & 0 & 0 \\ 0 & J_2^{-1} & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \cdots & J_r^{-1} & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} L^{-1}$$

$X^{\#}$ *is the group inverse of* X. *It satisfies:*

$$X = XX^{\#}X$$
$$X^{\#} = X^{\#}XX^{\#}$$

$XX^{\#}$ *is an oblique projection on the range of X and parallel to its nullspace.*

∎

4.3 STRUCTURED OPTIMALITY CONDITIONS

4.3.1 Eliminating G, K and A_c from the Optimality Conditions

The resolution of the full order LQG problem can be decomposed into two successive steps. The first one consists of solving two uncoupled Riccati equations which can be obtained by simplifying Eqs.(3.3.19-20) and replacing the compensator gains by their expressions as functions of P and Q. The solutions to the Riccati equations can then be used to obtain the optimal gains. If one desires to generalize the procedure to the constrained problem, one must be able to find analytic expressions of G, A_c and K as functions of P and Q. Sections 4.3.1.1 and 4.3.1.2 show how G, K and A_c are defined, and in most cases uniquely, as functions of P, Q and the parameters of the problem. A preliminary derivation consists of partitioning of Eqs.(3.3.19-20). They become:

$$0_{n\times n} = Q_{00}A^T + AQ_{00} + BGQ_{c0} + Q_{0c}G^TB^T + V \tag{4.3.1}$$

$$0_{n\times n_c} = Q_{0c}A_c^T + AQ_{0c} + BGQ_{cc} + Q_{00}C^TK^T \tag{4.3.2}$$

$$0_{n_c\times n} = Q_{c0}A^T + A_cQ_{c0} + KCQ_{00} + Q_{cc}G^TB^T \tag{4.3.3}$$

$$0_{n_c\times n_c} = Q_{cc}A_c^T + A_cQ_{cc} + KCQ_{0c} + Q_{c0}C^TK^T + KV_cK^T \tag{4.3.4}$$

$$0_{n\times n} = P_{00}A + A^TP_{00} + P_{0c}KC + C^TK^TP_{c0} + R \tag{4.3.5}$$

$$0_{n\times n_c} = P_{0c}A_c + A^TP_{0c} + P_{00}BG + C^TK^TP_{cc} \tag{4.3.6}$$

$$0_{n_c\times n} = P_{c0}A + A_c^TP_{c0} + P_{cc}KC + G^TB^TP_{00} \tag{4.3.7}$$

$$0_{n_c\times n_c} = P_{cc}A_c + A_c^TP_{cc} + P_{c0}BG + G^TB^TP_{0c} + G^TR_cG \tag{4.3.8}$$

4.3.1.1 <u>Solving for G and K</u>

Proposition 4.3: *The system of equations consisting of the architecture constraints and the optimality conditions for the control gain* G, Eqs.(3.3.17)*:*

$$G = E_G * g,$$
$$0 = \mathrm{Tr}E^{g}{}_i{}^T(R_c G Q_{cc} + B^T M_{0c}),\ i = 1,\cdots,n_g$$

and the system of equations consisting of the architecture constraints and the optimality conditions for K, Eqs.(3.3.18)*:*

$$K = E_K * k,$$
$$0 = \mathrm{Tr}E^{k}{}_i{}^T(P_{cc} K V_c + M_{c0} C^T),\ i = 1,\cdots,n_k$$

always admit at least one solution. Given P *and* Q, *it is always possible to find* G *and* K *that satisfy the optimality conditions of* Eqs.(3.3.17,18) *as well as the architecture constraints defined in Section* 3.3.1.1. ■

Proof: The problems satisfied by G and K are similar to the one covered by Theorem 4.1. One must check that their right hand sides satisfy the rank conditions expressed by Eq.(4.2.5). The following lemma must be proven first:

Lemma 4.1: *The following statements hold as long as* P *an* Q *are nonnegative*:

$\mathrm{Ker}(P_{cc})$ *is included in* $\mathrm{Ker}(P_{0c})$.

$\mathrm{Ker}(P_{cc})$ *is included in* $\mathrm{Ker}(G)$.

$\mathrm{Ker}(P_{cc})$ *is the unobservable subspace of* (G, A_c).

$\mathrm{Ker}(Q_{cc})$ *is included in* $\mathrm{Ker}(Q_{0c})$.

$\mathrm{Ker}(Q_{cc})$ *is included in* $\mathrm{Ker}(K^T)$.

$\mathrm{Ker}(Q_{cc})$ *is the uncontrollable subspace of* (A_c, K). ■

Proof: Let $v \in \mathrm{Ker}(P_{cc})$: $P_{cc}v = 0_{n_c}$. Consider the following vector X:

$$X = \begin{bmatrix} -P_{0c}v \\ \alpha v \end{bmatrix},\ \alpha \in \mathbb{R}$$

P being positive, $X^T P X$ is always nonnegative. Developing $X^T P X$:

$$X^T P X = v^T P_{0c}{}^T P_{00} P_{0c} v - 2\alpha v^T P_{0c}{}^T P_{0c} v$$

The expression is positive for all α if and only if $P_{0c}v = 0_n$. Thus $v \in \mathrm{Ker}(P_{0c})$ and $\mathrm{Ker}(P_{cc}) \subset \mathrm{Ker}(P_{0c})$.

Pre and postmultiply now Eq.(4.3.8) by v^T and v. Because both $P_{cc}v = 0_{nc}$ and $P_{0c}v = 0_n$, we get that $v^T G^T R_c G v = 0$. R_c being positive definite, $Gv = 0_m$ and $v \in Ker(G)$: hence, $Ker(P_{cc}) \subset Ker(G)$.

Postmultiply now Eq.(4.3.8) by v. Since $Gv = 0$, $P_{cc}v = 0_{nc}$, and $P_{0c}v = 0_n$, this implies that $P_{cc}A_c v = 0_{nc}$: $Ker(P_{cc})$ is invariant under A_c. $Ker(P_{cc})$ is therefore an invariant subspace of A_c, and every vector in this subspace satisfies $Gv = 0_m$. Thus $Ker(P_{cc})$ is included in the unobservable subspace of the pair (G, A_c). Similarly, consider v such that $A_c v = \lambda v$ and $Gv = 0_m$. Pre and postmultiplying Eq.(4.3.8) by v^T and v yields $2\lambda v^T P_{cc} v = 0$. If $\lambda \neq 0$, P_{cc} being positive, $P_{cc}v = 0_{nc}$ and the unobservable space of (G, A_c) is included in $Ker(P_{cc})$. If $\lambda = 0$, post multiplying Eq.(4.3.6) and Eq.(4.3.8) by v yields:

$$0_n = A^T P_{0c} v + C^T K^T P_{cc} v$$

$$0_{nc} = G^T B^T P_{0c} v + A_c^T P_{cc} v$$

These last two equalities can be written more succintly as:

$$0_{1 \times \tilde{n}} = v^T [P_{c0}\ P_{cc}] A_{cl}$$

A_{cl} being assumed strictly stable, this implies that $P_{cc}v = 0_{nc}$ and $P_{0c}v = 0_n$: $Ker(P_{cc})$ is the unobservable space of (G, A_c). The rest of the lemma is obtained by duality. This ends the proof of Lemma 4.1. □

Checking Eq.(4.2.5) on the two systems is now obvious once M_{0c} and M_{c0} have been developed according to Eqs.(3.3.2). This ends the proof of Proposition 4.3.

□

4.3.1.2 Solving for A_c

A_c does not appear explicitly in the optimality conditions Eq.(3.3.21). A system of equations similar to that of Theorem 4.1 can however be obtained. The combination Eq.(4.3.7)Q_{0c} + Eq.(4.3.8)Q_{cc} yields:

$$\begin{aligned} 0_{n_c \times n_c} = {} & P_{cc}A_c Q_{cc} + P_{c0} A Q_{0c} + P_{cc} K C Q_{0c} + P_{c0} B G Q_{cc} + \\ & + A_c^T (P_{c0} Q_{0c} + P_{cc} Q_{cc}) + \\ & + G^T (R_c G Q_{cc} + B^T (P_{00} Q_{0c} + P_{0c} Q_{cc})) \end{aligned}$$

or, using Eq.(3.3.2) and Eq.(3.3.9):

$$0_{n_c \times n_c} = P_{cc}A_cQ_{cc} + P_{c0}AQ_{0c} + P_{cc}KCQ_{0c} + P_{c0}BGQ_{cc} + \\ + A_c^TL_{A_c} + G^TL_G \qquad (4.3.9)$$

Similarly, the combination P_{c0}Eq.(4.3.2) + P_{cc}Eq.(4.3.4) yields:

$$0_{n_c \times n_c} = P_{cc}A_cQ_{cc} + P_{c0}AQ_{0c} + P_{cc}KCQ_{0c} + P_{c0}BGQ_{cc} + \\ + L_{A_c}A_c^T + L_K{}^TK \qquad (4.3.10)$$

A_c being block diagonal, it can be written as:

$$A_c = \sum_{i=1}^{p} \Pi_i{}^T A_i \Pi_i$$

where Π_i is defined in Section 3.2.3. Similarly, G and K can be written as:

$$G = \sum_{j=1}^{p} \sum_{i \in \mathcal{U}_j} \pi_i^{m\,T} G_{ij} \Pi_j$$

$$K = \sum_{i=1}^{p} \sum_{j \in \mathcal{Y}_i} \Pi_i{}^T K_{ij} \pi_j^l$$

Π_i and π_i^k satisfy the following properties:

$$\begin{aligned} \Pi_i\Pi_j{}^T &= 0_{n_i \times n_j} && \text{if } i \neq j, \\ &= I_{n_i} && \text{if } i = j, \\ \pi_i^k \pi_j^{k\,T} &= 0 && \text{if } i \neq j, \\ &= 1 && \text{if } i = j, \end{aligned} \qquad (4.3.11)$$

Using the expanded forms of A_c, G and K, and using the optimality conditions of Eqs.(3.3.21-23), one obtains the following expression by pre and postmultiplying Eq.(4.3.9) by Π_i and $\Pi_i{}^T$:

$$0_{n_i \times n_i} = \Pi_i(P_{cc}A_cQ_{cc} + P_{c0}BGQ_{cc} + Q_{cc}KCP_{0c})\Pi_i{}^T, \; i = 1,\cdots,p$$

or equivalently:

$$\mathrm{Tr}E^{a\,T}_{j}P_{cc}A_cQ_{cc} = -\mathrm{Tr}E^{a\,T}_{j}(P_{c0}AQ_{0c} + P_{cc}KCQ_{0c} + P_{c0}BGQ_{cc})$$

$$j = 1,\cdots,n_a \qquad (4.3.12)$$

The same system is obtained by pre and postmultiplying Eq.(4.3.10) by Π_i and Π_i^T. This system is similar to that of Theorem 4.1 and will produce A_c.

Proposition 4.4: *The system of equations consisting of the architecture constraints and the modified optimality conditions for* A_c*:*

$$A_c = E_{A_c} * a_c,$$

$$\mathrm{Tr}E^{a\,T}_{i}(P_{cc}A_cQ_{cc}) = -\mathrm{Tr}E^{a\,T}_{i}(P_{c0}AQ_{0c} + P_{c0}BGQ_{cc} + P_{cc}KCQ_{0c}),\ i = 1,\cdots,n_a$$

always admits at least one solution. Given P, *and* Q, G *and* K *can be computed (Proposition* 4.3*), and, consequently,* A_c *can always be determined.* ■

Proof: the proof is exactly similar to that of Proposition 4.3. Lemma 4.1 guarantees that the rank conditions are met. When the compensator is uncontrollable or unobservable, the dynamics of the corresponding mode is arbitrary. Its choice must correspond to a stable mode if one wants the overall closed loop system to be stable.

□

4.3.2 Structured Optimality Conditions

Proposition 4.5: *The following matrix equations hold if* A_{cl} *is asymptotically stable:*

$$P_{0c} = P_{0c}P_{cc}^{\dagger}P_{cc} \tag{4.3.13}$$
$$P_{c0} = P_{cc}P_{cc}^{\dagger}P_{c0} \tag{4.3.14}$$
$$Q_{0c} = Q_{0c}Q_{cc}^{\dagger}Q_{cc} \tag{4.3.15}$$
$$Q_{c0} = Q_{cc}Q_{cc}^{\dagger}Q_{c0} \tag{4.3.16}$$

■

Proof: From the properties of the generalized inverse seen above, we know that $P_{cc}^{\dagger}P_{cc}$ is an orthogonal projection parallel to $\mathrm{Ker}(P_{cc})$. Eq(4.3.13) is equivalent to showing that the nullspace of P_{cc} is included in that of P_{0c}. This result follows from Lemma 4.1. Similarly, Eq.(4.3.15) follows from lemma 4.1 and Eqs.(4.3.14,16) are just transposed forms of Eqs.(4.3.13,15). □

Theorem 4.2: *Problem* 3.1 *has a solution if there exists* $\mathbf{P}$, $\mathbf{Q}$, $\hat{\mathbf{P}}$, $\hat{\mathbf{Q}} \in \mathbb{R}^{n\times n}$, *symmetric positive,* P_{cc} *and* $Q_{cc} \in \mathbb{R}^{nc\times nc}$, *symmetric positive, as well as* Φ *and* $\Gamma \in \mathbb{R}^{n\times nc}$ *and* G, A_c *and* K *such that:*

$$0_{n\times n} = \mathbf{A}\mathbf{Q}+\mathbf{Q}\mathbf{A}^T+\mathbf{V}-\mathbf{Q}\Sigma\mathbf{Q}+(\Phi\mathbf{K}\mathbf{V}_c-\mathbf{Q}\mathbf{C}^T)\mathbf{V}_c^{-1}(\Phi\mathbf{K}\mathbf{V}_c-\mathbf{Q}\mathbf{C}^T)^T \tag{4.3.17}$$

$$0_{n\times n} = \mathbf{A}^T\mathbf{P}+\mathbf{P}\mathbf{A}+\mathbf{R}-\mathbf{P}\Sigma\mathbf{P}+(\Gamma\mathbf{G}\mathbf{R}_c-\mathbf{P}\mathbf{B})\mathbf{R}_c^{-1}(\Gamma\mathbf{G}^T\mathbf{R}_c-\mathbf{P}\mathbf{B})^T \tag{4.3.18}$$

$$0_{n\times n} = \mathbf{A}\hat{\mathbf{Q}}+\hat{\mathbf{Q}}\mathbf{A}^T+\Phi\mathbf{Q}_{cc}\mathbf{G}^T\mathbf{B}^T+\mathbf{B}\mathbf{G}\mathbf{Q}_{cc}\Phi^T+\mathbf{Q}\Sigma\mathbf{Q}-(\Phi\mathbf{K}\mathbf{V}_c-\mathbf{Q}\mathbf{C}^T)\mathbf{V}_c^{-1}(\Phi\mathbf{K}\mathbf{V}_c-\mathbf{Q}\mathbf{C}^T)^T \tag{4.3.19}$$

$$0_{n\times n} = \mathbf{A}^T\hat{\mathbf{P}}+\hat{\mathbf{P}}\mathbf{A}+\Gamma\mathbf{P}_{cc}\mathbf{K}\mathbf{C}+\mathbf{C}^T\mathbf{K}^T\mathbf{P}_{cc}\Gamma^T+\mathbf{P}\Sigma\mathbf{P}-(\Gamma\mathbf{G}^T\mathbf{R}_c-\mathbf{P}\mathbf{B})\mathbf{R}_c^{-1}(\Gamma\mathbf{G}^T\mathbf{R}_c-\mathbf{P}\mathbf{B})^T \tag{4.3.20}$$

$$0_{n_c\times n_c} = (A_c+KC\Phi)Q_{cc}+Q_{cc}(A_c+KC\Phi)^T+KV_cK^T \tag{4.3.21}$$

$$0_{n_c\times n_c} = (A_c+\Gamma^T BG)^T P_{cc}+P_{cc}(A_c+\Gamma^T BG)+G^T R_c G \tag{4.3.22}$$

$$0_{n_i\times n_i} = \Pi_i P_{cc}(I_{n_c} + \Gamma^T\Phi)Q_{cc}\Pi_i^T \qquad i = 1,\cdots,p \tag{4.3.23}$$

$$0_{1\times n_i} = \pi_j^m\left[R_cG + B^T\left[P\Phi + \Gamma P_{cc}(I_{n_c}+\Gamma^T\Phi)\right]\right]Q_{cc}\Pi_i^T,\ j \in \mathcal{U}_i,$$
$$i = 1,\cdots,p \tag{4.3.24}$$

$$0_{n_i\times 1} = \Pi_i P_{cc}\left[KV_c + \left[(I_{n_c} + \Gamma^T\Phi)Q_{cc}\Phi^T + \Gamma^T Q\right]C^T\right]\pi_j^{1T},\ j \in \mathcal{Y}_i$$
$$i = 1,\cdots,p \tag{4.3.25}$$

$$0_{n_i\times n_i} = \Pi_i P_{cc}(A_c + \Gamma^T A\Phi + \Gamma^T BG + KC\Phi)Q_{cc}\Pi_i^T, \qquad i = 1,\cdots,p \tag{4.3.26}$$

■

Proof: the proof is a matter of algebra. Define the following matrices,

$$\mathbf{P} = P_{00} - P_{0c}P_{cc}^{\dagger}P_{c0} \tag{4.3.27}$$
$$\mathbf{Q} = Q_{00} - Q_{0c}Q_{cc}^{\dagger}Q_{c0} \tag{4.3.28}$$
$$\hat{\mathbf{P}} = P_{0c}P_{cc}^{\dagger}P_{c0} \tag{4.3.29}$$
$$\hat{\mathbf{Q}} = Q_{0c}Q_{cc}^{\dagger}Q_{c0} \tag{4.3.30}$$
$$\Phi = Q_{0c}Q_{cc}^{\dagger} \tag{4.3.31}$$
$$\Gamma = P_{0c}P_{cc}^{\dagger} \tag{4.3.32}$$
$$\Sigma = BR_c^{-1}B^T \tag{4.3.33}$$
$$\bar{\Sigma} = C^T V_c^{-1} C \tag{4.3.34}$$

Eq. (4.3.17) = Eq. (4.3.1)–Eq. (4.3.2)$Q_{cc}^{\dagger}Q_{c0}$ – $Q_{0c}Q_{cc}^{\dagger}$Eq. (4.3.3), as well as using Eq. (4.3.4) to eliminate A_c and Proposition 4.5 to regroup terms.
Eq. (4.3.18) = Eq. (4.3.5)–Eq. (4.3.6)$P_{cc}^{\dagger}P_{c0}$ – $P_{0c}P_{cc}^{\dagger}$Eq. (4.3.7), as well as using Eq. (4.3.8) to eliminate A_c and Proposition 4.5 to regroup terms.
Eq. (4.3.19) = Eq. (4.3.1) – Eq. (4.3.17).
Eq. (4.3.20) = Eq. (4.3.5) – Eq. (4.3.18).
Eq. (4.3.21,22) = Eq. (4.3.4,8).
Eq. (4.3.23) = Eq. (3.3.21).
Eq. (4.3.24,25) = Eq. (3.3.22,23).
Eq. (4.3.26) = Eq. (4.3.12). □

The full order LQG problem and the reduced order LQG problem are both specific cases of the fixed architecture LQG problem, and both satisfy Eqs.(4.3.17-26). The optimality conditions are not so complicated in those cases: we shall now show how the increasing specificity of the architecture increases the coupling between the different optimality conditions.

4.3.2.1 Full Order LQG

The architecture parameters for the full order compensator problem are: $p = 1$, $n_c = n$, $\Pi_1 = I_n$. Eq.(4.3.23) becomes:

$$I_n = -\Gamma^T \Phi$$

Eq.(4.3.24,25) become:

$$G = -R_c^{-1} B^T P \Phi$$
$$K = -\Gamma^T Q C^T V_c^{-1}$$

This implies that the two positive terms in Eqs.(4.3.17,18) are zero and the two Riccati equations decouple. **P** and **Q** satisfy:

$$0_{n \times n} = AQ + QA^T + V - Q\Sigma Q$$
$$0_{n \times n} = A^T P + PA + R - P\Sigma P$$

Eq.(4.3.26) becomes:

$$A_c = -\Gamma^T (A - BG - KC)\Phi$$

The solution is the classic LQG solution with the two independent Control and Filter Riccati equations. The choice of Φ and Γ such that $\Gamma^T \Phi = -I_n$ correspond to choosing different state space realization for the controller. Eqs.(4.3.21,22) allow one to compute the optimal cost by producing Q_{cc} and P_{cc}.

4.3.2.2 Optimal Projection Equation

The OPE corresponds to $p = 1$ and $n_c < n$.

Eq.(4.3.23) becomes:

$$I_{n_c} = -\Gamma^T \Phi$$

As shown in [Hyl84], Eq.(4.3.23) implies that the matrix $\tau = -\Phi\Gamma^T \in \mathbb{R}^{n \times n}$ is idempotent: $\tau^2 = \tau$. τ is therefore a projection operator. Γ and Φ being two $n \times n_c$

matrices, the rank of τ is at most n_c. Eqs.(4.3.24,25) become:

$$G = -R_c^{-1}B^TP\Phi$$
$$K = -\Gamma^TQC^TV_c^{-1}$$

Defining

$$\tau_\perp = I_n - \tau$$
$$= I_n + \Phi\Gamma^T$$

it follows that:

$$\Phi KV_c - QC^T = -\tau_\perp QC^T$$
$$\Gamma GR_c - PB = -\tau_\perp^T PB$$

The modified Riccati equations Eqs.(4.3.17,18) and the Lyapunov equations Eqs.(4.3.19,20) become:

$$0_{n\times n} = AQ + QA^T + V - Q\Sigma Q + \tau_\perp Q\Sigma Q\tau_\perp^T \tag{4.3.35}$$
$$0_{n\times n} = A^TP + PA + R - P\Sigma P + \tau_\perp^T P\Sigma P\tau_\perp \tag{4.3.36}$$
$$0_{n\times n} = (A-P\Sigma)\hat{Q} + \hat{Q}(A-P\Sigma)^T + Q\Sigma Q - \tau_\perp Q\Sigma Q\tau_\perp^T \tag{4.3.37}$$
$$0_{n\times n} = (A-\Sigma Q)^T\hat{P} + \hat{P}(A-\Sigma Q) + P\Sigma P - \tau_\perp^T P\Sigma P\tau_\perp \tag{4.3.38}$$

The projection operator appears when combining Eq.(4.3.23) and Eqs.(4.3.29,30). From Eq.(4.3.23) the projection is:

$$\tau = -Q_{0c}Q_{cc}^{\dagger}P_{cc}^{\dagger}P_{c0}$$

The product $\hat{Q}\hat{P}$ is, Eqs.(4.3.29,30):

$$\hat{Q}\hat{P} = Q_{0c}Q_{cc}^{\dagger}(Q_{c0}P_{0c})P_{cc}^{\dagger}P_{c0}$$

Thus, τ can be found using the group inverse of $\hat{Q}\hat{P}$, Definition 4.2,

$$\tau = \hat{Q}\hat{P}(\hat{Q}\hat{P})^{\#} \tag{4.3.39}$$

By reducing the order of the compensator, one has introduced some coupling between the control and the filter problem. The separation principle does not hold anymore and one has to find the optimal oblique projection that couples control and filtering. Both problems have to be solved simultaneously. The dynamics of the compensator is simply obtained using Eq.(4.3.26). It yields simply:

$$A_c = -\Gamma^T(A - \Sigma P - Q\Sigma)\Phi \tag{4.3.40}$$

4.3.2.3 The Decentralized Fixed Order Control

We consider here the Decentralized Control Problem where K and G are block diagonal as defined in Section 3.5.3. This is the simplest form of constrained architecture in that G and K are block diagonal and can be computed from Eqs.(4.3.24,25) using simple matrix algebra. Since the control consists of independent loops, it can be shown, [Ber87b], that one can define p optimal reduced order problems and p projection operators. Each subcontroller satisfying the optimality conditions, it is optimal for the system composed of the original plant with the remaining control subcontrollers closed. Each independent compensator satisfies therefore the Optimal Projection Equations Eqs.(4.3.35-39) for the modified plant. Each subcontroller cannot be designed individually since the system on which the local loop is closed is the original plant with the remaining subcontrollers closed. There is, therefore, some loop coupling introduced in the problem. More precisely, Eq.(4.3.23) becomes:

$$0_{n_i \times n_i} = \Pi_i P_{cc}(I_{n_c} + \Gamma^T \Phi) Q_{cc} \Pi_i^T,\ i=1,\cdots,p$$

The set of equations is not enough to completely determine Γ and Φ. The optimal gains G and K are block diagonal Eqs.(3.5.1,2), $G = \text{blockdiag}(G_1, G_2,\cdots,G_p)$, $G_i \in \mathbb{R}^{m_i \times n_i}$, and $K = \text{blockdiag}(K_1, K_2,\cdots,K_p)$, $K_i \in \mathbb{R}^{n_i \times l_i}$. Eqs.(4.3.24,25) can be regrouped and written in matrix form as:

$$G_i = -R_i^{-1} B_i^T \left[P\Phi + \Gamma P_{cc}(I_{n_c} + \Gamma^T \Phi)) \right] Q_{cc} \Pi_i^T Q_{ii}^{-1}$$

$$K_i = -P_{ii}^{-1} \Pi_i P_{cc} \left[\Gamma^T Q + (I_{n_c} + \Gamma^T \Phi) Q_{cc} \Phi^T \right] C_i^T V_i^{-1}$$

G_i and K_i can be eliminated in that case, but an overlapping decomposition of the sensors and actuators among processors result in the more general systems described by Eqs.(4.3.24,25) whose solutions require the use of Theorem 4.1. The dynamics of the compensator A_c is found solving Eq.(4.3.26) which cannot be simplified. All the blocks must be determined at once (Proposition 4.4) unlike in the centralized case which resulted in the simple expression of Eq.(4.3.40). Two mechanisms couple the problem. The first form of coupling appears between the modified filter and control Riccati equations as $(\Phi K V_c - QC^T) V_c^{-1} (\Phi K V_c - QC^T)^T$ and $(\Gamma G^T R_c - PB) R_c^{-1} (\Gamma G^T R_c - PB)^T$ in Eqs.(4.3.17,18). The second form of coupling appears as loop coupling in the form of $P_{cc}(I_{n_c} + \Gamma^T \Phi) Q_{cc}$ which arises in

Eqs.(4.3.24,25). G and K can be solved as functions of P and Q and eliminated, but this will not result in a simple matrix expression. P_{cc} and Q_{cc} do not dissappear as the in previous cases. It indicates the coordinations between the subcontrollers. A_c is required to solve for P_{cc} and Q_{cc}. The structured form of the optimality conditions does not give rise to a decomposition of the optimality conditions as it was the case for the full order and for the reduced order LQG problem. One must solve simultaneously two modified Riccati equations coupled through four Lyapunov equations, and the control parameters must be found as well.

4.4 PROBLEM SINGULARITY

4.4.1 Cost Invariance

The necessary conditions for optimality as they appear in Eqs.(3.3.19-23) form a singular system of equations. In order to prove that fact, it will be shown that certain transformations on the control variables leave the cost unchanged.

Theorem 4.3: *Let* $N \in \mathbb{R}^{n_c \times n_c}$ *be block diagonal*

$$N = \text{blockdiag}(\, N_1, N_2, \cdots, N_p),$$

where $N_i \in \mathbb{R}^{n_i \times n_i}$*, invertible. Also define* $\tilde{N}$ *as:*

$$\tilde{N} = \text{blockdiag}(\, I_n, N)$$

The Lagrangian L *is invariant under the transformation:*

$$\begin{aligned} P &= \tilde{N}^T P \tilde{N} \\ Q &= \tilde{N}^{-1} Q \tilde{N}^{-T} \\ G &= GN \\ A_c &= N^{-1} A_c N \\ K &= N^{-1} K \end{aligned}$$

■

Proof: A_{cl} and R_{cl} and V_{cl} respectively become:

$$
\begin{aligned}
A_{cl} &= \tilde{N}^{-1}A_{cl}\tilde{N} \\
V_{cl} &= \tilde{N}^{-1}V_{cl}\tilde{N}^{-T} \\
R_{cl} &= \tilde{N}^{T}R_{cl}\tilde{N}
\end{aligned}
$$

Twice the Lagrangian of Eq.(3.2.6) becomes:

$$
\begin{aligned}
2\,L(\,GN,\ &N^{-1}A_cN,\ N^{-1}K,\ \tilde{N}^TP\tilde{N},\ \tilde{N}^{-1}Q\tilde{N}^{-T}) = \\
&\mathrm{Tr}\Big[(\tilde{N}^{-1}Q\tilde{N}^{-T})(\tilde{N}^TR_{cl}\tilde{N}) + (\tilde{N}^TP\tilde{N})(\tilde{N}^{-1}V_{cl}\tilde{N}^{-T}) + \\
&+ (\tilde{N}^{-1}A_{cl}\tilde{N})(\tilde{N}^{-1}Q\tilde{N}^{-T})(\tilde{N}^TP\tilde{N}) + (\tilde{N}^TP\tilde{N})(\tilde{N}^{-1}Q\tilde{N}^{-T})(\tilde{N}^TA_{cl}{}^T\tilde{N}^{-T})\Big] \\
&= \mathrm{Tr}(\tilde{N}^{-1}QR_{cl}\tilde{N}) + \mathrm{Tr}(\tilde{N}^TPV_{cl}\tilde{N}^{-T}) + \\
&+ \mathrm{Tr}(\tilde{N}^{-1}A_{cl}QP\tilde{N}) + \mathrm{Tr}(\tilde{N}^TA_{cl}{}^TPQ\tilde{N}^{-T}) \\
&= \mathrm{Tr}(\tilde{N}\tilde{N}^{-1}QR_{cl}) + \mathrm{Tr}(PV_{cl}\tilde{N}^{-T}\tilde{N}^T) + \\
&+ \mathrm{Tr}(\tilde{N}\tilde{N}^{-1}A_{cl}QP) + \mathrm{Tr}(A_{cl}{}^TPQ\tilde{N}^{-T}\tilde{N}^T) \\
&= \mathrm{Tr}(QR_{cl} + PV_{cl} + A_{cl}QP + PQA_{cl}{}^T) \\
&= 2L(\,G,\ A_c,\ K,\ P,\ Q) \qquad \square
\end{aligned}
$$

The transformation N consists of changing the internal realization of each subcontroller. Theorem 4.3 shows that the cost is invariant under such a transformation. Consider now that each N_i has the form:

$$N_i(\varepsilon) = I + \varepsilon M_i,$$

where $M_i \in \mathbb{R}^{n_i \times n_i}$ is any $n_i{\times}n_i$ matrix. For ε sufficiently small N_i is invertible. The differential of N_i with respect to ε is $N_{i_\varepsilon} = M_i$. Hence, $N_\varepsilon = \mathrm{blockdiag}(M_1, M_2,\cdots,M_p)$ and $\tilde{N}_\varepsilon = \mathrm{blockdiag}(0_{n\times n}, N_\varepsilon)$. One can write:

$$L(\varepsilon) = L(GN(\varepsilon), N(\varepsilon)^{-1}A_cN(\varepsilon), N(\varepsilon)^{-1}K, \tilde{N}(\varepsilon)^TP\tilde{N}(\varepsilon), \tilde{N}(\varepsilon)^{-1}Q\tilde{N}(\varepsilon)^{-T})$$

Consider further that P, Q, G, A_c, K satisfy the two Lyapunov equations Eqs.(3.3.19,20). Then, $P(\varepsilon)$, $Q(\varepsilon)$, $G(\varepsilon)$, $A_c(\varepsilon)$, $K(\varepsilon)$ satisfy Eqs.(3.3.19,20) for all ε. Indeed, pre and postmultiplying Eq.(3.3.19) by $\tilde{N}^T$:

$$
\begin{aligned}
0_{\tilde{n}\times\tilde{n}} &= \tilde{N}^TA_{cl}{}^T(\tilde{N}^{-T}\tilde{N}^T)P\tilde{N} + \tilde{N}^TP(\tilde{N}\tilde{N}^{-1})A_{cl}\tilde{N} + \tilde{N}^TR_{cl}\tilde{N} \\
&= A_{cl}(\varepsilon)^TP(\varepsilon) + P(\varepsilon)A_{cl}(\varepsilon) + R_{cl}(\varepsilon).
\end{aligned}
$$

Hence $P(\varepsilon)$ satisfies Eq.(3.3.19) for all ε. The proof for $Q(\varepsilon)$ is similar.

Theorem 4.3 states that $L(\varepsilon)$ is constant. Expanding $L(\varepsilon)$ to second order and using the matrix differentiation rules, one gets:

$$0 = \varepsilon\left[\mathrm{Tr}(L_G^T G_\varepsilon) + \mathrm{Tr}(L_{A_c}^T A_{c_\varepsilon}) + \mathrm{Tr}(L_K^T K_\varepsilon)\right] +$$
$$+ \varepsilon^2\left[\mathrm{Tr}(L_G^T G_{\varepsilon\varepsilon}) + \mathrm{Tr}(L_{A_c}^T A_{c_{\varepsilon\varepsilon}}) + \mathrm{Tr}(L_K^T K_{\varepsilon\varepsilon}) + \frac{1}{2}\xi_\varepsilon^T L_{\xi\xi}\xi_\varepsilon\right] + 0(\varepsilon^3)$$

where ξ is defined by Eq.(3.3.24) and where $L_{\xi\xi}$ of Eq.(3.3.25) is the Hessian of the Lagrangian as calculated in Sect.3.3.4. ξ_ε is the partial derivative of ξ with respect to ε. The Hessian calculated in Section 3.3.4 is computed assuming that P and Q vary as functions of G, A_c and K such that Eqs.(3.3.19,20) are always satisfied. Such is the case for $P(\varepsilon)$ and $Q(\varepsilon)$, which is the reason why $L_{\xi\xi}$ is used. Expanding to second order, N and N^{-1} become

$$N = I + \varepsilon N_\varepsilon + 0(\varepsilon^3)$$
$$N^{-1} = I - \varepsilon N_\varepsilon + \varepsilon^2 N_\varepsilon^2 + 0(\varepsilon^3)$$

Expanding $G(\varepsilon)$, $A_c(\varepsilon)$ and $K(\varepsilon)$ next, one obtains the following partial derivatives:

$$\begin{aligned} G_\varepsilon &= GN_\varepsilon \\ A_{c_\varepsilon} &= A_c N_\varepsilon - N_\varepsilon A_c \\ K_\varepsilon &= -N_\varepsilon K \\ G_{\varepsilon\varepsilon} &= 0_{m\times n_c} \\ A_{c_{\varepsilon\varepsilon}} &= -N_\varepsilon A_c N_\varepsilon + N_\varepsilon^2 A_c \\ K_{\varepsilon\varepsilon} &= N_\varepsilon^2 K \end{aligned} \tag{4.4.1}$$

$G(\varepsilon)$, $A_c(\varepsilon)$ and $K(\varepsilon)$ still satisfy the architecture constraints. It follows that the matrices G_ε, A_{c_ε} and K_ε are in the matrix subspaces defined by the architecture and one can define a vector ξ_ε using the notation of Definition 3.1 as:

$$\xi_\varepsilon = \begin{bmatrix} G_\varepsilon * E_G \\ A_{c_\varepsilon} * E_{A_c} \\ K_\varepsilon * E_K \end{bmatrix} \tag{4.4.2}$$

Hence, to second order:

$$0 = \varepsilon\left[\mathrm{Tr}L_{A_c}{}^T(A_c N_\varepsilon - N_\varepsilon A_c) + \mathrm{Tr}L_G{}^T GN_\varepsilon - \mathrm{Tr}L_K{}^T N_\varepsilon K\right] +$$
$$+ \varepsilon^2\left[\mathrm{Tr}L_{A_c}{}^T(N_\varepsilon^2 A_c - N_\varepsilon A_c N_\varepsilon) + \mathrm{Tr}L_K{}^T N_\varepsilon^2 K + \frac{1}{2}\xi_\varepsilon^T L_{\xi\xi}\xi_\varepsilon\right] +$$
$$+ 0(\varepsilon^3) \tag{4.4.3}$$

where N_ε is any block diagonal $n_c \times n_c$ matrix of the form: $N_\varepsilon = \sum_{i=1}^{p} \Pi_i^T N_i \Pi_i$. In particular, this implies that the ε term must be zero, or:

$$0_{n_i \times n_i} = \Pi_i \left[L_{A_c}^T A_c - A_c L_{A_c}^T + L_G^T G - K L_K^T \right] \Pi_i^T, \ i = 1, \cdots, p \quad (4.4.4)$$

This actually corresponds to the restriction of the difference Eq.(4.3.9) – Eq.(4.3.10) to its i^{th} diagonal block. Once L_P and L_Q are zero, L_G, L_{A_c} and L_K satisfy Eq.(4.4.2) which regroups $n_1^2 + n_2^2 + \cdots + n_p^2$ equations: the necessary conditions Eqs.(3.3.19-23) are therefore singular.

4.4.2 Minimal Set of Variables, Minimal Set of Equations

Singular problems are very difficult to solve numerically. One would like to reduce the number of variables as well as the number of optimality conditions such that the new system of equations has isolated solutions. The transformation should yield the remaining control parameters as a function of the reduced set of variables and the optimality conditions not retained in the reduced problem should be satisfied as well.

P and Q should be eliminated since they regroup a large number of variables – $\tilde{n}(\tilde{n}+1)$ variables when one enforces the symmetry of P and Q – and since they satisfy Lyapunov equations for which reliable numerical methods are available. One must try to constrain G, A_c and K and eliminate some of the conditions in Eqs.(3.3.21-23). Eq.(4.4.4) will then be used to check that all optimality conditions are met. The following subsections present three different ways of choosing the variables and the optimality conditions to reduce the singularity of the original formulation.

4.4.2.1 MIMO Controller Canonical Form

Proposition 4.6: *The following subset of the optimality conditions*

$$0_{\tilde{n}\times\tilde{n}} = A_{cl}^T P + P A_{cl} + R_{cl}$$

$$0_{\tilde{n}\times\tilde{n}} = A_{cl} Q + Q A_{cl}^T + V_{cl}$$

$$0_{n_i\times 1} = M_{ii}\, \pi_j^{n_i T}, \qquad j = n_i - l_i + 1, \cdots, n_i$$

$$0_{1\times n_i} = \pi_j^m L_G \Pi_i^T, \qquad j \in \mathcal{U}_i$$

is equivalent to the full set of conditions Eqs.(3.3.19-23) *provided that each subcontroller is in Controller Canonical Form,*

$A_c = \text{blockdiag}(A_1, A_2, \cdots, A_p)$,

$A_i = \begin{bmatrix} [A_{11} \; A_{12}] \\ I_{r_i} \; 0_{r_i\times l_i} \end{bmatrix}$, $A_{11} \in \mathbb{R}^{l_i\times r_i}$, $A_{12} \in \mathbb{R}^{l_i\times l_i}$, $r_i = n_i - l_i$

l_i = *number of sensors associated with subcontroller* i,

$K_{i_j} = \pi_j^{n_i T}$, n_i *by* 1 *column vector with all elements zero except the* j^{th} *which is* 1, K_{i_j} = *filter gain of the* j^{th} *sensor associated with the* i^{th} *subcontroller.*

■

Proof: before the main proof is derived, the structure of the i^{th} subcontroller will be detailed. Regrouping the sensor signals received by the i^{th} subcontroller into a l_i dimensional vector y_i and, similarly, regrouping the corresponding filter gains K_{ij} into the matrix K_i, the dynamics of the compensator become:

$$\dot{X}_i = A_i X_i + K_i y_i, \; K_i \in \mathbb{R}^{n_i\times l_i}, \; y_i \in \mathbb{R}^{l_i\times 1}$$

The structure of the compensator retained in Proposition 4.6 corresponds to the MIMO Controller Canonical Form, [Kai80], for the pair (A_i, K_i). K_i and A_i become:

$$K_i = \begin{array}{c} \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{array}{l} l_i \\ r_i, \end{array} \\ l_i \end{array} \quad A_i = \begin{array}{c} \begin{bmatrix} [A_{11} & A_{12}] \\ I & 0 \end{bmatrix} \begin{array}{l} l_i \\ r_i, \end{array} \\ r_i \quad l_i \end{array}$$

where I and 0 are the identity and null matrix of suitable dimension. The necessary conditions retained for the problem are:

Eqs.(3.3.19,20),

Eqs.(3.3.22), complete set

The last l_i columns of Eq.(3.3.21)

We must show that:

The $n_i - l_i$ first columns of Eq.(3.3.21) are satisfied

Eq.(3.3.23) is satisfied

Expanding Eq.(4.4.4),

$$0_{n_i \times n_i} = M_{ii}^T A_i - A_i M_{ii}^T + \Pi_i G^T L_G \Pi_i^T - \Pi_i K L_K^T \Pi_i^T \tag{4.4.5}$$

Since Eq.(3.3.22) is entirely satisfied and since G satisfies the architecture constraints the term $\Pi_i L_G^T G \Pi_i^T$ drops out. The last l_i columns of Eq.(3.3.21) being satisfied, M_{ii} is:

$$M_{ii} = [M\ 0_{n_i \times l_i}],\ M \in \mathbb{R}^{n_i \times r_i}$$

Regrouping the $L_{K_{ij}}$ in a matrix $L_{K_i} \in \mathbb{R}^{n_i \times l_i}$ like the K_{ij} were regrouped in K_i, the term $\Pi_i K L_K^T \Pi_i^T$ becomes simply $K_i L_{K_i}^T$: this follows from the fact that terms involving elements of L_K not associated with sensors used by loop i drop out since they get multiplied by zero entries of K. Expanding Eq.(4.4.5) in block form:

$$0_{n_i \times n_i} = \begin{bmatrix} M^T \\ 0_{l_i \times n_i} \end{bmatrix} A_i - \begin{bmatrix} A_{11} & A_{12} \\ I_{r_i} & 0_{r_i \times l_i} \end{bmatrix} \begin{bmatrix} M^T \\ 0_{l_i \times n_i} \end{bmatrix} - \begin{bmatrix} L_{K_i}^T \\ 0_{r_i \times n_i} \end{bmatrix}$$

$$= \begin{bmatrix} M^T A_i \\ 0_{l_i \times n_i} \end{bmatrix} - \begin{bmatrix} A_{11} M^T \\ M^T \end{bmatrix} - \begin{bmatrix} L_{K_i}^T \\ 0_{r_i \times n_i} \end{bmatrix} \tag{4.4.6}$$

Assume $l_i \geq r_i$. Then, the last r_i rows of Eq.(4.4.6) directly yield $M = 0_{n_i \times r_i}$. Looking back at Eq.(4.4.6), one gets that $L_{K_i} = 0_{n_i \times l_i}$.

If $l_i < r_i$, then define $d_i = r_i - l_i$ and partition M as $M = [M_1\ M_2]$, $M_1 \in \mathbb{R}^{n_i \times l_i}$, $M_2 \in \mathbb{R}^{n_i \times d_i}$. The last r_i rows of Eq.(4.4.6) become:

$$0_{r_i \times n_i} = \begin{bmatrix} M_2^T A_i \\ 0_{l_i \times n_i} \end{bmatrix} - \begin{bmatrix} M_1^T \\ M_2^T \end{bmatrix} \tag{4.4.7}$$

Notice that the two matrices in Eq.(4.4.7) do not share the same block partitioning: the first one has a $d_i \times n_i$ and a $l_i \times n_i$ block, the second matrix has a $l_i \times n_i$ and a $d_i \times n_i$ block. If $l_i \geq d_i$, then M_2 is zero. When $l_i < d_i$, M_2 has the form:

$$M_2 = [m_1\, m_2 \cdots m_j\, 0_{n_i \times l_i}],\ j = d_i - l_i,\ m_k \in \mathbb{R}^{n_i}, \text{ column vector.}$$

The l_i+1^{th} row of Eq.(4.4.7), counted from the bottom, is simply $m_j^T = 0_{1 \times n_i}$. Hence, M_2 becomes:

$$M_2 = [m_1\, m_2 \cdots m_{j-1}\, 0_{n_i \times l_i+1}].$$

Repeating the process, one can show that M_2 is zero. Consequently M_1 is zero, and using Eq.(4.4.6), L_{K_i} is zero. □

Subcontroller i being in MIMO Controller Canonical Form, n_i^2 parameters are fixed: they correspond to the $n_i - l_i$ last rows of the matrix A_i as well as all the entries of K_i. Similarly, the number of equations have been reduced: Eq.(3.3.23) has been completely relaxed and Eq.(3.3.21) has been reduced. These equations are, however, satisfied as soon as the remaining necessary conditions are met. Notice, however, that, whereas the last l_i columns of M_{ii} are set to zero, some of the first columns of M_{ii} are given as linear combinations of the last columns, as seen in the proof. The coefficients of the linear combinations are elements of the matrix A_i. The level of accuracy in setting the first columns of M_{ii} to zero will, thus, correspond to the level of accuracy to which the last columns are set to zero, multiplied by a gain which is of the order of the maximum element of A_i. The level of accuracy at which L_K is set to zero will also be an amplified version of the accuracy imposed on the last columns of M_{ii}. Hence, the values of A_i will be very influential in controlling the overall numerical accuracy of the solution using the MIMO Controller Canonical Form. A second shortcoming, but less critical, is that it is not possible to accommodate loops with more sensors than internal states. Indeed, if they are more sensors than they are internal states in the compensator, the matrix K_i will have more columns than rows and it will not be possible to put it in the canonical form specified in Proposition 4.6.

4.4.2.2 MIMO Observer Canonical Form

Proposition 4.7: *The following subset of the optimality conditions*

$$0_{\tilde{n}\times\tilde{n}} = A_{cl}{}^T P + PA_{cl} + R_{cl}$$
$$0_{\tilde{n}\times\tilde{n}} = A_{cl}Q + QA_{cl}{}^T + V_{cl}$$
$$0_{1\times n_i} = \pi_j^{n_i} M_{ii}, \qquad j = n_i - m_i + 1, \cdots, n_i$$
$$0_{n_i\times 1} = \Pi_i L_K \pi_j^{1\,T}, \qquad j \in \mathcal{Y}_i$$

is equivalent to the full set of conditions Eqs.(3.3.19-23) *provided that each subcontroller is in Observer Canonical Form,*

$A_c = \text{blockdiag}(A_1, A_2, \cdots, A_p)$,

$A_i = \begin{bmatrix} A_{11} & I_{r_i} \\ A_{21} & 0_{m_i\times r_i} \end{bmatrix}$, $A_{11} \in \mathbb{R}^{r_i\times m_i}$, $A_{21} \in \mathbb{R}^{m_i\times m_i}$, $r_i = n_i - m_i$

$m_i =$ *number of actuators associated with subcontroller* i,

$G_{i_j} = \pi_j^{n_i}$, 1 *by* n_i *row vector with all elements zero except the* j*th* *which is* 1,
$G_{i_j} =$ *control gain of the* j*th* *actuator associated with the* i*th* *subcontroller.*

■

Proof: the proof is the dual of the proof of Proposition 4.6. If we regroup all the actuator gains in one matrix G_i, then (G_i, A_i) is in MIMO Observer Canonical Form:

$$G_i = \begin{array}{c} \begin{bmatrix} I & 0 \end{bmatrix} m_i \\ \;m_i \;\; r_i \end{array}, \quad A_i = \begin{array}{cc} \begin{bmatrix} A_{11} & I \\ A_{21} & 0 \end{bmatrix} & \begin{array}{c} r_i \\ m_i \end{array} \\ \;m_i \quad r_i & \end{array}$$

Eq.(3.3.21) is only partially enforced. M_{ii} has the form:

$$M_{ii} = \begin{array}{cc} \begin{bmatrix} M_1 & M_2 \\ 0 & 0 \end{bmatrix} & \begin{array}{c} n_i - m_i \\ m_i \end{array} \end{array}$$

Again, n_i^2 parameters and n_i^2 equations have been eliminated. The parameters are the n_i-m_i last columns of A_i and all the entries of G_i, and the equations are the first n_i-m_i rows of Eq.(3.3.21) and the entire set of Eq.(3.3.22).

The shortcomings of the Observer Canonical Form are similar, or dual, to those of the Controller Canonical Form. The main one is the sensitivity of the accuracy which cannot be controlled, and the second one is the fact that one cannot have subcontrollers with more actuators than internal states.

4.4.2.3 Modal Form

Proposition 4.8: *The following subset of the optimality conditions*

$$
\begin{aligned}
0_{\tilde{n}\times\tilde{n}} &= A_{cl}^{T}P + PA_{cl} + R_{cl} \\
0_{\tilde{n}\times\tilde{n}} &= A_{cl}Q + QA_{cl}^{T} + V_{cl} \\
0_{1\times n_i} &= \pi^{m}_{j}L_{G}\Pi_i^{T}, && j \in \mathcal{U}_i \\
0_{n_i\times 1} &= \Pi_i L_K \pi^{l\,T}_{j}, && j \in \mathcal{Y}_i \\
0 &= M_{ii}(k_1,k_2),\ k_2\ \textit{even},\ k_1 = k_2-1,\ k_2, \\
0 &= M_{ii}(n_i,n_i)\ \textit{if}\ n_i\ \textit{is odd}
\end{aligned}
$$

is equivalent to the full set of conditions Eqs.(3.3.19-23) *provided that each subcontroller is in "real modal form"*:

$$
A_i = \mathrm{blockdiag}(A^i_1, A^i_2, \cdots, A^i_r, a^i_{r+1}),
$$

$$
A^i_j \in \mathbb{R}^{2\times 2},\ a^i_{r+1} \in \mathbb{R}\ \textit{if}\ n_i\ \textit{odd},
$$

$$
A^i_j = \begin{bmatrix} a^i_{j1} & a^i_{j2} \\ 1 & 0 \end{bmatrix} \qquad \blacksquare
$$

Proof: Eqs.(3.3.19,20,22,23) are all satisfied. A_c is almost in modal form: it is in real modal form if each 2×2 blocks have complex conjugate eigenvalues. Such a representation will be called the "real modal form" of the compensator. Each A_i has the form:

$$A_i = \begin{bmatrix} \begin{bmatrix} a^i{}_{11} & a^i{}_{12} \\ 1 & 0 \end{bmatrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} \cdots \\ \cdots \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{bmatrix} a^i{}_{21} & a^i{}_{22} \\ 1 & 0 \end{bmatrix} & & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \vdots \quad \vdots & & \ddots & & \vdots \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & & \begin{bmatrix} a^i{}_{r1} & a^i{}_{r2} \\ 1 & 0 \end{bmatrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \end{matrix} & & \begin{matrix} 0 & 0 \end{matrix} & a^i{}_{r+1} \end{bmatrix}$$

Repeating the partitioning of A_i on M_{ii}, one gets 2×2 blocks along with 2×1, 1×2, and a 1×1 block if n_i is odd:

$$M_{ii} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1r} & X_{1r+1} \\ X_{21} & X_{22} & \cdots & X_{2r} & X_{2r+1} \\ \vdots & \vdots & & \vdots & \vdots \\ X_{r1} & X_{r2} & \cdots & X_{rr} & X_{rr+1} \\ X_{r+11} & X_{r+12} & \cdots & X_{r+11} & X_{r+1r+1} \end{bmatrix},$$

Expanding Eq.(4.4.5) in block form, the off-diagonal block equations become:

$$0 = X_{kj}{}^T A^i{}_k - A^i{}_j X_{kj}{}^T, \tag{4.4.8}$$

where 0 is 2×2, 2×1 or 1×2 depending on the block being considered. Assuming that no two blocks $A^i{}_j$ and $A^i{}_k$ have the same eigenvalues, Proposition 4.1 implies that $X_{kj} = 0$. Looking now at a diagonal block element, the conditions satisfied by M_{ii} state that if $X_{kk} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}$, then $x_{12} = 0$, $x_{22} = 0$. If $k = r+1$, they state that $X_{kk} = 0$. The k^{th} diagonal block of Eq.(4.4.5) becomes:

$$0 = X_{kk}{}^T A^i{}_k - A^i{}_k X_{kk}{}^T$$

If $k = r+1$, $X_{kk} = 0$. For $i \neq r+1$:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_{12} & a_{22} \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} a_{12} & a_{22} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{11} & x_{21} \\ 0 & 0 \end{bmatrix}$$

Looking at the last row immediately yields $x_{11} = 0$ and $x_{21} = 0$. Hence, as long as no two block A_i and A_j share a common eigenvalue, the matrix M_{ii} is identically zero. All the optimality conditions are, therefore, satisfied. □

The variables describing A_i correspond to the ones used in [Ly85]. Their number is n_i. The set of variables is not minimal: it is indeed possible to scale G and K without modifying the cost or changing the structure of A_c. One must fix the scaling in order to have a unique modal realization. This operation does not lead to a simple elimination of some elements in G and K and it is not useful at this stage to go further. The modal description reduces the number of degrees of freedom that exist to change the realization of one subcontroller from n_i^2, for the i^{th} subcontroller, down to n_i.

Eq.(4.4.8) is obtained from Eq.(4.4.5) by assuming that L_G and L_K are equal to zero, or similarly, by assuming that Eqs.(3.3.22,23) are satisfied. Of course, one cannot satisfy these equations with infinite accuracy and in practice L_G and L_K will not be exactly zero. Hence the left hand side of Eq.(4.4.8) will not be exactly zero but equal to a residual given by Eq.(4.4.5) and which is $\Pi_i(KL_K{}^T - G^TL_G)\Pi_i{}^T$. The accuracy of the error on off-diagonal block elements of M_{ii} consequently corresponds to the accuracy at which Eqs.(3.3.22,23) are satisfied, times a gain which depends on the magnitudes of G and K, times a gain corresponding to the inverse transformation that yields X_{jk} in Eq.(4.4.8). The gain corresponding to the invertion of Eq.(4.4.8) can be very important if the eigenvalues inside one compensator are small, and if their difference is small as well. When one eliminates the redundant equations, the only means for controlling the accuracy of the equations that have been eliminated is to tighten the error on Eqs.(3.3.22,23), but such a process may be unsuccessful if the gains mentioned above are really large.

4.4.3 Singularity of the Hessian and Existence of Saddle Points

First order necessary conditions only show that a point is stationary: the variation of the Lagrangian with respect to the problem variables will be zero to first order at that point. One must look at the second order conditions to conclude on the nature of the stationary point. In this section, we will first show that the Hessian is singular if one does not use a minimal set of variables. We will also show that some solution points can be saddle points. This last characteristic is linked to the controllability and observability of the controller.

4.4.3.1 Singularity of the Hessian

The first order term in ε in Eq.(4.4.3) is zero at a stationary point. Hence, Eq.(4.4.3) becomes:

$$0 = \mathrm{Tr}L_{A_c}^{T}(N_{\varepsilon}^{2}A_c - N_{\varepsilon}A_cN_{\varepsilon}) - \mathrm{Tr}N_{\varepsilon}^{2}KL_{K}^{T} + \frac{1}{2}\,\xi_{\varepsilon}^{T}L_{\xi\xi}\xi_{\varepsilon}$$

where N_{ε} and ξ_{ε} are given in Eqs.(4.4.1,2). N_{ε} is a block diagonal matrix similar to A_c. Hence, the matrix $N_{\varepsilon}^{2}A_c - N_{\varepsilon}A_cN_{\varepsilon}$ shares the same block diagonal structure and constitutes, therefore, an admissible perturbation for A_c. Thus $\mathrm{Tr}L_{A_c}^{T}(N_{\varepsilon}^{2}A_c - N_{\varepsilon}A_cN_{\varepsilon})$ is zero. Similar argument shows that $\mathrm{Tr}N_{\varepsilon}^{2}KL_{K}^{T}$ is zero. At a stationary point, $L_{\xi\xi}$ satisfies:

$$0 = \xi_{\varepsilon}^{T}L_{\xi\xi}\xi_{\varepsilon},$$

where ξ_{ε} is defined by Eq.(4.4.2). If the stationary point is a minimum, this in turn implies that:

$$0 = L_{\xi\xi}\xi_{\varepsilon}, \tag{4.4.9}$$

Every ξ_{ε} obtained by changing the realization of the compensator following the transformation of Theorem 4.3 is in the nullspace of $L_{\xi\xi}$. In order to find a lower bound on the dimension of $\mathrm{Ker}(L_{\xi\xi})$, one can count the number of linearly independent such vectors ξ_{ε} that can be generated using Eq.(4.4.2). The next theorem shows that they are, in fact, all linearly independent.

Theorem 4.4: *Let the compensator* (G, A_c, K) *be a solution to the optimal fixed architecture control problem (Problem* 3.1*). The Hessian* $L_{\xi\xi}$ *is singular and its nullspace has a dimension greater or equal to* $n_1^2 + n_2^2 + \cdots + n_p^2$ ∎

Proof: consider a transformation $N(\varepsilon)$ as defined in Sect.4.4.1 such that:

$$0_{m\times n_c} = GN_{\varepsilon} \tag{4.4.10}$$
$$0_{n_c\times n_c} = -N_{\varepsilon}A_c + A_cN_{\varepsilon} \tag{4.4.11}$$
$$0_{n_c\times l} = -N_{\varepsilon}K \tag{4.4.12}$$

N_{ε} is block diagonal. For simplicity sake, it is assumed that $p = 1$ and that there is only one processor in the feedback loop. If there is more than one processor, one can consider one compensator at a time and augment the plant with the remaining loops. The proof can then be worked out for each compensator successively.

Let (λ,v) be a pair of eigenvalue and right eigenvector of A_c corresponding to an observable mode. Eq.(4.4.11) implies that $N_\varepsilon v$ is an eigenvector of A_c as long as $N_\varepsilon v$ is not zero. Postmultiplying Eq.(4.4.10) by v would imply that $GN_\varepsilon v = 0$ and that the mode $(\lambda,N_\varepsilon v)$ is unobservable, which is contrary to the hypotheses. Hence, $N_\varepsilon v = 0$ for every right eigenvector of A_c. If λ corresponds to a Jordan block of size r for A_c, consider v such that $(v, A_c v,\cdots,A_c^{r-1}v)$ spans the invariant space of A_c associated with λ. Eq.(4.4.11) implies that $(N_\varepsilon v, N_\varepsilon A_c v,\cdots,N_\varepsilon A_c^{r-1}v)$ is an invariant subspace of A_c associated with λ and Eq.(4.4.10) implies that this subspace is unobservable: $N_\varepsilon v$, $N_\varepsilon A_c v,\cdots,N_\varepsilon A_c^{r-1}v$ must therefore be zero. If λ corresponds to a controllable mode, then, calling w the left eigenvector associated with λ and using Eq.(4.4.12), $wN_\varepsilon = 0$. If λ corresponds to an irreducible block of size r for A_c, there exists w such that w, $wA_c,\cdots,wA_c^{r-1}$ spans the invariant subspace of A_c associated with λ such that wN_ε, $wA_cN_\varepsilon,\cdots,wA_c^{r-1}N_\varepsilon$ are all zero. Eq.(4.4.11) implies that N_ε and A_c share the same invariant subspaces. It has just been shown that if a mode is either controllable or observable, N_ε is zero on that particular subspace. Therefore, if each mode is either controllable or observable, N_ε is zero. Hence, the vectors ξ_ε's form an independent family of vectors. We have to consider now the unobservable and uncontrollable modes. A nonzero transformation N such that N_ε satisfies Eqs.(4.4.10-12) has the same invariant subspaces as A_c (Eq.(4.4.11)). Satisfying Eq.(4.4.10) and Eq.(4.4.12) implies that N_ε corresponds to scaling the uncontrollable and unobservable states. This means that the corresponding ξ_ε vector will be a linear combination of the remaining ξ_ε's. However, the transformation that corresponds to changing the eigenvalues of the uncontrollable and unobservable modes is one that leaves the cost unchanged. This transformation is independent from the others and one has a new independent vector to complete the nullspace of $L_{\xi\xi}$. Finally, counting the number of different possible linearly independent transformations N_ε, one finds that the dimension of the nullspace of $L_{\xi\xi}$ is $n_1^2 + n_2^2 + \cdots + n_p^2$.

□

Theorem 4.3 guarantees that the cost is invariant to all orders when the realization of the compensator is changed, but its transfer function is conserved. Hence, the problem is under-determined. The singularity of the Hessian uncovered in Theorem 4.4 corresponds only to the freedom of choosing one representation of te compensator transfer function. The problem is truly singular, or admits a critical point, when the Hessian has an extra zero eigenvalue that does not correspond to a change in the compensator realization.

4.4.3.2 Existence of Saddle Points

We consider in the following that there is a single compensator in the feedback loop and that G, A_c and K are block partitioned as follows:

$$G = [G_1\, G_2],\ A_c = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},\ K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}.$$

The triplet of matrices (G, A_c, K) can be expanded on a basis of matrices (Definition 3.1) and is completely defined by a column vector ξ containing the free entries of G, A_c and K. Assume that ξ is formed in the following fashion:

$$\xi = \begin{bmatrix} a_{11} \\ k_1 \\ g_1 \\ a_{21} \\ k_2 \\ a_{12} \\ g_2 \\ a_{22} \end{bmatrix} \tag{4.4.13}$$

where a_{11} is a column vector containing the free entries in A_{11}, k_1 is a column vector containing the free entries in K_1, etc. The Lagrangian can be partitioned accordingly:

$$L_{\xi\xi} = \begin{bmatrix}
L_{a_{11}a_{11}} & L_{a_{11}k_1} & L_{a_{11}g_1} & L_{a_{11}a_{21}} & L_{a_{11}k_2} & L_{a_{11}a_{12}} & L_{a_{11}g_2} & L_{a_{11}a_{22}} \\
L_{k_1a_{11}} & L_{k_1k_1} & L_{k_1g_1} & L_{k_1a_{21}} & L_{k_1k_2} & L_{k_1a_{12}} & L_{k_1g_2} & L_{k_1a_{22}} \\
L_{g_1a_{11}} & L_{g_1k_1} & L_{g_1g_1} & L_{g_1a_{21}} & L_{g_1k_2} & L_{g_1a_{12}} & L_{g_1g_2} & L_{g_1a_{22}} \\
L_{a_{21}a_{11}} & L_{a_{21}k_1} & L_{a_{21}g_1} & L_{a_{21}a_{21}} & L_{a_{21}k_2} & L_{a_{21}a_{12}} & L_{a_{21}g_2} & L_{a_{21}a_{22}} \\
L_{k_2a_{11}} & L_{k_2k_1} & L_{k_2g_1} & L_{k_2a_{21}} & L_{k_2k_2} & L_{k_2a_{12}} & L_{k_2g_2} & L_{k_2a_{22}} \\
L_{a_{12}a_{11}} & L_{a_{12}k_1} & L_{a_{12}g_1} & L_{a_{12}a_{21}} & L_{a_{12}k_2} & L_{a_{12}a_{12}} & L_{a_{12}g_2} & L_{a_{12}a_{22}} \\
L_{g_2a_{11}} & L_{g_2k_1} & L_{g_2g_1} & L_{g_2a_{21}} & L_{g_2k_2} & L_{g_2a_{12}} & L_{g_2g_2} & L_{g_2a_{22}} \\
L_{a_{22}a_{11}} & L_{a_{22}k_1} & L_{a_{22}g_1} & L_{a_{22}a_{21}} & L_{a_{22}k_2} & L_{a_{22}a_{12}} & L_{a_{22}g_2} & L_{a_{22}a_{22}}
\end{bmatrix}$$

Assume now that $A_{21} = 0$ and $K_2 = 0$. The modes corresponding to A_{22} are uncontrollable from K. The compensator matrices have the form:

$$G = [G_1\ G_2]\,,\ A_c = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},\ K = \begin{bmatrix} K_1 \\ 0 \end{bmatrix} \tag{4.4.14}$$

The following equations hold in that case:

$$\begin{array}{lll}
0 = L_{a_{11}a_{12}} & 0 = L_{a_{11}g_2} & 0 = L_{a_{11}a_{22}} \\
0 = L_{k_1a_{12}} & 0 = L_{k_1g_2} & 0 = L_{k_1a_{22}} \\
0 = L_{g_1a_{12}} & 0 = L_{g_1g_2} & 0 = L_{g_1a_{22}} \\
0 = L_{a_{12}a_{12}} & 0 = L_{a_{12}g_2} & 0 = L_{a_{12}a_{22}} \\
0 = L_{g_2a_{12}} & 0 = L_{g_2g_2} & 0 = L_{g_2a_{22}} \\
0 = L_{a_{22}a_{12}} & 0 = L_{a_{22}g_2} & 0 = L_{a_{22}a_{22}}
\end{array}$$

The proof is deferred to Appendix B. The proof consists of performing the derivations of Eqs.(3.3.24-33) in block form. The Hessian becomes:

$$L_{\xi\xi} = \begin{bmatrix}
L_{a_{11}a_{11}} & L_{a_{11}k_1} & L_{a_{11}g_1} & L_{a_{11}a_{21}} & L_{a_{11}k_2} & 0 & 0 & 0 \\
L_{k_1a_{11}} & L_{k_1k_1} & L_{k_1g_1} & L_{k_1a_{21}} & L_{k_1k_2} & 0 & 0 & 0 \\
L_{g_1a_{11}} & L_{g_1k_1} & L_{g_1g_1} & L_{g_1a_{21}} & L_{g_1k_2} & 0 & 0 & 0 \\
L_{a_{21}a_{11}} & L_{a_{21}k_1} & L_{a_{21}g_1} & L_{a_{21}a_{21}} & L_{a_{21}k_2} & L_{a_{21}a_{12}} & L_{a_{21}g_2} & L_{a_{21}a_{22}} \\
L_{k_2a_{11}} & L_{k_2k_1} & L_{k_2g_1} & L_{k_2a_{21}} & L_{k_2k_2} & L_{k_2a_{12}} & L_{k_2g_2} & L_{k_2a_{22}} \\
0 & 0 & 0 & L_{a_{12}a_{21}} & L_{a_{12}k_2} & 0 & 0 & 0 \\
0 & 0 & 0 & L_{g_2a_{21}} & L_{g_2k_2} & 0 & 0 & 0 \\
0 & 0 & 0 & L_{a_{22}a_{21}} & L_{a_{22}k_2} & 0 & 0 & 0
\end{bmatrix}$$

Such a matrix is not definite. Indeed, $L_{\xi\xi}$ can be written as:

$$L_{\xi\xi} = \begin{bmatrix} L_{11} & L_{12} & 0 \\ L_{21} & L_{22} & L_{23} \\ 0 & L_{32} & 0 \end{bmatrix},\ L_{12} = L_{21}^T,\ L_{23} = L_{32}^T \tag{4.4.15}$$

For any $\xi = \begin{bmatrix} 0 \\ \xi_2 \\ \alpha\xi_3 \end{bmatrix}$, $\xi^T L_{\xi\xi}\xi = \xi_2^T L_{22}\xi_2 + 2\alpha\xi_2^T L_{23}\xi_3$. Assuming that L_{23} is not identically zero, there exists ξ_2, ξ_3 and α such that $\xi^T L_{\xi\xi}\xi < 0$. There are also ξ_2, ξ_3 and α such that $\xi^T L_{\xi\xi}\xi > 0$. If L_{11} is positive, the compensator will correspond to a saddle point for the cost. □

Assume now that $G_2 = 0$ and $A_{12} = 0$. The modes corresponding to A_{22} are now unobservable from G. The compensator matrices have the form:

$$G = [G_1\ 0]\ ,\ A_c = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}\ ,\ K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \tag{4.4.16}$$

The following equations hold in that case:

$$\begin{array}{lll}
0 = L_{a_{11}a_{21}} & 0 = L_{a_{11}k_2} & 0 = L_{a_{11}a_{22}} \\
0 = L_{k_1a_{21}} & 0 = L_{k_1k_2} & 0 = L_{k_1a_{22}} \\
0 = L_{g_1a_{21}} & 0 = L_{g_1k_2} & 0 = L_{g_1a_{22}} \\
0 = L_{a_{21}a_{21}} & 0 = L_{a_{21}k_2} & 0 = L_{a_{21}a_{22}} \\
0 = L_{k_2a_{21}} & 0 = L_{k_2k_2} & 0 = L_{k_2a_{22}} \\
0 = L_{a_{22}a_{21}} & 0 = L_{a_{22}k_2} & 0 = L_{a_{22}a_{22}}
\end{array}$$

Reordering ξ as:

$$\xi = \begin{bmatrix} a_{11} \\ k_1 \\ g_1 \\ a_{12} \\ g_2 \\ a_{21} \\ k_2 \\ a_{22} \end{bmatrix} \tag{4.4.17}$$

$L_{\xi\xi}$ becomes:

$$L_{\xi\xi} = \begin{bmatrix}
L_{a_{11}a_{11}} & L_{a_{11}k_1} & L_{a_{11}g_1} & L_{a_{11}a_{12}} & L_{a_{11}g_2} & 0 & 0 & 0 \\
L_{k_1a_{11}} & L_{k_1k_1} & L_{k_1g_1} & L_{k_1a_{12}} & L_{k_1g_2} & 0 & 0 & 0 \\
L_{g_1a_{11}} & L_{g_1k_1} & L_{g_1g_1} & L_{g_1a_{12}} & L_{g_1g_2} & 0 & 0 & 0 \\
L_{a_{12}a_{11}} & L_{a_{12}k_1} & L_{a_{12}g_1} & L_{a_{12}a_{12}} & L_{a_{12}g_2} & L_{a_{12}a_{21}} & L_{a_{12}k_2} & L_{a_{12}a_{22}} \\
L_{g_2a_{11}} & L_{g_2k_1} & L_{g_2g_1} & L_{g_2a_{12}} & L_{g_2g_2} & L_{g_2a_{12}} & L_{g_2k_2} & L_{g_2a_{22}} \\
0 & 0 & 0 & L_{a_{21}a_{12}} & L_{a_{21}g_2} & 0 & 0 & 0 \\
0 & 0 & 0 & L_{k_2a_{12}} & L_{k_2g_2} & 0 & 0 & 0 \\
0 & 0 & 0 & L_{a_{22}a_{12}} & L_{a_{22}g_2} & 0 & 0 & 0
\end{bmatrix}$$

Such a matrix is, again, nondefinite.

Theorem 4.5: *Any stationary point such that the compensator (or one of the compensators) has an unobservable or uncontrollable mode is a saddle point. The Hessian* $L_{\xi\xi}$ *possesses in that case a nonpositive eigenvalue. If the eigenvalue is zero, its associated eigenvector does not belong to the part of the nullspace that corresponds to the changing of realization of the compensators. Such saddle points always exist if there exists a stabilizing solution for the fixed architecture problem obtained by keeping the same control architecture and by reducing the order of the compensators.* ∎

Proof: the nondefiniteness of $L_{\xi\xi}$ has been shown before. The second part of the theorem consists of proving that if the compensator has an uncontrollable or an unobservable mode that produces a singular eigenvalue in the Hessian, its associated eigenvector does not correspond to a change in compensator realization. Consider the case where A_{22} is uncontrollable. As seen from Eq.(4.4.15), the new eigenvector has the form:

$$\xi = \begin{bmatrix} 0 \\ \xi_2 \\ \alpha\xi_3 \end{bmatrix}$$

Hence, following Eq.(4.4.13), ξ leaves A_{11}, K_1 and G_1 invariant. A similarity transform N satisfying the same properties must share the block triangular partition of A_c, Eq.(4.4.14):

$$N = \begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{22} \end{bmatrix}$$

The first order variation of A_c and K will be such that A_{21} as well as K_2 remain invariant. This is equivalent to ξ_3 being zero. The form of $L_{\xi\xi}$ in Eq.(4.4.15) guarantees, however, that there is a negative, or a singular, eigenvalue with a corresponding eigenvector such that ξ_3 is not zero. Thus, the extra zero eigenvalue has an eigenvector which does not correspond to a change in the realization of the compensator.

Finally, assume that there is only one compensator ($p = 1$) and assume that we are trying to solve the reduced order problem for the system (C, A, B),

weigthing matrices R, V, R_c, V_c, with a compensator of order $n_c = n_{c0}$. Further assume that (G_1, A_{c1}, K_1) is the realization of a compensator that solves the reduced order LQG problem for the same plant and for the same weigthing matrices but with $n_c = n_{c0}-1$ and let P_1 and Q_1 be the correponding solution to the associated Lyapunov equations, $P_1, Q_1 \in \mathbb{R}^{\tilde{n}_1 \times \tilde{n}_1}$, $\tilde{n}_1 = n + n_{c0} - 1$.
(G_1, A_{c1}, K_1, P_1, Q_1) satisfy the optimality conditions Eqs.(3.3.19-23) with $n_c = n_{c0}-1$. Consider now the compensator given by:

$$G = [G_1\ 0],\ G \in \mathbb{R}^{m \times n_c}$$

$$A_c = \begin{bmatrix} A_{c1} & 0 \\ 0 & -1 \end{bmatrix},\ A_c \in \mathbb{R}^{n_c \times n_c}$$

$$K = \begin{bmatrix} K_1 \\ 0 \end{bmatrix},\ K \in \mathbb{R}^{n_c \times l}.$$

Defining the positive semidefinite matrices P and Q as:

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & 0 \end{bmatrix},\ P \in \mathbb{R}^{\tilde{n} \times \tilde{n}},\ \tilde{n} = n + n_{c0}$$

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & 0 \end{bmatrix},\ Q \in \mathbb{R}^{\tilde{n} \times \tilde{n}},\ \tilde{n} = n + n_{c0}$$

It is easy to verify that (G, A_c, K, P, Q) satisfy the optimality conditions Eqs.(3.3.19-23) for $n_c = n_{c0}$. The compensator has an uncontrollable or unobservable pole and is therefore a saddle point for $n_c = n_{c0}$. Any solution to the reduced order control problem where $n_c = n_{c0}$ obtained by appending an uncontrollable and unobservable mode to a controller that solves the reduced order control problem where $n_c = n_{c0}-1$ will be a stationary point which is not a local minimum but a saddle point. All the reduced order compensator solutions can, in particular, be extended by adding unobservable and uncontrollable poles to become full order compensator. These solutions are stationary point of the LQG problem, but they are saddle points.

4.5 CONCLUSION

It has been shown in this chapter that it is not possible to reduce the complexity of the problem by trying to write the optimality conditions in a more structured way. Two types of coupling occur when one looks at the fixed architecture problem: the first one is a coupling between the filtering and the control problems. The separation principle does not hold anymore. Such is already the case when one looks at the OPE problem. The structured problem consists then of solving two modified Riccati equations coupled by two Lyapunov equations. A new source of coupling occurs when one constrains the architecture and splits the feedback into p separate processors: Each subcontroller being optimal for the system consisting of the original plant and the remaining loops closed, the gains and dynamics of each compensator will depend on the gains and dynamics of the remaining feedback loops. It is not possible, in that case, to decompose the problem into a series of smaller problems that have to be solved sequentially. The structured problem consists of two modified Riccati equations coupled through four Lyapunov equations which yield the optimal coupling between the control and the filter problem as well as the optimal coordination between the various subcontrollers. The gains and dynamics of the compensators cannot be eliminated from the problem in a simple fashion.

The study of the optimality conditions derived in Chapter 3 have shown that the problem is singular. The cost depends only on the compensator transfer function, independent of the state space realization chosen to represent it. Any set of gains obtained by a similarity transformation of an optimal solution will, therefore, satisfy the optimality conditions. It is possible to find a minimum set of equations and a minimum set of variables that will remove this singularity. However, such an operation may generate numerical problems, since it is not possible to control directly the accuracy of the superfluous equations that have been eliminated. Finally, it was shown that if an optimal compensator has an unobservable or uncontrollable mode, it will be a saddle point for the cost. Optimal constrained solutions become, therefore, saddle points when they are expanded to problems with less stringent constraints that allow for compensators of higher orders.

HOMOTOPIC CONTINUATION

5.1 INTRODUCTION

The derivation of the necessary conditions for optimality has a limited value unless one can find a reliable way to solve for them. It provides an analytic expression for the gradient of the cost with respect to the free parameters of the problem. At this point, the use of any unconstrained minimization technique such as steepest descent, conjugate gradient, quasi-Newton or Newton methods should yield solutions corresponding to local minima for the cost. Optimization techniques are discussed in detail in numerous references such as [Lue69, Sca85]. This direct approach has been taken by many for solving reduced order LQG control problems [Bas75, Men75, Ly82, Ly85, Moe85, Cal89]. In [Cal89], the reduced order dynamic compensator problem is written as an optimal decentralized output feedback problem by adding integrators to the plant in order to account for the compensator states. The optimum is sought using a conjugate gradient method. Developed at Stanford University by Uy-Loi Ly, the program "SANDY" [Ly82] is one that has probably evolved as the most user oriented software for designing reduced order compensators. It may now include more constraints such as H_∞ sensitivity constraints and it also addresses some robustness issues in the design. In order to ease the problem of finding an initial stabilizing reduced order compensator, Ly considers a finite horizon H_2 norm. Once a solution is found, the final time is increased in order to obtain the asymptotic compensator. The minimization is carried out using a first order method based on the analytic form of the gradient. In [Gra74], Graupe includes the norm of the cost gradient with respect to the plant parameters in order to obtain an H_2 design less sensitive to modeling error. The problem is a different constrained H_2 design, but the derivation of the necessary condition for optimality is quite similar to the derivations shown in Chapter 3. The numerical procedure indicated for solving the problem is a steepest descent algorithm.

The efficiency of direct minimization procedures is very problem dependent. It is usually true that most algorithms need a good starting point in order to converge. This implies that one must design a near optimal initial stabilizing multiloop compensator with the appropriate architecture. This task is not simpler than guessing the optimal solution itself in many cases. Minimization algorithms have slower rates of convergence when the Hessian is singular at the optimum. One can choose to use reduced or minimal sets of parameters to describe the compensator, [Ly85, Cal89]. However, as indicated in Chapter 4, extreme sensitivity to parameter changes can arise from such a choice. Finally, one still has to worry about the existence of multiple solutions. The poor performance and the difficulty to apply a minimization scheme directly has led to the development of alternate solution techniques.

As indicated in [Hyl84], the reduced order compensator problem retains some structure that is captured by the Optimal Projection Equation formulation. [Gru86] uses an iterative method for solving the OPE similar to that of [Hyl83]. The method is based on successive iterations and reduction of the size of the eigenvalues of the coupling matrix τ. The method can be seen as an improvement of previously developed substitution algorithms such as the ones proposed in [Lev70, Joh70, And71] which are already an adaptation to the reduced order compensator problem of iterative methods for solving the full order compensator problem [Kwa72b]. The coupling matrix is set initially to be the identity matrix which corresponds to the full order LQG problem. The initial solution is therefore easy to obtain and, as some of the eigenvalues of τ go to zero, the matrix becomes a projection operator. The main disadvantage of the technique is that the successive iterations have no guarantee of converging even if a solution exists: the algorithm can be numerically unstable and can, therefore, fail to provide any answer at all. Designs similar to those presented in [Ly85] are reproduced in [Gru86] using the iterative technique. The problem is to optimally control a 7^{th} order plant corresponding to the longitudinal dynamics of an aircraft augmented by the wind gust dynamics using a 4^{th}, a 3^{rd}, and a 1^{st} order compensator. The iterative method appears to converge more slowly than the minimization techniques and to achieve a smaller accuracy, especially when searching for the 1st order compensator. Moreover, the technique cannot be used to synthesize a fixed architecture compensator since most of the structure of the solution has vanished, as shown in Chapter 4. The only obvious substitution scheme is,in that case, to find P and Q as functions of G, A_c and K by solving the Lyapunov equations Eqs.(3.3.19,20) and find, next, G, A_c and G as functions of P and Q by solving the systems of equations of Propositions 4.3 and 4.4. This approach is almost never successful. Another approach proposed in [Ber87b], and can only be applied to fully decentralized control (no sensor or actuator shared by any two subcontrollers), is to sucessively optimize each subcontroller. This has no guarantee of finding the optimal solution, and it may have poor convergence properties since the optimization of one subcontroller may have adverse effects on the optimality of the remaining loops which are kept fixed.

The only method found in the literature for solving the fixed architecture control problem in a global fashion has been that of [Wen80]. It considers the problem as an optimization problem with equality constraints: instead of looking for G, A_c and K in matrix spaces of reduced dimensions so that they satisfy the order and architecture constraints, the constraints are relaxed and the feedback may be a centralized full order system. The Lagrangian is then augmented with the weighted sum of the squares of the entries in G, A_c and K that should be zero. The algorithm becomes an unconstrained optimization problem for which a conjugate gradient method is applied. [Wen80] shows that if a solution to the constrained architecture problem exists, it will be one minimum of the augmented problem. A very simple output feedback problem is shown to illustrate the method. The advantage of this approach is that the initial design can be the full order LQG compensator. However, there is no guarantee that such an initial guess will converge to the minimum corresponding to the fixed architecture compensator. In the case of the dynamic compensator, the method may end up finding a specific state space representation of the full order compensator rather than converging on the fixed architecture compensator. One must then iterate on the starting point. Such a task may turn out to be very difficult. The weighting imposed on the structural constraints may then have to be very large and can make the problem numerically ill-conditioned. These problems do not appear, however, in the static output feedback problem treated in [Wen80].

Continuation, or homotopy methods have appeared to be a very promising approach for solving complex constrained control synthesis problem. Their successful use has been reported on many occasions in [Mon69, Lef85, Seb86, Kab87, Ric87, Ric89, Pet90]. As argued in [Ric89, Hyl90], these methods tend to be the only ones that work when the problem is numerically ill-conditioned.

The basis for the use of continuation methods is to have a problem that depends continuously on some design parameters so that the solution can be continuously differentiated with respect to them. The problem has the generic form

$$0_V = F(X,\alpha) \tag{5.1.1}$$

where F is a continuously differentiable function depending on a parameter α mapping a finite dimensional vector X into some vector space V. Differentiating Eq.(5.1.1), one gets

$$0_V = F_X \cdot X_\alpha + F_\alpha \tag{5.1.2}$$

Denoting by X_0 a solution to Eq.(5.1.1) for $\alpha = \alpha_0$, a solution to Eq.(5.1.1) for $\alpha = \alpha_1$ can be obtained by integrating Eq.(5.1.2)

$$X(\alpha_1) = X(\alpha_0) - \int_{\alpha_0}^{\alpha_1} F_X^{-1} F_\alpha \, d\alpha \tag{5.1.3}$$

Eq.(5.1.3) holds as long as the integrand inside the integral sign is a well defined quantity, which is equivalent to saying that the Implicit Function Theorem applies for every $\alpha \in [\alpha_0, \alpha_1]$, [Llo78]. Two solutions $X(\alpha_0)$ and $X(\alpha_1)$ will be qualitatively of the same nature if there exists a path of solutions connecting one to the other.

The main difficulty in applying homotopy is to be able to determine if all the solutions to the final problem can be reached through homotopy by following the solutions to the simplified problem. An attempt to answer these questions has been made in [Ric87, Ric89]. The theoretical basis to such a problem is covered by topological degree theory [Llo78, Eav83]. It is shown that under certain conditions the number of solutions to Eq.(5.1.1) remains constant, independent of α. The result clearly holds for the full order LQG problem, but it will be shown that a global result cannot be obtained for the more general constrained problem.

The remainder of this chapter is organized as follows. The next section shows how to use homotopy to solve the optimal fixed architecture control problem. The question of choosing a deformed simplified problem is addressed, the derivative of the optimality equations with respect to the homotopy parameter is given and it is shown that the singularity of the problem due to the freedom in choosing the compensator realization does not prevent one from defining an integrand and computing the integral in Eq.(5.1.3).

Based on these preliminary steps, a continuation algorithm is developed for solving the Fixed Architecture Control problem. The algorithm combines a simple Euler forward integration scheme to evaluate the integral and a minimization scheme using a mixed steepest gradient / Newton-Raphson scheme in order to control the error.

One can locally track the solution of any control problem using a continuation method. A more important question is to know if the property is global. To that effect, one must check that continuous paths connect the solutions of any two control problems with different design parameters. One must also be able to find the entire set of solutions for simplified problems. Finally, one must deal with critical points. Studying the question for the reduced order problem, which can be stated in the form of the Optimal Projection Equations [Hyl84], original examples are derived in this chapter to show that:

- not every stabilizing solution to the simplified problem connects to a stabilizing solution to the final problem;
- all solutions to simple diagonal problems cannot be systematically found;
- bifurcations can occur along a solution path;
- the nature of the solution can change along the path. More particularly, local minima can become saddle points and stabilizing solutions may become nonstabilizing solution in a continuous manner.

The contribution of this chapter is to characterize the behavior and the nature of the solutions to the constrained H_2 problem. The results derived in the following sections prove that homotopy cannot be used as a global tool to find all the solutions to the Fixed Architecture Control Problem and indicate that any successful numerical procedure must provide ways to abandon a solution which ceases to correspond to a stabilizing solution and a minimum. A second conclusion is that the choice of the initial simplified problem will be very important. A design example illustrates this fact and shows that, by not choosing a starting problem close enough to the problem of interest, one may not find a continuous path of solutions between the two problems. A proper choice of the initial problem, however, results in a very smooth behavior of the solution along the path and very good convergence properties.

5.2 HOMOTOPIC CONTINUATION METHOD

5.2.1 Deformed Problem and Initial Solution

In order to use a continuation method, one must find a family of control problems that depend on a parameter α such that $\alpha = 0$ corresponds to a problem to which some, or all the solutions are known and $\alpha = 1$ corresponds to the control problem one wishes to solve. Consider the Fixed Architecture Control Problem 3.1 whose design parameters are

(C_0,A_0,B_0)	plant $C_0 \in \mathbb{R}^{l\times n}$, $A_0 \in \mathbb{R}^{n\times n}$, $B_0 \in \mathbb{R}^{n\times m}$
V_0	process noise covariance, $V_0 \in \mathbb{R}^{n\times n}$,
V_{c0}	measurement noise covariance, $V_{c0} \in \mathbb{R}^{l\times l}$
R_0	penalty matrix on the states, $R_0 \in \mathbb{R}^{n\times n}$,
R_{c0}	penalty on the control inputs, $R_{c0} \in \mathbb{R}^{m\times m}$,

along with the specification of the architecture,

p	number of loops,
n_i	order of subcontroller i
$\mathcal{U}_i$,	set of actuator indices belonging to subcontroller i,
$\mathcal{Y}_i$,	set of sensor indices belonging to subcontroller i,

The parameters of the problem are assumed to have the following form:

$$A_0=\begin{bmatrix} A_{11} & 0 & \cdots & 0 & 0 \\ 0 & A_{22} & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & A_{pp} & 0 \\ 0 & 0 & \cdots & 0 & A_{p+1p+1} \end{bmatrix}, B_0=\begin{bmatrix} B_{11} & 0 & \cdots & 0 \\ 0 & B_{22} & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & B_{pp} \\ 0 & 0 & \cdots & 0 \end{bmatrix}, C_0=\begin{bmatrix} C_{11} & 0 & \cdots & 0 & 0 \\ 0 & C_{22} & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & C_{pp} & 0 \end{bmatrix}$$

$A_{ii} \in \mathbb{R}^{n_i \times n_i}$, n_i= order of compensator i, $i \leq p$,

$B_{ii} \in \mathbb{R}^{n_i \times \underline{m}_i}$, $\underline{m}_i \in \mathbb{N}$, $\underline{m}_i \leq m_i$ number of actuators retained in loop i as specified by $\mathcal{U}_i$

$C_{ii} \in \mathbb{R}^{\underline{l}_i \times n_i}$, $\underline{l}_i \in \mathbb{N}$, $\underline{l}_i \leq l_i$ number of sensors retained in loop i as specified by $\mathcal{Y}_i$,

$$V_0 = \begin{bmatrix} V_{11} & 0 & \cdots & 0 & 0 \\ 0 & V_{22} & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & V_{pp} & 0 \\ 0 & 0 & \cdots & 0 & V_{p+1p+1} \end{bmatrix}, \ V_{c0} = \text{blockdiag}(V_{10}, V_{20},\cdots,V_{p0}),$$

$V_{ii} \in \mathbb{R}^{n_i \times n_i}$, positive, $V_{i0} \in \mathbb{R}^{l_i \times l_i}$, positive definite,

$$R_0 = \begin{bmatrix} R_{11} & 0 & \cdots & 0 & 0 \\ 0 & R_{22} & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & R_{pp} & 0 \\ 0 & 0 & \cdots & 0 & R_{p+1p+1} \end{bmatrix}, \ R_{c0} = \text{blockdiag}(R_{10}, R_{20},\cdots,R_{p0}),$$

$R_{ii} \in \mathbb{R}^{n_i \times n_i}$, positive, $R_{i0} \in \mathbb{R}^{m_i \times m_i}$, positive definite.

The architecture is such that the first $\underline{m}_1$ actuators and $\underline{l}_1$ sensors are used by subcontroller 1, the next $\underline{m}_2$ actuators and $\underline{l}_2$ sensors are used by subcontroller 2, etc. This is not a restrictive hypothesis since one can renumber the sensors and actuators.

Systems for which the design problem simplifies are systems constituted of completely independent subsystems, each of them having its own sensors, actuators, independent cost and independent disturbance. The global cost is the sum of the costs defined for each subsystem. The dynamics of the subsystems are totally uncontrollable and unobservable from all actuators and sensors except their own. The fixed architecture problem decouples, in that case, in a number of smaller centralized full order control problems. If the architecture allows for a sensor to be shared by two or more subcontrollers, the shared sensor will affect only one of the smaller centralized control problems since only one subsystem is observable from this particular sensor. Hence, the effective number of nonzero rows $\underline{l}_i$ in C_{ii} may be smaller than l_i, the number of sensors connected to subcontroller i. Using a dual argument about controllability, one can see that the number of nonzero columns $\underline{m}_i$ in B_{ii} may be smaller than m_i, number of actuators used by subcontroller i.

The simplified system, defined above, is made of the aggregation of p separate smaller LQG problems. The plant is made of the aggregation of p+1 subsystems, p of which have their own set of sensors and actuators, the last one being both completely unobservable and completely uncontrollable. Neither the cost nor the various noises entering the systems couple the different subsystems together. Hence, for this very specific type of plant, the Fixed Architecture Control Problem decouples into p unconstrained LQG problems.

Consider, now, Problem 3.1 with the same architecture and same matrix dimensions, but with design parameters (C_1, A_1, B_1) for the plant, V_1 and V_{c1} for the process and measurement noise covariances, R_1 and R_{c1} for the penalty matrices on the states and the control inputs. Define the intermediate problems corresponding to the following parameters

$$A(\alpha) = A_0 + f_1(\alpha)(A_1 - A_0) \tag{5.2.1}$$

$$B(\alpha) = B_0 + f_2(\alpha)(B_1 - B_0) \tag{5.2.2}$$

$$C(\alpha) = C_0 + f_3(\alpha)(C_1 - C_0) \tag{5.2.3}$$

$$V(\alpha) = V_0 + f_4(\alpha)(V_1 - V_0) \tag{5.2.4}$$

$$R(\alpha) = R_0 + f_5(\alpha)(R_1 - R_0) \tag{5.2.5}$$

$$V_c(\alpha) = V_{c0} + f_6(\alpha)(V_{c1} - V_{c0}) \tag{5.2.6}$$

$$R_c(\alpha) = R_{c0} + f_7(\alpha)(R_{c1} - R_{c0}) \tag{5.2.7}$$

where $\alpha \in [\,0, 1]$, and where $f_i(\alpha)$ are right differentiable functions such that $f_i(0) = 0$ and $f_i(1) = 1$. Eqs.(5.2.1-7) define a one-parameter family of problems that starts as a series of simpler decoupled LQG problems ($\alpha = 0$) and continuously deforms into the actual control problem ($\alpha = 1$). If a continuous path of solutions exists, one can find solutions to the Fixed Architecture Control Problem. The simplest choice for f_i is to take $f_i(\alpha) = \alpha$. This produces a linear deformation of the parameters of the problem. A piecewise linear transformation can be obtained simply by defining intermediate problems and applying a linear transformation between them. For each new intermediate problem, the starting solution is the one obtained from the previous problem. The procedure is started with a simplified diagonal problem. The numerical procedure inplemented in the following sections utilizes a linear transformation of the parameters and allows, as just argued, piecewise linear deformation of the parameters.

5.2.2 Solving $L_{\xi\xi}\cdot\xi_\alpha + L_{\xi\alpha} = 0$

5.2.2.1 Analytical Aspect

Following the notation of Chapter 3, the Lagrangian L depends on a vector ξ containing the free parameters of the controller gains and dynamics (it is assumed here that the matrices P and Q have been eliminated by solving Eqs.(3.3.19,20)). The Lagrangian depends also on the design parameters which can be represented by a single parameter α if they are given by Eqs.(5.2.1-7). L can thus be formally written as $L(\xi,\alpha)$. The optimality condition becomes:

$$0 = L_\xi(\xi,\alpha) \tag{5.2.8}$$

Differentiating Eq.(5.2.8), one obtains the following equation

$$0 = L_{\xi\xi}\cdot\xi_\alpha + L_{\xi\alpha} \tag{5.2.9}$$

where $L_{\xi\xi}$ is the Hessian, ξ_α is a vector containing the derivative of ξ with respect to α and $L_{\xi\alpha}$ is the partial derivative of the optimality conditions with respect to α. $L_{\xi\alpha}$ can be obtained following steps similar to Eqs.(3.3.26-33). P_α and Q_α are solutions of the following Lyapunov equations:

$$0_{\tilde{n}\times\tilde{n}} = A_{cl}^T P_\alpha + P_\alpha A_{cl} + A_{cl_\alpha}^T P + P A_{cl_\alpha} + R_{cl_\alpha} \tag{5.2.10}$$

$$0_{\tilde{n}\times\tilde{n}} = A_{cl} Q_\alpha + Q_\alpha A_{cl}^T + A_{cl_\alpha} Q + Q A_{cl_\alpha}^T + V_{cl_\alpha} \tag{5.2.11}$$

where:

$$A_{cl_\alpha}=\begin{bmatrix}A_\alpha & B_\alpha G\\ KC_\alpha & 0\end{bmatrix},\ R_{cl_\alpha}=\begin{bmatrix}R_\alpha & 0\\ 0 & G^TR_{c_\alpha}G\end{bmatrix},\ V_{cl_\alpha}=\begin{bmatrix}V_\alpha & 0\\ 0 & KV_{c_\alpha}K^T\end{bmatrix}, \tag{5.2.12}$$

and where A_α, B_α, C_α, R_α, V_α, R_{c_α} and V_{c_α} are obtained by differentiating Eqs.(5.2.1-7) with respect to α. Define:

$$M_\alpha = P_\alpha Q + PQ_\alpha$$

Partition P_α, Q_α and M_α as:

$$P_\alpha=\begin{bmatrix}P^\alpha_{00} & P^\alpha_{0c}\\ P^\alpha_{c0} & P^\alpha_{cc}\end{bmatrix},\ Q_\alpha=\begin{bmatrix}Q^\alpha_{00} & Q^\alpha_{0c}\\ Q^\alpha_{c0} & Q^\alpha_{cc}\end{bmatrix},\ M_\alpha=\begin{bmatrix}M^\alpha_{00} & M^\alpha_{0c}\\ M^\alpha_{c0} & M^\alpha_{cc}\end{bmatrix}$$

Using the chain rule, $L_{A_c\alpha}$, $L_{G\alpha}$ and $L_{K\alpha}$ become:

$$L_{A_c\alpha} = M^\alpha_{cc}$$
$$L_{G\alpha} = R_{c_\alpha}GQ_{cc} + R_cG\,Q^\alpha_{cc} + B^TM^\alpha_{0c}$$
$$L_{K\alpha} = P_{cc}KV_{c_\alpha} + P^\alpha_{cc}KV_c + M^\alpha_{c0}C^T$$

Hence:

$$L_{a_i\alpha} = \mathrm{Tr}E^{a_i^T}M^\alpha_{cc} \tag{5.2.13}$$
$$L_{g_i\alpha} = \mathrm{Tr}E^{g_i^T}(R_{c_\alpha}GQ_{cc} + R_cG\,Q^\alpha_{cc} + B^TM^\alpha_{0c}) \tag{5.2.14}$$
$$L_{k_i\alpha} = \mathrm{Tr}E^{k_i^T}(P_{cc}KV_{c_\alpha} + P^\alpha_{cc}KV_c + M^\alpha_{c0}C^T) \tag{5.2.15}$$

$L_{\xi\alpha}$ is the column vector composed of the elements $L_{g_i\alpha}$, $L_{a_i\alpha}$ and $L_{k_i\alpha}$.

As seen in Sect 4.4, any change in the realization of the compensator transfer function leaves the cost invariant and, consequently, the Hessian $L_{\xi\xi}$ is singular. It can be shown, however, that the vector $L_{\xi\alpha}$ is always orthogonal to the vectors ξ_ε, Eq.(4.4.2), that span the nullspace of the Hessian $\mathrm{Ker}(L_{\xi\xi})$. Eq.(5.2.10) and Eq.(5.2.11) are the differentiated form of Eq.(3.3.19) and

Eq.(3.3.20) with respect to α. Because the two equations still hold after differentiation, Eq.(4.4.4) also holds in a differentiated form. Hence,

$$0_{n_i \times n_i} = \Pi_i(L^T_{A_c\alpha}A_c - A_cL^T_{A_c\alpha} + L^T_{G\alpha}G - KL^T_{K\alpha})\Pi_i^T, \; i = 1, \cdots, p$$

G, A_c and K are constant parameters in Eqs.(5.2.10,11) and consequently G_α A_{c_α} and K_α are zero. Consider now N_ε to be the gradient of a similarity transformation as defined in Theorem 4.3. N_ε is block diagonal and therefore, using the above result, the following holds,

$$0 = \mathrm{Tr}(L^T_{A_c\alpha}A_c - A_cL^T_{A_c\alpha} + L^T_{G\alpha}G - KL^T_{K\alpha})N_\varepsilon$$

or, after rearranging the expression using the properties of the trace,

$$0 = \mathrm{Tr}L^T_{A_c\alpha}(-N_\varepsilon A_c + A_cN_\varepsilon) + \mathrm{Tr}L^T_{G\alpha}GN_\varepsilon - \mathrm{Tr}L^T_{K\alpha}N_\varepsilon K$$

The right hand side of the equation corresponds to the matrix formulation, Eq.(3.3.1), of the inner product between $L_{\xi\alpha}$ and ξ_ε, where ξ_ε is given in Eq.(4.4.2). The above formula states therefore that $L_{\xi\alpha}$ is orthogonal to any of the basis vectors of $\mathrm{Ker}(L_{\xi\xi})$ that correspond to a change in the compensator realization: $L_{\xi\alpha}$ does not affect the realization of the compensator. As long as $L_{\xi\xi}$ does not have any other singularities, Eq.(5.2.9) will admit at least one solution. This result means that the real variable is not a vector but an equivalence class in the vector space of the compensator realizations. Two compensators are equivalent if they realize the same transfer function. The solution to the optimal control problem is in fact a transfer function independent of any particular state space representation. Factoring out the nullspace of $L_{\xi\xi}$ in the problem, one removes the singularity and obtains a problem similar to that of Section 5.1. In practice, the solution obtained by solving Eq.(5.2.9) contains two components: one is orthogonal to the nullspace of $L_{\xi\xi}$, $\mathrm{Ker}(L_{\xi\xi})$, and describes the changes in the compensator transfer function. The second component of ξ_α is parallel to $\mathrm{Ker}(L_{\xi\xi})$ and describes a change of realization of the compensator state space description. The freedom resulting from choosing the component of ξ_α along $\mathrm{Ker}(L_{\xi\xi})$ can be used to obtain better numerical properties.

5.2.2.2 Practical Aspect

The following methods can be used to find ξ_α numerically:

1) Use of a minimal set of equations and a minimal set of parameters:

As seen in Section 4.4.2.1 and Section 4.4.2.2, one can find a realization for the compensator such that only a reduced number of variables are needed. Similarly, only a reduced number of equations need to be satisfied. Calling ξ_0 a vector made of this reduced set of components, Eq.(5.2.9) becomes:

$$0 = L_{\xi_0 \xi_0} \cdot \xi_{0\alpha} + L_{\xi_0 \alpha}$$

where $L_{\xi_0 \xi_0}$ does not have a nullspace corresponding to a change in compensator realization.

2) Minimal set of parameters, complete set of equations:

The compensator is described by a minimal set of parameters ξ_0. All the necessary conditions are, however, considered. Eq.(5.2.9) becomes:

$$0 = L_{\xi \xi_0} \cdot \xi_{0\alpha} + L_{\xi \alpha}$$

This system has more equations than unknowns, but it has one solution that can be obtained using a least squares method.

3) Nonminimal set of parameters, full set of equations:

The compensator is described by a reduced but not minimal set ξ_1 such as the modal description of Section 4.4.2.3. Eq.(5.2.9) becomes:

$$0 = L_{\xi \xi_1} \cdot \xi_{1\alpha} + L_{\xi \alpha}$$

where ξ_1 is a reduced vector representing the nonminimal set of variables. The least squares solution can be obtained numerically.

5.3 A CONTINUATION ALGORITHM

5.3.1 General

The use of homotopy has transformed the resolution of the optimal control problem into a simple integration problem. Many numerical schemes have been developed for that purpose [Pre86] and the main difficulty encountered by any method is to stabilize the scheme and control the size of the error. These difficulties can be overcome when one uses a continuation scheme, since the calculated solution $\xi(\alpha)$ must always satisfy Eq.(5.2.8). It is therefore possible to monitor the error and take steps to reduce it. Different ways have been proposed to stabilize the integration. In [Mon69], a descent component is added to the gradient of the cost with respect to α. The forward step then tracks the solution path and tries at the same time to reduce the error inherited from the previous step. [Kab87] remarks a more systematic approach which consists of performing a minimization step after a forward step has been taken. A very similar approach is taken in [Ric87, Ric89] where the method relies in fact only on a local search of the solution: the gradient of the solution with respect to α is not calculated, and instead, the homotopy parameter is simply incremented. The previous solution is then used as the starting point of an iterative approach similar to [Gru86]. This last procedure suffers, however, the same shortcomings as the procedure developed in [Gru86] since the local search may itself be numerically unstable.

5.3.2 Structure of the Algorithm

The algorithm proposed here is a continuous homotopy algorithm with minimization steps to stabilize the forward integration. The gradient ξ_α is calculated and α is increased until the norm of the gradient $\|L_\xi\|$ is within 10% of a threshold ε_2. This step is denoted as the shooting step. The norm of the gradient is then reduced using a mixed steepest descent / Newton-Raphson method until the error becomes smaller than a second threshold ε_0. If the minimization fails, the shooting step is halved in order to start the minimization at a point closer to a solution point. The structure of the algorithm is the following:

<u>Step 1</u>: $\alpha = 0$
Find initial solution to the diagonal problem
or use existing solution of an already
known nondiagonal problem

<u>Step 2</u>: Compute the gradient ξ_{α_k}

<u>Step 3</u>: Shooting step:

find $\Delta\alpha_k$ such that

$$\alpha_{k+1} = \alpha_k + \Delta\alpha_k$$
$$\xi^0_{k+1} = \xi_k + \xi_{\alpha_k}\Delta\alpha_k$$
$$0.9\,\varepsilon_2 < \|L_\xi(\xi^0_{k+1})\| < 1.1\,\varepsilon_2$$

<u>Step 4</u>: Minimize L at α_{k+1}:

Initialize search at ξ^0_{k+1}.
If $\alpha < 1$, minimize until $\|L_\xi(\xi_{k+1})\| < \varepsilon_0$
If $\alpha = 1$, minimize until $\|L_\xi(\xi_{k+1})\| < \varepsilon_1$

<u>Step 5</u>: If minimization fails,

halve $\Delta\alpha_k$, repeat 4.

If minimization successful, accept ξ_{k+1}.

<u>Step 6</u>: Repeat Step 2 through 6 until $\alpha = 1$.

Two thresholds are used for minimization purposes. A coarser ε_0 is used throughout the integration. When $\alpha = 1$, a smaller threshold ε_1 is used to improve on the accuracy of the final solution. The value of ε_2 drives the size of the shooting step and should be set as large as possible. If ε_2 is too large, however, the minimization may start too far away from a solution and ε_2 must be chosen so that one stays in the region of rapid convergence of the minimization scheme. The minimization can be started with steepest descent steps which are more robust far away from a solution, thus allowing larger ε_2. When closer to a solution, the algorithm takes modified Newton's steps which have a higher rate of convergence. The Hessian must be calculated and inverted in order to get the direction in the shooting step and one can therefore use the same routine in order to get the descent direction of the modified Newton step. Figure 5.1 shows how the error evolves as a function of α. The size of the step varies depending on the sensitivity of the solution to plant parameter variations and design parameter variations.

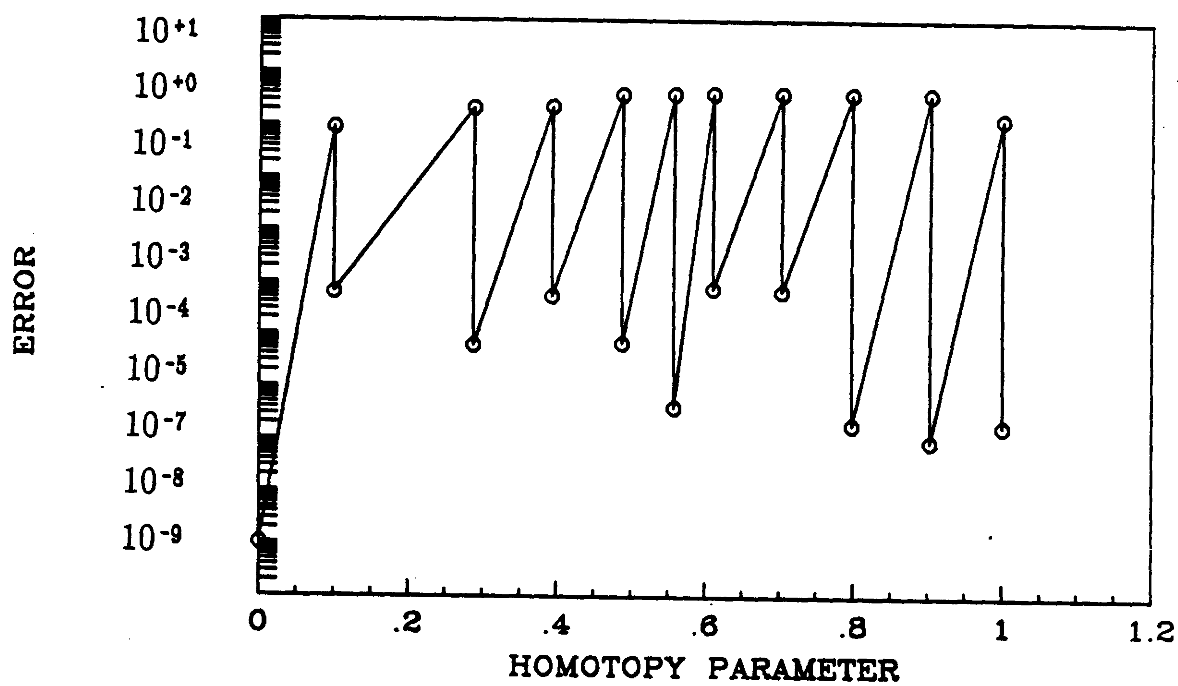

Figure 5.1: Variation of the Gradient Norm during the Homotopy

5.3.3 Choice of the Free Parameters

The general structure of the algorithm does not depend on the parameters chosen to describe the compensator in state space. The details of the calculations are however dependent on that choice as highlighted in Section 5.2.2.2. The modal form described in Section 4.4.2.3 appears to be a well conditioned parameterization that leaves only n_i extra degrees of freedom per loop. These extra degrees of freedom correspond to the possibility to change the scaling of the various states without modifying the block structure of the realization. Problems can occur, however, when eigenvalues from two different blocks merge and form a complex conjugate pair . The hypothetical root locus of Figure 5.2 illustrates such a problem. In the example depicted, A_c is four dimensional and has two diagonal blocks. The complex conjugate eigenvalues 1 and 2 belonging to one 2×2 block of A_c in modal form merge and then split on the real axis to give eigenvalues 5 and 6. Similarly the complex conjugate pair 3 and 4 which belongs to a second 2×2 block merge and split on the real axis to give eigenvalues 7 and 8. If the eigenvalues keep shifting and 5 and 8 become 9 and 12, and 6 and 7 become 10 and 11, 10 and 11 must then belong to the same 2×2 block in the modal representation of A_c. These eigenvalues come from two different blocks, however.

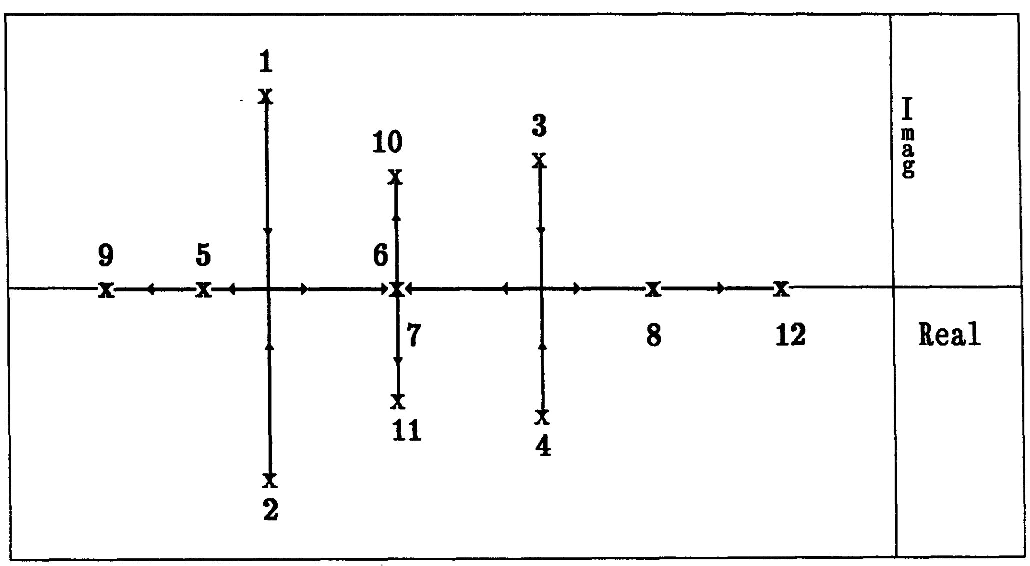

Figure 5.2: Hypothetical Root Locus

This implies that the blocks must be changed in order to allow 6 and 7 to become a complex conjugate pair of eigenvalues. The root locus cannot, therefore, be obtained by the continuous deformation of a matrix in modal form. The discontinuity problem must be overcome by adding some extra degrees of freedom and letting, for example, the dynamic matrices A_i be tridiagonal. Such a realization allows for $3 \times n_i - 2$ extra degrees of freedom to realize differently the transfer function of the i^{th} subcontroller,, versus n_i for the modal form, and n_i^2 for the most general representation. Let ξ, Eq.(3.3.24), be the vector of all the free entries in A_c, G and K that have not been set to zero when fixing the control architecture. Specifying a tridiagonal form for the A_i results in constraining some entries in ξ to be zero or, in other words, it constrains ξ to lie in a subspace S_D of $\mathbb{R}^{n_p}$, where n_p is the dimension of ξ. Let D be a diagonal matrix of size n_p whose diagonal is defined by:

$$\begin{aligned} D(i,i) &= 1 \text{ if } \xi_i \text{ is free} \\ D(i,i) &= 0 \text{ if } \xi_i \text{ is constrained to be 0.} \end{aligned} \tag{5.3.1}$$

Define the vector ξ_d as

$$\xi_d = D\xi \tag{5.3.2}$$

ξ_d is the expanded form of the vector containing the free parameters of the new problem. The solution to Eq.(5.2.9) will be constrained to be in S_D. Because the set of parameters is not minimal, one still has some freedom in choosing the solution in S_D. The solution chosen will be the one of smallest norm in S_D. The advantage in restricting the solution to be in S_D is that one does not have to compute the whole Hessian but only the columns corresponding to the free entries of ξ_d. This will, first, save computational time and, second, reduce the dimension of the nullspace of the Hessian on S_D.

5.3.4 Modified Steepest Descent Step

Denoting by ξ_{dk} the value of the vector at the k^{th} step of the minimization, and by g_k the error vector equal to

$$g_k = L_\xi(\xi_{dk})$$

the cost is to first order:

$$L(\xi_{dk}+\Delta\xi_{dk}) = L(\xi_{dk}) + g_k^T\Delta\xi_{dk} + 0(\|\Delta\xi_{dk}\|^2)$$

Taking a step $\Delta\xi_{dk}$ equal to

$$\Delta\xi_{dk} = -\sigma_k D g_k$$

the cost becomes

$$L(\xi_{dk}+\Delta\xi_{dk}) = L(\xi_{dk}) - \sigma_k g_k^T D g_k + 0(\sigma_k^2)$$

and will be locally decreasing for σ_k sufficiently small since D is a symmetric positive matrix thus making $g_k^T D g_k$ a positive quantity. If $g_k^T D g_k$ is zero, then Dg_k must be zero and, following the results of Section 4.4.2, the entire error vector g_k must be zero. Hence, the step is similar to a usual steepest descent step. σ_k is found to minimize the cost along the direction $-Dg_k$ using a bracketing technique [Sca85].

5.3.5 Modified Newton-Raphson Step

Newton's method has a quadratic rate of convergence when the Hessian matrix is positive definite. $L_{\xi\xi}$ is singular at any solution point as seen in Chapter 4, but the cost is invariant to every order in a change in the realization of the compensator transfer function. Hence, one can expect the convergence rate to remain quadratic close to the solution. Far away from a stationary point the Hessian regains its full rank since the gradient vector L_ξ, if it is not zero, will depend on the realization of the compensator. Some eigenvalues of the Hessian must then go to zero as the error is reduced. One cannot rely on the direction indicated by the eigenvectors associated with these eigenvalues in order to find a search direction since these eigenvalues may not be positive. The search direction $d\xi_{dk}$ is computed as follows. First, one must be sure that the search direction will be in S_D to preserve the type of realization chosen for the compensator. The following matrix L_{dd} is used instead of the Hessian:

$$L_{dd} = DL_{\xi\xi}D \tag{5.3.3}$$

where D is given in Eq.(5.3.1). L_{dd} maps S_D into itself and $S_D{}^{\perp}$ into zero. L_{dd} is a symmetric matrix, hence diagonalizable, and can be expanded in the following sum:

$$L_{dd} = \sum_{i=1}^{N_d} \lambda_i u_i u_i^T$$

where the u_i are the eigenvectors of L_{dd} belonging to S_D, where the λ_i's are the corresponding eigenvalues arranged in decreasing order, taking their sign into account, and where N_d is the number of free entries in ξ_d. The remaining eigenvectors span the nullspace of L_{dd} which coincides with $S_D{}^{\perp}$. Define the matrix H_k as:

$$H_k = \sum_{i=1}^{N_m} \frac{1}{\lambda_i} u_i u_i^T$$

where N_m is the minimal number of parameters required to fully describe the compensator's transfer function. As one approaches the solution, the remaining N_d–N_m smallest eigenvalues of L_{dd} go to zero. The search direction $d\xi_{dk}$ is defined by:

$$d\xi_{dk} = -H_k g_k$$

The step is admissible since H_k, like L_{dd}, maps $\mathbb{R}^{n_p}$ into S_D. To first order, the cost becomes

$$L(\xi_{dk}+\sigma_k d\xi_{dk}) = L(\xi_{dk}) - \sigma_k g_k^T H_k g_k + 0(\sigma_k^2)$$

and will be locally decreasing. σ_k can be chosen to minimize the cost along the search direction. The step becomes $\Delta\xi_{dk} = \sigma_k d\xi_{dk}$. The cost is reduced at every step and, eventually, the step σ_k will converge to 1, or to a full Newton step.

5.3.6 Convergence Properties

The minimization procedure used in the algorithm has convergence properties similar to those of other classic second order methods [Sca85]. When close to a minimum, the method is guaranteed to converge quadratically. Because a gradient step is used, and because line searches are performed during the minimization, the algorithm is guaranteed to provide a minimum. Since the minimization always start from a stabilizing solution, such a solution must exist. The convergence properties may be poor if the starting point is close to another stationary point which is not the minimum.

The shooting procedure guarantees that the starting solution obtained for the next minimization step will be close to a local minimum. As long as there is no critical point along the path, that is, the nullspace of the Hessian remains limited to those transformations that modify the compensator realization, and as long as the path connects stabilizing solutions, the shooting step will always produce a near optimal solution. The quality of the solution can be controlled using ε_2. The following sections study if the singular cases just mentioned can occur, that is, the possibility for the closed loop to become unstable, or the possibility for the solution to encounter a critical point. In such occurrences, the shooting step will tend toward zero, or the initial point it provides to the minimization will cease to be a local minimum.

5.4 ON THE NUMBER AND NATURE OF THE SOLUTIONS TO THE CONSTRAINED H_2 PROBLEM

5.4.1 General

$L_\xi(\xi,\alpha)$ is a continuously differentiable function of ξ and α. One can therefore track the solution to Eq.(5.2.8) at least locally, except possibly in some very pathological case corresponding to critical or bifurcation points. A more important question is whether or not, and under what assumptions, the property becomes global. [Ric87, Ric89] try to answer such questions using topological degree theory [Llo78]. The proof is worked for the full LQG problem, but does not generalize, however, to the fixed order control problem and, consequently, to the fixed architecture control problem as the following sections will show. The question will be examined for the reduced order problem for which the optimality conditions can be stated in terms of the Optimal Projection Equations (OPE). The following examples demonstrate that the OPE have possibly many algebraic solutions. Some of the solutions do not stabilize the plant while others do, and some stabilizing solutions can be local minima for the cost or saddle points. A series of examples will prove that the nature of the solution is not invariant along a solution path, and that stabilizing solutions can continuously become nonstabilizing, while minima can become saddles. These results point out that no global properties exist for the reduced order case, or for the fixed architecture case, which generalizes it.

5.4.2 Degree Theory

The following section presents the main definitions and results of degree theory. A complete treatment of the subject can be found in [Llo78, Eav83].

Definition 5.1: *Let D be a bounded, open subset of* $\mathbb{R}^n$, P *a point of* $\mathbb{R}^n$. $\overline{D}$ *is the closure of* D, ∂D *its boundary.* $C^1(\overline{D})$ *is the space of continuously differentiable functions defined on an open set containing* $\overline{D}$ *that maps* $\overline{D}$ *in* $\mathbb{R}^n$. *Let* $\phi \in C^1(\overline{D})$, $P \in \mathbb{R}^n$. $J_\phi = \det(\phi_X)$ *is the Jacobian of* ϕ. X *is said to be a regular point for* ϕ *if* $J_\phi \neq 0$. X *is a critical point otherwise.* $P = \phi(X)$ *is said to be a critical value if X is a critical point. The set of critical values is called the crease of the function.*

Suppose $\phi \in C^1(\overline{D})$, $P \notin \phi(\partial D)$, P $\notin$ *crease of* ϕ. *Define the degree of* ϕ *at* P *relative to* D *to be* deg(ϕ, D, P), *where*

$$\deg(\phi, D, P) = \sum_{X \in \phi^{-1}(P)} \operatorname{sign} J_\phi(X)$$

■

Since $\overline{D}$ is compact and P is not in the crease of ϕ, the problem $\phi(X) = P$ has a finite number of solutions and the summation is finite. Also, if deg(ϕ, D, P) $\neq 0$, $\phi(X) = P$ has at least one solution in D.

Theorem 5.1: *Consider the continuous mapping* $H: D \times [0,1] \rightarrow \mathbb{R}^n$, $h_t = H(X,t)$. *Assume that* $h_t(X) = P$ *has no solution on* ∂D *for any* $t \in [0,1]$. *Then,*

$$\deg(h_0, D, P) = \deg(h_1, D, P).$$

Let D *be an open, bounded set of* $\mathbb{C}^n$, *and* ϕ *an holomorphic mapping on* $\mathbb{C}^n$. *If* $P \notin \phi(\partial D)$, *then* deg(ϕ, D, P) ≥ 0. ■

Proof: the proof is in [Llo78]. The theorem states that the degree is invariant under homotopy. This means that, if no solution appears or disappears on the boundary, new solutions have to appear and disappear in pairs, one satisfying $J_\phi > 0$, the other $J_\phi < 0$. If the degree is nonzero, there is at least one continuous path of solutions connecting the solution of $h_0(X) = P$ to the solutions of $h_1(X) = P$. The homotopy invariance is illustrated by Figure 5.3 taken from [Llo78]. An holomorphic mapping is orientation preserving. Hence, when ϕ is analytic over the domain, the degree is exactly equal to the number of solutions of $\phi(X) = P$ for $P \in D$. In that case, the only way for a solution to appear or disappear is to go to infinity.

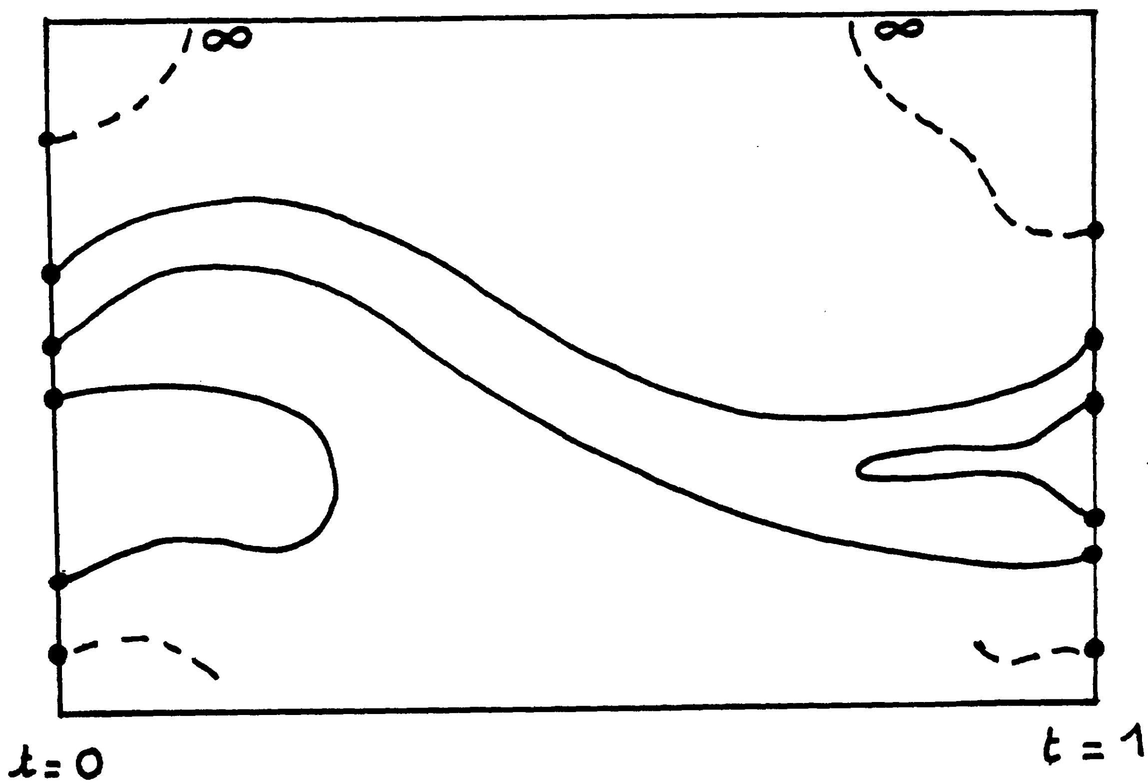

Figure 5.3: Solution Path under Homotopy
Dashed lines do not exist if degree is constant
Closing paths do not exist if the function is holomorphic

5.4.3 Boundedness of the Solution along a Path

Theorem 5.2: *Let* G, A_c, K, P, Q, $\hat{P}$, $\hat{Q}$, τ *be a solution to the OPE problem,* Eqs.(4.3.35–39), *for a plant such that*

(C, A, B) *is detectable and stabilizable,*

($R^{1/2}$, A, $V^{1/2}$) *is detectable and stabilizable.*

Assume that (G, A_c, K) *is both fully observable and fully controllable. Then, as the parameters of the problem* C, A, B, R, V, R_c *and* V_c *are continuously varied in a compact domain in such a way that the above hypotheses are always satisfied, and assuming also that* τ, $P\tau_{\perp}$ *and* $\tau_{\perp}Q$ *remain bounded along the solution path, it is true that* P *and* Q *remain bounded along the solution path, and that the corresponding matrices* G, A_c *and* K *remain bounded as well. No mechanism, however, prevents* τ, $P\tau_{\perp}$ *or* $\tau_{\perp}Q$ *from going to infinity depending on the particularity of the problem.* ∎

Proof:

1) <u>$P\tau$ and τQ remain bounded along the solution path</u>

The proof is by contradiction. Assume that there is a sequence of problems defined by (C_j, A_j, B_j) R_j, V_j, R_{cj} and V_{cj} satisfying the detectability and stabilizability hypotheses converging toward (C, A, B), R, V, R_c and V_c such that $P_j\tau_j$ goes to infinity. τ being assumed bounded, there is a sequence of unitary vectors u_j such that :

$$\sigma_j v_j = P_j u_j,\ v_j \text{ unitary},\ \sigma_j \to \infty \tag{5.4.1}$$

Pre and postmultiplying Eq.(4.3.35) by u_j^T and u_j, and using Eq.(5.4.1), we obtain:

$$0 = \sigma_j u_j^T A_j^T v_j + \sigma_j v_j^T A_j u_j + u_j^T R_j u_j - \sigma_j^2 v_j^T \Sigma_j v_j + u_j^T \tau_{j_\perp}^T P_j \Sigma_j P_j \tau_{j_\perp} u_j$$

$u_j^T \tau_{j_\perp}^T P_j \Sigma_j P_j \tau_{j_\perp} u_j$ remains bounded since $P\tau_{\perp}$ is bounded by assumption. Thus, in the limit $v^T\Sigma v = 0$, or, since Σ is positive,

$$0 = \Sigma^{1/2}v \tag{5.4.2}$$

Dividing Eq.(5.4.1) by σ_j and multiplying it by $P_j\ddagger$, one gets in the limit:

$$0 = P\ddagger v \tag{5.4.3}$$

where $P\ddagger$ is a generalized inverse defined as:

$$P\ddagger = W\begin{bmatrix} D^{-1} & 0 \\ 0 & I \end{bmatrix}W^T$$

W, and D being defined by

$$P = W^T\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}W, \text{ D diagonal, } D > 0, \text{ W unitary.}$$

Note that P is definite if ($R^{1/2}$, A) is completely observable, but it may have a nullspace if A has a stable mode not weighted by R. $P\ddagger P$ is an orthogonal projection parallel to Ker(P). Any vector v, the limit of a sequence of eigenvectors v_j such that $\sigma_j \to \infty$, satisfies Eq.(5.4.2) and Eq.(5.4.3). Postmultiplying now Eq.(4.3.35) by u_j, premultiplying by $P_j\ddagger$ and dividing by σ_j:

$$0 = P_j\ddagger A_j^T v_j + P_j\ddagger P_j A_j u_j/\sigma_j + P_j\ddagger R_j u_j/\sigma_j - P_j\ddagger P_j \Sigma_j v_j + P_j\ddagger \tau_{j_\perp}^T P_j \Sigma_j P_j \tau_{j_\perp} u_j/\sigma_j$$

$P_j\ddagger$ as well as $P_j\ddagger P_j$ are bounded, and in the limit $\Sigma_j v_j$ goes to zero. Hence, in the limit:

$$0 = P\ddagger A^T v \tag{5.4.4}$$

Combining Eq.(5.4.3) and Eq.(5.4.4), Ker($P\ddagger$) is invariant under A^T. A^T must have a mode defined by (λ_A, w_A) such that w_A is the limit of a sequence $w_j = P_j u_j/\sigma_j$ where σ_j goes to infinity. w_A satisfies Eq.(5.4.2). $P_j w_j$ must also go to infinity. Pre and postmultiplying Eq.(4.3.35) by w_j^H and w_j, one gets:

$$0 = w_j^H A_j^T w_j + w_j^H A_j w_j + w_j^H R_j w_j - w_j^H P_j \Sigma_j P_j w_j + w_j^H \tau_{j_\perp}^T P_j \Sigma_j P_j \tau_{j_\perp} w_j$$

Since R_j is bounded as well as $P_j\tau_{j_\perp}$, and since $w_j^H\Sigma_j w_j$ is positive, $w_j^H A_j^T w_j + w_j^H A_j w_j$ must be positive as j goes to infinity. Thus, in the limit,

$$2\text{Real}(\lambda_A) \geq 0$$

and from Eq.(5.4.2),

$$0 = \Sigma^{1/2} w_A$$

Thus, $(A, \Sigma^{1/2})$ is not stabilizable. This is contrary to hypothesis. Thus $P\tau$ must remain bounded. The proof for τQ can be obtained by duality.

2) <u>G, Ac, K remain bounded when τ, $P\tau$ and τQ remain bounded</u>

$P\tau$ is bounded. Hence $P\Phi$ is bounded: indeed, $\tau\Phi = \Phi$. This arises from the fact that $\tau = -\Phi\Gamma^T$ and $\Gamma^T\Phi = -I_{n_c}$. Thus G is bounded since

$$G = -R_c^{-1}B^T P\Phi$$

Similarly, τQ is bounded and $\Gamma^T\tau^T = -\Gamma$, thus:

$$K = -\Gamma^T Q C^T V_c^{-1}$$

is bounded. Finally,

$$A_c = -\Gamma^T(A - BG - KC)\Phi$$

is obviously bounded. □

The full order LQG problem is such that $\tau = I$. Theorem 5.2 implies that the solution to the problem remains in a compact domain which, in turn, implies that the number of solutions to the LQG problem is constant, as long as the detectability and stabilizability hypotheses are met, and that the problem parameters remain bounded as well. As for the reduced order problem, one can find counterexamples of $P\tau_\perp$ and $\tau_\perp Q$ being unbounded even though the above hypotheses hold. The unboundedness of P and Q always corresponds to a closed loop pole crossing the imaginary axis. Like the LQG problem, the reduced order control problem has solutions which satisfy the optimality conditions Eqs.(3.3.19-23) but for which P and Q are not positive, or, in other words, such that the closed loop is unstable. The full order solution guarantees that none of the unstable solutions will become stable and vice versa. This is not the case with the reduced order problem.

5.4.4 Unboundedness of $P\tau_\perp$ and $\tau_\perp Q$ along a Solution Path

This section presents examples of problems for which stabilizability and detectability conditions are met and where $P\tau_\perp$ and $\tau_\perp Q$ are unbounded.

5.4.4.1 Example 1

Consider the second order SISO system given by:

$$C = [1/2+\varepsilon \;\; 1/2-\varepsilon]\,,\; A = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix},\;\; B = \begin{bmatrix}1\\ 1\end{bmatrix},$$

along with the design cost parameters:

$$R = V = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\; R_c = V_c = 1.$$

The compensator is selected to be first order. The plant has two poles at -1 and $+1$ and a zero at -2ε. Writing the compensator in Controller Canonical form, the gain K is set to 1. The compensator is completely defined by its pole a_c and the control gain g which are two real numbers. The closed loop dynamic matrix is:

$$A_{cl} = \begin{bmatrix}1 & 0 & g\\ 0 & -1 & g\\ 1/2+\varepsilon & 1/2-\varepsilon & a_c\end{bmatrix}$$

The characteristic polynomial is:

$$\varphi(s) = s^3 - a_c s^2 - 2gs + a_c - 2\varepsilon g \qquad (5.4.5)$$

The polynomial admits stable roots if and only if:

$$a_c < 0$$
$$g < 0$$
$$a_c > 2\varepsilon g$$

Hence, ε must be positive and the system must have a minimum phase zero in order to be stabilizable by a first order compensator.

Let a_{c0} and g_0 be an optimal solution for $\varepsilon_0 > 0$. The corresponding compensator stabilizes the plant since the zero of the system is minimum phase. Consider now closing the loop around the system where $\varepsilon = -\varepsilon_0$ with a compensator such that $g = g_0$ and $a_c = -a_{c0}$. Using Eq.(5.4.5), the closed loop characteristic polynomial for the first system is:

$$\varphi(s) = s^3 - a_{c0}s^2 - 2g_0s + a_{c0} - 2\varepsilon_0 g_0$$

and for the second system:

$$\varphi(s) = s^3 + a_{c0}s^2 - 2g_0s - a_{c0} + 2\varepsilon_0 g_0$$

The two polynomials have the same coefficients for odd powers of s and coefficients of opposite sign for even powers of s: they admit, therefore, similar roots of opposite sign and the poles of the second system are the mirror images about the imaginary axis of the poles of the first system. If

$$P_0 = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

is the solution to Eq.(3.3.19) for $\varepsilon = \varepsilon_0$, $a_c = a_{c0}$ and $g = g_0$, then one can verify that

$$P_1 = \begin{bmatrix} -p_{22} & -p_{12} & p_{23} \\ -p_{21} & -p_{11} & p_{13} \\ p_{32} & p_{31} & -p_{33} \end{bmatrix}$$

is the solution to Eq.(3.3.19) for $\varepsilon = -\varepsilon_0$, $a_c = -a_{c0}$, $g = g_0$. Similarly, if

$$Q_0 = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$$

is the solution to Eq.(3.3.20) for the first problem,

$$Q_1 = \begin{bmatrix} -q_{22} & -q_{12} & q_{23} \\ -q_{21} & -q_{11} & q_{13} \\ q_{32} & q_{31} & -q_{33} \end{bmatrix}$$

is the solution to Eq.(3.3.20) for the second problem. One can verify that if P_0, Q_0, a_{c0}, g_0 and $k_0 = 1$ satisfy the optimality conditions Eqs.(3.3.19-23) for $\varepsilon = \varepsilon_0$, P_1, Q_1, $-a_{c0}$, g_0 and $k_1 = 1$ satisfy Eqs.(3.3.19-23) for $\varepsilon = -\varepsilon_0$. The associated cost $J_1 = \mathrm{Tr}(Q_1 R_{cl})$ is equal to $J_1 = -J_0$. The compensator satisfies the optimality conditions Eqs.(3.3.19-23) but yields a completely unstable closed loop system. As ε crosses the imaginary axis the cost becomes infinite. When the zero of the plant becomes nonminimum phase the cost J loses its physical meaning. Figure 5.4 shows the value of $\mathrm{Tr}(QR_{cl})$ as a function of the zero location. Figure 5.5 shows the control parameters as a function of the zero location.

As ε goes to zero, the best a first order compensator can do is put the closed loop poles on the imaginary axis. The cost for such a system becomes infinite and P and Q are unbounded. There is no solution for $\varepsilon = 0$ since P and Q are infinite, but looking at Figure 5.5, one can define a solution at $\varepsilon = 0$ by continuity. Hence a stabilizing solution may continuously become nonstabilizing, as the system parameters are changed and the optimal path followed. The open loop system remains observable and controllable for all α between -0.5 and 0.5. The closed loop poles cannot, however, be arbitrarily assigned.

5.4.4.2 Example 2

In the following example, the poles of the closed loop system can be assigned arbitrarily. Consider the control problem defined by:

$$A = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix},\ B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ R = V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ R_c = V_c = 1,\ n_c = 1,$$

Such a system is made of the aggregation of two completely separate LQG problems for two independent first order systems. One tries to control both systems optimally with a single first order compensator.

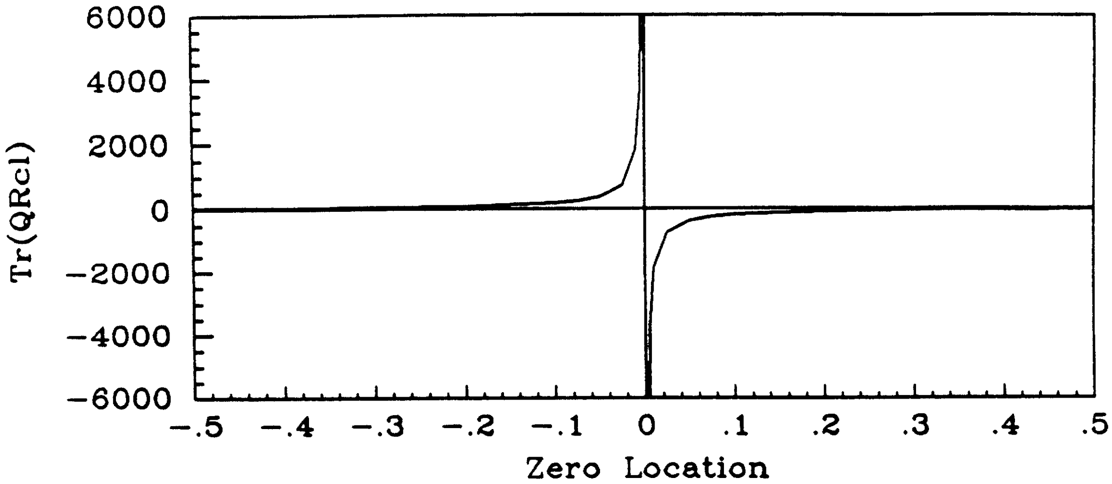

Figure 5.4: Variation of the Optimal Cost as a Function of Zero Location

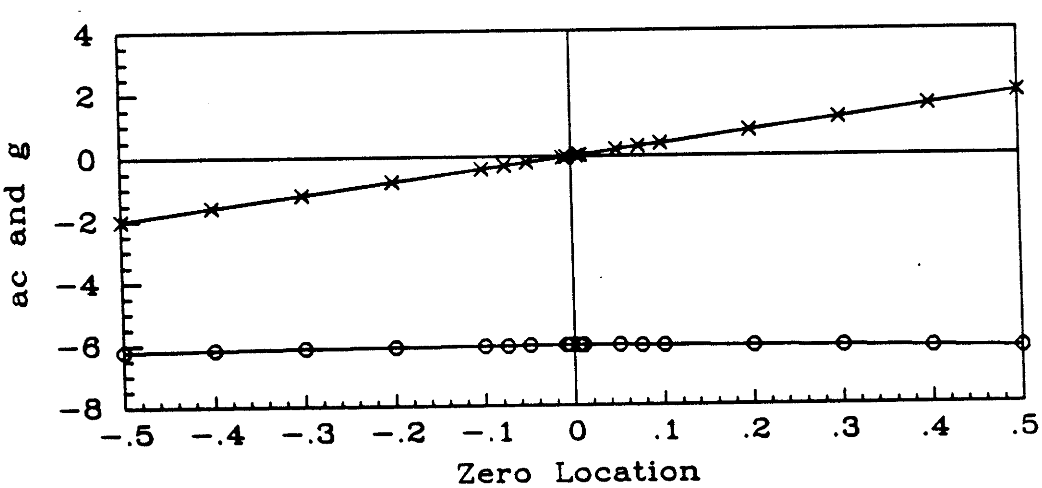

Figure 5.5: Variation of a_c (X) and g (0) as a Function of Zero Location

The projection has two obvious solutions:

$$\tau_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \text{ optimal control of the first subsystem,}$$

ignore the second subsystem,

$$\tau_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \text{ optimal control of the second subsystem,}$$

ignore the first subsystem.

All the optimality conditions can be satisfied with these choices for τ. When a_1 and a_2 are both negative, both solutions yield a stable closed loop. When a_1 is positive, the solution $\tau = \tau_2$, which corresponds to controlling the second subsystem only, becomes unstable. Similarly, if a_2 becomes unstable, $\tau = \tau_1$ yields an unstable closed loop.

Considering the solution $\tau = \tau_2$, the compensator obtained in this solution is independent of a_1. As a_1 becomes positive, the solution, which is constant, will become a nonstabilizing solution. Note, however, that as long as $a_1 \neq a_2$, the closed loop dynamics can be assigned arbitrarily. Choosing a realization such that $k_1 = k_2 = -1$, the closed loop dynamic matrix becomes:

$$A_c = \begin{bmatrix} a_1 & 0 & g_1 \\ 0 & a_2 & g_2 \\ -1 & -1 & a_c \end{bmatrix}$$

The characteristic polynomial is:

$$\varphi(s) = s^3 - (a_1+a_2+a_c)s^2 + (g_1+g_2+a_1a_2+a_c(a_1+a_2))s - g_1a_2 - g_2a_1 - a_1a_2a_c$$

Assume we want the characteristic polynomial to be:

$$\varphi(s) = s^3 + \sigma_2 s^2 + \sigma_1 s + \sigma_0$$

a_c, g_1 and g_2 must then satisfy the following linear system of equations:

$$\begin{bmatrix} a_1 + a_2 + \sigma_2 \\ -a_1a_2 + \sigma_1 \\ -\sigma_0 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ a_1+a_2 & 1 & 1 \\ a_1a_2 & a_2 & a_1 \end{bmatrix} \begin{bmatrix} a_c \\ g_1 \\ g_2 \end{bmatrix}$$

the determinant of the system is equal to $a_2 - a_1$. Hence, as long as the two subsystems do not have the same dynamics, the poles can be freely assigned using a first order dynamic compensator. This does not prevent the solution corresponding to τ_2 to become an unstable closed loop solution as a_1 becomes unstable.

5.4.5 Solutions to the Diagonal Problem

It is claimed in [Ric87, Ric89] that all the diagonal solutions to the reduced order problem can be found for diagonal systems. A diagonal problem is such that:

$$A = \mathrm{diag}(a_1, a_2, \cdots, a_n)$$

$$B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \; B_1 = \mathrm{diag}(b_1, b_2, \cdots, b_m)$$

$$C = [C_1 \; 0], \; C_1 = \mathrm{diag}(c_1, c_2, \cdots, c_l)$$

$$R = \mathrm{diag}(r_1, r_2, \cdots, r_n)$$

$$V = \mathrm{diag}(v_1, v_2, \cdots, v_n)$$

$$R_c = \mathrm{diag}(r_{c1}, r_{c2}, \cdots, r_{cm})$$

$$V_c = \mathrm{diag}(v_{c1}, v_{c2}, \cdots, v_{cl})$$

The problem consists of controlling n completely decoupled first order SISO systems with an n_c dimensional compensator. One obvious solution is to select n_c out of the n subsystems and control them independently using first order controllers. The problem then reduces to solving n_c independent first order LQG problems. The projections τ associated with such solutions are diagonal: $\tau(i, i) = 1$ if a_i is to be controlled, $\tau(i, i) = 0$ otherwise. Such solutions will be called diagonal solutions. Letting n_u be the number of unstable poles, one must control each of these modes in order to have a stable closed loop system. $n_c - n_u$ modes can still be controlled out of $n - n_u$ remaining stable modes. If m and l are larger than n_c, i.e. there are more controllable and observable subsystems than there are compensator modes, the number of such possible solutions corresponds to the combination $\binom{\inf(l,m) - n_u}{n_c - n_u}$ or 1 if the number is not defined.

More solutions may exist, however, even for diagonal problems, as the following examples will show. These solutions cannot be found systematically.

The second order system of Section 5.4.4.2 is considered for different pole locations and for different R and V matrices as well.

First, consider the case where $a_1 = 0.0$ and $a_2 = +0.1$. The open loop system has two unstable poles and cannot be stabilized with a diagonal τ. A first order controller can however stabilize the system. The following optimal solution is found by direct optimization for $R = V = R_c = V_c = I_2$:

$$a_c = -2.1530 \qquad K = [0.6736 \ 1.3294]$$

$$G = \begin{bmatrix} 0.6736 \\ -1.3294 \end{bmatrix} \qquad \tau = \begin{bmatrix} -0.4504 & -0.8082 \\ 0.8082 & 1.4504 \end{bmatrix}$$

The closed loop poles are:

$$\lambda = -0.0450, -1.0038, -1.0042$$

Consider now the plant poles at $a_1 = -0.01$ and $a_2 = +0.1$. This system can be obtained by continuously moving the pole a_1 from its previous value of $a_1 = 0.0$ to its new value of $a_1 = -0.01$. Using the homotopy algorithm shown in Section 5.3, the new solution for $a_1 = -0.01$ and $a_2 = 0.0$ is found to be:

$$a_c = -2.1375 \qquad K = [0.5103 \ -1.2412]$$

$$G = \begin{bmatrix} 0.5103 \\ 1.2412 \end{bmatrix} \qquad \tau = \begin{bmatrix} -0.2636 & -0.5771 \\ 0.5771 & 1.2636 \end{bmatrix}$$

The closed loop poles are:

$$\lambda = -0.0390, -1.0034, -1.0051$$

A second solution for that particular value of a_1 corresponds to a diagonal solution where a_2 is the pole being controlled. This yields:

$$a_c = -2.1100 \qquad K = [0 \ \ -1.0000]$$

$$G = \begin{bmatrix} 0 \\ 1.2210 \end{bmatrix} \qquad \tau = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

The closed loop poles are:

$$\lambda = -0.01, -1.0050, -1.0050$$

Finally, we consider the case where $a_1 = -0.01$ and $a_2 = -0.1$. The weighting matrices R and V are now taken to be $R = V = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$.

A nondiagonal solution was obtained numerically for that problem. The control parameters and projection are:

$$a_c = -3.6802 \qquad K = [-0.7037 \;\; -1.6535]$$

$$G = \begin{bmatrix} 0.7037 \\ 1.6535 \end{bmatrix} \qquad \tau = \begin{bmatrix} 0.1537 & -0.3606 \\ -0.3606 & 0.8463 \end{bmatrix}$$

The open loop system being stable, the two diagonal solutions corresponding to τ_1 and τ_2 are stabilizing solutions as well.

In summary, three examples of diagonal problems have been considered, one with two unstable poles, one with one stable and one unstable pole and one with two stable poles. In each case, it was possible to obtain solutions which are nondiagonal. Hence, even for simple problems, one cannot be certain to find all the solutions to the initial problem in a simple manner. The upper bound proposed in [Ric89] for the maximum number of diagonal solutions to the OPE underestimate the maximum number of stabilizing solutions to the problem. The nondiagonal solutions may also be the only one that connects to a stabilizing solution when all the plant poles are unstable.

5.4.6 Critical Solutions and Bifurcations

Solutions can appear and disappear when P and Q become infinite and a nonstabilizing solution becomes stabilizing and vice versa. A second mechanism for solutions to appear or disappear is when a critical point is encountered along the solution path. Following Definition 5.1, a critical point for the equation $\phi(X) = 0$ is a point at which ϕ_X is singular. In our particular case, this means that the nullspace of the Hessian is not composed only of those directions corresponding to a change in the state space realization of the compensator. If one uses a minimal set of parameters, the reduced Hessian will be column rank deficient if the solution is a critical point. Bifurcations can then occur, as illustrated in the following example.

Consider once more the second order system of Section 5.4.5 with its first order controller and $a_2 = +0.1$. Varying the first pole a_1, we track the diagonal solution corresponding to $\tau = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$. We consider here the entire set of parameters a_c, g_1, g_2, k_1, k_2 and compute the eigenvalues of the Hessian as a function of a_1 for the diagonal solution as a_1 moves toward the right half plane.

Table 5.1: Eigenvalues of the Hessian

a_1	s_1	s_2	s_3	s_4	s_5
-0.0300	0.000	2.020	0.062	3.590	38.32
-0.0280	0.000	0.094	0.062	3.590	43.13
-0.0279	0.000	-0.018	0.062	3.590	43.39
-0.0270	0.000	-1.098	0.062	3.590	45.92
-0.0250	0.000	-4.080	0.0621	3.590	52.48

The first eigenvalue of the Hessian, s_1, is always zero. It corresponds to the freedom in scaling the state variable representing the compensator. The eigenvalue s_2 is positive for $a_1 = -0.0280$ but becomes negative for $a_1 = -0.0279$. The solution starts out as a local minimum but then becomes a saddle point. The critical solution occurs for a value of $a_{1c} = -0.02791583$. Considering the nondiagonal solution found in the previous section for $a_1 = -0.010$ and integrating backward (i.e. reducing a_1), one finds that the solution merges with the diagonal solution at $a_1 = a_{1c}$. The corresponding derivatives of g_1 and k_1 with respect to a_1 become infinite. Figure 5.6 shows a plot of the control parameters as a function of a_1 and clearly shows the bifurcation occurring at $a_1 = a_{1c}$. Figure 5.6a shows the values of G(1) and G(2) as a function of a_1. The diagonal solution corresponds to the optimal control of the second subsystem: for this solution G(1) = 0 and G(2) is constant since the diagonal solution is independent from a_1. As a_1 comes closer to zero the second solution appears. The corresponding G(1) is not zero anymore and G(2) varies as well since the solution couples both of the system modes and the variation of a_1 influences now G(2). Figure 5.6b shows the variations of K(1) and K(2) as a function of a_1, Figure 5.6c shows the variations of a_c. The behavior of a_c and K is similar to that of G. Figure 5.6d shows the optimal cost for each solution. The cost of the diagonal solution rapidly increases as a_1 moves toward the right half plane and the closed loop system becomes unstable.

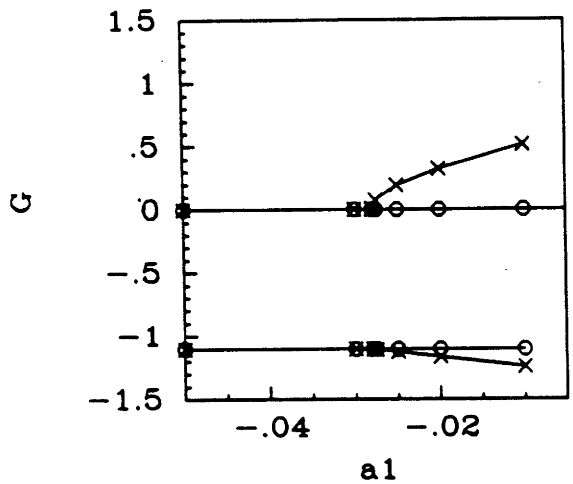

Figure 5.6a: Optimal Gain G
as a Function of a_1

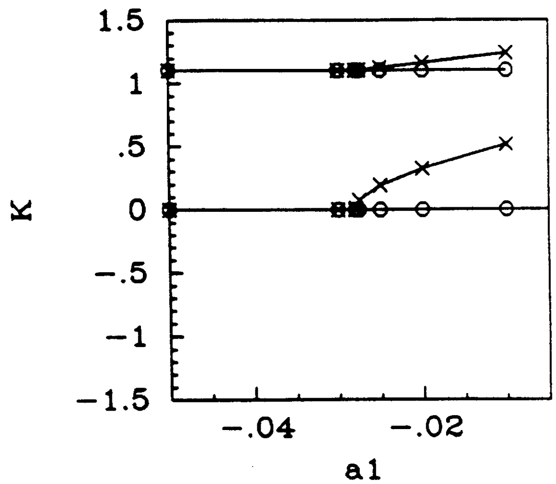

Figure 5.6b: Optimal Gain K
as a Function of a_1

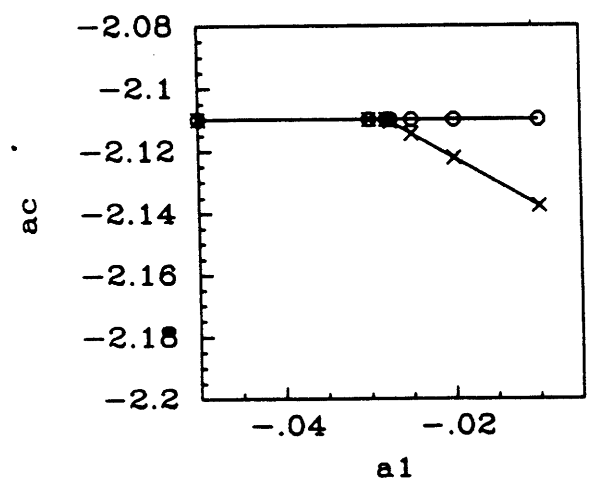

Figure 5.6c: Optimal Pole a_c
as a Function of a_1

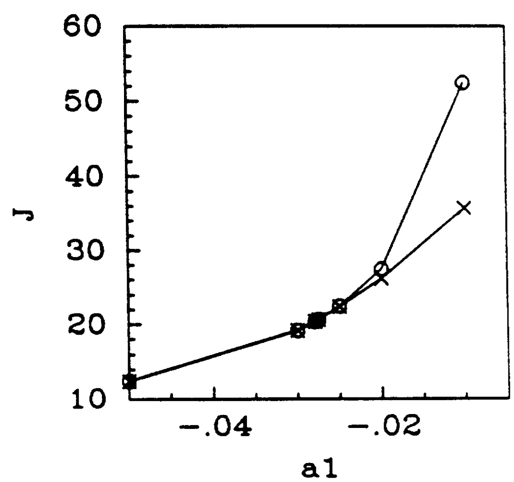

Figure 5.6d: Optimal Cost
as a Function of a_1

5.4.7 Multiple Local Minima

The following example shows multiple stabilizing solutions and, in particular, the occurrence of multiple local minima. The number of sensors and actuators is strictly smaller than the order of the compensator in this example, unlike the examples of multiple solutions previously shown. Consider the fourth order SISO system:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -2 & -0.1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1-\alpha \\ 0 \\ \alpha \end{bmatrix}$$

$$C = [\ 1 \quad 0 \quad 1 \quad 0\ ]$$

Take the LQG parameters to be:

$$R = V = I_4,$$
$$R_c = V_c = 1$$

A second order compensator is sought for the problem ($n_c = 2$). A solution $\mathcal{S}_0$ is found numerically for $\alpha = 0$ and a solution $\mathcal{S}_1$ is found numerically for $\alpha = 1$. The solution $\mathcal{S}_0$ is integrated from $\alpha = 0$ forward and the solution $\mathcal{S}_1$ is integrated from $\alpha = 1$ backward. For $\alpha = 0.055$ two solutions are found. From the forward integration, one gets:

$$A_c = \begin{bmatrix} -0.6905 & 0.4230 \\ -1.3465 & -0.6744 \end{bmatrix}, \qquad K = \begin{bmatrix} 0.5652 \\ -0.1005 \end{bmatrix}$$

$$G = [\ 0.0621 \ -0.5307], \qquad J = 29.3008$$

The eigenvalues of the corresponding Hessian are:

$$46.6288,\ 13.6840,\ 1.7978,\ 0.0771,\ 0.0,\ 0.0,\ 0.0,\ 0.0,$$

From the backward integration one gets the following solution:

$$A_c = \begin{bmatrix} -2.2471 & -0.9574 \\ -1.5384 & -2.1382 \end{bmatrix}, \qquad K = \begin{bmatrix} 1.2721 \\ 0.1405 \end{bmatrix}$$

$$G = [-0.1802 \ \ 0.1405], \qquad J = 29.3030$$

The eigenvalues of the corresponding Hessian are:

47.7989, 5.8213, 0.3235, 0.0015, 0.0, 0.0, 0.0, 0.0

Each Hessian has four zero eigenvalues that correspond to the freedom in selecting the state space representation of a second order transfer function. Notice also that the fourth eigenvalue becomes very small in both cases. Both solutions obtained are local minima.

As α is increased, the solution $\mathcal{S}_0$ encounters a critical point and becomes a saddle point, even though it still is a stabilizing compensator. Similarly, as α is decreased the solution $\mathcal{S}_1$ encounters a critical point and becomes a saddle point. At $\alpha = 0.055$, two local minima exist. The open loop system is controllable, observable and stable. Hence, the number of solutions to the OPE exceeds the upper bound given in [Ric87], even if only the minima are considered. Table 5.2 summarizes the characteristics of the two compensators for $\alpha = 0.055$.

Table 5.2: Compensator Characteristics, $\alpha = 0.055$

	Forward Solution	Backward Solution
Closed-loop Poles	$-0.4769 \pm j\ 0.5188$ $-0.2509 \pm j\ 1.0592$ $-0.0546 \pm j\ 1.4135$	-3.4983 -0.4864 $-0.2450 \pm j\ 1.0053$ $-0.0554 \pm j\ 1.4128$
Compensator Poles	$-0.6825 \pm j\ 0.7547$	-0.9778 -3.4075
Compensator Zero	-5.2212	3.8647

Even though the optimal costs are very close, the compensators are of very different natures. The first one is minimum phase and has two oscillatory poles close to the first mode of the system. The second compensator has a nonminimum phase zero, one slow real pole and one fast real pole. As α changes, one can vary each compensator in order to leave the cost stationary but, as α is varied, one compensator structure ceases to yield a minimum.

5.4.8 Accommodating Critical Points: Software Modification

Looking at the example of Section 5.4.6, one can see that the Hessian matrix has two eigenvalues equal to zero at the critical value $a_1 = a_{1c}$. One of the eigenvalues is the predicted singularity, the second one however characterizes the critical solution. The derivatives of the control parameters with respect to a_1 are not well defined at that particular solution. When a_1 is smaller than a_{1c}, the diagonal solution corresponds to a local minimum. When a_1 is larger than a_{1c}, however, the diagonal solution is a saddle point and the nondiagonal solution is the local minimum. If one tries to track the optimal solution as a_1 is varied from the left to the right of its critical value, one can detect the proximity of the critical point by checking the rank of the Hessian. One can then decide not to rely on the gradient ξ_α which may be ill-conditioned or not defined at all, and simply increment the value of a_1. The minimization step should then find the solution corresponding to the local minimum and abandon the diagonal solution to follow the nondiagonal solution. The solution becomes noncontinuous but remains valid for all values of a_1. More generally, The modification to the shooting step is the following:

Step 2.1: Compute $L_{\xi\xi_d}$

Step 2.2: If $L_{\xi\xi_d}$ has exactly N_d–N_m zero eigenvalues, compute ξ_{α_k}
Else, set $\xi_{\alpha_k} = 0$

N_d–N_m is the number of extra degrees of freedom left in ξ_d. Whenever a critical point is encountered along the solution path, the shooting step does not rely on the gradient ξ_α anymore, for ξ_α may be ill-conditioned or may not exist at all. In that case, one simply increments the homotopy parameter α and relies on the minimization routine to find the solution corresponding to a minimum. The procedure guarantees the convergence toward a minimum.

5.4.9 Conclusion

The optimality conditions for the LQG problem have multiple algebraic solutions. When the form of the controller is not constrained, the problem has the property that only one of the solutions stabilizes the plant and corresponds to a minimum for the cost. When the order of the compensator is reduced, however, this section has shown that the property is not valid anymore. Multiple stabilizing solutions can occur, corresponding to local minima or saddle points for the cost. Similarly, cases occur where no stabilizing solution exists. The section has also shown that the nature of a solution is not invariant under homotopy: minima can become saddles, stabilizing solutions can become nonstabilizing. These changes occur, however, in a smooth continuous fashion. All solutions, stabilizing, nonstabilizing, minima, saddle points cannot be obtained in any systematic manner even for simple problems like diagonal problems. Hence, the homotopy procedure will be guaranteed to track the global optimum only locally. Any global results require the tracking of <u>all</u> solutions.

5.5 PRACTICAL APPLICATION OF THE CONTINUATION METHOD

5.5.1 General

We consider in this section a practical application to illustrate the numerical problems that can arise when using a continuation method. The example is drawn from [Ber87b]. The system to control is made of a pair of simply supported Euler-Bernoulli beams connected by a spring. The system is depicted in Figure 5.7. Each beam has one rate sensor and one force actuator. Two vibrational modes are retained to describe each beam and the state space representation of the system is an eighth order interconnected model. The expression for the A, B and C matrices have been derived in [Ber87b]. There are:

$$
\begin{aligned}
&A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},\ B_1 = \begin{bmatrix} B_{11} \\ 0_{4x1} \end{bmatrix},\ B_2 = \begin{bmatrix} 0_{4x1} \\ B_{22} \end{bmatrix} \\
&C_1 = [C_{11}\ 0_{1x4}],\ \ C_2 = [0_{1x4}\ C_{22}]
\end{aligned}
\tag{5.5.1}
$$

where

$$A_{ii} =$$

$$
\begin{bmatrix}
0 & \omega_{1i} & 0 & 0 \\
-\omega_{1i}-(k/\omega_{1i})(\sin\pi c_i)^2 & -2\zeta_i\omega_{1i} & -(k/\omega_{2i})(\sin\pi c_i)(\sin 2\pi c_i) & 0 \\
0 & 0 & 0 & \omega_{2i} \\
-(k/\omega_{1i})(\sin\pi c_i)(\sin 2\pi c_i) & 0 & -\omega_{2i}-(k/\omega_{2i})(\sin 2\pi c_i)^2 & -2\zeta_i\omega_{2i}
\end{bmatrix}
$$

$$A_{ij} =$$

$$
\begin{bmatrix}
0 & 0 & 0 & 0 \\
(k/\omega_{1j})(\sin\pi c_i)(\sin\pi c_j) & 0 & (k/\omega_{2j})(\sin\pi c_i)(\sin 2\pi c_j) & 0 \\
0 & 0 & 0 & 0 \\
(k/\omega_{1j})(\sin\pi c_j)(\sin 2\pi c_i) & 0 & (k/\omega_{2j})(\sin 2\pi c_i)(\sin 2\pi c_j) & 0
\end{bmatrix}
$$

$$B_{ii} = \begin{bmatrix} 0 \\ -\sin\pi a_i \\ 0 \\ -\sin 2\pi a_i \end{bmatrix}, \quad C_{ii} = [\, 0 \ \sin\pi s_i \ 0 \ \sin 2\pi s_i]$$

where ω_{ij} and ζ_{ij} are respectively the j^{th} modal frequency and damping ratio of the i^{th} beam, k is the spring constant, c_i is the position of the spring attachment, a_i the actuator location and s_i the sensor location on the i^{th} beam, all distances being non dimensionalized by the beam length.

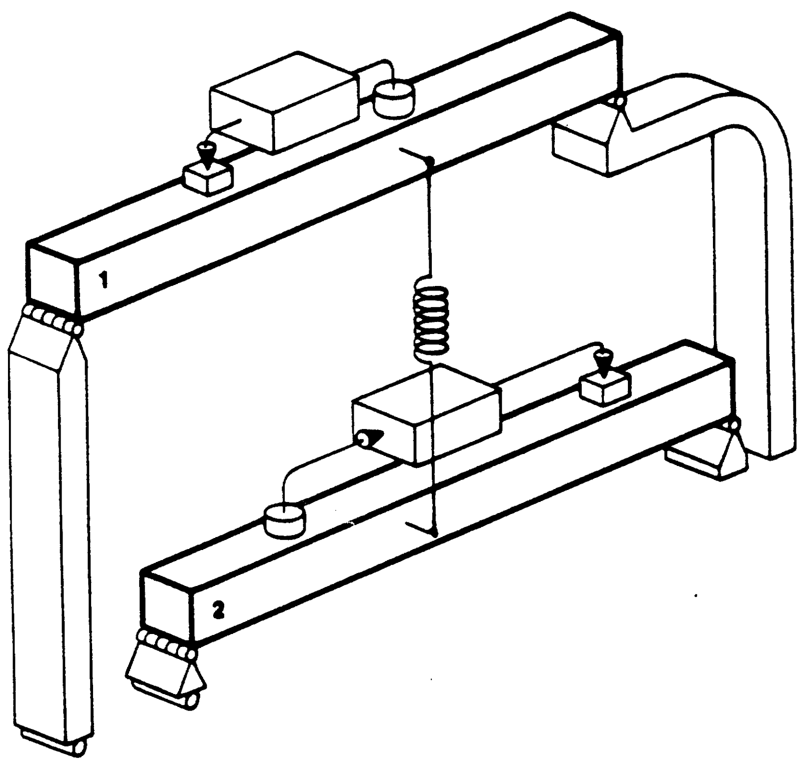

Figure 5.7: The two beam System of Bernstein

The parameters quoted in [Ber87b] are:

$$
\begin{aligned}
&w_{1i} = 1 \text{ rad/sec}, \ w_{2i} = 4 \text{ rad/sec}, \ \zeta_i = 0.0050 \\
&a_1 = 0.3, \ \ s_1 = 0.65, \ \ c_1 = 0.6 \\
&a_2 = 0.8, \ \ s_2 = 0.2, \ \ \ c_2 = 0.4
\end{aligned}
$$

The penalty on the states is given by:

$$
R = \text{blockdiag}\left[\begin{bmatrix}1 & 0 \\ 0 & 1/\omega_{11}\end{bmatrix}, \begin{bmatrix}1 & 0 \\ 0 & 1/\omega_{21}\end{bmatrix}, \begin{bmatrix}1 & 0 \\ 0 & 1/\omega_{12}\end{bmatrix}, \begin{bmatrix}1 & 0 \\ 0 & 1/\omega_{22}\end{bmatrix}\right]
$$

$$
V = \text{blockdiag}\left[\begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix}\right] \tag{5.5.2}
$$

$$
R_c = 0.1\, I_2, \ \ V_c = 0.1\, I_2
$$

The controller consists of two decentralized 4th order compensators, each of which uses the sensor and the actuator of one beam only.

5.5.2 Sequential Design

The optimization technique used in [Ber87b] consists of sequentially optimizing each compensator while the design of the remaining compensator is frozen and the corresponding loop is closed. The compensator that is optimized becomes at each step the optimal reduced order compensator for the system consisting of the original plant with the remaining compensator loops closed. The initial controllers are chosen to be the controllers obtained when the interconnection is ignored, i.e. when $k = 0$, in which case the problem decouples into two independent 4th order LQG problems. The results were reproduced using only the minimization part of the algorithm developed in Section 5.3.2 which yielded very satisfactory results in that case. Table 5.3 summarizes the results.

Table 5.3: Cost during Sequential Optimization

Design	Cost
Open Loop	163.2969
Full–Order LQG	11.0795
Suboptimal Decentralized	
assuming k = 0	29.4544
Redesign Subcontroller 1	14.5104
Redesign Subcontroller 2	12.4934
Redesign Subcontroller 1	12.0204
Redesign Subcontroller 2	11.9641
Redesign Subcontroller 1	11.9501
Redesign Subcontroller 2	11.9465
Redesign Subcontroller 1	11.9455
Redesign Subcontroller 2	11.9452

Note that there is a discrepancy between these results and those of [Ber87b] even for the full order LQG design. We attribute this to a mismatch between the parameters and the results quoted in the paper. The conclusions and basic behavior of the design procedure shown in the paper remain valid, however. Checking the optimality conditions simultaneously including both controllers, one finds that the error is equal to 3.9 10^{-3} after eighth redesigns. The cost, however, coincides already with the optimal cost to the first five significant figures.

5.5.3 Using Homotopy: a Continuous Solution Path

The two beams treated in the example are identical and the attachment points of the spring are symmetric with respect to the middle of each beam since $c_1 = 0.40$ (40% of the length from the left of the first beam) and $c_2 = 0.60$ (60% of the length from the left of the second beam). The interconnected system possesses therefore two modes which are independent of k, the first one at $\omega = 1$ rad/sec corresponding to the first bending modes of the beams oscillating in phase and the second one at $\omega = 4$ rad/sec corresponding to the second bending modes of the beams oscillating with opposite phase so that the spring is not stretched at any time. The remaining modes of the interconnected system depend strongly on the value of k. We try to use this property in order to find a simpler problem as a starting point. Transforming the system Eq.(5.5.1) in modal form and transforming accordingly the design parameters of Eq.(5.5.2), the terminal parameter values to use in the homotopy are:

$$A_1=\begin{bmatrix} -0.0050 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.0000 & -0.0050 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0200 & 3.9999 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3.9999 & -0.0200 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0112 & 3.1077 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3.1077 & -0.0112 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0138 & 5.6870 \\ 0 & 0 & 0 & 0 & 0 & 0 & -5.6870 & -0.0138 \end{bmatrix}$$

$$B_1=\begin{bmatrix} 0.0001 & -0.4523 & 0.0044 & -0.5316 & -0.0009 & 0.3443 & 0.1020 & -0.0035 \\ 0.0001 & -0.3286 & 0.0044 & -0.5316 & 0.0008 & -0.2970 & -0.1730 & 0.0040 \end{bmatrix}^T$$

$$C_1=\begin{bmatrix} -0.0041 & 0.7969 & 0.0096 & -0.7236 & 0.0060 & -0.2850 & 1.1992 & -0.0116 \\ -0.0027 & 0.5257 & 0.0113 & -0.8506 & -0.0023 & -0.2934 & -1.1118 & 0.0096 \end{bmatrix}$$

$$R_1=\begin{bmatrix} 1.6001 & -0.0080 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0080 & 1.6000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.6001 & 0.0080 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0080 & 0.4001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.7962 & -0.0298 & -0.0002 & -1.5047 \\ 0 & 0 & 0 & 0 & -0.0298 & 4.4039 & -1.3293 & 0.0221 \\ 0 & 0 & 0 & 0 & -0.0002 & -1.3293 & 1.1356 & -0.0078 \\ 0 & 0 & 0 & 0 & -1.5047 & 0.0221 & -0.0078 & 0.6103 \end{bmatrix}$$

$$V_1=\begin{bmatrix} 0.0000 & -0.0001 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0001 & 0.6250 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0000 & -0.0052 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0052 & 0.6249 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0000 & -0.0003 & -0.0006 & 0.0000 \\ 0 & 0 & 0 & 0 & -0.0003 & 0.1563 & 0.0000 & -0.0010 \\ 0 & 0 & 0 & 0 & -0.0006 & 0.0000 & 0.4946 & -0.0057 \\ 0 & 0 & 0 & 0 & 0.0000 & -0.0010 & -0.0057 & 0.0001 \end{bmatrix}$$

$$R_{c1}=\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad V_{c1}=\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

In order to get a simple initial solution, we need an initial problem that decouples into two fourth order LQG problems. Looking at the modal form of the problem, one can see that the only coupling comes from B_1 and C_1. In order to split the system into two subsystems we choose the following initial parameters:

$$B_0=\begin{bmatrix} 0.0001 & -0.4523 & 0.0044 & -0.5316 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0008 & -0.2970 & -0.1730 & 0.0040 \end{bmatrix}^T$$

$$C_0=\begin{bmatrix} -0.0041 & 0.7969 & 0.0096 & -0.7236 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0023 & -0.2934 & -1.1118 & 0.0096 \end{bmatrix}$$

$$\begin{aligned} A_0 &= A_1 \\ R_0 &= R_1 \\ V_0 &= V_1 \\ R_{c0} &= R_{c1} \\ V_{c0} &= V_{c1} \end{aligned}$$

The parameters are then continuously deformed following Eqs.(5.2.1-7) and the initial solution is integrated forward as a function of the homotopy parameter α. Table 5.4 summarizes the steps of the integration.

The shooting accuracy ε_2 is set initially to 0.1 . When the number of minimization steps is less than 6 the shooting accuracy is doubled in order to take larger steps. This strategy allows for large shooting steps. The accuracy on the solution during the integration has been set to $\varepsilon_0 = 10^{-4}$. Far better accuracies are attained, however. This is due to the fact that one extra iteration during the minimization process can bring the error down by 2 or 3 orders of magnitude especially if one is within the quadratic convergence region. The optimal cost J varies very smoothly as a function of α which explains the success of the homotopy. The controllers obtained at $\alpha = 1$ are given in Appendix C.

Table 5.4: Summary of Homotopy Procedure

α	Shooting Er. Minimized Er.	Shooting Cost Minimized Cost	Minim. Steps
0.100000	$0.91\ 10^{-1}$ $0.62\ 10^{-4}$	10.2082 10.1979	4
0.287500	$0.21\ 10^{-0}$ $0.59\ 10^{-5}$	10.6929 10.6367	7
0.392969	$0.22\ 10^{-0}$ $0.53\ 10^{-4}$	10.9170 10.9019	5
0.486719	$0.38\ 10^{-0}$ $0.66\ 10^{-5}$	11.1529 11.1112	6
0.557031	$0.39\ 10^{-0}$ $0.39\ 10^{-6}$	11.3005 11.2694	6
0.609766	$0.41\ 10^{-0}$ $0.75\ 10^{-4}$	11.4083 11.3756	6
0.703516	$0.40\ 10^{-0}$ $0.67\ 10^{-4}$	11.5967 11.5512	7
0.797266	$0.37\ 10^{-0}$ $0.19\ 10^{-6}$	11.7248 11.7072	6
0.903696	$0.40\ 10^{-0}$ $0.91\ 10^{-7}$	11.8683 11.8529	6
1.000000	$0.14\ 10^{-0}$ $0.18\ 10^{-6}$	11.9534 11.9450	4

5.5.4 Using Homotopy: a Discontinuous Solution Path

The system formed by the two beams is naturally decoupled when the stiffness of the interconnecting spring is zero, or in other words when the spring is removed. A very natural approach is to continuously increase the stiffness of the spring from $k = 0$ to $k = 10$. If one looks at the form of A in Eq.(5.5.1), one can see that it can be written as:

$$A = A_0 + k\Delta A$$

where, for the particular parameters chosen,

$$A_0 = \begin{bmatrix} 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.0000 & -0.0100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4.0000 & -0.0400 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.0000 & -0.0100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.0000 \\ 0 & 0 & 0 & 0 & 0 & 0 & -4.0000 & -0.0400 \end{bmatrix}$$

$$\Delta A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9.0451 & 0 & 1.3975 & 0 & 9.0451 & 0 & 1.3975 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5.5902 & 0 & -0.8637 & 0 & -5.5902 & 0 & -0.8637 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9.0451 & 0 & -1.3975 & 0 & -9.0451 & 0 & -1.3975 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5.5902 & 0 & -0.8637 & 0 & -5.5902 & 0 & -0.8637 & 0 \end{bmatrix}$$

The homotopy parameter α defined in Section 5.2.1 simply becomes $k/10$, where k is the stiffness of the currently deformed system. The remaining parameters of the problem B, C, R, V, R_c and V_c need not be changed since they naturally have the correct block diagonal structure.

The initial compensators are the two fourth order LQG solution to the problem with $k = 0$. Freezing the initial controllers and varying the stiffness of the spring, one finds that they stabilize the system for $0 \leq k \leq 2.8395$ and $k \geq 7.7310$. They do provide a stabilizing solution for $k = 10$ and constitute the starting point of the sequential design of [Ber87b]. However, the closed loop system is unstable whenever $2.8395 < k < 7.7310$ and the initial controllers are used. This indicates that the solution has a different character for small and large k. Starting the homotopy at $\alpha = 0$ with the initial decentralized solution one does indeed encounter a critical point at $\alpha = 0.5724$, or $k = 5.724$. At that point the algorithm cannot keep tracking the solution. The minimization step converges instead toward a very different solution. This step consumes most of the run time since the minimization starts with an initial solution that may be quite different in character to the minimum. Once the software has found a new local minimum, it can resume the forward integration starting with the new solution. As k reaches 10 the solution obtained is the same as the one found in the previous subsection using a different initial simplified problem. The solution at $k = 10$ was subsequently used as a starting point in order to do the homotopy backward and reduce the spring constant from 10 to 0. We will call the forward path the path of solutions obtained from the decoupled solution at $k = 0$ as k is increased and the backward path the path of solutions starting at $k = 10$ as k is decreased. As k keeps decreasing, the backward path also encounters a critical point: the integration cannot go pass $k = 1.795$ and the minimization then converges on the solution that is on the forward path. There is a whole range of values for k, between 1.795 and 5.724, where the system admits multiple solutions. The two solutions obtained for $\alpha = 0.57$ ($k = 5.7$) are as follows:

Compensators Dynamics from Forward Integration

$$A_{c1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -70.2930 & -287.3375 & -79.1406 & -11.2272 \end{bmatrix}, \; K_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -16.8490 & 525.6644 & -24.1238 & 27.0323 \end{bmatrix}$$

$$A_{c2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -23.9153 & -228.7647 & -64.7072 & -11.8874 \end{bmatrix}, \; K_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -48.5721 & 27.0888 & -66.0029 & 1.1453 \end{bmatrix}$$

$$G = \begin{bmatrix} G_1 & 0_{1\times4} \\ 0_{1\times4} & G_2 \end{bmatrix}, \; K = \begin{bmatrix} K_1 & 0_{4\times1} \\ 0_{4\times1} & K_2 \end{bmatrix}$$

Cost $J = 10.4692$

Compensators Dynamics from Minimization

$$A_{c1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -110.3980 & -297.0714 & -80.6289 & -11.4036 \end{bmatrix}, \quad K_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 36.6532 & 518.1627 & -21.6314 & 26.7397 \end{bmatrix}$$

$$A_{c2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -8.1725 & -20.2980 & -28.1007 & -3.5422 \end{bmatrix}, \quad K_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -12.0068 & 9.7086 & -12.1283 & -3.3435 \end{bmatrix}$$

$$G = \begin{bmatrix} G_1 & 0_{1\times 4} \\ 0_{1\times 4} & G_2 \end{bmatrix}, \quad K = \begin{bmatrix} K_1 & 0_{4\times 1} \\ 0_{4\times 1} & K_2 \end{bmatrix}$$

Cost $J = 10.4459$

The compensators are written in Controller Canonical form for easier comparison. The parameters are quite far apart between the two solutions. Both solutions satisfy the optimality conditions with an accuracy better than 10^{-10}. They yield a stable closed loop system implying that the matrices P and Q are non negative and correspond to minima. Figure 5.8 compares the closed loop poles obtained with the two different solutions.

Table 5.5 summarizes the results of the forward and backward integrations. The solution is unique at $k = 0$ and also appears to be unique at $k = 10$, as only one solution was found using either homotopy or direct minimization. Appendix C regroups the controllers obtained for various values of k in the forward or the backward integration.

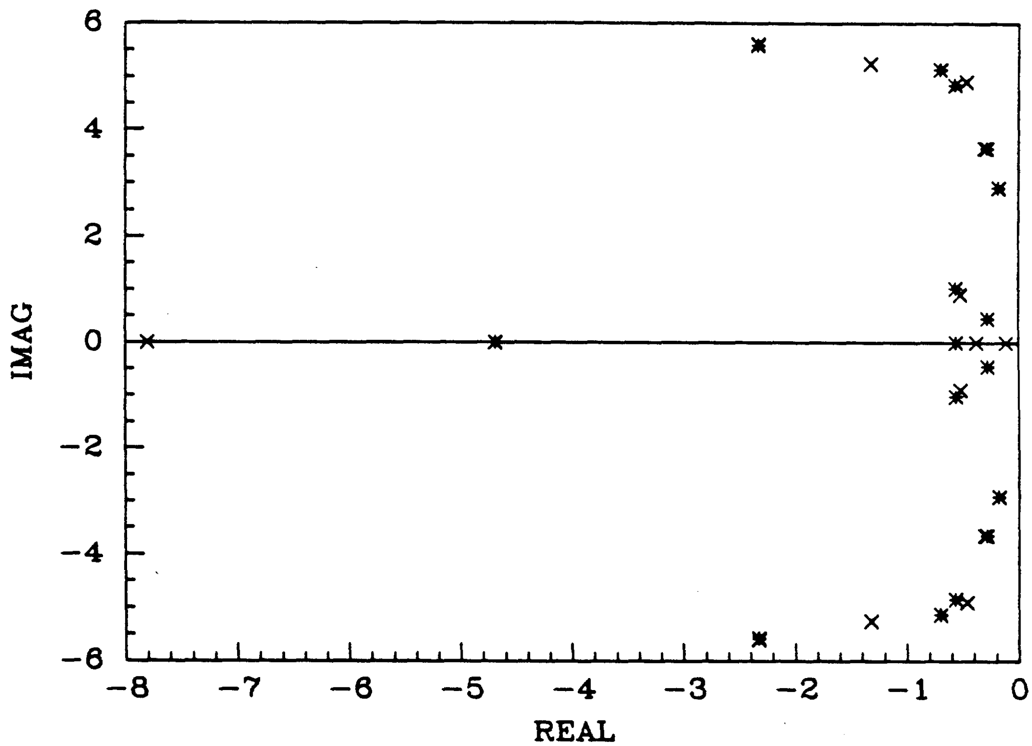

Figure 5.8: Comparison of the Closed Loop for k = 5.7
×: Forward and ∗: Backward Integration

Table 5.5: Cost on Forward and Backward Integration

α	Forward Integration Cost	Backward Integration Cost
0.0000	7.8926	*.*
0.1000	7.7483	*.*
0.1800	8.0781	8.1654
0.2000	8.1816	8.2587
0.3000	8.7914	8.8221
0.4000	9.4692	9.4779
0.5000	10.0875	10.0787
0.5700	10.4692	10.4459
0.5724	10.4816	10.4581
0.6000	*.*	10.5959
0.7000	*.*	11.0844
0.8000	*.*	11.5486
0.9000	*.*	11.8875
1.0000	*.*	11.9450

5.5.5 Dicussion

When the stiffness of the spring is large enough, one can recognize two types of modes in the composite system. The first type can be qualified as "group modes". It corresponds to the displacement in phase of the two beams in such a way that the spring is never elongated. This is made possible by the fact that both beams have similar dynamics. The second type of modes can be qualified as "spring modes". It corresponds to the motion of the two beams that will elongate the spring. When the spring stiffness is low, the dynamics of the composite system consist also of two types of modes. The first type, in that case, is associated with the motion of the first beam, and the second one to that of the second beam. The nature of the control obtained for small values of k corresponds mostly to the control of the beam modes since the only solution is a continuous deformation of the uncoupled solution obtained at $k = 0$. For large values of k, however, the control separates into the control of the spring the group modes since it is found by continuously deforming the compensator obtained by assuming that the sensors and actuators are such that group and spring modes can be estimated and controlled separately, while the stiffness k is unchanged, $k = 10$ (Section 5.5.3). For intermediate values of k, the situation is not clear and two types of control subsist, one connecting to the independent beam control at $k = 0$ and one connecting to the spring and group mode control at $k = 10$. The homotopy provides some insight in the physical meaning of the controller even when the nature is harder to identify. It appears that the initial problem must have a controller of the same nature as the final problem if one wants to track the solution. There may be more than one stabilizing compensator corresponding to a local minimum, which would mean that one has not identified the right architecture for the controller.

5.6 CONCLUSION

Continuation methods are reported more successful at solving complex control design problems than the direct optimization methods [Ric87, Hyl90]. A continuation algorithm has been derived in this chapter in order to solve the fixed architecture H_2 Optimal Control Problem. The algorithm combines a simple integration scheme with a minimization routine in order to control the error of the solution. The minimization scheme uses modified gradient steps along with modified Newton-Raphson steps. The modifications of the minimization methods are necessary in order to deal with the singularity of the problem since it admits families of solutions corresponding to the different state space realizations of the same transfer function.

A central issue in the use of homotopy is to determine the number of connecting solution paths between two problems as well as the possibility to find all solutions to simpler problems. It was found in this chapter that there is no systematic way to find all solutions to the constrained LQG problem even for diagonal problems. It was also found that the control problem admits multiple solutions of various natures. Some solutions correspond to nonstabilizing controllers while others yield a stable closed loop. Among the stabilizing solutions, some correspond to minima whereas others are saddles. Unfortunately, the nature of a solution is not invariant along a solution path: minima can become saddles and stabilizing solutions can continuously become nonstabilizing controllers. One must, therefore, determine all solutions, and not only the minima, and track all of them in order to have a global tool for solving the contrained LQG problem. One alternative is to rely on a minimization routine and follow noncontinuous solution paths when a critical point is found along the way. The success of the homotopy will strongly depend on how close the initial and the final problems are related. Take, for example, a system consisting of p weakly coupled subsystems. A natural architecture for that problem is to control each subsystem independently. A good starting solution can be obtained by setting the coupling terms to zero and to solve p independent LQG or reduced order LQG problems. One can then interpret the homotopy parameter α as being some norm of the coupling terms. Of course, if one keeps increasing the coupling terms to a point where the subsystems become strongly coupled, the decoupled solution does not bear any of the characteristics of the solution corresponding to the strongly coupled solution and one might expect the weak solution to vanish and a bifurcation to occur. A clear example of such an occurrence was demonstrated using an increasingly coupled set of beams controlled independently. Similarly, if the noise and the penalty matrices tend to couple the subsystems, one can expect to encounter difficulties in following the solution.

The algorithm can accommodate critical points along the solution path and track solutions corresponding to minima. This property was also demonstrated on the coupled beam problem. However, the algorithm can still fail to find a solution, whether it is because there is no solution, or because one has not found a path that connects to it. The selection of the architecture will play an important role for the success of the solution algorithm.

EXAMPLES OF FIXED ARCHITECTURE DESIGNS

6.1 INTRODUCTION

Some solutions to the fixed architecture problems have been already shown, which have been obtained with the numerical algorithm presented in Chapter 5. A broader validation is proposed in this chapter and consists of deriving reduced order compensators for the four disk drive system of Enns [Enn84]. This example was used in [Liu86] as a testbed for different compensator order reduction techniques, and it was also used in [Ric87, Hyl90] to validate the Optimal Projection Equations approach to find reduced order controllers. Enns' system is a flexible shaft supporting four dissimilar disks. A torque is applied to the first disk while the motion of the third disk is measured. Such a system is unstable and nonminimum phase. Reduced order controllers of order 2 to 6 are to be generated for increasing level of disturbance noise affecting the plant.

Once the confidence in the software abilities has been raised, more realistic problems involving decentralized controllers can be tried. Two lightly damped flexible structures were selected to provide a testbed for the design method. Both are fully instrumented experimental articles developed at the NASA Langley Research Center. Models, as well as specifications for the hardware components were available for both systems. The designs were, therefore, based on actual performance considerations and took into account hardware limitations such as maximum actuator authorities and noise levels, all given by the component specifications. The first experiment is the Mini-Mast, a 20 meter long truss structure which has been manufactured and assembled with space flight tolerances. The mast has a triangular section inscribed in a 1.4 meter diameter circle and is cantilevered at its base. Three torque wheels serve as the principal actuators at the top of the structure. Noncontacting sensors monitor the displacement of the truss vertices. The structure is an ideal testbed for performing vibration suppression experiments.

The second experiment is the SCOLE, or Spacecraft Control Laboratory Experiment. The article duplicates the dynamics of a composite satellite made of a large mass/inertia module (i.e the space shuttle) connected to a small mass/inertia module (i.e an antenna reflector) by a long flexible mast. The shuttle is simulated by a 500 pounds steel plate with appropriately scaled moments of inertia and is suspended by a single cable mounted to a universal joint near the center of gravity of the system. The reflector is connected to the shuttle by a stainless steel, 120 inch long tube and hangs down in order to reduce unnecessary loads. The reflector is a 24 inch side hexagon and is positioned horizontally in a nonsymmetric fashion relative to the shuttle. The reflector and mass both weigh around 5 pounds with no sensor and no actuator. Aircraft quality rate sensors and accelerometers are located both on the shuttle and on the reflector. The Line of Sight (LOS) pointing of the reflector is the typical control problem to investigate, where the flexibility and mass/inertia mismatch will naturally lead to problems of control/structure interaction.

The models used in the examples capture the main features of the systems they describe even though they may not represent the most current configurations of the experiments. The fixed architecture controllers obtained in the chapter will realistically illustrate the benefits and shortfalls of the approach. Designs will be compared to the unconstrained full order controllers in order to understand the implication of reducing the order and constraining the information flow.

6.2 DIRECT METHOD VERSUS INDIRECT METHOD: ENNS' EXAMPLE

6.2.1 Enns Four Disk Drive

The four disk system considered by Enns is an experiment originally developed at Stanford University [Enn84]. Its purpose was to study robust control designs. The four disk drive consists of a shaft whose torsional stiffness is small enough so that the system has slow lightly damped oscillatory poles. The control is performed by a torque motor connected to the first disk while a tachometer measures the rotation of the third disk. The system can be modeled with an eighth order transfer function. The transfer function has two poles at the origin if one assumes the shaft to be perfectly balanced and the bearings to be frictionless. The plant is therefore unstable and, because the sensor and the actuator are not collocated, it also happens to be nonminimum phase. Its nominal transfer function is:

$$G(s)=\frac{0.01(0.64s^5+0.235s^4+7.13s^3+100.02s^2+10.45s+99.55)}{s^2(s^6+0.161s^5+6.004s^4+0.5822s^3+9.985s^2+0.4073s+3.982)}$$

The uncertainty in the plant is introduced by allowing some mismatch between the inertia of the disks. Stability was found to be guaranteed as long as the loop transfer function had a shape contained in a region shown in Figure 6.1 [Enn84]. The loop transfer function properties can be obtained by using a full order LQG compensator when the problem parameters are properly chosen.. The work undertook by Enns was to reduce the order of the full order controllers found to meet the robustness constraints with a compensator reduction technique of his own. The work presented in [Liu86] was to extend the comparison and, considering the same full order controllers, reduce their order using various order reduction techniques, including a method of their own. Such order reduction methods are indirect since the design procedure always consists of finding a full order controller first, and then reducing it to meet the order constraints. Indirect methods are shown to fail to stabilize the plant in many cases [Liu86], especially when the disturbance entering the plant is high.

[Ric87, Hyl90], on the other hand, consider the LQG problems that generate the full order compensators that meet the robustness requirements and directly finds the reduced order controllers that solve the optimization problem. Direct methods were found to provide stabilizing compensators in all cases and the designs were extended to cases with much higher levels of disturbances entering the plant. The same reduced order LQG problems are to be solved with the newly developed software.

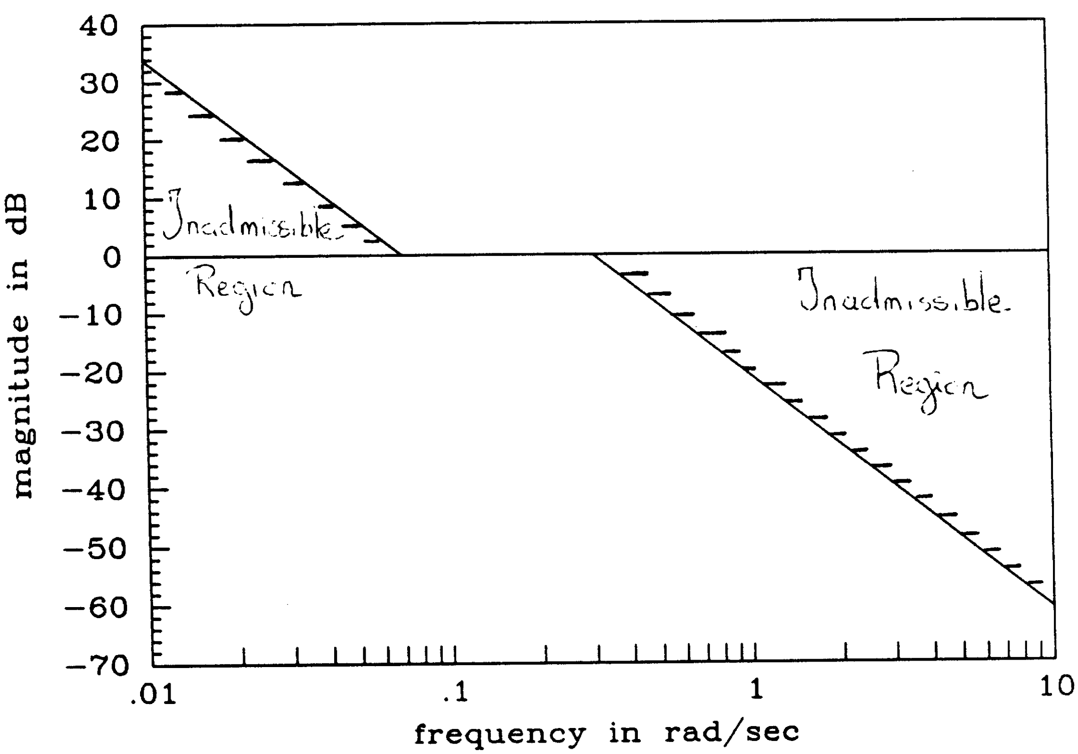

Figure 6.1: Admissible Region for the Loop Transfer Function in Enns' Example.

6.2.2 The LQG Problem

The nominal plant model corresponds to the four disks having the same inertias. Given the transfer function G(s) given in 6.2.1, a state space model can be derived. The Observer Canonical Form is chosen in [Liu86]. The A, B and C matrices are:

$$A = \begin{bmatrix} -0.1610 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6.0040 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -0.5822 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -9.9850 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -0.4073 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -3.9820 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0.640 \\ 0.235 \\ 7.130 \\ 100.020 \\ 10.450 \\ 99.550 \end{bmatrix}$$

$$C = [\ 1.0000\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

The plant poles and zeros are:

Poles	Zeros
0.0000	2.2616 ± j 5.1916
0.0000	–0.0199 ± j 0.9998
–0.0153 ± j 0.7648	–4.8506
–0.0282 ± j 1.4097	
–0.0370 ± j 1.8496	

The controlled output z is given by z = Hx, where H is:

$$H = [0\ \ 0\ \ 0\ \ 0\ \ 0.55\ 11\ \ 1.32\ 18]$$

The penalty on the control is one. The quadratic cost is given by R and R_c respectively equal to:

$$R\ = q_1 H^T H,\ \ q_1 = 10^{-6}$$
$$R_c = 1$$

The disturbance noise is modeled as a white noise being added to the control signal and the measurement noise has intensity one. V and V_c are therefore:

$$V\ = q_2 BB^T$$
$$V_c = 1$$

The robustness guarantees were shown to exist with the full order LQG compensator for values of q_2 ranging from 10^{-2} to 10^6 [Enn84]. Reduced order LQG problems of various orders have been derived for q_2 between 10^{-2} and 2000 by [Liu86], and between 10^{-2} and 10^5 by [Ric87, Hyl90].

6.2.3 Numerical Results using Homotopy

6.2.3.1 Results Summary

The extent of this present comparison is to design reduced order compensator for n_c between 2 and 6 and q_2 equal to 10^j, $j = -2$ to 5. Table 6.1 contains the value of J/q_2 for different compensator orders and different q_2.

Table 6.1: Values J/q_2 with Optimal Compensators of Increasing Order

q_2 \ n_c	2	3	4	5	6	8
10^{-2}	.22708400	.22708397	.22708050	.22708050	.22708044	.27080394
10^{-1}	.16709687	.16707101	.16678943	.16678943	.16677428	.16677295
10^0	.14673280	.14593113	.14378364	.14378357	.14335087	.14330784
10^1	.14293114	.14028275	.13662988	.13662986	.13369265	.13336824
10^2	.14249519	.13868584	.13505833	.13504924	.12819751	.12727027
10^3	.14245084	.13818969	.13475189	.13471420	.12432746	.12280585
10^4	.14244639	.13803300	.13470714	.13465987	.12199848	.11923043
10^5	.14244595	.13798344	.13470227	.13465367	.12064334	.11618572

Figure 6.2 shows the curves J/q_2 for the different compensator orders as a function of q_2. The curve allows an easy comparison with the results of [Ric87, Hyl90] and shows a close match except maybe for $n_c = 5$.

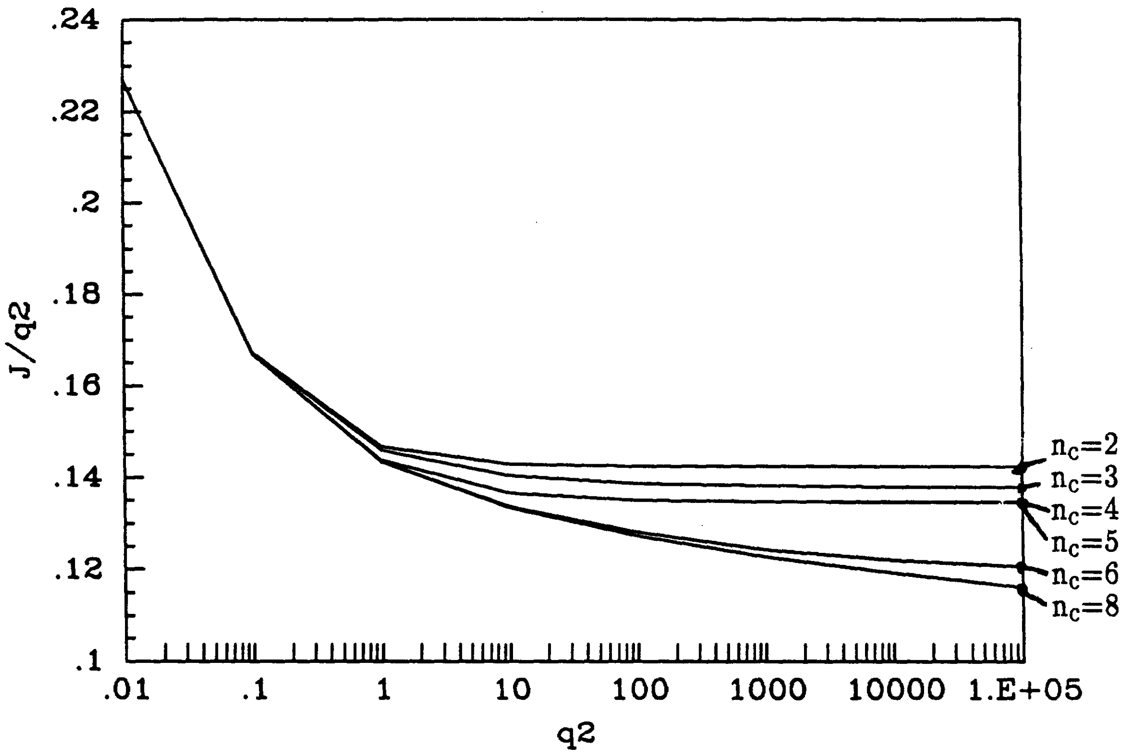

Figure 6.2: Optimal Value of J/q_2 for various Orders of Compensation

6.2.3.2 Convergence Properties

All designs were started at $q_2 = 10^{-2}$. For each value of n_c the LQG problem was transformed into a suitable diagonal problem and the compensator was tracked using the homotopy procedure of Chapter 5. Once the compensator for $q_2 = 10^{-2}$ was found, q_2 was increased, in powers of ten, to the various desired values while the solution just obtained was being tracked. The new solution was then used as a new starting point and q_2 increased again.

The open loop system is both unstable and nonminimum phase. The best achievable performance is therefore limited, and it is not possible to obtain zero error even with a full order LQG regulator [Kwa72a]. Consequently, the cost is insensitive to variation in compensator parameters once a good stabilizing compensator is found and when the variations only tend to improve the cost. The cost is, however, extremely sensitive to variations in the compensator parameters that reduce the stability margins. The problem is, therefore, badly conditioned, and the use of minimization steps encounters difficulties for large values of q_2. Second, fourth and fifth order compensators were obtained with no particular numerical difficulties. The fifth order compensators turn out to be very close to their fourth order counterparts. The loop transfer functions from compensator input to plant output are plotted for $n_c = 4$ and $n_c = 5$ and $q_2 = 1$ on Figure 6.3a, and for $n_c = 4$ and $n_c = 5$ and $q_2 = 10^4$ on Figure 6.3b. For any given q_2, the fifth order compensator has four poles and three zeros which are almost the same as that of the fourth order, plus a pole and a zero which almost but not exactly cancel (Table 6.2). Such a compensator was found to be a minimum for the cost: the eigenvalues of the Hessian calculated for $n_c = 5$ and $q_2 = 1$ split into two groups, the first one containing ten positive eigenvalues between $+3.5250\ 10^{+4}$ and $+1.6573\ 10^{-5}$, and the second group containing thirteen positive or negative eigenvalues whose magnitude is below $5.2371\ 10^{-9}$ (Figure 6.4). The tridiagonal realization of the fifth order SISO compensator requires twenty three parameters, a minimal set requiring ten only: one can see that the Hessian has a nullspace of appropriate dimension, and one can also see that the condition of the problem is bad, with a factor of 10^{+9} between the highest nonzero eigenvalue and the smallest nonzero eigenvalue.

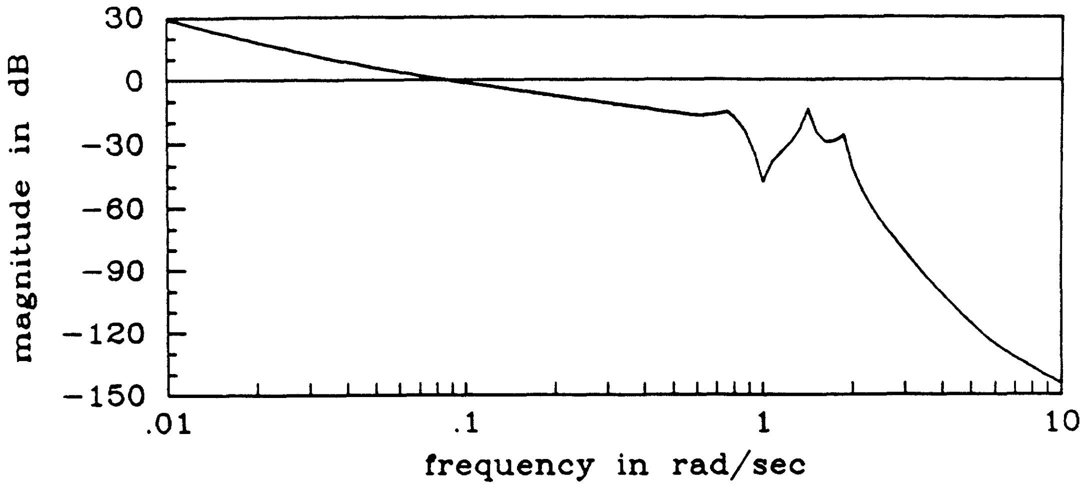

Figure 6.3a: Loop Transfer Function, $q_2 = 1$

———: $n_c = 4$, – – –: $n_c = 5$

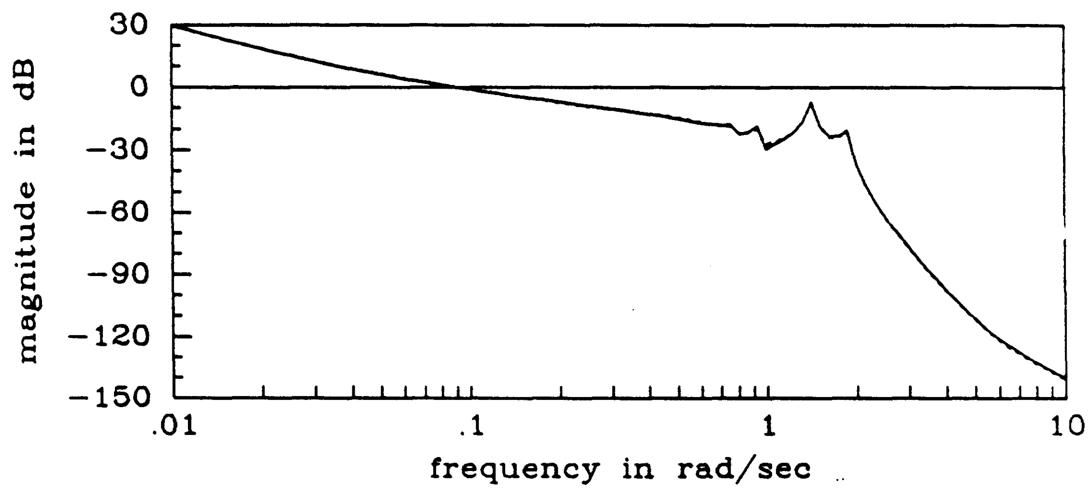

Figure 6.3b: Loop Transfer Function, $q_2 = 10^4$

———: $n_c = 4$, – – –: $n_c = 5$

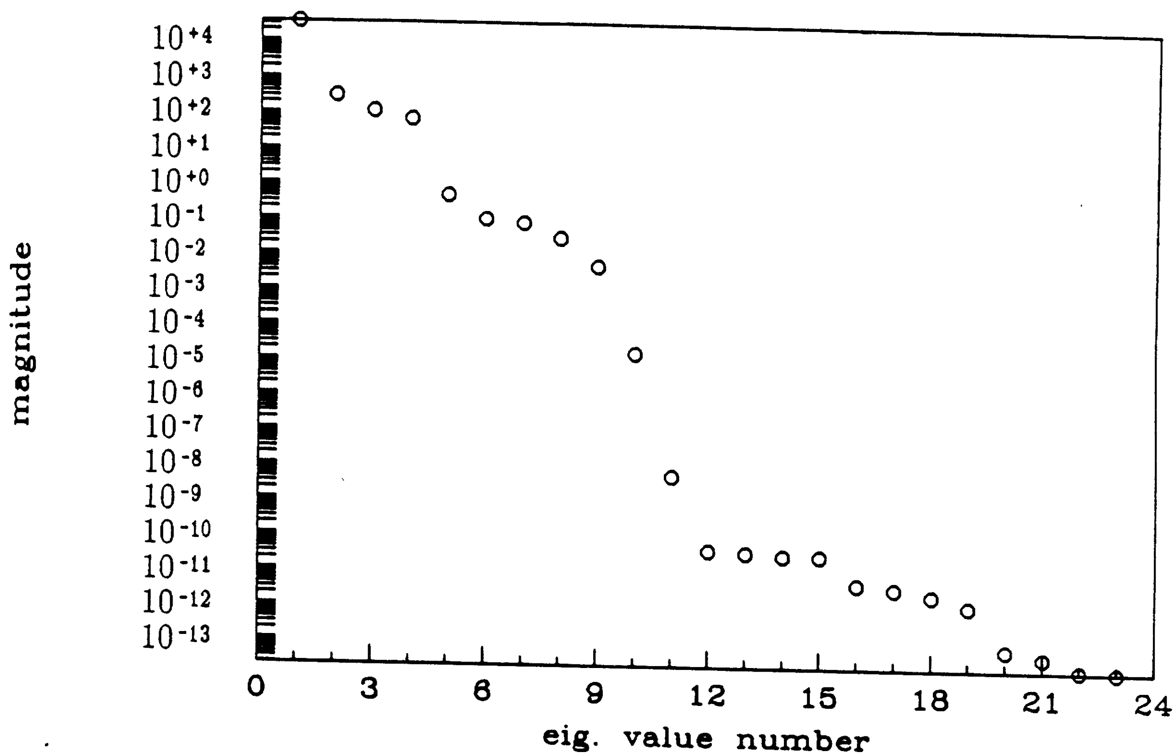

Figure 6.4: Magnitude of the Eigenvalues of the Hessian for $n_c = 5$ and $q_2 = 1$

Table 6.2: Compensators Poles and Zeros, 4th order vs 5th Order

q_2 = 1	n_c = 4	n_c = 5
Compensator Poles	−0.4005 ± j0.3254 −0.0649 ± j0.8089	−0.3999 ± j0.3254 −0.0650 ± j0.8090 −0.0282
Compensator Zeros	−0.0113 ± j0.7731 −0.0339	−0.0113 ± j0.7731 −0.0345 −0.0276
q_2 = 10,000	n_c = 4	n_c = 5
Compensator Poles	−0.3643 ± j0.4136 −0.0174 ± j0.9667	−0.3193 ± j0.4740 −0.0150 ± j0.9754 −0.2129
Compensator Zeros	−0.0138 ± j0.7742 −0.0357	−0.0134 ± j0.7719 −0.0340 −0.2609

Third and sixth order compensators had some difficulties to converge for large q_2. Minimization steps are usually performed with an accurate line search using a bracketing scheme. The line searches resulted in a very slow convergence of the error for large values of q_2. The corresponding variations of the cost were of 10^{-8} percent decrease per step. The infinity norm of the gradient (maximum absolute value of its elements) jumped back and forth between 10^2 and 10^{-3}. In order to cope with such a problem the software was modified to perform complete Newton-Raphson step with no line searches. The accuracy of the line search is highly dependent on the accuracy of the Lyapunov equation solver that eliminate P and Q. For large q_2, Q becomes larger and relatively less accurate, resulting in less accuracy on the cost. Similarly, the search direction is highly dependent on the eigenvectors of the very small eigenvalues and may be less accurate. Step 4 of the algorithm shown in Section 5.3.2 is modified as follows:

Step 4.1 Compute g_k, current gradient value

Step 4.2 if $\|g_k\| < \varepsilon_0$, stop

Step 4.3 Compute H_k

Step 4.4 Compute $d\xi_k = -H_k g_k$

Step 4.5 If $\|g_k\| < \varepsilon_s$,

$$\xi_{k+1} = \xi_k + d\xi_k$$ (Full Modified Newton Step)

Else

Find σ_k such that
$L(\xi_k + \sigma_k d\xi_k)$ is minimized:
line search via bracketing

$$\xi_{k+1} = \xi_k + \sigma_k d\xi_k$$

Go to Step 4.1

The full Newton step is still a modified step, like the one shown in Section 5.3.5, in order to deal with the singularity of the Hessian. The choice of the threshold ε_s below which no line search is performed is highly problem dependent. One wants to be close enough to the minimum so that the full step may converge. Indeed, a full step may increase the cost and result in a non converging sequence of steps. The threshold was set low for most of the runs. For large values of q_2 however ($q_2 = 1{,}000$ and above) and for $n_c = 3$ and $n_c = 6$ the threshold was set to 100. The best accuracy one was able to reach was 6.2 10^{-5} for $q_2 = 10^5$ and $n_c = 6$, and 1.9 10^{-7} for the same value of q_2 and $n_c = 3$.

6.2.3.3 Discussion

The transfer function from the process noise to the output of the system is the same as the open loop plant transfer function in this particular example. G(s) has eight poles and five zeros. As q_2 is increased, the poles of the Kalman filter designed for such a system go towards the minimum phase open loop zeros and the mirror image of the nonminimum phase zeros while the remaining poles go to infinity in a Butterworth pattern [Kwa72b]. Because of the separation principle, poles of the closed loop system with the full order LQG controller will follow such a pattern. This is not the case with reduced order compensators. Only for n_c = 3 do we have a closed loop pole going to infinity as $\sqrt{q_2}$.

Table 6.3: Fast Closed–Loop Poles, $n_c = 3$

q_2 = 1,000	α = –76.57	$\alpha/\sqrt{q_2}$ = –2.42
q_2 = 10,000	α = –241.12	$\alpha/\sqrt{q_2}$ = –2.41
q_2 = 100,000	α = –761.74	$\alpha/\sqrt{q_2}$ = –2.41

Some of the closed loop poles are near the open loop zeros or the reflections about the imaginary axis. For $q_2 = 10^5$ and $n_c = 6$, a pair of poles is at –0.0154 ± j 0.9978 while another pair is at –1.2138 ± j 5.1850. For $q_2 = 10^5$ and $n_c = 4$ a pair of closed loop poles is at –0.0186 ± j 0.9657. Looking at the trend, however, the same pair of poles was at –0.0190 ± j 0.9651 $n_c = 4$ and $q_2 = 10^4$. The real part does not converge toward –0.0199, the real part of the pair of complex conjugate open loop zeros. Figure 6.5 shows the root locus of the closed loop poles as a function of q_2 for $n_c = 4$.

Considering its simplicity, the reduced order compensators achieve performances comparable to the unconstrained optimum for very small orders(n_c = 2, 4). Figure 6.6 shows the impulse response obtained with compensators of order 2 to 6 and $q_2 = 1$. These various responses are also compared to the full order response. Figure 6.7 shows a step response obtained with a second and a fourth order compensator and compares it to the response obtained with the full order controller for $q_2 = 1$. The impulse response shows that higher modes are more highly damped as the order of the compensator is increased. The step response shows however that very good command following can be obtained already with a second order controller.

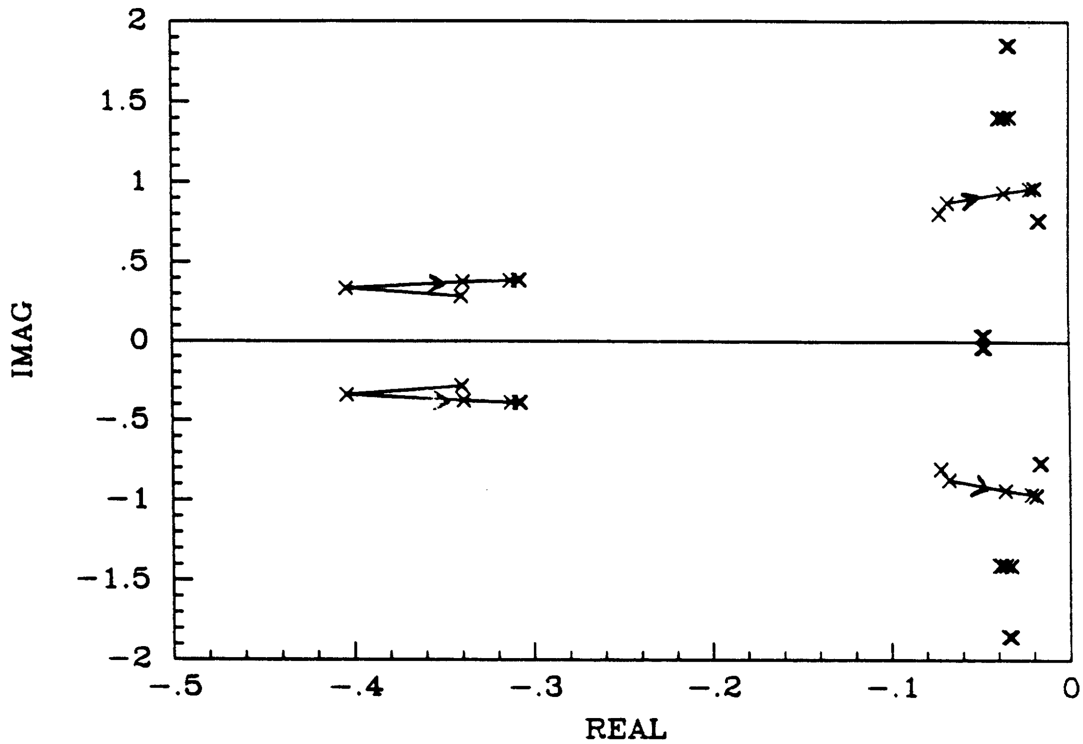

Figure 6.5: Root Locus of the Closed Loop Poles for $n_c = 4$ and $q_2 = 1$ to 10^5

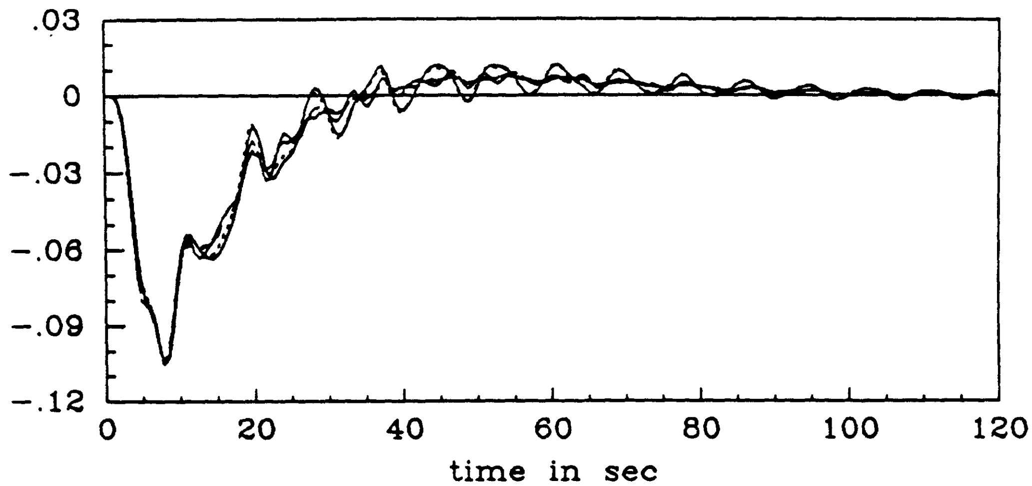

Figure 6.6: Impulse Response,
$n_c = 2, 3, 4, 5, 6, 8$

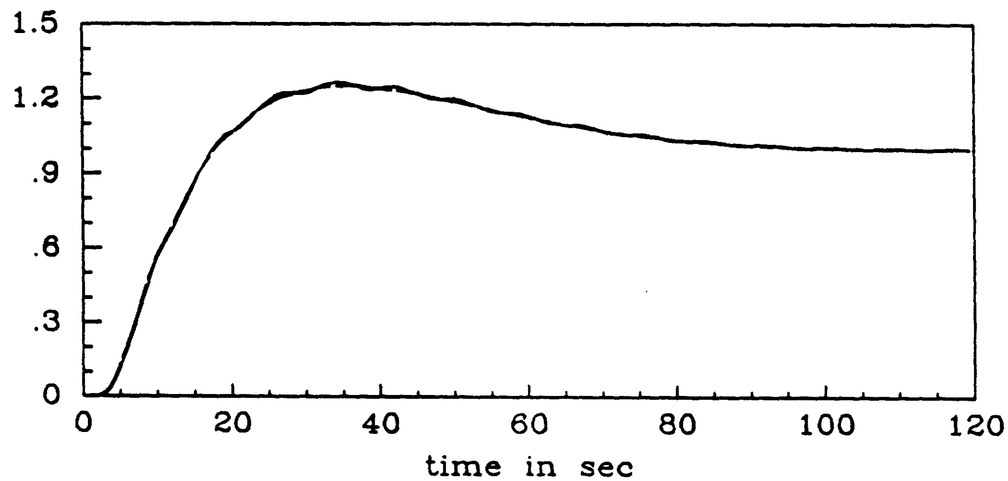

Figure 6.7: Step Response, $n_c = 2, 4, 6, 8$

6.3 THE NASA MINI–MAST

6.3.1 Description

The Mini-Mast [Nas89] is a 20 meter long generic space structure developed at the Structural Dynamics Research Laboratory at the NASA Langley Research Center. Its design duplicated except for its length the Mast truss envisioned for the COFS-I flight experiment. The materials as well as the manufacturing have flight quality specifications. The Mini-Mast has a three longeron construction forming a horizontal triangular cross section inscribed in a 1.4 meter diameter circle. 1.12 meter long battens connect the vertices of the triangles vertically to form a bay while diagonal elements provide stiffness in torsion and shear. The truss contains 18 repeating bays. It is cantilevered to the ground at its bottom. The structure has different possible configurations and can carry for example a tip mass to simulate a payload. This present example considers the mast only. Figure 6.8 shows a generic view of the beam and indicates the X-Y-Z reference frame that is used. The Z axis is vertical pointing up, while the X and Y axes are in the horizontal plane, the Y axis being normal to one of the faces of the triangular section of the beam (Figure 6.8a).

The Mini-Mast is fully instrumented to support active vibration isolation experiments. The principal actuators are three reaction wheels mounted at the top of the structure. The spin axes of the wheels are aligned respectively with the X, Y and Z axes and will be referred to as Wheel X, Wheel Y and Wheel Z. It also has a dual set of sensors: noncontacting displacement sensors are used to monitor the motion of the vertices of the truss in the horizontal plane and normally to the face of the structure (Figure 6.8b). The second set of sensors are high quality rate sensors and accelerometers.

The 'strawman' experiment proposed by NASA is to design a control system to minimize the relative deformation between Bay 18 (top) and Bay 10 (mid mast). An available model had been obtained through parameter identification of the structure using the noncontacting sensors only. Because accurate optical based sensors can be developed to monitor the relative deformation of the structure, it is not too unrealistic to use the displacement sensors located on Bay 10 and 18 as long as they are aggregated to provide three

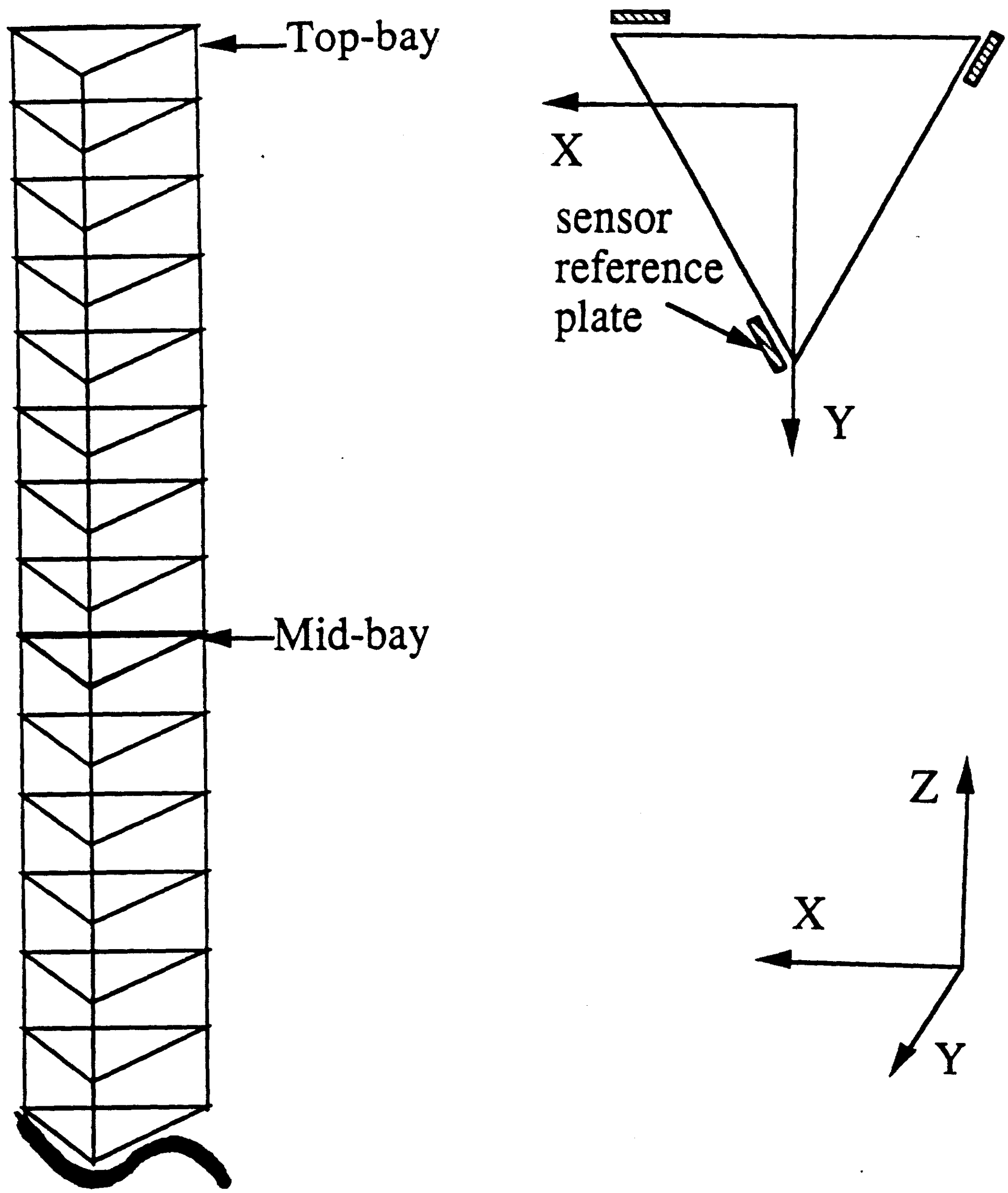

Figure 6.8: General Configuration of the NASA MINI-MAST

relative measurements, the relative torsion between Bay 18 and Bay 10, the relative displacement in the Y direction between Bay 18 and Bay 10 and the relative displacement between Bay 18 and Bay 10 in the X direction. A very simplified control law is sought, consisting of three decentralized second order compensators, the first one feeding the relative torsion to Wheel Z, the second one feeding the relative displacement along Y to Wheel X and, finally, the third one feeding the relative displacement along X to Wheel Y. The decentralized compensator will then be evaluated against its centralized full order counterpart.

6.3.2 The LQG Problem

6.3.2.1 The Mini-Mast Model

The first two bending modes of the Mini-Mast are at 0.86 Hz. The two modes are really close because of the symmetry of the beam. The first torsional mode appears at 4.30 Hz. the second bending modes are at 6.17 Hz. A hundred and eight modes then cluster around 15 Hz and correspond to the bending modes of the diagonal elements constituting the bays. The model to be used is a tenth order model that includes the first and second bending modes as well as the first torsional mode and was obtained through parameter identification on an early setup of the experiment. The modes and damping ratios are summarized in Table 6.4.

Table 6.4: frequencies and damping for the Mini–Mast

Mode description	Frequency in rad/sec	Damping Ratio
1st bending	5.3778	0.0323
1st bending	5.3702	0.0213
1st torsion	27.0133	0.0717
2nd bending	38.4440	0.0238
2nd bending	38.7478	0.0100

The main actuators are three reaction wheels mounted at the top of the structure and driven by DC motors. The dynamics of the motors are important and must be included in the model. The motors cannot deliver any DC torque: on the Mini-Mast, at low frequencies, the transfer functions from voltage to torque are three high pass filters whose corner frequencies are at 3 rad/sec or 0.48 Hz. At high frequencies, the transfer function from voltage to torque rolls off like a one pole system. The second corner frequencies are well above the frequencies of interest, around 350 rad/sec or 55 Hz, and need not be modeled. The final model becomes a thirteenth order model that includes the ten original structural poles of the beam and three poles to describe the dynamics of the wheels. The inputs are:

$u_1 = u_z$, command input on wheel Z in Nm
$u_2 = u_x$, command input on wheel X in Nm
$u_3 = u_y$, command input on wheel Y in Nm.

Six noncontacting displacement sensors are considered, the three monitoring the displacement of bay 18 (top of the mast) and those monitoring Bay 10 (mid mast). The information we want to extract are the relative displacements $X_{18} - X_{10}$ and $Y_{18} - Y_{10}$ as well as the relative angular deformation about the Z axis $\theta_{18} - \theta_{10}$. The information of the six sensors is therefore aggregated to yield only three outputs:

$$y_1 = \theta_{18} - \theta_{10} \text{ (radian)}$$
$$y_2 = Y_{18} - Y_{10} \text{ (meter)}$$
$$y_3 = X_{18} - X_{10} \text{ (meter)}$$

The corresponding A, B and C matrices are given in Appendix D.

6.3.2.2 Measurement and Process Noise

The noncontacting sensors have an RMS error value of 10^{-3} inch or 2.54 10^{-5} meter. After the measurements are aggregated, the measurement noises on y_1, y_2 and y_3 become:

$$V_c = \text{diag}(4.8092\ 10^{-10},\ 8.6021\ 10^{-10},\ 8.6021\ 10^{-10})$$

The disturbance noise is chosen as a random voltage driving the three torque wheels. The voltage is chosen to yield a 10 Nm RMS excitation:

$$V = 10^2 \, BB^T$$

6.3.2.3 Quadratic Cost

The maximum torque the motors can deliver is in the order of 50 Nm. The RMS value of the torque should not be higher than a third of the maximum torque. Using Bryson's rule, the weighting matrix R_c is set to:

$$R_c = (50/3)^{-2} \, I_3$$

The regulated values are y_1 y_2 and y_3: the control is to minimize the relative deformation between the top and the middle of the beam. The relative displacements are to be kept to within a millimeter whereas the angular displacement is to be kept within one milliradian (0.057 degree). The R matrix was selected as:

$$R = C^T \begin{bmatrix} 10^4 & 0 & 0 \\ 0 & 10^6 & 0 \\ 0 & 0 & 10^6 \end{bmatrix} C$$

A Reduced Order Decentralized Controller (RODC) consisting of three second order compensators between y_1 and u_1 (Wheel Z), y_2 and u_2 and y_3 and u_3 was designed. The homotopy procedure was started by canceling any coupling between the torsional mode and between the bending modes in the X plane and the Y plane. The second bending modes were also made unobservable and uncontrollable. As the coupling was continuously introduced, the solution was tracked and no singular points were encountered. The decentralized controller is given in Appendix D.

6.3.3 Design Comparison

The performance of the controllers can be judged on the RMS errors they achieve and the RMS inputs they required. Table 6.5 summarizes the results. Figure 6.9 shows a close up of the location of the open loop poles and the closed loop poles obtained with the two designs. The Full Order Centralized Controller (FOCC) results in three poles on the real axis close to the origin: they correspond to the controller attempting to invert the zeros of the high pass filter of the wheels located at the origin. These modes do not contribute to the cost. Notice also that the FOCC results in higher damping of the second bending modes: the damping ratios of the second bending modes with the FOCC are 0.0416 and 0.0255 respectively whereas the RODC can only achieve damping ratios equal to 0.0262 and 0.0192. Figure 6.10 shows the minimum and maximum singular values of the plant transfer function while Figure 6.11 presents a comparison of the maximum and minimum singular values of loop transfer functions from the compensator inputs to the systems outputs with the two compensators. The RODC follows the shape of the loop transfer function obtained with the FOCC but with reduced gains. At low frequencies the RODC does not try to invert the zero of the wheels and behave like a differentiator whereas the FOCC transfer function is flatter between its slow poles and the pole of the DC motor. The RODC design also yields smaller gains on the second bending modes, providing less damping.

Table 6.5: RMS Errors and RMS Inputs for the Open–Loop,RODC, and FOCC

RMS Value of	Open–Loop	Full–Order Centralized	Reduced–Order Decentralized
$\theta_{18}-\theta_{10}$ (rad)	$1.3198\ 10^{-2}$	$6.3020\ 10^{-3}$	$5.9202\ 10^{-3}$
$Y_{18}-Y_{10}$ (m)	$9.3478\ 10^{-3}$	$1.4251\ 10^{-3}$	$1.7618\ 10^{-3}$
$X_{18}-X_{10}$ (m)	$7.9484\ 10^{-3}$	$1.9466\ 10^{-3}$	$2.4860\ 10^{-3}$
u_z (Nm)	0	24.547	24.841
u_x (Nm)	0	22.258	26.449
u_y (Nm)	0	20.041	20.087

The torsional mode is already well damped and its RMS value falls within the specifications. The relative displacements between the two bays is reduced by a significant amount even though the actuators are used above the limit of 50/3 Nm. Hence, the displacement cannot be reduced to within a millimeter RMS. The designs were not iterated to make the control inputs fall within the specifications. The torque limitations make it impossible to achieve the required accuracy. These seemingly high RMS values are due to the large levels of excitation used in the problem.

The RODC achieves very good performance given its extreme simplicity. The performance degradation is 6.8% of the open loop RMS errors while the control RMS inputs are 18.8 % higher than the FOCC RMS inputs. Notice that the torsion is kept within a tighter bound. The Y channel also requires much larger controls from Wheel X. The reason for this is that the centralized compensator deliberately couples the axes X, Y and Z. It makes use of the fact that the torsional mode is well damped already so that it can use Wheel Z to get extra authority on the bending modes. The full order controller has also more authority on the second bending modes. Figure 6.12 show the transient of y_1 and y_2 with the two different controllers for an initial conditions mostly in bending in the Y axis. The response of y_2 $(Y_{18} - Y_{10})$ is very similar with both controllers, except for more overshoot and a smaller damping of the higher modes with the decentralized control scheme. The torsional response (y_1) is much different, however, and the centralized controller gives rise to a much higher transient. This is due to the fact that the compensator uses Wheel Z in order to damp out the oscillation in bending as well. When the system has initial conditions mostly in torsion, the FOCC tends to cancel the coupling between the torsional mode and the bending modes whereas the RODC lets the Z channel excite the remaining X and Y channels. Figure 6.13 shows the transient of y_1 and u_2 with the two controllers. The torsional response is almost identical with both schemes. One can see, however, that the decentralized controller cannot anticipate the error coming from the coupling which is going to excite the bending modes on the X and Y channels: the transient of u_2 (Y wheel) shows that the command resulting from the decentralized scheme lags behind the command of the centralized compensator and requires, therefore, higher torque levels. One can also see that the RODC has longer residual oscillations.

The controller transfer function relates three inputs to three outputs and is therefore made of nine SISO transfer functions: the off-diagonal transfer functions of the decentralized controller are identically zero. The infinity norms of the transfer functions give an indication of their relative importance. For the decentralized compensator, the matrix of infinity norms is

$$G_{RODC} = 10^4 \begin{bmatrix} 2.2262 & 0.0000 & 0.0000 \\ 0.0000 & 2.6752 & 0.0000 \\ 0.0000 & 0.0000 & 0.8026 \end{bmatrix}$$

For the centralized compensator, it is:

$$G_{FOCC} = 10^4 \begin{bmatrix} 3.5326 & 2.9372 & 6.3055 \\ 2.4415 & 8.3367 & 3.9215 \\ 7.8882 & 4.3139 & 6.2498 \end{bmatrix}$$

The off-diagonal elements of G_{FOCC} are comparable to the diagonal elements. This clearly illustrates the fact that the centralized controller couples inputs and outputs in order to obtain the maximum control authority in all three axes. The matrix G_{FOCC} does not indicate any clear simplification of the control structure. The performance obtained with the decentralized controller is, however, very satisfactory.

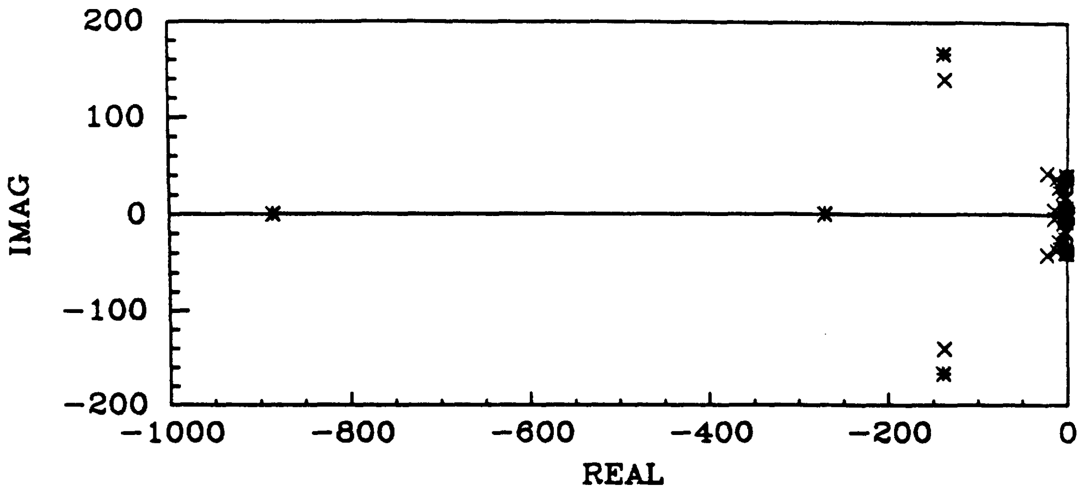

Figure 6.9a: Comparison of Open Loop and Closed Loop Poles,
o: Open Loop, ×: FOCC, ∗: RODC, Mini-Mast Example

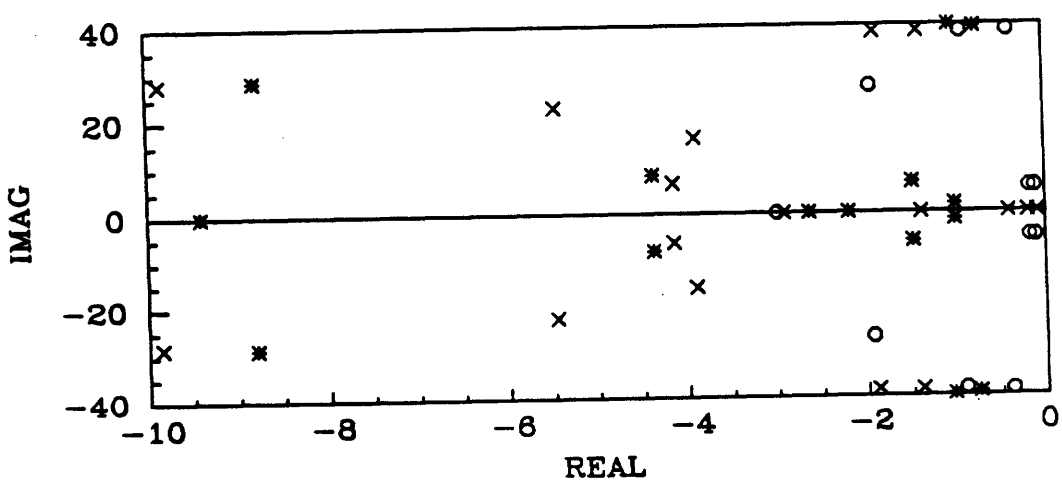

Figure 6.9b: Comparison of Open Loop and Closed Loop Poles,
o: Open Loop, ×: FOCC, ∗: RODC, Mini-Mast Example

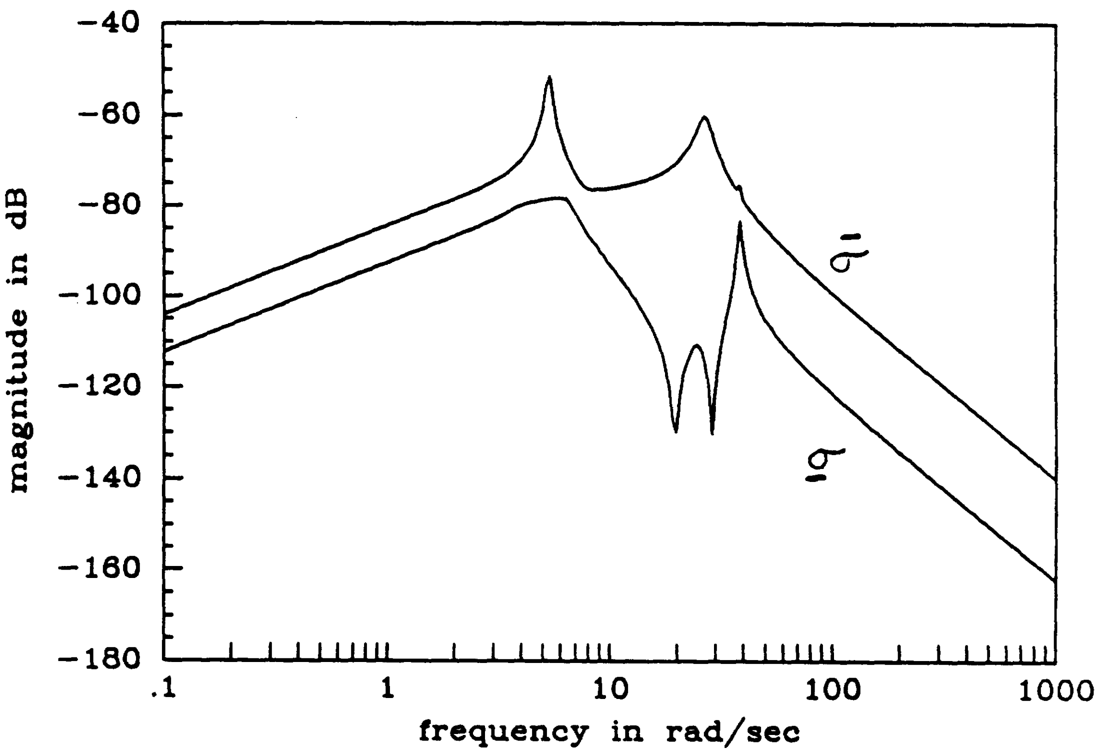

Figure 6.10: Minimum and Maximum Singular Values of the Open Loop Transfer Function, Mini-Mast Example

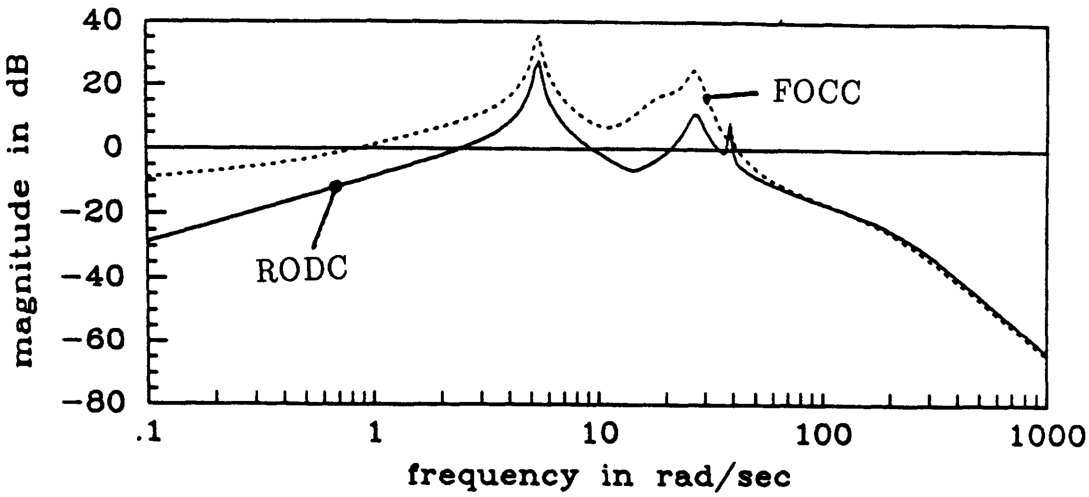

Figure 6.11a: Loop transfer Function, Maximum Singular Value,
———: RODC, – – –: FOCC

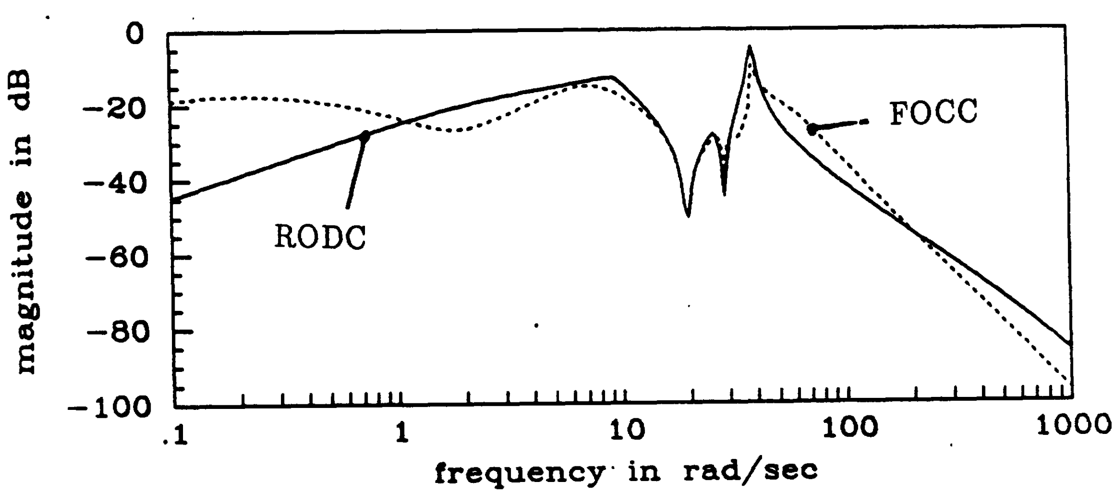

Figure 6.11b: Loop transfer Function, Minimum Singular Value,
———: RODC, – – –: FOCC

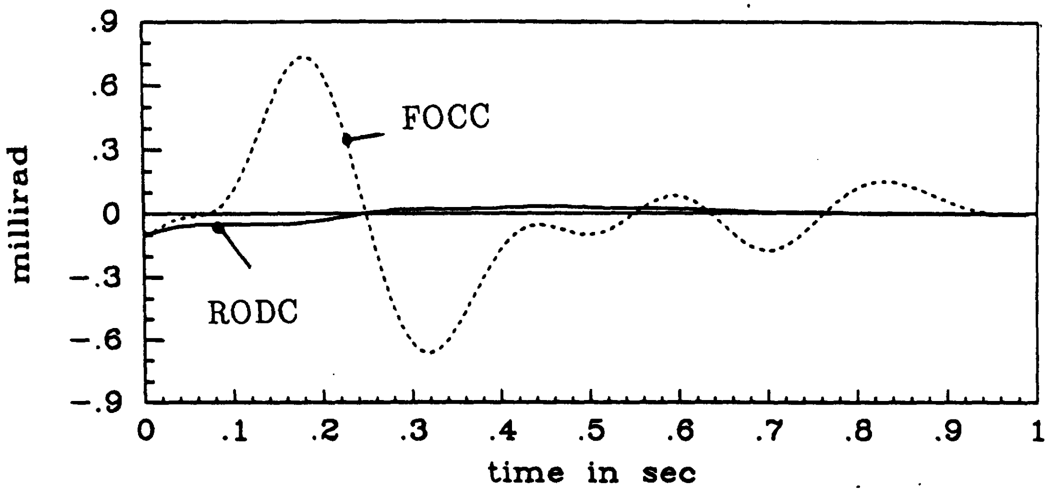

Figure 6.12a: $\Theta_{18} - \Theta_{10}$, Transient Response for an Initial Condition in Bending in the Y Axis, ———: RODC, - - -: FOCC

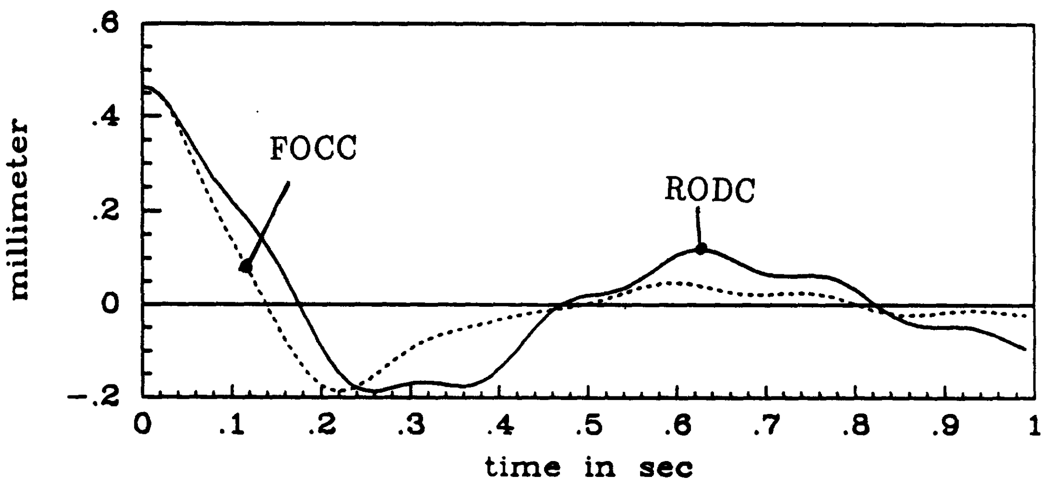

Figure 6.12b: $Y_{18} - Y_{10}$, Transient Response for an Initial Condition in Bending in the Y Axis, ———: RODC, - - -: FOCC

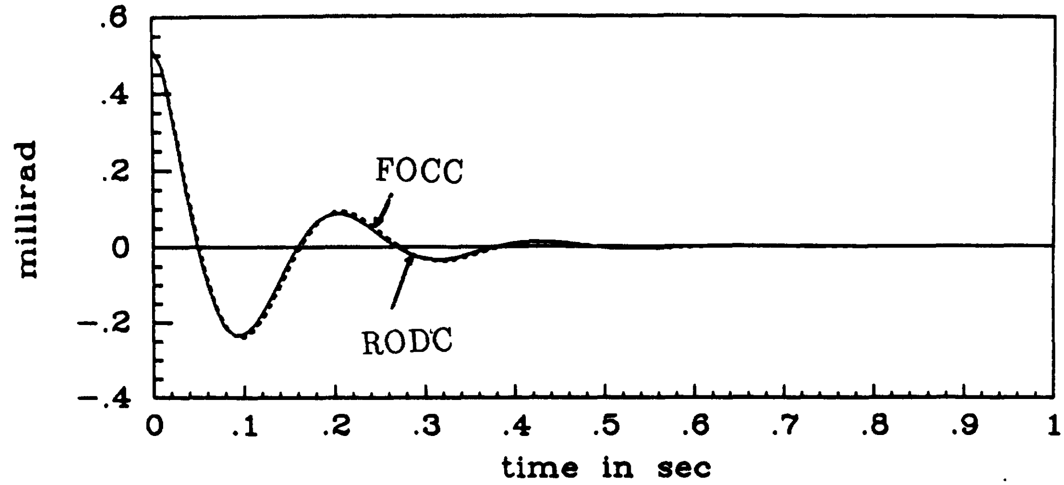

Figure 6.13a: $\Theta_{18} - \Theta_{10}$, Transient Response for an Initial Condition in Torsion, ———: RODC, – – –: FOCC

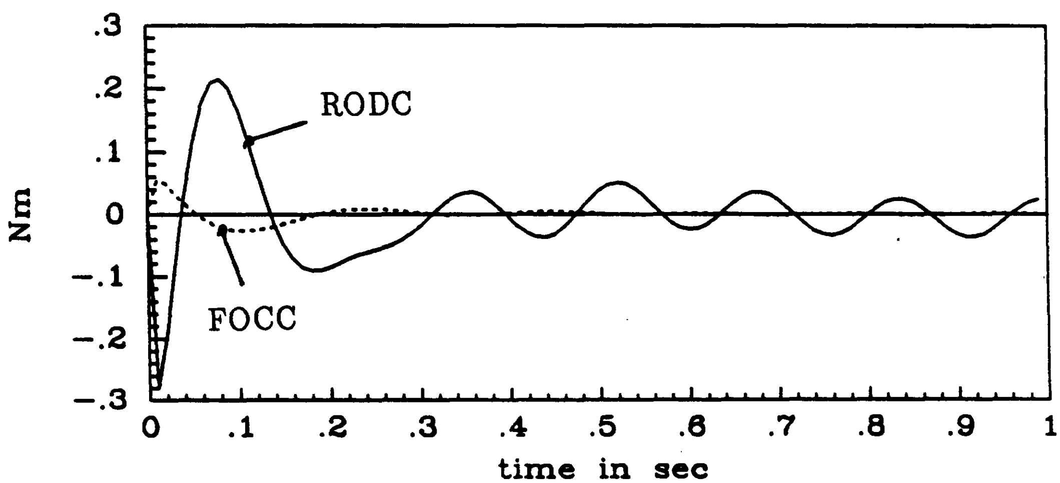

Figure 6.13b: Wheel X Torque Command, Transient Response for an Initial Condition in Torsion, ———: RODC, – – –: FOCC

6.4 THE SCOLE

6.4.1 Description

The SCOLE was constructed to provide a physical testbed for the investigation and validation of design techniques considering control structure interaction [Nas87]. A large plate representing a space shuttle model and weighing 500 pounds is suspended by a single cable through a universal joint located as close as possible to the center of mass of the entire article. A light, hexagonally shaped structure representing an antenna reflector is attached to the bottom of the plate by a long flexible mast. The reflector and the mast both weigh about 5 pounds. Figure 6.14 shows the basic SCOLE structural assembly. A reference frame is defined as follows (Figure 6.14): the Z axis is vertical, positive in the upward direction; the X axis is aligned with the axis of symmetry of the shuttle; the Y axis is along the right wing (assuming the shuttle's bay is below). The reflector is not deployed in a symmetric fashion: the reflector is attached horizontally to the mast and rotated to the right so that one of its sides is in the X direction (Figure 6.14). Aircraft quality rate sensors are available in all three axes both on the shuttle and at the end of the mast. More sensors, such as accelerometers can be used but are not considered here. The actuators consist of a Control Moment Gyros (CMG) on the shuttle and of three orthogonally mounted reaction wheels at the end of the mast. Torques in all three axes can be commanded both on the shuttle and on the reflector.

The experiment proposed here is to control the displacement and attitude of the reflector relative to the shuttle such that, if the control is perfect, the composite system should act like a rigid body. Various control architectures will be proposed in order to illustrate the effect of reducing the order of the compensator and constraining the information pattern.

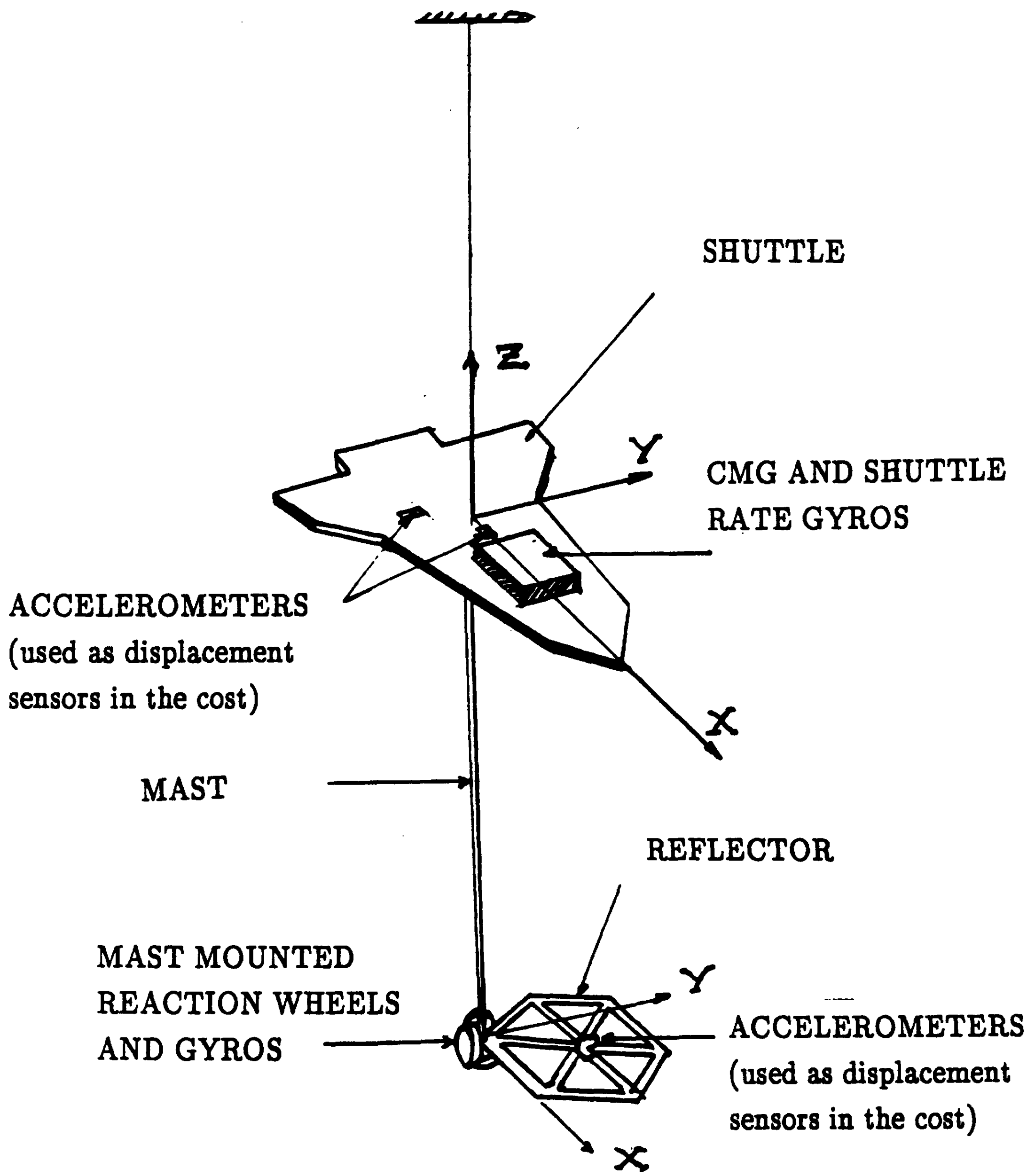

Figure 6.14: General Configuration of the SCOLE Experiment

6.4.2 The LQG Problem

6.4.2.1 General

The SCOLE has three marginally stable modes corresponding to the rigid body attitude motion about the universal joint. A small offset of the CG relative to the universal joint stabilizes the modes and makes the attitude observable from the rate sensors. This artifact makes the use of accelerometers unnecessary. The system also has two global pendulous modes corresponding to the swinging of the long attachment cable. Such modes are almost totally uncontrollable and unobservable from the actuators and sensors considered here and will be ignored. The remaining modes are flexible modes. The model used in this example utilizes the two first bending and the two second bending modes as well as the first torsional mode. Table 6.6 summarizes the frequencies and damping characteristics. The damping ratios were artificially set to 10^{-3}.

Table 6.6: Frequencies and Damping for the SCOLE

Mode description	Frequency in rad/sec	Damping Ratio
Rigid Body	0.174021	0.0010
Rigid Body	0.627081	0.0010
Rigid Body	1.009350	0.0010
1st Bending	3.554800	0.0010
1st Bending	4.007660	0.0010
1st Torsion	9.512380	0.0010
2nd Bending	18.496300	0.0010
2nd Bending	27.526200	0.0010

Six actuators are available for control. They are grouped as follows:

$u_1 = u_{sx}$, shuttle CMG, X input axis, lb-ft
$u_2 = u_{mx}$, mast mounted reaction wheel, X axis, lb-ft
$u_3 = u_{sy}$, shuttle CMG, Y input axis, lb-ft

$u_4 = u_{my}$, mast mounted reaction wheel, Y axis, lb-ft
$u_5 = u_{sz}$, shuttle CMG, Z input axis, lb-ft
$u_6 = u_{mz}$, mast mounted reaction wheel, Z axis, lb–ft

The six rate sensors are divided as follows:

$y_1 = y_{sx}$, shuttle rate sensor, X axis, rad/sec
$y_2 = y_{mx}$, mast mounted rate sensor, X axis, rad/sec
$y_3 = y_{sy}$, shuttle rate sensor, Y axis, rad/sec
$y_4 = y_{my}$, mast mounted rate sensor, Y axis, rad/sec
$y_5 = y_{sz}$, shuttle rate sensor, Z axis, rad/sec
$y_6 = y_{mz}$, mast mounted rate sensor, Z axis, rad/sec

The A, B and C matrix are provided in Appendix E.

6.4.2.2 Measurement and Process Noise

The noise properties of the various sensors can be found in [Nas87]. All rate sensors have an RMS noise of 0.005 rad/sec. The matrix V_c is:

$$V_c = 0.005^2 \, I_6$$

The process noise is taken as a random command on each of the actuators. The command is taken to be 0.3162 lb-ft RMS on the CMG's and 0.1 lb-ft on the mast mounted reaction wheels:

$$V = B \, \mathrm{diag}(0.1, 0.01, 0.1, 0.01, 0.1, 0.01) \, B^T$$

6.4.2.3 Quadratic Cost

The maximum torque the CMG can deliver is on the order of 1.5 lb-ft. The mast mounted reaction wheels are smaller and can only deliver 0.1042 lb-ft. Following Bryson's rule, the matrix R_c is chosen to be:

$$R_c = 3^2 \, \mathrm{diag}(\, 1.5^{-2}, 0.1042^{-2}, 1.5^{-2}, 0.1042^{-2}, 1.5^{-2}, 0.1042^{-2})$$

There are five controlled variables to the problem in the vector $z = Hx$. Three are the relative attitude between the shuttle and the reflector, the remaining two are the relative displacements in the X and Y direction between the shuttle and the center of the reflector. Denoting by θ_x, θ_y and θ_z the rotation about the X, Y and Z axis respectively, and by attaching a subscript 's' for quantities relative to the shuttle and by 'r' those relative to the reflector, the controlled variables are:

$$
\begin{aligned}
z_1 &= \theta_{sx} - \theta_{rx}, && \text{rad} \\
z_2 &= \theta_{sy} - \theta_{ry}, && \text{rad} \\
z_3 &= \theta_{sz} - \theta_{rz}, && \text{rad} \\
z_4 &= X_s - X_r, && \text{inch} \\
z_5 &= Y_s - Y_r, && \text{inch} \\
z &= Hx
\end{aligned}
$$

The matrix H is given in Appendix E. The design goal is to keep the relative angles z_1, z_2 and z_3 within 0.3000 millirad (1 minute of arc). The penalty on the displacement was chosen to correspond to a 4-minute-of-arc misalignement of the 120 inch long mast, or roughly 0.125 inch. The matrix R was chosen as:

$$R = H^T[\,\mathrm{diag}(\,10^{-7}, 10^{-7}, 10^{-7}, 1.6\ 10^{-2}, 1.6\ 10^{-2})]^{-1}\,H$$

6.4.2.4 Control Architecture

The general control structure one wishes to implement consists of three processors controlling the X, Y and Z axes. The eigenmotions corresponding to the bending modes were purposely tailored to be skewed at ± 45 degrees in the X-Y plane (see B and C matrices). This makes the control difficult for the axis decoupled compensator and uncovers interesting limitations of fixed architecture controllers. The first architecture is that of a Reduced Order Decentralized Controller (RODC). It can be summarized in the following table:

Processor	Order	Sensor	Actuator
1	4	1, 2	1, 2
2	4	3, 4	3, 4
3	4	5, 6	5, 6

An improved architecture is sought and an overlapping structure is investigated next. The compensator is similar to the RODC with the modification that Actuator 4 is now available to Processor 1 and Actuator 2 is available to Processor 2, thus giving more authority on the reflector in the X and Y axes. The Reduced Order Overlapping Controller (ROOC) is as follows:

Processor	Order	Sensor	Actuator
1	4	1, 2	1, 2, 4
2	4	3, 4	2, 3, 4
3	4	5, 6	5, 6

The total order of the compensator is still 12. In order to evaluate the effect of the fixed information structure, the orders of processors 1 and 2 are increased to 6. The last architecture is that of a Full Order Decentralized Controller (FODC):

Processor	Order	Sensor	Actuator
1	6	1, 2	1, 2
2	6	3, 4	3, 4
3	4	5, 6	5, 6

The solution to the RODC problem was obtained by solving the problem with the 12th order model first. The initial diagonal system was obtained by removing the second bending modes and by decoupling the problem into three 4th order LQG problems. The first torsional mode can easily be identified and was controlled along with the rigid body mode corresponding to the yaw by the Z channel. The mode shapes of the two first bending modes are strongly coupled in the X and Y axes. One was selected to be associated with the pitching mode and controlled by the Y channel while the last bending mode and the rolling mode were controlled by the X channel. As the coupling was introduced through the matrices R, V, B and C, the solution jumped from one path to another, to finally converge to a 12th order Full Order Decentralized Controller (FODC12) for the 12th order system. The second bending modes were subsequently introduced and the homotopy procedure used again. Because they are significantly low and undamped, and because they are strongly coupled in the X and Y direction, the solution path jumped again from one path to another in order to converge. A low accuracy was obtained on the RODC solution. The fact that both second bending modes are to be controlled forces one controller to have fast poles. This breaks the symmetry between the first and the second compensator and both have to be modified so that they jointly take care of the first two and second two bending modes with only four poles, the remaining poles dealing with the rigid body motion. The final solution is a 12th order reduced order decentralized controller (RODC) for the 16th order plant. The solution was used as a starting point for the ROOC problem and direct minimization without homotopy was used to obtained the solution. The FODC solution was found also through direct minimization starting with the FODC12 solution whose dynamics were augmented to comply with the extra poles introduced in the X and Y subcontrollers. Convergence on the FODC was extremely good and the accuracy high. All compensators are in Appendix E.

6.4.3 Design Comparison

The RMS values achieved by the different designs are summarized in Table 6.7. Table 6.8 summarizes the optimal cost and the minimal damping achieved.

The fixed architecture designs achieve performances very similar to that of the optimal solution: the worst performance degradation is 2.68 % of the closed loop RMS of z_2 and is obtained with the RODC. The required inputs are however significantly higher: the RMS of X axis of the CMG with the ROOC is 42.9 % higher than it is with the FOCC. Yet, it is still below 0.5 lb-ft which was chosen as the baseline. The RMS of u_6 (mast mounted reaction wheel, Z axis) is above the limit of 3.47 10^{-2} lb-ft which is a third of the maximum. torque. All designs require the same amount of RMS torque (within 0.6 %) from this particular actuator and the weight on u_6 should therefore be changed if one wants to iterate the designs. The overall performances are relatively poor even with the optimal compensator. The control authority on the CMG is low considering the large inertia of the system and the time constant of the closed loop rigid body modes is around one hundred seconds. The compensator must also consider the fact that the motion of the plate excites the vibrational modes and that the mast mounted wheels have also limited authority, limiting furthermore the bandwidth of the design.

Table 6.7: RMS Errors and RMS Control for the Open Loop, the Reduced Order Decentralized, Reduced Order Overlapping, Full Order Decentralized and Full Order Centralized Loops

RMS Value of	Open Loop	Reduced Order Decentralized	Reduced Order Overlapping
$\theta_{sx}-\theta_{rx}$ (rad)	2.6153 10^{-2}	2.8164 10^{-3}	2.8169 10^{-3}
$\theta_{sy}-\theta_{ry}$ (rad)	2.0418 10^{-2}	2.2852 10^{-3}	2.2761 10^{-3}
$\theta_{sz}-\theta_{rz}$ (rad)	6.1452 10^{-3}	9.9307 10^{-4}	9.9304 10^{-4}
$X_s - X_r$ (in)	2.8250	2.9532 10^{-1}	2.9533 10^{-1}
$Y_s - Y_r$ (in)	3.2883	3.4182 10^{-1}	3.4182 10^{-1}
u_{sx} (lb–ft)	0	5.6159 10^{-1}	5.6313 10^{-1}
u_{rx} (lb–ft)	0	3.8562 10^{-2}	3.7792 10^{-2}
u_{sy} (lb–ft)	0	3.6412 10^{-1}	3.6298 10^{-1}
u_{ry} (lb–ft)	0	3.3221 10^{-3}	1.3817 10^{-2}
u_{sz} (lb–ft)	0	3.7783 10^{-1}	3.7790 10^{-1}
u_{rz} (lb–ft)	0	6.2589 10^{-2}	6.2582 10^{-2}

Table 6.7: Cont'd

RMS Value of	Full Order Decentralized	Full Order Centralized
$\theta_{sx}-\theta_{rx}$ (rad)	2.7910 10^{-3}	2.7848 10^{-3}
$\theta_{sy}-\theta_{ry}$ (rad)	2.2318 10^{-3}	2.2254 10^{-3}
$\theta_{sz}-\theta_{rz}$ (rad)	9.9302 10^{-4}	9.9144 10^{-4}
$X_s - X_r$ (in)	2.9517 10^{-1}	2.9509 10^{-1}
$Y_s - Y_r$ (in)	3.4175 10^{-1}	3.4133 10^{-1}
u_{sx} (lb–ft)	4.3049 10^{-1}	3.9411 10^{-1}
u_{rx} (lb–ft)	2.4743 10^{-2}	2.5987 10^{-2}
u_{sy} (lb–ft)	3.7038 10^{-1}	3.1835 10^{-1}
u_{ry} (lb–ft)	3.2112 10^{-2}	2.9442 10^{-2}
u_{sz} (lb–ft)	3.7732 10^{-1}	2.4051 10^{-1}
u_{rz} (lb–ft)	6.2598 10^{-2}	6.2283 10^{-2}

Table 6.8: Optimal Cost and Minimum Damping for Different Control Architecture

Control Architecture	Cost	Minimum Damping Ratio, 2nd Bend.
Open-loop	12,560.89	$1.0000\ 10^{-3}$
RODC	161.07	$1.5385\ 10^{-3}$
ROOC	160.73	$1.7692\ 10^{-3}$
FODC	156.78	$4.9231\ 10^{-3}$
FOCC	155.38	$5.7692\ 10^{-3}$

Figure 6.15 presents a close-up of the closed loop poles and compare then to the open loop poles. Figure 6.15a shows the closed loop poles obtained with the RODC and the ROOC while Figure 6.15b shows the locations of the poles with the FODC and the FOCC. The rigid body modes are moved to very similar locations with all designs with the slowest closed loop pole around –0.03 rad/sec. All four designs provide similar amounts of damping to the first bending modes as well as the first torsional mode which does not appear on the figure. A noticeable difference occurs with the second bending modes however. The FODC achieves 14.6 % lower damping compared to the FOCC. The reduced order compensators, on the other hand, are unable to provide any significant amount of damping. Notice that the overlapping architecture provides equivalent damping of both second bending modes whereas the decentralized architecture results in the highest mode having a smaller damping ratio (Figure 6.15b).

Figure 6.16 shows the locations of the poles and zeros of the different compensators. All designs yield nonminimum phase zeros, with the fastest zeros being associated with the reduced order compensators (Figure 6.16c,d). Both full order compensators have a pair of lightly damped oscillatory poles close to the second bending modes of the plant in order to provide some damping. With a limited order, the RODC and the ROOC cannot achieve such pole locations. The result is that the bandwidths of Processor 1 and 2 split: the first subcontroller has two fast real poles around –5 and –27, the second subcontroller having its poles near the first bending modes. Figure 6.17 shows the minimum and maximum singular values of the open loop transfer function. Figure 6.18 shows the maximum singular values of the compensator transfer functions. Both full order designs have resonances located a the first and second bending modes as well as at the first torsional mode. The reduced order compensators, on the other hand, have a resonant peak at the first torsional mode, but must do some averaging between the first and the second bending modes. The damping provided by the reduced order controller will, therefore, be smaller. Figure 6.19 shows the maximum singular values of the loop transfer function from compensator inputs to plant outputs with the various controllers. The agreement of the curves at low frequencies is excellent. The FOCC has deeper valleys between the resonant peaks of the loop transfer function. The reduced order compensators, on the other hand, result in flatter curves.

The absence of symmetry between subcontroller 1 and 2 may be surprising since the bending modes have comparable observability and controllability properties from sensors and actuators in the X and Y directions. The fact that the second bending modes have to be controlled forces one of the controller to have a larger bandwidth, and the coordination between the controllers becomes more difficult when the architecture is specified, explaining in part the convergence problems. Because of the relative symmetry in X and Y, it is highly probable that a local minimum exists corresponding to Processor 2 having the highest bandwidth. When the decentralized solution obtained on the 12th order model is used to start the a direct optimization with the 16th order model, the successive compensators obtained through the iterations retain comparable bandwidths in X and Y. The minimization was not carried out completely, but one is confident that a more symmetric solution exists as well. The cost was, however, larger in that case, at about 164.5.

When the cost is made of several equivalent contributions, the unconstrained compensator will be able to minimize independently each of the contributions. When constraints are introduced, these contributions cannot be minimized independently anymore, which explains why many different trade-offs may occur and why compensators of very different character may be produced. All designs may have equivalent performances if one only looks at the value of the cost.

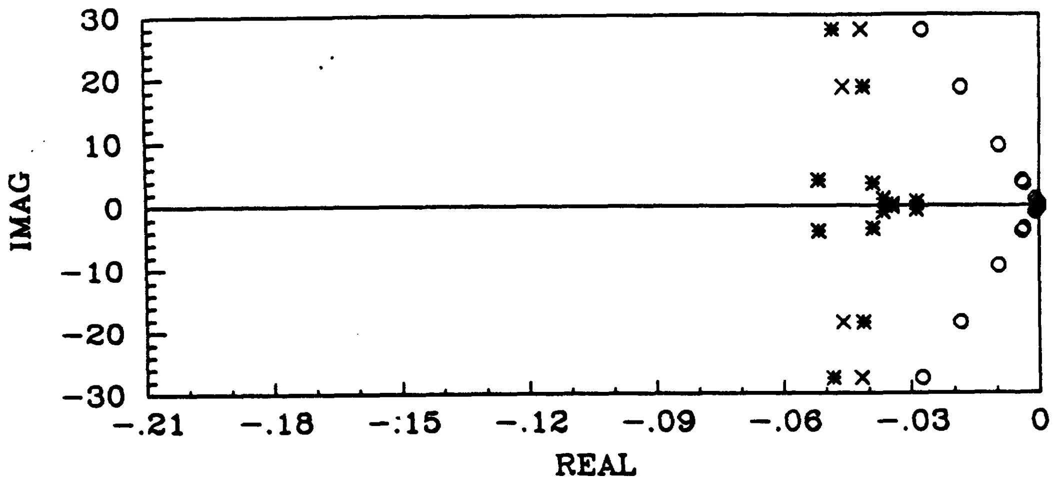

Figure 6.15a: Comparison of Open Loop and Closed Loop Poles:
o: Open Loop Poles, ×: RODC Closed Loop Poles,∗: ROOC Closed Loop Poles

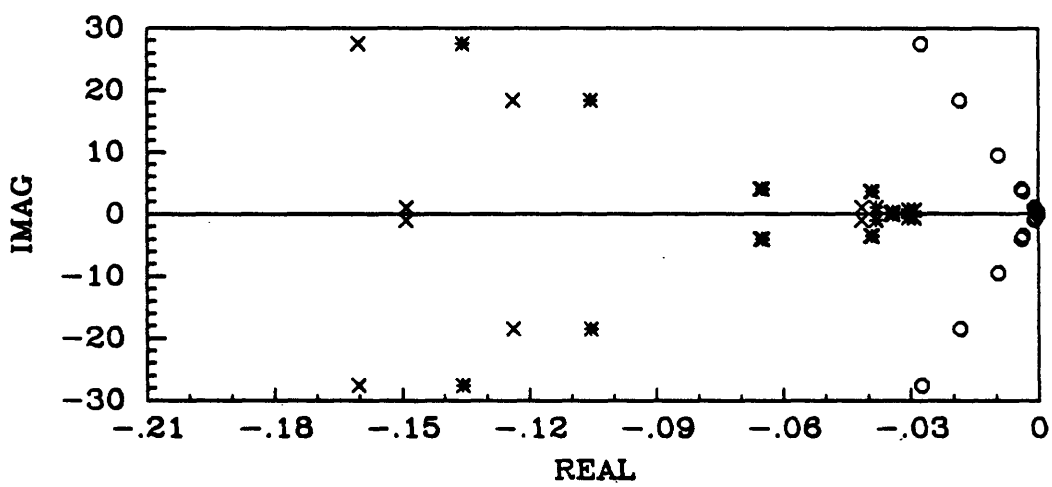

Figure 6.15b: Comparison of Open Loop and Closed Loop Poles:
o: Open Loop Poles, ×: FOCC Closed Loop Poles,∗: FODC Closed Loop Poles

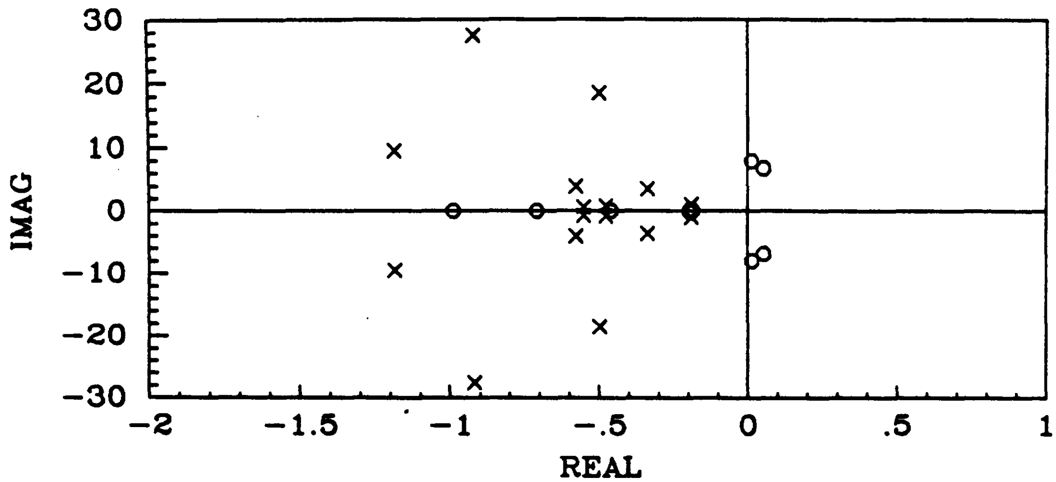

Figure 6.16a: Compensator Poles × and Zeros o, FOCC

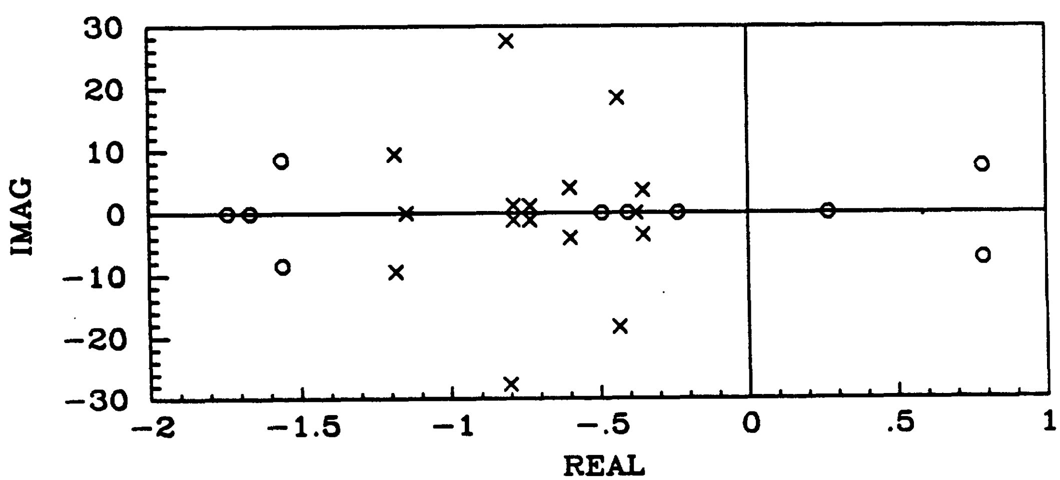

Figure 6.16b: Compensator Poles × and Zeros o, FODC

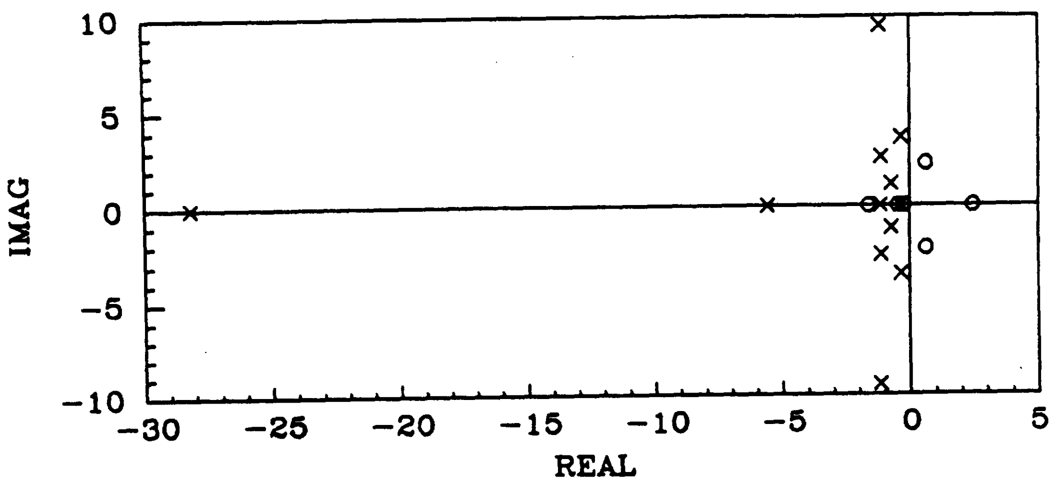

Figure 6.16c: Compensator Poles × and Zeros o, ROOC

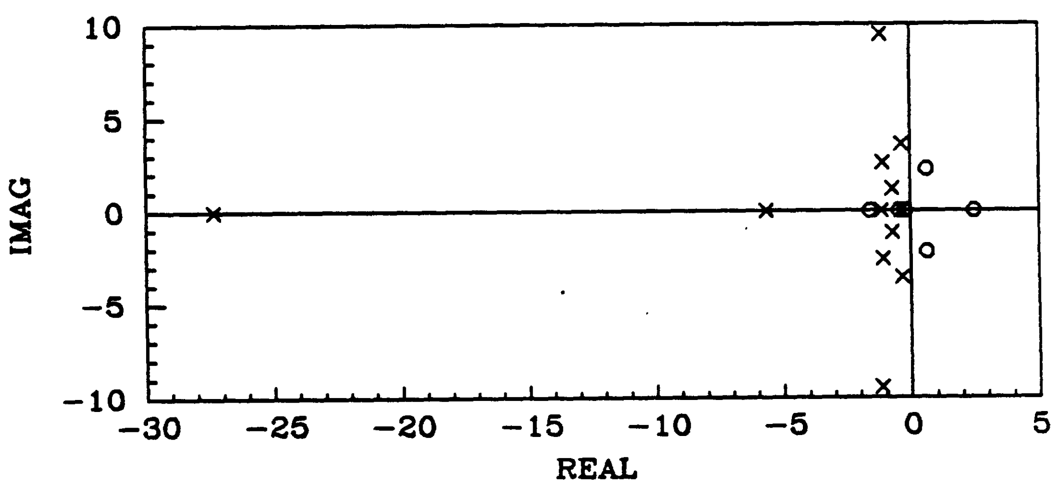

Figure 6.16d: Compensator Poles × and Zeros o, RODC

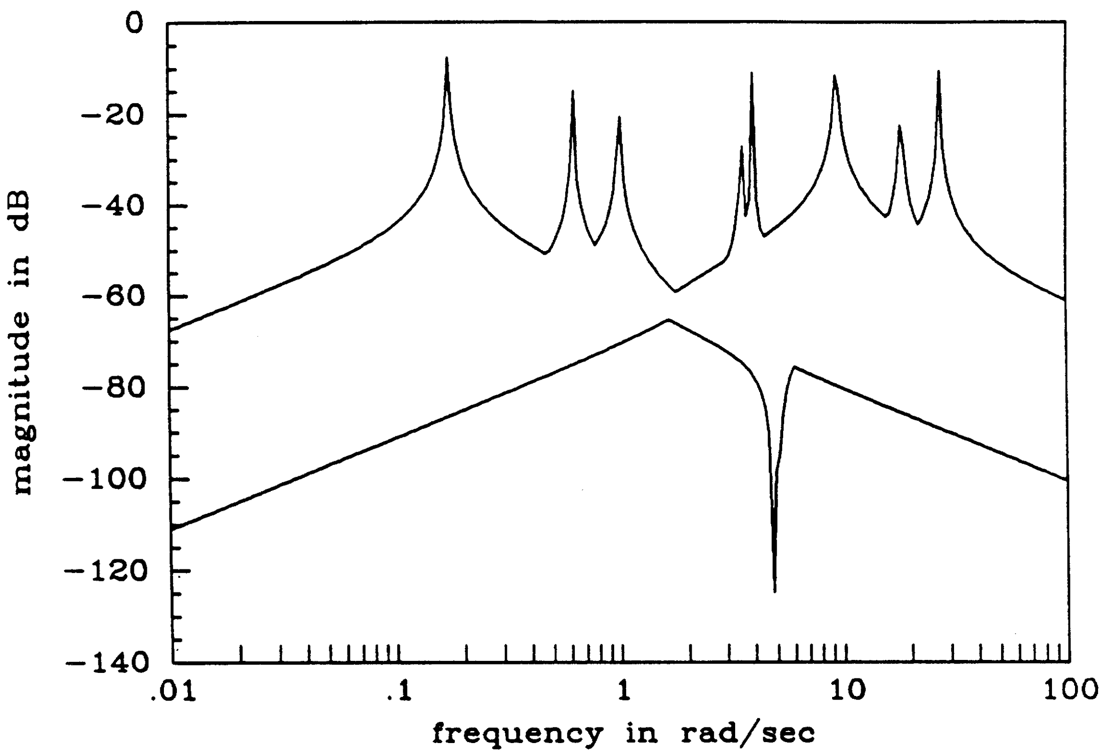

Figure 6.17: Minimum and Maximum Singular Values of the Open Loop Transfer Function, SCOLE Example

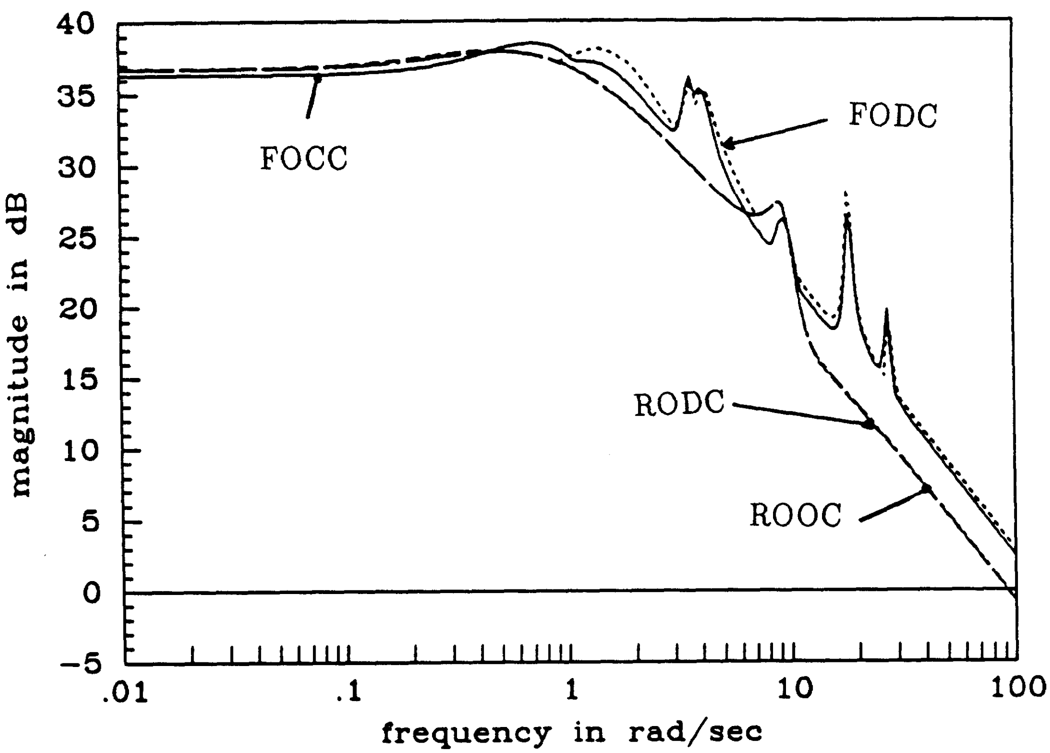

Figure 6.18: Maximum Singular Values of the Compensator Transfer Function: FOCC, FODC, ROOC, RODC, SCOLE Example

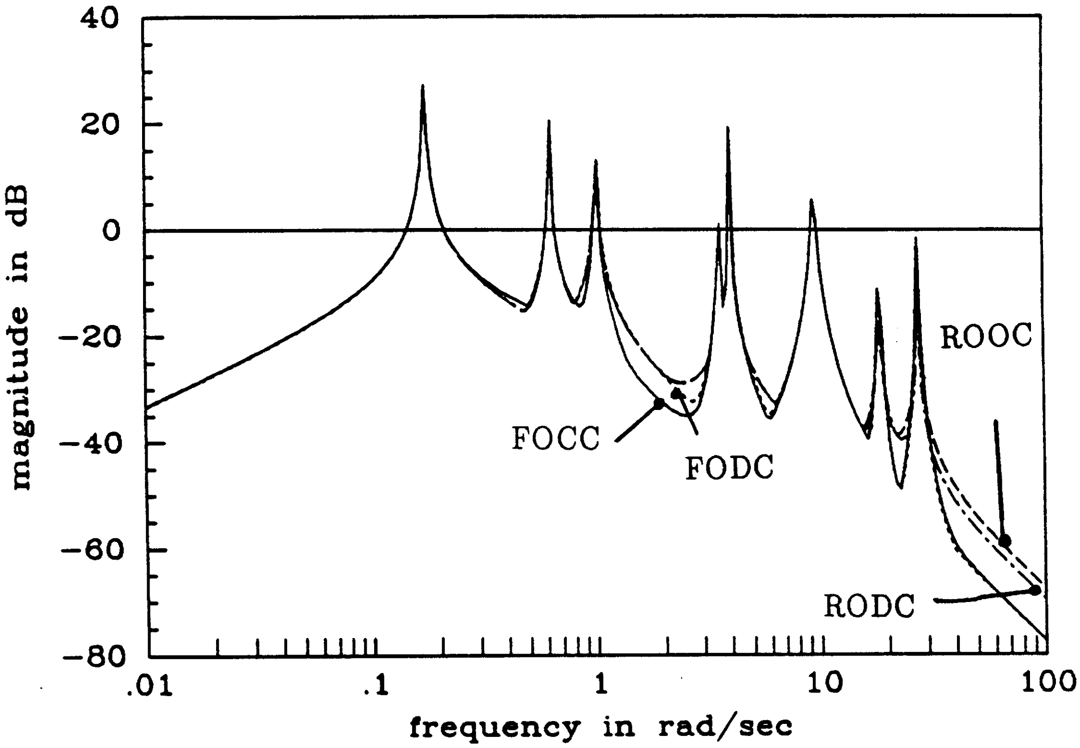

Figure 6.19: Maximum Singular Values of the Loop Transfer Function: FOCC, FODC, ROOC, RODC, SCOLE Example

6.5 CONCLUSION

6.5.1 On the Performance of the Fixed Architecture Controllers

This chapter has shown several examples of reduced order and fixed architecture control designs for realistic systems. Good performance was achieved by the constrained compensators. The nominal performance of the closed loop system is increased when more poles are added to the subcontrollers and when the constraints on the information flow and the control authority are relaxed. The SCOLE example has shown that the order constraints seem to be more important than the remaining architecture constraints, and more benefit is gained by increasing the order of the compensator than by letting sensors and actuators be shared by more than one subcontroller.

6.5.2 On the Convergence of the Algorithm

Inherent properties of the system can make the convergence of the homotopy procedure difficult. The four disk system of Enns being both unstable and nonminimum phase, the problem is numerically badly conditioned. Hence, the algorithm encounters some difficulties when the design tries to obtain high gain solutions. Again, the order of the compensator appears to be a very important parameter for the fast convergence of the algorithm. This was shown both on Enns' system and on the SCOLE. Problems occur when modes having comparable effects on the cost have to be controlled using a compensator whose order is such that it cannot tune itself to both dynamics. This effect was mostly observed on the SCOLE, where the RODC had to find some average way of controlling the first and second bending modes of the system. Excellent convergence properties where found with the Mini-Mast, where no bifurcation or singular point were found. The choice of the architecture appears to be, therefore, of paramount importance for a fast convergence of the algorithm.

6.5.3 On the Choice of the Control Architecture

The SCOLE example has shown that it is possible to obtain subcontrollers with separate bandwidths. The Mini-Mast example, on the other hand, shows that a locally decentralized control architecture, where three similar controllers are used to control three mostly decoupled channels, can also be found. The choice of an adequate control architecture is very important. The choice concerns both the order of the subcontrollers and the information flow in the compensator. As noted in chapter 1 and 2, some systems have asymptotic properties that make near optimal solutions easy to find, and where simpler control structures appear naturally. If these control structures are selected, the near optimal solutions can be improved by solving the constrained optimization problem and the homotopy algorithm is very likely to converge rapidly, as long as the plant is close enough to the ideal system for which the simplified controller is optimal. For weakly coupled systems, a locally decentralized architecture will achieve very high performance until the coupling reaches a limit for which the nature of the controller must change. Until this limit is reached, the homotopy procedure will have a high rate of convergence. For weakly connected systems, a two timescale control structure is near optimal. The slow control requires information and control authority on the global dynamics of the system, while the fast control requires local information and local conrtol authority. If such a structure is respected, the corresponding optimal H_2 fixed architecture controller will optimize the performance and will be found quite rapidly. As the slow and fast modes of the system merge, the control structure may have to change and the homotopy algorithm will encounter numerical difficulties.

A large structure is neither weakly coupled, nor is it weakly connected. Coherent behavior can be observed at all the resonant frequencies. Locally decentralized control should not, therefore, be used. A distribution of the control over the frequencies can, however, be envisioned. Such a scheme appears naturally in the form of independent modal control [Mei87]. The fixed architecture design procedure is, however, much more flexible, and potentially more robust. If the sensors and actuators are grouped in such a way that the resulting information and control authority is concentrated on one particular mode, the corresponding compensator will be tuned to that particular mode. The fact that higher modes are included in the design should prevent spillover.

The bandwidth separation obtained on the SCOLE resulted in poor convergence properties because both subcontrollers had similar information and control authority on the first as well as the second bending modes.

6.5.4 On Robustness

The issue of robustness has not been addressed in the examples. The H_2 fixed architecture control problem does not include any direct attempt at making the design more robust. Hence, the simplified controllers that have been obtained cannot be expected to tolerate more disturbance and perform well for larger uncertainties. An increase in robustness may, however, appear, simply because of the fact that constrained controllers are overall suboptimal solutions of the LQ problem. They will be less finely tuned to the model and may, consequently, tolerate higher level of uncertainties.

A quick assessment of the robustness properties will show, in the case of the SCOLE, that the RODC becomes unstable before the LQG design when the first torsional mode is made extremely soft and almost unstable. The unstability occurs on the second bending modes which are less damped with the RODC. The first bending and rigid modes will, however, change slightly more with the LQG controller. This rapid assessment will prove that improvements may occur, but that, once again, the architecture chosen for the design must be properly chosen, or the constrained design may be in fact less robust to plant uncertainties.

CONCLUSIONS AND RECOMMENDATIONS

7.1 SUMMARY

This thesis has extended the Linear Quadratic Gaussian design techniques to introduce the possibility to constrain the architecture of the compensator. The general form of feedback that it allows is a decentralized processing structure where the controller is made of p subcontrollers having totally uncoupled dynamics, each of which is connected to selected sensors and selected actuators. This general architecture can produce controllers of very different character. Full order, dynamic, centralized controllers, reduced order, dynamic, centralized controllers, decentralized, dynamic, controllers, and multi-timescale controllers all obey the general rules developed to constrain the control architecture. The decentralized static case has also been considered, but it has not been studied in details.

Optimality based techniques, and especially Linear Quadratic, or H_2, techniques, have been very successful at producing Multi-Input, Multi-Output compensators. The generalization of such methods was undertaken to allow the design of simpler feedback structures that follow hard implementation requirements such as limitation in the processing capabilities, complexity in the wiring and in data collecting, or modular assembly necessitating that each module have its own controller.

This thesis has posed the H_2 fixed architecture control problems (dynamic and static) and derived the necessary conditions for optimality for them. These conditions have the form of highly coupled matrix equations. The properties of these equations have been studied and the investigation has focused, in particular, on possible simplifications that could occur because of the structure of the problem. Such simplifications occur in the full order centralized problem, and to a lesser extent, in the reduced order centralized problem.

Homotopy methods have been reported successful at solving complex coupled matrix equations and continuation procedures have been developed to solve the reduced order control problem. This thesis has considered the use of a homotopy algorithm to solve the general H_2 fixed architecture dynamic control problem. The convergence properties have been studied at length and have led to a broader understanding of the number and the nature of the solutions to the constrained problem.

Design examples have shown the performance of simpler controllers for flexible space structures. The examples have also helped understood some of the issues in choosing the architecture of the controller and, more particularly, the effect of limiting the order of the overall compensator.

7.2 THESIS CONTRIBUTION

The contribution of this work has been to broaden the understanding of the effects of the architecture constraints on the H_2 optimal dynamic control problem. The structured conditions obtained in Chapter 4 have shown how the full order problem, which necessitates the solution of the two uncoupled Control and Filter Algebraic Equation (CARE and FARE), becomes more complicated as the order of the dynamics of the compensator is, first, reduced and, next, when the processing is decentralized. The reduction in the order of the compensator couples the filter and the control problem. This was shown in [Hyl84]. The optimal coupling requires the computation of a projection operator, and the CARE and FARE become modified Riccati equations, where the projection cancels out some part of the quadratic terms entering the equations. This thesis has shown that the decentralization of the processing also requires that the different subcontrollers be coordinated, so that they, as a whole, minimize the quadratic cost. The most general fixed architecture control problem requires the simultaneous solution of modified filter and control Riccati equations coupled through four Lyapunov equations which yield the optimal coupling, which is no longer a projection, and the optimal coordination between the subcontrollers.

This work has also shown that the solution of the optimality conditions can be decomposed and that the gains and the dynamics of all the subcontrollers can be solved as functions of the remaining variables in the problem. It is not possible, however, to obtain an analytic expression that would allow for a closed form substitution of these matrices in the remaining equations, in the most general case. Hence, the structured conditions do not lead to any noticeable simplification of the problem. The numerical algorithm developed in this work has, consequently, used the gain and dynamic matrices of the controllers as parameters and in order to solve a parameter optimization problem.

The H_2 fixed architecture control problem has an obvious solution when the system is made of totally independent subsystems. The motivation for using homotopy is that it should be possible to start from such simple solutions and follow the optimal solution as the parameters of the problem are changed from a simpler decoupled form to their actual values. This approach was suggested in [Ric87] for solving the reduced order control problem. This work has extended the idea and developed a procedure for the more general fixed architecture problem. In the process, one has developed an analytic expression for calculating the Hessian, or matrix of second derivatives of the cost with respect to the control parameters, and the nature of the solutions has been studied. The study has shown that the optimality conditions for the constrained LQG problem have many solutions. Some of the solutions yield stable closed loops, while others yield unstable closed loop. Among the stabilizing solutions, which are the only one of interest, some solutions are local minima and others are saddle points. The nature of the solution is not invariant under homotopy. When a critical point is encountered along the solution path, an eigenvalue of the Hessian becomes zero and may change sign. In that case, a minimum can become a saddle and a saddle can become a minimum. Bifurcating solutions may also occur. Another phenomenon is that the optimal solution that initially stabilizes the plant may become, in a continuous fashion, a nonstabilizing solution. The cost goes, in that case, to infinity, but the control parameters remain on a smooth path. A chosen architecture may not produce any stabilizing controllers for a particular system. If the unstable poles of such a system are continuously move in the left hand plane, the problem will then have a stabilizing solution.

Hence, the tracking of the stabilizing solutions only will not guarantee that a possibly stabilizing solution will be found. The homotopy must, therefore, track all the solutions to optimality conditions in order to be a global tool. The alternative taken in this work is to allow for noncontinuous solution path and let the algorithm look for local minima when a critical point is found. When the problem does not have any stabilizing compensator, the numerical scheme will stop converging toward the desired problem and the program will then abort.

The numerical examples have shown that the order constraints are the most stringent. If the order of the compensator is not large enough, the solution cannot tune itself to the dynamics of the system and tries to find an average that is hard to find in most cases. In general, the fixed architecture control problem converges rapidly and produce a high performance feedback if the problem admits near optimal solutions with the particular architecture chosen. For example, simpler near optimal controllers can be obtained for weakly coupled subsystems by ignoring the coupling and solving independent control problems. The optimization of these decentralized controllers will result in better performance of the closed loop system. The choice of the order of each subcontroller must leave enough freedom to let the dynamics of the controller tune itself to those of the system. When the architecture of the controller has too many constraints, the optimization produces a solution which tries to have some average action on different contributions to the cost and may result in poor overall performance. The homotopy is also less likely to converge continuously, since the starting solution has a character much different from that of the solution to the actual problem.

7.3 DIRECTIONS FOR FUTURE WORK

The first important issue to be studied is the choice of the control architecture. Both the order of the subcontrollers and the sensors and actuators that they use greatly influence the overall performance of the design. As shown in Chapter 6, the residual improvement obtained by increasing the order of the compensator may, however, be minimal above some number. The cost component ranking method of [Ske80] is a very interesting idea for selecting the order of the controller, since it breaks up the cost into several contributions from different part of the dynamics of the system. Hence, it can show what parts of the dynamics can be ignored, or need not be controlled. As for the choice of the sensors and actuators, [Ske83] and [Del90] have both proposed some schemes that, both, rely on weak coupling ideas. These studies can lead to general rules for choosing the architecture and should be pursued.

Numerical improvements should also result from a better choice of the architecture. If the unconstrained solution has already a marked decentralized character, the corresponding constrained solution will converge very rapidly. More generally, a more careful study on how to choose the inital problem should be undertaken. Alternate ways of getting the different gradients can also be studied. The current algorithm cannot handle very high order compensators with large number of sensors and actuators, since the number of parameters increases rapidly. The computation is, however, very well suited for parallel processing.

A second important issue that has not been investigated is the issue of robustness. Simplified controllers are only suboptimal if one considers the unconstrained LQG problem. If the architecture has been selected so that only the predominant dynamic effects are controlled, it is possible that the controller will not try to minimize the residual effects which may be the result of higher modes in the system, or coupling between subsystems. The constrained controller will, therefore, be less sensitive to modeling errors which are bound to be higher on the detailed description of the system. Being less finely tuned, the constrained controller may be more robust. This is not a guarantee, however. The H_2 problem itself has been modified to take robustness into consideration right away in the optimization. Considering the modified H_2 cost functional of [Ber87a, Che88], the problem can be generalized with the addition of the architecture constraints. The Filter Laypunov equation of Chapter 3 will become, in that case, a modified Riccati equation. The remaining optimality conditions will be slightly modified, but can still be obtained using matrix calculus as easily they were in the H_2 case.

Finally, the problem can also be extended to the constrained H_2/H_∞ problem. The reduced order case is presented in [Ber89], and the introduction of architecture constraints can be done in a way very similar to the one used for the H_2 problem, once the new cost functional has been defined. Again, a Riccati equation replaces the Filter Lyapunov equation in order to get the H_∞ bound. The numerical aspects of the robust control methods may, however, reach another level in complexity, and these methods may not be very pratical.

REFERENCES

[And71] B.D.O. Anderson, and J.B. Moore, *Linear Optimal Control*, Englewood Cliffs: Prentice-Hall, 1971.

[Aok68] M. Aoki, ' Control of large-scale dynamic systems by aggregation,' *IEEE Trans. Autom. Control*, AC-13, No.3, 1968, pp 246-253.

[Ath68] M. Athans, 'The matrix minimum principle,' *Information and Control*, 11, 1968, pp 592-606.

[Ath86] M. Athans, 'A tutorial on the LQG/LTR method,' *Proc. American Control Conference*, Seattle, WA., 1986.

[Bas75] S. Basuthakur and C.H. Knapp, 'Optimal constant controllers for stochastic linear systems,' *IEEE Trans. Autom. Control*, AC-20, Oct. 1975, pp 664-666.

[Ber85] D.S. Bernstein and D.C. Hyland, 'The optimal projection equations for reduced order state estimation,' *IEEE Trans. Autom. Control*, AC-30, No.6, June 1985, pp 583-585.

[Ber86a] D.S. Bernstein and D.C. Hyland, 'The optimal projection equations for finite dimensional fixed order dynamic compensation of infinite dimensional systems,' *SIAM J. Control and Optimization*, Vol.24, No.1, Jan. 1986, pp 122-151.

[Ber86b] D.S. Bernstein and and S.W. Greeley, 'Robust controller synthesis using the maximum entropy design equations, ' *IEEE Trans. Autom. Control*, AC-31, No.4, Apr. 1986, pp 362-364.

[Ber86c] D.S. Bernstein, L.D. Davis and D.C. Hyland, 'The optimal projection equations for reduced order, discrete time modeling, estimation and control,' *AIAA J. Guidance Control and Dynamics*, Vol. 9, No.3, May-June 1986, pp 288-293.

[Ber86d] D.S. Bernstein, L.D. Davis and S.W. Greeley, 'The optimal projection equations for fixed order sampled data dynamic compensation with computation delay,' *IEEE Trans. Autom. Control*, AC-31, No.9, Sept. 1986, pp 859-862.

[Ber87a] D.S. Bernstein, 'Robust static and dynamic output feedback stabilization: deterministic and stochastic perspectives,' *IEEE Trans. Autom. Control*, AC-32, No.12, Dec. 1987, pp 1076-1084.

[Ber87b] D.S. Bernstein, 'Sequential design of decentralized dynamic compensators using optimal projection equations,' *Int. J. Control*, Vol.46, No.5, 1987, pp 1569-1577.

[Ber89] D.S. Bernstein and W.M. Haddad, ' LQG control with H_∞ performance bound: a Riccati equation approach,' *IEEE Trans. Autom. Control*, AC-34, No.3, March 1989, pp 293-305.

[Bra70] F.M. Brasch and J.B. Pearson, 'Pole placement using dynamic compensators,' *IEEE Trans. Autom. Control*, AC-15, No.1, Feb. 1970, pp 34-43.

[Cal89] A.J. Calise, and V.R.P. Jonnalagadda, 'Fixed order compensator design based on frequency shaped cost functionals,' *AIAA J. Guidance Control and Dynamics*, Vol. 12, No.3, May-June 1989, pp 429-431.

[Cam79] S.L. Campbell and C.D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, London: Pitman, 1979.

[Chi71] M.R. Chidambara and R.B. Schainker, 'Lower order generalized aggregated model and suboptimal control,' *IEEE Trans. Autom. Control*, AC-16, No.3, Apr. 1971, pp 175-180.

[Che88] M. Cheung and S. Yurkovich, 'On the robustness of MEOP versus asymptotic LQG synthesis,' *IEEE Trans. Autom. Control*, AC-33, No.11, Nov. 1988, pp 1061-1065.

[Chg71] C.Y. Chong and M. Athans, 'On the stochastic control of linear systems with different information sets,' *IEEE Trans. Autom. Control*, AC-16, No.5, 1971, pp 423-430.

[Chg76] C.Y. Chong and M. Athans, 'On the periodic coordination of linear stochastic systems,' *Automatica*, Vol. 12, 1976, pp 321-335.

[Chw76] J.H. Chow, and P.V. Kokotovic, 'A decomposition of near optimum regulators for systems with slow and fast modes,' *IEEE Trans. Autom. Control*, AC-21, Oct 1976, pp 701-705.

[Chw82] J.H. Chow and P.V. Kokotovic, 'Time scale modeling of dynamic networks with sparse and weak connections,' in J.H. Chow ed., *Time Scale Modeling of Dynamic Networks with Application to Power Systems*, New York: Springer-Verlag, 1982, pp 310-353.

[Cor76] J.P. Corfmat and A.S. Morse, 'Decentralized control of linear multivariable systems,' *Automatica*, Vol. 12, pp 479-495.

[Dav76a] E.J. Davison, 'The robust control of servomechanism problem for linear time invariant multivariable systems,' *IEEE Trans. Autom. Control*, AC-21, No.1, Feb. 1976, pp 25-34.

[Dav76b] E.J. Davison, 'The robust decentralized control of a general servomechanism problem,' *IEEE Trans. Autom. Control*, AC-21, No.1, Feb. 1976, pp 14-24.

[Daz81] J.J D'azzo, and C.H. Houpis, *Linear Control Systems*, New York: Mc Graw-Hill, second ed., 1981.

[Del90] M.L. Delorenzo, 'Sensor and actuator selection for large space structure control,' *AIAA J. Guidance Control and Dynamics*, Vol.13, No.2, March-April 1990, pp 249-257.

[Doy79] J.C. Doyle, and G. Stein, 'Robustness via Observers,' *IEEE Trans. Autom. Control*, AC-24, No.4, Aug. 1979, pp 607-611.

[Eav83] B.C. Eaves, F.J. Gould, H.O. Peitgen and M.J Todd, *Homotopy Methods and Global Convergence*, New York: Plenum Press, 1983.

[Enn84] D. Enns, *Model Reduction for Control System Design*, PhD thesis, Stanford University, Dept. of Aeronautics and Astronautics, June 1984.

[Fra87] B.A. Francis, *An Course in H_∞ Control Theory,* New York: Springer Verlag, 1987.

[Gan59] F.R. Gantmacher, *The Theory of Matrices*, New York: Chelsea Publishing Company, 1959.

[Glo84] K. Glover, 'All optimal Hankel norm approximations of linear multivariable systems and their L_∞ error bounds,' *Int. J. Control*, Vol. 39, No.6, 1984, pp 1115-1193.

[Gra74] D. Graupe, 'Optimal linear control subject to sensitivity constraints,' *IEEE Trans. Autom. Control*, AC-19, Oct. 1974, pp 593-594.

[Gru86] A. Gruzen, 'Robust reduced order control of flexible structures,' SM Thesis, MIT, Dept. of Aeronautics and Astronautics, Feb. 1986.

[Gup80] N.K. Gupta, 'Frequency shaped cost functionals: extension of linear quadratic gaussian design methods,' *AIAA J. Guidance and Control*, Vol. 3, No.6, 1980, pp 529-535.

[Hal90] S.R. Hall, E.F. Crawley, J. How, and B. Ward, 'A hierarchic control architecture for intelligent structures,' accepted for publication, *AIAA J. Guidance Control and Dynamics*, 1990.

[How90] J. How, 'Local control design methodologies for hierarchic control architecture,' SM Thesis, MIT, Dept. of Aeronautics and Astronautics, Feb. 1990.

[Hod86] M. Hodzic and D.D. Siljak, 'Decentralized estimation and control with overlapping information sets,' *IEEE Trans. Autom. Control*, AC-32, No.1, Jan. 1986, pp 83-86.

[Hyl83] D.S. Hyland, 'The optimal projection equations approach to fixed order compensation: numerical methods and illustrative examples,' *21th AIAA Aerospace Science Meeting*, Reno, NV., Jan. 1983.

[Hyl84] D.S. Hyland and D.C. Bernstein, 'The optimal projection equations for fixed order compensation,' *IEEE Trans. Autom. Control*, AC-29, No.11, Nov. 1984, pp 1034-1037.

[Hyl85] D.C. Hyland and D.S. Bernstein, 'The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton and Moore,' *IEEE Trans. Autom. Control*, AC-30, No.12, Dec. 1985, pp 1201-1211.

[Hyl90] D.C. Hyland and S. Richter, 'On direct versus indirect methods for reduced order controller design,' *IEEE Trans. Autom. Control*, AC-35, No.3, March 1990, pp 377-379.

[Ift87] A. Iftar and Ü. Özgüner, 'Local LQG/LTR controller design for decentralized systems,' *IEEE Trans. Autom. Control*, AC-32, No.10, Oct. 87, pp 926-930.

[Ike80a] M. Ikeda and D.D. Siljak, 'decentralized stabilization of linear time varying systems,' *IEEE Trans. Autom. Control*, AC-25, No.1, Feb. 1980, pp 106-107.

[Ike80b] M. Ikeda and D.D. Siljak, 'Overlapping decompositions expansions and contractions of dynamic systems,' *Large Scale Systems*, Vol. 1, No.1, North-Holland, 1980, pp 29-38.

[Ike84] M. Ikeda, D.D. Siljak, and D. White, 'An inclusion principle for dynamic systems,' *IEEE Trans. Autom. Control*, AC-29, March 1984, pp 244-249.

[Joh70] T.L. Johnson and M. Athans, 'On the design of optimal constrained dynamic compensators for linear constant systems,' *IEEE Trans. Autom. Control*, AC-15, Dec. 1970, pp 658-660.

[Kab83] P.T. Kabamba, 'An integrated approach to reduced order control theory,' *Optimal Control Applications and Methods*, Vol. 4, 1983, pp 405–415.

[Kab87] P.T. Kabamba, 'A homotopy approach to feedback stabilization of linear systems,' *AIAA J. Guidance Control and Dynamics*, Vol. 10, No.5, Sept.-Oct. 1987, pp 422-432.

[Kai80] T. Kailath, *Linear systems,* Englewood Cliffs: Printice-Hall, 1980.

[Kha80] H.K. Khalil and P.V. Kokotovic, 'Decentralized stabilization of systems with slow and fast modes,' *Large Scale Systems*, Vol. 1, 1980, pp 141-148.

[Kha84] H.K. Khalil and Z. Gajic, 'Near optimum regulators for stochastic linear singularly perturbed systems,' *IEEE Trans. Autom. Control*, AC-29, No.6, June 1984.

[Kha87] H.K. Khalil, 'Output feedback control of linear two time scale systems,' *IEEE Trans. Autom. Control*, AC–32, No.9, Sept. 1987, pp 784-792.

[Kir70] D.E. Kirk, *Optimal Control Theory,* Englewood Cliffs: Prentice-Hall, 1970.

[Kra88] F.S. Kramer, and A.J. Calise, 'Fixed order dynamic compensation for multivariable linear systems,' *AIAA J. Guidance Control and Dynamics*, Vol. 11, No.1, Jan-Feb. 1988, pp 80-85.

[Kok81] P.V. Kokotovic, 'Subsystems time scales and multimodeling,' *Automatica*, Vol. 17, No.6, 1981, pp 789-795.

[Kwa72a] H. Kwakernaak and R. Sivan, 'The maximally achievable accuracy of linear optimal regulators and linear optimal filters,' *IEEE Trans. Autom. Control*, AC-17, No. 1, Feb. 1972, pp 79-86.

[Kwa72b] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, New York: Wiley-Interscience, 1972.

[Lef85] S. Lefebvre, S. Richter and R. Decarlo, 'A continuation algorithm for eigenvalue assignment by decentralized constant output,' *Int. J. Control*, Vol. 41, No.5, 1985, pp 1293-1299.

[Lev70] W.S. Levine and M. Athans, 'On the determination of the optimal constant output feedback gains for linear multivariable systems,' *IEEE Trans. Autom. Control*, AC-15, No.1, Feb. 1970, pp 44-48.

[Liu86] Y. Liu and B.D.O. Anderson, 'Controller reduction via stable factorization and balancing,' *Int. J. of Control*, Vol. 44, No.2, 1986, pp 507-531.

[Llo78] N.G. Lloyd, *Degree Theory,* London: Cambridge University Press, 1978.

[Lue69] D.G. Luenberger, *Optimization by Vector Space Methods*, New York: John Wiley & Sons, 1969.

[Lus85] D.W. Luse and H.K. Khalil, 'Frequency domain results for systems with slow and fast dynamics,' *IEEE Trans. Autom. Control*, AC-30, No.12, Dec. 1985, pp 1171-1179.

[Ly82] U.L. Ly, 'A design algorithm for robust low order controller,' SUDAAR 536, Stanford University, Nov. 1982.

[Ly85] U.L. Ly, A.E. Bryson, and R.H. Cannon, 'design of low order compensators using parameter optimization,' *Automatica*, Vol. 21, No.3, 1985, pp 315-318.

[Mac89] J.M. Maciejowski, *Multivariable Feedback Design*, Addison-Wesley, 1989.

[Mei88] L. Meirovitch, 'Control of distributed structures,' in Atluri and Amos (ed.), New York: Springer-Verlag, 1988.

[Men75] J.M. Mendel, and J. Feather, 'On the design of optimal time invariant compensators for linear stochastic time invariant systems,' *IEEE Trans. Autom. Control*, AC-20, No.9, Oct. 1975, pp 653-657.

[Moe85] D.D. Moerder, and A.J. Calise, 'Convergence of a numerical algorithm for calculating optimal output feedback gains,' *IEEE Trans. Autom. Control*, AC-30, No.9, 1985, pp 900-903.

[Mon69] R.C. Montgomery, and H.G. Hatch Jr., 'Application of differential synthesis to design multiaxis stability augmentation systems,' *AIAA J. Aircraft*, Vol. 6, No.4, 1969, pp 336-343.

[Mor81] B.C. Moore, 'Principal component analysis in linear systems: controllability, observability and system reduction,' *IEEE Trans. Autom. Control*, AC-26, No.1, Feb. 1981, pp 17-32.

[Nas87] J.P. Williams, and R.A. Rallo, 'The Spacecraft Control Laboratory Experiment (SCOLE) facility,' NASA Technical Memorandum 89057, Feb. 1987.

[Nas89] R. Pappa et al., *Mini-Mast CSI Testbed User's Guide*, March 1989.

[Oht86] Y. Ohta, D.D. Siljak and T. Matsumoto, 'Decentralized control using quasi-block diagonal dominance of transfer function matrices,' *IEEE Trans. Autom. Control*, AC-31, No.5, May 1986, pp 420-429.

[Opd90] Ph.C. Opdenacker, E.A. Jonckheere, M.G. Safonov, J.C. Juang, and M.S. Lukick, 'Reduced order compensator for flexible structures,' *AIAA J. Control Guidance and Dynamics*, Vol. 13, No.1, Jan.-Feb. 1990, pp 46-56.

[Ozg79] Ü. Özgüner, 'Near optimum control of composite systems: the multi timescale approach,' *IEEE Trans. Autom. Control*, AC-24, No.4, Aug. 1979, pp 652-655.

[Pet90] L.D. Peterson, 'An optimal projection controller for experimental truss structure,' submitted to the *AIAA J. Control Guidance and Dynamics*.

[Pre86] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes*, Cambridge University Press, 1986.

[Ric87] S. Richter, 'Reduced order control design via the optimal projection equations: a homotopy algorithm for global optimality,' *6th AIAA VPI&ISU symposium on dynamics and control of large structures*, Balcksburg, VA., 1987.

[Ric89] S. Richter, and E.G. Collins, 'A homotopy algorithm for reduced order design using the optimal projection equations,' *Proc. of the 28th CDC*, Tampa, FL., Dec. 1989, pp 506-511.

[Ros74] H.H. Rosenbrock, *Computer aided control system design*, London: Academic Press, 1974.

[Sak84] V.R. Saksena, J. O'Reilly, and P.V. Kokotovic, 'Singular perturbations and time scale methods in control theory: Survey 1976-1983,' *Automatica*, Vol. 20, No.3, 1984, pp 273-293.

[San78] N.R. Sandell, P. Varaiya, M. Athans and M. Safonov, 'Survey of decentralized control methods for large scale systems,' *IEEE Trans. Autom. Control*, AC-23, No.2, Apr. 1978, pp 108-128.

[Sca85] L.E. Scales, *Introduction to Nonlinear Optimization*, New York: Springer Verlag, 1985.

[Sch81] D.B. Schaechter, 'Optimal local control of flexible structure,' *AIAA J. Guidance and Control*, Vol. 4, No.1, 1981, pp 22-26.

[Seb86] D.R. Sebok, S. Richter, and R. Decarlo, 'Feedback gain optimization in decentralized eigenvalue assignment,' *Automatica*, Vol. 22, No.4, 1986, pp 433-447.

[Sil73] D.D. Siljak, 'On stability of large scale systems under structural perturbations,' *IEEE Trans. on Systems, Man, and Cybernetics*, July 1973, pp 415-417.

[Sil76] D.D. Siljak and M.B. Vukcevic, 'Decentralization, stabilization, and estimation of large scale systems,' *IEEE Trans. Autom. Control*, AC-21, June 1976, pp 363-366.

[Sil78] D.D. Siljak, *Large Scale Dynamic Systems*, North-Holland, 1978

[Sil79] D.D. Siljak, 'Overlapping decentralized control,' in A. Titli and M.G. Singh, *Handbook of Large Scale Systems Engineering Application*, North-Holland, 1979, pp 145-166.

[Sin75] M.G. Singh, A.W. Drew, and J.F. Coales, 'Comparisons of practical hierarchical conrol methods for interconnected dynamical systems,' *Automatica*, Vol. 11, 1975, pp 331-350.

[Sin78] M.G. Singh, *Decentralised Control*, Vol. 1, North-Holland, 1978.

[Sin80] M.G. Singh, *Dynamical Hierarchical Control*, North-Holland, 1978.

[Sir79] J.M. Siret, G. Michaïlesco, and P. Bertrand, 'On the use of aggregation techniques,' in A.Titli, and M.G. Singh, *Handbook of Large Scale Systems Engineering Application*, North-Holland, 1979, pp 20-37.

[Ske80] R.E. Skelton, 'Cost decomposition of linear systems with application to model reduction,' *Int. J. Control*, Vol. 32, 1980, pp 1031-1055

[Ske83] R.E. Skelton and M.L. Delorenzo, 'Selection of noisy actuators and sensors in linear stochastic systems,' *Large Scale Systems*, Vol. 4, 1983, pp 109-136.

[Ten77] D. Teneketzis and N.R. Sandell Jr., 'Linear regulator design for stochastic systems by a multiple timescale approach,' *IEEE Trans. Autom. Control*, AC-22, No.4, Aug. 77, pp 615-621.

[Wan73] S.H. Wang, and E.J. Davison, 'On the stabilization of decentralized control systems,' *IEEE Trans. Autom. Control*, AC-29, No.5, Oct. 1973, pp 473-478.

[Wen80] C.J. Wenk, and C.H. Knapp, 'Parameter optimization in linear systems with arbitrary constrained controller structure,' *IEEE Trans. Autom. Control*, AC-25, No.3, June 1980, pp 496-500.

[Wes84] G.S. West-Vukocich, E.J. Davison, and P.C. Hughes, 'The decentralized control of large flexible space structures,' *IEEE Trans. Autom. Control*, Vol. AC-29, Oct. 1984, pp 866-879.

[Wil70] D.A. Wilson, 'Optimum solution of model reduction problems,' *Proc. IEE*, Vol. 117, 1970, pp 1161-1165.

APPENDIX A

A.1 Proof of Theorem 4.1

Theorem 4.1: *Let* C, D, X, R *be matrices in* $\mathbb{R}^{m \times m}$, $\mathbb{R}^{n \times n}$, $\mathbb{R}^{m \times n}$ *and* $\mathbb{R}^{m \times n}$ *respectively,* C *and* D *symmetric, positive semidefinite. Let* $E_x = \{ E_1, E_2, \cdots, E_p\}$ *be a family of independent matrices on* $\mathbb{R}^{m \times n}$. *Consider the system of equations:*

$$X = E_x * x,\ x \in \mathbb{R}^p$$
$$\mathrm{Tr} E_j^T C X D = \mathrm{Tr} E_j^T R, \quad j = 1, \cdots, p \tag{4.2.4}$$

where $*$ *follows Definition* 3.1, *and* E_x *defines the subspace where* X *lies. Such a system is a linear system. It has a unique solution if* C *and* D *are definite. It has an infinite number of solutions, otherwise, if:*

$$\mathrm{Ker}(C) \subset \mathrm{Ker}(R^T)$$
$$\mathrm{Ker}(D) \subset \mathrm{Ker}(R) \tag{4.2.5}$$

■

Proof: The system of equations defined by Eq.(4.2.4) is clearly a linear system in the matrix variable X. X is required to be a linear combination of the matrices E_1, E_2. $\cdots$, E_p. Eq.(4.2.4) requires that the projection on E_j of the product CXD be equal to some specified value. The proof of the theorem necessitates some preliminary lemmas.

A.2 Preliminaries

Lemma A.1: *Given an* n×n *matrix* X, *which is symmetric, positive, semidefinite, with rank* p, *there exists an* n×p *matrix* R *which is full column rank, such that:*

$$X = RR^T$$

■

Proof: X being symmetric, it is diagonalizable. Its eigenvalue decomposition is:

$$X = T\begin{bmatrix}\Lambda_p & 0\\ 0 & 0\end{bmatrix}T^T$$

where T is unitary ($TT^T = I$), and where Λ_p is a diagonal matrix with strictly positive entries, [Gan59]. Block partitioning, the first p columns of T can be regrouped in a matrix T_1. Define the matrix $\sqrt{\Lambda}_p$ as the p-dimensional diagonal matrix whose diagonal elements are the square roots of the diagonal elements of Λ_p. The matrix R can then be defined as:

$$R = T_1\sqrt{\Lambda}_p \tag{A.1}$$

R is full column rank since $\sqrt{\Lambda}_p$is nonsingular and since T_1 is full column rank. Hence, it satisfies Lemma A.1. □

Lemma A.2: *Consider two n×n symmetric matrices* X *and* Y *such that* X *is positive semidefinite and* Y *is positive definite. Then,*

$$Tr(XY) \geq 0,$$

and $$Tr(XY) = 0 \text{ if and only if } X = 0_n$$

■

Proof: Y being positive definite, there is, according to Lemma A.1, a nonsingular n×n matrix S such that:

$$Y = SS^T \tag{A.2}$$

Using Eq.(A.2), we get:

$$\begin{aligned}Tr(XY) &= Tr(XSS^T)\\ &= Tr(S^TXS)\end{aligned}$$

X being positive semidefinite, S^TXS is also positive semidefinite. Its trace is equal to the sum of its eigenvalues. Hence, the trace is strictly positive, unless all eigenvalues of X are zero. X being symmetric, this would imply that X is identically zero.

□

Lemma A.3: *Let* C *and* D *be positive, semidefinite, such that:*

$$C = C_1C_1^T, \qquad C_1 \in \mathbb{R}^{m\times k}$$
$$D = D_1D_1^T, \qquad D_1 \in \mathbb{R}^{n\times l}$$

where k *and* l *are the respective ranks of* C *and* D. *Let* R *be a generic* m×n *matrix. Then, if the following is true,*

$$\mathrm{Ker}(C) \subset \mathrm{Ker}(R^T)$$
$$\mathrm{Ker}(D) \subset \mathrm{Ker}(R) \tag{A.3}$$

R *will satisfy*:

$$R = C_1(C_1^TC_1)^{-1}C_1^TRD_1(D_1^TD_1)^{-1}D_1^T \tag{A.4}$$

■

Proof: the existence of C_1 and D_1 is guaranteed by Lemma A.1. $C_1^TC_1$ and $D_1^TD_1$ are both invertible since C_1 and D_1 are both full column rank. $C_1(C_1^TC_1)^{-1}C_1^T$ is an orthogonal projection parallel to Ker(C). Indeed, if v is in Ker(C), then $C_1^Tv = 0$, and if v is perpendicular to Ker(C), $C_1^Tv = v$. Similarly, $D_1(D_1^TD_1)^{-1}D_1^T$ is an orthogonal projection parallel to Ker(D). Consider a general vector v in $\mathbb{R}^n$. v can be written as:

$$v = v_1 + v_2,$$

with $v_1 \in \mathrm{Ker}(D)^\perp$, $v_2 \in \mathrm{Ker}(D)^\perp$. Since $\mathrm{Ker}(D) \subset \mathrm{Ker}(R)$, $Rv = Rv_1$. Hence:

$$R = RD_1(D_1^TD_1)^{-1}D_1^T \tag{A.5}$$

Consider now a vector w in $\mathbb{R}^m$. w can be written as:

$$w = w_1 + w_2,$$

with $w_1 \in \mathrm{Ker}(C)^\perp$, $w_2 \in \mathrm{Ker}(C)^\perp$. Because $\mathrm{Ker}(C) \subset \mathrm{Ker}(R^T)$, $R^Tw = R^Tw$ and:

$$R^T = R^TC_1(C_1^TC_1)^{-1}C_1^T$$

or: $$R = C_1(C_1^TC_1)^{-1}C_1^TR \tag{A.6}$$

Combining Eq.(A.5) and Eq.(A.6), we get the desired equality. □

A.3 Uniqueness

The uniqueness of the solution is obtained by looking at the system of equations:

$$X = \sum_{j=1}^{p} x_jE_j \tag{A.7}$$

$$0 = \mathrm{Tr}E_j^TCXD, \quad j = 1, \cdots, p \tag{A.8}$$

(A.7) is the expanded version the expression $E_x{*}x$. Eqs.(A.7,8) constitute the homogeneous part of Eq.(4.2.4). Consider the following linear combination:

$$L = \sum_{j=1}^{p} x_j\mathrm{Tr}E_j^TCXD$$

$$= \mathrm{Tr}\left(\sum_{j=1}^{p} x_jE_j^TCXD\right) \tag{A.9}$$

One can recognize that the summation is only Eq.(A.7). Eq.(A.8) implies that L is zero. Hence, Eq.(A.9) yields:

$$0 = \mathrm{Tr}(X^TCXD)$$

If D is positive definite, X^TCX must be zero (Lemma A.2). If C is also positive definite, this implies that X must be zero. Hence, the homogeneous part of the linear system admits zero as its unique solution. The system having as many equations as unknowns, this implies that it has one and only one solution. □

A.3 Existence of a Solution when C or D are not Definite

Consider, now, the case where C or D are not definite. Assume that C has rank k and D has rank l. According to Lemma A.1, there exists a m×k dimensional matrix C_1 and a n×l dimensional matrix D_1 such that C_1 and D_1 are both full column rank, and such that:

$$D = D_1 D_1^T$$
$$C = C_1 C_1^T$$

Eq.(4.2.4) becomes

$$\mathrm{Tr} E_j^T R = \mathrm{Tr}(E_j^T C_1 C_1^T X D_1 D_1^T), \; j = 1, \cdots, p$$
$$= \mathrm{Tr}((C_1^T E_j D_1)^T C_1^T X D_1)$$

Define X_1 as: $$X_1 = C_1^T X D_1 \tag{A.10}$$

and define E_j^1 as:

$$E_j^1 = C_1^T E_j D_1 \tag{A.11}$$

The E_j^1 are k×l matrices. X_1 is also a k×l matrix. Assume, further, that Eq.(4.2.5) are satisfied. Lemma A.3 is, therefore satisfied by R. Defining R_1 as:

$$R_1 = (C_1^T C_1)^{-1} C_1^T R D_1 (D_1^T D_1)^{-1}$$

R_1 is an k×l matrix. X_1 satisfies the following system of equations:

$$X_1 = \sum_{j=1}^{p} x_j E_j^1$$
$$\mathrm{Tr} E_j^T C_1^T R_1 D_1 = \mathrm{Tr} E_j^T C_1 X_1 D_1^T, \; j = 1, \cdots, p \tag{A.12}$$

Using the properties of the trace operator, and using Eqs.(10,11), the system becomes:

$$X_1 = \sum_{j=1}^{p} x_j E_j^1$$
$$\mathrm{Tr} E_j^{1T} R_1 = \mathrm{Tr} E_j^{1T} X_1, \; j = 1, \cdots, p \tag{A.13}$$

X_1 satisfies a system of equations similar in every way to that defined in Theorem 4.1, where the matrices that play the role of C and D in that new problem are, respectively, I_k and I_l which are positive definite matrices. The problem defined by Eq.(A.13) has, therefore, one and only one solution. A general solution to Eq.(4.2.4) is:

$$X = C_1(C_1^T C_1)^{-1} X_1 (D_1^T D_1)^{-1} D_1^T + C_2 M D_2^T$$

where M is any $m-k \times n-l$ matrix, and where D_2 spans Ker(D) and C_2 spans Ker(C). □

APPENDIX B

B.1 Statement of the Problem

We consider, in the following, that there is a single compensator in the feedback loop and that G, A_c and K are block partitioned as follows:

$$G = \begin{bmatrix} G_1 & G_2 \end{bmatrix}, \quad A_c = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \tag{B.1}$$

The triplet of matrices (G, A_c, K) can be expanded on a basis of matrices (Definition 3.1) and is completely defined by a column vector ξ containing the free entries of G, A_c and K. Assume that ξ is formed in the following fashion:

$$\xi = \begin{bmatrix} a_{11} \\ k_1 \\ g_1 \\ a_{21} \\ k_2 \\ a_{12} \\ g_2 \\ a_{22} \end{bmatrix} \tag{B.2}$$

where a_{11} is a column vector containing the free entries in A_{11}, k_1 is a column vector containing the free entries in K_1, etc. The Lagrangian can be partitioned accordingly:

$$
L_{\xi\xi} = \begin{bmatrix}
L_{a_{11}a_{11}} & L_{a_{11}k_1} & L_{a_{11}g_1} & L_{a_{11}a_{21}} & L_{a_{11}k_2} & L_{a_{11}a_{12}} & L_{a_{11}g_2} & L_{a_{11}a_{22}} \\
L_{k_1a_{11}} & L_{k_1k_1} & L_{k_1g_1} & L_{k_1a_{21}} & L_{k_1k_2} & L_{k_1a_{12}} & L_{k_1g_2} & L_{k_1a_{22}} \\
L_{g_1a_{11}} & L_{g_1k_1} & L_{g_1g_1} & L_{g_1a_{21}} & L_{g_1k_2} & L_{g_1a_{12}} & L_{g_1g_2} & L_{g_1a_{22}} \\
L_{a_{21}a_{11}} & L_{a_{21}k_1} & L_{a_{21}g_1} & L_{a_{21}a_{21}} & L_{a_{21}k_2} & L_{a_{21}a_{12}} & L_{a_{21}g_2} & L_{a_{21}a_{22}} \\
L_{k_2a_{11}} & L_{k_2k_1} & L_{k_2g_1} & L_{k_2a_{21}} & L_{k_2k_2} & L_{k_2a_{12}} & L_{k_2g_2} & L_{k_2a_{22}} \\
L_{a_{12}a_{11}} & L_{a_{12}k_1} & L_{a_{12}g_1} & L_{a_{12}a_{21}} & L_{a_{12}k_2} & L_{a_{12}a_{12}} & L_{a_{12}g_2} & L_{a_{12}a_{22}} \\
L_{g_2a_{11}} & L_{g_2k_1} & L_{g_2g_1} & L_{g_2a_{21}} & L_{g_2k_2} & L_{g_2a_{12}} & L_{g_2g_2} & L_{g_2a_{22}} \\
L_{a_{22}a_{11}} & L_{a_{22}k_1} & L_{a_{22}g_1} & L_{a_{22}a_{21}} & L_{a_{22}k_2} & L_{a_{22}a_{12}} & L_{a_{22}g_2} & L_{a_{22}a_{22}}
\end{bmatrix}
$$

B.2 Uncontrollable Compensator

Theorem B.1: *Assume that $A_{21} = 0$ and $K_2 = 0$. The modes corresponding to A_{22} are uncontrollable from K. The compensator matrices have the form:*

$$
G = [G_1\ G_2],\quad A_c = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},\quad K = \begin{bmatrix} K_1 \\ 0 \end{bmatrix} \tag{B.3}
$$

The following equations hold in that case:

$$
\begin{array}{lll}
0 = L_{a_{11}a_{12}} & 0 = L_{a_{11}g_2} & 0 = L_{a_{11}a_{22}} \\
0 = L_{k_1a_{12}} & 0 = L_{k_1g_2} & 0 = L_{k_1a_{22}} \\
0 = L_{g_1a_{12}} & 0 = L_{g_1g_2} & 0 = L_{g_1a_{22}} \\
0 = L_{a_{12}a_{12}} & 0 = L_{a_{12}g_2} & 0 = L_{a_{12}a_{22}} \\
0 = L_{g_2a_{12}} & 0 = L_{g_2g_2} & 0 = L_{g_2a_{22}} \\
0 = L_{a_{22}a_{12}} & 0 = L_{a_{22}g_2} & 0 = L_{a_{22}a_{22}}
\end{array}
$$

∎

Proof: the matrix A_{cl}, P, Q and V_{cl} have special forms when the compensator is uncontrollable. Extending the partitioning of Eq.(3.3.2), and using Eq.(B.3), we get:

$$A_{cl} = \begin{bmatrix} A & BG_1 & BG_2 \\ K_1C & A_{11} & A_{12} \\ 0 & 0 & A_{22} \end{bmatrix}, \quad V_{cl} = \begin{bmatrix} V & 0 & 0 \\ 0 & K_1V_1K_1^T & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.4}$$

The corresponding Q becomes:

$$Q = \begin{bmatrix} Q_{00} & Q_{01} & 0 \\ Q_{10} & Q_{11} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.5}$$

Consider, now, a general perturbation of the compensator corresponding to a variation ΔA_{12} in A_{12}. Every matrix with superscript " ^{1} " corresponds to the derivative of that particular matrix in the direction ΔA_{12}.

$$A_{cl}^1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \Delta A_{12} \\ 0 & 0 & 0 \end{bmatrix}, \quad V_{cl}^1 = 0, \quad R_{cl}^1 = 0 \tag{B.6}$$

The matrix Q^1 satisfies the Lyapunov equation obtained by differentiating Eq.(3.3.20):

$$0 = A_{cl}Q^1 + Q^1A_{cl}^T + V_{cl}^1 + A_{cl}^1Q + QA_{cl}^{1T} \tag{B.7}$$

Given the form of Q and A_{cl}^{1T}, the term $A_{cl}^1Q + QA_{cl}^{1T}$ is zero. Similarly, P^1 satisfies:

$$0 = A_{cl}^TP^1 + P^1A_{cl} + R_{cl}^1 + A_{cl}^{1T}P + PA_{cl}^1 \tag{B.8}$$

R_{cl}^1 is zero (no dependence on A_{12}) and:

$$A_{cl}^{1T}P + PA_{cl}^1 = \begin{bmatrix} 0 & 0 & P_{01}\Delta A_{12} \\ 0 & 0 & P_{11}\Delta A_{12} \\ \Delta A_{12}^TP_{10} & \Delta A_{12}^TP_{11} & \Delta A_{12}^TP_{12}+P_{21}\Delta A_{12} \end{bmatrix}$$

Because of the form of A_{cl}, Eq.(B.4), the corresponding P^1 will be:

$$P^1 = \begin{bmatrix} 0 & 0 & P_{02} \\ 0 & 0 & P_{12} \\ P_{20} & P_{21} & P_{22} \end{bmatrix} \tag{B.9}$$

Hence Q^1 is zero. Using Eqs.(B.5,9) and the fact that $Q_1 = 0$, the matrix M^1 becomes:

$$\begin{aligned} M^1 &= P^1Q + PQ^1 \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ M_{20}{}^1 & M_{21}{}^1 & 0 \end{bmatrix} \end{aligned} \tag{B.10}$$

Hence, $M_{10}{}^1 = 0$, $M_{01}{}^1 = 0$, $M_{11}{}^1 = 0$, $M_{02}{}^1 = 0$, $M_{22}{}^1 = 0$ and $M_{12}{}^1 = 0$ for any perturbation in A_{12}. The matrix equation L_{A_c}, L_K and L_G depend on P, Q, G, A_c and K. Since they are matrix equations, one can look at some blocks only. The block partitioning follows the same rules as in Eqs.(3.3.8-13) or Proposition 3.1. When the matrices are varied in only one direction (corresponding, for example, to a given ΔA_{12}), the derivatives of L_{A_c}, L_K and L_G are matrices of similar dimensions that will be denoted by the superscript " ^{1} ". Using Eq.(3.3.8-10) and using Eq.(B.10), we get:

$$L_{A_{11}}^1 = M_{11}{}^1 = 0$$

$$L_{A_{12}}^1 = M_{12}{}^1 = 0$$

$$L_{A_{22}}^1 = M_{22}{}^1 = 0$$

This being true for any ΔA_{12}, and writing the various equations in vector form, we get the following equations:

$$0 = L_{a_{11}a_{12}}$$

$$0 = L_{a_{12}a_{12}}$$

$$0 = L_{a_{22}a_{12}}$$

G^1 is zero (no dependence on A_{12}), and Q^1 is zero. Hence, differentiating Eq.(3.3.9):

$$L_G{}^1 = B^T M_{0c}{}^1$$
$$= [B^T M_{01}{}^1 \; B^T M_{02}{}^1]$$

$M_{01}{}^1$ and $M_{02}{}^1$ being zero, Eq.(B.10), we obtain:

$$L_G{}^1 = 0$$

This being true for any ΔA_{12}, we have:

$$0 = L_{g_1 a_{12}}$$
$$0 = L_{g_2 a_{12}}$$

Finally, differentiating Eq.(3.3.10):

$$L_K{}^1 = P_{cc}{}^1 K V_c + M_{c0} C^T$$

$$\begin{bmatrix} L_{K_1}{}^1 \\ L_{K_2}{}^1 \end{bmatrix} = \begin{bmatrix} 0 & P_{12}{}^1 \\ P_{21}{}^1 & P_{22}{}^1 \end{bmatrix} \begin{bmatrix} K_1 V_{c1} \\ 0 \end{bmatrix} + \begin{bmatrix} M_{10} \\ M_{20} \end{bmatrix} C^T$$

Thus, $L_{K_1}{}^1 = 0$. This holds for any ΔA_{12}, therefore:

$$0 = L_{k_1 a_{12}}$$

In summary,

$$L_{a_{11} a_{12}} = 0, \; L_{a_{12} a_{12}} = 0, \; L_{a_{22} a_{12}} = 0, \; L_{g_1 a_{12}} = 0, \; L_{g_2 a_{12}} = 0, \; L_{k_1 a_{12}} = 0.$$

Similarly, one can consider a perturbation corresponding to a variation ΔG_2 in G_2. Computing $A_{cl}{}^1$, $V_{cl}{}^1$, $R_{cl}{}^1$, P^1, Q^1, M^1, etc. for such a perturbation, and using the special form of Q and A_{cl}, one would obtain the required equalities for

$L_{a_{11}g_2}$, $L_{k_1g_2}$ etc. Variations in A_{22} are handled in a similar way. The algebra is straightforward and the proof is not developed any further. □

B.3 Unobservable Compensator

Theorem B.2: *Assume that* $G_2 = 0$ *and* $A_{12} = 0$. *The modes corresponding to* A_{22} *are now unobservable from* G. *The compensator matrices have the form:*

$$G = \begin{bmatrix} G_1 & 0 \end{bmatrix}, \; A_c = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}, \; K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}$$

The following equations hold in that case:

$$\begin{array}{lll}
0 = L_{a_{11}a_{21}} & 0 = L_{a_{11}k_2} & 0 = L_{a_{11}a_{22}} \\
0 = L_{k_1a_{21}} & 0 = L_{k_1k_2} & 0 = L_{k_1a_{22}} \\
0 = L_{g_1a_{21}} & 0 = L_{g_1k_2} & 0 = L_{g_1a_{22}} \\
0 = L_{a_{21}a_{21}} & 0 = L_{a_{21}k_2} & 0 = L_{a_{21}a_{22}} \\
0 = L_{k_2a_{21}} & 0 = L_{k_2k_2} & 0 = L_{k_2a_{22}} \\
0 = L_{a_{22}a_{21}} & 0 = L_{a_{22}k_2} & 0 = L_{a_{22}a_{22}}
\end{array}$$

■

Proof: the proof follows the same steps as for the uncontrollable case. The roles of P and Q must however by changed. A rigorous method for obtaining the proof is to consider the dual problem which is exactly the case treated above. This is equivalent to exchanging the roles of A_{12} and A_{21}, B and C, G and K, P and Q.

□

APPENDIX C

The Two Coupled Beam Example

Compensator Realization, K = 10.0

```
AC1       =

  Columns   1 thru   3
 -2.079035073588129D-01   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00  -2.308225075874774D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00  -2.031506227256257D+00
  0.000000000000000D+00   0.000000000000000D+00  -8.857060679923778D+00

  Columns   4 thru   4
  0.000000000000000D+00
  0.000000000000000D+00
  8.857060679923778D+00
 -2.031506227256257D+00

AC2       =

  Columns   1 thru   3
 -9.534676858229525D-01   5.529691019591561D-01   0.000000000000000D+00
 -5.529691019591561D-01  -9.534676858229525D-01   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00  -2.628840477848499D+00
  0.000000000000000D+00   0.000000000000000D+00  -6.405916077110516D+00

  Columns   4 thru   4
  0.000000000000000D+00
  0.000000000000000D+00
  6.405916077110516D+00
 -2.628840477848499D+00
```

```
G          =

  Columns    1 thru    3
-9.018343190196548D+01   7.984149913067382D+01   4.271087093007880D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    4 thru    6
-7.030506036098839D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00  -2.102325903652823D+02   7.177298514734333D+01

  Columns    7 thru    8
 0.000000000000000D+00   0.000000000000000D+00
-1.250736350364199D+01  -1.253602048239188D+01
```

```
K          =

  5.947262021402900D-03   0.000000000000000D+00
  6.919208363010251D-02   0.000000000000000D+00
 -2.909482661829280D-01   0.000000000000000D+00
 -1.269356531593968D+00   0.000000000000000D+00
  0.000000000000000D+00  -1.273757948047070D-02
  0.000000000000000D+00   3.408798711234530D-02
  0.000000000000000D+00  -1.227401344698684D-01
  0.000000000000000D+00   6.239097032830571D-01
```

Compensator Realization,   K = 5.7

From Forward Integration

```
AC1        =

  Columns    1 thru    3
 0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   1.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
-7.029296936916157D+01  -2.873375079835937D+02  -7.914060682540577D+01

  Columns    4 thru    4
 0.000000000000000D+00
 0.000000000000000D+00
 1.000000000000000D+00
-1.122716676771891D+01
```

```
AC2          =

  Columns   1 thru   3
 0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   1.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
-2.391532215818170D+01  -2.287624710193553D+02  -6.470728600941434D+01

  Columns   4 thru   4
 0.000000000000000D+00
 0.000000000000000D+00
 1.000000000000000D+00
-1.188737140251781D+01

G            =

  Columns   1 thru   3
-1.684897187396559D+01   5.256643955089896D+02  -2.412383780909826D+01
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns   4 thru   6
 2.703235393689954D+01   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00  -4.857211024467728D+01   2.708882140765059D+01

  Columns   7 thru   8
 0.000000000000000D+00   0.000000000000000D+00
-6.600292621652838D+01   1.145331298540436D+00


K            =

   0.    0.
   0.    0.
   0.    0.
   1.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    1.
```

From Backward Integration

```
AC1        =

  Columns    1 thru    3
  0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   1.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 -1.103980332226334D+02  -2.970714396558536D+02  -8.062893898735915D+01

  Columns    4 thru    4
  0.000000000000000D+00
  0.000000000000000D+00
  1.000000000000000D+00
 -1.140356634338026D+01


AC2        =

  Columns    1 thru    3
  0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   1.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 -8.172508020674609D+00  -2.029796525695422D+01  -2.810074149284834D+01

  Columns    4 thru    4
  0.000000000000000D+00
  0.000000000000000D+00
  1.000000000000000D+00
 -3.542222040836909D+00


G          =

  Columns    1 thru    3
  3.665326060594805D+01   5.181627406219531D+02  -2.163142482950375D+01
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    4 thru    6
  2.673973951004057D+01   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00  -1.200677645050274D+01   9.708653820094518D+00

  Columns    7 thru    8
  0.000000000000000D+00   0.000000000000000D+00
 -1.212832336663966D+01  -3.343454231654758D+00
```

$$K = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

Compensator Realization, K = 2.0

From Forward Integration

```
AC1          =

 Columns    1 thru    3
 0.000000000000000D+00    1.000000000000000D+00    0.000000000000000D+00
 0.000000000000000D+00    0.000000000000000D+00    1.000000000000000D+00
 0.000000000000000D+00    0.000000000000000D+00    0.000000000000000D+00
-5.377240039142103D+01   -2.589761448167216D+02   -6.927498169338709D+01

 Columns    4 thru    4
 0.000000000000000D+00
 0.000000000000000D+00
 1.000000000000000D+00
-1.293760322819363D+01

AC2          =

 Columns    1 thru    3
 0.000000000000000D+00    1.000000000000000D+00    0.000000000000000D+00
 0.000000000000000D+00    0.000000000000000D+00    1.000000000000000D+00
 0.000000000000000D+00    0.000000000000000D+00    0.000000000000000D+00
-5.107853294669085D+01   -2.567588830829304D+02   -7.012270725157208D+01

 Columns    4 thru    4
 0.000000000000000D+00
 0.000000000000000D+00
 1.000000000000000D+00
-1.721950629700886D+01
G            =

 Columns    1 thru    3
-4.463482727234136D+01    3.558100923334868D+02   -4.135557038846034D+01
 0.000000000000000D+00    0.000000000000000D+00    0.000000000000000D+00

 Columns    4 thru    6
 2.086203640235739D+01    0.000000000000000D+00    0.000000000000000D+00
 0.000000000000000D+00   -7.375153371612239D+01    2.503599764242774D+02

 Columns    7 thru    8
 0.000000000000000D+00    0.000000000000000D+00
-5.267945704741912D+01    1.062395685978150D+01
```

$$K = \begin{bmatrix} 0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix}^T$$

From Backward Integration

```
AC1          =

  Columns   1 thru   3
 0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   1.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
-5.562832156014938D+01  -2.572736514448433D+02  -6.920396153950946D+01

  Columns   4 thru   4
 0.000000000000000D+00
 0.000000000000000D+00
 1.000000000000000D+00
-1.277191003374923D+01
AC2          =

  Columns   1 thru   3
 0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   1.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
-2.688681266883717D+00  -9.554476828248496D+00  -1.365350590976185D+01

  Columns   4 thru   4
 0.000000000000000D+00
 0.000000000000000D+00
 1.000000000000000D+00
-2.749640076610852D+00
G            =

  Columns   1 thru   3
-4.145852325290848D+01   3.576144134559132D+02  -4.094107595508636D+01
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns   4 thru   6
 2.099794814086120D+01   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00  -3.921259984394241D+00   5.406979739845054D+00

  Columns   7 thru   8
 0.000000000000000D+00   0.000000000000000D+00
 4.452324543692658D+00  -1.396126917737884D+00
```

$$K = \begin{bmatrix} 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}^T$$

APPENDIX D

The Mini-Mast Example

A, B and C Matrices

```
A          =

 Columns    1 thru    3
 0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
-7.297183768900000D+02  -3.873707220000000D+00  -5.835000000000000D-01
 0.000000000000000D+00   0.000000000000000D+00  -3.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

 Columns    4 thru    6
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
-2.892073284000000D+01  -3.474058800000000D-01   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00  -1.477941136000000D+03
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
```

```
 Columns    7 thru    9
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   1.849800000000000D-02   0.000000000000000D+00
  1.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 -1.829934400000000D+00   8.169900000000000D-03   0.000000000000000D+00
  0.000000000000000D+00  -3.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00  -2.883904804000000D+01
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   7.645500000000000D-03   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

 Columns   10 thru   12
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  1.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 -2.287705200000000D-01   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   1.000000000000000D+00
  0.000000000000000D+00  -1.501392004840000D+03  -7.749560000000000D-01
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

 Columns   13 thru   13
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
  8.169900000000000D-03
  0.000000000000000D+00
  0.000000000000000D+00
  1.236600000000000D-02
  0.000000000000000D+00
 -7.645500000000000D-03
 -3.000000000000000D+00
```

```
B          =

   0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
   1.945000000000000D-01   0.000000000000000D+00   0.000000000000000D+00
   1.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
   0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
   0.000000000000000D+00  -6.166000000000000D-03   0.000000000000000D+00
   0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
   0.000000000000000D+00  -2.723300000000000D-03  -2.723300000000000D-03
   0.000000000000000D+00   1.000000000000000D+00   0.000000000000000D+00
   0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
   0.000000000000000D+00   0.000000000000000D+00  -4.122000000000000D-03
   0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
   0.000000000000000D+00  -2.548500000000000D-03   2.548500000000000D-03
   0.000000000000000D+00   0.000000000000000D+00   1.000000000000000D+00


C          =

    Columns   1 thru   3
  -5.155723865871105D-01   0.000000000000000D+00   0.000000000000000D+00
   1.217340584459821D-02   0.000000000000000D+00   0.000000000000000D+00
   1.328160885211273D-01   0.000000000000000D+00   0.000000000000000D+00

    Columns   4 thru   6
  -4.637835277087724D-02   0.000000000000000D+00   3.448568984397917D-01
   6.867426749066858D-01   0.000000000000000D+00   7.980047502648306D-01
   1.310616739232647D-01   0.000000000000000D+00  -1.435335034713333D+00

    Columns   7 thru   9
   0.000000000000000D+00   0.000000000000000D+00   9.479046392193511D-02
   0.000000000000000D+00   0.000000000000000D+00  -4.630819187440592D-01
   0.000000000000000D+00   0.000000000000000D+00  -7.057893194448557D-01

    Columns  10 thru  12
   0.000000000000000D+00   1.738390029706574D-02   0.000000000000000D+00
   0.000000000000000D+00   7.225096195535450D-01   0.000000000000000D+00
   0.000000000000000D+00   1.208747919990667D+00   0.000000000000000D+00

    Columns  13 thru  13
   0.000000000000000D+00
   0.000000000000000D+00
   0.000000000000000D+00
```

Compensator Realization

```
AC         =

  Columns    1 thru    3
 -1.180454067738923D+02   1.788637642899564D+02   0.000000000000000D+00
 -1.699115055592872D+02  -1.735127477625835D+02   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00  -1.167665882744794D+02
  0.000000000000000D+00   0.000000000000000D+00   1.505594313684561D+02
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    4 thru    6
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  1.000736033206352D+02   0.000000000000000D+00   0.000000000000000D+00
 -1.717545570693279D+02   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00  -1.320299207529957D+02   1.617572786991442D+02
  0.000000000000000D+00   6.039716036385999D+02  -7.557162019544356D+02


G          =

  Columns    1 thru    3
 -1.042057101864683D+02  -5.715115708579130D+01   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00  -1.835434551904717D+03
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    4 thru    6
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  1.235443462786233D+03   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00  -2.488179791511258D+03   1.308543103519688D+03


K          =

 -7.415910802774555D+04   0.000000000000000D+00   0.000000000000000D+00
  2.176856265093304D+04   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00  -1.472271285185402D+03   0.000000000000000D+00
  0.000000000000000D+00   4.050313703112707D+03   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   8.104063369230693D+02
  0.000000000000000D+00   0.000000000000000D+00  -3.899472335264596D+03
```

APPENDIX E

The SCOLE Example

A, B, C, and H Matrices

$$A = \text{Blockdiag}(\ A_1,\ A_2,\ A_3,\ A_4,\ A_5,\ A_6,\ A_7,\ A_8)$$

A1 =

0.000000000000000D+00	1.000000000000000D+00
-3.028330844100000D-02	-3.480420000000000D-04

A2 =

0.000000000000000D+00	1.000000000000000D+00
-1.606133867560000D+01	-8.015320000000000D-03

A3 =

0.000000000000000D+00	1.000000000000000D+00
-3.932305805610000D-01	-1.254162000000000D-03

A4 =

0.000000000000000D+00	1.000000000000000D+00
-1.263660304000000D+01	-7.109600000000000D-03

A5 =

0.000000000000000D+00	1.000000000000000D+00
-1.018787422500000D+00	-2.018700000000000D-03

A6 =

0.000000000000000D+00	1.000000000000000D+00
-9.048537326440000D+01	-1.902476000000000D-02

A7 =

0.000000000000000D+00	1.000000000000000D+00
-3.421131136900000D+02	-3.699260000000000D-02

A8 =

0.000000000000000D+00	1.000000000000000D+00
-7.576916864400000D+02	-5.505240000000000D-02

B =

Columns 1 thru 3

0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
1.952620000000000D-02	7.359277000000000D-03	1.455764000000000D-02
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
-3.194602000000000D-02	4.868063000000000D-02	-8.035440000000000D-04
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
-2.034833000000000D-02	-8.952349000000000D-03	2.044872000000000D-02
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
1.183407000000000D-03	-2.049821000000000D-03	-2.005502000000000D-02
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
1.420762000000000D-02	8.294947000000000D-03	9.368484000000000D-03
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
-2.945981000000000D-04	8.194912000000000D-03	-5.244127000000000D-04
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
-5.629678000000000D-03	-1.693750000000000D-01	1.822702000000000D-03
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
3.279740000000000D-03	1.416774000000000D-01	3.040701000000000D-03

Columns 4 thru 6

0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
5.102775000000000D-03	1.802747000000000D-02	1.587412000000000D-02
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
1.368346000000000D-03	8.067914000000000D-04	2.128822000000000D-03
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
9.192607000000000D-03	-1.241117000000000D-03	-2.486406000000000D-03
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
4.581372000000000D-02	1.784269000000000D-04	2.524303000000000D-03
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
6.180635000000000D-03	-2.488448000000000D-02	-2.694900000000000D-02
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
1.804860000000000D-02	3.466187000000000D-03	-2.727312000000000D-01
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
1.016765000000000D-01	1.223136000000000D-05	-9.534877000000000D-04
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
2.413436000000000D-01	9.559862000000000D-05	-5.967419000000000D-03

$C = B^T$

```
H          =

  Columns    1 thru    3
  1.216689300000000D-02   0.000000000000000D+00  -8.062661000000000D-02
  9.454875000000000D-03   0.000000000000000D+00  -2.171888400000000D-03
  2.153340000000000D-03   0.000000000000000D+00  -1.322029300000000D-03
  1.313399080000000D+00   0.000000000000000D+00   1.769117300000000D-01
 -1.531592130000000D+00   0.000000000000000D+00  -2.807725100000000D+00

  Columns    4 thru    6
  0.000000000000000D+00  -1.139597100000000D-02   0.000000000000000D+00
  0.000000000000000D+00   1.125612300000000D-02   0.000000000000000D+00
  0.000000000000000D+00   1.245288000000000D-03   0.000000000000000D+00
  0.000000000000000D+00   1.464799810000000D+00   0.000000000000000D+00
  0.000000000000000D+00   1.527695060000000D+00   0.000000000000000D+00

  Columns    7 thru    9
  3.233233000000000D-03   0.000000000000000D+00   5.912673000000000D-03
 -6.586874000000000D-02   0.000000000000000D+00   3.187848000000000D-03
 -2.345876000000000D-03   0.000000000000000D+00   2.064510000000000D-03
  3.110866500000000D+00   0.000000000000000D+00   2.628658300000000D-01
  5.794685000000000D-02   0.000000000000000D+00  -8.768758300000000D-01

  Columns   10 thru   12
  0.000000000000000D+00  -8.489510100000000D-03   0.000000000000000D+00
  0.000000000000000D+00  -1.857300180000000D-02   0.000000000000000D+00
  0.000000000000000D+00   2.761971890000000D-01   0.000000000000000D+00
  0.000000000000000D+00  -4.705387627000000D+00   0.000000000000000D+00
  0.000000000000000D+00   2.879815417000000D+00   0.000000000000000D+00

  Columns   13 thru   15
  1.637451590000000D-01   0.000000000000000D+00  -1.383973080000000D-01
 -9.985369300000000D-02   0.000000000000000D+00  -2.383023060000000D-01
  9.656968300000000D-04   0.000000000000000D+00   6.062791960000000D-03
 -7.366108000000000D-02   0.000000000000000D+00   4.558728200000000D-01
 -1.131647800000000D-01   0.000000000000000D+00  -2.509094600000000D-01

  Columns   16 thru   16
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
```

Compensator Realization

Full Order Decentralized Controller (FODC)

```
AC1          =

  Columns    1 thru    3
 -9.442111804854230D-01   1.150156244784270D+00   0.000000000000000D+00
 -1.291528224232500D+00  -6.325646197810100D-01   1.060665439191030D-01
  0.000000000000000D+00   1.896260416589490D-01  -9.853691562892960D-01
  0.000000000000000D+00   0.000000000000000D+00  -4.578613503150420D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    4 thru    6
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  3.601014768050760D+00   0.000000000000000D+00   0.000000000000000D+00
 -2.149474009604760D-01  -2.979684296441010D-02   0.000000000000000D+00
 -6.104116706770492D-02  -1.940899138486310D-02   9.937930998759980D-01
  0.000000000000000D+00  -3.422079936873900D+02  -8.550553298981040D-01
AC2          =

  Columns    1 thru    3
 -1.011286073950710D+00   8.245857412760270D-01   0.000000000000000D+00
 -1.762356360014240D+00  -4.548275274844190D-01  -1.312517505533509D-02
  0.000000000000000D+00  -3.048697063421140D-01  -3.726679850544660D-01
  0.000000000000000D+00   0.000000000000000D+00  -3.861152826503860D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    4 thru    6
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  3.316180448878990D+00   0.000000000000000D+00   0.000000000000000D+00
 -3.348763257360610D-01  -1.484074835386789D-02   0.000000000000000D+00
  1.016718806673070D-01  -1.938862393176550D-02   1.007956219435680D+00
  0.000000000000000D+00  -7.577336634226750D+02  -1.592322681572410D+00
AC3          =

  Columns    1 thru    3
 -1.217437911953570D+00   1.001366049144530D-01   0.000000000000000D+00
 -6.447379652110310D-01  -2.513115776063470D-01  -1.020548460226800D+01
  0.000000000000000D+00   4.999311791407980D-01  -1.227581885589800D+00
  0.000000000000000D+00   0.000000000000000D+00  -9.065255817245430D+01

  Columns    4 thru    4
  0.000000000000000D+00
  0.000000000000000D+00
  9.320708504391700D-01
 -1.189189995799740D+00
```

```
G        =

  Columns     1 thru     3
 -1.497627168137560D+00  -2.179687253573830D+01  -2.427113332379030D+01
  7.046139185896553D-02  -5.621774018587642D-02   1.832527058947070D-01
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns     4 thru     6
  2.379891497353870D+01   1.547789455791690D+00   8.642564350005040D+00
 -2.784053694782230D-01  -1.509630792133590D+00   1.257462205920120D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns     7 thru     9
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  1.035168750928830D+01  -3.567322228675800D+01   9.432580519070980D+00
  1.518338505818330D-01  -7.160007973207559D-02  -9.580426753165213D-02
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    10 thru    12
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 -2.780796213405830D+01  -7.834760211794250D-01  -2.652217882497580D+00
  1.700614431789550D-01   9.025940009454570D-01  -1.018524203758040D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    13 thru    15
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 -2.551211780055560D+01  -1.297039603446080D+00  -1.708111078300700D+01
 -1.214572432472930D-01   1.069008445369250D+00   6.624425031834920D-01

  Columns    16 thru    16
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
  0.000000000000000D+00
 -5.582000963808510D+00
  2.014023680910640D+00
```

```
K          =

  Columns    1 thru    3
 -1.675990399772020D+00  -3.549850354123820D-01   0.000000000000000D+00
  3.049935635918950D+00   1.761095041762710D+00   0.000000000000000D+00
  2.349473724890200D-01  -1.258180876019330D+00   0.000000000000000D+00
 -1.609220278219210D+00   9.733671141093010D-01   0.000000000000000D+00
 -2.350279291770254D-04  -1.111273900775998D-02   0.000000000000000D+00
 -1.976656156378850D-01  -2.564598723191240D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00  -8.638884495154450D-01
  0.000000000000000D+00   0.000000000000000D+00   1.412744072409080D+00
  0.000000000000000D+00   0.000000000000000D+00   1.789904677837310D-01
  0.000000000000000D+00   0.000000000000000D+00   9.204970896987820D-01
  0.000000000000000D+00   0.000000000000000D+00  -6.148684425995775D-04
  0.000000000000000D+00   0.000000000000000D+00  -3.109799717393146D-02
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns    4 thru    6
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 -2.173933513752500D-01   0.000000000000000D+00   0.000000000000000D+00
  8.620324835512400D-01   0.000000000000000D+00   0.000000000000000D+00
  6.160360465203000D-01   0.000000000000000D+00   0.000000000000000D+00
 -1.134985373567730D+00   0.000000000000000D+00   0.000000000000000D+00
 -2.954762757515993D-02   0.000000000000000D+00   0.000000000000000D+00
  4.832985116285660D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   2.850631697258190D+00   2.600201224372080D+00
  0.000000000000000D+00  -4.468287885745900D+00  -5.127545367511940D+00
  0.000000000000000D+00  -3.051444495202544D-02  -1.254584600489570D-01
  0.000000000000000D+00   2.463348725006750D+00  -5.513569420366740D+00
```

Reduced Order Overlapping Controller (ROOC)

```
AC1          =

   Columns    1 thru    3
 -1.452480122503040D+00   1.756018673965760D+00   0.000000000000000D+00
 -3.945749143006660D+00  -9.453917704757270D-01   1.452359640575710D+00
  0.000000000000000D+00   6.124597937727510D-01  -5.434857699790040D+00
  0.000000000000000D+00   0.000000000000000D+00   4.815559257024294D-02

   Columns    4 thru    4
  0.000000000000000D+00
  0.000000000000000D+00
 -4.242551920910165D-02
 -2.820625832733400D+01

AC2          =

   Columns    1 thru    3
 -7.617234622765400D-01   1.145093819687130D+00   0.000000000000000D+00
 -1.162367431192810D+00  -7.020581071370500D-01   6.388667433456350D-03
  0.000000000000000D+00   6.036628526987111D-03  -3.580058235631030D-01
  0.000000000000000D+00   0.000000000000000D+00  -3.576542617906820D+00

   Columns    4 thru    4
  0.000000000000000D+00
  0.000000000000000D+00
  3.592468036601290D+00
 -3.607210693615690D-01

AC3          =

   Columns    1 thru    3
 -3.744937382207100D-01   4.440321278203877D-03   0.000000000000000D+00
  5.211323102396149D-05  -1.155080733245210D+00   1.999132732223541D-04
  0.000000000000000D+00  -3.431295200640864D-04  -1.181902941903580D+00
  0.000000000000000D+00   0.000000000000000D+00  -9.462820774112960D+00

   Columns    4 thru    4
  0.000000000000000D+00
  0.000000000000000D+00
  9.463190965482960D+00
 -1.181837776822930D+00
```

```
G        =

  Columns   1 thru   3
-1.290510683594130D+01  -9.790245433777640D+00   3.315510424881130D+01
-4.839641293091980D-01  -7.120113459013590D-02   1.003799507223110D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
-2.440443453543130D-01  -6.328110610962609D-02   4.822341386078310D-01
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns   4 thru   6
-1.424317237431190D+01   0.000000000000000D+00   0.000000000000000D+00
-1.192477216467900D+00  -2.977607304863700D-01   2.559817560055120D-01
 0.000000000000000D+00  -4.390837659469240D+01  -2.353468993149890D+01
-4.397766782423760D-01   5.217196223743558D-02  -2.275492837127810D-01
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns   7 thru   9
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 5.547968427926733D-03   1.003128858269810D-01   0.000000000000000D+00
-2.010080930011420D+01   1.676787980257970D+01   0.000000000000000D+00
 3.224638781333250D-01   2.002844248569479D-02   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00  -1.158801651275010D+00
 0.000000000000000D+00   0.000000000000000D+00  -1.132558684921652D-02

  Columns  10 thru  12
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 2.353739969690010D+01   5.202471966242030D+00   4.982855536787710D+00
 1.260671766822190D-01  -1.303457683121500D+00  -2.200843499772500D+00


K        =

  Columns   1 thru   3
-5.809550214825470D+00  -7.101654354472230D+00   0.000000000000000D+00
-3.864898476973230D+00   9.940574569681140D+00   0.000000000000000D+00
-1.290052111724840D+01   1.195747926759960D+01   0.000000000000000D+00
-1.237103993643130D+01   5.676304233116680D+01   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   7.043111422994730D-01
 0.000000000000000D+00   0.000000000000000D+00   1.457912948522780D+00
 0.000000000000000D+00   0.000000000000000D+00   2.706912314298850D-01
 0.000000000000000D+00   0.000000000000000D+00  -8.480459459031820D-01
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
```

```
  Columns   4 thru   6
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 5.484820600247960D-01   0.000000000000000D+00   0.000000000000000D+00
 5.827926334464460D-01   0.000000000000000D+00   0.000000000000000D+00
-1.422771360396130D+00   0.000000000000000D+00   0.000000000000000D+00
 6.928404246298890D-01   0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00  -8.701975130019260D+00  -8.173082448179380D+00
 0.000000000000000D+00  -3.828263831435710D+00  -3.516460132974570D+00
 0.000000000000000D+00  -5.717421164566010D-02   3.754464256402300D+00
 0.000000000000000D+00  -5.650144481426187D-02   5.333951023229490D+00
```

Reduced Order Decentralized Controller (RODC)

```
AC1        =

  Columns   1 thru   3
-1.111992785533652D+00   2.576050795703780D+00   0.000000000000000D+00
-2.576050795703780D+00  -1.111992785533652D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00  -5.673182641030859D+00
 0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns   4 thru   4
 0.000000000000000D+00
 0.000000000000000D+00
 0.000000000000000D+00
-2.736904086282898D+01


AC2        =

  Columns   1 thru   3
-7.396775299565394D-01   1.162367371964627D+00   0.000000000000000D+00
-1.162367371964627D+00  -7.396775299565394D-01   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00  -3.645515035669011D-01
 0.000000000000000D+00   0.000000000000000D+00  -3.586982807093651D+00

  Columns   4 thru   4
 0.000000000000000D+00
 0.000000000000000D+00
 3.586982807093651D+00
-3.645515035669011D-01
```

```
AC3      =

  Columns   1 thru   3
 -3.743785947237593D-01   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00  -1.154415465289202D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00  -1.181909869878743D+00
  0.000000000000000D+00   0.000000000000000D+00  -9.462953874635093D+00

  Columns   4 thru   4
  0.000000000000000D+00
  0.000000000000000D+00
  9.462953874635093D+00
 -1.181909869878743D+00


G        =

  Columns   1 thru   3
 -1.319467412327965D+01  -1.022142876249366D+01   3.250676231308169D+01
 -4.662204155665617D-01   5.780417735155261D-02   1.005911089655482D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns   4 thru   6
 -1.407132833599770D+01   0.000000000000000D+00   0.000000000000000D+00
 -1.218525838712585D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00  -4.390775396463485D+01  -2.353340353399004D+01
  0.000000000000000D+00   5.061740349871194D-02  -2.632993085269526D-01
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00

  Columns   7 thru   9
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
 -2.009971553343434D+01   1.677039343705295D+01   0.000000000000000D+00
  3.658758431640374D-01   6.227949922767444D-02   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00  -1.160482956685230D+00
  0.000000000000000D+00   0.000000000000000D+00  -1.135170654446138D-02

  Columns  10 thru  12
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  0.000000000000000D+00   0.000000000000000D+00   0.000000000000000D+00
  2.353688243823152D+01   5.202002102928775D+00   4.983149335076256D+00
  1.260500837257672D-01  -1.303727306918893D+00  -2.201094607597224D+00
```

K =

Columns 1 thru 3

-4.234145109844438D+00	-6.699670581919209D+00	0.000000000000000D+00
-3.167337972405387D+00	8.677837834388297D+00	0.000000000000000D+00
-1.267875145005293D+01	1.208112598855627D+01	0.000000000000000D+00
-1.344235581527970D+01	5.648377413710692D+01	0.000000000000000D+00
0.000000000000000D+00	0.000000000000000D+00	7.007341332126533D-01
0.000000000000000D+00	0.000000000000000D+00	1.463230999529574D+00
0.000000000000000D+00	0.000000000000000D+00	2.846859348618748D-01
0.000000000000000D+00	0.000000000000000D+00	-8.556351695664129D-01
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00

Columns 4 thru 6

0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
0.000000000000000D+00	0.000000000000000D+00	0.000000000000000D+00
5.686216681023230D-01	0.000000000000000D+00	0.000000000000000D+00
5.863207138422190D-01	0.000000000000000D+00	0.000000000000000D+00
-1.455093477442692D+00	0.000000000000000D+00	0.000000000000000D+00
6.826702771795415D-01	0.000000000000000D+00	0.000000000000000D+00
0.000000000000000D+00	-8.702224182658707D+00	-8.171422769313004D+00
0.000000000000000D+00	-3.827391309644286D+00	-3.515292916756586D+00
0.000000000000000D+00	-5.715999499660065D-02	3.754896128184332D+00
0.000000000000000D+00	-5.639468567612581D-02	5.333673574848407D+00

3859-65