State-Space Formulation for Structural Dynamics

by

Jose Luis Mendoza Zabala

B.S., Civil and Environmental Engineering (1994)
University of Texas at Austin

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Civil and Environmental Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1996

Author ....
Department of Civil and Environmental Engineering
April 24, 1996

Certified by ....
Jerome J. Connor
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by ....
Joseph M. Sussman
Chairman, Departmental Committee on Graduate Students

State-Space Formulation for Structural Dynamics

by

Jose Luis Mendoza Zabala

Submitted to the Department of Civil and Environmental Engineering
on April 24, 1996, in partial fulfillment of the
requirements for the degree of
Master of Science in Civil and Environmental Engineering

Abstract

In this thesis the state-space formulation is used for the analysis of structural dynamics. This formulation, being more general, does not present any restrictions on the characteristics of the damping. The problem is that for the state-space formulation the system matrix is nonsymmetric and therefore its eigenvalues and eigenvectors are complex which require further understanding of complex formulation. For that purpose the complex state-space formulation is expanded.

Also, we look at the computational efficiency of the formulation. The main computational burden in the analysis of large structures is the solution of the eigenproblem which for the nonsymmetric case is slightly altered. The eigenproblem solution for the nonsymmetric matrix is studied.

The state-space formulation is implemented in the analysis of active structural control. It is used for the study of various issues related to structural control. One issue is the discretization of the formulation for the application of digital control. Another is the effect of the time delay on the active control. Other topics are parameter sensitivity, optimization algorithms and nonlinear behavior. The simplicity of the state-space formulation provides a good framework for the investigation of all of these aspects.

Thesis Supervisor: Jerome J. Connor
Title: Professor of Civil and Environmental Engineering

Acknowledgments

I would like to express my gratitude to Prof. Jerome Connor. His lectures and his advice have been a source of inspiration for the last two years. I hope that some of that inspiration remains with me and I am able to project it in my future undertakings so as to make him proud.

Thank you, also, to all of those who have been with me during all this time. Specially Miguel, Jose and Jose Luis. Also Anteneh, Sang and the rest of my friends at MIT.

Finally, thank you to all my large family. Thank you to my parents. They grew me up and they deserve all the credit for my achievements. Thank you Michelle, for your continuous support during these two years.

Jose Luis

Contents

List of Figures

Chapter 1

Introduction

In many areas of engineering and applied science, differential equations are used for the analysis of the behavior of dynamical systems. Each discipline tends to formulate these equations in a way that best suits the problems of interest within that discipline and satisfies certain objectives such as ease of computation and transparency of the formulation. For the case of structural dynamics in civil engineering, the equations of motion are usually presented in the form of second order differential equations.

Solving these second order equations is not difficult. However, one is not taking advantage of the solution algorithms developed for first order equations. These algorithms are applicable to a broader range of equations and also are more easily implemented.

Converting the governing differential equations to a set of first order equations is the standard approach for most disciplines. These equations are referred to as the state space formulation. This study is concerned with computational aspects of the state-space formulation for structural systems. Of particular interest are how one can deal with arbitrary damping and the nonsymmetric eigenproblem and how computational time increases with the size of the system. Also, it focuses on the implementation of this formulation in the structural control problem.

1.1 The second order equation

In the field of structural engineering, the dynamic behavior of a structure is described by the following equation of motion:

$$\mathbf{m}\ddot{\mathbf{u}} + \mathbf{c}\dot{\mathbf{u}} + \mathbf{k}\mathbf{u} = \mathbf{f} \tag{1.1}$$

where $\mathbf{m}$ is the mass matrix, $\mathbf{c}$ is the damping matrix, $\mathbf{k}$ is the stiffness matrix and the vector $\mathbf{u}$ contains the displacements for the system and $\mathbf{f}$ defines the external excitation. This equation represents a system of second order linear differential equations.

This equation has been studied extensively and the solution is well known. There are different ways to obtain it, and understanding the equation and solution is simple. Simplicity is the main quality of this equation and approach; it is very physical and the solution is easy to interpret.

The solution for the case of a one degree of freedom linear system can be obtained in close form when there is no external excitation, that is, for free vibration. It involves an oscillatory factor and another term which, in all structural applications, decays exponentially. This solution is easy to derive and has the following form:

$$u(t) = e^{-\xi\omega t}\left(\frac{v_0 + \xi\omega u_0}{\omega_d}\sin\omega_d t + u_0\cos\omega_d t\right) \tag{1.2}$$

The solution for a linear system subjected to an exciting force $f(t)$ involves the use of Duhamel's integral. Conceptually, it sums the exponentially decaying responses due to a series of impulses determined by discretizing $f(t)$,

$$u(t) = \frac{1}{m\omega_d}\int_0^t e^{-\xi\omega(t-\tau)} f(\tau)\sin\omega_d(t-\tau)d\tau \tag{1.3}$$

Establishing the analytical solution is possible only for simple force-time relationships. Generally, it is necessary to use numerical integration for complex load-time histories.

On the other hand, it is always possible to obtain directly a numerical solution to the second order differential equations. There are various sophisticated schemes such as Newmark, Wilson or Constant Acceleration method.

This equation has also been solved by changing from the time domain to the frequency domain. This solution method for the single degree of freedom case decomposes the excitation into periodic excitations and obtains the solution to each of this periodic loads. The solution for the displacement due to a single periodic load is very simple and can be obtained algebraically. Then the solution for the general loading is obtained by combining the individual responses for each frequency. This method has been effectively applied in the design of structures. Since each input periodic load can be characterized by an amplification factor, the effect on the structure of a particular periodic loading can be readily established.

For the case of multiple degrees of freedom, one can use a coordinate transformation to convert the coupled system of equations to a set of uncoupled equations for the generalized coordinates. This makes the solution of the problem much easier for different reasons. Firstly, because we can solve each second order equation individually and then assemble the response. Secondly, because in structural engineering, we deal with a large number of degrees of freedom of which only a few are really significant. The use of only a few of the generalized coordinates allow us to obtain a good idea of the response of the system with much less computational cost.

The way to perform the transformation is by using the eigenvectors of the system matrix. This means that we have to solve for the eigenvalues of a second order eigenvalue problem. In the case of second order equations one way to solve this complicated problem is by ignoring the damping term when finding the eigenvalues and eigenvectors. The damping matrix is then assumed to be proportional to the stiffness and damping matrices so that when the transformation is performed the equations uncouple completely.

1.2 State-space formulation

In the state-space formulation the unknown variables are those quantities that are necessary to completely describe the state of the system at any time. For a structural system, the state variables are the displacements and velocities. Instead of working with n equations of motion for the n degrees of freedom, we break the second order equations into $2n$ first order equations. The equation for the case of multiple degree of freedom is,

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{BF} \tag{1.4}$$

where the vector $\mathbf{X}$ contains the displacements and velocities, $\mathbf{A}$ contains the system parameters, $\mathbf{F}$ contains the external excitation and $\mathbf{B}$ is called the locator matrix.

Even though we are changing the form of the differential equation, obviously the solution has to be the same. However, since the form is different, the solution process changes. Also, conceptually, this formulation is different, more general although less physical than the second order equation.

In the case of the state space formulation, since the equation is of first order, the eigenvalue problem does not have to be truncated and all the characteristics of the system are embedded in the system matrix A. The damping matrix can be arbitrarily chosen and does not have to be proportional to mass or stiffness. This flexibility in representing the damping is convenient for damping based control devices.

For the multiple degree of freedom case, the eigenproblem has to be solved with the computational burden that the system matrix is $2n$ by $2n$. Then, one changes to the state-space generalized coordinates. Although these coordinates are complex and have less physical meaning than the generalized coordinates for the proportional damping case, it is still necessary to change to generalized coordinates in order to reduce the computational effort without significantly affecting the accuracy.

The state space formulation system matrix is not symmetric and that is why the eigenvalues and eigenvectors occur in complex conjugate pairs. Since the solution is real, the different complex solutions to the state-space equation need to be combined on a certain way to produce the "real" solution.

1.3 Advantages of the state-space formulation

Next we discuss some issues which suggest the need for the state-space formulation as an alternative to the second order equation.

1.3.1 A more general formulation; arbitrary damping

One advantage of the state space formulation is that it allows one to specify an arbitrary damping matrix. The corresponding disadvantage is that an arbitrary choice leads to a nonsymmetric eigenproblem. This is an additional computational cost that has to be considered.

It has been argued that since the size of the system matrix is generally very large for structural systems, the computational efficiency of the solution becomes an issue and shifting to the state space formulation as opposed to proportional damping becomes too expensive in computer time and not worth the advantages it might offer. However, the validity of that judgement is subject to change with the passage of time. Computers are evolving faster than they can be commercialized. By the time that a new computer is adopted, a new faster one is already being introduced. Nowadays, one can perform computations in a few seconds that would have taken minutes ten years ago or hours twenty years ago. The change of factors like this one make us consider again mathematical procedures that we had avoided earlier for different reasons.

Still, we are forced to minimize the number of operations and to optimize the use of the computational power. Later, the methods developed for computation are studied for the case of the state space formulation.

If there are complications that may arise, for this formulation there are also some reasons that push us to investigate and study it carefully. The main reason mentioned earlier was the ability to define an arbitrary damping matrix, although there are other issues to take into consideration.

1.3.2 A more general formulation; conceptual level

Conceptually, this formulation provides a more general although less physical perspective of the dynamics problem. For the most basic dynamical system, the state is just proportional to the rate of change of this state. Something as important as the state of the structural system is defined uniquely only by its position and by its velocity, and contains all the information about the distribution of the energy within the system. The strain, representing the potential energy in the system and the momentum for the actual energy in the system. The characteristics of the first order dynamical system are embedded on a single system matrix which in itself contains all the information about the particular system.

1.3.3 A more general formulation; interdisciplinary

The energy approach to problems in dynamics is always the most general. Being more general, this approach converges easier to other areas of engineering. For instance, it does not require particular interpretations from structural engineering related to the damping of the system. One more advantage is derived from the generality of the state-space formulation. That is, the model that we work with is shared among more disciplines as opposed to a very particular formulation which would isolate the work performed within the discipline. By working at a common mathematical level with other disciplines of engineering, coherence between disciplines is achieved.

For instance, it is the formulation that will allow us to implement in the best way control theory to structures which in the future may be important for applications. We will concentrate on this characteristic of the state-space formulation later. It will be used for the study of the structural control problem.

1.3.4 A more general formulation; numerical implementation

There are also practical aspects in which the state space formulation can be very interesting. One of these is for the numerical analysis in the time domain for which we need to use numerical integration. In this formulation we are using a first order differential equation which is much easier to integrate in time using simple methods such as Euler or Runge-Kutta.

1.3.5 Advantages to structural engineering

In summary, we separate three levels at which the formulation could be interesting. One is the academic, other is the research and, finally, the commercial analysis level.

At the academic level, it is extremely convenient for students since it is conceptually interesting and applies to a much larger kind of systems than just structural systems. The workings of dynamical systems should be mastered by any engineer from any discipline even if he or she is going to be working at a very particular level in an engineering system. All systems have a first order rate of change and modifications deriving from this concept one can get to particular formulations that may be of interest to any civil engineering students.

At the level of research, it is certainly useful when working in areas that specifically implement this approach. In structural control research, it present many advantages and possibly in areas related to energy absorption and dissipation. Definitely, it is interesting to work on it and investigate it with greater detail before discarding it. Then, we can explain why we do not use it. Maybe, if it is not efficient or interesting today we can refer back to it in the future when time might have made it a more convenient approach.

At an industrial level, it is more questionable if it is the best formulation. The main objective at the level of some practical applications is functionality and effectivity. If this is the case, we should compare this formulation to other that are used currently. The main obstacle are the computations that need to be carried out to obtain a few eignvectors and eigenvalues from the system matrix.

1.4 State-space formulation implementation for control

One of the qualities that we mentioned of the state-space formulation is that it is interdisciplinary. One of those disciplines which implements state-space formulation is control. Control applies to any dynamical process in which men actuate and in all natural processes in which men do not. A controller modifies the characteristics of the system in order to make it perform better when trying to achieve some objectives. These objectives are usually the optimization of some variable, may it be the most effective use of water resources or the least deformations in a structure.

It is noticeable how even though traditionally all the areas mentioned have faced analytical problems that where solved by methods and notations particular to each, in the area of control all these disciplines seem to converge into one formulation very similar since the control problem although applying to all of them is essentially independent of the area which it models.

It is the field of electrical engineering the one that has investigated the most the problem of control with the state-space formulation. Also, it has been successfully applied to areas such as mechanical engineering for robotics and other applications, in airports for traffic control, in aerospace for aircraft and spacecraft control or in operations research for control of processes. One of those processes can be considered the construction process of large projects. Also, the theory is used for transportation engineering to control the traffic flows. In the management of water resources we also try to control flows of water to optimize its use. The theory is also implemented in financial models for the control of the economy. It is even used in some scientific areas traditionally less analytical like the Social Sciences.

Extensive literature has been produced using this formulation applied to control in many areas of engineering and some exposure to it or familiarity with it would be very beneficial to any field of engineering that deals with system dynamics, which are basically all the fields and structural engineering in particular. This theory, control theory, has not been investigated much within the area of structural engineering though. Traditionally a building was designed with some given characteristics of stiffness, mass and damping and it was left to take as an input the random external excitation of stochastic excitations such as earthquakes. The structural system were designed to be passive and not active. Consequently, structural engineering is probably one of the few engineering disciplines that has little or no presence of control theory within its curricula. However, the problem of system control applies to all dynamical systems so it should be a requirement in the education of an engineer and it would add much value to the understanding of engineering in general.

In the case of an actively controlled structure the state-space formulation has an additional term. This is the term that represents the control input. It has a similar form to the external excitation but the value of the input is determined through optimization instead of being stochastic. In the following equation the input is assumed to be in the form of accelerations,

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B_g}\mathbf{a_g} + \mathbf{B_f}\mathbf{F} \tag{1.5}$$

If the control input F is zero then we are left with the traditional dynamic equation of motion for the passive structure. From this we deduce that the control problem is analytically a more general problem. With the addition of the control action a broad range of new issues are raised that need to be well understood before this theory can be used in actual structural control.

The first problem raised with the introduction of control to the model is the selection of the mentioned control. To find the control force we will have to solve an optimization problem. There will be different versions to the optimization depending on what basis we use for the choice. We can be more or less strict with the deformation or be able to use more or less energy for control.

The are more issues that require our attention on the structural control problem such as the effect of control on stability or the potential problems that arise when the applied control is not working exactly at with the timing we wanted it to work. There are still other questions that are addressed in this thesis.

1.5 Control application to structures

It is important to incorporate the control theory to the analytical tools of the structural engineering. This is most evident when we consider the application of actual control to real civil engineering structures.

That possibility has had considerable impact in structural engineering in recent years. The interesting and challenging idea of controlling a structural system in order to reduce the effects of external excitations may turn into a reality in the beginning of the next century. This excitations are produced by nature in a rather random way in the form of earthquakes, strong winds or even water pressures for the case of off-shore structures. The effects of these phenomena are well known to be destructive. The cost is very large and in many cases the losses are human lives. In the last decades many have been the places that have suffered earthquakes. China, Japan, Mexico, Guatemala and The United States to name just a few examples.

The effects of this earthquakes are especially destructive for larger structures or buildings which usually contain more people and more value. The technological advances in construction technology and the growth of overpopulated metropolitan areas all over the world will probably increase very much the amount of buildings and therefore increase the amount of human lives and value exposed to the destructive effects of earthquakes. This means that efforts should be directed to the protection of these structures against these agents and control offers a very good solution to the problem. The idea of active control is to apply controlling forces that will tend to counteract the effect of external excitation on structures.

The control can be of two main classes. These are passive and active control. In the case of passive control, the controlling device is set within the system to reduce deformations in the structure for any external excitation. The action of the device is not dependant on the excitation. Active control, will work on the structures depending on the excitation. If the actuators act on the building based on the deformation or the actual position of the system it is closed loop control. If the action is taken based only on the excitation that is coming to the structure then it is called open loop.

Some theoretical aspects of active control are going to be the topic of other chapters so we will wait to expand the concepts. The problem of how to apply active control is a more complicated one and it is still a question how to efficiently generate the forces to control large structures. The theory of active control has already been implemented succesfully in some small to mid size buildings in Japan as well as in models in laboratories. Japan is one of the countries pioneering the implementation of active control. A country which has been struck many times by earthquakes. In addition, the spacial constrains on its cities force construction industry to grow the building structures vertically, increasing their exposure to earthquakes.

1.6 General outline

In later chapters, we will offer a detailed expansion of some of the operations we need to perform on the state space formulation in order to apply it to structural analysis. Also, we will look at the solution of the particular eigenproblem posed by the state space formulation when producing a nonsymmetric system matrix. We will use the characteristics of this formulation to obtain information regarding the behavior of the system when subjected to the control action. In particular information related to stability. We will also use the formulation to present some simulation schemes of actual control to observe the influence of some real control problems. Some of these are the discrete model, optimization and time delay.

Chapter 2

State Space Formulation

This chapter presents the state-space formulation for the dynamic analysis of structures as opposed to the second order differential equation. This formulation is developed and verified for the structural analysis problem. This will help us understand the possibilities of this formulation and its advantages in the analysis of the many aspects of the dynamic behavior of structures.

First, we will outline the steps that uncouple the multiple degree of freedom system of equations using the proportional damping. Then, we uncouple the equations for the state-space formulation. This formulation, through the use of generalized coordinates, leads to complex eigenvalues as well as eigenvectors. By applying another change of coordinates, we produce a real system of equations. The solution of the eigenproblem is expanded for the single degree of freedom case. We present the general expansion of the real solution for the state-space equation.

2.1 Proportional damping

There are many procedures to set up the equations of motion for a dynamical system. It is possible to use the equation of Newton, or D'alambert's principle. We can apply the Lagrangian equations of motion or even Hamilton's equation. All these equations are based on the same basic principles but the way to solve them is not exactly the same. Which equations to use depends on the characteristics of the particular problem.

The equation of motion for a single degree of freedom structural system undergoing free vibration has the form:

$$m\ddot{u} + c\dot{u} + ku = 0 \tag{2.1}$$

To solve this problem we assume a solution of the form,

$$u = ae^{\lambda t} \tag{2.2}$$

Substituting these back we get the arbitrary constants a from the initial conditions. As shown, for a single equation that is easily solved in the time domain with the use of the exponential notation. Euler's equation is then used to arrive to a real solution even though the frequency of the system may be complex.

The problem is more complicated for the case of multiple degrees of freedom. In this case we need to solve an eigenproblem in order to obtain the eigenvalues and the eigenvectors that uncouple the equations. For the general damped case, we have a quadratic eigenproblem which leads to complex eigenvectors. However, to avoid dealing with complex operations, this problem has traditionally been solved for the undamped case which gives real eigenvectors. Then, damping is introduced in the particular equations in various ways, some of which are very effective. That is the case for the proportional damping.

For a typical structural system, the number of nodes is relatively large and the number of degrees of freedom may be a few thousands. The matrix describing the characteristics of such a system is therefore very large. Given the constraints on the computer capabilities, we are forced to reduce the amount of computation to a minimum without a considerable loss of information. This is done by transforming the system of equations to the generalized coordinates. We decompose the displacements of the structure into orthogonal modes, and concentrate on the behavior of the dominant ones.

This transformation is performed for different reasons. On one hand, some modes are excited more than others by a given excitation. Also the magnitude of displacement for only a few modes accounts for a very large percentage of the displacement of a structure. Finally, the less dominant modes are usually harder to find and since they are less accurate therefore do not really add useful information.

To solve the eigenproblem using proportional damping, we diagonalize the problem for the mass and stiffness matrices ignoring the damping. This eigenproblem comes from,

$$\mathbf{M}\ddot{\mathbf{X}} + \mathbf{K}\mathbf{X} = \mathbf{0} \tag{2.3}$$

which is the same as,

$$(\mathbf{M}^{-1}\mathbf{K} - \mathbf{I}\lambda)\mathbf{X} = \mathbf{0} \tag{2.4}$$

where λ is the square of the frequency and the eigenvalues of the eigenproblem. The capital letters represent vectors or matrices. By ignoring the damping matrix we are left with the general eigenproblem. This problem is simple for two reasons. It is only an n by n matrix and it is symmetric thus producing real eigenvectors.

With the solution of this eigenproblem we are able to diagonalize the mass and stiffness matrices. Then the damping matrix is selected proportional to the stiffness and mass matrices, which are diagonalized by the transformation. This way the damping matrix is also diagonalized when we perform the change of coordinates. One specifies the damping for two of the modes only, the rest of the modal dampings are dependant on these two. The selection of the modal damping presents a different problem than the selection of damping for an individual member.

2.2 State-space formulation

In the state-space formulation, the unknowns define the state of the system. For a system with n degrees of freedom, these unknowns are the velocity and displacement for each degree of freedom, a total of $2n$ variables . From the second order differential equations of motion, we can obtain n equations involving velocity and position. The other n equation come from a very simple relation,

$$\dot{u} = \frac{du}{dt} \tag{2.5}$$

When we put together these $2n$ equations, we have a system of first order differential equations. The equation, including a term for the control action and another for the external excitation is the following:

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{B_f F} + \mathbf{B_g a_g} \tag{2.6}$$

As explained before, the vector $\mathbf{X}$ is a vector of size $2n$ containing the displacement and velocities at the nodes of the structure. It completely defines the state of the system at any time.

$$\mathbf{X} = \begin{bmatrix} \mathbf{U} \\ \dot{\mathbf{U}} \end{bmatrix} \tag{2.7}$$

The vector $\dot{\mathbf{X}}$ represents the first order change of the state of the system. The matrix $\mathbf{A}$ contains the characteristics of the system,

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M^{-1}K} & -\mathbf{M^{-1}C} \end{bmatrix} \tag{2.8}$$

The matrix $\mathbf{F}$ contains the control forces, and $\mathbf{B_f}$ is a locator matrix for the control forces of the following form:

$$\mathbf{B_f} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M^{-1}E_f} \end{bmatrix} \tag{2.9}$$

The matrix $\mathbf{B_g}$ is another locator for the acceleration of the excitation.

$$\mathbf{B_f} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{E} \end{bmatrix} \tag{2.10}$$

The $\mathbf{E}$ and $\mathbf{E_f}$ matrices contain just ones on the places where the accelerations or the forces respectively are located and simply zeros where there are no accelerations or forces. Finally, the vector $\mathbf{a_g}$ contains the input accelerations.

2.2.1 Eigenproblem for the nonsymmetric matrix

By reducing the system from n second order equations to $2n$ first order equations, we are producing a system matrix which is not symmetric. Eventually, this will present a nonsymmetric eigenproblem which is a drawback of this formulation.

The solution for this eigenproblem can be obtained using different iterative methods. It is necessary to study these methods. This will give us an idea of how much more computational time is required to solve the nonsymmetric eigenproblem for the state space formulation as compared to the method of proportional damping.

Our next step will be to expand the solution to the state-space equation and to the nonsymmetric eigenproblem to gain more understanding of the formulation.

The eigenproblem is found by plugging the solution to the equation back into the original equation. For the case of free vibration, one has the following equation:

$$\dot{\mathbf{X}} = \mathbf{AX} \tag{2.11}$$

For the first order matrix differential equation we can assume the following solution:

$$\mathbf{X} = \sum \mathbf{C_i} e^{\lambda_i t} \tag{2.12}$$

In this case the vector $\mathbf{C_i}$ represents the vector with the arbitrary constants to be determined from the initial conditions. And, λ_i shall be determined by plugging the solution back into the state-space equation for the homogeneous case.

$$\lambda_i \mathbf{X} = \mathbf{AX} \tag{2.13}$$

then,

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{X} = \mathbf{0} \tag{2.14}$$

This is the standard eigenvalue problem with the feature that the **A** matrix is not symmetric.

The solution of the eigenvalue problem posed by the state space equation should give us the $2n$ complex eigenvalues and eigenvectors of the **A** matrix. This set of $2n$ eigenvectors is composed of n eigenvectors and their n conjugates, that is, the complex eigenvalues and their corresponding eigenvectors occur in conjugate pairs.

By diagonalizing the **A** matrix, one uncouples the equations and solve for the individual variables. First, one must obtain the vectors that diagonalize the matrix.

We know that,

$$\mathbf{A}\mathbf{V_j} = \lambda_j \mathbf{V_j} \tag{2.15}$$

$$\mathbf{W_j^T}\mathbf{A} = \lambda_j^* \mathbf{W_j} \tag{2.16}$$

Every $\mathbf{V_j}$ satisfies

$$(\mathbf{A} - \lambda_j \mathbf{I})\mathbf{V_j} = \mathbf{0} \tag{2.17}$$

and every $\mathbf{W_j}$ satisfies

$$\mathbf{W_j^T}(\lambda_j^* \mathbf{I} - \mathbf{A}) = \mathbf{0} \tag{2.18}$$

It is simple to prove that the eigenvectors of **A**, and the left eigenvectors of **A** have the same eigenvalues. Premultiplying the two eigenproblems by $\mathbf{W_j^T}$ and $\mathbf{V_j^T}$ respectively we get,

$$\mathbf{W_j^T}\mathbf{A}\mathbf{V_j} = \lambda_j \mathbf{W_j^T}\mathbf{V_j} \tag{2.19}$$

$$\mathbf{V_j^T}\mathbf{A^T}\mathbf{W_j} = \lambda_j^* \mathbf{V_j^T}\mathbf{W_j} \tag{2.20}$$

Since,

$$(\mathbf{V_j^T}\mathbf{A^T}\mathbf{W_j})^{\mathbf{T}} = \mathbf{W_j^T}\mathbf{A}\mathbf{V_j} \tag{2.21}$$

$$\mathbf{W_j^T}\mathbf{A}\mathbf{V_j} = \lambda_j^* \mathbf{W_j^T}\mathbf{V_j} \tag{2.22}$$

Follows that,

$$\lambda_j^* = \lambda_j \tag{2.23}$$

Now we show that the left eigenvectors of $\mathbf{A}$ are also the eigenvectors of $\mathbf{A^T}$.

$$\mathbf{A^T W_i} = \lambda_i \mathbf{W_i} \tag{2.24}$$

And transposing,

$$\mathbf{W_i^T A} = \lambda_i \mathbf{W_i^T} \tag{2.25}$$

and then,

$$\mathbf{W_i^T}[\lambda_i \mathbf{I} - \mathbf{A}] = \mathbf{0} \tag{2.26}$$

left eigenvectors of $\mathbf{A}$ are also the eigenvectors of $\mathbf{A^T}$. Since,

$$\mathbf{A^T W_i} = \lambda_i \mathbf{W_i} \tag{2.27}$$

And transposing,

$$\mathbf{W_i^T A} = \lambda_i \mathbf{W_i^T} \tag{2.28}$$

and then,

$$\mathbf{W_i^T}[\lambda_i \mathbf{I} - \mathbf{A}] = \mathbf{0} \tag{2.29}$$

Once we proved that the eigenvalues for $\mathbf{A}$ and $\mathbf{A^T}$ are the same, we must prove another important point. In order to uncouple the equations we need $2n$ orthogonal eigenvectors. This means that the set of eigenvectors of $\mathbf{A}$ and $\mathbf{A^T}$ must be orthogonal to each other. To prove orthogonality premultiply again by $\mathbf{W_i^T}$ and by $\mathbf{V_j^T}$ to get,

$$\mathbf{W_i^T A V_j} = \lambda_j \mathbf{W_i^T V_j} \tag{2.30}$$

$$\mathbf{V_j^T A^T W_i} = \lambda_i \mathbf{V_j^T W_i} \tag{2.31}$$

Transposing the second equation we get,

$$\mathbf{W_i^T A V_j} = \lambda_i \mathbf{W_i^T V_j} \tag{2.32}$$

and substracting the two equation leads to,

$$0 = (\lambda_j - \lambda_i)\mathbf{W_i^T V_j} \tag{2.33}$$

It follows that,

$$\mathbf{W_i^T V_j} = 0 \quad for \quad j \ not \ equal \ to \ i \tag{2.34}$$

There is a comment to make regarding the eigenvectors $\mathbf{W_i}$ and $\mathbf{V_i}$. If we define the $\mathbf{V}$ matrix such that,

$$\mathbf{V} = [\mathbf{V_1 V_2 V_3 ... V_n}] \tag{2.35}$$

the matrix $\mathbf{V^{-1}}$ has the vectors $\mathbf{W_i^T}$, normalized, for rows. This follows directly from,

$$\mathbf{W_i^T V_j} = 0 \quad for \quad j \ not \ equal \ to \ i \tag{2.36}$$

$$\mathbf{W_i^T V_j} = 1 \quad for \quad j = i \quad after \quad normalizing \tag{2.37}$$

The right and left eigenvectors of $\mathbf{A}$ diagonalize the matrix, but the magnitude of the diagonal entries depends on the constants used when solving for the eigenvectors. This is because the eigenvectors have specified direction but not magnitude. If we obtain $\mathbf{V^{-1}}$ from the left eigenvectors we must go through normalization. This is done by first finding the product,

$$\mathbf{W_i^T V_i} = c_{ii} \tag{2.38}$$

The normalized vectors are,

$$\mathbf{W_{i,n}^T V_{i,n}} = \frac{\mathbf{W_i^T} V_i}{c_{ii}} \tag{2.39}$$

where the subscript n means 'normalized'.

This equation, however, does not specify how c_{ii} is divided between $\mathbf{W_i}$ and $\mathbf{V_i}$. Although the vectors will be normalized anyway, it is convenient to apply some judgement later on, at this point. We should set a standard of normalization which may save complications. One standard is to set the first entry of $\mathbf{V_i}$ equal to one.

2.3 Change of coordinates

2.3.1 Transformation to generalized coodinates

We have an orthogonal and normalized basis. We now change the coordinates in the state space equation from natural coordinates to generalized coordinates

As we stated before, the eigenvectors and eigenvalues from the **A** matrix occur in conjugate pairs. Therefore, if we represent the conjugate of any vector **V** by $\tilde{\mathbf{V}}$ we can express any vector **X** as a combination of its eigenvectors as follows,

$$\mathbf{X} = \mathbf{V}\mathbf{q} + \tilde{\mathbf{V}}\tilde{\mathbf{q}} \tag{2.40}$$

The factors that multiply the eigenvectors are the generalized coordinates which occur also in conjugate pairs and are represented by **q** and $\tilde{\mathbf{q}}$. In order to perform this modal decomposition, we combine the conjugate terms in global matrices represented with a *g* subscript,

$$\mathbf{V_g} = \begin{bmatrix} \mathbf{V} & \tilde{\mathbf{V}} \end{bmatrix} \tag{2.41}$$

$$\mathbf{q_g} = \begin{bmatrix} \mathbf{q} \\ \tilde{\mathbf{q}} \end{bmatrix} \tag{2.42}$$

$$\mathbf{W_g^T} = \begin{bmatrix} \mathbf{W^T} \\ \tilde{\mathbf{W}}^\mathbf{T} \end{bmatrix} \tag{2.43}$$

We have the tools to transform the state space equation from natural coordinates to generalized coordinates. Starting with

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{BF} \tag{2.44}$$

substituting for $\mathbf{X} = \mathbf{V_g q_g}$,

$$\mathbf{V_g \dot{q}_g} = \mathbf{A V_g q_g} + \mathbf{BF} \tag{2.45}$$

and expanding

$$\begin{bmatrix} \mathbf{V} & \tilde{\mathbf{V}} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{V} & \tilde{\mathbf{V}} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \tilde{\mathbf{q}} \end{bmatrix} + \mathbf{BF} \tag{2.46}$$

leads to

$$\mathbf{V}\dot{\mathbf{q}} + \tilde{\mathbf{V}}\dot{\tilde{\mathbf{q}}} = \mathbf{AVq} + \mathbf{A}\tilde{\mathbf{V}}\tilde{\mathbf{q}} + \mathbf{BF} \tag{2.47}$$

Also,

$$\mathbf{AV_g} = \mathbf{V_g \Lambda_g} \tag{2.48}$$

The matrix $\mathbf{\Lambda_g}$ can be decomposed into its conjugate eigenvalues as we did before with the conjugate eigenvectors,

$$\mathbf{\Lambda_g} = \begin{bmatrix} \mathbf{\Lambda} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Lambda}} \end{bmatrix} \tag{2.49}$$

The matrices $\mathbf{\Lambda}$ and $\tilde{\mathbf{\Lambda}}$ are diagonal matrices containing the eigenvalues and conjugate eigenvalues respectively.

Then, the next step is to use the matrix $\mathbf{V_g^{-1}}$ to obtain the state space equation in the uncoupled form. By premultiplying all the state space equation by,

$$\mathbf{V_g^{-1}} = \mathbf{W_g^T} \tag{2.50}$$

we finally obtain the uncoupled equation.

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Lambda}} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \tilde{\mathbf{q}} \end{bmatrix} + \mathbf{W_g^T BF} \tag{2.51}$$

or the two following equations,

$$\dot{\mathbf{q}} = \mathbf{\Lambda q} + \mathbf{W^T BF} \quad (n \;\; equations) \tag{2.52}$$

$$\dot{\tilde{\mathbf{q}}} = \tilde{\mathbf{\Lambda}}\tilde{\mathbf{q}} + \tilde{\mathbf{W}}^{\mathbf{T}}\mathbf{BF} \quad (n \;\; equations) \tag{2.53}$$

At this point there are a set of $2n$ uncoupled equations on the $2n$ state-space modal coordinates. One has to face the problem that this equations involve complex factors. Still, we know that the solution to the problem, the positions and the velocities in **X**, has to be real (there is not such a thing as an imaginary position in space...yet). For this we will perform a second transformation by decomposing the complex terms into imaginary and real parts, and use these as variables. Then, we just solve the equations.

2.3.2 Transformation to real modal coordinates q_R and q_I

The transformation that we will apply is, for a one degree of freedom system,

$$q = q_R + iq_I \tag{2.54}$$

$$\tilde{q} = q_R - iq_I \tag{2.55}$$

which means,

$$q_g = \begin{bmatrix} q \\ \tilde{q} \end{bmatrix} = \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} \begin{bmatrix} q_R \\ q_I \end{bmatrix} \tag{2.56}$$

Defining **Q** and **T** as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q_R} \\ \mathbf{q_I} \end{bmatrix} \tag{2.57}$$

$$\mathbf{T} = \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} \tag{2.58}$$

the transformation is expressed by

$$\mathbf{q_g} = \mathbf{TQ} \tag{2.59}$$

We will also use the inverse of the transformation matrix **T** so that it does not appear on the transformed equation. This matrix has this form

$$\mathbf{T^{-1}} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{-i}{2} & \frac{i}{2} \end{bmatrix} \tag{2.60}$$

So when applying the transformation to the modal state space equations we get,

$$\mathbf{T\dot{Q}} = \mathbf{\Lambda_g TQ} + \mathbf{W_g^T BF} \tag{2.61}$$

Next we premultiply by $\mathbf{T^{-1}}$ which together with $\mathbf{T}$ transforms the $\mathbf{\Lambda_g}$ matrix. The resulting matrix which we will call $\mathbf{A}^*$ will be a skew symmetric matrix. Defining,

$$\mathbf{A}^* = \begin{bmatrix} \mathbf{\Lambda_R} & -\mathbf{\Lambda_I} \\ \mathbf{\Lambda_I} & \mathbf{\Lambda_R} \end{bmatrix} \tag{2.62}$$

where Λ_R and Λ_I are diagonal matrices containing the real and imaginary parts of the eigenvalues respectively.

Also define,

$$\mathbf{B}^* = \mathbf{T^{-1} W_g^T B} \tag{2.63}$$

Finally, the real modal state space equation once transformed is,

$$\mathbf{\dot{Q}} = \mathbf{A^* Q} + \mathbf{B^* F} \tag{2.64}$$

This transformation is easily expanded to the case of multiple degrees of freedom where the $\mathbf{T}$ would be,

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & i\mathbf{I} \\ \mathbf{I} & -i\mathbf{I} \end{bmatrix} \tag{2.65}$$

where $\mathbf{I}$ is an n by n identity matrix.

2.3.3 Overall transformation

.

Since overall we are performing a change of coordinates, one can find what matrix performs both transformations in one step. Since,

$$\mathbf{X} = \mathbf{V_g q_g} \tag{2.66}$$

and,

$$\mathbf{q_g} = \mathbf{TQ} \tag{2.67}$$

it follows that

$$\mathbf{X} = \mathbf{V_g TQ} \tag{2.68}$$

We will call the overall transformation $\mathbf{Z}$, such that,

$$\mathbf{X} = \mathbf{ZQ} \tag{2.69}$$

where

$$\mathbf{Z} = \mathbf{V_g T} \tag{2.70}$$

We can carry out the multiplication to obtain one matrix. Defining for each vector,

$$\mathbf{V_j} = \mathbf{V_{R,j}} + i\mathbf{V_{I,j}} \tag{2.71}$$

then,

$$\mathbf{Z} = \begin{bmatrix} \mathbf{V} & \tilde{\mathbf{V}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & i\mathbf{I} \\ \mathbf{I} & -i\mathbf{I} \end{bmatrix} \tag{2.72}$$

that results in,

$$\mathbf{Z} = 2 \begin{bmatrix} \mathbf{V_R} & -\mathbf{V_I} \end{bmatrix} \tag{2.73}$$

where,

$$\mathbf{V_R} = \begin{bmatrix} \mathbf{V_{R,1}} & \mathbf{V_{R,2}} & \dots & \mathbf{V_{R,n}} \end{bmatrix} \tag{2.74}$$

and similarly for $\mathbf{V_I}$.

To sum up, one goes from

$$\mathbf{X} = \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix} \tag{2.75}$$

to,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q_R} \\ \mathbf{q_I} \end{bmatrix} \tag{2.76}$$

using the transformation,

$$\mathbf{Z} = 2\begin{bmatrix} \mathbf{V_R} & -\mathbf{V_I} \end{bmatrix} \tag{2.77}$$

After we solve the transformed equation for $\mathbf{Q}$, one obtains the solution in natural coordinates using,

$$\mathbf{X} = \mathbf{V_R q_R} + \mathbf{V_I q_I} \tag{2.78}$$

2.4 The one degree of freedom case

Although the state space formulation is just an alternate way of representing the traditional formulation, its useful to go through some of the steps in the solution for the one degree of freedom case. This process will demonstrate that both formulations lead to the same solution as expected. Also, it gives a better understanding of the problem by following some of the unknowns through the operations. Finally, it shall be possible to compare the similarities between both procedures.

For the one degree of freedom case, where ω is the natural frequency, ω_d is the damped frequency and ξ is the damping, the system matrix in state-space is the following:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\xi\omega \end{bmatrix} \tag{2.79}$$

which can easily be derived from the second order differential equation of motion. The eigenvalues of this matrix are,

$$\lambda_{1,2} = -\xi\omega \pm i\omega\sqrt{1-\xi^2} = \lambda_R \pm i\lambda_I \tag{2.80}$$

The solution has the form,

$$x = ce^{\lambda t} \tag{2.81}$$

Differentiating with respect to time,

$$\dot{x} = \lambda ce^{\lambda t} \tag{2.82}$$

and assembling the state vector leads to

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 1 \\ \lambda \end{bmatrix} ce^{\lambda t} \tag{2.83}$$

Then, the eigenvectors have the form

$$\mathbf{V_1} = \begin{bmatrix} 1 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \lambda \end{bmatrix} \quad \textit{and} \quad \mathbf{V_2} = \begin{bmatrix} 1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \tilde{\lambda} \end{bmatrix} \tag{2.84}$$

The left eigenvectors from $\mathbf{A^T}$ are,

$$\mathbf{A^T} = \begin{bmatrix} 0 & -\omega^2 \\ 1 & -2\xi\omega \end{bmatrix} \tag{2.85}$$

Clearly the eigenvalues are the same, and the eigenvectors can also be found to be proportional to,

$$\mathbf{W_1} = \begin{bmatrix} -\omega^2 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} -\omega^2 \\ \lambda \end{bmatrix} \quad \textit{and} \quad \mathbf{W_2} = \begin{bmatrix} -\omega^2 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -\omega^2 \\ \tilde{\lambda} \end{bmatrix} \tag{2.86}$$

Before we use the eigenvectors, they must be normalized. This is done, as we explained earlier, by multiplying the vectors by themselves to get a scalar and then dividing the vectors by that scalar. We already proved that they were orthogonal.

Let us verify that the eigenvectors that we obtained before are orthogonal.

$$\begin{bmatrix} -\omega^2 & \lambda_1 \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{\lambda} \end{bmatrix} = -\omega^2 + \lambda\tilde{\lambda} = 0 \tag{2.87}$$

Now, we normalize,

$$\begin{bmatrix} -\omega^2 & \lambda \end{bmatrix} \begin{bmatrix} 1 \\ \lambda \end{bmatrix} = -2\omega^2(1 - \xi^2 + 2i\xi\sqrt{1-\xi^2}) = -2\omega^2 s \tag{2.88}$$

That is the normalizing factor to use for the one degree of freedom case. One has to decide how to split the factor among the two eigenvectors. This will not make a difference to the results but it is convenient to make a good choice to obtain simpler arithmetic. The choice here will be to include a one in the first entry of the eigenvectors from **A**. This way we obtain a normalized **V** that will be represented with a subscript of n. This matrix of eigenvectors with a first entry of one is known as the Vandermonde matrix. For this case,

$$\mathbf{V_n} = \begin{bmatrix} 1 & 1 \\ \lambda & \tilde{\lambda} \end{bmatrix} \tag{2.89}$$

Since we started with this vectors, we will divide $\mathbf{W^T}$ by the scalar we obtained to get the normalized vectors for this case. For the undamped case,

$$\mathbf{W_n} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -\frac{i}{\omega} & \frac{i}{\omega} \end{bmatrix} \tag{2.90}$$

Next we multiply $\mathbf{W_n^T A V_n}$ to get the complex eigenvalues for the undamped case which are,

$$\mathbf{W_n^T A V_n} = \begin{bmatrix} i\omega & 0 \\ 0 & -i\omega \end{bmatrix} \tag{2.91}$$

Now we can perform the second transformation using the T matrix which gives the A^* matrix for the one degree of freedom case,

$$\mathbf{A}^* = \begin{bmatrix} -\xi\omega & -\omega_d \\ \omega_d & -\xi\omega \end{bmatrix} \tag{2.92}$$

The overall transformation for this case is done by,

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 \\ -\xi\omega & \omega_d \end{bmatrix} \tag{2.93}$$

for the single degree of freedom case.

2.4.1 From diagonal state space to second order equation.

Now that we have transformed the state-space formulation we can relate the transformed q_R and q_I to the original x and $\dot{x}$ for the one degree of freedom and homogeneous case. The transformed state-space equation is,

$$\dot{q_R} = \lambda_R q_R - \lambda_I q_I \tag{2.94}$$

$$\dot{q_I} = \lambda_I q_R + \lambda_R q_I \tag{2.95}$$

Now we solve for q_I and differentiate.

$$q_I = -\frac{\dot{q_R}}{\lambda_I} + \frac{\lambda_R}{\lambda_I} q_R \tag{2.96}$$

$$\dot{q_I} = -\frac{\ddot{q_R}}{\lambda_I} + \frac{\lambda_R}{\lambda_I} \dot{q_R} \tag{2.97}$$

and substituting in Eq.(2.92) yields

$$\ddot{q_R} - 2\lambda_R \dot{q_R} + (\lambda_R^2 + \lambda_I^2) q_R = 0 \tag{2.98}$$

This equation reduces to

$$\ddot{q_R} - 2\xi\omega \dot{q_R} + \omega^2 q_R = 0 \tag{2.99}$$

Solving for q_R is equivalent to solving for x.

2.5 Real expansion of the state-space solution.

Next we prove that the expansion of the complex solution of x gives a real solution. If we try to expand the solution vector $\mathbf{X}$ into its complex eigenvectors we will get the following expression:

$$\mathbf{X} = \mathbf{V}\mathbf{C}e^{\mathbf{\Lambda}t} + \tilde{\mathbf{V}}\tilde{\mathbf{C}}e^{\tilde{\mathbf{\Lambda}}t} \tag{2.100}$$

where

$$\mathbf{q} = \left[c_i e^{\lambda_i t}\right] \tag{2.101}$$

If we use the transformation to convert the matrices in the equation to real coordinates with the $\mathbf{T}$ matrix, we can say,

$$\mathbf{V} = \mathbf{V_R} + i\mathbf{V_I} \tag{2.102}$$

$$\tilde{\mathbf{V}} = \mathbf{V_R} - i\mathbf{V_I} \tag{2.103}$$

$$\mathbf{C} = \mathbf{C_R} + i\mathbf{C_I} \tag{2.104}$$

$$\tilde{\mathbf{C}} = \mathbf{C_R} - i\mathbf{C_I} \tag{2.105}$$

Also, one splits the exponentials using the following expression:

$$e^{\lambda t} = e^{\lambda_R t} e^{i\lambda_I t} \tag{2.106}$$

We can do the this expansion for each degree of freedom so that by using the mentioned transformations together with,

$$e^{i\lambda_I t} + e^{-i\lambda_I t} = 2\cos\lambda_I t \tag{2.107}$$

we can rewrite the expression for the single degree of freedom case.

$$\mathbf{x} = e^{\lambda_R t}\{\cos\lambda_I t(C_R\mathbf{V_R} - C_I\mathbf{V_I}) + \sin\lambda_I t(-C_R\mathbf{V_I} - C_I\mathbf{V_R})\} \tag{2.108}$$

which is the complete real solution for both the displacement and velocity.

When we substitute the for the normalized eigenvectors for the one degree of freedom case we get,

$$\mathbf{V_R} = \begin{bmatrix} 1 \\ \lambda_R \end{bmatrix} V^* \tag{2.109}$$

$$\mathbf{V_I} = \begin{bmatrix} 0 \\ \lambda_I \end{bmatrix} V^* \tag{2.110}$$

where the V^* is just a constant used for normalization. What looks like this:

$$\mathbf{X} = e^{\lambda_R t}\{\cos\lambda_I t \begin{bmatrix} C_R \\ C_R\lambda_R - C_I\lambda_I \end{bmatrix} V^* + \sin\lambda_I t \begin{bmatrix} -C_I \\ -C_R\lambda_I - C_I\lambda_R \end{bmatrix} V^*\} \tag{2.111}$$

Although this was shown for a one degree of freedom system it can be easily extended for the multidegree of freedom case by summing over all the modes. The general form is

$$\mathbf{X} = \sum C_j e^{\lambda_j t}\mathbf{V_j} + \tilde{C_j} e^{\tilde{\lambda_j} t}\tilde{\mathbf{V_j}} \tag{2.112}$$

For the one degree of freedom we can find for $t = 0$ the initial conditions. If we assume,

$$\mathbf{X(0)} = \begin{bmatrix} u(0) \\ \dot{u}(0) \end{bmatrix} \tag{2.113}$$

this leads to,

$$\begin{aligned} V^*C_R &= u(0) && (2.114) \\ V^*C_R\lambda_R - V^*C_I\lambda_I &= \dot{u}(0) && (2.115) \end{aligned}$$

For $\dot{u}(0)$ equal to zero,

$$u = e^{\lambda_R t}\cos\lambda_I t u(0) \tag{2.116}$$

$$C_R V^* = u(0) \tag{2.117}$$

When $u(0) = 0$,

$$C_R = 0 \tag{2.118}$$

$$C_I = \frac{-\dot{u}(0)}{V^*\lambda_I} \tag{2.119}$$

For the case of no damping we have

$$\lambda_I = \omega \tag{2.120}$$

$$V^* = 1 \tag{2.121}$$

2.5.1 Relationship between constants in the solution

We will try to relate the generalized coordinates to the constants C_R and C_I. The real solution for the displacement and velocity was expressed earlier in terms of the generalized coordinates $\mathbf{q}$ and $\tilde{\mathbf{q}}$ in the following form:

$$\mathbf{X} = \mathbf{Vq} + \tilde{\mathbf{V}}\tilde{\mathbf{q}} \tag{2.122}$$

Also,

$$\mathbf{X} = \mathbf{VC}e^{\mathbf{\Lambda}t} + \tilde{\mathbf{V}}\tilde{\mathbf{C}}\mathbf{e}^{\tilde{\mathbf{\Lambda}}\mathbf{t}} \tag{2.123}$$

Therefore with this equation and the expansion we just used, we should be able to get a relationship between the coordinates q_R and q_I and the C_R and C_I for one degree of freedom.

We have

$$q = (C_R + iC_I)e^{(\lambda_R + i\lambda_I)t} \tag{2.124}$$

$$\tilde{q} = (C_R - iC_I)e^{(\lambda_R - i\lambda_I)t} \tag{2.125}$$

Now,

$$q + \tilde{q} = 2q_R \tag{2.126}$$

$$-iq + i\tilde{q} = 2q_I \tag{2.127}$$

Then,

$$Ce^{\lambda t} + \tilde{C}e^{\tilde{\lambda} t} = 2q_R \tag{2.128}$$

$$-iCe^{\lambda t} + i\tilde{C}e^{\tilde{\lambda} t} = 2q_I \tag{2.129}$$

now expanding the way we did before,

$$q_R \;=\; e^{\lambda_R t}\{C_R(e^{i\lambda_I t} + e^{-i\lambda_I t}) + iC_I(e^{i\lambda_I t} - e^{-i\lambda_I t})\} \tag{2.130}$$

$$q_R \;=\; e^{\lambda_R t}\{C_R(\cos\lambda_I t) - C_I(\sin\lambda_I t)\} \tag{2.131}$$

By doing the same derivation with the equation about q_I we get similarly,

$$q_I = e^{\lambda_R t}\{C_I(\cos\lambda_I t) + C_R(\sin\lambda_I t)\} \tag{2.132}$$

so the equations can be put in matrix form:

$$\begin{bmatrix} q_R \\ q_I \end{bmatrix} = e^{\lambda_R t} \begin{bmatrix} C_R & -C_I \\ C_I & C_R \end{bmatrix} \begin{bmatrix} \cos\lambda_I t \\ \sin\lambda_I t \end{bmatrix} \tag{2.133}$$

This equation give us a good idea about what is the relation between one system of coordinates and the other. Of course time had to be included in the relation since q is time dependant and C simply reflect the initial conditions. Also, we observe that at $t = 0$ the variables are just equal to each other.

Chapter 3

Overview of eigenproblem for the nonsymmetric matrix

There is an important aspect of the formulation that we have to consider. The state space formulation has to be computationally feasible or at least not very inefficient in comparison to another formulations or procedures if we intend to implement it.

The analysis in structural engineering has to deal generally with a larger number of equations and therefore with larger matrices than other fields such as electrical engineering. It is necessary then to change the system from natural coordinates to the generalized coordinates, which allows us to reduce the number of variables. This is possible because in this coordinate system some coordinates are significantly larger than others. These correspond to the principal modes of the system. In order to change the coordinates we are forced to solve for the eigenvalues of the system matrix. This is, in general, computationally the most expensive step in the analysis, especially when we are dealing with hundreds or even thousands of degrees of freedom.

There are different methods to solve for the eigenvalues and eigenvectors of a matrix. This topic has been exhaustively studied in the field of civil engineering for the reason mentioned above. The efficiency of the solver for the eigenproblem can decide whether the solution of a problem is reasonable or if the amount of computation makes it prohibitive.

We have to establish how much more expensive the transformation to modal coordinates is for the state space formulation in comparison to the classical formulation. This is difficult because there are many ways to solve the eigenproblem, and each particular problem or system matrix requires a method adjusted to its characteristics.

Next, we will overview the alternative solution methods for the eigenproblem when the matrix is symmetric. This is the most typical eigenproblem and the one that has been studied the most in structural engineering. Then, in order to compare the algorithms, we present an iterative solution method for the non symmetric eigenvalue. It proves that there is also a simple iterative solution for this problem even though the eigenvectors are complex.

3.1 Eigenproblem solution with proportional damping

The equation for the damped free-vibration problem for the multi degree of freedom system has the form,

$$\mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{0} \tag{3.1}$$

The solution to this equation was given before:

$$\mathbf{U} = e^{\lambda t}\mathbf{q} \tag{3.2}$$

Substituting this solution into the original equation results in

$$(\lambda^2\mathbf{M} + \lambda\mathbf{C} + K)\mathbf{q} = \mathbf{0} \tag{3.3}$$

This equation defines the quadratic eigenvalue problem which has a difficult solution. One way to solve it is to set the determinant of the matrix to zero and solving for the roots of the polynomial. This procedure is very difficult to implement for larger systems.

The quadratic eigenvalue problem has been avoided by ignoring the term representing the damping, and solving the standard eigenvalue problem.

$$(\mathbf{M}^{-1}\mathbf{K} + \lambda^2\mathbf{I})\mathbf{q} \tag{3.4}$$

The eigenvalues turn out to be pure imaginary as expected since we considered no damping. The problem that we deal with in this case is a standard eigenvalue problem of size n by n. One advantage of this problem is that the size of the matrix is just the number of degrees of freedom while the number of variables is actually $2n$. However, it is also very important that the mass and the stiffness matrix are symmetric and then the eigenproblem is symmetric too.

The eigenvalues, which are pure imaginary, are conveniently turned into sin and cos in the expanded solution. This allows for a simpler interpretation of it for analysts less familiar with the exponential formulation which, on the other hand, if understood can provide an easier and more representative formulation.

By using proportional damping we have simplified the formulation but we have lost generality and possibly valuable information regarding the variables that we left out of the eigenproblem, namely the velocities.

An exhaustive research of the many different solution techniques for the eigenvalue problem would fill volumes with information that go beyond the scope of this thesis. Some of these become very intricate and with the increasing computational speed of computers and the increase in their availability these methods may become less important.

The methods used generally for the solution of the symmetric eigenproblem are of three different types: transformation , vector iteration, polynomial methods.

3.1.1 Transformation methods

The idea of the transformation methods is to apply successive rotations to the system matrix in order to diagonalize it without actually changing the eigenvalues of the system. This method diagonalizes the system matrix so that we obtain all the eigenvalues at the same time. As we said before in the case of reduction of the order of the size of a problem we do not need all the eigenvalues but only a small percentage of them. The methods are appropriate for rather small matrices. It is not convenient for sparse matrices since the transformation do not use this advantage in the solution, in fact some rotation that tend to diagonalize the matrix may place a non zero entry where there was already a zero.

3.1.2 Vector iteration methods

The second type of solution algorithms are knows as the iteration methods. The iteration methods are specially suited for large eigenproblem where only a few eigenvalues are being studied since they are found one by one and we can even specify what range of eigenvalues we are looking for.

The vector iteration methods allow for the succesive calculation of the interesting eigenvalues starting by the largest, smallest or for a range of values. This method will be explained in more detail since it will be the method most appropriate for our kind of problems in which we aim only for some important and most significant eigenvalues. Also it is applicable for the nonsymmetric eigenvalue problem. This will be shown later using the basic procedures that immediately follow.

The basic principle of the vector iteration methods also called the power methods is the following. Suppose we define the eigenvalues of a matrix such that,

$$|\lambda_1| < |\lambda_2| < |\lambda_3| < ... < |\lambda_n| \tag{3.5}$$

Also since the eigenvectors of an n by n matrix form an orthogonal basis for the N-dimensional space, we know that an arbitrary vector $\mathbf{u}$ on that space can be represented as a linear combination of these eigenvectors such that,

$$\mathbf{u} = c_1\mathbf{q_1} + c_2\mathbf{q_2} + ... + c_N\mathbf{q_N} \tag{3.6}$$

if we premultiply this arbitrary vector by the system matrix on both sides we get,

$$\mathbf{Au} = c_1\mathbf{Aq_1} + c_2\mathbf{Aq_2} + ... + c_N\mathbf{Aq_N} \tag{3.7}$$

$$\mathbf{Au} = c_1\lambda_1\mathbf{q_1} + c_2\lambda_2\mathbf{q_2} + \ldots + c_N\lambda_N\mathbf{q_N} \tag{3.8}$$

if we repeat the multiplication iteratively k times we get,

$$\mathbf{A^k u} = c_1\lambda_1^k\mathbf{q_1} + c_2\lambda_2^k\mathbf{q_2} + \ldots + c_N\lambda_N^k\mathbf{q_N} \tag{3.9}$$

after many iterations we realize that most of the terms will vanish except for the term

$$\mathbf{u_k} = \mathbf{A^k u} = c_N\lambda_N^k\mathbf{q_N} \tag{3.10}$$

so that it is the value that we will converge to after enough iterations.

We can get information about the convergence characteristics if we put the equation in the following form,

$$\mathbf{A^k u} = \lambda_N^k \left[c_1 \left(\frac{\lambda_1}{\lambda_N}\right)^k \mathbf{q_1} + c_2 \left(\frac{\lambda_2}{\lambda_N}\right)^k \mathbf{q_2} + \ldots + c_{N-1} \left(\frac{\lambda_{N-1}}{\lambda_N}\right)^k \mathbf{q_{N-1}} + c_N\mathbf{q_N} \right] \tag{3.11}$$

The value of an eigenvalue can be then estimated from the expression,

$$\lambda_N = \rho_{k+1} = \frac{\mathbf{u_{k+1}^T u_{k+1}}}{\mathbf{u_{k+1}^T u_k}} \tag{3.12}$$

In order to keep the entries of the eigenvectors with small size we should perform a simple normalization of the vectors every iteration, or few iterations, using a simple expression such as,

$$\mathbf{u_{k+1}} = \frac{\bar{\mathbf{u}}_{\mathbf{k+1}}}{\left(\bar{\mathbf{u}}_{\mathbf{k+1}}^{\mathbf{T}}\bar{\mathbf{u}}_{\mathbf{k+1}}\right)}^{\frac{1}{2}} \tag{3.13}$$

Here is a list of the steps for the typical vector iteration algorithm:,

$$\bar{\mathbf{u}}_{\mathbf{k+1}} = \mathbf{Au_k} \tag{3.14}$$

$$\rho_{k+1} = \frac{\mathbf{u_{k+1}^T u_{k+1}}}{\mathbf{u_{k+1}^T u_k}} \tag{3.15}$$

$$\mathbf{u}_{\mathbf{k+1}} = \frac{\bar{\mathbf{u}}_{\mathbf{k+1}}}{\left(\bar{\mathbf{u}}^{\mathbf{T}}_{\mathbf{k+1}}\bar{\mathbf{u}}_{\mathbf{k+1}}\right)}^{\frac{1}{2}} \tag{3.16}$$

There is a method that presents a slight variation from the algorithm presented and it is the inverse vector iteration method. It presents some characteristics that make it more interesting for us. It will converge to the lowest eigenvalue that is to the dominant frequency in the case of structural analysis. The general eigenvalue problem is,

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{q} = \frac{1}{\lambda}\mathbf{q} \tag{3.17}$$

which is the same as,

$$\mathbf{D}\mathbf{q} = \gamma\mathbf{q} \tag{3.18}$$

where the eigenvalues are

$$\gamma = \frac{1}{\omega^2} \tag{3.19}$$

The algorithm presented above will converge to the lowest eigenvalue and to the corresponding eigenvector. In order to get other eigenvectors and eigenvalues we must make the next trial vector orthogonal to the vector that we already found, so that no matter how many times we multiply it by the system matrix it never converges to it. Because it is orthogonal, the coefficient that multiplies the eigenvector already found is zero and the algorithm converges to the second eigenvalue. This procedure is called the Gram-Schmidt orthogonalization. We apply it by selecting the new trial vector such that

$$\tilde{\mathbf{u}} = \mathbf{u} - c_1\mathbf{q}_1 \tag{3.20}$$

Since for the symmetric case

$$c_i = \frac{\mathbf{q}_{\mathbf{i}}^{\mathbf{T}}\mathbf{u}}{\mathbf{q}_{\mathbf{i}}^{\mathbf{T}}\mathbf{q}_{\mathbf{i}}} \tag{3.21}$$

then the trial vector should be

$$\tilde{\mathbf{u}} = \mathbf{u} - \frac{\mathbf{q}_{\mathbf{1}}^{\mathbf{T}}\mathbf{M}\mathbf{u}}{\mathbf{q}_{\mathbf{1}}^{\mathbf{T}}\mathbf{M}\mathbf{q}_1}\mathbf{q}_1 \tag{3.22}$$

We can rewrite

$$\tilde{\mathbf{u}} = \left(\mathbf{I} - \frac{\mathbf{q_1 q_1^T M}}{\mathbf{q_1^T M q_1}}\mathbf{q_1}\right)\mathbf{u} \tag{3.23}$$

We can perform this same task in the form of a sweeping matrix that sweeps the old system matrix **D** from the mode that we already removed. This is performed by multiplying the system matrix by

$$\mathbf{S_1} = \mathbf{I} - \frac{\mathbf{q_1 q_1^T M}}{\mathbf{q_1^T M q_1}}\mathbf{q_1} \tag{3.24}$$

To find an eigenvalue which is neither the largest nor the smallest we have to apply a shift to aim for any range of eigenvalues. This procedure is briefly explained later.

3.1.3 Polynomial Iteration Techniques

The other type of method of solution for the eigenproblem apply polynomial iteration techniques. This method is based on the condition that the determinant of the matrix must vanish when we substract the eigenvalues from the diagonal entries. This condition is expressed by,

$$p(\lambda_{(i)}) = 0 \tag{3.25}$$

When we expand the determinant we get a polynomial. If we can solve explicitly for the eigenvalues embedded in the polynomial which is for very small systems, then the method is very simple. However for the systems that we are dealing with, we are forced to use numerical methods to find the roots of the polynomial or the eigenvalues of the matrix. Iterative methods allow us to find these roots. With these methods we can also find as many eigenvalues as we need.

3.1.4 Solution Tools

There are important tools that may improve the solution algorithms for the eigenproblem. Before we look at more complex methods for solving the eigenproblem which implement these tools, we introduce them. They speed up and insure the convergence process.

-Shifts. If we perform a shift to the eigenvalues, the new eigenproblem will have the eigenvalues of the old problem plus the amount of the shift and the same eigenvectors. If we apply the shift,

$$\lambda = \mu + \delta \tag{3.26}$$

The new eigenproblem will be,

$$(\mathbf{K} - \mu\mathbf{M})\mathbf{q} = \delta\mathbf{M}\mathbf{q} \tag{3.27}$$

so that,

$$\hat{\mathbf{D}}\mathbf{q} = \hat{\gamma}\mathbf{q} \tag{3.28}$$

The change to the new problem can be very convenient for different reasons. It can help us avoid the problem of a rigid body mode for which the eigenvalue is zero by shifting its value. It can also help us when combined with the iteration methods.

As we saw after k iterations of an arbitrary vector some items in the expanded vector vanished. When we divided and multiplied all the expression by the largest eigenvalue the equation became:

$$\mathbf{A^k u} = \lambda_N^k \left[c_1 \left(\frac{\lambda_1}{\lambda_N}\right)^k \mathbf{q_1} + c_2 \left(\frac{\lambda_2}{\lambda_N}\right)^k \mathbf{q_2} + \ldots + c_{N-1} \left(\frac{\lambda_{N-1}}{\lambda_N}\right)^k \mathbf{q_{N-1}} + c_N \mathbf{q_N} \right] \tag{3.29}$$

Therefore after enough iterations all the terms would vanish except for the one containing the largest eigenvalue. By applying shifts to the axis of the eigenvalues we can force the largest eigenvalue to be within any range we want. Therefore we will converge to any eigenvalue we want if it becomes the largest when the shift is applied.

-Sturms Sequence check. When we apply an arbitrary shift μ to the eigenvalues in our system matrix we get a new system matrix $\mathbf{A} - \mathbf{I}\lambda$. In the Sturms sequence check, which is not presented here, when we perform the $\mathbf{LDL^T}$ factorization of the matrix the number of negative elements in $\mathbf{D}$ indicates us how many eigenvalues there are for the matrix less than μ. This is a very useful check for some iterative methods in which we search one eigenvalue at a time without knowing some times if we might have skipped one. The check will tell us if when we performed a shift to speed up the process we left an eigenvalue behind.

3.2 Combined methods

We presented the basic methods and properties that we use for the solution of the eigenproblem. However, there are cases in which we combine the use of different methods to make the most efficient use of them. Two of these methods are subspace iteration and the determinant search method.

3.2.1 Subspace Iteration

The subspace iteration uses transformation and iterative techniques. It can also implement some of the tools to expedite the solution process.

For the subspace iteration method we are trying to solve for the p lowest eigenvectors and eigenvalues. The subspace iteration method is ideal when we are dealing with large systems in which we are interested in only a few really significant eigenvectors. In order to find these eigenvectors we first use a method similar to vector iteration but in which we iterate p eigenvectors at the same time in the following way,

$$\mathbf{K\bar{X}_{k+1} = MX_k} \tag{3.30}$$

for the case of proportional damping solution, and

$$\mathbf{\bar{X}_{k+1} = AX_k} \tag{3.31}$$

for the solution of the state space formulation which is already in the form of the standard eigenvalue problem.

With this iteration the vectors will converge to the eigenvectors. However, this does not insure that this vectors will approach different eigenvectors. In other words, we should make sure that this eigenvectors span the subspace in which we will work. This is achieved by solving an eigenproblem in the subspace. In order to solve this problem in the subspace we need to find the projection of the system matrix or matrices onto the reduced space. For the case of general eigenvalue problem we solve,

$$\mathbf{K_{k+1}} = \mathbf{\bar{X}^T_{k+1} K \bar{X}_{k+1}} \tag{3.32}$$

Once we have the reduced matrix we are able to solve for the new improved set of p eigenvectors from this smaller eigenproblem,

$$\mathbf{K_{k+1} Q_{k+1}} = \mathbf{M_{k+1} Q_{k+1} \Lambda_{k+1}} \tag{3.33}$$

The vectors that we get from here are ready for a new iteration which will apply again the two steps. Namely, forcing the vectors to approach the eigenvectors and then solving the eigenproblem in the subspace so that they tend to span the subspace. In summary, this is the algorithm,

$$\mathbf{K \bar{X}_{k+1}} = \mathbf{M X_k} \tag{3.34}$$

$$\mathbf{K_{k+1}} = \mathbf{\bar{X}^T_{k+1} K \bar{X}_{k+1}} \tag{3.35}$$

$$\mathbf{M_{k+1}} = \mathbf{\bar{X}^T_{k+1} M \bar{X}_{k+1}} \tag{3.36}$$

$$\mathbf{K_{k+1} Q_{k+1}} = \mathbf{M_{k+1} Q_{k+1} \Lambda_{k+1}} \tag{3.37}$$

At this point, it is necessary to introduce an equation to normalize the vectors.

$$\mathbf{X_{k+1}} = \mathbf{\bar{X}_{k+1} Q_{k+1}} \tag{3.38}$$

There are still some issues to address regarding how to improve the method to make it more efficient. There is an eigenvalue problem to be solved several times. In this eigenproblem we are searching for all eigenvectors and the size of the problem is small. For this reasons the method to use could be a transformation method such as

Jacobi.

Also, the starting set of eigenvectors have to be chosen carefully because the number of iterations required to get to the solution depends very much on these. For the procedure and characteristics of this method we can see that it is well suited for our problem which deals with large matrices finding a few eigenvalues in an efficient manner.

3.2.2 Determinant search

This method is based on the iteration of the characteristic polynomial. It has some variations which make it more efficient. Like the polynomial iteration it is best for problems with small bandwidth and it can be used to search only for a few eigenvalues. In the procedure, we use polynomial iteration techniques to search for the zeros. We also use deflation to find roots other than the first one. To make sure that we are finding the right roots we apply the Sturm sequence check with the $\mathbf{LDL}^{\mathbf{T}}$ factorization.

In addition, we also use shifts to make it even more efficient. We first use polynomial iterations to find an appropriate shift that will approximate as close as possible the next eigenvalue on its axis. Once we are close enough to the eigenvalue, vector iteration can be very effective finding the eigenvector exactly.

3.3 Eigenproblem for the state-space equation

.

From the brief introduction made above to eigensolvers we can see what methods are available. From the kind of problems that we work with we can deduce which method are best for us and what kind of operations are needed. The problem is that although this method have been used very effectively solving the general and standard eigenvalue problem they have not been designed specifically to deal with the eigenproblem that the state-space formulation presents.

The solution to the large dynamical problem in the state space formulation goes through the solution of the eigenproblem for the non symmetric matrix. The problem is standard, which is better, although it still requires the inversion of the mass matrix to create the system matrix with the operational costs that it involves and also the risks of ill conditioning. The eigenvectors and eigenvalues are complex. This loss of symmetry and added complexity will not allow us to use some methods, which however were not suited for our problem anyway, and will force some variation on other methods. The solution is still possible and not much more complicated than it was for the symmetric case. One application of the vector iteration method is presented by Humar and is developed in this section.

The names we will use for the variables and its real and imaginary parts will be the same that we used earlier such that,

$$\lambda = \lambda_R + i\lambda_I \tag{3.39}$$

$$\mathbf{V_j} = \mathbf{V_{R,j}} + i\mathbf{V_{I,j}} \tag{3.40}$$

If we start with some initial trial vector which could be real,

$$\mathbf{V_0} = \mathbf{V_{R,0}} + i\mathbf{V_{I,0}} \tag{3.41}$$

and perform the first iteration multiplying by the system matrix such that,

$$\mathbf{D}\left(\mathbf{V_{R,k}} + i\mathbf{V_{I,k}}\right) = (\lambda_R + i\lambda_I)(\mathbf{V_{R,k}} + \mathbf{iV_{I,k}}) \tag{3.42}$$

then we can equate real and imaginary parts which gives,

$$\mathbf{DV_{R,k}} = \lambda_R\mathbf{V_{R,k}} - \lambda_I\mathbf{V_{I,k}} \tag{3.43}$$

$$\mathbf{DV_{R,k}} = \mathbf{V_{R,k+1}} \tag{3.44}$$

$$\mathbf{DV_{I,k}} = \lambda_R\mathbf{V_{I,k}} - \lambda_I\mathbf{V_{R,k}} \tag{3.45}$$

$$\mathbf{DV_{R,k}} = \mathbf{V_{I,k+1}} \tag{3.46}$$

what we do is to eliminate the imaginary part $\mathbf{V_I}$ and leave $\mathbf{V_{R,k}}$ as a function of $\mathbf{V_{R,k+1}}$ and $\mathbf{V_{R,k+2}}$ so that we can perform iterations on the real parts. If we perform the next iteration these should be,

$$\mathbf{V_{R,k+2}} - 2\lambda_R \mathbf{V_{R,k+1}} + (\lambda_R^2 + \lambda_I^2)\mathbf{V_{R,k}} \tag{3.47}$$

$$\mathbf{V_{R,k+3}} - 2\lambda_R \mathbf{V_{R,k+2}} + (\lambda_R^2 + \lambda_I^2)\mathbf{V_{R,k+1}} \tag{3.48}$$

Using this two equations we can get an expression for the real part of the eigenvalue, which is given next,

$$\lambda_R = \frac{1}{2}\frac{V_{R,k}V_{R,k+3} - V_{R,k+1}V_{R,k+2}}{V_{R,k}V_{R,k+2} - V_{R,k+1}V_{R,k+1}} \tag{3.49}$$

Also the real part of the eigenvector will converge. Then we use the following expressions to get the imaginary parts of both eigenvectors and eigenvalues,

$$\lambda_R^2 + \lambda_I^2 = \frac{V_{R,k+1}V_{R,k+3} - V_{R,k+2}V_{R,k+2}}{V_{R,k}V_{R,k+2} - V_{R,k+1}V_{R,k+1}} \tag{3.50}$$

$$\mathbf{V_{I,k}} = \frac{\lambda_R}{\lambda_I}\mathbf{V_{R,k}} - \frac{1}{\lambda_I}\mathbf{V_{R,k+1}} \tag{3.51}$$

Then we must select a new trial vector for iteration on the second eigenvector. The selection of the new trial vector has to have some constrains that insure orthogonality with respect to the eigenvector that we already found. This is done with an analogous procedure to the Gram-Schmidt orthogonalization.

3.4 Conclusions

This brief introduction help us consider to what extent it is more complicated to include the damping matrix in the state space formulation as opposed to the use of proportional damping. However, the detailed analysis of the computational cost of each method is not developed here but is left as a further step on the analysis of the formulation. Still, we should look at this problem with the new perspective provided by the rapidly evolving computing machines.

There are some considerations that can be made to guide further investigation of the algorithms.

For instance there may be improvements to the methods derived from the particular form of the system matrix in the state-space formulation. It has one quarter of the matrix filled with zeroes and another taken by an identity matrix. That kind of sparsity requires special attention for it may provide advantages.

Also, we must consider wether the size of the system matrix, $2n$, really affects the computational time considerably. We only search for p eigenvalues. For a system twice the size we should need twice the number of modes to describe its shape with the same accuracy. However, if for the state-space matrix we search for $2p$ eigenvectors, they still occur in conjugate pairs, such that we only really need to search for p eigenvectors and the other p vectors we automatically know. With the tools described above it would be possible to avoid the computational cost that those eigenvectors would require.

Furthermore, from the second chapter we deduced that the lower entries of the eigenvector were just the upper entries times the eigenvalue that corresponds to that eigenvector. That is another peculiarity of the eigenproblem for this particular non-symmetric system matrix that should be considered to improve further the solution algorithms.

Chapter 4

Formulation For Control

The state-space formulation can be employed in the structural control problem. The formulation represents an appropriate analytical tool for both the continuous and discrete time control. Since the application of control to structures is at an initial stage and the cost of the implementation is very high, an exhaustive study of the possible problems that may be encountered is required. All deviations from the original model have to be studied thoroughly by means of simulation and sensitivity analysis. In the following sections some of those alterations to the initial model are examined.

Continuous control is discussed first. Then, we model the problem of discrete control, which although conceptually analogous to the continuous time control, it is different and also more realistic. We also incorporate the effect of time delay in the formulation.

Finally, we investigate other important issues such as parameter sensitivity, that need to be taken into account.

4.1 Continuous-time control

The state space formulation presented earlier is specialized for control by incorporating a control force, **F**, in the equilibrium equation. This leads to

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{B_f F} + \mathbf{B_g a_g} \tag{4.1}$$

Where $\mathbf{X}$ is of size $2n$ and contains the displacement and velocity of the nodes for the structure,

$$\mathbf{X} = \begin{bmatrix} \mathbf{U} \\ \dot{\mathbf{U}} \end{bmatrix} \tag{4.2}$$

$\dot{\mathbf{X}}$ represents the first order change of the state of the system, $\mathbf{A}$ contains the characteristics of the system,

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M^{-1}K} & -\mathbf{M^{-1}C} \end{bmatrix} \tag{4.3}$$

$\mathbf{F}$ contains the control forces, and $\mathbf{B_f}$ is a locator matrix for the control forces of the form,

$$\mathbf{B_f} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M^{-1}E_f} \end{bmatrix} \tag{4.4}$$

and $\mathbf{B_g}$ is another locator matrix for the acceleration of the excitation

$$\mathbf{B_f} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{E} \end{bmatrix} \tag{4.5}$$

The $\mathbf{E}$ and $\mathbf{E_f}$ matrices contain just ones on the places where the accelerations or the forces respectively are located and simply zeros where there are no accelerations or forces. Finally the vector $\mathbf{a_g}$ contains the input accelerations. In what follows, the nature of the control force $\mathbf{F}$ is discussed.

Continuous negative linear feedback control

The control forces in the type of systems that we will investigate will be chosen depending on the state of the system at a given time. The measuring devices will read the state of the system and based on it, an action will be taken, that is, a control force will be applied. The selection of the appropriate force will be made with an algorithm which optimizes a performance index. This kind of control is called closed loop control and uses the concept of feedback in the sense that it uses the output to generate a new input.

For our analysis we consider linear feedback, i.e. the case where the feedback force is linearly proportional to the system state vector. The control force is expressed as

$$\mathbf{F} = -\mathbf{K_f}\mathbf{X} \tag{4.6}$$

where $\mathbf{X}$ represents the state of the system at that time, and $\mathbf{K_f}$ is a gain matrix calculated through an optimization process. Since the state vector of the system contains both velocities and displacements, the terms velocity feedback and displacement feedback are used to differentiate the two contributions. The partitioned form is

$$\mathbf{F} = -\begin{bmatrix} K_d & K_v \end{bmatrix} \begin{bmatrix} U \\ \dot{U} \end{bmatrix} \tag{4.7}$$

where d and v refer to the displacement and velocity contributions.

Later, we will discuss how $\mathbf{K}$ is established.

4.1.1 Continuous feedback control with time delay

There is an important factor to take into account when redefining our model. It is called time delay which is represented with the symbol τ. It represents the effect of the interval between the time the sensors are read to the time the action is taken. It consists of the time intervals involved in sending the signal to the computer from the sensors, computing the action to be taken and sending the signal to the actuator for it to apply the action. This time delay cannot be avoided and it is present on any control device. Its effect on the performance of the system may be significant and therefore it should be studied carefully. Mathematically, this delay is easy to represent and conceptually easy to understand. However, the improved model is much more complicated to solve analytically. In later sections, we will examine how much information we can draw from the model through analysis and also present some models for simulation.

In the state-space formulation the delay has the effect that a force obtained from the state at a time $(t - \tau)$ will act at a time t, therefore the state space equation becomes,

$$\mathbf{\dot{X}(t) = AX(t) + B_f F(t - \tau) + B_g a_g(t)} \tag{4.8}$$

This formulation for the continuous case, which gives interesting information regarding the behavior of the delayed system, will be revised later for the discrete formulation.

4.2 Discrete time control

The main adjustment to make to the state-space formulation is the discretization of the control action. Since the readings and control forces are established at discrete times, the control algorithm need to be expressed in terms of quantities evaluated at these discrete times.

The solution of the differential equation for the time interval of size Δt from $k\Delta t$ to $k\Delta t + \Delta t$ is given by the sum of the decaying initial position at time $k\Delta t$ plus Duhamel's integral over the interval for the control input in that interval. The analytical solution over the interval is,

$$\begin{aligned} \mathbf{X}(k\Delta t + \Delta t) = e^{\mathbf{A}\Delta t}\mathbf{X}_{k\Delta t} + \textstyle\int_{k\Delta t}^{k\Delta t+\Delta t} e^{\mathbf{A}(k\Delta t+\Delta t-\tau)}\mathbf{B}_f\mathbf{F}(\tau)d\tau \\ + \textstyle\int_{k\Delta t}^{k\Delta t+\Delta t} e^{\mathbf{A}(k\Delta t+\Delta t-\tau)}\mathbf{B}_g\mathbf{a}_g(\tau)d\tau \end{aligned} \tag{4.9}$$

We will have to redefine the system matrices for this discrete case in order to have a formulation similar to the formulation for the continuous case. For simplicity, we will assume constant control action over the interval, that is a zero order hold. Our objective is to reduce the formulation to:

$$\mathbf{\dot{X}(j + 1) = \hat{A}X(j) + \hat{B}_f F(j) + \hat{B}_g a_g(j)} \tag{4.10}$$

We will forget for now about the term for the external excitation since for the up-coming stability analysis we will not need it. By defining the following:

$$\mathbf{\hat{A}} = \mathbf{e^{A\Delta t}} \tag{4.11}$$

and,

$$\mathbf{\hat{B}_f} = \mathbf{\bar{A}B_f} \tag{4.12}$$

where,

$$\mathbf{\bar{A}} = \mathbf{A^{-1}} \left[\mathbf{e^{A\Delta t}}\right] \tag{4.13}$$

The equation will be in the state-space form for the discrete time case.

Discrete linear feedback control

The reason why we introduce a discrete state-space formulation for the control problem is that the control force is actually updated at discrete points in time, when readings are taken from the current state of the system. This does not change the characteristics of the control vector from the ones mentioned for the continuous case. We still use negative linear feedback control, although the vector $\mathbf{F_j}$ is calculated only for those times for which we have a reading.

4.2.1 Discrete control with time delay

The discrete formulation has to be modified further if it is going to take into account also the time delay. The modification is analogous to that for the continuous case. Simply, the control action will be applied a few intervals after it is computed. In terms of the formulation, this case is represented by the following state-space equation:

$$\mathbf{\dot{X}(j+1)} = \mathbf{\hat{A}X(j)} + \mathbf{\hat{B}_f F(j-}\nu\mathbf{)} + \mathbf{\hat{B}_g a_g(j)} \tag{4.14}$$

where ν is just the ratio of the time delay over the time interval.

4.3 The optimization model

The study and implementation of structural control introduces a new aspect to conventional structural analysis. In control we have to select a force to apply to the structure. As in other selection problems, the choice depends on what are the goals that we try to achieve with the control force and what are the constraint. Basically, since we usually cannot apply the force that achieves all of our goals precisely, we have to try to get as close as possible, that is, we search for an optimum solution.

The methodology we apply, incorporates into one functional a series of factors such as the square of the energy that the control forces require or the square of the difference between the desired deformation and the actual deformation. We find the force by minimizing this functional. For the design of linear control systems, quadratic performance indices are used, and the problem reduces to a linear quadratic optimization problem.

The control optimization problem involves searching for an admissible control force f and trajectory in the state space x^*, such that the performance index $\mathbf{J}$ defined as

$$\mathbf{J} = s(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{f}(t), t)dt \tag{4.15}$$

is minimized.

The regulator problem is concerned with finding the control input such that the system is driven from an initial state to a constant, presumably zero, final state. It is linear regulator in the case in which the control is a linear function of the state as mentioned earlier.

In the structural control problem the functions are vectors since we discretized our system. In this case the quadratic performance index is the following,

$$J = \frac{1}{2}\mathbf{X}^{\mathbf{T}}(t_f)S\mathbf{X}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left[\mathbf{X}^{\mathbf{T}}(t)Q(t)\mathbf{X}(t) + \mathbf{F}^{\mathbf{T}}(t)R(t)\mathbf{F}(t)\right] dt \tag{4.16}$$

where $\mathbf{S}$ and $\mathbf{Q}$ are real symmetric positive semidefinite matrices and $\mathbf{R}$ is real symmetric positive definite. The matrices $\mathbf{Q}$ and $\mathbf{R}$ are used to penalize the size of different state variables or control forces respectively. For simplicity, it is best to make them diagonal. This way the penalties are uncoupled which should be the case.

From the derivative of the functional we obtain a system of differential equations on the state variables $\mathbf{X}$ and the costate variables $\mathbf{P}$ plus some boundary conditions. From this, we can solve for the optimal control vector as a function of the costate variable,

$$\mathbf{F} = -\mathbf{R}^{-1}(t)\mathbf{B}^{\mathbf{T}}(t)\mathbf{P}(\mathbf{t}) \tag{4.17}$$

We have a linear relationship between the state and costate variable $\mathbf{P}$.

$$\mathbf{P} = \mathbf{H}(\mathbf{t})\mathbf{X} \tag{4.18}$$

Then,

$$\mathbf{F} = -\mathbf{R}^{-1}(t)\mathbf{B}^{\mathbf{T}}(t)\mathbf{H}(\mathbf{t})\mathbf{X} \tag{4.19}$$

The system of equations can be satisfied for all t when $\mathbf{H}$ satisfies,

$$\dot{\mathbf{H}}(t) = -\mathbf{Q}(t) - \mathbf{A}^{\mathbf{T}}(t)\mathbf{H}(t) - \mathbf{H}(t)\mathbf{A}(t) + \mathbf{H}(t)\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^{\mathbf{T}}(t)\mathbf{H}(\mathbf{t}) \tag{4.20}$$

This equation is known as the Riccati equation and $\mathbf{H}$ is the Riccati matrix.

If the system is controllable, and matrices $\mathbf{A}$, $\mathbf{B_f}$, $\mathbf{Q}$ and $\mathbf{R}$ are constant which is our case, the matrix $\mathbf{H}$ approaches a constant value as the final time approaches infinity. Then, a limiting form of the Riccati equation is

$$-\mathbf{Q} - \mathbf{A}^{\mathbf{T}}\mathbf{H} - \mathbf{H}\mathbf{A} + \mathbf{H}\mathbf{B_f}\mathbf{R}^{-1}\mathbf{B_f^T}\mathbf{H} = \mathbf{0} \tag{4.21}$$

This equation is known as the algebraic Riccati equation. Assuming $\mathbf{H}$ is constant, i.e. taking the solution of the algebraic Riccati equation is reasonable. It is a valid solution, that is, the assumption that the Riccati matrix is constant is good.

In summary, the selection of the optimal control force requires the following series of steps. First, we must select the matrices $\mathbf{Q}$ and $\mathbf{R}$ according to our priorities. Then, we solve the algebraic Riccati equation (ARE). Various programs such as MATLAB contain routines for this computation. The solution yields the Riccati matrix $\mathbf{H}$. From this we can get the control forces using,

$$\mathbf{F} = -\mathbf{R}^{-1}\mathbf{B_f^T}\mathbf{H}X = -\mathbf{K_f}\mathbf{X} \tag{4.22}$$

Finally, we include this term in the state space equations for control,

$$\dot{\mathbf{X}} = \mathbf{AX} - \mathbf{B_f K_f X} + \mathbf{B_g a_g} \tag{4.23}$$

4.3.1 The optimization model for discrete control

The process of optimization described above has to be modified for discrete control. There are different approaches. Connor and Klink present a procedure in which the cost functional J is developed for one time interval. Optimization of this functional leads to the following expression

$$\mathbf{F} = -\mathbf{K}_{f,j}\mathbf{X}_j + \mathbf{G}_j\mathbf{a}_{g,j} \tag{4.24}$$

where

$$\mathbf{K}_{f,j} = \mathbf{L}_j \mathbf{e}^{\mathbf{A}_j \Delta t} \tag{4.25}$$

$$\mathbf{G}_j = \mathbf{L}_j \bar{\mathbf{A}}_j \mathbf{B}_{g,j} \tag{4.26}$$

$$\mathbf{L}_j = [\mathbf{R}_j + \{\mathbf{B}_{f,j}^T \bar{\mathbf{A}}_j^T \mathbf{Q}_j \bar{\mathbf{A}}_j \mathbf{B}_{f,j}]^{-1} [\mathbf{B}_{f,j}^T \bar{\mathbf{A}}_j^T \mathbf{Q}_j] \tag{4.27}$$

$$\bar{\mathbf{A}} = \Delta t \left[\mathbf{I} + \sum_{k=1}^{\infty} \frac{\mathbf{A}_j^k (\Delta t)^k}{(k+1)!} \right] \tag{4.28}$$

This algorithm allows for updating the system matrices and the time interval at each discrete time.

There is another procedure which gives essentially the same results when similar cases are treated. In this procedure, the performance index is discretized and summed over the total time. This procedure does not allow for updating of the system matrices.

The optimization of the functional results in a recursive equation. It is actually the discrete algebraic Riccati equation.

To solve for the gain matrix, it is necessary to assume a final point to start iterating backwards. If the value is chosen large enough, the iteration will lead to an initial value for the gain. We also can assume that the gain is constant, like in the continuous case, and use the initial value as a constant gain. The recursive equation is

$$\mathbf{H}_k = \mathbf{A}^T[\mathbf{H}_{k+1} - \mathbf{H}_{k+1}\mathbf{B}(\mathbf{R} + \mathbf{B}^T\mathbf{H}_{k+1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{H}_{k+1}] + \mathbf{Q} \tag{4.29}$$

Once we get $\mathbf{H}$ the gain is calculated with

$$\mathbf{K} = (\mathbf{R} + \mathbf{B}^T\mathbf{H}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{H}\mathbf{A} \tag{4.30}$$

If the matrices and time interval are constants and the initial value for the iteration is large, both procedures result in the same gain.

4.4 Other issues

In this section we identify other issues that need to be considered carefully prior to implementation of structural control. They represent problems that may arise when implementing the control on real structural systems.

4.4.1 Parameter sensitivity

In the algorithms for structural control we are being as precise as possible but we must not forget the limits of that precision. If we would obtain extremely accurate calculations on the simulation of the structural behavior when subjected to control forces we could get a very good estimate on the time dependant variables present in the differential equations. But these variables are not the only factors in the equations. There are also parameters that depend on the system being considered.

Slight changes in those parameters can sometimes make our solutions and therefore our conclusions completely wrong. Furthermore, in this case certain parameters are known to be very difficult to asses. For example, the estimate of the stiffness of a large structure can vary by fifty percent from the actual value. This variability in the parameters has to be studied and dealt with thoroughly in the analysis. Large structures have an additional complication; the size of the systems does not allow for inexpensive and simple testing or experimentation. Also, since it is a new area of research, there is only limited data on real applications.

All of these considerations reinforce the need for parameter sensitivity analysis. That is, how much does the information drawn from the model, such as deformations and conditions for stability, change when we make small and some time large changes in the system parameters, the mass, the damping and the stiffness. These sensitivity can be assessed with analytical tools, although they can get extremely complicated for real cases.

Sensitivity theory can give us useful information regarding the stability of the system when subjected to changes in the parameters. We are mainly concerned with the determination of the sensitivity of the eigenvalues and the eigenvectors of $\mathbf{A}$ to changes in the system parameters, that is, to changes in $\mathbf{A}$ itself. We asses the sensitivity of the eigenvalues and eigenvectors only. They alone describe the sensitivity of the system.

There are two main approaches to this problem. The first approach observes the changes in the eigenvectors and eigenvalues as each element in the system matrix $\mathbf{A}$ changes using the first derivative of those values with respect to each of these elements.

The other approach uses perturbation theory. Basically, we perturb the system matrix in some way and observe how this perturbation carries to the eigenvalues and eigenvectors.

4.4.2 Controllability and observability

This section refers to other two variations that our real system will present when compared to the initial ideal model. The first refers to the fact that for a large system we will not be able to put control forces at every degree of freedom. The second addresses the problem that for the same large system we cannot put measuring devices at every degree of freedom. This means that the vector space of the control force vector and output vector are smaller than the vector space of the deformations and forces in the system.

This is always going to be a problem for obvious reasons. It is simply impossible economically to control and to measure all the structure. For this reason, this problems has to be properly addressed.

Basically, we will be concerned mainly with modes that are not controllable or observable rather than with actual degrees of freedom. This makes sense since we are approaching the control problem from the modal point of view for we noted that it is only a few modes that are going to do most of the damage in the system. The complete control formulation for either natural or modal coordinates is,

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B_f}\mathbf{F} + \mathbf{B_g}\mathbf{a_g} \tag{4.31}$$

$$\mathbf{Y} = \mathbf{C}\mathbf{X} + \mathbf{D}\mathbf{U} \tag{4.32}$$

The matrix $\mathbf{B}$ is going to locate the forces and therefore it contains the information on how many control forces are we applying and where. The matrix $\mathbf{C}$ on the other hand determines which state variables we are reading to get the output vector $\mathbf{Y}$. Therefore, in this matrix our ability to read the different deformations is described. In a real system the optimization process for the selection of a control force is not carried out with the state variables $\mathbf{X}$ but with the available output from the system in the vector $\mathbf{Y}$. That is limited as we see in the formulas by the $\mathbf{C}$ matrix.

For the problem of controllability it is conceptually simple to understand how can we determine wether a mode is going to be controllable or not. It is a matter of orthogonality. If the multiplication $\mathbf{W_j}\mathbf{B_f}$ is null, clearly our control forces are orthogonal to that particular mode. Therefore, we will have no effect on that mode. In the same way as for the so called participation factors for the case of an external excitation, the larger this product is the more effect and control that we have over that mode.

For the observability, conceptually we have the same situation . In this case we multiply the matrix **C** by the eigenvectors and check that the main modes are not orthogonal to the output matrix **C**. It is from this modes that we need the information since we need to apply control based on their feedback.

4.4.3 Model reduction in modal analysis

Another issue to consider is the model reduction. The key steps in the analysis is the transformation to modal coordinates and the discarding of those modes which have little contribution to the global response.

It is necessary in an actual structural system to test different possible inputs to the system. This way, we are able to asses the magnitude of the various modal participation factors and verify our assumption as to the modes which are released. The actual number of modes depends on the size of the building and the excitation and is generally less than 10.

4.4.4 Nonlinearities

A common assumption in structural engineering is that systems behave linearly, which is never true although some times it is very close to it. Close enough, so that we can assume it is linear with no loss of accuracy. However, in dynamics it is convenient to take into account the effect of nonlinear behavior in the system.

This is most important in the case of control where the excitation is an earthquake. This is so, because in those circumstances the nonlinear behavior may become significant.

The nonlinear behavior can come from the excessive deformations, which increase second order effects. It can also be due to nonlinear behavior of the material. Due to the excesive deformations, the level of stress in the members reaches yielding and the material starts to show plastic behavior.

Both of this effect should be analyzed carefully and the state space formulation is an excellent tool for this kind of analysis. It provides direct access to the equations for displacement and velocity in the time domain, and it is in the form of a first order differential equation as opposed to second order equation.

This problem presents a very important area of research in structural control. Application of the concepts presented in this thesis must be preceded either by studies which show that the levels of deformation and stress do not lead to nonlinear behavior or a thorough study of nonlinear structural control.

Chapter 5

Stability Analysis and Simulation

We have introduced the use of state-space formulation in structural control and control in general as a problem that deserves our attention as civil engineers. We have already presented the formulation needed for the modelling and analysis of the structural control problem, that is, the differential equations in the state-space form for the continuous and discrete time cases. Now we are ready to use this model for the analysis and study of systems.

The first option will be to obtain analytical expressions describing the structural behavior. To get an analytical expression for the response of the system to arbitrary inputs is difficult and therefore this possibility is not considered. However , there are important characteristics of the system that can be obtained explicitly from the system matrix. The state space formulation has the advantage that all the information about the system is contained in one single matrix. The stability of the system is the most important piece of information that we can draw from this matrix.

The other alternative is generating the response numerically for a range of system parameters and loadings and identifying instability through numerical implementation. This approach is employed when it is too difficult to obtain an explicit analytical solution.

5.1 Analysis

In what follows, we present the analysis for continuous and discrete control in that order. The main purpose will be to obtain the system matrix for each of the systems. From the system matrix, we obtain the conditions for stability.

5.1.1 The continuous case

The continuous case without control

The analytical solution for this equation can be expressed as a convolution integral. If we define,

$$\mathbf{X(0)} = X_0 \tag{5.1}$$

then,

$$\mathbf{X(t)} = \mathbf{e^{At}X_0} + \int_0^t \mathbf{e^{A(t-\tau)}B_g a_g}(\tau)\mathbf{d}\tau \tag{5.2}$$

We showed that for the case of no external excitation the free vibration response for any initial perturbation uncouples when we transform to generalized coordinates. The solution can be expressed as a combination of the eigenvectors,

$$\mathbf{X} = \sum_{j=1}^{2n} \nu_j e^{\lambda_j t}\mathbf{V_j} \tag{5.3}$$

where ν_j represent arbitrary complex constants determined from the initial conditions such that,

$$\nu_j = \mathbf{W_j^T X_0} \tag{5.4}$$

Noting this equation, the homogeneous solution expands to

$$\mathbf{X} = \sum_{j=1}^{2n} \mathbf{W_j^T X_0 e^{\lambda_j t} V_j} \tag{5.5}$$

This solution is real even though the eigenvectors are complex.

For stability analysis, we look at the free vibration response

$$\mathbf{X} = \sum_{j=1}^{2n} \mathbf{W_j^T X_0 e^{\lambda_j t} V_j} \tag{5.6}$$

The eigenvalues occur in complex conjugate pairs. The imaginary part of each exponential is then transformed into periodic functions and these are multiplied by the exponential of the real part which is,

$$e^{\lambda_R t} = e^{-\xi\omega t} \tag{5.7}$$

Therefore, for the solution to be bounded, the frequency and the damping have to be positive. This is the only condition for stability in the continuous linear case. Since this is always the case we conclude that this case is unconditionally stable.

The continuous case with feedback control

Ideally, the feedback to the system should occur instantaneously and continuously. However, it is neither one or the other. In this section we will explain what the behavior of the system would be if this was the case. It should give a better understanding since it is possible to get an analytical solution for the conditions for stability in this case.

We investigate what may be the effect of feedback control on the stability of the system. The stability of the system for the case of no control depended on the eigenvalues of the system matrix. That is also the case when we apply the linear negative feedback control, but in this case the system matrix is altered by the control gain. Let us look at the controlled system matrix for the case of no external excitation.

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{B_f F} \tag{5.8}$$

where $\mathbf{F}$ the control force is,

$$\mathbf{F} = -\mathbf{K_f X} \tag{5.9}$$

then we can write,

$$\dot{\mathbf{X}} = (\mathbf{A} - \mathbf{B_f K_f})\mathbf{X} \tag{5.10}$$

To asses the stability we will look at the eigenvalues of the single degree of freedom system matrix with control, $\mathbf{A_c}$.

$$\mathbf{A_c} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} - \frac{k_d}{m} & -\frac{c}{m} - \frac{k_v}{m} \end{bmatrix} \tag{5.11}$$

For the single degree of freedom case, the eigenvalues of this matrix are,

$$\lambda = \lambda_R \pm \lambda_I \tag{5.12}$$

where,

$$\lambda_R = -\frac{c + k_v}{2m} \tag{5.13}$$

$$\lambda_I = \sqrt{\left[\frac{k + k_d}{m}\right] - \left[\frac{c + k_v}{2m}\right]^2} \tag{5.14}$$

We can draw important conclusions about the effect of the control on stability. First, we see that the system is stable for any velocity feedback, assuming k_v is always positive for negative feedback. The displacement feedback has no effect on the real part and therefore no effect on stability.

The continuous case with control and time delay

Before we proceed to the simulation model, we first extend the analytical solution for the continuous time model for the case of time delay. The state-space equation with time delay in the feedback is,

$$\mathbf{X}\dot{(}\mathbf{t}) = \mathbf{AX(t)} + \mathbf{B_f F}(\mathbf{t} - \tau) + \mathbf{B_g a_g(t)} \tag{5.15}$$

In order to study the stability, we look at the eigenvalues of the system matrix. The system matrix for the delayed system is determined by assuming the solution to have the following form

$$\mathbf{X} = \mathbf{C}\mathbf{e}^{\mathbf{\Lambda t}}\mathbf{V} \tag{5.16}$$

where the row vector $\mathbf{C}$ contains the constants with information regarding initial conditions, the matrix $\mathbf{\Lambda}$ is diagonal with eigenvalues in these entries, and the matrix $\mathbf{V}$ contains the eigenvectors. The response is a superposition of $2n$ initially disturbed, oscillating and decaying mode shapes. If we plug in this solution for free vibration we get the system matrix,

$$\dot{\mathbf{X}} = (\mathbf{A} - \mathbf{B_f}\mathbf{K_f}\mathbf{e}^{-\mathbf{\Lambda}\tau})\mathbf{X} \tag{5.17}$$

Conceptually, this is the same control force as before, only that since the system has linear feedback control the magnitude of the force is extrapolated backwards with a negative exponential factor with frequency $\mathbf{\Lambda}$ for a time τ, which is the delay time.

As before, we can solve for the single degree of freedom case in order to study the effect of control and delay simultaneously. The system matrix for this case would be,

$$\mathbf{A_c} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} - \frac{k_d}{m}e^{\lambda_1} & -\frac{c}{m} - \frac{k_v}{m}e^{\lambda_2} \end{bmatrix} \tag{5.18}$$

The requirement for stability is that the real part of the eigenvalues be negative. However, to obtain an analytical solution for this problem is very difficult. Connor and Klink use two different approximations. One is a first order estimate of the exponential, the other uses Pade approximation. The solution is still complicated because the effect of displacement feedback and velocity feedback is not the same. In this case, we have to study the effect of each with the time delay, where the effects of these are coupled.

From the approximations used by Connor and Klink we can draw conclusions. For pure displacement feedback and no delay the system is unstable. The poles in this case go in the positive direction of the real axis in the complex plane. For pure velocity feedback and delay the system is stable for the smaller values of feedback and for some margin of delay. The more time delay, obviously the closer we are to the unstable side of the complex plane with our poles.

Connor and Klink provide an analytical solution for the largest time delay that the system can take before it becomes unstable. This solution is available for a single degree of freedom system. For the case in which the system has many degrees of freedom, the equations have to be uncoupled. Then, the maximum time delay can be found for each degree of freedom independently. We can uncouple the system matrix using modal coordinates, however we have to uncouple also the control forces to be able to analyze each equation alone.

Since it is not guaranteed that the control forces are going to uncouple, we will use simulation to get more information on the behavior of the multidegree of freedom systems when there is time delay in the control.

5.1.2 The discrete model, analytical solution

The discrete time case without control

Earlier, we presented the analytical solution for the continuous time case with discrete control. Now we can try to use this model to draw conclusions regarding the conditions for stability in this case. We assume the discrete time model,

$$\dot{\mathbf{X}}(\mathbf{j}+\mathbf{1}) = \hat{\mathbf{A}}\mathbf{X}(\mathbf{j}) + \hat{\mathbf{B}}_{\mathbf{f}}\mathbf{F}(\mathbf{j}) + \hat{\mathbf{B}}_{\mathbf{g}}\mathbf{a}_{\mathbf{g}}(\mathbf{j}) \tag{5.19}$$

For the free vibration case with we have,

$$\dot{\mathbf{X}}(\mathbf{j}+\mathbf{1}) = \hat{\mathbf{A}}\mathbf{X}(\mathbf{j}) \tag{5.20}$$

Each solution can be obtained by multiplying the previous one by $\hat{\mathbf{A}}$. So after j iteration we will have raised $\hat{\mathbf{A}}$ to the power of j . This means that the solution is stable if $\hat{\mathbf{A}}^j$ is stable for any number j, and from the nature of $\hat{\mathbf{A}}$ we know that this is true if its eigenvalues have modulus less than one. This is equivalent for the continuous case to requiring the real part to be negative.

For the equation,

$$\dot{\mathbf{X}}(\mathbf{j}+\mathbf{1}) = \hat{\mathbf{A}}\mathbf{X}(\mathbf{j}) \tag{5.21}$$

We can assume a solution of the form,

$$\mathbf{X_j} = \rho^{\mathbf{j}}\mathbf{V} \tag{5.22}$$

where ρ contains the eigenvalues of the controlled matrix and $\mathbf{V}$ contains the eigenvectors of $\hat{\mathbf{A}}$.

The discrete time case with control

Again, we need the system matrix for the study of stability. Since this is a discrete control problem we will have a system matrix made of exponential terms like before. Therefore in order to be sure that the system is stable, all the eigenvalues of this matrix must have a modulus less than one. The state equation for free vibration is,

$$\dot{\mathbf{X}}(j+1) = \hat{\mathbf{A}}\mathbf{X}(\mathbf{j}) + \hat{\mathbf{B}}_{\mathbf{f}}\mathbf{F}(\mathbf{j}) \tag{5.23}$$

That means that, similarly to the continuous case, the controlled system matrix is,

$$\tilde{\mathbf{A}} = \hat{\mathbf{A}} - \hat{\mathbf{B}}_{\mathbf{f}}\mathbf{K_f} \tag{5.24}$$

The solution again has the form,

$$\mathbf{X_j} = \rho^{\mathbf{j}}\mathbf{V} \tag{5.25}$$

where ρ contains the eigenvalues of the controlled matrix and $\mathbf{V}$ contains the eigenvectors of $\hat{\mathbf{A}}$.

Discrete control with delay, analytical solution

In this case the state-space equation is,

$$\dot{\mathbf{X}}(\mathbf{j}+\mathbf{1}) = \hat{\mathbf{A}}\mathbf{X}(\mathbf{j}) + \hat{\mathbf{B}}_{\mathbf{f}}\mathbf{F}(\mathbf{j}-\nu) \tag{5.26}$$

Here, ν is the ratio of time delay over discrete time step. The controlled system matrix is more complicated because the delay, analogous to the continuous time case, introduces an exponential term in the eigenproblem which is extrapolating backward in time to get the control force proportional to the earlier time. Plugging in the solution,

$$\mathbf{X_j} = \rho^{\mathbf{j}}\mathbf{V} \tag{5.27}$$

we have the eigenvalue problem

$$\left[\hat{\mathbf{A}} - \rho\mathbf{I} - \rho^{-\nu}\hat{\mathbf{B}}_{\mathbf{f}}\mathbf{K_f}\right]\mathbf{V} = \mathbf{0} \tag{5.28}$$

Connor and Klink have developed analytical solutions for the case of a single degree of freedom to determine the value of the ratio ν for which the poles of the system are in the imaginary axis, that is, the system becomes unstable. Like in the continuous time case, the solution is available for the maximum values of the time delay and time step for which the system becomes unstable.

For the case of multiple degrees of freedom, if the damping matrix is made proportional, the control forces uncouple and it is possible to obtain the limiting values of time delay and time step for which the system is stable. However, if the state-space formulation is used, the equations do not uncouple and it becomes much more convenient to use simulation.

5.2 Simulation

Next we present the models which help us find the conditions for stability by using simulation instead of analysis.

5.2.1 The discretization of the structural response

In the first place, we will study the model of the procedure used to solve the equation numerically. The stability of this model does not have anything to do with the stability of the actual system. With this difference equations we are trying to obtain a solution close to the response of the real system. If the discrete solution is far from the real one, for having the steps in time too long for example, the numerical solution may be unstable while the actual response of the system does not. Needless to say we do not want that to happen.

There are different schemes available for the iterative solution of this kind of problem and two of those methods are the famous Euler and Runge-Kutta methods. For simplicity we started modelling with a first degree Euler method. It is the simplest version but for the purpose of this paper it is appropriate. From this we can understand much of how the system works. For this model we can integrate very fast so we can use a small step in time, this way we do not influence the results with the integration scheme and we are able to focus on other aspects of the system behavior.

One of the conveniences of the state-space formulation is that it presents a first order differential equation. That means that if we know the state of the system at any given time, simply by multiplying it by the system matrix we can get the slope or rate of change at that time. This way, in order to find the position at a close time we only have to use a small step in order to keep our result accurate enough and stable. For the case of no external input and no time delay our state equation is,

$$\dot{\mathbf{X}} = \mathbf{AX} \tag{5.29}$$

We will use $\mathbf{X(1)}$ to represent $\mathbf{X(t_1)}$ and similarly for $\mathbf{t_0}$, such that our time step is defined as,

$$t_1 - t_0 = \Delta t \tag{5.30}$$

therefore for an initial disturbance $\mathbf{X(0)}$, in order to get a good estimate of $\mathbf{X(1)}$ we use the simplest estimate.

$$\mathbf{X(1)} = \mathbf{X(0)} + \Delta t\mathbf{AX(0)} \tag{5.31}$$

In this case to incorporate the external excitation, in the form of an earthquake, and the control forces should be very easy. In fact, the ground acceleration just produces accelerations in the nodes, which are first order changes on some of the state variables. Same applies to the control forces, so the iterative equation becomes,

$$\mathbf{X}(\mathbf{1}) = \mathbf{X}(\mathbf{0}) + \Delta t(\mathbf{A}\mathbf{X}(\mathbf{0}) + \mathbf{B_g a_g}(\mathbf{0}) + \mathbf{B_f F}(\mathbf{0})) \tag{5.32}$$

The earthquake record must be available at each time step or interpolated if not available. If we want our analysis to be satisfactory we will concentrate on minimizing the Δt mentioned above.

5.2.2 Simulation of discrete readings

Once we have found an iterative method to model the behavior of the actual structure we are ready to simulate the control action with iterative procedures. We must incorporate effects such as discrete readings and time delay in the simulation model.

In order to incorporate the fact that readings are taken at discrete times and actions are taken at those times only we will make a loop such that the state used to calculate the control forces to act at any time remains the same for a few of the following iterations on the integration scheme, presented earlier. This means that if we want our loop to work we have to make the time interval in the discrete readings a multiple of the integration step Δt. If we want to be precise in the intervals for the discrete readings we can either reduce more and more Δt or to calculate the ratio of the two intervals and use the closest integer as the number of iterations that the control force should remain constant. We will call the reading intervals with the letter ρ.

The algorithm has to start with a reading of a deformed structure, calculate the optimum control action and apply it constantly until a new reading is received which allow us to calculate a new force. Until we get the new reading, the control force is held constant for the following intervals. The iterative procedure is just the Euler method presented above in which the control force is changed only every few intervals. In the simulation program, a test has to be made before the new state is calculated. In this test we check if the number of the interval is a multiple of the ratio of the

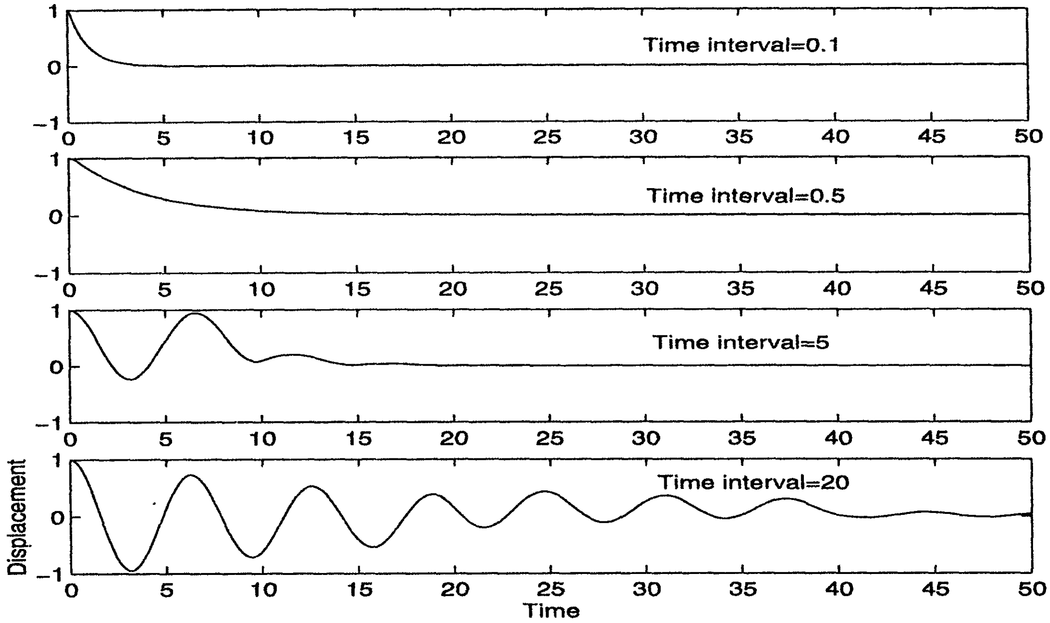

Figure 5-1: Response for initial displacement for different values of time interval

integration step and ρ.

There is the possibility that new readings are available at shorter intervals than new forces can be applied. In that case, the speed of the load generator controls over the speed of the readings, but this case is not considered.

We apply control with this model to a one degree of freedom system. Consider the simple case where $m = 1$, $c = 0.1$ and $k = 1$. In this case the state-space system matrix is simply

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & -0.1 \end{bmatrix} \tag{5.33}$$

We simulate the behavior of the system when subjected to an initial disturbance of

$$\mathbf{X_0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5.34}$$

that is an initial displacement of 1. The matrices $\mathbf{Q}$ and $\mathbf{R}$ are simply identity matrices. We use an integration step of 0.01 for 10 seconds and then try different time steps in the discrete control to observe the changes in the behavior.

From the figure we observe that when the time interval is small the control is very effective. However, the control is less when the interval is larger. Finally, we have no control for very large intervals. In that case, the response is almost free vibration.

We can observe that from the magnitude of the gain matrix. When the time interval is 0.1 the gain matrix is

$$\begin{bmatrix} kd & kv \end{bmatrix} = \begin{bmatrix} 7.793 & 9.582 \end{bmatrix} \tag{5.35}$$

and when the interval is 5 we get

$$\begin{bmatrix} kd & kv \end{bmatrix} = \begin{bmatrix} -0.2710 & -0.6816 \end{bmatrix} \tag{5.36}$$

As the time interval becomes larger, the control becomes smaller. We observe, that the system does not become unstable for larger time intervals, the system is still unconditionally stable.

5.2.3 Simulation of the discrete model with time delay

Next, we just update the algorithm mentioned above for the discrete control for the simulation of time delay. We want the force that should start acting on the system at a given point in time to start to act a few intervals later. The number of intervals is simply calculated dividing the delay time by the integration step.

The implementation of the algorithm for simulation presents similar problems to the ones presented above for the case of discrete readings. We have to choose the time delay such that it is a multiple of the integration step. Alternatively, we can do like for the discrete case, use the closest integer to the ratio of time delay to step. If we are very interested in a precise time delay we can improve the integration scheme for the dyanmic behavior.

That is the complete simulation scheme in which we are, in some way, running a discrete model on top of another. The underlying discrete model represents the continuous behavior of the structure and the one on top the actual discrete behavior

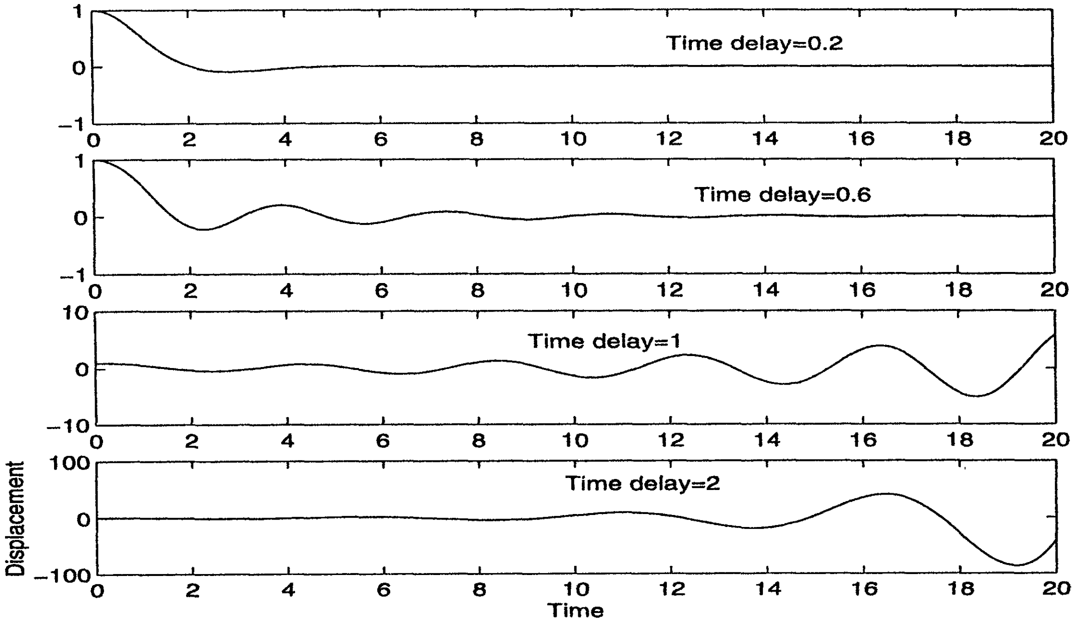

Figure 5-2: Response to initial displacement for different values of the time delay

of the digital control tools.

We try an example in which the increment in the time delay has a large effect on the performance of control system. In this case, an excessive time delay leads to instability. We use the same system as before with a time interval of 0.2 and with different values for the delay.

For a small time delay, the system is controlled very effectively. Around the value of 0.9 for the time delay, the system is unstable. For larger values we can see the effect of the time delay is very exagerated.

5.2.4 More simulation

Variations in the weighting matrix R

By changing the values in the **R** matrix we affect directly the optimization process. The larger the entries in the matrix, the more expensive it is to apply forces, and the smaller these are so that we have less control of the system.

In this case we vary the parameter **R** and observe the changes in the response

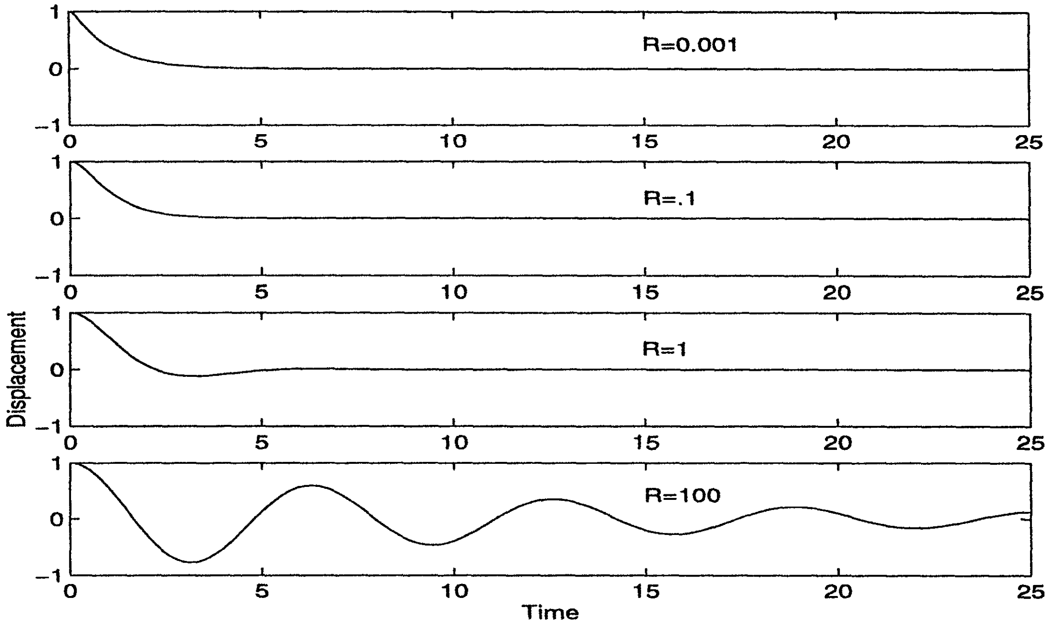

Figure 5-3: Response for initial displacement for different values of matrix R

and the changes in the magnitude of the control force. The values that we tried, for a time interval of 0.2 are 0.001, 0.1, 1, 100.

In the first figure we see that as expected the control is less as the value of **R** is increased.

In the plot of the control force, we see the large reduction in the magnitude of the control forces as the cost increases.

Modal control

We change to modal coordinates and apply the control to these coordinates. The system matrix that we used earlier

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & -0.1 \end{bmatrix} \tag{5.37}$$

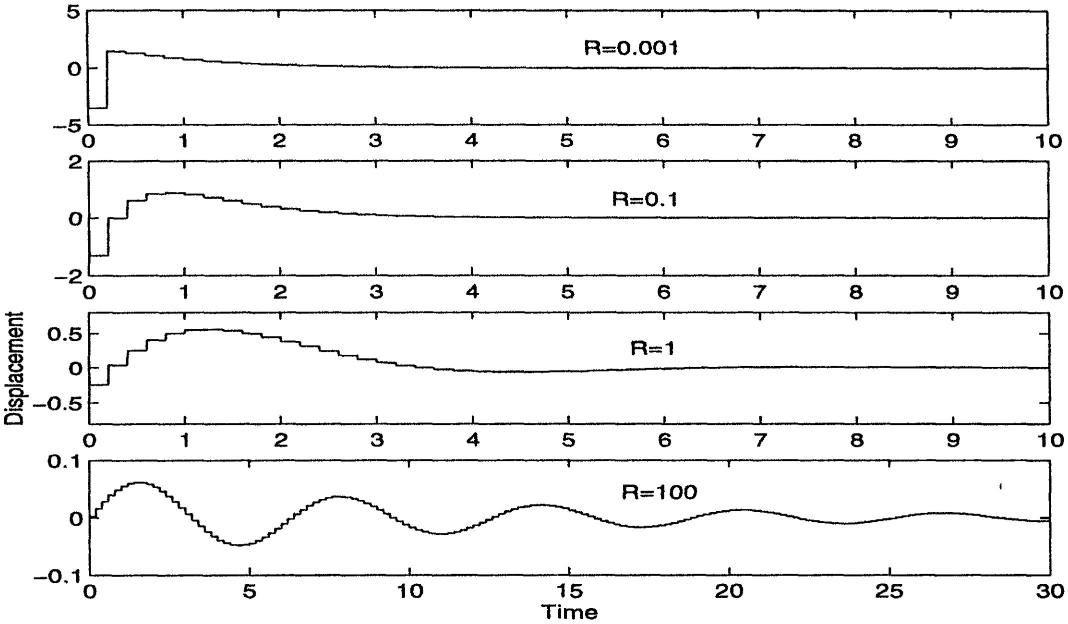

Figure 5-4: Control forces applied for different values for R

has the eigenvalues given in the diagonal entries of Λ.

$$\Lambda = \begin{bmatrix} -0.0500 + 0.9987i & 0 \\ 0 & -0.0500 - 0.9987i \end{bmatrix} \tag{5.38}$$

therefore, when we change the coordinates, the system matrix is

$$\mathbf{A}^* = \begin{bmatrix} -0.0500 & -0.9987 \\ 0.9987 & -0.0500 \end{bmatrix} \tag{5.39}$$

Also, the matrix $\mathbf{B_f}$ becomes

$$\mathbf{B_f^*} = \begin{bmatrix} 0.0500 \\ -0.9987 \end{bmatrix} \tag{5.40}$$

We solve numerically the control problem for these new system equation and obtain the response in the figure. For a single degree of freedom we know that the response of the first mode is simply the response of the system. So if we plot the response of

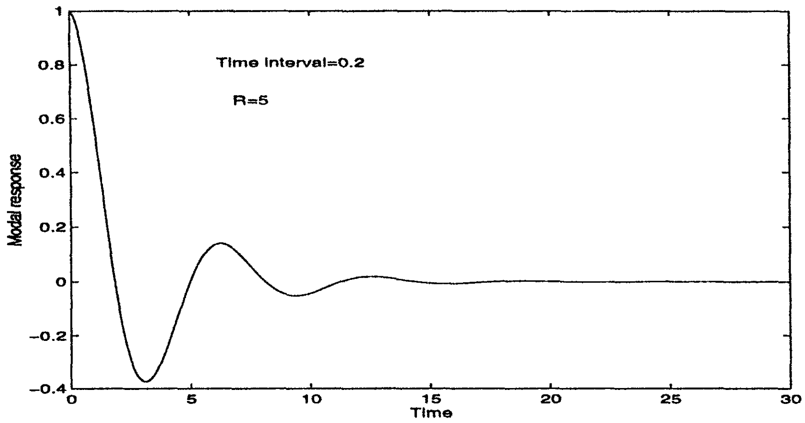

Figure 5-5: Response of the first mode for initial displacement in a single degree of freedom system

the first coordinate of the transformed state-space vector we get the same result than when we plot the response of the displacement in natural coordinates.

Bibliography

[1] Klaus-Jurgen Bathe. *Finite Element Procedures in Engineering Analysis.* Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[2] James A. Cadzow and Hinrich R. Martens. *Discrete-time and Computer Control Systems.* Prentice Hall, Englewood Cliffs, New Jersey, 1970.

[3] A. R. Collar and A. Simpson. *Matrices and Engineering Dynamics.* Ellis Horwood Limited, Chichester, England, 1987.

[4] Jerome J. Connor and Boutros S. A. Klink. *Introduction to Motion Based Design.* Plenum Press, Cambridge, Mass., 1995.

[5] John J. D'Azzo and Constantine H. Houpis. *Lineal Control Systems Analysis and Design.* McGraw Hill, Inc., New York, N.Y., 1995.

[6] Paul M. Derusso, Rob J. Roy, and Charles M. Close. *State Variables for Engineers.* John Wiley and Sons, Inc., New York, N.Y., 1965.

[7] R. A. Frazer, W. J. Duncan, and A. R. Collar. *Elementary Matrices.* The Macmillan Company, New York, N.Y., 1946.

[8] Jagmohan L. Humar. *Dynamics of Structures.* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[9] Vladimir Kucera. *Analysis and Design of Discrete Linear Control Systems.* Prentice Hall Intenational, Hertfordshire, U. K., 1991.

[10] Frank L. Lewis. *Applied Optimal Control and Estimation.* Prentice Hall, 1992.

[11] Leonard Meirovitch. *Dynamics and control of structures.* John Wiley and Sons, Inc., New York, N.Y., 1990.

[12] P. Hr. Petkov, N. D. Christov, and M. M. Konstantinov. *Computational Methods for Linear Control Systems.* Prentice Hall International, Hertfordshire, U. K., 1991.

[13] Reinhardt M. Rosenberg. *Analytical Dynamics of Discrete Systems.* Plenum Press, New York, N.Y., 10 October 1977.

[14] S. Timoshenko and D. H. Young. *Discrete-time and Computer Control Systems.* McGraw Hill Book Company, Inc., New York, N.Y., 1948.

[15] LaMar K. Timothy and Blair E. Bona. *State Space Analysis: an Introduction.* McGraw Hill, Inc., New York, N.Y., 1968.

[16] Hugh F. VanLandingham. *Introduction to Digital Control Systems.* Macmillan Publishing Company, New York, N.Y., 1985.