ADAPTIVE STRUCTURAL CONTROL

by

Bernard K. Lee

B. Eng. in Civil and Structural Engineering,

University of Bristol, UK, 1997

SUBMITTED TO THE DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING

IN CIVIL & ENVIRONMENTAL ENGINEERING

at

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1998

Signature of Author ______________________________

Department of Civil and Environmental Engineering

May 8, 1998

Certified by ______________________________

Professor Jerome J. Connor

Thesis Supervisor

Accepted by ______________________________

Professor Joseph M. Sussman

Chairman, Departmental Committee on Graduate Studies

ADAPTIVE STRUCTURAL CONTROL

by

Bernard K. Lee

Submitted to the Department of Civil and Environmental Engineering on May 8, 1998, in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Civil and Environmental Engineering

ABSTRACT

In this complex, ever-changing world, the demand for the performance and functionality of buildings is constantly increasing. Due to the scarcity of resources and information, there is a need to continually improve our design and analysis methods. Only then will there be improvements in technology. Adaptive control provides us with a means to this end.

Adaptive technology covers a very broad scope. The aim of this thesis is to take a close look at some important aspects of adaptive control and how it can be applied to structural systems. Some potential applications are developed and discussed, for example the personalized environmental control. An attempt is also made to combine AI technologies like neural networks, knowledge based systems, and fuzzy logic, with control theory in order to achieve an adaptive behavior.

Optimization techniques are integral to adaptive control. Many adaptive controls involve the use of advanced optimization methods. A MATLAB application is developed to carry out an optimal stiffness design. This application attempts to calculate a total stiffness value that will best satisfy both wind and earthquake spectrums.

Another MATLAB application is developed to demonstrate the learning capability of neural networks. By giving the network sufficient neurons, an adequately long training time, and numerous training cycles, the network can be shown to exhibit good learning characteristics.

The third short application demonstrates the use of neural networks to predict input signals. After the network has been trained, it is able to predict the input signal over the next time step, from data gathered from delayed input signals. It is found that with the use of more delayed signals and more neurons, the neuron is able to handle more complex inputs.

Finally, an attempt was made to tie everything together to the holistic picture of adaptive controls.

Thesis Supervisor: Professor Jerome J. Connor

Title: Professor of Civil and Environmental Engineering

ACKNOWLEDGEMENTS

This thesis concludes my experience with the M. Eng. program under the CEE department at MIT. Many people have granted me their help in the writing of this thesis, and I would like to make use of this opportunity to express my appreciation and gratitude.

In particular, I would like to thank Professor Jerome J. Connor, my thesis advisor, for his steady encouragement and excellent guidance. He was instrumental in inspiring my work in this non-conventional field for Civil Engineering. It has indeed been stimulating.

I would also like to thank Professor Kevin Amaratunga and Professor Annaswamy for their advice and insight.

I would also like to express my gratitude towards the administrators of the M. Eng program: Professor Dave H. Marks, Professor Connor and Charlie Helliwell. My fellow colleagues in the M. Eng program also deserve much credit. Throughout the 9 months of this program, many of them have given their help, both in academic work and moral support. They also provided great companionship.

I also wish to thank the members in my “Smart Technologies” project team, Lisa, Maria, and Simonetta. It had been great working with you guys. Special thanks goes to Tim, who helped me with many aspects of my work, given his previous experience.

My brief 1 year here has been made all the more memorable by my friends, Tracey, Allen, Cheewe, Terence, Weiyang, Bryan, Kelvin, Kok Keong, Xuanhui, Choonsiang, Cheewee and Waikit, just to name a few.

I wish to thank my parents for instilling the spirit of independence in me and giving me free leash in whatever I want to do in life, while at the same time, giving me their care and support. And to my sister, Esther: thanks for all the times that you have prayed for me and shared with me.

Finally, I would like to express my gratitude to my sponsor, the Public Service Commission of Singapore, for giving me this scholarship to study abroad. This opportunity has not only increased my professional knowledge, but also broadened my horizons.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation and Objectives

Buildings are becoming more complicated due to increased owner and occupant demands, and the constant need for innovation and technological improvement. As a result, controls have been used rather extensively, for the simple fact that they enable the structure to respond to changes in external stimuli. However, more needs to be done in order to improve our understanding and use of controls.

The objective of this thesis is to address the use of adaptive controls in intelligent structures. Adaptive controls can be implemented in intelligent structures in a myriad number of ways, ranging from Heating, Ventilating and Air Conditioning (HVAC) systems and Diagnostics to building stiffness and buckling. A brief account of the background developments and some existing applications will help illustrate the motivations for this thesis and the various issues addressed.

A brief discussion will be carried out on the various advanced technologies used to effect adaptive controls. Some applications pertinent to intelligent structures will also be presented with a detailed discussion and formulation performed for a variable stiffness control.

1.2 Background

The concept of adaptive controls originated from the vision of systems that can improve their performance by acting in the environment, observing the consequences of these actions, and optimizing their structure and parameters.

Adaptive control has its roots in the mid-fifties. Classical control theory as we know it now, having made significant progress during the World War II years, had many successful applications and became the control tool of preference. Classical control theory uses frequency domain models. Nonlinearities and time varying effects must be handled through exploiting the robustness margin of the control loop. The performance of the loop is thus not constant but changes with the operating point. As a consequence, when the time variations and nonlinearities are severe, it will not be easy to find a controller that can cope.

The problem that initiated adaptive control was of a plant with a linear part and a time varying gain. The gain varied slowly, compared to the natural dynamics of the linear part, or changed abruptly due to effects outside the plant's control loop. Moreover, the performance requirements on the closed loop were very tight. Therefore, a single simple classical controller could not cope with the expected range of gain variations.

The problem can be approached as follows. Firstly, assuming the gain is known, a controller that meets the required specifications can be designed using classical theory. Next, compensate the gain by its inverse, and then control the thus compensated plant with the designed controller. Of course, the gain being unknown, has to be identified. This combination of a linear control design with known parameters, together with the identification of a parameterized model of the unknown (here the gain) came to be known as adaptive control.

The *raison d'etre* of adaptive controls is to meet performance criteria over a large range of varying operating conditions. Large here means that a single simple controller will not be able to cope. On top of that, it is also the aim to have similar performance over the whole operating range. This adds to the difficulty.

Since then, adaptive controls have undergone significant development [1, 2, 3]. These involve some debates on the precise definition of adaptive controls, and proofs in stability, and robustness issues. The area of adaptive control has grown to be one of the richest in terms of algorithms, design techniques, analytical tools and modifications [1,2, 3, 4]. Only a few of these will be mentioned in this thesis, with respect to applications in high performance structures.

1.3 Applications

Adaptive controls can be applied to a wide variety of problems ranging from ore crushing to hemodialysis. Example applications can be found in most literature and a number of webpages [1, 2, 3, 4, 5]. In many of these instances, the performance of the system was greatly improved compared to that through the use of classical controls. This is mainly because the adaptive controller can be less cautiously tuned to handle worst case situations, and therefore provide better control.

Comparatively, the application of adaptive controls in buildings has been less developed. This is because more emphasis is usually given to cheapness of design rather than high performance. In aeronautical structures, there is more justification for adaptive controls because they improve performance and might even possibly reduce maintenance costs.

However, civil and structural engineers are gradually accepting adaptive technology. Improvements in human comfort and functional efficiency will be two factors that will continue to push this development. The potential benefits are great, both for indoor and outdoor applications.

Outdoors

Outdoor structures can be designed to respond to changes in external loading, light intensity, climatic conditions or time. These will be able to continually change their states, be it opacity, geometry or stiffness. For example, the stiffness or damping of a building can be adjusted in accordance with the type, strength or frequency of the loading experienced. In this way, structural integrity can be preserved, while at the same time minimizing the amount of redundancy. With the use of neural networks and other AI technologies, it might even be possible to design the building to cope with unusual loads.

The forms of roofs and membrane structures can also be constantly altered in response to changes in wind strengths and distributions. This can be carried out together with the use of Computational Fluid Dynamics to achieve a more accurate simulation of the force and pressure distributions on the structure. Architectural aspects of designs can also be enhanced by tuning the control of movable structures, for example, those by Santiago Calatrava [7] to changes in lighting conditions. Innovative structures, like the unfolding structures designed by Chuck Hoberman [8] will also benefit from the use of adaptive controls.

Indoors

HVAC is an area where adaptive controls can be put to good use. HVAC controls the indoor environments in which people work [14]. Traditionally, HVAC systems are only simple systems that are directly controlled from switches operated by users; there is very low versatility. Energy is often wasted due to the fact that heating or air conditioning is often left switched on even though nobody occupies the room. Recently, many integrated actively controlled HVAC systems have been used.

Adaptive controls can be used to manage personalized environmental controls. Fuzzy Logic can be used to characterize the parameters of the indoor environment and thus allow users to conveniently set their preferences. Each of the users in the building will have a smart Identification (ID) Card. Besides providing access to the building, this card will store information about the user's environment preferences. Whenever a person enters a room, the HVAC system will recognize his presence and set the environmental conditions to that user's particular preference. Neural Networks can then be used to learn the users' preferences for different situations and even predict and alter the environment depending on external temperature, number of people in the room or the time of day.

Certain sensors can also be carried on the user to measure his body temperature and then send that information via infrared rays to the control panel [9]. If the body temperature is high, the control system will know that the user has just done some walking or exercise and therefore reduce the room's temperature. With continuous data collection from the sensors on the user's body, the control system will be able to alter the room temperature accordingly, raising it gradually as the user's body temperature falls.

Simple models can easily be implemented for most of these systems, and they have proven to be rather effective. Such applications are most successful in digital signal processing. Bernard Widrow, one of the co-founders of the Least Mean Square (LMS) algorithm for use of neural networks in controls, has had much success in this field.

Many software applications have also been written to perform adaptive control. In more recent times, with the combined use of neural networks and other AI technologies, these controls can be made to learn the characteristics of various other systems and thus enable the controller to cope with new inputs. However, stability issues in such applications are still not fully resolved.

These are only some possible applications. Adaptive technology is an exciting field of work. Researchers have been trying to emulate human cognitive behavior in machines for a long time. Due to the gradual maturing of the research and development, the number of possible applications are increasing. Some of the above mentioned ideas are bold, but they must not be cast away, in order for there to be technological breakthroughs.

Chapter 2

Adaptive Technology

2.1 Characteristics

A good design for a control system is not easy. It also requires a large amount of time and money, as well as highly skilled personnel. However, an accurate design is only possible if the system if fully deterministic. In many real engineering systems, this is not the case. Many assumptions and approximations have to be made. A possible solution will be to use robustness in the design, that is, to make the controller immune or stable to the time-varying changes. However, there is a trade-off between robustness and the control quality. The more robustness there is, the slower the controller will be.

Therefore, it makes sense to have a controller that does not need to be very accurately designed from the start. This is possible since the controller can act on the environment, observe the consequences of these actions, and then optimize the structure and/or the parameters in order to improve its modeling accuracy. This is more or less the definition of adaptive controls. Below are some practical characteristics of adaptive controls:

1. The design of an adaptive control must be based on measured data. This criterion excludes control systems designed by methods based on the knowledge of the system. Clearly, some problem-specific definitions such as stating which variables can be measured, lie within this criterion. However, the measured data must contribute most of the information for the design.

2. The control must be performed automatically. All essential problem-dependent choices must be taken algorithmically, and not by the designer.

3. The design must be performed in real time. This will depend on the sampling rates used in the application and also the lifecycle of the application and of the changes to which it has to be adapted. Obviously, the more time a control system has to react to the environmental changes, the more feasible it will be to implement that control. The computational platform used is also very important. An adaptive algorithm will be more feasible to implement on a powerful parallel computer, than say, a microprocessor.

4. An adaptive control cannot be easily distinguished from feedback control, in general, and robust control, in particular. Feedback control action is computed from measured data automatically and in real time. While by definition, adaptive controllers have to contain parameters whose values are changing with the adaptation, the parameters can also be viewed as the state of the controller that is evolving under the influence of the data. The control parameters should be changed by a separate adaptation process or algorithm, typically taking place in a time frame some orders of magnitude slower than the control itself.

2.2 Technical Concepts

Adaptive controls can be loosely defined as systems that consist of a primary feedback that handles process signal variations and secondary feedback that handles process parameter changes. The primary feedback is the part that corresponds to active control and the secondary feedback part is what makes the control adaptive.

An example of a typical system can be described by the following linear differential equations:

$$\mathbf{x} = \mathbf{Ax} + \mathbf{Bu}\ , \qquad \mathbf{x}\,(0) = \mathbf{x_0}$$
$$\mathbf{y} = \mathbf{C}^T\mathbf{x} + \mathbf{Du} \qquad (1)$$

where $\mathbf{x} \in \boldsymbol{R}_n$ is the state of the model, $\mathbf{u} \in \boldsymbol{R}_r$ the system input, and $\mathbf{y} \in \boldsymbol{R}_l$ the system model output. The matrices $\mathbf{A} \in \boldsymbol{R}_{nxn}$, $\mathbf{B} \in \boldsymbol{R}_{nxr}$, $\mathbf{C} \in \boldsymbol{R}_{nxl}$ and $\mathbf{D} \in \boldsymbol{R}_{lxr}$ can be completely unknown and changing with time or operating conditions. A controller based on this linear model will be easier to understand and implement than one that utilizes a more accurate but non-linear model.

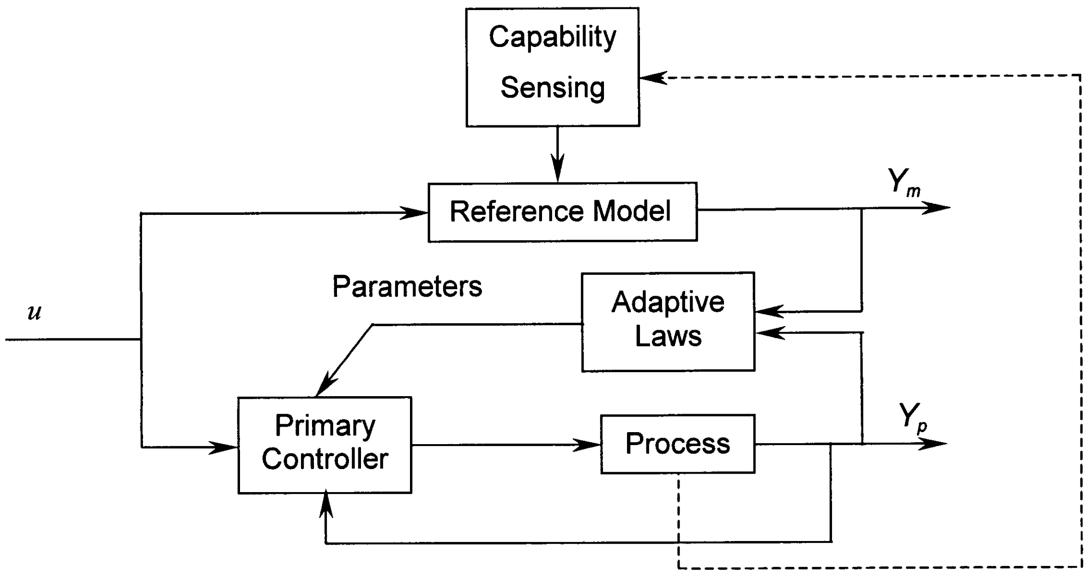


Fig. 1: Model reference adaptive control.

Figure 1 shows a scheme of a Model Reference Adaptive Control (MRAC). [3] Whitacker first introduced this technique in 1958. There is a primary controller that is used to obtain suitable closed-loop behavior. However, because the process parameters are either unknown or time varying, a fixed parameter setting for the controller cannot be obtained such that the closed-loop behavior is acceptable under all circumstances. The desired process response to a specified input is represented in the parametrically defined reference model. An adaptation mechanism then compares the process output $\mathbf{y_p}$ with the model output $\mathbf{y_m}$, and calculates a suitable parameter setting such that the error between these 2 outputs tends to zero. The process state, $\mathbf{x_p}$, and the process input $\mathbf{u}$ or the reference signal $\mathbf{r}$ may also be used by the adaptation mechanism.

As shown in Fig. 1, the MRAC scheme consists of 2 feedback loops: the primary (lower) loop which operates at a higher rate and the secondary (upper) loop that contains the adaptive mechanism. Due to this lag in the speed of the secondary loop, the process parameters are assumed to vary more slowly than the process states.

In the secondary loop, the adaptive laws will use the output error to produce an estimation and institute a change in the process parameters. This is a feature of an indirect scheme. A direct scheme will be one where the adaptation leads directly to the controller parameter changes without an explicit parameter estimation part.

Due to the fact that adaptive control algorithms are designed to handle unknown, non-linear or time-varying parameters in actual systems, they are prone to instability. If the known bounds of the controller response to the nonlinearities in the system had not been properly set up, even small disturbances can cause the adaptive scheme to go unstable. Therefore the methods used to derive the adaptive laws are very important. These methods will have to ensure robustness, which guarantees signal boundedness in the presence of "reasonable" classes of unmodeled dynamics and bounded disturbances as well as performance error bounds that are of the order of the modeling error. It was not until these instabilities could be counteracted in the mid-1980s, that adaptive schemes can be properly utilized.

So far, the design of an adaptive control has been discussed. However, other than the control algorithm, other decision support capabilities in the form of rule-based systems and neural networks are also necessary. These provide the capability to adjust the system parameter dynamically and introduce the possibility of a learning system.

2.3 Active and Adaptive Materials

Active materials are often incorporated with logic to coordinate the behavior of a building. Many of these materials can also be considered as adaptive since in effecting the control, they usually undergo a change of state. These materials are often used as actuators in controlling the stiffness or damping of a structure or element. A piezoelectric ceramic is one example. It produces a voltage proportional to its displacement and vice versa. Many commercial products using such materials are available in the market, an example of which can be seen in Fig. 3. There are generally 3 types: electro-rheological fluids, electrostrictive ceramics, and shape memory alloys.

2.3.1 Electro-Rheological Fluids (ERF)

ERF's are colloidal suspensions. Their viscosity and shear yield stress can be controlled within a certain range by an electrical field applied to it. 2 common examples are corn starch and corn oil, or zeolite and silicone oil [9]. This phenomenon was first observed by W. M. Winslow in 1947. The particles suspended in the fluid get polarized by the electric field and form chain-like structures (Fig. 2) between the electrodes along the direction of the electric field. These structures prevent free flow of the ERF and effectively transform it into a plastic. For example, for field strengths of about 3kV/mm, ERFs 'solidify' with static and dynamic yield stresses as high as 10kPa and 5kPa respectively. The electric field effectively changes the stiffness and damping of the structure. This liquid-to-solid transition is fully reversible. On top of that, these changes can be reversed in just a few milliseconds. Therefore, ERFs offer excellent control capabilities.

ERFs have been used extensively, in engineering devices such as brakes, clutches, hydraulic valves, dampers and engine mounts. However, ERFs have their drawbacks. Problems like leakage, corrosion, evaporation, phases or compounds separation etc., often occur. Moreover, as much as the ERF can improve the dynamic response of the system, it cannot improve the static response significantly since it is a fluid. Even if it is constantly active, the ERF will only have a maximum yield stress of a few hundred Pascals. Therefore, its usefulness is limited.

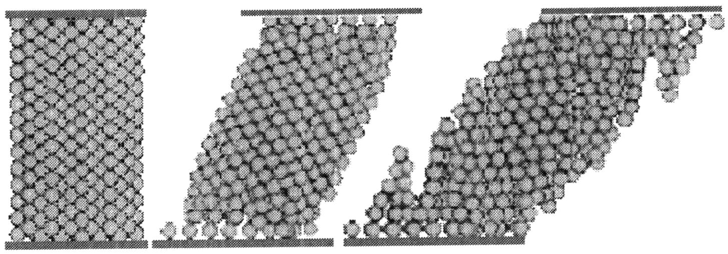


Fig. 2: Snapshots of the suspension particles of ERF under shear loading. The initial structure is a body centered tetragonal lattice consisting of 822 particles.

Magneto-Rheological Fluids (MRF) have slightly better qualities. They are higher in strength and require lower fields to activate the fluids [9].

2.3.2 Electrostrictive Ceramics

Fig. 3: Electrostrictive Actuator.

Electrostrictive ceramics produce a stress or strain if a voltage is applied to it. However, they differ from piezoelectric materials in that its strain, $x = ME^2$ whereas for piezoelectric materials, $x = dE$, where M and d are electrostrictive and piezoelectric constants respectively. Both types of ceramics have a fast response (10μs) and can produce up to 0.1% strains, but the electrostrictive ceramic can generate a higher pressure, up to 1 MPa. Thus piezoelectric materials lack repeatability because of their high hysteresis and creep. Fig. 3 shows an electrostrictive actuator.

Electrostrictive materials have better characteristics because of a lead-magnesium-niobate (PMN) crystal, a class of ferroelectric material with superior properties for motion control applications. The PMN stack has a multi-layer configuration with very thin layers (125 to 250 μm) that are diffusion bonded during the manufacturing process. The net positive displacement is a superposition of the strain from the individual layers.

Piezoelectric materials, on the other hand consist of lead-zirconate-titanate (PZT) based ceramics. When using the PZT stack, electric currents are applied across two conductive plates separated by a poled ferroelectric material.

Thus, due to the fact that PMN materials are not poled, they are inherently more stable. It does not experience the long-term creep present in PZT materials. PMN materials also undergo lesser hysteresis (about 3% compared to 15% for PZT). As such they allow better repeatability since the original state is not lost. More detailed properties and characteristics of these materials are mentioned in [10].

2.3.3 Shape Memory Alloys

Shape memory alloys (SMAs) [4, 9] are a class of metallic materials that exhibit the ability to elastically deform over very large strains with a nearly infinite fatigue life. NITINOL-55, a nickel-titanium alloy, has an elastic strain capacity of over 6%, more than 30 times that of structural steel. In addition, these materials have very high internal damping. The stress-strain diagram of the axially loaded material is shown in Fig. 4.

The material is elastic up to the martensite inducing stress, undergoes molecular phase change to martensite, and then becomes elastic again up to ultimate load. Any martensitic transformation remains after the removal of stress until the temperature is raised above the transition temperature, at which the martensite reverts back to austenite, removing the "permanent" strain.

Devices incorporating these materials in dampers can be installed in new and existing buildings to modify structural behavior under strong lateral loads. Prototype devices have been developed [5] which produce a nearly square hysteresis loop, providing significant energy dissipation capabilities. These devices could be incorporated into existing braces or in braces added to moment frame structures. Tuning the device to the particular force, displacement, and hysteretic characteristics is achieved by altering the cross sectional area, length and configuration of the material or device. Unfortunately, the variable stiffness effect of a SMA is coupled with its stress/strain variations.

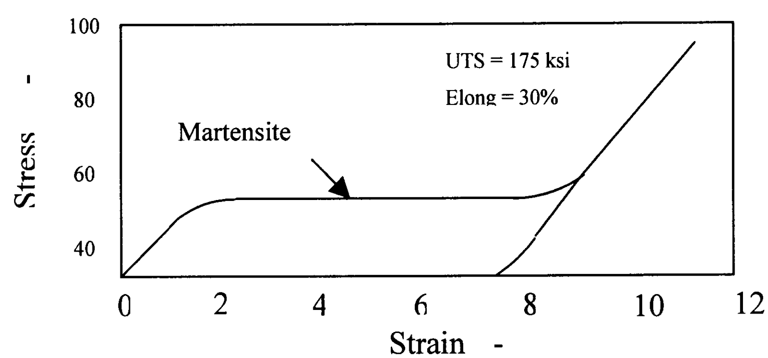


Fig. 4: Shape Memory Hysteresis.

Chapter 3

Intelligent Controls

Control theory has long been implemented using algorithms. In order to get a more flexible and practical model, other elements such as logic, sequencing, reasoning, learning and heuristics should be used. Thus, it follows that Artificial Intelligence (AI) technologies should be applied to the control designs. The most frequently used methodologies are knowledge-based systems, neural networks and fuzzy logic control.

Both control theory and AI technologies have been the center of interest for many researchers and scientists worldwide [3, 4, 5, 11, 12], since the potential benefits and innovations can be enormous. This has spawned the use of hybrid methods such as Neurocontrol and Neuro-Fuzzy Logic [3,12].

Controllers are often tuned using heuristic rules-of-thumb. In adaptive control, heuristics in the form of safety jackets often have codes that are significantly larger than that of the control algorithm. Such problems are ill structured and therefore do not provide good algorithmic solutions. Thus knowledge-based solutions should be employed.

3.1 Knowledge Based Systems

Knowledge based, or expert control systems can be used to extract and condense knowledge about control system design and operation in order to automate tasks normally performed by experienced system engineers (i.e. experts). An expert system is an interactive computer-based decision tool that uses both facts and heuristics to solve difficult decision problems, based on knowledge acquired from an expert.

Knowledge based controls, unlike conventional ones, do not require the programmer to specify how the program should achieve its objectives through an algorithm. For example, for a rule-based expert system, any of the rules can be selected for use if they are relevant to the problem at hand. Therefore, the program statement order does not have a rigid control flow. By doing this, the program becomes much more versatile and flexible compared to hard-wired logic and sequential programs. This is achieved through expressing heuristics as rules.

Another advantage will be that expert systems separate the numerical algorithms from the heuristics. This greatly facilitates the writing of programs since both parts can be kept separate. The heuristics can be rule-based, so that the contained knowledge can grow as more rules are added to the system. This continual refinement helps to check the correctness and performance of the system.

Various expert system shells and techniques have already been implemented [5, 11]. For example, NEXPERT Object is a commercial expert system shell that uses a combination of rules and objects for its knowledge representation. Gensym Corporation's G2 is an application development tool for building real-time knowledge based process control systems.

3.2 Neural Networks

An artificial neural network is a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain.

As can be seen in Fig. 5, artificial neural networks are arranged in layers: an input, output and middle layer. All the nodes in the network are interconnected and each of these connections is assigned a weight, which will be trained [12, 13]. These weights are the memory units of the neural network and the values of the weights represent the current state of knowledge of the network. As various sets of inputs are passed through the system, the weights are altered accordingly to improve its simulation and knowledge of the subject of study. Essentially, neural nets learn by examples. This is one of its setbacks since it will require some time for training before it can be used in the real system effectively. However, once it is trained to recognize the various states and conditions of a complex system, it will only require 1 cycle to detect or identify a specific state or condition.

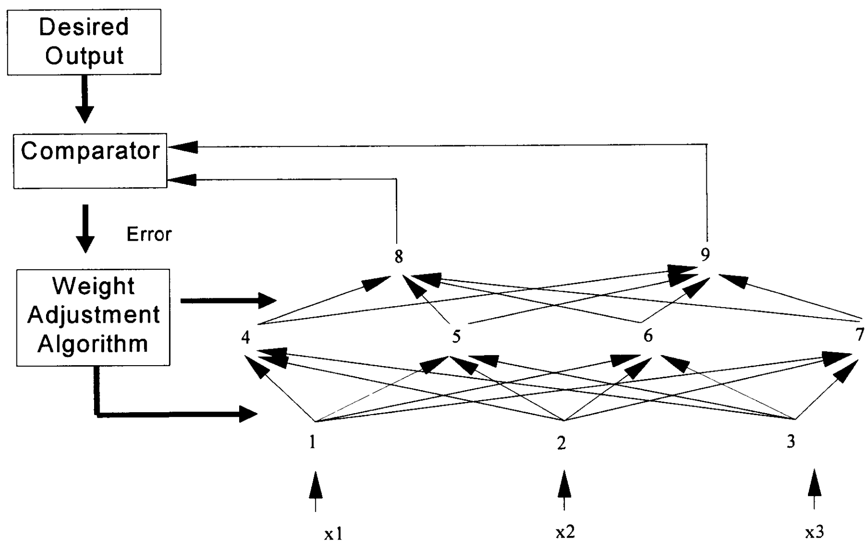


Fig. 5: Neural Networks.

Well-trained neural networks are robust because they are fault tolerant. Due to the fact that the information storage is distributed over all the weights, even if some of the connections are faulty, the overall system will not be drastically affected. For example, a typical neural network might make use of about 1000 weights, so if one or a few of these are not working properly, the percentage effect is negligible.

3.3 Fuzzy Logic

Fuzzy logic is another control technique that has great potential. It enables linguistic description of system variables. For example, the temperature in a room can be described as warm, cool or average as shown in Fig. 6. The universe of discourse represents the set of possible temperature values in terms of degrees Fahrenheit. The fuzzy values: warm, average and cool will then be mapped through different membership weights to the universe of discourse. In this way, each fuzzy value can be tailored to consist of a differentiated range of temperatures. This stage is called Fuzzification [13].

A set of if/then rules will then be applied to these fuzzy values. For example, a rule can be set such that if the temperature is warm and the humidity is high, the air conditioner will be switched on. Of course this is only a simple example. A full system consists of numerous fuzzy linguistic variables such as temperature, which in turn is made up of their own fuzzy values. The set of if/then rules will also be exhaustive, so as to cover every possible user demand. In this way, a flexible and well-designed HVAC control system can be set up. This stage is called Inference.

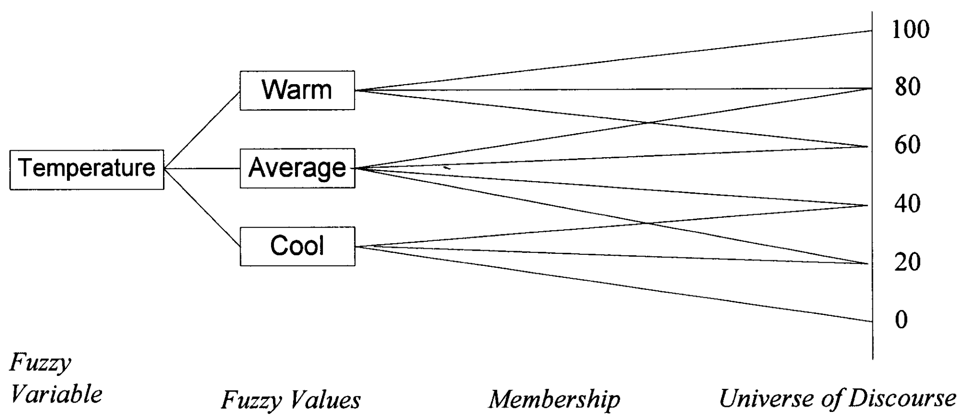


Fig. 6: Fuzzy linguistic values.

The remaining task will be to convert the output from the inferences to actual crisp values, which will then be sent as input to each of the localized temperature or humidity controls. This is Defuzzification.

The output actions produced will then be fed back to the system controller so that the parameter changes can be monitored or changed.

3.4 Neurocontrol

Neurocontrol, by the sense of the word, represents an incorporation of neural network technology into control design. It is a dynamic research field that has attracted considerable attention from the scientific and control engineering community in recent years. This section has been included here to emphasize the potential of combining the use of neural networks and classical control theory in order to bring about adaptive capabilities.

Among the possible applications of neural networks, such as classification, filtering and control, the latter is the most important. This is because applications such as pattern recognition and classification, or filtering, already have rather well established solutions and approximations. On the other hand, there are only a few universally applicable nonlinear control design approaches available. Therefore, neural networks can be put to great use in this field through its learning capability.

The biological roots of neural networks are responsible for the widespread use of the term *learning* to describe the process during which the network parameters are changed to improve the performance of the neural network based system. Another frequently used term, *training*, refers to instances when the parameter tuning is directed to deliberately selected situations. Both terms have a common mathematical equivalent: optimization. Since neurocontrollers are usually designed on digital computers with the help of numerical mathematics, it makes sense to exploit this mathematical connection as much as possible.

Fundamental control tasks are all functional approximation tasks. These approximations seek the optimal feedback law, the system model that is most consistent with measured data, and the best approximation of the strategic utility function. In addition to the optimization method, a space of parameterized functions is needed. In this functional space, instances are sought that represent the best solution of the functional approximation task, that is, the minimum of the fundamental cost function.

Chapter 4

Control Theory and Concepts

4.1 Control Design and Modeling

The first step in the design of a control is to establish a working model. A thorough understanding of the system will have to be achieved in order to express its characteristics as a set of mathematical equations [2]. The model will be based on these equations, which map the inputs u(t) to the outputs y(t). In variable stiffness control for dynamic behavior of a building, the inputs will either be the seismic or wind loads, and the outputs will be the appropriate changes to the stiffness and natural frequency of the structure. The goal then is to make the system produce as exact an output as possible. In this case, it will be the optimal response to a given excitation. This will require detailed knowledge of the building's parameters. However, due to the complexity of physical systems, such as the one mentioned above, an exact output will often not be possible. Simplifications will have to be introduced. For example,

a) Linearization around operating points, whereby Taylor's series expansion and approximation or fitting of experimental data to linear models etc., are used to approximate the model around the area of concern.

b) Model order reduction techniques, whereby small effects and phenomena outside the frequency range of interest are neglected to simplify the model. For example, in dynamic studies, only the first 2 or 3 natural frequencies are usually considered.

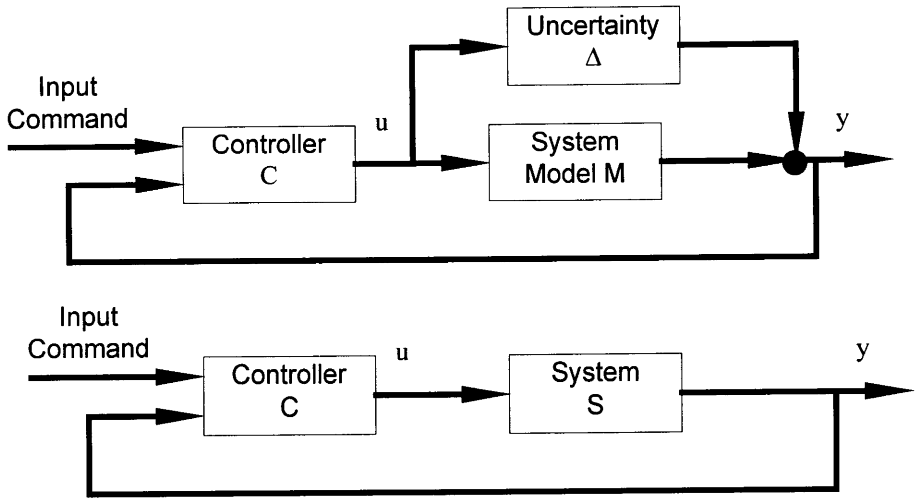


Fig. 7: Model design.

Once the model is obtained, the controller can then be designed. However, since the model is only an approximation of the actual system, the effect of any discrepancy between the system and the model on the performance of the controller will not be known until the controller is actually applied to the system. Fortunately, there is a way to circumvent this. That is to introduce a class of system model uncertainties denoted by Δ (as shown in Fig. 7) that might probably appear in the system. Since Δ is an uncertainty, it cannot be represented by explicit equations. Its characterization will then have to be in terms of some known bounds. This will enable the engineer to better predict the behavior of the model in response to such uncertainties and therefore alter the model accordingly to make it more robust with respect to Δ. That is, to make it less sensitive to Δ.

During the design, consideration will also have to be given to the available type of computer, the type of interface devices between the computer and the system, processing power and accuracy limitations. If not properly considered, the model will not be practical even though it approximates the actual system very well. This is very important especially because most adaptive algorithms are computationally intensive.

Finally, once the controller has been designed and tested to be sufficiently robust, implementation will follow. The model will then be fine-tuned according to the monitored performance of the controller in order to achieve a better characterization. This tuning is often done through trial and error, which is very tedious. Adaptive techniques should be utilized to enable the controller to "learn" about the system and thus conduct self-tuning.

4.2 Use of Controls in Structural Systems

Conventionally, structural variables are kept independent of the control variables. The design is usually carried out for the structure first, with constraints on the allowable stresses, displacements at the nodes, and natural frequencies etc. Once that is done, the control variables can then be designed with constraints on closed-loop eigenvalues, control effort, reliability, and sensor and actuator locations etc. However, this sequential method does not take into consideration the dynamic interaction between the control and structural variables and therefore, does not produce an optimal solution to the problem set up.

Experimental results have shown that slight structural changes can improve the control system considerably. Therefore, it will only make sense to integrate both aspects of the design together [22]. There had been considerable effort in recent years towards this end, and cross handling and coupling effects have been better understood.

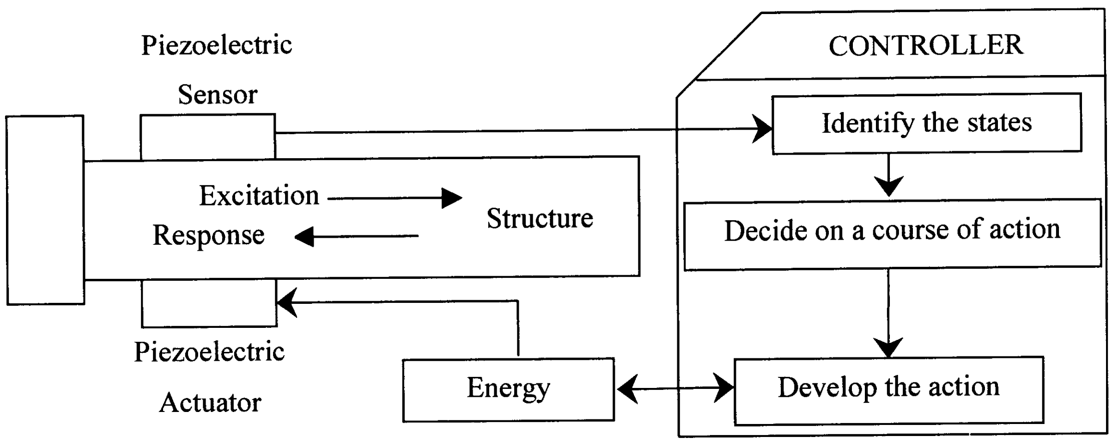


Fig. 8 *: Schematic of an active/adaptive control system.*

Designing adaptive controls and structures involve knowledge of a number of fields, namely, electronic and electrical engineering, control science, materials, structural dynamics and deformable solids. Due to this inter-disciplinary nature, adaptive systems are not easy to design.

In general, an adaptive structure has the ability to produce an active response to an external stimulus (Fig. 8). The idea is to integrate transducer materials into the sensors and actuators. So, once a stimulus is sensed, the signal output from the sensors will be sent through the use of feedback control laws to electrical devices. These devices then process the signals and feed them back to the actuators, which will in turn induce loads in the structure according to a prescribed design criterion.

The detailed design of active or adaptive controls for structures requires an accurate model of both the actual structure and the control-structure interaction. This is most often done with Finite Element Methods (FEM), which allow handling of even complex geometries.

Below are the dynamic equations for a building structure based on finite element formulations:

$$[M]\{\ddot{x}(t)\}+[C]\{\dot{x}(t)\}+[K]\{x(t)\}=[b]\{U(t)\}+\{f(t)\} \tag{2}$$

where *[M]* is the system mass matrix, *[C]* is the structural (passive) viscous damping matrix, *[K]* is the structural stiffness matrix, *{f(t)}* is the vector of nodal external forces, *[b]* is the matrix that places the control forces (actuator forces) at nodal degrees of freedom, *{U(t)}* is the vector of actuator forces, and *{x(t)}* is the vector of nodal displacements and rotations. The number of degrees of freedom can be denoted by n and the number of actuators denoted by n_a. Therefore, *[M], [C]* and *[K]* are n x n matrices, [b] is an n x n_a matrix, and *{U(t)}* is a vector of dimension n_a.

The system mass matrix *[M]* includes the masses of the sensors and actuators, as well as the building itself. The damping matrix *[C]* inherent to the structure is usually assumed to be either proportional to the stiffness or completely ignored.

Using state space formulations, the above equation can be transformed to 2 first order equations:

$$[M]\{\dot{X}_1(t)\}+[C]\{\dot{X}_2(t)\}+[K]\{X_2(t)\}=[b]\{U(t)\}+\{f(t)\} \tag{3}$$

$$[M]\{\dot{X}_2(t)\}=[M]\{X_1(t)\} \tag{4}$$

where

$$\{X(t)\} = \begin{Bmatrix}\{X_1(t)\}\\ \{X_2(t)\}\end{Bmatrix} = \begin{Bmatrix}\{\dot{x}(t)\}\\ \{x(t)\}\end{Bmatrix} \tag{5}$$

In classical control theory, equation (3) is written as

$$\{\dot{X}(t)\}-[A]\{X(t)\}+[B]\{U(t)\}+\{F(t)\} \tag{6}$$

where

$$[A] = \begin{bmatrix} -[M]^{-1}[C] & -[M]^{-1}[K] \\ [I] & [0] \end{bmatrix} \tag{7a}$$

$$[B] = \begin{bmatrix} [M]^{-1}[b] \\ [0] \end{bmatrix} \tag{7b}$$

$$\{F(t)\} = \begin{Bmatrix} [M]^{-1}\{f(t)\} \\ \{0\} \end{Bmatrix} \tag{7c}$$

Equation (3) has the advantage that the *[M]* and *[K]* matrices are explicitly known, so they provide a better basis for constructing approximations, which will be used extensively in the optimization process. Equation (7) will be more suitable for analysis of control theory concepts like observability and controllability.

Sensors

Sensors enable the control system to pick up direct or indirect measurements of displacements and velocities at the given degrees of freedom. They can be expressed in state space form as follows:

$$\{Y(t)\} = [Cc]\{X(t)\} \tag{8}$$

where

$$[Cc] = \begin{bmatrix} [0] & [Cp] \\ [Cv] & [0] \end{bmatrix} \tag{9}$$

Y(t) is the output of the sensors, and *[Cp]* and *[Cv]* are the matrices expressing the locations of the displacement and velocity sensors respectively. The number of velocity sensors can be denoted by n_v, the number of displacement sensors by n_p and the total number of sensors by n_s.

Model Reduction

Model reduction can be performed by using modal superposition, and then retain only the first few important modes. Therefore,

$$\{x(t)\} = [\phi]\{q(t)\} \tag{10}$$

$$\ddot{q}_i + 2\xi_i\omega_i\dot{q}_i + \omega_i^2 q_i = \{\phi\}_i^T[b]\{U\} + \{\phi\}_i^T\{f\} \tag{11}$$

where ω_i , $i=1, \ldots\ldots N_R$ are the natural frequencies, and ξ_i , $i=1, \ldots\ldots N_R$ are the modal damping ratios, and N_R is the number of modes that are retained in the reduced model. The detailed derivations can be found in [15, 16].

A model reduction is useful because it reduces the dimensions of the problem and makes it more manageable. However, only the retained modes can be controlled.

4.3 Feedback

Feedback is essential in a control system. Feedback loops occur whenever part of an output of some system is connected back into one of its inputs (Fig. 9). Depending on whether the connection is such as to add the output to an input ("positive" feedback) or such as to subtract the output from some input ("negative" feedback), the whole system will behave entirely differently.

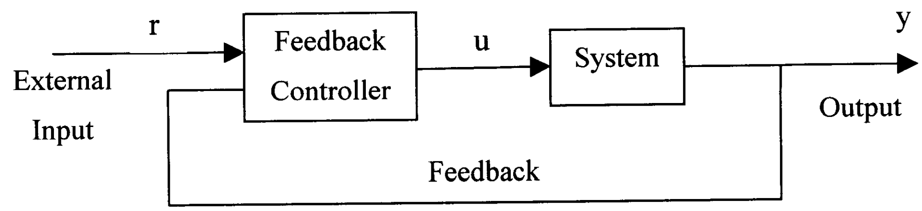


***Fig. 9** : Simple Feedback System.*

Feedback needs to be properly designed and used. Otherwise, it may create more problems than it solves. Feedback can be used for a variety of purposes. Stabilizing an unstable system is one of its prime uses. Without proper feedback, many open loop systems will be rendered useless, given the large amount of uncertainties in the initial conditions and the inaccuracies in the model. The classic example for such a use is an initially unstable system. The force required to restore equilibrium to an initial perturbation to the system can be calculated using Newton's laws and Lyapunov's first law. The details of this usage can be found in [17].

Another application of feedback is to reduce the sensitivity of systems to parameters, noise, and nonlinear distortion. Usually, a high loop gain will lower the sensitivity of a system to load disturbances and nonlinear perturbations. However, if the noise is substantial, a high loop gain tends to make the noise transmitted to the output high. Therefore, a compromise between the choice of gain and frequency characteristics of the loop transmission develops. Feedback can be used to achieve a good design, especially for a multivariable system.

Feedback can also be used for optimization. A functional called a performance index or cost functional is usually minimized with respect to control parameters. This is further illustrated in Chapter 4.5.

4.4 Feedback Control

There are two basic approaches to feedback control for linear systems. The first and simplest is direct output feedback in which the vector of control forces is considered proportional to the measured outputs :

$$\{U(t)\} = -[Hp]\{Yp(t)\} - [Hv]\{Yv(t)\} \quad (12)$$

where [Hp] is the displacement gain matrix with dimensions n_a x n_p, and [Hv] is the velocity gain matrix with dimensions n_a x n_v. If a sensor and actuator pair is at the same degree of freedom in the structure, the controller is collocated. If the sensor and the actuator are at different degrees of freedom, then the controller is non-collocated.

Two types of output feedback controller models are used. The first is the axial controller, which is modeled as a truss-spring element. The axial controller is always collocated by definition. The other type of control element is the general controller. This type can be either collocated or non-collocated.

The second type of feedback control is full state feedback :

$$\{U(t)\} = -[H]\{X(t)\} \quad (13)$$

where $\{X(t)\}$ is the state vector.

Clearly, this second law incorporates all the information for the system and therefore enables more design flexibility. This equation can be further expanded into :

$$\{U(t)\} = -[Hp]\{x(t)\} - [Hv]\{\{\dot{x}(t)\} \quad (14)$$

where *[Hp]* and *[Hv]* represent the position and velocity gain matrices respectively. Putting this into Equation (1), we get :

$$[M]\{\ddot{x}\}+[C+bHv]\{\dot{x}\}+[K+bHp]\{x\}=\{f\} \tag{15}$$

where $[C + bHv] = [C_A]$ = augmented damping matrix,

$[K + bH_p] = [K_A]$ = augmented stiffness matrix.

From the control point of view, the aim is to design the matrices $[H_p]$ and $[H_v]$ within certain limits. The three most common design approaches are: (I) pole placement, in which the gains are determined such that a pre-assigned stability of the system is achieved, (II) optimal control, which is mostly used for state feedback, and (III) direct optimization, where the gains are used directly as design variables.

The above formulation can be used for both the full model and the reduced model. It must also be mentioned here that stability, observability and controllability issues must also be considered in the design process. However, these issues will not be discussed here. Further reference can be found from [15].

4.5 Optimal Control

Optimal control theory can be used to design the controller based on the linear quadratic regulator. The performance index is :

$$J=\int_0^\infty(\{X(t)\}^T[Q]\{X(t)\}+\{U(t)\}^T[R]\{U(t)\})dt \tag{16}$$

where [Q] and [R] are state and control weighting matrices. The matrix [Q] is considered to be positive semi-definite and [R] is assumed to be positive definite. The performance index J represents a combination of the state vector norm and the control effort.

Thus, the selection of the elements of [Q] and [R] determines the closed-loop damping, which is directly related to the time required to control the disturbances and the energy required by the controllers.

The optimal solution for {U(t)} is obtained by minimizing the performance index J subject to the system dynamics (Eq XX). Assuming a state feedback control law of the form

$$\{U(t)\} = -[H]\{X(t)\} \tag{17}$$

the optimal gain matrix [H] is given by

$$[H] = [R]^{-1}[B_C]^T[P] \tag{18}$$

where matrix [P] satisfies the following non-linear algebraic matrix equation, called the Riccati equation :

$$[P]^T[A] + [A]^T[P] - [P][B_C][R]^{-1}[B_C]^T[P] + [Q] = [0] \tag{19}$$

The closed loop system (with no external disturbances) is then given by

$$\{\dot{X}\} = [A_C]\{X\} \tag{20}$$

where

$$[A_C] = [A] - [B_C][R]^{-1}[B_C]^T[P] \tag{21}$$

If the system is controllable and [R] is positive definite, then there is a unique solution [P] for the Riccati equation, this solution is positive definite, and the closed loop system is stable. The assumption that the system is controllable, allows the system to be stabilized to any degree. In other words, one can find a feedback such that the eigenvalues of the closed loop system can be made to have real parts less than or equal to an arbitrary negative constant. For this purpose, the Riccati equation can be modified thus:

$$[P_\alpha]^T([A]+\alpha[I])+([A]+\alpha[I])^T[P_\alpha]-[P_\alpha]^T[B_C][R]^{-1}[B_C]^T[P_\alpha]+[Q]=[0] \qquad (22)$$

where $a >= 0$ and the closed loop matrix $[A_C]$ has eigenvalues $(\sigma + j\omega)$ such that $\sigma <= -\alpha$.

By solving the Riccati equations for the feedback gains, the optimality conditions for the control subproblem are automatically satisfied and the gains become implicit functions of the structural variables (Eq. 22).

4.6 MIT Rule for Adaptive Controls

The MIT rule was introduced in the early fifties [1]. It refers to the combination of model reference control together with a gradient-type parameter update law. This can be best illustrated by an example, in this case, a feedforward problem.

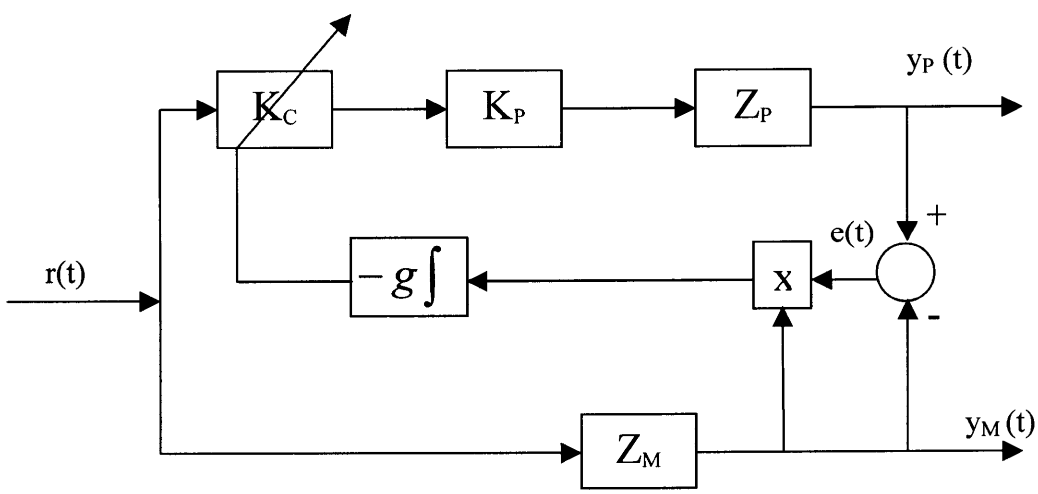


***Fig. 10** : MIT rule, feedforward case.*

Refer to Fig. 10, which shows a plant characterized by a time-invariant linear stable system with transfer function $Z_p(\xi)$ in cascade with an unknown scalar gain $K_p(t)$ of known sign. Without loss of generality, assume that $K_p>0$ and that $Z_p(0)=1$. The reference model has transfer function $Z_m(s)$, $Z_m(0)=1$. The goal is to compensate the plant gain by a scalar pre-compensator $K_c(t)$ so that $K_c(t)K_p(t)=1$. The control objective is to have the plant output y_p track the model output y_m. In other words, to have Z_m as close to Z_p as possible. A mismatch $K_c(t)K_p(t)=1$ is identified via the error between the plant output $y_p(t)$ and the model output $y_m(t)$. An external signal r, assumed to be piecewise continuous, drives both the plant and the model.

In particular, K_c needs to be adjusted online so as to minimize the mean square error : $\frac{1}{T}\int_0^T e^2(t)dt$ where $e(t) = y_p(t) - y_m(t)$. Due to the fact that K_c and K_p are time varying, this minimization can only be done approximately. The MIT rule uses a gradient approximating approach. Instead of minimizing the entire mean square error, one only updates K_c, to minimize the present error :

$$\dot{K}_c \approx -\frac{\delta}{\delta K_c} e^2(t) \tag{23}$$

The approximation sign is used to indicate that it is impossible to implement the right hand side. Since y_m is independent of K_c, we get :

$$\dot{K}_c \approx -e(t)\frac{\delta}{\delta K_c} y_p(t) \tag{24}$$

Assuming that K_c is constant, which is only approximately true, the partial derivative can be evaluated as :

$$\dot{K}_c \approx -e(t)[Z_p K_p r](t) = -e(t)[Tr](t) \tag{25}$$

[Tr](t) can be interpreted as the output of the operator T driven by the input r, at time t. However, the update in Equation (25) cannot be implemented since it requires full knowledge of the plant gain K_p and the plant transfer function Z_p. Assuming K_p to be nearly constant,

$$[Z_p K_p r](t) \approx K_p(t)[Z_p r](t) \tag{26}$$

To implement a gradient descent, the scaling due to K_p is largely irrelevant; knowledge of its sign will be enough. Hence, provided one ignores the K_p dependence and accepts that K_p is only slowly time varying and positive at all time, and, because Z_m is supposed to be a good model for Z_p, one implements :

$$\dot{K}_c \approx -ge(t)y_m(t) \tag{27}$$

The scalar gain g determines the adaptation speed. Due to the assumptions made, it is reasonable to set g to be a small positive constant. The update in equation (27) is known as the MIT rule for adaptive control.

The overall system is linear time varying. However, it has properties that depend nonlinearly on the reference input signal. This non-linear dependence is characteristic for all adaptive systems. Thus, full analysis is very difficult. The closed loop can be represented in state space form as such :

$$\dot{x}_p = A_p x_p + b_p K_p(t) r(t) K_c \tag{28}$$

$$\dot{x}_m = A_m x_m + b_m r(t) \tag{29}$$

$$\dot{K}_c \approx -g c_m x_m(t)(c_p x_p(t) - c_m x_m(t)) \tag{30}$$

The transfer function Z_p is realized as $Z_p(\xi) = c_p(\xi I - A_p)^{-1} b_p$. The triple (c_p, A_p, b_p) represents a minimal realization for the plant transfer function. The same applies for the model, $Z_m(\xi) = c_m(\xi I - A_m)^{-1} b_m$, (c_m, A_m, b_m) is a minimal realization for the model transfer function. The variable x_p is the plant state, and x_m is the model state. The inputs are r and the plant gain K_p.

The interesting properties of adaptive systems can be divided into 2 general categories : ideal behavior and robustness properties. Under ideal behavior, the system response is known since the plant model is identical to the actual plant, and the system is only excited by the reference signal. The robustness properties refer to the changes in the behavior when the adaptive system is operated under conditions perturbed from the ideal, for example, in the presence of plant-model mismatch and disturbance signals.

Chapter 5

Variable Stiffness Control

The field of active variable stiffness controls is relatively new. Up to date, much of it is found only in research work. There are very few actual cases where it has been implemented. Most of these are for applications in aircraft engineering in order to control the stiffness of the wings in response to the loading conditions. On the Civil Engineering side, Kajima Corporation Incorporated has implemented an active variable stiffness (AVS) system in one of its research institutes (Fig. 11). This is one of the first uses of variable stiffness controls in buildings.

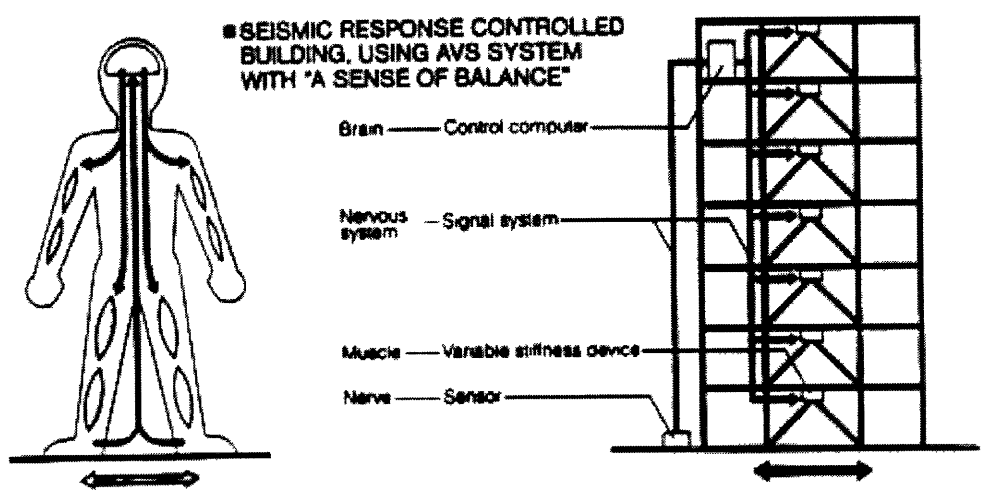


Fig. 11 *: Active Variable Stiffness Control, Kajima Corporation.*

The (AVS) system utilizes a combination of braces in an on/off state to alter the stiffness of the structure. This is a non-resonant control system that continuously monitors the external excitation due to an earthquake and alters the stiffness of the building to avoid a resonant state [16]. This is a form of active control. A measurement and control device, consisting of an accelerometer, is placed on the first floor of the building. This device feeds the earthquake input into a motion analyzer. The analyzer consists of several special band-pass filters, which approximate the response transfer characteristics to each stiffness type, i.e. with the braces locked or unlocked or some other combinations [16]. The number of stiffness types in the AVS system to be used in the building design will be predetermined by the designer. In this way, different stiffness modes can be adopted in response to the excitation.

This can be taken a step further by introducing some adaptive techniques. Any model that is created after a complex system, say a building, will always have errors in its design. The assumed loading, behavior or excitation frequencies will not be exactly the same as that of the actual system. In many situations, these actual parameters also vary with time or are non-linear. Therefore, adaptive control techniques will have to be used to alter the model even after it has been implemented into the real system. As the control handles different sets of data, it will gradually learn how the parameters change or need to be changed in order to achieve an accurate estimate of the actual behavior and thus apply the right solution technique.

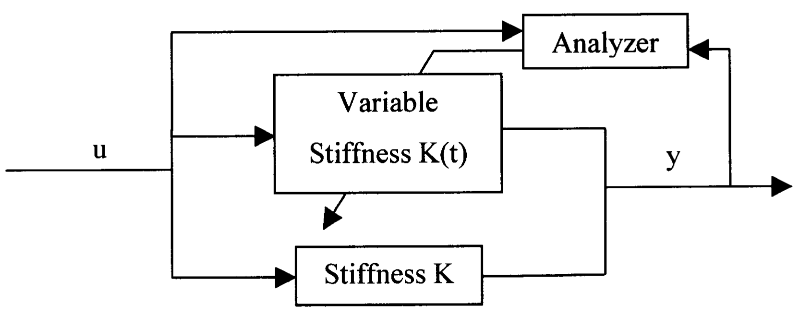


***Fig. 12** : Variable stiffness control.*

For example, earthquake data from pre-tremors, or seismic data collection centers will enable the control system to alter the process model after the preliminary diagnosis of the earthquake, as shown in Fig. 12. Once the earthquake hits, the system will quickly change its stiffness accordingly, to avoid resonant behavior, while at the same time gather data continuously from the output of the process, and feed it back to an estimator or analyzer. The analyzer will then determine the appropriate parameter change to introduce into the model and controller (i.e. the stiffness type K(t)) so that eventually, the error of the system can be minimized. This will lead to accurate responses to wind or earthquake excitation.

Chapter 6

Simulation

In order to achieve a better understanding of the functions and uses of adaptive controls in buildings, some programs have been developed in MATLAB [21]. There are 3 programs, one dealing with optimizing the sum of the stiffness in a building, one demonstrating the training of a neural network, and the last one showing how a neural net adapts and predicts an input signal.

6.1 Stiffness Optimization

Optimized response and behavior is the best and most efficient type of response. In the use of adaptive controls, optimization techniques are usually used to achieve the best-fit solution, which the control will use to adapt the system. Adaptiveness means that the state parameters of the system are changed in order to effect the control. Most of the work is done either in the optimization algorithm or the neural networks.

Building structures excite differently under different types of loads. Thus, an adaptive control can be aptly implemented here so as to avoid excessive design for extreme conditions. In this way, structural redundancy can be prevented since unusual loads and more extreme loads can be handled by the adaptive control. When there is a change in the external loading conditions, new optimized values for the stiffness or damping of the structure will be calculated by the adaptive controller. These values will then be used by the controller, either to achieve equilibrium or to avoid resonance (see Chapter 5).

Once a building has been built, it will be exposed to wind and earthquake loads. Each type of load will operate over a range of frequencies, around 0.2 to 20 Hz for earthquakes and around 0.1 Hz for wind. Each of these will excite the few modes with natural frequencies within that range. Therefore, the response of the structure within that range will have to be designed for.

In the analysis performed, a 3-story building with an assumed mass was used. The stiffness can be changed for each analysis and the damping was assumed to be 2% proportional to the stiffness.

Next, the building was subjected to an earthquake load. A famous, well-studied set of earthquake data was used (Fig. 13). This contains the north-south component of the ground motion measured at a site in El Centro, California during the Imperial Valley, California earthquake of May 18, 1940 [18].

An acceleration response spectrum is shown in Fig. 14. This shows the response of the structure to the El Centro earthquake. Shown in Fig. 15 is the elastic pseudo-acceleration design spectrum within which the structural design will be safe for any moderate design earthquakes. The elastic design spectrum provides a basis for calculating the design force and deformation, for the system to remain elastic.

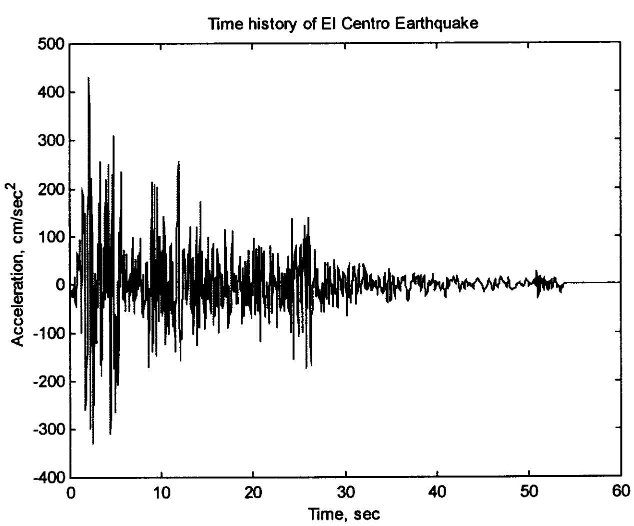


Fig. 13: El Centro Earthquake.

Usually, for a particular building, a number of earthquakes are simulated and the resulting response spectra plotted. Each of these will be similar to the spectrum shown in Fig. 14, except that they will have different peaks at different frequencies. Different values of stiffness and damping will affect the response. By plotting a number of these spectra together, average values, of the spectral velocity, for example, can be obtained over a range of frequencies. This ensemble average can then be used as the seismic design spectrum for the building, assuming a certain damping ratio. Fig. 16 shows a comparison of a standard design spectrum with an elastic response spectrum for El Centro ground motion.

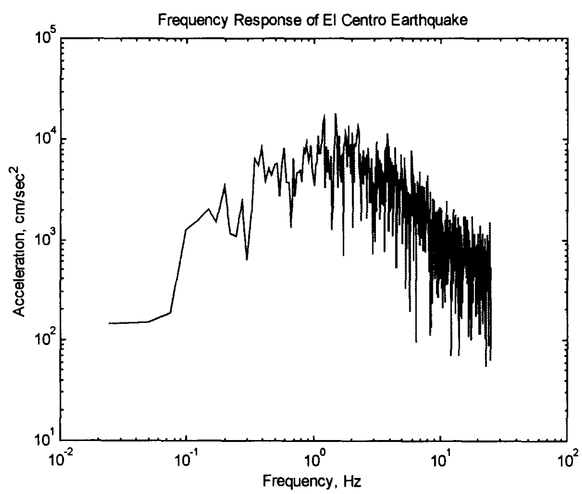


Fig. 14: El Centro Earthquake Response Spectrum.

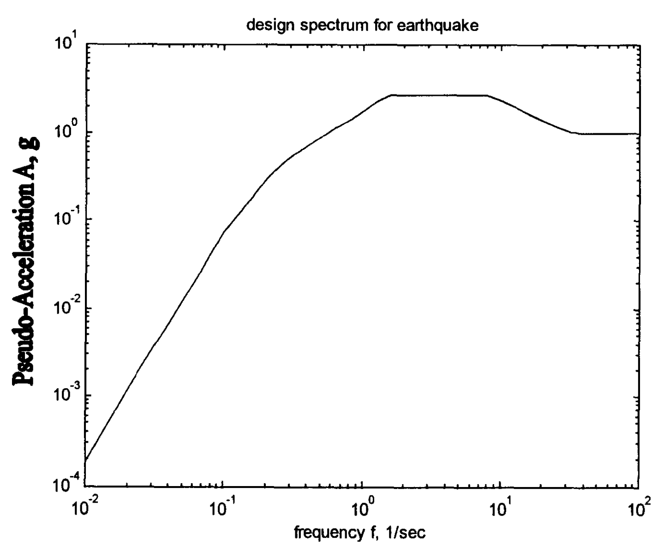


Fig. 15: Elastic pseudo-acceleration design spectrum.

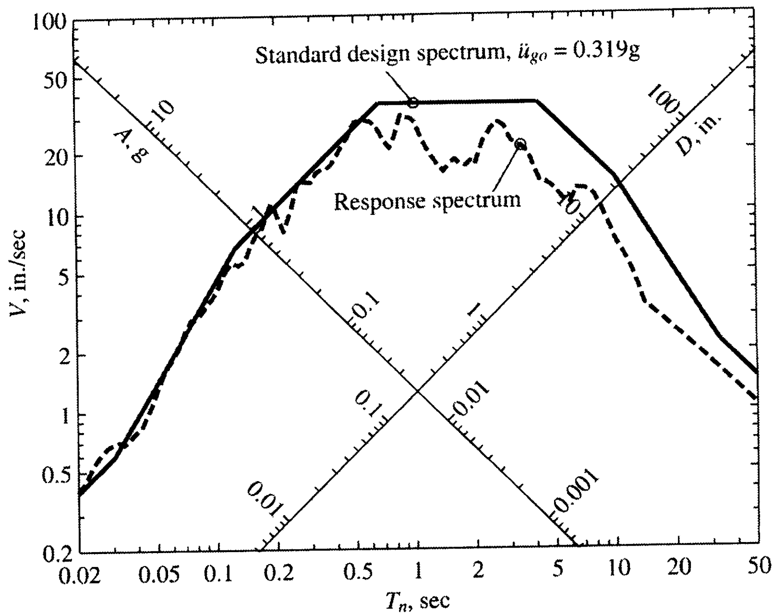


Fig. 16: Comparison of Standard Design Spectrum with Elastic Response Spectrum for El Centro ground motion; $\xi = 5\%$.

A state space formulation is used to obtain the transfer function of the structure:

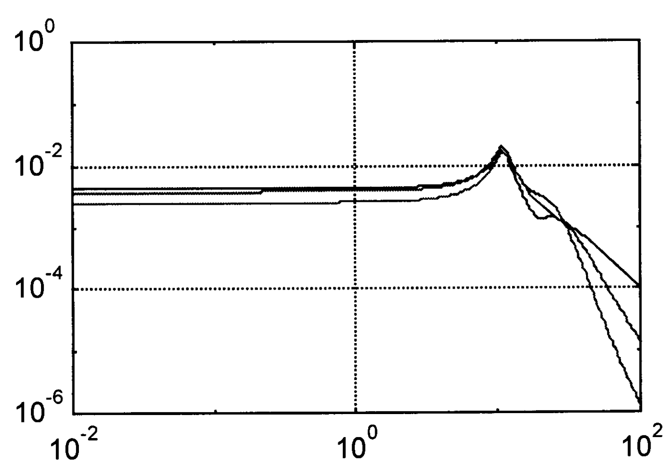
$$\dot{x} = Ax + Bu_g \qquad (31)$$
$$y = Cx$$

Analysis in the time domain can be done using the Duhamel integral, by superimposing the response to each short-duration impulse. It can also be done in the frequency domain by superimposing the responses to the harmonic components of the loading. However, both methods are not suited for nonlinear analyses due to the use of superposition. Therefore, step-by-step methods like the piecewise exact method, Euler-Gauss procedure, Newmark's method, or the linear acceleration method, are used instead [19]. These basically treat each time-step as an independent analysis problem. They are time domain approaches and can be used to solve linear problems as well.

It was decided to carry out a state-space to transfer function conversion. The transfer function was then used to conduct the analyses in the frequency domain. Time domain analyses are usually not used for simple structures since they require more number crunching. In situations when the equation of motion contains parameters that are frequency dependent, such as stiffness k or damping c, the frequency domain approach is more superior than the time domain approach.

The difference between the displacements at each story, that is, the inter-story displacements, are then plotted (Fig. 17). All the 3 plots contain peaks around the same frequency level.

High-rise buildings have low natural frequencies. Therefore, the optimal design for this case will be the lowest realistic value of stiffness, since only the seismic design spectrum has to be satisfied. To satisfy the spectrum means that for a certain value of stiffness and damping, the response of the building to the particular earthquake falls below the design spectrum. However, if wind loads are considered at the same time, this will not be the case. A compromise will have to be achieved. This is because wind load spectra are almost opposite to earthquake spectra. At low frequencies, the earthquake spectrum decreases with the frequency, while the wind spectrum increases. Therefore, an optimal solution for the stiffness has to be obtained while satisfying both earthquake and wind spectra.



Initial stiffness design : [150 100 50]; Criterion : 0.0195

Minimum stiffness : [100 75 50]

Fig. 17: Inter-story displacements.

The maximum natural frequencies for low-rise buildings, on the other hand, are usually higher, at around 3 Hz. Therefore, the optimal stiffness design is earthquake dominated in this case. Hence, the best stiffness to use for the design will be the most economical and realistic value.

A note needs to be mentioned here. Lowering the stiffness of the structure will cause the maximum displacements for each of the modes to occur at a lower frequency. The magnitude of the response will also increase at low frequencies. Therefore, higher values of damping might have to be used.

The attempt to include the wind spectrum was not very successful, due to a lack of accurate data. Hence, this was not pursued further. However, the concept and possibility of this method is clear. With accurate data for the wind spectrum, an optimized solution can be obtained with the consideration of both the earthquake and wind spectrums. This result will be interesting since a stiffness design that meets the criteria for earthquake loads might not coincide with the stiffness design for wind loads.

Nonlinear optimization with constraints was performed using the MATLAB function (Appendix I), *constr('obj3', [k1 k2 k3])*. Only the earthquake spectrum was used. The design values, objective functions and the constraints used in the analysis are all defined in the function, *obj3(k)*. The objective function is :

$$\min \sum k_i \tag{32}$$

where i = 1, 2, ... n represents the stories in the building. The following constraints are used:

$$\begin{aligned} &|x_i - x_{i-1}| = \delta_i \\ &\delta_i(\omega(k))\big|_{\max} \le \delta^* \\ &k_i \ge k^* \end{aligned} \tag{33}$$

where x_i represents the displacements of the stories, δ_i is the inter-story displacement, and k_i is the story stiffness values. The inter-story displacements are analyzed in the frequency domain. The maximum displacement can be identified by locating the peak in the transfer function, and finding the corresponding frequency at which it occurs. Eventually, this can be converted back to the time domain to obtain a time history response.

Various initial values of the stiffness were tried to observe the effect of the optimization. 2 approaches were taken. The design criterion for the frequency response of the inter-story displacements was fixed. Analyses were then performed to find an optimized sum of the stiffness for each floor of the structure, which is lower than the initial sum. With a limiting displacement of 0.0195, and an initial stiffness of [150 100 50], a lower value of the sum of the stiffness was found [134.6 101.4 50].

The other approach taken was to keep the sum of the stiffness unchanged. Various values of limiting displacement were then tested to find whether it is possible to satisfy a more stringent criterion with the same total value of stiffness. It was found that with a total stiffness value of 300 ([150 100 50]), and an initial criterion of 0.0195, a lower criterion of 0.0181 could be achieved. The optimized stiffness values obtained for the 3 stories are [141.37 107.7 50].

Achieving an optimal stiffness design is very desirable since this leads to savings in building costs. Optimal stiffness design has great potential for use in structures due to the increasing complexities of designs today. It can also be further expanded to achieve a more powerful procedure, i.e. with the inclusion of damping optimization. Such optimization procedures are required in order for adaptive control techniques to work well.

6.2 Learning of a Neural Network

The next program demonstrates the training of a neural network (Appendix II). The idea is based on backpropagation. Input vectors and the corresponding output vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way defined by the user. This is very similar to a famous example of the learning of T and J signal shifts demonstrated by Bernard Widrow in a 1960's television program [20].

Here the Elman Network (Fig. 18) is used, which is an example of a recurrent network. The Elman network is a two-layer network with feedback in the first layer. This recurrent connection allows the Elman network to both detect and generate time-varying patterns.

The Elman network has neurons that use a tangent-sigmoid transfer function (Fig. 19) in its hidden (recurrent) layer, and neurons that use a linear transfer function in its output layer. These transfer functions are both backpropagation functions. This combination allows the network to approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons to handle the required level of complexity.

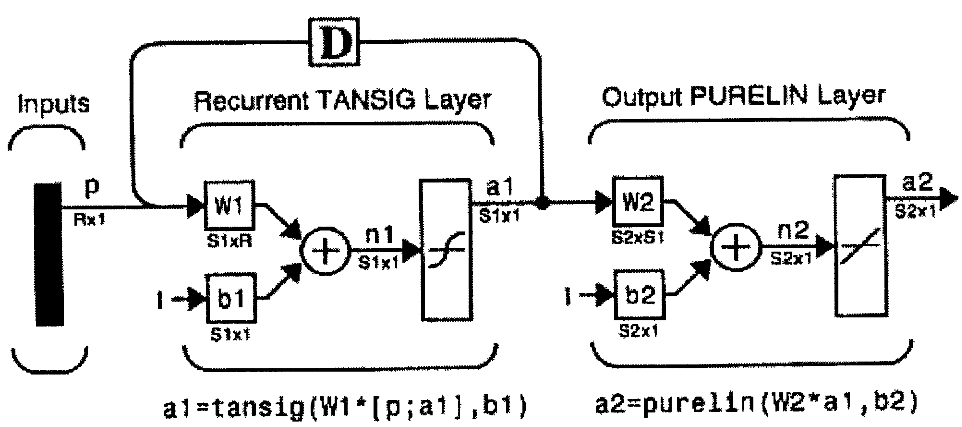


Fig. 18 : The Elman Network.

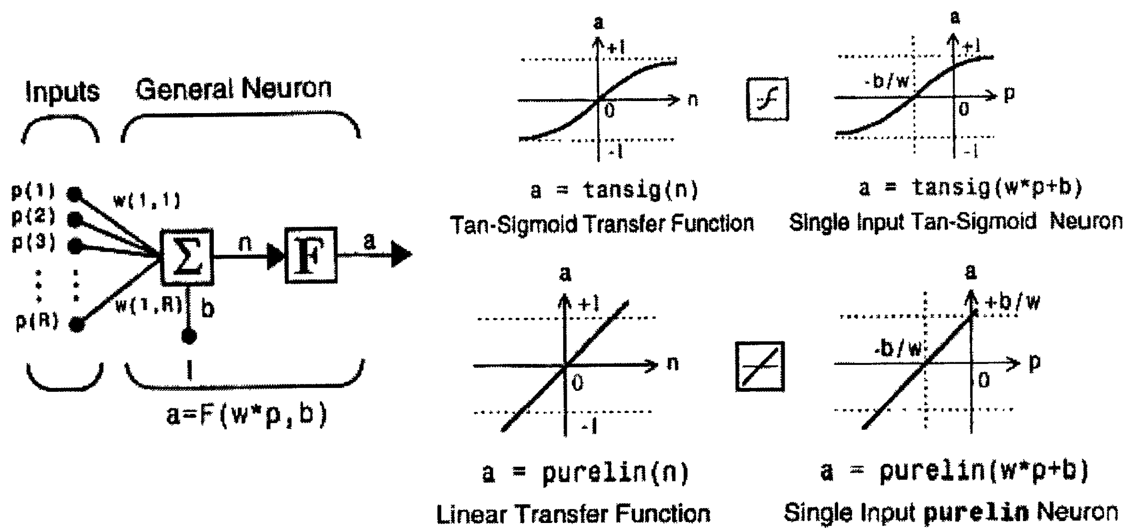


Fig. 19 : 2 Types of Neurons used in MATLAB.

There are a number of ways to improve the learning of a neural network. Here, 3 methods are shown: increasing the number of recurrent neurons, increasing the training time, and carrying out a number of training sets.

Using a larger number of neurons can enable the neural network to handle complex functions. Therefore, as can be seen from Fig. 20 to Fig. 21, by increasing the number of neurons from 10 to 100, the network was able to approximate the function more accurately. However, care must be taken not to overdo this, since (in Fig. 22), with 200 neurons, there is a deterioration of the approximation. Therefore the right amount of neurons to be used should be determined by experimentation. Using more neurons also increases the training time, which is undesirable.

The training time can also be increased (Fig. 23). Longer training times allow the network to achieve a better fit over the function that it is being trained. However, there is not an obvious benefit from the approximation of other functions, after this initial training.

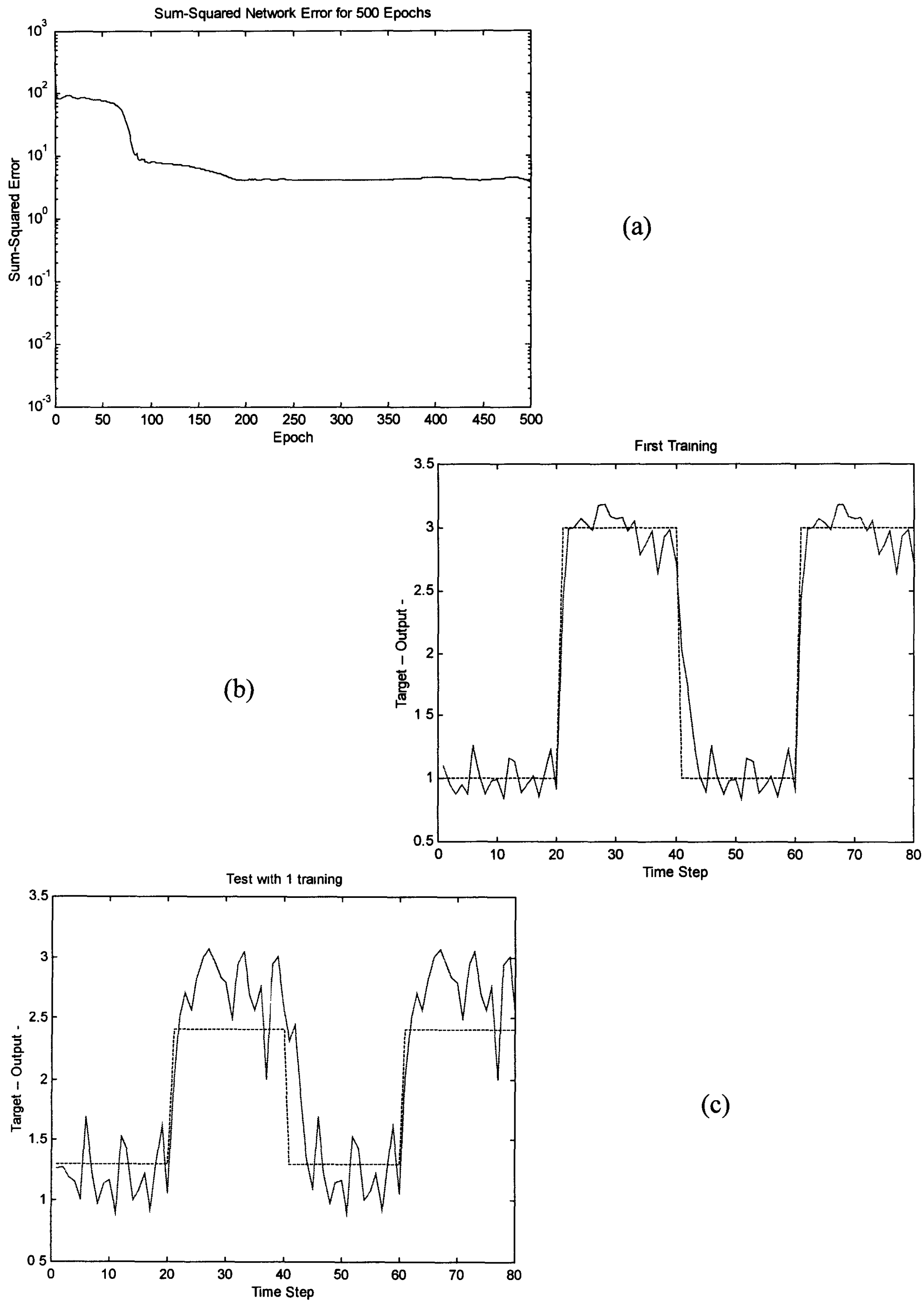


Fig. 20 : Normal Training of a Neural Network using 10 recurrent neurons, 500 epochs, and 1 training input set.

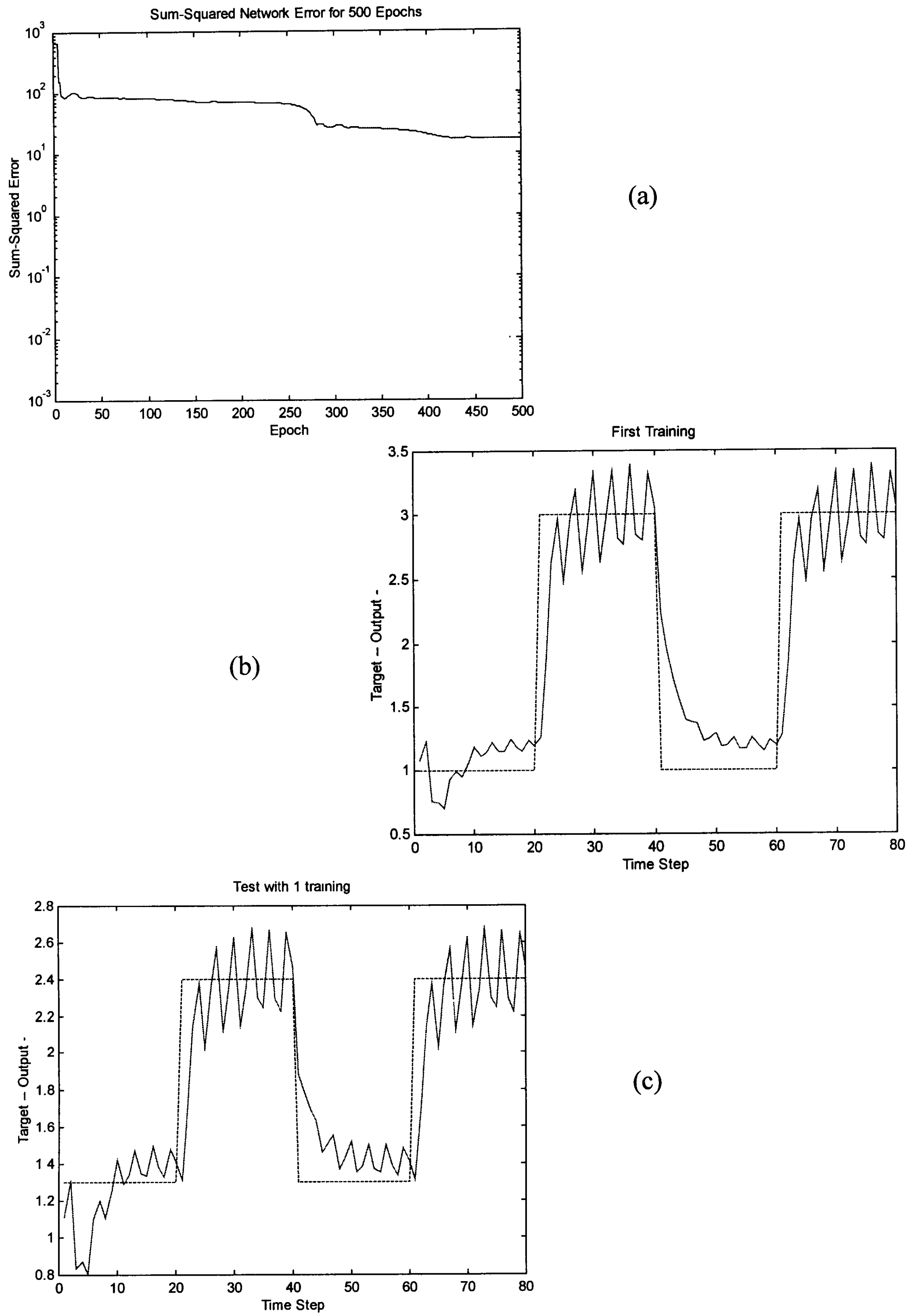


Fig. 21 : Using 100 recurrent neurons.

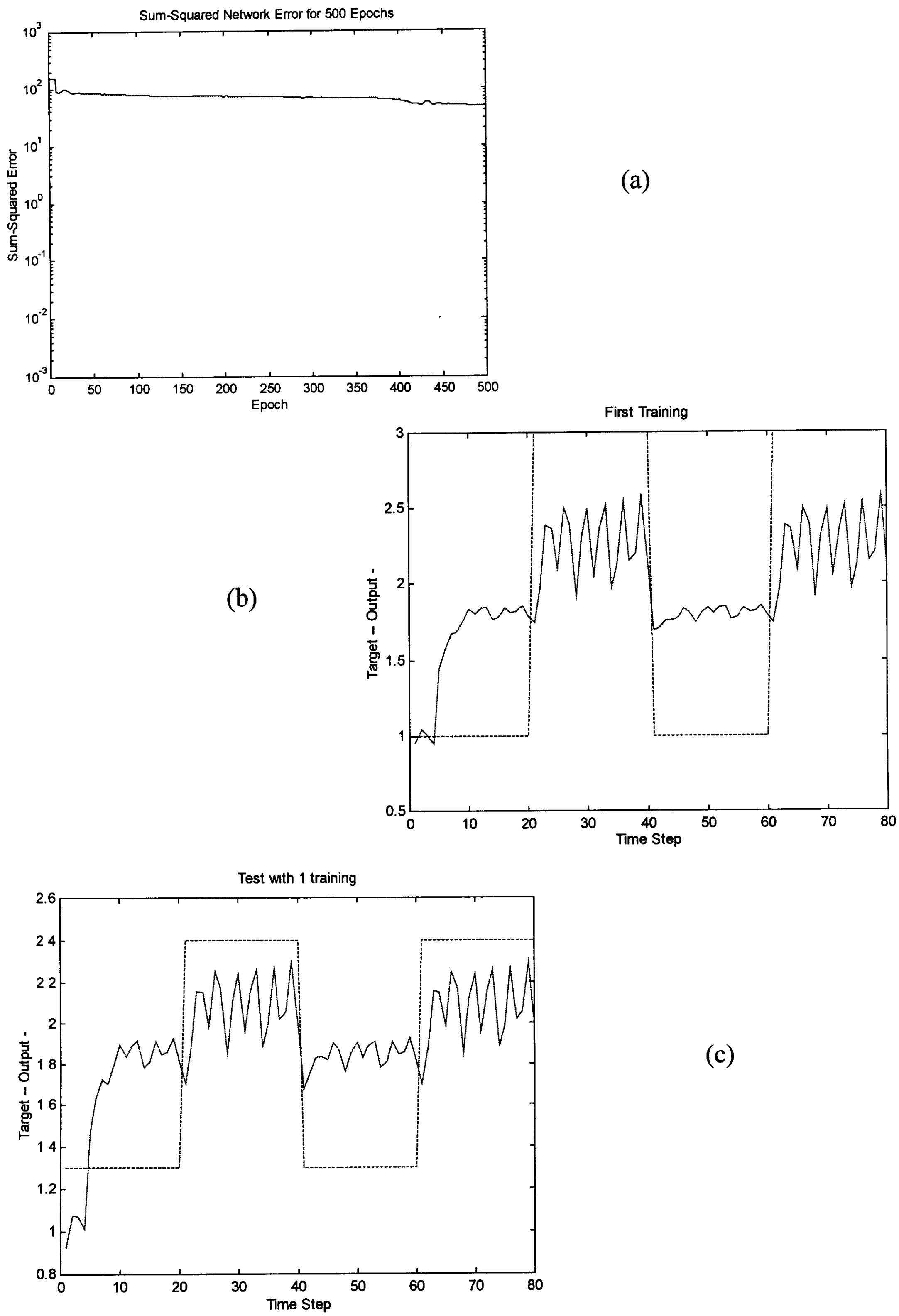
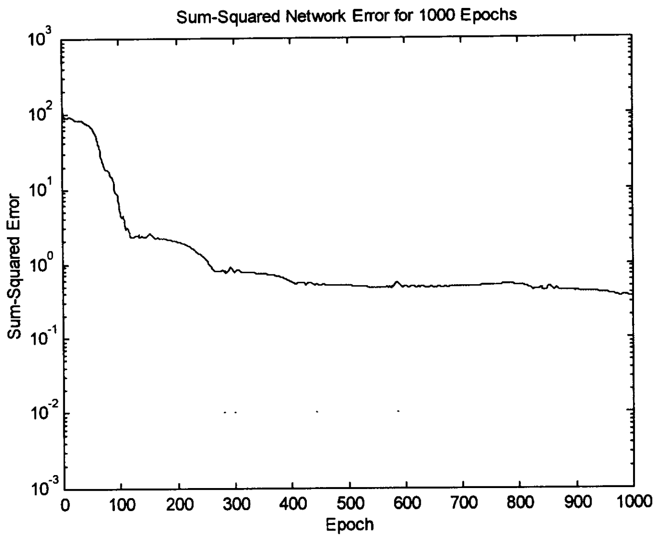
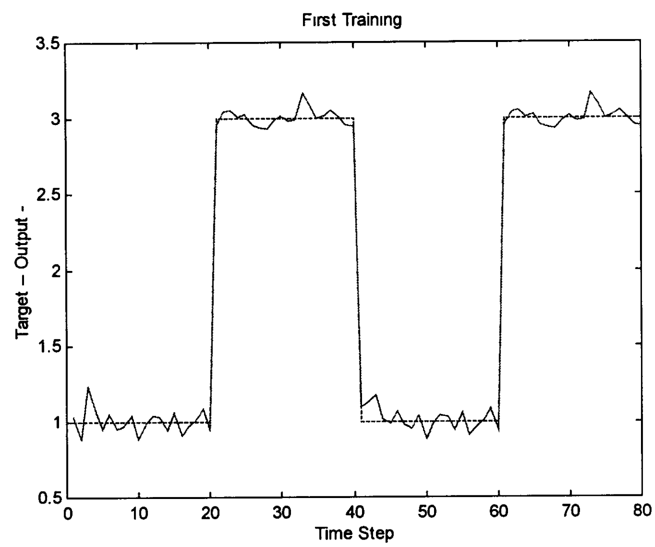


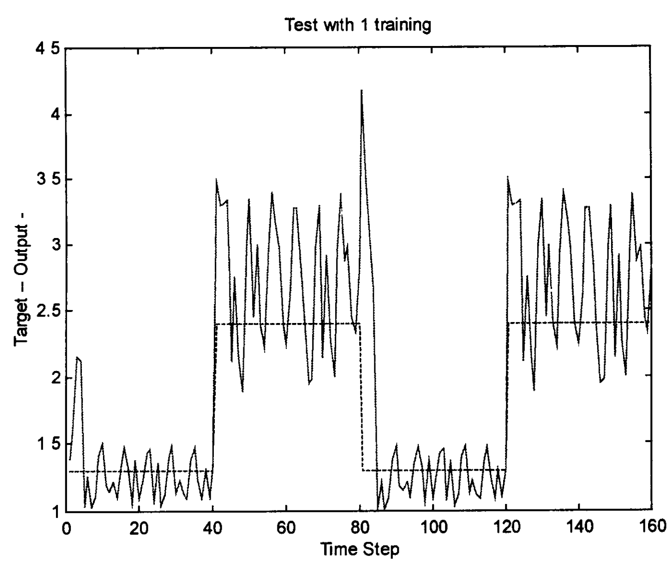
Fig. 22 : Using 200 recurrent neurons.



(a)



(b)



(c)

Fig. 23 : Using 1000 epochs.

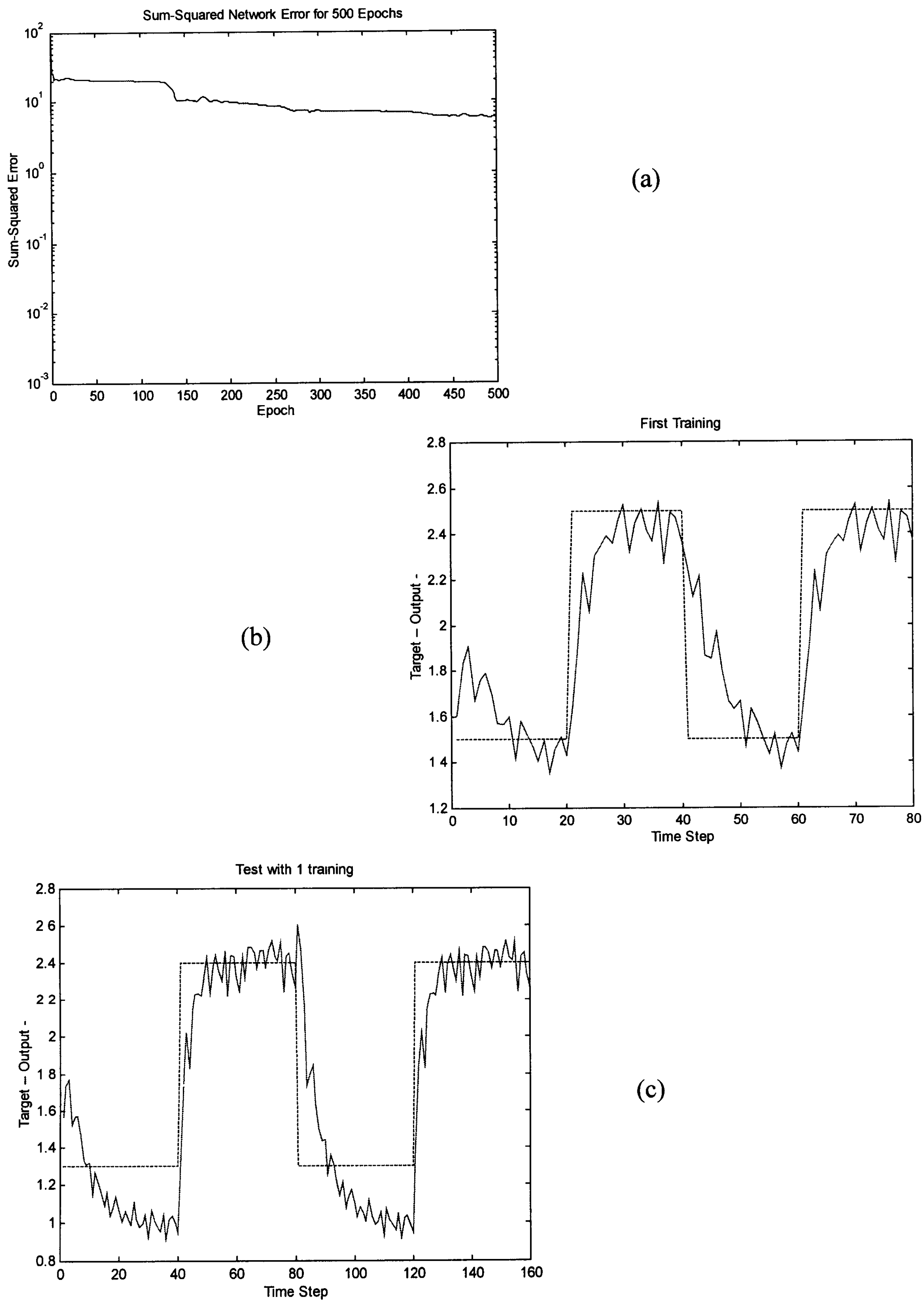


Fig. 24 : Using 2 training input sets.

The network can be trained to respond appropriately to a variety of input functions. In this way, a desired response behavior can be "programmed" into the network, thus allowing it to produce a sufficiently accurately output to any arbitrary input. This can be seen in Fig. 24. With an additional training set of input, the output approximation has been improved.

Obviously, it follows that these improvement methods should be used together to achieve the best effect. Several training cycles can be carried out with a sufficiently long training time using a minimal number of neurons.

A recurrent network such as the Elman network can be used to help effect personal environmental controls. As mentioned before, the users in a room might have different temperature or humidity preferences. This variation in preferences can be transmitted to the recurrent network as a series of target functions. A Computational Fluid Dynamics model can be used to provide continuous feedback of actual conditions in the room. The neural network will then attempt to achieve the best fit of the actual output (in this case the temperature and the humidity) to the target output.

This method might also be applied to an active stiffness control of a building. However, actual earthquake and wind loads are much harder to predict and approximate.

6.3 Predicting Response

The third program trains a linear neuron adaptively to predict a time series. Due to this adaptive training, the linear network will be able to respond to changes between the past and present values of the input function.

The signal to be predicted is a cosine wave lasting for about 7 seconds, with a sampling rate of about 20 samples per second. Initially it has a frequency of 1 Hz, then after 4 seconds, its frequency doubles. After another 2 seconds, it doubles again. A plot of the signal is shown in Fig. 25.

At each time step, the past 5 values of the signal will be sent to the network as inputs. The network will then attempt to predict the output for the next time step. The function *delaysig* is used to obtain the 5 delayed signals.

The network has only 1 neuron since there is only one output value per time step. The function *initlin* creates the initial weights and biases for the neuron.

adaptwh is used to train the neuron adaptively on input/target signals P and t. A learning rate of 0.1 was used. Once the neuron has been trained, it can be tested to compare the simulated output with the input signals.

As can be seen from Fig. 25, the neural network took about 1.5 seconds to track the initial input signal. This is about 30 samples. Then the output matches the target fairly well, until the change in the frequency occurs. The network then has to re-adapt to this new frequency. However, the time taken is much shorter since the network had already picked up the general signal behavior (cosine function). The second adaptation is achieved in an even shorter time. A plot of the error between the target and the output signals is also plotted to further illustrate this (Fig. 26).

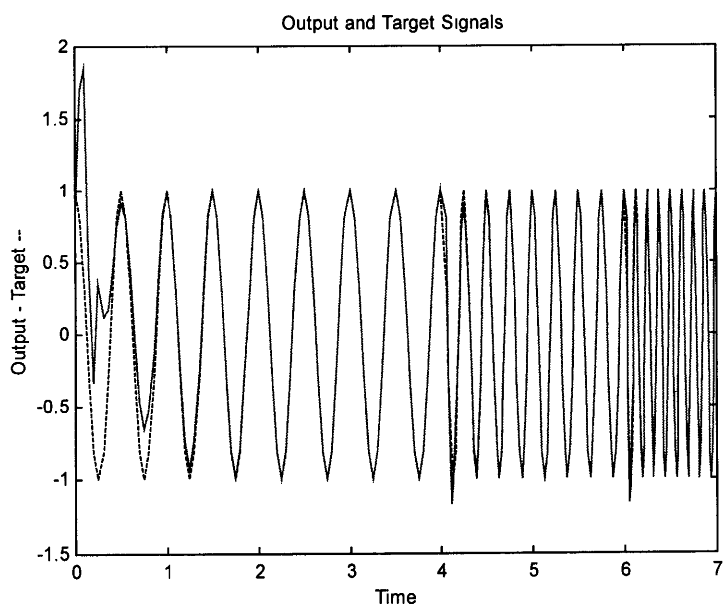


Fig. 25 : Plot of Target and Output Signals (Prediction)

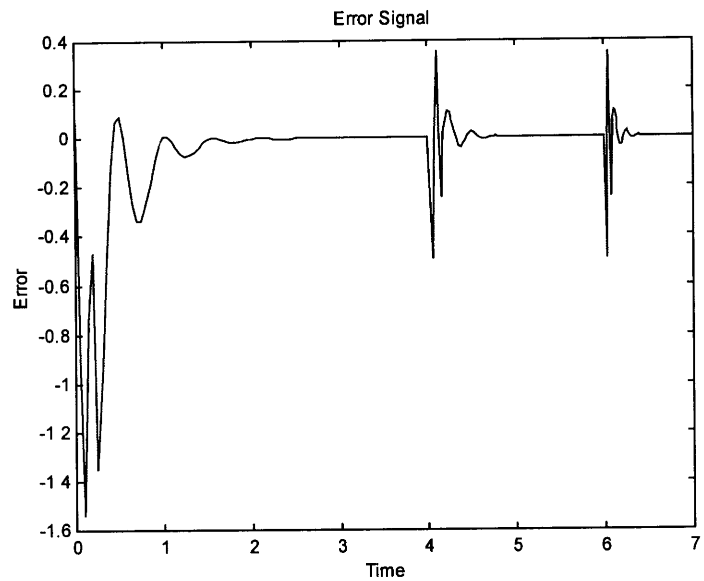


Fig. 26 : Plot of the error signal (Prediction)

The linear network modeled the input signal quite well. The adaptation was very quick too. Only 30 samples were required to pick up the input signal behavior. This is quite impressive given that typical signal processing applications can sample at around 20kHz. This will translate the 30 samples to only about 1.5 milliseconds.

The example demonstrated was for linear behavior. However, a linear network can also be applied to nonlinear behavior, albeit without very good accuracy. Nevertheless, this is still good enough for many applications. For very nonlinear problems, or for low error limits, either backpropagation or radial basis networks can be used. Such capability will allow the neural network to handle structural vibrations. Backpropagation can be used to train the neural network to understand the behavior of the type of input signals encountered. Adaptation algorithms can then be used in sync to predict the possible future signals. Giving the linear network many delayed input signals will provide it with more information in order to calculate the best fit prediction with the lowest error value.

Work had been done using neural networks to control wind and earthquake vibrations [23]. While this had only been carried out for a non-linear, 1 degree of freedom system, the results were promising. It showed that neural network controllers are able to reduce building displacements due to wind and earthquake forces. This might be achieved by extending the programs developed together with this thesis. It might not be possible to predict earthquakes exactly yet, but with this technology and assuming rapid progress in the field, it might be possible one day to read in data from pre-tremors, and then rapidly predict the response for the next time step or further. This prediction can then be used to alter the building stiffness or damping accordingly, in order to avoid resonant behavior.

Chapter 7

Conclusions

A number of conclusions can be derived from this study. Adaptive control is still an emergent technology. However, it has great potential for applications in building technology. Due to the fact that it is mainly established in signal processing, the applications in structural engineering are rather limited. In recent years, there has been a maturing of the relevant technologies, and more civil engineers are becoming more willing to look at these controls as possibilities to improve structural design and maintenance. Adaptive control can be applied to a multitude of structural systems, especially the building automation system (BAS). Such applications will result in a more efficient use of the available facilities, better coordination and control, lower maintenance costs, and improved user comfort.

Adaptive Technology combines knowledge from many fields. Electronics Engineering, Control Theory and AI Technology are some of the examples. Neural networks, knowledge-based systems, and fuzzy logic can all be used to design the adaptive controller. Many of these make use of the same concepts as adaptive control. Various types of feedback can be used to enable the controller to decide which parameters to change and to what extent. By doing so, the adaptive controller can improve the performance of the structure in response to changes in the environmental conditions. Active and adaptive materials like Electrostrictive Ceramics and Shape Memory Alloys can be used to effect the adaptation as sensors and actuators.

Classical control theory can be used, together with some appropriate modifications, to provide us with the equations for designing the control, and understanding how it interacts with the structure. Optimization techniques are also extensively used. An optimized design will help reduce costs, and enable peak performance to be achieved. Therefore, it is incorporated into adaptive control as a task to be carried out at every adaptation step. In this way, the controller will continuously adapt the structural behavior to the optimized state.

Adaptive control techniques have not been properly developed for structural problems yet. This is because buildings are subjected to conditions that are different to those usually encountered in traditional applications of adaptive control. The earthquake for example, is very difficult to characterize, and therefore it is very hard to control.

A few applications have been developed in order to demonstrate some of the concepts involved with adaptive control. An optimal stiffness design was developed to obtain the stiffness values that will best satisfy both the wind and earthquake spectrums. Even though there are some problems due to the lack of wind spectrum information, the methodology can be applied reasonably to high-rise buildings. This optimal solution can be derived for the structure concerned at every stage under time varying conditions. This will play a part in the whole adaptation process.

An application was also developed to look into the training of the response of a neural network to certain types of input signals. The neural network was shown to exhibit learning behavior. By varying the parameters of the adaptation process, the quality of the learning can be changed. Therefore, by using an appropriate number of neurons, a relatively long training time, and numerous training cycles to condition the network, a good behavior of the network can be achieved. This will allow an accurate detection and emulation of the input signals sent to the network. This has great potential to be applied to HVAC systems in order to effect a personalized environmental control. The different environmental settings across a room can be fed into a neural network in the form of signals. The network will then attempt to emulate these signals and thus produced the desired settings in the room.

The last application deals with predicting input signals. Once the neural network has been trained adaptively, it will be able to pick up a signal's characteristics and attempt to predict the behavior of the signal over the next time step or more. The more repetitive the signal, the faster the network will be able to carry out the adaptation. With the use of more neurons, the network will be able to handle more complicated input signals and even non-linear signals. However, random signals like earthquakes are still out of reach.

In this ever-complex world of ours, there is a constant need to improve. The demands on a building's performance and functionality are getting higher and higher. In order to cope with this, innovative design methods have to be introduced to enable the structure to adopt certain smart characteristics that will enhance its performance. Controls, and adaptive controls in particular, provide such a potential solution. Adaptive controls will allow us to act on the environmental conditions, observe the consequences of these actions and then utilize optimization and learning to achieve the desired behavior.

Chapter 8

References

1. Mareels I., Polderman J. W., "Adaptive Systems: An Introduction", Birkhauser, Boston, 1996
2. Ioannou P. A., Sun J., "Robust Adaptive Control", Prentice Hall, New Jersey, 1996
3. Butler H., "Model Reference Adaptive Control – (from theory to practice)", Prentice Hall, UK, 1992
4. Hryces, T., "Neurocontrol: Towards an Industrial Control Methodology", John Wiley & Sons, New York, 1997
5. Culshaw B., Gardiner P. T., McDonach A., "First European Conference on Smart Structures and Materials", co-published by Institute of Physics Publishing, Bristol, and EOS/SPIE EUROPTO Series SPIE Volume 1777, 1992
6. Tzou H. S., Anderson G. L., "Intelligent Structural Systems", Kluwer Academic Publishers, The Netherlands, 1992
7. Tzonis A., "Movement, Structure and the work of Santiago Calatrava", Birkhauser, Basel, Boston, 1995
8. http://www.hoberman.com/fold/, "Hoberman Associates, Inc., Unfolding Structures"
9. Douay A., "Optimal Variable Stiffness Feedback Control for Structural Vibration Suppression", M.S. Thesis, Department of Aeronautics and Astronautics, MIT, 1993
10. http://www.newport.com/tutorials/Electro_vs_Piezo_Actuators.html, "Electro vs Piezo Actuators"
11. Proceedings Singapore International conference on Intelligent Controls and Instrumentation, IEEE Singapore Section, Singapore, 1992
12. Page G. F., Gomm J. B., Williams D., "Application of Neural Networks to Modelling and Control", Chapman & Hall, London, 1993

13. Tsoukalas L. H., Uhrig R. E., "Fuzzy and Neural Approaches in Engineering", Wiley-Interscience, New York, 1997

14. Tao W. K. Y., Janis R. R., "Mechanical and Electrical Systems in Buildings", Prentice Hall, New Jersey, 1997

15. Herskovits J., "Advances in Structural Optimization", Kluwer Academic Publishers, the Netherlands, 1995

16. Connor J. J., Klink B. S. A., Introduction to Motion Based Design, Computational Mechanics Publications, MA, 1996

17. Cruz J. B., Jr., "Feedback Systems", McGraw-Hill Inc., USA, 1972

18. Berg G. V., "Elements of Structural Dynamics", Prentice-Hall International Editions, USA, 1988

19. Clough R. W., Penzien J., "Dynamics of Structures", McGraw-Hill Inc., USA, 1993

20. Widrow B., "Adaptive Signal Processing and Adaptive Neural Networks (Videorecording)", IEEE Educational Activities, IEEE, Piscataway, NJ, 1992

21. Demuth H., Beale M., "Neural Network Toolbox for use with MATLAB", The Mathworks, Inc., MA, 1994

22. Weisshaar T. A., Newsom J. R., Gilbert M. G., and Zeiler T. A., "Integrated Structure/Control Design-Present Methodology and Future Opportunities" ICAS-86-4.8.1, 1986

23. Nerves A. C., Krishnan R., Singh M. P., "Modeling, Simulation and Analysis of Active Control of Structures with Nonlinearities Using Neural Networks", Engineering Mechnanics, Proceedings of the 10th Conference on Engineering Mechanics, May 21-24 1995, v2, Boulder, CO, USA, Sponsored by: ASCE New York, USA, p 1054-1057

Chapter 9

Appendices

Appendix I : Optimal Stiffness Design

```
%******************************************************************
% Function to minimize the sum of the stiffness of all levels
% of a simple structure. The degrees of freedom can be easily
% increased.
%           k:  initial stiffness
%******************************************************************

function [f,g]=obj3(k)

% X'= AX + Bu;
% y = CX + Du;

% hC: constraint for frequency response of inter-floor displacement
% hC=0.0528* [ 13.5 0.89 0.37 1 ];
hC=0.049* [ 13.5 0.89 0.37 1 ];

kC=[ 100 75 50];

M=[ 100 0 0; 0 100 0; 0 0 50]/386.088;
invM= inv(M);

K=[ k(1)+k(2) -k(2) 0; -k(2) k(2)+k(3) -k(3); 0 -k(3) k(3)];

% 2% damping
C= K*0.02;

A=[ zeros(3) eye(3); -invM*K -invM*C];

B=[ 0;0;0;1;1;1];

D=[ 0];

dt=0.02;

i=-2:dt:2;

w=10.^i;


% convert state-space to transfer function: 1st inter-floor displacement
%------------------------------------------------------------------

C=[ 1 0 0 0 0 0];

[num1,den1]= ss2tf( A, B, C, D, 1);
```

```
% convert state-space to transfer function: 2nd inter-floor displacement
%-----------------------------------------------------------------------

C=[ -1 1 0 0 0 0];

[num2,den2]= ss2tf( A, B, C, D, 1);

% convert state-space to transfer function: 3rd inter-floor displacement
%-----------------------------------------------------------------------

C=[ 0 -1 1 0 0 0];

[num3,den3]= ss2tf( A, B, C, D, 1);

% frequency response of inter-floor displacements
[h1, W]=freqs( num1, den1, w);
[h2, W]=freqs( num2, den2, w);
[h3, W]=freqs( num3, den3, w);

% objective function
f= sum(k);

t1= floor(1/dt);
t2= floor(1.87/dt);
t3= floor(3.47/dt);

% inequality constraints: design criteria of the inter-floor
displacements
g(1)=  max(abs(h1( 1:t1)))- hC(1);
g(2)=  max(abs(h2( 1:t1)))- hC(1);
g(3)=  max(abs(h3( 1:t1)))- hC(1);
g(4)=  max(abs(h1(t1:t2)))- hC(2);
g(5)=  max(abs(h2(t1:t2)))- hC(2);
g(6)=  max(abs(h3(t1:t2)))- hC(2);
g(7)=  max(abs(h1(t2:t3)))- hC(3);
g(8)=  max(abs(h2(t2:t3)))- hC(3);
g(9)=  max(abs(h3(t2:t3)))- hC(3);
g(10)= max(abs(h1(t3:201)))- hC(4);
g(11)= max(abs(h2(t3:201)))- hC(4);
g(12)= max(abs(h3(t3:201)))- hC(4);

% inequality constraints: design criteria of the required stiffness
g(13)= kC(1)-k(1);
g(14)= kC(2)-k(2);
g(15)= kC(3)-k(3);

% END
```

Appendix II : Learning of a Neural Network

```
%*********************************************
% Program to train an Elman Network.
% Different parameters can be changed in order
% to achieve different training qualities.
% The neuron can be used to detect and emulate
% signals.
%*********************************************


% Single waves
p1=1*cos(1:20);
p2=3*cos(1:20);

% Target waves
t1=1*ones(1,20);
t2=3*ones(1,20);

% Sequenced waves
p=[p1 p2 p1 p2];
t=[t1 t2 t1 t2];

% 1 input
R=1;

% 1 output neuron
s2=1;

% Recurrent Neurons (to be changed: 10, 100, 200)
s1=10;

% Transfer function (use to change the training time)
% i.e. tp=[5,1000,0.01,0.001,1.05,0.7,0.95,1.04] => 1000 epochs used
tp=[5,500,0.01,0.001,1.05,0.7,0.95,1.04];

% initialize the neuron
[W1,B1,W2,B2] = initelm([-3 +3],s1,s2);

% train the neuron
[W1,B1,W2,B2,TE] = trainelm(W1,B1,W2,B2,p,t,tp);
pause

% An Example of a second training set
%-------------------------------------------------

p3=1.5*cos(1:20);
p4=2.5*cos(1:20);

t3=1.5*ones(1,20);
t4=2.5*ones(1,20);

pp=[p3 p4 p3 p4];
tt=[t3 t4 t3 t4];
```

```
R=1;
s2=1;
s1=10;
tp=[5,500,0.01,0.001,1.05,0.7,0.95,1.04];

[W1,B1,W2,B2] = initelm([-3 +3],s1,s2);
[W1,B1,W2,B2,TE] = trainelm(W1,B1,W2,B2,pp,tt,tp);
pause

%--------------------------------------------------

% Simulate the neural network
a=simuelm(p,W1,B1,W2,B2);
time=1:length(p);

% Plot the graph
plot(time,t,'--',time,a)
xlabel('Time    Step');ylabel('Target    --    Output    -');title('First
Training');
pause

% END
```

Appendix III : Predicting Response

```
%*****************************************************
% Program to predict the response of input signals.
% Using a number of delayed signals from the input
% (target) function, further similar response can
% be predicted.
%*****************************************************

% Setting the target function
time1=0:0.05:4;
time2=4.05:0.025:6;
time3=6.025:0.0125:7;
time =[time1 time2 time3];
t=[cos(4*pi*time1) cos(8*pi*time2) cos(16*pi*time3)];

% Setting 5 delayed signals as inputs
P = delaysig(t,1,5)

% Initialize the weights can biases
[w,b]=initlin(P,t);

% Train the neuron adaptively using the Widrow-Hoff rule
[a,e,w,b]=adaptwh(w,b,P,t,0.1);

plot(time,t,'--',time,a)
xlabel('Time');ylabel('Output - Target --');
title('Output and Target Signals');
pause

plot(time,e)
xlabel('Time');ylabel('Error');title('Error Signal');

% END
```

Appendix IV : MATLAB Functions used

1. SS2TF State-space to transfer function conversion.

[NUM,DEN] = SS2TF(A,B,C,D,iu) calculates the transfer function:

$$H(s) = \frac{NUM(s)}{DEN(s)} = C(sI-A)^{-1} B + D$$

of the system:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

from the iu'th input. Vector DEN contains the coefficients of the denominator in descending powers of s. The numerator coefficients are returned in matrix NUM with as many rows as there are outputs y.

See also TF2SS.

2. NORM Matrix or vector norm.

For matrices...

NORM(X) is the largest singular value of X, max(svd(X)).

NORM(X,2) is the same as NORM(X).

NORM(X,1) is the 1-norm of X, the largest column sum,

= max(sum(abs((X)))).

NORM(X,inf) is the infinity norm of X, the largest row sum,

= max(sum(abs((X')))).

NORM(X,'fro') is the Frobenius norm, sqrt(sum(diag(X'*X))).

NORM(X,P) is available for matrix X only if P is 1, 2, inf or 'fro'.

For vectors...

NORM(V,P) = sum(abs(V).^P)^(1/P).

NORM(V) = norm(V,2).

NORM(V,inf) = max(abs(V)).

NORM(V,-inf) = min(abs(V)).

See also COND, CONDEST, NORMEST.

Overloaded methods

help lti/norm.m

3. DELAYSIG Create delayed signal matrix from signal matrix.

DELAYSIG(X,D)

X - SxT matrix with S-element column vectors for T timesteps.

D - Maximum delay.

Returns signal X delayed by 0, 1, ..., and D2 timesteps.

DELAYSIG(X,D1,D2)

X - SxT matrix with S-element column vectors for T timesteps.

D1 - Minimum delay.

D2 - Maximum delay.

Returns signal X delayed by D1, D1+1, ..., and D2 timesteps.

The signal X can be a row vector of values, or a matrix of (column) vectors.

EXAMPLE: X = [1 2 3 4 5; 10 9 8 7 6];

Y = delaysig(X,1,3)

4. INITLIN Initialize linear layer.

[W,B] = INITLIN(R,S)

R - Number of inputs to the layer.

S - Number of neurons in layer.

Returns:

W - SxR Weight matrix.

B - Bias (column) vector.

[W,B] = INITLIN(P,T)

P - RxQ matrix of input vectors.

T - SxQ matrix of target outputs.

Returns weights and biases.

EXAMPLE: [w,b] = initlin(2,3)

p = [1; 2; 3];

a = simulin(p,w,b)

See also NNINIT, LINNET, SOLVELIN, SIMULIN, LEARNWH, ADAPTWH, TRAINWH.

5. ADAPTWH **Adapt linear layer with Widrow-Hoff rule.**

[A,E,W,B] = ADAPTWH(W,B,P,T,lr)

W - SxR weight matrix.

B - Sx1 bias vector.

P - RxQ matrix of input vectors.

T - SxQ matrix of target vectors.

lr - Learning rate (optional, default = 0.1).

Returns:

A - output of adaptive linear filter.

E - error of adaptive linear filter.

W - new weight matrix

B - new weights & biases.

See also ADAPTFUN, LINNET, SIMLIN, SOLVELIN, INITLIN, LEARNWH, TRAINLIN.

```
EXAMPLE: time = 0:1:40;
        p = sin(time);
        t = p*2+1;
        [w,b] = initlin(p,t);
[a,e,w,b] = adaptwh(w,b,p,t,1.0);
plot(time,t,'+',time,a)
label('time','output - target +','Output and Target Signals')
```

6. INITELM Initialize Elman recurrent network.

[W1,B1,W2,B2] = INITELM(P,S1,S2)

P - RxQ matrix of input vectors.

S1 - Number of hidden TANSIG neurons.

S2 - Number of output PURELIN neurons.

Returns:

W1 - Weight matrix for recurrent layer.

B1 - Bias (column) vector for recurrent layer.

W2 - Weight matrix for output layer (from first layer).

B2 - Bias (column) vector for output layer.

[W1,B1,W2,B2] = INITELM(P,S1,T)

T - SxQ matrix of target outputs.

Returns weights and biases.

IMPORTANT: Each ith row of P must contain expected
min and max values for the ith input.

EXAMPLE: P = [sin(0:100); cos([0:100]*2)];
t = 2*P(1,:) - 3*P(2,:);
[W1,b1,W2,b2] = initelm(P,2,t);
See also ELMAN, SIMELM, TRAINELM

7. TRAINELM Train Elman recurrent network.

[W1,B1,W2,B2,TE,TR] = TRAINELM(W1,B1,W2,B2,P,T,TP)
W1 - Weight matrix for first layer (from input & feedback).
B1 - Bias (column) vector for first layer.
W2 - Weight matrix for second layer (from first layer).
B2 - Bias (column) vector for second layer.
P - Input (column) vectors arranged in time.
T - Target (column) vectors arranged in time.
TP - Vector of training parameters (optional).
Returns:
W1,B1,W2,B2 - New weights and biases.
TE - Number of epochs trained.
TR - Record of errors throughout training.

Training parameters are:
TP(1) - Number of epochs between display (default = 5).
TP(2) - Maximum number of epochs to train (default = 500).
TP(3) - Sum squared error goal (default = 0.01).
TP(4) - Initial adaptive learning rate (default = 0.001).
TP(5) - Ratio to increase learning rate (default = 1.05).
TP(6) - Ratio to decrease learning rate (default = 0.7).
TP(7) - Momentum constant (default = 0.95).
TP(8) - Error ratio (default = 1.04).
Missing parameters and NaN's are replaced with defaults.

See also NNTRAIN, ELMAN, INITELM, SIMUELM.

8. SIMUELM Simulates an Elman recurrent network.

[A1,A2] = SIMUELM(P,W1,B1,W2,B2,A1)

P - Input (column) vectors to network arranged in time.

W1 - Weight matrix for recurrent layer.

B1 - Bias (column) vector for recurrent layer.

W2 - Weight matrix for output layer.

B2 - Bias (column) vector for output layer.

A1 - Initial output vector of recurrent layer (optional).

Returns:

A1 - Output (column) vectors of recurrent layer in time.

A2 - Output (column) vectors of output layer in time.

A2 = SIMUELM(P,W1,B1,W2,B2,A1)

Returns only the output vectors.

```
EXAMPLE: [W1,b1,W2,b2] = initelm([-5 5; 0 2],4,1);
      p = [3; 1.5];
      a = simuelm(p,W1,b1,W2,b2)
```

See also NNSIM, ELMAN, INITELM, TRAINELM.

9. FREQS Laplace-transform (s-domain) frequency response.

H = FREQS(B,A,W) returns the complex frequency response vector H of the filter B/A:

```
            nb-1     nb-2
 B(s)  b(1)s  + b(2)s  + ... + b(nb)
H(s) = ---- = --------------------------------
            na-1     na-2
 A(s)  a(1)s  + a(2)s  + ... + a(na)
```

given the numerator and denominator coefficients in vectors B and A. The frequency response is evaluated at the points specified in vector W. The magnitude and phase can be graphed by calling FREQS(B,A,W) with no output arguments.

[H,W] = FREQS(B,A) automatically picks a set of 200 frequencies W on which the frequency response is computed. FREQS(B,A,N) picks N frequencies.

See also LOGSPACE, POLYVAL, INVFREQS, and FREQZ.