TIME DELAY COMPENSATION IN FINITE-ORDER MODELS

OF ACOUSTIC WAVE PROPAGATION

IN HOMOGENOUS MEDIA

_____

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri

_____

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

_____

by

NILESH V SALVI

Dr. Jinglu Tan, Thesis Supervisor

MAY 2011

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

TIME DELAY COMPENSATION IN FINITE ORDER MODELS OF ACOUSTIC
WAVE PROPAGATION IN HOMOGENOUS MEDIA

presented by Nilesh Salvi,

a candidate for the degree of master of science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Jinglu Tan, Biological Engineering

Professor Ali Bulent Koc, Biological Engineering

Professor Roger Fales, Mechanical & Aerospace Engineering

ACKNOWLEDGEMENT

TABLE OF CONTENTS

iv

LIST OF FIGURES

LIST OF TABLES

TIME DELAY COMPENSATION IN FINITE-ORDER MODELS
OF ACOUSTIC WAVE PROPAGATION IN HOMOGENOUS MEDIA

Nilesh Salvi

Dr. Jinglu Tan, Thesis Supervisor

ABSTRACT

Finite-order models do not completely account for the delay in acoustic wave propagation and thus require an additional phase correction, besides parameter adjustments to fit experimental measurements. As a consequence, it is necessary to determine the time or phase delay of a finite-order model as a function of excitation frequency and model order.

In this work a homogenous, one-dimensional medium is discretized in finite a number of elements. Two methods were developed to derive the transfer function of wave transmission for an arbitrary number of elements. Results from the two methods were verified with transfer functions computed from state space models developed in the time domain. The transfer functions were used to evaluate the model time delays and consequently the needed additional time delay corrections for a given system. Experimental data were collected and used, to verify utility of the method. By providing the time delay correction, the method helps enhance the model parameter estimation process.

CHAPTER 1

INTRODUCTION

1.1   Problem Statement

Propagation of a sound wave through a medium is a phenomenon dependent upon the medium properties. Depending upon the viscous and elastic nature of the medium material, sound wave propagates through it at a different velocity. This velocity for a known input wave, if measured to certain accuracy, can become an identity for the material.

The wave equation is an approximate hyperbolic partial differential equation that is considered a good representation of small-amplitude wave motion. It is an infinite-order model that gives a useful linear approximation to real non-linear phenomenon. The following is the acoustic wave equation in one dimension.

$$\frac{\partial^2 p}{\partial x^2} - \left(\frac{1}{c^2}\right)\frac{\partial^2 p}{\partial t^2} = 0 \tag{1.1}$$

where, p is the acoustic pressure (the local deviation from the ambient pressure) in the medium, t is time, x is the spatial co-ordinate and c is the speed of sound wave propagation given by,

$$c = \sqrt{\frac{B}{\rho}}$$

1

where, B is the coefficient of stiffness or bulk modulus, the modulus of bulk elasticity and ρ is the medium density.

The use of finite-order models facilitates the use of the linear system theory in obtaining solutions for wave propagation. The medium is divided in to a finite number of discrete elements as shown in Fig. 1.1. Every discrete element is connected to the neighboring elements through connections given by the Kelvin-Voigt network model.
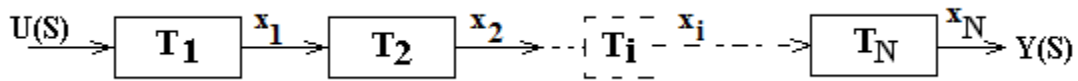
*Fig. 1.1 **Finite-order representation of a propagation medium (Order = 2N)***

The first element is subject to an external force, or pressure. The other end is held fixed. In the case of a periodic external force stimulus on the first element, all the elements are to oscillate about their mean rest positions and not to undergo any permanent translation. By using the boundary conditions, the transfer function for each element can be obtained successively. This provides a model for the entire length of the propagation medium and time domain response can be obtained by convolution of the input and the transfer functions.

The direct method for sound speed estimation makes use of time taken by a wave to travel through a medium of known length. This time is often measured in terms of

phase lag between the measured wave and the input wave. The method used to obtain this time-of-flight through measurement is illustrated in Chapter 4.

The term 'order' in this work, is two times the number of discrete elements, which are connected in a series to represent the length of the propagation medium. For a homogenous medium, all the elements have the exact same material properties. For the current method of discretization, all the elements have exactly the same geometry. In light of the incompleteness of finite models (Fagin 1993), an explicit phase correction will be required to compensate for the modeling errors.
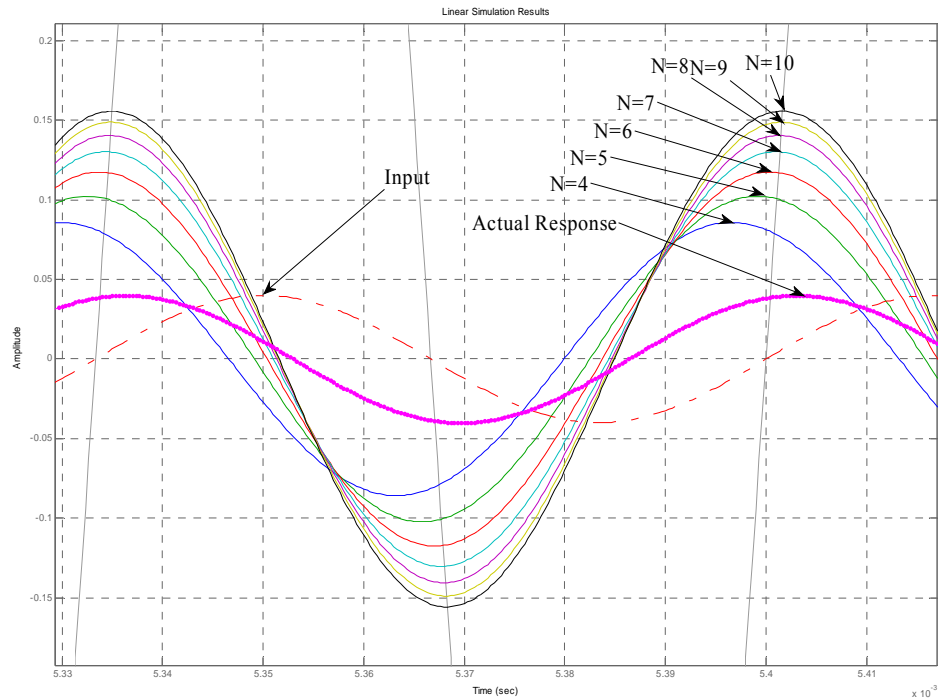
*Fig. 1.2 **Comparison with modeled and actual response to sinusoidal input for varied N***

As shown in Figures 1.2 and 1.3, the response obtained with a finite model does not match the measured response. The phase lag from the input in the model response increases with increasing number of elements and consequently gets closer to the measured response. But as far as the number of elements does not reach ∞, complete agreement with experiment data is not obtained.
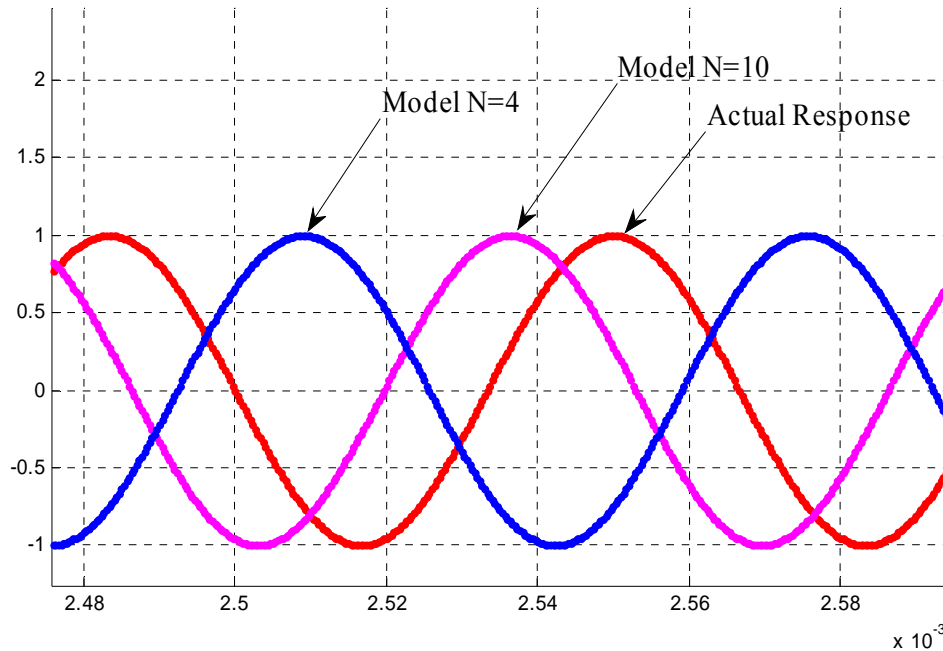
Fig. 1.3 **Gain compensated illustration of insufficiency of finite-order models to produce time delay matching the actual time-of-flight**

4

1.2 Research Objectives

The overall goal of this research was to find a method to correct time delay errors in finite-order models for sound wave propagation. The specific objectives are:

1. To analyze the relationship between time delay given by finite-order sound wave propagation models and their order for homogenous media,

2. To formulate time delay correction for finite-order models, in terms of model order and input frequency, and

3. To validate the method with experimental data.

CHAPTER 2

LITERATURE REVIEW

The task of modeling acoustic wave propagation requires selecting proper methods to design, analyze and validate the results. Previous work done on sound speed measurement and wave modeling is important for this research. This chapter summarizes the relevant previous work.

2.1 Sound Wave Modeling

The acoustic wave equation, as mentioned in Chapter 1, is a mathematical representation of the physical wave phenomenon in the form of a hyperbolic partial differential equation. There have been various linear methods proposed to model the wave phenomenon. According to Billingham and King (2001), more realistic differential equations exist for waves that allow the speed of wave propagation to vary with its frequency; a phenomenon known as dispersion. In this case, wave velocity must be replaced by the phase velocity, which is the rate at which the phase of the wave propagates in space. For a more comprehensive definition of phase velocity, refer to Appendix B.

According to Dameron (1979), the two basic equations for linearized ultrasonic wave propagation in one dimension are Euler's equation of motion

$$\rho_0 \frac{\partial v}{\partial t} = -\frac{\partial P}{\partial x} \tag{2.1}$$

and the continuity equation

$$\rho_0 \frac{\partial v}{\partial t} + v\frac{\partial \rho_0}{\partial x} = -\frac{\partial \rho}{\partial t} \tag{2.2}$$

where ρ is mass density, P is pressure and v is particle velocity. ρ_0 is the density in the absence of the wave. These equations have assumed that,

$$v \ll c = (\rho_0 k_0)^{-(\frac{1}{2})} \tag{2.3}$$

where k_0 is compressibility of the medium material in the absence of an acoustic wave. The material variables ρ_0 and k_0 are assumed to be arbitrary functions of space and not time.

Euler's equation is a special case of the conservation of momentum equation for a non-viscous (invisid) medium. A third equation is needed relating the small changes in pressure P to small changes in density ρ because of wave motion; this is the equation of state for the fluid. Thermodynamics tells us that, in general, pressure P is a function of two thermodynamic variables, for example, density ρ and entropy S. Since acoustic oscillations are assumed small, we can neglect thermal conduction from wave-induced temperature differences so that the wave process is adiabatic as well as inviscid. Expanding ρ in a Taylor series of P and keeping only the linear term, we have,

7

$$\rho - \rho_0 = \frac{\partial \rho_0}{\partial P}\Big|_s (P - P_0) \tag{2.4}$$

where

$$\frac{\partial \rho_0}{\partial P}\Big|_s = k_0\rho_0 \triangleq \frac{1}{c^2(x)}$$

$\rho - \rho_0$ and $P - P_0$ are the density and pressure fluctuations caused by the wave. ρ_0 and Po are the values in the absence of the wave. k_0 is the compressibility, also constant, to the first order. $\frac{\partial \rho_0}{\partial P}\Big|_s$ is defined to be $\frac{1}{c^2(x)}$ from thermodynamics.

To get a wave equation for the wave variable P, we first take partial derivatives of Eqn. (2.1) with respect to x and partial derivatives of Eqn. (2.2) with respect to time t. Combining the resulting equations yields,

$$\frac{\partial^2 P}{\partial x^2} - \frac{\partial^2 \rho}{\partial t^2} = 0 \tag{2.5}$$

$$\frac{\partial^2 P}{\partial x^2} - \left(\frac{1}{c^2(x)}\right)\frac{\partial^2 P}{\partial t^2} = 0 \tag{2.6}$$

which is the wave equation presented in Chapter 1.

Acoustic signal processing is performed based on discrete system models. But, acoustic signals are produced by acoustic and mechanical processes of continuous media,

which are governed by wave equations (Tohyama and Koike, 1998). Banta (1965), in a study of finite amplitude sound waves, used the Taylor series expansion of the wave equation about the boundary, to exploit the greater availability of electronic computers for evaluating iterative procedures and for general broadening of knowledge about nonlinear equations and their solutions.

Kang (2007) has proposed a modeling technique based on the fact that the dynamic displacement of any point in a waveguide can be determined by superimposing the amplitudes of the wave components traveling along the waveguide, where the wave numbers of the constituent waves are defined in the Laplace domain. The proposed analysis approach results in recursive computational algorithms that always involve computations of fixed-size matrices regardless of the number of elements, which can be implemented into highly efficient computer codes. Since neither exact nor approximate Eigen solutions are required as a priori, this method is suitable for the forced response analysis.

For simulation purposes, as is indicated, there are two limiting cases that may affect the analysis accuracy and/or numerical efficiency of the wave approach, viz.

A. When the waveguide contains a *very* small amount of inertia or flexibility such as mass-less elements or rigid bodies, it results in making the wavelength of the constituent waves larger than the span length of the waveguide

B. When the waveguide is *extremely* flexible or its response contains a very sharp impulsive spike such as the impulse response example discussed above, it results in making the wavelength of the constituent waves unrealistically small.

In practice, however, most engineering systems and structures have reasonable amounts of inertia, flexibility, and also damping such that these two limiting cases will be rarely encountered. Hence, in practice, the results obtained as model output will require an implicit correction in order to match the experimental data.

An alternative approach to the analysis of the dynamic response of one-dimensional distributed parameter systems is applying the concepts of wave motions in elastic waveguides. The analysis technique is demonstrated using the flexural vibrations of multi-span damped beams with general support and boundary conditions; however other one-dimensional systems such as strings, rods, and higher order beams can be treated in the same manner. The proposed approach allows a systematic formulation that yields exact, closed-form, distributed transfer functions from which the transient response and frequency response solutions can be obtained.

The propagation media are the series of discrete elements lumped together to form a continuous system. The joints between discrete elements form mathematical discontinuities which demand to be modeled separately. When the wave packet is incident upon a series of discontinuities along its traveling path, it is computationally more efficient to employ the concepts of generalized wave reflection and transmission

matrices, in particular when the free or forced vibration analysis of a multi-span beam is sought. These matrices relate the amplitudes of incoming and outgoing waves at a point of discontinuity (Kang, 2007).

According to Mäkilä and Partington (2003), often no accurate dynamical model of a real system is available through first principle modeling, so one need to perform input–output experiments on the system for the purpose of modeling via system identification. A linear time invariant approximation F of G is said to be a best LTI approximation of G if,

$$\|G - F\|_i = inf \|G - H\|_i \tag{2.7}$$

where the infimum is taken over all causal LTI systems H such that $\|H\|_i < \infty$. Mäkilä and Partington (2003) have estimated both a good LTI model and the size of the un-modeled dynamics.

According to Nagurka and Huang (2006), the use of linear mechanical model parameter namely, Mass, Spring and Damper to model any non-linear physical parameters such as elasticity, friction and inertia provide predictions of the equivalent stiffness, damping, natural frequency, damping ratio and coefficient of restitution for a forced vibration induced by known input force. Work of Nagurka and Huang (2006) has addressed to linear damping issues such as aerodynamic drag as well as non-linear velocity dependent drag characteristics.

According to Drozdov (1996), use of the Kelvin-Voigt model to connect discrete elements provides good results in case of constant or periodic stress as compared to the Maxwell model. In this model (Fig. 2.1) both the elements (elasticity and viscosity) experience the same stress.

$$\sigma = \sigma_{visc} + \sigma_{elast} \tag{2.8}$$

$$\sigma = \varepsilon_{elast}E + \eta\frac{d\varepsilon_{visc}}{dt} \tag{2.9}$$
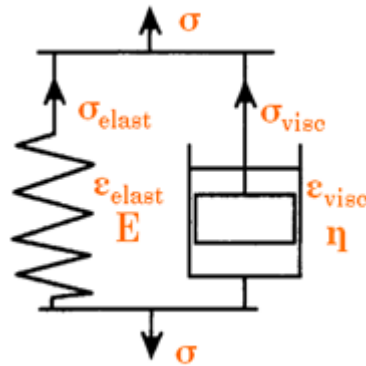
Fig. 2.1 **Kelvin-Voigt Model**

where σ is stress, ε is strain, E is modulus of elasticity and η is damping coefficient.

The stresses are assumed to be additive and overall stress is to remain constant.

$$\varepsilon(t) = \frac{\sigma_0}{E}\left[1 - e^{-\lambda t}\right] \tag{2.10}$$

12

where $\lambda = \frac{E}{\eta}$ is rate of relaxation, σ_0 is stress in absence of wave and t is time.

The Kelvin-Voigt model is great for modeling creep. But it does not account for instantaneous strain or permanent deformation. It is perfect to be applied in cases when loading is not too high.

Analytical methods, such as in Rabenstein and Schetelig (1998), make use of geometry are based on the concept of acoustic rays or plane waves. This simplifying assumption allows adapting various methods from computer graphics. They require a modest numerical expense and neglect some acoustical effects, like diffraction. Computational methods use techniques from numerical mathematics to solve the acoustic wave equation directly. They are numerically expensive, but they model the acoustical effects correctly.

Rabenstein and Schetelig (1998) have presented a numerical method for the simulation of dynamic wave propagation in three space coordinates. A multidimensional wave digital filter algorithm is implemented which is suitable for the simulation of 3D sound propagation in enclosures with complex geometries. The algorithm is capable of modeling free space propagation, reflections at boundaries with arbitrary reflection factors and diffraction at openings. Conventional methods for the assessment of room acoustics model the sound propagation in analogy to the propagation of light. More advanced computational methods rely on the numerical solution of the wave equation.

The propagation of sound waves in air is governed by two basic relations for the acoustic pressure $p(x,t)$ and the acoustic fluid velocity vector $v(x,t)$. Pressure p and velocity v denote small-amplitude acoustic signals, which depend on time t and the space vector x. These basic relations are the equation of motion and the equation of continuity. Under reasonable simplifications, they are given by,

$$\rho_0 \frac{\partial}{\partial t} v(x,t) + grad\, p(x,t) = 0 \tag{2.11}$$

$$\frac{\partial}{\partial t} p(x,t) + \rho_0 c^2 div\, v(x,t) = 0 \tag{2.12}$$

where ρ_0 is the static density of the air and c is the speed of sound. Equations (2.11) and (2.12) describe free-space propagation only. They have to be complemented by boundary conditions when finite enclosures are present. The corresponding conditions can be stated in terms of the reaction factors or the wall impedances of an enclosure (Rabenstein and Schetelig, 1998).

The method proposed in Rabenstein and Zayati (1999) is also based on multidimensional wave digital principles. It allows a physically exact numerical modeling of the relevant acoustical effects and yields robust algorithms. Unlike previous methods (Rabenstein and Schetelig, 1998), Rabenstein and Zayati (1999) have straightaway derived an algorithm from the basic laws of physics applied to a multidimensional discrete state space formulation of dynamic wave propagation in three

14

space coordinates, without making use of multidimensional network theory and complex analysis.

For the cases where exact solution is difficult to obtain, which is often the case while modeling non-linear physical phenomenon, the asymptotic solution of systems of linear partial differential equations is preferred, such as discussed in Adam (1982). It obtains the asymptotic functional form of the solution of a scalar wave equation with constant coefficients. In Kagawa et al. (1992), an approximate wave equation which can describe the effect of diffraction with reasonable accuracy, is derived based on the equations of fluid dynamics. Only the generation of the second harmonic wave is considered, with the higher order harmonics being neglected.

Morgül (2002) has discussed the stabilization of the wave equation in a bounded domain by means of a dynamic boundary control law. The transfer function of the controller may contain simple poles on the imaginary axis. This type of controller is proposed for the asymptotic stabilization of the wave equation.

2.2 Sound Speed Measurement

Sound speed being a critical unknown in the present work that is both estimated through model and measured by experiment, various methods exercised in past works for sound velocity measurement are reviewed.

The properties of a fluid are normally determined using invasive methods. These methods may lead to possibly contaminating or consuming the sample. When only very small amounts of a valuable sample exist, noninvasive measurement methods are preferred (Srinivasan et al., 2009). The instrument most commonly used in the fruit industry for the determination of fruit firmness is the hand-held penetrometer. Such instruments measure the peak force required to plunge a cylinder of known diameter to a given distance into the fruit pulp, usually with the skin removed. This method though simple and is universally accepted. However, the method is destructive. A non-destructive method of food firmness evaluation has obvious applications (Subedi and Walsh, 2009).

To measure sound velocity, Subedi and Walsh (2009) have used an instrument that consists of a 'hand gun' in which a trigger releases a spring-loaded plastic plunger. The plunger moves along the barrel of the gun to lightly tap the fruit surface. This action produces a vibration in the fruit, which is detected by two unidirectional microphones located at the end of the gun barrel, but at different distances from the point of plunger impact. The time difference in maximum signal detection by the two microphones is used

to calculate sound velocity. The 'hand gun' is held such that a spacer peg at the end of the barrel touches the fruit, and the two microphones are in the near vicinity of the fruit surface. Averages of five measurements are taken for each reading. The destructive penetrometer reading, taken from the same position on the fruit is used for comparison (Subedi and Walsh, 2009).

The acoustic wave properties are the basis for several noninvasive qualitative measurement techniques. Conventional methods determine the sound speed by measuring the medium path length propagated by a pulsed wave and the corresponding time-of-flight (TOF). TOF measurements are valuable in the estimation of distances, displacements and velocities of moving objects, phase differences of wave pulses, temperature of the atmosphere, and so on. Srinivasan et al. (2009) has recommended a technique to measure an index descriptive of the stage of ripening of fruits for the sound velocity measurement.

In the setup, two microphones are spatially separated by a known distance of one meter, collinear with an acoustic source, as shown in Fig. 2.2. The distance is compared with the value obtained from the product of time-of-flight and acoustic velocity $\sqrt{\gamma RT}$, where, γ is the ratio of specific heats, R is the specific gas constant of the intervening air medium, and T is the absolute temperature in K. The signals for time-of-flight calculations are simultaneously acquired from two ¼ inch pre-polarized condenser microphones. One microphone is placed at a distance of 0.1 m from the source and the other is placed at a distance of 1 m from the location of first microphone.
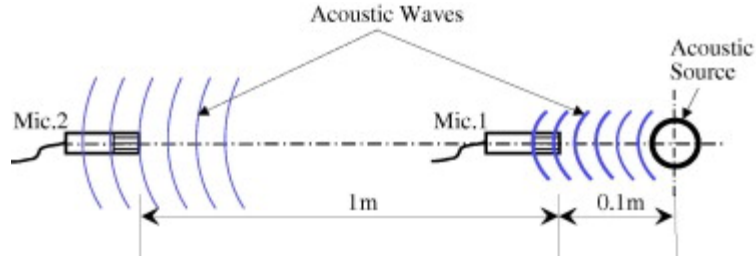
*Fig. 2.2 **Schematic of setup for time-of-flight measurement used in Srinivasan et al. (2009)***

The time-of-flight analyses are performed using both unfiltered and filtered data. In the first case, microphone data are analyzed without any filters; such data are termed as "unfiltered" cases. In the second case, band-pass filtering of the data around the prominent frequency is carried out using an analog filter. These data sets are termed as "filtered" data.

For continuous signals, the cross-correlation function can be defined as:

$$R_{xy}(\tau) = \int_0^\infty x(t)y(t-\tau)dt \qquad (2.13)$$

where τ is the lag parameter, which in the present case is the time-of-flight between the two microphone signals $x(t)$ and $y(t)$. The value of τ which corresponds to the peak value of the cross-correlation function R_{xy}, gives the most probable estimate for the time-of-flight of the acoustic signal, between the two microphone locations. For data that is

acquired digitally, the discrete form of the correlation function between the signals $x(t)$

and $y(t)$ is given by the following equation:

$$R_{xy}(m) = \sum_{n=0}^{N-m-1} x_{n+m} y_n \qquad m \geq 0$$

$$R_{xy}(m) = R_{xy}(-m) \qquad m < 0 \tag{2.14}$$

where N is the length of vectors x and y. Thus, the R_{xy} vector has length $2N - 1$. The

lag index corresponding to the maximum value of this vector $m = m \cdot R_{max}$ is reckoned

as the index corresponding to the time delay for the acoustic signal to travel between the

two microphone locations. The time-of-flight and the distance of travel are then

calculated as $TOF = m \cdot R_{max} \cdot \Delta t$ and $L = TOF \times \sqrt{\gamma RT}$, respectively, where Δt is the

reciprocal of the sampling frequency. The estimated error in the distance (ΔL) based on

TOF is calculated as follows:

$$\Delta L \ (\%) = \left(\frac{1}{L_{actual}}\right) \sqrt{\sum_{i=1}^{n} \frac{(L_i - L_{actual})^2}{n}} \times 100 \tag{2.15}$$

where, L_i is the estimated distance from TOF measurements for the ith trial, L_{actual} is the

actual distance between the two microphones (1 m) and n is the number of trials of the

measurement, ranging from 10 to 60, for the various experiments reported in the paper.

Srinivasan et al. (2009) have investigated the effects of sound source on time-of-flight measurements and demonstrates that signal filtering around the dominant spectral components of sound signal enhances the accuracy of TOF measurement. However, this approach puts a restriction in terms of a priori information regarding the propagation path length.

Among other methods such as Pereira et al. (2000), the wave speed is determined without the need of the path length. A transmitting transducer sends a pulsed wave into the medium (constant wave speed along the beam axis) and the backscattered signal is collected by a hydrophone placed at two distinct positions near the transmitted beam. The time-delay profile, between gated windows of the two RF-signals received by the hydrophone, is determined using a cross-correlation method. Also, a theoretical time-delay profile is determined considering the wave speed as a parameter. The measured wave speed is obtained upon minimization of the RMS error between theoretical and experimental time-delay profiles. The work of Dameron (1979) is based on the de-convolution of backscattered signals, to determine the wave speed without the need of the propagation path length.

Levy et al. (2006) has given a way to measure sound speed dispersion. This research uses continuous wave dispersion as the index for soft tissue characterization by ultrasound for which sound velocity measurement plays a critical role. It combines a short pulse transmission followed by a long burst comprised of two frequencies, one

being double that of the other. The method allows the determination of the speed of sound dispersion using a single transmission.

A non-contact and non-invasive optical probing of deflected laser beams because of normally incident degenerated shock waves is used for sound speed measurement in the method proposed by Jung et al. (2005). In this work the shock waves from an exploding wire are degenerated to ordinary sound waves at a distance exceeding 0.23 m in air from the source.

Ophir et al. (1991) have proposed an experiment where a TACT transducer imparts an accurate trans-axial compression to the tissue and the corresponding change in the arrival time of an echo at a range of interest is measured. This procedure results in a biased speed estimate whose value is range dependent. The theoretical function is fitted to the experimental estimates, from which the unbiased sound speed is then computed.

Guillon et al. (1998) have pointed out the frequency dependency of sound wave speed in terms of phase velocity. The inter-receiver broadband estimation technique utilizes the usable frequency bandwidths of the transducers to demonstrate the agreement of the results obtained by two sets of transducers within the experimental uncertainty. The Biot theory of propagation of sound in porous media is a Lagrangian semi-phenomenological theory, based on a few hypotheses:

1) The elementary volume must be much smaller than acoustical wavelength and larger than maximal size of the pores,

2) The solid frame is elastic and isotropic,

3) The fluid is continuous and saturates the medium uniformly.

The curve obtained from experiment data presents a decrease of the sound speed with increasing frequency, whereas the Biot theory predicts a flat curve in the used frequency range. The conclusions suggest that the Biot theory cannot be directly used for the chosen medium (water saturated sand). The paper also suggests performing measurements at lower frequencies (Guillon et al., 1998).

2.3 Finite-order Models

Despite the fact that almost none of the key theorems and tools of model theory, such as the completeness theorem and the compactness theorem, apply to finite structures, finite models are popular. As per Fagin (1993), the reason for popularity in the computer era is the connection with databases and logic programming, for the fact that it is possible to think of a database as simply a finite structure.

The finite-order models are true representation of system dynamics within tolerance value. Yet, in terms of stability, models may have a varied performance because of local truncation error, in the case of the finite difference method (Ataie-Ashtiani et al., 1999), discretization error, in the case of the finite element method (Schmidt et al., 2009

22

and Kagawa et al., 1998) and un-modeled dynamics in the case of the reduced order method (Yae and Inman, 1991). The major drawback of an explicit time integration method is its conditional stability. This means there is a critical time step which must not be exceeded in the analysis. The magnitude of this critical time step depends on the largest natural frequency of the linearized system (Fagin, 1993).

In the dynamic modeling of a system, finite element analysis (Yae and Inman, 1991) employs reduction techniques that remove some of the 'insignificant' physical coordinates, *i.e.* reduced degrees of freedom at a node point. Despite such reduction, the resultant model is still too large for control design. This allows further reduction as is frequently done in control design by approximating a large dynamical system with a fewer number of state variables. A problem, however, arises because a model usually undergoes, before being reduced, some form of coordinate transformations that destroy the physical meanings of the states. To correct such a solution, Yae and Inman (1991) have given a method that expresses a reduced model in terms of a subset of the original states. The proposed method starts with a dynamic model that is originated and reduced in finite element analysis. The model is then converted to a state-space form and reduced further by the internal balancing method. At this stage, being in the balanced coordinate system, the states in the reduced model have no apparent resemblance to those of the original model. The model reduction procedure used is illustrated as follows,

$$(A, B, C, x) \overset{P}{\Leftrightarrow} \left(\overline{A}, \overline{B}, \overline{C}, \overline{x} \right) \qquad\qquad 2n$$

23

After model reduction,

$$(A_r, B_r \ C_r \ x_r) \overset{P_r}{\Leftrightarrow} \left(\overline{A_r}, \overline{B_r}, \overline{C_r}, \overline{x_r}\right) \qquad 2n - k$$

where, x consists of $2n - k$ elements of x and $\overline{x_r}$ consists of $2n - k$ elements of $\overline{x}$. So, k represents the number of states discarded in model reduction. In addition, $(A_r, B_r \ C_r \ x)$ is Original State Space representation of the system and $\left(\overline{A_r}, \overline{B_r}, \overline{C_r}, \overline{x}\right)$ is balanced system.

The model reduction method used here is based on the internal balancing method and is developed to represent the reduced model with a subset of the states in the original model. The proposed method takes a finite element model that is reduced by Guyan's reduction, converts it into the state space form, and applies the balanced model reduction. Through another transformation that is derived from the deleted states in reduction, the model is filly expressed by a subset of the original states. The method thereby provides a clear, physical relationship between the states in the reduced model and those in the original model. The states in the reduced model are selected directly from the original states, thus retaining the same physical meanings as in the original model (Yae and Inman, 1991).

Kagawa et al. (1998) have proposed a transmission-line matrix (TLM) model which is a space and time discretization method for computation of electromagnetic fields, applied in simulating sound wave propagation. Two dimensional and axisymmetric TLM elements are first developed for both linear and non-

24

linear sound field applications. Analogous and digital equivalent circuit expressions are developed for the two dimensional TLM elements. Their Transfer Function characteristics are examined theoretically. A digital filter expression equivalent to the two-dimensional TLM element is also developed.

To check the validity of the TLM modeling as proposed, a simple problem of plane wave propagation in an acoustic tube is first examined. The tube is modeled as a series of TLM elements as shown in Fig. 2.3a. In the model, one end of the tube is terminated by the non-reflective boundary while another end is driven by the velocity source of a train of pulses of single-shot sine envelope with wavelength λ and the side walls are considered to be rigid. The element length Δl is chosen to be from $\frac{\lambda}{10}$ to $\frac{\lambda}{40}$ in order to check the accuracy.

Fig. 2.3b shows the pressure waveforms as the sine wave with the amplitude of 0.5 Pa propagates until the time 12T, where T is the period of the sine wave. It is shown that the wave propagates at the speed of $\frac{c_0}{\sqrt{2}}$. The discretization errors become pronounced as the wave propagates. It decreases as the element length becomes shorter. As with other numerical methods, the TLM method also requires the fine mesh for an accurate solution, to reduce fluctuation errors or discretization errors. A two dimensional Gaussian Filtering is given to improve upon discretization error. Improved results after filtering can be seen in Fig. 2.3c.

Rigid wall

Velocity drive → Non-reflective boundary

|← 10λ →|

(a) Acoustic tube model

3 T — 0.5 Pa ↕ — ↔ 2λ

6 T

9 T

12 T

Δl= λ/10   Δl= λ/20   Δl= λ/40

(b) TLM solutions without filtering

3 T — 0.5 Pa ↕ — ↔ 2λ

6 T

9 T

12 T

Δl= λ/10   Δl= λ/20   Δl= λ/40

(c) TLM solutions with filtering
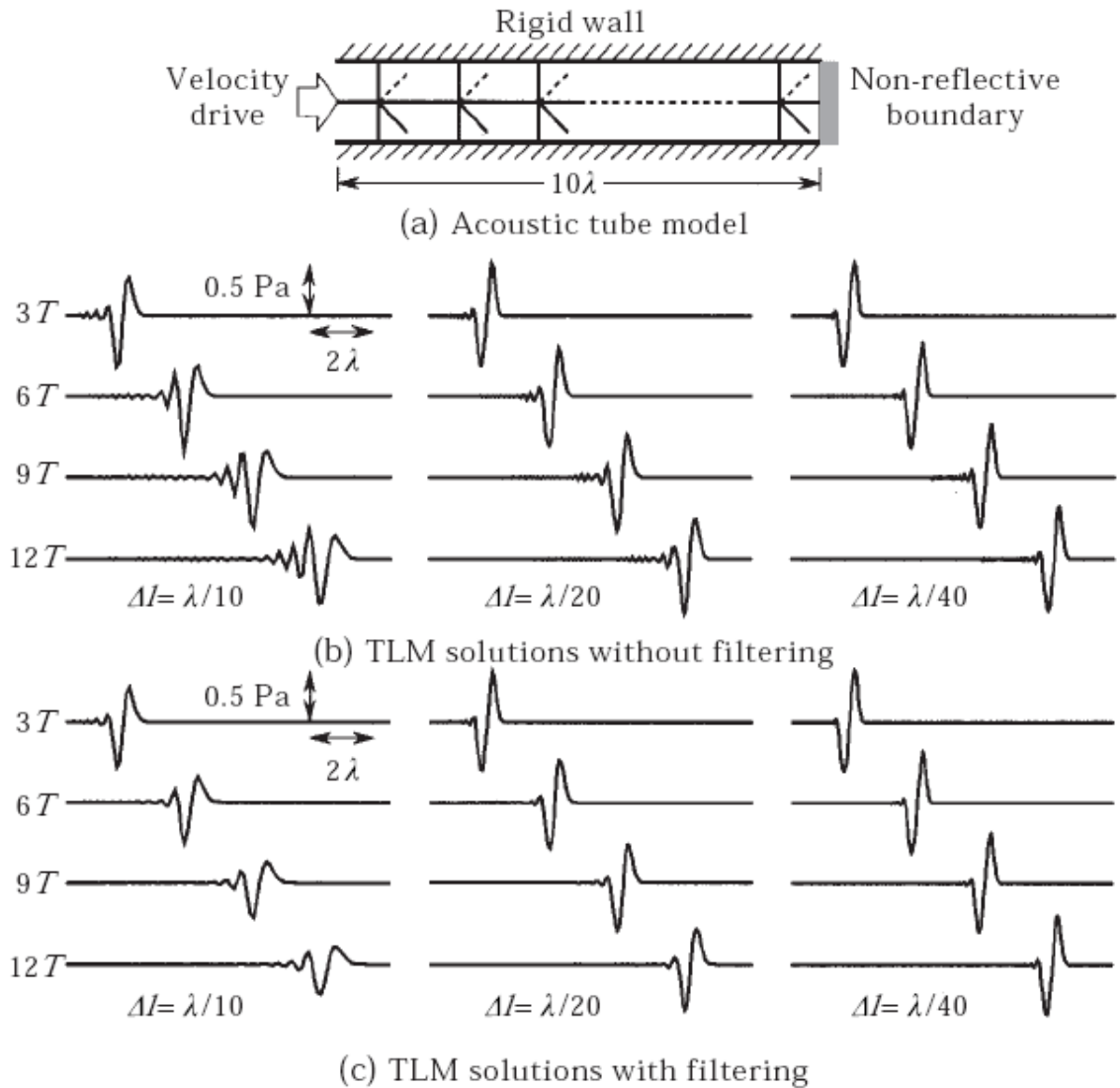
Fig. 2.3 Acoustic tube model and TLM solutions for a single-shot sine wave and solutions with the filtering capability (Kagawa et al., 1998)

Some simulated examples are then demonstrated for the two-dimensional and axisymmetric sound wave propagation problems. The results are compared with finite element solutions.

2.4 Modeling Error Correction

The Finite-order model output is not directly an exact match to the true response. The phase correction offered to compensate for the modeling error need to be function of the Input Frequency and Order of the model. Shaw et al. (2010) have proposed a highly problem dependent technique for estimating error in a finite model. Whether or not it becomes more widely used as a tool for assessing the reliability of engineering computations seems to depend on whether some form of a priori assessment of its own reliability becomes available, as well as on whether codes evolve to cope with the inherent complexity of the implementation.

Fawcett (1985) has used non-linear least squares theory and the Levenberg-Marquardt method of solution, to invert in the time domain, full wave field observations for two dimensional acoustic velocity variations. The singular value decomposition of the linearized problems' Jacobian allows for the determination of the portions of parameter space that have been well or poorly resolved.

From an analogy in reduced order controller design, a well-known and often used technique for obtaining a simpler controller is model order reduction. One can either approximate the system or design a low order controller for the simple plant model, or one can design a controller for the full order plant and reduce the order of the resulting controller. The problem of conditional stability still holds in this case. With Robust design techniques, Al-Saggaf and Franklin (1987) derive an L^{∞} bound on model error for

a method of order reduction of discrete linear multivariable systems based on balancing. Yoneya et al. (2002) has proposed a method to test whether the model error of a system is in a pre-determined bound given with a weight function using frequency response data of the system obtained by experiment. The model error bound problem specified in Al-Saggaf and Franklin (1987) is converted to a stability problem of a set of systems in Yoneya et al. (2002) and the stability test is performed to obtain model with minimum modeling error.

The exact and unique solution to the design of phase-lead and phase-lag compensation proposed by Wang (2003) has provided an alternate approach towards formulation for modeling error correction. With the desired gains in the magnitude and phase known, a priori at a given frequency, a lag compensator can be designed thus offering implicit phase correction to the model output.

A major motivation of research in identification for control (Reinelt et al., 1999) is to achieve robust stability of a real system. Finite-order models cursed by the drawback of conditional stability will be benefitted with robust design technique. Thus, it is customary to identify not only a nominal model, but also an uncertainty set, i.e. a set of models to be considered in the design process. For example, if non-parametric design methods are adopted, it is natural to look for uncertainty regions in the Bode plot. A further consequence of the control objective is that it might be sufficient to estimate the real plant well up to a certain frequency (somewhere in the region of the cross over) and to tolerate a larger uncertainty for higher frequencies. Reinelt et al. (1999) has elaborated

28

on three recent concepts delivering a nominal model and the related uncertainty in the Bode plot: the Stochastic Embedding technique, Set Membership Identification and Model Error Modeling. The first is a frequency domain method, while the other two work in the time domain. The main features of these methods are highlighted and reliable nominal models along with acceptable related uncertainties were obtained in all three cases.

Alvarez-Ramfrez and Sueraz (2000) and Sun et al. (1994) have shown how modeling error compensation techniques can be extended for the case of single input/single output, minimum-phase, linear time variant systems. The new element in these papers is an idea to achieve robust control via high-gain observer alone without explicit design of high-gain feedback. Instead of designing a robust state feedback to dominate the uncertain term, the uncertain term is viewed as an extra state that is estimated using a high-gain observer.

Among the computational methods, Parameswaran and Raol (1994) have presented algorithms for estimation of deterministic model error in the assumed models of nonlinear discrete and continuous time systems. The explicit model error time histories are parameterized using least squares method. According to Kolodziej and Mook (2011), the parameterized models relative to the true model explain the deterministic deficiency in the chosen models, in the sense of minimum model error. The algorithms have appealing features of extended Kalman filter. The numerical simulation results are obtained by implementing the algorithms in MATLAB.

Ljung and Lei (1995) have explained what a typical model validation test implies in terms of the model error, expressed in the frequency domain. Discussion is made upon the principles by which confidence in a model can be reached through validation techniques and about how the distance to a "true" description can be estimated. Research in Ljung and Lei (1995) emphasizes how the typical model validation procedure gives a direct measure of the model error of the model test, without referring to its ensemble properties.

CHAPTER 3

SYSTEM DEFINITION AND MODELING

As described in Section 1.1, the propagation medium is specified in terms of its density, stiffness and frictional properties. In order to obtain a finite-order model for a propagation medium of finite length, the medium is divided into N elements (Fig. 3.1). The first element in the network is under direct action of an external force and the rightmost end next to the final element is held fixed.
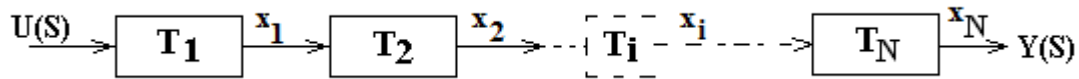
*Fig. 3.1 **Finite-order representation of medium***

For this research, propagation medium is assumed to be homogeneous. Each intermediate element represents a mass-spring-damper unit connected to the two other similar units as shown in Fig. 3.2, forming a Kelvin-Voigt network model. It forms a single-input single-output system with external force as input $U(s)$ and the reactive force experienced by the fixed end as system output $Y(s)$.

Each element is described in terms of displacements it undergoes along the direction of wave propagation through the medium. Each element is assumed to have mass m, stiffness k and friction or damping coefficient b. Every model coefficient consists of these three constants of a given homogenous medium.

This chapter elaborates on the derivation of the system transfer function representing wave propagation in terms of material properties *m, b* and *k*. A method is explained to derive a series of transfer functions by a recursive technique and consequently to obtain a complete system model. This method, however, results in a circumstantially stable system because of numerical artifacts. An alternate method is proposed in which, the system is reduced to a subset of original states to obtain a simpler system. Finally the mapping process is used to obtain the true system model from the reduced sub-system.

3.1 System Transfer Function

The system here is a propagation medium of finite length divided in a known number of elements N. Transfer function derivation starts with using the boundary condition to obtain the transfer function for the final element in the network. Then, transfer functions are obtained for other elements by a recursive method explained ahead. The consecutive substitution of transfer functions gives the desired system model.

3.1.1 Recursive Form of Series Transfer Functions

Each element in the system is represented by a transfer function between two consecutive state variables x_i $(i = 2, 3, ..., N)$ or between a state variable x_1 and input u.

$$T_1 = \frac{X_1(s)}{U(s)}; \ T_2 = \frac{X_2(s)}{X_1(s)}; \ T_3 = \frac{X_3(s)}{X_2(s)}; \ ... \ T_i = \frac{X_i(s)}{X_{i-1}(s)}; \ ... \ T_N = \frac{X_N(s)}{X_{N-1}(s)} \qquad (3.1)$$

where $T_i(s)$ $(i = 1,2, ..., N)$ denotes transfer function, $X_i(s)$ $(i = 1,2, ..., N)$ is Laplace transform of x_i displacements of the element $i, U(s)$ is the Laplace transform of external force u and s is the Laplace variable.

The transfer function for each element is obtained from the force balance of the element. Considering the fact that all the elements of a homogenous medium are identical in terms of their properties, a generic index i can be applied to all the non-terminal (all except for the first and the last) elements as shown in Fig.3.2.
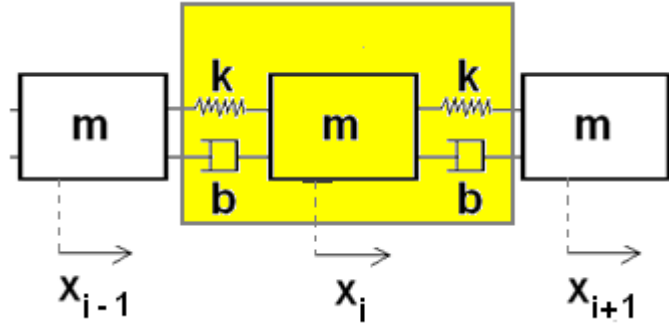
*Fig.3.2 **Illustration of forces acting on a non-terminal element***

Force balance of element i gives,

$$m\frac{d^2x_i}{dt^2} + b\frac{d}{dt}(x_i - x_{i+1}) + k(x_i - x_{i+1}) + b\frac{d}{dt}(x_i - x_{i-1}) + k(x_i - x_{i-1}) = 0$$

Expanding and rearranging gives,

33

$$m\frac{d^2x_i}{dt^2} = -2b\frac{dx_i}{dt} - 2kx_i + b\frac{dx_{i+1}}{dt} + kx_{i+1} + b\frac{dx_{i-1}}{dt} + kx_{i-1} = 0 \quad (3.2)$$

Taking the Laplase transform on both sides of the Eqn. (3.2), we get,

$$ms^2X_i(s) = (bs + k)X_{i-1}(s) - 2(bs + k)X_i(s) + (bs + k)X_{i+1}(s). \quad (3.3)$$

Eqn. (3.3) is generic relation that, by using a given boundry condition, will provide a recursive formula for the element transfer functions.

Since one end of medium is assumed fixed, for $i = N$

$$x_{i+1} = x_{N+1} = 0$$

Applying this to Eqn. (3.3) we get,

$$ms^2X_N(s) = (bs + k)X_{N-1}(s) - 2(bs + k)X_N(s)$$

Then the transfer function for element N is

$$T_N(s) = \frac{X_N(s)}{X_{N-1}(s)} = \frac{bs + k}{ms^2 + 2bs + 2k}.$$

This is an expression that will appear in the system transfer functions. It is the transfer function for the Nth or the final element in the discretized medium. For convenience, we denote it by G, i.e.

$$G(s) = T_N(s) = \frac{X_N(s)}{X_{N-1}(s)} = \frac{bs + k}{ms^2 + 2bs + 2k} \qquad (3.5)$$

or,

$$X_N(s) = GX_{N-1}(s) \qquad (3.6)$$

Now we consider the penultimate state variable x_{N-1} for which Eqn. (3.3) becomes

$$ms^2 X_{N-1}(s) = (bs + k)X_{N-2}(s) - 2(bs + k)X_{N-1}(s) + (bs + k)X_N(s)$$

$$\qquad \cdots (3.7)$$

In Eqn. (3.7), three state variables are present, one of which can be eliminated by using Eqn. (3.6). So,

$$ms^2 X_{N-1}(s) = (bs + k)X_{N-2}(s) - 2(bs + k)X_{N-1}(s) + (bs + k)GX_{N-1}(s)$$

Eqn. (3.8) relates the current state x_{N-1} to the next state x_{N-2} only.

The transfer function for element $N - 1$ is,

$$T_{N-1}(s) = \frac{X_{N-1}(s)}{X_{N-2}(s)} = \frac{G}{1 - G^2} \qquad (3.8)$$

where Eqn. (3.5) has been used.

Similarly, the transfer function $T_{N-2}(s)$ for $N-2^{th}$ element is,

$$T_{N-2}(s) = \frac{G(1-G^2)}{1-2G^2} = \frac{G}{1-G\left(\frac{G}{1-G^2}\right)} = \frac{G}{1-GT_{N-1}(s)}$$

By repeating the procedure for the next elements, a recursive relationship for the series of transfer functions is obtained which is as follows:

$$T_i(s) = \frac{X_i(s)}{X_{i-1}(s)} = \frac{G}{1-GT_{i+1}(s)} \qquad (i = 2,3, ..., N) \qquad (3.9)$$

As a boundary condition $x_{N+1} = 0$, hence $T_{N+1}(s) = 0$ and by Eqn. (3.9),

$$T_N(s) = \frac{G}{1-0} = G$$

So, Eqn. (3.9) applies to $i = N$ as well.
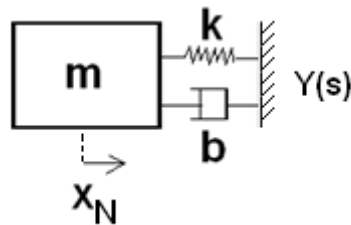
Fig. 3.3 **Output gain between the fixed boundary and the final element**

The model output at the boundary shown in Fig. 3.3, is related to the final element through the relation

36

$$y(t) = b\frac{dx_N}{dt} + kx_N$$

Taking Laplace transform we get

$$\frac{Y(s)}{X_N(s)} = bs + k$$

By multiplying all the element transfer functions and the output gain, the complete transfer function of the entire medium is obtained as,

$$H(s) = \frac{Y(s)}{U(s)} = T_1(s) \cdot T_2(s) \cdot T_3(s) \cdot T_4(s) \cdot \ldots \cdot T_N(s) \cdot \frac{Y(s)}{X_N(s)}$$

$$= T_1(s) \cdot \frac{G^{N-1} \cdot (bs + k)}{\{1 - GT_3(s)\}\{1 - GT_4(s)\} \ldots \{1 - GT_N(s)\}} \qquad (3.10)$$

where

$$G = T_N(s) = \frac{bs + k}{ms^2 + 2bs + 2k},$$

$$T_i(s) = \frac{G}{1 - GT_{i+1}(s)} \ldots for\ i = 2, \ldots N, \qquad and$$

$$T_1(s) = \frac{X_1(s)}{U(s)}.$$

Equation (3.10) gives the required system model for a given homogenous medium. The only factor that remains unknown is the transfer function for the first element which is directly subjected to external force.

3.1.2 Transfer Function for the first element $T_1(s)$

For the first element ($i = 1$), the forces are shown in Fig.3.3,
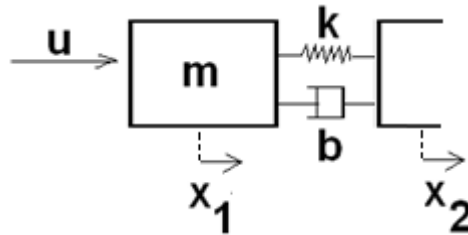
Fig. 3.4 **Free Body Diagram of the First Element**

Force balance for the first element gives,

$$m\frac{d^2x_1}{dt^2} + b\left\{\frac{dx_1}{dt} - \frac{dx_2}{dt}\right\} + k\{x_1 - x_2\} = u(t)$$

Taking Laplace Transform we get,

$$ms^2X_1(s) + bs\{X_1(s) - X_2(s)\} + k\{X_1(s) - X_2(s)\} = U(s)$$

Therefore, by using $X_2(s) = T_2(s) \cdot X_1(s)$,

38

$$[ms^2 + bs\{1 - T_2(s)\} + k\{1 - T_2(s)\}] \cdot X_1(s) = U(s)$$

Hence,

$$T_1(s) = \frac{X_1(s)}{U(s)} = \frac{1}{ms^2 + \{1 - T_2(s)\}bs + \{1 - T_2(s)\}k} \tag{3.11}$$

Substituting Eqn. (3.11) in Eqn. (3.10) for the total transfer function, we get,

$$H(s) = \frac{G^{N-1} \cdot (bs + k)}{\{ms^2 + (bs + k)(1 - T_2(s))\} \cdot \{1 - GT_3(s)\} \cdot \{1 - GT_4(s)\} \dots \{1 - GT_N(s)\}}$$

$$\dots (3.12)$$

3.1.3 Illustrative Example

Consider the medium discretized into 5 elements (*i.e., N* $=$ 5) shown in Fig. 3.4
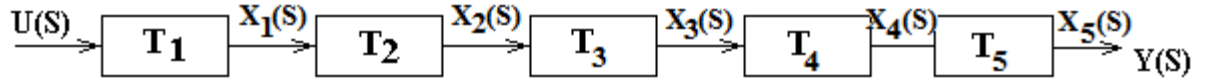
*Fig. 3.5 **Medium discretized into five elements***

Applying Eqn. (3.9), the following transfer functions are obtained. All poles and zeros are determined by medium parameters m, b, and k.

$$T_5(s) = G$$

$$T_4(s) = \frac{G}{1 - G^2}$$

$$T_3(s) = \frac{G(1 - G^2)}{1 - 2G^2}$$

$$T_2(s) = \frac{G(1 - 2G^2)}{1 - 3G^2 + G^4}$$

where,

$$G = \frac{bs + k}{ms^2 + 2bs + 2k}$$

The total transfer function $H(s)$ is,

$$H(s) = \frac{Y(s)}{U(s)} = T_1(s) \cdot T_2(s) \cdot T_3(s) \cdot T_4(s) \cdot T_5(s) \cdot (bs + k)$$

$$= T_1(s) \cdot \frac{G^4}{1 - 3G^2 + G^4} \cdot (bs + k)$$

where

$$T_1(s) = \frac{1}{ms^2 + (1 - T_2(s))(bs + k)}.$$

Per Eqn. (3.12) the total transfer function $H(s)$ for $N = 5$ is,

$$H(s) = \frac{G^4 \cdot (bs + k)}{\left\{ms^2 + \left(1 - \frac{G(1 - 2G^2)}{1 - 3G^2 + G^4}\right)(bs + k)\right\}(1 - 3G^2 + G^4)}.$$

3.1.4 *Frequency Response*

Dependency of the system on input frequency is illustrated in Fig. 3.6 by the bode plot of transfer function $H(s)$ using Eqn. (3.12) The plot shows the variations in magnitude and phase as function of input frequency for different orders of the model.
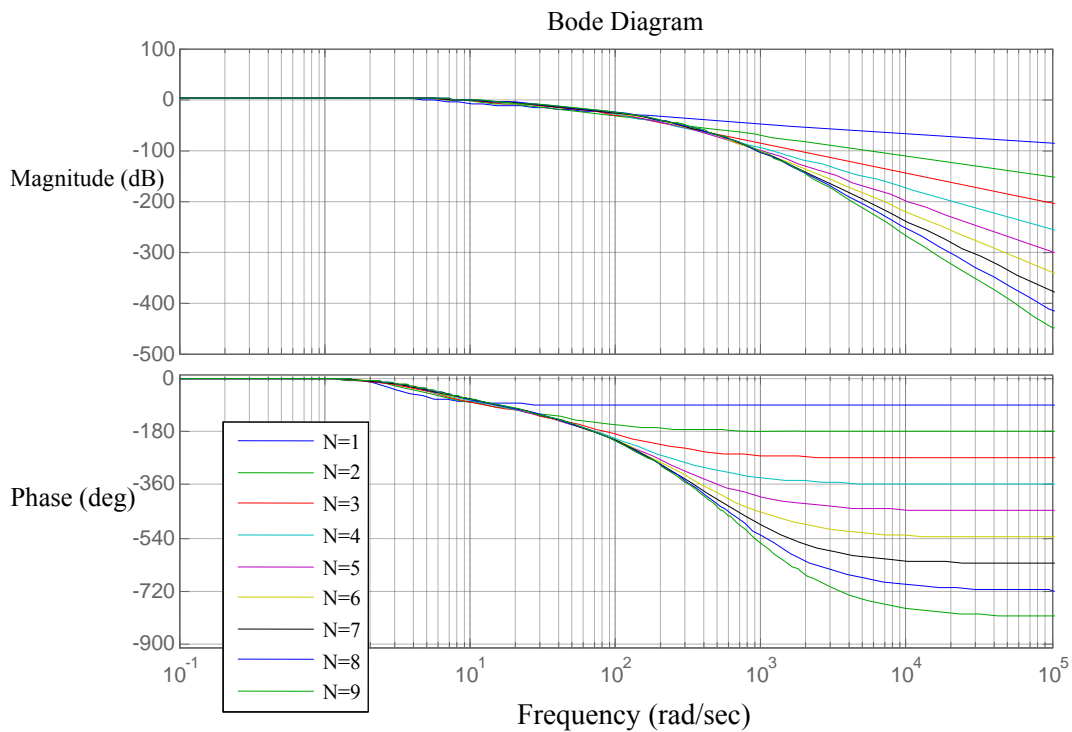
Fig. 3.6 ***Bode Plot for N=1 to 9***

The magnitude plot indicates that the corner frequency increases with increasing order of the model $2N$. This results from the leftward shifting of poles in the s-plane. For each unit increment in N, the phase lag increases by 90° at high frequencies.

The phase lag continues to increase with increasing order of the model, but the increases occur at higher and higher frequencies. Since a given phase lag (say, 90°) implies less time delay at higher frequencies, the added time delay by the added phase lag diminishes. As a result, the time delay given by infinite order model (N → ∞) at infinite frequency should converge. For a given frequency, the added phase angle by increasing N diminishes resulting in an upper bound for the phase lag.

In other words, for an infinitely high order model, phase lag and thus the time delay will converge. The data acquired for varied frequencies are shown in Table 4.1 of the next chapter.

Time domain simulation of the transfer function $H(s)$ obtained from Eqn. (3.12), by using built-in MATLAB routines, is stable only for limited order of the model. Flowchart in Fig. 3.7 illustrates the recursive transfer function method used to generate $H(s)$ for simulation purpose. The simulation turns unstable for higher values of N. Fig. 3.8 shows that, with increasing order of the model, poles shift leftwards and farther from the origin.

Fig. 3.7 **Flowchart for the recursive transfer function method**

```
                          ( Start )
                              |
                              v
              +-------------------------------+
              | Calculate element properties  |
              |     from bulk properties      |
              +-------------------------------+
                              |
                              v
              +-------------------------------+
              |    Generate sinusoidal        |
              |           input               |
              +-------------------------------+
                              |
                              v
              /-------------------------------/
             /     Import experiment          /
            /    data from external file     /
           /-------------------------------/
                              |
                              v
```

Define end element TF,

$$T_N(s) = G = \frac{bs+k}{ms^2 + 2bs + 2k}$$

Evaluate TF for next element using,

$$T_i(s) = \frac{G}{1 - GT_{i+1}(s)}$$

Is this the second element?

No

Yes

Evaluate

$$T_1(s) = \frac{1}{ms^2 + (1 - T_2(s))bs + (1 - T_2(s))k}$$

$$H(s) = T_1(s) \cdot T_2(s) \cdot \ldots \cdot T_N(s) \cdot (bs + k)$$

Return $H(s)$
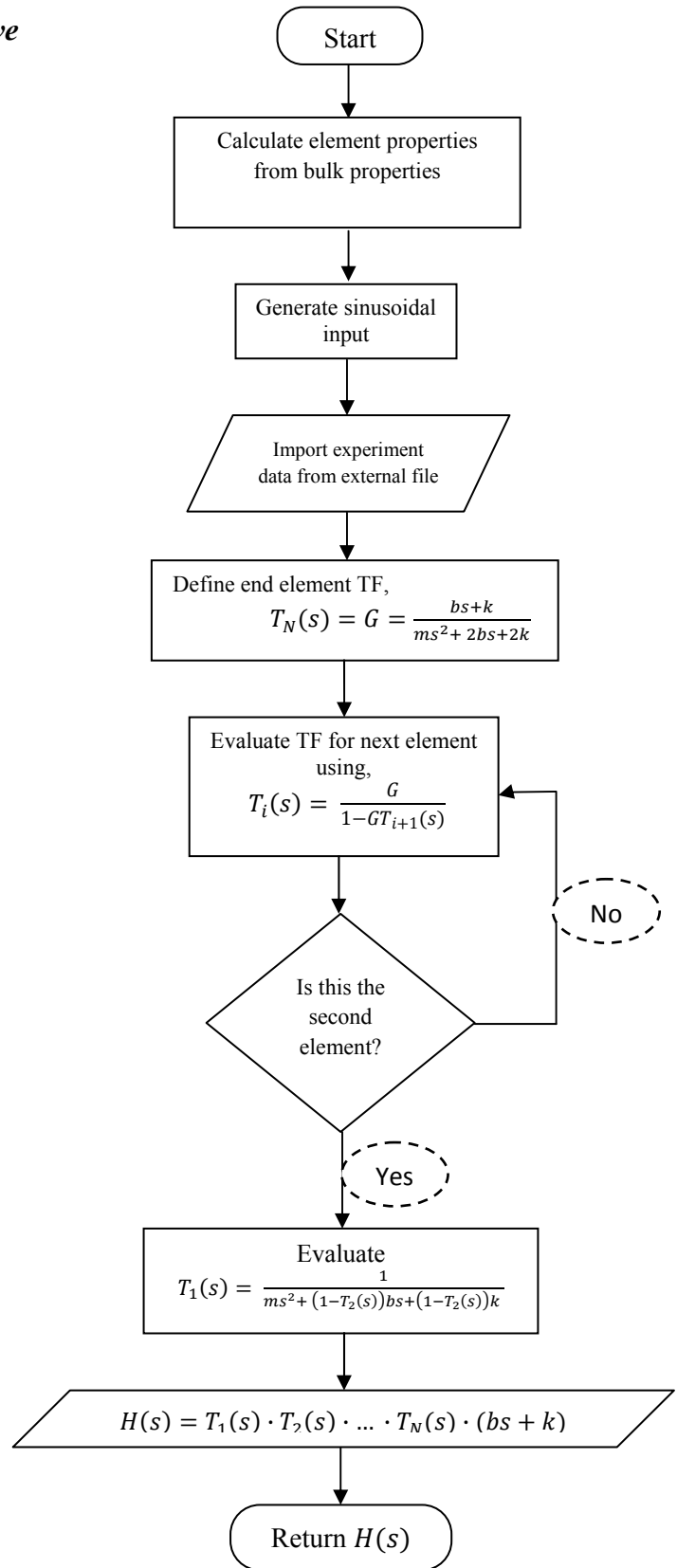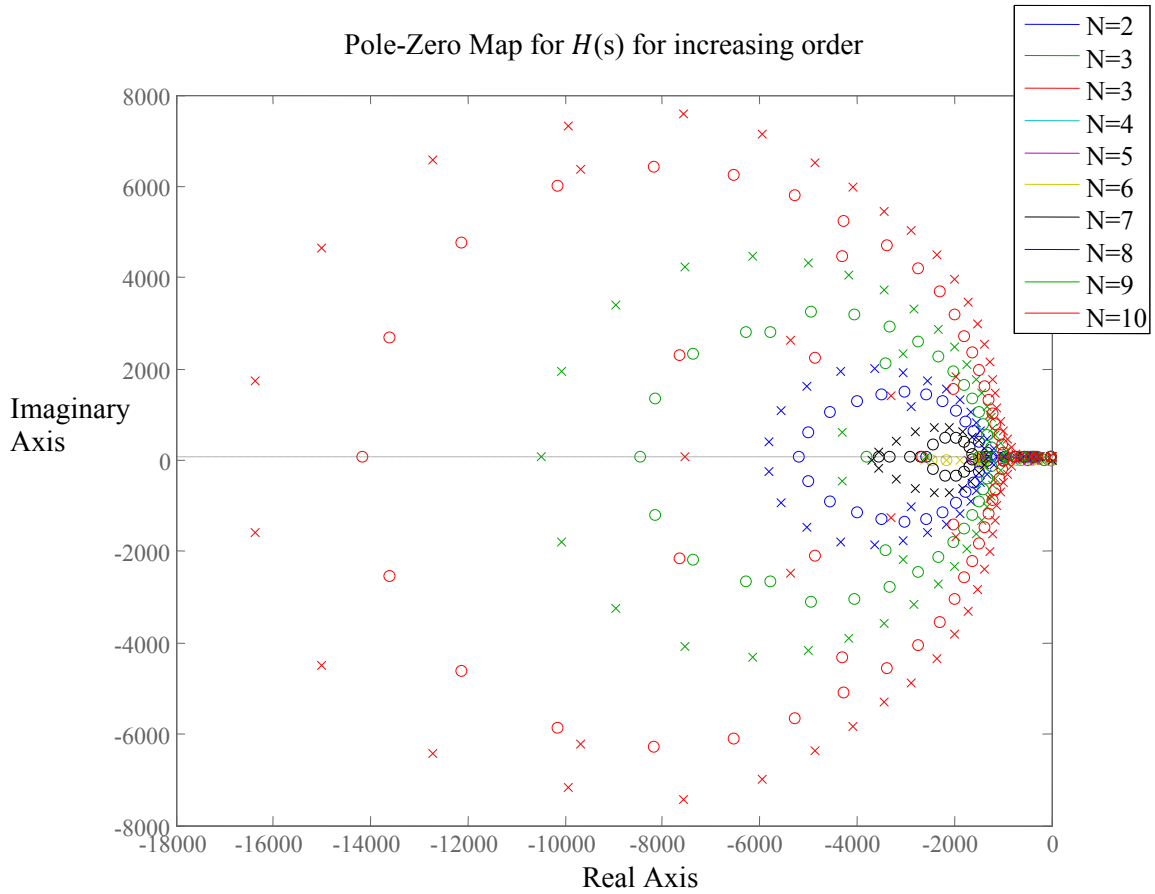
Fig. 3.8 ***Pole-zero plots for the transfer function derived by the recursive method***

For a set of chosen property values (viz. *m*, *b* and *k*), the highest order for which the solution remains stable turns out to be 10. The maximum order for stability varies with medium properties.

Figure 3.9 shows a detailed view of the poles and zeros that lie in the right half of the *s* plane when $N = 11$, resulting in instability. These values are numerical artifacts, which are absent when the equation is solved analytically.

45
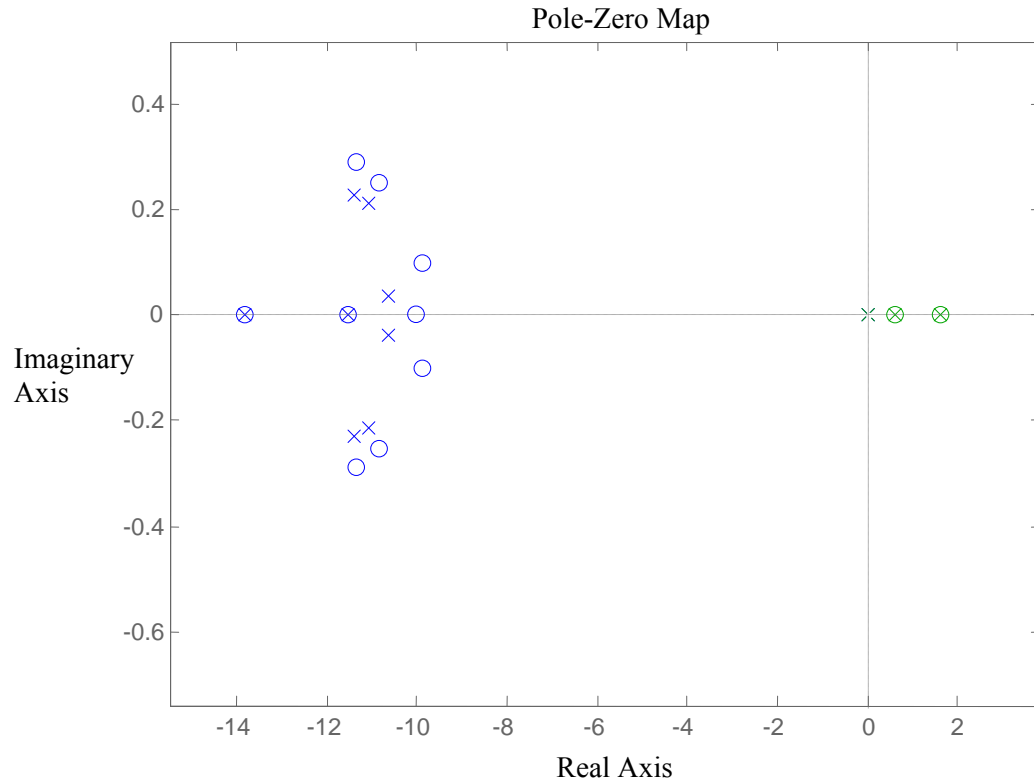
Fig. 3.9 **Magnified view showing numerical artifacts of positive poles, causing simulation instability**

The zero present at $s = -10$ is the only valid zero analytically. The unstable pole and zero pairs are extremely close and they should exactly cancel. The fact that they don't cancel in numerical simulations, causes instability.

3.2 Pattern Method for General Transfer Function

The generic form of transfer function obtained in Eqn. (3.10) is

$$H(s) = T_1(s) \cdot \frac{G^{N-1}}{\{1 - GT_3(s)\}\{1 - GT_4(s)\} \dots \{1 - GT_N(s)\}} \cdot (bs + k)$$

This requires knowledge of the denominator polynomial, which is in fact the numerator of $(1 - GT_3)$ that depends upon the order of the model $2N$.

The numerical issue with evaluating this polynomial through the recursive method is because of the repeated use of the equation, which increases the inaccuracies with increasing order of the model. Fig. 3.9 indicates that the non-cancelling RHS pole-zero pairs result in simulation instability for high number of elements. To find out one polynomial, which is the numerator of the first parenthesis in above the equation, we have to evaluate other $N - 2$ series transfer functions. A method is needed to avoid evaluating all those intermediate transfer functions.

The problem can be overcome by finding an algebraic relation to generate the polynomial coefficients in order to obtain the overall transfer function without numerical recursion.

3.2.1 Coefficient pattern in G

To obtain such a formula, the coefficients of the denominator polynomial, in G for up to $N = 25$ are listed in Table. 3.1. There is an obvious relationship among the coefficients as illustrated in Fig. 3.10.
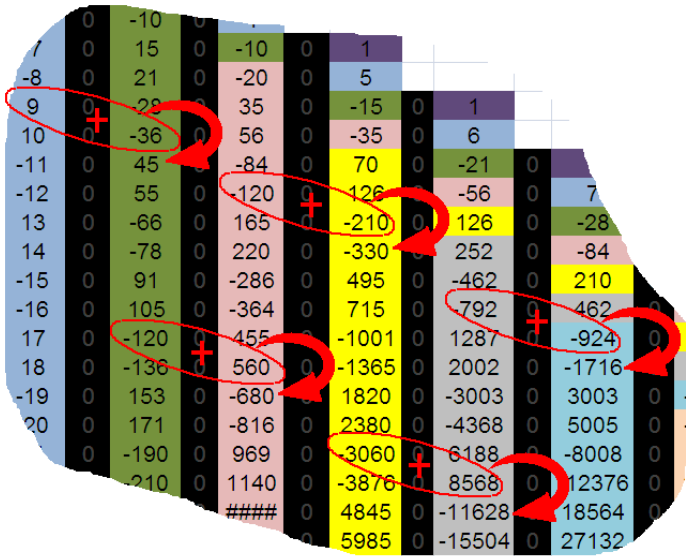
*Fig. 3.10 **Illustration for the pattern: Addition of magnitude of the numbers encircled gives magnitude of the number pointed in next row***

$Coeff(G_N^n)$, symbolizes the coefficient of the G^n term in the denominator polynomial for transfer function $H(s)$ for N elements, as shown in Eqn. (3.12). The sign of each coefficient is determined by its position in the matrix. The final formula obtained is as follows,

$$Coeff(G_N^n) = \left\{(-1)^{\left|\frac{n}{2}\right|+\left|\frac{N}{2}\right|}\right\}.\left\{|Coeff(G_{N-1}^n)| + |Coeff(G_{N-2}^{n-2})|\right\} \qquad (3.13)$$

The pattern is for the denominator polynomial in G, instead of s. After obtaining the desired transfer function $H(G)$ with the formula in Eqn. (3.13), $H(s)$ can be obtained with the relation given by Eqn. (3.5).

With this, we are able to generate the denominator polynomial of $(1 - GT_3)$ in Eqn. (3.10) and thus an entirely algebraic method for composing $H(G)$ is obtained by using Equations (3.10) and (3.13) in the form,

$$H(G) = T_1 \cdot \frac{G^{N-1}}{a_1 G^N + a_2 G^{N-2} + \ldots + a_{N-1} G^2 + a_N} \cdot (bs + k). \qquad (3.15)$$

In the Matlab script titled '*G-pattern*' included in Appendix C, Eqn. (3.13) is used to generate a lower diagonal matrix of size $(N \times n)$. The desired polynomial coefficients $(a_1, a_2, \ldots a_{N-1}, a_N)$ are the non-zero elements of row corresponding to N in coefficient matrix. Table 3.1 shows the coefficient values generated for up to $N = 25$ in a lower diagonal matrix form.

Observe the like-colored blocks along vertical and diagonal. They match.

power of G / N	n→ 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	-1																								
2	-1	0	1																						
3	-1	0	2																						
4	-1	0	-3	0	1																				
5	-1	0	-4	0	3																				
6	-1	0	5	0	-6	0	1																		
7	-1	0	6	0	-10	0	4																		
8	-1	0	-7	0	15	0	-10	0	1																
9	-1	0	-8	0	21	0	-20	0	5																
10	-1	0	9	0	-28	0	35	0	-15	0	1														
11	-1	0	10	0	-36	0	56	0	-35	0	6														
12	-1	0	-11	0	45	0	-84	0	70	0	-21	0	1												
13	-1	0	-12	0	55	0	-120	0	126	0	-56	0	7												
14	-1	0	13	0	-66	0	165	0	-210	0	126	0	-28	0	1										
15	-1	0	14	0	-78	0	220	0	-330	0	252	0	-84	0	8										
16	-1	0	-15	0	91	0	-286	0	495	0	-462	0	210	0	-36	0	1								
17	-1	0	-16	0	105	0	-364	0	715	0	-792	0	462	0	-120	0	9								
18	-1	0	17	0	-120	0	455	0	-1001	0	1287	0	-924	0	330	0	-45	0	1						
19	-1	0	18	0	-136	0	560	0	-1365	0	2002	0	-1716	0	792	0	-165	0	10						
20	-1	0	-19	0	153	0	-680	0	1820	0	-3003	0	3003	0	-1716	0	495	0	-55	0	1				
21	-1	0	-20	0	171	0	-816	0	2380	0	-4368	0	5005	0	-3432	0	1287	0	-220	0	11				
22	-1	0	21	0	-190	0	969	0	-3060	0	6188	0	-8008	0	6435	0	-3003	0	715	0	-66	0	1		
23	-1	0	22	0	-210	0	1140	0	-3876	0	8568	0	-12376	0	11440	0	-6435	0	2002	0	-286	0	12		
24	1	0	-23	0	231	0	-1330	0	4845	0	-11628	0	18564	0	-19448	0	12870	0	-5005	0	1001	0	-78	0	1
25	1	0	-24	0	253	0	-1540	0	5985	0	-15504	0	27132	0	-31824	0	24310	0	-11440	0	3003	0	-364	0	13

Table 3.1. Coefficient matrix for N=1 to 25

Let the denominator polynomial obtained from Eqn. (3.13), which is of the form $a_1 G^N + a_2 G^{N-2} + ... + a_{N-1} G^2 + a_N$, be $P(G)$, so that,

$$H(G) = T_1(s) \cdot \frac{G^{N-1}}{P(G)} \cdot (bs + k) \tag{3.16}$$

where $T_1(s)$ obtained with Eqn. (3.11) and can also be obtained, by replacing $T_2(s)$ with $T_2(G)$ as follows,

$$T_1(s) = \frac{X_1(s)}{U(s)} = \frac{1}{ms^2 - \{2 - T_2(G)\}bs + \{2 - T_2(G)\}k} \tag{3.17}$$

where $T_2(G)$ is obtained with the coefficient matrix as,

$$T_2(G) = \frac{G \cdot N(G)}{P(G)}$$

where

$P(G)$ is polynomial formed by using N^{th} row of the for coefficient matrix and $N(G)$ is polynomial formed by using $N - 1^{th}$ row.

Up to this point H is still in the form $H(G)$ and needs further expansion in terms of s, which will give the true poles and zeros of the system, in s.

3.2.2 'G' to 's' Mapping

From Eqn. (3.10), the necessary and sufficient conditions for determining the poles are

$$\left(1 - GT_3(s)\right) = 0, \left(1 - GT_4(s)\right) = 0, \dots, \left(1 - GT_N(s)\right) = 0$$

or,

$$GT_3(s) = 1, GT_4(s) = 1, \dots, GT_N(s) = 1$$

$$T_3(s) = \frac{1}{G}, T_4(s) = \frac{1}{G}, \dots, T_N(s) = \frac{1}{G}$$

But $T_N = G$. Considering $N = 4$:

$$G = \frac{1}{G} \rightarrow \boldsymbol{G = \pm 1} \tag{3.18}$$

$$\frac{G}{1 - G^2} = \frac{G(1 - G^2)}{1 - 2G^2} = \frac{G(1 - 2G^2)}{1 - 3G^2 + G^4} \tag{3.19}$$

From equations (3.18) and (3.19) we get the following,

$$\frac{G}{1 - G^2} = \pm 1$$

$$G^2 + G - 1 = 0 \quad and \quad -G^2 + G + 1 = 0$$

52

$$G = \pm 0.618 \quad and \quad G = \pm 1.618$$

$$\frac{G(1-G^2)}{1-2G^2} = \pm 1$$

$$-G^3 + 2G^2 + G - 1 = 0 \quad and \quad -G^3 - 2G^2 + G + 1 = 0$$

$$G = \pm 2.247 \quad and \quad G = \pm 0.8019 \quad and \quad G = \pm 0.555$$

$$\frac{G(1-2G^2)}{1-3G^2+G^4} = \pm 1$$

$$G^4 - 2G^3 + 3G^2 + G - 1 = 0 \quad and \quad G^4 - 2G^3 - 3G^2 + G + 1 = 0$$

$$G = \pm 2.87 \quad and \quad G = \pm 0.65 \quad and \quad G = \pm 1 \quad and \quad G = \pm 0.53$$

For each value of G obtained above (*in bold*), there exist two values of s, i.e., poles in the s-plane, which can be obtained by solving the quadratic equation,

$$G = \frac{bs + k}{ms^2 + 2bk + 2k}$$

or

$$mGs^2 + (2bG - b)s + (2kG - k) = 0 \qquad (3.20)$$

53

For a chosen set of property values (m, b and k), simulation based on the transfer function resulting by the recursive method, became unstable after $N = 10$ because of certain un-cancelled pole-zero pairs. Further, the maximum N value for stability depended upon the values of m, b and k.

By the pattern technique, the number of poles is always $2N$ and hence no extra poles and zeros are generated. Figures 3.11 and 3.12 show that, for the same parametric values as used before, the pole-zero plots for the transfer functions obtained by the pattern method retain stability for up to $N = 56$. The instability issue is avoided. The algorithm for this method is illustrated as a flowchart in Fig. 3.13.

In summary, we have available:

- Method 1: *Recursive Method* that evaluates the element transfer functions by a recursive formula given in Eqn. (3.12) and provides the system transfer function multiplication of the element transfer functions. The resulting transfer function is highly prone to numerical inaccuracies in simulation.

- Method 2: *Pattern Method* that makes use of an algebraic pattern to generate the system transfer function in G plane and then maps the poles in G to those in s by making use of a quadratic roots formula. The simulation results are stable for high orders of the model.

54

The transcription function obtained from the pattern method results in a pole-zero map as shown in Fig. 3.11. This agrees with the results obtained with the recursive method (Fig. 3.6) in terms of the leftward shifting of poles for higher order models. Unlike the recursive method, there are no un-cancelled pole-zero pairs.
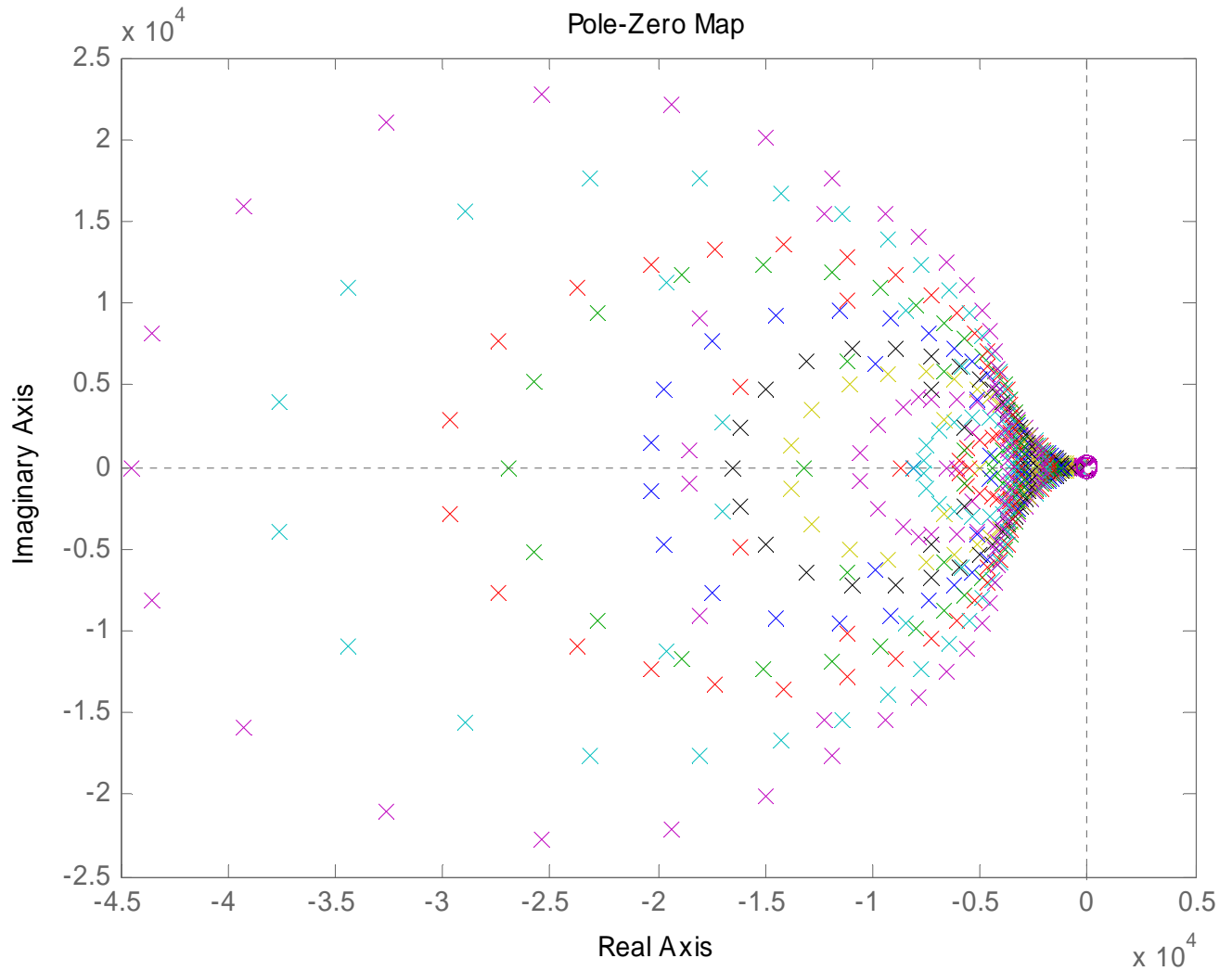
Fig.3.11 ***Pole-zero map from the pattern method, (for N=2 to 56)***

Unstable simulation occurred when $N = 58$ and the poles are shown in Fig. 3.12. MATLAB does not provide a plot for $N > 58$ because of insufficient memory. The numerical instability for $N \geq 58$ observed could be because of limitation in the Matlab algorithm rather than the pattern method.
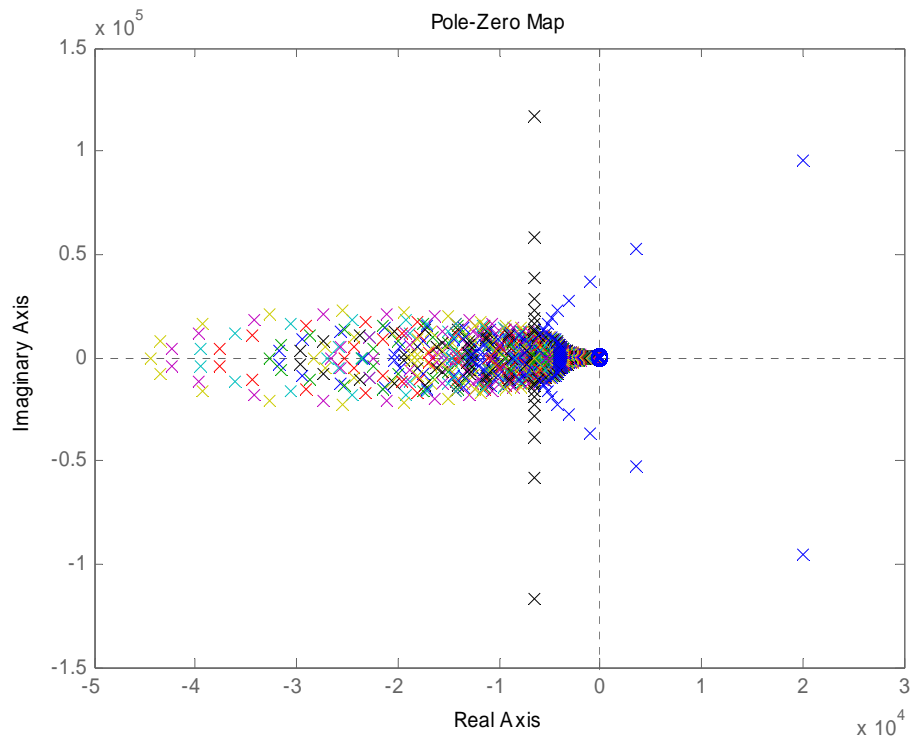
Fig. 3.12 **Poles are on vertical line $R_e(s) = -\dfrac{b}{k}$ for N=57, and move into RHS for N=58 leading to instability of simulation**

*Fig. 3.13 **Flow chart for pattern method to evaluate the transfer function model***

```
                              ┌───────────┐
                              │   Start   │
                              └─────┬─────┘
                                    ▼
                    ┌──────────────────────────────┐
                    │           Calculate           │
                    │ element properties from bulk  │
                    │   properties m, b and k       │
                    └──────────────┬────────────────┘
                                   ▼
```

Define final element TF,

$$T_N(s) = G = \frac{bs+k}{ms^2 + 2bs + 2k}$$

Generate co-efficient matrix *GMat* by using the pattern:

$$Coeff(G_N^n) = \left\{(-1)^{1+\left|\frac{n}{2}\right|+\left|\frac{N}{2}\right|}\right\} \cdot \{|Coeff(G_{N-1}^n)| + |Coeff(G_{N-2}^{n-2})|\}$$

Use final row of *GMat* to directly generate TF of the form:

$$H(G) = \frac{G^{n-1} \cdot (bs+k)}{a_1 G^N + a_2 G^{N-2} + \ldots + a_{N-1} G^2 + a_N}$$

Generate $T_2(G)$ using final two rows of *GMat* as numerator and denominator

Evaluate

$$T_1(G) = \frac{1}{ms^2 + (1 - T_2(G))bs + (1 - T_2(G))k}$$

Obtain $H(G) \rightarrow H(s)$ by solving N quadratic equations and their 2N roots as poles of the transfer function

Return $H(s)$

```
                              ┌───────────┐
                              │   Stop    │
                              └───────────┘
```

57

The pattern obtained, provides a way to generate the true number of poles without artificial poles and zeros because of numerical inaccuracies. Eqn. (3.20) gives a direct relation between poles in G-domain and poles in s-domain.

An important result is that the poles increase in number and move leftward in the s-plane with increasing N. In other words, for increasing number of elements, smaller and smaller phase lags or time delays are added for a given frequency as illustrated in a future chapter.

3.3 Model Validation

The results obtained with the proposed method can be further validated. For that purpose, the state space method was used as a reference, which is an alternative and more conventional method for transfer function estimation of finite-state systems. Both models give the same transfer function and generate similar responses to a given input.

3.3.1 State Space Method

Consider the cases of $N=3$ or 4. The system matrices $(A, B, C, D)_N$ can be easily shown as follows,

$$
\mathbf{A_3} = \begin{bmatrix}
-\dfrac{k}{m} & \dfrac{k}{m} & 0 & -\dfrac{b}{m} & \dfrac{b}{m} & 0 \\
-\dfrac{k}{m} & \dfrac{2k}{m} & -\dfrac{k}{m} & -\dfrac{b}{m} & \dfrac{2b}{m} & -\dfrac{b}{m} \\
0 & k & -\dfrac{2k}{m} & 0 & \dfrac{b}{m} & -\dfrac{2b}{m} \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

$$
\mathbf{A_4} = \begin{bmatrix}
-\dfrac{k}{m} & \dfrac{k}{m} & 0 & 0 & -\dfrac{b}{m} & \dfrac{b}{m} & 0 & 0 \\
-\dfrac{k}{m} & \dfrac{2k}{m} & -\dfrac{k}{m} & 0 & -\dfrac{b}{m} & \dfrac{2b}{m} & -\dfrac{b}{m} & 0 \\
0 & \dfrac{k}{m} & -\dfrac{2k}{m} & \dfrac{k}{m} & 0 & \dfrac{b}{m} & -\dfrac{2b}{m} & \dfrac{b}{m} \\
0 & 0 & \dfrac{k}{m} & -\dfrac{2k}{m} & 0 & 0 & \dfrac{b}{m} & -\dfrac{2b}{m} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

$$\mathbf{B_3} = \begin{bmatrix} -\dfrac{1}{m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{B_4} = \begin{bmatrix} -\dfrac{1}{m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{C_3} = \begin{bmatrix} 0 & 0 & k & 0 & 0 & b \end{bmatrix}$$

$$\mathbf{C_4} = \begin{bmatrix} 0 & 0 & 0 & k & 0 & 0 & 0 & b \end{bmatrix}$$

$$\mathbf{D_3} = 0$$

$$\mathbf{D_4} = 0$$

where subscripts 3 and 4 indicate the number of elements. Appendix A includes the procedure to obtain these matrices from force balances.

3.3.2 Algorithm to produce state space model

Among A, B, C and D the last three matrices are trivial to evaluate. Changes in N correspond to shifting the location of non-zero cells.

Matrix A, the system matrix, requires further analysis. From Fig. 3.14 it can be seen that matrix A can be divided into four equal quadrants such that the upper two quadrants consist of a tridiagonal matrix each. One of the lower quadrants is a zero matrix and the other is a unit matrix, both of order $N \times N$. The two tri-diagonal matrices are exactly the same except for a different constant multiplier.
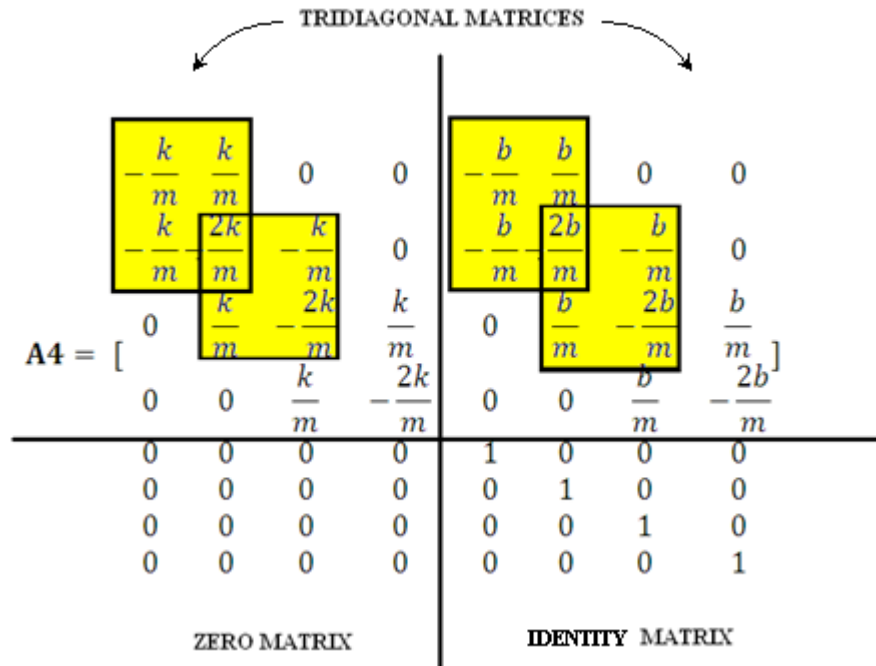
Fig. 3.14 *Structure of System Matrix A*

Further in the band diagonal, a 2×2 matrix repeatedly appears as shaded in Fig. 3.14. This matrix appears to be of the form,

$$Q \times \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}$$

where $Q = \frac{k}{m}$ for the upper left quadrant and $Q = \frac{b}{m}$ for upper right, for the chosen arrangement of the state variables. The only exception is that, the first element of the left quadrant is replaced by 1 instead of 2.

By connecting the four quadrant matrices, the desired matrix A was obtained and the results were verified for higher values of order N. The flow diagram is shown in Fig. 3.15 and Matlab implementation is in Appendix C.

The transfer function obtained exactly match the proposed pattern transfer function method, for a number of N values tested. This further verifies the recursive method and the pattern method, which were shown to give identical results earlier.

Fig. 3.15 **Flowchart for automated generation of state space model for given number of state variables**

Start

Initialize
$A = [0]_{NxN}$
Mask $= \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}$

Mask the upper edge of upper diagonal of A

Shift mask downward along the diagonal by one element

Is it end of diagonal?

No

Yes

Set $A(1,1) = 1$

Multiply the tridiagonal matrices by their respective constant ($\frac{k}{m}$ or $\frac{b}{m}$)

B = transpose of [N zeros, 1/m, (n-1) zeros]
C = [n-1 zeros, k, n-1 zeros, b]

Return transfer function as,
$H(s) = C(sI - A)^{-1}B$

Stop

63
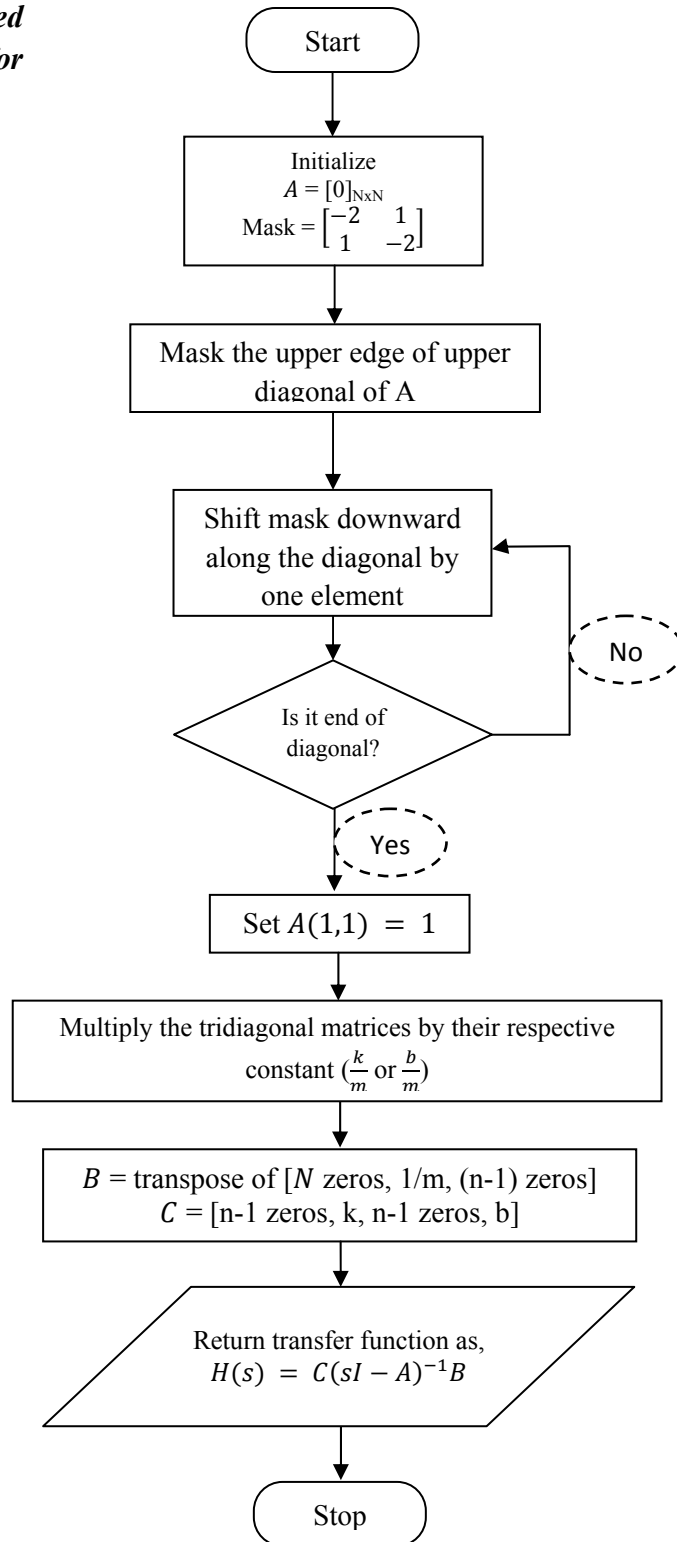
CHAPTER 4

EXPERIMENTS

Experimental data was collected to verify the methods developed. In this chapter, the instrumentation and the experiment method used are described. The methods developed in Chapter 3 were applied to give a time delay correction so that the model could be made to match experimental measurements by parameter adjustments.

4.1.   Instrumentation

Time-of-flight is a measure for speed of sound through a medium of propagation with known dimensions. This time of flight is measured in terms of the phase difference between a periodic input and output with an experimental setup shown in Fig. 4.1.

The experimental setup consists of a function generator, transducers, an oscilloscope connected as shown in Fig. 4.1. Overlapping waveforms of input and output are obtained to observe the phase difference between the two. This phase difference is the true time delay in the response for a given input frequency.

Source: A 15-MHz function generator (Model DS340, Stanford Research Systems) is the source supplying a known sinusoidal wave to the emitter transducer.
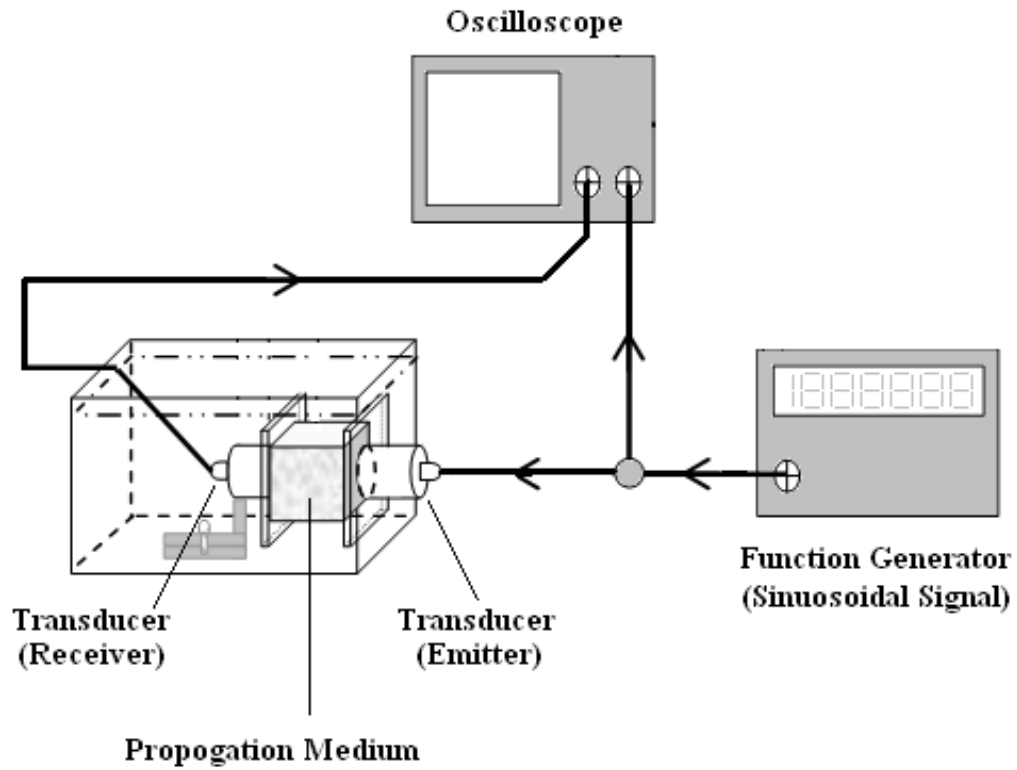
*Fig. 4.1 **Experiment Setup***

Transducer: Immersion-type transducers (PANAMETRIC 5052UA #1) were used for the emitter-receiver pair. The emitter was fixed in position on the rectangular chamber, while the receiver could be moved and fastened to give varied lengths of the propagation medium. The transducer details are in Appendix D.

4.2.   Experimental Errors

The various sources of error in measurement were individually dealt with and compensated for in either analytical or mathematical ways.

4.2.1.   Error in length

To determine the sound velocity from time-of-flight measurements, accurate medium lengths are needed. The unknown thickness of the transducer wear plates and geometry could result in significant errors in the medium length measurements, which need to be compensated for.

Frequency variation is known to result in change in time-of-flight (Guillon et al., 1998). But medium length variation should not result in changed in measured sound velocity, which can be expressed as,

$$c = \frac{L + \Delta L}{\Delta T} \qquad (4.1)$$

where ΔT is time of flight measure, L is a nominal medium length measured between the two transducer wear plates and ΔL is, a length correction which should be constant for given transducer pair. For two or more different values of L, we should get the same value of c from observed ΔT, for a given frequency, which can be expressed as,

$$\frac{L_1 + \Delta L}{\Delta T_1} = \frac{L_2 + \Delta L}{\Delta T_2} = \frac{L_3 + \Delta L}{\Delta T_3} = \cdots \tag{4.2}$$

From this, it is possible to determine ΔL by solving one of the equations or finding the optimal solution for several equations, which was done in this work.   For the experimental setup used, the value of ΔL turned out to be 14.32 mm.

4.2.2.   Error in time (*number of skipped cycles*)

The time delay between input and output waveforms could be many whole cycles plus a fraction of a cycle. An error could be made in counting the number of cycles in a time-of-flight, especially at high frequencies. To avoid such an error, measurements were initiated at short medium length and low input frequency, in which case a full cycle could not be easily overlooked and the measured sound velocity could still be in the expected range for the medium used. This base measurement and incremental increases in length and frequency helped ensure readings of the phase delay.

4.3. Experiment Data and Analysis

The time-of-flight was obtained by comparing the input and output waveforms on the oscilloscope with the 'Measure' feature. By positioning the two pointers at the zero-crossing points, the time delay in milliseconds could be obtained.

Table 4.1 lists the time-of-flight readings obtained for three different medium lengths and several input frequencies ranging from 2 to 100 kHz transmitted through water as a medium. The correction in length measurement obtained in Section 4.2.1 was applied.

With ambient conditions held constant, the sound velocity should remain unchanged for a given propagation medium and input frequency. Three readings of phase difference were obtained for each length of medium and input frequency. The three readings were averaged and used to compute sound velocity. The sound speed increased with input frequency. These variations are summarized in graphical form in Figures 4.2 and 4.3.

Nominal Length (ΔL = 14.23 mm)	L=96 mm		L=55 mm		L=14 mm		Avg. Sound velocity for given input frequency in m/s
f in KHz ⇓	Δt (ms)	Sound velocity (m/s)	Δt (ms)	Sound velocity (m/s)	Δt (ms)	Sound velocity (m/s)	
2	133	820.00	84	821.43	35	827.07	822.83
4	133	823.53	84	821.43	34	827.07	824.01
5	133	823.53	83.4	827.34	34	827.07	825.98
7	131	840.84	82.2	839.42	33.3	839.69	839.98
10	129.1	851.06	81	851.85	32.9	852.05	851.65
12	127.8	861.54	80.2	860.35	32.5	860.72	860.87
14	126.3	872.27	79.2	871.21	32.1	870.94	871.47
15	124.3	886.08	78	884.62	31.6	884.96	885.22
16	123	894.57	77.2	893.78	31.3	894.31	894.22
18	119.6	921.05	75	920.00	30.4	919.73	920.26
19	118	933.33	74	932.43	30	932.20	932.66
20	116.4	945.95	73	945.21	29.6	945.02	945.39
21	116.4	979.02	70.5	978.72	28.6	945.02	967.59
22	116.4	945.94	73	945.21	29.6	945.02	945.39
23	114	965.52	71.5	965.03	29	964.91	965.15
24	113.2	972.22	71	971.83	28.8	971.73	971.93
25	112.6	952.38	71.5	965.03	29.4	976.91	964.77
26	112.4	979.02	70.5	978.72	28.6	978.65	978.80
27	109.2	1085.27	69.5	992.81	25.8	1007.33	1028.47
28	107.6	1021.90	67.5	1022.22	27.4	1022.30	1022.14
29	101.2	1085.27	63.5	1086.61	25.8	1086.96	1086.28
31	98.5	1191.9	58	1189.65	23.5	1116.75	1165.97
32	96.4	1142.86	60.9	1133.01	24.5	1141.08	1138.98
33	95.3	1152.26	60.8	1134.87	24.3	1154.25	1147.13
35	93.5	1138.21	60.5	1140.49	24.6	1176.47	1151.73
38	90	1222.71	56.5	1221.24	22.9	1222.22	1222.06
42	88.5	1244.44	56.5	1221.24	22.5	1242.93	1236.21
46	87.3	1233.48	56	1232.14	22.7	1260.02	1241.88
50	84.2	1308.41	52.8	1306.82	21.4	1306.41	1307.21
55	82.8	1327.01	52	1326.92	21.1	1328.50	1327.48
70	81.6	1346.15	51.2	1347.65	20.8	1348.04	1347.28
80	78.4	1400.00	49.2	1402.44	20	1403.06	1401.83
90	77.8	1414.14	48.8	1413.93	19.8	1413.88	1413.98
100	76.5	1435.90	48	1437.50	19.5	1437.91	1437.10

Table 4.1 **Experimental Data**

The slope of the curves in Fig. 4.2 represents the rate at which the time-of-flight decreases or sound velocity increases with input frequency. This rate appears to increase with the length of medium. It is known that sound velocity varies with frequency because of the existence of damping.
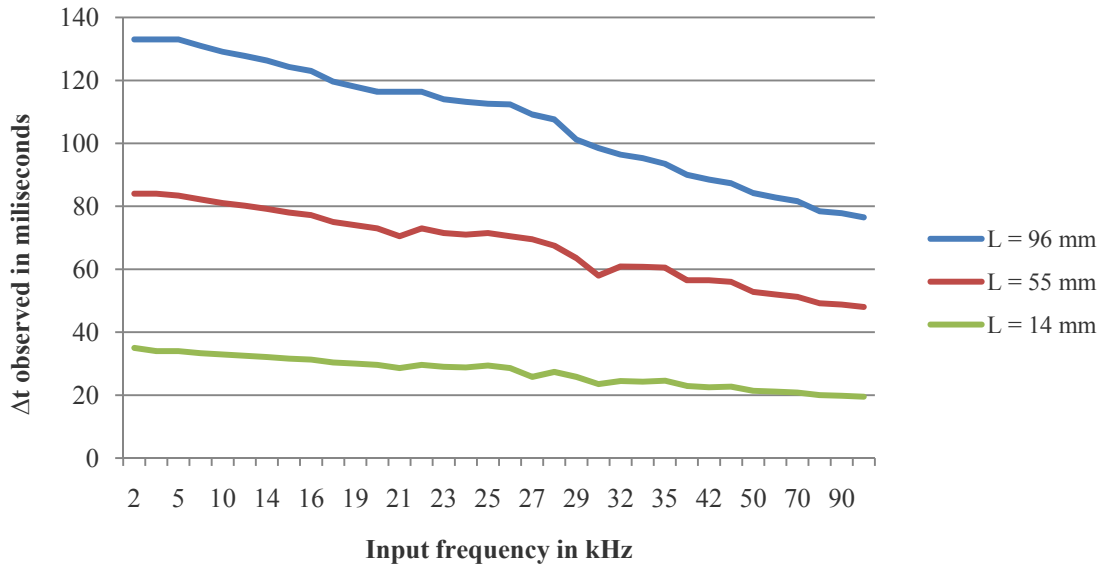
*Fig. 4.2 **Variations in time of flight Δt with respect to input frequency***

This rate dependence on medium length may seem to indicate that the medium damping did not vary linearly or proportionally with length, but this is not the case. The medium used was actually a composite of water of length *L* plus another material (wear plates) which was corrected with an effective length ΔL. The second material of fixed length (as opposed to proportional) would affect the time delay differently for different medium *L*. When ΔL was applied in computing sound velocity from Δ*t* and *L*, the three

curves overlapped (Fig. 4.3), indicating that the water medium exhibited little nonlinear damping effects.
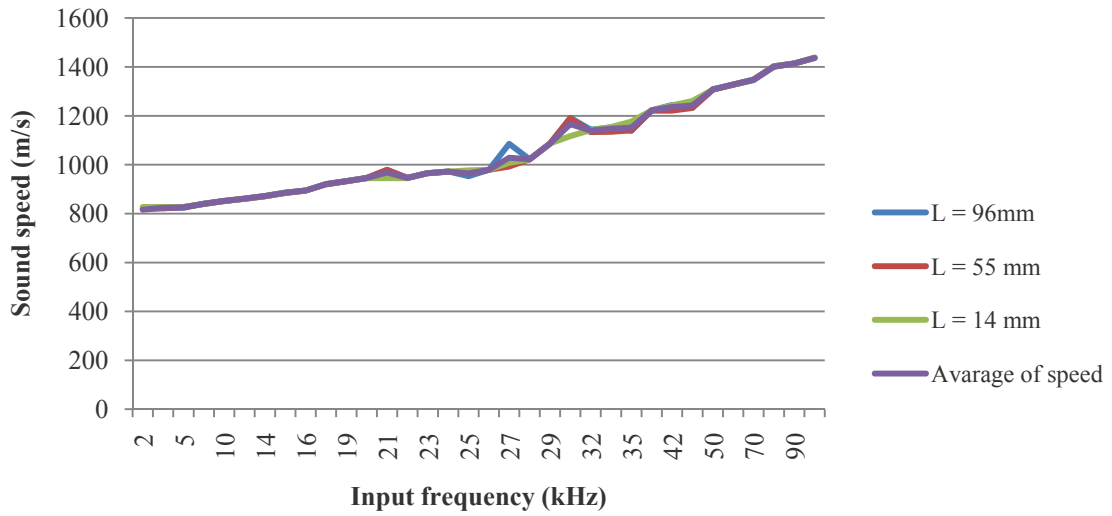

*Fig. 4.3 **Sound speed as function of input frequency***

4.4 Time Delay Compensation

An objective of this work was to determine the time delay in a finite-order model so that the model can be made to fit the experimental data by adjusting the model parameters. For given medium properties (m, b and k), frequency ω and number of element N, a finite-order model $H(s)$ can be obtained by the methods presented in Chapter 3. From $H(s)$, the phase angle could be calculated as

$$\Phi(\omega) = \arg\big(Num(j\omega)\big) - \arg\big(Den(j\omega)\big) \qquad (4.3)$$

where transfer function

$$H(j\omega) = \frac{Num(j\omega)}{Den(j\omega)}$$

and ω is angular frequency. Algorithm flowchart is shown in Fig. 4.4. The time delay in seconds is

$$t = \frac{\Phi(\omega)}{\omega} \qquad (4.4)$$

Time delay is the time-of-flight given by the model. This time of flight, the model output, was then plotted against both increasing number of elements N and increasing input frequency f to observe simultaneous effect of the two (Fig. 4.5).

*Fig. 4.4 **Flowchart for phase delay determination***

Start

Initialize
P and N as the array of poles and zeros Ends of two loops with number of poles and zeros and phase angle
$\Phi = 0$

Calculate phase angle contribution because of first pole as, $||pole||$

$\Phi = \Phi + ||pole||$

No

Is this the final pole?

Yes

$\Phi = \Phi + ||zero||$

No

Is this the final zero?

Yes

Return Φ

Stop
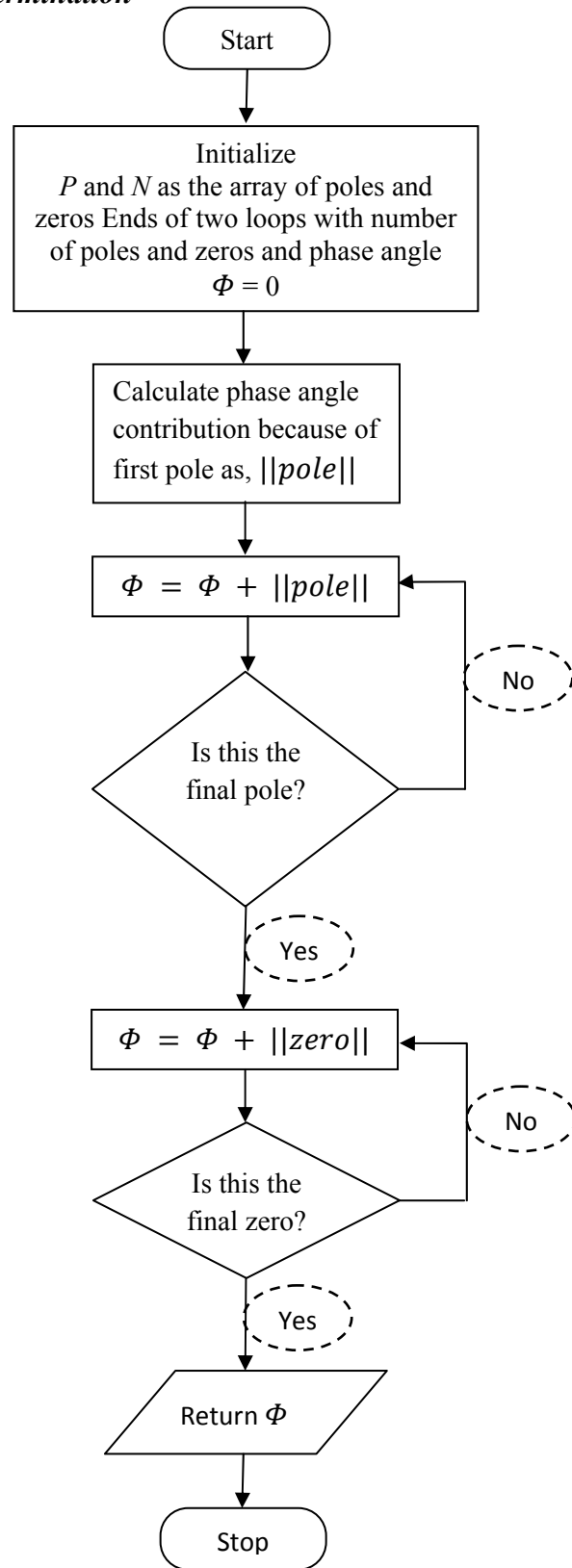
Fig. 4.5 ***Surface plot for time delay by model vs. number of elements and input frequency***

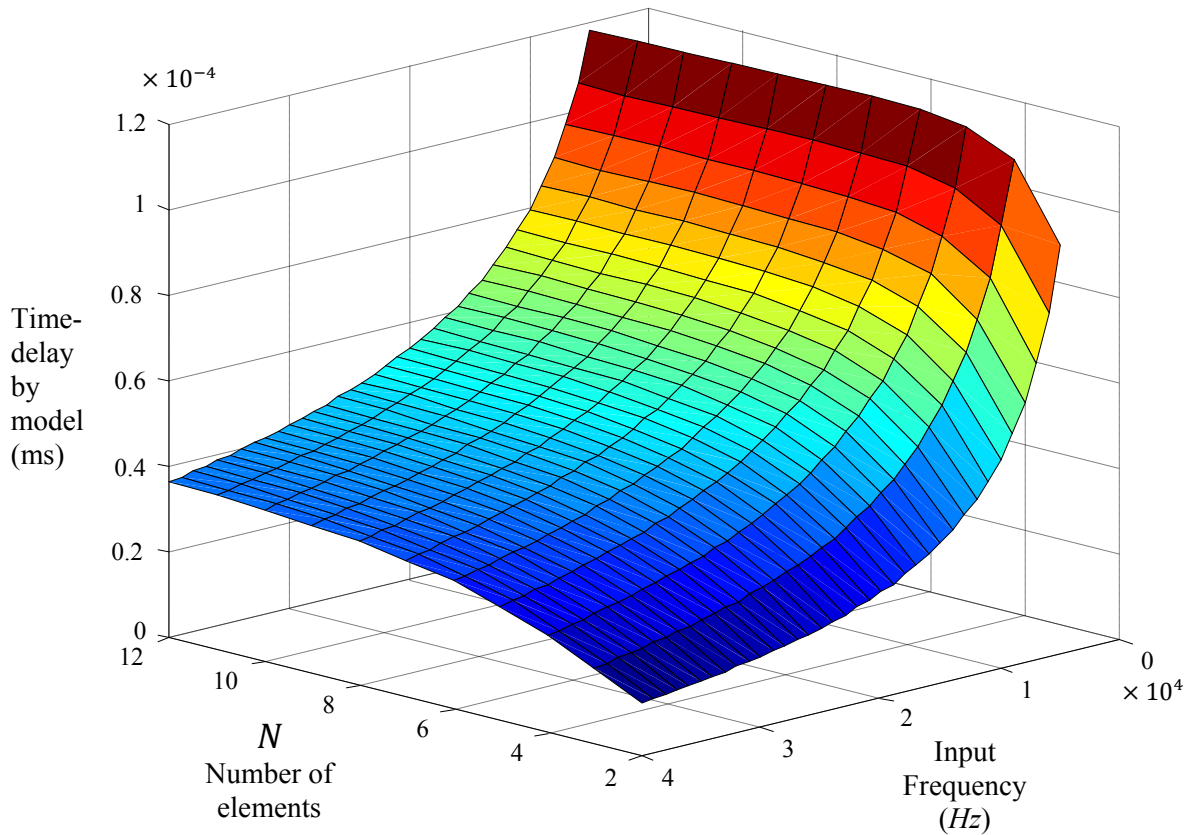Medium properties and geometry were kept constant. For a given frequency, the time-of-flight increases rapidly in the beginning and starts to saturate at certain high number of elements. This agrees with the earlier analysis that the time delay will converge when N increases.

Time of flight decreases with increasing input frequency, which results in greater sound velocity for higher frequencies. This agrees with the observation made from experimental data shown in Table 4.1.

The model thus gives expected overall behavior in terms of time delay. The model and experimental data of a given input (sinusoidal) are plotted together in Fig. 4.6.
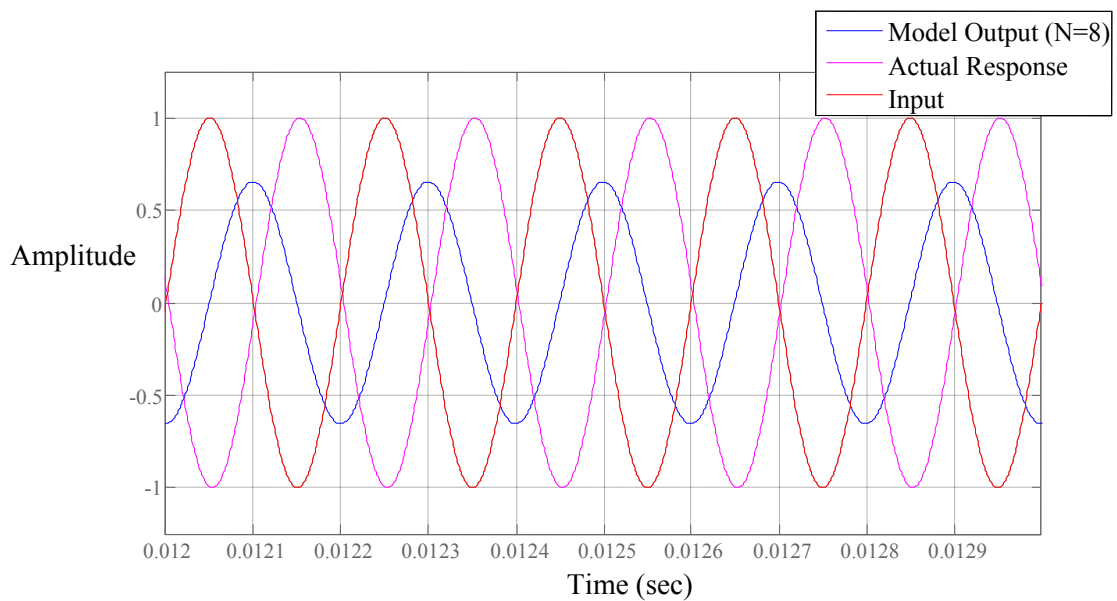

*Fig. 4.6 **Model response** $(N = 8)$ **compared with experiment result***

As Fig. 4.6 shows, the model has a time delay deficiency δt,

$$\delta t = \Delta t - \frac{\Phi}{\omega} \tag{4.5}$$

where

Φ : Phase delay provided by the model,

ω : Angular frequency, and

Δt : The actual time of flight measured in experiment

An added time shift δt will provide a course correction, for aligning the model with experimental data. After the course correction, there can still be major difference between the modeled and the actual response in amplitude which depends on the medium property values. The model parameters which are functions of *m, b* and *k* can be adjusted to fit the model to experimental data.

One way is to obtain a set of parameter values by using a least square based estimation method as shown in Waluyo (2010). Given the scope of this research however, implementing such an algorithm is left for the future research. To show the capabilities of the proposed method, in helping match the actual response, trail-and-error was used to obtain parameter values that provide the closest fitting to experimental measurements.

An algorithm for time delay correction and parameter adjustment is shown by the flowchart in Fig. 4.8 After time delay correction and parameter adjustment the model response closely match with the actual response as shown in Fig. 4.7.
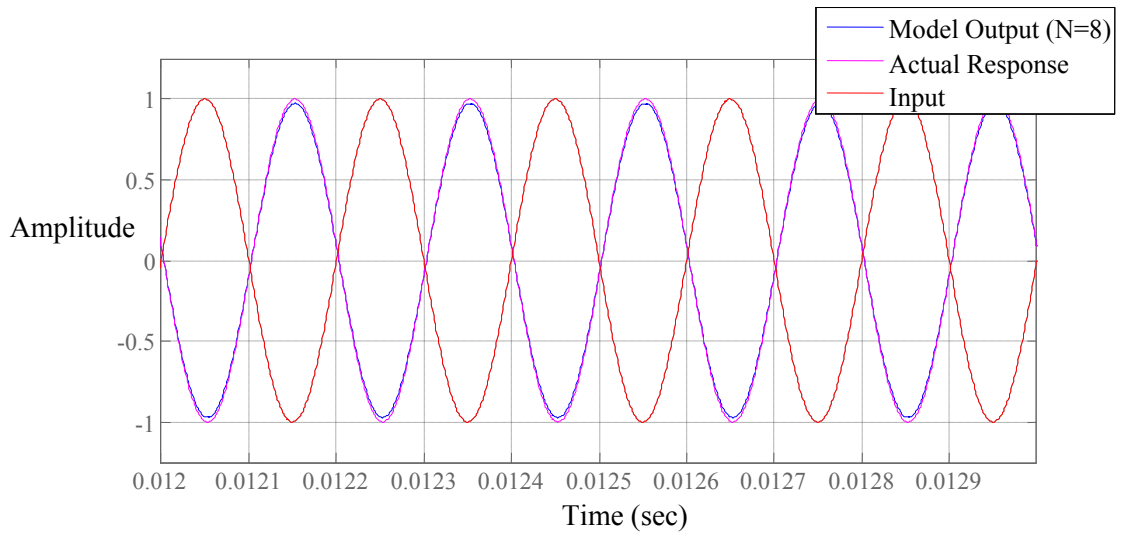
Fig. 4.7 **Phase and gain compensated model response compared with measured response**

Fig. 4.8 ***Algorithm flowchart for time delay compensation and parameter adjustment***
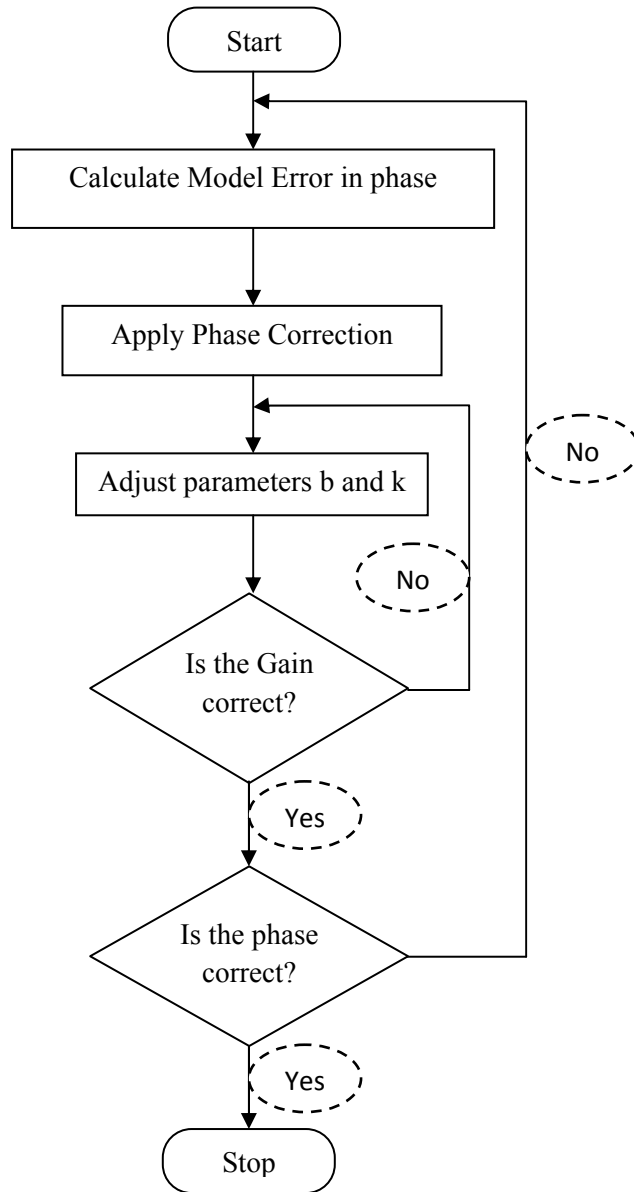
In summary, the methods developed allow compensation of the phase lag in a given model and thus estimation of the necessary additional time-delay correction. This provides a method to iteratively adjust the model parameters to fit experimental measurements.

CHAPTER 5

SUMMARY AND FUTURE WORK

5.1   *Summary*

Finite-order linear models allow solution of acoustic wave problems by taking advantage of the vast array of existing linear system techniques, but these models usually do not sufficiently account for the time delays in wave transmission. As a consequence, good fit to experimental data is often not achievable by adjusting the model parameters alone and an additional time shift in the model is necessary. Determining this essential extra time delay entails knowledge of the time (or phase) delay already in a model, which can be determined if the model is known. As a result, it is desirable to derive a general model structure of arbitrary order. In this work, two methods were developed to give a linear model of desirable order based on a Kelvin-Voigt network: a recursive method and a coefficient pattern method.

The recursive method took advantage of a generic element transfer function and a given boundary condition to derive an overall system transfer function by successive substitution. The transfer function model obtained can be extended to an arbitrary order or number of discretized elements (N), but it is not in a factorized form to allow elimination of redundant equal zero-pole pairs resulting from successive substitution. In numerical simulations and analysis, these superfluous zeros and poles could cause computational instability and difficulty for high model orders.

79

The coefficient pattern method was based on observed patterns in the coefficients of the numerator and denominator polynomials of the system transfer function. The method gives the same results as the recursive method and does but does not suffer from the numerical problems associated with redundant poles and zeros. Results from both methods were further verified with transfer functions computed from state space models developed in the time domain.

With a model structure extendable to a desirable order and given model parameters, the time delay existing in a model can be computed as a function of excitation frequency. This allowed the formulation of an algorithm to determine the extra time delay needed to fit a model to a transmitted wave with a measured time-of-flight. Experimental data were collected and used to test the algorithm. The experimental data were also used to verify time delay properties revealed by the system model.

5.2   *Recommended Future Work*

- Implementation of a parameter estimation algorithm to be used in tandem with the developed algorithm for time delay compensation in fitting a finite-order linear model to experimental measurements,

- An improved correction technique for the error in measured length of propagation medium,

- An algorithm to find the initial conditions that would not generate a transient response in time-domain simulation, and

- Extending the scope of this work to non-homogeneous media by accounting for the effects of medium property variations and echoes from medium interfaces.

State Space Transfer Function Method
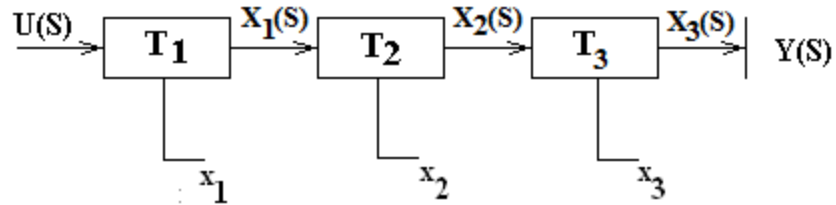
Consider above system model of order n,

State Space Representation of it shall be obtained in the following manner.

Since the displacement of every element is an independent variable, it can be considered as the state variable.

$y(t)$, the displacement of the final element which is the measure for output velocity and acceleration and hence the force experienced by the fixed support.

For first element,

$$m\ddot{x}_1 = -kx_1 + kx_2 - b\dot{x}_1 + b\dot{x}_2 + u$$

$$u = m\ddot{x}_1 + kx_1 - kx_2 + b\dot{x}_1 - b\dot{x}_2 \dots \text{(I)}$$

For second element,

$$k(x_1 - x_2) + b(\dot{x}_1 - \dot{x}_2) = m\ddot{x}_2 + k(x_2 - x_3) + b(\dot{x}_2 - \dot{x}_3)$$

$$\therefore m\ddot{x}_2 = -kx_1 + 2kx_2 - kx_3 - b\dot{x}_1 + 2b\dot{x}_2 - b\dot{x}_3 \dots \text{(II)}$$

For final element,

$$k(x_2 - x_3) + b(\dot{x}_2 - \dot{x}_3) = m\ddot{x}_3 + k(x_3) + b(\dot{x}_3)$$

$$\therefore m\ddot{x}_2 = -kx_1 + 2kx_2 - kx_3 - b\dot{x}_1 + 2b\dot{x}_2 - b\dot{x}_3 \dots \text{(III)}$$

Rearranging equations in Matrix form we get,

$$
\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} =
\begin{bmatrix}
-\dfrac{k}{m} & \dfrac{k}{m} & 0 & -\dfrac{b}{m} & \dfrac{b}{m} & 0 \\
-\dfrac{k}{m} & \dfrac{2k}{m} & -\dfrac{k}{m} & -\dfrac{b}{m} & \dfrac{2b}{m} & -\dfrac{b}{m} \\
0 & k & -\dfrac{2k}{m} & 0 & \dfrac{b}{m} & -\dfrac{2b}{m} \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} +
\begin{bmatrix} -\dfrac{1}{m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} . u(t)
$$

$$
y = \begin{bmatrix} 0 & 0 & k & 0 & 0 & b \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} + [0].u(t)
$$

Thus we have system matrices A, B, C and D for the model of the order $N = 3$.

These can be substituted in the following formula, to obtain State Space Transfer

Function $H(s)$ for the desired model. $\boldsymbol{H(s)} = \boldsymbol{C(sI - A)^{-1}B}$

APPENDIX B

Phase velocity

The phase velocity of a wave is the rate at which the phase of the wave propagates in space. This is the speed at which the phase of any one frequency component of the wave travels. For such a component, any given phase of the wave (for example, the crest) will appear to travel at the phase velocity. The phase velocity is given in terms of the wavelength λ (lambda) and period T as,

$$v_p = \frac{\lambda}{T}$$

Or, equivalently, in terms of the angular frequency ω and wave number k by,

$$v_p = \frac{\omega}{k}$$

In a dispersive medium, the phase velocity varies with frequency and is not necessarily the same as the group velocity of the wave, which is the rate at which changes in amplitude (known as the *envelope* of the wave) propagate.

The phase velocity of electromagnetic radiation may, under certain circumstances, (for example anomalous dispersion) exceed the speed of light in a vacuum, but this does not indicate any superluminal information or energy transfer. It was theoretically described by physicists such as Arnold Sommerfeld and Léon Brillouin.

Programs

      This section includes MATLAB Programs scripted and used during the research

work for the simulation and data visualization purposes. First is the Main body of the

program and subsequent are the individually scripted modules documented in the order of

their occurrence in the main script.

```matlab
% following program is a MATLAB script to generate a finite-order model
% of acoustic wave propagation through a one dimensional medium of
% known parametric properties. This program was written by Mr. Nilesh V
% Salvi to serve an automation purpose in Graduate Thesis Research work
% held during year 2010-2011 at University of Missouri.

clc;clear all;close all;
s = tf([1 0],1);

L = 55e-3+14e-3;                              % Medium Bulk Properties
M = .01; B = 2; K = 1e5;

F = import(data(2:30,1));                     % Experimental data
w = import(data(2:30,4));*1e-6;               % imported from external
                                              % file to the program

nmax = 12; nmin = 2;          % limits for number of elements used in test

dT_Mat = zeros(length(F),nmax-nmin+1);

for q=1:length(F)
    f=F(q); w=W(q);                     % observed delay and input
frequency
    [u,t]=gensig('sin',1/f,150/f,0.005/f);
                                        % generating model input sine wave
    for n=nmin:nmax
        m=M/n; k=K*n; b=B/n;            % calculating element properties
                                        % from bulk properties
        G_tf = tf([b k],[m 2*b 2*k]);    % TF for final element

        G-pattern;                      % script to generate TF in G plane
                                        % using a recursive pattern in
                                        % polynomial coefficients

        G2smapping;                    % script for mapping TF from G to s plane
```

```matlab
        ssMatrix_test;              % script to generate State Space Model

        figure(f);lsim(H,u,t);grid on;hold on;
                                    % plots sinusoidal response of model
        figure(f);lsim(Hss,u,t);grid on;hold on;
                                    % response of SSTF models
        axis([0/f 5/f -1 1]);

        plot(t+w,u,'r');            % plots response of experiment
        title 'Response to the sinusoidal input';

        timedelay_calc;                 % obtained by shifting time line
                                        % calculates time delay offered
                                        % and error given by the model

        figure(f);lsim(H,u,t);grid on;hold on;
                                        % plotting again after course
                                        % correction

        figure(2); pzplot(H); hold on;grid on;      % pole zero plot
        figure(2); pzplot(Hss);
        figure(4); margin(H); hold on; grid on;     % Bode plot
        figure(4); margin(Hss); hold on; grid on;
        title 'frequency response for increasing order of the model';
    end
end
hold off;

phase_surface_plot;         % script that generates surface plot for
                            % time of flight vs. number of elements
                            % and input frequency

man_phase_correct;          % script to generate error surface plot
                            % w.r.t. experimental data

%--------------------------end of main script--------------------------
```

```matlab
% following script generates the 2-dimentional array of coefficients
% using a recursive algebraic pattern from which a transfer function is
% obtained in G plane.

GMat = zeros(n+1,n+2);              % Pattern to generate TF
for i=1:size(GMat,1)
    GMat(i,1) = (-1)^floor(i/2);
end
for j=3:size(GMat,2)
    for l=j:size(GMat,1)
        GMat(l,j)=((-1)^(floor((j-1)/2)+...
            floor((l+1)/2)))*(abs(GMat(l-1,j))+...
            abs(GMat(l-2,j-2)));
    end
end

GMat=fliplr(GMat);                  % to meet matlab's sequence of
                                    % polynomial coefficients in TF

Root_G=roots(GMat(n,:));            % poles of TF in G-plane

Root_T2_G_N=roots(GMat(n-1,:));     % numerator polynomial of T2
Root_T2_G_D=roots(GMat(n,:));       % denominator polynomial of T2


%--------------------------end of this module-------------------------
```

```matlab
% following script performs mapping for the TF obtained in G plane to
% the s plane, such that for each pole in G, there are 2 poles in s
% plane which are obtained by solving for a simple quadratic relation N
% times

Root_S=zeros(2*n,1);
Root_T2_S_N=zeros(2*n-2,1);
Root_T2_S_D=zeros(2*n,1);

for j=1:size(Root_G,1)                            % G to s mapping
    G = Root_G(j);

    A1 = m*G;
    B1 = 2*b*G-b;
    C1 = 2*k*G-k;

    S1 = (-B1+sqrt(B1^2-4*A1*C1))/(2*A1);
    S2 = (-B1-sqrt(B1^2-4*A1*C1))/(2*A1);

    Root_S(2*j-1) = S1; Root_S(2*j) = S2;
end

for j=1:size(Root_T2_G_N,1)
    G = Root_T2_G_N(j);

    A1 = m*G;
    B1 = 2*b*G-b;
    C1 = 2*k*G-k;

    S1 = (-B1+sqrt(B1^2-4*A1*C1))/(2*A1);
    S2 = (-B1-sqrt(B1^2-4*A1*C1))/(2*A1);

    Root_T2_S_N(2*j-1) = S1; Root_T2_S_N(2*j) = S2;
end

for j=1:size(Root_T2_G_D,1)
    G = Root_T2_G_D(j);

    A1 = m*G;
    B1 = 2*b*G-b;
    C1 = 2*k*G-k;

    S1 = (-B1+sqrt(B1^2-4*A1*C1))/(2*A1);
    S2 = (-B1-sqrt(B1^2-4*A1*C1))/(2*A1);

    Root_T2_S_D(2*j-1) = S1; Root_T2_S_D(2*j) = S2;
end

[N,D] = zp2tf(1, Root_S',1);                % building TF in s plane

[N_T2,D_T2] = zp2tf(Root_T2_S_N,Root_T2_S_D,1);

                    % Building T2 to obtain T1 through a recursive relation
```

89

```matlab
T2_emp = G_tf*tf(N_T2,D_T2);
T1_emp = 1/(m*s*s+(1-T2_emp)*(b*s+k));

H = (G_tf^n-1)*T1_emp*tf(N,D)*(b*s+k);     % required TF in s-plane

%-------------------------end of this module------------------------
```

```matlab
% following is the script to generate system matrices in terms of
% elemental parametric values A = [2n x 2n], B = [2n x 1], C =[1 x 2n]
% and D = 0 for given value of number of elements

A2_ = [-2 1;1 -2];
A2_11= zeros(n,n);

for i=1:n-1
    A2_11(i,i) = A2_(1,1);
    A2_11(i,i+1) = A2_(1,2);
    A2_11(i+1,i) = A2_(2,1);
    A2_11(i+1,i+1) = A2_(2,2);
end
A2_11(1,1)=-1;
A2 = [zeros(n,n) eye(n)
      (k/m)*A2_11 (b/m)*A2_11];

B2 = [zeros(1,n) 1/m zeros(1,n-1)]';

C2 = [zeros(1,n-1) k zeros(1,n-1) b];

D2 = 0;

Hss = C2*inv(s*eye(size(A2))-A2)*B2+D2;

%-------------------------end of this module-------------------------
```

```matlab
% following is the script to obtain the phase angle of the transfer
% function model and from that to obtain the time-of-flight using input
% frequency

phi = 0; p=pole(H); z=zero(H);   % array of poles and zeros
lp=length(p); lz=length(z);      % number of poles and zeros
for i=1:lp
    phi = phi - atan(2*pi*f/p(i));  % 'phi' initialized in element_prop
end
for i=1:lz
    phi = phi + atan(2*pi*f/z(i));
end
dT_Mat(q,n-nmin+1) = phi/(2*pi*f);  % time delay in seconds

w=abs(w-(phi/(2*pi*f)));
H = H*exp(-w*s);                     % applying the course correction to TF

%-------------------------end of this module-------------------------
```

```
% following is the script to generate surface plot of time of flight
% against input frequency and number of elements in proposed model

[N1,F1] = meshgrid(NE,F);              % Phase lag surface plot w.r.t N and
F
figure(11); surf(N1,F1,dT_Mat);grid on;
xlabel 'number of elements';ylabel 'Input Frequency';zlabel 'phase
angle'

%-------------------------end of this module------------------------
```

APPENDIX D

Transducer Details

An immersion-type transducer used, is a longitudinal wave transducer with a 1/4 wavelength layer acoustically matched to water. Immersion transducers have sealed cases allowing them to be completely submerged under water when used with a waterproof cable. By using water as both a couplant and delay line, immersion transducers are ideal for use in scanning applications where consistent coupling to the part is essential.

Advantages of immersion type transducers:

- The immersion technique provides a means of uniform coupling

- Quarter wavelength matching layer increases sound energy output

- Corrosion resistant 303 stainless steel case with chrome-plated brass connectors

- Proprietary RF shielding for improved signal-to-noise characteristics in critical applications

- Can be focused spherically (spot) or cylindrically (line)

- Customer specified focal length concentrates the sound beam to increase sensitivity to small reflectors

Advantages of Large Diameter Casing type transducer used:

- Large element diameters increase near field length allowing for longer focal lengths

- Larger diameters can increase scanning index

- Low frequency, large element diameter designs available for challenging applications

Applications for this type of transducer:

- Automated scanning

- On-line thickness gauging

- High speed flaw detection in pipe, bar, tube, plate, and other similar components

- Time-of-flight and amplitude based imaging

- Through transmission testing

- Material analysis and velocity measurements

Usage Note: Transducers should not be submerged for periods exceeding 8 hours. Allow 16 hours of dry time to ensure the life of the unit.

APPENDIX E

Tridiagonal Matrix

Tridiagonal Matrix is a square Matrix with nonzero elements only on the diagonal and slots horizontally or vertically adjacent to the diagonal (i.e., along the 'subdiagonal or first diagonal below' and 'superdiagonal or first diagonal above').

$$
\begin{bmatrix}
a_{11} & a_{12} & 0 & 0 & \cdots & 0 & 0 \\
a_{21} & a_{32} & a_{23} & \ddots & \ddots & 0 & 0 \\
0 & a_{32} & a_{33} & \ddots & \ddots & a_{n-2,n-1} & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & a_{n-1,n-1} & a_{n-1,n} \\
0 & 0 & \cdots & \cdots & \cdots & a_{n,n-1} & a_{n,n}
\end{bmatrix}
$$

REFERENCES

Adam, J. A. (1982). Asymptotic solutions and spectral theory of linear wave equations. Physics Reports, 86, 5:217-316.

Al-Saggaf, U., Franklin, G., (1987). An error Bound for a Discrete Reduced Order Model of a Linear Multivariable System. IEEE Transactions on Automatic Control, 32, 9:815-819.

Alvarez-Ramfrez, J., Sueraz, R. (2000). Stabilization of a class of linear time-varying systems via modeling error compensation. IEEE Transactions on Automatic Control, 45, 4:738 – 741.

Ataie-Ashtiani, B., Lockington, D. A., Volker, R. E. (1999). Truncation errors in finite difference models for solute transport equation. with first-order reaction. Journal of Contaminant Hydrology, 35, 4:409-428.

Banta E. D. (1965). Lossless propagation of one-dimensional, finite amplitude sound waves. Journal of Mathematical Analysis and Applications, 10, 1:166-173.

Billinghan, J., King, A. C. (2000). *Wave Motion.* New York, New York: Cambridge University Press, 1st edition, 476 pages.

Dameron, D. H. (1979). Determination of the Acoustic Velocity in Tissues Using an Inhomogeneous Media Model. IEEE Transactions on Sonics Ultrasonics 26, 2:69-74.

Drozdov, A. D. (1996). *Finite elasticity and viscoelasticity: a course in the nonlinear mechanics of Solids*. World Scientific Publishing Company, 248-259.

Fagin, R. (1993). Finite Model Theory – A personal perspective, Theoretical Computer Science, 116:3-31.

Fawcett, J. (1985). Two dimensional velocity inversion of the acoustic wave equation. Wave Motion, 7, 6:503-513.

Guillon, L., Moussatov, A., Brouard, B., Ayrault, C. (1998). Propagation of Sound in Sands: Measurements and Modeling. OCEANS Conference Proceedings, 1:380-384.

Jung, S. S., Kim, Y. T., Pu, Y. C., Kim, M. G., Kim, H. C. (2005). Non-contact sound speed measurement by optical probing of beam deflection because of sound wave. Ultrasonics, 12, 6:12-16.

Kagawa, Y., Tsuchiya, T., Fujii, B., Fujiokaf, K. (1998). Discrete Huygen's Model Approach To Sound Wave Propagation. Journal of Sound and Vibration 218, 3:419-444.

Kagawa, Y., Tsuchiya, T., Yamabuchi, T., Kawabe, H., Fujii, T. (1992). Finite element simulation of non-linear sound wave propagation. Journal of Sound and Vibration 154, 1:125-145.

Kang, B. (2007). Transfer Functions of One-Dimensional Distributed Parameter Systems by Wave Approach. Journal of Vibrations and Acoustics, 129, 2:193-201.

Kolodziej, J. R., Mook, D. (2011). Model Determination using the Minimum Model Error Estimation. Nonlinear Dynamics, 63, 4:735-753.

Levy, Y., Agnon, Y., Azhari, H. (2006). Measurement of speed of sound dispersion in soft tissues using a double frequency continuous wave method. Ultrasound in Med. and Biol., 32, 7:1065-1071.

Ljung, L., Lei, G. (1995), Estimating the Total Model Error from a Standard Validation Test. Proceedings of the 34th IEEE Conference on Decision and Control, 2:1651-1656.

Mäkilä, P. M., Partington, J. R. (2003). On linear models for nonlinear systems. Automatica, 39, 1:1-13.

Morgül, Ömer (2002). An exponential stability result for the wave equation. Automatica, 38, 4:731-735.

Nagurka, M., Huang, S. (2006). Mass-spring-damper model of a bouncing ball. International Journal of Engineering Education, 22, 2:393-401.

Ophir, J., Moriya, T., Yazdi, Y. (1991). A single transducer trans-axial compression technique for the estimation of sound speed in biological tissues. Ultrasonic Imaging, 13:269-279.

Parameswaran, V., Raol, J. R., (1994). Estimation of model error for nonlinear system Identification. IEEE Proceedings in Control Theory and Applications, 141, 6:403-408.

Pereira, F. R., Pereira, W. C. A., Machado, J. C. (2000). Ultrasonic wave speed measurement using the time-delay profile of RF-backscattered signals. Ultrasonics, Volume 38, 1-8:708-710.

Rabenstein, R., Schetelig, T. (1998). A multidimensional wave digital filter algorithm for the 3D simulation of room acoustics. Proceedings to Image and Multidimensional Digital Signal Processing Workshop, IMDSP, 255-258.

Rabenstein, R., Zayati, A. (1999). A Direct Method To Computational Acoustics. Acoustics, Speech, and Signal Processing. ICASSP '99, Proceedings, 957 – 960.

Reinelt, W.; Garulli, A.; Ljung, L., Braslavsky, J. H., Vicino, A. (1999). Model Error Concepts in Identification for Control Decision and Control. Proceedings of the 38th IEEE Conference, 2:1488 – 1493.

Schmidt, H., Alber, T., Wehner, T., Blakytny, R., Wilke, H. (2009). Discretization error when using finite element models: Analysis and evaluation of an underestimated problem. Journal of Biomechanics, 42, 12:1926-1934.

Shaw, S., Warby, M. K. and Whiteman, J. R. (2010). Discretization error and modeling error in the context of the rapid inflation of hyper-elastic membranes. IMA Journal of Numerical Analysis, 30:302–333.

Srinivasan, K., Sundararajan, T., Narayanan, S., Joshi, T. J. S. (2009). Effects of acoustic source and filtering on time-of-flight measurements. Applied Acoustics, 70, 8:1061-1072.

Subedi, P. P., Walsh, K. B. (2009). Non-invasive techniques for measurement of fresh fruit firmness. Postharvest Biology and Technology, Volume 51, Issue 3, 297-304.

Sun, J., Olbrot, A.W., Polis M. P. (1994). Robust stabilization and robust performance using model reference control and modeling error compensation. IEEE transactions on Automatic Control, 39, 3:630-634.

Tohyama, M., Koike, T. (1998). *Fundamentals of Acoustic Signal Processing*. Academic Press, 1st edition, 321 Pages.

Waluyo, S. (2010). Modeling approach to ultrasound evaluation of biological material properties. PhD dissertation, University of Missouri.

Wang, F. (2003). The Exact and Unique Solution for Phase-Lead and Phase-Lag Compensation. IEEE Transactions on Education, 46, 2:258-262.

Yae, K. H., Inman, D. J. (1991). Control-Oriented Order Reduction of Finite Element Model. American Control Conference, $2187 - 2189$.

Yoneya, A., Kobayashi, H., Togari, Y. (2002). On Evaluation of Model Error Bound from Experimental Data. SICE, Proceedings of the 41st SICE Annual Conference, 1:545-549.

VITA

Nilesh Vijay Salvi was born in Mumbai, India on January 3, 1985, the son of Vijay Janardan Salvi and Vidyut Vijay Salvi. He went to high school at Shailendra Education Society affiliated with the Maharashtra State Board. After finishing junior college studies in science at Royal College of Science, he attended Vidyavardhini College of Engineering, University of Mumbai.

He graduated with a Bachelors degree in Instrumentation Engineering and then worked as an Electronic Design Engineer in India for the next two years. In Fall 2008, he joined the University of Missouri as a graduate student in the Department of Electrical Engineering. Later, he switched his major to Biological Engineering to pursue this research for his thesis.

Permanent Address: 1402 Orchid, Gundecha's Valley of Flowers, Thakur Village, Kandivali (E), Mumbai 400101, Maharashtra, India

Personal Email: nileshsalvi@gmail.com