LUMPED PARAMETER MODELING OF FLUID LINE DYNAMICS WITH TURBULENT FLOW CONDITIONS

by

YI-WEI HUANG

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2012

Acknowledgements

I would like to take this opportunity to thank all the persons who have contributed in the different aspects of this study. I would never have been able to finish my dissertation without the guidance of my committee members, encouragement from friends, and support from my family.

I would like to express my deepest gratitude to my advisor, Dr. David A. Hullender, for his excellent guidance, caring, encouragement, and patiently corrected my writing. With his generous help, I have learned a lot about fluid dynamics modeling and utilization of MATLAB software, which has been a valuable tool during my dissertation, and I consider myself fortunate to be associated with him. I would also like to thank Dr. Robert L. Woods for his valuable comments and suggestions that helped to answer specific questions and prevented me from obtaining the wrong conclusions. In addition, this dissertation could not have been achieved without the valuable guidance and constructive criticism from the remaining committee members: Dr. S. Nomura, Dr. D. R. Wilson, and Dr. Brian Dennis.

Finally, and certainly not least, I wish to thank my family, especially my parents, for their support and encouragement throughout my study at the University of Texas at Arlington.

November 23, 2011

Abstract

LUMPED PARAMETER MODELING OF

FLUID LINE DYNAMICS WITH

TURBULENT FLOW

CONDITIONS

Yi-Wei Huang, PhD

The University of Texas at Arlington, 2012

Supervising Professor: David Hullender

A procedure for obtaining rational polynomial transfer functions representing ordinary differential equations for time domain transient response simulations of systems with fluid transmission lines with incompressible turbulent flow is formulated and presented. The method is based on the use of an inverse frequency least squares algorithm applied to the distributed parameter model for laminar flow with added lumped resistance to match the additional resistance associated with turbulence. Guidelines for using the algorithm are included with focus on the use of weighting parameters, the number of resonant modes to be included, and the bandwidth of the pressure/flow pulsation dynamics in the line. Frequency and time domain comparisons are presented demonstrating the accuracy of the transfer function formulations.

For special cases where the use of a lumped parameter model is preferred or is necessary for modeling and simulating the fluid pulsations in a line, guidelines for tapering the capacitance and inertance lump sizes for laminar and turbulent flow are presented. The improvement in matching the true mode frequencies and the improvement in the accuracy of transient responses achieved by tapering the size of the

lumps as compared to using equal size lumps is demonstrated for a variety of boundary conditions associated with different causalities.

The ability to obtain accurate rational polynomial transfer functions for pressure/flow pulsations in lines for both laminar and turbulent flow conditions, using either the inverse frequency algorithm or a tapered lumped model, makes it possible to include the line as a component in the model for a total dynamic system represented by ordinary differential equations for purposes of time domain simulations and the use of linear system analysis and control algorithms on the total system.

Table of Contents

List of Illustrations

List of Tables

Nomenclature

A Area of the line cross section, m^2

B_r Bessel function ratio defined by Equation (2.7)

$B_{r\sigma}$ Bessel function ratio with Prandtl number defined by Equation (2.8)

c Speed of sound in fluid, $=\sqrt{\beta/\rho}$ (m/s)

c_n Weight coefficient of line capacitance

C_s Sum capacitance of line

D Diameter of line (m)

D_n Dimensionless dissipation number defined by Equation (2.11)

f Friction factor defined by Equation (4.2)

i_n Weight coefficient of line inertance

I_s Sum inertance of line

$J_0()$ Bessel functions of zero order of the first kind

$J_1()$ Bessel functions of first order of the first kind

L Length of line (m)

N Number of approximation order

P_a Upstream pressure (N/m^2)

P_b Downstream pressure (N/m^2)

ΔP_a Sudden increase or decrease occurs in the pressure P_a

ΔP_b Sudden increase or decrease occurs in the pressure P_b

Q_a Upstream volumetric flow rate (m^3/s)

Q_b Downstream volumetric flow rate (m^3/s)

Q_e Equivalent steady state flow rate before time 0 (m^3/s)

ΔQ_a Sudden increase or decrease occurs in the flow Q_a

ΔQ_b Sudden increase or decrease occurs in the flow Q_b

r Internal radius of fluid line (m)

r_n Weight coefficient of line resistance

R_n Reynolds number

R_r Resistance ratio $= R_v/R_s$

R_s Sum resistance of line

R_t Equivalent resistance of line for turbulent flow defined by Equation (4.9)

R_v Linear resistance valve attached on the line

s Laplace operator

$\bar{s}$ Normalized Laplace operator $= (r^2/v)s$

t Time (sec)

U Flow speed (m/s)

Z_c Characteristic impedance of line defined by Equation (2.6)

Z_0 Impedance constant defined by Equation (2.9)

β Bulk modulus of fluid (N/m^2)

γ Specific heat ratio

Γ Propagation operator

μ Dynamic viscosity $= \rho\nu$

ν Kinematic viscosity (m^2/s)

ρ Density (kg/m^3)

σ Prandtl number

ω Frequency (rad/s)

ω_n Natural frequency (rad/s)

ω_v Viscous frequency $= \nu/r^2$

Chapter 1

Introduction

The subject of pressure and flow pulsations in fluid lines or pipes resulting from changes in the boundary conditions at the ends of the line has been researched and studied for decades [1, 2, 3, 4]. Consider the schematic of a fluid line shown in Figure 1.1. The pressures at the ends of the line are denoted by P_a and P_b; the volumetric flow rates at the ends are denoted by Q_a and Q_b. Sudden changes in these pressure or flow variables result in pressure and flow pulsations in the line. For example, assume that the flow through the line is initially constant or zero. Assume a sudden increase or decrease occurs in the pressure P_a, ΔP_a, associated with a control or measurement signal, or, consider a sudden increase in the flow Q_a, ΔQ_a, associated with a hydraulic fracturing process or 'fracking'. These boundary condition changes will cause pressure and flow pulsations along the line. Pulsations will also occur if there is a sudden decrease in the pressure P_b, ΔP_b. This decrease could be associated with a break in the line or the sudden opening of a valve causing a sudden increase in the flow Q_b, ΔQ_b. An additional example are pulsations associated with a sudden closing of a valve; in this case, the flow Q_b suddenly decreases or even goes to zero creating a blocked end condition on the line. It is not uncommon for the need to perform time domain simulations of these pulsations and also, time domain simulations of total systems in which these lines are internal system components.

Figure 1.1 Schematic of a fluid transmission line

There are numerous time domain methods for simulating the fluid dynamics in a line to analyze these pressure and flow pulsations. For example, for laminar flow conditions, the distributed parameter partial differential equations can be solved using the method-of-characteristics [5]. This technique provides an accurate solution but can be difficult to use for time varying boundary conditions and cannot be used if a line is an integral component in a total system in which eigenvalues and/or mode frequencies are to be analyzed. A more versatile solution technique is to approximate the laminar flow distributed parameter partial differential equations by ordinary differential equations using an inverse frequency technique first applied to fluid line dynamics by Wongputorn [6]. This approach allows for the boundary conditions to be variables which may be associated with other components such as actuators, valves, other fluid lines, etc.; this approach is particularly attractive when some of the other components are defined by ordinary differential equations. A review of the procedure for formulating ordinary differential equations for the pressure/flow pulsations in a line starting with the partial differential equations for the distributed parameter model is reviewed in Chapter 2 of this document.

Then there is the very popular approach of formulating lumped parameter models for the fluid dynamics in a line; the idea is to represent the resistance, inertance, and capacitance properties of the fluid in the line by a series of lumped resistances, inertances, and capacitances. Supposedly, the smaller the size of the lumps and the larger the number of lumps, the greater the accuracy of the overall model; as will be demonstrated in Chapter 3, however, the use of a larger number of smaller lumps does

improve the accuracy but does not necessarily achieve the true mode frequencies and is lacking when matching the true pressure/flow pulsation dynamics within the line.

In this author's opinion, the most versatile time domain method of modeling and simulating dynamic systems containing fluid lines is the method of approximating the distributed parameter equations for the line by ordinary differential equations [6, 8]. Unfortunately, to this date, the distributed parameter equations are for pulsations superimposed onto laminar flow conditions. For systems with fluid lines with turbulent flow, the lumped parameter modeling technique has been essentially the most common method for simulating these pressure/flow pulsations. Most likely, this is because the lumped parameter model approach, as well as the inverse frequency method, provides ordinary differential equations for the fluid lines that can be meshed with the algebraic and ordinary differential equations associated with the remaining components of a total system. The question arises, however, as to the best way to formulate the lumped parameter model especially if the flow through the lines is turbulent.

The overall objective of this research is to formulate a method for developing a reasonably accurate model for a dynamic system containing fluid lines with potentially turbulent flow. Throughout this study, numerous approaches to achieving this objective have been explored. However, as will be demonstrated, the most accurate approach found in this study seems to be to use a combination of a laminar flow distributed parameter model with a lumped resistance to account for the additional flow resistance associated with turbulence. For turbulent flow conditions, recent publications [7, 8] pertaining to steady turbulent flow in lines have introduced a way to formulate a more realistic turbulent resistance lumped model based on an empirical friction factor equation which is a function of the Reynolds number.

To develop an approach for using a combination of the distributed parameter model with a lumped resistance for turbulent conditions, Chapter 2 provides a review of the method for achieving ordinary differential equation approximations for the laminar flow distributed parameter model based on the inverse frequency concept [6, 8, 9]. As will be shown, the accuracy of a transfer function approximation depends on the choice of the frequency range to be matched, the weighting factor on each of the frequencies, and the order of numerator and denominator. The initial objective of using the inverse frequency concept was to formulate a method for tapering the size of the lumps within a lumped parameter laminar flow model in order to obtain a match of the dynamics with the distributed parameter model. The goal was to then extend the tapering algorithm to formulating lumped parameter models to lines with turbulent flow. Chapter 3 presents the results of attempting to use tapered lumped models to match the true fluid dynamics within a line with laminar flow.

As mentioned above, Chapter 3 focuses on the method to achieve an accurate lumped parameter model for the line with the size of the individual lumps tapered in order to achieve the true mode frequencies of the pressure/flow pulsations when superimposed on laminar steady flow conditions. The need to taper the size of the lumps in the model stems from the inability to achieve the true mode frequencies by simply increasing the number of lumps of equal size. The idea to taper the lumps in order to achieve the mode frequencies in fluid lines was first introduced by Paynter [10]. As will be demonstrated, the tapering method does allow one to match the true mode frequencies but the lumped model has limited accuracy due to the fact that a lumped model does not provide necessary numerator derivatives in the transfer functions. As mentioned, Chapter 3 focused on laminar flow conditions only. The objective of Chapter 4 is to extend the concepts in Chapter 3 to turbulent flow conditions.

The initial objective of research presented in Chapter 4 was to extend the tapered lumped model concepts to turbulent flow conditions by incorporating an additional lumped resistance based on an empirical friction factor model associated with turbulent flow. Tables with varied boundary conditions, fluid properties, and line parameters are provided for guidance on establishing levels of tapering. After formulating the tables for tapering lumps for lines with turbulent flow, however, it became obvious that the limitation of no input derivatives in the transfer functions could be avoided by using the transfer functions from the inverse frequency algorithm directly instead of attempting to match the model with a tapered lumped model. Thus, the objective changed to investigating how to use the inverse frequency algorithm directly for turbulent flow conditions. A recommended approach for modeling pressure/flow pulsations superimposed on steady turbulent flow is thus presented in Chapter 4. All of the analysis of Chapters 2 - 4 is applied to a specific set of boundary conditions consisting of ΔP_a as input pressure, ΔQ_b as output flow rate, and $\Delta P_b = 0$. These techniques are extended to other sets of potential boundary conditions in Chapter 5.

Three different cases of boundary conditions are analyzed for the distributed parameter and lumped parameter models in Chapter 5. As will be demonstrated, similarities in the resulting models for different boundary condition assumptions permits application of one set of results for a particular set of boundary conditions to the model for a different set of boundary conditions. As will be demonstrated in Chapter 5, results for only two types of model formulations are required for showing the similarities and differences in the form of the distributed parameter model and the lumped parameter model. Consequently, it is possible to use suggested coefficients for tapering the lumps for more than a single set of boundary conditions. The tapering coefficients for lumped

models are formulated and listed in tables along with assumed lumped resistances that account for turbulent flow conditions.

The conclusions and recommendations for future research are included in Chapter 6.

Chapter 2

Rational Polynomial Approximations

Distributed parameter models for pressure and flow pulsations in lines for laminar flow conditions have been tested and proven to be very accurate using graphical and frequency domain analysis techniques. However, it is often desirable to obtain ordinary differential equations for these pulsations in lines for purposes of time domain simulations in special cases. There have been many algorithms presented for approximating the distributed parameter equations by ordinary differential equations depending on various simplifying assumptions [1, 4]. It appears that the most versatile and consistently accurate approach, however, is based on an inverse frequency technique formulated by Levi [9] and then applied to the transmission line problem by Wongputorn [6, 8]. This chapter is devoted to reviewing the application of this inverse frequency method to fluid line dynamics with emphasis on techniques for fine tuning the process of generating ordinary differential equations from the partial differential equations.

2.1 Distributed Parameter Model

In reality, the inertance, capacitance, and resistance of a fluid transmission line are continuously distributed from beginning to end and defined by partial differential equations for laminar flow conditions. This is in contrast to the lumped parameter model, which assumes the inertance, capacitance, and resistance are lumped independently at discrete locations.

The distributed parameter model is derived by considering a control volume of cross section of a round tube. To develop its general equations, the following assumptions are made.

1. Rigid walls

2. Uniform cross section line

3. One dimensional flow

4. Fluid velocity is small compared to the speed of sound in the line

Assume the following boundary conditions:

$$\begin{aligned} P(0,t) &= P_a \quad Q(0,t) = Q_a \\ P(L,t) &= P_b \quad Q(L,t) = Q_b \end{aligned} \tag{2.1}$$

As documented in Ref. [1], the corresponding solution of the partial differential equations for the distributed parameter model in matrix form is shown below.

$$\begin{bmatrix} Q_a \\ Q_b \end{bmatrix} = \begin{bmatrix} \dfrac{\cosh\Gamma}{Z_c \sinh\Gamma} & -\dfrac{1}{Z_c \sinh\Gamma} \\ \dfrac{1}{Z_c \sinh\Gamma} & -\dfrac{\cosh\Gamma}{Z_c \sinh\Gamma} \end{bmatrix} \begin{bmatrix} P_a \\ P_b \end{bmatrix} \tag{2.2}$$

The equations for the propagation operator and characteristic impedance vary depending on assumptions regarding friction, heat transfer, and compressibility. For example, the lossless model [1, 4] is assumes that there is no dissipation due to viscosity and heat transfer. So the propagation operator is

$$\Gamma = \frac{L}{c} s \tag{2.3}$$

and the characteristic impedance is

$$Z_c = \frac{\beta}{Ac} = \frac{\rho c}{A} \tag{2.4}$$

For the case of the dissipative model [1, 4, 11, 12] that includes viscosity and heat transfer, the propagation operator and the characteristic impedance become,

$$\Gamma = \frac{Ls}{c} \sqrt{\frac{1+(\gamma-1)B_{r\sigma}}{1-B_r}} \tag{2.5}$$

$$Z_c = \frac{Z_0}{\sqrt{(1-B_r)(1+(\gamma-1)B_{r\sigma})}} \tag{2.6}$$

where

$$B_r = \frac{2J_1(j\sqrt{\frac{r^2 s}{\nu}})}{j\sqrt{\frac{r^2 s}{\nu}} \cdot J_0(j\sqrt{\frac{r^2 s}{\nu}})} \tag{2.7}$$

$$B_{r\sigma} = \frac{2J_1(j\sqrt{\frac{\sigma r^2 s}{\nu}})}{j\sqrt{\frac{\sigma r^2 s}{\nu}} \cdot J_0(j\sqrt{\frac{\sigma r^2 s}{\nu}})} \tag{2.8}$$

$$Z_0 = \frac{\rho c}{A} \tag{2.9}$$

Here we introduce the dimensionless dissipation number [13, 14], D_n, defined by Equation (2.11) and the normalized Laplace operator, $\bar{s}$, and viscous frequency, ω_ν, both defined by Equation (2.12) which is used to simplify the equation. The equation for the propagation operator becomes

$$\Gamma = D_n \bar{s} \sqrt{\frac{1+(\gamma-1)B_{r\sigma}}{1-B_r}} \tag{2.10}$$

where

$$D_n = \frac{\nu L}{c r^2} \tag{2.11}$$

$$\bar{s} = \frac{r^2}{\nu} s = \frac{s}{\omega_\nu} \tag{2.12}$$

$$B_r = \frac{2J_1(j\sqrt{\bar{s}})}{j\sqrt{\bar{s}} \cdot J_0(j\sqrt{\bar{s}})} \tag{2.13}$$

$$B_{r\sigma} = \frac{2J_1(j\sqrt{\sigma\bar{s}})}{j\sqrt{\sigma\bar{s}} \cdot J_0(j\sqrt{\sigma\bar{s}})} \tag{2.14}$$

$$c = \sqrt{\frac{\beta}{\rho}} \tag{2.15}$$

When formulating rational polynomial approximations for the fluid transmission line model, the poles and zeros depend on the magnitude of the dissipation number D_n. In other words, the transfer function of the model is a function of D_n.

The remainder of Chapter 2 and Chapters 3 - 4 focus on a specific set of boundary conditions which have ΔP_a as input pressure with $\Delta P_b = 0$, and ΔQ_b & ΔQ_a as output flow rates. The distributed parameter model equation becomes

$$\Delta Q_b = \left[\frac{1}{Z\sinh\Gamma}\right]\Delta P_a \quad \text{and} \quad \Delta Q_a = \left[\frac{\cosh\Gamma}{Z\sinh\Gamma}\right]\Delta P_a$$

But in Chapter 5, we will introduce two other boundary condition formulations and address how to apply the techniques in Chapters 2 - 4 to these other cases. One of these cases has ΔP_b as input with $\Delta P_a = 0$, and ΔQ_a & ΔQ_b as outputs. So the distributed parameter model equation will become

$$\Delta Q_a = \left[\frac{-1}{Z\sinh\Gamma}\right]\Delta P_b \quad \text{and} \quad \Delta Q_b = \left[\frac{-\cosh\Gamma}{Z\sinh\Gamma}\right]\Delta P_b$$

Another case has ΔQ_b as input with $\Delta P_a = 0$, and ΔQ_a & ΔP_b as outputs. Its distributed parameter model equation will become

$$\Delta Q_a = \left[\frac{1}{\cosh \Gamma}\right]\Delta Q_b \quad \text{and} \quad \Delta P_b = \left[\frac{-Z \sinh \Gamma}{\cosh \Gamma}\right]\Delta Q_b$$

As will be explained and demonstrated in Chapter 5, the similarities of these transfer functions for different inputs and outputs makes it possible to apply results for model tapering to more than one set of boundary conditions.

2.2Curve Fit Approximation of the Distributed Parameter Model

In references [6, 8], a new approach is introduced to obtain rational polynomial transfer function approximations that very accurately match the frequency response properties of the original function over a designated frequency band. This approach is to match the frequency response using the least squares algorithm of Levi [9]. The actual function is represented by frequency response data of the system taken from measurements or generated directly from an analytical function.

For the case of the dissipative distributed parameter model of the circular rigid fluid transmission line, this approach will match this frequency response data generated from Equation (2.2). These techniques are utilized in an m-file named ‘invfreqs’ available in the Signal Processing toolbox within MATLAB. By formulating a new transfer function in MATLAB representing an ordinary differential equation, the results allow for time domain simulations and analysis techniques. Thus, the objective is to obtain a transfer function approximation representing an ordinary differential equation that accurately models the pressure and flow dynamics by performing an inverse frequency least squares curve fit to the frequency response of a transfer function representing the solution to the partial differential equations. The approximation will be a ratio of rational polynomials of the Laplace operator s or a normalized version of s.

To demonstrate the process, consider the following example. The flow through the line is initially steady with upstream pressure $P_a = P_i$ and downstream pressure

$P_b = P_o$. Pressure/flow pulsations will be created from a sudden increase or decrease in pressure ΔP_a in P_a. That is, after time 0, $P_a = P_i + \Delta P_a$. The schematic is shown in Figure 2.1.

Assuming that the downstream pressure remains constant, the flow rate ΔQ_b becomes

$$\Delta Q_b(\bar{s}) = \left[\frac{1}{Z \sinh \Gamma}\right] \Delta P_a(\bar{s}) \tag{2.16}$$

Normalizing $\Delta Q_b(\bar{s})$ by steady flow rate $Q_e = (P_i - P_o)/R_s$, where R_s is the laminar steady flow line resistance, gives

$$\begin{aligned} \frac{\Delta Q_b(\bar{s})}{Q_e} &= \left[\frac{1}{Z \sinh \Gamma}\right] \frac{\Delta P_a(\bar{s})}{(P_i - P_o)/R_s} \\ &= \left[\frac{R_s}{Z \sinh \Gamma}\right] \frac{\Delta P_a(\bar{s})}{(P_i - P_o)} \end{aligned} \tag{2.17}$$

The transfer function for the normalized flow rate and the normalized pressure input simplifies to

$$\frac{\Delta Q_b(\bar{s})/Q_e}{\Delta P_a(\bar{s})/(P_s - P_o)} = \frac{R_s}{Z \sinh \Gamma} \tag{2.18}$$

Figure 2.1 Schematic of a fluid transmission line for laminar flow

As mentioned above, the objective is to obtain a rational polynomial approximation for the transfer function such as in Equation (2.17). An approximation that

will provide accurate time domain simulations should include all modes out through the bandwidth of the original transfer function and should have the correct DC gain.

Since the transfer functions obtained from solving the partial differential equations are not linear, the easiest way determine the bandwidth and the number of modes that need to be included is by trial and error using an educated guess for the number of modes, the order of the system, and the mode frequencies. Consider the following example.

The fluid used in this example is MIL-L-87257, whose properties are defined in the MATLAB Hydraulic Utilities Library. These properties and the dimensions of the line are listed below.

Line Properties:

D = 0.003175 Diameter of line, m

L = 20 Length of line, m

T1 = 27 Avg. temperature, ℃

Fluid Properties:

Fluid MIL-F-87257

KVis = 7.6179e-006 Kinematic viscosity, m^2/s

Den = 855.24 Density, kg/m^3

Bulk = 1.8246e+009 Bulk modulus, N/m^2

If all of the modes are 2nd order thus representing complex eigenvalues, the order N of the transfer function approximation needs to be at least two times the number of modes that need to be included. Considering that some of the modes will be 1st order representing real eigenvalues, the number of modes in the approximation will be slightly greater than N/2. The only way to really know the number of modes needed and the frequencies involved is to start with educated guesses. It may prove beneficial to use

equations for the mode frequencies. In Hullender's paper [13, 14], equations for approximating the mode natural frequencies and damping ratios for air and liquids are presented. For a liquid, the equation is

$$\log_{10}\omega_n = 1.0178\log_{10}\lambda_c + 0.42966 \tag{2.19}$$

where

$$\lambda_c = \left(i - \frac{1}{2}\right) \Big/ D_n \quad i = 1, 2, 3, \ldots \tag{2.20}$$

Using Equation (2.19) with the fluid properties and line dimensions in the example above gives the first 6 natural frequencies to be 68.767, 139.24, 210.37, 281.94, 353.83, and 425.97 rad/sec. Until a frequency response is generated, however, a reasonable estimate of the number of modes needed and the bandwidth will remain unknown. For instance, assume that the bandwidth is between 281 and 353 rad/sec. We will designate this frequency by $\omega_{\max}$.

In this case, the transfer function must accurately approximate the first four 2nd order modes and any significant 1st order modes in this frequency range. Thus, a 10th order transfer function is selected; for this designated frequency range, a 10th order will accommodate two first order modes in addition to the four 2nd order modes.

To insure that an accurate DC gain is achieved, a lower bound on the frequency range must also be assumed; for this example, $\omega_{\min} = 0.11\ rad/\text{sec}$ is chosen. Finally, the order of the numerator of the transfer function approximation must be chosen. In all probability, the most accurate approximation will be achieved with a numerator order N-1.

Adjustment of $\omega_{\max}$ may be needed depending on the comparison result of the figure of frequency responses. The second consideration of accuracy is that the DC gain should be as close to 1 as possible while adjusting the maximum frequency $\omega_{\max}$.

The MATLAB command for converting the frequency response data to a transfer function is

$$[\text{num}, \text{den}] = \text{invfreqs}(\text{h}, \text{w}, \text{n}, \text{m}, \text{wt}, \text{iter})$$

Where 'h' is the frequency response data corresponding to the frequency data 'w', associated with the transfer function in Equation (2.18). The 'wt' is a vector of weighting factors and has the same length as 'w'. The 'wt' can be substituted with '[ ]' to obtain a weighting vector of all ones. The 'iter' parameter tells "invfreqs" to end the iteration when the solution has converged, or after 'iter' iterations, whichever comes first.

Using the MATLAB command above, the 9th/10th order transfer function approximation for the fluid and line properties listed above is

$$\frac{\Delta Q_b(\bar{s})}{Q_e} = \frac{3.454\bar{s}^9 + \cdots + 2.845\times10^{18}\bar{s} + 5.463\times10^{19}}{\bar{s}^{10} + \cdots + 1.206\times10^{19}\bar{s} + 5.461\times10^{19}} \frac{\Delta P_a(\bar{s})}{(P_i - P_o)} \tag{2.21}$$

A comparison of the frequency response plot of the original transfer function and the 10th order approximation listed above is shown in Figure 2.2. Note the accuracy of the approximation at all frequencies including the resonant peaks out through the 4th second order mode. Also note that the 4th resonant peak is at about -5 dB which is slightly below the standard bandwidth cutoff of -3 dB. It is also important to note that the DC gain is of the approximation is 1.0003 which is essentially equal to the true value of 1.

The eigenvalues, damping ratios, and natural frequencies of the approximation are listed below. It is of interest to compare the mode frequencies with the values predicted using Equation (2.19). The frequencies are reasonably close to the estimated

values. A comparison of these values with transfer functions corresponding to lumped parameter models with zero order numerators will be of interest in the next section.

Eigenvalue	Damping	Freq. (rad/s)
-5.77e+000		5.77e+000
-2.36e+001		2.36e+001
-6.74e+000 + 6.98e+001i	9.62e-002	7.01e+001
-6.74e+000 - 6.98e+001i	9.62e-002	7.01e+001
-8.92e+000 + 1.43e+002i	6.23e-002	1.43e+002
-8.92e+000 - 1.43e+002i	6.23e-002	1.43e+002
-1.19e+001 + 2.17e+002i	5.50e-002	2.17e+002
-1.19e+001 - 2.17e+002i	5.50e-002	2.17e+002
-1.14e+001 + 2.93e+002i	3.89e-002	2.93e+002
-1.14e+001 - 2.93e+002i	3.89e-002	2.93e+002

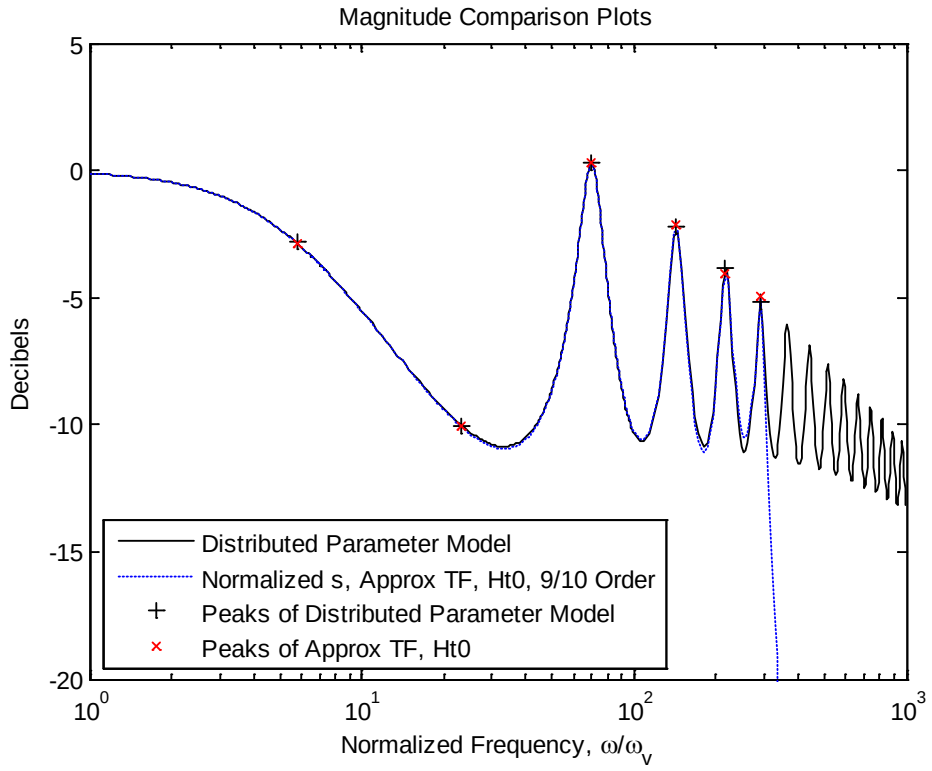


Figure 2.2 Comparison of frequency responses of distributed parameter model and $9^{th}/10^{th}$ order approximation model

A quantitative comparison of the resonant peaks is provided in Table 2.1.

Table 2.1 Comparing the resonant peaks of the distributed parameter and $9^{th}/10^{th}$ models

Normalized resonant frequencies	69.759	142.97	216.91	292.92
Resonant peaks of distributed model (dB)	0.32074	-2.2435	-3.8644	-5.1345
Resonant peaks of transfer function approximation (dB)	0.28908	-2.1411	-4.0538	-4.9807

2.3 Zero Order Numerator Transfer Function Approximation

The use of lumped parameter models for line dynamics is very popular. However, it can be shown that the transfer function for a lumped parameter model will always have

a zero order numerator. Consequently, any study pertaining to the number and size of lumps to use in a lumped model for a line must start with the assumption that the order of the numerator is zero. Thus, to match the transfer function coefficients of the lumped parameter model from the frequency response of the distributed parameter model, the order of the transfer function of the curve fit approximation will be set to zero.

In section 2.2, a 10^{th} order approximation was obtained for the purpose of having a transfer function that can be used to accurately model the time response of all of the modes with modal frequencies within the bandwidth of the system. This model is thus assumed to provide an 'exact' or 'true' time response for comparison with lower order transfer functions without numerator dynamics such as occur with lumped parameter models for the line. For instance, a lumped parameter line model with two lumped capacitances and two lumped inertances will be 4^{th} order with a 0^{th} order numerator. For the line and fluid properties in the example in section 2.2, a 4^{th} order transfer function will at best accurately model only one resonant mode.

It is not uncommon to get an error warning from MATLAB while using the m-file of 'invfreqs' due to the restriction of a zero order numerator in the transfer function. The best way to avoid this error is to adjust the weighting factors wt by increasing the weight around the first real eigenvalue and mode frequencies. For example, the weighting factors are set to 2 for the frequency range of 2~30 $rad/\sec$, and the rest set to 1. The frequency range of the approximation is set from 0.11 to 87 $rad/\sec$. The results are as follows:

DCgainHt1 = 0.99236

Transfer function:

$$\frac{\Delta Q_b(\bar{s})}{Q_e} = \frac{2.863\times10^7}{\bar{s}^4+867.6\bar{s}^3+1.688\times10^4\bar{s}^2+4.336\times10^6\bar{s}+2.885\times10^7}\frac{\Delta P_a(\bar{s})}{(P_i-P_o)} \quad (2.22)$$

Comparing the first mode eigenvalues, frequencies, damping ratios, and frequency response plots (Figure 2.2 and Figure 2.3) to those of the 10th order model reveals that the results are reasonably accurate.

Eigenvalue	Damping	Freq. (rad/s)
-6.72e+000		6.72e+000
-3.53e+000 + 7.05e+001i	5.00e-002	7.06e+001
-3.53e+000 - 7.05e+001i	5.00e-002	7.06e+001
-8.84e+002		8.84e+002

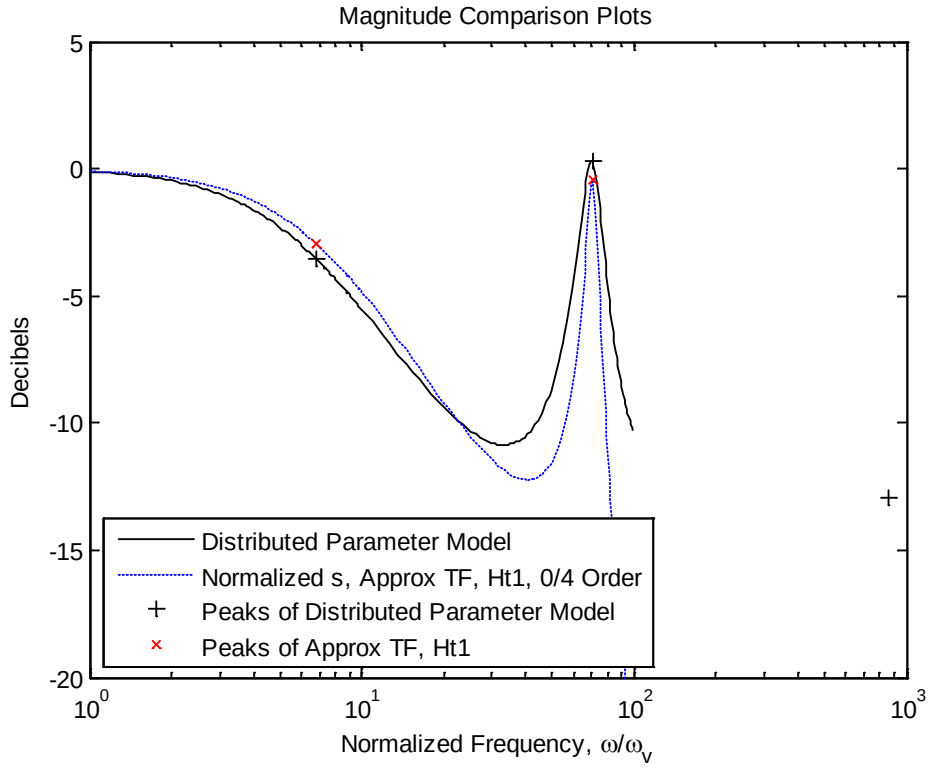


Figure 2.3 Comparison of frequency responses of distributed parameter model and 0/4th order approximation model

A quantitative comparison of the resonant peaks of the first 2nd order mode is provided in Table 2.2. The error is seen to be 0.26866-(-0.46306)=0.73172 dB which is relatively very accurate.

Table 2.2 Magnitude of distributed parameter model and 0/4th order TF approximation

Normalized resonant frequencies	70.556
Resonant peaks of distributed model (dB)	0.26866
Resonant peaks of transfer function approximation (dB)	-0.46306

In Figure 2.4, we see that the 4th order approximation does not exactly match the 10th order approximation mainly due to the 0 order numerator of the transfer function. However, its mode frequencies and DC gain are very accurate.

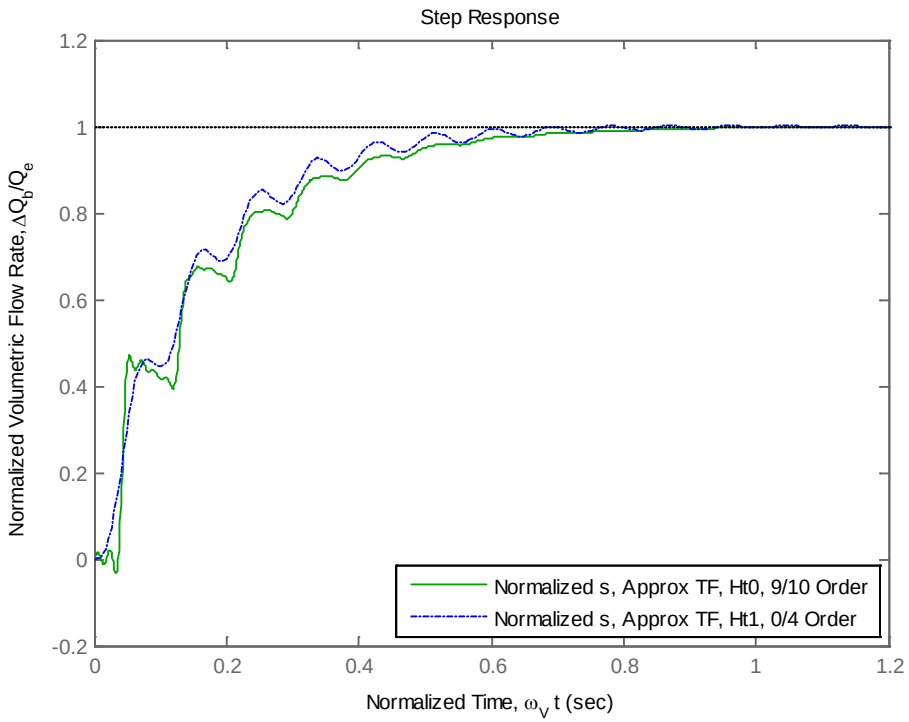


Figure 2.4 Comparison of step response of 9th/10th order and 0/4th order transfer function approximation

This chapter has focused on the procedure for generating a rational polynomial transfer function approximation for a partial differential equation transfer function defined in the frequency domain. A 10^{th} order transfer function representing the 'exact' or 'true' model was generated for a specific set of line dimensions and fluid properties. This model was then used to determine the accuracy and limitations of lower order transfer functions as might be formulated from lumped parameter models. Chapter 3 pertains to methods for formulating tapered lumped models and higher order transfer functions such as the 10^{th} order obtained in this section will be used as a baseline for determining the accuracy of the tapered models in the time domain.

Chapter 3

Lumped Parameter Model

For pressure/flow pulsations in lines with turbulent flow conditions, the most accurate method for formulating the model is debatable. Previous studies proposed that inertance and capacitance properties of the line do not change with the flow being laminar or turbulent but only that the resistance will change with an increase in the Reynolds number. The approach in this study is to simply add additional lumped resistance to the distributed parameter model for a line; the additional resistance is sufficient to produce the total steady flow resistance for turbulent flow through the line. In all probability, the most accurate approach is to distribute the resistance in one or more lumps along the line; use of more than one lump, however, requires that the distributed parameter model be divided into distributed segments. The scope of this study is limited to a single distributed parameter segment with a single lumped resistance for the turbulence; a thorough understanding of this relatively simple model is of interest before extending the research to a more complex segmented model.

A popular approach to modeling the pressure/flow dynamics in lines with turbulence is the use of a lumped parameter model. The model consists of a lumped inertance in series with a lumped capacitance in series with a lumped resistance. The actual ordering of the lumps may vary depending on the causalities associated with the boundary conditions; however, the resulting transfer functions always end up with a zero order numerator which limits the accuracy of the models.

The objective of the following section of this document is to explore the improvement in achieving the true mode frequencies and more accurate transient responses using tapered lumps in comparison to the traditional method of equal sized

lumps. Guidelines for tapering the lumps as a function of the dissipation number D_n are presented.

A lumped parameter model assumes that the effects of inertance, capacitance, and resistance can be treated as being concentrated separately in the line.

The Figure 3.1 shows a typical lumped parameter model, which has a single lump for the line inertance, a single lump for the line capacitance, and a single lump for the line resistance; all the impedances are in series. And Figure 3.2 is a supposedly more accurate model that has two times the number of lumps equally weighted.

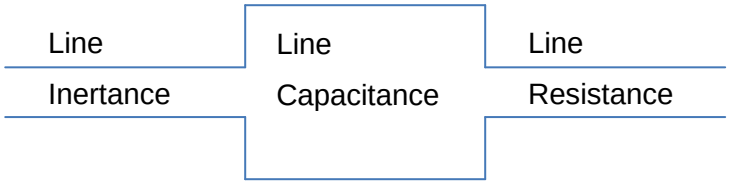


Figure 3.1 Typical single lumped parameter model

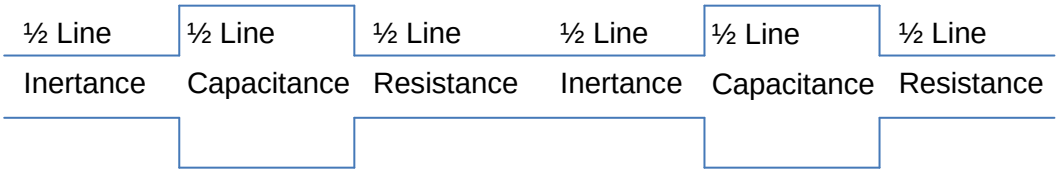


Figure 3.2 Typical two lumped parameter model

3.1 Equal Lumped Parameter Model

Traditionally, if we have more than one lumped parameter for inertance, capacitance, and resistance, the value of each lump is assumed to be the same respectively. A better result would be expected when the numbers of lumped parameter are increased. However, it may not give a good result due to its incorrect mode frequencies.

The schematic of the 4th order lumped parameter model shown in Figure 3.3 is used for an example to demonstrate the above phenomenon. Although the location and number of lumped resistances does affect the result but not the order of the transfer function, a single resistance lump will be placed at the right end of the line in the examples to follow.

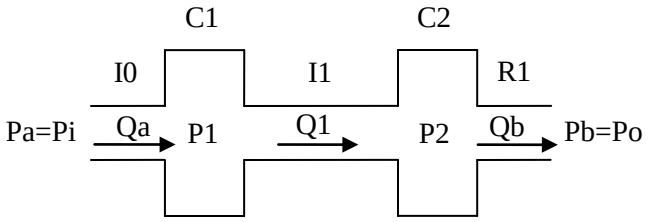


Figure 3.3 0/4th order lumped parameter model

We can derive the symbolic transfer function:

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{R_1}{\left(\mathrm{C_1C_2I_0I_1R_1}\right)\mathrm{s}^4 + \left(\mathrm{C_1I_0I_1}\right)\mathrm{s}^3 + \left(\mathrm{C_1I_0R_1 + C_2I_0R_1 + C_2I_1R_1}\right)s^2 + \left(I_0 + I_1\right)s + R_1} \frac{\Delta P_a(s)}{\left(P_i - P_o\right)} \tag{3.1}$$

Let $I_0 = I_1 = 1/2 \times I_s$, $C_1 = C_2 = 1/2 \times C_s$, and $R_1 = R_s$ with the line and fluid properties used in Chapter 2, the 4th order equal lumped parameter model becomes,

Transfer function:

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{5.2244e010}{114.79s^4 + 50633.0s^3 + 7.3466e006s^2 + 2.1604e009s + 5.2244e010} \frac{\Delta P_a(s)}{\left(P_i - P_o\right)} \tag{3.2}$$

Eigenvalue	Damping	Freq. (rad/s)
-2.61e+001		2.61e+001
-1.12e+001 + 2.10e+002i	5.30e-002	2.11e+002
-1.12e+001 - 2.10e+002i	5.30e-002	2.11e+002

-3.93e+002	3.93e+002

In order to compare the results of an equal lumped parameter model with the transfer function approximation of the distributed parameter model, the process of un-normalization of $\bar{s}$ in both approximate transfer functions is needed. That is to convert $\bar{s}$ to s, which $\bar{s} = s \times (r^2/\nu)$. Then the results of the example in Chapter 2 become

10th order Transfer Function Approximation:

Transfer function:

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{3.001\times10^{-24}s^9 + \cdots + 0.01723s + 1}{2.875\times10^{-25}s^{10} + \cdots + 0.07307s + 1}\frac{\Delta P_a(s)}{(P_i - P_o)} \tag{3.3}$$

Eigenvalue	Damping	Freq. (rad/s)
-1.74e+001		1.74e+001
-7.00e+001		7.00e+001
-2.04e+001 + 2.11e+002i	9.62e-002	2.12e+002
-2.04e+001 - 2.11e+002i	9.62e-002	2.12e+002
-2.71e+001 + 4.32e+002i	6.26e-002	4.33e+002
-2.71e+001 - 4.32e+002i	6.26e-002	4.33e+002
-3.59e+001 + 6.56e+002i	5.47e-002	6.57e+002
-3.59e+001 - 6.56e+002i	5.47e-002	6.57e+002
-3.42e+001 + 8.85e+002i	3.86e-002	8.86e+002
-3.42e+001 - 8.85e+002i	3.86e-002	8.86e+002

4th order Transfer Function Approximation:

Transfer function:

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{1}{4.152\times10^{-10}s^4 + 1.089\times10^{-6}s^3 + 6.404\times10^{-5}s^2 + 0.04972s + 1}\frac{\Delta P_a(s)}{(P_i - P_o)} \tag{3.4}$$

Eigenvalue	Damping	Freq. (rad/s)
-2.05e+001		2.05e+001
-1.07e+001 + 2.13e+002i	5.03e-002	2.14e+002
-1.07e+001 - 2.13e+002i	5.03e-002	2.14e+002
-2.58e+003		2.58e+003

From above data, we can see the mode frequencies of the equal lumped parameter model and the first 2^{nd} order mode of the $9^{th}/10^{th}$ order approximate transfer function are reasonably close. In addition, the equal lumped parameter model has a larger error in the first real eigenvalue. Figure 3.4 shows that the step response of the equal lumped parameter model does not match well with the $9^{th}/10^{th}$ order approximate transfer function. The $0/4^{th}$ order approximate transfer function time response is shown for comparison. Frequency domain comparisons of the transfer functions are shown in Figure 3.5.

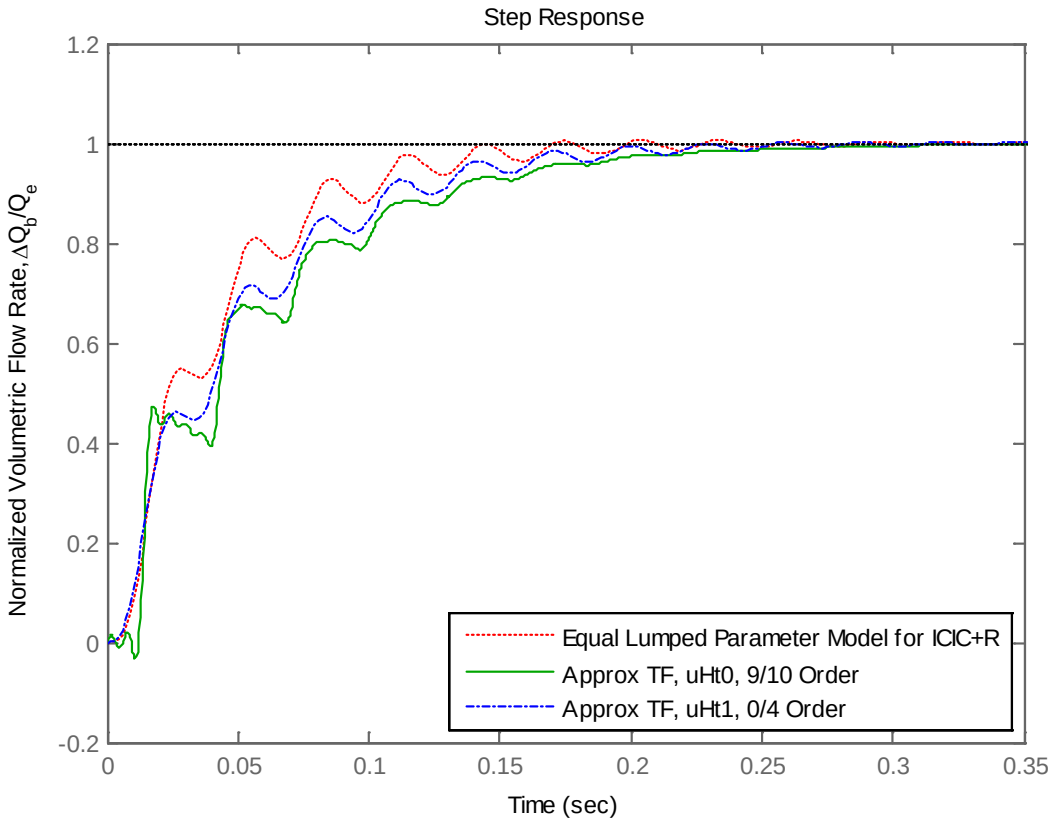


Figure 3.4 Comparison of step responses of equal lumped parameter model, $9^{th}/10^{th}$ order and $0/4^{th}$ order approximate transfer function model

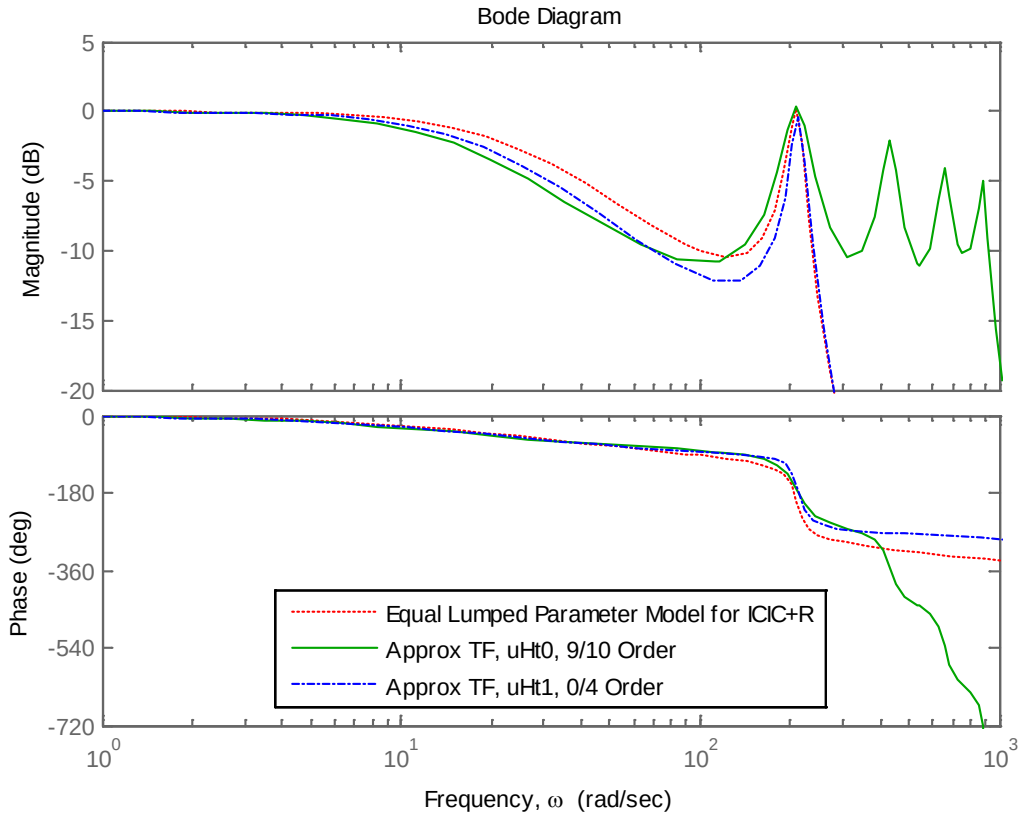


Figure 3.5 Comparison of bode plot of equal lumped parameter model, $9^{th}/10^{th}$ order and $0/4^{th}$ order approximate transfer function

3.2 Tapered Lumped Parameter Model with Lossless Effect

To improve the lumped model, Professor Paynter's patent [10], which allows the model to have only a few lumps but hit frequencies exactly for zero resistance, is utilized. Figure 3.6 shows a network diagram for an n-segment tapered model.

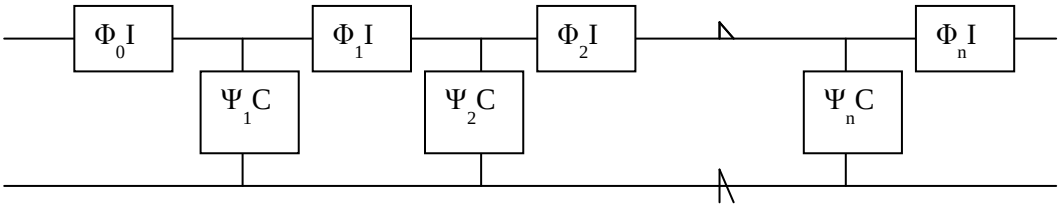


Figure 3.6 Network diagram for an n-segment tapered model

The values of the Φ's and Ψ's are dependent on the number of segments, n. Values of Φ and Ψ for n up to 4 are given in below Table.

Table 3.1 Taper coefficients Φ and Ψ for Paynter's model

n->	1	2	3	4
Φ_0	1.000	.250	.142	.099
Ψ_1		.541	.289	.199
Φ_1		.750	.311	.205
Ψ_2			.367	.218
Φ_2			.547	.244
Ψ_3				.295
Φ_3				.452

The lossless model of the distributed parameter model will be used to compare with the Paynter's lumped parameter model. The propagation operator and the characteristic impedance from Equations (2.3) and (2.4) will be applied to the lossless model.

An example is shown below to demonstrate the comparisons of the model of a $8^{th}/9^{th}$ order approximate transfer function, the equal lumped parameter model of $0/5^{th}$ order transfer function, and Paynter's tapered lumped parameter model of $0/5^{th}$ order transfer function. The model of $8^{th}/9^{th}$ order approximate transfer function will be assumed to be the exact model of the line.

To get the transfer function of lossless effect for equal lumped parameter model and Paynter's tapered lumped parameter model, the symbolic transfer function of the lumped parameter mode is needed and shown below.

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{1}{(C_1C_2I_0I_1I_2)s^5 + (C_1I_0I_1 + C_1I_0I_2 + C_2I_0I_2 + C_2I_1I_2)s^3 + (I_0 + I_1 + I_2)s} \frac{\Delta P_a(s)}{(P_i - P_o)} \tag{3.5}$$

The properties of the line and fluid are the same as those in the example in Chapter 2 except for the viscosity of the fluid; it is assumed to be zero as used in Equation (3.5). The transfer functions and mode frequencies of each model are presented below.

$8^{th}/9^{th}$ order Transfer Function Approximation of Distributed Parameter Model

Transfer function:

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{3.141\times10^{-9}s^8 + \cdots + 2.029\times10^{-11}s + 1.379\times10^{13}}{6.734s^9 + 4.42\times10^{-20}s^8 + \cdots + 0.0002587s^2 + 2.978\times10^{22}s + 1} \frac{\Delta P_a(s)}{(P_i - P_o)} \tag{3.6}$$

Mode frequencies (rad/s) = [229, 459, 688, 918]

Equal Lumped Parameter Model: ICICI, 5th order

Transfer function:

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{1}{0.7032s^5 + 9.001e004s^3 + 2.16e009s} \frac{\Delta P_a(s)}{(P_i - P_o)} \tag{3.7}$$

Mode frequencies (rad/s) = [179, 310]

Paynter's Tapered Lumped Parameter Model: ICICI, 5th order

Transfer function:

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{1}{0.1946s^5 + 5.11e004s^3 + 2.16e009s} \frac{\Delta P_a(s)}{(P_i - P_o)} \tag{3.8}$$

Mode frequencies (rad/s) = [230, 458]

From the above data and Figure 3.7, we can see the mode frequencies of the Paynter's tapered lumped parameter model match well with the $8^{th}/9^{th}$ order approximate transfer function of lossless distributed parameter model, but the equal lumped parameter model does not.

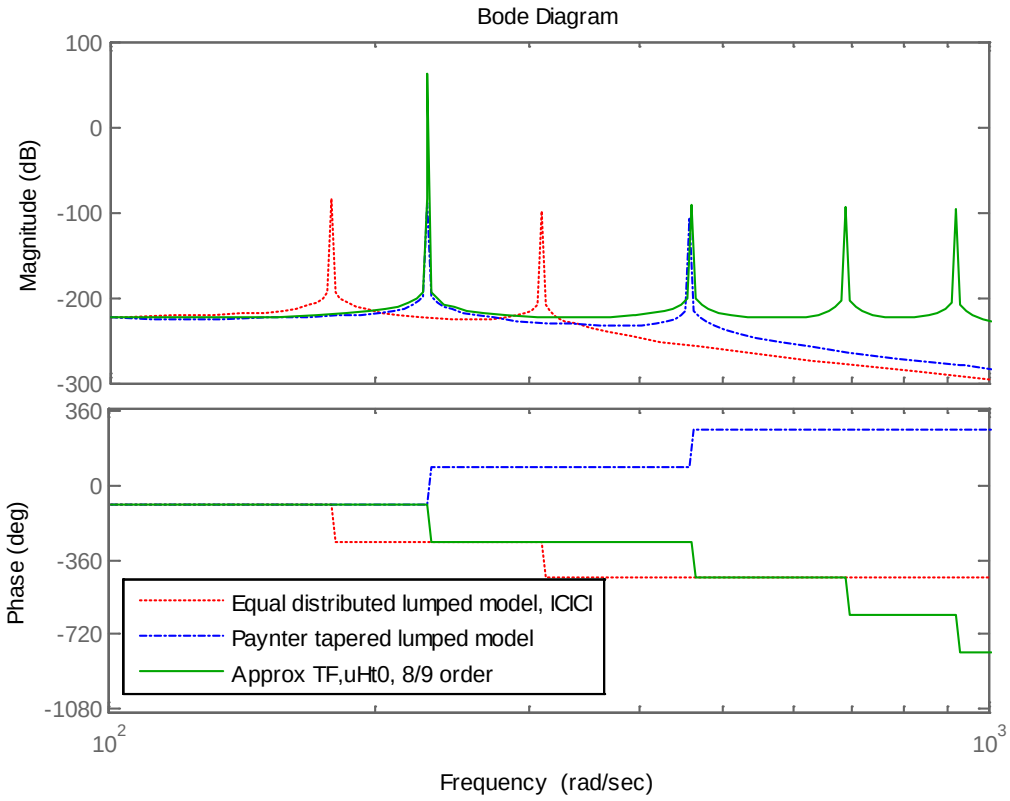


Figure 3.7 Bode plot of $8^{th}/9^{th}$ order approximate transfer function model and $0/5^{th}$ order transfer functions of both equal and Paynter's tapered lumped parameter models

This approves the Paynter's tapered lumped parameter model does achieve the true mode frequencies for lossless models. How about the models with resistance? Dr. Paynter suggests attaching the resistance at the end of the lossless model. The next section demonstrates the results.

3.3 Tapered Lumped Parameter Model with Viscosity Effect

To apply the viscosity effect to Paynter's tapered lumped parameter model, we can simply add the resistance at the end of the line. The following shows the transfer function, eigenvalues, damping ratios, and mode frequencies of Paynter's tapered lumped parameter model.

Paynter's Tapered Lumped Parameter Model with Resistance: ICIR, 0/3rd order

Transfer function:

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{1}{7.865\times10^{-7}s^3 + 2.536\times10^{-5}s^2 + 0.04135s + 1}\frac{\Delta P_a(s)}{(P_i - P_o)} \tag{3.9}$$

Eigenvalue	Damping	Freq. (rad/s)
-2.43e+001		2.43e+001
-3.99e+000 + 2.29e+002i	1.74e-002	2.29e+002
-3.99e+000 - 2.29e+002i	1.74e-002	2.29e+002

In Chapter 3.1, the Equation (3.3) and data of the 10th order approximate transfer function of distributed parameter model presents an accurate model. By comparing above data with the accurate model, Paynter's model with viscosity effect obviously has larger errors for the eigenvalues and mode frequencies than the 5th order approximate transfer function shown in Equation (3.4).

Figure 3.8 and Figure 3.9 show the inaccuracy of Paynter's lumped model with viscosity effect in time response. Also, Figure 3.10 clearly shows the first mode frequency of his model does not match the accurate model at all.

Therefore, we can conclude that the tapering of the lumped parameter model should change with the resistance

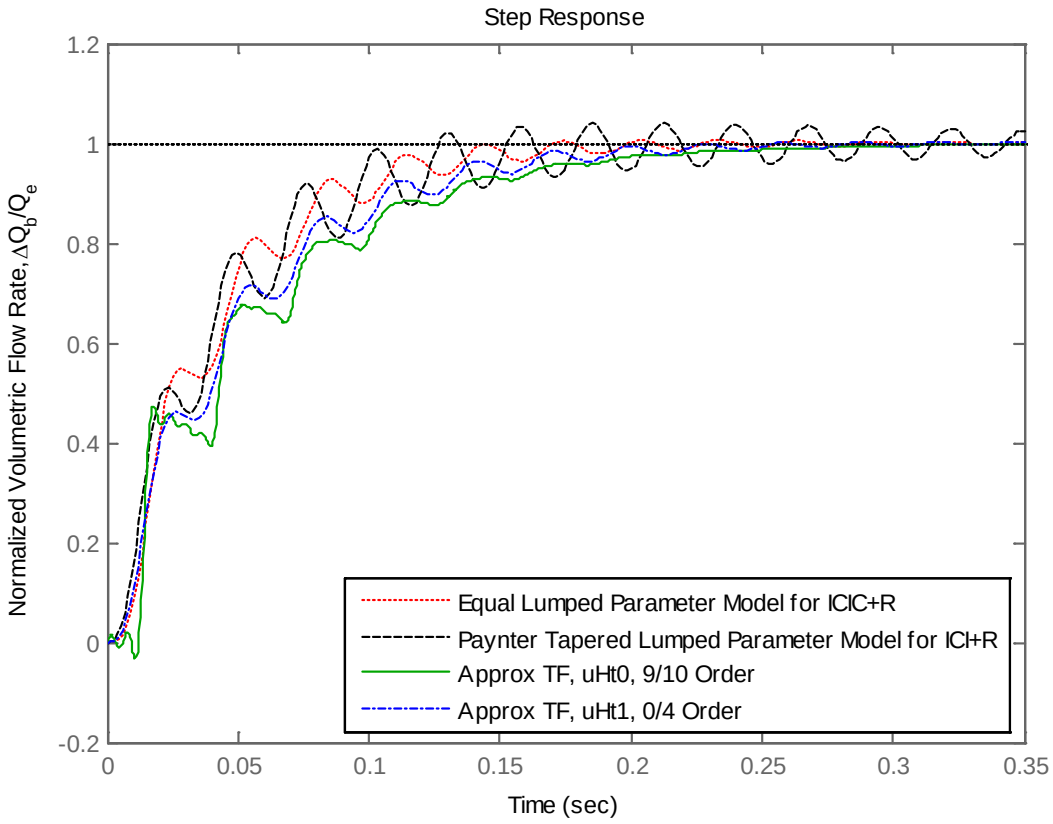


Figure 3.8 Comparison of step responses (a)

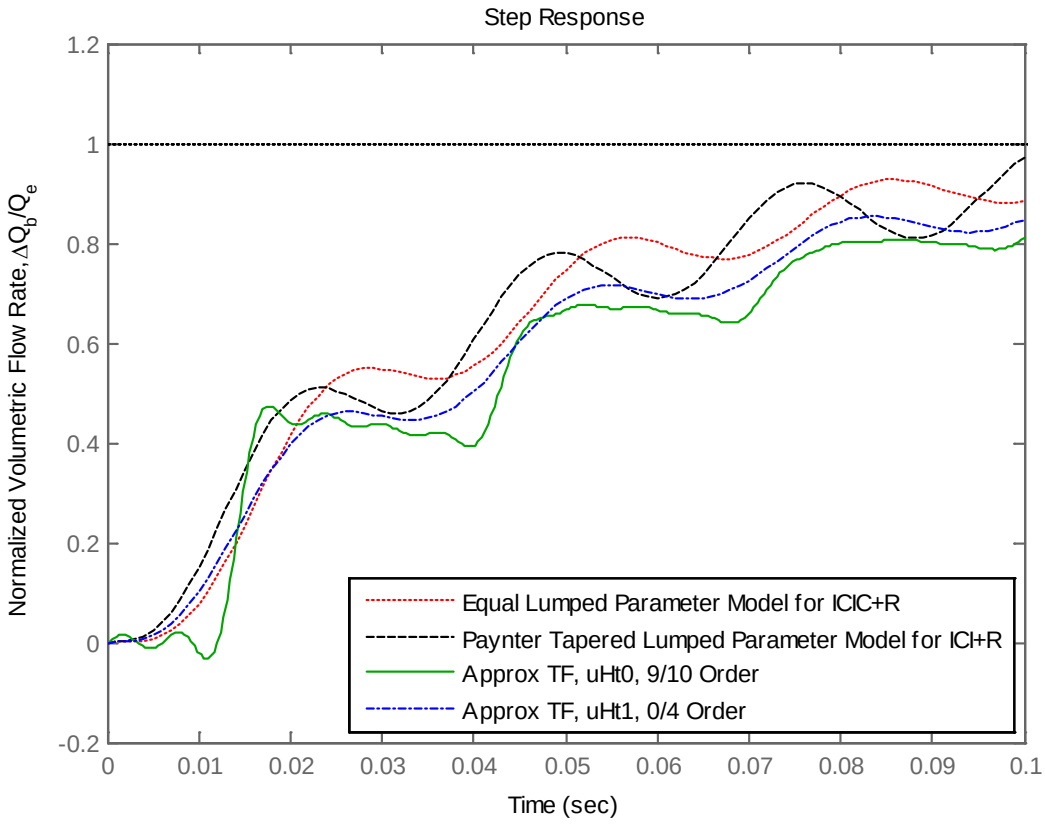


Figure 3.9 Comparison of step responses (b)

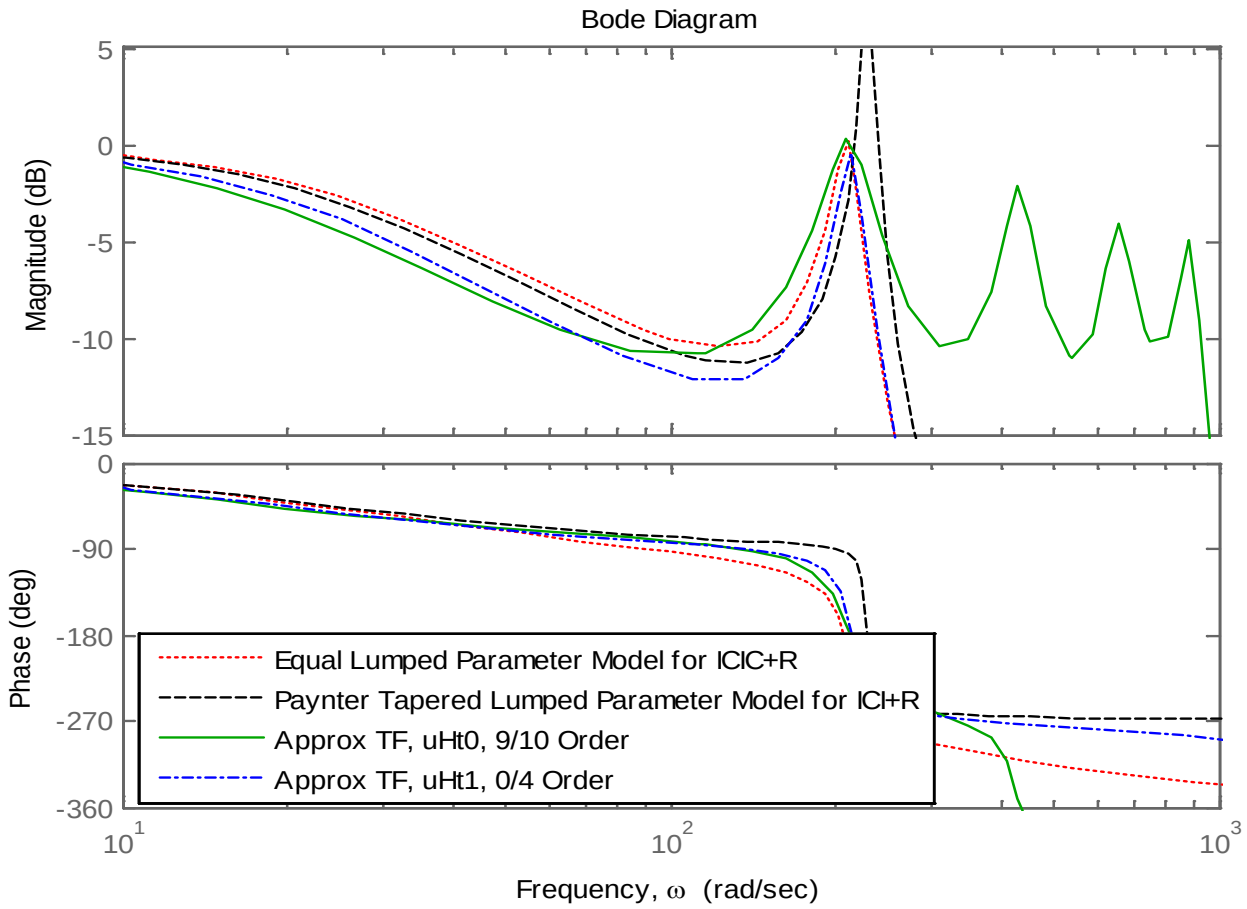


Figure 3.10 Comparison of frequency responses

Therefore a proper tapering for the lumped parameter model is needed. Since we can get an approximate transfer function with a zero order numerator from the distrabuted parameter model, we can attempt to use this transfer funciton to represent a lumped parameter model? The idea is to match the denominator coefficients of the approximate transfer function of distributed parameter model with the coefficients of the transfer function for the tapered lumped parameter model.

The 0/4th order approximate transfer function and the 4th order lumped parameter model illustrated in Figure 3.3, is used as the example to demonstrate how to solve the coefficients of inertance and capacitance of the lumped parameter.

First, define I_s, C_s, R_s as the sum of inertance, capacitance, and resistance correspondingly, and i_0, i_1, c_1, c_2 as their coefficients. Assume $I_0 = i_0 \times I_s$, $I_1 = i_1 \times I_s$, $C_1 = c_1 \times C_s$, $C_2 = c_2 \times C_s$ and $R_1 = 1 \times R_s$. The Equation (3.1) can be rewritten as

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{R_s}{\begin{aligned}&\left[I_s{}^2 C_s{}^2 R_s \left(c_1 c_2 i_0 i_1\right)\right] s^4 + \left[I_s{}^2 C_s \left(c_1 i_0 i_1\right)\right] s^3 \\ &+ \left[I_s C_s R_s \left(c_1 i_0 + c_2 i_0 + c_2 i_1\right)\right] s^2 + \left[I_s \left(i_0 + i_1\right)\right] s + R_s\end{aligned}} \frac{\Delta P_a(s)}{\left(P_i - P_o\right)} \tag{3.10}$$

where

$$I_s = \frac{\rho L}{\pi r^2} \tag{3.11}$$

$$C_s = \frac{\pi r^2 L}{\beta} \tag{3.12}$$

$$R_s = \frac{128 \rho \nu L}{\pi (2r)^4} = \frac{8 \rho \nu L}{\pi r^4} \tag{3.13}$$

$$s = \frac{\nu}{r^2} \bar{s} \tag{3.14}$$

$$D_n^2 = \left(\frac{\nu L}{cr^2}\right)^2 = \left(\frac{\nu L}{\sqrt{\beta/\rho}r^2}\right)^2 = \frac{\rho\nu^2 L^2}{\beta r^4} \tag{3.15}$$

Substitute line and fluid properties and $\bar{s}$ into the $I_s, C_s,$ and R_s of Equation (3.10) and sum up into the format of D_n.

The coefficients of the transfer function can be rewritten separately as

$$\bar{s}^4 : \frac{\rho^2\nu^4 L^4}{\beta^2 r^8}\left(i_0 i_1 c_1 c_2\right) = \left(i_0 i_1 c_1 c_2\right) \times D_n^4$$

$$\bar{s}^3 : \frac{\rho\nu^2 L^2}{\beta r^4}\frac{(i_0 i_1 c_1)}{8} = \frac{(i_0 i_1 c_1)}{8} \times D_n^2$$

$$\bar{s}^2 : \frac{\rho\nu^2 L^2}{\beta r^4}\left(i_0 c_1 + i_0 c_2 + i_1 c_2\right) = \left(i_0 c_1 + i_0 c_2 + i_1 c_2\right) \times D_n^2$$

$$\bar{s}^1 : \frac{(\mathrm{i}_0 + \mathrm{i}_1)}{8}$$

$$\bar{s}^0 : 1$$

We can see that the coefficients of the symbolic transfer function of the lumped parameter model are a function of D_n, just like the coefficients of the approximate transfer function of distributed parameter model are also a function of D_n.

Then we use symbolic math in MATLAB to solve for the taper coefficients, i_0, i_1, c_1, c_2, with the coefficients of Equation (3.4). Notice that, we have 4 equations for solving for 4 unknowns $(i_0 \quad i_1 \quad c_1 \quad c_2)$. In this example, we get the results for coefficient, i_0, i_1, c_1, c_2, for the I's and C's parameters:

$$wI = \begin{bmatrix} i_0 & i_1 \end{bmatrix} = \begin{bmatrix} 0.61822 & 0.58403 \end{bmatrix} \tag{3.16}$$

$$wC = \begin{bmatrix} c_1 & c_2 \end{bmatrix} = \begin{bmatrix} 0.38895 & 0.084098 \end{bmatrix} \tag{3.17}$$

3.4 Tapering with Varied D_n

Since the I's and C's coefficients are functions of D_n, we can repeat the whole procedure to get a table of I's and C's coefficients versus D_n for tapering the lumped parameter models. Here we use the 4th order tapered lumped parameter model as the example to calculate the I's and C's coefficients with varied D_n values, which are controlled by changing the length of the line in this case. Results are shown in Table 3.2.

Table 3.2 I's and C's coefficients of 0/4th order tapered lumped parameter model with varied D_n

L (m)	20	16	12	8	4	2
D_n	0.04139	0.033112	0.024834	0.016556	0.0082781	0.004139
Ratio	1	4/5	3/5	2/5	1/5	1/10
wI(1)	0.62301	0.64816	0.73168	0.76697	0.87463	0.97119
wI(2)	0.5877	0.55963	0.53946	0.52408	0.42157	0.32426
wC(1)	0.38633	0.38401	0.37327	0.36128	0.38123	0.43782
wC(2)	0.081268	0.073349	0.022999	2.7138e-13	2.6126e-12	2.064e-11

L (m)	1.6	1.2	0.8
D_n	0.0033112	0.0024834	0.0016556
Ratio	4/50	3/50	2/50
wI(1)	0.98956	0.9958	1.0472
wI(2)	0.2972	0.30305	0.25645
wC(1)	0.46348	0.47242	0.50005
wC(2)	1.5053e-12	1.3934e-12	0.51971

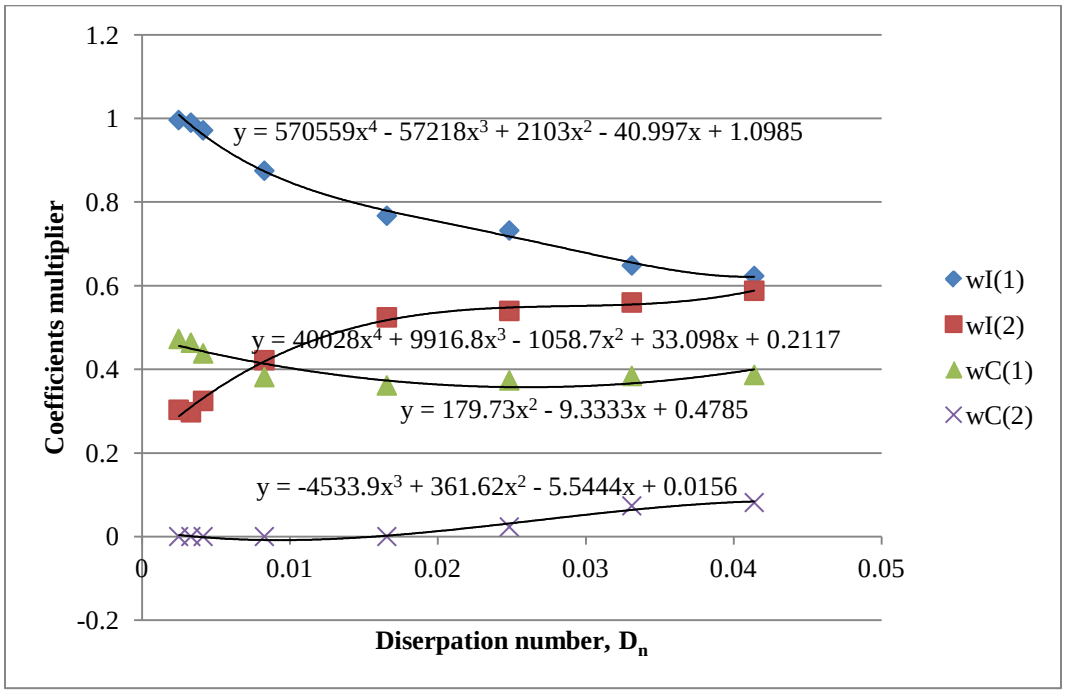


Figure 3.11 I 's and C 's coefficients of 0/4th order tapered lumped parameter model with varied D_n

To find the relation of these coefficients, they are plotted as a scatter diagram and described by polynomial equations using MS Excel function. These results are shown in Figure 3.11 as a function of D_n. However, a proper equation for the wC(2) cannot be found due to the last value of wC(2) in Table 3.2 is significantly increased to 0.51971. Also, these line equations do not express the left side of the data accurately.

To improve this and have a better data line approximation, the last data for $D_n = 0.0016556$ are taken off and $\log(D_n)$ is used. The results are shown in Figure 3.12.

For laminar flow conditions, after the value of D_n is calculated, we can get any lump coefficients of the lumped parameter model with these equations

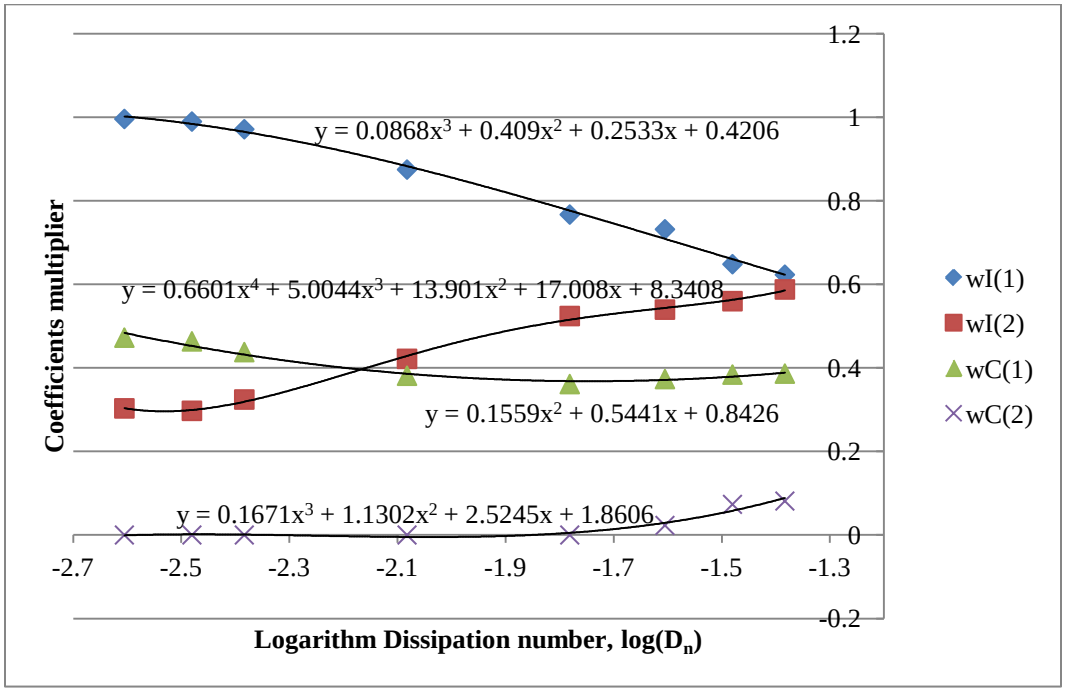


Figure 3.12 I 's and C 's coefficients of 0/4th order tapered lumped parameter model with varied $\log(D_n)$ in x-axis

This analysis technique will be extended to turbulent flow cases in Chapter 4.

Chapter 4

Tapered Lumped Parameter Model for Turbulent Flow

In Chapter 2 and Chapter 3, procedures were developed to formulate accurate approximate models results in the frequency and time domains. Also, empirical formulas were formulated for tapering the size of the lumps in lumped parameter models for laminar flow conditions.

In Chapter 4, we would like to extend this concept to turbulent flow. Since inertance and capacitance properties of the line are assumed to not be a function of the Reynolds number, we can add an extra resistance, defined as R_v, at the end of the line represented by the distributed parameter model to achieve the appropriate turbulent flow resistance. Then repeating the procedures in Chapter 2 and Chapter 3, guidelines for tapering the size of the lumps will be formulated for small pressure/flow pulsations in the line with turbulent flow.

The pressure/flow pulsations are assumed to be relatively small because the resistance of the line is a nonlinear function of the Reynolds number. Therefore, the case of a small amplitude disturbance superimposed on a gross turbulent flow is applied here. The next section of this document pertains to the derivation of the equation for the equivalent resistance R_v required to achieve a steady flow resistance for turbulent flow. This resistance will be added to the end of the line which will be modeled using the distributed parameter equations.

4.1 Turbulent Flow Resistance

For incompressible steady flow of the circular transmission line, the summation of the forces on the fluid in a small section of pipe is

$$\int_{P_i}^{P_o} dP = -\int_{L_1}^{L_2} \frac{f \rho U^2}{2D} dx \tag{4.1}$$

The empirical friction factor f [7, 15, 16] of a smooth pipe for turbulent flow is

$$f \approx 0.3164 / R_n^{0.25} \tag{4.2}$$

where R_n is the Reynolds number defined by

$$R_n = \frac{4\rho Q_e}{\mu \pi D} \tag{4.3}$$

The velocity of the flow is defined by

$$U = \frac{Q_e}{A} = \frac{4Q_e}{\pi D^2} \tag{4.4}$$

Substituting these equations into Equation (4.1) and performing the integration gives

$$P_i - P_o = \frac{0.2414 \rho^{0.75} \mu^{0.25} L}{D^{4.75}} Q_e^{1.75} = C_f Q_e^{1.75} \tag{4.5}$$

The lumped resistance R_v is defined by the following equation

$$P_i - P_o = C_f Q_e^{1.75} = (R_s + R_v) Q_e \tag{4.6}$$

Thus, solving for R_v gives

$$R_v = C_f Q_e^{0.75} - R_s \tag{4.7}$$

Or in terms of the Reynolds number, Equation (4.7) becomes

$$\begin{aligned} R_v &= \left(\frac{0.2414 \rho^{0.75} \mu^{0.25} L}{D^{4.75}} \right) \left(\frac{\pi D \mu R_n}{4\rho} \right)^{0.75} - \frac{128 \mu L}{\pi D^4} \\ &= \left(\frac{0.2414 \mu L}{D^4} \right) \left(\frac{\pi}{4} \right)^{0.75} R_n^{0.75} - \frac{128 \mu L}{\pi D^4} \end{aligned} \tag{4.8}$$

For $R_v = 0$, $R_n = 1187.6$. So, the Reynolds number at which turbulent resistance equals the laminar resistance is 1187.6. The above equation for R_v gives a value that can be used as long as $R_n > 1187.6$.

It is desirable to represent R_v as a normalized resistance R_r defined by

$$R_r = \frac{R_v}{R_s} = \left(\frac{0.2414\pi}{128}\right)\left(\frac{\pi}{4}\right)^{0.75} R_n^{0.75} - 1 = 0.004943 R_n^{0.75} - 1 \tag{4.9}$$

or

$$R_n^{0.75} = 202.3044(R_r + 1) \tag{4.10}$$

4.2 Distributed Parameter Model

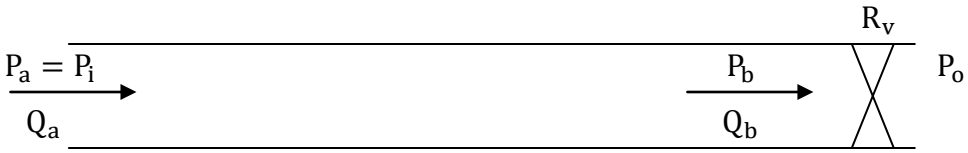


Figure 4.1 Schematic of a fluid transmission line for turbulent flow

To apply the turbulent flow resistance to the distributed parameter model, an additional linear resistance R_v will be attached to the end of the line so the total resistance will be equal to the turbulent flow resistance R_t.

See Figure 4.1 for the schematic. Consider the example shown below. The flow through the line is initially steady with upstream pressure $P_a = P_i$ and downstream pressure P_o. Pressure/flow pulsations will be created from a slight increase or decrease

in pressure P_a, ΔP_a. Therefore, like Equation (2.18) for laminar flow conditions, the transfer function for normalized flow rate for the normalized pressure input becomes,

$$\frac{\Delta Q_b(\bar{s})}{Q_e} = \left[\frac{R_t}{Z \sinh \Gamma + \cosh \Gamma R_v} \right] \frac{\Delta P_a(\bar{s})}{(P_i - P_o)} \tag{4.11}$$

where $R_v = R_t - R_s$

Using the example presented in Chapter 2 and Chapter 3, the effects of a gradual transition from laminar to turbulent flow are presented in Table 4.1

Table 4.1 Effects of increasing from laminar to turbulent flow

	1st mode frequency	Damping ζ_1	Natural frequency ω_1
$R_r = 0$, Laminar flow	5.77	0.0962	70.1
$R_r = 1$, $R_n = 2992.6$	12	0.205	69.8
$R_r = 2$, $R_n = 5138.5$	18.4	0.342	70.4
$R_r = 3$, $R_n = 7540.9$	22.2	0.581	68.1

It is observed that the damping ratio and 1st mode frequency increase as the Reynolds number increases.

4.3 Curve Fit Approximation of the Distributed Parameter Model

Before doing the curve fit approximation, we need to assume the Reynolds number for the turbulent flow in order to have the turbulent flow equivalent resistance.

To demonstrate, we start with $R_n = 10544.37$, which is turbulent flow condition. The flow rate can be derived.

$$Q_0 = R_n \frac{\nu A}{D} \text{ or } R_n \frac{\mu \pi D}{4\rho} \tag{4.12}$$

Properties of line and fluid are the same as in Chapter 2. Therefore, Equation (4.9), the turbulent flow equivalent resistance becomes

$$R_t = 2.6872e+011$$

Then we repeat the same procedures as with laminar flow condition for turbulent flow condition to do the curve fitting. As in the previous chapters, the 9th/10th order transfer function approximation will be used to represent the true model since it provides an accurate least squares fit for the dominant modes out through the bandwidth of the pulsation frequencies. Setting the upper bound on frequency to be 353 ($rad/\sec$) results in the so called 'true' transfer function for comparison with other models for turbulent flow conditions:

$$\frac{\Delta Q_b(\bar{s})}{Q_e} = \frac{1.935\times10^{-21}\bar{s}^9 + \cdots - 1.761\times10^{-5}\bar{s}^2 - 0.003475\bar{s} + 1}{1.222\times10^{-22}\bar{s}^{10} + \cdots + 0.008033\bar{s}^2 + 0.03415\bar{s} + 1}\frac{\Delta P_a(\bar{s})}{(P_i - P_o)} \qquad (4.13)$$

The eigenvalues, damping ratios, and natural frequencies are listed below for comparison with lumped parameter models with zero order numerators to be formulated.

Eigenvalue	Damping	Freq. (rad/s)
-2.98e+001 + 3.80e+001i	6.17e-001	4.83e+001
-2.98e+001 - 3.80e+001i	6.17e-001	4.83e+001
-2.81e+001 + 1.10e+002i	2.48e-001	1.13e+002
-2.81e+001 - 1.10e+002i	2.48e-001	1.13e+002
-3.73e+001 + 1.75e+002i	2.08e-001	1.79e+002
-3.73e+001 - 1.75e+002i	2.08e-001	1.79e+002
-4.69e+001 + 2.60e+002i	1.77e-001	2.65e+002
-4.69e+001 - 2.60e+002i	1.77e-001	2.65e+002
-3.34e+001 + 3.47e+002i	9.59e-002	3.49e+002

-3.34e+001 - 3.47e+002i 9.59e-002 3.49e+002

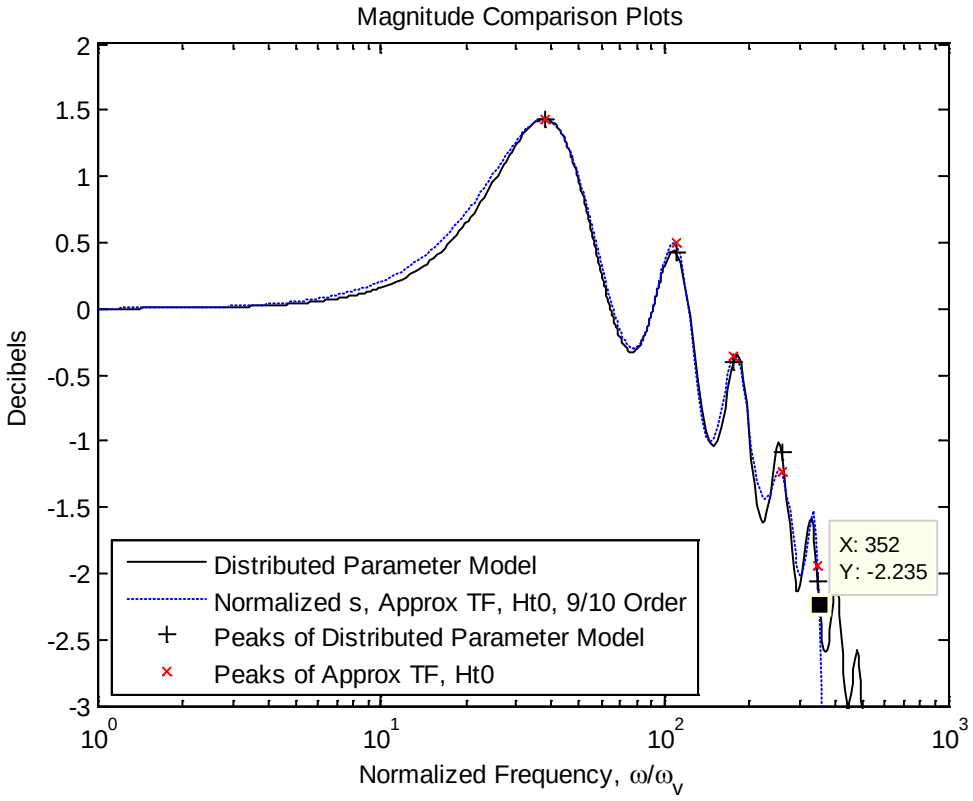


Figure 4.2 Comparison of frequency responses of distributed parameter model and $9^{th}/10^{th}$ order approximation model

Figure 4.2 shows the accuracy of the 'true' model in regard to matching resonant peaks and frequencies. A quantitative comparison of the resonant peaks is provided in Table 4.2.

Table 4.2 Comparing the resonant peaks of the distributed parameter and $9^{th}/10^{th}$ models

Normalized resonant frequencies	38.004	109.86	175.01	260.38	347.28
Resonant peaks of distributed model (dB)	1.4367	0.42623	-0.4	-1.0828	-2.0572
Resonant peaks of transfer function approximation (dB)	1.4262	0.49274	-0.36327	-1.2373	-1.9373

Selection of the upper bound on the frequency for the least squares curve fit may make a big difference in the accuracy of the results. For example, the results of an attempt to include too many modes in the fit with too low of an order for the transfer function are provided in Figure 4.3 and Table 4.3. The upper frequency bound was set to 120 $rad/\sec$ in an attempt to include the second 2nd order mode. The accuracy of the resulting transfer function is poor.

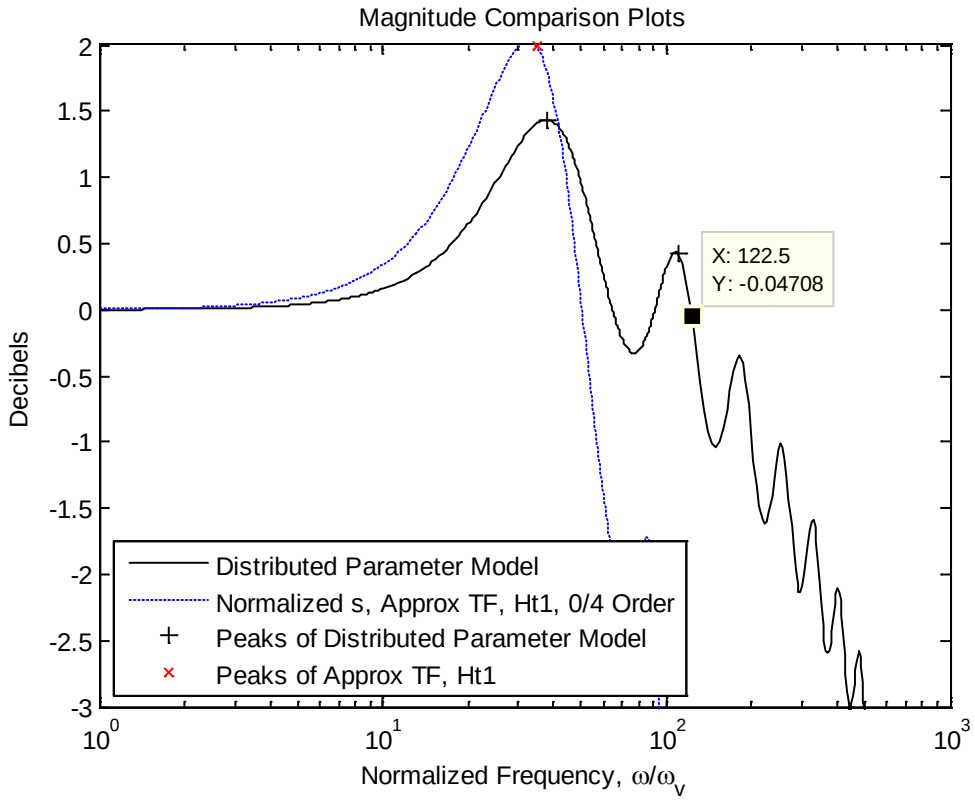


Figure 4.3 Comparison of frequency responses of distributed parameter model and 0/4th order approximation model (a)

A quantitative comparison of the resonant peaks of the first and second 2nd order modes are provided in Table 4.3. The error of the first 2nd order mode is seen to be 1.9945-1.43667=0.55783 dB which is not really significant if we consider the magnitude

scale; an error less than 1dB, is small. But the error of the second 2nd order mode is 0.42623-(-2.4211)=2.8473 dB which is not acceptable.

Table 4.3 Magnitude of distributed parameter model and 0/4th order TF approximation

Normalized resonant frequencies	34.831	91.992
Resonant peaks of distributed model (dB)	1.4367	0.42623
Resonant peaks of transfer function approximation (dB)	1.9945	-2.4211

In order to improve the results, the maximum frequency range is reduced to only cover the first mode frequency and adjusted until the first mode peak of the 4th order transfer function approximation matches the peak of the distributed parameter model. In this case, the upper bound on frequency is set to be 73 ($rad/\sec$). This kind of adjustment will be applied to all similar cases. Examination of Figure 4.4 shows the improvement. The resulting transfer function for turbulent flow is

$$\frac{\Delta Q_b(\bar{s})}{Q_e} = \frac{1}{9.536\times10^{-8}\bar{s}^4 + 7.042\times10^{-6}\bar{s}^3 + 0.0008799\bar{s}^2 + 0.03299\bar{s} + 1} \frac{\Delta P_a(\bar{s})}{(P_i - P_o)} \tag{4.14}$$

Eigenvalue	Damping	Freq. (rad/s)
-2.39e+001 + 3.29e+001i	5.88e-001	4.07e+001
-2.39e+001 - 3.29e+001i	5.88e-001	4.07e+001
-1.30e+001 + 7.85e+001i	1.63e-001	7.95e+001
-1.30e+001 - 7.85e+001i	1.63e-001	7.95e+001

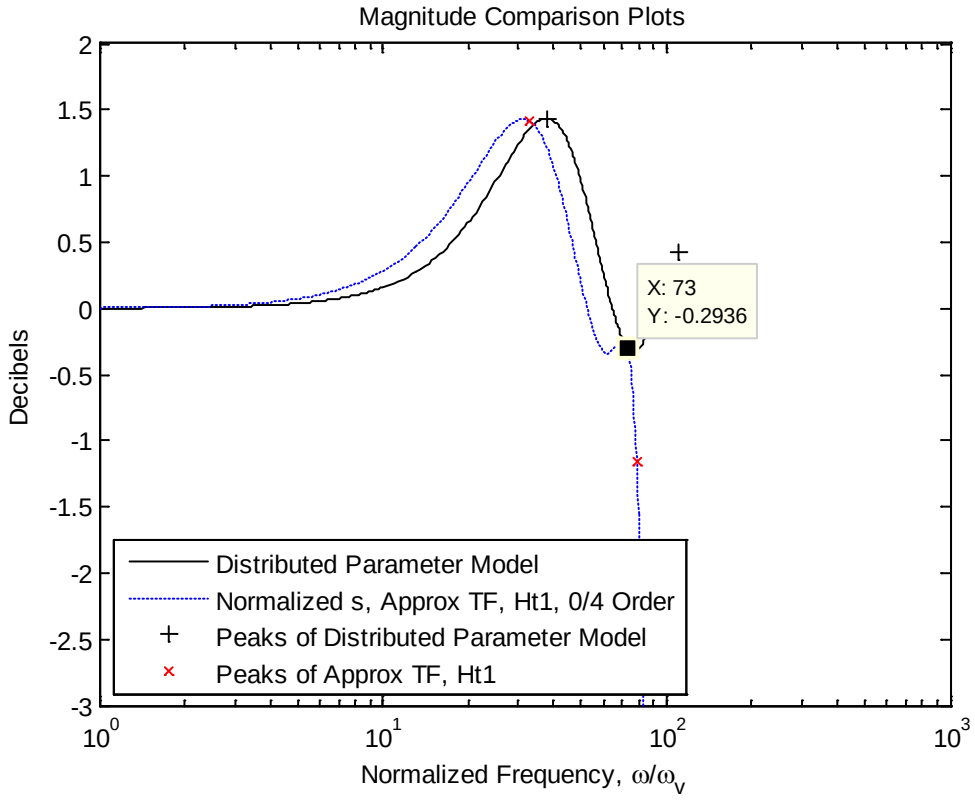


Figure 4.4 Comparison of frequency responses of distributed parameter model and 0/4th order approximation model (b)

A quantitative comparison of the resonant peaks of the first 2nd order mode is provided in Table 4.4. The error of the first 2nd order mode is seen to be 1.4367-1.4159=0.0208 dB which is relatively very accurate.

Table 4.4 Magnitude of distributed parameter model and 0/4th order TF approximation

Normalized resonant frequencies	32.935
Resonant peaks of distributed model (dB)	1.4367
Resonant peaks of transfer function approximation (dB)	1.4159

4.4 Equal Lumped Parameter Model

This section shows the difference between 4th order equal lumped parameter model and models of 10th and 4th order approximate transfer functions of the distributed parameter model with turbulent flow conditions. Recall that the 10th order approximate transfer function represents what is considered to be the 'true' transfer function for comparing and evaluating other transfer function models. The 4th order approximate transfer function represents a reasonably accurate model out through the first 2nd order mode; it will be used to find the coefficients of inertance and capacitance of a tapered lumped parameter model.

4th order equal lumped parameter model:

For 4th order equal lumped parameter model, we assume that $C_1 = C_2 = 0.5 \times C_s$, $I_0 = I_1 = 0.5 \times I_s$ and $R_1 = R_t$. The transfer function, eigenvalues and mode frequencies become

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{1}{2.197 \times 10^{-9} s^4 + 1.884 \times 10^{-7} s^3 + 0.0001406 s^2 + 0.00804 s + 1} \frac{\Delta P_a(s)}{\left(P_i - P_o\right)} \tag{4.15}$$

Eigenvalue	Damping	Freq. (rad/s)
-3.21e+001 + 8.58e+001i	3.50e-001	9.16e+001
-3.21e+001 - 8.58e+001i	3.50e-001	9.16e+001
-1.08e+001 + 2.33e+002i	4.64e-002	2.33e+002
-1.08e+001 - 2.33e+002i	4.64e-002	2.33e+002

In order to compare the time responses of the equal lumped parameter model with the transfer function approximations of the distributed parameter model, $\overline{s}$ must first be un-normalized using $\overline{s} = s \times (r^2 / \nu)$. The transfer functions can then be compared to the 'true' transfer function. The result is

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{9.183\times10^{-26}s^9 + \cdots - 1.928\times10^{-6}s^2 - 0.00115s + 1}{1.918\times10^{-27}s^{10} + \cdots + 8.791\times10^{-5}s^2 + 0.0113s + 1}\frac{\Delta P_a(s)}{\left(P_i - P_o\right)} \tag{4.16}$$

Eigenvalue	Damping	Freq. (rad/s)
-9.01e+001 + 1.15e+002i	6.17e-001	1.46e+002
-9.01e+001 - 1.15e+002i	6.17e-001	1.46e+002
-8.50e+001 + 3.32e+002i	2.48e-001	3.43e+002
-8.50e+001 - 3.32e+002i	2.48e-001	3.43e+002
-1.13e+002 + 5.29e+002i	2.08e-001	5.41e+002
-1.13e+002 - 5.29e+002i	2.08e-001	5.41e+002
-1.42e+002 + 7.87e+002i	1.77e-001	8.00e+002
-1.42e+002 - 7.87e+002i	1.77e-001	8.00e+002
-1.01e+002 + 1.05e+003i	9.59e-002	1.05e+003
-1.01e+002 - 1.05e+003i	9.59e-002	1.05e+003

The 4th order transfer function approximation for turbulent flow conditions is found to be

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{1}{1.142\times10^{-9}s^4 + 2.55\times10^{-7}s^3 + 9.63\times10^{-5}s^2 + 0.01091s + 1}\frac{\Delta P_a(s)}{\left(P_i - P_o\right)} \tag{4.17}$$

Eigenvalue	Damping	Freq. (rad/s)
-7.24e+001 + 9.96e+001i	5.88e-001	1.23e+002
-7.24e+001 - 9.96e+001i	5.88e-001	1.23e+002
-3.92e+001 + 2.37e+002i	1.63e-001	2.40e+002
-3.92e+001 - 2.37e+002i	1.63e-001	2.40e+002

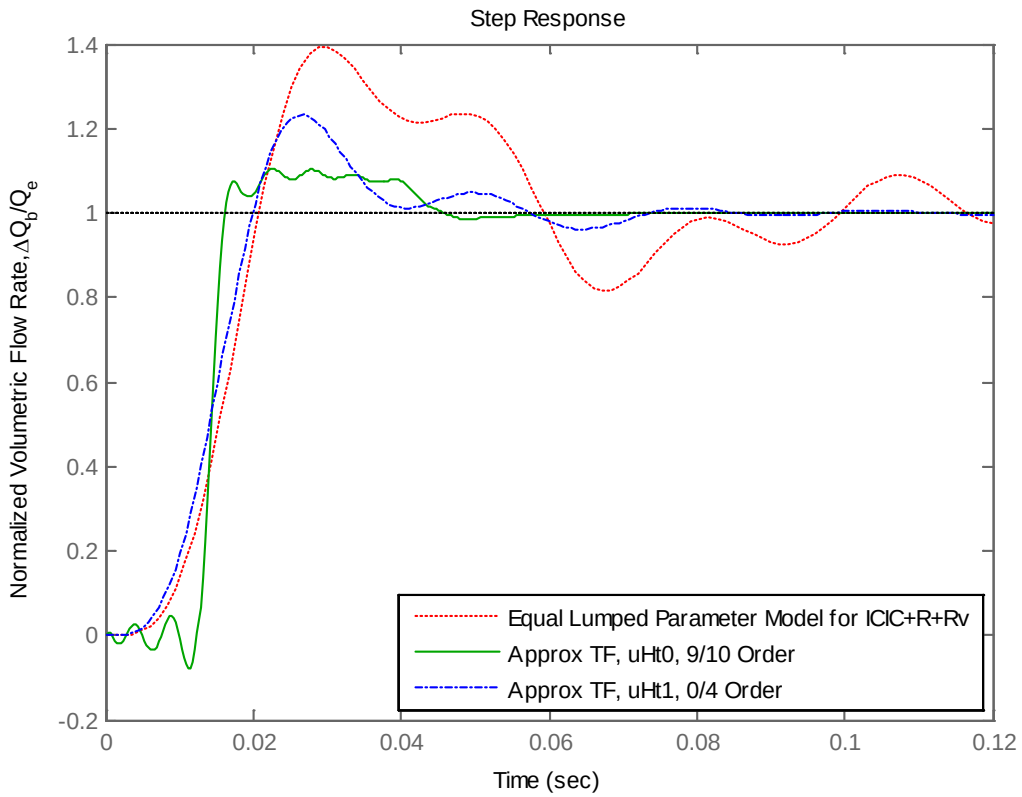


Figure 4.5 Comparison of step responses of equal lumped parameter model, $9^{th}/10^{th}$ order and $0/4^{th}$ order approximate transfer function models

From the above data and Figure 4.5 and Figure 4.6, there are significant differences between the model with equal size lumps and both models of $9^{th}/10^{th}$ order and $0/4^{th}$ order transfer functions. It is clear that the model of $0/4^{th}$ order approximate transfer function is closer to the 'true' model than the model with equal size lumps. Therefore, this $0/4^{th}$ order approximate transfer function will be used in the next section to solve for the tapering coefficients of the inertance and capacitance lumps.

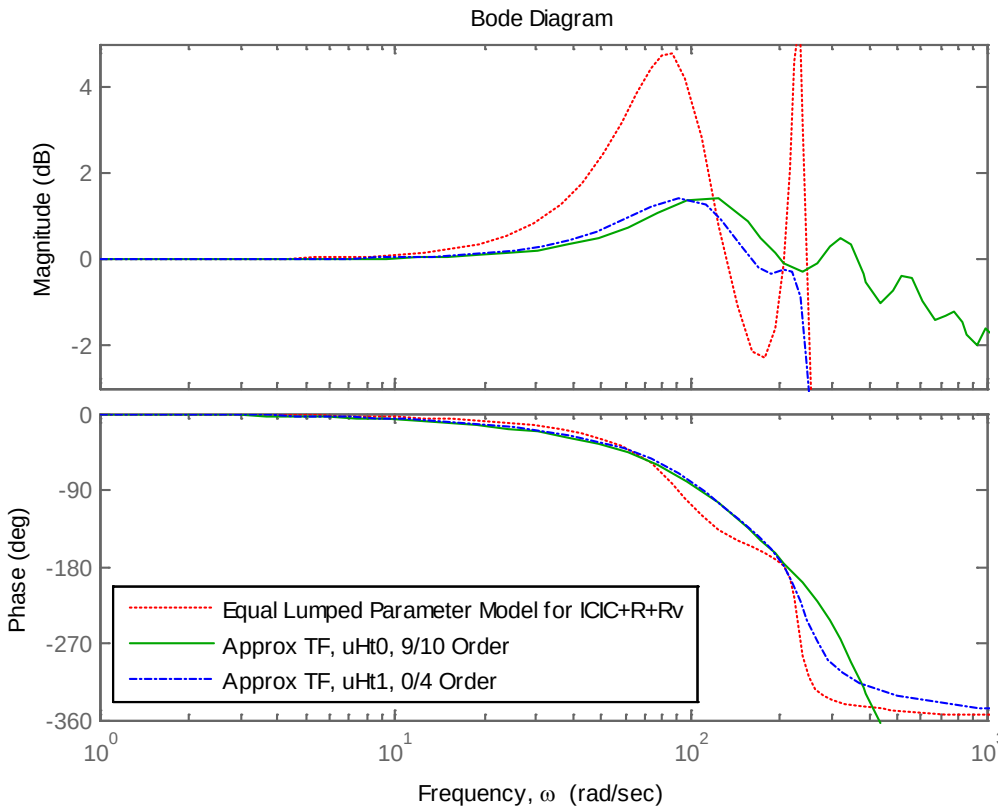


Figure 4.6 Comparison of bode plots of equal lumped parameter model, 9th/10th order and 0/4th order approximate transfer function models

4.5 Lumped Parameter Model Tapering with Varied D_n & R_n

Referring to Figure 3.3, we assume $I_0 = i_0 I_s$, $I_1 = i_1 I_s$, $C_1 = c_1 C_s$, $C_2 = c_2 C_s$ and $R_1 = R_t$. The same procedure is repeated as was done for laminar flow conditions in Chapter 3. The symbolic transfer function of the 4th order lumped parameter model is

$$\frac{\Delta Q_b(s)}{Q_e} = \frac{R_t}{\left[I_s{}^2 C_s{}^2 R_s \left(c_1 c_2 i_0 i_1\right)\right] s^4 + \left[I_s{}^2 C_s \left(c_1 i_0 i_1\right)\right] s^3 + \left[I_s C_s R_s \left(c_1 i_0 + c_2 i_0 + c_2 i_1\right)\right] s^2 + \left[I_s \left(i_0 + i_1\right)\right] s + R_t} \frac{\Delta P_a(s)}{\left(P_i - P_o\right)} \qquad (4.18)$$

After solving the symbolic math with MATLAB by comparing Equation (4.17) and Equation (4.18), we get the coefficients for the I's and C's for $D_n = 0.04139$ with $R_n = 10544.37$:

$$wI = \begin{bmatrix} i_0 & i_1 \end{bmatrix} = \begin{bmatrix} 0.68837 & 0.66893 \end{bmatrix} \quad (4.19)$$

$$wC = \begin{bmatrix} c_1 & c_2 \end{bmatrix} = \begin{bmatrix} 0.36733 & 0.1921 \end{bmatrix} \quad (4.20)$$

The coefficients are different than those for laminar flow. Therefore, new tables are needed. As was done for laminar flow, we varied only the length of the line to get the corresponding values of D_n for computing the coefficients.

The resulting coefficients for $R_n = 10544.37$ and different values of D_n are shown in Table 4.5 and Figure 4.7.

Table 4.5 I 's and C 's coefficients of 0/4th order tapered lumped parameter model with varied D_n, for $R_n = 10,544$

L (m)	20	16	12	8	4	2
D_n	0.04139	0.033114	0.024834	0.016557	0.0082778	0.004139
Ratio	1	4/5	3/5	2/5	1/5	1/10
wI(1)	0.68837	x	0.61696	0.6307	0.62692	0.65081
wI(2)	0.66893	x	0.60042	0.57044	0.5105	0.45644
wC(1)	0.36733	x	0.43945	0.46648	0.40276	0.39793
wC(2)	0.1921	x	0.17319	0.11895	0.11729	0.11267

L (m)	1.6	1.2	0.8	0.4	0.2
D_n	0.0033113	0.0024834	0.0016556	0.0008278	0.0004139
Ratio	4/50	3/50	2/50	1/50	1/100
wI(1)	0.66408	0.72043	0.72674	0.79124	0.85847
wI(2)	0.443	0.4549	0.42	0.37899	0.32999
wC(1)	0.39989	0.38127	0.39407	0.40473	0.43202
wC(2)	1.2976e-013	1.7723e-013	9.7274e-013	3.0053e-012	2.9128e-011

From the Table 4.5, we see that the column for $D_n = 0.033114$ is incomplete; for these cases the solution did not converge. That is because the difficulty of achieving a good curve fit when there is a real eigenvalue located in front of the first 2nd order mode in the frequency response of the distributed parameter model.

To find empirical equations for these coefficients, the data in Table 4.5 was plotted as a scatter diagram and described with polynomial equations using the MS Excel function. This is shown in Figure 4.7 as a function of $\log(D_n)$.

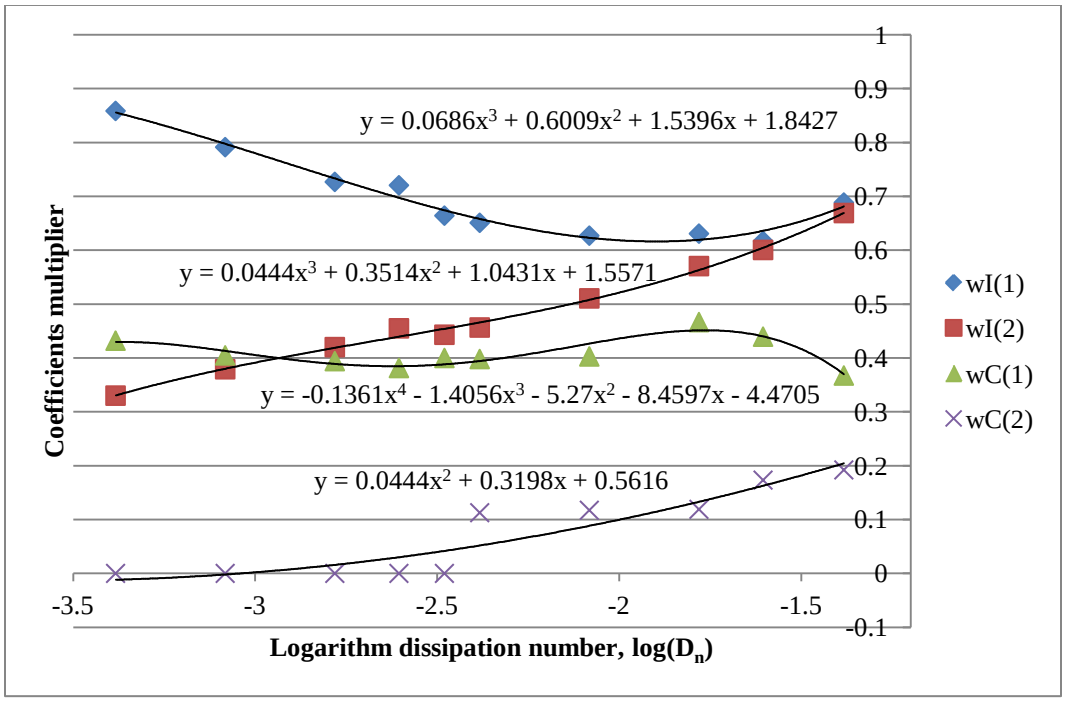


Figure 4.7 I's and C's coefficients of 0/4th order tapered lumped parameter model with $\log(D_n)$ in x-axis, for $R_n = 10,544$

More results are shown below in tables and figures as a function of $\log(D_n)$ for turbulent flow $R_r = 10 \text{ and } 100$ which correspond to Reynolds numbers of $R_n = 29,053 \text{ and } 558,602$.

Table 4.6 I 's and C 's coefficients of 0/4th order tapered lumped parameter model with varied D_n, for $R_n = 29,053$

L (m)	20	16	12	8	4	2	1.6
D_n	0.04139	0.033112	0.024834	0.016556	0.0082781	0.004139	0.0033112
Ratio	1	4/5	3/5	2/5	1/5	1/10	4/50
wI(1)	0.99341	0.84938	0.70615	0.59849	0.6091	0.62187	0.62023
wI(2)	0.65043	0.53336	0.45543	0.41951	0.51182	0.50236	0.49535
wC(1)	0.21022	0.23068	0.25758	0.26814	0.37973	0.38015	0.38787
wC(2)	0.18113	0.2163	0.25727	0.2829	0.17453	0.10111	0.11306

L (m)	1.2	0.8	0.4	0.2
D_n	0.0024834	0.0016556	0.00082781	0.0004139
Ratio	3/50	2/50	1/50	1/100
wI(1)	0.62086	0.64886	0.68949	0.72005
wI(2)	0.46815	0.4665	0.43296	0.39775
wC(1)	0.39166	0.38126	0.38829	0.40148
wC(2)	0.142	0.11337	0.099763	0.13629

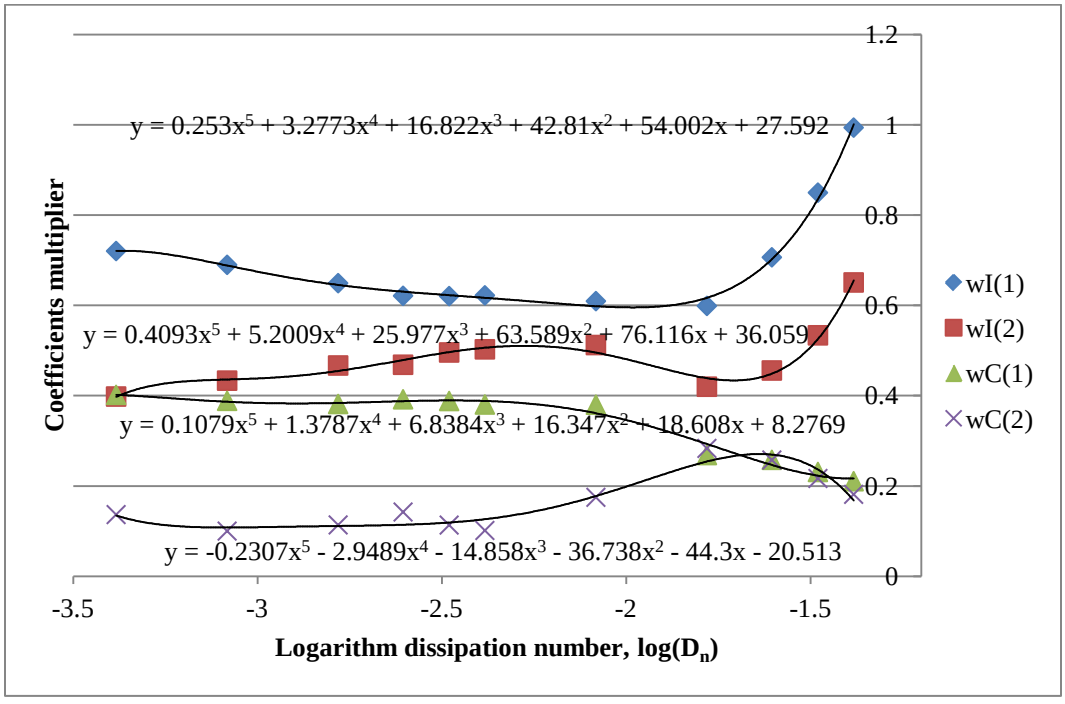


Figure 4.8 I 's and C 's coefficients of 0/4th order tapered lumped parameter model with $\log(D_n)$ x-axis, for $R_n = 29,053$

Table 4.7 I 's and C 's coefficients of 0/4th order tapered lumped parameter model with varied D_n, for $R_n = 558,602$

L (m)	20	16	12	8	4	2	1.6
D_n	0.04139	0.033112	0.024834	0.016556	0.0082781	0.004139	0.0033112
Ratio	1	4/5	3/5	2/5	1/5	1/10	4/50
wI(1)	0.50442	0.36951	0.25469	0.16235	0.091907	0.064734	0.061261
wI(2)	0.27674	0.20638	0.14562	0.094432	0.055663	0.044341	0.041117
wC(1)	0.41782	0.55363	0.78573	1.218	2.0062	2.5362	2.7576
wC(2)	0.41903	0.57162	0.80705	1.2178	2.0556	2.7786	2.8323

L (m)	1.2	0.8	0.4	0.2
D_n	0.0024834	0.0016556	0.00082781	0.0004139
Ratio	3/50	2/50	1/50	1/100
wI(1)	0.059152	0.046495	0.059259	0.060449
wI(2)	0.039308	0.039637	0.051228	0.051736
wC(1)	2.8381	3.0306	3.6128	3.6506
wC(2)	2.8938	2.6668	1.9599	1.4214

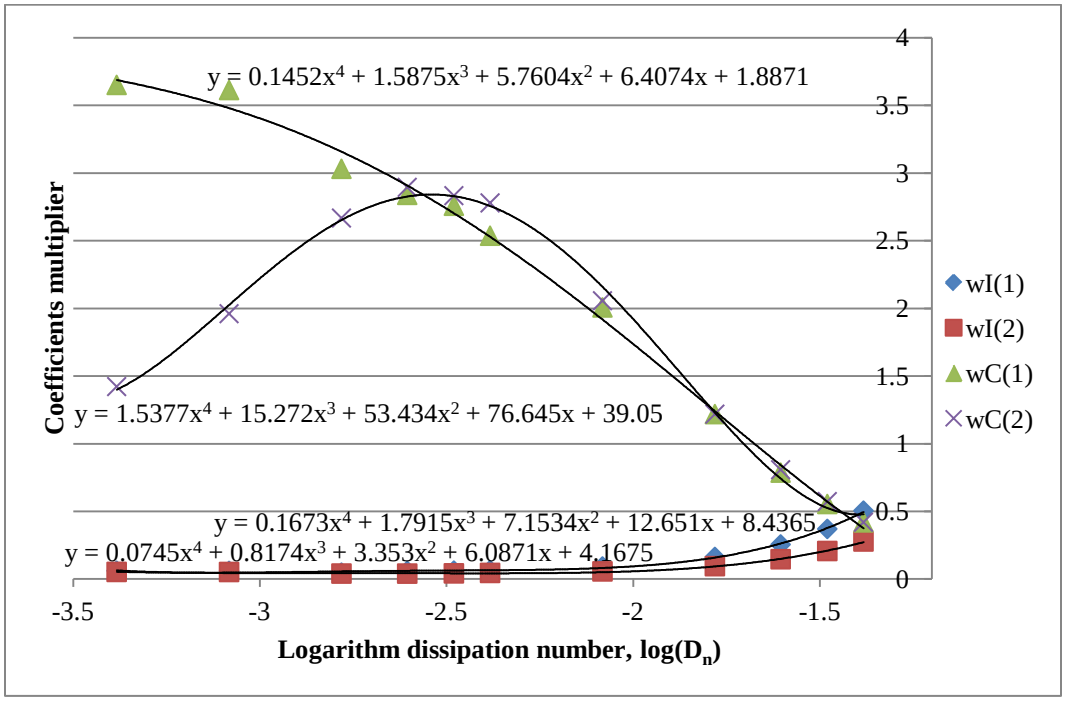


Figure 4.9 I 's and C 's coefficients of 0/4th order tapered lumped parameter model with $\log(D_n)$ x-axis, for $R_n = 558,602$

Chapter 5

Models for an Assortment of Boundary Conditions

The examples in the previous chapters were based on the specific set of boundary conditions associated with the output being the perturbation in flow and the input being a perturbation in pressure. That is,

$$\Delta Q_b(\bar{s}) = \left[\frac{1}{Z \sinh \Gamma}\right] \Delta P_a(\bar{s})$$

The objective of this chapter is to demonstrate how to use the results in the previous chapters to other sets of boundary conditions. Three applications of the line are used to show the similarities and differences of each model in Chapter 5. As in the previous chapters, an additional resistance R_v is added to the end of the line to account for turbulence.

The similarities and differences of each application in the distributed parameter model and the tapered lumped parameter model will be demonstrated. Also, the I's and C's coefficients for each application will be presented with tables and polynomial equations from corresponding figures of the I's and C's coefficients of tapered lumped parameters with varied resistance ratio, R_r, and dissipation number, D_n.

5.1 Distributed Parameter Model Transfer Functions

Case 1:

Consider the schematic of a fluid line shown in Figure 5.1. The pressures at the ends of the line are denoted by P_a and P_o; the volumetric flow rates at the ends are denoted by Q_a and Q_b; R_v is a linear resistance attached at the right end of the line for

turbulence. Assume that the flow through the line is initially constant, then a sudden increase or decrease occurs in the pressure P_a, ΔP_a, associated with a control or measurement signal. This input will cause pressure and flow pulsations along the line.

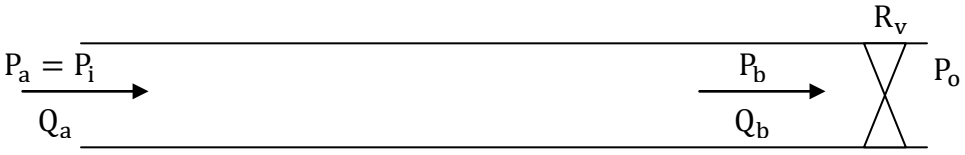


Figure 5.1 Schematic of a fluid transmission line for Case 1

The flow rate Q_b equation of distributed parameter model for case 1 is given from Equation (2.2). Then the equations of this model are

$$\begin{aligned} \Delta P_b - \Delta P_o &= R_v \cdot \Delta Q_b \\ \Delta Q_b &= \frac{1}{Z \sinh \Gamma} \Delta P_a - \frac{\cosh \Gamma}{Z \sinh \Gamma} \Delta P_b \end{aligned} \tag{5.1}$$

Since $\Delta P_o = 0$, the ΔQ_b can be rewritten as

$$\begin{aligned} \Delta Q_b &= \frac{1}{Z \sinh \Gamma} \Delta P_a - \frac{\cosh \Gamma}{Z \sinh \Gamma} (\Delta P_o + R_v \Delta Q_b) \\ &= \frac{1}{Z \sinh \Gamma + R_v \cosh \Gamma} \Delta P_a \end{aligned} \tag{5.2}$$

For steady state, the flow rate through the line will be the same value, Q_e, and the pressure difference becomes $(P_i - P_o)$. The steady state equation can be written as

$$P_i - P_o = \left(R_s + R_v \right) Q_e \tag{5.3}$$

Finally, the transfer function of the normalized flow rate ΔQ_b becomes

$$\frac{\Delta Q_b}{Q_e} = \frac{R_s + R_v}{Z \sinh \Gamma + R_v \cosh \Gamma} \frac{\Delta P_a}{\left(P_i - P_o\right)}$$

or (5.4)

$$= \frac{1 + R_r}{\frac{Z}{R_s} \sinh \Gamma + R_r \cosh \Gamma} \frac{\Delta P_a}{\left(P_i - P_o\right)}$$

where R_r is defined as $R_r = R_v / R_s$

Using the same procedure, the ΔQ_a and the transfer function of the normalized ΔQ_a is

$$\Delta Q_a = \frac{Z \cosh \Gamma + R_v \sinh \Gamma}{Z(Z \sinh \Gamma + R_v \cosh \Gamma)} \Delta P_a \tag{5.5}$$

$$\frac{\Delta Q_a}{Q_e} = \frac{(Z \cosh \Gamma + R_v \sinh \Gamma)(R_v + R_s)}{Z(Z \sinh \Gamma + R_v \cosh \Gamma)} \frac{\Delta P_a}{\left(P_i - P_o\right)}$$

or (5.6)

$$= \frac{(\frac{Z}{R_s} \cosh \Gamma + R_r \sinh \Gamma)(R_r + 1)}{Z(\frac{Z}{R_s} \sinh \Gamma + R_r \cosh \Gamma)} \frac{\Delta P_a}{\left(P_i - P_o\right)}$$

The transfer function of ΔP_b can be shown to be

$$\Delta P_b = \frac{R_v}{Z \sinh \Gamma + R_v \cosh \Gamma} \Delta P_a \tag{5.7}$$

Case 2:

Consider the schematic of a fluid line shown in Figure 5.2. The pressures at the ends of the line are denoted by P_i and P_b; the volumetric flow rates at the ends are denoted by Q_a and Q_b; R_v is a linear resistance attached at the right end of line to account for turbulence. Assume that the flow through the line is initially constant.

Pulsations will also occur if there is a sudden decrease in the pressure P_b, ΔP_b. This decrease could be associated with a break in the line or the sudden opening of a valve causing a sudden increase in the flow Q_b, ΔQ_b.

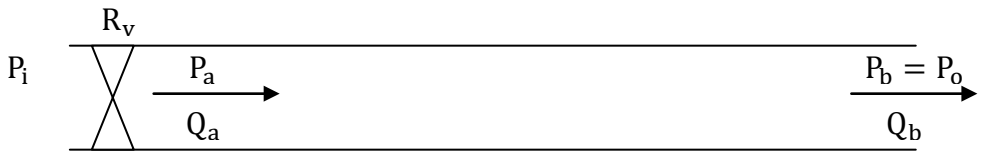


Figure 5.2 Schematic of a fluid transmission line for Case 2

The flow rate Q_a equation of the distributed parameter model for case 2 is given from the Equation (2.2). The resulting equations of this model are

$$\begin{aligned} \Delta P_i - \Delta P_a &= R_v \cdot \Delta Q_a \\ \Delta Q_a &= \frac{\cosh\Gamma}{Z\sinh\Gamma}\Delta P_a - \frac{1}{Z\sinh\Gamma}\Delta P_b \end{aligned} \tag{5.8}$$

Since $\Delta P_i = 0$, the ΔQ_a can be rewritten as

$$\begin{aligned} \Delta Q_a &= \frac{\cosh\Gamma}{Z\sinh\Gamma}(\Delta P_i - R_v \Delta Q_a) - \frac{1}{Z\sinh\Gamma}\Delta P_b \\ &= \frac{-1}{Z\sinh\Gamma + R_v\cosh\Gamma}\Delta P_b \end{aligned} \tag{5.9}$$

In steady state, the flow rate difference through the line will be the same value, Q_e, and the pressure difference becomes $(P_i - P_o)$. The steady state equation can be written as $P_i - P_o = (R_s + R_v)Q_e$.

Finally, the transfer function of the normalized flow rate ΔQ_a becomes

$$\frac{\Delta Q_a}{Q_e} = \frac{-\left(R_s + R_v\right)}{Z\sinh\Gamma + R_v\cosh\Gamma}\frac{\Delta P_b}{\left(P_i - P_o\right)}$$

or (5.10)

$$= \frac{-\left(1 + R_r\right)}{\frac{Z}{R_s}\sinh\Gamma + R_r\cosh\Gamma}\frac{\Delta P_b}{\left(P_i - P_o\right)}$$

Using the same procedure, the transfer function of the normalized ΔQ_b is

$$\Delta Q_b = -\frac{Z\cosh\Gamma + R_v\sinh\Gamma}{Z(Z\sinh\Gamma + R_v\cosh\Gamma)}\Delta P_b \tag{5.11}$$

$$\frac{\Delta Q_b}{Q_e} = \frac{-(Z\cosh\Gamma + R_v\sinh\Gamma)(R_s + R_v)}{Z(Z\sinh\Gamma + R_v\cosh\Gamma)}\frac{\Delta P_b}{\left(P_i - P_o\right)}$$

or (5.12)

$$= \frac{-(\frac{Z}{R_s}\cosh\Gamma + R_r\sinh\Gamma)(1 + R_r)}{Z(\frac{Z}{R_s}\sinh\Gamma + R_r\cosh\Gamma)}\frac{\Delta P_b}{\left(P_i - P_o\right)}$$

Also, the transfer function of the ΔP_a can be shown to be

$$\Delta P_a = \frac{R_v}{Z\sinh\Gamma + R_v\cosh\Gamma}\Delta P_b \tag{5.13}$$

Case 3:

Consider the schematic of a fluid line shown in Figure 5.3. The pressures at the ends of the line are denoted by P_i and P_b; the volumetric flow rates at the ends are denoted by Q_a and Q_b; R_v is a linear resistance attached at the right end of line to account for turbulence. Assume that the flow through the line is initially constant. An additional example are pulsations associated with a sudden closing of a valve; in this

case, the flow Q_b suddenly decreases or even goes to zero creating a blocked end condition on the line.

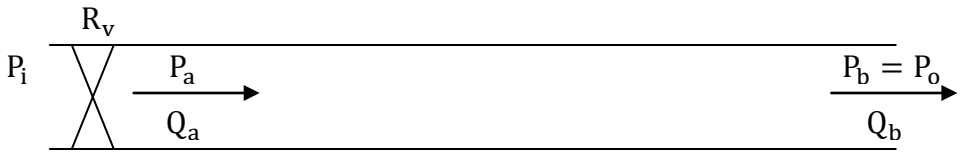


Figure 5.3 Schematic of a fluid transmission line for Case 3

The flow rate Q_a equation for the distributed parameter model for case 3 is obtained from the Equation (2.2). Then the equations of this model are

$$\begin{aligned} \Delta P_i - \Delta P_a &= R_v \cdot \Delta Q_a \\ \Delta Q_a &= \frac{\sinh \Gamma}{Z \cosh \Gamma} \Delta P_a + \frac{1}{\cosh \Gamma} \Delta Q_b \end{aligned} \tag{5.14}$$

Since $\Delta P_i = 0$, the ΔQ_a can be rewritten as

$$\begin{aligned} \Delta Q_a &= \frac{\sinh \Gamma}{Z \cosh \Gamma} (\Delta P_i - R_v \Delta Q_a) + \frac{1}{\cosh \Gamma} \Delta Q_b \\ &= \frac{Z}{Z \cosh \Gamma + R_v \sinh \Gamma} \Delta Q_b \end{aligned} \tag{5.15}$$

In steady state, the flow rate difference through the line will be the same value, Q_e, and the pressure difference becomes $(P_i - P_o)$. The steady state equation can be written as $P_i - P_o = \left(R_s + R_v\right) Q_e$.

Then, the transfer function of the normalized flow rate ΔQ_a becomes

$$\frac{\Delta Q_a}{Q_e} = \frac{Z}{Z\cosh\Gamma + R_v \sinh\Gamma}\frac{\Delta Q_b}{Q_e}$$

or (5.16)

$$= \frac{\frac{Z}{R_s}}{\frac{Z}{R_s}\cosh\Gamma + R_r \sinh\Gamma}\frac{\Delta Q_b}{Q_e}$$

Using the same procedure, the transfer function of the normalized ΔP_b is obtained as follows.

$$\Delta P_b = -\frac{Z(Z\sinh\Gamma + R_v\cosh\Gamma)}{Z\cosh\Gamma + R_v\sinh\Gamma}\Delta Q_b \tag{5.17}$$

$$\frac{\Delta P_b}{(P_i - P_o)} = -\frac{Z\,(Z\sinh\Gamma + R_v\cosh\Gamma)/(R_s + R_v)}{Z\cosh\Gamma + R_v\sinh\Gamma}\frac{\Delta Q_b}{Q_e}$$

or (5.18)

$$= -\frac{\frac{Z}{R_s}\left(\frac{Z}{R_s}\sinh\Gamma + R_r\cosh\Gamma\right)\Big/(1+R_r)}{\frac{Z}{R_s}\cosh\Gamma + R_r\sinh\Gamma}\frac{\Delta Q_b}{Q_e}$$

Likewise, the transfer function of the normalized ΔP_a can be derived as follows.

$$\Delta P_a = -\frac{ZR_v}{Z\cosh\Gamma + R_v\sinh\Gamma}\Delta Q_b \tag{5.19}$$

$$\frac{\Delta P_a}{(P_i - P_o)} = \frac{-Z\,R_v/(R_s + R_v)}{Z\cosh\Gamma + R_v\sinh\Gamma}\frac{\Delta Q_b}{Q_e}$$

or (5.20)

$$= \frac{-\frac{Z}{R_s}R_r/(1+R_r)}{\frac{Z}{R_s}\cosh\Gamma + R_r\sinh\Gamma}\frac{\Delta Q_b}{Q_e}$$

By inspecting the transfer functions obtained for each case, we see that case 1 and case 2 has similar transfer functions. That is, compare the transfer functions in Equations (5.4) and. (5.10). The same is true for Equations (5.6) and. (5.12). Thus, modeling simplifications obtained for one of the cases and be used for the other cases without having to redo the curve fitting and analysis.

Case 3 is a different model from case 1 and 2 because it has a change in the volumetric flow rate as the input instead of a pressure change.

5.2 Tapered Lumped Parameter Model

Using the boundary conditions described in Chapter 5.1, lumped parameter modeling is used to demonstrate the similarity and difference of each case.

Case 1:

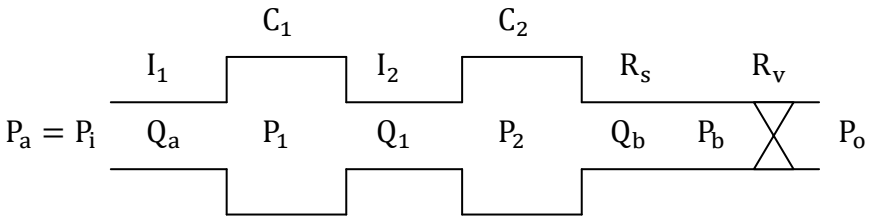


Figure 5.4 Schematic of lumped parameter model of a line for Case 1

The equations of the lumped parameter model are

$$\begin{aligned} P_a - P_1 &= I_1 s Q_a \\ Q_a - Q_2 &= C_1 s P_1 \\ P_1 - P_2 &= I_2 s Q_2 \\ Q_2 - Q_b &= C_2 s P_2 \\ P_2 - P_b &= \left(R_s + R_v\right) Q_b \end{aligned} \tag{5.21}$$

By solving the symbolic equations, we have symbolic transfer functions of normalized ΔQ_b and ΔQ_a.

$$\frac{\Delta Q_b}{Q_e} = \frac{1}{\begin{array}{l}(C_1C_2I_1I_2)s^4 + (C_1I_1I_2)/(R_s+R_v)\,s^3 + \\ (C_1I_1 + C_2I_1 + C_2I_2)s^2 + (I_1+I_2)/(R_s+R_v)\,s+1\end{array}} \frac{\Delta P_a}{\left(P_i - P_o\right)} \tag{5.22}$$

$$\frac{\Delta Q_a}{Q_e} = \frac{C_1C_2I_1(R_s+R_v)s^3 + (C_1I_2)s^2 + (C_1+C_2)(R_s+R_v)s+1}{\begin{array}{l}(C_1C_2I_1I_2)s^4 + (C_1I_1I_2)/(R_s+R_v)\,s^3 + \\ (C_1I_1 + C_2I_1 + C_2I_2)s^2 + (I_1+I_2)/(R_s+R_v)\,s+1\end{array}} \frac{\Delta P_a}{\left(P_i - P_o\right)} \tag{5.23}$$

Case 2:

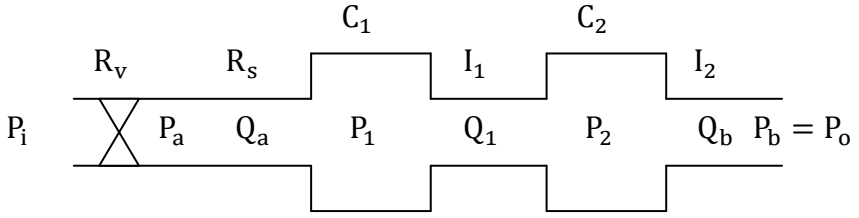


Figure 5.5 Schematic of lumped parameter model of a line for Case 2

The equations of the lumped parameter model are

$$\begin{aligned} P_i - P_1 &= \left(R_s + R_v\right)Q_a \\ Q_a - Q_1 &= C_1 s P_1 \\ P_1 - P_2 &= I_1 s Q_1 \\ Q_1 - Q_b &= C_2 s P_2 \\ P_2 - P_b &= I_2 s Q_b \end{aligned} \tag{5.24}$$

By solving the symbolic equations, we have symbolic transfer functions of normalized ΔQ_a and ΔQ_b.

$$\frac{\Delta Q_a}{Q_e} = -\frac{1}{\begin{array}{l}(C_1C_2I_1I_2)s^4 + (C_2I_1I_2)/(R_s+R_v)\,s^3 + \\ (C_1I_1 + C_1I_2 + C_2I_2)s^2 + (I_1+I_2)/(R_s+R_v)\,s+1\end{array}} \frac{\Delta P_b}{\left(P_i - P_o\right)} \tag{5.25}$$

$$\frac{\Delta Q_b}{Q_e} = -\frac{C_1C_2I_2(R_s+R_s)s^3+(C_2I_1)s^2+(C_1+C_2)(R_s+R_v)s+1}{(C_1C_2I_1I_2)s^4+(C_2I_1I_2)/(R_s+R_v)\,s^3+ \\ (C_1I_1+C_1I_2+C_2I_2)s^2+(I_1+I_2)/(R_s+R_v)\,s+1}\frac{\Delta P_b}{(P_i-P_o)} \tag{5.26}$$

Comparing (5.22) with (5.25), it is observed that these transfer functions are identical except for the negative sign if C_1 and C_2 are switched and I_1 and I_2 are switched. The same is true for the transfer functions in (5.23) and (5.26).

Case 3:

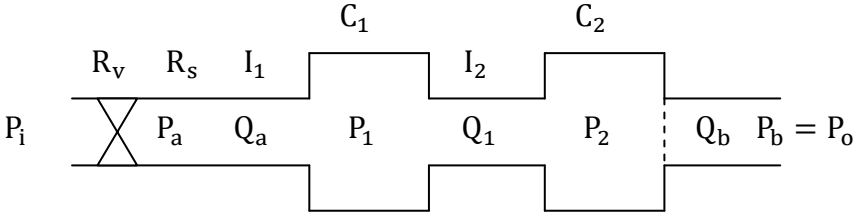


Figure 5.6 Schematic of lumped parameter model of a line for Case 3

The equations of the lumped parameter model are

$$\begin{aligned} P_i - P_a &= (R_s+R_v)Q_a + I_1sQ_a \\ Q_a - Q_2 &= C_1sP_1 \\ P_1 - P_b &= I_2sQ_2 \\ Q_2 - Q_b &= C_2sP_b \end{aligned} \tag{5.27}$$

By solving the symbolic equations, we have symbolic transfer functions of normalized ΔQ_a and ΔP_a.

$$\frac{\Delta Q_a}{Q_e} = \frac{1}{(C_1C_2I_1I_2)s^4+(C_1C_2I_2(R_s+R_v))s^3+ \\ (C_1I_1+C_2I_1+C_2I_2)s^2+(C_1+C_2)(R_s+R_v)s+1}\frac{\Delta Q_b}{Q_e} \tag{5.28}$$

$$\frac{\Delta P_b}{\left(P_i - P_o\right)} = -\frac{\begin{array}{c}(C_1 I_1 I_2)/(R_s + R_v)\, s^3 + (C_1 I_2) s^2 + (I_1 + I_2)/(R_s + R_v)\, s \\ +1\end{array}}{\begin{array}{l}(C_1 C_2 I_1 I_2) s^4 + (C_1 C_2 I_2 (R_s + R_v)) s^3 + \\ (C_1 I_1 + C_2 I_1 + C_2 I_2) s^2 + (C_1 + C_2)(R_s + R_v) s + 1\end{array}} \frac{\Delta Q_b}{Q_e} \tag{5.29}$$

5.3 Demonstration with Turbulent Flow Conditions

The same procedures used in Chapter 4 can be applied to Case 1 and Case 3 without Case 2 since Case 1 and Case 2 are considered as the same type of model.

To start, the ratio R_r for turbulent flow is introduced.

The turbulent resistance equation [15, 16] for incompressible steady flow of the smooth circular transmission line is

$$F(Q) = P = C_f Q^{1.75} \tag{5.30}$$

See the derivation of C_f in Chapter 4.1.

$$C_f = \frac{0.2414 \rho^{0.75} \mu^{0.25} L}{d^{4.75}} \tag{5.31}$$

Rewrite the Equation (5.30) using the equivalent resistance gives

$$F(Q) = P = C_f Q_e^{1.75} = R_t Q_e \tag{5.32}$$

$$Q_e = R_n \nu A / D = R_n \nu \pi D / 4 \tag{5.33}$$

$$\mu = \rho \nu \tag{5.34}$$

The turbulent resistance R_t is defined as

$$\begin{aligned} R_t &= C_f Q_0^{0.75} \\ &= \left(\frac{0.2414 \rho^{0.75} \left(\rho \nu\right)^{0.25} L}{D^{4.75}} \right) \left(\frac{R_n \nu \pi D}{4} \right)^{0.75} \end{aligned} \tag{5.35}$$

Also,

$$R_t = R_s\left(1+R_r\right) = \frac{128\rho\nu L}{\pi D^4}\left(1+R_r\right) \tag{5.36}$$

Combining Equations (5.35) and (5.36), we got the resistance ratio R_r corresponding to Reynolds number R_n.

$$\begin{aligned} \frac{128}{\pi}\left(1+R_r\right) &= \frac{0.2414 * R_n^{0.75} * \pi^{0.75}}{4^{0.75}} \\ \left(1+R_r\right) &= \left[\frac{\pi^{1.75}}{1499.746}\right] R_n^{0.75} = \frac{R_n^{0.75}}{202.3044} \end{aligned} \tag{5.37}$$

We will use R_r instead of R_n in developing the tables of inertance and capacitance lump coefficients of the lumped parameter model.

Since the Equation (5.37) is derived using the friction factor from Blasius [16], it is good for $3000 < R_n < 10^5$ that is corresponding to $1 < R_r < 26.8$.

Therefore, we will consider using the value $R_r = 0 \sim 40$ to develop the tables and show the results in figures to evaluate their polynomial equations.

The following tables and corresponding figures of I's and C's coefficients for the lumped parameter model were obtained values of R_r for both Case 1 and Case 3 with four dissipation numbers $Dn = [0.04139 \quad 0.033112 \quad 0.024834 \quad 0.016556]$,.

Case 1:

Table 5.1 Case 1, 4th order lumped parameter model tapering, D_n =0.04139

R_r	0	1	2	3	5	10
wI(1)	0.67565	0.63807	0.666	0.66756	0.64684	0.85694
wI(2)	0.62843	0.63237	0.67057	0.73826	0.53435	0.63461
wC(1)	0.36092	0.39772	0.39337	0.41552	0.27422	0.22705
wC(2)	0.04627	0.1445	0.16237	0.16711	0.21762	0.19415

Table 5.1—*Continued*

R_r	20	30	40
wI(1)	1.2949	1.6989	2.0999
wI(2)	0.87367	1.1094	1.3459
wC(1)	0.1618	0.12696	0.10464
wC(2)	0.14242	0.11144	0.091229

Table 5.2 Case 1, 4th order lumped parameter model tapering, D_n =0.033112

R_r	0	5	10	20	30	40
wI(1)	0.70421	0.64838	0.74072	1.055	1.3418	1.6223
wI(2)	0.5968	0.48054	0.51592	0.67018	0.82147	0.97269
wC(1)	0.35947	0.2556	0.2535	0.18746	0.15118	0.12727
wC(2)	8.0582e-009	0.24715	0.23023	0.17836	0.14472	0.12139

Table 5.3 Case 1, 4th order lumped parameter model tapering, D_n =0.024834

R_r	0	5	10	20	30	40
wI(1)	0.71531	0.6227	0.67297	0.85005	1.0309	1.2098
wI(2)	0.545	0.65156	0.4862	0.558	0.64872	0.74253
wC(1)	0.37294	0.44008	0.26491	0.22279	0.18899	0.16422
wC(2)	1.2074e-013	0.17758	0.25307	0.21754	0.18497	0.15996

Table 5.4 Case 1, 4th order lumped parameter model tapering, D_n =0.016556

R_r	0	5	10	20	30	40
wI(1)	0.77202	0.621	0.53675	0.67328	0.78159	0.88038
wI(2)	0.50403	0.57672	0.45742	0.47972	0.51201	0.55196
wC(1)	0.37201	0.42995	0.27475	0.24696	0.22581	0.20801
wC(2)	4.0381e-013	0.15957	0.25067	0.24818	0.23195	0.2129

0 real + 2 complex eigenvalue
1 real + 1 complex eigenvalue

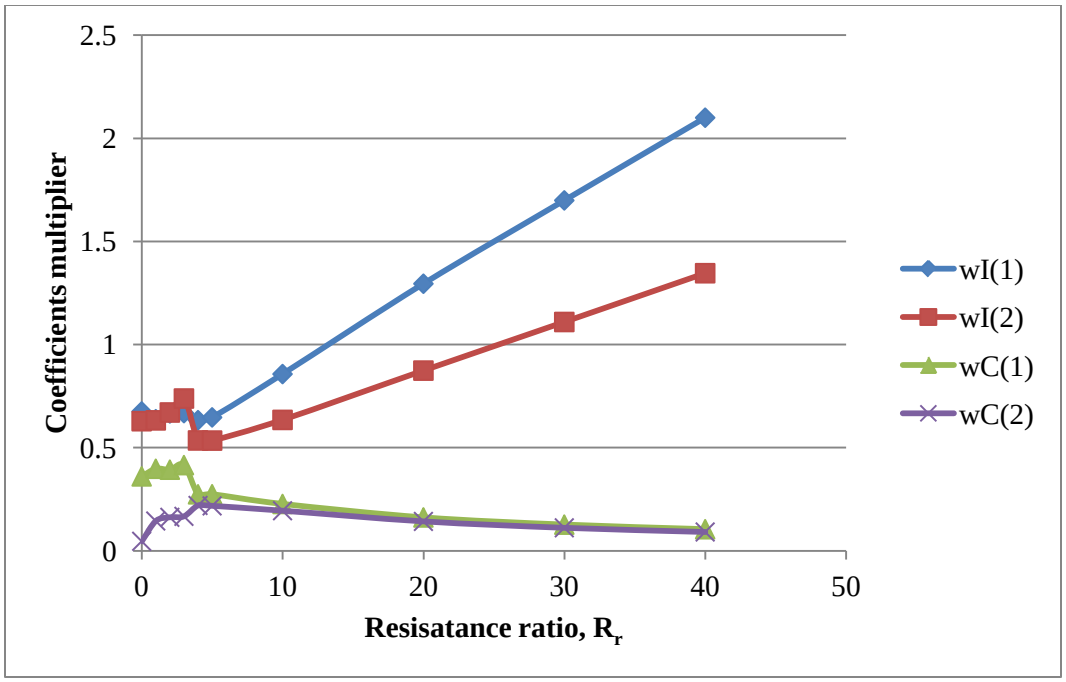


Figure 5.7 Case 1, 4th order lumped parameter model tapering, D_n =0.04139 (a)

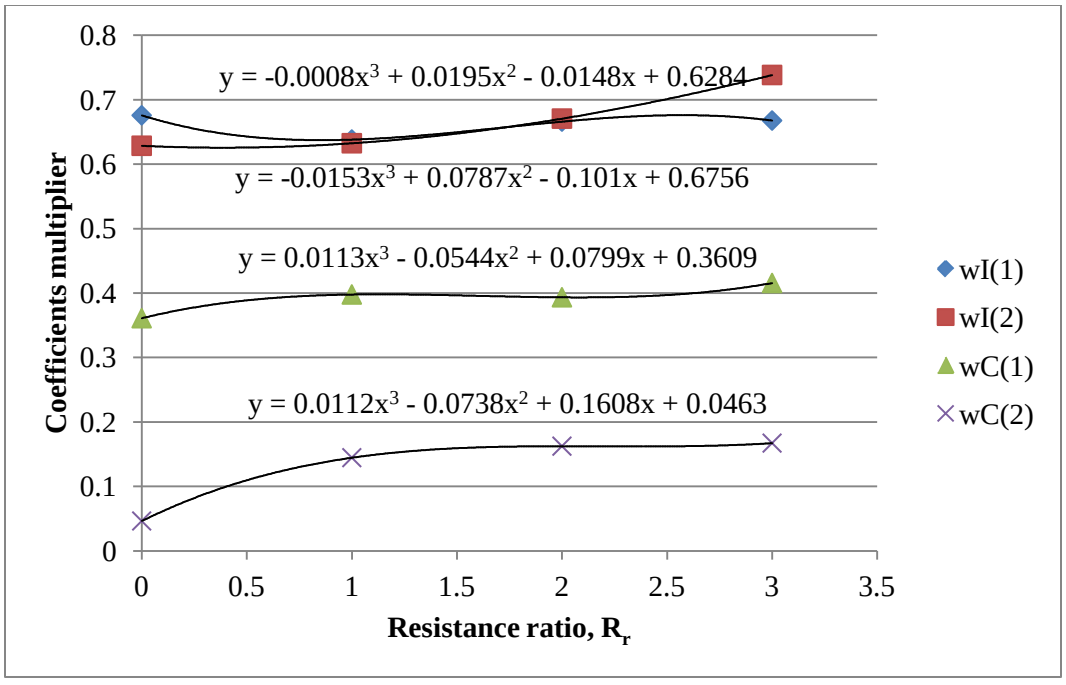


Figure 5.8 Case 1, 4th order lumped parameter model tapering, D_n =0.04139 (b)

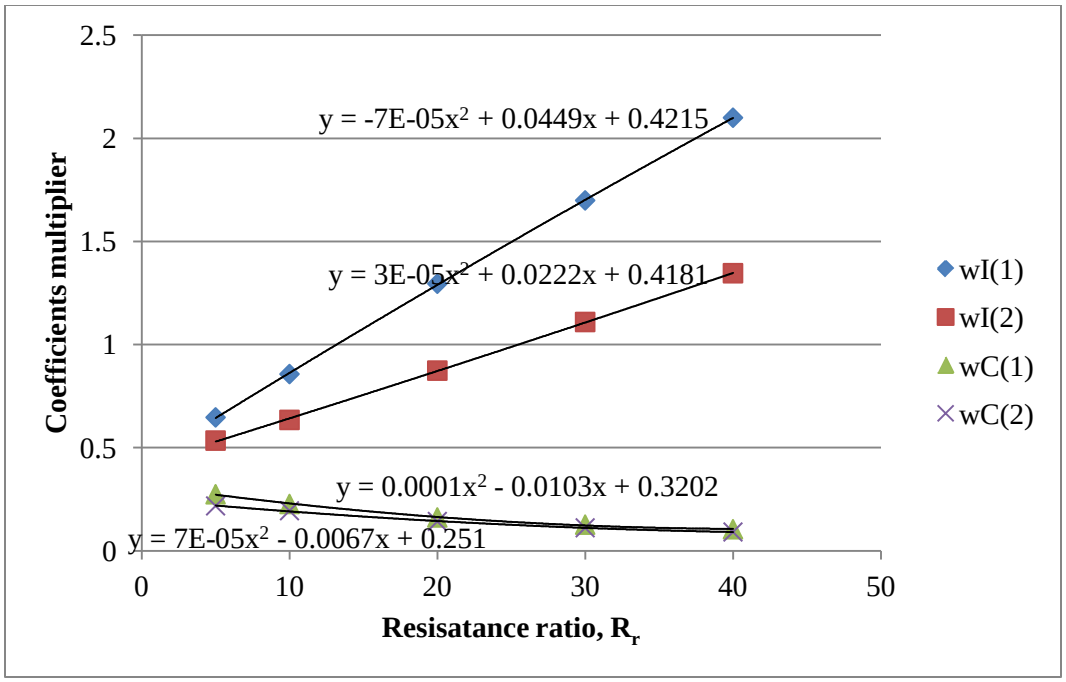


Figure 5.9 Case 1, 4th order lumped parameter model tapering, D_n =0.04139 (c)

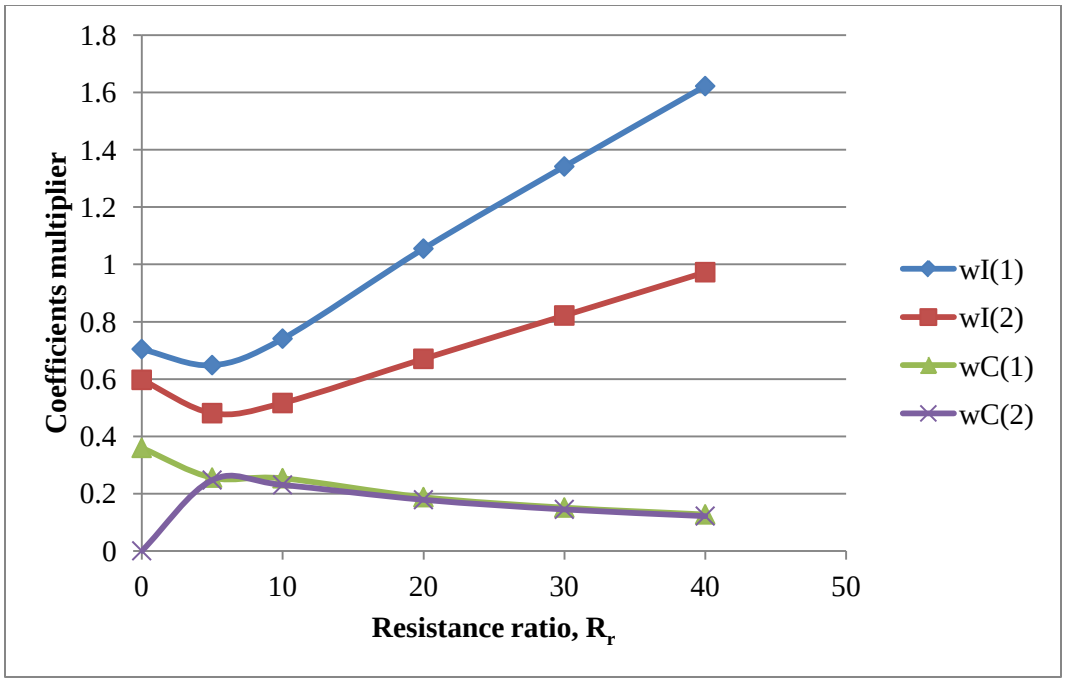


Figure 5.10 Case 1, 4th order lumped parameter model tapering, D_n =0.033112 (a)

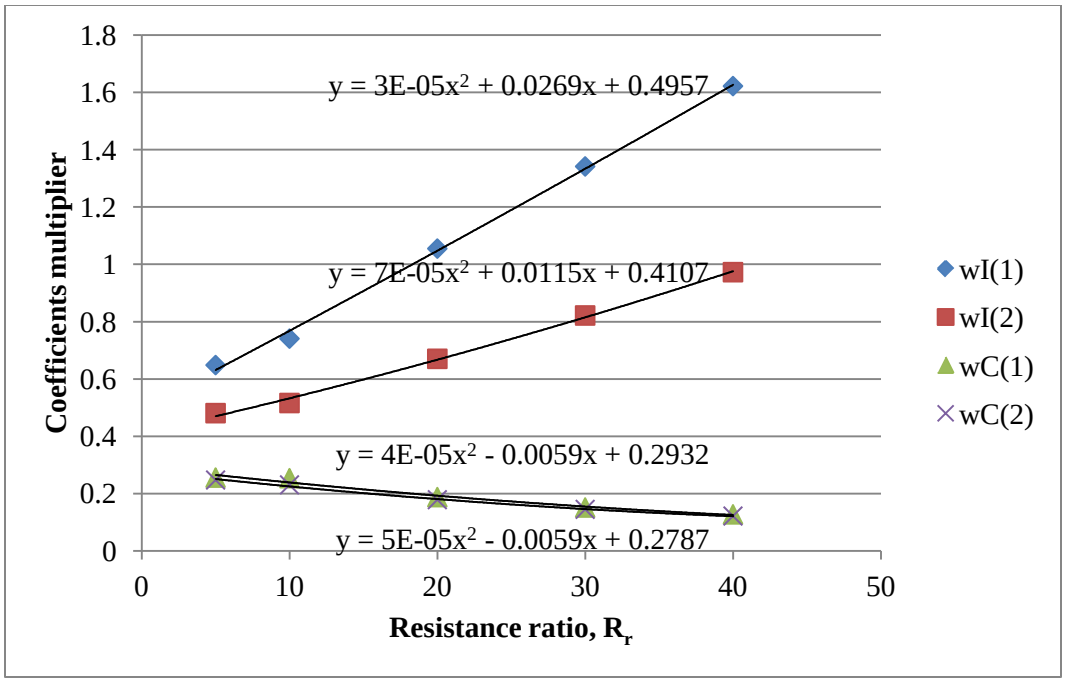


Figure 5.11 Case 1, 4th order lumped parameter model tapering, D_n =0.033112 (b)

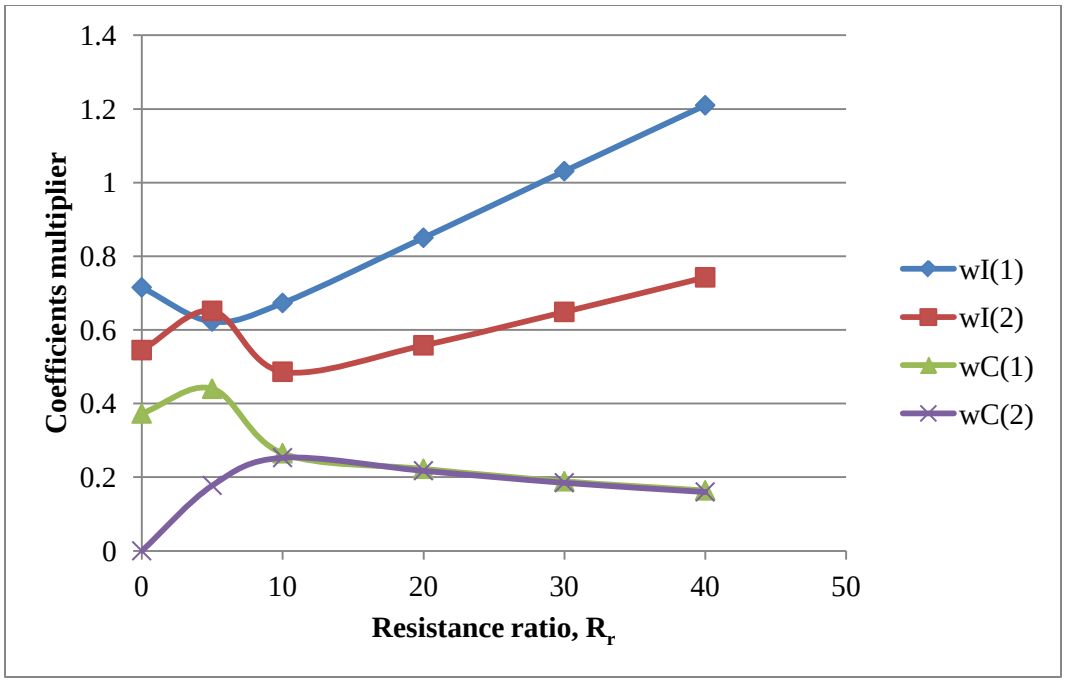


Figure 5.12 Case 1, 4th order lumped parameter model tapering, D_n =0.024834 (a)

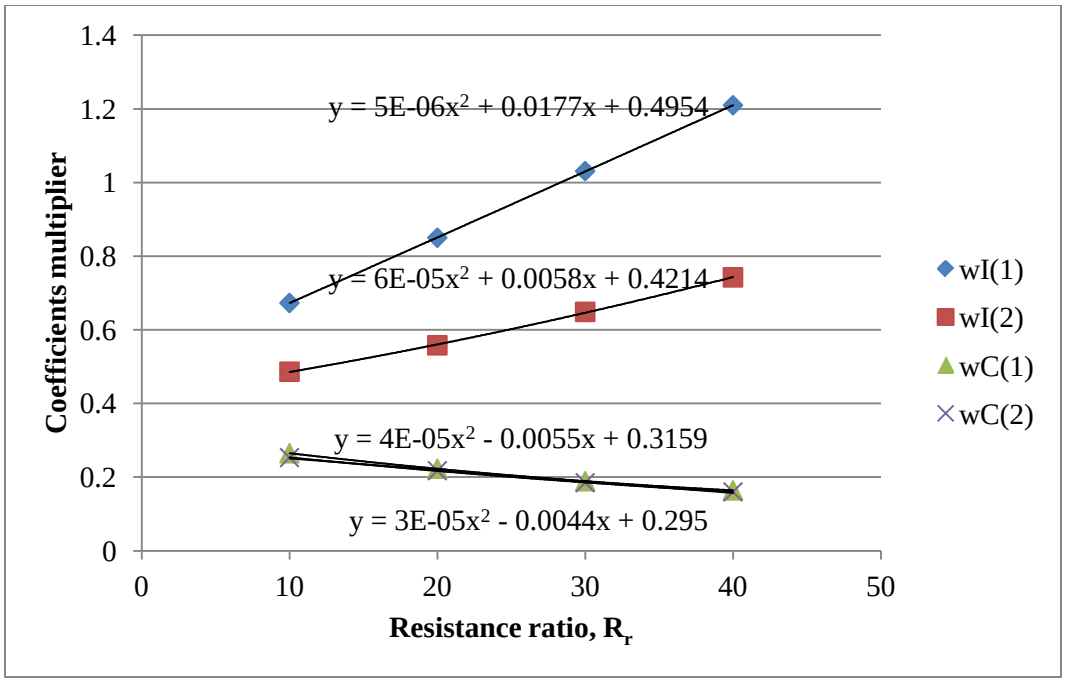


Figure 5.13 Case 1, 4th order lumped parameter model tapering, D_n=0.024834 (b)

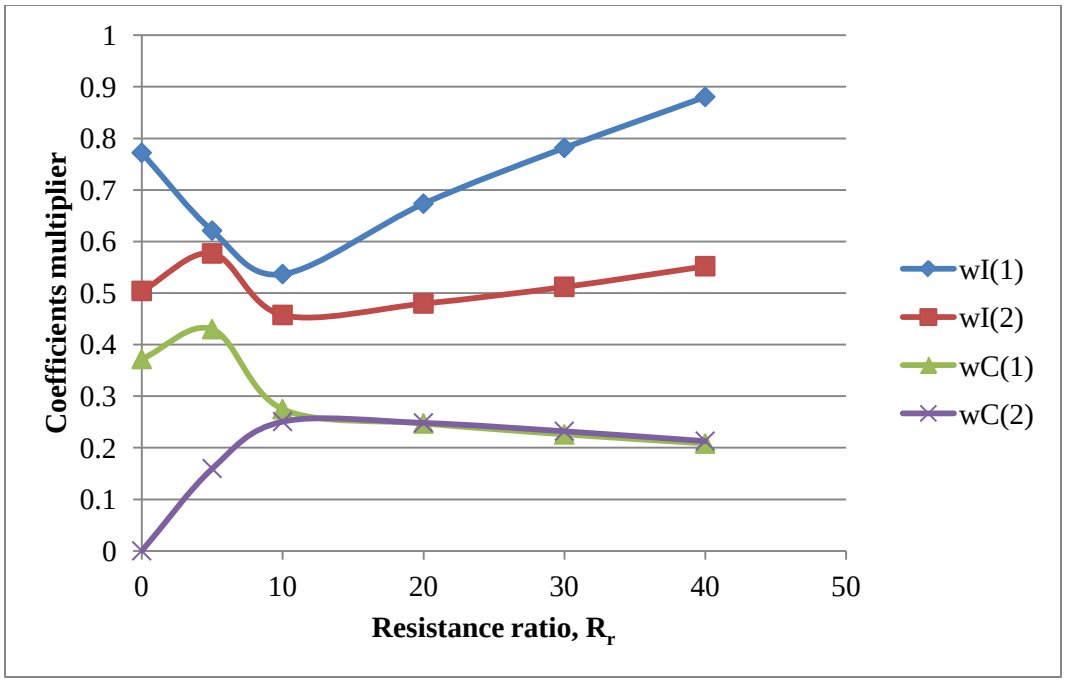


Figure 5.14 Case 1, 4th order lumped parameter model tapering, D_n=0.016556 (a)

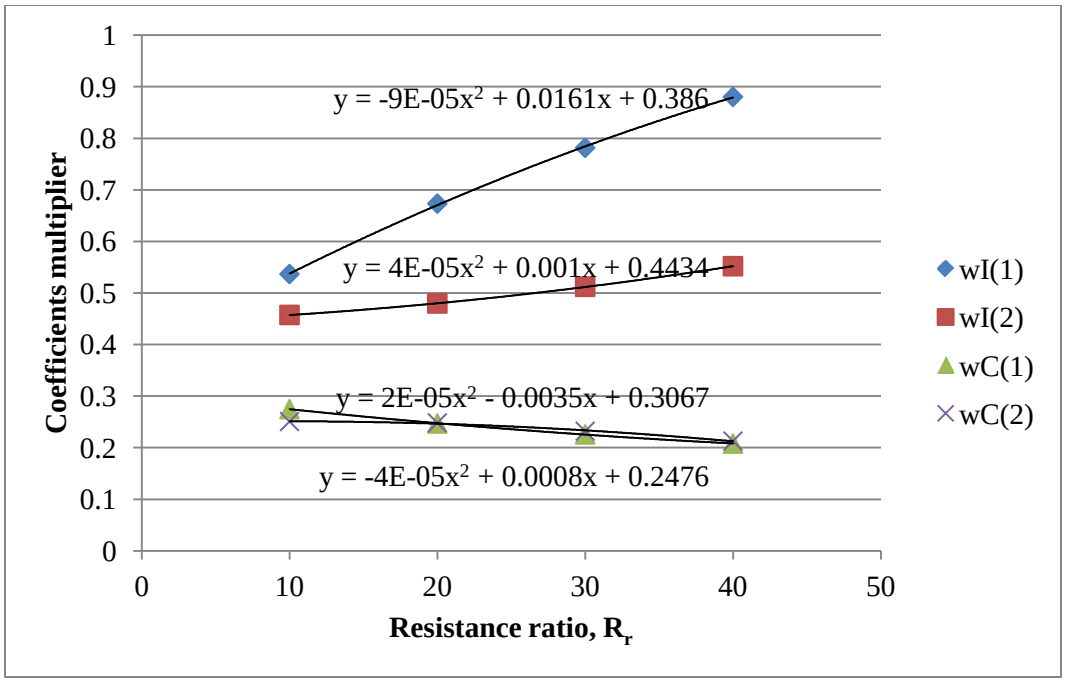


Figure 5.15 Case 1, 4th order lumped parameter model tapering, D_n =0.016556 (b)

Case 3:

Table 5.5 Case 3, 4th order lumped parameter model tapering, D_n =0.04139

R_r	0	5	7	10	20
wI(1)	0.22525	1.9977e-013	2.2478e-013	1.5768e-013	4.0036e-012
wI(2)	0.76972	0.78239	0.84706	0.873	1.2124
wC(1)	0.27642	0.19913	0.16164	0.17434	0.1078
wC(2)	0.50364	0.69622	0.78325	0.80196	0.87867

R_r	30	40
wI(1)	7.2353e-012	9.0931e-012
wI(2)	1.4999	1.7893
wC(1)	0.085312	0.070519
wC(2)	0.90577	0.92257

0 real + 2 complex eigenvalue
1 real + 1 complex eigenvalue

Table 5.6 Case 3, 4th order lumped parameter model tapering, D_n =0.033112

R_r	0	1	3.35	5	7	10
wl(1)	0.18931	0.16779	1.6457e-008	1.1163e-013	8.0521e-014	1.9305e-013
wl(2)	0.71865	0.61118	0.89957	0.74697	0.76561	0.82039
wC(1)	0.32575	0.31824	0.26666	0.16101	0.17651	0.16515
wC(2)	0.5415	0.66768	1.2193	0.63565	0.75015	0.7998

R_r	20	30	40
wl(1)	7.4052e-013	2.7134e-012	3.5392e-012
wl(2)	1.0116	1.2063	1.4027
wC(1)	0.13018	0.10678	0.090399
wC(2)	0.85802	0.88625	0.9045

Table 5.7 Case 3, 4th order lumped parameter model tapering, D_n =0.024834

R_r	0	3.35	5	7	10	20
wl(1)	0.17027	0.096278	0.0063061	2.1106e-013	8.3616e-015	2.2657e-013
wl(2)	0.62898	0.55155	0.83925	0.70887	0.73362	0.86128
wC(1)	0.36319	0.26593	0.23636	0.15711	0.17733	0.14763
wC(2)	0.59609	0.87088	1.5269	0.67216	0.76955	0.84024

R_r	30	40
wl(1)	3.441e-012	3.7482e-012
wl(2)	0.98917	1.1173
wC(1)	0.12704	0.11121
wC(2)	0.86697	0.88493

Table 5.8 Case 3, 4th order lumped parameter model tapering, D_n =0.016556

R_r	0	5	7	10	20	30
wl(1)	0.15929	0.095588	6.3845e-012	1.584e-013	1.4535e-012	2.178e-012
wl(2)	0.5012	0.49182	0.90161	0.7416	0.72909	0.79905
wC(1)	0.41248	0.29683	0.41744	0.17539	0.17727	0.1608
wC(2)	0.6883	0.87896	0.41744	0.62584	0.80227	0.833

R_r	40
wl(1)	1.1083e-012
wl(2)	0.86875
wC(1)	0.14633
wC(2)	0.85112

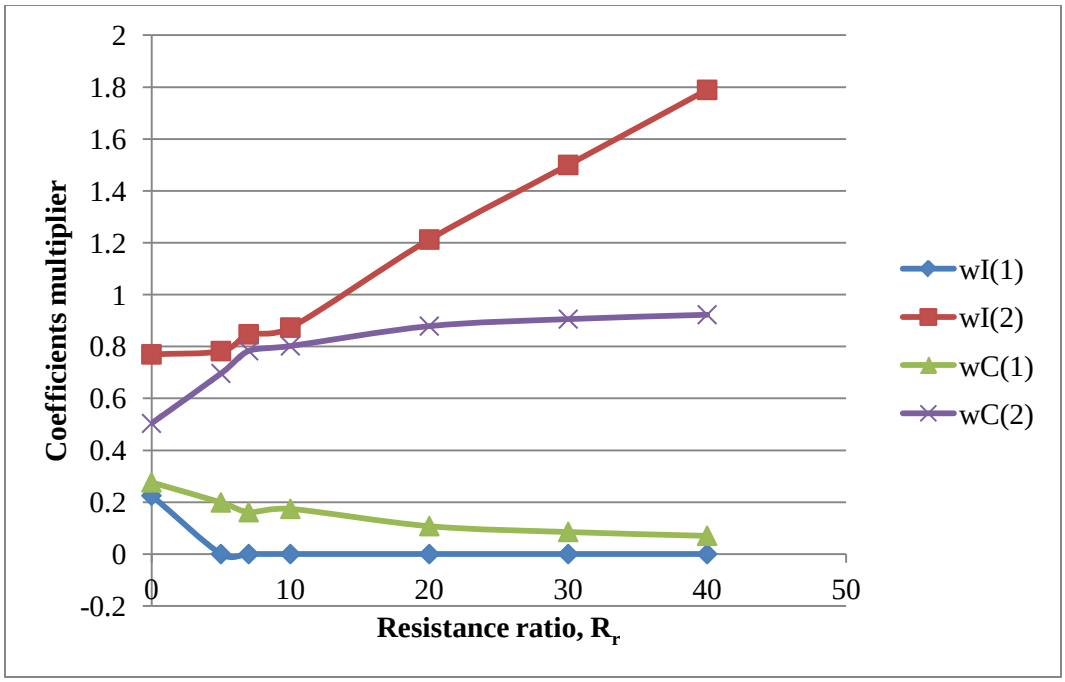


Figure 5.16 Case 3, 4th order lumped parameter model tapering, D_n =0.04139 (a)

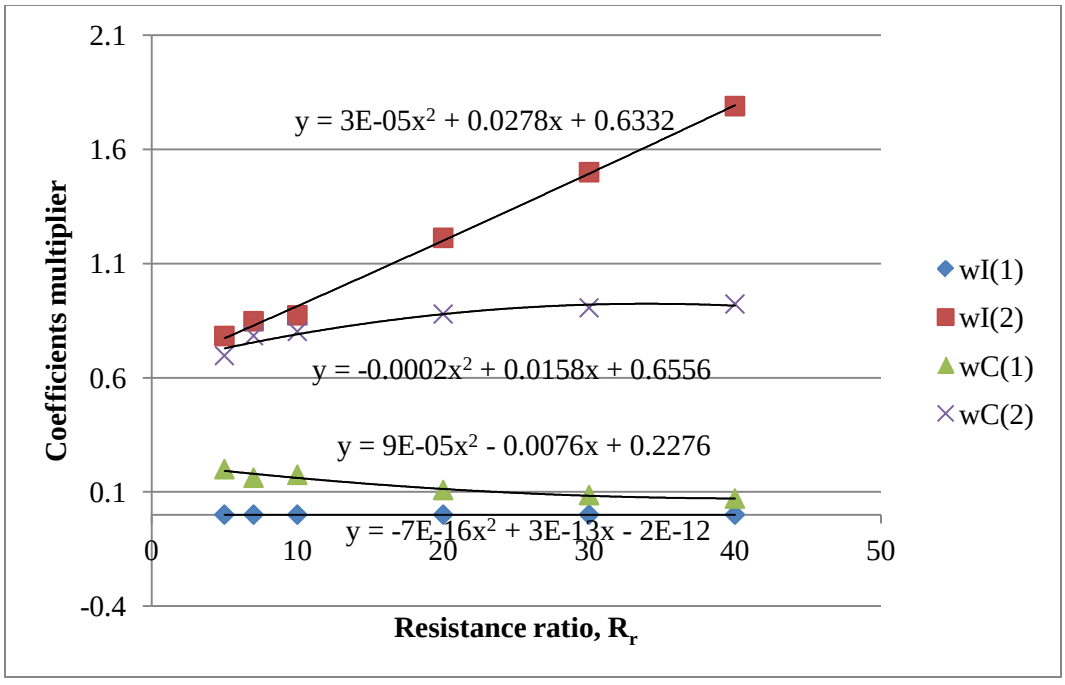


Figure 5.17 Case 3, 4th order lumped parameter model tapering, D_n =0.04139 (b)

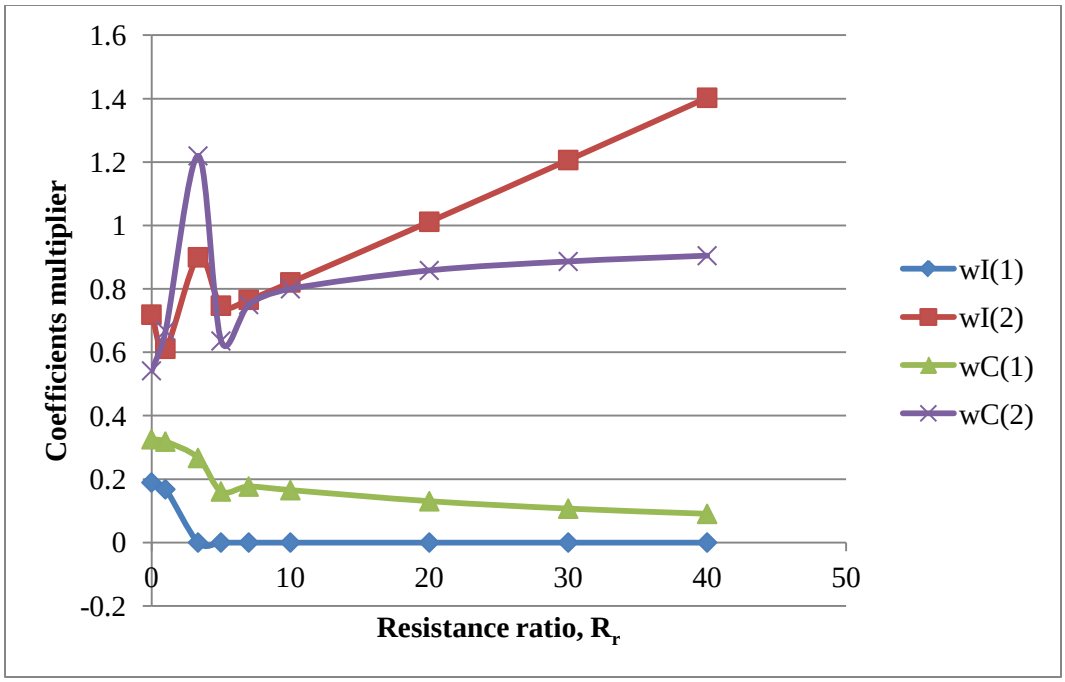


Figure 5.18 Case 3, 4th order lumped parameter model tapering, D_n =0.033112 (a)

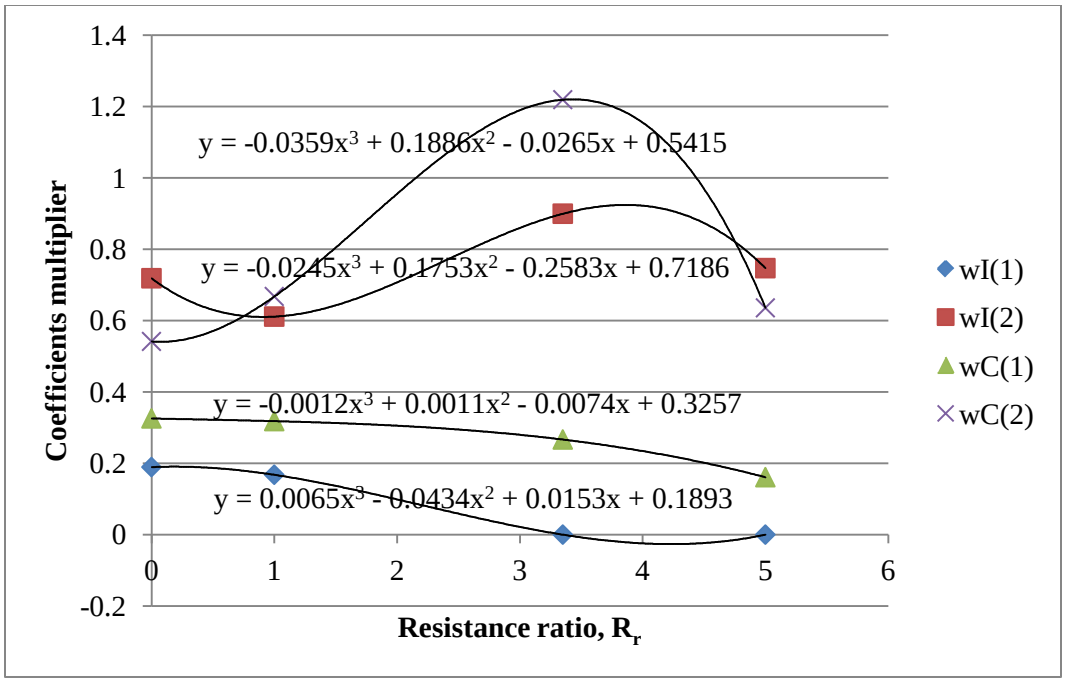


Figure 5.19 Case 3, 4th order lumped parameter model tapering, D_n =0.033112 (b)

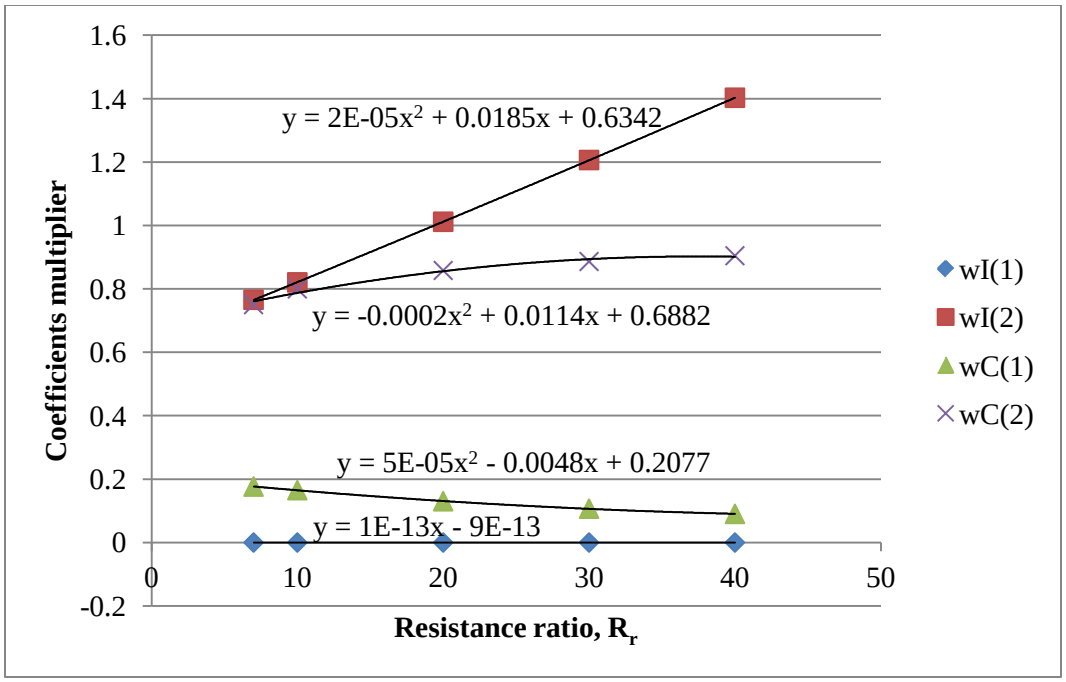


Figure 5.20 Case 3, 4th order lumped parameter model tapering, D_n=0.033112 (c)

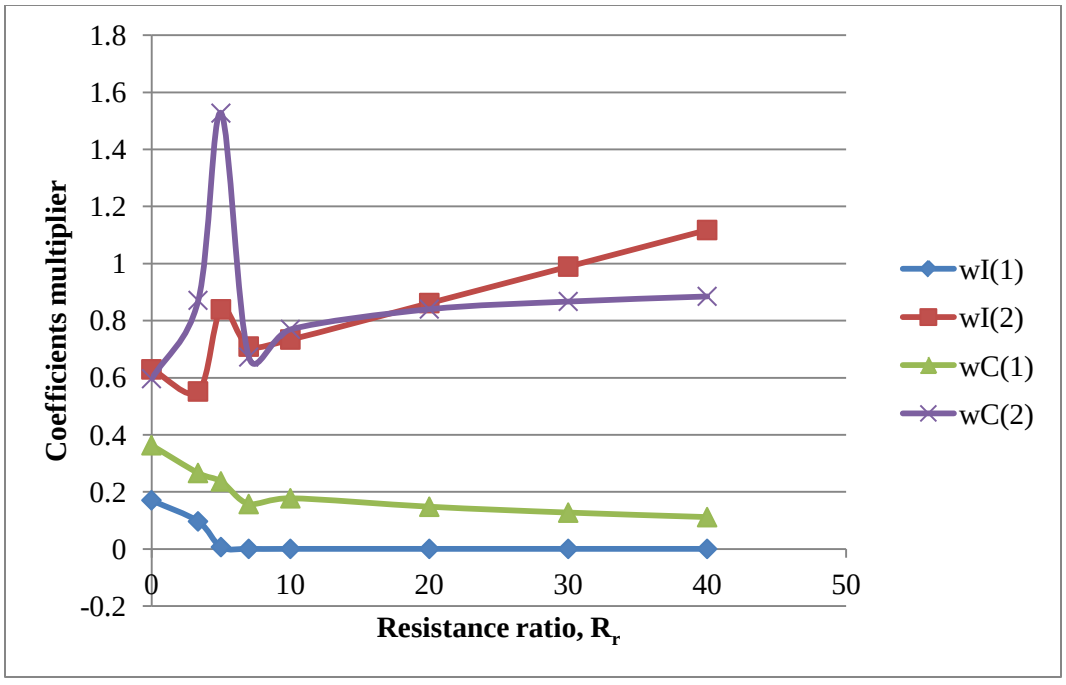


Figure 5.21 Case 3, 4th order lumped parameter model tapering, D_n=0.024834 (a)

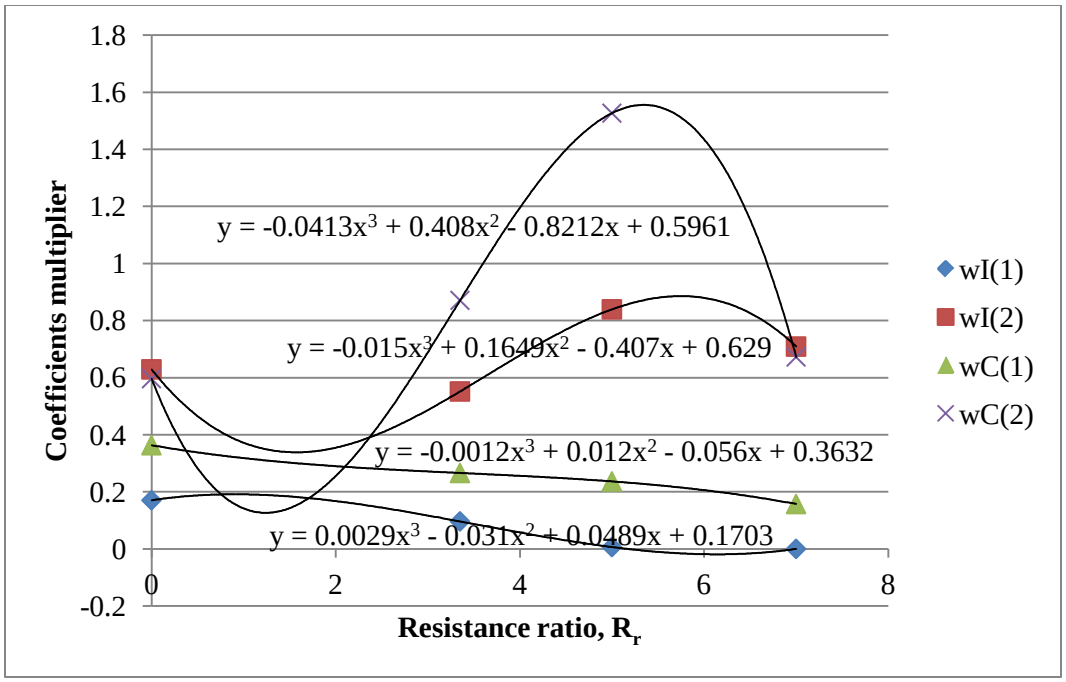


Figure 5.22 Case 3, 4th order lumped parameter model tapering, D_n =0.024834 (b)

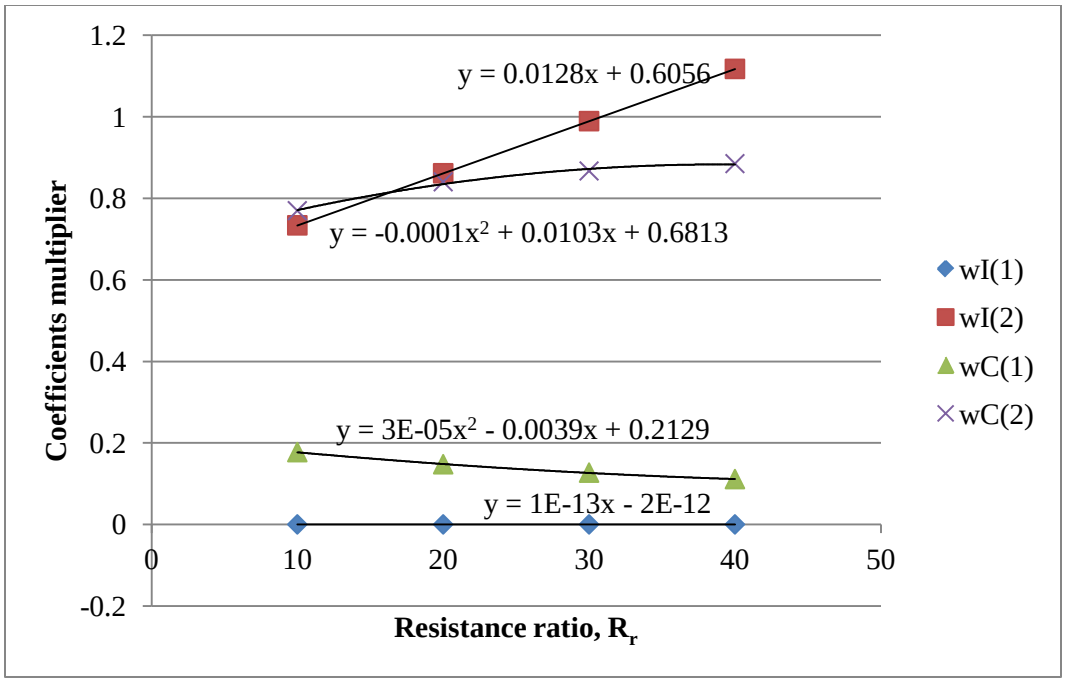


Figure 5.23 Case 3, 4th order lumped parameter model tapering, D_n =0.024834 (c)

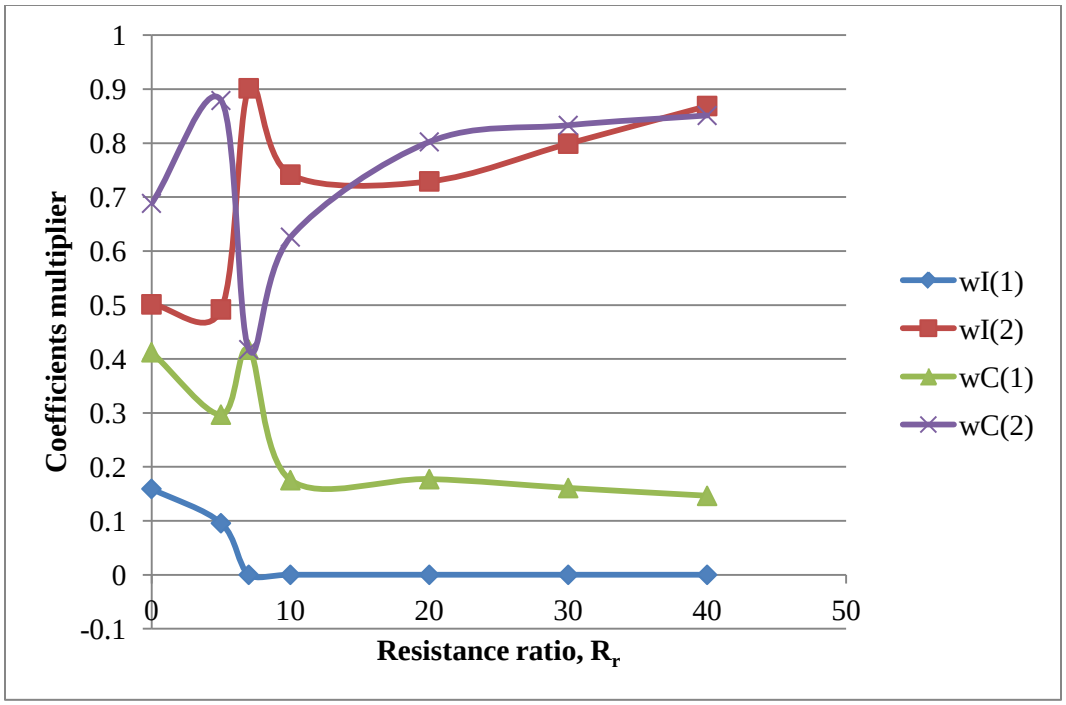


Figure 5.24 Case 3, 4th order lumped parameter model tapering, D_n =0.016556 (a)

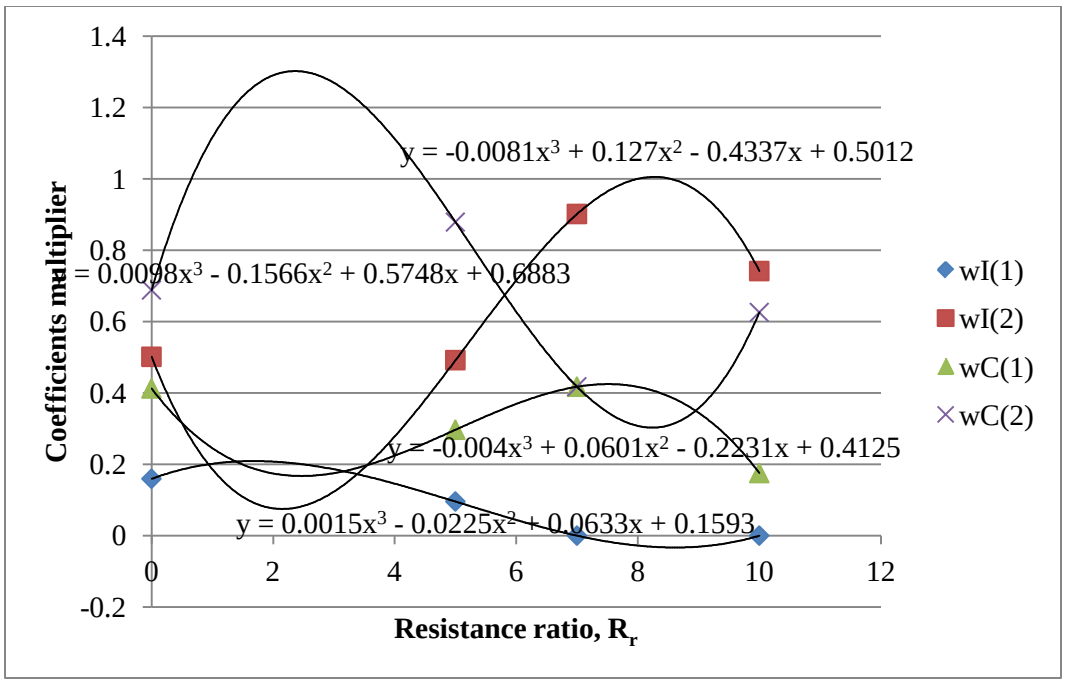


Figure 5.25 Case 3, 4th order lumped parameter model tapering, D_n =0.016556 (b)

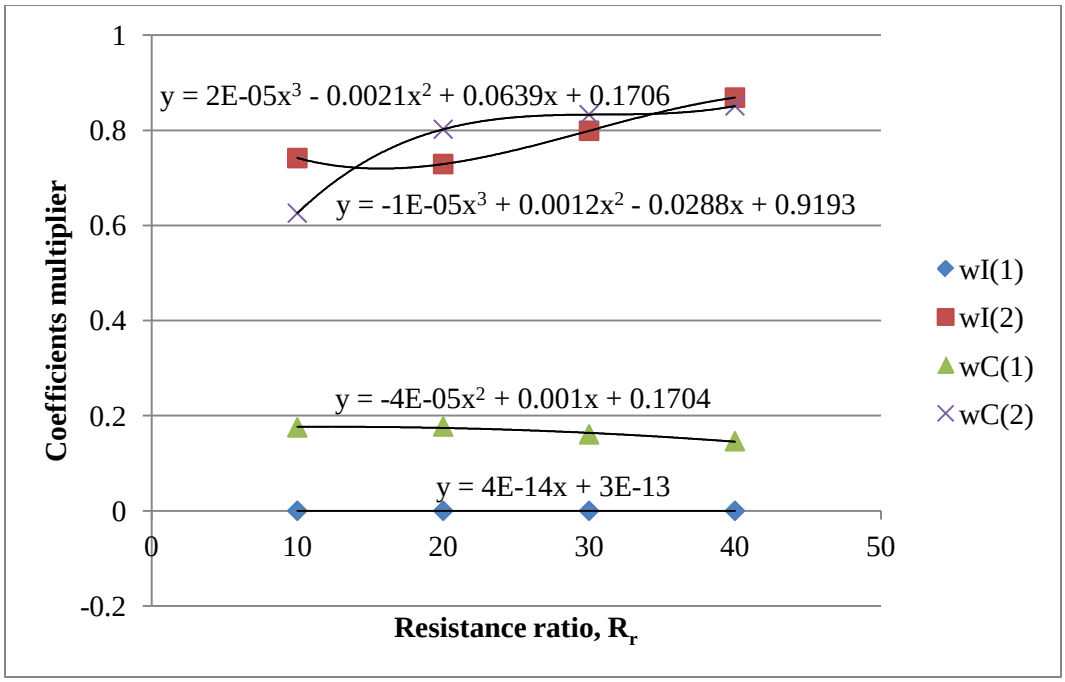


Figure 5.26 Case 3, 4th order lumped parameter model tapering, D_n =0.016556 (c)

5.4 Example Demonstrating Similarities of Transfer Functions

Chapter 5 has focused on variations of the line transfer functions for different inputs and outputs. It has been demonstrated that the results of optimizing the tapering coefficients for one set of boundary conditions can be also used for other transfer functions if certain guidelines are followed. This section contains an example demonstrating this versatility for different inputs and outputs.

Consider a fluid line with steady flow with a Reynolds number of 20,000, which corresponds to R_r = 7.3132. Also, assume a line and fluid with the same properties as the example in Chapter 2. This gives D_n = 0.04139. For ΔP_a the input and ΔQ_b the output, this is the boundary conditions for Case 1. Thus, the polynomial equations in Figure 5.9 can be used to calculate the coefficients of the lumps which correspond to the lumped parameter model shown in Figure 5.4. The results are as follows:

$$\begin{bmatrix} wI(1) & wI(2) & wC(1) & wC(2) \end{bmatrix} = \begin{bmatrix} 0.75003 & 0.58023 & 0.25022 & 0.20575 \end{bmatrix}$$

Therefore, with the same properties of line and fluid in Chapter 2, the lumps of inertance and capacitance will be

$$\begin{aligned} I_1 &= wI(1) * I_s = 0.75003 * 2.1604 * 10^9 = 1.6204 * 10^9 \\ I_2 &= wI(2) * I_s = 0.58023 * 2.1604 * 10^9 = 1.2535 * 10^9 \\ C_1 &= wC(1) * C_s = 0.25022 * 8.6784 * 10^{-14} = 2.1715 * 10^{14} \\ C_2 &= wC(2) * C_s = 0.20575 * 8.6784 * 10^{-14} = 1.7856 * 10^{-14} \end{aligned} \tag{5.38}$$

Then applying the lumps of inertance and capacitance to Equation (5.22) and (5.23) gives the transfer functions for the normalized flow perturbations ΔQ_b and ΔQ_a.

$$\frac{\Delta Q_b}{Q_e} = \frac{1}{(7.876 * 10^{-10})s^4 + (1.016 * 10^{-7})s^3 + (8.65 * 10^{-5})s^2 + (0.006617)s + 1} \frac{\Delta P_a}{(P_i - P_o)} \tag{5.39}$$

$$\frac{\Delta Q_a}{Q_e} = \frac{(2.729 * 10^{-7})s^3 + (2.722 * 10^{-5})s^2 + (0.01719)s + 1}{(7.876 * 10^{-10})s^4 + (1.016 * 10^{-7})s^3 + (8.65 * 10^{-5})s^2 + (0.006617)s + 1} \frac{\Delta P_a}{(P_i - P_o)} \tag{5.40}$$

As demonstrated for Case 2 in Section 5.2, these tapered coefficients can also be used for an input of ΔP_b and output of ΔQ_a which are boundary conditions for Case 2. This is possible as long as you swap wI(1) and wI(2) and also swap wC(1) and wC(2). Swapping these inertance and capacitance weighting parameters and recalculating the values for the lumped inertances and capacitances gives

$$\begin{bmatrix} I_1 & I_2 & C_1 & C_2 \end{bmatrix} = \begin{bmatrix} 1.2535 * 10^9 & 1.6204 * 10^9 & 1.7856 * 10^{-14} & 2.1715 * 10^{14} \end{bmatrix}$$

Substituting these values for the inertances and capacitances into Equations (5.25) and (5.26) gives the transfer functions for normalized ΔQ_a and ΔQ_b.

$$\frac{\Delta Q_a}{Q_e} = \frac{-1}{(7.876*10^{-10})s^4 + (1.016*10^{-7})s^3 + (8.65*10^{-5})s^2 + (0.006617)s + 1} \frac{\Delta P_b}{(P_i - P_o)} \tag{5.41}$$

$$\frac{\Delta Q_b}{Q_e} = -\frac{(2.729*10^{-7})s^3 + (2.722*10^{-5})s^2 + (0.01719)s + 1}{(7.876*10^{-10})s^4 + (1.016*10^{-7})s^3 + (8.65*10^{-5})s^2 + (0.006617)s + 1} \frac{\Delta P_b}{(P_i - P_o)} \tag{5.42}$$

It is important to note that the transfer functions in Equations (5.41) and (5.42) are identical to the transfer functions in Equations (5.39) and (5.40) if the "a's" and "b's" are swapped and negative signs are added. This is the reason that the tapering coefficients for Case 1 can be used in Case 2 if the weighting coefficients are swapped.

To demonstrate the accuracy of the 4th order transfer function of the tapered lumped parameter model with turbulence, comparisons with the 10th order approximation of the distributed parameter model with added resistance for turbulence are shown in Figure 5.27 and Figure 5.28. Considering that the tapered lumped model is only 4th order with a zero order numerator, the comparison reveals that both models formulated to account for turbulence are very closely matched in both the time domain and the frequency domain.

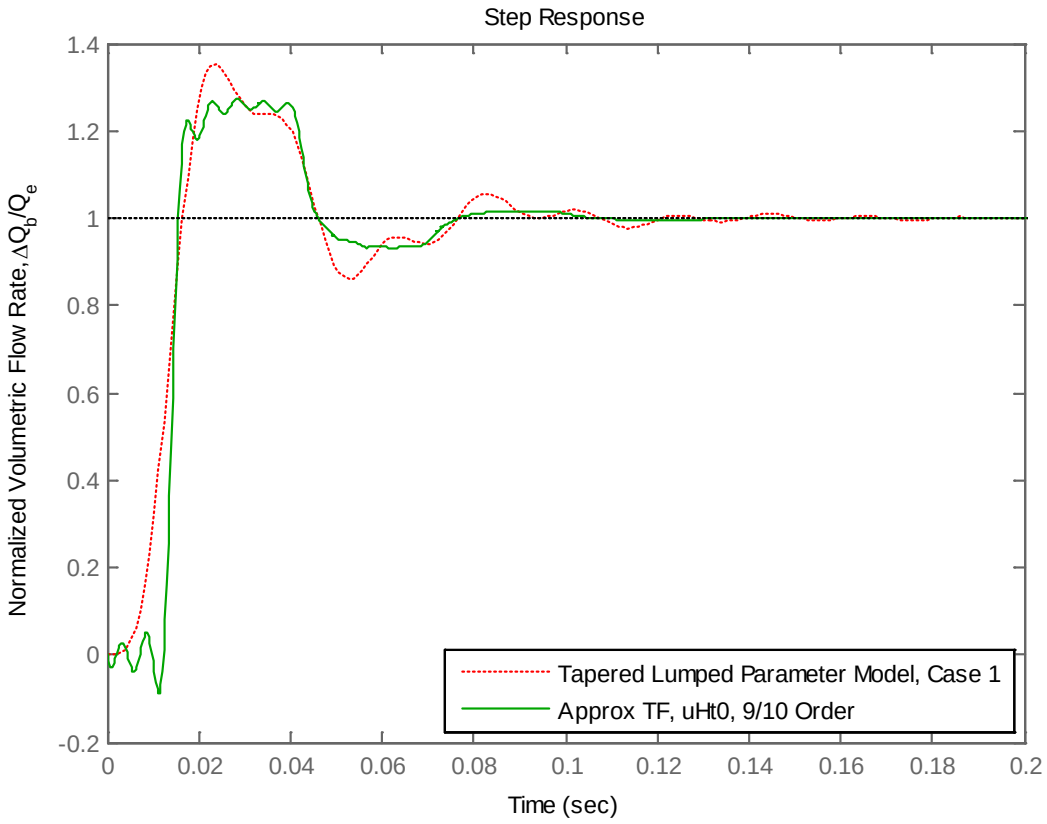


Figure 5.27 Comparison of step responses of 4th order tapered lumped parameter model and 9th/10th order approximate transfer function model

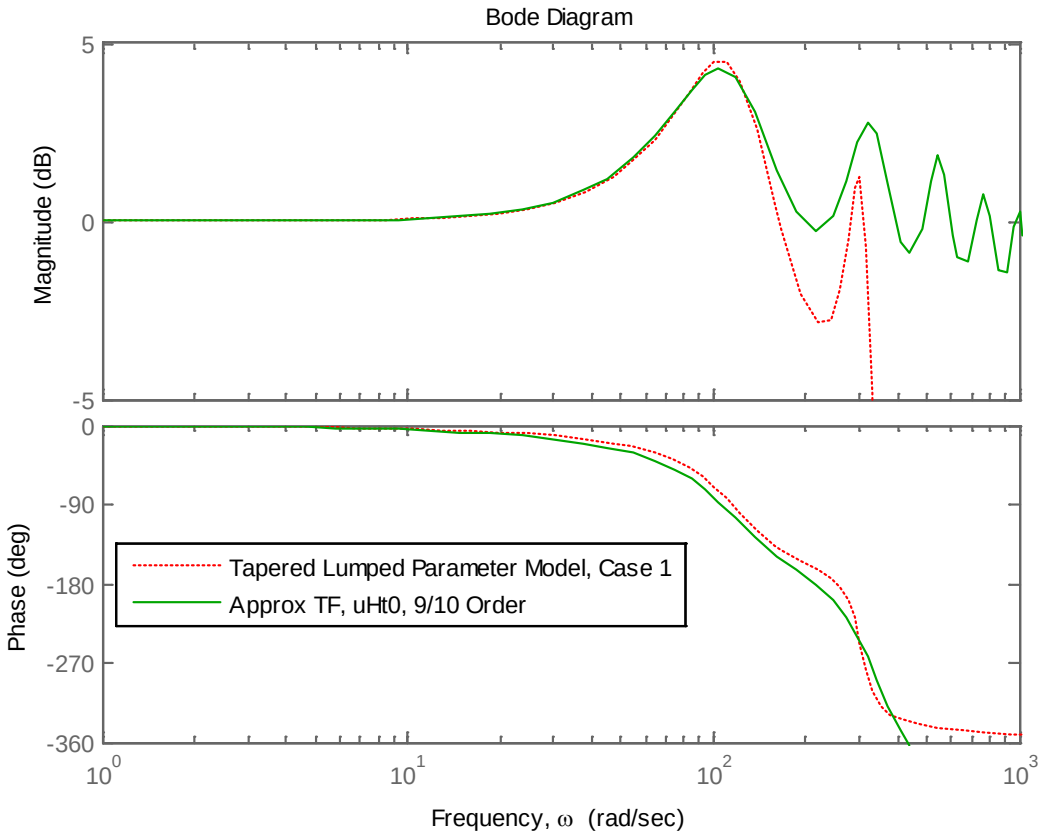


Figure 5.28 Comparison of bode plots of 4th order tapered lumped parameter model and 9th/10th order approximate transfer function model

Chapter 6

Conclusion and Future Work

The overall objective of this research has been to formulate a method for simulating the time domain pressure/flow pulsations in fluid lines for conditions of turbulent flow. The initial approach was to use a lumped parameter model for the inertance, capacitance, and resistance fluid properties where the size of the lumps was tapered so as to match the mode frequencies to a laminar flow distributed parameter model with an additional lumped resistance added to the end of the line to achieve the supposedly true steady flow resistance for turbulent flow through the line. The rationale behind this approach was based on the assumption that the inertance and capacitance properties of the fluid in a line are not dependent on the Reynolds number whereas the flow resistance is very dependent on the Reynolds number. In addition, it is a fact that equal size lumps, although many in number, will not provide either the true mode frequencies or a sufficiently accurate time domain response.

Although this approach of using a tapered lumped model achieved the correct mode frequencies if the lumps were appropriately tapered, the time domain response of the lumped model in some cases was not acceptably accurate due to the fact that the transfer function for a lumped model is zero order. That is, the transfer function has no numerator dynamics whereas an accurate transfer function approximation for a distributed parameter model may have up to (N-1) numerator terms where N is the order of the denominator.

At this point in the research, it was realized that there was no need to try to formulate a tapered lumped parameter model with an additional resistance to obtain a model for time domain simulations since the distributed parameter model with the added

resistance could be converted directly to an approximate rational polynomial transfer function that would by default match the distributed parameter model very accurately. Thus, a transfer function with the required numerator terms could be achieved which would not only provide the true mode frequencies but also an accurate time domain transient response. For special cases where the modeling and simulation dictate the use of a lumped parameter model, equations for sizing the tapered lumps were presented.

The procedure for determining the size of the lumped resistance to add to the end of the line to achieve the equivalent flow resistance for turbulence for a specific Reynolds number was presented in Chapter 4. In terms of the ratio of the added resistance to the laminar flow resistance, R_r, it is believed that large values of this ratio are not desirable as the model would tend to represent a nearly blocked line. So, results for ratios of 1, 2, and 3, which correspond to increasing values of the Reynolds number, were formulated and presented.

For larger Reynolds numbers requiring larger values of R_r, it is proposed that the line be divided into distributed parameter sections divided by lumped resistances. The rational polynomial transfer function approximation for the entire line would then be obtained from this sectioned distributed parameter model with the lumped resistances. Demonstration and development of guidelines for obtaining such models for increasing Reynolds numbers was not included in the scope of this research but definitely recommended for future studies. In addition, the technique of using the sectioned distributed parameter model with lumped resistances can be extended to compressible fluids using the explicit formula for compressible flow resistance in reference [7]; this topic is also recommended for future research. And finally, an experimental study to verify the accuracy and limitations of using the laminar flow distributed parameter model with

lumped resistance to model pressure/flow pulsations in flow through lines with turbulent flow conditions is also recommended for future work.

Appendix A

MATLAB Code for Laminar Flow Condition

```
% Series set, ICIC+R, 4th order, for laminar flow
%% Properties of liquid
clear all
format shortG
fprintf('%%%% Properties of liquid \n');
r=1/8/2*2.54e-2     % m
D=2*r;
% L=20*[1 4/5 3/5 2/5 1/5 1/10 4/50 3/50 2/50 1/50 1/100]
L=20               % m
A=pi*r^2;          % m^2
T1=27              % avg. temperature, C
% KVis=7.6179e-006; Bulk=1.8246e+009; Den=855.24; c=1460.6;

% Fluid: MIL-F-87257
% query fluid properties of stock SimHydraulics fluids
info = sh_stockfluidproperties;
name = info.f_87257.name
% info.f_87257.plot()  % plots fluid properties as a function of temperature
[KVis, Den, Bulk] = info.f_87257.prop(T1)
%      Input Units:
%         Temperature ..... C
%      Output Units:
%         Kinematic Viscosity ... m^2/s
%         Density ..... kg/m^3
%         Bulk Modulus ..... Pa
% KVis=KVis/20;      % Make the Viscosity a very small number

c=sqrt(Bulk/Den)    % Speed of sound, m/sec
DVis=KVis*Den;      % Dynamic (/Absolute) viscosity
II=Den*L/A          % Inertance
CC=A*L/Bulk         % Capacitance
RR=128*DVis*L/(pi*D^4)  % Resistance (Laminar flow)
Dn=KVis*L/(c*r^2)   % Dissipation number, dimensionless

%% Distributed parameter model (only for Laminar flow)
warning off all
syms s C1 C2
% B=2*besselj(1,j*sqrt(r^2*s/KVis))/(j*sqrt(r^2*s/KVis)*...
%     besselj(0,j*sqrt(r^2*s/KVis)));     % using s
B=2*besselj(1,j*sqrt(s))/(j*sqrt(s)*...
   besselj(0,j*sqrt(s)));      % s here mean s_bar
Zo=Den*c/(pi*r^2);  Z=Zo/sqrt(1-B);
% Dn=KVis*L/(c*r^2)
```

```
% Gamma=Dn*(r^2*s/KVis)/sqrt(1-B);       % using s
Gamma=Dn*(s)/sqrt(1-B);         % s here mean s_bar

% Transfer function to be approximated.
H=C2*RR;      % Q(s)/Ps(s)*R=Q2b(s)/U(s), (normalized Q2)/(unit step input)
H=subs(H,'C1',cosh(Gamma)/(Z*sinh(Gamma))); H=subs(H,'C2',1/(Z*sinh(Gamma)));

% Lambda_s: dimensionless root index for sinhG polynomials
Lambda_s=[];Wn_bar=[];Wn=[];Damp=[];
for m=1:6
   Lambda_s(m)=m/Dn;
   Wn_bar(m)=10^(1.0178*log10(Lambda_s(m))+0.42966);
   Wn(m)=Wn_bar(m)*KVis/r^2; % for varied m
   Damp(m)=10^( 0.0032734*log10(Lambda_s(m))^5 ...
      - 0.01333*log10(Lambda_s(m))^4 - 0.071493*log10(Lambda_s(m))^3 ...
      + 0.36569*log10(Lambda_s(m))^2 - 1.1256*log10(Lambda_s(m)) ...
      + 0.047715 );
%    W(m)=Wn(m)*(1-Damp(m)^2)^0.5;
end
Lambda_s, Wn_bar, Wn, Damp

%% Approximation curve fit TF(freq. response), numerator order = (denominator order -
1)
fprintf('%%%% Approximation curve fit TF, numerator order = (denominator order - 1) \n');
% To have at least 1 complex term, (Nord-Nreal)>1
Nord=10;     % number of denominator order
Nreal=1;    % assume minimum number of real terms
aa=Wn_bar( fix((Nord-Nreal)/2) ); bb=Wn_bar( fix((Nord-Nreal)/2)+1 ); % for s_bar
% aa=Wn( fix((Nord-Nreal)/2) ); bb=Wn( fix((Nord-Nreal)/2)+1 ); % for s
aaa=log10(aa); bbb=log10(bb);cc=10^((2*aaa+bbb)/3);
Wmin=.11; Wmax=cc

% Generate frequency data used for the curve fit and comparison plot.
% NP is the number of frequency points per decade
NP=200;
[w,wc]=genfreqs(Wmin,Wmax,NP);  % sub m-file, genfreqs.m, is used
N=length(w);
NC=length(wc);
TF=[];TFc=[];
for k=1:N
   sw=j*wc(k); % Generate values for s.
   TF(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
   TFc(k)=TF(k);
```

```
end
N1=N+1;
for k=N1:NC     % Additional frequencies for accuracy comparisons.
   sw=j*wc(k);
   TFc(k)=subs(H,s,sw);% Additional data for the accuracy comparison.
end

% Create a Transfer Function by curve fitting the frequency response
   % Get numerator & denominator
[num0,den0]=invfreqs(TF,w,Nord-1,Nord,[],100);
% Rewrite TF, make coefficient of lowest order be 1
n=length(den0);
Nnum0=num0/den0(n);
Nden0=den0/den0(n);
Ht0=tf(Nnum0,Nden0);
DCgainHt0=dcgain(Ht0)
% Adjust the dcgain to 1
Nnum0=Nnum0/DCgainHt0;
Ht0=tf(Nnum0,Nden0)
damp(Ht0)
[PR_Ht0]=pfract(Nnum0,Nden0)    % Call partial fraction function, pfract.m
DCgain_PR_Ht0=dcgain(PR_Ht0)

% Mark the peak value of Freq. Resp. of Distributed parameter model & approx TF.
   % Find damped natural frequencies, which correspond to peak values
[Wn_Ht0,Z_Ht0] = damp(Ht0);
Nw=length(Wn_Ht0);
Wd_Ht0=[];
for k=1:Nw
   if Z_Ht0(k)<1
      Wd_Ht0(k)=Wn_Ht0(k)*(1-Z_Ht0(k)^2)^0.5;
   else
      Wd_Ht0(k)=Wn_Ht0(k);
   end
end
   % Peak values of Distributed parameter model
Ht0_c=[];
for k=1:Nw
   sw=j*Wd_Ht0(k); % Generate values for s.
   Ht0_c(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
end
MTF_Wd_Ht0=20*log10(abs(Ht0_c));    % Convert to dB
   % Peak values of approx TF.
```

```
Hc0_Wd_Ht0=freqs(Nnum0,Nden0,Wd_Ht0);   % Complex frequency response
MHc0_Wd_Ht0=20*log10(abs(Hc0_Wd_Ht0));  % Convert to dB

% Generate Freq. Resp. plots to determine the accuracy of curve fit.
Hc0=freqs(Nnum0,Nden0,wc);
MHc0=20*log10(abs(Hc0));
% AHc1=angle(Hc1)*180/pi;
MTF=20*log10(abs(TFc));
% ATF=angle(TFc)*180/pi;
figure
semilogx(wc,MTF,'k',wc,MHc0,'b:',...
   Wd_Ht0,MTF_Wd_Ht0,'k+', Wd_Ht0,MHc0_Wd_Ht0,'rx');
title('Magnitude Comparison Plots');
xlabel('Normalized Frequency, \omega/\omega_v');ylabel('Decibels');
 legend('Distributed Parameter Model','Normalized s, Approx TF, Ht0, 9/10 Order',...
   'Peaks of Distributed Parameter Model','Peaks of Approx TF, Ht0',3)
% xlim([10^0, 2*10^5]);% Dn=0.0001043
% ylim([-70,5]);
% xlim([10^0,10^3]);  % Dn=0.01043
% ylim([-40,0]);
% xlim([1, 1e4]);     % Dn=0.04139 *[2/50]
% ylim([-50,5]);
% xlim([1, 1e4]);     % Dn=0.04139 *[3/50]
% ylim([-40,5]);
% xlim([1, 5e3]);     % Dn=0.04139 *[1/5, 1/10, 4/50]
% ylim([-40,5]);
% xlim([1, 1e3]);     % Dn=0.04139 *[2/5]
% ylim([-30,5]);
xlim([1, 1e3]);     % Dn=0.04139 *[1, 4/5, 3/5]
ylim([-20,5]);

   % Define damped natural freq.& peak values for real and complex
   % eigenvalues
Wd_RealEig_Ht0=[]; Wd_CompEig_Ht0=[];Peak_Distrib_0=[];Peak_Ht0=[];
k=1;
while k<length(Wd_Ht0)
   if Wd_Ht0(k)~=Wd_Ht0(k+1)
      Wd_RealEig_Ht0=[Wd_RealEig_Ht0, Wd_Ht0(k)];
      Peak_Distrib_0=[Peak_Distrib_0, MTF_Wd_Ht0(k)];
      Peak_Ht0=[Peak_Ht0, MHc0_Wd_Ht0(k)];
      k=k+1;
   else
      Wd_CompEig_Ht0=[Wd_CompEig_Ht0, Wd_Ht0(k)];
```

```
        Peak_Distrib_0=[Peak_Distrib_0, MTF_Wd_Ht0(k)];
        Peak_Ht0=[Peak_Ht0, MHc0_Wd_Ht0(k)];
        k=k+2;
    end
end
Wd_RealEig_Ht0, Wd_CompEig_Ht0, Peak_Distrib_0, Peak_Ht0

%% Approximation curve fit TF (freq. response), numerator order = 0
fprintf('%%%%% Approximation curve fit TF, numerator order = 0 \n');
% To have at least 1 complex term, (Nord-Nreal)>1
Nord=4;     % number of denominator order
Nreal=1;    % assume minimum number of real terms
aa=Wn_bar( fix((Nord-Nreal)/2) ); bb=Wn_bar( fix((Nord-Nreal)/2)+1 ); % for s_bar
% aa=Wn( fix((Nord-Nreal)/2) ); bb=Wn( fix((Nord-Nreal)/2)+1 ); % for s
aaa=log10(aa); bbb=log10(bb);cc=10^((2*aaa+bbb)/3);
Wmin=.12; Wmax=cc

% Generate frequency data used for the curve fit and comparison plot.
% NP is the number of frequency points per decade
NP=100;
[w,wc]=genfreqs(Wmin,Wmax,NP);  % sub m-file, genfreqs.m, is used
N=length(w);
NC=length(wc);
TF=[];TFc=[];
for k=1:N
    sw=j*wc(k); % Generate values for s.
    TF(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
    TFc(k)=TF(k);
end
N1=N+1;
for k=N1:NC     % Additional frequencies for accuracy comparisons.
    sw=j*wc(k);
    TFc(k)=subs(H,s,sw);% Additional data for the accuracy comparison.
end

% Create a Transfer Function by curve fitting the frequency response
    % Define a vector of weighting factors as default value 1
wt=ones(1,N);
    % Define a vector of weighting factors
for k=1:N
        % frequency range for real eigenvalue rad/sec
    if w(k)>2 && w(k)<30
        wt(k)=2;
```

```
      % frequency range for first complex eigenvalue rad/sec
   elseif (w(k)>(Wd_CompEig_Ht0(1)*0.7)) && (w(k)<(Wd_CompEig_Ht0(1)*1.3))
      wt(k)=1;
   end
end
   % Get numerator & denominator
[num1,den1]=invfreqs(TF,w,0,Nord,wt,100);
% Rewrite TF, make coefficient of lowest order be 1
n=length(den1);
Nnum1=num1/den1(n);
Nden1=den1/den1(n);
Ht1=tf(Nnum1,Nden1);
DCgainHt1=dcgain(Ht1)
% Adjust the DCgain to 1
Nnum1=Nnum1/DCgainHt1;
Ht1=tf(Nnum1,Nden1)
damp(Ht1)
[PR_Ht1]=pfract(Nnum1,Nden1)    % Call partial fraction function, pfract.m
DCgain_PR_Ht1=dcgain(PR_Ht1)

% Mark the peak values of Freq. Resp. of Distributed parameter model & Approx TF.
[Wn_Ht1,Z_Ht1] = damp(Ht1);
Nw=length(Wn_Ht1);
   % Find damped natural frequencies, which correspond to peak values
for k=1:Nw
   if Z_Ht1(k)<1
      Wd_Ht1(k)=Wn_Ht1(k)*(1-Z_Ht1(k)^2)^0.5;
   else
      Wd_Ht1(k)=Wn_Ht1(k);
   end
end
   % Peak values of Distributed parameter model
Ht1_c=[];
for k=1:Nw
   sw=j*Wd_Ht1(k); % Generate values for s.
   Ht1_c(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
end
MTF_Wd_Ht1=20*log10(abs(Ht1_c));    % Convert to dB
   % Peak values of Normalized s, Approx TF.
Hc1_Wd_Ht1=freqs(Nnum1,Nden1,Wd_Ht1);   % Complex frequency response
MHc1_Wd_Ht1=20*log10(abs(Hc1_Wd_Ht1));  % Convert to dB

% Generate Freq. Resp. plots to determine the accuracy of curve fit.
```

```
Hc1=freqs(Nnum1,Nden1,wc);
MHc1=20*log10(abs(Hc1));
% AHc1=angle(Hc1)*180/pi;
MTF=20*log10(abs(TFc));
% ATF=angle(TFc)*180/pi;
figure
semilogx(wc,MTF,'k',wc,MHc1,'b:',...
   Wd_Ht1,MTF_Wd_Ht1,'k+', Wd_Ht1,MHc1_Wd_Ht1,'rx');
title('Magnitude Comparison Plots');
xlabel('Normalized Frequency, \omega/\omega_v');ylabel('Decibels');
legend('Distributed Parameter Model','Normalized s, Approx TF, Ht1, 0/4 Order',...
   'Peaks of Distributed Parameter Model','Peaks of Approx TF, Ht1',3)
% xlim([10^-2, 10^6]);% Dn=0.0001043
% ylim([-120,0]);
% xlim([10^0,10^3]);  % Dn=0.01043
% ylim([-70,-10]);
% xlim([1, 1e4]);     % Dn=0.04139 *[2/50]
% ylim([-50,5]);
% xlim([1, 1e4]);     % Dn=0.04139 *[3/50]
% ylim([-40,5]);
% xlim([1, 5e3]);     % Dn=0.04139 *[1/5, 1/10, 4/50]
% ylim([-40,5]);
% xlim([1, 1e3]);     % Dn=0.04139 *[2/5]
% ylim([-30,5]);
xlim([1, 1e3]);     % Dn=0.04139 *[1, 4/5, 3/5]
ylim([-20,5]);

   % Define damped natural freq.& peak values for real and complex
   % eigenvalues
Wd_RealEig_Ht1=[]; Wd_CompEig_Ht1=[];Peak_Distrib_1=[];Peak_Ht1=[];
k=1;
while k<length(Wd_Ht1)
   if Wd_Ht1(k)~=Wd_Ht1(k+1)
      Wd_RealEig_Ht1=[Wd_RealEig_Ht1, Wd_Ht1(k)];
      Peak_Distrib_1=[Peak_Distrib_1, MTF_Wd_Ht1(k)];
      Peak_Ht1=[Peak_Ht1, MHc1_Wd_Ht1(k)];
      k=k+1;
   else
      Wd_CompEig_Ht1=[Wd_CompEig_Ht1, Wd_Ht1(k)];
      Peak_Distrib_1=[Peak_Distrib_1, MTF_Wd_Ht1(k)];
      Peak_Ht1=[Peak_Ht1, MHc1_Wd_Ht1(k)];
      k=k+2;
   end
```

```
end
Wd_RealEig_Ht1, Wd_CompEig_Ht1, Peak_Distrib_1, Peak_Ht1

%% Comparison of step response of 9th/10th order and 0/4th order transfer function
approximation
figure, step(Ht0,'g',Ht1,'b-.')
legend('Normalized s, Approx TF, Ht0, 9/10 Order',...
    'Normalized s, Approx TF, Ht1, 0/4 Order', 4)
% title('Step Response')
xlabel('Normalized Time, \omega_V t');
ylabel('Normalized Volumetric Flow Rate, \DeltaQ_b/Q_e ');

%% Un-Normalized TF, convert above s_bar to s, s_bar=s*(r^2/KVis)
fprintf('%%%% Un-Normalized TF, convert above s_bar to s, s_bar=s*(r^2/KVis) \n');
ord=length(Nnum0);
uNnum0=[];uNden0=[];
for a=ord:-1:1
    uNnum0(a)=Nnum0(a)*(r^2/KVis)^(ord-a);
end
ord=length(Nden0);
for a=ord:-1:1
    uNden0(a)=Nden0(a)*(r^2/KVis)^(ord-a);
end
uHt0=tf(uNnum0,uNden0)
damp(uHt0)
[PR_uHt0]=pfract(uNnum0,uNden0)
DCgain_PR_uHt0=dcgain(PR_uHt0)

ord=length(Nnum1);
uNnum1=[];uNden1=[];
for a=ord:-1:1
    uNnum1(a)=Nnum1(a)*(r^2/KVis)^(ord-a);
end
ord=length(Nden1);
for a=ord:-1:1
    uNden1(a)=Nden1(a)*(r^2/KVis)^(ord-a);
end
uHt1=tf(uNnum1,uNden1)
damp(uHt1)
[PR_uHt1]=pfract(uNnum1,uNden1)
DCgain_PR_uHt1=dcgain(PR_uHt1)

%% Equal Distributed Series set: ICIC+R, 4th order
```

```
fprintf('%%%% Equal Distributed Series set: ICIC+R, 4th order \n');
syms s Ps I0 I1 C1 C2 R1
% let I0=Is*i0; I1=Is*i1;
% C1=Cs*c1; C2=Cs*c2;
% R1=Rs*1;
H1=solve('Ps-P1=I0*s*Q0','Q0-Q1=C1*s*P1',...
   'P1-P2=I1*s*Q1','Q1-Q2=C2*s*P2',...
   'P2=R1*Q2','P1,P2,Q0,Q1,Q2');
% Q2(s)/(Ps(s)/R)=Q2b(s)/U(s), (normalized Q2)/(unit step input)
h1=collect(H1.Q2/Ps*(R1),s); % TF=Q/(Ps/R)
pretty(h1)

% Equal distributed parameter
digits(5);
h11=subs(h1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RR]);     % laminar
%     % Use Resistance of turbulent, RRt
% h11=subs(h1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RRt]);    % turbulent
h11=vpa(h11);
pretty(h11)

[numh1,denh1]=numden(h1);
[CoeNum1,SS1]=coeffs(numh1,s);  % coefficient from high to low order
[CoeDen1,SS2]=coeffs(denh1,s);  % coefficient from high to low order

% let coefficient of lowest order be 1
n=length(CoeDen1);
NCoeNum1=CoeNum1/CoeDen1(n);
NCoeDen1=CoeDen1/CoeDen1(n);

% Assume this is a equal distributed parameter model
NumLump1=subs(NCoeNum1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RR]);
DenLump1=subs(NCoeDen1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RR]);
%     % Use Resistance of turbulent, RRt
% NumLump1=subs(NCoeNum1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RRt]);
% DenLump1=subs(NCoeDen1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RRt]);

TFLump1=tf(NumLump1,DenLump1)
% eig(TFLump1)
damp(TFLump1)
[PF_TFLump1]=pfract(NumLump1,DenLump1)
DCgain_PF_TFLump1=dcgain(PF_TFLump1)

figure, step(TFLump1,':r',uHt0,'g',uHt1,'-.b')
```

```
% title('Step Response')
ylabel('Normalized Volumetric Flow Rate, \DeltaQ_b/Q_e ');
legend('Equal Lumped Parameter Model for ICIC+R',...
    'Approx TF, uHt0, 9/10 Order',...
    'Approx TF, uHt1, 0/4 Order', 4)

figure, uH=bodeplot(TFLump1,':r',uHt0,'g',uHt1,'-.b');
p=getoptions(uH);    % Create a plot options handle p.
p.PhaseMatching = 'on';
% p.Xlim{1} = [1e-2 5e2];   % Dn=0.00010431
% p.Ylim{1} = [-80 0];
% p.Xlim{1} = [1 2e3];     % Dn=0.010431
% p.Ylim{1} = [-40 0];
% p.Xlim{1} = [1 5e2];     % Dn=0.10431
% p.Ylim{1} = [-15 6];
% p.Xlim{1} = [1 5e4];      % Dn=0.04139 *[2/50]
% p.Ylim{1} = [-50 5];
% p.Ylim{2} = [-720 0];
% p.Xlim{1} = [1 1e4];      % Dn=0.04139 *[1/10, 4/50, 3/50]
% p.Ylim{1} = [-40 5];
% p.Ylim{2} = [-720 0];
% p.Xlim{1} = [1 6e3];      % Dn=0.04139 *[1/5]
% p.Ylim{1} = [-40 5];
% p.Ylim{2} = [-720 0];
% p.Xlim{1} = [1 2e3];       % Dn=0.04139 *[2/5]
% p.Ylim{1} = [-30 5];
% p.Ylim{2} = [-720 0];
p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[1, 4/5, 3/5]
p.Ylim{1} = [-20 5];
p.Ylim{2} = [-720 0];

setoptions(uH,p);    % Apply plot options to the Bode plot and render.
legend('Equal Lumped Parameter Model for ICIC+R',...
    'Approx TF, uHt0, 9/10 Order',...
    'Approx TF, uHt1, 0/4 Order', 'Location','Best')

%% Paynter's lumped parameter model
fprintf('%%%% Tapered lumped parameter model: ICI+R, 3th order \n');
syms s Ps I0 I1 C1 R1
% let I0=Is*i0; I1=Is*i1;
% C1=Cs*c1; C2=Cs*c2;
% R1=Rs*1;
H3=solve('Ps-P1=I0*s*Q0','Q0-Q1=C1*s*P1',...
```

```
    'P1-P2=I1*s*Q1',...
    'P2=R1*Q1','P1,P2,Q0,Q1');
% Q2(s)/(Ps(s)/R)=Q2b(s)/U(s), (normalized Q2)/(unit step input)
h3=collect(H3.Q1/Ps*(R1),s); % TF=Q/(Ps/R)

% Paynter's lumped parameter
[numh3,denh3]=numden(h3);
[CoeNum3,SS1]=coeffs(numh3,s);  % coefficient from high to low order
[CoeDen3,SS2]=coeffs(denh3,s);  % coefficient from high to low order

% let coefficient of lowest order be 1
n=length(CoeDen3);
NCoeNum3=CoeNum3/CoeDen3(n);
NCoeDen3=CoeDen3/CoeDen3(n);

NumLump3=subs(NCoeNum3,[I0,I1,C1,C2,R1],[II*.25,II*.75,CC*.541,0,RR]);
DenLump3=subs(NCoeDen3,[I0,I1,C1,C2,R1],[II*.25,II*.75,CC*.541,0,RR]);

TFLump3=tf(NumLump3,DenLump3)
% eig(TFLump1)
damp(TFLump3)
[PF_TFLump3]=pfract(NumLump3,DenLump3)
DCgain_PF_TFLump3=dcgain(PF_TFLump3)

figure, step(TFLump1,':r',TFLump3,'--k',uHt0,'g',uHt1,'-.b', .35)
% title('Step Response')
ylabel('Normalized Volumetric Flow Rate, \DeltaQ_b/Q_e ');
legend('Equal Lumped Parameter Model for ICIC+R',...
    'Paynter Tapered Lumped Parameter Model for ICI+R',...
    'Approx TF, uHt0, 9/10 Order',...
    'Approx TF, uHt1, 0/4 Order', 4)

figure, uH=bodeplot(TFLump1,':r',TFLump3,'--k',uHt0,'g',uHt1,'-.b');
p=getoptions(uH);    % Create a plot options handle p.
p.PhaseMatching = 'on';
% p.Xlim{1} = [1e-2 5e2];   % Dn=0.00010431
% p.Ylim{1} = [-80 0];
% p.Xlim{1} = [1 2e3];     % Dn=0.010431
% p.Ylim{1} = [-40 0];
% p.Xlim{1} = [1 5e2];     % Dn=0.10431
% p.Ylim{1} = [-15 6];
% p.Xlim{1} = [1 5e4];      % Dn=0.04139 *[2/50]
% p.Ylim{1} = [-50 5];
```

```
% p.Ylim{2} = [-720 0];
% p.Xlim{1} = [1 1e4];      % Dn=0.04139 *[1/10, 4/50, 3/50]
% p.Ylim{1} = [-40 5];
% p.Ylim{2} = [-720 0];
% p.Xlim{1} = [1 6e3];      % Dn=0.04139 *[1/5]
% p.Ylim{1} = [-40 5];
% p.Ylim{2} = [-720 0];
% p.Xlim{1} = [1 2e3];       % Dn=0.04139 *[2/5]
% p.Ylim{1} = [-30 5];
% p.Ylim{2} = [-720 0];
p.Xlim{1} = [1e1 1e3];       % Dn=0.04139 *[1, 4/5, 3/5]
p.Ylim{1} = [-15 5];
p.Ylim{2} = [-360 0];

setoptions(uH,p);    % Apply plot options to the Bode plot and render.
legend('Equal Lumped Parameter Model for ICIC+R',...
   'Paynter Tapered Lumped Parameter Model for ICI+R',...
   'Approx TF, uHt0, 9/10 Order',...
   'Approx TF, uHt1, 0/4 Order', 'Location','Best')

%% Solve symbolic equations of series set lumped parameter model
fprintf('%%%% Solve symbolic equations of series set lumped parameter model \n');
warning on
syms s Ps i0 i1 c1 c2 r1 Is Cs Rs
% let I0=Is*i0; I1=Is*i1;
% C1=Cs*c1; C2=Cs*c2;
% R1=Rs*1;
H2=solve('Ps-P1=Is*i0*s*Q0','Q0-Q1=Cs*c1*s*P1',...
   'P1-P2=Is*i1*s*Q1','Q1-Q2=Cs*c2*s*P2',...
   'P2=Rs*1*Q2','P1,P2,Q0,Q1,Q2');
% Q(s)/Ps(s)*R=Qb(s)/U(s), (normalized Q)/(unit step input)
h2=collect(H2.Q2/(Ps/Rs),s); % TF=Q/(Ps/R)
pretty(h2)

[numh2,denh2]=numden(h2);
[CoeNum2,SS1]=coeffs(numh2,s);  % coefficient from high to low order
[CoeDen2,SS2]=coeffs(denh2,s);  % coefficient from high to low order

% let coefficient of lowest order be 1
n=length(CoeDen2);
NCoeNum2=CoeNum2/CoeDen2(n);
NCoeDen2=CoeDen2/CoeDen2(n);
```

```
% 0/4th order approximation numerical TF
uHt1=tf(uNnum1,uNden1)

F=solve(NCoeDen2(1)-uNden1(1), NCoeDen2(2)-uNden1(2),...
    NCoeDen2(3)-uNden1(3), NCoeDen2(4)-uNden1(4),...
    'i0,i1,c1,c2');

% the weight of each parameter
wIs=[F.i0, F.i1];
wCs=[F.c1, F.c2];
wI=subs(vpa(wIs),{Is,Cs,Rs},{II,CC,RR})
wC=subs(vpa(wCs),{Is,Cs,Rs},{II,CC,RR})
.
```

Appendix B

MATLAB Code for Turbulent Flow Condition

```
% Series set, ICIC+R+Rv(approx R, using slop), 4th order, turbulent flow
% Varied Dn(=<0.04139) with Rn=10544
%% Properties of liquid
clear all
% format shortG
fprintf('%%%% Properties of liquid \n');
% D=1/8 *(2.54e-2) *[1,1.118,1.291,1.5811,2.2361,3.1623,3.5355,4.0825,5,7.0711,10]
D=1/8 *(2.54e-2)    % diameter, m
r=D/2;              % radius, m
% L=20*[1 4/5 3/5 2/5 1/5 1/10 4/50 3/50 2/50 1/50 1/100]
L=20                % m
A=pi*r^2;           % m^2
T1=27               % avg. temperature, C
% KVis=7.6179e-006; c=1460.6; Den=855.24;

% Fluid: MIL-F-87257
% query fluid properties of stock SimHydraulics fluids
info = sh_stockfluidproperties;
name = info.f_87257.name
% info.f_87257.plot()  % plots fluid properties as a function of temperature
[KVis, Den, Bulk] = info.f_87257.prop(T1)
%     Input Units:
%        Temperature ..... C
%     Output Units:
%        Kinematic Viscosity ... m^2/s
%        Density ..... kg/m^3
%        Bulk Modulus ..... Pa
% KVis=KVis/20;      % Make the Viscosity a very small number

c=sqrt(Bulk/Den)    % Speed of sound, m/sec
DVis=KVis*Den;      % Dynamic (/Absolute) viscosity
II=Den*L/A          % Inertance
CC=A*L/Bulk         % Capacitance
RR=128*DVis*L/(pi*D^4)  % Resistance (Laminar flow)
% For Turbulent flow (Rn > 4000)
Rn=10544.37             % Reynolds number
% Rn=558,602             % Reynolds number, Rv=100*RR
Qr=Rn*KVis*A/D;     % Flow rate for Rn
   % Resistance equation of turbulent flow; P=Cf*Q^1.75
Cf=(0.2414*Den^0.75*DVis^0.25*L/D^4.75);
fQr=Cf*Qr^1.75;     % P=f(Qr)+df(Qr)*(Q-Qr)
dfQr=1.75*Cf*Qr^0.75;
RRt=dfQr            % Resistance (Turbulent flow)
```

```
RRv=RRt-RR          % Resistance (added valve)
Dn=KVis*L/(c*r^2)   % Dissipation number, dimensionless,
                    % *[1,4/5,3/5,2/5,1/5,1/10,4/50,3/50,2/50,1/50,1/100]

Rr=RRv/RR           % Define the ratio as Rr

%% Distributed parameter model (only for Laminar flow)
warning off all
syms s C1 C2
% B=2*besselj(1,j*sqrt(r^2*s/KVis))/(j*sqrt(r^2*s/KVis)*...
%     besselj(0,j*sqrt(r^2*s/KVis)));     % using s
B=2*besselj(1,j*sqrt(s))/(j*sqrt(s)*...
   besselj(0,j*sqrt(s)));     % s here mean s_bar
Zo=Den*c/(pi*r^2);  Z=Zo/sqrt(1-B);
% Dn=KVis*L/(c*r^2)
% Gamma=Dn*(r^2*s/KVis)/sqrt(1-B);       % using s
Gamma=Dn*(s)/sqrt(1-B);       % s here mean s_bar

% Transfer function to be approximated.
H=C2/(1+C1*RRv)*(RR+RRv);   % Q(s)/Ps(s)*R=Q2b(s)/U(s), (normalized Q2)/(unit step
input)
H=subs(H,'C1',cosh(Gamma)/(Z*sinh(Gamma))); H=subs(H,'C2',1/(Z*sinh(Gamma)));

% Lambda_s: dimensionless root index for sinhG polynomials
Lambda_s=[];Wn_bar=[];Wn=[];Damp=[];
for m=1:6
   Lambda_s(m)=m/Dn;
   Wn_bar(m)=10^(1.0178*log10(Lambda_s(m))+0.42966);
%     Wn(m)=Wn_bar(m)*KVis/m/r^2; % for varied Dn
   Wn(m)=Wn_bar(m)*KVis/r^2; % for varied m
   Damp(m)=10^( 0.0032734*log10(Lambda_s(m))^5 ...
      - 0.01333*log10(Lambda_s(m))^4 - 0.071493*log10(Lambda_s(m))^3 ...
      + 0.36569*log10(Lambda_s(m))^2 - 1.1256*log10(Lambda_s(m)) ...
      + 0.047715 );
%     W(m)=Wn(m)*(1-Damp(m)^2)^0.5;
end
Lambda_s, Wn_bar, Wn, Damp

%% Approximation curve fit TF, numerator order = (denominator order - 1)
fprintf('%%%% Approximation curve fit TF, numerator order = (denominator order - 1) \n');
% To have at least 1 complex term, (Nord-Nreal)>1
Nord=10;     % number of denominator order
Wmin=1.1; Wmax=Wn_bar(Nord/2)    % Dn=0.04139 *[1]
```

```
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*.97    % Dn=0.04139 *[4/5]
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*.97    % Dn=0.04139 *[3/5]
% Wmin=1.1; Wmax=738 % Wn_bar(Nord/2-1)*1.00    % Dn=0.04139 *[2/5]
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*1.00    % Dn=0.04139 *[1/5]
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*1.00    % Dn=0.04139 *[1/10]
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*1.00    % Dn=0.04139 *[4/50]
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*1.00    % Dn=0.04139 *[3/50]
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*1.03    % Dn=0.04139 *[2/50]
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*1.03    % Dn=0.04139 *[1/50]
% Wmin=1.1; Wmax=Wn_bar(Nord/2-1)*1.03    % Dn=0.04139 *[1/100]

% Generate frequency data used for the curve fit and comparison plot.
% NP is the number of frequency points per decade
NP=200; % Dn=0.04139 *[1,3/5,2/5,1/5,1/10,4/50,3/50]
% NP=300; % Dn=0.04139 *[2/50,1/50,1/100]
[w,wc]=genfreqs(Wmin,Wmax,NP);  % sub m-file, genfreqs.m, is used
N=length(w);
NC=length(wc);
TF=[];TFc=[];
for k=1:N
   sw=1i*wc(k); % Generate values for s.
   TF(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
   TFc(k)=TF(k);
end
N1=N+1;
for k=N1:NC     % Additional frequencies for accuracy comparisons.
   sw=1i*wc(k);
   TFc(k)=subs(H,s,sw);% Additional data for the accuracy comparison.
end

% Create a Transfer Function by curve fitting the frequency response
   % Get numerator & denominator
[num0,den0]=invfreqs(TF,w,Nord-1,Nord,[],100);
% Rewrite TF, make coefficient of lowest order be 1
n=length(den0);
Nnum0=num0/den0(n);
Nden0=den0/den0(n);
Ht0=tf(Nnum0,Nden0);
DCgainHt0=dcgain(Ht0)
% Adjust the dcgain to 1
Nnum0=Nnum0/DCgainHt0;
Ht0=tf(Nnum0,Nden0)
damp(Ht0)
```

```
[PR_Ht0]=pfract(Nnum0,Nden0)    % Call partial fraction function, pfract.m
DCgain_PR_Ht0=dcgain(PR_Ht0)

% Mark the peak value of Freq. Resp. of Distributed parameter model & approx TF.
   % Find damped natural frequencies, which correspond to peak values
[Wn_Ht0,Z_Ht0] = damp(Ht0);
Nw=length(Wn_Ht0);
Wd_Ht0=[];
for k=1:Nw
   if Z_Ht0(k)<1
      Wd_Ht0(k)=Wn_Ht0(k)*(1-Z_Ht0(k)^2)^0.5;
   else
      Wd_Ht0(k)=Wn_Ht0(k);
   end
end
   % Peak values of Distributed parameter model
Ht0_c=[];
for k=1:Nw
   sw=1i*Wd_Ht0(k); % Generate values for s.
   Ht0_c(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
end
MTF_Wd_Ht0=20*log10(abs(Ht0_c));    % Convert to dB
   % Peak values of 9/10 order approx TF.
Hc0_Wd_Ht0=freqs(Nnum0,Nden0,Wd_Ht0);   % Complex frequency response
MHc0_Wd_Ht0=20*log10(abs(Hc0_Wd_Ht0));  % Convert to dB
%     % Peak values of 0/10 order approx TF.
% Hc02_Wd_Ht0=freqs(1,Nden0,Wd_Ht0);   % Complex frequency response
% MHc02_Wd_Ht0=20*log10(abs(Hc02_Wd_Ht0));  % Convert to dB

% Generate Freq. Resp. plots to determine the accuracy of curve fit.
   % Ht0, 9/10 order approximate TF
Hc0=freqs(Nnum0,Nden0,wc);
MHc0=20*log10(abs(Hc0));
% AHc1=angle(Hc1)*180/pi;
%     % Ht02, 0/10 order approximate TF
% Hc02=freqs(1,Nden0,wc);
% MHc02=20*log10(abs(Hc02));
   % distributed model
MTF=20*log10(abs(TFc));
% ATF=angle(TFc)*180/pi;
figure
semilogx(wc,MTF,'k',wc,MHc0,'b:',...
   Wd_Ht0,MTF_Wd_Ht0,'k+', Wd_Ht0,MHc0_Wd_Ht0,'rx');
```

```
title('Magnitude Comparison Plots');
xlabel('Normalized Frequency, \omega/\omega_v');ylabel('Decibels');
legend('Distributed Parameter Model','Normalized s, Approx TF, Ht0, 9/10 Order',...
   'Peaks of Distributed Parameter Model','Peaks of Approx TF, Ht0',3)
xlim([1, 1e3]); % Dn=0.04139 *[1]
ylim([-3,2]);
% xlim([1, 5e2]); % Dn=0.04139 *[4/5],
% ylim([-2.5,0.5]);
% xlim([1, 1e3]); % Dn=0.04139 *[3/5],
% ylim([-3.5,0]);
% xlim([1, 1e3]); % Dn=0.04139 *[2/5],
% ylim([-6,0]);
% xlim([1, 2e3]); % Dn=0.04139 *[1/5],
% ylim([-11,0]);
% xlim([1, 4e3]); % Dn=0.04139 *[1/10],
% ylim([-17,0]);
% xlim([1, 5e3]); % Dn=0.04139 *[4/50],
% ylim([-20,0]);
% xlim([1, 1e4]); % Dn=0.04139 *[3/50],
% ylim([-25,0]);
% xlim([1, 1e4]); % Dn=0.04139 *[2/50],
% ylim([-30,0]);
% xlim([1, 3e4]); % Dn=0.04139 *[1/50]
% ylim([-35,0]);
% xlim([1, 5e4]); % Dn=0.04139 *[1/100],
% ylim([-40,0]);

   % Damped natural freq. of real and complex eigenvalues
   % Peak values of distributed model and approx TF
Wd_RealEig_Ht0=[]; Wd_CompEig_Ht0=[];Peak_Distrib_0=[];Peak_Ht0=[];
k1=1;
while k1<length(Wd_Ht0)
   if Wd_Ht0(k1)~=Wd_Ht0(k1+1)
      Wd_RealEig_Ht0=[Wd_RealEig_Ht0, Wd_Ht0(k1)];
      Peak_Ht0=[Peak_Ht0, MHc0_Wd_Ht0(k1)];
      k1=k1+1;
   else
      Wd_CompEig_Ht0=[Wd_CompEig_Ht0, Wd_Ht0(k1)];
      Peak_Ht0=[Peak_Ht0, MHc0_Wd_Ht0(k1)];
      k1=k1+2;
   end
end
if k1==length(Wd_Ht0)
```

```
    Wd_RealEig_Ht0=[Wd_RealEig_Ht0, Wd_Ht0(k1)];
    Peak_Ht0=[Peak_Ht0, MHc0_Wd_Ht0(k1)];
end
k2=1;
while k2<length(MTF_Wd_Ht0)
    if MTF_Wd_Ht0(k2)~=MTF_Wd_Ht0(k2+1)
        Peak_Distrib_0=[Peak_Distrib_0, MTF_Wd_Ht0(k2)];
        k2=k2+1;
    else
        Peak_Distrib_0=[Peak_Distrib_0, MTF_Wd_Ht0(k2)];
        k2=k2+2;
    end
end
if k2==length(MTF_Wd_Ht0)
    Peak_Distrib_0=[Peak_Distrib_0, MTF_Wd_Ht0(k2)];
end

Wd_RealEig_Ht0, Wd_CompEig_Ht0, Peak_Distrib_0, Peak_Ht0
Peak_difference0=(Peak_Ht0-Peak_Distrib_0)

%% Approximation curve fit TF, numerator order = 0
fprintf('%%%% Approximation curve fit TF, numerator order = 0 \n');
% To have at least 1 complex term, (Nord-Nreal)>1
Nord=4;     % number of denominator order
% Nreal=0;    % assume minimum number of real terms
% aa=Wd_CompEig_Ht0( fix((Nord-Nreal)/2) );   % for s_bar
% bb=Wd_CompEig_Ht0( fix((Nord-Nreal)/2)+1 ); % for s_bar
% aaa=log10(aa); bbb=log10(bb);cc=10^((4*aaa+bbb)/5);
% Wmin=1.1; Wmax=cc       % Dn=0.04139 *[1]
Wmin=1.1; Wmax=73   % Wd_CompEig_Ht0(Nord/2-1)*1.9       % Dn=0.04139 *[1]
% Wmin=1.1; Wmax=60       % Dn=0.04139 *[4/5]
% Wmin=1.1; Wmax=110      % Dn=0.04139 *[3/5]
% Wmin=1.1; Wmax=149      % Dn=0.04139 *[2/5]
% Wmin=1.1; Wmax= Wd_CompEig_Ht0(Nord/2-1)*1      % Dn=0.04139 *[1/5]
% Wmin=1.1; Wmax=802 % Wd_CompEig_Ht0(Nord/2-1)*1      % Dn=0.04139 *[1/10]
% Wmin=1.1; Wmax=1040 % Wd_CompEig_Ht0(Nord/2-1)*1.1      % Dn=0.04139 *[4/50]
% Wmin=1.1; Wmax=Wd_CompEig_Ht0(Nord/2-1)*1.1      % Dn=0.04139 *[3/50]
% Wmin=1.1; Wmax=2480 % Wd_CompEig_Ht0(Nord/2-1)*1.3      % Dn=0.04139 *[2/50]
% Wmin=1.1; Wmax=Wd_CompEig_Ht0(Nord/2-1)*1.3      % Dn=0.04139 *[1/50]
% Wmin=1.1; Wmax=Wd_CompEig_Ht0(Nord/2-1)*1.3      % Dn=0.04139 *[1/100]

% 4th order for the first mode
% Wmin=1.1; Wmax=73 % Wd_CompEig_Ht0(Nord/2-1)*1.9    % Dn=0.04139,
```

```
% % 4th order for first 2 modes
% Wmin=1.1; Wmax=Wn_bar(Nord/2)-10    % Dn=0.04139 *[1]

% Generate frequency data used for the curve fit and comparison plot.
% NP is the number of frequency points per decade
NP=200;
[w,wc]=genfreqs(Wmin,Wmax,NP);  % sub m-file, genfreqs.m, is used
N=length(w);
NC=length(wc);
TF=[];TFc=[];
for k=1:N
   sw=1i*wc(k); % Generate values for s.
   TF(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
   TFc(k)=TF(k);
end
N1=N+1;
for k=N1:NC     % Additional frequencies for accuracy comparisons.
   sw=1i*wc(k);
   TFc(k)=subs(H,s,sw);% Additional data for the accuracy comparison.
end

%%
% Create a Transfer Function by curve fitting the frequency response
   % Define a vector of weighting factors as default value 1
wt=ones(1,N);
   % Define a vector of weighting factors
for k=1:N
   % frequency range for real eigenvalue rad/sec
   if w(k)>10 && w(k)<30  % Dn=0.04139 *[1,4/5]
%     if w(k)>20 && w(k)<50  % Dn=0.04139 *[3/5,2/5,1/5,1/10,4/50,3/50,1/50]
%     if w(k)>20 && w(k)<100  % Dn=0.04139 *[2/50,1/100]
%        wt(k)=1;   % Dn=0.04139 *[3/5,2/5,1/5,1/10,4/50,3/50,1/50,1/100]
%        wt(k)=2;   % Dn=0.04139 *[2/50]
      wt(k)=1;   % Dn=0.04139 *[1]

   % frequency range for first complex eigenvalue rad/sec
   elseif (w(k)>(Wd_CompEig_Ht0(1)*0.9)) && (w(k)<(Wd_CompEig_Ht0(1)*1.2)) %
Dn=0.04139 *[1,4/5]
%     elseif (w(k)>(Wd_CompEig_Ht0(1)*0.9)) && (w(k)<(Wd_CompEig_Ht0(1)*1.2)) %
Dn=0.04139 *[3/5,2/5,1/5,1/10,4/50,3/50]
%     elseif (w(k)>(Wd_CompEig_Ht0(1)*0.9)) && (w(k)<(Wd_CompEig_Ht0(1)*1.4)) %
Dn=0.04139 *[2/50,1/50,1/100]
%        wt(k)=1;    % Dn=0.04139 *[3/5,2/5,1/10,4/50,3/50,1/100]
```

```
%        wt(k)=2;    % Dn=0.04139 *[1/5]
%        wt(k)=6;    % Dn=0.04139 *[2/50]
%        wt(k)=3;    % Dn=0.04139 *[1/50]
%        wt(k)=3;    % Dn=0.04139 *[4/5]
       wt(k)=1;    % Dn=0.04139 *[1]

   % frequency range for second complex eigenvalue rad/sec
   elseif (w(k)>(Wd_CompEig_Ht0(2)*0.9)) && (w(k)<(Wd_CompEig_Ht0(2)*1.4))
%        wt(k)=10;    % Dn=0.04139 *[1]
   end
end
   % Get numerator & denominator
[num1,den1]=invfreqs(TF,w,0,Nord,wt,100);
% Rewrite TF, make coefficient of lowest order be 1
n=length(den1);
Nnum1=num1/den1(n);
Nden1=den1/den1(n);
Ht1=tf(Nnum1,Nden1);
DCgainHt1=dcgain(Ht1)
% Adjust the DCgain to 1
Nnum1=Nnum1/DCgainHt1;
Ht1=tf(Nnum1,Nden1)
damp(Ht1)
[PR_Ht1]=pfract(Nnum1,Nden1)    % Call partial fraction function, pfract.m
DCgain_PR_Ht1=dcgain(PR_Ht1)

% Mark the peak values of Freq. Resp. of Distributed parameter model & Approx TF.
[Wn_Ht1,Z_Ht1] = damp(Ht1);
Nw=length(Wn_Ht1);
   % Find damped natural frequencies, which correspond to peak values
Wd_Ht1=[];
for k=1:Nw
   if Z_Ht1(k)<1
      Wd_Ht1(k)=Wn_Ht1(k)*(1-Z_Ht1(k)^2)^0.5;
   else
      Wd_Ht1(k)=Wn_Ht1(k);
   end
end
% Wd_Ht1
   % Peak values of Distributed parameter model
Ht1_c=[];
for k=1:Nw
   sw=1i*Wd_Ht0(k); % Generate values for s.
```

```
   Ht1_c(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
end
MTF_Wd_Ht1=20*log10(abs(Ht1_c));    % Convert to dB
   % Peak values of approx TF.
Hc1_Wd_Ht1=freqs(Nnum1,Nden1,Wd_Ht1);   % Complex frequency response
MHc1_Wd_Ht1=20*log10(abs(Hc1_Wd_Ht1));  % Convert to dB

% Generate Freq. Resp. plots to determine the accuracy of curve fit.
Hc1=freqs(Nnum1,Nden1,wc);
MHc1=20*log10(abs(Hc1));
% AHc1=angle(Hc1)*180/pi;
MTF=20*log10(abs(TFc));
% ATF=angle(TFc)*180/pi;
figure
semilogx(wc,MTF,'k',wc,MHc1,'b:',...
   Wd_Ht0(1:4),MTF_Wd_Ht1,'k+', Wd_Ht1,MHc1_Wd_Ht1,'rx');

title('Magnitude Comparison Plots');
xlabel('Normalized Frequency, \omega/\omega_v');ylabel('Decibels');
legend('Distributed Parameter Model','Normalized s, Approx TF, Ht1, 0/4 Order',...
   'Peaks of Distributed Parameter Model','Peaks of Approx TF, Ht1',3)
xlim([1, 1e3]); % Dn=0.04139 *[1],
ylim([-3,2]);
% xlim([1, 2e2]); % Dn=0.04139 *[4/5],
% ylim([-1,1]);
% xlim([1, 1e3]); % Dn=0.04139 *[3/5],
% ylim([-3.5,0]);
% xlim([1, 1e3]); % Dn=0.04139 *[2/5],
% ylim([-6,0]);
% xlim([1, 2e3]); % Dn=0.04139 *[1/5],
% ylim([-11,0]);
% xlim([1, 4e3]); % Dn=0.04139 *[1/10],
% ylim([-17,0]);
% xlim([1, 5e3]); % Dn=0.04139 *[4/50],
% ylim([-20,0]);
% xlim([1, 1e4]); % Dn=0.04139 *[3/50],
% ylim([-25,0]);
% xlim([1, 1e4]); % Dn=0.04139 *[2/50],
% ylim([-30,0]);
% xlim([1, 3e4]); % Dn=0.04139 *[1/50],
% ylim([-35,0]);
% xlim([1, 5e4]); % Dn=0.04139 *[1/100],
% ylim([-40,0]);
```

```
    % Damped natural freq. of real and complex eigenvalues
    % Peak values of distributed model and approx TF
Wd_RealEig_Ht1=[]; Wd_CompEig_Ht1=[];Peak_Distrib_1=[];Peak_Ht1=[];
k1=1;
while k1<length(Wd_Ht1)
    if Wd_Ht1(k1)~=Wd_Ht1(k1+1)
        Wd_RealEig_Ht1=[Wd_RealEig_Ht1, Wd_Ht1(k1)];
        Peak_Ht1=[Peak_Ht1, MHc1_Wd_Ht1(k1)];
        k1=k1+1;
    else
        Wd_CompEig_Ht1=[Wd_CompEig_Ht1, Wd_Ht1(k1)];
        Peak_Ht1=[Peak_Ht1, MHc1_Wd_Ht1(k1)];
        k1=k1+2;
    end
end
if k1==length(Wd_Ht1)
    Wd_RealEig_Ht1=[Wd_RealEig_Ht1, Wd_Ht1(k1)];
    Peak_Ht1=[Peak_Ht1, MHc1_Wd_Ht1(k1)];
end
k2=1;
while k2<length(MTF_Wd_Ht1)
    if MTF_Wd_Ht1(k2)~=MTF_Wd_Ht1(k2+1)
        Peak_Distrib_1=[Peak_Distrib_1, MTF_Wd_Ht1(k2)];
        k2=k2+1;
    else
        Peak_Distrib_1=[Peak_Distrib_1, MTF_Wd_Ht1(k2)];
        k2=k2+2;
    end
end
if k2==length(MTF_Wd_Ht1)
    Peak_Distrib_1=[Peak_Distrib_1, MTF_Wd_Ht1(k2)];
end
Wd_RealEig_Ht1, Wd_CompEig_Ht1, Peak_Distrib_1, Peak_Ht1
% Peak_difference1=(Peak_Ht1-Peak_Distrib_1)

%% Comparison of step response of 9th/10th order and 0/4th order transfer function
approximation
figure, step(Ht0,'g-',Ht1,'b--')
legend('Normalized s, Approx TF, Ht0, 9/10 Order',...
    'Normalized s, Approx TF, Ht1, 0/4 Order', 4)
% title('Step Response')
xlabel('Normalized Time, \omega_V t');
```

```
ylabel('Normalized Volumetric Flow Rate, \DeltaQ_b/Q_e ');

%% Un-Normalized TF, convert above s_bar to s, s_bar=s*(r^2/KVis)
fprintf('%%%% Un-Normalized TF, convert above s_bar to s, s_bar=s*(r^2/KVis) \n');
ord=length(Nnum0);
uNnum0=[];uNden0=[];
for a=ord:-1:1
   uNnum0(a)=Nnum0(a)*(r^2/KVis)^(ord-a);
end
ord=length(Nden0);
for a=ord:-1:1
   uNden0(a)=Nden0(a)*(r^2/KVis)^(ord-a);
end
uHt0=tf(uNnum0,uNden0)
damp(uHt0)
[PR_uHt0]=pfract(uNnum0,uNden0)
DCgain_PR_uHt0=dcgain(PR_uHt0)

ord=length(Nnum1);
uNnum1=[];uNden1=[];
for a=ord:-1:1
   uNnum1(a)=Nnum1(a)*(r^2/KVis)^(ord-a);
end
ord=length(Nden1);
for a=ord:-1:1
   uNden1(a)=Nden1(a)*(r^2/KVis)^(ord-a);
end
uHt1=tf(uNnum1,uNden1)
damp(uHt1)
[PR_uHt1]=pfract(uNnum1,uNden1)
DCgain_PR_uHt1=dcgain(PR_uHt1)

%% Equal Lumped Series set: ICIC+(R+Rv), 4th order
fprintf('%%%% Equal Lumped Series set: ICIC+R+Rv, 4th order \n');
syms s Ps I0 I1 C1 C2 R1
% let I0=Is*i0; I1=Is*i1;
% C1=Cs*c1; C2=Cs*c2;
% R1=Rs*1;
H1=solve('Ps-P1=I0*s*Q0','Q0-Q1=C1*s*P1',...
   'P1-P2=I1*s*Q1','Q1-Q2=C2*s*P2',...
   'P2=R1*Q2','P1,P2,Q0,Q1,Q2');
% Q2(s)/(Ps(s)/R)=Q2b(s)/U(s), (normalized Q2)/(unit step input)
h1=collect(H1.Q2/Ps*(R1),s); % TF=Q/(Ps/R)
```

```
% h1=collect(h1,Ps);
pretty(h1)

% Equal distributed parameter
digits(5);
% h11=subs(h1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RR]);     % laminar
%     % Use Resistance of turbulent, RRt
h11=subs(h1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RRt]);    % turbulent
h11=vpa(h11);
pretty(h11)

[numh1,denh1]=numden(h1);
[CoeNum1,SS1]=coeffs(numh1,s);  % coefficient from high to low order
[CoeDen1,SS2]=coeffs(denh1,s);  % coefficient from high to low order

% let coefficient of lowest order be 1
n=length(CoeDen1);
NCoeNum1=CoeNum1/CoeDen1(n);
NCoeDen1=CoeDen1/CoeDen1(n);

% Assume this is a equal distributed parameter model
    % Use Resistance of turbulent, RRt
NumLump1=subs(NCoeNum1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RRt]);
DenLump1=subs(NCoeDen1,[I0,I1,C1,C2,R1],[II/2,II/2,CC/2,CC/2,RRt]);

TFLump1=tf(NumLump1,DenLump1)
% eig(TFLump1)
damp(TFLump1)
[PF_TFLump1]=pfract(NumLump1,DenLump1)
DCgain_PF_TFLump1=dcgain(PF_TFLump1)

figure, step(TFLump1,':r',uHt0,'g',uHt1,'-.b', .12)
% title('Step response')
ylabel('Normalized Volumetric Flow Rate, \DeltaQ_b/Q_e ');
legend('Equal Lumped Parameter Model for ICIC+R+Rv',...
   'Approx TF, uHt0, 9/10 Order',...
   'Approx TF, uHt1, 0/4 Order', 4)

figure, uH=bodeplot(TFLump1,':r',uHt0,'g',uHt1,'-.b');
p=getoptions(uH);    % Create a plot options handle p.
p.PhaseMatching = 'on';
p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[1,4/5],
p.Ylim{1} = [-3 5];
```

```
p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[3/5],
% p.Ylim{1} = [-4 3];
% p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[2/5],
% p.Ylim{1} = [-6 2];
% p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[1/5],
% p.Ylim{1} = [-12 2];
% p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[1/10],
% p.Ylim{1} = [-18 2];
% p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[4/50],
% p.Ylim{1} = [-20 1];
% p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[3/50],
% p.Ylim{1} = [-23 1];
% p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[2/50],
% p.Ylim{1} = [-30 0];
% p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[1/50],
% p.Ylim{1} = [-35 0];
% p.Ylim{2} = [-360 0];
% p.Xlim{1} = [1 1e3];       % Dn=0.04139 *[1/100],
% p.Ylim{1} = [-40 0];
% p.Ylim{2} = [-360 0];

setoptions(uH,p);    % Apply plot options to the Bode plot and render.
legend('Equal Lumped Parameter Model for ICIC+R+Rv',...
    'Approx TF, uHt0, 9/10 Order',...
    'Approx TF, uHt1, 0/4 Order', 'Location','Best')

%% Solve symbolic equations of series set lumped parameter model
fprintf('%%%% Solve symbolic equations of series set lumped parameter model \n');
warning on
syms s Ps i0 i1 c1 c2 r1 Is Cs Rs
% let I0=Is*i0; I1=Is*i1;
% C1=Cs*c1; C2=Cs*c2;
% R1=Rs*1;
H2=solve('Ps-P1=Is*i0*s*Q0','Q0-Q1=Cs*c1*s*P1',...
    'P1-P2=Is*i1*s*Q1','Q1-Q2=Cs*c2*s*P2',...
```

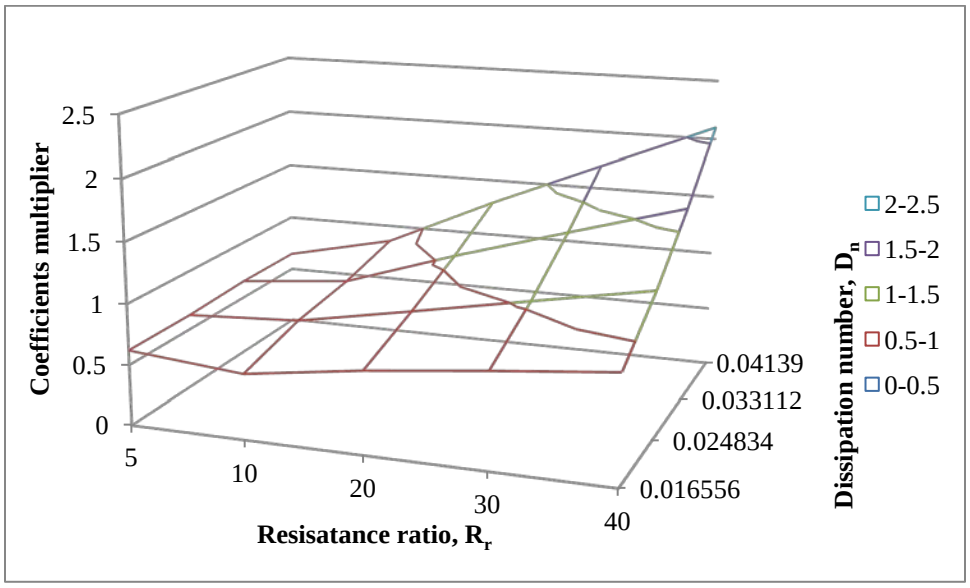
```
    'P2=Rs*1*Q2','P1,P2,Q0,Q1,Q2');

% Q(s)/Ps(s)*R=Qb(s)/U(s), (normalized Q)/(unit step input)
h2=collect(H2.Q2/(Ps/Rs),s); % TF=Q/(Ps/R)
pretty(h2)

[numh2,denh2]=numden(h2);
[CoeNum2,SS1]=coeffs(numh2,s);  % coefficient from high to low order
[CoeDen2,SS2]=coeffs(denh2,s);  % coefficient from high to low order

% let coefficient of lowest order be 1
n=length(CoeDen2);
NCoeNum2=CoeNum2/CoeDen2(n);
NCoeDen2=CoeDen2/CoeDen2(n);

% 0/4th order approximation numerical TF
uHt1=tf(uNnum1,uNden1)

F=solve(NCoeDen2(1)-uNden1(1), NCoeDen2(2)-uNden1(2),...
    NCoeDen2(3)-uNden1(3), NCoeDen2(4)-uNden1(4),...
    'i0,i1,c1,c2');

% the weight of each parameter
wIs=[F.i0, F.i1];
wCs=[F.c1, F.c2];
%     % Use Resistance of turbulent, RRt
wI=subs(vpa(wIs),{Is,Cs,Rs},{II,CC,RRt})
wC=subs(vpa(wCs),{Is,Cs,Rs},{II,CC,RRt})
```

Appendix C

I 's and C 's Coefficients vs. D_n vs. R_r for Case 1

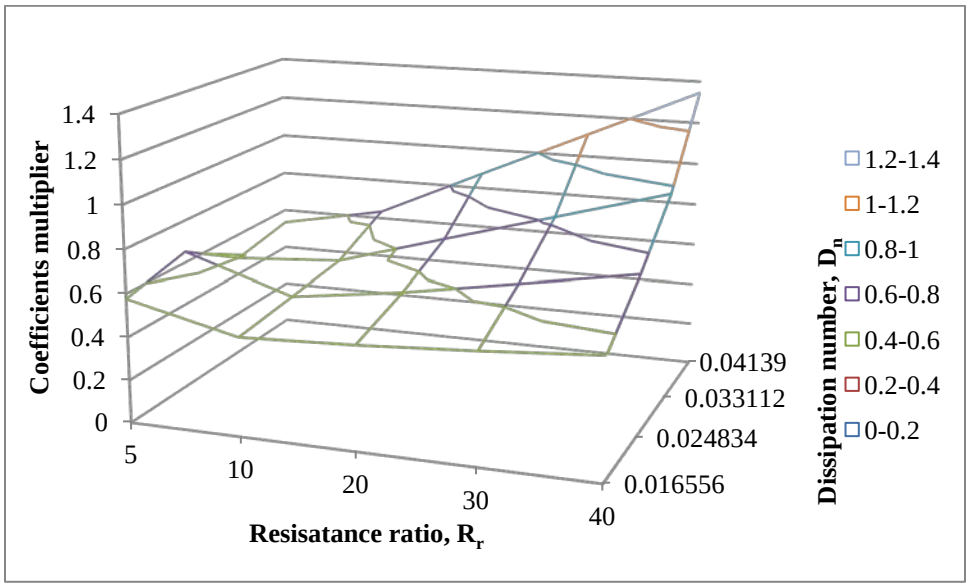
C. 1 First I 's coefficient, wl(1), vs R_r vs D_n

D_n \ R_r	5	10	20	30	40
0.016556	0.621	0.53675	0.67328	0.78159	0.88038
0.024834	0.6227	0.67297	0.85005	1.0309	1.2098
0.033112	0.64838	0.74072	1.055	1.3418	1.6223
0.04139	0.64684	0.85694	1.2949	1.6989	2.0999



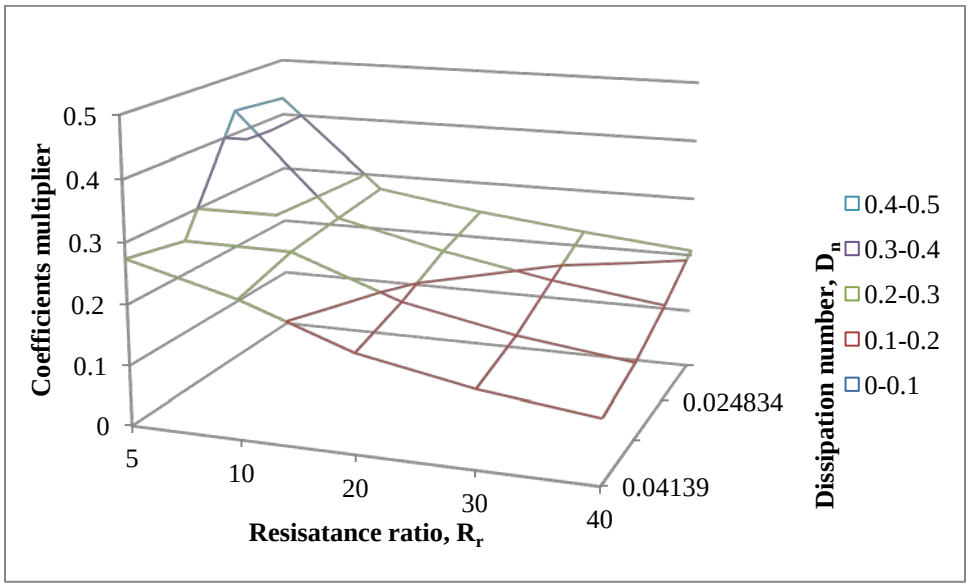
C. 2 Second I 's coefficient, wl(2), vs R_r vs D_n

D_n \ R_r	5	10	20	30	40
0.016556	0.57672	0.45742	0.47972	0.51201	0.55196
0.024834	0.65156	0.4862	0.558	0.64872	0.74253
0.033112	0.48054	0.51592	0.67018	0.82147	0.97269
0.04139	0.53435	0.63461	0.87367	1.1094	1.3459



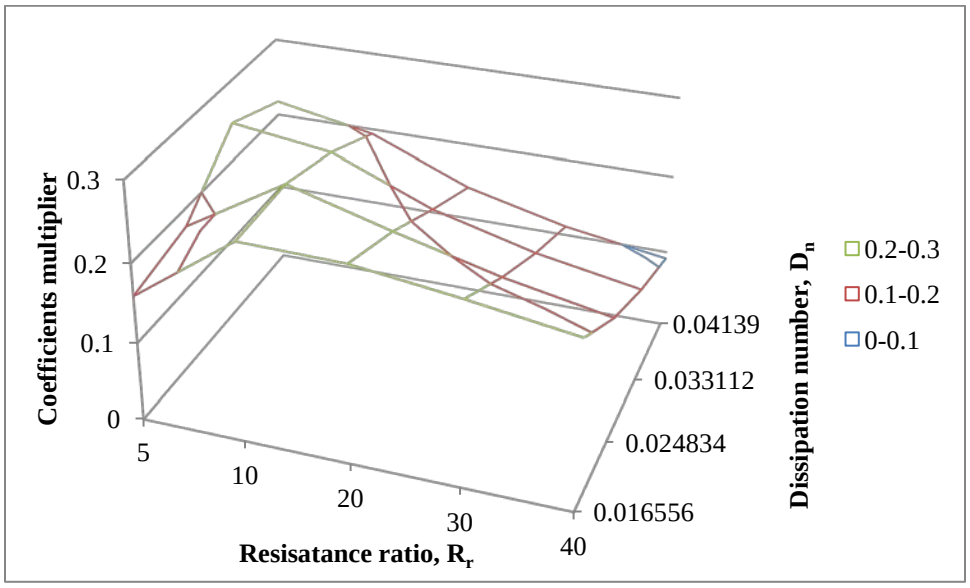
C. 3 First C 's coefficient, wC(1), vs R_r vs D_n

D_n \ R_r	5	10	20	30	40
0.016556	0.42995	0.27475	0.24696	0.22581	0.20801
0.024834	0.44008	0.26491	0.22279	0.18899	0.16422
0.033112	0.2556	0.2535	0.18746	0.15118	0.12727
0.04139	0.27422	0.22705	0.1618	0.12696	0.10464



C. 4 Second C's coefficient, wC(2), vs R_r vs D_n

D_n \ R_r	5	10	20	30	40
0.016556	0.15957	0.25067	0.24818	0.23195	0.2129
0.024834	0.17758	0.25307	0.21754	0.18497	0.15996
0.033112	0.24715	0.23023	0.17836	0.14472	0.12139
0.04139	0.21762	0.19415	0.14242	0.11144	0.091229



References

[1] Goodson, R. E. and Leonard, R. G., “A Survey of Modeling Techniques for Fluid Transmission Line Transients,” *Journal of Basic Engineering*, ASME Transaction, Series D, Vol. 94, June, 1972.

[2] Hullender, D., Woods, R., et. al., "Fluid Transmission Line Dynamics", A special publication of the ASME Winter Annual meeting, Boston, Massachusetts, November, 1983.

[3] Brown F. T., Margolis D. L., Shah R. P., “Small-Amplitude Frequency Behavior of Fluid Lines with Turbulent Flow,” *Journal of Basic Engineering*, Transactions of the ASME, Series D, 91(4), 678-693, December, 1969.

[4] Nursilo, W. S., “Fluid Transmission Line Dynamics,” Ph.D. Dissertation, University of Texas at Arlington, August, 2000.

[5] Jianing, P, “A Novel Technique for Analyzing Hydraulic Transients By Method of Characteristics,” 4th ASME_JSME Fluids Engineering Conference, FEDSM2003-45250, Honolulu, Hawaii, July, 2003.

[6] Wongputorn, P., Hullender, D., Woods, R., and King, J., “Application of MATLAB Functions of Simulation of Systems with Lines with Fluid Transients,” ASME *Journal of Fluids Engineering*, FE-03-1080, 2004.

[7] Hullender, D. A., Woods, R. L., and Huang, Y. W., “Single Phase Compressible Steady Flow in Pipes,” *Journal of Fluids Engineering* (ASME), Vol.132, No.1, 014502, 2010.

[8] Hullender, D., “Transfer Function Approximations for High or Infinite-Order Engineering Systems,” ME5305 Dynamic Systems Modeling Class Notebooks, The University of Texas at Arlington, September, 2009-2012.

[9] Levi, E. C., “Complex-Curve Fitting,” IRE Transactions on Automatic Control, AC-4, pp.37-44, May, 1959.

[10] Paynter, H. M., “Lumped Structures, Methods, and Apparatus and Approximations of Uniform Media by Lumped Structures,” U.S. Patent No. 3,044,703, July, 1962

[11] Gerlach, C. R., “Dynamic Models for Viscous Fluid Transmission Lines,” Proceedings of the 10th Joint Automatic Control Conference, Boulder, Colorado, 1969.

[12] Gerlach, C. R., “The Dynamics of Viscous Fluid Transmission Lines with Particular Emphasis on Higher Mode Propagation,” Ph.D. Dissertation, Oklahoma State University, July, 1966.

[13] Hullender, D. and Healey, A. J., “Rational Polynomial Approximation for Fluid Transmission Line Models,” Special Publication of the *Journal of Dynamic Systems, Measurement, and Control*, ASME, Nov., 1981.

[14] Hullender, D., Hsue, C., "Modal Approximations For the Fluid Dynamics of Hydraulic and Pneumatic Transmission Lines", Special Publication of the *Journal of Dynamic Systems, Measurement, and Control*, ASME, March, 1983.

[15] Moody, L. F., “Friction Factors for Pipe Flow,” Transactions of the ASME, Vol. 66, pp.671-684, 1944.

[16] Blasius, H., Forschungsarbeiten des Deutsch. Ing., No 131., 1913.

[17] Hullender, D., Woods, R., Hsue, C., "Time Domain Simulation of Fluid Transmission Lines Using Minimum Order State Variable Models", Special Publication of the *Journal of Dynamic Systems, Measurement, and Control*, ASME, June, 1983.

[18] Hullender, D., Healey, A., "State Variable Representation of Modal Approximation For Fluid Transmission Line Models", Special publication of the *Journal of Dynamic Systems, Measurement and Control*, ASME, Nov., 1981.

[19] Hullender, D., Woods, R., Hunn, R., "Transfer Functions for Fluid Transmission Lines with Differential Signals", Special Publication of the *Journal of Dynamic Systems Measurement, and Control*, ASME, Nov. 1981.

Biographical Information

Yi-Wei Huang was born on April 16, 1975 in Taipei, Taiwan. He attended Tatung University in Taipei, Taiwan and received his Bachelor of Science in Mechanical Engineering in June 1997.

He then joined the Army of Taiwan, during which time he was selected to have further training, and reached the rank of Corporal. He was released from the military service in 1999.

The next year, he was admitted to the McNeese State University in Lake Charles, Louisiana to pursue the Master program and completed his Master of Science in Mechanical Engineering in December 2002. The title of his Master's thesis was "The Intelligent House"

He was accepted an offer to pursue a Ph.D. degree in Mechanical Engineering at the University of Texas at Arlington and earned his doctoral degree in December 2012. His research interests are fluid transmission line dynamics and modeling with computer-aided software, which was the subject of his dissertation. During his study, he had a publication of "Single Phase Compressible Steady Flow in Pipes" in Journal of Fluids Engineering (ASME).