

CONSIDERATIONS FOR GENERATING ACCURATE
LINEAR TRANSFER FUNCTIONS FOR
LAMINAR FLOW TRANSMISSION
LINE DYNAMICS

by

SINA JASTEH

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

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2015

Copyright © by Sina Jasteh 2015

All Rights Reserved



Acknowledgements

I take this opportunity to express my sincere thanks to Professor Hullender, My supervising professor, for providing me with all the valuable instructions on the course of my project.

April 14, 2015

Abstract

CONSIDERATIONS FOR GENERATING ACCURATE
LINEAR TRANSFER FUNCTIONS FOR
LAMINAR FLOW TRANSMISSION
LINE DYNAMICS

Sina Jasteh, M.S.

The University of Texas at Arlington, 2015

Supervising Professor: David Hullender

Modeling fluid lines for applications such as hydraulic fracturing has been studied for many years, and we know that the inverse frequency method presents very accurate solutions for systems with lines in almost all laminar flow cases. However, utilizing this method is not easy for all users, and requires experience and understanding of the method. In this study a simplified method is introduced for accurately modeling the transients in lines which requires minimum experience in utilizing inverse frequency methods; the accuracy and limits of this simplified method are studied. Also, alternative approaches are presented which improved the accuracy of the final results.

Table of Contents

Acknowledgements	iii
Abstract	iv
List of Illustrations	vi
List of Tables	vii
Chapter 1 Introduction.....	1
Chapter 2 Generating Transfer Function Coefficients in Loops	9
Chapter 3 Modifying Tabulated Coefficients	15
Chapter 4 Direct Overall System Approximation	27
Appendix A The MATLAB Code of the Function Generating Component's Transfer Function's Data	34
Appendix B Fit Objects For the Transfer Function Coefficients.....	37
Appendix C New Tables for the ZChoshOverSinh Component's Transfer Function's Coefficients	39
Appendix D New Tables for the ZOverSinh Component's Transfer Function's Coefficients.....	43
Appendix E linfn MATLAB function	46
Appendix F FrackingTF input example function for linfn MATLAB function.....	49
Appendix G BrakeTF input example function for linfn MATLAB function.....	51
References	53
Biographical Information.....	54

List of Illustrations

Figure 1	Schematic of a fluid line.....	2
Figure 2	Frequency response of the infinite order Hydraulic Fracturing model.....	5
Figure 3	Fifth order model of ZCoshOverSinh component.....	11
Figure 4	A fit object for a coefficient of the ZCoshOverSinh component	12
Figure 5	Magnitude plot comparison of the fits generated using tables and loop data...	13
Figure 6	Fifth order model of ZCoshOverSinh component using data available in old tables.....	16
Figure 7	Fifth order model of ZCoshOverSinh component using new maximum frequency of fitting.....	17
Figure 8	Fifth order model of ZOverSinh generated by the values from the old table....	19
Figure 9	Magnitude plot comparison of fits generated using different methods.....	20
Figure 10	Time response comparison of system models.....	22
Figure 11	Fifth order model of ZOverSinh component for a low Dn value.....	24
Figure 12	Magnitude plot comparison of fits generated using different methods for a low Dn value	25
Figure 13	Time response comparison of system models for a low Dn value.....	26
Figure 14	hydraulic fracturing example fourth order direct fit using linfn MATLAB function.....	30
Figure 15	Brake example schematic.....	31
Figure 16	Brake example fifth order direct fit using linfn MATLAB function.....	32

List of Tables

Table 1 Poles and nat. freq. of ZCoshOverSinh component using the old table coefficient data	16
Table 2 poles and nat. freq. of ZCoshOverSinh component using new maximum frequency.....	18
Table 3 poles of overall system using different system models	21

Chapter 1

Introduction

In this study, the problem of accurately simulating the transients in a fluid line with laminar flow is being investigated; Three different methods are introduced to describe the flow of fluid through a line. These methods are described for application of the line in hydraulic fracturing process. The first method to generate a simulation model of a line to be used in hydraulic fracturing analysis is to generate a model using the inverse frequency algorithm directly on the frequency response of the total system, which is available as a function in MATLAB. This method could be complicated to utilize for some users, and also requires the ability to set up the MATLAB function settings and data points. The second method is developed by Huang, Hullender and Woods [1] which is a simplified method to create models for lines using pre-calculated data; In this method the inverse frequency is used on each component of the equation describing the flow of fluid in a line used in hydraulic fracturing application, and the results are stored in tables which allows users to use these tables to generate models for a broad range of line dimensions and fluid properties. Many attempts have been made in order to improve the tabulated data, and to improve the models generated by them, such as adding more data points, and adjusting the data available in the tables. Also, efficiency of the second method is compared to the first method for different line and fluid properties for the time response of the overall system. In the end a third method which uses a MATLAB function is introduced that uses the overall system transfer function instead of tables but uses algorithms to simplify the process and minimize the settings required to get a low approximation of the overall system transfer function. This method simplifies the process of direct curve fitting for overall system transfer function frequency response.

In order to start with the modeling process, first, we need to have a good understanding of the nature of the problem. The problem that is being investigated in this study has applications in hydraulic fracturing. In the process of Hydraulic Fracturing fluid is pumped into pipes, and at the end of each pipe, fluid penetrates the rock fissures. Pumping fluid in these lines with pulses with a frequency equal to the resonant frequency of the lines can result in amplification of these pulses, and more penetration of the fluid in the fissures. Accurately modeling these lines is crucial in finding the correct resonant frequencies of the lines, and amplification of input pulses.

These lines can be components of a more complex system. To obtain a total system linear model all the components in the system have to be linear. Therefore, to model a system with fluid lines as components of the system, linear ordinary differential equations have to be obtained for each of the fluid lines present in the system.



Figure 1 Schematic of a fluid line

Figure 1 represents a line with fully developed laminar flow input; P_{in} and Q_{in} are pressure and volumetric rate of the flow which enters the line, and P_{out} and Q_{out} are pressure and volumetric rate of the flow exiting the line.

The input pulses that enter this line and the output generated by these pulses can be describe by changes in P and Q values, ΔP and ΔQ . These pressure and flow

perturbations for one-dimensional laminar flow in a line are described by the infinite order model in the equation 1:

$$\begin{bmatrix} \Delta P_{In} \\ \Delta P_{out} \end{bmatrix} = \begin{bmatrix} \frac{Z \cosh \Gamma}{\sinh \Gamma} & -\frac{Z}{\sinh \Gamma} \\ \frac{Z}{\sinh \Gamma} & -\frac{Z \cosh \Gamma}{\sinh \Gamma} \end{bmatrix} \begin{bmatrix} \Delta Q_{In} \\ \Delta Q_{out} \end{bmatrix} \quad (1)$$

The Γ and Z functions are functions of zero and first order Bessel functions.

From equation 1 we have:

$$\Delta P_{out} = \left[\frac{Z}{\sinh \Gamma} \right] \Delta Q_{In} - \left[\frac{Z \cosh \Gamma}{\sinh \Gamma} \right] \Delta Q_{out} \quad (2)$$

Equation 2 describes the changes in the output pressure as a function of changes in output and input flow rates in a fluid line.

In hydraulic fracturing, the change in the flow rate at the end of the line is a function of pressure change at the exit point and resistance of the rock fissures; This resistance is not linear, but for simplicity it is assumed to be linear as in Darcy's equation, and approximately six times the resistance of the line.

$$\Delta Q_{out} = f(\Delta P_{out})$$

$$\Delta P_{out} = 6R_L \Delta Q_{out}$$

Plugging this equation into equation 2 we have:

$$\Delta Q_b = \frac{\frac{Z}{\sinh \Gamma}}{6R_L + \frac{Z \cosh \Gamma}{\sinh \Gamma}} \Delta Q_a \quad (3)$$

Equation 3 describes the overall system infinite order transfer function for hydraulic fracturing.

In equation 3:

$$Z(s) = \frac{\rho c}{\pi r^2} \frac{1}{\sqrt{1-B}}$$

$$B(s) = \frac{2J_1 i \sqrt{s}}{i \sqrt{s} J_0 i \sqrt{s}}$$

$$\Gamma(s) = \frac{D_n \bar{s}}{\sqrt{1-B}}$$

J_1 and J_0 are Bessel functions of first and zero order

c is the speed of sound in the fluid which can be found from the following

formula: $c = \sqrt{\frac{\beta}{\rho}}$

β is bulk modulus, and ρ is density of the fluid.

D_n is the line dissipation number: $D_n = \frac{vL}{cr^2}$

\bar{s} is the normalized Laplace operator which can be found by this formula: $\bar{s} = \frac{r^2}{v}$

R_L is steady flow resistance in line: $R_L = \frac{8\mu L}{\pi r^4}$

L is the length of line in meters.

μ is absolute viscosity (Ns/m²)

The equation 3 represents the infinite order model of the system, and the frequency response of this infinite order model can be found in the figure 2.

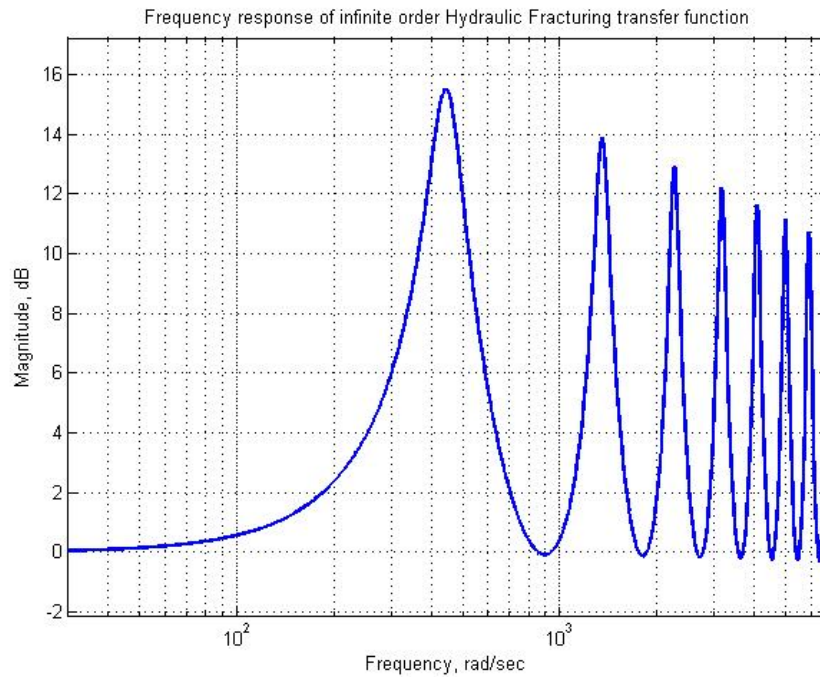


Figure 2 Frequency response of the infinite order hydraulic fracturing model

The plot 2 is generated for a system containing a line with the following properties:

$$\beta = 1.8246 \times 10^9 \frac{N}{m^2} \quad \rho = 855.24 \frac{Kg}{m^3}$$

$$\text{Kinematic Viscosity} = 7.618 \times 10^{-6} \frac{m^2}{s}$$

$$\text{Radius, } r = 0.003175 \text{ m} \quad \text{length, } L = 5 \text{ m}$$

The aim is to find the time domain approximation of the system for a range of frequencies containing the major modes of the hydraulic fracturing example, and accurately approximating the lowest resonant frequencies; The time domain analysis of this model can be a good representation of the infinite order model of the system in this example depending on the application.

There are different algorithms that provide linear stable differential equations to model a system. The modeling methods presented in this study use the inverse frequency algorithm, which is one of the most accurate algorithms to generate transfer functions from frequency response data. This method matches the frequency response of the true system on a range of frequencies with minimum number of poles and zeros needed.

Inverse frequency algorithm uses least square curve fitting methods on the frequency response of the system to acquire an accurate linear model of the system; This model approximates the system's frequency response on a range of frequencies which the algorithm is used on. In addition, inverse frequency algorithm provides the option to choose the order of the model required for the line by the user as well as the frequency range of application.

We know that using inverse frequency algorithm directly on the frequency response of an overall system frequency response is one of the most accurate and reliable methods to get a linear model for a system. However, using inverse frequency algorithm requires experience and knowledge of utilizing the algorithm; For instance, generating frequency data points from the infinite order model is one of the most important factors in finding a good fit for the frequency response, and getting an accurate model for the system. Finding this frequency range involves choosing a good lower and upper frequency limits which is only possible through trial and error, and requires experience in curve fitting. Also, generating enough number of frequency response data points from the infinite order model can be challenging for some users. The other difficulty associated with use of inverse frequency method directly is that the settings and arrangement of frequencies have to change for each change in a property of the system such as a change in fluid or line properties. This means that the settings for lower and

upper frequencies of fitting and possibly the number of data points has to be adjusted for any alterations in the system transfer function, and this process can be time consuming.

The method developed by Huang, Hullender and Woods [1] allows the user to acquire a reasonably accurate low order linear transfer function fitting the true system's transfer function frequency response on the range of the first two resonant frequencies of the overall system without using the inverse frequency algorithm directly and complexities that comes with it.

In this method the curve fitting using inverse frequency algorithm is being done for each individual hyperbolic infinite order transfer functions in the equation 3, $\frac{z}{\sinh\Gamma}$ and $\frac{z\cosh\Gamma}{\sinh\Gamma}$. Transfer function coefficients generated using inverse frequency algorithm are tabulated and stored to be called later for different applications. Tabulated data will provide the coefficients for transfer functions for a broad range of line specifications such as diameter and length of the line, fluid properties such as fluid density, viscosity and bulk modulus. The coefficients generated by inverse frequency algorithm, are tabulated based on "dissipation numbers" which are functions of line and fluid properties. By changing dissipation number values in a range from 0.00001 to 0.5, and generating coefficients for the tables, a broad range of line and fluid properties is covered. Any other values between the values present in the table can be simply interpolated and extracted from tables, and plugged into the transfer function to generate the required components of the model.

There are questions about this process to be answered: How many data points in each table are required for good interpolation results? Is this method effective and accurate for all range of dissipation numbers presented in this study? These are among the questions that are investigated in the course of this study.

The approach and the tabulated data are presented in the paper by by Huang, Hullender and Woods [1]. The first aim in this study, is investigating the accuracy and the range that this method is applicable, and secondly, if possible, find a way to improve the results.

Chapter 2

Generating Transfer Function Coefficients in Loops

The first approach was to increase the accuracy by adding more data points which was initially assumed to be the reason behind some inaccuracy in the initial table. It was assumed that by finding accurate fits up to the frequency of the second pole of each component, and adding more data points, the final results are going to be more accurate.

The first attempt to generate an accurate table of coefficients was to generate these coefficients in a loop using an algorithm to choose a range of frequencies for the fitting and apply inverse frequency algorithm on this range. The focus in this step is on generating accurate fit for each of the hyperbolic causality transfer functions which are used in the hydraulic fracturing model, equation 3. The benefit of generating coefficient data in a loop is that very high number of data points can be generated in loops which may be time consuming to generate by running the functions manually.

There are two hyperbolic components in the hydraulic fracturing equation, equation 3. The more challenging component to fit using inverse frequency is the $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component. In this report the fitting process is explained for this component. But, similar process has been applied to $\frac{Z}{\sinh\Gamma}$ component. The MATLAB code of the function generating $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component transfer function data is available in appendix A.

A fifth order transfer function is fitted to $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component; the order of fitting is assumed to be enough to get an accurate fit for the first two modes of this function. The fitting process is done in a loop; first, dissipation value numbers (Dn) generated logarithmically for a range of Dn values. The reason that the Dn values are spread logarithmically is that the coefficients associated with smaller Dn values tend to be more

difficult to interpolate, so the concentration of data points are more focused on the smaller Dn values.

The range of frequencies used in this loop is decided by a function which identifies the local maximums on the frequency response magnitude plot of the infinite order function, and uses local maximums on the plot to decide the maximum frequency of fitting. The MATLAB function that finds the local maximums of a vector is directly applied to the frequency response generated from the infinite order transfer function.

The maximum frequency of the fitting is chosen by the formula presented in equation 4 for the range of Dn values between 0.00001 and 0.0001:

$$\text{maximum frequency of fitting} = \text{frequency of second local maxima} - 0.04 \times (\text{frequency of second local maxima} - \text{frequency first local maxima}) \quad (4)$$

It should be mentioned that the 0.04 value in the equation 4 has been found by trial and error. The maximum frequency generated by this formula in the loop for Dn values between 0.00001 and 0.0001 will result in a fit that covers the first pole of the infinite order model, and up to the second pole.

In figure 3 the frequency response of $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ function, and the fifth order fit to this function for the Dn value of 0.00001 can be seen.

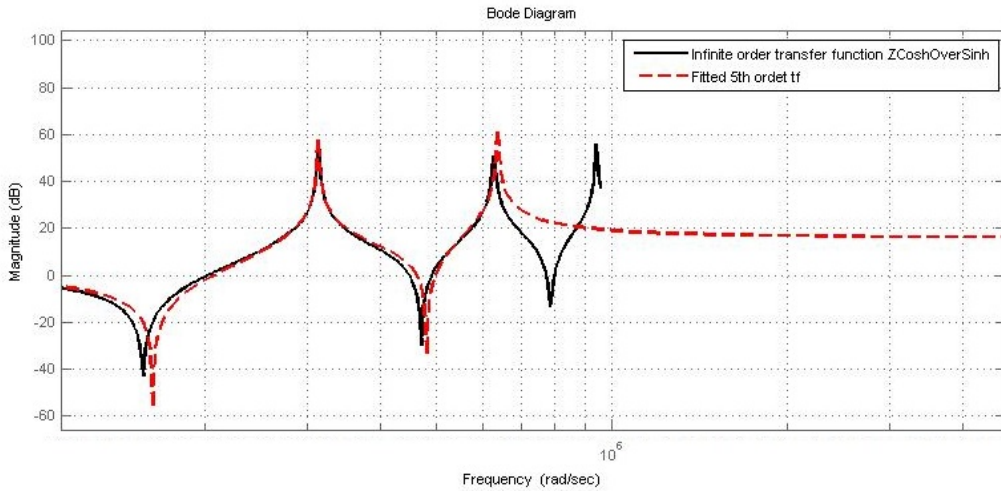


Figure 3 Fifth order model of $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component

The Maximum frequency generated by the formula above for the earlier example ($D_n=0.00001$) is 6.26×10^5 .

Weighting factors could be added to the fitting process, although no significant change in the accuracy of the fit on the critical frequencies was observed using weighting factors. Choosing a correct maximum frequency for fitting had much more impact in the accuracy than weighting factors.

Since running these loops for high number of iterations can be time consuming, the transfer functions generated in the loops is saved, and called separately to generate fit objects for each of the transfer function coefficients. A curve is fitted to each coefficient value for a range of D_n values. For example, Figure 4 shows the change of one of the transfer function coefficients of the $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ fifth order model for D_n values between 0.00001 and 0.0001

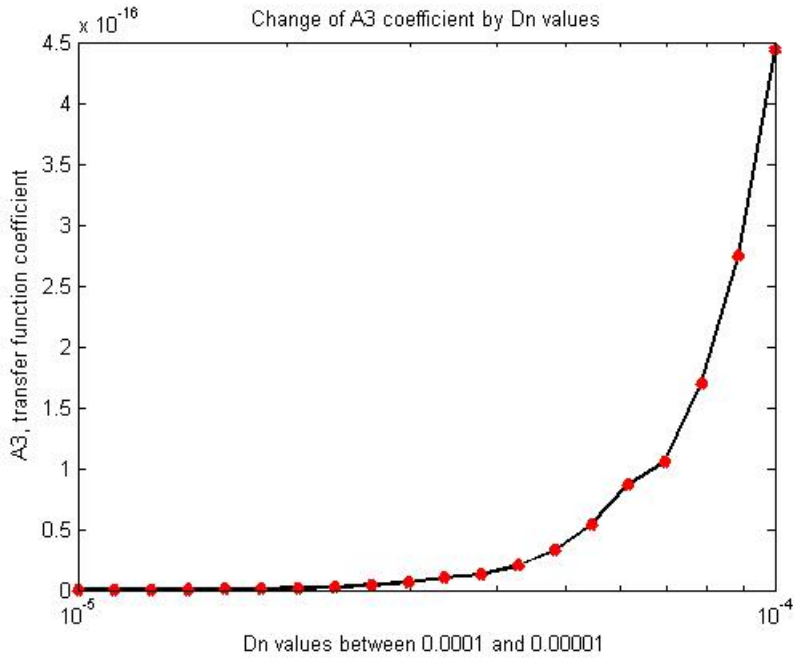


Figure 4 A fit object for a coefficient of the $\frac{Zcosh\Gamma}{sinh\Gamma}$ component

The curve fitted to the coefficient data plot will be the source for extracting new coefficient values for a desired new function. For instance, eight curve fit objects, similar to the plot above, are generated for $\frac{Zcosh\Gamma}{sinh\Gamma}$ component. New coefficients can be extracted from these fits for new transfer functions. The MATLAB files for the coefficient fittings are available in the appendix B.

In this example there are 20 data points logarithmically spaced from Dn value of 0.00001 to 0.0001. It is expected that adding data points will result in more accurate final transfer functions. However, in the figure 4 it can be seen that this is not the case. Figure 5 is the frequency response plot of equation 3 which the coefficients of the $\frac{Zcosh\Gamma}{sinh\Gamma}$ component of this transfer function is generated using the loop. A fifth order fit for the $\frac{Zcosh\Gamma}{sinh\Gamma}$ component, and a fourth order fit for the $\frac{Z}{sinh\Gamma}$ component are combined using the

equation 3 to make a ninth order fit for the equation 3 transfer function. Figure 5 is for D_n value of 0.000087 .

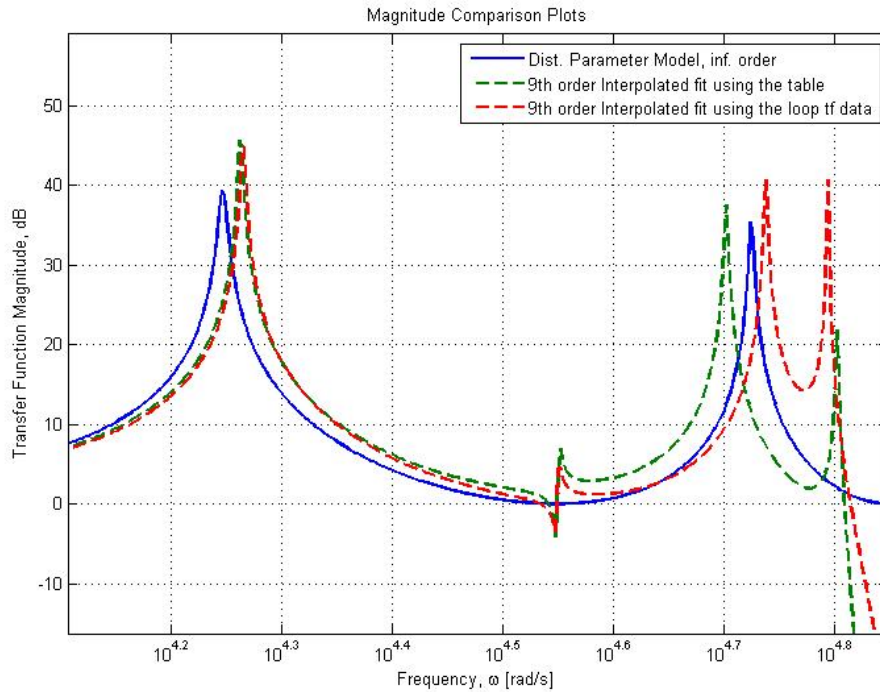


Figure 5 Magnitude plot comparison of the fits generated using tables and loop data

It is evident from the Figure 5 that there is no significant improvement in fitting the first pole using this method, and neither of the fits are matching the first and second poles of the true equation 3 transfer function.

Looking at the results from this method and comparing them with the results from the original tables we can conclude that the inaccuracy of the results from the original tables were not result of shortage of data points in the table, and using the loop method to increase the number of data points didn't result in better final fit. In other words, having a large number of D_n values (20 or more) is not necessary, and accurate results can be

achieved with fewer numbers of data points using interpolation. Overall, no significant improvements has been observed by generating fitted transfer function coefficients using a loop with high number of cycles. By looking at the individual fits of the components of the hydraulic fracturing equation 3 model it could be conclude that although we have good fits for each of the components the combination of these functions to make the final transfer function doesn't even match the first pole correctly.

During the process of fitting I realized that the most important factor in the fitting process using inverse frequency algorithm is the maximum frequency of fitting, and this frequency decides if we are going to have a good fit for the first pole or not when combining transfer functions. Using the right maximum frequency for the fitting process was crucial for fitting the first pole, and also the first dip in the magnitude plot of the $\frac{z\cosh\Gamma}{\sinh\Gamma}$ component. In the next chapter, the process of choosing the right maximum frequency for equation 3 will be explained, and the results are compared to the model generated directly by inverse frequency algorithm.

Chapter 3

Modifying Tabulated Coefficients

In the second attempt to improve the tabulated coefficients' data, the attention is on getting an accurate fit for the combined transfer components in equation 3 instead of trying to obtain an accurate fit up to the second pole for each individual component of the transfer function. Transfer function coefficients for each of the components of the hydraulic fracturing equation were generated, and were plugged directly into the equation 3. The final transfer function response was compared to the true transfer function response. By changing the range of frequency of fitting for each of the components and looking at the combined transfer function in equation 3, new frequency ranges have been used to obtain a good fit for hydraulic fracturing example equation.

In the paper by Huang, Hullender and Woods [1] "Baseline models" which are high order models are generated for comparison purposes. In the example below, an 11th order model for the baseline model for the $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component for Dn value of 0.001 was used as a baseline model, which fits the infinite order model very well through the first two poles of the function. This baseline model gives us an accurate estimation of the first two poles of the function. The first pole is at 3100 rad/sec and the second pole is at 6230 rad/sec.

In the figure 6 and figure 7 the $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component's fit using the coefficients available in the old table of coefficients, and a fit with alternative maximum frequency of fitting are compared. Note that the maximum frequency of fitting in the figure 6 is 6100 rad/sec; this frequency is very close to the frequency of the second pole of the infinite order function which is at 6230 rad/sec.

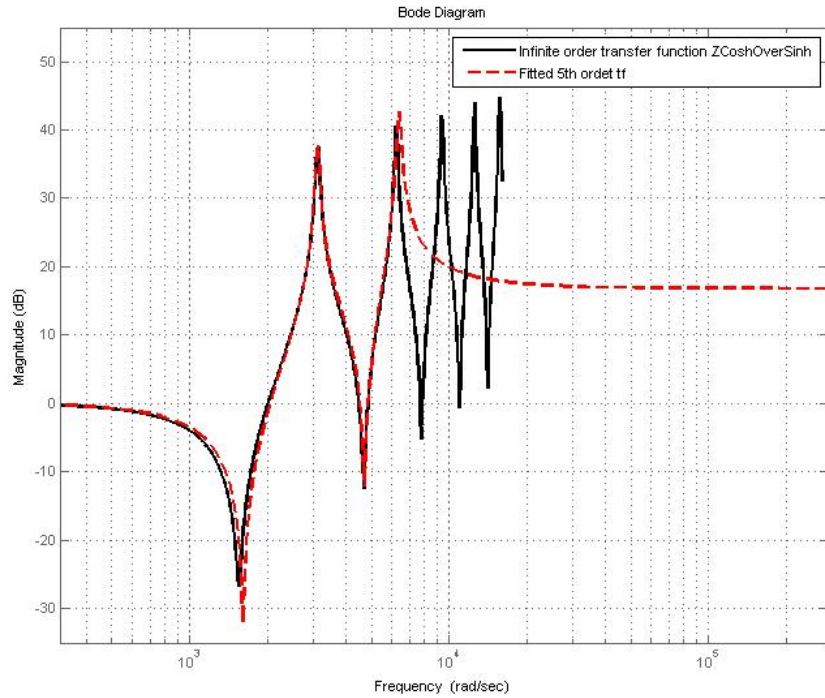


Figure 6 Fifth order model of $\frac{Z_{\cosh\Gamma}}{\sinh\Gamma}$ component using data available in old tables

The position of the poles and the frequencies of the poles for this fifth order fit is available in the table below:

Table 1 poles and nat. freq. of $\frac{Z_{\cosh\Gamma}}{\sinh\Gamma}$ component using the old table coefficient data

Eigenvalue	Damping	Frequency (rad/s)
-4.00e+001 + 3.10e+003i	1.29e-002	3.10e+003
-4.00e+001 - 3.10e+003i	1.29e-002	3.10e+003
-9.04e+001 + 6.38e+003i	1.42e-002	6.38e+003
-9.04e+001 - 6.38e+003i	1.42e-002	6.38e+003

looking at the Figure 6 and Table 1 we can see that the fifth order model is matching the first pole very accurately, and the fitting estimation for the second pole is very close to the original second pole but not as accurate as the first pole. Figure 7 shows the magnitude response of a model generated using the method which focuses on the accuracy of the overall system fit instead of individual fits. In this fit the maximum frequency of fitting is 5000 rad/sec, which is after the second dip in the plot but way before the second pole. The position of the poles and the frequencies for this fifth order fit is available in Table 2.

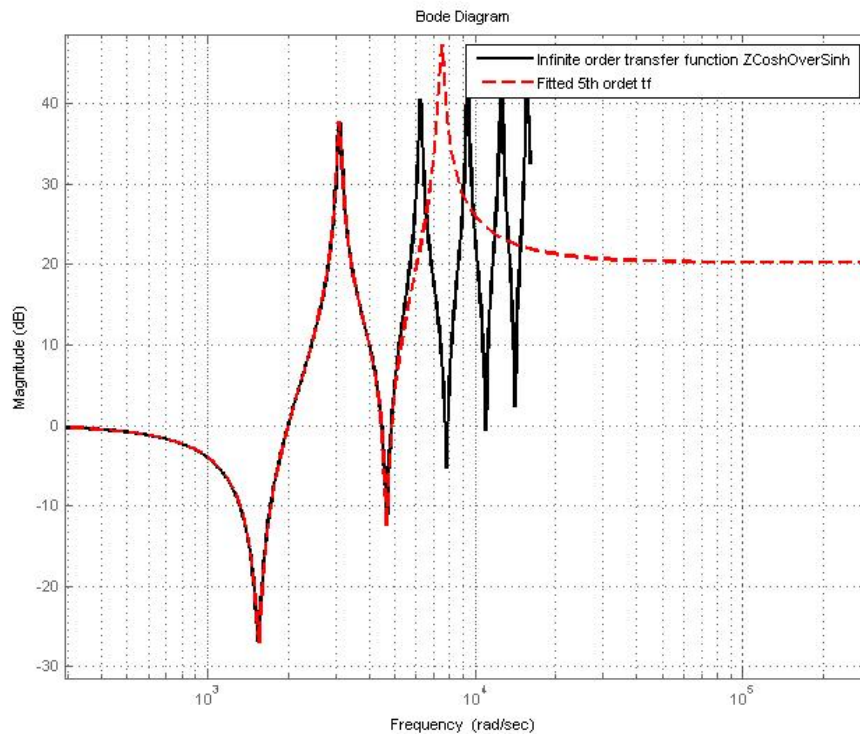


Figure 7 Fifth order model of $\frac{Z_{cosh}\Gamma}{\sinh\Gamma}$ component using new maximum frequency of fitting

Table 2 poles and nat. freq. of $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component using new maximum frequency

Eigenvalue	Damping	Frequency (rad/s)
-4.02e+001 + 3.10e+003i	1.30e-002	3.10e+003
-4.02e+001 - 3.10e+003i	1.30e-002	3.10e+003
-1.18e+002 + 7.48e+003i	1.58e-002	7.48e+003
-1.18e+002 - 7.48e+003i	1.58e-002	7.48e+003

It can be seen in the figure 7 that the first pole is matching very accurately, while the second pole is not matching. However, note that the fifth order fit is matching the infinite order function all the way after the second dip in the plot.

Now we look at the plot for the $\frac{Z}{\sinh\Gamma}$ component which is generated by the values from the old table, and the range of fitting used for this component.

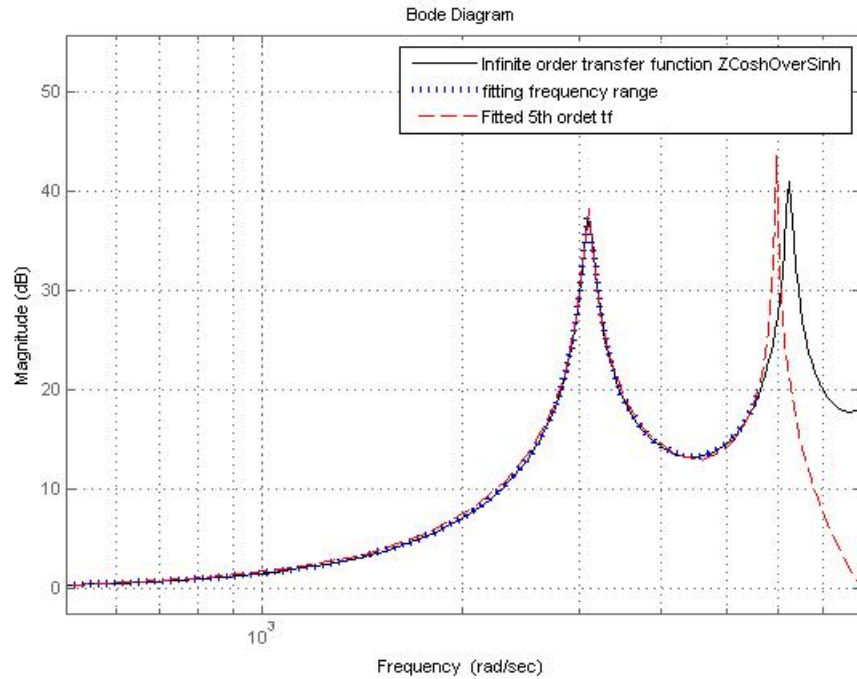


Figure 8 Fifth order model of $\frac{Z}{\sinh\Gamma}$ generated by the values from the old table

By looking at the graphs and using different maximum frequencies for the fitting of the $\frac{Z}{\sinh\Gamma}$ component, and by trial and error, it could be conclude that the fitting of the second pole of both of the components is not significant in the combined transfer function result for equation 3, as long as the first pole is fitting well throughout the middle of the dip in the magnitude plot for each component. In the Figure 8 the frequency range is chosen in a way that the first pole fits accurately throughout the maximum frequency chosen. Extending the maximum frequency of fitting more than this will result in inaccuracy in the first pole fitting.

The results are analyzed in order to observe how combining these transfer functions will affect the final transfer function in equation 3. Note that in the Figure 9, the

$\frac{Z}{\sinh\Gamma}$ component transfer function is generated using the old available tables, and only the

$\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component coefficients are different between two fits.

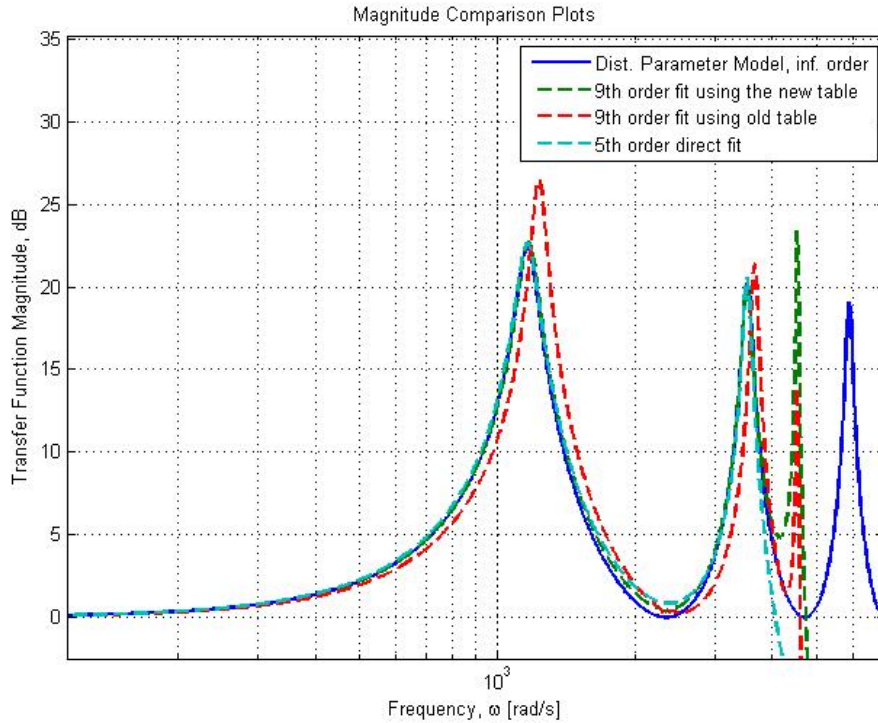


Figure 9 Magnitude plot comparison of fits generated using different methods

Figure 9 shows that even though the fit for individual components at their frequency of the second pole is not fitting particularly well the overall combined functions fits the true transfer function frequency response of hydraulic fracturing example equation very well up to the second pole. To compare the accuracy of the fits generated by the table values, a fifth order model is fitted to the true frequency response. In the Table 3 the frequency of the poles of these models are compared:

Table 3 poles of overall system using different system models

Fifth order direct fit	Fit using New table	Fit using Old table
$-5.87e+001 + 1.16e+003i$	$-5.66e+001 + 1.17e+003i$	$-3.83e+001 + 1.23e+003i$
$-5.87e+001 - 1.16e+003i$	$-5.66e+001 - 1.17e+003i$	$-3.83e+001 - 1.23e+003i$
$-6.03e+001 + 3.52e+003i$	$-3.08e+001 + 2.34e+003i$	$-3.08e+001 + 2.34e+003i$
$-6.03e+001 - 3.52e+003i$	$-3.08e+001 - 2.34e+003i$	$-3.08e+001 - 2.34e+003i$
$-5.43e+014$	$-7.75e+001 + 3.52e+003i$	$-5.45e+001 + 3.66e+003i$
	$-7.75e+001 - 3.52e+003i$	$-5.45e+001 - 3.66e+003i$
	$-1.71e+001 + 4.54e+003i$	$-1.71e+001 + 4.54e+003i$
	$-1.71e+001 - 4.54e+003i$	$-1.71e+001 - 4.54e+003i$
	$-1.62e+005$	$-9.48e+004$

In the Table 3 it can be seen that the first pole of the new table's model is matching the first pole of the fifth order direct fit's model. The pole at 2340 rad/sec has failed to be canceled in the new and old tables' models. The third pole of the new table's model which is at 3520 rad/sec is matching the second pole of the direct fit.

For time response comparison, time response of these two transfer functions are compared with the time response of the 5th order direct inverse frequency fit. The time response of the system is generated for a unit pulse series with a frequency equal to the true frequency of the first pole of the system. The Figure 10 shows the time response of the three models explained above for Dn value of 0.001, the frequency of pulses are at 1170 rad/sec, the first resonant frequency of the system.

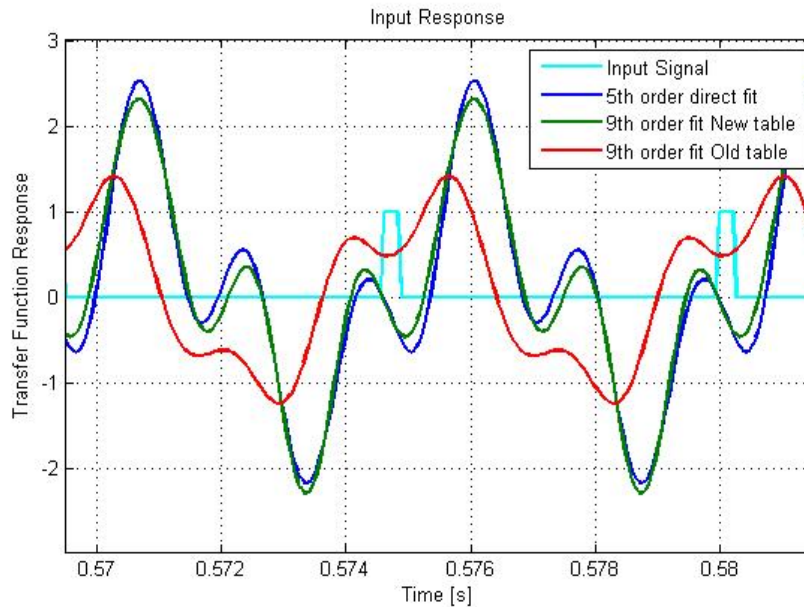


Figure 10 Time response comparison of system models

It is evident from Figure 10 that using the model generated by the new table's coefficients up to 92% of the amplification of the fifth order model can be achieved. This is significantly higher than only 55% for the model generated using the old tables.

It should be mentioned that the amplification we see in the plot above is only for the first two modes of the system, and adding more modes will result in higher amplification. However, the first two modes of the system have the highest impact in the amplification of the pulse series.

By looking at the accuracy of the fits using tabulated data, and comparing them to direct fit, one may conclude that this is a good alternative to using inverse frequency directly. This is true, however, the same quality of fitting couldn't be achieved for the very low D_n values. Specifically, the fifth order model for the $\frac{Z \cosh \Gamma}{\sinh \Gamma}$ component wouldn't match

the infinite order model well enough no matter what frequency range or weighting factors are chosen for the fitting process.

For the low values of dissipation number the order of changes in the frequency response plot are 10^5 , and at high frequencies a small inaccuracy in the fitting can result in inefficiency of the method.

In the Figure 11 the frequency range of the fit for the $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ component is also decided by the combined transfer function results for hydraulic fracturing example similar to the previous example for the dissipation number value of 0.001, but it can be seen that the model is not fitting the first dip in the magnitude plot as well as the previous example. Note that the frequency range chosen for this fitting is the best frequency range possible to match the first mode of the final infinite order model of the overall system, and reducing and increasing the maximum frequency of fitting didn't improve the results. For the frequencies in the order of 10^5 , this method cannot deliver the same relative accuracy as with low Dn values such as 0.0000123; In the Figure 11 the frequency response of the model of the component for this Dn value can be seen.

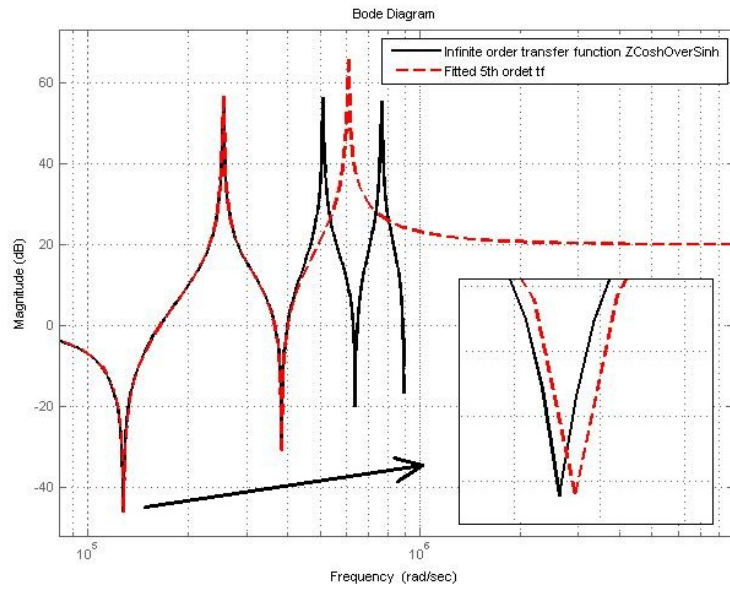


Figure 11 Fifth order model of $\frac{Z_{cosh}\Gamma}{\sinh\Gamma}$ component for a low Dn value

The frequency response magnitude plot of the combined transfer functions for the Dn value of 0.0000123 for both new and old tables, and also the fifth order direct fit is available in the Figure 12.

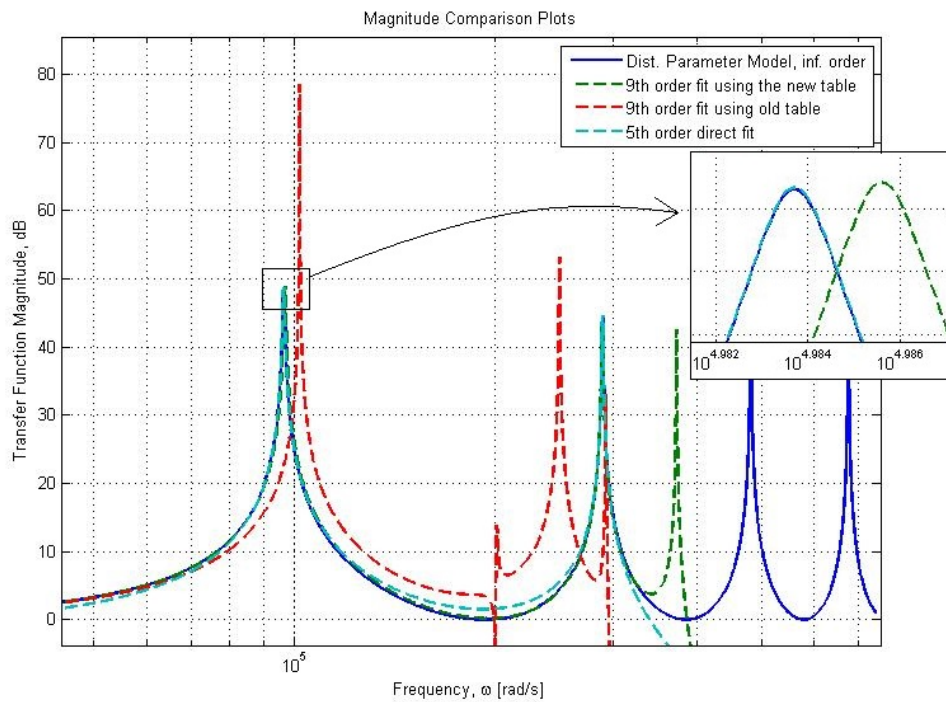


Figure 12 Magnitude plot comparison of fits generated using different methods for a low Dn value

Figure 12 shows the failure of the fifth order model of the $\frac{Zcosh\Gamma}{sinh\Gamma}$ component to fit the infinite order model with an accuracy suitable for the order changes in the frequency response (In this case, 10^5 rad/sec). The result of this inaccuracy can be seen clearly in the pulse series time response of the model in the Figure 13.

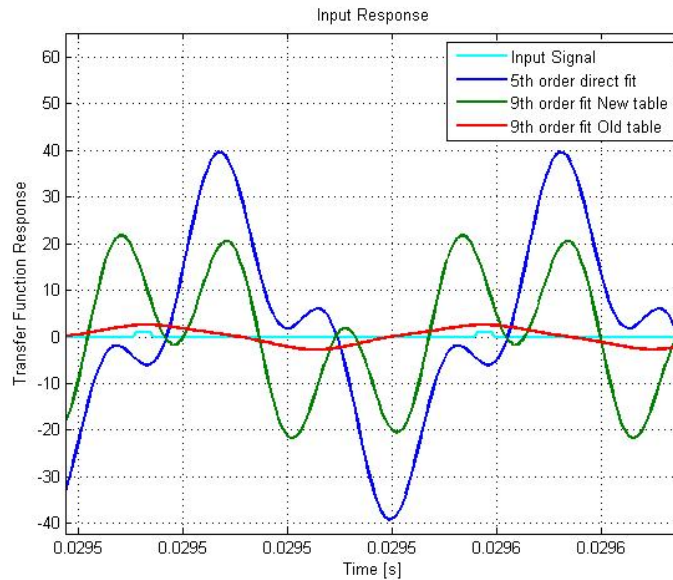


Figure 13 Time response comparison of system models for a low Dn value

From Figure 13 it is clear that the model generated using the new table value is not working efficiently, and the efficiency of the method using tabulated data compared to the direct fit method has dropped to 50%. This value was above 90% for higher Dn values.

To ensure accurate interpolated coefficients from tables, several data points were added to the tables. The new tables are available in the appendix C and D.

right maximum frequency will be explained, and the results are compared to the model generated directly by inverse frequency algorithm.

Chapter 4

Direct Overall System Approximation

The difficulties associated with the choice of maximum frequency used with the inverse frequency in the method that combines low order linear transfer functions to get the total transfer functions were explained in the chapter 3. Lacking a practical guide line for choosing maximum frequency of fitting for components in the method explained in chapter 3 makes this method inefficient and impractical. In other words one needs to look at the frequency response of the combined transfer function and compare it to the true frequency response of the infinite order transfer function, and change the maximum frequency of fitting for components accordingly to get a good combined fit. This method is impractical, and the coefficient tables generated with this method are not useful for new equations with different arrangement of components. For instance, the plots in the chapter 2 and 3 are generated for the hydraulic fracturing equation, and respectively the fitting process and maximum frequency of fitting for each component in these two chapters are generated exclusively for this equation. Although, tables generated for the hydraulic fracturing equation components can be used for different resistances in the rock fissures simplified model.

The complications and the restrictions associated with this method reinforced the idea that one would be better off using the inverse frequency method directly on the overall system transfer function, and get a fit for the whole system instead of the components like the method in the chapters 2 and 3. This method is more accurate and is applicable to any system transfer function. However, the direct use of inverse

frequency should be facilitated in a way that applying the method becomes easy for inexperienced users; therefore, therein lies the motivation for this third approach.

A MATLAB file is developed to provide a simplified way to acquire low order transfer function approximation of an overall system. This MATLAB file has the name "linfn". The code for "linfn" M-file and instruction on how to use this function is available in the appendix E. linfn takes a function as an input which contains the equation in for the overall system and all the variables and values associated with them expressed in frequency domain. The MATLAB input function which has the overall system equation should take a frequency as an input, and returns the response to that frequency as an output. The user specifies the name of the function in string format, and the order of approximation needed for the output linear transfer function when calling the linfn function. The input function being in a separate function rather than combined in the body of the main MATLAB function, linfn, brings the advantage of changing the input without the need of changing the settings in the linfn. With every change in the input equation such as lines and fluids property changes the user does not need to change the settings in the linfn function. Therefore, the process is faster and more convenient for users which is one of the major motivations in developing this M-file.

The linfn M-file automatically generates frequency data points by calling the input function in a loop. An algorithm decides the maximum frequency needed to be used with the inverse frequency function based on the order of approximation needed. The frequency of the modes and the order of the approximation is the deciding factors while selecting the maximum frequency of fitting. The algorithm that decides the maximum frequency uses a function that finds the local maximums in the frequency responses' vector with the respective frequency vector indices. Based on the number of the modes

needed in the approximation the respective local maximum will be used in the process of choosing maximum frequency of fitting.

Figure 14 shows the frequency response and a fourth order approximation of the equation 3 generated using the `linfn` M-file. The M-file used as an input function to the "`linfn`" function for the hydraulic fracturing example is available in appendix F. The maximum frequency required for the fitting is decided in a way that covers the first two modes of the equation 3 transfer function. The fourth order fit is matching the frequency response of the infinite order model perfectly through the first two modes of the infinite order transfer function. If the user chooses a higher order approximation, for example a sixth order approximation, the algorithm will choose the maximum frequency of the fitting in a way that covers the first three modes. The following MATLAB command was used to utilize the "`linfn`" function for the hydraulic fracturing example to generate the plot in the figure 14:

```
>> linfn('FrackingTF',0.1,10000,4)
```

The "FrackingTF" input file example is available in appendix F.

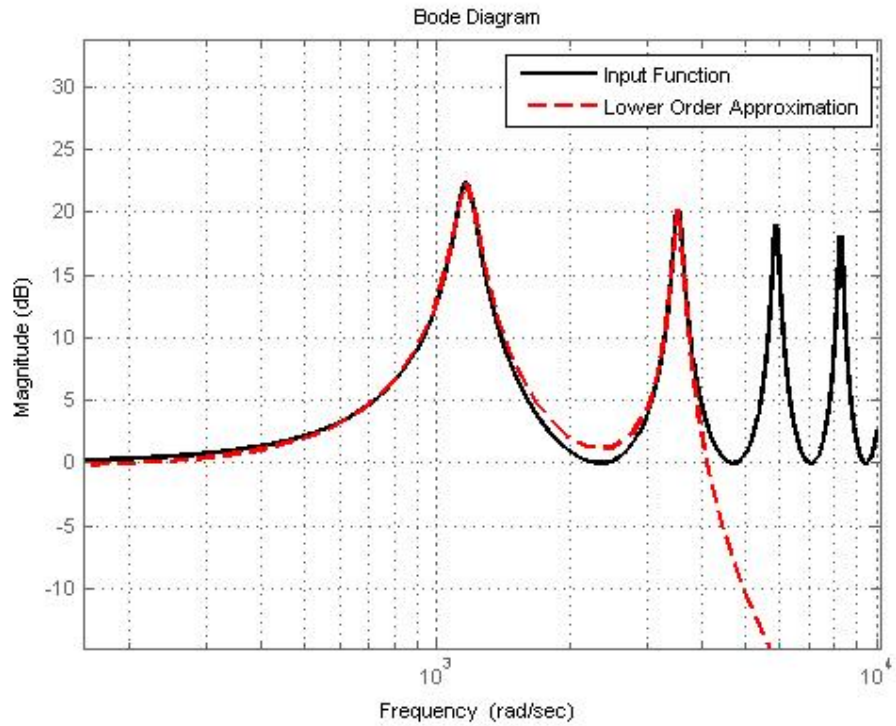


Figure 14 hydraulic fracturing example fourth order direct fit using `linfn` MATLAB function

In another example, the input function is the hydraulic brake system example equation in the paper by Huang, Hullender and Woods. Figure 15 shows the schematic for the hydraulic brake example.

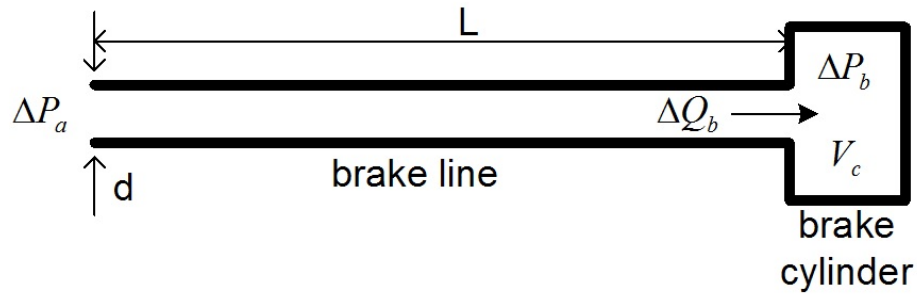


Figure 15 Brake example schematic

The equation 5 describes the pressure change in the brake cylinder with respect to changes in pressure at the start of the brake line due to an opening of a valve or changes in the brake fluid supply system.

$$\Delta P_b = \frac{1}{\frac{V_c \cosh \Gamma}{\beta + Z \sinh \Gamma}} \Delta P_a \quad (5)$$

In equation 5, V_c is the volume of the cylinder, and β is the bulk modulus. Using the equation 5 as a part of the input function for the M-file `linfn`, a fifth order approximation is generated for the equation 5. The M-file used as an input function to the "linfn" function for the hydraulic fracturing example is available in appendix G. Figure 16 shows the plot for the fifth order fit for the brake example. The following MATLAB command was used to utilize the "linfn" function for the hydraulic brake example to generate the plot in the figure 16:

```
>> linfn('BrakeTF',0.1,5000,5)
```

The "BrakeTF" input file example is available in appendix G.

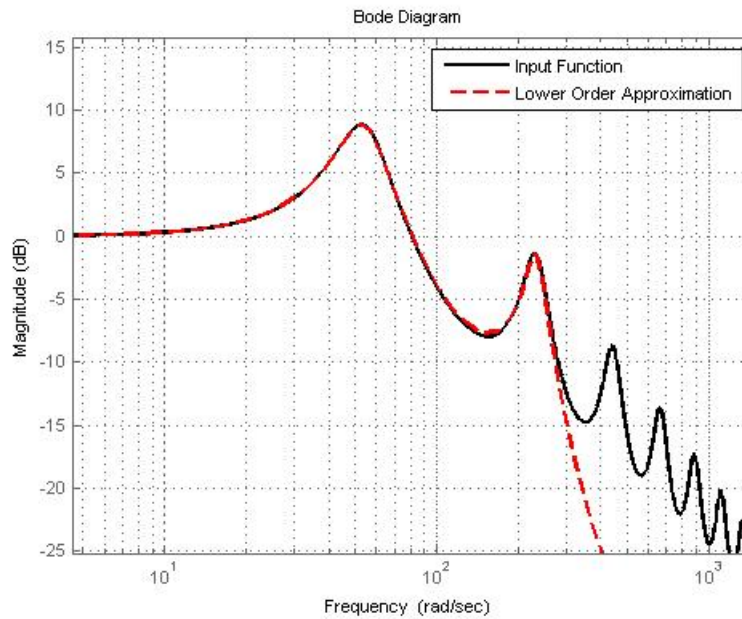


Figure 16 Brake example fifth order direct fit using linfn MATLAB function

Conclusion:

The method and the results in the chapter 2 and 3 of this study reveals the complication and difficulties associated with generating tables and choosing maximum frequency of fitting for components of the hydraulic fracturing example equation. Moreover, the results in chapter 3 revealed that the tables generated for hydraulic fracturing example equation components are only applicable for that form of equation, and these tables cannot be used for other equations with the same components but different arrangements. These limits were the motivation to develop a simple method to use the inverse frequency algorithm directly on the overall system transfer function instead of the components of the transfer function. The MATLAB function introduced in chapter 4 only requires the user to specify the order of approximation needed, and provides the name of the function to be approximated. An inexperienced user who

doesn't have knowledge of choosing the maximum frequency of fitting and distribution of frequency data points can easily utilize this MATLAB function, and get a lower order approximation of the input function very easily.

Appendix A

The MATLAB Code of the Function Generating $\frac{Zcosh\Gamma}{sinh\Gamma}$ Component's Transfer Function's

Data

```

lo=20; % loop iterations
Dnv=logspace(log10(0.00001),log10(0.0001),lo); % data point distribution
for v=1:lo

Dn=Dnv(v);

Bulk=1.8246e9;Den=855.24;KVis=7.6179e-6;
Wmin=0.001;
Wmax=960000;
c=sqrt(Bulk/Den); % Speed of sound, m/sec
DVis=KVis*Den; % Dynamic (/Absolute) viscosity

r=0.0254/8;
D=2*r;A=pi*r^2;
L=c*Dn*r^2/KVis; %calculate L to get the desired Dn

RR=128*DVis*L/(pi*D^4); % Resistance (Laminar flow)

syms s C1

B=2*besselj(1,1i*sqrt(s))/(1i*sqrt(s))*...
    besselj(0,1i*sqrt(s)); % s here is normalized, s_bar=r^2*s/KVis
Zo=Den*c/(pi*r^2); Z=Zo/sqrt(1-B);

Gamma=Dn*(s)/sqrt(1-B); % s here is normalized, s_bar

% Transfer function to be approximated: normalized Zcosh/sinh
% Pa=[Zcosh/sinh]Qa-[Z/sinh]Qb
H=C1*8*Dn^2/RR;% zero freq gain of sZcosh/sinh is RR/8Dn^2
H=subs(H,'C1',s*Z*cosh(Gamma)/sinh(Gamma));% multiplied by s to cancel s in denominator
%% Generating frequency points
NP=300;
w=genfreqs2(Wmin,Wmax,NP); % sub m-file, genfreqs.m, is used
N=length(w);
%NC=length(wc);
TF=[];TFc=[];
for k=1:N
    sw=1i*w(k); % Generate values for s.
    TF(k)=subs(H,s,sw); % Generate tf data points for the curve fit.
end

MTF=20*log10(abs(TF));

MTF1=smooth(MTF); %smoothing the data

%% local maximum
%MTF2=20*log10(abs(TF1));
%MTF3=smooth(MTF2);
[maxval,indmax]=lmax(MTF1,2);
frmax= w(indmax);

```

```

pp=1;
for q=1:length(frmax)
    if frmax(q) >= 10*Wmin
        frmax2(pp)=frmax(q);
        pp=pp+1;
    end
end
if length(frmax2) == 1
    maxi2=frmax2(1);
else
    maxi2=frmax2(2)
    maxi2=(frmax2(2)-0.04*(frmax2(2)-frmax2(1))); %the maximum %frequency is decided by
percentage of difference between first and %second local maximum of frequency response
magnitude
end

%% Frequency points for inver

Wmina=.001;
w2=genfreqs2(Wmina,maxi2,NP);
N2=length(w2);

TF2=[];TFc2=[];
for k=1:N2
    sw2=1i*w2(k); % Generate values for s.
    TF2(k)=subs(H,s,sw2); % Generate tf data points for the curve fit.
end

%%

wt2=ones(N2,1);
wt2=getInvFreqWeight(w2,TF2); % Weighting factors

%%
NO=5; %order
[num2,den2]=invfreqs(TF2,w2,NO-1,NO,wt2,100);
ZCHt1(v)=tf(num2,den2);
DCgainHt1(v)=dcgain(ZCHt1(v));
% adjust dc gain to 1
num2=num2/DCgainHt1(v);
ZCHt1(v)=tf(num2,den2);
n=length(den2);

P=pole(ZCHt1(v));
Mnum2=num2/(-P(1));
Mden2=conv([1 -P(2)],conv([1 -P(3)],conv([1 -P(4)],[1 -P(5)])));
Mnum2=Mnum2/real(Mden2(n-1));
Mden2=real(Mden2)/real(Mden2(n-1));
fittedZC(v)=tf(Mnum2,Mden2); % The transfer functions are saved here
end

```

Appendix B

Fit Objects For the Transfer Function Coefficients

```

function [NGZCoshOverSinh, GZCoshOverSinh] =
TFdataZCoshOverSinh(Den,Beta,Vis,d,L)
%% part 1
%clear all
%close all
r=d/2;
Mu=Vis*Den;
DVis=Vis;
Dnin=Vis*L*sqrt(Den/Beta)/r^2;
D=2*r;
RL=128*Mu*L/(pi*d^4);
Wv=Vis/r^2;
%%
load('Zcosh1e-5to1e-4.mat') %Saved transfer Function data values are %loaded here
tf1=fittedZC;

Dnv=logspace(log10(0.00001),log10(0.0001),20);

n=length(tf1);
for lo= 1: n

[num,den]=tfdata(tf1(lo),'v');

A4(lo)=num(1);
A3(lo)=num(2);
A2(lo)=num(3);
A1(lo)=num(4);
B4(lo)=den(1);
B3(lo)=den(2);
B2(lo)=den(3);
B1(lo)=den(4);

end

%% fit objects are created in this section
fA11=fit(Dnv,'A1','smoothingspline'); A11O=fA11(Dnin);
fA21=fit(Dnv,'A2','smoothingspline'); A21O=fA21(Dnin);
fA31=fit(Dnv,'A3','smoothingspline'); A31O=fA31(Dnin);
fA41=fit(Dnv,'A4','smoothingspline'); A41O=fA41(Dnin);
fB11=fit(Dnv,'B1','smoothingspline'); B11O=fB11(Dnin);
fB21=fit(Dnv,'B2','smoothingspline'); B21O=fB21(Dnin);
fB31=fit(Dnv,'B3','smoothingspline'); B31O=fB31(Dnin);
fB41=fit(Dnv,'B4','smoothingspline'); B41O=fB41(Dnin);

NGZCoshOverSinh=tf([A41O A31O A21O A11O 1],[B41O B31O B21O B11O 1 0]);

GZCoshOverSinh=tf([A41O/Wv^4 A31O/Wv^3 A21O/Wv^2 A11O/Wv 1]*RL/...
(8*Dnin^2),[B41O/Wv^5 B31O/Wv^4 B21O/Wv^3 B11O/Wv^2 1/Wv 0]);

en

```

Appendix C

New Tables for the $\frac{Z\cosh\Gamma}{\sinh\Gamma}$ Component's Transfer Function's Coefficients

```

function [NGZCoshOverSinh,GZCoshOverSinh] = ZCoshOverSinhTF(Den,Beta,Vis,d,L)
% NGZCoshOverSinh=[];
% GZCoshOverSinh=[];
format shortg

r=d/2;
Mu=Vis*Den;
Dn=Vis*L*sqrt(Den/Beta)/r^2;

RL=128*Mu*L/(pi*d^4);
Wv=Vis/r^2;

if Dn>0.5
    error('Dn is greater than 0.5 which is outside the range for the tabulated coefficients');
end
if Dn<=0.999999999e-5
    error('Dn is less than or equal to 1e-5 which is outside the range for the tabulated coefficients');
end

DN=[.9999999e-5 0.0000123 0.0000149 1.75e-5 2.0e-5 2.5e-5 2.8e-5 3e-5 4e-5 5e-5
7.5e-5 1e-4 1.2e-4 1.4e-4 2e-4 2.4e-4 3e-4 4e-4 5e-4 5.3125e-4...
5.625e-4 6.25e-4 7.5e-4 1e-3 1.25e-3 1.75e-3 2.5e-3 3e-3 4e-3 5e-3 ...
6e-3 7.5e-3 0.01 0.012 .015 0.0175 0.02 0.025 0.03 0.0325 0.035 0.0375 0.05 0.075
0.1 0.15 0.2 0.25 0.3 0.4 0.5];
A4=[1.8074e-022 4.1756e-022 8.7942e-022 1.6466e-021 2.9027e-021 6.9859e-021
1.1026e-020 1.0938e-20 4.6605e-020 1.136e-019 ...
5.4988e-019 1.815e-018 3.8148e-018 7.0248e-018 2.8676e-017 6.155e-017 1.5251e-
16 4.7601e-16 1.1385e-015 1.2767E-15 ...
1.5923E-15 2.8746e-015 5.9527e-015 1.9116e-014 4.6013e-014 1.8032e-013
7.6724e-013 1.5933e-012 5.1472e-012 1.2358e-11 ...
2.5926e-011 6.5288e-011 2.1507e-010 4.5246e-010 1.1185e-009 ...
2.1299e-009 3.6698e-009 9.2666e-009 1.9641e-008 2.7259e-008 3.7217e-008
4.9587e-008 1.6276e-007 8.9504e-007 ...
3.0102e-006 1.9115e-5 7.8567e-5 2.6864e-4 5.4481e-4 2.3728e-3 8.8698e-5];
A3=[2.7258e-019 6.0004e-019 1.0687e-018 1.7827e-018 3.4022e-018 6.45e-018
9.6533e-018 -8.8215e-20 3.8249e-017 7.6277e-017 ...
2.6869e-016 8.579e-016 1.7956e-015 2.8816e-015 9.1331e-015 2.0073e-014
4.3461e-14 1.1169e-13 2.2812e-013 1.0472E-13 ...
1.2077E-13 5.5569e-013 1.0428e-012 3.0902e-012 6.0864e-012 2.0737e-011 7.51e-
011 1.5842e-010 4.1532e-010 8.0502e-010 ...
1.545e-009 3.6385e-009 1.115e-008 2.0836e-008 4.5113e-008 ...
8.2156e-008 1.3797e-007 3.1743e-007 5.7555e-007 7.7427e-007 1.0552e-006
1.3608e-006 4.3462e-006 1.8694e-005 ...
5.4184e-005 2.0882e-4 7.2327e-4 0.0025004 0.0051553 0.019135 1.5492e-2];
A2=[4.4177e-011 6.7864e-011 9.7003e-011 1.3251e-010 1.801e-010 2.7322e-010
3.4327e-010 3.5557e-10 7.2067e-010 1.1043e-009 ...

```

2.4294e-009 4.4142e-009 6.5158e-009 8.8876e-009 1.751e-008 2.6095e-008
 4.0525e-8 7.1059e-8 1.1073e-007 1.1842E-07 ...
 1.3304E-07 1.767e-007 2.5435e-007 4.6156e-007 7.0653e-007 1.3985e-006 2.8998e-006
 4.2643e-006 7.5964e-006 1.163e-005 ...
 1.6867e-005 2.6852e-005 4.9502e-005 7.1427e-005 0.00011213 0.00015586
 0.00020656 0.0003303 0.00047601 ...
 0.00056271 0.0006624 0.00076578 0.0014244 0.0033527 0.0061812 0.015975 ...
 0.033723 0.062404 0.09418 0.22309 6.66695e-2];
 A1=[2.5057e-008 3.7192e-008 4.3021e-008 5.0778e-008 7.8134e-008 9.0283e-008
 1.0796e-007 -2.4709e-8 2.2128e-007 2.7366e-007 ...
 4.0246e-007 7.6443e-007 1.1544e-006 1.4018e-006 1.934e-006 3.2487e-006
 4.3372e-6 5.7452e-6 7.8966e-006 1.0854E-06 ...
 1.4457E-06 1.2868e-005 1.6777e-005 2.831e-005 3.4368e-005 5.9545e-005
 0.00010749 0.00015448 0.00023421 0.00027334 ...
 0.00036317 0.00055865 0.00097063 0.0012555 0.0017092 ...
 0.0022953 0.0029347 0.0042345 0.0052973 0.0060726 0.0071491 0.0080001
 0.013505 0.026248 ...
 0.042584 0.077152 0.14023 0.25631 0.36907 0.71874 0.81392];
 B4=[2.1075e-023 4.1965e-023 1.1061e-022 2.1994e-022 2.719e-022 8.9385e-022
 1.4006e-021 2.0945e-21 4.4543e-021 1.3793e-020 ...
 7.6218e-020 2.2402e-019 3.6719e-019 6.4252e-019 3.7888e-018 6.184e-018
 1.7679e-17 6.2297e-17 1.4512e-016 2.1464E-16 ...
 2.6359E-16 3.3115e-016 6.8866e-016 1.8562e-015 5.5703e-015 2.1531e-014
 8.5756e-014 1.3813e-013 5.0556e-013 1.5639e-012 ...
 3.269e-012 7.6715e-012 2.0057e-011 4.6816e-011 1.2488e-010 ...
 2.1368e-010 3.2677e-010 7.7656e-010 2.0647e-009 2.7721e-009 3.3963e-009
 4.5139e-009 1.0723e-008 6.8218e-008 ...
 2.5271e-007 1.8734E-06 0.000006021 0.000011108 0.000023639 ...
 0.000051464 1.8294e-9];
 B3=[5.621e-020 1.1389e-019 2.3552e-019 4.3132e-019 6.568e-019 1.4997e-018
 2.2294e-018 7.2692e-018 1.6542e-017 7.1217e-017 ...
 1.8726e-016 3.4084e-016 2.2287e-015 1.3683e-016 1.8311e-15 3.7676e-015
 9.0157e-15 2.7959e-14 5.3584e-014 7.1986E-14 ...
 8.0956e-14 1.1397e-013 2.1414e-013 5.8789e-013 1.322e-012 4.427e-012 1.5112e-011
 3.0471e-011 7.752e-011 1.8936e-010 ...
 3.631e-010 7.7731e-010 2.1284e-009 4.0805e-009 9.3764e-009 1.5968e-008
 2.5451e-008 5.9637e-008 1.1651e-007 ...
 1.51e-007 1.9466e-007 2.5081e-007 7.9861e-007 3.189e-006 9.1577e-006 ...
 0.000066712 0.00024507 0.00078121 0.0013009 0.0045048 4.9083e-4];
 B2=[1.2233e-011 1.8102e-011 2.7466e-011 3.8198e-011 4.7359e-011 7.7639e-011
 9.7331e-011 1.1446e-10 1.9027e-010 3.0889e-010 ...
 7.067e-010 1.2407e-009 1.7214e-009 2.3252e-009 5.0237e-009 6.9557e-009
 1.1159e-8 2.0257e-8 3.142e-008 3.6505E-08 ...
 4.0748E-08 4.8587e-008 7.009e-008 1.2181e-007 1.9713e-007 3.8815e-007 7.8972e-007
 1.0987e-006 2.0038e-006 3.269e-006 ...
 4.7321e-006 7.3803e-006 1.2839e-005 1.8929e-005 3.0244e-005 ...
 4.089e-005 5.2887e-005 8.33e-005 0.00012554 0.00014745 0.00016943 0.00019561
 0.0003447 0.00082421 ...
 0.0015395 0.0038347 0.0072286 0.012104 0.018575 0.03882 2.2687e-2];


```

B1=[1.1955e-008 1.646e-008 2.191e-008 2.829e-008 3.5234e-008 4.8494e-008
5.7505e-008 5.6917e-8 9.85e-008 1.3633e-007 ...
2.526e-007 3.8537e-007 5.1256e-007 4.2681e-007 1.1094e-006 1.4295e-006
2.0446e-6 3.3012e-6 4.3588e-006 4.8277E-06 ...
7.9791e-006 6.0856e-006 8.1879e-6 1.2573e-005 1.7376e-005 2.9293e-005 5.0043e-
005 6.9797e-005 0.00010285 0.00014785 ...
0.00019603 0.00027384 0.00043389 0.00056945 0.00081238 ...
0.0010345 0.0012815 0.0018691 0.0024528 0.0027547 0.0031105 0.0034769
0.0059298 0.010797 0.017047 ...
0.044685 0.083234 0.13742 0.17392 0.32726 8.9281e-2];

```

```

a4=interp1(DN,A4,Dn);a3=interp1(DN,A3,Dn);a2=interp1(DN,A2,Dn);a1=interp1(DN,A1,
Dn);
b4=interp1(DN,B4,Dn);b3=interp1(DN,B3,Dn);b2=interp1(DN,B2,Dn);b1=interp1(DN,B1,
Dn);

```

```

% fprintf('Normalized Transfer Function, 8Dn^2ZCosh/[RLSinh]')
NGZCoshOverSinh=tf([a4 a3 a2 a1 1],[b4 b3 b2 b1 1 0]);

```

```

GZCoshOverSinh=tf([a4/Wv^4 a3/Wv^3 a2/Wv^2 a1/Wv 1]*RL/...
(8*Dn^2),[b4/Wv^5 b3/Wv^4 b2/Wv^3 b1/Wv^2 1/Wv 0]);
end

```

Appendix D

New Tables for the $\frac{z}{\sinh \Gamma}$ Component's Transfer Function's Coefficients

```

function [NGZoverSinh,GZoverSinh] = ZoverSinhTF(Den,Beta,Vis,d,L )
% NGZoverSinh=[];GZoverSinh=[];
format short g
r=d/2;Mu=Vis*Den;Dn=Vis*L*sqrt(Den/Beta)/r^2 ; % Dissipation number
RL=128*Mu*L/(pi*d^4); % Steady state flow resistance
Wv=Vis/r^2 ; % Normalizing frequency, rad/sec

if Dn>0.5
    error('Dn is greater than 0.5 which is outside the range for the tabulated coefficients');
end
if Dn<=0.999999999e-5
    error('Dn is less than or equal to 1e-5 which is outside the range for the tabulated
coefficients');
end

DN=[.999999999e-5 0.0000123 0.0000149 1.75e-5 0.000025 0.000028 0.00004 0.00005
0.000075 0.0001 ...
0.00012 0.00014 0.0002 0.00024 .0003 .0004 .0005 .000625 .00075 .000875 ...
0.001 .00125 .002 0.0025 .003 .004 0.005 .006 0.0075 .008 ...
.009 0.01 0.012 0.015 .0175 0.02 0.025 0.03 0.0325 0.035 .0375 0.05 0.075 0.0875
0.094 0.1 ...
0.1125 0.125 0.1375 0.15 0.175 0.2 0.215 0.225 0.25 0.275 0.3 0.4 0.45 0.5];
A3=[1.312e-019 1.945e-019 3.827e-019 1.013e-018 1.8628e-018 2.2231e-018 7.8934e-
018 1.717e-017 8.7531e-017 2.8628e-016 ...
4.7039e-016 8.7249e-016 5.241e-15 6.2725e-015 1.6043e-014 2.8178e-14 1.0464e-
013 1.7212e-013 3.8587e-013 5.086e-14 ...
9.5279e-13 3.2871e-12 1.423e-11 3.0054e-11 4.949e-11 8.3464e-011 1.8781e-010
8.5115e-10 1.3224e-9 1.9478e-9 ...
3.1992e-9 3.9679e-9 7.7303e-009 1.6198e-008 2.7117e-008 2.5888e-008 5.8571e-
008 1.4473e-007 1.7595e-007 2.1629e-007 2.8267e-007 6.7532e-007 2.9045e-006
7.9063e-6 ...
1.0449e-5 1.2583e-5 1.9094e-5 2.7134e-5 3.7502e-5 5.185e-5 9.851e-5 ...
1.6846e-4 0.00015593 0.00029794 2.9e-4 1.5898e-4 0.00053852 8.848e-4 .0011349
.0011513];
A2=[-6e-012 -7.754e-012 -1.2e-011 -3.574e-011 -3.1923e-011 -3.2667e-011 -6.9201e-
011 -1.054e-010 -3.0003e-010 -5.5151e-010 ...
-7.4237e-010 -1.0655e-009 -2.7155e-9 -3.2592e-009 -5.6473e-9 -8.2351e-9 -1.6437e-
008 -2.2607e-008 -3.6843e-008 -4.1006e-8 ...
-5.8649e-8 -1.2826e-7 -2.7098e-7 -4.2738e-7 -5.2434e-7 -7.4136e-007 -1.2071e-006 -
2.3246e-006 -3.977e-6 -5.1135e-6 ...
-6.9593e-6 -7.5531e-6 -1.0009e-005 -1.669e-005 -2.272e-005 -2.1424e-005 -3.5605e-
005 -6.1517e-005 -6.9438e-005 -7.7788e-005 ...
-9.2056e-005 -0.00014948 -0.00036778 -6.7473e-4 -7.9662e-4 -8.9249e-4 -1.1419e-3
...
-1.4011e-3 -1.6821e-3 -1.9974e-3 -2.7501e-3 -3.5422e-3 -0.0035806 -0.0045154 -
4.6697e-3 ...
-3.625e-3 -0.0058645 -6.5903e-3 -.0050703 -.0020768];
A1=[8.66e-009 6.641e-009 9.783e-009 3.226e-008 1.3122e-008 6.6813e-009 1.3417e-
008 1.72e-008 7.116e-008 1.5057e-007 ...

```

1.46e-007 8.9978e-7 2.216e-007 5.8478e-007 1.0929e-6 6.7799e-7 2.7178e-006
 2.2591e-006 4.1618e-006 2.9032e-6 ...
 5.1312e-6 1.6212e-5 2.2747e-5 2.9428e-5 2.8521e-5 1.1455e-005 1.8303e-005
 9.1169e-005 1.3856e-4 2.0799e-4 ...
 2.8644e-4 2.4907e-4 0.00029946 0.00039067 0.00046459 0.00012807 0.00021186
 0.00060269 0.00053345 0.0004802 0.00058165 ...
 0.00041565 0.00076965 4.2226e-3 4.8285e-3 4.6969e-3 5.085e-3 ...
 4.9452e-3 4.6868e-3 4.7561e-3 6.5592e-3 8.8334e-3 -0.00031767 0.018141 ...
 -2.5887e-3 -.021021 -0.0085529 -.046738 -.070166 -.10371];
 B4=[2.666e-023 6.299e-023 1.341e-022 2.424e-022 1.0974e-021 1.9086e-021 7.802e-
 021 1.933e-020 8.9638e-020 2.7555e-019 ...
 5.8511e-019 1.0688e-018 4.2483e-18 9.1911e-018 2.2125e-017 7.4651e-17 1.7093e-
 016 4.3044e-016 8.7683e-016 1.7293e-15 ...
 2.884e-15 6.6938e-15 4.5115e-14 1.1038e-13 2.3097e-13 8.5528e-013 2.0764e-012
 3.7182e-12 9.5737e-12 1.2135e-11 ...
 1.9468e-11 3.0411e-11 6.334e-011 1.5729e-010 2.96e-010 5.8258e-010 1.4218e-009
 2.8063e-009 3.9607e-009 5.4662e-009 7.1898e-009 2.4982e-008 1.3065e-007 ...
 2.2412e-7 3.0012e-7 3.8894e-7 6.3074e-7 9.7325e-7 1.4378e-6 ...
 2.0407e-6 3.7506e-6 6.3872e-6 9.144e-006 1.0259e-005 1.6248e-5 3.0509e-5 ...
 3.3921e-5 1.1688e-4 .00018201 .00027347];
 B3=[3.478e-020 6.939e-020 1.412e-019 3.342e-019 8.6052e-019 1.1902e-018 4.2614e-
 018 9.165e-018 4.0875e-017 1.1097e-016 ...
 2.0938e-016 3.6384e-016 1.3706e-15 2.4115e-015 5.3697e-015 1.458e-14 3.2492e-
 014 7.0632e-014 1.3712e-013 2.2546e-13 ...
 3.659e-13 9.1883e-13 4.4724e-12 9.9672e-12 1.7938e-11 4.7498e-011 1.0694e-010
 2.5474e-10 4.9855e-10 6.6092e-10 ...
 1.0407e-9 1.4443e-9 2.6536e-009 6.2065e-009 1.0815e-008 1.5838e-008 3.6794e-
 008 7.3905e-008 9.8811e-008 1.2811e-007 1.6725e-007 4.7206e-007 2.2178e-006 ...
 4.2328e-6 5.5882e-6 7.0208e-6 1.0997e-5 1.6355e-5 2.3429e-5 ...
 3.2655e-5 5.9464e-5 9.953e-5 0.00012604 0.00015899 2.2156e-4 3.3114e-4 ...
 4.4059e-4 1.3172e-3 .0019912 .0028388];
 B2=[1.278e-011 1.947e-011 2.85e-011 3.892e-011 8.084e-011 1.037e-010 2.1082e-010
 3.306e-010 7.3007e-010 1.2911e-009 ...
 1.8694e-009 2.5379e-009 5.1406e-9 7.4642e-009 1.1639e-8 2.1005e-8 3.2416e-008
 5.104e-008 7.3312e-008 1.0121e-7 ...
 1.3166e-7 2.0404e-7 5.2748e-7 8.2579e-7 1.1952e-6 2.2069e-006 3.4554e-006
 4.8686e-006 7.6805e-6 8.7167e-6 ...
 1.1069e-5 1.3762e-5 1.9916e-005 3.1431e-005 4.3126e-005 5.8249e-005 ...
 9.1726e-005 0.00013174 0.00015589 0.00018233 0.00020993 0.00038607
 0.00089666 1.2217e-3 1.4202e-3 ...
 1.6178e-3 2.0719e-3 2.584e-3 3.1538e-3 3.7793e-3 5.2043e-3 6.8943e-3 0.0081137
 ...
 0.0089566 .011056 .014172 .016313 .030686 .038805 .04767];
 B1=[9.536e-009 1.264e-008 1.728e-008 2.75e-008 3.6898e-008 4.1448e-008 7.1898e-
 008 9.959e-008 1.9317e-007 2.9841e-007 ...
 3.8973e-007 4.9673e-007 9.1115e-7 1.1204e-006 1.5877e-6 2.3973e-6 3.4437e-006
 4.7776e-006 6.3977e-006 7.8248e-6 ...
 9.7332e-6 1.5169e-5 2.935e-5 4.1522e-5 5.2255e-5 7.6888e-005 0.00010964
 0.00015583 2.244e-4 2.5832e-4 ...
 3.1857e-4 3.6119e-4 0.00046417 0.00068213 0.00086707 0.00096219 ...

Appendix E
linfn MATLAB function

```

function linfn(funName,wmin,wmax,order)
%
% linfn(funName,wmin,wmax,order)
% Generates a transfer function approximation of a function, funName;
% The input function is in frequency domain format and is generated in a separate M-file
% the name of transfer function has to be
% specified in string format, for example: 'BrakeTF'
% The approximation is achieved by curve fitting the original transfer
% function frequency response over the specified frequency range.
% The user has to specify a range of wmin rad/sec to wmax rad/sec
% range of frequency initially.
% The maximum frequency needs to be specified as an input for generating frequency
% response of the original function.
% The program will produce an error if the maximum frequency is too low
% for the order of approximation specified.
% Once a maximum frequency is specified the program will decide on the maximum
% frequency needed for curve fitting based on the order of approximation.
% The user needs to choose a min frequency initially, and base on the
% output results, choose a lower min frequency if the approximation
% fails to fit the original function at low frequencies.
%
%file name "linfn.m"
%funName= input transfer function in string format
%wmin=minimum frequency (rad/sec) for a good approximation
%wmax=maximum frequency (rad/sec) for generating frequency response
%order=desired order of the approximation

func=str2func(funName); % converts the string function name to function format

mm=floor(order/2); %modes

%%

w=logspace(log10(wmin),log10(wmax),10000);
N=length(w);
H = zeros(N,1);
for k=1:N
    s=w(k)*1i;
    H(k) = func(s); % Generate tf data points for the curve fit.;
end

MTF=20*log10(abs(H));

MTF1=smooth(MTF); %smoothing the data
%%
%the maximum frequency is decided here.
%there is no need to specify the maximum frequency accurately; This part

```

```

%calculates the maximum frequency of the fitting using the order of
%approximation
[maxval,indmax]=lmax(MTF1,2);
frmax= w(indmax);
pp=1;
frmax2=[];
for q=1:length(frmax)
    if frmax(q) >= 10*wmin
        frmax2(pp)=frmax(q);
        pp=pp+1;
    end
end

if isempty(frmax2)
    error('The maximum frequency is too low.')
    RETURN
end

if length(frmax2) == 1
    maxi2=frmax2(1);
else
    maxi2=frmax2(mm)
    maxi2=(frmax2(mm)+0.1*(frmax2(mm)-frmax2(mm-1)));
end

%%
w2=logspace(log10(wmin),log10(maxi2),100*order);
N2=length(w2);
for k=1:N2
    s=w2(k)*1i;
    H2(k) = func(s); % Generate tf data points for the curve fit.;
end
weight=ones(N2,1);
[numa,dena]=invfreqs(H2,w2,order-1,order,weight,30); %30 iterations
Ht1=tf(numa,dena)
figure
semilogx(w,MTF1,'k','LineWidth',2)
hold on;bodemag(Ht1,'r--')
legend('Input Function','Lower Order Approximation')
h = findobj(gcf,'type','line');
set(h,'linewidth',2);
grid on
grid minor
damp(Ht1)
end

```

Appendix F

FrackingTF input example function for linfo MATLAB function


```

function resp = FrackingTF(s)
% This function takes one frequency as an input
% and returns a response to that frequency.

Beta=1.8246e9; %bulk modulus N/m^2
Den=855.24; %density Kg/m^3
KVis=7.6179e-6; %kinematic viscosity m^2/s
DVis=KVis*Den; % absolute or dynamic viscosity Ns/m^2
r = 0.003175;
c = sqrt(Beta/Den); % Speed of sound, m/sec

%Dn=0.001;
%L=c*Dn*r^2/KVis;
L=1.9328;
RL = 8*DVis*L / (pi*r^4); % Resistance (Laminar flow)
Dn = KVis * L / (c * r^2);

s_bar = r^2 * s / KVis;
B = 2*besselj(1, 1i*sqrt(s_bar)) / (1i*sqrt(s_bar) * besselj(0, 1i*sqrt(s_bar)));
Zo = Den*c / (pi*r^2);
Z = Zo / sqrt(1-B);
Gamma = Dn*s_bar / sqrt(1-B);

resp = Z / (6*RL*sinh(Gamma) + Z*cosh(Gamma));
end

```

Appendix G

BrakeTF input example function for linfo MATLAB function

```

function resp = BrakeTF(s)
% This function takes one frequency as an input
% and returns a response to that frequency.

Beta = 1.8246e9;% bulk modulus N/m^2
Den = 855.24; % density Kg/m^3
r = 0.0015875;
KVis = 7.6179e-6; % kinematic viscosity m^2/s
L = 20;
Vc = L*pi*r^2; % Assume capacitance volume is equal to line volume
c = sqrt(Beta/Den); % Speed of sound, m/sec

Dn = KVis * L / (c * r^2);

s_bar = r^2 * s / KVis;
B = 2*besselj(1,1i*sqrt(s_bar))/(1i*sqrt(s_bar)*besselj(0,1i*sqrt(s_bar)));
Zo = Den*c/(pi*r^2);
Z = Zo/sqrt(1-B);
Gamma = Dn*(s_bar)/sqrt(1-B);

resp = 1 / (Vc*s/Beta * Z*sinh(Gamma) + cosh(Gamma));
end

```

References

- [1] Yi-Wei Huang, David Hullender, Robert Woods, "A Simplified Approach to Accurately Analyzing the Effects of Fluid Transients in Lines", 2014
- [2] "Imax" MATLAB function is developed by Serge Koptenko, Guigne International Ltd.

Biographical Information

Sina started his academic career as an undergraduate student in mechanical engineering in 2008. The focus of his undergraduate program was on heat transfer and fluids; He later worked on a project to evaluate the energy efficiency of his university's building. Sina participated in summer internships in two of the major engineering companies in Tehran. He later started his graduate studies in mechanical engineering in University of Texas Arlington. As a masters student, he took courses in modeling and controls, and worked on a project for modeling fluid lines. His career goal is to do modeling and consulting for engineering projects.