Rochester Institute of Technology

RIT Digital Institutional Repository

Theses

2000

Machine tool spindle design

Jamie Hoyt

Follow this and additional works at: https://repository.rit.edu/theses

Recommended Citation

Hoyt, Jamie, "Machine tool spindle design" (2000). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

MACHINE TOOL
SPINDLE DESIGN

by

Jamie A. Hoyt

A Thesis Submitted in
Partial Fulfillment of the
Requirements for the

MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

Approved by:

Dr. Wayne W. Walter Department of Mechanical Engineering	________________ Thesis Advisor)
Dr. Hany A. Ghoneim Department of Mechanical Engineering	________________
Dr. Kevin B. Kochersberger Department of Mechanical Engineering	________________
Dr. Satish Kandlikar Department of Mechanical Engineering	________________

DEPARTMENT OF MECHANICAL ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY

MAY, 2000

Permission granted

Title of Thesis: Machine Tool Spindle Design

I, Jamie A. Hoyt, hereby grant permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date: 5/17/00 Signature of Author: ____________________

Abstract

In modern machine tool applications the performance of a machine tool is judged by its ability to produce work-pieces accurately and efficiently. The stiffness of the machine tool spindle has a profound impact on the overall machine performance. The work presented here provides a tool for machine tool spindle designers to develop spindles that are sufficiently stiff to meet their needs. The analysis presented here is divided into three main sections.

The first portion is a static analysis. The static analysis calculates the lateral deflection of the spindle-bearing system. A *Matlab* program was developed that allows the user to enter the spindle parameters into a batch file and obtain the plots of the deformed shape of the spindle.

The next portion is a dynamic analysis of the spindle. This portion includes both the modes of vibration and the forced response. The modal analysis treats the spindle as a continuous Euler-Bernoulli beam. A numerical method for handling the steps in the shaft and applied boundary conditions was developed that could be extended to many other applications in rotor dynamics. A *Matlab* program was developed for the dynamic analysis. This program provides a designer with plots of the mode shapes and forced response given the spindle design parameters.

The final section is an optimization of the spindle design. Given constraints on the location and stiffness of the support bearings, a *Matlab* program will return values for these parameters resulting in the spindle configuration that presents the minimum deflection at the spindle's gauge line.

Table of Contents

List of Tables

List of Figures

Nomenclature

Symbol

A	Cross sectional area of beam [in^2]
A	Gradient vector of inequality constraint [dependent on constraint]
a_1	Location of drive pulley [in]
a_2	Location of rear bearing [in]
a_3	Location of front bearing [in]
a_4	Location of gauge line [in]
b_k	Location of kth joint in spindle shaft [in]
c	Gradient vector [in/in]
d	Vector of design changes [unitless]
$g_i(\mathbf{x})$	Ith inequality constraint [dependent on contraint]
D	Distance between support bearings [in]
E	Young's modulus [psi]
f	Quadratic subproblem [unitless]
F_c	Cutting force [lbf]
F_{cub}	Unbalance force due to cutting tool [lbf]
F_d	Drive force [lbf]
F_{de}	Equivalent drive force [lbf]
F_{dub}	Unbalance force due to unbalance of drive pulley [lbf]
$f(\mathbf{x})$	Cost function [in]
I_n	Moment of inertia of nth beam segment [in^4]

K_f Lateral stiffness of front bearing [lbf/in]

K_{fmax} Maximum lateral stiffness of front bearing [lbf/in]

K_{ft} Torsional stiffness of front bearing [in-lbf]

K_i Generalized stiffness [lbf/in]

K_r Lateral stiffness of rear bearing [lbf/in]

K_{fmax} Maximum lateral stiffness of rear bearing [lbf/in]

M(x) Bending moment in spindle shaft [in-lbf]

$M_{applied}$ Externally applied moments [in-lbf]

M_b Reaction moment at front bearing [in-lbf]

M_{be} Equivalent reaction moment at front bearing [lbf]

M_i Generalized mass [lbf-s^2/in]

M_k Moment induced at kth joint in spindle shaft [in-lbf]

M_T Moment at guage line due to cutting force and tool length [in-lbf]

OH Cantilever, distance between front bearing and gauge line [in]

q_i Generalized coordinate [in]

Q_i Generalized force [lbf]

R Penalty parameter [unitless]

R_f Reaction force at front bearing [lbf]

R_{fe} Equivalent reaction force at front bearing [lbf]

R_r Reaction force at rear bearing [lbf]

R_{re} Equivalent reaction force at rear bearing [lbf]

s_i Slack variable for ith constraint [unitless]

T	Kinetic energy [in-lbf]
t_j	Step size [dependent on design variable]
$[T_{i,j}]$	Transformation matrix between ith and jth beam segments [unitless]
ti	Length of cutting tool [in]
u	Strain energy [in-lbf]
U	Potential energy [in-lbf]
u_i	Lagrange multiplier [unitless]
V	Maximum constraint violation [dependent on constraint]
V(x)	Shear force in spindle shaft [lbf]
V_k	Shear force induced at kth joint in spindle shaft [lbf]
x	Axial position along shaft [in]
$\mathbf{x}$	Vector of design variables
x_0	Fraction of moment exerted by front bearing [unitless]
y_b	Elastic deflection of spindle shaft [in.]
y_s	Elastic deflection of spindle shaft [in.]
δ_f	Deflection at front bearing [in]
δq_i	Virtual displacement [in]
δ_r	Deflection at rear bearing [in]
δw_i	Virtual work [in-lbf]
ε_1	Convergence criteria [unitless]
ε_2	Maximum allowable constraint violation [unitless]
ϕ_i	Ith normal mode [unitless]

Φ	Descent function [in]
ρ	Mass density [lbf-s^2/in^4]
ω	Circular frequency [rad/sec]

1.0 Introduction:

Great demands are placed on the capabilities of today's modern machine tools to produce parts that are dimensionally correct with increasing accuracy and throughput. Some of the machine tool components that impact the accuracy and throughput of the machine are the drive systems, way systems, control and feedback systems, and finally the machine tool spindle. The machine tool spindle is the element of the machine that either supports the work-piece or the cutting tool. In addition to being a support structure, the spindle also rotates at high rates of speed to provide relative motion between the work-piece and the cutting tool. Therefore the spindle has a direct impact on both the throughput (material removal rate), and the accuracy of the finished part.

According to Lewinschal (1985), the most common requirements of a machine tool spindle are:

- High running accuracy
- High speed capability
- Great stiffness
- Low and even running temperature
- Minimum need of maintenance

Often in machine tool spindles these parameters will conflict with each other. In order to achieve a higher speed capability the designer must trade off spindle stiffness for speed or visa-versa. The spindle designer must carefully weigh the requirements of the user to determine the best possible balance of these parameters.

The goal of this research is to provide a tool for a spindle designer to aid in the evaluation of the spindle stiffness. High running accuracy, high operating speed capability, low and even running temperature, and minimum need of maintenance are typically functions of the bearing's geometry, manufacturing, lubrication, and method of mounting. If the spindle designer is able to quantify the stiffness requirements for the bearing he can then work with the bearing manufacturer to select the proper bearings for the application.

Al-Shareef et al. (1990) developed a quasi-static method of analyzing machine tool spindles. Their analysis takes the amplitude of the dynamic forces and applies them to a static model of the spindle-bearing system. For the static analysis the deflection contribution of the spindle shaft and the deflection contribution of the spindle support bearings are superimposed to obtain the total deflection of the system.

The static analysis of the spindle shaft assumes that a stepped flexible shaft is pinned in the location of the support bearings. The analysis of this flexible shaft consists of a transformation from a stepped shaft to a uniform shaft. This transformation yielded additional shear and bending moments at each of the joints in the shaft. The resulting uniform shaft was analyzed using classical mechanics.

The deflection contribution of the spindle support bearings assumes a rigid shaft supported by linear springs. The reaction forces yielded the deflection at each of the springs. Essentially, the deflection contribution of the bearings is a straight line fit between the resulting deformed positions of the springs.

In addition to the static analysis an optimization of the deflection at the end of the spindle was presented. The optimization analysis consisted primarily of varying the spindle design parameters and looking at the effect on the resulting deflection at the spindle gauge line. Plots were presented illustrating the effect of the variation of these parameters. The following conclusions were drawn from these plots.

- In the design of a spindle there exist an optimum ratio of the bearing spacing to the overhang of the spindle. As the flexural stiffness increases and the ratio of front to rear bearing stiffness decreases the optimum bearing-overhang decreases.
- A dimensionless flexural stiffness ($K_f(OH)^3/EI$) of greater than 5 results in minimum deflection at the cutting tool. The deflection at the end, or gauge line, of the spindle is very sensitive to the flexural stiffness for magnitudes less than 5.
- Having more than 3 steps in the shaft is desirable for obtaining minimal deflection values.
- The magnitude, position, and direction of the driving force greatly effects the deflection at the gauge line. For each scenario there exists an optimum location of the drive pulley.

In Lewinchal (1983) a similar study on the variation of spindle design parameters was presented. Plots were generated that illustrated the effect of the bearing spacing-overhang ratio on the spindle stiffness for support bearings of varying stiffness. From these plots it could be concluded that for very stiff support bearings the optimum spacing between the bearings becomes shorter. It could also be concluded that if the spindle has a long overhang the stiffness of the bearings has a lesser impact on the stiffness of the spindle.

Other work in the optimum design of machine tool spindles was also done in Montusiewicz et al. (1997). In this work a model of a machine tool spindle supported by hydrostatic bearings was presented. The study consisted of applying a four-stage multicriterion optimization strategy to a static model of a spindle. The objective of the analysis was to reduce the radial and axial deflection of a spindle, the total mass of the spindle, the total power loss of the bearings, and finally the size of the bearings. The analysis divides the spindle system into four subsystems. Each of these systems are optimized locally, and finally integrated to provide a global optimization. The outcome of this analysis was a computer aided optimum design package. This package allows spindle designers to interactively design an optimum spindle, inputting required design variables throughout the optimization process.

A qualitative dynamic analysis of a machine tool spindle was presented in Al-Shareef et al. (1991). Traditionally in the dynamic analysis of machine tool spindles the first mode is thought to be responsible for poor cutting quality. The purpose of this work was to assess this assumption. There was concern that this would not be the case since the range of operating frequencies for a given spindle often excite the higher modes. The first four modes for an example spindle were solved for analytically and compared to experimental results. The modal analysis presented ignores damping and rotational affects. The authors site an experimental study that proved there to be little difference between the non-rotational natural frequencies and the rotational critical speeds. By looking at the individual mode shapes they found that the first mode contributed the most to the deflection at the tool to work-piece interface. All other modes in the operating frequency range exhibited nodal characteristics at this interface. Since the excitation force would be exerted here they concluded that the first mode would indeed be most accountable for poor cutting quality. However they also noted that at the higher modes there was significant deflection at the location of the support bearings. This could result in the degradation of these bearing and an eventual loss of spindle stiffness.

Some other works, pertaining more generally to the field of rotor dynamics, were also researched. Two of these works deal primarily with the extension of the conventional transformation matrix (CTM) technique. In the work done by Curti et al. (1993) an expression for an 8 x 8 dynamic stiffness matrix of a rotating Timoshenko beam is derived and related to the conventional 4 x 4 dynamic stiffness matrix. This provides for the inclusion of anisotropic supports.

In work done by Murphy (1993) a polynomial transfer matrix was developed to replace the conventional transfer matrix for modal and forced response analyses. The advantage of the polynomial transfer matrix is an increase in computational speed of 3.5 to 100 times over the conventional transfer matrix. Example problems were analyzed using both the CTM and PTM methods as well as a finite element analysis. The results for all three cases were identical and the speed of the PTM method was considerably faster.

2.0 Static Analysis:

The static analysis calculates the lateral deflection of the spindle. Figure 2.1 illustrates the model under scrutiny. The following assumptions were necessary to perform the analysis:

1. The spindle shaft is assumed to be an Euler-Bernoulli Beam.
2. The spindle is subjected to a cutting force, a drive force, and the reaction forces at the bearings. The drive force must be applied behind the rear bearing.
3. The torsional and axial deflections of the spindle shaft are neglected.
4. The centerline of the spindle shaft is exactly inline with the centerline of the bearing bores. There is no contribution to the lateral deflection due to manufacturing misalignment.
5. The spindle housing and the cutting tool are both assumed to have an infinite stiffness.
6. It is assumed that the spindle is supported by only two bearings. This is common for most machine tool spindles. Manufacturability precludes the use of more than two bearings in most spindles.
7. The contribution of transverse shear deformation to the overall lateral deflection is assumed to be negligible. It was observed in a study conducted by Al-Shareef and Brandon, that the contribution of shear deformation is dependent on the ratio between the length of the spindle and the spindle nose overhang. The shear deflection for short spindles with small overhangs contributes more to the overall

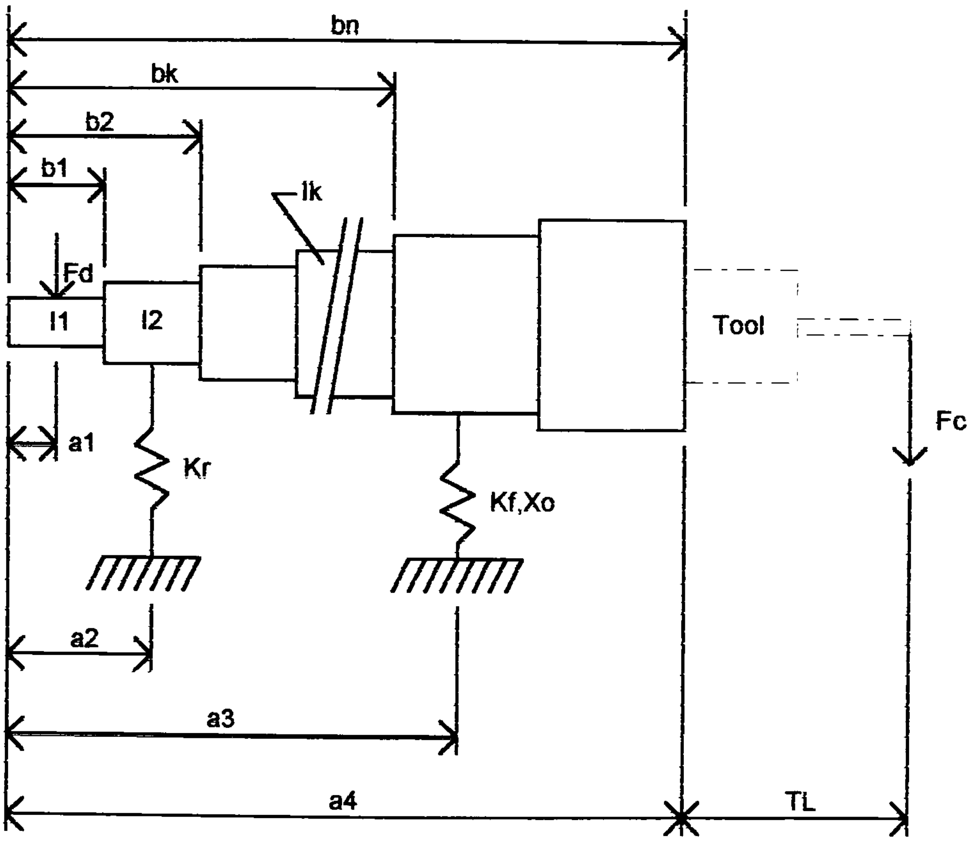

Figure 2.1 Spindle Model

deflection than longer, more slender spindles. A variety of spindles were analyzed in this study and a maximum contribution of 12 per cent was found (Al-Shareef et al., 1990)

Superposition was employed to calculate the lateral deflection of the spindle. The elastic deformation of the spindle shaft, y_s and the deflection of the spindle bearings, y_b were superimposed to calculate the overall deflection of the spindle (see figs. 2.2a and 2.2b). Equation 2.1 gives the overall deflection of the spindle.

$$y_t = y_s + y_b \tag{2.1}$$

2.1 Deformation of Elastic Shaft:

For the elastic contribution of the spindle shaft Al-Shareef and Brandon propose a method to transform the stepped spindle shaft to a uniform shaft (Al-Shareef et. al, 1990). This approach will be employed in this analysis. When the shaft is transformed there is a moment, M_k and shear force, V_k induced at each step in the shaft (fig. 2.3). In addition the applied forces and reactions must be transformed into equivalent forces applied to beam segments with larger bending moments of inertia. These equivalent forces are noted using the subcript "e" (i.e. $F_d \rightarrow F_{de}$).

The deflection of the uniform beam can be easily analyzed using conventional beam theory and singularity functions. The singularity functions will be represented by expressions in < >. If the value of the expression within these brackets is less than zero the function becomes zero (i.e. $<2-4>^2 = 0$). If the value of the expression is greater than zero, the function simply becomes the expression within the brackets (i.e. $<4-2>^2 = (4-2)^2$).

The shear force, V(x) of the uniform beam can be found to be:

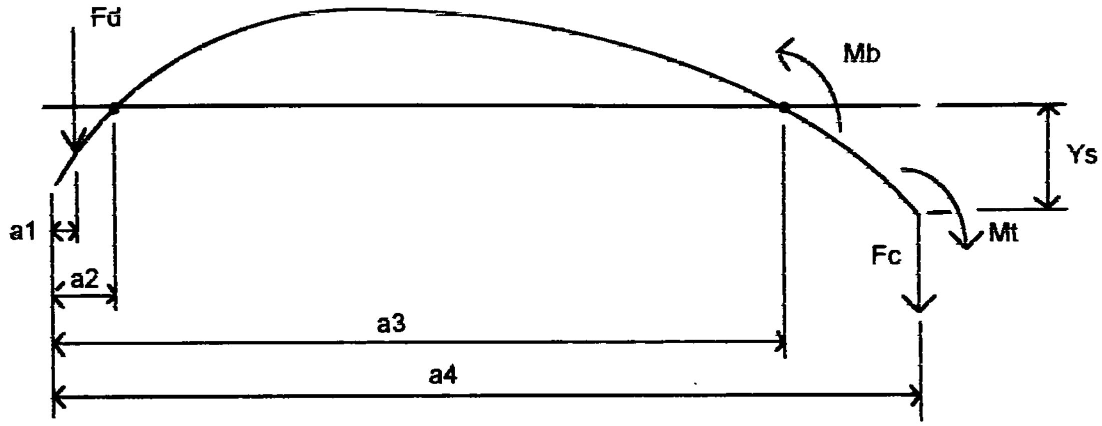

Figure 2.2a Elastic Deflection of Spindle Shaft

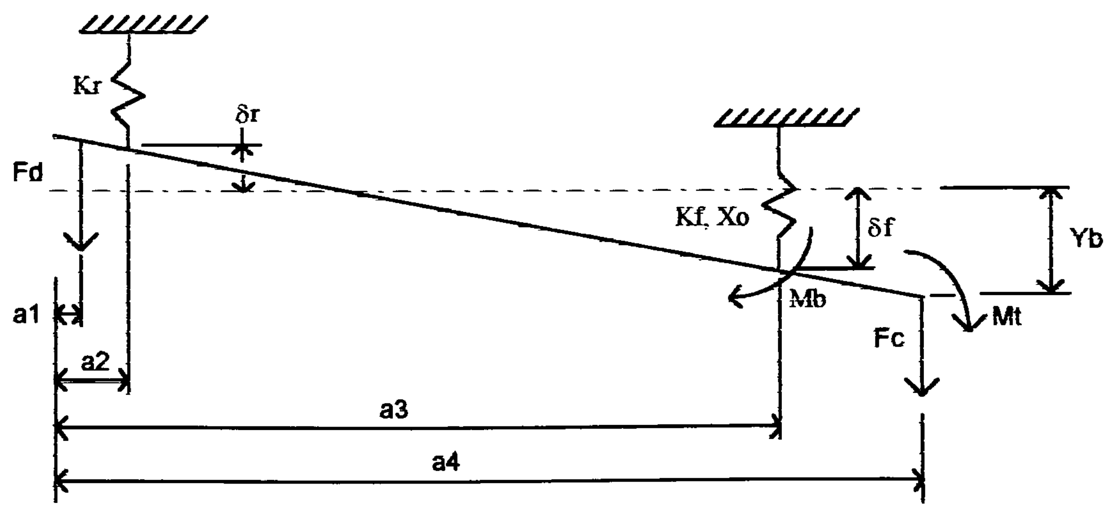

Figure 2.2b Deflection of Spindle Bearings

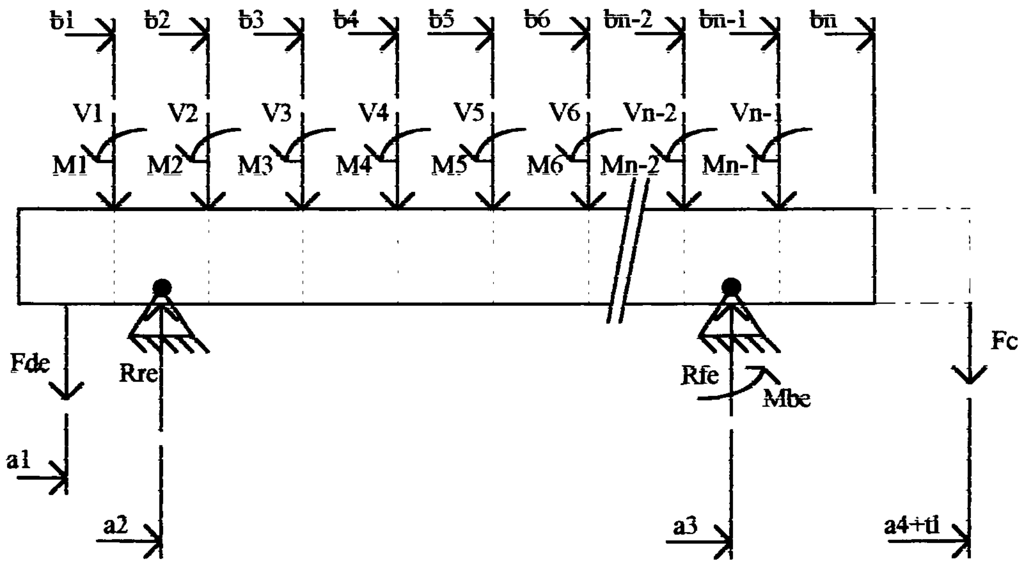

Figure 2.3 Model of Uniform Beam

$$
\begin{aligned}
V(x) &= F_{de}\langle x-a_1\rangle^0 + Fc\langle x-a_4-tl\rangle^0 - R_{re}\langle x-a_2\rangle^0 \\
&- R_{fe}\langle x-a_3\rangle^0 + \sum_{k=1}^{n} V_k\langle x-b_k\rangle^0
\end{aligned}
\tag{2.2}
$$

The moment of the beam, M(x) becomes:

$$
M(x) = \int V(x)dx + M_{applied} \tag{2.3}
$$

$$
\begin{aligned}
M(x) &= F_{de}\langle x-a_1\rangle^1 + F_c\langle x-a_4-tl\rangle^1 - R_{re}\langle x-a_2\rangle^1 - R_{fe}\langle x-a_3\rangle^1 \\
&+ \sum_{k=1}^{n} V_k\langle x-b_k\rangle^1 + \sum_{k=1}^{n} M_k\langle x-b_k\rangle^0 + M_{be}\langle x-a_3\rangle^0
\end{aligned}
\tag{2.4}
$$

The slope of the beam, θ(x) becomes:

$$
\theta(x) = \frac{1}{EI_n}\int M(x)dx \tag{2.5}
$$

$$
\theta(x) = \frac{1}{EI_n}\left(
\begin{aligned}
&\frac{F_{de}}{2}\langle x-a_1\rangle^2 + \frac{F_c}{2}\langle x-a_4-tl\rangle^2 - \frac{R_{re}}{2}\langle x-a_2\rangle^2 - \frac{R_{fe}}{2}\langle x-a_3\rangle^2 \\
&+ \sum_{k=1}^{n}\frac{V_k}{2}\langle x-b_k\rangle + \sum_{k=1}^{n} M_k\langle x-b_k\rangle^1 + M_{be}\langle x-a_2\rangle^1 + q_1
\end{aligned}
\right)
\tag{2.6}
$$

Integrating the slope of the beam yields the elastic deflection, $y_s(x)$:

$$
y_s(x) = \frac{1}{EI_n}\left(
\begin{aligned}
&\frac{F_{de}}{6}\langle x-a_1\rangle^3 + \frac{F_c}{6}\langle x-a_4-tl\rangle^3 - \frac{R_{re}}{6}\langle x-a_2\rangle^3 \\
&- \frac{R_{fe}}{6}\langle x-a_3\rangle^3 + \sum_{k=1}^{n}\frac{V_k}{6}\langle x-b_k\rangle^3 + \sum_{k=1}^{n}\frac{M_k}{2}\langle x-b_k\rangle^2 \\
&+ \frac{M_{be}}{2}\langle x-a_3\rangle^2 + q_1 x + q_2
\end{aligned}
\right)
\tag{2.7}
$$

The integration constants, q_1 & q_2 can be found by applying the following boundary conditions:

$$
\begin{aligned}
y_s(x = a_2) &= 0 \\
y_s(x = a_3) &= 0
\end{aligned}
$$

Solving for the integration constants yields:

$$q_2 = -\frac{F_{de}}{6}\langle a_2 - a_1\rangle^3 - \sum_{k=1}^{n}\frac{V_k}{6}\langle a_2 - b_k\rangle^3 - \sum_{k=1}^{n}\frac{M_k}{2}\langle a_2 - b_k\rangle^2 - q_1 a_2 \tag{2.8}$$

$$q_1 = \frac{1}{(a_2 - a_3)}\left\{\begin{array}{l} \frac{F_{de}}{6}\left(\langle a_3 - a_1\rangle^3 - \langle a_2 - a_1\rangle^3\right) - \frac{R_{re}}{6}\langle a_3 - a_2\rangle^3 \\ + \sum_{k=1}^{n}\frac{V_k}{6}\left(\langle a_3 - b_k\rangle^3 - \langle a_2 - b_k\rangle^3\right) \\ + \sum_{k=1}^{n}\frac{M_k}{2}\left(\langle a_3 - b_k\rangle^2 - \langle a_2 - b_k\rangle^2\right) \end{array}\right\} \tag{2.9}$$

The derivation for the moments and shear forces induced, and the equivalent applied forces when the stepped shaft is transformed into a uniform shaft will now be presented. The derivation begins by looking at the internal shear and bending moments for an arbitrary segment in the stepped beam (Fig 2.4). An illustration of the shear and bending moment diagrams is also offered (Fig. 2.5).

From the shear and bending moment diagrams it was found that:

$$V(x) = V_l = V_r \tag{2.10}$$

and

$$M(x) = M_l - V_l x \tag{2.11}$$

From Castigliano's Second Theorem:

$$\frac{\partial U}{\partial V} = y \tag{2.12}$$

and

$$\frac{\partial U}{\partial M} = \theta \tag{2.13}$$

The strain energy, U for one-dimensional bending is known to be:

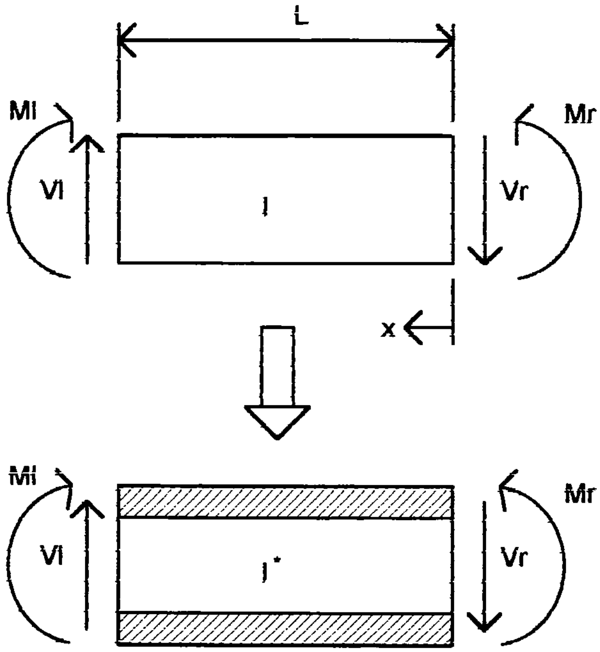

Figure 2.4 Transformation of a Beam Segment

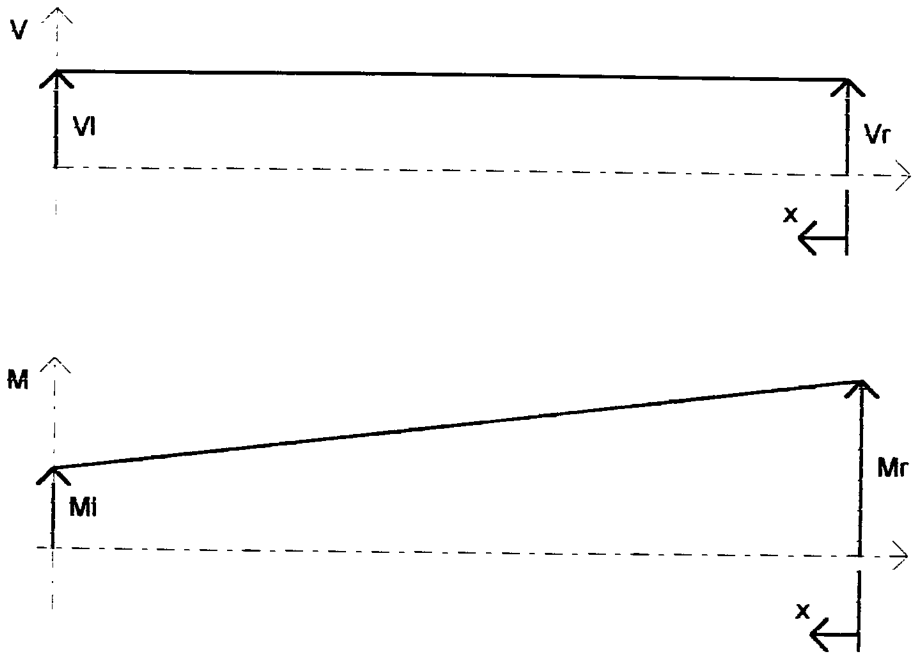

Figure 2.5 Shear and Bending Moment Diagrams for a Beam Segment

$$U = \int_0^l \frac{M^2}{2EI}\, dx \tag{2.14}$$

It should be noted that this expression for the strain energy does not include any contribution due to transverse shear deformation. Substituting equation (2.14) into equations (2.12) and (2.13), with eqns. (2.10) and (2.11) for the original beam segment (prior to the transformation to the uniform shaft), yields the following y and θ:

$$y = \frac{-1}{EI}\left\{\frac{M_r l^2}{2} - \frac{V_r l^3}{3}\right\} \tag{2.15}$$

$$\theta = \frac{1}{EI}\left\{M_r l - \frac{V_r l^2}{2}\right\} \tag{2.16}$$

Similarly, when the analysis is repeated for the segment after its transformation the deflection and slope, y* and θ* are found to be:

$$y^* = \frac{-1}{EI^*}\left\{\frac{M_r l^2}{2} - \frac{V_r l^3}{3}\right\} \tag{2.17}$$

$$\theta^* = \frac{1}{EI^*}\left\{M_r l - \frac{V_r l^2}{2}\right\} \tag{2.18}$$

The differences in y and θ must be compensated for with the induced shear force and bending moment.

$$\Delta y = -\left\{\frac{1}{EI} - \frac{1}{EI^*}\right\}\left\{\frac{M_r l^2}{2} - \frac{V_r l^3}{3}\right\} \tag{2.19}$$

$$\Delta\theta = \left\{\frac{1}{EI} - \frac{1}{EI^*}\right\}\left\{M_r l - \frac{V_r l^2}{2}\right\} \tag{2.20}$$

Therefore:

$$\left\{\frac{1}{EI}-\frac{1}{EI^*}\right\}\left\{\frac{M_r l^2}{2}-\frac{V_r l^3}{6}\right\}=\left\{\frac{1}{EI^*}\right\}\left\{\frac{M_{ind} l^2}{2}-\frac{V_{ind} l^3}{6}\right\} \tag{2.21}$$

$$\left\{\frac{1}{EI}-\frac{1}{EI^*}\right\}\left\{M_r l-\frac{V_r l^2}{2}\right\}=\left\{\frac{1}{EI^*}\right\}\left\{M_{ind} l-\frac{V_{ind} l^2}{2}\right\} \tag{2.22}$$

Multiplying eqn. (2.22) by –l/2 and adding eqn. (2.21) yields:

$$V_{ind}=I^*\left[\frac{1}{I}-\frac{1}{I^*}\right]V_r \tag{2.23}$$

Substituting eqn. (2.23) into eqn. (2.21) yields:

$$M_{ind}=I^*\left[\frac{1}{I}-\frac{1}{I^*}\right]M_r \tag{2.24}$$

This analysis can be repeated to find an induced shear and bending moment at each step in the shaft. The induced force and moment now become applied forces to a beam segment with a moment of inertia of I^* This analysis can be extended to show that all applied forces must be scaled by a factor of I_N/I. Where I_N is the moment of inertia of the uniform beam (largest moment of inertia in the stepped shaft), and I is the moment of inertia of the segment that the force is applied to. Therefore the induced forces and moments become:

$$V_k=I_n\left[\frac{1}{I}-\frac{1}{I^*}\right]V_r \tag{2.25}$$

$$M_k=I_n\left[\frac{1}{I}-\frac{1}{I^*}\right]M_r \tag{2.26}$$

where:

$$V_r=R_r\langle x-a_2\rangle^0+R_f\langle x-a_3\rangle^0-f_d\langle x-a_1\rangle^0-f_c\langle x-a_4-tl\rangle^0 \tag{2.27}$$

$$M_r = R_r\langle b_k - a_2\rangle\langle b_k - a_2\rangle^0 + R_f\langle b_k - a_3\rangle\langle b_k - a_3\rangle^0 - f_d\langle b_k - a_1\rangle\langle b_k - a_1\rangle^0 - f_c\langle b_k - a_4 - tl\rangle\langle b_k - a_4 - tl\rangle^0 - mb\langle b_k - a_3\rangle^0 \tag{2.28}$$

The cutting, driving, and reaction forces from the stepped spindle shaft must also be scaled to provide equivalent applied forces on the uniform shaft. The scaling of these forces yields:

$$F_{de} = \frac{I}{I_{fc}} F_d \tag{2.29}$$

$$R_{re} = \frac{I}{I_{Rr}} R_r \tag{2.30}$$

$$R_{fe} = \frac{I}{I_{Rf}} R_f \tag{2.31}$$

$$M_{be} = \frac{I}{I_{Mb}} M_b \tag{2.32}$$

2.2 Deflection of Bearings:

The deflection contribution of the spindle bearings was calculated by assuming that the spindle is a rigid shaft supported by two flexible bearings (Figure 2.2b). The cutting force F_c, and the driving force F_d were used to solve for the reactions at the bearing. The two reaction forces were used to calculate δ_r and δ_f, the deflections at the two bearings. The deflection contribution of the bearings is a straight line through δ_r and δ_f.

$$y_b = \frac{(\delta_r - \delta_f)(x - a_2) + \delta_r(a_3 - a_2)}{(a_3 - a_2)} \tag{2.33}$$

Where

$$\delta_r = \frac{\left(f_c(a_4 + tl - a_3) - mb - f_d(a_3 - a_2)\right)}{(a_3 - a_2)K_r} \tag{2.34}$$

$$\delta_f = \frac{f_d(a_3 - a_2) + mb - f_c(a_4 + tl - a_3) - (f_c + f_d)(a_{3-}a_2)}{(a_3 - a_2)K_f} \tag{2.35}$$

$$m_b = f_c(a_4 + tl - a_3)x_o \tag{2.36}$$

2.3 *Matlab* Solution:

A program was developed using *Matlab* to automate the static analysis of the spindle shaft. The user must simply enter the geometry, loads, and support parameters into a spreadsheet called a "batch file" A copy of the batch file template is presented in Appendix A. The *Matlab* programming code used to automate the static analysis can be found in Appendix B.

An example of the analysis for a simple spindle is presented here. Figure 2.6 illustrates the batch file for the static analysis. Upon the completion of the batch file the program will read the file and report a geometric representation of the spindle. The plot illustrates the geometry of the shaft as well as the locations of the bearings, cutting force, and drive force (see figure 2.7). This feedback allows the user to easily check for mistakes in the batch file. With all the information correct the program calculates and reports plots of the deflection contribution of the elastic shaft (figure 2.8a) and the deflection contribution of the bearings (figure 2.8b). Finally the program reports a plot of the total deformation of the spindle (see figure 2.9).

Batch File:

Geometry:

Number of Sections(#):	6

Section (#)	Length (in)	Outer Diameter (in)	Inner Diameter (in)	Area (in^2)	Moment of Inertia (in^4)
1	3	2.25	2	0.83449	0.47265783
2	3	2.375	2.125	0.88357	0.560861726
3	3	2.5	2.25	0.93266	0.659419991
4	3	2.625	2.375	0.98175	0.76890787
5	3	2.75	2.5	1.03084	0.889900605
6	3	2.875	2.625	1.07992	1.022973438
7				0	0
8				0	0
9				0	0
10				0	0
11				0	0
12				0	0
13				0	0
14				0	0
15				0	0

Bearings:

Lat. Stiffness of Rear Bearing (lb/in):	100000
Lat. Stiffness of Front Bearing (lb/in.):	500000
Tor. Stiffness of Front Bearing (in-lb):	10000
Fraction of mom. on Front Bearing:	0.1
Location of Rear Bearing (in.)	7.5
Location of Front Bearing (in.)	13.5

Pulley:

Location of Pulley (in.)	4.5
Mass of Pulley (lb-s^2/in):	[illegible]
Static Belt Tension (lb):	50
Harmonic Drive Force (lb):	50
Drive Frequency (Hz):	133.33
Pulley Unbalance (lb-s^2):	[illegible]

Tool:

Mass of Tool (lb-s^2/in):	[illegible]
Static Cutting Force (lb):	300
Harmonic Cutting Force (lb):	[illegible]
Cutting Frequency (Hz):	133.33
Tool Unbalance (lb-s^2):	[illegible]
Length of Tool (in)	2

Speed:

Spindle Shaft Speed (rpm):	16.66666667

Material Properties:

Modulus of Elasticity (psi):	30000000
Density (lb/in^3):	0.289

Figure 2.6 Batch File for *Matlab* Solution

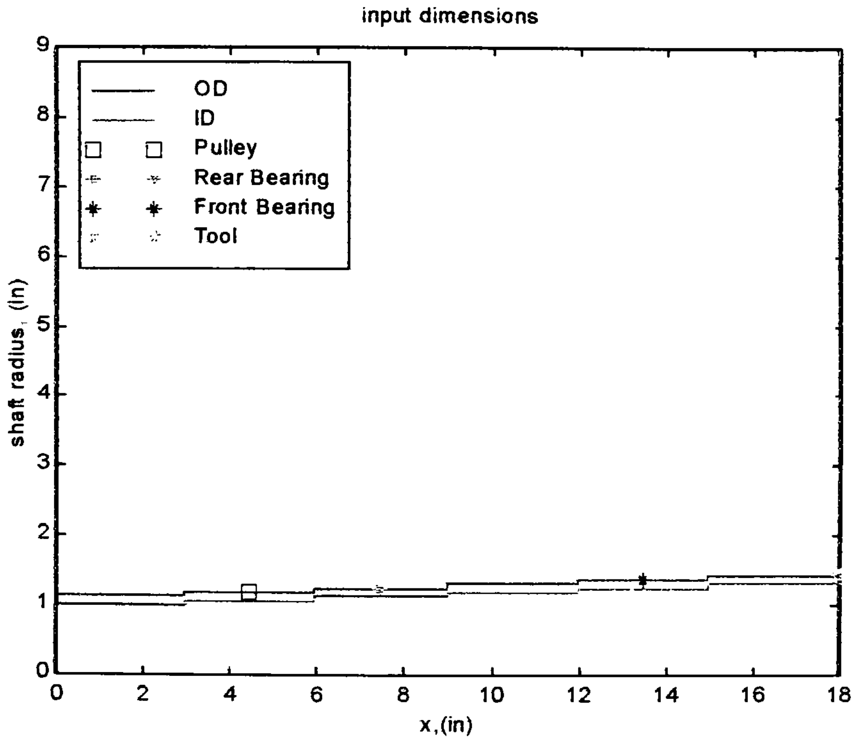

Figure 2.7 *Matlab* Representation of Geometry

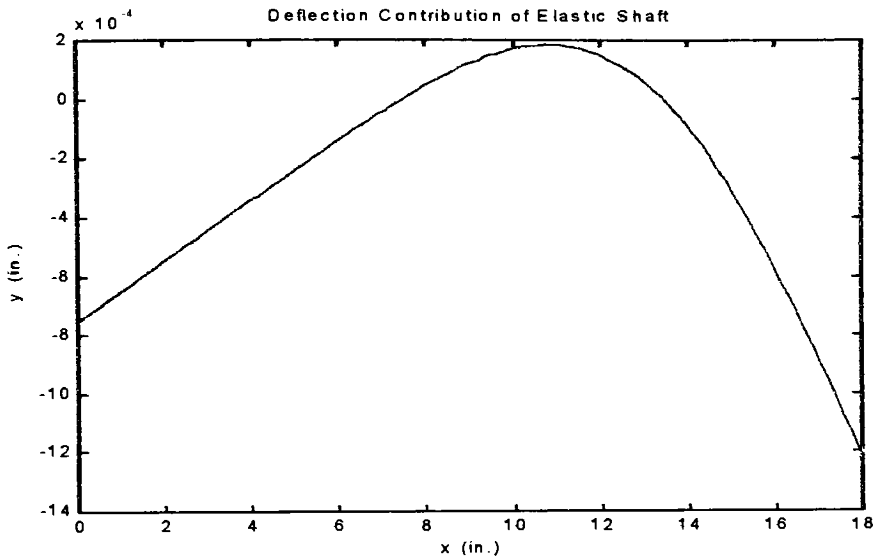

Figure 2.8a Deflection Contribution of Elastic Shaft

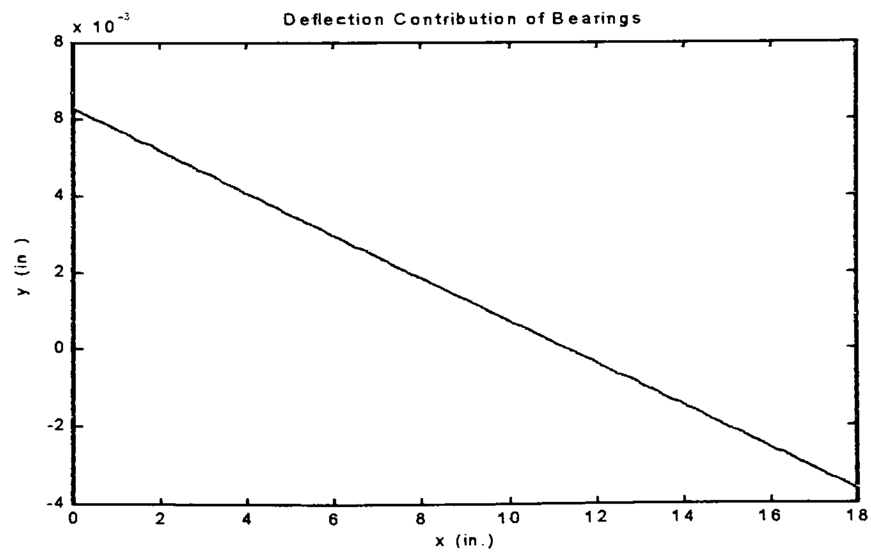

Figure 2.8b Deflection Contribution of Support Bearings

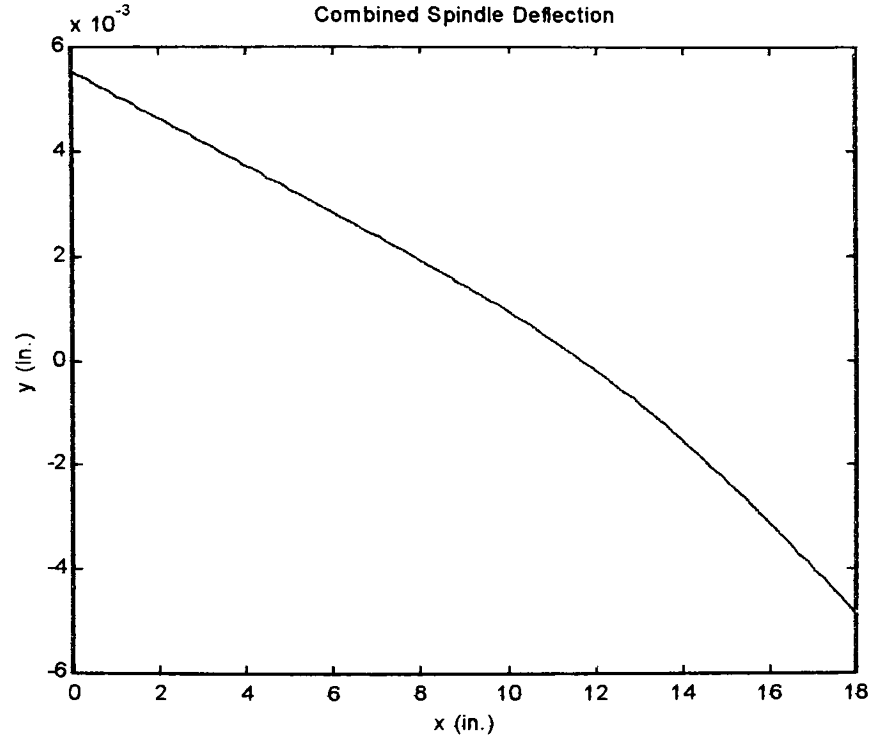

Figure 2.9 Total Deflection of Spindle

In order to confirm the results offered by the program, a finite element analysis of the sample spindle was performed using *Ansys*. The spindle was modeled using one-dimensional linearly elastic beam elements. The bearings were modeled using linear spring elements. The cutting force was transformed into a force moment couple and applied at the end of the spindle shaft in order to account for the tool length. Figure 2.10 compares the deflections of the shaft using both methods. It is clear from the plot that there is an excellent correlation between the two analyses.

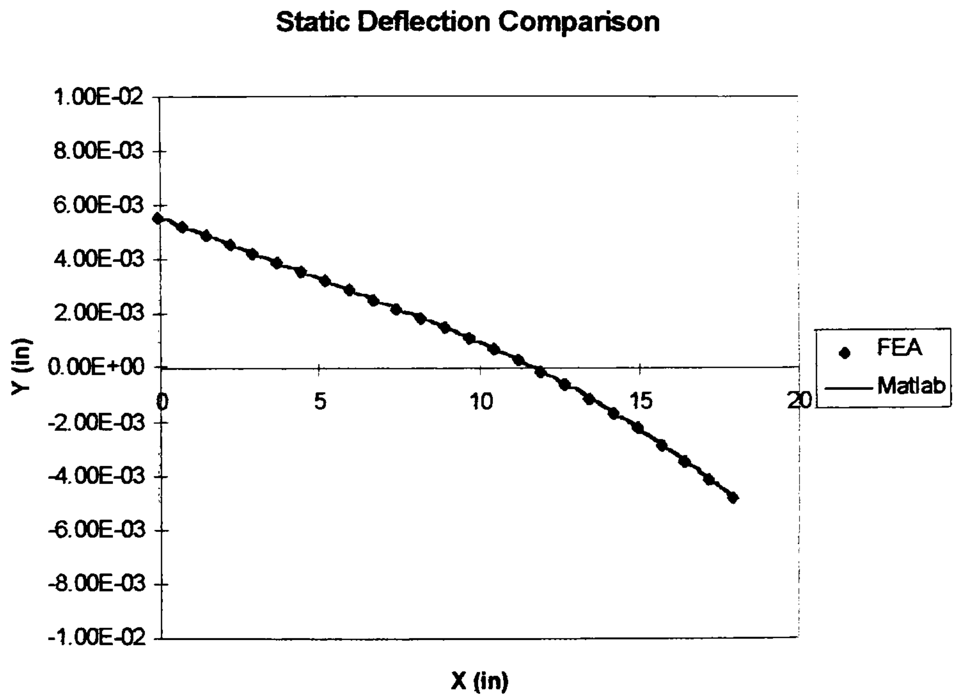

Figure 2.10 Comparison of Total Spindle Deflection (FEA vs. *Matlab*)

3.0 Dynamic Analysis:

The dynamic analysis for the spindle shaft consists of two portions. The first part of the analysis is the modal analysis. The beam is treated as a continuous system for this portion of the analysis. The second part of the analysis solves for the deflection of the spindle by means of modal superposition. The following assumptions were made in order to perform the analysis:

1. The spindle shaft is assumed to be an Euler-Bernoulli Beam.
2. The spindle is subjected to a cutting force ($F_c Sin(\omega_c t)$), a drive force ($F_d Sin(\omega_d t)$), unbalance forces ($F_{cub} Sin(\omega t)$ & ($F_{dub} Sin(\omega t)$), and the reaction forces at the bearings. The drive force must be applied behind the rear bearing. The cutting force and drive force are assumed to be harmonic.
3. The masses of the pulley and cutting tool are assumed to be concentrated. The mass of the pulley is assumed to be concentrated at the centerline of the pulley. The mass of the tool is assumed to be concentrated at the end of the spindle shaft. This point is often referred to as the gauge line.
4. There is no unbalance excitation introduced by the spindle shaft.
5. The rotational affects of the spindle shaft are neglected.
6. The torsional and axial deflections of the spindle shaft are neglected.
7. The centerline of the spindle shaft is exactly inline with the centerline of the bearing bores. There is no contribution to the lateral deflection due to manufacturing misalignment.

8. The spindle housing and the cutting tool are both assumed to have an infinite stiffness.
9. It is assumed that the spindle is supported by only two bearings. This is common for most machine tool spindles. Manufacturability typically precludes the use of more than two bearings in most spindles.
10. The contribution of transverse shear deformation to the overall lateral deflection is assumed to be negligible.
11. Damping is neglected in the dynamic analysis.

The model scrutinized in the dynamic analysis is very similar to the model used in the static analysis. One major difference is the use of a torsional spring to represent the torsional stiffness of the front support bearings. In addition the masses of the pulley and cutting tool are included. See figure 3.1 for the dynamic model under scrutiny.

3.1 Modal Analysis:

The foundation for the modal analysis is the derivation of the wave equation for the lateral vibration of a continuous Euler-Bernoulli beam. Figure 3.2 represents the free body diagram of an differential element of an E-B beam. Applying Newton's second law to the beam element it can be shown that:

$$\frac{\partial V}{\partial x} = -\rho A \frac{\partial^2 y}{\partial t^2} \tag{3.1}$$

and

$$V = \frac{\partial M}{\partial x} \tag{3.2}$$

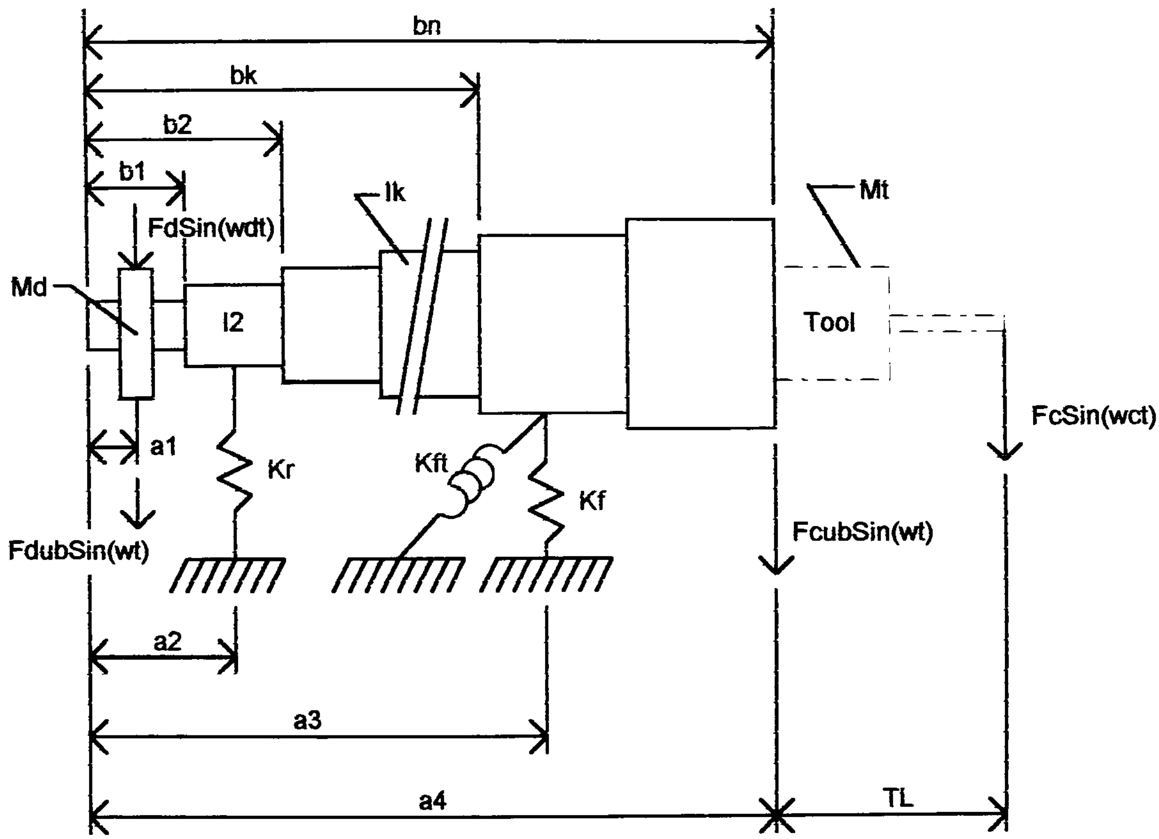

Figure 3.1 Dynamic Spindle Model

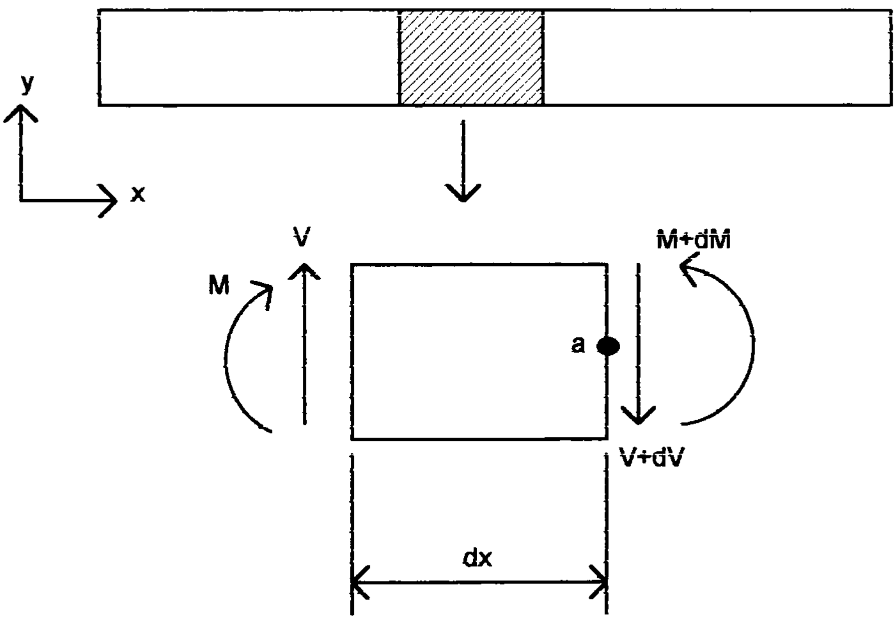

Figure 3.2 Differential Element of an E-B Beam

It can also be shown from Strengths of Materials that:

$$M = EI\frac{\partial^2 y}{\partial x^2} \tag{3.3}$$

Substituting eq. 3.3 into 3.2 yields:

$$V = EI\frac{\partial^3 y}{\partial x^3} \tag{3.4}$$

Finally, substituting eq. 3.4 into 3.1 and rearranging yields:

$$\frac{\partial^2 y}{\partial t^2} + \frac{EI}{\rho A}\frac{\partial^4 y}{\partial x^4} = 0 \tag{3.5}$$

The following harmonic solution to eq. 3.5 was assumed:

$$y(x,t) = \bar{y}(x)\sin \omega t \tag{3.6}$$

Substituting the assumed solution (eq. 3.6), into the differential equation (3.5) yields the following forth-order differential equation:

$$\frac{\partial^4 \bar{y}}{\partial x^4} - \beta^4 \bar{y} = 0 \tag{3.7}$$

where:

$$\beta^4 = \frac{\rho A \varpi^2}{EI} \tag{3.8}$$

It can be shown that the general solution to the preceding forth-order differential equation is:

$$\bar{y}(x) = A\cosh \beta x + B\sinh \beta x + C\cos \beta x + D\sin \beta x \tag{3.9}$$

Equation 3.9 represents the wave equation for an E-B beam. The mode shapes for a beam can be found by substituting values for β that correspond to the resonant frequencies. The constants A,B,C, and D can be solved for by applying the boundary condition for the beam.

A systematic method involving numerical methods was developed to solve for the resonant frequencies and their corresponding mode shapes. This method is not exclusive to the spindle problem at hand. It can be extended to the lateral vibration of many Euler-Bernoulli problems. Listed below are the steps to this method:

1. Establish the boundary conditions for the system.
2. Collect the system of equations into matrix form.
3. Using Gaussian Elimination numerically reduce the matrix.
4. Using the Bisection Method or a comparable root finding method solve for the resonant frequency, β.
5. Back substitute to find the constants A,B,C and D for the beam segment.

Figure 3.3 represents a simple beam used to illustrate this approach. The beam under scrutiny here is a uniform E-B beam fixed at both ends. The first step is to find the boundary conditions. Since the beam is fixed-fixed, the displacement and rotation at x = 0,l are both equal to zero. Expressed mathematically:

$$\bar{y}(0) = 0 \tag{3.10}$$

$$\bar{y}'(0) = 0 \tag{3.11}$$

$$\bar{y}(l) = 0 \tag{3.12}$$

$$\bar{y}'(l) = 0 \tag{3.13}$$

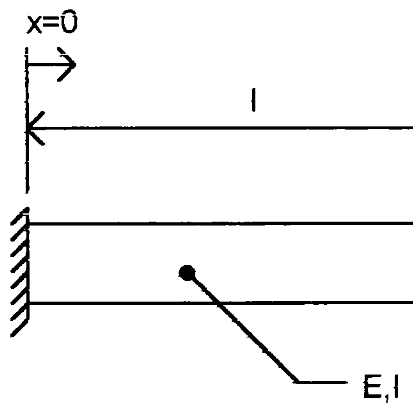

Figure 3.3 Sample Euler-Bernoulli Beam

Substituting Eq. 3.9 into Eqs. 3.10-3.13 yields:

$$\bar{y}(0) = A + C = 0 \tag{3.14}$$

$$\bar{y}'(0) = B + D = 0 \tag{3.15}$$

$$\bar{y}(l) = A\cosh(\beta l) + B\sinh(\beta l) + C\cos(\beta l) + D\sin(\beta l) = 0 \tag{3.16}$$

$$\bar{y}'(l) = A\sinh(\beta l) + B\cosh(\beta l) - C\sin(\beta l) + D\cos(\beta l) = 0 \tag{3.17}$$

The next step is to collect this system of four equations into matrix form. This yields Eq. 3.18:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \cosh(\beta l) & \sinh(\beta l) & \cos(\beta l) & \sin(\beta l) \\ \sinh(\beta l) & \cosh(\beta l) & -\sin(\beta l) & \cos(\beta l) \end{bmatrix} \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.18}$$

Step (3) reduces the matrix in eqn. 3.18 using Guass-Jordan elimination. The reduced system is illustrated in eqn. 3.19:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & \dfrac{\sin(\beta l) - \sinh(\beta l)}{\cos(\beta l) - \cosh(\beta l)} \\ 0 & 0 & 0 & \cos(\beta l)\cosh(\beta l) - 1 \end{bmatrix} \begin{Bmatrix} A \\ B \\ C \\ D \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.19}$$

The reduced system can be used to solve for the resonant frequencies, β_j.

$$\left(\cos(\beta l)\cosh(\beta l) - 1\right)D = 0 \tag{3.20}$$

If D was equal to zero, then A,B, and C would also equal zero. This would not be a meaningful result. Therefore it can be concluded that:

$$\left(\cos(\beta l)\cosh(\beta l) - 1\right) = 0 \tag{3.21}$$

This is where the root finding method suggested in step (4) comes into place. The roots of eq. 3.21 lead to the resonant frequencies of the system. Solving for the roots yields:

$$\beta_1 l, \beta_2 l, \beta_3 l = 4.7, 7.8, 11.0$$

After solving for the roots the final step is to back substitute to obtain the constants A,B,C and D. Begin the substitution by assuming that D=1. Working backward from D it can be shown that the remaining constants are:

$$C = \frac{\sinh(\beta l) - \sin(\beta l)}{\cos(\beta l) - \cosh(\beta l)}$$

$$B = -1$$

$$A = \frac{\sin(\beta l) - \sinh(\beta l)}{\cos(\beta l) - \cosh(\beta l)}$$

Substituting these constants into eq. 3.9 yields the mode shape for the sample beam. The equation for the mode shapes becomes:

$$\begin{aligned} \bar{y}(x) &= \frac{\sin(\beta_j l) - \sinh(\beta_j l)}{\cos(\beta_j l) - \cosh(\beta_j l)} \cosh(\beta_j x) + \sinh(\beta_j x) \\ &+ \frac{\sinh(\beta_j l) - \sin(\beta_j l)}{\cos(\beta_j l) - \cosh(\beta_j l)} \cos(\beta_j x) + \sin(\beta_j x) \end{aligned} \qquad j=1,2,.. \qquad (3.22)$$

Figure 3.4 illustrates the first three mode shapes for the sample beam.

The method described here can be applied to find the mode shapes for all uniform E-B beam problems. However if the beam is stepped, as is the case with the spindle shaft, there needs to be a set of boundary conditions for each beam segment. This leads

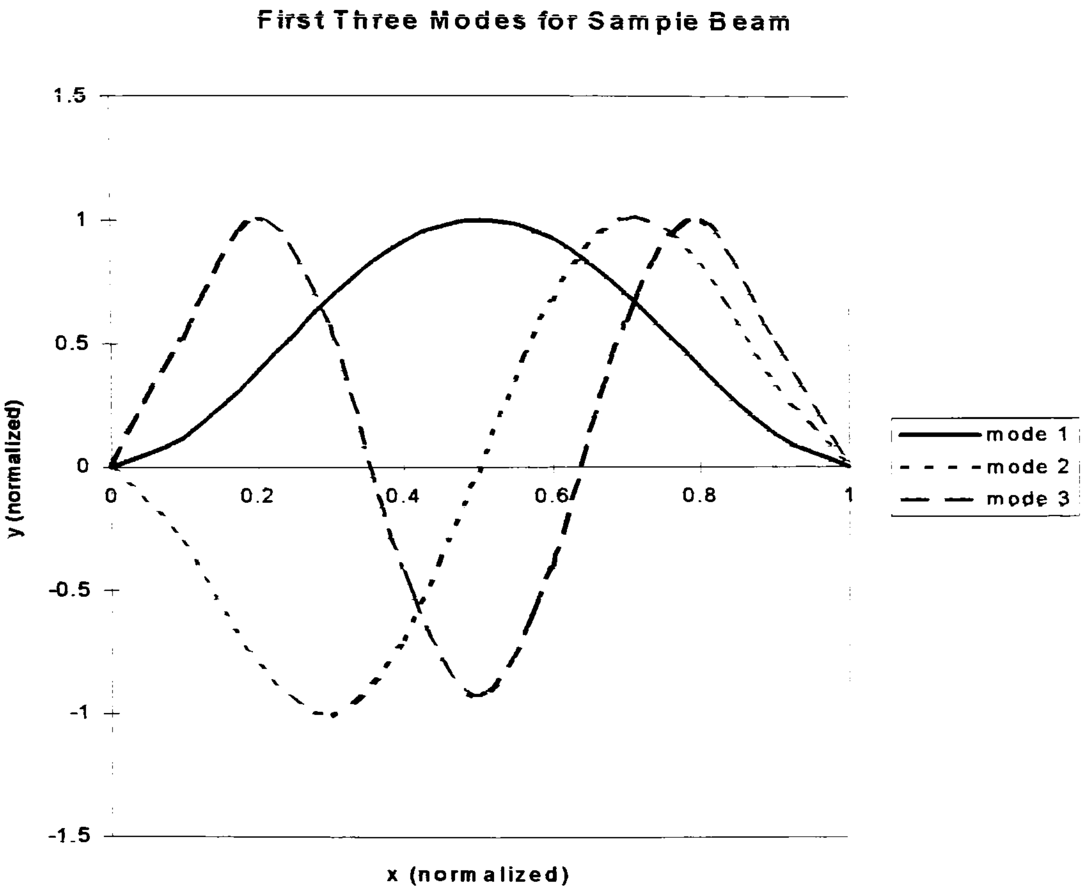

Figure 3.4 Mode Shapes for Sample Beam

to a very large system of equations. Aside from the problem of having a very large system, the number of steps would change for different spindles. This would make automation very difficult. A transformation matrix was developed to handle the steps in the shaft. The transformation matrix relates the constants on one side of a step to the constants on the other side of the step. This makes the number of equations in the system independent of the number of steps in the shaft.

The development of this transformation matrix begins by looking at an arbitrary step in an Euler-Bernoulli beam (see figure 3.5). In order for continuity to exist the deflection, slope, moment, and shear force at the joint must be the same for both beam segments.

$$\bar{y}_1(l) = \bar{y}_2(l) \tag{3.23}$$

$$\bar{y}_1'(l) = \bar{y}_2'(l) \tag{3.24}$$

$$(EI)_1 \frac{d^2\bar{y}_1(l)}{dx^2} = (EI)_2 \frac{d^2\bar{y}_2(l)}{dx^2} \tag{3.25}$$

$$(EI)_1 \frac{d^3\bar{y}_1(l)}{dx^3} = (EI)_2 \frac{d^3\bar{y}_2(l)}{dx^3} \tag{3.26}$$

Substituting eq. 3.8 yields:

$$\begin{aligned} &A_1 \cosh(\beta_1 l) + B_1 \sinh(\beta_1 l) + C_1 \cos(\beta_1 l) + D_1 \sin(\beta_1 l) = \\ &A_2 \cosh(\beta_2 l) + B_2 \sinh(\beta_2 l) + C_2 \cos(\beta_2 l) + D_2 \sin(\beta_2 l) \end{aligned} \tag{3.27}$$

$$\begin{aligned} &\beta_1 \left(A_1 \sinh(\beta_1 l) + B_1 \cosh(\beta_1 l) - C_1 \sin(\beta_1 l) + D_1 \cos(\beta_1 l)\right) = \\ &\beta_2 \left(A_2 \sinh(\beta_2 l) + B_2 \cosh(\beta_2 l) - C_2 \sin(\beta_2 l) + D_2 \cos(\beta_2 l)\right) \end{aligned} \tag{3.28}$$

$$\begin{aligned} &\beta_1^2 \left(A_1 \cosh(\beta_1 l) + B_1 \sinh(\beta_1 l) - C_1 \cos(\beta_1 l) - D_1 \sin(\beta_1 l)\right) = \\ &\beta_2^2 \left(A_2 \cosh(\beta_2 l) + B_2 \sinh(\beta_2 l) - C_2 \cos(\beta_2 l) - D_2 \sin(\beta_2 l)\right) \end{aligned} \tag{3.29}$$

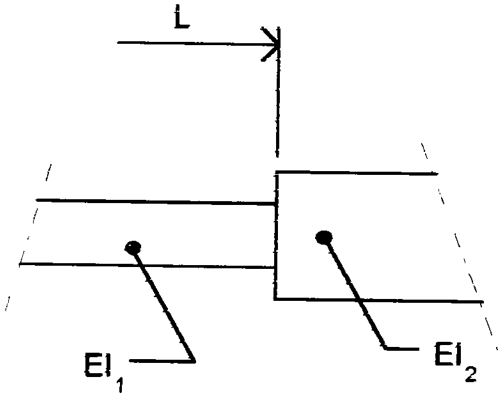

Figure 3.5 Step in E-B Beam

$$\begin{aligned}&\beta_1^{\,3}\left(A_1 \sinh(\beta_1 l) + B_1 \cosh(\beta_1 l) + C_1 \sin(\beta_1 l) - D_1 \cos(\beta_1 l)\right) = \\ &\beta_2^{\,3}\left(A_2 \sinh(\beta_2 l) + B_2 \cosh(\beta_2 l) + C_2 \sin(\beta_2 l) - D_2 \cos(\beta_2 l)\right)\end{aligned} \tag{3.30}$$

The system of four equations and eight unknowns can be collected into matrix form.

$$\begin{bmatrix} \cosh(\beta_1 l) & \sinh(\beta_1 l) & \cos(\beta_1 l) & \sin(\beta_1 l) & -\cosh(\beta_2 l) & -\sinh(\beta_2 l) & -\cos(\beta_2 l) & -\sin(\beta_2 l) \\ \sinh(\beta_1 l) & \cosh(\beta_1 l) & -\sin(\beta_1 l) & \cos(\beta_1 l) & -\sinh(\beta_2 l) & -\cosh(\beta_2 l) & \sin(\beta_2 l) & -\cos(\beta_2 l) \\ \cosh(\beta_1 l) & \sinh(\beta_1 l) & -\cos(\beta_1 l) & -\sin(\beta_1 l) & -\frac{(EI)_2 \beta_2^{\,2}}{(EI)_1 \beta_1^{\,2}}\cosh(\beta_2 l) & -\frac{(EI)_2 \beta_2^{\,2}}{(EI)_1 \beta_1^{\,2}}\sinh(\beta_2 l) & \frac{(EI)_2 \beta_2^{\,2}}{(EI)_1 \beta_1^{\,2}}\cos(\beta_2 l) & \frac{(EI)_2 \beta_2^{\,2}}{(EI)_1 \beta_1^{\,2}}\sin(\beta_2 l) \\ \sinh(\beta_1 l) & \cosh(\beta_1 l) & \sin(\beta_1 l) & -\cos(\beta_1 l) & -\frac{(EI)_2 \beta_2^{\,3}}{(EI)_1 \beta_1^{\,3}}\sinh(\beta_2 l) & -\frac{(EI)_2 \beta_2^{\,3}}{(EI)_1 \beta_1^{\,3}}\cosh(\beta_2 l) & -\frac{(EI)_2 \beta_2^{\,3}}{(EI)_1 \beta_1^{\,3}}\sinh(\beta_2 l) & \frac{(EI)_2 \beta_2^{\,3}}{(EI)_1 \beta_1^{\,3}}\cosh(\beta_2 l) \end{bmatrix}$$

$$* \begin{Bmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \\ A_2 \\ B_2 \\ C_2 \\ D_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.31}$$

Using Gauss-Jordan elimination followed by back substitution a relationship can be found between A_1-D_1 and A_2-D_2. Two ratios, R_1 and r_1, were defined to simplify the relationship.

$$R_1 = \frac{(EI)_2}{(EI)_1} \tag{3.32}$$

$$r_1 = \frac{\beta_2}{\beta_1} \tag{3.33}$$

$$
\begin{aligned}
A_1 = {} & \frac{\left[r_1 R_1^{\,2}+1\right]}{2}\{\cosh(\beta_1 l)\cos(\beta_2 l) - R_1 \sinh(\beta_1 l)\sinh(\beta_2 l)\}A_2 \\
& + \frac{\left[r_1 R_1^{\,2}+1\right]}{2}\{\cosh(\beta_1 l)\sinh(\beta_2 l) - R_1 \sinh(\beta_1 l)\cosh(\beta_2 l)\}B_2 \\
& - \frac{\left[r_1 R_1^{\,2}-1\right]}{2}\{\cosh(\beta_1 l)\cos(\beta_2 l) + R_1 \sinh(\beta_1 l)\sin(\beta_2 l)\}C_2 \\
& - \frac{\left[r_1 R_1^{\,2}-1\right]}{2}\{\cosh(\beta_1 l)\sin(\beta_2 l) - R_1 \sinh(\beta_1 l)\cos(\beta_2 l)\}D_2
\end{aligned} \tag{3.34}
$$

$$
\begin{aligned}
B_1 = {} & \frac{\left[r_1 R_1^{\,2}+1\right]}{2}\{R_1 \cosh(\beta_1 l)\sinh(\beta_2 l) - \sinh(\beta_1 l)\cosh(\beta_2 l)\}A_2 \\
& + \frac{\left[r_1 R_1^{\,2}+1\right]}{2}\{R_1 \cosh(\beta_1 l)\cosh(\beta_2 l) - \sinh(\beta_1 l)\sinh(\beta_2 l)\}B_2 \\
& + \frac{\left[r_1 R_1^{\,2}-1\right]}{2}\{R_1 \cosh(\beta_1 l)\sin(\beta_2 l) + \sinh(\beta_1 l)\cos(\beta_2 l)\}C_2 \\
& + \frac{\left[r_1 R_1^{\,2}-1\right]}{2}\{\sinh(\beta_1 l)\sin(\beta_2 l) - R_1 \cosh(\beta_1 l)\cos(\beta_2 l)\}D_2
\end{aligned} \tag{3.35}
$$

$$
\begin{aligned}
C_1 = {} & \frac{\left[r_1 R_1^{\,2}-1\right]}{2}\{R_1 \sin(\beta_1 l)\sinh(\beta_2 l) - \cos(\beta_1 l)\cosh(\beta_2 l)\}A_2 \\
& + \frac{\left[r_1 R_1^{\,2}-1\right]}{2}\{R_1 \sin(\beta_1 l)\cosh(\beta_2 l) - \sinh(\beta_1 l)\cos(\beta_2 l)\}B_2 \\
& + \frac{\left[r_1 R_1^{\,2}+1\right]}{2}\{\cos(\beta_1 l)\cos(\beta_2 l) + R_1 \sin(\beta_1 l)\sin(\beta_2 l)\}C_2 \\
& + \frac{\left[r_1 R_1^{\,2}+1\right]}{2}\{\cos(\beta_1 l)\sin(\beta_2 l) - R_1 \cos(\beta_1 l)\sin(\beta_2 l)\}D_2
\end{aligned} \tag{3.36}
$$

$$
\begin{aligned}
D_1 = &-\frac{\left[r_1 R_1^2 - 1\right]}{2}\left\{R_1 \cos(\beta_1 l)\sinh(\beta_2 l) + \sin(\beta_1 l)\cosh(\beta_2 l)\right\}A_2 \\
&-\frac{\left[r_1 R_1^2 - 1\right]}{2}\left\{R_1 \cos(\beta_1 l)\cosh(\beta_2 l) + \sin(\beta_1 l)\sinh(\beta_2 l)\right\}B_2 \\
&-\frac{\left[r_1 R_1^2 + 1\right]}{2}\left\{R_1 \cos(\beta_1 l)\sin(\beta_2 l) - \sin(\beta_1 l)\sin(\beta_2 l)\right\}C_2 \\
&+\frac{\left[r_1 R_1^2 + 1\right]}{2}\left\{R_1 \cos(\beta_1 l)\cos(\beta_2 l) + \sin(\beta_1 l)\sin(\beta_2 l)\right\}D_2
\end{aligned}
\tag{3.37}
$$

The coefficients from eqns. 3.34-3.37 can be collected into a transformation matrix [T], such that:

$$
\begin{Bmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \end{Bmatrix} = \left[T_{21}\right]\begin{Bmatrix} A_2 \\ B_2 \\ C_2 \\ D_2 \end{Bmatrix}
\tag{3.38}
$$

The use of the transformation matrix can be illustrated by expanding the sample beam problem to include steps in the beam (see figure 3.6). Applying the boundary conditions would result in the following system of equations:

$$
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cosh(\beta_3 l_3) & \sinh(\beta_3 l_3) & \cos(\beta_3 l_3) & \sin(\beta_3 l_3) \\
0 & 0 & 0 & 0 & \sinh(\beta_3 l_3) & \cosh(\beta_3 l_3) & -\sin(\beta_3 l_3) & \cos(\beta_3 l_3)
\end{bmatrix}
\begin{Bmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \\ A_3 \\ B_3 \\ C_3 \\ D_3 \end{Bmatrix}
=
\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}
\tag{3.39}
$$

If transformation matrices were not used, the only way to solve the system of equations would be to relate A_1-D_1 to A_3-D_3 by including the continuity equations. This would increase the size of the system to 12 equations and 12 unknowns. It would also make the

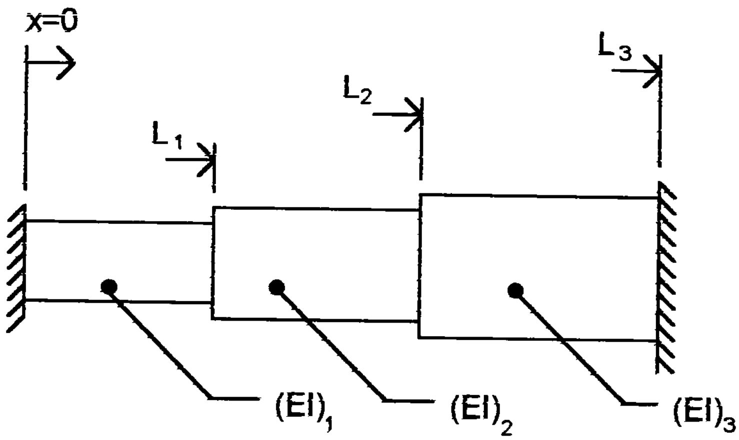

Figure 3.6 Sample Stepped Euler-Bernoulli Beam

size of the system dependent on the number of steps in the shaft. This in-turn would make automation more difficult. If the transformation matrices were used the system of equations would be reduced to 4 equations and 4 unknowns, regardless of the number of steps in the shaft. The first two equations in the system become:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} [T_{32}][T_{21}] \begin{Bmatrix} A_3 \\ B_3 \\ C_3 \\ D_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.40}$$

The last two equations will be the same as represented in eqn. 3.39. Once the system of equations is developed steps 3-5 of the pre-described method can be used to solve for the resonant frequencies and their corresponding mode shapes.

The five-step process and transformation matrix can now be combined and applied to find the frequencies and modes shapes of the spindle depicted in Figure 3.1. In order to encompass all of the externally applied boundary conditions the beam must be divided into four sections. Figures 3.7a-3.7d depict the four subdivisions. The first section is between the rear free end and the drive pulley. The second section is between the pulley and the rear support bearing. The third section is between the rear and front support bearings. The forth and final section is between the front support bearing and the cutting tool. There will be four constants for each of the four sections for a total of sixteen constants.

Beginning with the free end of section one, the shear force and bending moment at x = 0 are both equal to zero.

Expressed mathematically:

$$V_1(0) = EI\frac{d^3\overline{y}_1}{dx^3} = 0 \tag{3.41}$$

and

$$M_1(0) = EI\frac{d^2\overline{y}_1}{dx^2} = 0 \tag{3.42}$$

Substituting eqn. 3.9 into equations 3.10 and 3.11 and setting x equal to zero yields:

$$B_1 - D_1 = 0 \tag{3.43}$$

and

$$A_1 - D_1 = 0 \tag{3.44}$$

At the junction between sections 1 and 2 there are four boundary conditions. The first three conditions involve the deflection, slope and bending moment at the joint between sections 1 and 2. Since there are no externally applied moments, and the structure is continuous, the deflection, slope, and bending moment at the joint must be equal for both sections. Therefore:

$$\overline{y}_1(a_1) = \overline{y}_2(a_1) \tag{3.45}$$

$$\overline{y}_1'(a_1) = \overline{y}_2'(a_1) \tag{3.46}$$

$$EI\overline{y}_1''(a_1) = EI\overline{y}_2''(a_1) \tag{3.47}$$

Substituting equation 3.9 into equations 3.45-3.47 yields:

$$\begin{aligned} &A_1\cosh(\beta a_1) + B_1\sinh(\beta a_1) + C_1\cos(\beta a_1) + D_1\sin(\beta a_1) \\ &- A_2\cosh(\beta a_1) - B_2\sinh(\beta a_1) - C_2\cos(\beta a_1) - D_2\sin(\beta a_1) = 0 \end{aligned} \tag{3.48}$$

$$\begin{aligned} &A_1\sinh(\beta a_1) + B_1\cosh(\beta a_1) - C_1\sin(\beta a_1) + D_1\cos(\beta a_1) \\ &- A_2\sinh(\beta a_1) - B_2\cosh(\beta a_1) + C_2\sin(\beta a_1) - D_2\cos(\beta a_1) = 0 \end{aligned} \tag{3.49}$$

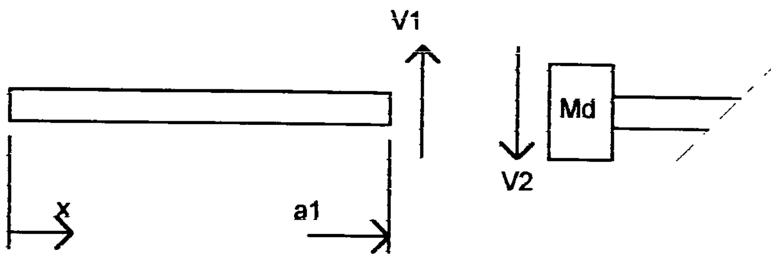

Figure 3.7a Boundary Conditions for Section 1

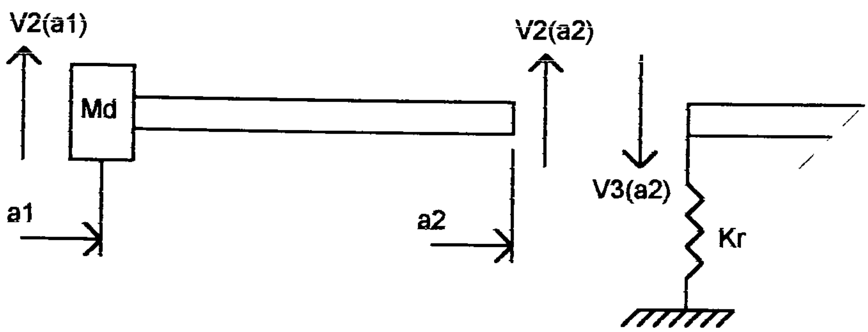

Figure 3.7b Boundary Conditions for Section 2

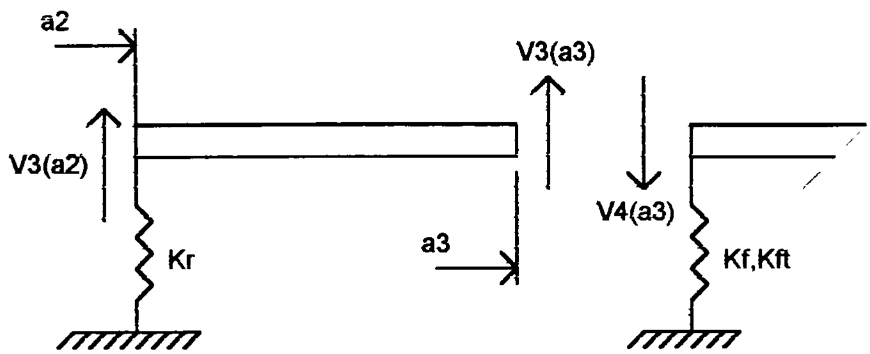

Figure 3.7c Boundary Conditions for Section 3

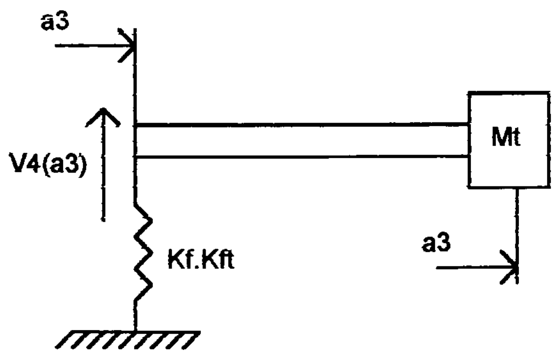

Figure 3.7d Boundary Conditions for Section 4

$$A_1 \cosh(\beta a_1) + B_1 \sinh(\beta a_1) - C_1 \cos(\beta a_1) - D_1 \sin(\beta a_1)$$
$$- A_2 \cosh(\beta a_1) - B_2 \sinh(\beta a_1) + C_2 \cos(\beta a_1) + D_2 \sin(\beta a_1) = 0 \tag{3.50}$$

The forth boundary condition at this joint is affected by the mass of the pulley. The mass of the pulley introduces an external shear force. Figure 3.8 illustrates the free body diagram at the joint. The shear force introduced by the mass is equal to the D'Alembert force associated with the pulley mass.

Therefore:

$$V_m = m_p \ddot{y} = -m_d \varpi^2 \bar{y}_2(a_1) \tag{3.51}$$

For equilibrium at the joint:

$$V_1(a_1) - V_2(a_1) = V_m \tag{3.52}$$

$$EI\bar{y}_1'''(a_1) - EI\bar{y}_2'''(a_1) = -m_d \omega^2 \bar{y}_2(a_1) \tag{3.53}$$

Substituting equation 3.9 into equation 3.53:

$$A_1 \sinh(\beta a_1) + B_1 \cosh(\beta a_1) + C_1 \sin(\beta a_1) - D_1 \cos(\beta a_1)$$
$$- A_2[\sinh(\beta a_1) - \frac{m_d \omega^2}{\beta^3 EI} \cosh(\beta a_1)] - B_2[\cosh(\beta a_1) - \frac{m_d \omega^2}{\beta^3 EI} \sinh(\beta a_1)] \tag{3.54}$$
$$+ C_2[\sin(\beta a_1) - \frac{m_d \omega^2}{\beta^3 EI} \cos(\beta a_1)] + D_2[\cos(\beta a_1) - \frac{m_d \omega^2}{\beta^3 EI} \sin(\beta a_1)] = 0$$

The first three boundary conditions for the joint between the second and third section are the same as the boundary conditions between the first and second joint.

Therefore:

$$A_2 \cosh(\beta(a_2 - a_1)) + B_2 \sinh(\beta(a_2 - a_1)) + C_2 \cos(\beta(a_2 - a_1))$$
$$+ D_2 \sin(\beta(a_2 - a_1)) - A_3 \cosh(\beta(a_2 - a_1)) - B_3 \sinh(\beta(a_2 - a_1)) \tag{3.55}$$
$$- C_3 \cos(\beta(a_2 - a_1)) - D_3 \sin(\beta(a_2 - a_1)) = 0$$

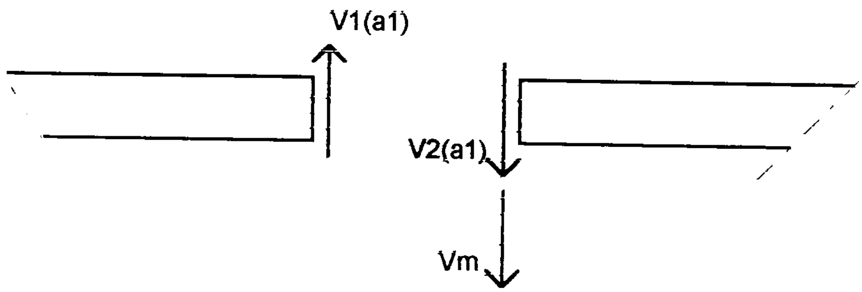

Figure 3.8 Free Body Diagram of Joint 1

$$A_2 \sinh(\beta(a_2 - a_1)) + B_2 \cosh(\beta(a_2 - a_1)) - C_2 \sin(\beta(a_2 - a_1))$$
$$+ D_2 \cos(\beta(a_2 - a_1)) - A_3 \sinh(\beta(a_2 - a_1)) - B_3 \cosh(\beta(a_2 - a_1)) \quad (3.56)$$
$$+ C_3 \sin(\beta(a_2 - a_1)) - D_3 \cos(\beta(a_2 - a_1)) = 0$$

$$A_2 \cosh(\beta(a_2 - a_1)) + B_2 \sinh(\beta(a_2 - a_1)) - C_2 \cos(\beta(a_2 - a_1))$$
$$- D_2 \sin(\beta(a_2 - a_1)) - A_3 \cosh(\beta(a_2 - a_1)) - B_3 \sinh(\beta(a_2 - a_1)) \quad (3.57)$$
$$+ C_3 \cos(\beta(a_2 - a_1)) + D_3 \sin(\beta(a_2 - a_1)) = 0$$

For the forth boundary condition at this joint the shear force introduced by the rear support bearing must be accounted. Figure 3.9 illustrates the free body diagram at the joint. The shear force introduced by the bearing is proportional to the shaft's displacement at the joint.

$$V_{kr} = K_r \bar{y}_3(a_2) \quad (3.58)$$

For equilibrium at the joint:

$$V_2(a_2) - V_3(a_2) = V_{kr} \quad (3.59)$$

$$EI\bar{y}_2'''(a_2) - EI\bar{y}_3'''(a_2) = K_r \bar{y}_2(a_2) \quad (3.60)$$

Substituting equation 3.9 into equation 3.60:

$$A_2 \sinh(\beta(a_2 - a_1)) + B_2 \cosh(\beta(a_2 - a_1)) + C_2 \sin(\beta(a_2 - a_1))$$
$$- D_2 \cos(\beta(a_2 - a_1)) - A_3[\sinh(\beta(a_2 - a_1)) + \frac{K_r}{\beta^3 EI} \cosh(\beta(a_2 - a_1))]$$
$$- B_3[\cosh(\beta(a_2 - a_1)) + \frac{K_r}{\beta^3 EI} \sinh(\beta(a_2 - a_1))] + C_3[\sin(\beta(a_2 - a_1)) \quad (3.61)$$
$$+ \frac{K_r}{\beta^3 EI} \cos(\beta(a_2 - a_1))] + D_3[\cos(\beta(a_2 - a_1)) - \frac{K_r}{\beta^3 EI} \sin(\beta(a_2 - a_1))] = 0$$

The first two boundary conditions for the joint between the third and fourth section are comprised of the continuity conditions ($y_3 = y_4$ and $y_3' = y_4'$).

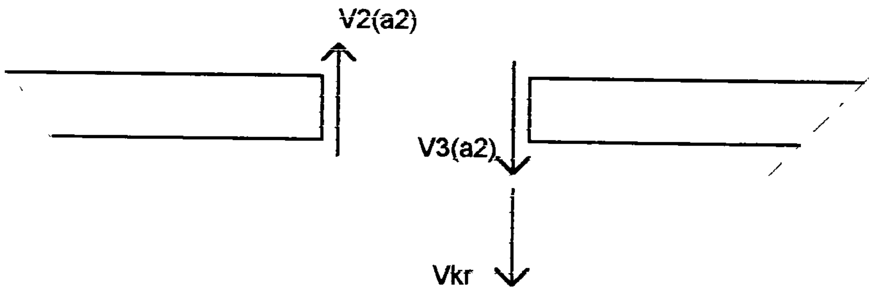

Figure 3.9 Free Body Diagram of Joint 2

Therefore:

$$\begin{aligned}
&A_2\cosh(\beta(a_2-a_1))+B_2\sinh(\beta(a_2-a_1))+C_2\cos(\beta(a_2-a_1))\\
&+D_2\sin(\beta(a_2-a_1))-A_3\cosh(\beta(a_2-a_1))-B_3\sinh(\beta(a_2-a_1))\\
&-C_3\cos(\beta(a_2-a_1))-D_3\sin(\beta(a_2-a_1))=0
\end{aligned} \tag{3.62}$$

$$\begin{aligned}
&A_2\sinh(\beta(a_2-a_1))+B_2\cosh(\beta(a_2-a_1))-C_2\sin(\beta(a_2-a_1))\\
&+D_2\cos(\beta(a_2-a_1))-A_3\sinh(\beta(a_2-a_1))-B_3\cosh(\beta(a_2-a_1))\\
&+C_3\sin(\beta(a_2-a_1))-D_3\cos(\beta(a_2-a_1))=0
\end{aligned} \tag{3.63}$$

The third and forth boundary conditions are influenced by the bending moment and shear force associated with the torsional and lateral stiffness of the front support bearing. Figure 3.10 illustrates the free body diagram at joint 3.

For equilibrium at the joint:

$$V_3(a_3)-V_4(a_3)=V_{kf} \tag{3.64}$$

$$EI\bar{y}_3'''(a_3)-EI\bar{y}_4'''(a_3)=K_f\bar{y}_4(a_3) \tag{3.65}$$

and

$$M_3(a_3)-M_4(a_3)=M_{ktf} \tag{3.66}$$

$$EI\bar{y}_3''(a_3)-EI\bar{y}_4''(a_3)=K_{tf}\bar{y}_4'(a_3) \tag{3.67}$$

Substituting eq. 3.9 into eqs. 3.65 and 3.67 yields:

$$\begin{aligned}
&A_3\sinh(\beta(a_3-a_2))+B_3\cosh(\beta(a_3-a_2))+C_3\sin(\beta(a_3-a_2))\\
&-D_3\cos(\beta(a_3-a_2))-A_4[\sinh(\beta(a_3-a_2))+\frac{K_f}{\beta^3EI}\cosh(\beta(a_3-a_2))]\\
&-B_4[\cosh(\beta(a_3-a_2))+\frac{K_f}{\beta^3EI}\sinh(\beta(a_3-a_2))]+C_4[\sin(\beta(a_3-a_2))\\
&+\frac{K_f}{\beta^3EI}\cos(\beta(a_3-a_2))]+D_4[\cos(\beta(a_3-a_2))-\frac{K_f}{\beta^3EI}\sin(\beta(a_3-a_2))]=0
\end{aligned} \tag{3.68}$$

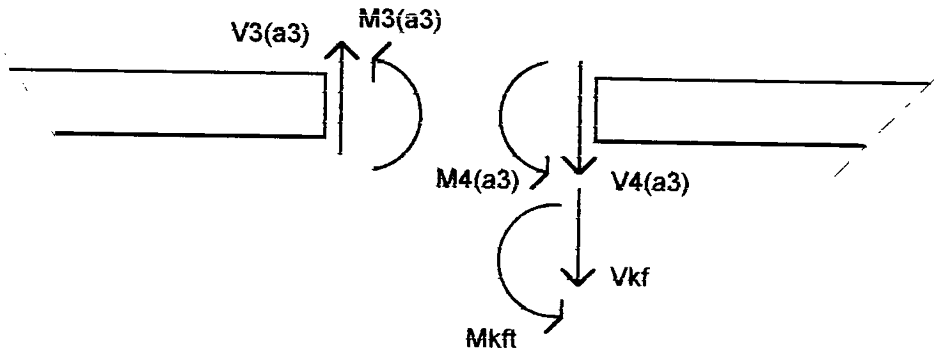

Figure 3.10 Free Body Diagram of Joint 3

and

$$
\begin{aligned}
&A_3 \cosh(\beta(a_3 - a_2)) + B_3 \sinh(\beta(a_3 - a_2)) - C_3 \cos(\beta(a_3 - a_2)) \\
&- D_3 \sin(\beta(a_3 - a_2)) - A_4[\cosh(\beta(a_3 - a_2)) + \frac{K_{tf}}{\beta^2 EI} \sinh(\beta(a_3 - a_2))] \\
&- B_4[\sinh(\beta(a_2 - a_1)) + \frac{K_{tf}}{\beta^2 EI} \cosh(\beta(a_2 - a_1))] + C_4[\cos(\beta(a_2 - a_1)) \\
&+ \frac{K_{tf}}{\beta^2 EI} \sin(\beta(a_3 - a_2))] + D_4[\sin(\beta(a_3 - a_2)) - \frac{K_r}{\beta^2 EI} \cos(\beta(a_3 - a_2))] = 0
\end{aligned}
\tag{3.69}
$$

The final two boundary conditions are related to the cutting end of the spindle. The first of these conditions relates to the bending moment. Since the rotary inertia of the cutting tool is neglected the moment at the end of the spindle is equal to zero. Therefore:

$$M_1(0) = EI \frac{d^2 \bar{y}_1}{dx^2} = 0 \tag{3.70}$$

$$
\begin{aligned}
&A_4 \cosh(\beta(a_4 - a_3)) + B_4 \sinh(\beta(a_4 - a_3)) \\
&- C_4 \cos(\beta(a_4 - a_3)) - D_4 \sin(\beta(a_4 - a_3)) = 0
\end{aligned}
\tag{3.71}
$$

The last boundary condition involves the shear force at the end of the shaft. The shear force is equal to the D'Alembert force associated with the mass of the tool. Therfore:

$$V(a_4) = m_t \ddot{y}(a_4) = -m_t \omega^2 \bar{y}(a_4) \tag{3.72}$$

Substituting equation 3.9:

$$A_4[\sinh(\beta(a_4-a_3))+\frac{m_t\omega^2}{\beta^3 EI}\cosh(\beta(a_4-a_3))]+B_4[\cosh(\beta(a_4-a_3))$$
$$+\frac{m_t\omega^2}{\beta^3 EI}\sinh(\beta(a_4-a_3))]+C_4[\sin(\beta(a_4-a_3))+\frac{m_t\omega^2}{\beta^3 EI}\cos(\beta(a_4-a_3))] \quad (3.73)$$
$$-D_4[\cos(\beta(a_4-a_3))-\frac{m_t\omega^2}{\beta^3 EI}\sin(\beta(a_4-a_3))]=0$$

In order to solve the set of simultaneous equations, the set of sixteen equations and sixteen unknowns were collected into a matrix.

3.2 Matlab Solution for Mode Shapes:

A program was developed using *Matlab* to automate the modal analysis of the spindle shaft. Data is collected and entered into a spreadsheet. This sheet acts as the batch file for the modal analysis. Much like the static analysis, the user must enter the geometry, mass information, and support parameters into the batch file. A copy of the batch file template is presented in Appendix A. The *Matlab* programming code used to automate the modal analysis can be found in Appendix C.

An example of the analysis for a simple spindle is presented here. Figure 3.11 illustrates the batch file for the modal analysis. The "grayed out" information does not pertain to the modal analysis. Upon the completion of the batch file the program will read the file and report a geometric representation of the information. With all the information correct the program calculates and reports the resonant frequencies for the sample spindle. The sample spindle was also modeled using *Ansys*. A comparison between the FEA and analytical results for the first three modes is presented in Figures 3.12 through 3.14.

Batch File:

Geometry:

Number of Sections(#): 6

Section (#)	Length (in)	Outer Diameter (in)	Inner Diameter (in)	Area (in^2)	Moment of Inertia (in^4)
1	3	2.25	2	0.83449	0.47265783
2	3	2.375	2.125	0.88357	0.560861726
3	3	2.5	2.25	0.93266	0.659419991
4	3	2.625	2.375	0.98175	0.76890787
5	3	2.75	2.5	1 03084	0.889900605
6	3	2.875	2.625	1 07992	1.022973438
7				0	0
8				0	0
9				0	0
10				0	0
11				0	0
12				0	0
13				0	0
14				0	0
15				0	0

Bearings:

Lat. Stiffness of Rear Bearing (lb/in):	100000
Lat. Stiffness of Front Bearing (lb/in.):	500000
Tor. Stiffness of Front Bearing (in-lb):	10000
Fraction of [illegible] on Front Bearing:	0.1
Location of Rear Bearing (in.)	7.5
Location of Front Bearing (in.)	13.5

Pulley:

Location of Pulley (in.)	4.5
Mass of Pulley (lb-s^2/in):	0 007763975
Static Belt Tension (lb):	50
Harmonic Drive Force (lb):	50
Drive Frequency (Hz):	133.33
Pulley Unbalance (lb-s^2):	0.002279727

Tool:

Mass of Tool (lb-s^2/in):	0 025879917
Static Cutting Force (lb):	300
Harmonic Cutting Force (lb):	300
Cutting Frequency (Hz):	133.33
Tool Unbalance (lb-s^2):	0.004559453
Length of Tool (in)	2

Speed:

Spindle Shaft Speed (Hz):	16.66666667

Material Properties:

Modulus of Elasticity (psi):	30000000
Density (lb/in^3):	0.289

Figure 3.11 Batch File for Sample Spindle

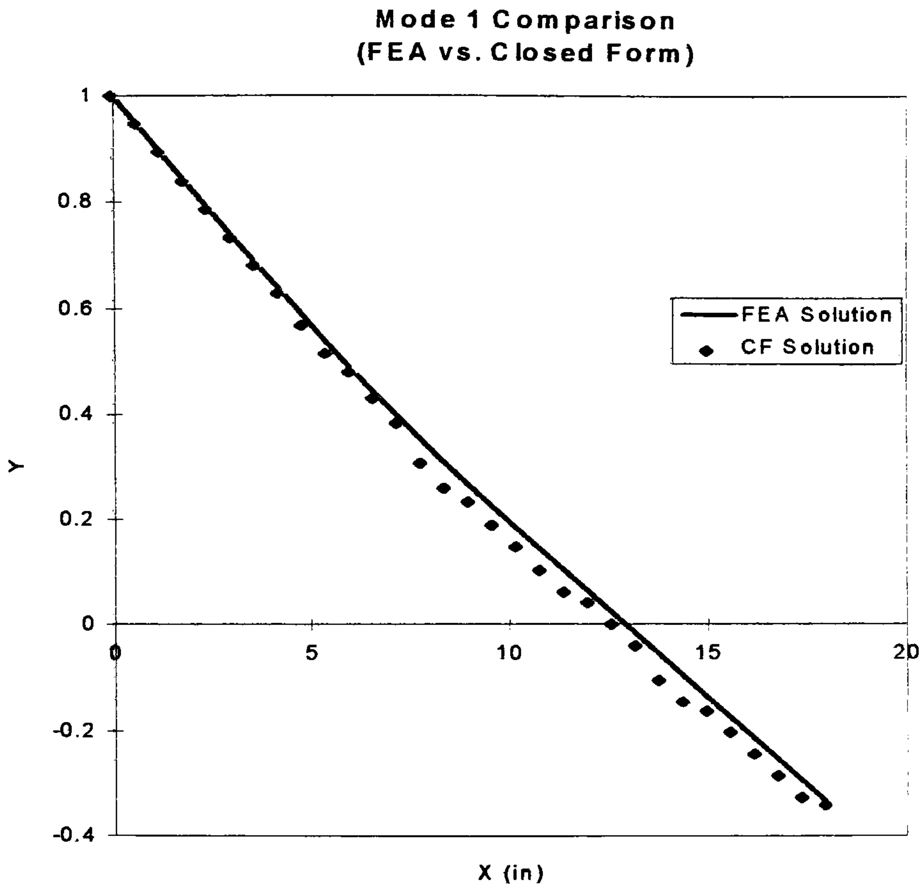

Figure 3.12 Mode 1 Comparison

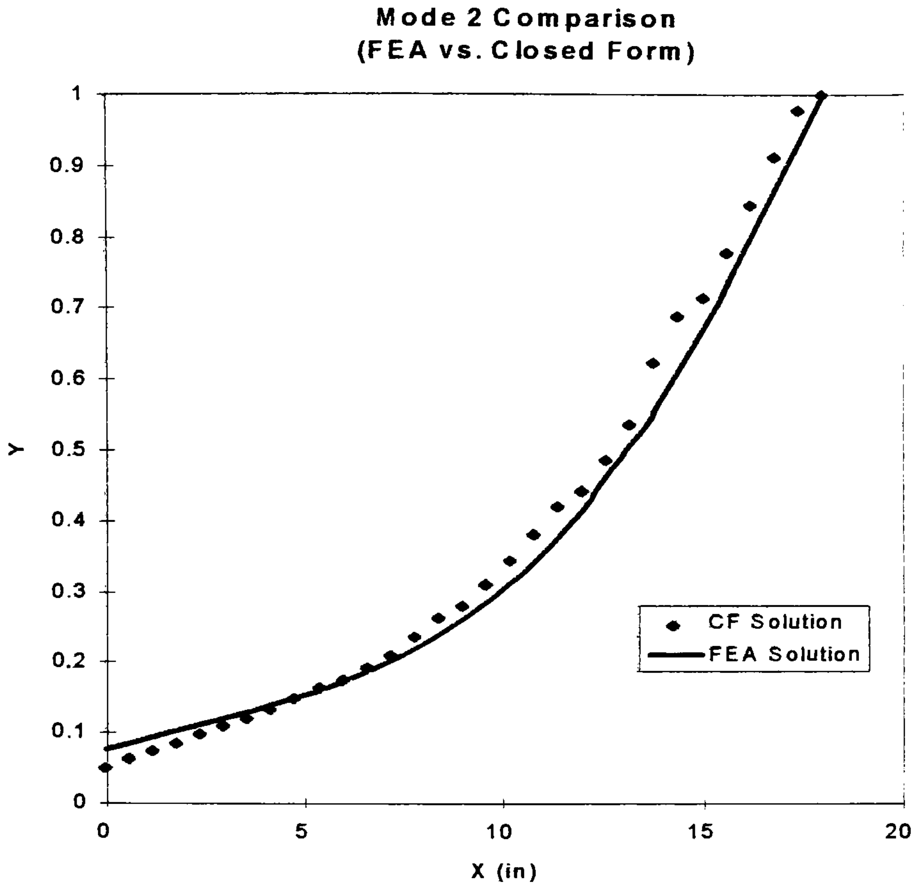

Figure 3.13 Mode 2 Comparison

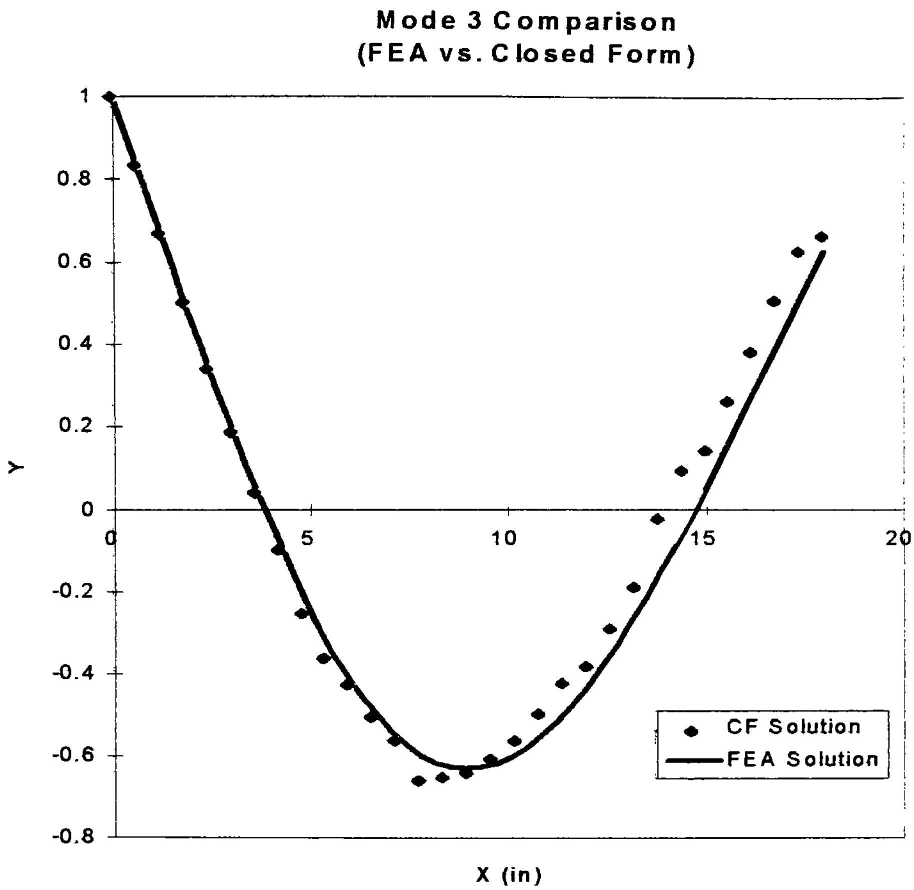

Figure 3.14 Mode 3 Comparison

The resonant frequencies for the first three modes are compared in table 3.1. It is clear from the table that the two methods correlate very closely for the two methods.

Table 3.1 Comparison of Resonant Frequencies (FEA vs. Analytical)

Mode	FEA	Analytical	Difference
	(Hz)	(Hz)	(%)
1	19.26	19.17	0.4672897
2	61.28	61.72	0.7180157
3	95.57	100.33	4.9806425

3.3 Forced Response:

The forced response of the spindle is calculated using a numeric modal summation procedure. The development of the forced response begins with the equation of motion for a beam, Dahleh et. al, (1989).

$$\left[EIy''(x,t)\right]'' + m(x)\ddot{y}(x,t) = f(x,t) \tag{3.73}$$

The normal modes for the beam, $\phi_i(x)$, must satisfy the following equation:

$$(EI\phi_i'')'' - \omega_i^2 m(x)\phi_i = 0 \tag{3.74}$$

In addition to eqn. 3.74, since the normal modes are orthogonal they must also satisfy the following equation:

$$\int_0^l \phi_i \phi_j dx = 0 \quad \text{for } i \neq j \tag{3.75}$$

The solution to the forced response can be represented in terms $\phi_i(x)$ as:

$$y(x,t) = \sum_i \phi_i(x) q_i(t) \tag{3.76}$$

Where $q_i(t)$ is the generalized coordinate. The generalized coordinate can be realized using the Lagrange Equation. Looking first at the kinetic energy yields:

$$T = \frac{1}{2}\int_0^l \dot{y}^2(x,t)m(x)dx$$

Substituting eqn. 3.76 for y(x,t) yields:

$$T = \frac{1}{2}\sum_i \sum_j q_i q_j \int_0^l \phi_i \phi_j m(x)dx$$

$$T = \frac{1}{2}\sum_i M_i \dot{q}_i^2 \tag{3.77}$$

Where the generalized mass, M_i is defined as:

$$M_i = \int_0^l \phi_i^2(x)m(x)dx \tag{3.78}$$

The potential energy, U can be defined as:

$$U = \frac{1}{2}\int_0^l EIy''^2(x,t)dx$$

$$U = \frac{1}{2}\sum_i \sum_j \phi_i \phi_j \int_0^l EI\phi_i'' \phi_j'' dx$$

$$U = \frac{1}{2}\sum_i K_i q_i^2 \tag{3.79}$$

Where the generalized stiffness, K_i equals:

$$K_i = \int_0^l EI[\phi_i''(x)]^2 dx \tag{3.80}$$

If eqn. 3.74 is substituted into eqn. 3.79 it can be shown that:

$$U = -\frac{1}{2}\sum_i \omega_i^2 M_i q_i^2 \tag{3.81}$$

A generalized force, Q_i can be defined by looking at the work done by a virtual displacement, δq_i.

$$\delta w_i = \int_0^l f(x,t)\sum_i \phi_i \delta q_i dx$$

rearranging:

$$\delta w_i = \sum_i \delta q_i Q_i \tag{3.82}$$

where:

$$Q_i = \int_0^l f(x,t)\phi_i(x)dx \tag{3.83}$$

From the Lagrange Equation:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = Q_i \tag{3.84}$$

Substituting for the kinetic energy, potential energy, and the generalized force yields the following differential equation:

$$\ddot{q}_i + \omega_i^2 q_i = \frac{\int_0^l f(x,t)\phi_i(x)dx}{\int_0^l \phi_i^2(x)m(x)dx} \tag{3.85}$$

where:

$$\omega_i^2 = \frac{K_i}{M_i} \tag{3.86}$$

The model of the spindle assumes four simple harmonic loads. The harmonic loads include the drive force, cutting force, unbalance of the pulley and unbalance of the cutting tool (see fig. 3.1). All four of the forces are assumed to be in phase with each other and of the form:

$$f(x,t) = F(x)\sin(\omega t) \tag{3.87}$$

Each of the forces are applied to a single point. Assuming the force is applied at x = x_o, it can be described using the delta dirac function as:

$$f(x,t) = F\sin(\omega t)\delta(x - x_o) \tag{3.88}$$

By definition the delta dirac function is equal to zero for all x not equal to x_o. Further it can be shown that:

$$\int_0^\infty F(x)\delta(x - x_o)dx = F(x_o) \tag{3.89}$$

Substituting this relationship into eqn. 3.85 yields:

$$\ddot{q}_i + \omega_i^2 q_i = \frac{\phi_i(x_o)F\sin(\omega t)}{\int_0^l \phi_i^2(x)m(x)dx} \tag{3.90}$$

Assuming the following solution to eqn. 3.90:

$$q_i(t) = q_i \sin(\omega t) \tag{3.91}$$

yields:

$$q_i = \frac{\phi_i(x_o)F}{\left(\omega_i^2 - \omega^2\right)\int_0^l \phi_i^2 m dx} \tag{3.92}$$

The denominator of eqn. 3.92 must be broken down for the four sections of the spindle and each of the segments (steps) in the shaft described in the modal analysis.

$$q_i = \frac{\phi_i(x_o)F}{\left(\omega_i^2 - \omega^2\right)\sum_{n=1}^{N} m_n \sum_{k=1}^{4} \int_0^{a_k} \left(A_k \cosh(\beta_i x) + B_k \sinh(\beta_i x) + C_k \cos(\beta_i x) + D_k \sin(\beta_i x)\right)^2 dx} \tag{3.93}$$

For this analysis only the summation of the first four modes were utilized. After the first four modes the difference between the resonant frequencies and the drive frequencies become large and q_i approaches zero. Therefore the steady state response becomes:

$$Y = \phi_1 q_1 + \phi_2 q_2 + \phi_3 q_3 + \phi_4 q_4 \tag{3.94}$$

The deflections, Y were calculated for each of the four excitation forces and superposed to yield the total forced response:

$$Y_t = Y_{fc} + Y_{fdub} + Y_{fc} + Y_{fcub} \tag{3.95}$$

<u>3.4 Matlab Solution for Forced Response:</u>

A program was developed using *Matlab* to automate the calculation of the forced response for the spindle shaft. The magnitude and frequency of the excitation forces is entered into a batch file. In addition to the load information the program reads the first four modes calculated in the modal analysis program. The *Matlab* programming code used to automate the forced response can be found in Appendix C.

An example of the analysis for a simple spindle is presented here. Figure 3.15 illustrates the batch file used for this example problem. The "grayed out" information does not pertain to this analysis. It should be noted that the program will not function

Batch File:

Geometry:

Number of Sections(#):	6

Section (#)	Length (in)	Outer Diameter (in)	Inner Diameter (in)	Area (in^2)	Moment of Inertia (in^4)
1	3	2.25	2	0.834486	0.47265783
2	3	2.375	2.125	0.883573	0.560861726
3	3	2.5	2.25	0.93266	0.659419991
4	3	2.625	2.375	0.981748	0.76890787
5	3	2.75	2.5	1.030835	0.889900605
6	3	2.875	2.625	1.079922	1.022973438
7				0	0
8				0	0
9				0	0
10				0	0
11				0	0
12				0	0
13				0	0
14				0	0
15				0	0

Bearings:

Lat. Stiffness of Rear Bearing (lb/in):	100000
Lat. Stiffness of Front Bearing (lb/in.):	500000
Tor. Stiffness of Front Bearing (in-lb):	10000
Fraction of mom. on Front Bearing:	[illegible]
Location of Rear Bearing (in.)	7.5
Location of Front Bearing (in.)	13.5

Pulley:

Location of Pulley (in.)	4.5
Mass of Pulley (lb-s^2/in):	0.007763975
Static Belt Tension (lb):	[illegible]
Harmonic Drive Force (lb):	50
Drive Frequency (Hz):	133.33
Pulley Unbalance (lb-s^2):	0.002279727

Tool:

Mass of Tool (lb-s^2/in):	0.025879917
Static Cutting Force (lb):	[illegible]
Harmonic Cutting Force (lb):	300
Cutting Frequency (Hz):	133.33
Tool Unbalance (lb-s^2):	0.004559453
Length of Tool (in)	2

Speed:

Spindle Shaft Speed (Hz):	16.66666667

Material Properties:

Modulus of Elasticity (psi):	30000000
Density (lb/in^3):	0.289

Figure 3.15 Batch File for Sample Spindle Forced Response

properly if the modal analysis from section 3.2 is not completed first. The sample spindle was also modeled using *Ansys*. A comparison between the FEA and analytical results for the forced response is presented in Figure 3.16. The comparison between the FEA and analytical responses shows a close correlation between the two methods. There is a 6.5% difference in the deflection at the tool for the two methods.

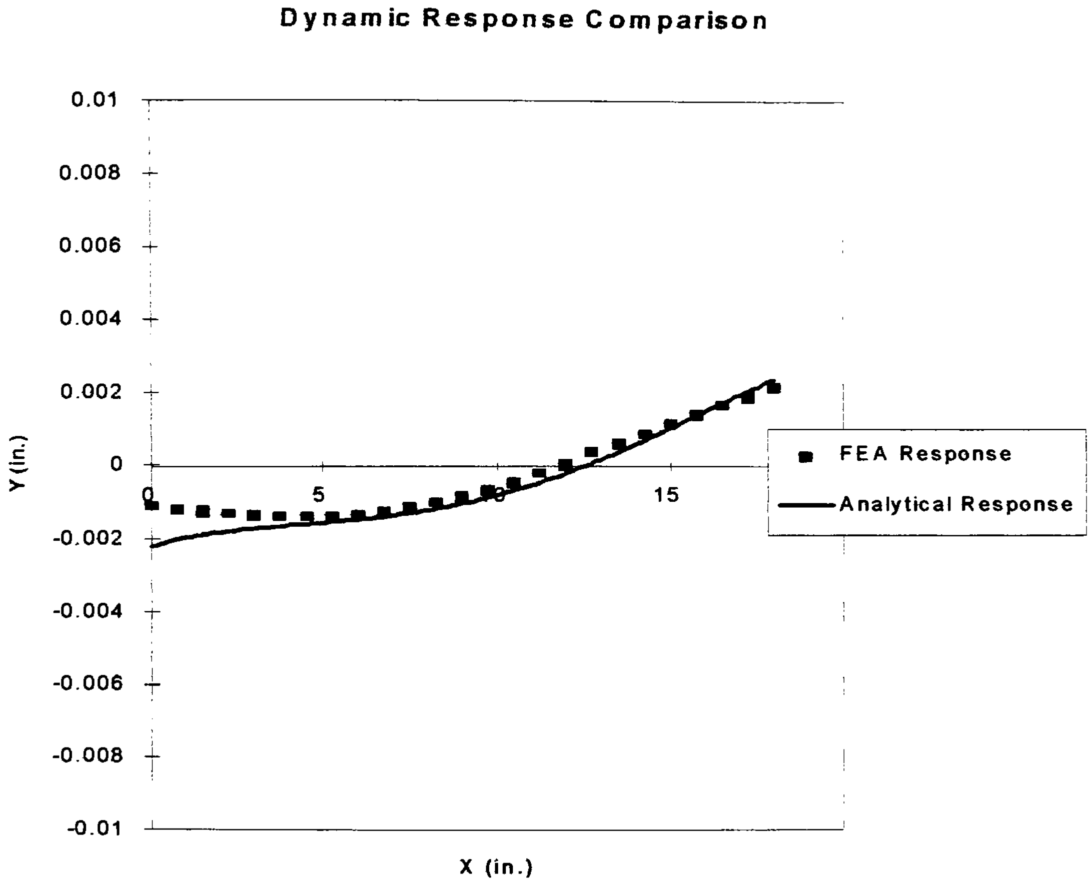

Figure 3.16 Comparison of Forced Response FEA vs. Analytical

4.0 Optimization Analysis:

The optimization analysis consists of minimizing the deflection of the spindle shaft at the gauge line (see figure 4.1). This analysis builds upon the static as well as the dynamic analysis. Optimal parameters are offered for both cases. The following assumptions apply to the optimization analysis:

1. The design variables for this analysis are the lateral stiffness and the position of the two bearings. All other parameters are assumed to be constant.
2. Each design iteration is approximated using the Taylor series expansion. This approximation is required to define a quadratic programming subproblem.
3. The optimization point may or may not be the global minimum. However the values assure a local minimum.

4.1 Optimization Model:

The development of the optimization problem rests in minimizing a cost function, f(**x**), where **x** is the design variable vector. For the optimization of the machine tool spindle the cost function, f is defined as the deflection at the spindle's gauge line.

$$f(\mathbf{x}) = y_t(a_4) \tag{4.1}$$

Given values for the design parameters, a value for $y_t(a_4)$ can be obtained numerically using the *Matlab* routines developed in Chapters 2& 3. The design variables, **x** are listed in table 4.1. The remainder of the spindle design parameters are assumed to be fixed. This is a fairly accurate assessment since for an existing spindle design the other parameters would significantly influence the supporting components (i.e. gearbox and spindle housing).

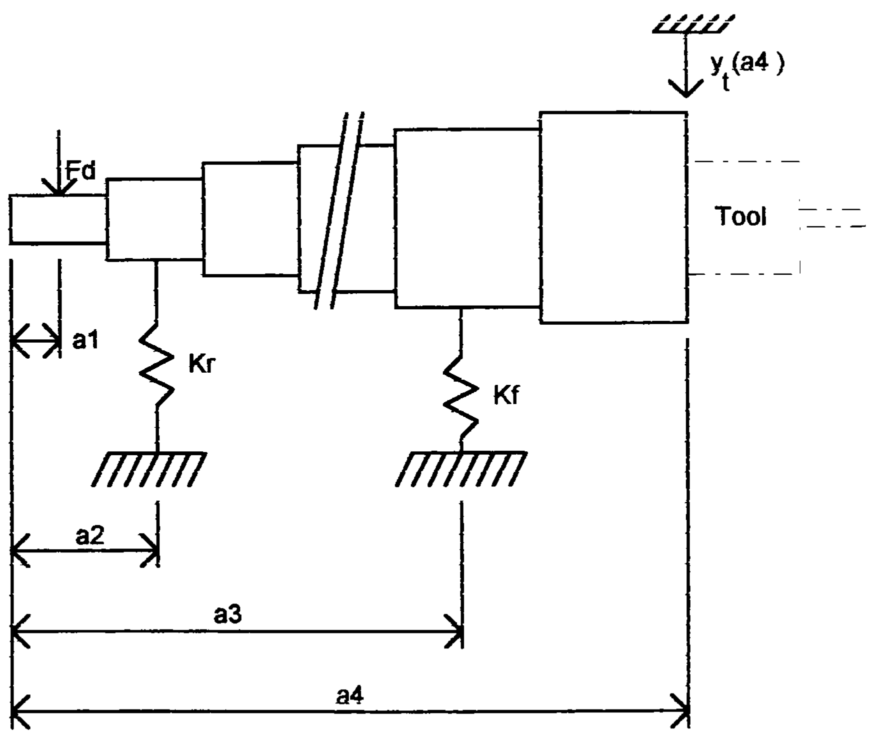

Figure 4.1 Optimization Model of Spindle

Table 4.1 Table of Design Variables

Design Variable Vector	Parameter
x(1)	a(2), postion of rear brg
x(2)	a(3), position of front brg
x(3)	Kf, lateral stiffness of front brg
x(4)	Kr, lateral stiffness of rear brg

General constrained optimum design defines the following equality and inequality constraints respectively:

$$h_j(\mathbf{x}) = 0 \tag{4.2}$$

$$g_i(\mathbf{x}) \le 0 \tag{4.3}$$

For this optimization problem there exists no equality constraints. The following equations define the inequality constraints.

$$x_1 \ge a_1 + D \tag{4.4}$$

$$x_2 \le a_4 - OH \tag{4.5}$$

$$x_3 \le K_{f\max} \tag{4.6}$$

$$x_4 \le K_{r\max} \tag{4.7}$$

To summarize these constraints, the first constraint (eqn. 4.4) stipulates that the location of the rear bearing must be beyond the location of the pulley by a distance, D. This is required to ensure that the pulley is "outboard" of the support bearings and there is sufficient spacing to accommodate the width of the pulley and the width of the bearing. The second constraint (eqn. 4.5) requires that there exist a sufficient overhang to accommodate features in the spindle shaft to accept and support the tool. The third and forth constraints (eqns. 4.6-4.7) ensure that the bearings' stiffness values are physically obtainable. Without these constraints the optimization could potentially specify a bearing with an infinite lateral stiffness.

Prior to developing the process used for this optimization analysis it is important to first introduce the Lagrange function and the Lagrange Multiplier Theorem, Arora, (1989). For general constrained optimization the form of the Lagrange equation is:

$$L(\mathbf{x},\mathbf{v},\mathbf{u}) = f(\mathbf{x}) + \sum_{j=1}^{n} v_j h_j(\mathbf{x}) + \sum_{i=1}^{m} u_i (g_i(\mathbf{x}) + s_i^2) \tag{4.8}$$

Since there are no equality constraints in this analysis the Lagrange equation reduces to:

$$L(\mathbf{x},\mathbf{u}) = f(\mathbf{x}) + \sum_{i=1}^{m} u_i (g_i(\mathbf{x}) + s_i^2) \tag{4.9}$$

From eqn 4.9, f($\mathbf{x}$) is the cost function, m is the number of constraint equations, u is the lagrange mulitplier for the i^{th} constraint equation, g_i is the i^{th} constraint equation, and s_i the slack variable for the i^{th} constraint equation. The slack variable is a constant that converts the inequality constraint to an equality constraint.

$$g_i(\mathbf{x}) + s_i^2 = 0 \tag{4.10}$$

If the design point, $\mathbf{x}$ is a local minimum the Lagrange Mulitplier Theorem stipulates the following Kuhn-Tucker Conditions:

$$\frac{\partial L}{\partial x_j} = 0 \qquad \text{for j = 1 to n} \tag{4.11}$$

$$\frac{\partial L}{\partial u_i} = 0 \qquad \text{for i = 1 to m} \tag{4.12}$$

$$\frac{\partial L}{\partial s_i} = 0 \qquad \text{for i = 1 to m} \tag{4.13}$$

If the i^{th} constraint is inactive s_i is equal to zero. If the i^{th} constraint is active u_i is equal to zero. Therefore:

$$u_i s_i = 0 \tag{4.14}$$

The Lagrange multipliers and the slack variables can be found by solving this system of equations (eqns 4.11-4.14).

In order to apply numerical methods to solve for the design change an approximate quadratic programming subproblem (QP subproblem) was defined. The QP subproblem can be obtained from a Taylor series expansion of the cost function. It has a quadratic cost function and linear constraints. The problem is defined as the minimization of:

$$\bar{f} = \mathbf{c}^t\mathbf{d}^{(k)} + 0.5(\mathbf{d}^t\mathbf{d})^{(k)} \tag{4.15}$$

where:

$$\bar{f} \cong f(\mathbf{x}^{(k)} + \mathbf{d}^{(k)}) - f(\mathbf{x}^{(k)}) \tag{4.16}$$

subject to the following constraints:

$$\mathbf{A}^t\mathbf{d}^{(k)} \leq \mathbf{p} \tag{4.17}$$

where $\mathbf{d}^{(k)}$ is a vector of changes in the design variables for the k^{th} design point, $\mathbf{c}$ is a vector containing the gradient of the cost function $f(\mathbf{x}^{(k)})$, and $\mathbf{A}$ is the gradient of the inequality constraints.

In order to solve the QP problem a search direction and a step size must be determined. The constrained steepest descent method was used to solve for these two entities. When no constraints exist the search direction is simply in the direction of the negative of the gradient vector (**d**=-**c**). In the case of the spindle optimization constraints exist and they must be included in the development of a search direction.

In order to accommodate the constraints in solving for a search direction a descent function must be defined for the constrained problem. A descent function must possess two properties. First, it must be equal to the cost function at the optimum point. Next it must allow for a unit step size near the optimum point. This is important because a unit step size will yield a high rate of convergence. The Pshenichny's descent function Φ was chosen since it obeys these two rules.

$$\Phi(\mathbf{x}) = f(\mathbf{x}) + RV(\mathbf{x}) \tag{4.18}$$

In eqn 4.18, R is the penalty parameter and V is the maximum constraint violation. The user specifies the initial value of R. A subsequent value for R is calculated at the end each iteration in the optimization process. In order to satisfy the necessary condition the penalty parameter must be greater than or equal to the sum of Lagrange multipliers at the k^{th} iteration.

$$R \geq r_k \tag{4.19}$$

For the m constraint equations, r_k is defined as:

$$r_k = \sum_{i=1}^{m} u_i^{\,k} \tag{4.20}$$

where u_i^k is the Lagrange multiplier for the i^{th} constraint at the k^{th} design point. The Lagrange multipliers can be found by solving the system of equations previously mentioned (eqns. 4.11-4.14).

The maximum constraint violation at the k^{th} iteration, V_k is defined as:

$$V_k = \max\{0; g_1, g_2, \ldots, g_m\} \tag{4.21}$$

The next step in solving the optimization problem is to define a step size determination procedure. The decent function will yield the search direction, the step size determination will dictate how far to adjust the design variables in that direction. For this analysis an inexact line search method was used. For this method a sequence of trial step sizes, t_j was defined.

$$t_j = \left(\frac{1}{2}\right)^j \quad \text{for j=0,1,2,\ldots} \tag{4.22}$$

Each iteration begins with the trial step size $t_0=1$. If a defined descent condition is not satisfied the step size is cut in half ($t_1=1/2$). For a step size iteration, j and a descent iteration, k the new design variable vector is defined as:

$$\mathbf{x}^{(k+1,j)} = \mathbf{x}^{(k)} + t_j \mathbf{d}^{(k)} \tag{4.23}$$

The acceptable step size will be the smallest integer j that satisfies the descent condition.

$$\Phi_{k+1,j} \leq \Phi_k - t_j \beta_k \tag{4.24}$$

where $\Phi_{k+1,j}$ is the descent function defined in eqn. 4.11 evaluated at the trial step size. The constant β_k is found using the search direction, $\mathbf{d}^{(k)}$

$$\beta_k = \gamma \left\| \mathbf{d}^{(k)} \right\|^2 \tag{4.25}$$

The constant γ is specified by the user and has a value between 0 and 1. The value of γ affects the allowable step size. Larger values of γ will result in smaller values for the step size. The end result is a slower rate of convergence. Alternatively very small values for

γ can lead to instabilities in the optimization process. Typically experimentation takes place to find a suitable value for the engineering problem being solved.

Iterations of search direction and step size are continued until the method converges on a local minimum for the cost function. Convergence is defined as the design point were

$$\|\mathbf{d}\| \leq \varepsilon_1 \tag{4.26}$$

where ε_1 is a specified small positive number.

4.2 The Constrained Steepest Descent Algorithm:

A CSD algorithm was used to optimize the design variables in a spindle shaft. This section describes the steps to this algorithm.

The first step to the CSD algorithm is to set the counter, k equal to zero. At this step initial values for the design variables **x**, the penalty parameter R, the constant γ, and the convergence criteria ε_1. An additional convergence criteria was also added to the analysis. It was stipulated that the maximum constraint violation, V_k must not exceed a predefined value ε_2. This assures that design points with excessive constraint violations are not allowed. A value for this constant is also needed at this step. Since the goal of this analysis is to optimize an existing spindle design the initial values for the design variables would simply be the parameters used in the existing design. The initial value for the penalty parameter was defined as R=1. The constant γ was defined as 0.5. Finally the values for the convergence constants ε_1 and ε_2 were both defined to be 0.1.

The next step is to calculate values for the cost and constraint functions as well as their gradients. It is important to note here that the design variables were normalized for this analysis. Since the magnitudes of the variables vary significantly it would be inappropriate to use there gradients in obtaining a search directions and step size. The gradient of the cost function was calculated by applying the forward difference method to the static and dynamic model previously developed.

$$\frac{\partial f}{\partial x_j} = \frac{f(x_j + \Delta_j) - f(x_j)}{\Delta_j} \tag{4.27}$$

where

$$\Delta_j = 0.001 x_j \tag{4.28}$$

The forward difference method was selected because it only required two calculations of the cost function. This helped to speed up the time to convergence. The central difference method would have required three calculations for each design variable. All of the constraints are linear so their gradients were easily obtainable analytically. The final calculation at this step is the maximum constraint violation V_k (see eqn. 4.21).

The third step is to use the information from the first two steps to define the QP subproblem (eqns. 4.15-4.17). At this point the QP subproblem can be used to solve for the search direction, **d** and Lagrange multipliers, **u**. In order to obtain these values the cost function, f(**x**) in the Lagrange Equation must be replaced by the QP cost function (eqn. 4.15) and the system of equations, 4.11 – 4.14, must be solved simultaneously.

The forth step is to check the convergence criteria to see that:

$$\|\mathbf{d}\| \le \varepsilon_1$$

and

$$V_k \leq \varepsilon_2$$

If these conditions are satisfied the algorithm has converged and the analysis can stop. Otherwise continue to step five of the algorithm.

The next step of the analysis is to modify the penalty parameter, R. For the k^{th} iteration the new penalty parameter R_{k+1} becomes:

$$R = \max\{R_k, r_k\} \tag{4.29}$$

where R_k is the existing penalty parameter and r_k is the sum of the Lagrange multipliers calculated in third step of the algorithm. By updating the penalty parameter the necessary condition will always be satisfied.

Next the step size must be determined. The inexact line search method previously developed was used here to calculate the proper step size. Once the step size is determined the design point can be indexed. Therefore:

$$\mathbf{x}^{(k+1,j)} = \mathbf{x}^{(k)} + t_j \mathbf{d}^{(k)}$$

The final step to the algorithm is to index the counter, k=k+1, and repeat all but the first step. Iterations will continue until convergence is reached.

4.3 Matlab Solution:

The CSD algorithm was implemented for the optimization of the spindle shaft using *Matlab*. The optimization applied to both the static and the dynamic models developed in Chapters 3 and 4. The optimization program will return optimal values for the lateral stiffness and location of the spindle support bearings. As in previous chapters the Matlab program reads in a batch file. The batch file allows the user to define the spindle parameters. The optimization constants used by the CSD algorithm were hard coded into the program. Therefore the user of the program does not have the flexibility to change these. The programming code used to perform the optimization analysis can be found in Appendix D.

The sample spindle analyzed statically and dynamically in Chapters 2 and 3 was also optimized to demonstrate the Optimization program. Figure 4.2 illustrates the batch file read in by the program. The optimum parameters returned by the program for the static and dynamic analysis are listed in tables 4.2 and 4.3 respectively.

Table 4.2 Optimum Values for Static Analysis

Design Variable	Optimum Value	Original Value
a_2 (in.)	7.47	7.5
a_3 (in.)	16.09	13.5
K_f (lbf/in.)	1000000	500000
K_r (lbf/in.)	1000000	100000

Table 4.3 Optimum Values for Dynamic Analysis

Design Variable	Optimum Value	Original Value
a_2 (in.)	7.3	7.5
a_3 (in.)	16.15	13.5
K_f (lbf/in.)	1000000	500000
K_r (lbf/in.)	1000000	100000

The static deflection of the existing spindle was approximately .0048" The corresponding static deflection of the optimized spindle was approximately .00079". The optimization reduced the deflection by a factor of 6. For the dynamic analysis the initial deflection was about .0023" The corresponding deflection of the optimized spindle was approximately .00025". Here the deflection improved by a factor of 9. The optimization

Batch File:

Geometry:

Number of Sections(#): 6

Section (#)	Length (in)	Outer Diameter (in)	Inner Diameter (in)	Area (in^2)	Moment of Inertia (in^4)
1	3	2.25	2	0.83449	0.47265783
2	3	2.375	2.125	0.88357	0.560861726
3	3	2.5	2.25	0.93266	0.659419991
4	3	2.625	2.375	0.98175	0.76890787
5	3	2.75	2.5	1.03084	0.889900605
6	3	2.875	2.625	1.07992	1.022973438
7				0	0
8				0	0
9				0	0
10				0	0
11				0	0
12				0	0
13				0	0
14				0	0
15				0	0

Bearings:

Lat. Stiffness of Rear Bearing (lb/in):	100000
Lat. Stiffness of Front Bearing (lb/in):	500000
Tor. Stiffness of Front Bearing (in-lb):	10000
Fraction of mom. on Front Bearing:	0.1
Location of Rear Bearing (in.)	7.5
Location of Front Bearing (in.)	13.5

Pulley:

Location of Pulley (in.)	4.5
Mass of Pulley (lb-s^2/in):	0.007763975
Static Belt Tension (lb):	50
Harmonic Drive Force (lb):	50
Drive Frequency (Hz):	133.33
Pulley Unbalance (lb-s^2):	0.002279727

Tool:

Mass of Tool (lb-s^2/in):	0.025879917
Static Cutting Force (lb):	300
Harmonic Cutting Force (lb):	300
Cutting Frequency (Hz):	133.33
Tool Unbalance (lb-s^2):	0.004559453
Length of Tool (in)	2

Speed:

Spindle Shaft Speed (rpm): 16.66666667

Material Properties:

Modulus of Elasticity (psi):	30000000
Density (lb/in^3):	0.289

Figure 4.2 Batch File for Sample Spindle Problem

improved the deflection of the spindle significantly for both the static and dynamic models.

5.0 Conclusion:

The analysis of a machine tool spindle began by developing a model to solve for the static lateral deflection. A program was developed using *Matlab* that reads in the geometry and load information for a spindle and reports plots of the lateral deflection. The user must simply enter the appropriate information into a spreadsheet, termed a "batch file" and the program will supply a plot of the spindles deformed shape. A sample spindle was analyzed using this program and the FEA program *Ansys*. Both analyses yielded comparable results.

Next the analysis was extended to include the dynamic response of the spindle. Again a program was developed that would read in a batch file containing the appropriate geometry and load information for the spindle. The program would then report plots of the first four mode shapes of the spindle as well as a plot of its dynamic forced response. A sample spindle was also analyzed using the dynamic program and the results compared to an FEA analysis. The FEA analysis agreed very closely with the dynamic analysis program for both the mode shapes and lateral deflection.

Finally a program was developed that would optimize the spindle by minimizing the deflection at the interface between the cutting tool and the spindle shaft. The program performs an optimization for both the static and dynamic analyses. The program optimizes the location and stiffness of the spindle support bearings. A sample spindle was optimized both statically and dynamically using the program. One key result of these analyses was that the optimum parameters for both the static and dynamic analysis were approximately the same. This is very important because the complexity and time required to analyze and optimize the spindle dynamically was significantly more than that of the static analysis and optimization. Therefore if the goal of a spindle designer is to optimize an existing spindle design, it could be done statically with much less effort and time than it could be done dynamically.

Although these programs are very powerful in designing a machine tool spindle, further work could make them of more value to a spindle designer. The first recommendation would be to equip these programs with a graphical user interface. This would make the interface between the designer and the program much more user friendly.

The next recommendation would be to make the programming code more efficient. Currently it takes a considerable amount of time to process the dynamic analysis and optimization. There are several iterations in each of these analyses. The dynamic optimization takes several hours to run.

Another recommendation for future work would be to enhance the dynamic loads applied to the spindle model. Currently harmonic loads are assumed for the cutting force and drive force. The accuracy of the forced response could be further refined by taking measurements of the cutting force and drive force for an existing machine tool spindles. This could be taken one step further by creating a database of the cutting forces for various cutting tools. If a spindle was to primarily use one type of cutting tool the load information could be read into the dynamic program from the database.

One final suggestion for future work would be to work with a bearing manufacturer to create a database of bearings and their stiffness values. This would provide the stiffness for the static and dynamic analyses as well as the maximum allowable stiffness values for the optimization analyses.

REFERENCES

[1] Al-Shareef, K.J.H, Brandon, J.A., *"On the Applicability of Modal and Response Representations in the Dynamic Analysis of Machine Tool Spindle Bearing Systems," Journal of Engineering Manufacture,* Vol. 205, 1991, pp. 139-145.

[2] Al-Shareef, K.J.H, Brandon, J.A., *"On the Quasi-Static Design of Machine Tool Spindles," Journal of Engineering Manufacture,* Vol. 204, 1990, pp. 91-104.

[3] Arora, J, *Introduction to Optimum Design,* McGraw-Hill Inc., Boston, 1989.

[4] Curti,G., Raffa, F.A., Vatta, F., *"Steady-State Unbalance Response of Continuous Rotors on Anisotropic Supports," Vibration of Rotating Systems, Vol.* 60, ASME 1993, pp. 27-34.

[5] Dahleh, M.D., Thomson, W.T., *Theory of Vibration with Applications,* Prentice-Hall, Inc., London, 1998.

[6] Lewinchal, L., *"Machine Tool Spindle Applications," SKF Industries, Inc. Engineering and Research,* SKF Norden, Feb 1983.

[7] Lewinchal, L., *"Machine Tool Spindle Applications," SKF Industries, Inc. Engineering and Research,* SKF Norden, Feb 1985.

[8] Montusiewicz,J., Osycska,A., *"Computer Aided Optimum Design of Machine Tool Spindle Systems with Hydrostatic Bearings" Journal of Engineering Manufacture,* Vol. 211, 1997, pp. 43-51.

[9] Murphy,B., *"Improved Rotordynamics Unbalance Response Calculations Using the Polynomial Method", Vibration of Rotating Systems, Vol.* 60, ASME 1993, pp. 35-42.

Appendix A

Template for Batch File

Batch File:

Geometry:

Number of Sections(#):	6

Section (#)	Length (in)	Outer Diameter (in)	Inner Diameter (in)	Area (in^2)	Moment of Inertia (in^4)
1	3	2.25	2	0.83449	0.47265783
2	3	2.375	2.125	0.88357	0.560861726
3	3	2.5	2.25	0.93266	0.659419991
4	3	2.625	2.375	0.98175	0.76890787
5	3	2.75	2.5	1.03084	0.889900605
6	3	2.875	2.625	1.07992	1.022973438
7				0	0
8				0	0
9				0	0
10				0	0
11				0	0
12				0	0
13				0	0
14				0	0
15				0	0

Bearings:

Lat. Stiffness of Rear Bearing (lb/in):	100000
Lat. Stiffness of Front Bearing (lb/in.):	500000
Tor. Stiffness of Front Bearing (in-lb):	10000
Fraction of mom. on Front Bearing:	0.1
Location of Rear Bearing (in.)	7.5
Location of Front Bearing (in.)	13.5

Pulley:

Location of Pulley (in.)	4.5
Mass of Pulley (lb-s^2/in):	0.007763975
Static Belt Tension (lb):	50
Harmonic Drive Force (lb):	50
Drive Frequency (Hz):	133.33
Pulley Unbalance (lb-s^2):	0.002279727

Tool:

Mass of Tool (lb-s^2/in):	0.025879917
Static Cutting Force (lb):	300
Harmonic Cutting Force (lb):	300
Cutting Frequency (Hz):	133.33
Tool Unbalance (lb-s^2):	0.004559453
Length of Tool (in)	2

Speed:

Spindle Shaft Speed (rpm):	16.66666667

Material Properties:

Modulus of Elasticity (psi):	30000000
Density (lb/in^3):	0.289

Appendix B

Matlab Programs for the Static Analysis

```
% Filename : spindlec
% This file is the parent to all other subroutines that will be employed
% to perform the static and dynamic analysis.
% Revision "C"

clear

% Reads in geometry and loading information from "Batch1.wk1"

constantsc

% Calculates the static deformation.

deformationc

% Calculates the dynamic response.

dynamicc
```

```
% Filename : "Constantsc"
% This subroutine collects the constants required to perform the static and dynamic
analysis.
% The subroutine requires a spreadsheet "batch" to be in the folder "RevC"
% "batch" contains all the inputs required to drive the analysis.

clear;

temp = 0;

ct = wk1read('Batch1');

metool = ct(49,4);

mepulley = ct(41,4);

speed = ct(53,4);

tlunbal = metool*(speed*2*pi)^2;

plunbal = mepulley*(speed*2*pi)^2;

ftharm = ct(47,4);

fpharm = ct(39,4);

freqpulley = ct(40,4);

freqtool = ct(48,4);

E = ct(57,4);

density = ct(58,4);

N = ct(5,4);

Mpulley  = ct(37,4);

Mtool = ct(45,4);

Kr = ct(27,4);

Kf = ct(28,4);
```

```
Ktf = ct(29,4);

x1 = ct(30,4);

tl = ct(50,4);

f1 = ct(38,4);

f2 = ct(46,4);

for n = 1:N;

   b(n) = ct(8+n,2)+temp;

   temp = b(n);

   Area(n) = ct(8+n,5);

   Inertia(n) = ct(8+n,6);

   R_out(n) = ct(8+n,3)/2;

   R_in(n) = ct(8+n,4)/2;

end

Omsft = ct(53,4);

a(1) = ct(36,4);

a(2) = ct(31,4);

a(3) = ct(32,4);

a(4) = b(N);


if a(1) <= b(1)

   XsecA(1) = Area(1);

   XsecI(1) = Inertia(1);

   XsecR(1) = R_out(1);
```

```
end

for n = 1:N
  if b(n)< a(1)
    if b(n+1)>a(1)
      XsecA(1) = Area(n+1);
      XsecI(1) = Inertia(n+1);
      XsecR(1) = R_out(n+1);
    end
  end
  if b(n)< a(2)
    if b(n+1)>a(2)
      XsecA(2) = Area(n+1);
      XsecI(2) = Inertia(n+1);
      XsecR(2) = R_out(n+1);
    end
  end
  if b(n)< a(3)
    if b(n+1)>a(3)
      XsecA(3) = Area(n+1);
      XsecI(3) = Inertia(n+1);
      XsecR(3) = R_out(n+1);
```

```
    end
  end
end
XsecA(4) = Area(N);
XsecI(4) = Inertia(N);
XsecR(4) = R_out(N);
for n = 1:(N-1);
  r(n) = Inertia(n+1)/Inertia(n);
  R(n) = (Area(n+1)/Inertia(n+1))^.25/(Area(n)/Inertia(n))^.25;
end
x(1) = 0;
for n = 1:N
  z = 2*n;
  x(z) = b(n);
  if z < 2*N;
    x(z+1) = b(n);
  end
end

for n = 1:N
  z = 2*n-1;
  Outer(z) = R_out(n);
  Outer(z+1) = R_out(n);
```

```
  Inner(z) = R_in(n);

  Inner(z+1)= R_in(n);

end

plot(x,Outer,x,Inner,a(1),XsecR(1),'s',a(2),XsecR(2),'*',a(3),XsecR(3),'*',a(4),XsecR(4),'p
');

xlabel('x,(in)')

ylabel('shaft radius, (in)')

title('input dimensions')

legend('OD','ID','Pulley','Rear Bearing','Front Bearing','Tool',2)

axis([0,b(N),0,(b(N))/2])
```

```
% Filename : deformationc
% This subroutine will calculate the static lateral deflection of the spindle
% Revision "C"


% Contribution due to bearing deformation

deflbrgc

% Contribution due to shaft deformation

deflshftc


yt = yb + ys;

% Creates Plots of the shaft deformation

plotsc
```

```
% Filename : deflbrgc
% This subroutine will calcuate the deformation of the bearings
% Revision "C"

x = linspace(0,(a(4)),100);

mb = f2 * ((a(4)+tl)-a(3))*x1;

R1 = (f1*(a(3)-a(1))+mb-f2*((a(4)+tl)-a(3)))/(a(3)-a(2));
R2 = f1 + f2 - R1;


delta1 = -R1/Kr;
delta2 = -R2/Kf;

m = (delta2-delta1)/(a(3)-a(2));

for n = 1:100

   yb(n) = m*(x(n)-a(2)) + delta1;

end
```

```
% Filename: deflshftc
% This subroutine will calculate the deformation of the elastic shaft
% in rigid supports
% This portion of the routine will scale the loads:
% Revision "C"

for j = 1:3

  n(j) = 1;

  for i = 2:N

    if a(j) > b(i-1)

      if a(j) <= b(i)

        n(j) = i;

      else

      end

    else

      n(j)=n(j);

    end

  end

end


f1e = Inertia(N)/Inertia(n(1))*f1;

R1e = Inertia(N)/Inertia(n(2))*R1;

mbe = Inertia(N)/Inertia(n(3))*mb;

R2e = Inertia(N)/Inertia(n(3))*R2;


for i = 1:(N-1)
```

```
  voi = R1*sing(b(i),a(2),0) + R2*sing(b(i),a(3),0) - f1*sing(b(i),a(1),0);

  moi = R1 * (b(i)-a(2)) * sing(b(i),a(2),0) + R2*(b(i)-a(3))*sing(b(i),a(3),0)...
    - f1*(b(i)-a(1))*sing(b(i),a(1),0) - mb*sing(b(i),a(3),0);

  v(i) = Inertia(N)*(1/Inertia(i) - 1/Inertia(i+1))*voi;

  m(i) = Inertia(N)*(1/Inertia(i) - 1/Inertia(i+1))*moi;

end

x = linspace(0,a(4),100);

temp1 = 0;

temp2 = 0;

for r = 1:(N-1)

  temp1 = temp1 + v(r)/6*sing(a(3),b(r),3) - v(r)/6*sing(a(2),b(r),3)...
    + m(r)/2*sing(a(3),b(r),2) - m(r)/2*sing(a(2),b(r),2);

  temp2 = temp2 + v(r)/6*sing(a(2),b(r),3) + m(r)/2*sing(a(2),b(r),2);

end


q1 = 1/(a(2)-a(3))*(f1e/6*sing(a(3),a(1),3) - R1e/6*sing(a(3),a(2),3) + temp1 -
f1e/6*sing(a(2),a(1),3));

q2 = -f1e/6*sing(a(2),a(1),3) - temp2 - q1*a(2);


for r = 1:100

  temp3 = 0;

  for s = 1:(N-1)

    temp3 = temp3 + v(s)/6*sing(x(r),b(s),3) + m(s)/2*sing(x(r),b(s),2);

  end

  ys(r) = -1/(E*Inertia(N))*(f1e/6*sing(x(r),a(1),3) - R1e/6*sing(x(r),a(2),3)...
```

```
   -R2e/6*sing(x(r),a(3),3) + temp3 + mbe/2*sing(x(r),a(3),2) + q1*x(r) + q2);
end
```

```
% Filename   plotsc
% This subroutine will create plots of the shaft deformation
% Revision "C"

figure

plot(x,yb)

title('Deflection Contribution of Bearings')

figure

plot(x,ys)

title('Deflection Contribution of Elastic Shaft')

figure

plot(x,yt)

title('Combined Spindle Deflection')
```

```
function sing = f(o,p,q)
% Function : Singularity
% This subroutine defines a new function to be used in subsequent calculations
if o<p
  sing = 0;
else
  sing = (o-p)^q;
end
```

Appendix C

Matlab Programs for the Dynamic Analysis

```
% Filename : "dynamicc"
% Rev C
% This subroutine solves for the dynamic response of the spindle.
% Must read in the batch file prior to executing this subroutine.


% Solves for the eigenvalues.

frequencyc

% Solves for the eigenvectors.

modec

% Solves for the dynamic response.

forcedc
```

```
% Filename : "frequencyc"
% This subroutine is used to plot the frequency equation.
% There needs to be a matrix A that drives this subroutine.


Omega = linspace(1,2000,50);

for n = 1:50;

  w = Omega(n);

  X1 = (density*XsecA(1)*w^2/E/XsecI(1))^.25*a(1);

  X2 = (density*XsecA(2)*w^2/E/XsecI(2))^.25*a(2);

  X3 = (density*XsecA(3)*w^2/E/XsecI(3))^.25*a(3);

  X4 = (density*XsecA(4)*w^2/E/XsecI(4))^.25*a(4);

  Transferc

  C1 = Mpulley*X1/(density*XsecA(1)*a(1));

  C2 = Kr/(X2/a(2))^3/E/XsecI(2);

  C3 = Ktf/(X3/a(3))^2/E/XsecI(3);

  C4 = Kf/(X3/a(3))^3/E/XsecI(3);

  C5 = Mtool*X4/(density*XsecA(4)*a(4));

  temp1(1,:) = [1 0 -1 0];

  temp1(2,:) = [0 1 0 -1];

  Q1 = temp1*T1;

  temp2(1,:) = [-(sinh(X1)-C1*cosh(X1)) -(cosh(X1)-C1*sinh(X1)) -(sin(X1)-
C1*cos(X1)) (cos(X1)+C1*sin(X1))];

  temp2(2,:) = [-cosh(X1) -sinh(X1) -cos(X1) -sin(X1)];

  temp2(3,:) = [-sinh(X1) -cosh(X1) sin(X1) -cos(X1)];
```

```
temp2(4,:) = [-cosh(X1) -sinh(X1) cos(X1) sin(X1)];

Q2 = temp2*T2;

temp3(1,:) = [ sinh(X2) cosh(X2) sin(X2) -cos(X2)];

temp3(2,:) = [-cosh(X2) -sinh(X2) -cos(X2) -sin(X2)];

temp3(3,:) = [-sinh(X2) -cosh(X2) sin(X2) -cos(X2)];

temp3(4,:) = [-cosh(X2) -sinh(X2) cos(X2) sin(X2)];

Q3 = temp3*T3;

temp4(1,:) = [ sinh(X3) cosh(X3) sin(X3) -cos(X3)];

temp4(2,:) = [-cosh(X3) -sinh(X3) -cos(X3) -sin(X3)];

temp4(3,:) = [-sinh(X3) -cosh(X3) sin(X3) -cos(X3)];

temp4(4,:) = [cosh(X3) sinh(X3) -cos(X3) -sin(X3)];

Q4 = temp4*T4;

A(1,:) = [Q1(1,1) Q1(1,2) Q1(1,3) Q1(1,4) 0 0 0 0 0 0 0 0 0 0 0 0];

A(2,:) = [Q1(2,1) Q1(2,2) Q1(2,3) Q1(2,4) 0 0 0 0 0 0 0 0 0 0 0 0];

A(3,:) = [sinh(X1) cosh(X1) sin(X1) -cos(X1) Q2(1,1) Q2(1,2) Q2(1,3) Q2(1,4) 0 0 0 0
0 0 0 0];

A(4,:) = [cosh(X1) sinh(X1) cos(X1) sin(X1) Q2(2,1) Q2(2,2) Q2(2,3) Q2(2,4) 0 0 0 0
0 0 0 0];

A(5,:) = [sinh(X1) cosh(X1) -sin(X1) cos(X1) Q2(3,1) Q2(3,2) Q2(3,3) Q2(3,4) 0 0 0 0
0 0 0 0];

A(6,:) = [cosh(X1) sinh(X1) -cos(X1) -sin(X1) Q2(4,1) Q2(4,2) Q2(4,3) Q2(4,4) 0 0 0
0 0 0 0 0];

A(7,:) = [0 0 0 0 (C2*cosh(X2)-sinh(X2)) (C2*sinh(X2)-cosh(X2)) (C2*cos(X2)-
sin(X2)) (C2*sin(X2)+cos(X2)) Q3(1,1) Q3(1,2) Q3(1,3) Q3(1,4) 0 0 0 0];

A(8,:) = [0 0 0 0 cosh(X2) sinh(X2) cos(X2) sin(X2) Q3(2,1) Q3(2,2) Q3(2,3) Q3(2,4)
```

```
0 0 0 0];

  A(9,:) = [0 0 0 0 sinh(X2) cosh(X2) -sin(X2) cos(X2) Q3(3,1) Q3(3,2) Q3(3,3) Q3(3,4)
0 0 0 0];

  A(10,:) = [0 0 0 0 cosh(X2) sinh(X2) -cos(X2) -sin(X2) Q3(4,1) Q3(4,2) Q3(4,3)
Q3(4,4) 0 0 0 0];

  A(11,:) = [0 0 0 0 0 0 0 0 (C4*cosh(X3)-sinh(X3)) (C4*sinh(X3)-cosh(X3))
(C4*cos(X3)-sin(X3)) (C4*sin(X3)+cos(X3)) Q4(1,1) Q4(1,2) Q4(1,3) Q4(1,4)];

  A(12,:) = [0 0 0 0 0 0 0 0 cosh(X3) sinh(X3) cos(X3) sin(X3) Q4(2,1) Q4(2,2) Q4(2,3)
Q4(2,4)];

  A(13,:) = [0 0 0 0 0 0 0 0 sinh(X3) cosh(X3) -sin(X3) cos(X3) Q4(3,1) Q4(3,2) Q4(3,3)
Q4(3,4)];

  A(14,:) = [0 0 0 0 0 0 0 0 (C3*sinh(X3)-cosh(X3)) (C3*cosh(X3)-sinh(X3)) (-
C3*sin(X3)+cos(X3)) (C3*cos(X3)+sin(X3)) Q4(4,1) Q4(4,2) Q4(4,3) Q4(4,4)];

  A(15,:) = [0 0 0 0 0 0 0 0 0 0 0 0 cosh(X4) sinh(X4) -cos(X4) -sin(X4)];

  A(16,:) = [0 0 0 0 0 0 0 0 0 0 0 0 (sinh(X4)+C5*cosh(X4)) (cosh(X4)+C5*sinh(X4))
(sin(X4)+C5*cos(X4)) (-cos(X4)+C5*sin(X4))];

  for k = 1:15

    A(k,:)=A(k,:)/A(k,k);

    for p = k+1:16

      A(p,:) = A(p,:)-A(k,:)*A(p,k);

    end

  end

  freq(n) = A(16,16);

end

n = 1;

for s = 2:50;
```

```
  sign = freq(s)*freq(s-1);
  if sign < 0
    if abs((freq(s)-freq(s-1))/(Omega(s)-Omega(s-1)))<.25
        bisectc
    end
  end
end
```

```
% Filename : "transferj"
% This subroutine creates the matrices to transform the constants.

T1=eye(4);

T2=eye(4);

T3=eye(4);

T4=eye(4);

for counter = 1:(N-1)

  beta1 = (Area(counter)*density*w^2/(E*Inertia(counter)))^.25;

  beta2 = (Area(counter+1)*density*w^2/(E*Inertia(counter+1)))^.25;

  r1 = r(counter);

  R1 = R(counter);

  l1 = b(counter);

  flag = 0;

  T = zeros(4,4);

  T(1,1) = ((r1*R1^2+1)/2*(cosh(beta1*l1)*cosh(beta2*l1)-
R1*sinh(beta1*l1)*sinh(beta2*l1)));

  T(1,2) = ((r1*R1^2+1)/2*(cosh(beta1*l1)*sinh(beta2*l1) -
R1*sinh(beta1*l1)*cosh(beta2*l1)));

  T(1,3) = -((r1*R1^2-1)/2*(cosh(beta1*l1)*cos(beta2*l1)+
R1*sinh(beta1*l1)*sin(beta2*l1)));

  T(1,4) = -((r1*R1^2-1)/2*(cosh(beta1*l1)*sin(beta2*l1) -
R1*sinh(beta1*l1)*cos(beta2*l1)));

  T(2,1) = ((r1*R1^2+1)/2*(R1*cosh(beta1*l1)*sinh(beta2*l1)-
sinh(beta1*l1)*cosh(beta2*l1)));

  T(2,2) = ((r1*R1^2+1)/2*(R1*cosh(beta1*l1)*cosh(beta2*l1) -
sinh(beta1*l1)*sinh(beta2*l1)));
```

```
T(2,3) = ((r1*R1^2-1)/2*(R1*cosh(beta1*l1)*sin(beta2*l1)+
sinh(beta1*l1)*cos(beta2*l1)));

T(2,4) = ((r1*R1^2-1)/2*(sinh(beta1*l1)*sin(beta2*l1)-
R1*cosh(beta1*l1)*cos(beta2*l1)));

T(3,1) = ((r1*R1^2-1)/2*(R1*sin(beta1*l1)*sinh(beta2*l1) -
cos(beta1*l1)*cosh(beta2*l1)));

T(3,2) = ((r1*R1^2-1)/2*(R1*sin(beta1*l1)*cosh(beta2*l1) -
cos(beta1*l1)*sinh(beta2*l1)));

T(3,3) = ((r1*R1^2+1)/2*(cos(beta1*l1)*cos(beta2*l1)+
R1*sin(beta1*l1)*sin(beta2*l1)));

T(3,4) = ((r1*R1^2+1)/2*(cos(beta1*l1)*sin(beta2*l1) -
R1*sin(beta1*l1)*cos(beta2*l1)));

T(4,1) = -((r1*R1^2-1)/2*(R1*cos(beta1*l1)*sinh(beta2*l1)+
sin(beta1*l1)*cosh(beta2*l1)));

T(4,2) = -((r1*R1^2-1)/2*(R1*cos(beta1*l1)*cosh(beta2*l1)+
sin(beta1*l1)*sinh(beta2*l1)));

T(4,3) = ((r1*R1^2+1)/2*(sin(beta1*l1)*cos(beta2*l1)-
R1*cos(beta1*l1)*sin(beta2*l1)));

T(4,4) = ((r1*R1^2+1)/2*(sin(beta1*l1)*sin(beta2*l1)+
R1*cos(beta1*l1)*cos(beta2*l1)));

if a(3)< b(counter)

  T4 = T4*T;

  flag = 1;

end

if a(2)< b(counter)

  if flag < 1;

    T3 = T3*T;
```

```
      flag = 1;
    end
  end

  if a(1) < b(counter)
    if flag < 1;
      T2 = T2*T;
      flag = 1;
    end
  end
  if flag < 1;
    T1 = T1*T;
    flag=0;
  end
end
```

```
% Filename : "bisectc"
% This subroutine refines the incremental root finding search
% using the bisection method.

Xlow = Omega(s-1);

Xhigh = Omega(s);

freq_ref = freq(s-1);

freq_new = freq(s-1);

while abs(Xhigh-Xlow)>1e-12;

  Xnew = .5*(Xlow+Xhigh);

  w = Xnew;

  X1 = (density*XsecA(1)*w^2/E/XsecI(1))^.25*a(1);

  X2 = (density*XsecA(2)*w^2/E/XsecI(2))^.25*a(2);

  X3 = (density*XsecA(3)*w^2/E/XsecI(3))^.25*a(3);

  X4 = (density*XsecA(4)*w^2/E/XsecI(4))^.25*a(4);

  Transferc

  C1 = Mpulley*X1/(density*XsecA(1)*a(1));

  C2 = Kr/(X2/a(2))^3/E/XsecI(2);

  C3 = Ktf/(X3/a(3))^2/E/XsecI(3);

  C4 = Kf/(X3/a(3))^3/E/XsecI(3);

  C5 = Mtool*X4/(density*XsecA(4)*a(4));

  temp1(1,:) = [1 0 -1 0];

  temp1(2,:) = [0 1 0 -1];

  Q1 = temp1*T1;
```

```
temp2(1,:) = [-(sinh(X1)-C1*cosh(X1)) -(cosh(X1)-C1*sinh(X1)) -(sin(X1)-
C1*cos(X1)) (cos(X1)+C1*sin(X1))];

temp2(2,:) = [-cosh(X1) -sinh(X1) -cos(X1) -sin(X1)];

temp2(3,:) = [-sinh(X1) -cosh(X1) sin(X1) -cos(X1)];

temp2(4,:) = [-cosh(X1) -sinh(X1) cos(X1) sin(X1)];

Q2 = temp2*T2;

temp3(1,:) = [ sinh(X2) cosh(X2) sin(X2) -cos(X2)];

temp3(2,:) = [-cosh(X2) -sinh(X2) -cos(X2) -sin(X2)];

temp3(3,:) = [-sinh(X2) -cosh(X2) sin(X2) -cos(X2)];

temp3(4,:) = [-cosh(X2) -sinh(X2) cos(X2) sin(X2)];

Q3 = temp3*T3;

temp4(1,:) = [ sinh(X3) cosh(X3) sin(X3) -cos(X3)];

temp4(2,:) = [-cosh(X3) -sinh(X3) -cos(X3) -sin(X3)];

temp4(3,:) = [-sinh(X3) -cosh(X3) sin(X3) -cos(X3)];

temp4(4,:) = [cosh(X3) sinh(X3) -cos(X3) -sin(X3)];

Q4 = temp4*T4;

A(1,:) = [Q1(1,1) Q1(1,2) Q1(1,3) Q1(1,4) 0 0 0 0 0 0 0 0 0 0 0 0];

A(2,:) = [Q1(2,1) Q1(2,2) Q1(2,3) Q1(2,4) 0 0 0 0 0 0 0 0 0 0 0 0];

A(3,:) = [sinh(X1) cosh(X1) sin(X1) -cos(X1) Q2(1,1) Q2(1,2) Q2(1,3) Q2(1,4) 0 0 0 0
0 0 0 0];

A(4,:) = [cosh(X1) sinh(X1) cos(X1) sin(X1) Q2(2,1) Q2(2,2) Q2(2,3) Q2(2,4) 0 0 0 0
0 0 0 0];

A(5,:) = [sinh(X1) cosh(X1) -sin(X1) cos(X1) Q2(3,1) Q2(3,2) Q2(3,3) Q2(3,4) 0 0 0 0
0 0 0 0];
```

```
A(6,:) = [cosh(X1) sinh(X1) -cos(X1) -sin(X1) Q2(4,1) Q2(4,2) Q2(4,3) Q2(4,4) 0 0 0
0 0 0 0 0];

A(7,:) = [0 0 0 0 (C2*cosh(X2)-sinh(X2)) (C2*sinh(X2)-cosh(X2)) (C2*cos(X2)-
sin(X2)) (C2*sin(X2)+cos(X2)) Q3(1,1) Q3(1,2) Q3(1,3) Q3(1,4) 0 0 0 0];

A(8,:) = [0 0 0 0 cosh(X2) sinh(X2) cos(X2) sin(X2) Q3(2,1) Q3(2,2) Q3(2,3) Q3(2,4)
0 0 0 0];

A(9,:) = [0 0 0 0 sinh(X2) cosh(X2) -sin(X2) cos(X2) Q3(3,1) Q3(3,2) Q3(3,3) Q3(3,4)
0 0 0 0];

A(10,:) = [0 0 0 0 cosh(X2) sinh(X2) -cos(X2) -sin(X2) Q3(4,1) Q3(4,2) Q3(4,3)
Q3(4,4) 0 0 0 0];

A(11,:) = [0 0 0 0 0 0 0 0 (C4*cosh(X3)-sinh(X3)) (C4*sinh(X3)-cosh(X3))
(C4*cos(X3)-sin(X3)) (C4*sin(X3)+cos(X3)) Q4(1,1) Q4(1,2) Q4(1,3) Q4(1,4)];

A(12,:) = [0 0 0 0 0 0 0 0 cosh(X3) sinh(X3) cos(X3) sin(X3) Q4(2,1) Q4(2,2) Q4(2,3)
Q4(2,4)];

A(13,:) = [0 0 0 0 0 0 0 0 sinh(X3) cosh(X3) -sin(X3) cos(X3) Q4(3,1) Q4(3,2) Q4(3,3)
Q4(3,4)];

A(14,:) = [0 0 0 0 0 0 0 0 (C3*sinh(X3)-cosh(X3)) (C3*cosh(X3)-sinh(X3)) (-
C3*sin(X3)+cos(X3)) (C3*cos(X3)+sin(X3)) Q4(4,1) Q4(4,2) Q4(4,3) Q4(4,4)];

A(15,:) = [0 0 0 0 0 0 0 0 0 0 0 0 cosh(X4) sinh(X4) -cos(X4) -sin(X4)];

A(16,:) = [0 0 0 0 0 0 0 0 0 0 0 0 (sinh(X4)+C5*cosh(X4)) (cosh(X4)+C5*sinh(X4))
(sin(X4)+C5*cos(X4)) (-cos(X4)+C5*sin(X4))];

for k = 1:15

  A(k,:)=A(k,:)/A(k,k);

  for p = k+1:16

    A(p,:) = A(p,:)-A(k,:)*A(p,k);

  end

end
```

```
freq_new = A(16,16);
   if (freq_new*freq_ref)< 0
      Xhigh = Xnew;
   else
      Xlow = Xnew;
      freq_ref = freq_new;
   end
end
Root(n) = Xnew; % This will be in rad/s
Frequency(n) = Xnew/2/pi;
n = n+1;
```

```
% Filename : "modec"
% This subroutine calculates the first (4) mode shapes.

global c

for m = 1:4

  w = Root(m);

  X1 = (density*XsecA(1)*w^2/E/XsecI(1))^.25*a(1);

  X2 = (density*XsecA(2)*w^2/E/XsecI(2))^.25*a(2);

  X3 = (density*XsecA(3)*w^2/E/XsecI(3))^.25*a(3);

  X4 = (density*XsecA(4)*w^2/E/XsecI(4))^.25*a(4);

  Transferc

  C1 = Mpulley*X1/(density*XsecA(1)*a(1));

  C2 = Kr/(X2/a(2))^3/E/XsecI(2);

  C3 = Ktf/(X3/a(3))^2/E/XsecI(3);

  C4 = Kf/(X3/a(3))^3/E/XsecI(3);

  C5 = Mtool*X4/(density*XsecA(4)*a(4));

  temp1(1,:) = [1 0 -1 0];

  temp1(2,:) = [0 1 0 -1];

  Q1 = temp1*T1;

  temp2(1,:) = [-(sinh(X1)-C1*cosh(X1)) -(cosh(X1)-C1*sinh(X1)) -(sin(X1)-
C1*cos(X1)) (cos(X1)+C1*sin(X1))];

  temp2(2,:) = [-cosh(X1) -sinh(X1) -cos(X1) -sin(X1)];

  temp2(3,:) = [-sinh(X1) -cosh(X1) sin(X1) -cos(X1)];

  temp2(4,:) = [-cosh(X1) -sinh(X1) cos(X1) sin(X1)];
```

```
Q2 = temp2*T2;

temp3(1,:) = [ sinh(X2) cosh(X2) sin(X2) -cos(X2)];

temp3(2,:) = [-cosh(X2) -sinh(X2) -cos(X2) -sin(X2)];

temp3(3,:) = [-sinh(X2) -cosh(X2) sin(X2) -cos(X2)];

temp3(4,:) = [-cosh(X2) -sinh(X2) cos(X2) sin(X2)];

Q3 = temp3*T3;

temp4(1,:) = [ sinh(X3) cosh(X3) sin(X3) -cos(X3)];

temp4(2,:) = [-cosh(X3) -sinh(X3) -cos(X3) -sin(X3)];

temp4(3,:) = [-sinh(X3) -cosh(X3) sin(X3) -cos(X3)];

temp4(4,:) = [cosh(X3) sinh(X3) -cos(X3) -sin(X3)];

Q4 = temp4*T4;

A(1,:) = [Q1(1,1) Q1(1,2) Q1(1,3) Q1(1,4) 0 0 0 0 0 0 0 0 0 0 0 0];

A(2,:) = [Q1(2,1) Q1(2,2) Q1(2,3) Q1(2,4) 0 0 0 0 0 0 0 0 0 0 0 0];

A(3,:) = [sinh(X1) cosh(X1) sin(X1) -cos(X1) Q2(1,1) Q2(1,2) Q2(1,3) Q2(1,4) 0 0 0 0
0 0 0 0];

A(4,:) = [cosh(X1) sinh(X1) cos(X1) sin(X1) Q2(2,1) Q2(2,2) Q2(2,3) Q2(2,4) 0 0 0 0
0 0 0 0];

A(5,:) = [sinh(X1) cosh(X1) -sin(X1) cos(X1) Q2(3,1) Q2(3,2) Q2(3,3) Q2(3,4) 0 0 0 0
0 0 0 0];

A(6,:) = [cosh(X1) sinh(X1) -cos(X1) -sin(X1) Q2(4,1) Q2(4,2) Q2(4,3) Q2(4,4) 0 0 0
0 0 0 0 0];

A(7,:) = [0 0 0 0 (C2*cosh(X2)-sinh(X2)) (C2*sinh(X2)-cosh(X2)) (C2*cos(X2)-
sin(X2)) (C2*sin(X2)+cos(X2)) Q3(1,1) Q3(1,2) Q3(1,3) Q3(1,4) 0 0 0 0];

A(8,:) = [0 0 0 0 cosh(X2) sinh(X2) cos(X2) sin(X2) Q3(2,1) Q3(2,2) Q3(2,3) Q3(2,4)
0 0 0 0];
```

```
A(9,:) = [0 0 0 0 sinh(X2) cosh(X2) -sin(X2) cos(X2) Q3(3,1) Q3(3,2) Q3(3,3) Q3(3,4)
0 0 0 0];

A(10,:) = [0 0 0 0 cosh(X2) sinh(X2) -cos(X2) -sin(X2) Q3(4,1) Q3(4,2) Q3(4,3)
Q3(4,4) 0 0 0 0];

A(11,:) = [0 0 0 0 0 0 0 0 (C4*cosh(X3)-sinh(X3)) (C4*sinh(X3)-cosh(X3))
(C4*cos(X3)-sin(X3)) (C4*sin(X3)+cos(X3)) Q4(1,1) Q4(1,2) Q4(1,3) Q4(1,4)];

A(12,:) = [0 0 0 0 0 0 0 0 cosh(X3) sinh(X3) cos(X3) sin(X3) Q4(2,1) Q4(2,2) Q4(2,3)
Q4(2,4)];

A(13,:) = [0 0 0 0 0 0 0 0 sinh(X3) cosh(X3) -sin(X3) cos(X3) Q4(3,1) Q4(3,2) Q4(3,3)
Q4(3,4)];

A(14,:) = [0 0 0 0 0 0 0 0 (C3*sinh(X3)-cosh(X3)) (C3*cosh(X3)-sinh(X3)) (-
C3*sin(X3)+cos(X3)) (C3*cos(X3)+sin(X3)) Q4(4,1) Q4(4,2) Q4(4,3) Q4(4,4)];

A(15,:) = [0 0 0 0 0 0 0 0 0 0 0 0 cosh(X4) sinh(X4) -cos(X4) -sin(X4)];

A(16,:) = [0 0 0 0 0 0 0 0 0 0 0 0 (sinh(X4)+C5*cosh(X4)) (cosh(X4)+C5*sinh(X4))
(sin(X4)+C5*cos(X4)) (-cos(X4)+C5*sin(X4))];

for k = 1:15

  A(k,:)=A(k,:)/A(k,k);

  for p = k+1:16

    A(p,:) = A(p,:)-A(k,:)*A(p,k);

  end

end

c(16,m) = 1;

for k = 1:15;

  g = 16-k;

  temp = 0;

  for p = g+1:16;
```

```
    temp = temp + A(g,p)*c(p,m);
  end

  c(g,m) = -temp;

end

x1 = linspace(0,a(4),31);

for n = 1:31

  x = x1(n);

  delta = 12;

  if x1(n)<a(3)

    delta = 8;

    if x1(n)<a(2)

      delta = 4;

      if x1(n)<a(1);

      delta = 0;

    end

  end

end

z = 1;

for i = 1:N

  if x >= b(i);

    z = i;

  end
```

```
end

ydyn(n,m) =
c((delta+1),m)*cosh(x*(density*Area(z)*w^2/E/Inertia(z))^.25)+c((delta+2),m)*sinh(x*(
density*Area(z)*w^2/E/Inertia(z))^.25)+c((delta+3),m)*cos(x*(density*Area(z)*w^2/E/I
nertia(z))^.25)+c((delta+4),m)*sin(x*(density*Area(z)*w^2/E/Inertia(z))^.25);

end


figure

plot(x1,ydyn(:,m),'*')

L = sprintf('mode %0.5g',m);

title(L)

end
```

```
% Filename : "modec"
% This subroutine calculates the first (4) mode shapes.

global c

for m = 1:4

  w = Root(m);

  X1 = (density*XsecA(1)*w^2/E/XsecI(1))^.25*a(1);

  X2 = (density*XsecA(2)*w^2/E/XsecI(2))^.25*a(2);

  X3 = (density*XsecA(3)*w^2/E/XsecI(3))^.25*a(3);

  X4 = (density*XsecA(4)*w^2/E/XsecI(4))^.25*a(4);

  Transferc

  C1 = Mpulley*X1/(density*XsecA(1)*a(1));

  C2 = Kr/(X2/a(2))^3/E/XsecI(2);

  C3 = Ktf/(X3/a(3))^2/E/XsecI(3);

  C4 = Kf/(X3/a(3))^3/E/XsecI(3);

  C5 = Mtool*X4/(density*XsecA(4)*a(4));

  temp1(1,:) = [1 0 -1 0];

  temp1(2,:) = [0 1 0 -1];

  Q1 = temp1*T1;

  temp2(1,:) = [-(sinh(X1)-C1*cosh(X1)) -(cosh(X1)-C1*sinh(X1)) -(sin(X1)-
C1*cos(X1)) (cos(X1)+C1*sin(X1))];

  temp2(2,:) = [-cosh(X1) -sinh(X1) -cos(X1) -sin(X1)];

  temp2(3,:) = [-sinh(X1) -cosh(X1) sin(X1) -cos(X1)];

  temp2(4,:) = [-cosh(X1) -sinh(X1) cos(X1) sin(X1)];
```

```
Q2 = temp2*T2;

temp3(1,:) = [ sinh(X2) cosh(X2) sin(X2) -cos(X2)];

temp3(2,:) = [-cosh(X2) -sinh(X2) -cos(X2) -sin(X2)];

temp3(3,:) = [-sinh(X2) -cosh(X2) sin(X2) -cos(X2)];

temp3(4,:) = [-cosh(X2) -sinh(X2) cos(X2) sin(X2)];

Q3 = temp3*T3;

temp4(1,:) = [ sinh(X3) cosh(X3) sin(X3) -cos(X3)];

temp4(2,:) = [-cosh(X3) -sinh(X3) -cos(X3) -sin(X3)];

temp4(3,:) = [-sinh(X3) -cosh(X3) sin(X3) -cos(X3)];

temp4(4,:) = [cosh(X3) sinh(X3) -cos(X3) -sin(X3)];

Q4 = temp4*T4;

A(1,:) = [Q1(1,1) Q1(1,2) Q1(1,3) Q1(1,4) 0 0 0 0 0 0 0 0 0 0 0 0];

A(2,:) = [Q1(2,1) Q1(2,2) Q1(2,3) Q1(2,4) 0 0 0 0 0 0 0 0 0 0 0 0];

A(3,:) = [sinh(X1) cosh(X1) sin(X1) -cos(X1) Q2(1,1) Q2(1,2) Q2(1,3) Q2(1,4) 0 0 0 0
0 0 0 0];

A(4,:) = [cosh(X1) sinh(X1) cos(X1) sin(X1) Q2(2,1) Q2(2,2) Q2(2,3) Q2(2,4) 0 0 0 0
0 0 0 0];

A(5,:) = [sinh(X1) cosh(X1) -sin(X1) cos(X1) Q2(3,1) Q2(3,2) Q2(3,3) Q2(3,4) 0 0 0 0
0 0 0 0];

A(6,:) = [cosh(X1) sinh(X1) -cos(X1) -sin(X1) Q2(4,1) Q2(4,2) Q2(4,3) Q2(4,4) 0 0 0
0 0 0 0 0];

A(7,:) = [0 0 0 0 (C2*cosh(X2)-sinh(X2)) (C2*sinh(X2)-cosh(X2)) (C2*cos(X2)-
sin(X2)) (C2*sin(X2)+cos(X2)) Q3(1,1) Q3(1,2) Q3(1,3) Q3(1,4) 0 0 0 0];

A(8,:) = [0 0 0 0 cosh(X2) sinh(X2) cos(X2) sin(X2) Q3(2,1) Q3(2,2) Q3(2,3) Q3(2,4)
0 0 0 0];
```

```
A(9,:) = [0 0 0 0 sinh(X2) cosh(X2) -sin(X2) cos(X2) Q3(3,1) Q3(3,2) Q3(3,3) Q3(3,4)
0 0 0 0];

A(10,:) = [0 0 0 0 cosh(X2) sinh(X2) -cos(X2) -sin(X2) Q3(4,1) Q3(4,2) Q3(4,3)
Q3(4,4) 0 0 0 0];

A(11,:) = [0 0 0 0 0 0 0 0 (C4*cosh(X3)-sinh(X3)) (C4*sinh(X3)-cosh(X3))
(C4*cos(X3)-sin(X3)) (C4*sin(X3)+cos(X3)) Q4(1,1) Q4(1,2) Q4(1,3) Q4(1,4)];

A(12,:) = [0 0 0 0 0 0 0 0 cosh(X3) sinh(X3) cos(X3) sin(X3) Q4(2,1) Q4(2,2) Q4(2,3)
Q4(2,4)];

A(13,:) = [0 0 0 0 0 0 0 0 sinh(X3) cosh(X3) -sin(X3) cos(X3) Q4(3,1) Q4(3,2) Q4(3,3)
Q4(3,4)];

A(14,:) = [0 0 0 0 0 0 0 0 (C3*sinh(X3)-cosh(X3)) (C3*cosh(X3)-sinh(X3)) (-
C3*sin(X3)+cos(X3)) (C3*cos(X3)+sin(X3)) Q4(4,1) Q4(4,2) Q4(4,3) Q4(4,4)];

A(15,:) = [0 0 0 0 0 0 0 0 0 0 0 0 cosh(X4) sinh(X4) -cos(X4) -sin(X4)];

A(16,:) = [0 0 0 0 0 0 0 0 0 0 0 0 (sinh(X4)+C5*cosh(X4)) (cosh(X4)+C5*sinh(X4))
(sin(X4)+C5*cos(X4)) (-cos(X4)+C5*sin(X4))];

for k = 1:15

  A(k,:)=A(k,:)/A(k,k);

  for p = k+1:16

    A(p,:) = A(p,:)-A(k,:)*A(p,k);

  end

end

c(16,m) = 1;

for k = 1:15;

  g = 16-k;

  temp = 0;

  for p = g+1:16;
```

```
    temp = temp + A(g,p)*c(p,m);
  end

  c(g,m) = -temp;

end

x1 = linspace(0,a(4),31);

for n = 1:31

  x = x1(n);

  delta = 12;

  if x1(n)<a(3)

    delta = 8;

    if x1(n)<a(2)

      delta = 4;

      if x1(n)<a(1);

      delta = 0;

    end

  end

end

z = 1;

for i = 1:N

  if x >= b(i);

    z = i;

  end
```

```
end

ydyn(n,m) =
c((delta+1),m)*cosh(x*(density*Area(z)*w^2/E/Inertia(z))^.25)+c((delta+2),m)*sinh(x*(density*Area(z)*w^2/E/Inertia(z))^.25)+c((delta+3),m)*cos(x*(density*Area(z)*w^2/E/Inertia(z))^.25)+c((delta+4),m)*sin(x*(density*Area(z)*w^2/E/Inertia(z))^.25);

end

figure

plot(x1,ydyn(:,m),'*')

L = sprintf('mode %0.5g',m);

title(L)

end
```

```
function integ = f(XH,XL,md,sec,beta)

% Function : integral of phi^2 from Xlow to Xhigh

% This subroutine defines a new function to be used in subsequent calculations

global c


integ = 1/beta*(c(sec,md)^2/2*(cosh(beta*XH)*sinh(beta*XH)+beta*XH)...
  + c(sec+1,md)^2/2*(cosh(beta*XH)*sinh(beta*XH)-beta*XH)...
  + c(sec+2,md)^2/2*(cos(beta*XH)*sin(beta*XH)+beta*XH)...
  + c(sec+3,md)^2/2*(-cos(beta*XH)*sin(beta*XH)+beta*XH)...
  + c(sec,md)*c(sec+1,md)*(sinh(beta*XH))^2 +
c(sec,md)*c(sec+2,md)*(cosh(beta*XH)*sin(beta*XH)+sinh(beta*XH)*cos(beta*XH))...
  + c(sec,md)*c(sec+3,md)*(sinh(beta*XH)*sin(beta*XH)-
cosh(beta*XH)*cos(beta*XH))...
  +
c(sec+1,md)*c(sec+2,md)*(sinh(beta*XH)*sin(beta*XH)+cosh(beta*XH)*cos(beta*XH)
)...
  + c(sec+1,md)*c(sec+3,md)*(cosh(beta*XH)*sin(beta*XH)-
sinh(beta*XH)*cos(beta*XH))...
  + c(sec+2,md)*c(sec+3,md)*(sin(beta*XH))^2)...
  -1/beta*(c(sec,md)^2/2*(cosh(beta*XL)*sinh(beta*XL)+beta*XL)...
  + c(sec+1,md)^2/2*(cosh(beta*XL)*sinh(beta*XL)-beta*XL)...
  + c(sec+2,md)^2/2*(cos(beta*XL)*sin(beta*XL)+beta*XL)...
  + c(sec+3,md)^2/2*(-cos(beta*XL)*sin(beta*XL)+beta*XL)...
  + c(sec,md)*c(sec+1,md)*(sinh(beta*XL))^2 +
c(sec,md)*c(sec+2,md)*(cosh(beta*XL)*sin(beta*XL)+sinh(beta*XL)*cos(beta*XL))...
  + c(sec,md)*c(sec+3,md)*(sinh(beta*XL)*sin(beta*XL)-
cosh(beta*XL)*cos(beta*XL))...
  +
c(sec+1,md)*c(sec+2,md)*(sinh(beta*XL)*sin(beta*XL)+cosh(beta*XL)*cos(beta*XL))
...
  + c(sec+1,md)*c(sec+3,md)*(cosh(beta*XL)*sin(beta*XL)-
sinh(beta*XL)*cos(beta*XL))...
  + c(sec+2,md)*c(sec+3,md)*(sin(beta*XL))^2);
```

Appendix D

Matlab Programs for the Optimization

```
% filename : "optimize"
% this file launches the optimization subroutines

constantso

stop = 0;

for nn = 1:1000;

  if stop <= 0;

    % Step 1;

    kk=1;

    R(1) = 1;

    gamma = .5;

    eps1 = .01;

    eps2 = .01;

    Hess = eye(4);

    Kr_max = 1e6;

    Kf_max = 1e6;

    DELTA = .5;

    OH = 2;

    xn(1) = a(2)/(a(1)+DELTA);

    xn(2) = a(3)/(a(4)-OH);

    xn(3) = Kf/Kf_max;

    xn(4) = Kr/Kr_max;

    % Step 2
```

```
deformationo
yt(100)
constraint
temp = max(G);
if temp < 0
  V = 0;
else
  V = temp;
end
gradient
dytn(1) = dyt(1)*(a(1)+DELTA)
dytn(2) = dyt(2)*(a(4)-OH)
dytn(3) = dyt(3)*Kf_max
dytn(4) = dyt(4)*Kr_max
% Step 3
multiplier
% Step 4

if abs_dd < eps1
  stop = 1;
  if V > eps2
    stop = 0;
```

```
      end
    end
    % Step 5
    rr = uu(1) + uu(2) + uu(3) +  uu(4);
    R((kk+1)) = max(rr,R(kk));
    % Step 6
    step_size
    % Step 7
    kk = kk+1;
  end
end
```

```
% filename · "gradient"
% this subroutine calculates the gradient of yt with respect to the design variables
% the design variables are Kf, Kr, a(1), a(2), a(3)


deformationo

yt_k = yt(100);

% the gradient of yt w.r.t. a(2):

Delta1 = a(2)*.001;

a(2) = a(2)+Delta1;

deformationo

yt_kp = yt(100);

dyt(1) = (yt_kp-yt_k)/Delta1;

% the gradient of yt w.r.t. a(3):

Delta2 = a(3)*.01;

a(3) = a(3)+Delta2;

deformationo

yt_kp = yt(100);

dyt(2) = (yt_kp-yt_k)/Delta2;

% the gradient of yt w.r.t. Kf:

Delta3 = Kf*.01;

Kf = Kf+Delta3;

deformationo

yt_kp = yt(100);

dyt(3) = (yt_kp-yt_k)/Delta3;
```

```
% the gradient of yt w.r.t. Kr:

Delta4 = Kr*.01;

Kr = Kr+Delta4;

deformationo

yt_kp = yt(100);

dyt(4) = (yt_kp-yt_k)/Delta4;
```

```
% Filename : "constraint"
% This subroutine calculates the constraint violations and their derivatives.

G = [(-xn(1)+1) (xn(2)-1) (xn(3)-1) (xn(4)-1)];

dG1 = [-1 0 0 0];

dG2 = [0 1 0 0];

dG3 = [0 0 1 0];

dG4 = [0 0 0 1];
```

```
% Filename . "multiplier"
% This file calculates the Lagrange Multipliers

clear AA CC

if xn(1) >= 1

  uu(1) = 0;

  ss(1) = sqrt(xn(1)-1);

  sw1 = 0;

else

  ss(1) = 0;

  sw1 = 1;

end

if xn(2) <= 1

  uu(2) = 0;

  ss(2) = sqrt(xn(2)-1);

  sw2 = 0;

else

  ss(2) = 0;

  sw2 = 1;

end

if xn(3) <= 1

  uu(3) = 0;

  ss(3) = sqrt(xn(3)-1);

  sw3 = 0;
```

```
else
  ss(3) = 0;
  sw3 = 1;
end
if xn(4) <= 1
  uu(4) = 0;
  sw4 = 0;
  ss(4) = sqrt(xn(4)-1);
else
  ss(4) = 0;
  sw4 = 1;
end
binary = sw1*2^3+sw2*2^2+sw3*2^1+sw4*2^0
if binary == 0
  mult0
end
if binary == 1
  mult1
end
if binary == 2
  mult2
end
```

```
if binary == 2
  mult2
end
if binary == 3
  mult3
end
if binary == 4
  mult4
end
if binary == 5
  mult5
end
if binary == 6
  mult6
end
if binary == 7
  mult7
end
if binary == 8
  mult8
end
if binary == 9
```

```
  mult9
end
if binary == 10
  mult10
end
if binary == 11
  mult11
end
if binary == 12
  mult12
end
if binary == 13
  mult13
end
if binary == 14
  mult14
end
if binary == 15
  mult15
end

abs_dd = sqrt(dd(1)^2+dd(2)^2+dd(3)^2+dd(4)^2)
%if sw1 == 0
```

```
% ss(1) = sqrt(dd(1)-dd(2)-DELTA/a(4));
%end
%if sw2 == 0
% ss(2) = sqrt(dd(3)-1);

%end
%if sw3 == 0
% ss(3) = sqrt(dd(4)-1);
%end
%if sw4 == 0
% ss(4) = sqrt(dd(5)-1);
%end
```

```
% Filename . "multiplier"
% This file calculates the Lagrange Multipliers

clear AA CC

if xn(1) >= 1

  uu(1) = 0;

  ss(1) = sqrt(xn(1)-1);

  sw1 = 0;

else

  ss(1) = 0;

  sw1 = 1;

end

if xn(2) <= 1

  uu(2) = 0;

  ss(2) = sqrt(xn(2)-1);

  sw2 = 0;

else

  ss(2) = 0;

  sw2 = 1;

end

if xn(3) <= 1

  uu(3) = 0;

  ss(3) = sqrt(xn(3)-1);

  sw3 = 0;
```

```
else
  ss(3) = 0;
  sw3 = 1;
end
if xn(4) <= 1
  uu(4) = 0;
  sw4 = 0;
  ss(4) = sqrt(xn(4)-1);
else
  ss(4) = 0;
  sw4 = 1;
end
binary = sw1*2^3+sw2*2^2+sw3*2^1+sw4*2^0
if binary == 0
  mult0
end
if binary == 1
  mult1
end
if binary == 2
  mult2
end
```

```
if binary == 2
  mult2
end
if binary == 3
  mult3
end
if binary == 4
  mult4
end
if binary == 5
  mult5
end
if binary == 6
  mult6
end
if binary == 7
  mult7
end
if binary == 8
  mult8
end
if binary == 9
```

```
  mult9
end
if binary == 10
  mult10
end
if binary == 11
  mult11
end
if binary == 12
  mult12
end
if binary == 13
  mult13
end
if binary == 14
  mult14
end
if binary == 15
  mult15
end

abs_dd = sqrt(dd(1)^2+dd(2)^2+dd(3)^2+dd(4)^2)
%if sw1 == 0
```

```
 %  ss(1) = sqrt(dd(1)-dd(2)-DELTA/a(4));
%end
%if sw2 == 0
 %  ss(2) = sqrt(dd(3)-1);

%end
%if sw3 == 0
 %  ss(3) = sqrt(dd(4)-1);
%end
%if sw4 == 0
 %  ss(4) = sqrt(dd(5)-1);
%end
```

```
% Filname : "mult0"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1))];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2))];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3))];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4))];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

dd = inv(AA)*CC
```

```
% Filname : "mult0"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1))];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2))];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3))];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4))];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

dd = inv(AA)*CC
```

```
% Filname : "mult2"

dd(4) = -G(3);

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,4)+Hess(4,1)) 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,4)+Hess(4,2)) 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,4)+Hess(4,3)) 1];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) Hess(4,4) 0];

CC(1,:) = [-dytn(1)-(Hess(1,3)+Hess(3,1))*dd(3)];

CC(2,:) = [-dytn(2)-(Hess(2,3)+Hess(3,2))*dd(3)];

CC(3,:) = [-dytn(3)-Hess(3,3)*dd(3)];

CC(4,:) = [-dytn(4)-(Hess(3,4)+Hess(4,3))*dd(3)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(4,1) = temp_mult(3,1);

u(3) = temp_mult(4,1);
```

```
% Filname : "mult3"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) 0 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 0 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 1 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 1];

AA(5,:) = [0 0 1 0 0 0];

AA(6,:) = [0 0 0 1 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(3)];

CC(6,:) = [-G(4)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);

u(3) = temp_mult(5,1);

u(4) = temp_mult(6,1);
```

```
% Filname : "mult4"

dd(2,1) = -G(2);

AA(1,:) = [(Hess(1,1)) (Hess(1,3)+Hess(3,1)) (Hess(1,4)+Hess(4,1)) 0];

AA(2,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,4)+Hess(4,3)) 1];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,3)) (Hess(4,3)+Hess(3,4)) 0];

AA(4,:) = [(Hess(4,1)+Hess(1,4)) (Hess(4,3)+Hess(3,4)) (Hess(4,4)) 0];

CC(1,:) = [-dytn(1)-(Hess(1,2)+Hess(2,1))*dd(2)];

CC(2,:) = [-dytn(2)-(Hess(2,2))*dd(2)];

CC(3,:) = [-dytn(3)-(Hess(3,2)+Hess(2,3))*dd(2)];

CC(4,:) = [-dytn(4)-(Hess(4,2)+Hess(2,4))*dd(2)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(3,1) = temp_mult(2,1);

dd(4,1) = temp_mult(3,1);

u(2) = temp_mult(4,1);
```

```
% Filname : "mult5"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) 0 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 1 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 1];

AA(5,:) = [0 1 0 0 0 0];

AA(6,:) = [0 0 0 1 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(2)];

CC(6,:) = [-G(4)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);

u(2) = temp_mult(5,1);

u(4) = temp_mult(6,1);
```

```
% Filname : "mult6"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) 0 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 1 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0 1];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 0];

AA(5,:) = [0 1 0 0 0 0];

AA(6,:) = [0 0 1 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(2)];

CC(6,:) = [-G(3)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);

u(2) = temp_mult(5,1);

u(3) = temp_mult(6,1);
```

```
% Filname · "mult7"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) 0 0 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 1 0 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0 1 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 0 1];

AA(5,:) = [0 1 0 0 0 0 0];

AA(6,:) = [0 0 1 0 0 0 0];

AA(7,:) = [0 0 0 1 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(2)];

CC(6,:) = [-G(3)];

CC(7,:) = [-G(4)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);
```

```
u(2) = temp_mult(6,1);
u(3) = temp_mult(7,1);
u(4) = temp_mult(8,1);
```

```
% Filname : "mult8"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) 1];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0];

AA(5,:) = [1 0 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(6,:) = [-G(1)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);

u(1) = temp_mult(5,1);
```

```
% Filname : "mult9"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) -1 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 0 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 1];

AA(6,:) = [-1 0 0 0 0 0];

AA(7,:) = [0 0 0 1 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(1)];

CC(6,:) = [-G(4)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);

u(1) = temp_mult(5,1);

u(4) = temp_mult(6,1);
```

```
% Filname : "mult10"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) -1 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 0 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0 1];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 0];

AA(5,:) = [-1 1 0 0 0 0 0];

AA(6,:) = [0 0 0 1 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(1)];

CC(6,:) = [-G(3)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);

u(1) = temp_mult(5,1);

u(3) = temp_mult(6,1);
```

```
% Filname : "mult11"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) -1 0 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 0 0 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0 1 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 0 1];

AA(5,:) = [-1 0 0 0 0 0 0];

AA(6,:) = [0 0 1 0 0 0 0];

AA(7,:) = [0 0 0 1 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(1)];

CC(6,:) = [-G(3)];

CC(7,:) = [-G(4)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);
```

```
u(1) = temp_mult(5,1);
u(3) = temp_mult(6,1);
u(4) = temp_mult(7,1);
```

```
% Filname : "mult12"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) -1 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 1 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 0];

AA(5,:) = [-1 0 0 0 0 0];

AA(6,:) = [0 1 0 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(1)];

CC(6,:) = [-G(2)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);

u(1) = temp_mult(5,1);

u(2) = temp_mult(6,1);
```

```
% Filname : "mult13"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) (Hess(1,5)+Hess(5,1)) -1 0 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) (Hess(2,5)+Hess(5,2)) 0 1 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) (Hess(3,5)+Hess(5,3)) 0 0 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) (Hess(4,5)+Hess(5,4)) 0 0 1];

AA(5,:) = [-1 0 0 0 0 0 0 0];

AA(6,:) = [0 1 0 0 0 0 0 0];

AA(7,:) = [0 0 0 1 0 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(1)];

CC(6,:) = [-G(2)];

CC(7,:) = [-G(4)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);
```

```
u(1) = temp_mult(5,1);
u(2) = temp_mult(6,1);
u(4) = temp_mult(7,1);
```

```
% Filname : "mult14"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) -1 0 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) 0 1 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) 0 0 1];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) 0 0 0];

AA(6,:) = [-1 0 0 0 0 0 0];

AA(7,:) = [0 1 0 0 0 0 0];

AA(8,:) = [0 0 1 0 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(1)];

CC(6,:) = [-G(2)];

CC(7,:) = [-G(3)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);

dd(3,1) = temp_mult(3,1);

dd(4,1) = temp_mult(4,1);
```

```
u(1) = temp_mult(5,1);
u(2) = temp_mult(6,1);
u(3) = temp_mult(7,1);
```

```
% Filname : "mult15"

AA(1,:) = [(Hess(1,1)) (Hess(1,2)+Hess(2,1)) (Hess(1,3)+Hess(3,1))
(Hess(1,4)+Hess(4,1)) (Hess(1,5)+Hess(5,1)) -1 0 0 0];

AA(2,:) = [(Hess(2,1)+Hess(1,2)) (Hess(2,2)) (Hess(2,3)+Hess(3,2))
(Hess(2,4)+Hess(4,2)) (Hess(2,5)+Hess(5,2)) 0 1 0 0];

AA(3,:) = [(Hess(3,1)+Hess(1,3)) (Hess(3,2)+Hess(2,3)) (Hess(3,3))
(Hess(3,4)+Hess(4,3)) (Hess(3,5)+Hess(5,3)) 0 0 1 0];

AA(4,:) = [(Hess(4,1)+Hess(4,1)) (Hess(4,2)+Hess(2,4)) (Hess(4,3)+Hess(3,4))
(Hess(4,4)) (Hess(4,5)+Hess(5,4)) 0 0 0 1];

AA(5,:) = [-1 0 0 0 0 0 0 0];

AA(6,:) = [0 1 0 0 0 0 0 0];

AA(7,:) = [0 0 0 1 0 0 0 0];

AA(8,:) = [0 0 0 0 1 0 0 0];

CC(1,:) = [-dytn(1)];

CC(2,:) = [-dytn(2)];

CC(3,:) = [-dytn(3)];

CC(4,:) = [-dytn(4)];

CC(5,:) = [-G(1)];

CC(6,:) = [-G(2)];

CC(7,:) = [-G(3)];

CC(8,:) = [-G(4)];

temp_mult = inv(AA)*CC;

dd(1,1) = temp_mult(1,1);

dd(2,1) = temp_mult(2,1);
```

```
dd(3,1) = temp_mult(3,1);
dd(4,1) = temp_mult(4,1);
u(1) = temp_mult(5,1);
u(2) = temp_mult(6,1);
u(3) = temp_mult(7,1);
u(4) = temp_mult(8,1);
```

```
% filename : "step_size"
% This subroutine determines the proper step size for the optimization procedure

flag = 0;

Beta = gamma*(abs_dd)^2;

PHI_o = Y + R(kk+1)*V;

mm = 0;

for flag = 0;

  t = (1/2)^mm

  X = [xn(1) xn(2) xn(3) xn(4)] + t*transpose(dd)

  a(2) = X(1)*(a(1)+DELTA);

  a(3) = X(2)*(a(4)-OH);

  Kf = X(3)*Kf_max;

  Kr = X(4)*Kr_max;

  deformationo

  PHI_1 = Y+R(kk+1)*V;

  if PHI_1 >= PHI_o - t*Beta

    mm = mm+1;

    flag = 0;

  else

    flag = 1;

    a(2) = X(1);

        a(3) = X(2);

        Kf = X(3);
```