Computer Simulation of the
Bristol Compressor Suspension System Dynamics

by
Ramakant P. Arcot

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute & State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Mechanical Engineering

APPROVED:

Reginald G. Mitchiner, Chairman

Robert G. Leonard

Charles F. Reinholtz

November 17, 1993
Blacksburg, Virginia
Computer Simulation of the
Bristol Compressor Suspension System Dynamics

by
Ramakant P. Arcot
Reginald G. Mitchiner, Chairman
Mechanical Engineering
(ABSTRACT)

The objective of this research is the computer simulation of the vibrations of the suspension system of a two-cylinder reciprocating compressor. A theoretical model is developed to describe the various steps undertaken to calculate the response of this six-degree-of-freedom rigid system. The response, which is in the form of a displacement vector, serves as the input to a computer animation of the motion of the orbit of the compressor with respect to the four suspension system springs.

The theoretical model is developed by calculating (1) the System Mass and Inertial Matrix, (2) the Gyroscopic Matrix, (3) the Total Assembly Stiffness Matrix, and (4) the Shaking Forces and Moments Matrix. Experimental and finite element methods used to evaluate the parameters required to calculate these matrices are also discussed.

An eigenanalysis is performed to calculate the eigenvalue frequencies and eigenvectors for the system. The force analysis is performed to calculate the forcing function in the time domain for the first 40 harmonics. The Fast Fourier Transform method is used to transform the forcing function from the time domain to the frequency domain. The validity of the results are checked by simultaneously developing another model using IMP (Integrated Mechanisms Program). The response is then calculated in original coordinates, after performing a modal transformation.

Finally, the response, which is a displacement vector, is utilized by an animation program in PHIGS (Programmer's Hierarchical Interactive Graphics Standard) to animate the motion of the orbit of the compressor.
Acknowledgments

First of all, I would like to express my sincere gratitude to Dr. R.G. Mitchiner, my major professor and graduate advisor, for his guidance, encouragement and personal interest during the development and completion of this research.

I am also very grateful to Dr. R.G. Leonard and Dr. C.F. Reinholtz for serving on my graduate committee. Their guidance and support have been invaluable.

I would also like to thank Dr. L.D. Mitchell, Dr. C.E. Knight and Dr. C.J. Hurst for their invaluable guidance, and without whose support this research might not have been possible.

Special thanks go to Bristol Compressors Inc., the sponsors of this project, represented by Mr. David Giiliam, for his endless supply of drawings and information pertaining to the research.

My thanks also go to the instrument shop personnel and the machinists in the mechanical engineering department of VPI&SU. Without the professional help of Mr. Johnny Cox, Mr. Billy Shepherd and Mr. Benjamin Poe, this work could not have been completed.

Last, but not the least, I would like to thank my parents, my brothers and sister, for all the love they bestowed upon me throughout my life.
# Table of Contents

1.0 Introduction ........................................................................................................ 1

2.0 Literature Review ............................................................................................. 6
  2.1 Force analysis in reciprocating machinery .................................................. 6
  2.2 Computer Simulation ................................................................................... 8

3.0 Theoretical Development of the Model ......................................................... 11
  3.1 Dynamics of Rigid Body Motion .................................................................... 11
  3.2 System Equations of Motion ......................................................................... 12
    3.2.1 Coordinate system for the compressor assembly ................................... 17
    3.2.2 System Mass and Inertial Matrix ........................................................... 18
    3.2.3 System Gyroscopic Matrix ................................................................... 21
    3.2.4 Total System Stiffness Matrix ............................................................... 26
      3.2.4.1 Spring Numbering Convention ..................................................... 26
      3.2.4.2 The Coiled Spring ......................................................................... 26
      3.2.4.3 Derivation of the Element Stiffness Matrix .................................... 27
      3.2.4.4 Computation of the Total Assembly Stiffness Matrix .................... 31
    3.2.5 Force and Moment Matrix ..................................................................... 32
      3.2.5.1 Application of equivalent masses to replace the connecting rod ...... 32
      3.2.5.2 Force analysis of a slider crank mechanism using point masses ...... 34
      3.2.5.3 Effect of reaction forces at the crank pin and the wrist pin ............ 37
      3.2.5.4 Effect of shaft torque at the center of mass .................................... 39
      3.2.5.5 Effect of forces due to rotating counterweights ......................... 39
3.2.5.6 The Total Shaking Forces and Moments Matrix ........................................ 42
3.2.6 Response Analysis of the Compressor System ............................................... 43

4.0 Evaluation of Parameters ............................................................................. 45
4.1 Center of mass of the assembly ................................................................. 45
4.2 Moments of Inertia of the assembly .......................................................... 50
  4.2.1 Effect of damping on oscillations ......................................................... 63
4.3 Determination of Part Weight Values ......................................................... 65
4.4 Calculation of System Mass and Inertial Matrix ....................................... 65
4.5 Center of mass of the rotor ............................................................... 67
4.6 Moment of inertia of the rotor ............................................................. 72
4.7 Computation of the System Gyroscopic Matrix ....................................... 75
4.8 Calculation of coordinates of spring attachment points ....................... 77
4.9 Lateral Stiffnesses of Springs ............................................................. 78
4.10 Moment Stiffnesses of Springs ............................................................ 82
4.11 Total Assembly Stiffness Matrix ........................................................... 85
4.12 Eigenanalysis comparison between analytical and experimental methods .................. 88
4.13 Application of point masses to the connecting rod ................................... 92
4.14 The Total Shaking Forces and Moments Matrix ....................................... 94

5.0 Discussion of Results ................................................................................... 104
5.1 Verification with Integrated Mechanisms Program (IMP) ....................... 104
5.2 Response analysis of the system ............................................................. 111
6.0 Conclusion and Recommendations ................................................................. 113

6.1 Animation of the orbit of the compressor ..................................................... 113

6.2 Recommendation for further research ......................................................... 121

   6.2.1 Development of a variable speed vibration model .............................. 121

Appendix A. Mass Properties ........................................................................ 122

Appendix B. Total Assembly Stiffness Matrix ................................................... 123

Appendix C. Response program BRESP.FOR ..................................................... 124

Appendix D. Comparison of eigenvalues ......................................................... 144

Appendix E. Output of program BRESP.FOR .................................................... 145

Appendix F. Force analysis subroutines ........................................................... 150

Appendix G. Integrated Mechanisms Program ............................................... 154

Appendix H. Spring Stiffnesses - Finite Element Solution ................................ 158

Appendix I. Orbit Animation Program BR_ANIM.C ....................................... 161

Appendix J. Experimental determination of spring stiffnesses ....................... 194

References ...................................................................................................... 196

Vita ................................................................................................................. 198

Table of Contents
List of Illustrations

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The Bristol H-25A two-cylinder reciprocating compressor</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>The compressor analysis process</td>
<td>5</td>
</tr>
<tr>
<td>3.</td>
<td>The Bristol H-25A compressor assembly</td>
<td>14</td>
</tr>
<tr>
<td>4.</td>
<td>A general dynamic rigid body system</td>
<td>15</td>
</tr>
<tr>
<td>5.</td>
<td>Location of the system coordinate axes</td>
<td>19</td>
</tr>
<tr>
<td>6.</td>
<td>Gyroscopic action in a system (from Mabie and Reinholtz [11])</td>
<td>23</td>
</tr>
<tr>
<td>7.</td>
<td>Gyroscopic moments due to rotation of the rotor</td>
<td>25</td>
</tr>
<tr>
<td>8.</td>
<td>A six-degree-of-freedom spring in 3-D space</td>
<td>28</td>
</tr>
<tr>
<td>9.</td>
<td>Replacement of connecting rod using equivalent masses</td>
<td>33</td>
</tr>
<tr>
<td>10.</td>
<td>Force analysis of a typical slider-crank mechanism</td>
<td>35</td>
</tr>
<tr>
<td>11.</td>
<td>Inertia forces due to a rotating counterweight</td>
<td>41</td>
</tr>
<tr>
<td>12.</td>
<td>Determination of the center of mass</td>
<td>46</td>
</tr>
<tr>
<td>13.</td>
<td>The circular plate for determining inertia</td>
<td>51</td>
</tr>
<tr>
<td>14.</td>
<td>The plug for the spherical bearing to hold the circular plate</td>
<td>52</td>
</tr>
<tr>
<td>15.</td>
<td>The oscillating circular plate for the torsional pendulum</td>
<td>54</td>
</tr>
<tr>
<td>16.</td>
<td>Determination of the moment of inertia of the assembly about X-direction</td>
<td>57</td>
</tr>
<tr>
<td>17.</td>
<td>Determination of the moment of inertia of the assembly about Y-direction</td>
<td>60</td>
</tr>
<tr>
<td>18.</td>
<td>Determination of the moment of inertia of the assembly about Z-direction</td>
<td>62</td>
</tr>
<tr>
<td>19.</td>
<td>Determination of the center of mass in the axial direction</td>
<td>68</td>
</tr>
<tr>
<td>20.</td>
<td>A solid model of the H-25A rotor</td>
<td>69</td>
</tr>
<tr>
<td>21.</td>
<td>Determination of the center of mass in the radial direction</td>
<td>71</td>
</tr>
<tr>
<td>22.</td>
<td>The triangular plate for holding the rotor</td>
<td>74</td>
</tr>
<tr>
<td>23.</td>
<td>Spring under combined axial and lateral loading (from Wahl [14])</td>
<td>79</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>24</td>
<td>Chart for finding factor $C_i$ (from Wahl [14])</td>
<td>80</td>
</tr>
<tr>
<td>25</td>
<td>Spring subjected to moment in the plane of its axis (from Wahl [14])</td>
<td>83</td>
</tr>
<tr>
<td>26</td>
<td>Eigen function minimum vs. stiffness plots for the top spring</td>
<td>90</td>
</tr>
<tr>
<td>27</td>
<td>Eigen function minimum vs. stiffness plots for the side spring</td>
<td>91</td>
</tr>
<tr>
<td>28</td>
<td>Application of point masses to the connecting rod</td>
<td>93</td>
</tr>
<tr>
<td>29</td>
<td>The assumed P-V diagram for the H-25A compressor (k=1.30)</td>
<td>95</td>
</tr>
<tr>
<td>30</td>
<td>The output torque of the rotor</td>
<td>96</td>
</tr>
<tr>
<td>31</td>
<td>Piston displacement vs. angular rotation of the crank</td>
<td>97</td>
</tr>
<tr>
<td>32</td>
<td>Force in the X-direction acting at the center of mass of the assembly</td>
<td>98</td>
</tr>
<tr>
<td>33</td>
<td>Force in the Y-direction acting at the center of mass of the assembly</td>
<td>99</td>
</tr>
<tr>
<td>34</td>
<td>Moment about X-direction acting at the center of mass of the assembly</td>
<td>100</td>
</tr>
<tr>
<td>35</td>
<td>Moment about Y-direction acting at the center of mass of the assembly</td>
<td>101</td>
</tr>
<tr>
<td>36</td>
<td>Moment about Z-direction acting at the center of mass of the assembly</td>
<td>102</td>
</tr>
<tr>
<td>37</td>
<td>The H-25A assembly as defined in IMP</td>
<td>105</td>
</tr>
<tr>
<td>38</td>
<td>Plots of pressure force vs. crank angle for the two pistons (k=1.30)</td>
<td>106</td>
</tr>
<tr>
<td>39</td>
<td>Comparison of forces in X between FORTRAN program and IMP</td>
<td>108</td>
</tr>
<tr>
<td>40</td>
<td>Comparison of forces in Y between FORTRAN program and IMP</td>
<td>109</td>
</tr>
<tr>
<td>41</td>
<td>Comparison of moments about Z between FORTRAN program and IMP</td>
<td>110</td>
</tr>
<tr>
<td>42</td>
<td>Animation of the orbit of motion of the assembly using PHIGS</td>
<td>114</td>
</tr>
<tr>
<td>43</td>
<td>Displacement in X vs. number of time intervals</td>
<td>116</td>
</tr>
<tr>
<td>44</td>
<td>Displacement in Y vs. number of time intervals</td>
<td>117</td>
</tr>
<tr>
<td>45</td>
<td>Rotation about X vs. number of time intervals</td>
<td>118</td>
</tr>
<tr>
<td>46</td>
<td>Rotation about Y vs. number of time intervals</td>
<td>119</td>
</tr>
<tr>
<td>47</td>
<td>Rotation about Z vs. number of time intervals</td>
<td>120</td>
</tr>
<tr>
<td>48</td>
<td>A sample FRF for the suspension</td>
<td>196</td>
</tr>
</tbody>
</table>

List of Illustrations viii
Nomenclature

The following symbols are used throughout this research.

\( M_g \)  System Mass and Inertial Matrix

\( C_g \)  System Gyroscopic Matrix

\( K_g \)  Total Assembly Stiffness Matrix

\( F_g \)  Shaking Forces and Moments Matrix

\( \omega \)  Angular velocity of the crankshaft

\( X_g \)  Displacement vector for the center of mass of the frame

\( x \)  Positive displacement in the X-direction

\( y \)  Positive displacement in the Y-direction

\( z \)  Positive displacement in the Z-direction

\( \theta_x \)  Counter-clockwise angular rotation about the X-direction

\( \theta_y \)  Counter-clockwise angular rotation about the Y-direction

\( \theta_z \)  Counter-clockwise angular rotation about the Z-direction

\( M_f \)  Mass matrix of the system

\( I_f \)  Inertia matrix of the system

\( M_x \)  Gyroscopic moment about the X-direction

\( M_y \)  Gyroscopic moment about the Y-direction

\( K_{xx} \)  Lateral stiffness of a spring element in the X-direction

\( K_{yy} \)  Lateral stiffness of a spring element in the Y-direction

\( K_{zz} \)  Axial stiffness of a spring element in the Z-direction

\( K_{belix} \)  Moment stiffness of a spring element about the X-direction
$K_{byby}$  Moment stiffness of a spring element about the Y-direction

$K_{teke}$  Torsional stiffness of a spring element about the Z-direction

$R_x$  X coordinate of the spring attachment point

$R_y$  Y coordinate of the spring attachment point

$R_z$  Z coordinate of the spring attachment point

$K_{Spring#1}$  Stiffness matrix of the top spring

$K_{Spring#2}$  Stiffness matrix of the left side spring

$K_{Spring#3}$  Stiffness matrix of the right side spring

$K_{Shockloop}$  Stiffness matrix of the shockloop

$M_{rot}$  Rotating point mass at the crank-pin center of the connecting rod

$M_{trans}$  Translating point mass at the wrist-pin center of the connecting rod

$W$  Weight of the connecting rod

$l_{rot}$  Center of mass and crank-pin center distance of the connecting rod

$l_{trans}$  Center of mass and wrist-pin center distance of the connecting rod

$l$  Moment of inertia of the connecting rod

$M_{cw}$  Counterweight mass added to the crank

$F_{12}$  Main bearing force at the crank

$F_{14}$  Reaction force from the cylinder to the piston

$F_B$  Total force acting at the wrist-pin center

$P$  Gas pressure force

$F_{O4}$  Force developed due to the acceleration of the piston

$\theta$  Angle of the crank

$d$  Perpendicular distance between the connecting rod and the cylinder axis

$h$  Distance between the crank-pin center and the wrist-pin center

$T$  Output torque developed
\( T_s \) \hspace{1cm} \text{Shaft torque}

\( F_{41} \) \hspace{1cm} \text{Reaction of the piston on the block or the frame}

\( F_{21} \) \hspace{1cm} \text{Reaction of the crank on the main bearings}

\( F_{g\text{REACTION}} \) \hspace{1cm} \text{Force and Moments matrix due to reaction forces}

\( F_{g\text{TOTCW}} \) \hspace{1cm} \text{Force and Moments matrix due to counterweight forces}

\( F_{g\text{TORQUE}} \) \hspace{1cm} \text{Torque effect matrix}

\( (R_{xcpin}, R_{ycpin}, R_{zcpin}) \) \hspace{1cm} \text{Position vector of crank-pin center with respect to frame coordinate axes}

\( (R_{xwpin}, R_{ypwpin}, R_{zwpin}) \) \hspace{1cm} \text{Position vector of wrist-pin center with respect to frame coordinate axes}

\( P \) \hspace{1cm} \text{Modal matrix of the system}

\( m \) \hspace{1cm} \text{Generalized Mass Matrix}

\( c \) \hspace{1cm} \text{Generalized Gyroscopic Matrix}

\( k \) \hspace{1cm} \text{Generalized Stiffness Matrix}

\( F_o \) \hspace{1cm} \text{Generalized Forcing Function Matrix}
1.0 Introduction

The main purpose of the suspension system of a reciprocating compressor is the isolation of the compressor motions and forces from the hermetic shell surrounding the compressor. Isolation of these forces and motions would prevent vibration of the shell and transmission of vibration from the shell into the structure supporting the compressor. The noise radiation from the shell could also be reduced by attenuating the vibration of the shell.

In order to help the designer predict the response of the compressor for various proposed suspension designs, a vibration model and a simulation of the compressor will not only reduce the time and effort needed to find improved designs, but would also prove useful to test the designs before such designs are built into a compressor. This study is an effort to develop a modeling and simulation program for a two-cylinder reciprocating freon compressor, elastically mounted to a hermetic shell. The hermetic shell is assumed to have negligible motion compared to the compressor frame.

The objectives of this research are (1) prediction of the orbit of the compressor, (2) generation of a forcing function matrix, which could be used later for a finite element solution of the hermetic shell, and (3) suggest changes in design in order to reduce the transmission of vibration from the frame of the compressor to the shell.

An extensive survey of the available literature was done in the areas of reciprocating compressor force analysis and simulation of reciprocating compressor motion. Chapter 2 presents
brief descriptions of some of the studies done by other researchers in the fields of reciprocating compressor simulation and other relevant areas.

An analytical model for the simulation of the suspension system of the compressor was developed taking into account the results of the literature survey. This modeling was divided into three efforts, namely i) eigenanalysis, ii) force analysis and iii) orbit response analysis of the system.

An eigenanalysis is performed on the compressor frame - springs arrangement to give the eigenvalues and the normalized modal matrix. The springs under consideration are the top spring, the two side springs (from Fig. 5), and the shockloop which is treated as a spring because of its attachment to the muffler port. The stiffness matrix for each spring is developed using the method of stiffness influence coefficients, and is referenced to the center of mass of the assembly. The total assembly stiffness matrix is obtained by summing all the stiffness matrices for the four springs mentioned above.

In the force analysis, the mass of each connecting rod is replaced by two masses, one at the wrist pin end, called the translating mass, and another at the crank pin end called the rotating mass. The forces due to all rotating and translating masses, including the counterweights attached to the rotor and the crankshaft, are summed up to give i) the reaction of the frame on the piston and ii) the reaction of the frame on the main crank bearing. The reaction torque developed due to rotation of the rotor is also taken into consideration in the analysis. The forces and moments due to these reactions, all translated to the center of mass of the compressor assembly, are calculated. Then a Fast Fourier Transform for the first 40 harmonics is taken consecutively to give the coefficients of the cosine and sine terms of the forcing function matrix, and thus, convert the forcing function

1. Introduction 2
from the time domain to the frequency domain. Also, the effect of gyroscopic moments due to the rotation of the rotor armature are taken into account to give the total system gyroscopic matrix.

In the orbit response analysis, the set of equations of motion is solved for the displacement vector. The displacement vector then acts as an input to a time simulation. The simulation program is designed for a Bristol H-25A, single stage twin cylinder freon reciprocating compressor using advanced graphic techniques by running it with a PHIGS interface. An assembly of the Bristol H-25A compressor is shown in Fig. 1.

A summary of the total analysis process as defined and implemented in this research is shown in Fig. 2.
Figure 1. The Bristol H-25A two cylinder reciprocating compressor
Figure 2. The compressor analysis process
2.0 Literature Review

2.1 Force analysis in reciprocating machinery

Hirschhorn [4] in 1968, emphasizing the inertia effects due to the shaking force in reciprocating machinery, suggested that in order to simplify calculations, the connecting rod be replaced by two statically equivalent point masses, one at the wrist pin and the other at the crank pin. The wrist pin mass is then lumped together with the other reciprocating masses, and the crank pin mass is added to the rotating parts. Since the static replacement of the mass of the connecting rod changes neither the total mass nor the position of the center of gravity, the resultant of the inertia forces of the two point masses has the same magnitude and direction as the actual inertia force of the connecting rod. However, its line of action lies on the axis of rotation of the crankshaft. Also, the inertia force due to the rotating mass of the connecting rod is easily reduced to zero by balancing it with a mass added to the counterweight to the crank, to induce an inertia force, equal in magnitude as the inertia force due to the rotating mass of the connecting rod, but opposite in direction. Since the gas forces generated in the cylinder are internal and, as such, have no external effects, the only force that must be given consideration is the inertia force due to the reciprocating mass of the connecting rod. This force varies periodically in magnitude and sense along a fixed line of action, and is known as the 'shaking force'. If unchecked, it may excite dangerous vibrations of adjacent equipment or of the building in which the machine is housed.

Mabie and Ocvirk [2] in 1975 proposed the use of approximate kinetically equivalent point masses to replace the mass of the connecting rod in the internal combustion engine mechanism.
One of the point masses was located at the wrist-pin axis and the other at the crank-pin axis. These masses would generate forces, one of which would act on the line of reciprocation of the wrist-pin, and the other which would always act radially outward on the crank line when the crank was rotating at a uniform speed. By adding a counterweight to the crank, so that an inertia force is induced to balance the radially acting force of the connecting rod mass, the masses rotating with the crank were balanced, so that no forces from these masses would act on the main bearings. Also, as there were no transverse components of forces to bend or shear the connecting rod, its force component was found to be acting in the axial direction. This would make it possible to undertake the force analysis of the engine without superposition.

Solodilov and Shchepeitl'nikov [1] in 1988 proposed the balancing of crank and connecting rod mechanisms by the null vector method. They substantiated the feasibility of use of the null vector method to balance hermetic and transport piston compressors. Their theory of the mechanism proved that the position of the mass center of a crank and connecting rod mechanism was equal to the geometrical sum of the principal point vectors of the moving links. The principal point vectors, positioning the mass centers of the links of a crank and connecting rod mechanism could be determined analytically or experimentally, and when the unbalances in masses were eliminated, complete static balancing of the mechanism using the null vector method could be achieved. However the null vector method is rarely used in practice owing to the constructional difficulties in placing the correcting masses of the connecting rod and the crank within the housings of piston machines.

Kui [3] in 1988 presented the method and principle of loading the connecting rod by a large centrifugal machine used for photoclastic experiments, to elucidate the fact that the connecting rod of an internal combustion engine is subjected to inertia loading of its own
distributed mass. Kui [3] also developed a theoretical formula simulating the acceleration along each of the cross sections of the connecting rod and, therefore solved the problem of measuring the rod's stress according to the distributed mass method. This theory is also known as the whipping phenomena. The results of his work show that a real inertia load can be exerted on the connecting rod, in accordance with distributed mass and acceleration of each cross section, thus eliminating the approximation of the replacement system of two-masses for the connecting rod.

2.2 Computer simulation

In 1982, Hamilton [7], Professor of Mechanical Engineering, in the Ray W. Herrick Laboratories of the school of Mechanical Engineering at Purdue University, one of the centers of research and development for analysis of positive displacement compressors, developed a modeling and simulation program for a single cylinder compressor, elastically sprung from a rigid hermetic shell.

The development of the compressor vibration model was made on the following assumptions.

1. The hermetic shell containing the compressor has zero motion.
2. All body forces due to the gravity field, (i.e., vibration is about the static equilibrium position) are omitted.
3. The effect of the discharge pressure within the discharge line is omitted, due to the stiffening of the shockloop.

2. Literature Review
4. The motions of the compressor frame are assumed to be of such levels that linear assumptions are sufficiently accurate.

Chucholowski and Woschni [6] in 1987 developed a computer program which was able to simulate the piston slap motion in a diesel engine more precisely than it was possible hitherto. The physical structure of the program is described in their paper on the computer simulation of piston slap motion. Parallel to the development of the program, measurements on a single-cylinder diesel-engine were made to obtain the necessary boundary conditions and to be able to check the results of the calculations. It has been proved in their work that the computer program is able to predict the piston slap motion for different sizes and constructions of pistons. In the fitted pistons of the Bristol H-25A, slap is minimized.

Soong [5] in 1988 presented an experimental investigation in the area of development of analytical and computational models for predicting the dynamic response of mechanical systems comprising assemblages of rigid bodies with clearance at the joints. He suggested a dearth of experimental investigations in this area which focus on furnishing complementary response data, consequently raising a question on the predictive capabilities of these theoretical models. Soong [5] expressed his goal of filling in this void by undertaking a comprehensive experimental study of a slider-crank mechanism in which the radial clearance at the wrist pin bearing was carefully controlled. Response data at different operating speeds are presented, and finally, design guidelines are distilled from the experimental data to relate the response of the system to the response of a similar system without bearing clearance.

The simulation was developed such that the shaking forces and moments vector will be a function of the crank angular motion, due to the acceleration of the elements within the compressor frame.

2. Literature Review
Vector and matrix operations were used for calculations, which made it convenient for programming the operation on a computer. One of the major factors affecting the accuracy of the modeling and simulation of the vibrations of the compressor was the assumption that the shaking forces due to the moving parts within the compressor could be calculated assuming that the crankshaft of the compressor rotated with uniform velocity, thus having zero angular acceleration. A more accurate analysis could be pursued by assuming the crankshaft to rotate with variable speed, and calculate the instantaneous velocity for each instantaneous angular rotation of the crank. The force analysis is also based upon the assumption that there is no force acting in the Z direction at the center of mass of the assembly. The balancing of the connecting rod with two masses is a very suitable method to approximate the mass of the connecting rod, because, with little error, the approximation could save considerable effort.
3.0 Theoretical Development of the Model

3.1 Dynamics of Rigid Body Motion

The study of the dynamics of classical rigid body motion is based upon the laws of motion published in 1687 by Sir Isaac Newton. According to Newton's second law

The rate of change of linear momentum of a body with respect to time is proportional to the force acting upon it and occurs in the direction of action of the force.\[8\]

This law is often stated in terms of force, mass and acceleration as

The force acting on a body is equal to the product of its mass and its acceleration.

In equation form, translational motion of a body can be described by writing the second law as

\[ \sum F = ma \]  

(1)

where \( \Sigma F \) is the sum of the forces acting on the body, and \( m \) and \( a \) are the mass and acceleration of the mass center of the body, respectively.

To describe rotational motion of a body, the equation can be written as

\[ \sum M = J\alpha \]  

(2)

where \( \Sigma M \) is the sum of the moments acting on the body, \( J \) is the mass moment of inertia of the body about its axis of rotation, and \( \alpha \) is the angular acceleration about that axis.
For dynamics of a rigid body in three dimensional space, these equations are written as

\[ \sum F_x = m\ddot{x} \quad \sum F_y = m\ddot{y} \quad \sum F_z = m\ddot{z} \]  
\[ \sum M_x = J\ddot{\theta}_x \quad \sum M_y = J\ddot{\theta}_y \quad \sum M_z = J\ddot{\theta}_z \]  

where \( \sum F_x \), \( \sum F_y \), and \( \sum F_z \) represent the forces due to linear accelerations in the respective coordinate directions, and \( \sum M_x \), \( \sum M_y \), and \( \sum M_z \) represent the moments due to the rotational accelerations about the respective coordinate axes.

3.2 System Equations of Motion

The set of ordinary differential equations used to simulate the vibrations of the compressor frame are developed in this chapter. The assumptions on which the modeling is based are as follows.

1. The hermetic shell containing the compressor has zero motion.

2. The frequency response is calculated for frequencies above the 2 kHz range, as considerable modal activity is expected to occur at that frequency range.

3. The compressor frame, rotor, crank, connecting rod and piston are assumed to be rigid bodies.

4. The discharge line, known as the shockloop, is treated as a spring for calculation purposes.

5. All the springs, i.e., the top spring, two side springs and the shockloop are assumed to be massless linearly deformable bodies. Hence the surging modes of the spring are not taken into

3. Theoretical Development of the Model
account in this analytical model, as compared to the finite element model where these surging modes are considered.

6. The inertial forces of the rotor, crank, connecting rod and the piston are calculated with respect to their positions from the center of mass of the assembly.

7. The compression and expansion processes are expressed as polytropic processes and the suction and discharge processes are expressed as constant pressure processes.

A schematic of a typical Bristol H-25A, two cylinder compressor is shown in Fig. 3. The compressor consists of a frame, rotor, piston assembly and mounting springs. The piston assembly consists of the crank, connecting rod and the piston. The mounting springs comprise four springs. These are the top mounting spring, the two "identical" side mounting spring (shown in Fig. 5), and the shockloop which is also treated as a spring. The inertial coordinate system is placed at the equilibrium position of the center of gravity of the frame and the motion of the frame is given in terms of this coordinate system.

A general equation of motion describing rigid body dynamics will include contributions from terms having i) the mass, ii) the stiffness and iii) the coefficient of damping. Applying Newton's second law,

\[ \sum F_x = m\ddot{x} \]  

we have for the general system shown in Fig. 4

3. Theoretical Development of the Model
Figure 3. The Bristol H-25A compressor assembly
Figure 4. A general dynamic rigid body system
\[ m\ddot{x} = -c\dot{x} - kx \]  

(6)

where \( m \) is the mass of the body, \( k \) is its stiffness, and \( c \) is the damping in the system.

Considering a dynamic rigid body system with an applied time varying force \( F(t) \)

\[ M\ddot{x} + C\dot{x} + Kx = F(t) \]  

(7)

This general equation may be applied in the form of matrices for systems with more than one degree-of-freedom. For an unforced system the equation of motion appears as

\[ M\ddot{x} + C\dot{x} + Kx = 0 \]  

(8)

and for the forced system the equation of motion can be written as

\[ M_g\ddot{x} + C_g\dot{x} + K_gx = F_g(t) \]  

(9)

Normally, for a multi-degree-of-freedom body, i.e., a body that has more than one degree-of-freedom, its number of degrees-of-freedom represent the number of coordinates required to fully describe its position and orientation in space. Hence it is common to describe a real system in terms of its number of degrees of freedom.

The mathematical model will be in the following format

\[ M_g(\ddot{X}_g) + C_g(\dot{X}_g) + K_g(X_g) = [F_g] \]  

(10)

3. Theoretical Development of the Model
where \([M_g]\) denotes the system mass and inertial matrix, \([C_g]\) denotes the system damping matrix, 
\([K_g]\) denotes the total assembly stiffness matrix and \([F_g]\) denotes the shaking forces and moments matrix for the system, which will be a function of the angular velocity of the crankshaft, \(\omega\), due to the acceleration of the elements within the frame of the compressor.

The frame center of gravity displacement vector is

\[
(X_e)^T = [x, y, z, \theta_x, \theta_y, \theta_z]
\]  

(11)

The frame center of gravity velocity vector is

\[
(\dot{X}_e)^T = [\dot{x}, \dot{y}, \dot{z}, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z]
\]  

(12)

The frame center of gravity acceleration vector is

\[
(\ddot{X}_e)^T = [\ddot{x}, \ddot{y}, \ddot{z}, \ddot{\theta}_x, \ddot{\theta}_y, \ddot{\theta}_z]
\]  

(13)

The following sections describe the development of the coordinate system, the system matrices \([M_g], [C_g], [K_g]\) and the shaking forces and moments matrix \([F_g]\).

3.2.1 Coordinate system for the compressor assembly

A three-dimensional coordinate system was adopted for the compressor assembly, the origin of which was located at the center of mass of the assembly. The positive direction of \(X\) coordinate axis was chosen in a direction perpendicular to the cylinder axis and along the face of the cylinder. The positive direction of the \(Y\) coordinate axis was chosen in a direction parallel to the cylinder axis towards the muffler port. Maxwell's Right Hand Rule was applied to determine the \(Z\) axis, whose positive direction was parallel to the crankshaft axis towards the motor cap (from Fig. 5).
For all cases, displacements along each positive coordinate axis were taken as positive displacements, and all counter-clockwise rotations about each positive coordinate axis were taken as positive rotations, consistent with the Right Hand rule, as shown in Fig. 5.

The location of the center of mass of the assembly will become an important aspect, because that would locate the coordinate axes for the system, with respect to which all the calculations are undertaken, e.g., the location of the attachment points of the springs would be with respect to the system coordinate axes, and the location of any of the compressor parts like the rotor, crankshaft or the piston would also be with respect to the system coordinate axes. The location of the center of mass with respect to an origin located at the Top Dead Center of the bottom cylinder (Ref. Fig. 3), is

\[(x_o, y_o, z_o) = (0.01, -3.51, 5.71)\]

3.2.2 System Mass and Inertial Matrix

The mass and inertial matrix emphasizes the mass and inertial properties of the compressor assembly. It includes both translational and rotational terms for the frame such as

\[
\begin{bmatrix}
M_e & 0 \\
0 & I_e
\end{bmatrix}
\]

(14)
Figure 5. Location of the system coordinate axes
where the mass and moment of inertia sections of the system mass and inertia matrix are

\[
M_{f'} = \begin{bmatrix}
M_f & 0 & 0 \\
0 & M_f & 0 \\
0 & 0 & M_f
\end{bmatrix}
\]  

(15)

and

\[
I_{fc} = \begin{bmatrix}
I_{xx} & -I_{xy} & -I_{xz} \\
-I_{yx} & I_{yy} & -I_{yz} \\
-I_{zx} & -I_{zy} & I_{zz}
\end{bmatrix}
\]

(16)

As the fixed coordinate axes of the frame are generally not the principal coordinate axes of the frame, the moment of inertia matrix for the frame has terms contributed by cross-products of inertia.

Hence the system mass and inertial matrix can be written as

\[
[M_s] = \begin{bmatrix}
M_f & 0 & 0 & 0 & 0 & 0 \\
0 & M_f & 0 & 0 & 0 & 0 \\
0 & 0 & M_f & 0 & 0 & 0 \\
0 & 0 & 0 & I_{xx} & -I_{xy} & -I_{xz} \\
0 & 0 & 0 & -I_{yx} & I_{yy} & -I_{yz} \\
0 & 0 & 0 & -I_{zx} & -I_{zy} & I_{zz}
\end{bmatrix}
\]

(17)
3.2.3 System Gyroscopic Matrix

In a system that has rotating parts having high moment of inertia, or high rotational speeds, gyroscopic forces are in action when the system is changing direction of motion. Automotive vehicles undergoing roadway turns at high velocity or engine parts as well as the propeller shaft of an airplane, circling before landing, are under the action of gyroscopic effects due to turns and pullouts.

According to Newton's second law of motion, discussed by Greenwood [8], which is also discussed in section 3.1, the rate of change of angular momentum with respect to time is proportional to an applied torque. Mathematically

\[ T = \frac{d}{dt}(l\omega) \]  

(18)

Also since

\[ H = l\omega \]  

(19)

Therefore,

\[ T = \frac{dH}{dt} \]  

(20)

where \( H \) is the angular momentum of the spinning body, \( l \) is its moment of inertia, and \( \omega \) is its angular speed.

3. Theoretical Development of the Model
According to Mabie and Reinholtz [11], for the case shown in Fig. 6, gyroscopic action results if the spin axis is made to change its angular position with respect to the x axis, with the rotation of the body at an angular velocity, \( \omega \), about the spin axis. For a small angular displacement, \( \Delta \theta \), of the spin axis, the magnitude of the change in the angular momentum, \( \Delta H \), is

\[
\Delta H = (I \omega) \Delta \theta
\]  

(21)

The time rate of change of angular momentum for a system whose spin axis displaces angularly at the rate of

\[
\omega_p = \frac{d \theta}{dt}
\]  

(22)

is

\[
\frac{dH}{dt} = \lim_{\Delta t \to 0} \frac{\Delta H}{\Delta t} = \lim_{\Delta t \to 0} (I \omega) \frac{\Delta \theta}{\Delta t} = I \omega \frac{d \theta}{dt}
\]  

(23)

Therefore,

\[
\frac{dH}{dt} = I \omega \omega_p
\]  

(24)

and

\[
T = I \omega \omega_p
\]  

(25)

3. Theoretical Development of the Model
Figure 6. Gyroscopic action in a system (from Mabie and Reinholtz [11])
where $T$ is the torque applied, which is a couple called the gyroscopic couple.

There are important gyroscopic actions set up in the frame due to the high speed rotation of the electric motor armature, or the rotor, with respect to the frame, and the rotation of the frame about the center of mass of the assembly. Figure 7 shows the two important gyroscopic moments due to the armature rotation, which depend upon the angular speed of the crank. Using Eq. 25, these gyroscopic moments are calculated and can be written as

$$M_x = l_x \dot{\theta}_y$$  \hspace{1cm} (26)
$$M_y = -l_x \dot{\theta}_x$$  \hspace{1cm} (27)

The system gyroscopic matrix can be written as

$$[C_g] = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & l_x \dot{\theta} & 0 & 0 \\
0 & 0 & 0 & -l_x \dot{\theta} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 
\end{bmatrix}$$ \hspace{1cm} (28)

Also, the applied torque can be written as

$$T = C_g \dot{X}_g = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & \dot{x} \\
0 & 0 & 0 & 0 & 0 & 0 & \dot{y} \\
0 & 0 & 0 & 0 & 0 & 0 & \dot{z} \\
0 & 0 & 0 & l_x \dot{\theta} & 0 & 0 & \dot{\theta}_x \\
0 & 0 & 0 & -l_x \dot{\theta} & 0 & 0 & \dot{\theta}_y \\
0 & 0 & 0 & 0 & 0 & 0 & \dot{\theta}_z
\end{bmatrix}$$

3. Theoretical Development of the Model
Figure 7. Gyroscopic moments due to rotation of the rotor
3.2.4 Total System Stiffness Matrix

3.2.4.1 Spring Numbering Convention

The position of the attachment points of the three springs with respect to the three-dimensional coordinate system at the center of mass of the assembly are shown in Fig. 5. The top mounting spring is numbered as Spring #1. The left side mounting spring as viewed from the positive Y-axis is numbered as Spring #2. The right side mounting spring as viewed from the positive Y-axis is numbered as Spring #3. The shockloop which connects the muffler port to the gas outlet is numbered as Spring #4.

3.2.4.2 The Coiled Spring

The Bristol H-25A compressor assembly is supported by two helical coil springs, one on each side, and a third spring, which engages the top of the motor cap. As a spring is a flexible mechanical link between two bodies in a mechanical system, it is a general element in the stiffness matrix of the system. When such a spring is deformed, the spring exerts a force that is proportional to the displacement, but is opposite in direction. Mathematically, this can be expressed as

\[ F = kx \]  

(29)

where \( k \) is the spring constant for the given spring.
3.2.4.3 Derivation of Element Stiffness Matrix

Figure 8 shows a six-degree-of-freedom spring in three-dimensional space consisting of three translational springs and three rotational springs in and about the X, Y, and Z directions, respectively. This element would be free to translate in three different directions, namely, in the X, Y, and Z directions. The element would also be free to rotate about three different coordinate directions, namely, about the X, Y, and Z directions. Hence this element would have six degrees of freedom.

As discussed by Kelly [9], the method of stiffness influence coefficients is used to determine the element's stiffness matrix. The procedure using the influence coefficient method can be described as follows:

A unit displacement is assigned, maintaining all other displacements and rotations in their static-equilibrium position. The system of forces required to maintain this as an equilibrium position is then calculated. The set of forces applied at the locations whose displacements define the generalized coordinates yields the elements in the respective columns of the stiffness matrix. If a rotation is applied instead of a displacement, maintaining all other rotations and displacements in their static-equilibrium position, then the set of moments required to maintain this as an equilibrium position yields the elements in the respective columns of the stiffness matrix.

Hence, applying a unit displacement in the X-direction, the set of forces required to maintain the system in equilibrium are
Figure 8. A six-degree-of-freedom spring in 3-D space
\[ k_{11} = K_{xx} \]
\[ k_{21} = K_{xy} \]
\[ k_{31} = K_{xz} \] (30)

The set of moments required to keep the system in equilibrium are

\[ k_{41} = K_{x\theta_x} - K_{xy} R_x + K_{xz} R_y \]
\[ k_{51} = K_{x\theta_y} + K_{xx} R_z - K_{xz} R_x \]
\[ k_{61} = K_{x\theta_z} - K_{xz} R_y + K_{xy} R_z \] (31)

Applying a unit displacement in the Y-direction,

\[ k_{12} = K_{xy} \]
\[ k_{22} = K_{yy} \]
\[ k_{32} = K_{yz} \]
\[ k_{42} = K_{y\theta_x} - K_{yx} R_z + K_{yz} R_y \]
\[ k_{52} = K_{y\theta_y} + K_{xy} R_x - K_{yz} R_x \]
\[ k_{62} = K_{y\theta_z} - K_{yx} R_y + K_{yy} R_z \] (32)

Applying a unit displacement in the Z-direction,

\[ k_{13} = K_{xz} \]
\[ k_{23} = K_{yz} \]
\[ k_{33} = K_{zz} \]
\[ k_{43} = K_{z\theta_x} - K_{zx} R_x + K_{zz} R_y \]
\[ k_{53} = K_{z\theta_y} + K_{xz} R_z - K_{zz} R_x \]
\[ k_{63} = K_{z\theta_z} - K_{zx} R_y + K_{yz} R_z \] (33)

3. Theoretical Development of the Model
Similarly, applying a unit rotation about the X-direction

\[
\begin{align*}
  k_{14} &= K_{x0} + K_{xy} R_y + K_{xx} R_x \\
  k_{24} &= K_{y0} - K_{yx} R_y + K_{yy} R_x \\
  k_{34} &= K_{z0} - K_{zy} R_y + K_{zz} R_x \\
  k_{44} &= K_{0,0} - 2K_{xy} R_x R_y + K_{yy} R_y^2 + K_{xx} R_x^2 \\
  k_{54} &= K_{0,0} - K_{xy} R_y^2 + K_{yy} R_x R_y + K_{xx} R_x R_y - K_{zz} R_x R_y \\
  k_{64} &= K_{0,0} - K_{xy} R_y^2 + K_{yy} R_x R_y + K_{yy} R_x R_y - K_{xx} R_x R_y 
\end{align*}
\]  

(34)

Applying a unit rotation about the Y-direction,

\[
\begin{align*}
  k_{15} &= K_{x0} + K_{xy} R_y - K_{xx} R_x \\
  k_{25} &= K_{y0} - K_{yx} R_y + K_{yy} R_x \\
  k_{35} &= K_{z0} - K_{zy} R_y + K_{zz} R_x \\
  k_{45} &= K_{0,0} - K_{xy} R_y^2 + K_{yy} R_x R_y + K_{xx} R_x R_y - K_{zz} R_x R_y \\
  k_{55} &= K_{0,0} - 2K_{xy} R_x R_y + K_{yy} R_y^2 + K_{xx} R_x^2 \\
  k_{65} &= K_{0,0} - K_{xy} R_y^2 + K_{yy} R_x R_y + K_{yy} R_x R_y - K_{xx} R_x R_y 
\end{align*}
\]  

(35)

Applying a unit rotation about the Z-direction,

\[
\begin{align*}
  k_{16} &= K_{x0} + K_{xy} R_y - K_{xx} R_x \\
  k_{26} &= K_{y0} - K_{yx} R_y + K_{yy} R_x \\
  k_{36} &= K_{z0} - K_{zy} R_y + K_{zz} R_x \\
  k_{46} &= K_{0,0} - K_{xy} R_y^2 + K_{yy} R_x R_y + K_{xx} R_x R_y - K_{zz} R_x R_y \\
  k_{56} &= K_{0,0} - K_{xy} R_y^2 + K_{yy} R_x R_y + K_{yy} R_x R_y - K_{xx} R_x R_y \\
  k_{66} &= K_{0,0} - 2K_{xy} R_x R_y + K_{yy} R_y^2 + K_{xx} R_x^2 
\end{align*}
\]  

(36)

3. Theoretical Development of the Model
The element stiffness matrix can then be written as

$$[K_{elem}] = \begin{bmatrix}
  k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\
  k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\
  k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\
  k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\
  k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\
  k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66}
\end{bmatrix}$$  \hspace{1cm} (37)

where all the elements of the matrix can be obtained from Eqs. 30 through 36. It should also be observed that the element stiffness matrix is symmetric.

3.2.4.4 Computation of Total Assembly Stiffness Matrix

The element stiffness matrix is computed for Spring #1, Spring #2 and Spring #3, i.e., the top mounting spring, and the two side mounting springs, using the procedure described in Sec. 3.2.4.3. The stiffness matrix for the shockloop is obtained by the finite element analysis method, as computed by Ramani [10], and is shown in Appendix H. The total assembly stiffness matrix for the compressor is then obtained by superpositioning these four spring stiffness matrices, as in Eq. 38. It is to be mentioned here that the total assembly stiffness matrix is computed with respect to the center of mass of the assembly about the static equilibrium position.

$$[K_z] = [K_{Spring\#1}] + [K_{Spring\#2}] + [K_{Spring\#3}] + [K_{Shockloop}]$$  \hspace{1cm} (38)

3. Theoretical Development of the Model
3.2.5 Force and Moment Matrix

3.2.5.1 Application of equivalent masses to replace the connecting rod

In the analysis of reciprocating compressors and automotive and aircraft piston engines, dynamically equivalent systems of two masses are most commonly used to replace the mass of connecting rods in these machines. Figure 9 shows a typical connecting rod, whose weight, \( W \), moment of inertia, \( I \), and center of gravity are given. As discussed by Mabie and Reinholtz [11], by arbitrarily locating one of the equivalent masses, \( M_{\text{trans}} \) at the wrist-pin bearing, one of the inertia forces is determined from the acceleration of the piston. The location of the second mass, \( M_{\text{rot}} \), is determined by applying the following three laws of equivalents.

Applying the law of equivalence of mass, which states that the sum of the point masses must be equal to the mass of the link,

\[
M_{\text{rot}} + M_{\text{trans}} = \frac{W}{g}
\]  

(39)

Applying the law of equivalence of mass center, which states that the mass center of the system of the two point masses must be at the mass center of the link,

\[
M_{\text{rot}}l_{\text{rot}} - M_{\text{trans}}l_{\text{trans}} = 0
\]  

(40)

Applying the law of equivalence of moment of inertia, which states that the sum of moments of inertia of the point masses about the center of gravity, must be equal to the moment of inertia

\[
M_{\text{rot}}l_{\text{rot}}^2 + M_{\text{trans}}l_{\text{trans}}^2 = I
\]  

(41)

3. Theoretical Development of the Model
Figure 9. Replacement of connecting rod using equivalent masses
By solving Eq. 39 and Eq. 40, simultaneously, the following equations are obtained:

\[
M_{\text{rot}} = \frac{W}{g} \left( \frac{l_{\text{trans}}}{l_{\text{rot}} + l_{\text{trans}}} \right)
\]

(42)

\[
M_{\text{trans}} = \frac{W}{g} \left( \frac{l_{\text{rot}}}{l_{\text{rot}} + l_{\text{trans}}} \right)
\]

(43)

Substituting these equations in Eq. 41, the following equation is obtained:

\[
l_{\text{rot}} l_{\text{trans}} = l \left( \frac{g}{W} \right)
\]

(44)

Because of the shape of the connecting rod, and the nearness of point C to point A, as shown in Fig. 9, an approximation with little error in the value for the moment of inertia, I, could be made that

\[
l_{\text{rot}} = l_{\text{pin}}
\]

(45)

Hence, the second mass is placed at the crank-pin center, and the inertia force may be determined from the acceleration of the crank pin.

### 3.2.5.2 Force Analysis of a slider crank mechanism using point masses

Figure 10 shows a typical slider crank mechanism of a compressor, whose crankshaft is rotating counter-clockwise, with approximate kinetically equivalent point masses replacing the connecting rod. The ground is numbered as link 1, the crank is numbered as link 2, the connecting rod is numbered as link 3, and the piston is numbered as link 4, in this four-bar linkage. One of the point masses is located at the wrist-pin center, known as the translating mass, and the other is
Figure 10. Force analysis of a typical slider crank mechanism

3. Theoretical Development of the Model
From the force polygon obtained from Fig. 10,

\[ F_{12} = -\frac{F_B}{\cos \phi} \]  \hspace{1cm} (49)

and

\[ F_{14} = -F_B \tan \phi \]  \hspace{1cm} (50)

The output torque can be calculated by using either of the following equations

\[ T = F_{12}d \]  \hspace{1cm} (51)

or

\[ T = F_{14}h \]  \hspace{1cm} (52)

3.2.5.3 Effect of reaction forces at the crank-pin and the wrist-pin

The reactions of the assembly on the frame are among the most important reaction forces in the force analysis of the compressor system. These can be classified as (i) the reaction of the piston on the block or frame, \( F_{41} \), and (ii) the reaction of the crank on the main bearings, \( F_{21} \). These forces contribute their effects at the center of mass of the assembly as the crank rotates and the piston reciprocates simultaneously.

The effect of the reaction force in the X-direction at the center of mass would then be the sum of the reaction forces in the X-direction, contributed by \( F_{41} \) and \( F_{21} \), when considering action within one cylinder.

Mathematically, this can be written as

\[ F_{x,cyl} = (F_{41,x,cyl} + F_{21,x,cyl}) \]  \hspace{1cm} (53)

3. Theoretical Development of the Model
The effect of the reaction force in the Y-direction at the center of mass would similarly be the sum of the reaction forces in the Y-direction, contributed by $F_{41}$, $F_{21}$, for one cylinder.

Mathematically, this can be written as

$$F_{y_{cyl}} = (F_{41y_{cyl}} + F_{21y_{cyl}}) \quad (54)$$

As there is no reaction force acting in the Z-direction, the effect of the force in that direction would be zero. Hence

$$F_{z_{cyl}} = 0 \quad (55)$$

The moments of the reaction forces about the origin of the center of mass coordinate axes are calculated using the cross-product relation:

$$\vec{M}_{cyl} = \vec{R}_{cyl} \times \vec{F}_{cyl} \quad (56)$$

In equation form, these can be written as

$$M_{x_{cyl}} = -R_{x_{spin}} F_{21y_{cyl}} - R_{y_{spin}} F_{41y_{cyl}} \quad (57)$$

$$M_{y_{cyl}} = R_{x_{spin}} F_{21x_{cyl}} + R_{y_{spin}} F_{41x_{cyl}} \quad (58)$$

$$M_{z_{cyl}} = R_{x_{spin}} F_{21y_{cyl}} + R_{y_{spin}} F_{41y_{cyl}} - R_{x_{spin}} F_{21x_{cyl}} - R_{y_{spin}} F_{41x_{cyl}} \quad (59)$$

The reaction forces and moments matrix for each cylinder can then be written as

$$F_{cyl} = [F_{x_{cyl}}, F_{y_{cyl}}, F_{z_{cyl}}, M_{x_{cyl}}, M_{y_{cyl}}, M_{z_{cyl}}]^{T} \quad (60)$$
all counterweights rotating about the crankshaft axis. From Fig. 11, it can be observed that there are two components of the inertia force of the counterweight; (i) the radial component of force and (ii) the tangential component of force. The magnitudes of the radial and tangential forces are given by

\[ F_{RAD} = M_{CW} \rho_{CW} \omega^2 \]  

(63)

and

\[ F_{TAN} = M_{CW} \rho_{CW} \alpha \]  

(64)

In vector form and in the center of mass coordinate axes, the counterweight inertia force is given by

\[ F_{CWX} = F_{TAN} \cos(\theta + \lambda) - F_{RAD} \sin(\theta + \lambda) \]  

(65)

\[ F_{CWT} = F_{RAD} \cos(\theta + \lambda) + F_{TAN} \sin(\theta + \lambda) \]  

(66)

\[ F_{CWZ} = 0 \]  

(67)

The inertia moments of the counterweight inertia forces about the origin of the center of mass coordinate axes are calculated using the cross-product relation:

\[ \overline{M}_{CW} = \overline{R}_{CW} \times \overline{F}_{CW} \]  

(68)

In scalar equation form, these can be written as

\[ M_{CWX} = -R_{CWZ} F_{CWT} \]  

(69)

\[ M_{CWT} = R_{CWZ} F_{CWX} \]  

(70)

\[ M_{CWZ} = R_{CWX} F_{CWT} - R_{CWT} F_{CWX} \]  

(71)

Thus, the inertial forcing and moment matrix for each rotating counterweight can be written as

\[ (F_{CW}) = \begin{bmatrix} F_{CWX} & F_{CWT} & F_{CWZ} & M_{CWX} & M_{CWT} & M_{CWZ} \end{bmatrix}^T \]  

(72)

3. Theoretical Development of the Model
3. Theoretical Development of the Model
The following are the important rotating counterweights for which the inertial forcing and moment matrices are calculated and superposed to give the total counterweight forcing and moment matrix.

i) The equivalent mass of the connecting rod of cylinder #1 (lower cylinder), located at its crank-pin center, and known as the rotating connecting rod mass #1.

ii) The equivalent mass of the connecting rod of cylinder #2 (upper cylinder), located at its crank-pin center, and known as the rotating connecting rod mass #2.

iii) The counterweight attached to the rotating rotor.

iv) The counterweight attached to the crankshaft to supply the necessary balance to the system.

Then the total counterweight forcing and moment matrix can be written as

\[ (F_{gTOTCW}) = (F_{gCW})_{connrod1} + (F_{gCW})_{connrod2} + (F_{gCW})_{rotor} + (F_{gCW})_{cshaft} \]  \hspace{1cm} (73)

3.2.5.6 The Total Shaking Forces and Moments Matrix

The total shaking forces and moments matrix is obtained by superposing (i) the total force and moment matrix due to reaction forces, (ii) the torque effect matrix, and (iii) the total counterweight forcing and moment matrix.

In equation form, this can be written as:

\[ [F_g] = (F_{gREACTION}) + (F_{gTORQUE}) + (F_{gTOTCW}) \]  \hspace{1cm} (74)
3.2.6 Response Analysis of the Compressor System

The equation of motion for the compressor assembly, a six-degree-of-freedom system, with gyroscopic terms, and arbitrary excitation, \( F_g \), can be presented in the matrix form as

\[
M_g(\ddot{X}_g) + C_g(\dot{X}_g) + K_g(X_g) = [F_g]
\]

This is generally a set of six coupled equations. We also know that the solution of the homogenous undamped equation

\[
M_g(\ddot{X}_g) + K_g(X_g) = 0
\]

leads to the eigenvalues and eigenvectors that describe the normal modes, \( X_1, X_2, \ldots, X_6 \) of the system, and the modal matrix, \( P \). As discussed by Thomson [12], when the six normal modes, or eigenvectors, are assembled into a square matrix with each normal mode represented by a column, we call that square matrix the *modal matrix*, \( P \). Thus the modal matrix for this system can be written as

\[
P = \begin{bmatrix}
  \{x_1\} & \{x_1\} & \{x_1\} & \{x_1\} & \{x_1\} & \{x_1\} \\
  \{x_2\} & \{x_2\} & \{x_2\} & \{x_2\} & \{x_2\} & \{x_2\} \\
  \{x_3\} & \{x_3\} & \{x_3\} & \{x_3\} & \{x_3\} & \{x_3\} \\
  \{x_4\} & \{x_4\} & \{x_4\} & \{x_4\} & \{x_4\} & \{x_4\} \\
  \{x_5\} & \{x_5\} & \{x_5\} & \{x_5\} & \{x_5\} & \{x_5\} \\
  \{x_6\}_1 & \{x_6\}_2 & \{x_6\}_3 & \{x_6\}_4 & \{x_6\}_5 & \{x_6\}_6 \\
\end{bmatrix}
\]  

or

3. Theoretical Development of the Model

43
\[ P = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 \end{bmatrix} \] (78)

According to Thomson [12], if we let \( X = PY \) and premultiply Eq. 76 by \( P^T \) throughout, we obtain

\[ P^T [ M_e ] P(\dot{Y}) + P^T [ C_g ] P(\dot{Y}) + P^T [ K_g ] P(Y) = P^T [ F_e ] \] (79)

where the matrices for the mass, gyroscopic, stiffness, and force are called their respective generalized matrices. Equation 79 is generally uncoupled and can be written as

\[ m \ddot{Y} + c \dot{Y} + kY = F \sin \omega t \] (80)

which is the sine function for a forced harmonic motion, whose solution, as discussed by Thomson [12] is given by

\[ Y_i = \frac{F_i / k_{ij}}{\sqrt{1 - \left( \frac{\omega}{\omega_n} \right)^2 + \left( \frac{2 \xi \omega}{\omega_n} \right)^2}} \]

and when \([C_g]\) is proportional to \([M_e]\) or \([K_g]\), will have the form

\[ Y_i = \frac{F_i}{\sqrt{k_{ii} - m_{ii} \omega^2 + \left( c_{ii} w \right)^2}} \] (81)

and the response in the original coordinates is given by

\[ X = PY \] (82)

3. Theoretical Development of the Model
4.0 Evaluation of Parameters

4.1 Center of mass of the assembly

The determination of the center of mass of the assembly is a very important step, as it plays a major role in the force analysis, as well as the response analysis of the system. All the forces and moments are calculated with respect to a coordinate system, the origin of which is defined to lie at the center of mass of the compressor assembly. As the compressor could not be supported easily on a perfectly horizontal plane, the inclined plane method is used to calculate the center of mass of the assembly, the method being explained as follows:

Figure 12 shows a body being inclined about its axis, with respect to which the center of mass is to be determined. If point \( A \) is the point of loading on a weighing scale, and point \( B \) is the point of rest on a flat surface, knowing the angle of inclination, \( \theta \), and the weight scale reading \( W_f \) and the distance between supports, \( l \), we can calculate the distance, \( d \), from point \( B \) to the center of mass. This can be done by taking moments about the point \( B \), as we know that the weight of the body, i.e., the compressor, \( W_{\text{comp}} \), acts downwards, at its center of mass. Taking moments about point \( B \), we have

\[
(W_f)l \cos \theta = W_{\text{comp}} d
\]

or

\[
d = \frac{W_f l \cos \theta}{W_{\text{comp}}}
\]

\( (83) \)
Figure 12. Determination of the center of mass
For simplicity in calculations, the origin for calculating the center of mass of the compressor assembly was chosen at the center of the bottom cylinder, i.e., cylinder #1, as shown in Fig. 3, with the X, Y, and Z directions being the same as the coordinate system used throughout this research, which is discussed in Sec. 3.2.1. The weight of the compressor assembly, $W_{\text{comp}}$, consisted of parts whose mass values are shown in Appendix A. The calculations for the center of mass of the assembly follow.

**Calculations for X-axis:**

Weight of the compressor assembly, $W_{\text{comp}} = 79.6$ lb

Weighing scale reading, $W_f = 10.6$ lb

Length between supports, $l = 4.84$ in

Angle of inclination, $\theta = 10.4^\circ$

Using Eq. 84, we have

$$d = \frac{10.6 \times 4.84 \times \cos(10.4^\circ)}{79.6}$$

or

$$d = 0.632 \text{ in}$$

Therefore, the distance of the center of mass along AB from point B or

$$OB = \frac{d}{\cos\theta}$$

(85)

Using Eq. 85, we have

$$OB = \frac{0.632}{\cos(10.4^\circ)}$$
or \( OB = 0.643 \text{ in} \)

Hence, the X coordinate of the center of mass (as per the cylinder coordinate axes)

\[ x_o = -(0.632 - 0.643) = 0.010 \text{ in} \]

or \[ x_o = 0.010 \text{ in} \] (86)

**Calculations for Y-axis:**

Weight of the compressor assembly, \( W_{\text{comp}} = 79.6 \text{ lb} \)

Weighing scale reading, \( W_f = 6.25 \text{ lb} \)

Length between supports, \( l = 5.39 \text{ in} \)

Angle of inclination, \( \theta = 78.2^\circ \)

Using Eq. 84, we have

\[
d = \frac{6.25 \times 5.39 \times \cos(78.2^\circ)}{79.6}
\]

or \( d = 0.146 \text{ in} \)

Therefore, the distance of the center of mass along \( AB \) from point \( B \), using Eq. 85,

\[
OB = \frac{0.146}{\cos(78.2^\circ)}
\]

or \( OB = 0.423 \text{ in} \)

Hence, the Y coordinate of the center of mass (as per the cylinder coordinate axes)

\[ y_o = -(3.93 - 0.42) = -3.51 \text{ in} \]

or \[ y_o = -3.51 \text{ in} \] (87)

4. Evaluation of Parameters
Calculations for Z-axis:

Weight of the compressor assembly, $W_{\text{comp}} = 79.6 \text{ lb}$

Weighing scale reading, $W_f = 34.8 \text{ lb}$

Length between supports, $l = 8.27 \text{ in}$

Angle of inclination, $\theta = 22.2^\circ$

Using Eq. 84, we have

$$d = \frac{34.8 \times 8.27 \times \cos(22.2^\circ)}{79.6}$$

or

$$d = 3.35 \text{ in}$$

Therefore, the distance of the center of mass along AB from point B, using Eq. 85,

$$OB = \frac{3.35}{\cos(22.2^\circ)}$$

or

$$OB = 3.61 \text{ in}$$

Hence, the Z coordinate of the center of mass (as per the cylinder coordinate axes)

$$z_o = (2.10 + 3.61) = 5.71 \text{ in}$$

or

$$z_o = 5.71 \text{ in}$$   \hspace{1cm} (88)

Hence the center of mass of the compressor assembly, with respect to Fig. 3, can be written as

$$(x_o, y_o, z_o) = (0.01, -3.51, 5.71)$$   \hspace{1cm} (89)

4.2 Moments of Inertia for the Assembly
As the mass moment of inertia for the compressor assembly was not available, and to accommodate all the parts that were attached in the actual assembly, the mass moments of inertia about the X, Y, and Z coordinate directions at the center of mass of the assembly, had to be determined.

The following experimental method, called the torsional pendulum method, discussed by Holman [13] was adopted to accurately determine the mass moments of inertia for the compressor assembly.

As shown in Fig. 13, a piece of aluminum sheet metal was cut out into a circular shape. The dimensions of the plate are as follows:

\[ D = \text{Diameter of the circular plate, 24.0 in} \]

\[ W = \text{Weight of the plate, 17.3 lbf} \]

The effects of the circular hole in the center and the four holes at diametrically opposite ends of the circular sheet (shown in Fig. 13) were neglected as their combined mass amounted to less than 1% of the circular plate's weight. The circular plate was then hung from supports with assumed massless steel rods. In order to hold the circular plate with these steel rods, each about 0.092 in. in diameter and 24 in. in length, four spherical bearings were selected and suitable plugs were designed to pass through the bearings, so that the bearing could rest on the conical portion of the hole in the circular plate, and thus help the plate to oscillate easily. The dimensions of the plug designed to hold the plate is shown in Fig. 14.
Figure 13. The circular plate for determining inertia

MATERIAL: ALUMINIUM

ALL DIMENSIONS ARE IN INCHES

4. Evaluation of Parameters
Figure 14. The plug for the spherical bearing to hold the circular plate
The equation of motion for the angular oscillation of the circular plate about a normal through the plate centroid axis can be written as follows:

Applying Newton’s second law, as discussed by Greenwood [8], for the free-body diagram, shown in Fig. 15,

\[ I \ddot{\theta} = \Sigma M \]  \hspace{1cm} (90)

or

\[ I \ddot{\theta} = -\frac{W r \theta}{4L} (r) - \frac{W r \theta}{4L} (r) - \frac{W r \theta}{4L} (r) - \frac{W r \theta}{4L} (r) \]  \hspace{1cm} (91)

or

\[ I \ddot{\theta} = -\frac{W r^2 \theta}{L} \]  \hspace{1cm} (92)

Therefore

\[ I \ddot{\theta} + \frac{W r^2 \theta}{L} = 0 \]  \hspace{1cm} (93)

From the above Eq. 93,

\[ \omega_n^2 = \frac{1}{2 \pi} \sqrt{\frac{W r^2}{L I}} \]

Re-writing the above equation, we have

\[ I = \frac{W r^2}{4\pi^2 L \omega_n^2} \]  \hspace{1cm} (94)

4. Evaluation of Parameters
Figure 15. The oscillating circular plate used as the torsional pendulum
Substituting the known values:

\[ I = \frac{17.3(12)^2}{4\pi^2(24)\omega_n^2} \]  

(95)

From the above equation, the mass moment of inertia of the circular plate can be determined by substituting for \( \omega_n \), the natural frequency of oscillation of the circular plate. The plate was given a small angular twist from its equilibrium position and the time period for 25 oscillations of the circular plate about the vertical axis was measured with the help of a chronometer.

\[ \omega_n = \frac{1}{\tau} \]  

(96)

where \( \tau \) is the period of oscillation, and is given by the ratio of the time taken to the number of oscillations. Mathematically

\[ \tau = \frac{29.2}{25} \]

or \[ \tau = 1.17 \text{ s} \]

and \[ \omega_n = 0.86 \text{ s}^{-1} \]

Substituting these in Eq. 95, we have

\[ I_{\text{plate}} = 3.59 \text{ lbf-in-sec}^2 \]

or

\[ I_{\text{plate}} = 1384 \text{ lbf-in}^2 \]  

(97)

4. Evaluation of Parameters
After determining the moment of inertia of the plate alone, the compressor had to be held on the plate such that the coordinate axis about which the mass moment of inertia of the assembly that had to be determined was vertical. In order to achieve this, fixtures made out of wood, were designed to hold the compressor vertical in the X, Y, and Z directions.

**Mass Moment of Inertia about X, I_{xx}**

The compressor assembly was placed on the circular plate, such that the X-axis of the center of mass coordinate system was vertical, as shown in Fig. 16. The assembly was firmly placed on the circular plate with the center of mass of the assembly in line with the center of mass of the circular plate. The equation of motion for this system can be obtained by modifying Eq. 93 as follows:

\[
(I_{\text{plate}} + I_{\text{comp}}) \ddot{\theta} + \frac{(W_{\text{plate}} + W_{\text{comp}}) r^2}{L} \theta = 0
\]

(98)

From the above Eq. 98

\[
\omega_n = \frac{1}{2\pi} \sqrt{\frac{(W_{\text{plate}} + W_{\text{comp}}) r^2}{L(I_{\text{plate}} + I_{\text{comp}})}}
\]

(99)

Re-writing Eq. 99, we have

\[
I_{\text{comp}} = \frac{(W_{\text{plate}} + W_{\text{comp}}) r^2}{4\pi^2 L \omega_n^2} - I_{\text{plate}}
\]

(100)

or from Eq. 94, Eq. 96 and Eq. 100, we get

\[
I_{\text{comp}} = \frac{W_{\text{comp}} r^2 \tau_{\text{comp}}^2}{4\pi^2 L} + \frac{W_{\text{plate}} r^2 (\tau_{\text{comp}}^2 - \tau_{\text{plate}}^2)}{4\pi^2 L}
\]

(101)

4. Evaluation of Parameters
Figure 16. Determination of the moment of inertia of the assembly about X direction

4. Evaluation of Parameters
For the case where the X-direction is vertical, the values were

Weight of the inner assembly, $W_{\text{comp}} = 79.6 \text{ lb}$

Weight of the circular plate + fixture, $W_{\text{plate}} = 17.3 + 4.5 = 21.8 \text{ lb}$

Length of the cord, $l = 24 \text{ in}$

Acceleration due to gravity, $g = 386 \text{ in/s}^2$

Radius of circular plate, $r = 12 \text{ in}$.

The plate with the fixture was first given a small angle of twist from its equilibrium position and the time period, $\tau_{\text{plate}}$, for 5 oscillations of the circular plate and fixture about the vertical axis was measured using a chronometer.

\[
\tau_{\text{plate}} = \frac{5.8}{5}
\]

or

\[
\tau_{\text{plate}} = 1.16 \text{ s}
\]

The compressor was then placed on the fixture and plate, and the procedure was repeated to calculate the time period, $\tau_{\text{comp}}$, for 15 oscillations of the circular plate, fixture and compressor.

\[
\tau_{\text{comp}} = \frac{10.7}{15}
\]

\[
\tau_{\text{comp}} = 0.713 \text{ s}
\]

Substituting all values in Eq. 101, we have

\[
I_{xx} = 1305 \text{ lbf} - \text{in}^2
\]  

(102)

**Mass Moment of Inertia about Y, $I_{yy}$**

The compressor assembly was placed on the circular plate, such that the Y-axis of the center of mass coordinate system was vertical, as shown in Fig. 17. The assembly was firmly placed on the

4. Evaluation of Parameters
circular plate with the center of mass of the assembly in line with the center of mass of the circular plate.

For the case where the Y-direction is vertical, the values were

Weight of the inner assembly, \( W_{\text{comp}} = 79.6 \text{ lb} \)

Weight of the circular plate + fixture, \( W_{\text{plate}} = 17.3 + 4.9 = 22.2 \text{ lb} \)

Length of the cord, \( l = 24.0 \text{ in} \)

Acceleration due to gravity, \( g = 386 \text{ in/s}^2 \)

Radius of circular plate, \( r = 12.0 \text{ in} \).

The plate with the fixture, was first given a small angle of twist from its equilibrium position and the time period, \( \tau_{\text{plate}} \), for 3 oscillations of the circular plate and fixture about the vertical axis was measured using a chronometer.

\[
\tau_{\text{plate}} = \frac{3.54}{3}
\]

or

\[
\tau_{\text{plate}} = 1.18 \text{ s}
\]

The compressor was then placed on the fixture and plate, and the procedure was repeated to calculate the time period, \( \tau_{\text{comp}} \), for 20 oscillations of the circular plate, fixture and compressor.

\[
\tau_{\text{comp}} = \frac{14.1}{20}
\]

\[
\tau_{\text{comp}} = 0.70 \text{ s}
\]

Substituting all values in Eq. 101, we have

\[
I_{yy} = 1145 \text{ lb} \cdot \text{in}^2 \quad \text{(103)}
\]
Figure 17. Determination of the moment of inertia of the assembly about Y direction

4. Evaluation of Parameters
**Mass Moment of Inertia about Z, \( I_{zz} \)**

The compressor assembly was placed on the circular plate, such that the Z-axis of the center of mass coordinate system was vertical, as shown in Fig. 18. The assembly was firmly placed on the circular plate with the center of mass of the assembly in line with the center of mass of the circular plate.

For the case where the Z-direction is vertical, the values were

- Weight of the inner assembly, \( W_{\text{comp}} = 79.6 \text{ lb} \)
- Weight of the circular plate + fixture, \( W_{\text{plate}} = 17.3 + 3.5 = 20.8 \text{ lb} \)
- Length of the cord, \( l = 24.0 \text{ in} \)
- Acceleration due to gravity, \( g = 386 \text{ in/s}^2 \)
- Radius of circular plate, \( r = 12.0 \text{ in} \)

The plate with the fixture, was first given a small angle of twist from its equilibrium position and the time period, \( \tau_{\text{plate}} \), for 6 oscillations of the circular plate and fixture about the vertical axis was measured using a tachometer.

\[
\tau_{\text{plate}} = \frac{7.38}{6}
\]

or

\[
\tau_{\text{plate}} = 1.23 \text{ s}
\]

The compressor was then placed on the fixture and plate, and the procedure was repeated to calculate the time period, \( \tau_{\text{comp}} \), for 16 oscillations of the circular plate, fixture and compressor.

\[
\tau_{\text{comp}} = \frac{10.2}{16}
\]

\[
\tau_{\text{comp}} = 0.63 \text{ s}
\]

Substituting all values in Eq. 101, we have

\[
I_{zz} = 561 \text{ lbf} - \text{in}^2
\]  
(104)

4. Evaluation of Parameters
Figure 18. Determination of the moment of inertia of the assembly about Z direction
4.2.1 Effect of damping on oscillations

In order to obtain accuracy in the calculation of mass moments of inertia of the compressor assembly, a study was made on the effect of damping in the system on the oscillations. As discussed by Thomson [12], a convenient way to determine the amount of damping present in a system is to measure the rate of decay of free oscillations. This could be done by calculating the logarithmic decrement, which is defined as the natural logarithm of the ratio of any two successive amplitudes.

Mathematically, the logarithmic decrement for two successive amplitudes \( x_1 \) and \( x_2 \),

\[
\delta = \ln \left( \frac{x_1}{x_2} \right)
\]

or

\[
\delta = \ln (\zeta \omega_n \tau_d) = \zeta \omega_n \tau_d
\]

or

\[
\delta = \zeta \omega_n \frac{\tau_n}{\sqrt{1-\zeta^2}} = 2\pi \frac{\zeta}{\sqrt{1-\zeta^2}}
\]

since

\[
\tau_d = \frac{\tau_n}{\sqrt{1-\zeta^2}}
\]

and

\[
\tau_n = \frac{2\pi}{\omega_n}
\]

For the case when \( \zeta < 0.1 \), Eq. 107 becomes

\[
\delta = 2\pi \zeta
\]

Rewriting Eq. 110, the coefficient of damping

\[
\zeta = \frac{\delta}{2\pi}
\]
For determining the amount of damping present in the X, Y, and Z directions, a scale was mounted on the periphery of the circular plate, with a pointer to measure the amplitudes of oscillation, as the compressor, with the fixture, oscillated, and the period of oscillation decreased with each successive oscillation. The amplitudes were measured for 5 successive oscillations, and the logarithmic decrement between two successive amplitudes, and hence the damping ratio for that case was repeatedly calculated using Eq. 107 and Eq. 111. The average for the damping ratios in the three coordinate directions were obtained as follows:

\[
\begin{align*}
\zeta_x &= 0.012 \\
\zeta_y &= 0.013 \\
\zeta_z &= 0.017
\end{align*}
\]

For the X-direction:

Using Eq. 108, to calculate \( \tau_n \), we have

\[
\tau_n = \tau_d \sqrt{1 - \frac{\zeta_x^2}{\sigma_x^2}}
\]

where

\[
\tau_d = \tau
\]

Substituting \( \tau_d = \tau = 0.713 \text{ s} \) and \( \zeta_x = 0.012 \) in Eq. 112, we have

\[
\tau_n = 0.713 \text{ s}
\]

Substituting all values in Eq. 101 and \( \tau = \tau_n \), we have

\[
I_{xx} = 1300 \text{ lbf-in}^2
\]

For the Y-direction:

Using Eq. 112, to calculate \( \tau_n \), we have

Substituting \( \tau_d = \tau = 0.704 \text{ s} \) and \( \zeta_y = 0.013 \) in Eq. 112, we have

\[
\tau_n = 0.704 \text{ s}
\]
Substituting all values in Eq. 101 and \( \tau = \tau_n \), we have

\[
I_{yy} = 1144 \text{ lbf-in}^2
\]  \hspace{1cm} (115)

**For the Z-direction:**

Using Eq. 112, to calculate \( \tau_n \), we have

Substituting \( \tau_d = \tau = 0.64 \) s and \( \zeta = .017 \) in Eq. 112, we have

\[
\tau_n = 0.64 \text{ s}
\]

Substituting all values in Eq. 101 and \( \tau = \tau_n \), we have

\[
I_{zz} = 560 \text{ lbf-in}^2
\]  \hspace{1cm} (116)

**4.3 Determination of Part Weight Values**

The various parts of the compressor were weighed on a digital electronic scale, which had an accuracy of 0.01 lb in its readings. The various results are listed in Appendix A. It should be mentioned here that the cylinder head was weighed with its gasket in place, and that the stator of the H-25A's motor consisted of its body and the windings.

**4.4 Calculation of System Mass and Inertial Matrix**

Reviewing the results obtained,

Weight of the assembly,

\[
W_{\text{comp}} = 79.6 \text{ lb}
\]

Therefore

Mass of the assembly,

\[
M_f = \frac{W_{\text{comp}}}{g}
\]  \hspace{1cm} (117)
Substituting the value in Eq. 117, we get

\[ M_f = 0.206 \text{ lbf-in-s}^2 \]  

(118)

Also the values for \( I_{xx} \), \( I_{yy} \), and \( I_{zz} \) in terms of (lbf-in-s\(^2\)) units, can be written as

\[ I_{xx} = 3.38 \text{ lbf-in-s}^2 \]  

(119)

\[ I_{yy} = 2.97 \text{ lbf-in-s}^2 \]  

(120)

\[ I_{zz} = 1.45 \text{ lbf-in-s}^2 \]  

(121)

The product of inertia terms for the compressor assembly were obtained from the solid modeling by Ramani [10], and are as follows

\[ I_{xy} = 0.00 \text{ lbf-in-s}^2 \]  

(122)

\[ I_{yz} = -0.49 \text{ lbf-in-s}^2 \]  

(123)

\[ I_{xz} = 0.00 \text{ lbf-in-s}^2 \]  

(124)

Substituting these values into Eq. 17 for \([M_g]\), as discussed in Sec. 3.2.2, we have

\[
[M_g] = \begin{bmatrix}
0.206 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.206 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.206 & 0 & 0 & 0 \\
0 & 0 & 0 & 3.37 & 0 & 0 \\
0 & 0 & 0 & 0 & 2.97 & 0.49 \\
0 & 0 & 0 & 0 & 0.49 & 1.45
\end{bmatrix}
\]  

(125)
4.5 Center of mass of the rotor

The center of mass in the axial direction was calculated using the method discussed below. As shown in Fig. 19a, one end of a body is suspended on a weighing scale, and the other end is suspended on a knife edge, such that the whole system is horizontal. If \( L \) is the distance between the supports, \( W_s \) is the weigh scale reading, and \( W \) is the weight of the body, which acts downwards at its center of mass located at point \( O \), we can calculate the distance of the center of mass from either support by taking moments about point B, as shown in Fig. 19b.

Mathematically, this can be written as

\[
W_s L = W y = W(L - x)
\]

or

\[
x = \frac{(W - W_s)L}{W}
\]  

(126)  

(127)

**Y-direction:**

Weight of the rotor, \( W = 9.00 \text{ lb} \)

Weighing scale reading, \( W_s = 3.83 \text{ lb} \)

Distance between supports, \( L = 4.50 \text{ in} \)

Using Eq. 127,

\[
x = 2.59 \text{ in.}
\]

which implies that

\[
y_o = -.089 \text{ in.}
\]

(128)

where \( y_o \) is the distance of the center of mass of the rotor in the Y-direction, as shown in Fig. 20.
Figure 19. Determination of the center of mass in the axial direction
Figure 20. A solid model of the H-25A rotor
For the calculation of the center of mass in the radial direction, the following theory was used.

The portion of the body having a circular cross section is supported on two circular rollers as shown in Fig. 21a. We know that the weight of the body always acts downwards at its center of mass. If \( e \) is the distance of the center of mass from the geometric center of the circular portion, and if an imbalance is provided to the body by attaching a small particle, whose weight is \( W_u \), and whose center of mass lies at a distance \( e_u \) from the center of the circular portion, then knowing the angle of displacement, \( \theta \), as shown in Fig. 21b, through which the body rotates, we can calculate the center of mass of the original body as follows

Taking moments about point \( O \), as shown in Fig. 21c, we have

\[
W_u e_u \cos \theta = W e \sin \theta \tag{129}
\]

or

\[
e = \frac{W_u e_u}{W \tan \theta} \tag{130}
\]

**X-direction:**

Weight of the rotor, \( W = 9.00 \text{ lb} \)

Weight of the imbalance, \( W_u = 0.047 \text{ lb} \)

Distance of center of mass of imbalance from origin, \( e_u = 1.50 \text{ in} \)

Angle of rotation, \( \theta = 223^\circ \)

Using Eq. 130, we have

\[e = 0.008 \text{ in.}\]

or

\[x_e = 0.008 \text{ in.}\tag{131}\]

4. Evaluation of Parameters
Figure 21. Determination of the center of mass in the radial direction
Z-direction:

Weight of the rotor, \( W = 9.00 \text{ lb} \)

Weight of the imbalance, \( W_u = 0.047 \text{ lb} \)

Distance of center of mass of imbalance from origin, \( e_u = 1.50 \text{ in} \)

Angle of rotation, \( \theta = 313^\circ \)

Using Eq. 130, we have,

\[
e = -0.007 \text{ in.}
\]

or

\[
z_o = -0.007 \text{ in.} \tag{132}
\]

Hence the center of mass of the rotor is as follows

\[
(x_o, y_o, z_o) = (0.008, -0.089, -0.007) \tag{133}
\]

with respect to the rotor shown in Fig. 20.

4.6 Moment of inertia of the rotor

The moment of inertia of the rotor are calculated using the torsional pendulum method as discussed in Sec. 4.2. The mass of the rotor being less, as compared to the mass of the assembly, a small triangular plate was chosen, whose dimensions are shown in Fig. 22. The moment of inertia of the triangular plate can be calculated using the relation

\[
I = \frac{Wa^2}{4\pi^2 L\omega_n^2} \tag{134}
\]

where

\( W = \text{Weight of the plate} = 0.234 \text{ lb} \)

4. Evaluation of Parameters
\( a = \) Distance of centroid from any vertex = 6.00 in

\( L = \) Length of the cord to suspend = 13.7 in

\( \omega = \) Frequency of oscillation = 0.853 rad/s

Substituting all these values, we have

\[ I = 0.002 \text{ lbf-in-sec}^2 \quad (135) \]

For the rotor, the same equations were used, as developed in Sec. 4.2.

Modifying Eq. 101 for the rotor, we have

\[ I_{rotor} = \frac{W_{rotor}a^2 \tau_{rotor}^2}{4\pi^2 L} + \frac{W_{plate}a^2}{4\pi^2 L} (\tau_{comp}^2 - \tau_{rotor}^2) \quad (136) \]

**Mass Moment of Inertia about Y, \text{I}_{yy}**

The rotor was placed on the triangular plate such that the Y-axis of the center of mass coordinate system was vertical. The rotor was firmly placed on the triangular plate with the center of mass of the assembly in line with the center of mass of the triangular plate.

For the case where the Y-direction is vertical, the values were

Weight of the rotor, \( W_{rotor} = 9.00 \text{ lb} \)

Weight of the triangular plate, \( W_{plate} = 0.234 \text{ lb} \)

Length of the cord, \( L = 13.7 \text{ in} \)

Acceleration due to gravity, \( g \equiv 386 \text{ in/s}^2 \)

Distance of centroid from the vertex of the plate, \( a = 6.00 \text{ in} \).

The plate with the fixture, was first given a small angle of twist from its equilibrium position and the time period, \( \tau_{plate} \), for 20 oscillations of the triangular plate and fixture about the vertical axis was measured using a chronometer.

4. **Evaluation of Parameters**
Figure 22. The triangular plate for holding the rotor
Mathematically,
\[ \tau_{\text{plate}} = \frac{23.4}{20} \]
or
\[ \tau_{\text{plate}} = 1.17 \text{ s} \]

The rotor was then placed on the plate, and the procedure was repeated to calculate the time period, \( \tau_{\text{rotor}} \), for 20 oscillations of the triangular plate and rotor.

\[ \tau_{\text{rotor}} = \frac{81.1}{20} \]

\[ \tau_{\text{rotor}} = 4.06 \text{ s} \]

Substituting all values in Eq. 136, we have
\[ I_{yy} = 10.1 \text{ lbf} \cdot \text{in}^2 \quad (137) \]

4.7 Computation of the System Gyroscopic Matrix

As shown in Fig. 20, the Y-axis is considered to be the vertical axis for the rotor, in the calculations for the center of mass and the moment of inertia, and since the Z-axis is the vertical axis for all calculations pertaining to the frame of the assembly, we have to keep track of coordinate transformations when calculating the System Gyroscopic Matrix.

Hence, the moment of inertia of the crankshaft about its vertical axis, with respect to the global coordinate system is

\[ I_{zz} = \frac{10.1}{386} \]

or
\[ I_{zz} = 0.026 \text{ lbf-in-sec}^2 \]

4. Evaluation of Parameters
As the crankshaft and the rotor rotate at 3450 rpm, we know that the constant angular velocity of the system is given by

\[
\dot{\theta} = \omega = 361 \text{ rad/s}
\]  

(138)

Using Eq. 26 and Eq. 27, as discussed in Sec. 3.2.3, to calculate the moments that form the Gyroscopic Matrix, which are rewritten as

\[M_x = I_\alpha \dot{\theta} \dot{\theta}
\]

and

\[M_y = -I_\alpha \dot{\theta} \dot{\theta}
\]

Using Eq. 28, as discussed in Sec. 3.2.3, we can calculate the System Gyroscopic Matrix, which can be written as

\[
[C_s] = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 9.48 & 0 \\
0 & 0 & 0 & -9.48 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \text{ lbf-in-sec/rad}
\]  

(139)
4.8 Determination of coordinates of spring attachment points

The total assembly stiffness matrix, is the sum of the stiffness matrices for all four springs under consideration, namely i) the top spring, ii) the two side springs and iii) the shockloop.

In order to determine the individual spring stiffness matrices, the position of the attachment point of each spring to the frame of the compressor, has to be determined. These coordinates are determined in three-dimensional space, with respect to the global Cartesian coordinate system whose origin lies at the center of mass of the compressor assembly. This could be done by collecting information and data obtained in the drawings for the compressor assembly, and the results were as follows.

\[ (r_{1x}, r_{1y}, r_{1z}) = (-0.010, -0.925, 7.43) \]  \hspace{2cm} (140)

\[ (r_{2x}, r_{2y}, r_{2z}) = (-3.20, -.269, -2.48) \]  \hspace{2cm} (141)

\[ (r_{3x}, r_{3y}, r_{3z}) = (3.18, -.269, -2.48) \]  \hspace{2cm} (142)

\[ (r_{4x}, r_{4y}, r_{4z}) = (1.11, 4.14, 2.48) \]  \hspace{2cm} (143)

4. Evaluation of Parameters
4.9 Lateral Stiffnesses of Springs

As discussed by Wahl [14], helical springs, when used as vibration isolators, are laterally loaded by a force, $F$, while being compressed by a vertical force, $P$, the only resistance to the lateral deflection being the stiffness of the spring. This is illustrated in Fig. 23, discussed by Wahl [14]. A theoretical analysis by Wahl [14] shows that in such a case, the lateral stiffness $K_{xx}$ or $K_{yy}$, is reduced by the presence of the axial load. For steel springs of round wire with $E = 30 \times 10^6$ psi and $G = 11.5 \times 10^6$ psi, the stiffness, $K_{xx}$ or $K_{yy}$, in the lateral direction is given by

$$K_{xx} = K_{yy} = \frac{F}{\delta_x} = \frac{10^6 d^4}{C_l n D \left(0.204 d^2 + 0.265 D^2\right)} \quad (144)$$

where

$\delta_x$ = Lateral deflection due to force $F$

$l_o$ = Free length

$h_s$ = Compressed length of spring = $l_o - \delta_s$

$\delta_s$ = Vertical deflection due to load $P$

$C_l$ = A factor depending on $\delta_s/l_o$ and $d/D$

$d$ = Diameter of the wire

$D$ = Mean diameter of the spring

Values of $C_l$ may be taken from the chart shown in Fig. 24, as used by Wahl[14], which is a plot of $l_o/D$ vs. $C_l$.

The ratio of axial stiffness, $K_{zz} = P/\delta_s$, to lateral stiffness, $K_{xx}$, for steel springs of round wire with $E = 30 \times 10^6$ psi and $G = 11.5 \times 10^6$ psi, as derived by Wahl [14], is given by

4. Evaluation of Parameters
Figure 23. Spring under combined axial and lateral loading (from Wahl [14])

4. Evaluation of Parameters
Figure 24. Chart for finding factor from $C_f$ (from Wahl [14])
\[
\frac{K_{xx}}{K_{xx}} = 1.44 C_t \left(0.204 \frac{h_s^2}{D^2} + 0.265 \right)
\]  
(145)

*For the Top Spring:*

\(l_o = \text{Free length} = 1.58 - 0.296 = 1.28 \text{ in}\)

\(h_s = \text{Compressed length of spring} = l_o - \delta_{st} = 0.892 \text{ in}\)

\(\delta_{st} = \text{Vertical deflection due to load} P = 0.387 \text{ in}\)

\(D = \text{Mean diameter of the spring} = 1.11 \text{ in}\)

\(C_t = \text{A factor depending on} \delta_{st}/l_o \text{ and } l_o/D = 1.85 \text{ (From Fig. 24)}\)

From Eq. 144, Lateral Stiffness,

\[K_{xx} = K_{yy} = 136 \text{ lb/in}\]  
(146)

From Eq. 145, Axial Stiffness,

\[K_{zz} = 100 \text{ lb/in}\]  
(147)

Ramani [10] calculated the lateral and axial stiffnesses by developing a solid model for the top spring (Ref. Appendix H). They were \(K_{xx} = K_{yy} = 146 \text{ lb/in} \text{ and } K_{zz} = 100 \text{ lb/in}\).

*For the Side Springs:*

\(l_o = \text{Free length} = 2.67 - 1.29 = 1.38 \text{ in}\)

\(h_s = \text{Compressed length of spring} = l_o - \delta_{st} = 1.25 \text{ in}\)

\(\delta_{st} = \text{Vertical deflection due to load} P = 0.131 \text{ in}\)

\(D = \text{Mean diameter of the spring} = 0.808 \text{ in}\)

\(C_t = \text{A factor depending on} \delta_{st}/l_o \text{ and } l_o/D = 1.65 \text{ (From Fig. 24)}\)

From Eq. 144, Lateral Stiffness,

\[K_{xx} = K_{yy} = 562 \text{ lb/in}\]  
(148)

From Eq. 145, Axial Stiffness,

\[K_{zz} = 450 \text{ lb/in}\]  
(149)

4. Evaluation of Parameters
The lateral and axial stiffnesses calculated by Ramani [10] for the side spring (Ref. Appendix H.) were $K_{xx} = K_{yy} = 503$ lb/in and $K_{zz} = 478$ lb/in.

4.10 Moment Stiffnesses of Springs

Figure 25a, from Wahl[14], shows a coiled spring subjected to moment in the plane of the axis of the spring. By considering a quarter coil subject to a moment, $M$, at its end, the moment being represented by a vector and at a cross section at an angle, $\varphi$, the bending moment, $M_b$, will be $M \cos \varphi$, while the twisting moment, $M_t$, will be $M \sin \varphi$. Considering a length $ds = r d\varphi$, as discussed by Wahl[14], and shown in Fig. 25b, the component of the angular twist about the axis of the moment due to $ds$ will be

$$d\theta = \frac{M_c ds \cos \varphi}{EI} + \frac{M_c ds \sin \varphi}{GI_p}$$

(150)

$\theta = \text{angular twist of a single coil of the spring}$

$M = \text{moment in the plane of the spring}$

$n = \text{number of active coils of the spring}$

$d = \text{diameter of the wire}$

As discussed by Wahl [14], the total angular twist, $\theta$, for one complete turn will be four times the integral of this between $\varphi = 0$ and $\varphi = \pi /2$. Thus

$$\theta = 4 \int_0^{\pi/2} \left( \frac{r M_c}{EI} \cos^2 \varphi + \frac{r M_c}{GI_p} \sin^2 \varphi \right) d\varphi$$

(151)

or

$$\theta = \frac{\pi Mr}{EI} \left( 1 + \frac{EI}{GI_p} \right)$$

(152)

4. Evaluation of Parameters
Figure 25. Spring subjected to moment in the plane of its axis (from Wahl[14])
where

\( I = \text{Area moment of inertia} \)

\( I_p = \text{Polar moment of inertia} = 2I \) \text{ (for circular cross section)}

\( EI = \text{Flexural rigidity of the cross section of the wire} \)

\( GI_p = \text{Torsional rigidity of the cross section of the wire} \)

Rewriting Eq. 152 for a wire having a circular cross section, we have

\[
\theta = \frac{\pi Mr}{EI} \left( \frac{2G + E}{2G} \right) \tag{153}
\]

The moment stiffness about the lateral direction can then be written as

\[
K_{\theta \theta} = K_{\phi \phi} = \frac{M}{\theta} = \frac{EI}{\pi r} \left( \frac{2G}{2G + E} \right) \tag{154}
\]

where

\( M = \text{Moment applied at the end of the spring coil} \) \tag{155}

**For the Top Spring:**

Mean diameter of the coil, \( d = 0.148 \) in

Modulus of rigidity, \( G = 11.5 \times 10^6 \) psi

Modulus of elasticity, \( E = 30 \times 10^6 \) psi

Area moment of inertia, \( I = \pi d^4/64 = 0.00 \) in\(^4\)

Mean radius of the spring, \( r = 1.11/2 = 0.555 \) in

From Eq. 154, Moment Stiffness,

\[
K_{\theta \theta} = K_{\phi \phi} = 175 \text{ lb-in/rad} \tag{156}
\]

Ramani[10] calculated the moment stiffnesses by developing a solid model for the top spring. They were \( K_{\theta \theta} = K_{\phi \phi} = 102 \text{ lb-in/rad} \) and \( K_{\theta \theta} = 0.00 \text{ lb-in/rad} \).

4. Evaluation of Parameters
For the Side Springs:

Mean diameter of the coil, \( d = 0.207 \) in

Area moment of inertia, \( I = \pi d^4/64 = 0.00 \) in\(^4\)

Modulus of rigidity, \( G = 11.5 \times 10^6 \) psi

Modulus of elasticity, \( E = 30 \times 10^6 \) psi

Mean radius of the spring, \( r = 0.808/2 = 0.404 \) in

From Eq. 154, Moment Stiffness,

\[
K_{\theta_c\theta_c} = K_{\theta_y\theta_y} = 924 \text{ lb-in/rad} \quad (157)
\]

The moment stiffnesses as calculated by Ramani [10], using a solid model for the side spring were \( K_{\theta_c\theta_c} = K_{\theta_y\theta_y} = 412 \) lb-in/rad and \( K_{\theta_c\theta_c} = 0.00 \) lb-in/rad.

4.11 Total Assembly Stiffness Matrix

As derived in Sec. 3.2.4.3, using Eq. 37, the element stiffness matrix for each spring, \( K_{\text{elem}} \), can be written. The simplified form of \( K_{\text{elem}} \) is shown in Appendix B. Using Appendix B for \( K_{\text{elem}} \), the co-ordinates of the spring attachment points from Sec. 4.8, and the values for the stiffnesses from Sec. 4.10, we can write the element stiffness matrices as follows:

Element Stiffness Matrix for Spring #1

Lateral Stiffness, \( K_{xx} = K_{yy} = 136 \) lb/in

Axial Stiffness, \( K_{xx} = 100 \) lb/in

Moment Stiffness, \( K_{\theta_c\theta_c} = K_{\theta_y\theta_y} = 175 \) lb-in/rad

Spring Attachment point co-ordinates \((x, y, z) = (-0.010, -0.925, 7.43)\)
Therefore, Element Stiffness Matrix for Spring #1 (Top Spring)

\[
K_{\text{Spring1}} = \begin{bmatrix}
136 & 0 & 0 & 0 & 1015 & 126 \\
0 & 136 & 0 & -1015 & 0 & -141 \\
0 & 0 & 100 & -92.5 & 1.03 & 0 \\
0 & -1015 & -92.5 & 7808 & -953 & 10.5 \\
1015 & 0 & 1.03 & -9.53 & 7723 & 939 \\
126 & -1.41 & 0 & 10.5 & 939 & 117
\end{bmatrix}
\]  

(158)

Element Stiffness Matrix for Spring #2

Lateral Stiffness, \(K_{xx} = K_{yy} = 562 \ \text{lb/in}\)

Axial Stiffness, \(K_{zz} = 450 \ \text{lb/in}\)

Moment Stiffness, \(K_{\text{hex}} = K_{\text{hyz}} = 924 \ \text{lb-in/rad}\)

Spring Attachment point co-ordinates \( (R_x, R_y, R_z) = (-3.20, -0.27, -2.48) \)

Therefore, Element Stiffness Matrix for Spring #2 (Side Spring #1)

\[
K_{\text{Spring2}} = \begin{bmatrix}
562 & 0 & 0 & 0 & -1394 & 151 \\
0 & 562 & 0 & 1394 & 0 & -1798 \\
0 & 0 & 450 & -121 & 1439 & 0 \\
0 & 1394 & -121 & 4413 & -387 & -4458 \\
-1394 & 0 & 1439 & -387 & 8983 & -375 \\
151 & -1798 & 0 & -4458 & -375 & 5791
\end{bmatrix}
\]  

(159)

Element Stiffness Matrix for Spring #3

Lateral Stiffness, \(K_{xx} = K_{yy} = 562 \ \text{lb/in}\)

Axial Stiffness, \(K_{zz} = 450 \ \text{lb/in}\)

Moment Stiffness, \(K_{\text{hex}} = K_{\text{hyz}} = 924 \ \text{lb-in/rad}\)

Spring Attachment point co-ordinates \( (R_x, R_y, R_z) = (3.18, -0.27, -2.48) \)

4. Evaluation of Parameters
Therefore, Element Stiffness Matrix for Spring #3 (Side Spring #2)

\[
K_{\text{Spring#3}} = \begin{bmatrix}
562 & 0 & 0 & 0 & -1394 & 151 \\
0 & 562 & 0 & 1394 & 0 & 1787 \\
0 & 0 & 450 & -121 & -1429 & 0 \\
0 & 1394 & -121 & 4413 & 384 & 4429 \\
-1394 & 0 & -1429 & 384 & 8923 & -375 \\
151 & 1787 & 0 & 4429 & -375 & 5716
\end{bmatrix}
\]  
(160)

Element Stiffness Matrix for Spring #4

The stiffnesses for Spring #4, or the shockloop, were calculated by Ramani [10], by performing a finite element solution of the shockloop, which was grounded at the bottom node, and which had unit displacements applied at its top node, in six coordinate directions. The following were the results:

- Lateral Stiffness, \(K_{xx} = 66.6\) lb/in
- Lateral Stiffness, \(K_{yy} = 94.3\) lb/in
- Axial Stiffness, \(K_{zz} = 79.2\) lb/in
- Moment Stiffness, \(K_{\theta_x\theta_x} = 5224\) lb-in/rad
- Moment Stiffness, \(K_{\theta_y\theta_y} = 4813\) lb-in/rad
- Moment Stiffness, \(K_{\theta_t\theta_t} = 1780\) lb-in/rad

Spring Attachment point co-ordinates \((R_x, R_y, R_z) = (1.11, 4.14, 2.47)\)

Therefore, Element Stiffness Matrix for Spring #4 (Shockloop)
\[
K_{\text{Shockloop}} =
\begin{bmatrix}
66.6 & 18.9 & 14.1 & 102 & -351 & -43.5 \\
18.9 & 94.3 & 36.4 & 474 & -141 & 110 \\
14.1 & 36.4 & 79.2 & 220 & -169 & 32.9 \\
102 & 474 & 220 & 6410 & -900 & 300 \\
-351 & -141 & -169 & -900 & 5240 & -2196 \\
-43.5 & 110 & 32.9 & 300 & -2196 & 2862 \\
\end{bmatrix}
\] (161)

The Total Assembly Stiffness Matrix can be written by summing up Eq. 158, Eq. 159, Eq. 160, and Eq. 161, and as discussed in Eq. 38 in Sec. 3.2.4.3.

Therefore the Total Assembly Stiffness Matrix,

\[
K_g =
\begin{bmatrix}
1328 & 18.9 & 14.1 & 101 & -2123 & 385 \\
18.9 & 1355 & 36.4 & 2246 & -141 & 96.7 \\
14.1 & 36.4 & 1079 & -115 & -159 & 32.9 \\
101 & 2246 & -115 & 23046 & -903 & 282 \\
-2123 & -141 & -159 & -903 & 30870 & -2006 \\
385 & 96.7 & 32.9 & 282 & -2006 & 14486 \\
\end{bmatrix}
\] (162)

4.12 Eigenanalysis comparison between analytical and experimental methods

<table>
<thead>
<tr>
<th>Eigenvalue frequency mode#</th>
<th>Analytical eigenvalue (Hz)</th>
<th>Experimental eigenvalue (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>2</td>
<td>11.3</td>
<td>10.5</td>
</tr>
<tr>
<td>3</td>
<td>11.6</td>
<td>11.9</td>
</tr>
<tr>
<td>4</td>
<td>15.4</td>
<td>13.2</td>
</tr>
<tr>
<td>5</td>
<td>15.4</td>
<td>16.0</td>
</tr>
<tr>
<td>6</td>
<td>17.8</td>
<td>18.9</td>
</tr>
</tbody>
</table>

A comparison of the analytically obtained eigenvalue frequencies and the experimentally obtained eigenvalue frequencies is shown above. A detailed output of the program BRESP.FOR,

4. Evaluation of Parameters
showing eigenvalues and eigenvectors, is presented in Appendix E. The analytical determination of stiffnesses was done as mentioned above. Appendix J discusses the method of experimental determination. In order to achieve a correlation of results between the analytical and the experimental methods of modal analysis, a sensitivity solution for the stiffnesses, was attempted by starting with values, determined by performing a finite element solution for the top spring and the side spring. The goal was to obtain the values for these stiffness variables, using an iterative method like the Newton-Raphson method, as discussed in Carnahan [15], and approximating the eigenfunction minimum, by obtaining a least square approximation between the eigenvalues generated by each new stiffness value and the experimental eigenvalues. An accuracy of 0.01 was achieved in terms of the least square minimum, which can be defined as the sum of the squares of the differences between the analytical and the experimental eigenvalue frequencies. A table is presented in Appendix D which shows the changes in the various stiffnesses.

In the case of the top spring, it can be observed from Fig. 26a that the lateral stiffness of the spring started to increase steadily from 147 lb/in to 411 lb/in, in proportion to the decrease in the eigenfunction minimum. As shown in Fig. 26b, the axial stiffness remained constant at 100 lb/in, as the eigenfunction minimum decreased from 6.7 units to .57 units, and then the axial stiffness started to increase steadily from 100 lb/in to 374 lb/in for the proportional decrease in the eigenfunction minimum.

In the case of the side spring, it can be observed from Fig. 27a, that the lateral stiffness of the spring started to decrease steadily from 507 lb/in to 212 lb/in, in proportion to the decrease in

4. Evaluation of Parameters
Figure 26. Eigen function minimum vs. stiffness plots for the top spring.
Figure 27. Eigen function minimum vs. stiffness plots for the side spring
the eigenfunction minimum. As shown in Fig. 27b, the axial stiffness first started to increase from 478 lb/in to 676 lb/in, as the eigenfunction minimum decreased from 6.7 units to 4.5 units. The axial stiffness then remained constant at 676 lb/in, as the eigenfunction minimum decreased from 4.5 units to 1.7 units, and then the axial stiffness started to increase steadily from 676 lb/in to 483 lb/in, for the proportional decrease in the eigenfunction minimum.

4.13 Application of point masses to the connecting rod

As discussed in Sec. 3.2.5.1, two point masses, one at the rotating center, and the other at the translating end, are used to replace the mass of the connecting rod. As shown in Fig. 28, the following are the dimensions of the connecting rods used in the H-25A assembly.

Distance between centers, \( l = 2.38 \) in

Distance between rotating center and center of mass, \( l_{rot} = 1.75 \) in.

Distance between translating center and center of mass, \( l_{trans} = 0.625 \) in.

Weight of the connecting rod, \( W = 0.131 \) lb (from a finite element solution by Ramani [10])

Applying Eq. 42 from Sec. 3.2.5.1, we have

Weight of point mass at rotating end,

\[
W_{rot} = 0.077 \text{ lb}
\]  \hspace{1cm} (163)

Applying Eq. 43 from Sec. 3.2.5.1, we have

Weight of point mass at translating end, \( W_{trans} = 0.054 \) lb
Figure 28. Application of point masses to the connecting rod

$W_{rot} = 0.076\text{ lb}$

$W_{trans} = 0.789\text{ lb}$

$2.375 \pm 0.001$

65° REF
But the total weight acting at the translating end would be the sum of the weight of the point mass placed at the wrist pin center of the connecting rod, the weight of the piston and the weight of the wrist pin.

Therefore weight of mass acting at the translating end,

\[ W_{\text{trans}} = W'_{\text{trans}} + W_{\text{pist}} + W_{\text{wpin}} \]  

(164)

Applying Eq. 164, we have

\[ W_{\text{trans}} = 0.789 \text{ lb} \]  

(165)

4.14 The Total Shaking Forces and Moments Matrix

As discussed in Sec. 3.2.5.6, the total shaking forces and moments matrix, \( F_g \), is obtained from Eq. 74, which can be rewritten as

\[ [F_g] = (F_{g\text{REACTION}}) + (F_{g\text{TORQUE}}) + (F_{g\text{TOTCW}}) \]

where

\( F_{g\text{REACTION}} = \) The force and moment matrix due to reaction forces
\( F_{g\text{TORQUE}} = \) The torque effect matrix
\( F_{g\text{TOTCW}} = \) The total counterweight forcing and moment matrix

Source code lists for the subroutines, COMPR, which was used to calculate \( F_{g\text{REACTION}} \) and BRCWFM, which was used to calculate \( F_{g\text{TOTCW}} \), can be obtained in Appendix F.

The results of the force analysis are shown from Fig. 29 through Fig. 36.

4. Evaluation of Parameters
Figure 29. The assumed P-V diagram for the H-25A compressor (k=1.30)
Figure 30. The output torque of the rotor

4. Evaluation of Parameters
DISPLACEMENT OF PISTON #1 vs. TIME

DISPLACEMENT OF PISTON #2 vs. TIME

Figure 31. Piston displacement vs. angular rotation of the crank
Figure 32. Force in the X-direction acting at the center of mass of the assembly
Figure 33. Force in the Y-direction acting at the center of mass of the assembly
Figure 34. Moment about X-direction acting at the center of mass of the assembly
Figure 35. Moment about Y-direction acting at the center of mass of the assembly.

4. Evaluation of Parameters
Figure 36. Moment about Z-direction acting at the center of mass of the assembly
Fig. 29 shows the P-V diagram for the H-25A assembly. The minimum and maximum values of the gas pressure are 77 psi and 298 psi respectively. Fig. 30 shows the plot of the output torque from the rotor, as a function of crank angle. The maximum output torque developed is about 245 lb-in. As shown in Fig. 31, the displacement of the two pistons, when plotted as a function of the angular rotation of the crank, are observed to be about 180° out of phase with each other.

Fig. 33 shows the force in the Y-direction, which is the direction parallel to the axes of the two cylinders. As expected, the force in the Z-direction, experienced at the center of mass of the assembly, would be zero, as there is no vertical motion. The moments experienced at the center of mass in the three different co-ordinate directions are shown in Figs. 34 through 36. The moments in the X and Y directions are mainly due to the rotating counterweights attached to the rotor and the crankshaft.
5.0 Discussion of Results

5.1 Verification with Integrated Mechanisms Program (IMP)

The results from the force analysis were verified by defining the compressor assembly model using a kinematic analysis package called Integrated Mechanisms Program (IMP) and comparing the results. The assembly model, shown in Fig. 37 was defined based on the following conditions.

i) The global coordinate system in IMP was defined, such that its origin was at the bottom of the crankshaft, and had the same direction vectors as the global coordinate system in the FORTRAN program, whose origin was at the center of mass of the assembly.

ii) The local X-axis of the crank corresponded with the global Y-direction, as shown in Fig. 37. This had to be done in order to enable the crank rotation to start with the piston in the bottom cylinder to lie at its Top Dead Center (TDC), and the piston in the top cylinder to lie at its Bottom Dead Center (BDC), as was the case with the FORTRAN program.

iii) A negative angle rotation was required for the crank to rotate in the anti-clockwise direction about the global Z-axis.

iv) The pressure forces on the two pistons were read from two separate data files, one file corresponding to each piston. Plots for the pressure forces vs. crank angle are shown in Fig. 38.
Figure 37. The H-25A assembly as defined in IMP
Figure 38. Plot of pressure force vs. crank angle for the two pistons (k = 1.30)
v) The necessary sign changes for the forces were made so that comparisons between the IMP solution and the FORTRAN program solution were made in the same coordinate directions.

vi) An important drawback of IMP was that it would not consider coupling due to forces acting at the joints. The FORTRAN program solution calculated moments due to these forces.

vii) Figure 39 shows the comparison of solutions of the forces acting in the X global direction. Both the solutions do not consider coupling due to forces at the joints.

viii) Figure 40 shows the comparison of solutions of the forces experienced in the Y global direction. The coupling due to forces is neglected in this case.

ix) Figure 41 shows the comparison of moments about the Z global direction. As the coupling is neglected, this moment would be equal to the resisting torque of the load, \( T_z \).

The forces obtained from the FORTRAN program compared well with the IMP solution, as is observed from these plots.

The source code for the input to the Integrated Mechanisms Program to perform the force analysis on the Bristol H-25A compressor assembly is shown in Appendix G.

5. Discussion of Results
Figure 39. Comparison of forces in X between FORTRAN program and IMP
Figure 40. Comparison of forces in Y between FORTRAN program and IMP
Figure 41. Comparison of moments about Z, between FORTRAN program and IMP
5.2 Response analysis of the system

After the eigenanalysis and the force analysis were performed on the Bristol H-25A reciprocating compressor, the response of the system was calculated. First, the forcing function, obtained in the time domain, for the first 40 harmonics, was converted to the frequency domain, using the Fast Fourier Transform method, as discussed by Brigham [16]. The forcing function, which would serve as input to the finite element solution of the system was written to a data file in the form of a three-dimensional matrix of order (40x6x2), corresponding to the first 40 harmonics of the forces and moments about the three global coordinate directions, having the cosine and sine components. The 40 harmonics were considered, as the finite element solution was proposed to make a comparison with the experimental solution at frequency levels up to 2250 Hz.

Using Eq. 81, which is discussed in Sec. 3.2.6, to calculate the response in the transformed coordinates, \( \tilde{Y} \), and thus, using Eq. 82, the response in the original coordinates, \( X \), were calculated.

Rewriting the equation of motion in the transformed coordinates,

\[
m\ddot{Y} + c\dot{Y} + kY = F\sin \omega t
\]

which is the sine function for a forced harmonic motion, whose solution, as discussed by Thomson [12] is given by

\[
Y_i = \frac{F_i / k_{ii}}{\sqrt{1 - \left(\frac{\omega}{\omega_n}\right)^2 + \left(\frac{2\xi\omega}{\omega_n}\right)^2}}
\]
and when $[C_e]$ is proportional to $[M_y]$ or $[K_e]$, will have the form

$$Y_i = \frac{F_i}{\sqrt{\left(k_{ii} - m_{ii}\omega^2\right)^2 + (c_{ii}\omega)^2}}$$

and the response in the original coordinates is given by

$$X = PY$$

The displacement vector for the center of mass of the frame, also known as the X vector was divided into two two-dimensional vectors, one representing the cosine component and the other representing the sine component of displacement, and each having an order (40x6).

The output from the response program, BRESP.FOR is shown in Appendix E.

5. Discussion of Results
6.0 Conclusion and Recommendations

6.1 Animation of orbit of motion of the compressor

A computer program (shown in Appendix I) was written in PHIGS (Programmer's Hierarchical Interactive Graphics Standard) to draw the frame of the assembly, supported by the three springs, and to show its animation of orbit of motion using the X displacement vector obtained in Sec. 5.2.

The animation, as viewed from the three global coordinate directions was also shown simultaneously, by opening three partial window views. The program would run on a RISC/6000 machine, in a UNIX environment, having a PHIGS interface. The PHIGS response program was user friendly, allowing the user to vary the displacement and rotation scales to obtain the necessary viewing. A frozen screen picture during animation, of the same program is shown in Fig. 42.

The displacements were then calculated as a function of time, within one time period, using the following equation, as discussed by Thomson [12], that any periodic motion can be represented by a series of sines and cosines that are harmonically related. Thus if \( X(t) \) is a periodic function of the period, \( \tau \), it is represented by the Fourier series

\[
X(t) = \frac{a_0}{2} + a_1 \cos \omega_1 t + a_2 \cos \omega_2 t + \ldots + b_1 \sin \omega_1 t + b_2 \sin \omega_2 t + \ldots
\]  

(166)

The time period corresponding to a constant speed rotation of 3450 rpm, \( \tau \), was divided into 100 intervals and the plots for the displacement vs. number of time intervals were taken.
Figure 42. Animation of the orbit of motion of the assembly using PHIGS
The displacement plots that are shown from Fig. 43 through Fig. 47 are obtained from the output of the PHIGS computer program BR_ANIM.C. Fig. 43 shows the plot of displacement in X in one time period. As the global X-direction is parallel to the line connecting the centers of the two side springs, the displacement would be smaller in value (maximum X being 0.004 in), as compared to the displacement in Y (maximum Y being 0.015 in), whose direction is parallel to the axis of the two cylinders.

The displacement in Y vs. the number of time intervals is shown in Fig. 44, and this displacement is mainly due to the recircuation of the pistons in that direction.

The displacement in Z, as one would expect would be zero, as the force in the vertical direction, i.e. the Z direction is zero.

The rotation about the X and Y directions are shown in Fig. 45 and Fig. 46 respectively, and the main contributions in this case are from the moments due to forces developed by rotating and reciprocating masses, including the counterweight masses attached to the rotor and the crankshaft.

The rotation about the Z direction is shown in Fig. 47, and this is due to the sum of the moments due to the shaft torque $T_r$, and the moment due to forces generated by rotating and reciprocating masses.

It could be concluded from this research that one of the major factors affecting the accuracy of the modeling and simulation of the vibrations of the compressor, was the involvement of moments generated due to forces acting at various joints. The FORTRAN program for the force analysis of the compressor used the relations to calculate the moments developed due to these forces at the

6. Conclusion and Recommendations
Figure 43. Displacement in X vs. number of time intervals
Figure 44. Displacement in Y vs. number of time intervals
Figure 45. Rotation about X vs. number of time intervals
Figure 46. Rotation about Y vs. number of time intervals
Figure 47. Rotation about Z vs. number of time intervals
center of mass of the assembly, and thus improvised the solution as compared to that obtained from the Integrated Mechanisms Program (IMP).

6.2 Recommendation for further research

6.2.1 Development of a variable speed vibration model

The model used in this research made an assumption that the crankshaft rotated with uniform angular velocity. An improvement in this model could be obtained by developing a variable speed model. The model will use a dynamic model of the induction motor of the type developed by Stanley [17]. This model when coupled with the forced model developed herein, will allow reasonable fluctuations in the speed of the crankshaft.
APPENDIX A. Mass Properties

**BRISTOL H-25 A PART WEIGHT VALUES**

<table>
<thead>
<tr>
<th>NAME OF THE COMPONENT</th>
<th>WEIGHT IN lb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. PISTON</td>
<td>.659</td>
</tr>
<tr>
<td>2. WRIST PIN</td>
<td>.075</td>
</tr>
<tr>
<td>3. CONNECTING ROD</td>
<td>.131</td>
</tr>
<tr>
<td>4. CRANKSHAFT</td>
<td>3.26</td>
</tr>
<tr>
<td>5. CRANKCASE</td>
<td>18.3</td>
</tr>
<tr>
<td>6. MUFFLER ASSEMBLY</td>
<td>1.49</td>
</tr>
<tr>
<td>7. VALVE PLATE</td>
<td>.578</td>
</tr>
<tr>
<td>8. CYLINDER HEAD</td>
<td>2.94</td>
</tr>
<tr>
<td>9. PILOT BEARING</td>
<td>.299</td>
</tr>
<tr>
<td>10. STATOR</td>
<td>40.1</td>
</tr>
<tr>
<td>11. ROTOR</td>
<td>9.00</td>
</tr>
</tbody>
</table>
APPENDIX B. Total Assembly Stiffness Matrix

\[
[K_{\text{elem}}] =
\]

\[
\begin{bmatrix}
K_a & K_a & K_a & K_a & K_a - K_a R_e + K_a P_e \\
K_a & K_e & K_e & K_e & K_e - K_e R_e + K_e P_e \\
K_a & K_e & K_e & K_e & K_e - K_e R_e + K_e P_e \\
K_a & K_a - K_a R_e + K_a R_e & K_a - K_a R_e + K_a R_e & K_a - K_a R_e + K_a R_e & K_a - K_a R_e + K_a R_e \\
K_a & K_a - K_a R_e + K_a R_e & K_a - K_a R_e + K_a R_e & K_a - K_a R_e + K_a R_e & K_a - K_a R_e + K_a R_e \\
\end{bmatrix}
\]
APPENDIX C. Response program BRESP.FOR

*****************************************************************************
C C C
C TITLE: BRISTOL H25-A RESPONSE ANALYSIS
C WRITTEN BY: RAMAKANT P. ARCOT & REGINALD G. MITCHINER
C SOURCE CODE: BRESP.FOR
C C
C*****************************************************************************
C Description: This program calculates the steady state response for 6 degrees
C of freedom. An eigenanalysis is performed and the steady state response in the
C form of the original coordinates (X) will be reported.
C
C GLOBAL VARIABLES:
C N - NUMBER OF DEGREES OF FREEDOM
C M - MASS MATRIX
C K - STIFFNESS MATRIX
C C - GYROSCOPIC MATRIX
C VEC - MATRIX CONTAINING EIGEN VECTORS COLUMNWISE
C VALS - MATRIX CONTAINING EIGEN VALUES
C F - FORCING FUNCTION MATRIX (COEFFICIENTS)
C P - NORMALIZED MODAL MATRIX
C PT - TRANPOSED MODAL MATRIX
C PI - INVERTED MODAL MATRIX
C GM - GENERALIZED MASS MATRIX
C GC - GENERALIZED GYROSCOPIC MATRIX
C GK - GENERALIZED STIFFNESS MATRIX
C GF - GENERALIZED FORCE MATRIX
C X - MATRIX CONTAINING COSINE & SINE COEFFICIENTS (X)

PROGRAM STFEIG

COMMON/DATA/XRESP(6,2,41)
REAL MASS,STIFF,GYRO
REAL RX,RX,RZ,KXX,KYY,KZZ,KTXTX
1 ,KTTY,KTZ,KTXTKXTY,KTXTY,KTXT
2 ,KTTY,KTYZ,KTXT,KTYKTXT
3 ,KTXTX,KTXTY,KXX,KYZ,KZX
DIMENSION MASS(6,6),STIFF(6,6),GYRO(6,6),FORCE1(6,2,41)
DIMENSION RX(10),RY(10),RZ(10),KXX(10),KYY(10),KZZ(10)
1 ,KTTY(10),KTXTZ(10),KTXTX(10),KTXTY(10),KTXTZ(10),KTXT(10)
2 ,KTTY(10),KXT(10),KXTZ(10),KXTY(10),KTXT(10),KTXTY(10)
3 ,KTXTX(10),KTXTY(10),KXX(10),KYZ(10),KZX(10),KTXTX(10)
4 ,TSTF(6,6),TSTSL(6,6),TRAD(6),THZ(6),TRPM(6)
DIMENSION FORCE(6,2,41),P(6,6),PT(6,6),PI(6,6),GM(6,6),GK(6,6)
1,X(6,2),GF(6,2),GC(6,6),VECS(6,6),VALS(6),TEMP(6),Y(6,2),FCS(6,2)
CHARACTER STRING*10,PROMPT*2
INTEGER NN,NCOL,ROW,TJCOL

C
C Open input & output files
C
OPEN (8, FILE = 'FORCE.DAT', STATUS = 'OLD')
OPEN (9, FILE = 'RESPONSE.OUT', STATUS = 'NEW')
OPEN (78, FILE = 'YCOFF.DAT', STATUS = 'NEW')
OPEN (79, FILE = 'XCOFF.DAT', STATUS = 'NEW')

C
C Initialize all working matrices
C
DO 8 I=1,10
RX(I)=0.
RY(I)=0.
RZ(I)=0.
KXX(I)=0.
KYY(I)=0.
KZZ(I)=0.
KTXTX(I)=0.
KTYTY(I)=0.
KTZTZ(I)=0.
KXY(I)=0.
KZX(I)=0.
KXX(I)=0.
KXY(I)=0.
KZ(I)=0.
KTY(I)=0.
KZ(I)=0.
KTX(I)=0.
KTY(I)=0.
KZ(I)=0.
KXTX(I)=0.
KTY(I)=0.
KZ(I)=0.
KTX(I)=0.
KTY(I)=0.
KZ(I)=0.
KTX(I)=0.
KTY(I)=0.
KZ(I)=0.
KTXT(I)=0.
KTY(I)=0.
KZ(I)=0.
KT(I)=0.

8 KTYTZ(I)=0.

Appendix C.  Response Program BRESP.FOR
C
DO 9 I=1,6
DO 9 J=1,6
STIFF(I,J)=0.
GYRO(I,J)=0.
9 MASS(I,J)=0.

DO 20 I=1,6
DO 20 J=1,2
DO 20 K=1,41
20 FORCE(I,J,K) = 0.

C
C N = 3450 RPM
C
FREQ = 361.28

C
C Enter the matrix values
C
C
C Define the number of springs
C
NSPR = 4
C
C SPRING 1 (Top Spring)
C
RX(1)=-.0103
RY(1)=-.9252
RZ(1)=7.4342
KXX(1)=136.56
KYY(1)=136.56
KZZ(1)=100.
KTXX(1)=175.85
KTYTY(1)=175.85
KTZTZ(1)=0.
C
C SPRING 2 (Left side spring as seen from muffler side)
C
RX(2)=-3.1978
RY(2)=-.26895
RZ(2)=.24794
KXX(2)=562.28
KYY(2)=562.28
KZZ(2)=450.
KTXX(2)=924.47
KTYTY(2)=924.47
KTTZT(2)=0.
C
C SPRING 3 (Right side spring as seen from muffler side)

Appendix C. Response Program BRESP.FOR
C
RX(3)=+3.1772
RY(3)=-.26895
RZ(3)=-2.4794
KXX(3)=562.28
KYY(3)=562.28
KZZ(3)=450.
KTXTYX(3)=924.47
KTYTYZ(3)=924.47
KZTZTZ(3)=0.

C
C SPRING 4  (Shockloop part that enters the muffler)
C
RX(4)=1.1147
RY(4)=4.1373
RZ(4)=2.4737
KXX(4)=66.570
KYY(4)=94.330
KZZ(4)=79.150
KTXTYX(4)=5224.6
KTYTYZ(4)=4813.0
KZTZTZ(4)=1780.0

KXY(4) = 18.98
KZX(4) = 14.14
KXTX(4) = 90.1
KXTY(4) = -499.1
KXTZ(4) = 210.8

KYZ(4) = 36.44
KTYTX(4) = 556.5
KTYT(4) = -147.2
KYTZ(4) = 83.07

KZTX(4) = -17.66
KZTY(4) = -116.0
KZTZ(4) = 50.75

KTXTY(4) = -663.6
KTXTZ(4) = 439.7
KTYTZ(4) = -1587.0

C
C Load the mass matrix here
C

MASS(1,1)=79.55/386.
MASS(2,2)=79.55/386.
MASS(3,3)=79.55/386.
MASS(4,4)=1303.80654/386.
MASS(5,5)=1144.499571/386.
MASS(6,6)=560.02008/386.
MASS(4,5)=-0.1991078/386.
MASS(5,4)=0.1991078/386.
MASS(4,6)=-0.008400733/386.
MASS(6,4)=-0.008400733/386.
MASS(5,6)=+189.1067/386.
MASS(6,5)=+189.1067/386.

C
C Load the Gyroscopic Matrix here
C

C IROT * OMEGA

GYRO(4,5) = 10.1262*FREQ/386.
GYRO(5,4) = -10.1262*FREQ/386.

WRITE(*,*) 'Mass Matrix = '
WRITE(*,33) (MASS(I,J),J=1,6),I=1,6
READ(*,29) PROMPT
WRITE(9,*) 'Mass Matrix = '
WRITE(9,33) (MASS(I,J),J=1,6),I=1,6
WRITE(9,*)

WRITE(*,*) 'Gyroscopic Matrix = '
WRITE(*,33) (GYRO(I,J),J=1,6),I=1,6
READ(*,29) PROMPT
WRITE(9,*) 'Gyroscopic Matrix = '
WRITE(9,33) (GYRO(I,J),J=1,6),I=1,6
WRITE(9,*)

C
C Load the forcing function matrix here FORCE(6,2,41)
C

DO 101 I=1,6
DO 101 K=1,41
READ(8,21) FORCE(I,1,K),FORCE(I,2,K)
101 CONTINUE

21 FORMAT (10X,F14.6,10X,F14.6)

DO 107 I=1,41
WRITE(78,787)FORCE(I,1,1),FORCE(I,1,2),FORCE(I,2,1),FORCE(I,2,2),
+ FORCE(3,1,1),FORCE(3,2,1),FORCE(4,1,1),FORCE(4,2,1),
+ FORCE(5,1,1),FORCE(5,2,1),FORCE(6,1,1),FORCE(6,2,1)
107 CONTINUE
C Loop through each spring
C
DO 30 K=1,NSPR
C
C Set up temporary stiffness matrix for each spring
C
DO 35 L=1,6
DO 35 M=1,6
35 TSTF(L,M)=0.
C
C Diagonal terms:
TSTF(1,1)= KXX(K)
TSTF(2,2)= KYY(K)
TSTF(3,3)= KZZ(K)
TSTF(4,4)= KYY(K)*(RZ(K)**2) + KZZ(K)*(RY(K)**2) + KTXTX(K) -
1 2.0*KYZ(K)*RY(K)*RZ(K)
TSTF(5,5)= (KXX(K)*(RZ(K)**2) + KZZ(K)*(RX(K)**2)) + KTYTY(K) -
1 2.0*KZX(K)*RZ(K)*RX(K)
TSTF(6,6)= (KXX(K)*(RZ(K)**2) + KYY(K)*(RX(K)**2)) + KZZ(K)*RY(K) -
1 2.0*KXY(K)*RX(K)*RY(K)
C
C REMAINING TERMS OF ROW 1:
TSTF(1,2)= KXY(K)
TSTF(1,3)= KZX(K)
TSTF(1,4)= KTXTX(K) + KZX(K)*RY(K) - KXY(K)*RZ(K)
TSTF(1,5)= KTYTY(K) + KXX(K)*RZ(K) - KZX(K)*RX(K)
TSTF(1,6)= KTXTZ(K) + KXY(K)*RX(K) - KXX(K)*RY(K)
C
C REMAINING TERMS OF ROW 2:
TSTF(2,3)= KYZ(K)
TSTF(2,4)= KTXTX(K) + KYZ(K)*RY(K) - KYY(K)*RZ(K)
TSTF(2,5)= KTYTY(K) + KXY(K)*RZ(K) - KYZ(K)*RX(K)
TSTF(2,6)= KTXTZ(K) + KYY(K)*RX(K) - KXY(K)*RY(K)
C
C REMAINING TERMS OF ROW 3:
TSTF(3,4)= KZTX(K) + KZZ(K)*RY(K) - KYZ(K)*RZ(K)
TSTF(3,5)= KZTY(K) + KZX(K)*RZ(K) - KZZ(K)*RX(K)
TSTF(3,6)= KZTXT(K) + KYZ(K)*RX(K) - KZX(K)*RY(K)
C
C REMAINING TERMS OF ROW 4:
TSTF(4,5)= KTXTY(K) + KYZ(K)*RX(K)*RZ(K) + KZX(K)*RY(K)*RZ(K)
I - KZZ(K)*RX(K)*RY(K) - KXY(K)*(RZ(K)**2)
TSTF(4,6) = KTZX(K) + KYZ(K)*RX(K)*RY(K) + KXY(K)*RY(K)*RZ(K)
I - KYY(K)*RX(K)*RZ(K) - KZX(K)*(RY(K)**2)

C REMAINING TERMS OF ROW 5:

TSTF(5,6) = KTTYZ(K) + KXY(K)*RX(K)*RZ(K) + KZX(K)*RX(K)*RY(K)
I - KXX(K)*RY(K)*RZ(K) - KYZ(K)*(RX(K)**2)

DO 36 IR=1,6
DO 36 JC=1,6
IF(IR.GT.JC) TSTF(IR,JC)=TSTF(JC,IR)
36  CONTINUE

C WRITE(9,*) 'SPRING #,K
WRITE(*,*) 'SPRING #,K
CALL MATPRT(TSTF,'TSTFS')

C C C
C Now update the system stiffness matrix
C
DO 37 I=1,6
DO 37 J=1,6
37  STIFF(I,J)=STIFF(I,J)+TSTF(I,J)
C 38  CONTINUE
C
C
C*** Write the mass matrix, stiffness matrix, the frequency &
C the forcing function to the screen, row-wise, five elements
C per line.

WRITE(*,*) 'The input data and the calculated output follow:'
WRITE(*,*) 'The elements of each matrix are written row-wise,'
WRITE(*,*) 'six elements to a line, except for the forcing'
WRITE(*,*) 'function and the steady state response in X where'
WRITE(*,*) 'they are written two elements a line. Please press'
WRITE(*,*) '<---| or RETURN to continue after a pause:'
WRITE(*,*) '
WRITE(*,*) '
READ(*,29) PROMPT

WRITE (*,*) 'Stiffness Matrix ='
WRITE(*,33) ((STIFF(I,J),J=1,6),I=1,6)
READ(*,29) PROMPT
WRITE (*,32) FREQ
READ(*,29) PROMPT
WRITE (9,*) 'Stiffness Matrix ='
WRITE(9,33) ((STIFF(I,J),J=1,6),I=1,6)
WRITE(9,*)
WRITE (9,32) FREQ

32 FORMAT(' Excitation frequency = ', G10.5, ' rad/sec')
33 FORMAT(2X,6G12.5)

DO 999 I = 1,6
DO 888 J = 1,6
C*** Reduce the problem to a standard symmetric eigen value
C problem

P(I,J) = STIFF(I,J) / (SQRT(MASS(I,J)) * SQRT(MASS(J,J)))
888 CONTINUE
999 CONTINUE

DO 333 I = 1,6
DO 444 J = 1,6
C*** Check the symmetry of the resultant matrix
IF((P(I,J) - P(J,I)) .LT. 1.0E-7) THEN
SYM = 0
ELSE SYM = 1
ENDIF
444 CONTINUE
333 CONTINUE

IF (SYM .EQ. 0) THEN
WRITE(*,*),
+ 'The matrix [1/SQRT(M1)]*[K]*[1/SQRT(M1)] is symmetric !!!'
ELSE
WRITE(*,*),
+ 'The matrix [1/SQRT(M1)]*[K]*[1/SQRT(M1)] is non-symmetric !!!'
ENDIF
READ(*,29) PROMPT
WRITE(*,*),'Please wait ....'
WRITE(*,*) ''

C*** Solve the eigen value problem by Jacobi's Method

CALL JACOBI(P,VECS,VALS,CHK,6)
WRITE(*,*) ' ORTHOGONALITY CHECK = ',CHK

C*** Change the arrangement of the eigen vectors from
C descending to ascending and storing it in P

DO 666 COL = 1,6
NN = 6 + 1 - COL
DO 777 ROW = 1,6
P(ROW,NN) = VECS(ROW,COL)

Appendix C. Response Program BRESP.FOR
C*** Print eigen values and eigen vectors on the screen

DO 1000 JCOL=1,6
   TJCOL = 6+1-JCOL
   TRAD(TJCOL) = SQRT(VALS(TJCOL))
   THZ(TJCOL) = TRAD(TJCOL)/(2*3.141592654)
   TRPM(TJCOL) = THZ(TJCOL)*60.
   WRITE(*,*) ' EIGENVALUE', JCOL,'=', 'VALS(TJCOL)
   WRITE(9,*), 'EIGENVALUE', JCOL,'=', 'VALS(TJCOL)
DO 2000 IROW=1,6
   P(IROW,JCOL) = P(IROW,JCOL) / SQRT(MASS(IROW,IROW))
   WRITE(*,*) ' EIGENVECTOR', IROW,'=', 'P(IROW,JCOL)
2000 WRITE(9,*), 'EIGENVECTOR', IROW,'=', 'P(IROW,JCOL)
IF(JCOL.EQ.3 .OR. JCOL.EQ.6) READ(*,29) PROMPT
1000 CONTINUE

WRITE(*,*)
WRITE(*,*) '----------------------------------
WRITE(*,*) 'EIGENVALUE FREQUENCIES'
WRITE(*,*) '----------------------------------
WRITE(*,*)
   + ' RAD/SEC HZ ','
   + ' RPM '

WRITE(9,*)
WRITE(9,*)
   + '----------------------------------
WRITE(9,*)
   + ' EIGENVALUE FREQUENCIES'
WRITE(9,*)
   + '----------------------------------
WRITE(9,*)
   + ' RAD/SEC HZ '
   + ' RPM '

DO 109 I = 6,1,-1
WRITE(*,*)
WRITE(*,*) TRAD(I),',THZ(I),',TRPM(I)
WRITE(9,*)
WRITE(9,*), TRAD(I),',THZ(I),',TRPM(I)
109 CONTINUE

C*** Normalize the Modal matrix, P

NN = 3
DO 40 J = 1,6
DO 41 I = 1,6

IF (ABS(P(I,J)).GT.ABS(TEMP(J))) THEN
TEMP(J) = P(I,J)
ENDIF

41 CONTINUE
40 CONTINUE

DO 50 J = 1,6
DO 51 K = 1,6
P(K,J) = P(K,J)/TEMP(J)
51 CONTINUE
50 CONTINUE

READ(*,29) PROMPT
WRITE(*,*) 'Normalized Modal Matrix ='
WRITE(*,34) ((P(I,J),J=1,6),I=1,6)

WRITE(9,*) 'Normalized Modal Matrix ='
WRITE(9,34) ((P(I,J),J=1,6),I=1,6)
WRITE(9,*)

DO 60 I = 1,6
DO 70 J = 1,6
PT(I,J) = P(I,J)
P(I,J) = PT(I,J)
70 CONTINUE
60 CONTINUE
READ(*,29) PROMPT

C*** Calculate Transposed modal matrix (PT), Inverted modal matrix (PI), Generalized mass matrix (GM), Generalized stiffness matrix (GK), and report to the screen.

CALL TRANS(PT,6)

WRITE(*,*) 'Transposed Modal Matrix ='
WRITE(*,34) ((PT(I,J),J=1,6),I=1,6)

WRITE(9,*) 'Transposed Modal Matrix ='
WRITE(9,34) ((PT(I,J),J=1,6),I=1,6)
WRITE(9,*)
READ(*,29) PROMPT

CALL INVERT(PI,6)

WRITE(*,*) 'Inverted Modal Matrix ='
WRITE(*,34) ((PI(I,J),J=1,6),I=1,6)
READ(*,29) PROMPT
C CALCULATE THE GENERALIZED MASS & STIFFNESS MATRICES - GM, GK

CALL PROD(MASS,P,VECS,6,6,6)
CALL PROD(PT,VECS,GM,6,6,6)
CALL PROD(STIFF,P,VECS,6,6,6)
CALL PROD(PT,VECS,GK,6,6,6)
CALL PROD(GYRO,P,VECS,6,6,6)
CALL PROD(PT,VECS,GC,6,6,6)

WRITE(*,*) 'Generalized Mass Matrix = ' 
WRITE(*,33) ((GM(I,J),J=1,6),I=1,6) 
READ(*,29) PROMPT

WRITE(*,*) 'Generalized Stiff Matrix = ' 
WRITE(*,33) ((GK(I,J),J=1,6),I=1,6) 
READ(*,29) PROMPT

WRITE(*,*) 'Generalized Gyro Matrix = ' 
WRITE(*,33) ((GC(I,J),J=1,6),I=1,6) 
READ(*,29) PROMPT

WRITE(9,*), 'Generalized Mass Matrix = ' 
WRITE(9,33) ((GM(I,J),J=1,6),I=1,6) 
WRITE(9,*), 'Generalized Stiff Matrix = ' 
WRITE(9,33) ((GK(I,J),J=1,6),I=1,6) 
WRITE(9,*), 'Generalized Gyro Matrix = ' 
WRITE(9,33) ((GC(I,J),J=1,6),I=1,6)

C --- Start looping for calculation of response for 40 harmonics
C

DO 301 K=1,41

DO 401 I=1,6
  DO 401 J=1,2
    FCS(I,J)=FORCE(I,J,K)
  401 CALL PROD(PT,FCS,GF,6,6,2)

C*******************************************************************************

DO 402 I = 1,6
  DO 402 J = 1,2
  402 FORCE(I,J,K)= GF(I,J)

C*******************************************************************************

C*** Use the magnification factor to calculate steady state
C (Y) response
DO 111 I = 1,6
DO 222 J = 1,2
Y(I,J) =
1 G(F(I,J)/SQR(T((GK(I,J)-GM(I,J))*FREQ**2)**2)) +
1 (GC(I,J)*FREQ)**2
222 CONTINUE
111 CONTINUE

C*** Use the equation X = PY to calculate the steady state
C (X) response and print it on the screen

CALL PROD(P,Y,X,6,6,2)

29 FORMAT (A2)
34 FORMAT (2X,6F12.5)
45 FORMAT (2X,2F15.5)

DO 501 I = 1,6
DO 501 J = 1,2
501 XRESP(I,J,K) = X(I,J)
601 CONTINUE

C******************************************************************************************************************************************

DO 463 I= 1,41
   WRITE(79,789)FORCE1(1,1,I),FORCE1(1,2,I),FORCE1(2,1,I),
  + FORCE1(2,2,I),FORCE1(3,1,I),FORCE1(3,2,I),FORCE1(4,1,I),
  + FORCE1(4,2,I),FORCE1(5,1,I),FORCE1(5,2,I),FORCE1(6,1,I),
  + FORCE1(6,2,I)
463 CONTINUE
789 FORMAT (12(1X,F12.6))

C******************************************************************************************************************************************

C Call the subroutine to plot the co-efficient C vs. Harmonic #
C
CALL PLTRSP

OPEN (11,FILE = 'RESPONSE.DAT',STATUS = 'NEW')
WRITE(11,*) 'RESPONSE IN ORIGINAL (X) CO-ORDINATES'
WRITE(11,*) 'FOR THE BRISTOL H-25 A COMPRESSOR'
DO 601 K= 1,41
   M = K - 1
   WRITE(11,*) 'FREQUENCY HARMONIC NO. ', M
601 DO 601 I = 1,6
   WRITE (11,55) XRESP(I,1,K),XRESP(I,2,K)

Appendix C. Response Program BRESP.FOR 135
OPEN (13, FILE = 'anim.dat', STATUS = 'NEW')

WRITE(13, *) 'Disp_Vec.Cos[41][6] = '

DO 225 K = 1, 41
  WRITE (13, 667) XRESP(1, 1, K), XRESP(2, 1, K), XRESP(3, 1, K),
    XRESP(4, 1, K), XRESP(5, 1, K), XRESP(6, 1, K)
225 CONTINUE

WRITE(13, *) 'Disp_Vec.Sin[41][6] = '

DO 551 K = 1, 41
  WRITE (13, 667) XRESP(1, 2, K), XRESP(2, 2, K), XRESP(3, 2, K),
    XRESP(4, 2, K), XRESP(5, 2, K), XRESP(6, 2, K)
551 CONTINUE

667 FORMAT ('(', 6(F15.7, ',', '))'
55 FORMAT(10X, F14.6, 10X, F14.6)
    CALL FINITT(0, 0)
STOP
END

C>>> Subroutine JACOBI calculates the eigen values & eigen vectors and checks the orthogonality of the vectors
C
SUBROUTINE JACOBI(MIN,VECS,VALS,CHK,N)
C
C ARGUMENT LIST:
C
C MIN - INPUT MATRIX, MUST BE SYMMETRIC - N BY N
C VALS - VECTOR OF EIGENVALUES, ORDERED LARGEST TO SMALLEST - LENGTH N
C VECS - MATRIX OF EIGENVECTORS, ORDERED COLUMNWISE ACCORDING TO
C         VECS - N BY N
C CHK - CHECK OF ORTHOGONALITY OF 1ST AND LAST EIGENVECTORS - REAL*4
C CHECK SHOULD APPROACH 0 FOR BEST PERFORMANCE OF ROUTINE
C N - SIZE OF ARRAYS, MUST BE 10 OR LESS
C
C THIS DOUBLE PRECISION SUBROUTINE DETERMINES THE EIGENVALUES AND
C EIGENVECTORS OF A SYMMETRIC MATRIX BY THE JACOBI METHOD. THE
C EIGENVALUES ARE ORDERED FROM THE LARGEST TO THE SMALLEST.
C
C ADAPTED FROM PROGRAM IN JAMES, SMITH, AND WOLFORD,
C "APPLIED NUMERICAL METHODS FOR DIGITAL COMPUTATION, 3RD EDITION",
C
C IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 MIN(N,N), VECS(N,N), VALS(N), CHK
DIMENSION A(10,10), RT(10,10)
DO 2 I=1,N
  DO 2 J=1,N

Appendix C. Response Program BRESP.FOR 136
A(I,J)=DBLE(MIN(I,J))

C GENERATE N X N IDENTITY MATRIX WHICH EVENTUALLY CONTAIN THE EIGENVECTORS

C DO 7 I=1,N
    DO 7 J=1,N
    RT(I,J)=0.0
    IF(I.EQ.J) RT(I,J)=1.0
7 CONTINUE
    NSWEEP=0
9 NRSKIP=0

C BEGIN A SWEEP WHICH WILL TRANSFORM EACH OFF-DIAGONAL ELEMENT IN TURN TO ZERO

C NMIN1=N-1
10 DO 19 I=1,NMIN1
    IP1=I+1
    DO 18 J=IP1,N
        AV=0.5*(A(I,J)+A(I,I))
        DIFF=A(I,I)-A(I,J)
        RAD=DSQRT(DIFF*DIFF+4.*AV*AV)
18    CONTINUE
19    CONTINUE

C CHECK TO SEE IF RAD IS ZERO. IF IT IS, NO ROTATION WILL BE PERFORMED
C SINCE CALCULATION OF SINE AND COSINE WOULD CAUSE OVERFLOWS.

C IF(RAD.EQ.0.) GO TO 14

C CHECK TO SEE IF DIFF IS NEGATIVE. IF IT IS, THE ELEMENTS A(I,I) AND A(J,J)
C ON THE DIAGONAL NEED TO BE INTERCHANGED; THEREFORE A ROTATION WILL BE
C PERFORMED.

C IF(DIFF.LT.0.) GO TO 12
    IF(DABS(A(I,I)).EQ.DABS(A(I,J))+100.*DABS(AV)) GO TO 10
    GO TO 11
10    IF(DABS(A(J,J)).EQ.DABS(A(J,I))+100.*DABS(AV)) GO TO 14
11    COSINE=DSQRT((RAD+DIFF)/(2.*RAD))
    SINE=AV/(RAD*COSINE)
    GO TO 13

C ON THIS PATH, ROTATIONS WILL ALWAYS BE PERFORMED SINCE ELEMENTS ON THE C DIAGONAL NEED REORDERING.

C 12    SINE=DSQRT((RAD-DIFF)/(2.*RAD))
    IF(AV.LT.0.) SINE=-SINE
    COSINE=AV/(RAD*SINE)

C CHECK TO SEE IF SIN(THETA) IS NEGLIGIBLE. IF IT IS, SKIP THE ROTATION

C 13    IF(1.LT.1.+DABS(SINE)) GO TO 15
14    NRSKIP=NRSKIP+1
    GO TO 18

C Appendix C. Response Program BRESP.FOR
C PERFORM THE ROTATION, PREMULTIPLY BY THE ROTATION MATRIX.
C
!5 DO 16 K=1,N
   Q=A(I,K)
   A(I,K)=COSINE*Q+SINE*A(J,K)
   A(J,K)=SINE*Q+COSINE*A(I,K)
16 CONTINUE
C
POSTMULTIPLY BY THE TRANSFORM OF THE ROTATION ANGLE
C
   DO 17 K=1,N
   Q=A(K,I)
   A(K,I)=COSINE*Q+SINE*A(K,J)
   A(K,J)=SINE*Q+COSINE*A(K,I)
17 CONTINUE
18 CONTINUE
19 CONTINUE
C
POSTMULTIPLY THE CURRENT PRODUCT OF ALL THE RT MATRICES UP TO THIS POINT
C BY THE CURRENT RT MATRIX. THIS PRODUCT OF RT MATRICES WHICH HAS THE
FORTRAN
C VARIABLE NAME RT WILL EVENTUALLY BECOME THE MATRIX CONTAINING THE
C EIGENVECTORS.
C
   Q=RT(K,I)
   RT(K,I)=COSINE*Q+SINE*RT(K,J)
   RT(K,J)=SINE*Q+COSINE*RT(K,I)
17 CONTINUE
18 CONTINUE
19 CONTINUE
C
KEEP A TALLY OF THE NUMBER OF SWEEPS
C
   NSWEEP=NSWEEP+1
   IF(NSWEEP.GT.50) GO TO 21
   IF(NRSKIP.LT.N*(N-1)/2) GO TO 9
C
AS AN ACCURACY CHECK, SEE IF THE DOT PRODUCT OF THE FIRST AND LAST
C EIGENVECTORS IS NEAR ZERO.
C
21 PROD=0.
   DO 28 J=1,N
      PROD=PROD+RT(J,1)*RT(J,N)
28 CONTINUE
   CHK=SNGL(PROD)
C
MAP DBLE PRECISION ARRAYS BACK INTO SNGL PRECISION ARRAYS
C
C VALS CONTAINS THE EIGENVALUES IN DESCENDING ORDER
C
   DO 30 J=1,N
30   VALS(J)=SNGL(A(J,J))
C
C VECS CONTAINS THE EIGENVECTORS IN COLUMN ORDER ACCORDING TO VALS
C
   DO 27 J=1,N
27   C

Appendix C. Response Program BRESP.FOR
C J IS THE EIGENVALUE INDEX
C
DO 25 I=1,N
  VECS(I,J)=RT(I,J)
25 CONTINUE
27 CONTINUE
RETURN
END

C>>> SUBROUTINE TO CALCULATE THE TRANSPOSE OF A MATRIX

SUBROUTINE TRANS(A,N)
  DIMENSION A(N,N)
  NN = N-1
  DO 3 I=1,NN
  K = I+1
  DO 3 J =K,N
  TEMP = A(I,J)
  C TRANSPOSE THE ELEMENTS
  A(I,J) = A(I,J)
 3 A(I,J) = TEMP
RETURN
END

C>>> SUBROUTINE TO CALCULATE THE INVERSE OF A MATRIX

SUBROUTINE INVERT(A,N)
  DIMENSION A(N,N)
C CALCULATE ELEMENTS OF REDUCED MATRIX
  DO 6 K=1,N
C CALCULATE NEW ELEMENTS OF PIVOT ROW
  DO 4 J=1,N
  IF(J.EQ.K) GO TO 4
  A(K,J) = A(K,J)/A(K,K)
 4 CONTINUE
C CALCULATE ELEMENT REPLACING PIVOT ELEMENT
  A(K,K) = 1.0/A(K,K)
C CALCULATE NEW ELEMENTS NOT IN PIVOT ROW OR PIVOT COLUMN
  DO 5 I=1,N
  IF(I.EQ.K) GO TO 65
5  CONTINUE
  DO 5 J=1,N
  IF(J.EQ.K) GO TO 5
  A(I,J) = A(I,J) - (A(K,J)*A(I,K))
 5 CONTINUE
C CALCULATE REPLACEMENT ELEMENTS FOR PIVOT COLUMN
C EXCEPT PIVOT ELEMENT
  DO 6 I=1,N
  IF(I.EQ.K) GO TO 6
  A(I,K) = -A(I,K)*A(K,K)
 6 CONTINUE
RETURN

Appendix C. Response Program BRESP.FOR
C>>>>> SUBROUTINE TO CALCULATE PRODUCT OF TWO MATRICES A & B
C AND STORE IT IN C

SUBROUTINE PROD(A,B,C,M,N,O)
INTEGER M,N,O
DIMENSION A(M,N),B(N,O),C(M,O)

DO 7 J = 1,M
DO 8 J = 1,O
DO 9 K = 1,N
IF(K.EQ.1) THEN
CSUM = 0.0
ENDIF
C CALCULATE THE PRODUCT OF A AND B AND STORE THE
C ELEMENTS ROW-WISE IN C
CSUM = CSUM + A(I,K)*B(K,J)
9 CONTINUE
C(I,J) = CSUM
8 CONTINUE
7 CONTINUE
RETURN
END

C SUBROUTINE MATPRT TO PRINT A MATRIX

SUBROUTINE MATPRT(MTRX,CHARS)
CHARACTER*5 CHAR
CHARACTER*1 STRING
REAL MTRX(6,6)
WRITE(*,*)
WRITE(9,*)
WRITE(*,'(3X,A)') CHAR
WRITE(9,'(3X,A)') CHAR
DO 10 IR=1,6
WRITE(9,15) (MTRX(IR,IC),IC=1,6)
WRITE(9,*)
10 WRITE(*,15) (MTRX(IR,IC),IC=1,6)
15 FORMAT(2X,6G12.5)
WRITE(*,*)
WRITE(*,'PRESS RETURN TO CONTINUE')
READ(*,'(A1)') STRING
RETURN
END

SUBROUTINE PLTRSP
COMMON/DATA/XRESPP(6,2,4)
DIMENSION XDATA(42),YDATA(42),ZDATA(42),TXDATA(42),TYDATA(42),TZDATA(42)
1 TDAT42,HMDATA(42)
DO 10 I = 1,42
   XDATA(I) = 0.
   YDATA(I) = 0.
   ZDATA(I) = 0.
   TXDATA(I) = 0.
   TYDATA(I) = 0.
   TZDATA(I) = 0.
   HMDATA(I) = 0.
10    CONTINUE

   HMDATA(1) = 40.
   XDATA(1) = 40.
   YDATA(1) = 40.
   ZDATA(1) = 40.
   TXDATA(1) = 40.
   TYDATA(1) = 40.
   TZDATA(1) = 40.

DO 20 K = 2,41
   HMDATA(K) = K - 1
   XDATA(K) = SQRT( XRESP(1,1,K)**2 + XRESP(1,2,K)**2 )
   YDATA(K) = SQRT( XRESP(2,1,K)**2 + XRESP(2,2,K)**2 )
   ZDATA(K) = SQRT( XRESP(3,1,K)**2 + XRESP(3,2,K)**2 )
   TXDATA(K) = SQRT( XRESP(4,1,K)**2 + XRESP(4,2,K)**2 )
   TYDATA(K) = SQRT( XRESP(5,1,K)**2 + XRESP(5,2,K)**2 )
   TZDATA(K) = SQRT( XRESP(6,1,K)**2 + XRESP(6,2,K)**2 )
20    CONTINUE

666   FORMAT ('(',6(F15.7,',')

CALL INITT (10)

C
C   COLUMN IDENTIFICATION
C
C   1 - DISP IN X VS. NO. OF HARMONICS
C   2 - DISP IN Y VS. NO. OF HARMONICS
C   3 - DISP IN Z VS. NO. OF HARMONICS
C   4 - ROT ABOUT X VS. NO. OF HARMONICS
C   5 - ROT ABOUT Y VS. NO. OF HARMONICS
C   6 - ROT ABOUT Z VS. NO. OF HARMONICS
C
11   CALL ERASE
     CALL ANMODE

C
C   WRITE(*,*)'BRISTOL H-25A COMPRESSOR'
   WRITE(*,*)'----------------------------------'
   WRITE(*,*)''
   WRITE(*,*)''
   WRITE(*,*)'PLOT OF FFT CO-EFFICIENT "C" VS. HARMONICS'
   WRITE(*,*)''
   WRITE(*,12)

Appendix C. Response Program BRESP.FOR
12  FORMAT(/,'SELECT APPROPRIATE OPTION:/',
1 'DISP IN X VS. NO. OF HARMONICS - 1/',
2 'DISP IN Y VS. NO. OF HARMONICS - 2/',
3 'DISP IN Z VS. NO. OF HARMONICS - 3/',
4 'ROT ABOUT X VS. NO. OF HARMONICS - 4/',
5 'ROT ABOUT Y VS. NO. OF HARMONICS - 5/',
6 'ROT ABOUT Z VS. NO. OF HARMONICS - 6/',
7 'QUIT - 11
   >> (11) ')
READ(*,13,END=11,ERR=11) IRPLY
13  FORMAT(I2)
   IF(IRPLY.EQ.0) IRPLY=11
   IF(IRPLY.LT.1 OR IRPLY.GT.12) GO TO 11
   IF(IRPLY.EQ.11) RETURN
   CALL ERASE
   CALL ANMODE
   CALL BINITT
C
   GO TO (100,200,300,400,500,600), IRPLY
C
   C X VS. HARMONICS
C
   100  CALL ERASE
       CALL BINITT
       CALL XFRM(2)
       CALL VBARST(4,10,0)
       CALL CHECK(HMDATA,XDATA)
       CALL DISPLAY(HMDATA,XDATA)
       CALL FRAME
       CALL LINE(0)
       CALL TINPUT (l)
       CALL ERASE
       GO TO 11

   200  CALL ERASE
       CALL BINITT
       CALL XFRM(2)
       CALL VBARST(4,10,0)
       CALL CHECK(HMDATA,YDATA)
       CALL DISPLAY(HMDATA,YDATA)
       CALL FRAME
       CALL LINE(0)
       CALL TINPUT (l)
       CALL ERASE
       GO TO 11

   300  CALL ERASE
       CALL BINITT
       CALL XFRM(2)
       CALL VBARST(4,10,0)
       CALL CHECK(HMDATA,ZDATA)
       CALL DISPLAY(HMDATA,ZDATA)
CALL FRAME
CALL LINE(0)
CALL TINPUT (I)
CALL ERASE
GO TO 11

400 CALL ERASE
   CALL BINITT
   CALL XFRM(2)
   CALL VBARST(4,10,0)
   CALL CHECK(HMDATA,TXDATA)
   CALL DISPLAY(HMDATA,TXDATA)
   CALL FRAME
   CALL LINE(0)
   CALL TINPUT (I)
   CALL ERASE
   GO TO 11

500 CALL ERASE
   CALL BINITT
   CALL XFRM(2)
   CALL VBARST(4,10,0)
   CALL CHECK(HMDATA,TYDATA)
   CALL DISPLAY(HMDATA,TYDATA)
   CALL FRAME
   CALL LINE(0)
   CALL TINPUT (I)
   CALL ERASE
   GO TO 11

600 CALL ERASE
   CALL BINITT
   CALL XFRM(2)
   CALL VBARST(4,10,0)
   CALL CHECK(HMDATA,TZDATA)
   CALL DISPLAY(HMDATA,TZDATA)
   CALL FRAME
   CALL LINE(0)
   CALL TINPUT (I)
   CALL ERASE
   GO TO 11
END
APPENDIX D. Comparison of eigenvalues

**NEWTON RAPHSON & LEAST SQUARE APPROXIMATION**

<table>
<thead>
<tr>
<th>SPRING STIFFNESS</th>
<th>TOP SPRING</th>
<th>SIDE SPRING</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{xx}$</td>
<td>146.72</td>
<td>411.68</td>
</tr>
<tr>
<td>$K_{zz}$</td>
<td>100.21</td>
<td>374.26</td>
</tr>
<tr>
<td>$K_{yz}$</td>
<td>-43.40</td>
<td>-43.40</td>
</tr>
<tr>
<td>$K_{\theta x \theta x}$</td>
<td>101.50</td>
<td>101.50</td>
</tr>
<tr>
<td>$K_{\theta x \theta y}$</td>
<td>51.27</td>
<td>51.27</td>
</tr>
<tr>
<td>$K_{\theta y \theta x}$</td>
<td>-5.24</td>
<td>-5.24</td>
</tr>
<tr>
<td>$K_{x \theta x}$</td>
<td>5.02</td>
<td>78.08</td>
</tr>
<tr>
<td>$K_{x \theta y}$</td>
<td>91.65</td>
<td>91.65</td>
</tr>
<tr>
<td>$K_{x \theta z}$</td>
<td>-8.40</td>
<td>-171.84</td>
</tr>
<tr>
<td>$K_{y \theta x}$</td>
<td>-91.12</td>
<td>-91.12</td>
</tr>
<tr>
<td>$K_{z \theta x}$</td>
<td>6.43</td>
<td>6.42</td>
</tr>
<tr>
<td>$K_{z \theta z}$</td>
<td>-6.19</td>
<td>-6.19</td>
</tr>
</tbody>
</table>
APPENDIX E. Output of program BRESP.FOR

****************************************
C
C TITLE: OUTPUT OF PROGRAM BRESP.FOR
C DESCRIPTION: DETAILS OF RESULTS OF SYSTEM RESPONSE
C NAME OF FILE: RESPONSE.OUT
C
C
C****************************************

Mass Matrix =
.20609  .00000  .00000  .00000  .00000  .00000
.00000  .20609  .00000  .00000  .00000  .00000
.00000  .00000  .20609  .00000  .00000  .00000
.00000  .00000  .00000  .3.3777  -.51582E-03  -.21764E-04
.00000  .00000  .00000  -.51582E-03  2.9650  .48991
.00600  .00000  .00000  -.21764E-04  .48991  1.4508

Gyrosopic Matrix =
.00000  .00000  .00000  .00000  .00000  .00000
.00000  .00000  .00000  .00000  .00000  .00000
.00000  .00000  .00000  .00000  .00000  .00000
.00000  .00000  .00000  .00000  .9.4777  .00000
.00000  .00000  .0000  -.9.4777  .00000  .00000
.00000  .00000  .0000  .0000  .0000  -.00000

SPRING #  1

TSTFS
136.56  .00000  .00000  .00000  1015.2  126.35
.00000  136.56  .00000  -1015.2  .00000  -1.4066
.00000  .00000  100.30  -.92.520  1.0300  .00000
.00000  -1015.2  -92.520  7808.8  -.95296  10.457
1015.2  .00000  1.0300  -.95296  7723.2  939.28
126.35  -1.4066  .00000  10.457  939.28  116.91

Appendix E. Output of program BRESP.FOR  145
<table>
<thead>
<tr>
<th>SPRING #</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTFS</td>
<td></td>
</tr>
<tr>
<td>562.28</td>
<td>.00000</td>
</tr>
<tr>
<td>.00000</td>
<td>562.28</td>
</tr>
<tr>
<td>.00000</td>
<td>.00000</td>
</tr>
<tr>
<td>.00000</td>
<td>1394.1</td>
</tr>
<tr>
<td>-1394.1</td>
<td>.00000</td>
</tr>
<tr>
<td>151.23</td>
<td>-1798.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPRING #</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTFS</td>
<td></td>
</tr>
<tr>
<td>562.28</td>
<td>.00000</td>
</tr>
<tr>
<td>.00000</td>
<td>562.28</td>
</tr>
<tr>
<td>.00000</td>
<td>.00000</td>
</tr>
<tr>
<td>.00000</td>
<td>1394.1</td>
</tr>
<tr>
<td>-1394.1</td>
<td>.00000</td>
</tr>
<tr>
<td>151.23</td>
<td>1786.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPRING #</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTFS</td>
<td></td>
</tr>
<tr>
<td>18.980</td>
<td>94.330</td>
</tr>
<tr>
<td>14.140</td>
<td>36.440</td>
</tr>
<tr>
<td>101.65</td>
<td>473.92</td>
</tr>
<tr>
<td>-350.19</td>
<td>-140.87</td>
</tr>
<tr>
<td>-43.463</td>
<td>109.69</td>
</tr>
</tbody>
</table>

Excitation frequency = 361.28 rad/sec
Stiffness Matrix =

<table>
<thead>
<tr>
<th></th>
<th>1327.7</th>
<th>18.980</th>
<th>14.140</th>
<th>101.65</th>
<th>-2123.2</th>
<th>385.33</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.980</td>
<td>1355.5</td>
<td>36.440</td>
<td>2246.9</td>
<td>-114.91</td>
<td>-158.95</td>
<td>32.868</td>
</tr>
<tr>
<td>14.140</td>
<td>36.440</td>
<td>1079.2</td>
<td>-114.91</td>
<td>23046.9</td>
<td>-903.02</td>
<td>281.60</td>
</tr>
<tr>
<td>101.65</td>
<td>2246.9</td>
<td>-114.91</td>
<td>23046.9</td>
<td>-903.02</td>
<td>30870.0</td>
<td>-2006.7</td>
</tr>
<tr>
<td>-2123.2</td>
<td>-140.87</td>
<td>-158.95</td>
<td>-903.02</td>
<td>-2006.7</td>
<td>14486.0</td>
<td></td>
</tr>
<tr>
<td>385.33</td>
<td>96.704</td>
<td>32.868</td>
<td>281.60</td>
<td>-2006.7</td>
<td>14486.0</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{EIGENVALUE}(1) = 3962.969000 \\
\text{EIGENVECTOR}(1) = 1.861116E-002 \\
\text{EIGENVECTOR}(2) = -1.5677740 \\
\text{EIGENVECTOR}(3) = 3.851098E-001 \\
\text{EIGENVECTOR}(4) = 3.699765E-001 \\
\text{EIGENVECTOR}(5) = 1.180736E-002 \\
\text{EIGENVECTOR}(6) = 5.871044E-003 \\
\text{EIGENVALUE}(2) = 5051.643000 \\
\text{EIGENVECTOR}(1) = 1.9481570 \\
\text{EIGENVECTOR}(2) = 6.899126E-002 \\
\text{EIGENVECTOR}(3) = 2.806006E-001 \\
\text{EIGENVECTOR}(4) = -1.297651E-002 \\
\text{EIGENVECTOR}(5) = 2.586563E-001 \\
\text{EIGENVECTOR}(6) = -3.407943E-002 \\
\text{EIGENVALUE}(3) = 5273.139000 \\
\text{EIGENVECTOR}(1) = -2.893004E-001 \\
\text{EIGENVECTOR}(2) = 2.683645E-001 \\
\text{EIGENVECTOR}(3) = 2.1495170 \\
\text{EIGENVECTOR}(4) = -6.592725E-002 \\
\text{EIGENVECTOR}(5) = -1.941118E-002 \\
\text{EIGENVECTOR}(6) = -8.161565E-004 \\
\text{EIGENVALUE}(4) = 9326.952000 \\
\text{EIGENVECTOR}(1) = 1.417683E-003 \\
\text{EIGENVECTOR}(2) = -1.4110310 \\
\text{EIGENVECTOR}(3) = -3.687264E-003 \\
\text{EIGENVECTOR}(4) = -3.673199E-001 \\
\text{EIGENVECTOR}(5) = 2.416377E-002 \\
\text{EIGENVECTOR}(6) = 3.018672E-001 \\
\text{EIGENVALUE}(5) = 9418.945000 \\
\text{EIGENVECTOR}(1) = -3.898884E-001 \\
\text{EIGENVECTOR}(2) = 5.346553E-001 \\
\text{EIGENVECTOR}(3) = -2.060692E-002 \\
\text{EIGENVECTOR}(4) = 1.291776E-001 \\
\text{EIGENVECTOR}(5) = 2.470242E-001 \\
\text{EIGENVECTOR}(6) = 6.807684E-001 \\
\text{EIGENVALUE}(6) = 12440.930000 \\
\text{EIGENVECTOR}(1) = 9.060476E-001 \\
\text{EIGENVECTOR}(2) = 2.018289E-001 \\
\text{EIGENVECTOR}(3) = 6.622787E-002 \\
\text{EIGENVECTOR}(4) = 5.548849E-002 \\
\text{EIGENVECTOR}(5) = -4.563345E-001 \\
\text{EIGENVECTOR}(6) = 3.653698E-001
### Eigenvalue Frequencies

<table>
<thead>
<tr>
<th>RAD/SEC</th>
<th>HZ</th>
<th>RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>62.9521200</td>
<td>10.0191400</td>
<td>601.1485000</td>
</tr>
<tr>
<td>71.0749100</td>
<td>11.3119200</td>
<td>678.7153000</td>
</tr>
<tr>
<td>72.6163906</td>
<td>11.5572600</td>
<td>693.4354000</td>
</tr>
<tr>
<td>96.5761500</td>
<td>15.3705700</td>
<td>922.2343000</td>
</tr>
<tr>
<td>97.0512500</td>
<td>15.4461900</td>
<td>926.7712000</td>
</tr>
<tr>
<td>111.5389000</td>
<td>17.7519700</td>
<td>1065.1180000</td>
</tr>
</tbody>
</table>

Normalized Modal Matrix =

\[
\begin{bmatrix}
-0.01187 & 1.00000 & -0.13459 & -0.00100 & -0.57272 & 1.00000 \\
1.00000 & 0.03541 & -0.12485 & 1.00000 & 0.78537 & 0.22276 \\
-0.24564 & 0.14403 & 1.00000 & 0.0261 & -0.03027 & 0.07310 \\
-0.23599 & -0.00666 & -0.03056 & 0.26032 & 0.18975 & 0.06124 \\
-0.00753 & 0.13277 & -0.00903 & -0.01712 & 0.36286 & -0.50365 \\
-0.00374 & -0.01749 & -0.00038 & -0.21393 & 1.00000 & 0.40326
\end{bmatrix}
\]

Transposed Modal Matrix =

\[
\begin{bmatrix}
-0.01187 & 1.00000 & -0.24564 & -0.23599 & -0.00753 & -0.00374 \\
1.00000 & 0.03541 & 0.14403 & 0.00666 & 0.01327 & 0.01749 \\
-0.13459 & 0.12485 & 0.00000 & -0.03056 & -0.00903 & -0.00038 \\
-0.00100 & 0.00000 & 0.00261 & 0.26032 & -0.01712 & -0.21393 \\
-0.57272 & 0.78537 & -0.03027 & 0.18975 & 0.36286 & 1.00000 \\
1.00000 & 0.22276 & 0.07310 & 0.06124 & -0.50365 & 0.40326
\end{bmatrix}
\]

Generalized Mass Matrix =

\[
\begin{bmatrix}
0.40687 & -16300E-03 & 16740E-04 & 81861E-03 & -43053E-02 & -62286E-03 \\
-16300E-03 & 25121 & 54751E-04 & 13787E-01 & 61924E-01 & 30541E-01 \\
16744E-04 & 54751E-04 & 21643 & 95046E-03 & 44844E-02 & 16978E-02 \\
81860E-03 & 13787E-01 & 95046E-03 & 50585 & 46473E-01 & 49470E-01 \\
-43053E-02 & 61924E-01 & -44844E-02 & -46473E-01 & 2.5132 & -1.7503 \\
-62285E-03 & 30541E-01 & -16978E-02 & 49470E-01 & -1.7502 & 1.0192
\end{bmatrix}
\]

Appendix E. Output of program BRESP.FOR
Generalized Stiffness Matrix =

\[
\begin{bmatrix}
1612.3 & .37236E-05 & .69352E-06 & .10392E-03 & .10716E-03 & -.29342E-04 \\
-.47331E-05 & 1331.0 & .33341E-04 & -.26600E-05 & .58910E-04 & -.19604E-03 \\
-.26500E-04 & .37724E-04 & 1141.3 & -.26442E-04 & -.21130E-04 & .68772E-05 \\
.96612E-04 & -.20389E-05 & -.26615E-04 & 4684.5 & .25230E-03 & .44848E-04 \\
.12589E-03 & .76294E-04 & -.45300E-04 & .24414E-03 & 20324. & .48828E-03 \\
.35708E-04 & -.26670E-03 & .54056E-04 & .64548E-04 & .28755E-03 & 15155. \\
\end{bmatrix}
\]

Generalized Gyroscopic Matrix =

\[
\begin{bmatrix}
.16479E-08 & -.29743 & .18016E-01 & .56884E-01 & -.79804 & 1.1309 \\
.29743 & -.51503E-09 & .39027E-01 & -.32649 & -.26168 & -.45269E-01 \\
-.18016E-01 & -.39027E-01 & .10209E-09 & .27241E-01 & -.88864E-01 & .15113 \\
-.56884E-01 & .32649 & -.27241E-01 & -.92586E-09 & .92606 & -.12327 \\
.79804 & .26168 & .88864E-01 & -.92606 & .53438E-08 & -.11164 \\
-1.1309 & .45269E-01 & -.15113 & 1.2327 & 1.1164 & -.23892E-07 \\
\end{bmatrix}
\]
SUBROUTINE BRCWFM(FX,FY,MX,MY,MZ,WCW,RHOCW,OMEGA,1,ALPHA,THETA,GAMMA,RCW)

C   THIS SUBROUTINE COMPUTES THE FORCES AND MOMENTS DUE
C   TO ROTATING COUNTERWEIGHTS.
C   WCW - Weight of rotating counterweight
C   RHOCW - Radius of rotating counterweight's center w.r.t. cylinder axes
C   RCW  - Position vector of rotating counterweight w.r.t. center of mass
C   FRAD - Radial component of force w.r.t. cylinder axes
C   FTAN - Tangential component of force w.r.t. cylinder axes

DIMENSION RCW(3)

FRAD = (WCW/386.)*(RHOCW)*(OMEGA**2)
FTAN = (WCW/386.)*(RHOCW)*ALPHA

C   ----Force in X at center of mass
FX = FTAN*COS(THETA+GAMMA) - FRAD*SIN(THETA+GAMMA)

C   ----Force in Y at center of mass
FY = FRAD*COS(THETA+GAMMA) + FTAN*SIN(THETA+GAMMA)

C   ----Force in Z at center of mass
FZ = 0.

C   ----Moment about X due to rotation of counterweight
MX = -RCW(3)*FY

C   ----Moment about Y due to rotation of counterweight
MY = +RCW(3)*FX

C   ----Moment about Z due to rotation of counterweight
MZ = +RCW(1)*FY - RCW(2)*FX

RETURN
END
SUBROUTINE COMPR(ANGLE, OMEGA, ALPHA)

C THIS SUBROUTINE COMPUTES ALL CYLINDER, PISTON, CROD, CRANKSHAFT, 
C AND BLOCK FORCES FOR A SINGLE CYLINDER.
C
C VARIABLES:
C 1  X - FROM TDC
C 2  XDOT - PISTON VELOCITY
C 3  XDDOT - PISTON ACCELERATION
C 4  PRESSABS - CYLINDER PRESSURE ABSOLUTE
C 5  PRESS - DIFFERENTIAL PRESSURE ACROSS PISTON
C 6  PF - PRESSURE FORCE ON PISTON
C 7  T - TOTAL TORQUE - POS EQUIV TO DEMAND
C 8  F14X - X FORCE CASE ON PIST IN CASE FRAME OF REF
C 9  F14Y - Y FORCE
C 10 F12X - X FORCE CASE ON CRANK IN CASE FRAME OF REF
C 11 F12Y - Y FORCE
C 12 VOL - VOLUME IN CYLINDER
C 13 F0 - D'ALEMBERT INERTIAL FORCE OF RECIP MASS
C 14 F41X - X FORCE PISTON ON CASE IN CASE FRAME OF REF
C 15 F41Y - Y FORCE
C 16 F21X - X FORCE CRANK ON CASE IN CASE FRAME OF REF
C 17 F21Y - Y FORCE
C 18 FCASEX - FORCE COMPONENT IN X DUE TO CRANK ROTATION
C 19 FCASEY - FORCE COMPONENT IN Y DUE TO CRANK ROTATION

REAL L
COMMON/CYL/1, X, XDOT, XDDOT, PRESSABS, PRESS, PF, T, F14X, F14Y, F12X,
  1 F12Y, VOL, F0, F41X, F41Y, F21X, F21Y, FCASEX, FCASEY
DATA TWP0/6.283185/
DATA GAMMA/0.0/, R/.500/, L/2.3750/, WTRANS/.7890/,
  1 WROT/.07686/, AREA/2.9560/,
  2 PMAX/298.0/, PMIN/77.0/, VCL/0.0600/, VMAX/2.4830/,
  3 XN/1.3000/, RCW/.500/, CYLANG/0.0/

C ANGLE - INPUT REFERENCE CRANK ANGLE
C GAMMA - DIFFERENCE BETWEEN CRANK REFERENCE AND CYL CRANK
C
C THETA = ANGLE - GAMMA - CYLANG

C POSITIVE DISPLACEMENT, VELOCITY, AND ACCELERATION ARE IN
C THE DIRECTION FROM TDC TOWARD BDC
C
C X = R*Cos(THETA) - Sqrt((L**2 - R**2) Sin(THETA)**2) + (R + L)
XDOT = OMEGA*(R*Sin(THETA) + .5*R**2*Sin(2.*THETA))/Sqrt(L**2 - R**2*Sin
  1 (THETA)**2))
XDDOT = ALPHA*(R*Sin(THETA) + .5*R**2*Sin(2.*THETA))/Sqrt(L**2 - R**2*Sin
  1 THETA)**2) + OMEGA**2*(R*COS(THETA) + R**2*COS(2.*THETA))/Sqrt(2*L**2 - R**2*Sin(THETA)**2) + .25*R**4*Sin(2.*THETA)**2)/Sqrt((L**2- 
  3 R**2*Sin(THETA)**2)**3))

Appendix F. Force analysis subroutines
C
C PHI - ANGLE BETWEEN CYL CL AND CR CL
C D - RADIUS OF CR FORCE ABOUT CRANK CENTER
C F0 - INERTIAL FORCE OF TRANS MASS
C
WPIST = .7352
PHI = ASIN(R * SIN(THETA) / L)
D = R * COS(2.0 * ATAN(1.0) - (THETA + PHI))
F0 = WPIST / 386.0 * XDDOT
C
C CALCULATE INSTANTANEOUS VOLUME IN CYLINDER
C VOL = X * AREA + VCL
C
IF(XDOT .GE. 0.0) PRESSABS = PMAX / (VOL / VCL) ** XN
IF(XDOT .LT. 0.0) PRESSABS = PMIN * (VMAX / VOL) ** XN
IF(PRESSABS .GT. PMAX) PRESSABS = PMAX
IF(PRESSABS .LT. PMIN) PRESSABS = PMIN
C
C CORRECT PRESSURE FOR CRANKCASE PRESSURE
C PRESS = PRESSABS - PMIN
C
PF - PRESSURE FORCE
FB3 - D'ALEMBERT'S FORCE DUE TO CONN ROD LOADING
FB - SUM OF INERTIAL AND PRESSURE FORCES
C
FB3 = WTRANS / 386.0 * XDDOT
PF = PRESS * AREA
FB = PF - (F0 + FB3)
C
Direction of +ve FB is from TDC to BDC
C
output torque
C TORQUE - POSITIVE TORQUE REPRESENTS DEMAND BY COMPRESSOR ON P MOVER
C T = -FB * D / COS(PHI)
C
C BLOCK FORCES WHICH ACT ON THE CRANK/PISTON
C POSITIVE Y IS FROM CRNK CNTR TOWARD PISTON
C POSITIVE X IS TO RIGHT WITH POSITIVE OMEGA CCW
C
C TRANSFORM INTO REF FRAME OF CRANKCASE
C
F14 = FB * TAN(PHI)
F14X = F14
F14Y = 0.0
C
F12 = FB / COS(PHI)
F12X = F12 * SIN(PHI)
F12Y = F12 * COS(PHI)

Appendix F. Force analysis subroutines

152
REATIONS OF MECHANISM ON BLOCK

\[ F_{41X} = F_{14X} \]
\[ F_{41Y} = F_{14Y} \]
\[ F_{21X} = F_{12X} \]
\[ F_{21Y} = F_{12Y} \]

RETURN
END
APPENDIX G. Integrated Mechanisms Program

REMARK:               ____________________________________________
REMARK:               ____________________________________________
REMARK: TITLE: FORCE ANALYSIS ON THE BRISTOL H-25 A COMPRESSOR
REMARK: SOURCE: INTEGRATED MECHANISMS PROGRAM
REMARK: WRITTEN BY: RAMAKANT P. ARCOT AND REGINALD G. MITCHENER
REMARK:               ____________________________________________

REMARK: ZERO(GRAVITY)=386.
UNIT(MASS)=.0025901
GROUND=CASE
REVO(CRNK,CR1)=A
REVO(CRNK,CR2)=B
REVO(CR1,P1)=C
REVO(CR2,P2)=D
PRISM(P1,CASE)=E
PRISM(P2,CASE)=F
REVO(CRNK,CASE)=MAIN
DATA:REVO(A)=0,.5,.2,1.25/0,.5,3/0,0,2,1.25/0,2.875,2,1.25
DATA:REVO(B)=0,-.5,4,2.5/0,-.5,5/0,0,4,2.5/0,1,875,4,2.5
DATA:REVO(C)=0,2,875,2,1.25/0,2.875,3/0,5,2,1.25/0,1,2,1.25
DATA:REVO(D)=0,1,875,4,2.5/0,1,875,5/0,0,4,2.5/0,1,4,250

REMARK: FOR THE ROTATION OF THE CRANK TO START IN A COUNTER
REMARK: CLO-WSIDE DIRECTION WITH THE PISTON IN THE BOTTOM
REMARK: CYLINDER AT ITS TDC, THE LOCAL X-AXIS OF THE CRNK &
REMARK: CASE ARE SELECTED TO LIE IN THE GLOBAL Y-DIRECTIONS

DATA:REVO(MAIN)=0,0,0,0,0,1,0,1,0,0,1,0

DATA:PRISM(E)=0,2,875,2,1.25/0,3,5,2,1.25/-1,2,875,2,1.25
DATA:PRISM(F)=0,1,875,4,2.5/-0,3,4,250/-1,1,875,4,250

REMARK: MASSES FOR THE TWO PISTONS AND CONNECTING RODS

DATA:MASS(P1,C)=.789/0,0,0
DATA:MASS(CR1,A)=.07686/0,00,00,4,9137

DATA:MASS(P2,D)=.789/0,0,0
DATA:MASS(CR2,B)=.07686/0,00,00,4,9137

REMARK: POINT DATA FOR GRAPHICS

REMARK: POINT DEFINITIONS FOR THE CRANKSHAFT, ROTOR

POINT(CRNK)=PT0,PT1,PT2,PT3,PT4,PT5,PT6,PT7,PT8,PT9
DATA:POINT(PT0,ABS)=0,0,0
DATA:POINT(PT1,ABS)=0,0,1,75
DATA:POINT(PT2,ABS)=0,.5,1,75
DATA:POINT(PT3,ABS)=0,.5,2,5
DATA:POINT(PT4,ABS)=0,.0,2,5

Appendix G. Integrated Mechanisms Program
DATA:POINT(PT5,ABS)=0,0,3.875
DATA:POINT(PT6,ABS)=0,-0.5,3.875
DATA:POINT(PT7,ABS)=0,-0.5,4.625
DATA:POINT(PT8,ABS)=0,0,4.625
DATA:POINT(PT9,ABS)=0,0,11.0

POINT(CRNK)=PT10,PT11,PT12,PT13,PT14,PT15,PT10
POINT(CRNK)=PT16,PT17,PT18,PT19,PT20,PT21,PT16
POINT(CRNK)=PT10,PT16
POINT(CRNK)=PT11,PT17
POINT(CRNK)=PT12,PT18
POINT(CRNK)=PT13,PT19
POINT(CRNK)=PT14,PT20
POINT(CRNK)=PT15,PT21

DATA:POINT(PT10,ABS)=0,-1.5,9.75
DATA:POINT(PT11,ABS)=1.3,-75,9.75
DATA:POINT(PT12,ABS)=1.3,75,9.75
DATA:POINT(PT13,ABS)=0,1.5,9.75
DATA:POINT(PT14,ABS)=1.3,75,9.75
DATA:POINT(PT15,ABS)=1.3,-75,9.75
DATA:POINT(PT16,ABS)=0,-1.5,5.75
DATA:POINT(PT17,ABS)=1.3,-75,5.75
DATA:POINT(PT18,ABS)=1.3,75,5.75
DATA:POINT(PT19,ABS)=0,1.5,5.75
DATA:POINT(PT20,ABS)=1.3,75,5.75
DATA:POINT(PT21,ABS)=1.3,-75,5.75

REMARK: POINT DEFINITIONS FOR THE CONNECTING RODS

POINT(CR1)=CRP1,CRP2
DATA:POINT(CRP1,ABS)=0,0.500,2.125
DATA:POINT(CRP2,ABS)=0,2.875,2.125

POINT(CR2)=CRP3,CRP4
DATA:POINT(CRP3,ABS)=0,-.500,4.25
DATA:POINT(CRP4,ABS)=0,1.875,4.25

REMARK: POINT DEFINITIONS FOR THE PISTONS

POINT(P1)=Q1,Q2,Q3,Q4,Q5,Q6,Q7
POINT(P1)=Q7,Q8,Q9,Q10,Q11,Q12,Q7
POINT(P1)=Q1,Q8
POINT(P1)=Q2,Q9
POINT(P1)=Q3,Q10
POINT(P1)=Q4,Q11
POINT(P1)=Q5,Q12
POINT(P1)=Q6,Q7
POINT(P1)=Q13
DATA:POINT(Q1,ABS)=0.9375,3.375,2.125
DATA:POINT(Q2,ABS)=0.46875,3.375,2.9369
DATA:POINT(Q3,ABS)=-0.46875,3.375,2.9369
DATA:POINT(Q4,ABS)=-0.9375,3.375,2.125

Appendix G. Integrated Mechanisms Program
DATA: POINT(Q5,ABS) = -0.46875, 3.375, 1.3131
DATA: POINT(Q6,ABS) = 0.46875, 3.375, 1.3131
DATA: POINT(Q8,ABS) = 0.9375, 2.375, 2.125
DATA: POINT(Q9,ABS) = 0.46875, 2.375, 2.9369
DATA: POINT(Q10,ABS) = -0.46875, 2.375, 2.9369
DATA: POINT(Q11,ABS) = -0.9375, 2.375, 2.125
DATA: POINT(Q12,ABS) = -0.46875, 2.375, 1.3131
DATA: POINT(Q7,ABS) = 0.46875, 2.375, 1.3131
DATA: POINT(Q13,ABS) = 0.0, 3.375, 2.125

POINT(P2) = R1, R2, R3, R4, R5, R6, R1
POINT(P2) = R7, R8, R9, R10, R11, R12, R7
POINT(P2) = R1, R8
POINT(P2) = R2, R9
POINT(P2) = R3, R10
POINT(P2) = R4, R11
POINT(P2) = R5, R12
POINT(P2) = R6, R7
POINT(P2) = R13
DATA: POINT(R1,ABS) = 0.9375, 2.375, 4.25
DATA: POINT(R2,ABS) = 0.46875, 2.375, 5.0619
DATA: POINT(R3,ABS) = -0.46875, 2.375, 5.0619
DATA: POINT(R4,ABS) = -0.9375, 2.375, 4.25
DATA: POINT(R5,ABS) = -0.46875, 2.375, 3.4381
DATA: POINT(R6,ABS) = 0.46875, 2.375, 3.4381
DATA: POINT(R8,ABS) = 0.9375, 1.375, 4.25
DATA: POINT(R9,ABS) = 0.46875, 1.375, 5.0619
DATA: POINT(R10,ABS) = -0.46875, 1.375, 5.0619
DATA: POINT(R11,ABS) = -0.9375, 1.375, 4.25
DATA: POINT(R12,ABS) = -0.46875, 1.375, 3.4381
DATA: POINT(R7,ABS) = 0.46875, 1.375, 3.4381
DATA: POINT(R13,ABS) = 0.0, 2.375, 4.25

REMARK: POINT DEFINITIONS FOR THE GROUND

POINT(CASE) = PC1, PC2, PC3, PC4, PC5
DATA: POINT(PC1,ABS) = 0, 2.375, 1.125
DATA: POINT(PC2,ABS) = 0, 3.375, 1.125
DATA: POINT(PC3,ABS) = 0, 3.375, 5.25
DATA: POINT(PC4,ABS) = 0, 2.375, 5.25
DATA: POINT(PC5,ABS) = 0, 1.375, 5.25

REMARK: GRAPHIC VISIBILITY AND COLOR SETTINGS

SHOW: RED = CRNK
SHOW: WHITE = CASE
SHOW: BLUE = CR1
SHOW: BLUE = CR2
SHOW: YELLOW = P1
SHOW: YELLOW = P2

REMARK: GRAPHIC COMMANDS

AXES (0) ALL

Appendix G. Integrated Mechanisms Program
ERASE

REMARK: TEKTRONIX 4010 DEVICE SPECIFICATION

DEVICE = T4010
ZOOM(0.25)
ZOOM(1.25)
DATA: VIEW=ISOMETRIC

REMARK: THET IS THE ANGLE OF ROTATION

DRAW
DATA: POSITION(MAIN) = 0,-4.90
DATA: VELOCITY(MAIN) = 381.26
VALUE(THET)=(0 - POSITION(MAIN,1))

REMARK: PRE1, PRE2 ARE PRESSURE FORCES ON P1,P2

VALUE(PRE1)=TABLE((THET)): PF1
VALUE(PRE2)=TABLE((THET)): PF2

DATA: FORCE(E)= PRE1,NONE
DATA: FORCE(F)= PRE2,NONE
PLOT FORCE(E,F,MAIN)
PRINT ON = FOR.DAT
LIST FORCE(MAIN)
EXEC

RETURN
APPENDIX H. Spring Stiffnesses - Finite Element Solution

The stiffness matrices for the top spring, the side spring and the shockloop were calculated by Ramani [10], by fixing one end of each spring to the ground and applying a unit force in each direction. The local stiffness matrices for the top spring and the side spring were as follows:

i) Top Spring Local Stiffness Matrix:

\[
K_{loc1} = \begin{bmatrix}
146.80 & 0.00 & -9.887 & 1.4924 & 91.652 & -8.3939 \\
0.00 & 145.95 & 0.00 & -91.12 & 0.5196 & 0.00 \\
-9.887 & 0.00 & 100.21 & 6.4368 & -6.1715 & -1.5608 \\
1.4924 & -91.12 & 6.4368 & 101.80 & 0.60743 & -0.0843 \\
91.652 & 0.5196 & -6.1715 & 6.0743 & 101.71 & -5.24 \\
-8.3939 & 0.00 & -1.5608 & -0.0843 & -5.24 & 51.27
\end{bmatrix}
\]

ii) Side Spring Local Stiffness Matrix:

\[
K_{loc2} = \begin{bmatrix}
498.61 & 1.0137 & -0.05893 & 5.0156 & -362.64 & -1.7522 \\
1.0137 & 503.77 & -43.404 & 366.23 & 0.70146 & -0.0193 \\
-0.05893 & -43.404 & 478.19 & -29.664 & 0.30572 & -6.192 \\
5.0156 & 366.23 & -29.664 & 411.66 & -2.57 & -0.0537 \\
-362.64 & 0.70146 & 0.30572 & -2.57 & 409.12 & 1.2743 \\
-1.7522 & -0.0193 & -6.192 & -0.0537 & 1.2743 & 166.01
\end{bmatrix}
\]

iii) Shockloop Local Stiffness Matrix:

\[
K_{loc3} = \begin{bmatrix}
66.57 & 18.98 & 14.14 & 90.10 & -499.10 & 210.80 \\
18.98 & 94.33 & 36.44 & 556.50 & -147.20 & 83.07 \\
14.14 & 36.44 & 79.15 & -17.66 & -116.00 & 50.75 \\
90.10 & 556.50 & -17.66 & 5224.0 & -663.60 & 439.70 \\
-499.10 & -147.20 & -116.00 & -663.60 & 4813.0 & -1587.0 \\
210.80 & 83.07 & 50.75 & 439.70 & -1587.0 & 1780.0
\end{bmatrix}
\]
These stiffness matrices were then applied in the Response Analysis Program BRESP.FOR, and using Appendix B to calculate the Element Stiffness Matrix for each spring, the Total Assembly Stiffness Matrix was calculated. The results from the program were as follows.

<table>
<thead>
<tr>
<th>SPRING #</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Stiffness Matrix =</td>
<td></td>
</tr>
<tr>
<td>146.80</td>
<td>.00000</td>
</tr>
<tr>
<td>.00000</td>
<td>145.95</td>
</tr>
<tr>
<td>-.98870</td>
<td>.00000</td>
</tr>
<tr>
<td>2.4071</td>
<td>-1.1761</td>
</tr>
<tr>
<td>1183.0</td>
<td>.51960</td>
</tr>
<tr>
<td>127.43</td>
<td>-1.5033</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPRING #</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Stiffness Matrix =</td>
<td></td>
</tr>
<tr>
<td>498.61</td>
<td>1.0137</td>
</tr>
<tr>
<td>1.0137</td>
<td>503.77</td>
</tr>
<tr>
<td>-.58930E-01</td>
<td>-43.404</td>
</tr>
<tr>
<td>7.5448</td>
<td>1627.0</td>
</tr>
<tr>
<td>-1599.1</td>
<td>-140.61</td>
</tr>
<tr>
<td>129.11</td>
<td>-1610.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPRING #</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Stiffness Matrix =</td>
<td></td>
</tr>
<tr>
<td>498.61</td>
<td>1.0137</td>
</tr>
<tr>
<td>1.0137</td>
<td>503.77</td>
</tr>
<tr>
<td>-.58930E-01</td>
<td>-43.404</td>
</tr>
<tr>
<td>7.5448</td>
<td>1627.0</td>
</tr>
<tr>
<td>-1598.7</td>
<td>136.09</td>
</tr>
<tr>
<td>135.57</td>
<td>1600.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPRING #</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Stiffness Matrix =</td>
<td></td>
</tr>
<tr>
<td>18.980</td>
<td>94.330</td>
</tr>
<tr>
<td>14.140</td>
<td>36.440</td>
</tr>
<tr>
<td>101.65</td>
<td>473.92</td>
</tr>
<tr>
<td>-350.19</td>
<td>-140.87</td>
</tr>
<tr>
<td>-43.463</td>
<td>109.69</td>
</tr>
</tbody>
</table>
Total Assembly Stiffness Matrix =

\[
\begin{array}{cccccc}
1210.6 & 21.007 & 13.033 & 119.15 & -2365.0 & 348.64 \\
21.007 & 1247.8 & -50.368 & 2551.7 & -144.87 & 98.319 \\
13.033 & -50.368 & 1135.7 & -398.39 & -165.43 & 18.871 \\
119.15 & 2551.7 & -398.39 & 21866. & -915.67 & 287.08 \\
-2365.0 & -144.87 & -165.43 & -915.67 & 30121. & -971.97 \\
348.64 & 98.319 & 18.871 & 287.08 & -971.97 & 13680.
\end{array}
\]

The Eigenanalysis was performed and the results were as follows:

```
EIGENVALUE FREQUENCIES
```

<table>
<thead>
<tr>
<th>RAD/SEC</th>
<th>HZ</th>
<th>RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>56.4621200</td>
<td>8.9862250</td>
<td>539.1735000</td>
</tr>
<tr>
<td>65.4485200</td>
<td>10.4164600</td>
<td>624.9874000</td>
</tr>
<tr>
<td>73.7725600</td>
<td>11.7412700</td>
<td>704.4760000</td>
</tr>
<tr>
<td>95.8142800</td>
<td>15.2493200</td>
<td>914.9589000</td>
</tr>
<tr>
<td>97.0028200</td>
<td>15.4384800</td>
<td>926.3087000</td>
</tr>
<tr>
<td>109.5322000</td>
<td>17.4325900</td>
<td>1045.9550000</td>
</tr>
</tbody>
</table>
```
APPENDIX I. Orbit Animation Program BR_ANIM.C

#include <stdio.h>
#include <math.h>
#include "home/aroc/tocfcl/pass.h"
#include <afanc.h>
#include <ctype.h>

/**********************************************************
 TITLE: BRISTOL H-25 A COMPRESSOR SYSTEM ANIMATION
 WORKSTATION: RISC/6000 WITH PHIGS INTERFACE & UNIX ENVIRONMENT
 AUTHOR: RAMAKANT P. ARCOT
 WRITTEN AT: VIRGINIA TECH C.A.D. LABORATORY (M.E. DEPT)
 SOURCE CODE: BR_ANIM.C
**********************************************************/

#define WORKSTATION_ID (Pint) 1 /* Workstation identifier */
#define MAIN_MENU (Pint) 1 /* Main Menu structure id */
#define GEOM_MENU (Pint) 2 /* Geometry menu structure id */
#define ANAL_MENU (Pint) 3 /* Analysis menu structure id */

static Ppoint outer_border[4] = {{-1,1},{-1,0.9},{1,0.9},{1,1}}; /* outer border*/
static Ppoint outer_border1[4] = {{0,0.83},{0,0.73},{0.46,0.73},{0.46,0.83}};
static Ppoint inner_border[4] =
    {{-1,9}, {-1,75}, {-293,75}, {-293,9}}; 
static Ppoint inner_border2[4] =
    {{-293,75}, {-293,9}, {35,9}, {35,75}};
static Ppoint inner_border3[4] =
    {{35,75}, {35,9}, {1,9}, {1,75}};
static Ppoint Geom_menu_border[4] = {{-1,9}, {-1,-75}, {1,-75}, {1,9}}; /* Geom Menu border */
static Ppoint Main_menu_border[4] = {{-1,9}, {-1,-75}, {1,75}, {1,9}}; /* Main menu border */
static Ppoint an_border[4] =
    {{-1,9}, {-1,75}, {0,-0.9}}; 
static Ppoint an_border2[4] =
    {{0,9}, {0,75}, {1,0,75}, {1,0,9}}; /* Analysis Menu Border */

static Ppoint Pan_menu_border[4] =
    {{-1,-1}, {-1,1}, {1,1}, {1,-1}}; /* Pan Menu border */

static Ppoint_list inbls1 = {4,inner_border1};
static Ppoint_list inbls2 = {4,inner_border2};
static Ppoint_list inbls3 = {4,inner_border3};
static Ppoint_list_list inbls1s1 = {1,&inbls1};
static Ppoint_list_list inbls2s2 = {1,&inbls2};
static Ppoint_list_list inbls3s3 = {1,&inbls3};
static Ppoint_list an1 = \{4,an\_border1\};
static Ppoint_list an2 = \{4,an\_border2\};
static Ppoint_list_list anew1 = \{1,\&an1\};
static Ppoint_list_list anew2 = \{1,\&an2\};

static float disp\_scale1 = 1.0;    // Scaling factor for displacement */
static float disp\_scale2 = 2.0;    // Scaling factor for rotations */
static float case\_len = .55;       // Length of the crank case */
static float cyl\_dia = .20;        // Piston cylinder diameter */
static float rotor\_len = .90;       // Length of the rotor */
static float time = 100.0;           // Time in sec for oscillation */
static int no\_increments = 100;     // Number of divisions for time interval */
static float no\_repeats = 5;        // Number of repeats */

static Pvec3 view\_up\_vec, view\_plane\_normal;    // View up vector, view plane normal */
static Ppoint3 view\_ref\_pt;                      // View reference point */
static Pview\_rep3 view\_rep1, view\_rep2;           // View representation matrices */
static Pview\_map3 view\_map;                      // View mapping matrix */
static Pint error;

static Ppoint3 fixed\_pt1, fixed\_pt2;
static float angle, alpha, distance;    // alpha = angle between Conn rod & horizontal */
static char junk[2];

/* Points for web */

static Ppoint3 web\_points[5] =
{ {0.4,0.0,0.4}, {-0.4,0.0,0.4}, {0.4,0.0,-0.4}, {0.4,0.0,0.4} };

/* Points for crank case faces */

static Ppoint3 face1[5] =
{ {-225,-55,3}, {425,-55,3}, {425,0,3}, {-225,0,3}, {-225,-55,3} };
static Ppoint3 face2[5] =
{ {425,-55,3}, {425,0,0,3}, {425,0,-3}, {425,-55,3}, {425,-55,3} };
static Ppoint3 face3[5] =
{ {425,0,-3}, {425,-55,-3}, {-225,-55,-3}, {-225,0,-3}, {425,0,-3} };
static Ppoint3 face4[5] =
{ {-225,-55,-3}, {-225,0,-3}, {-225,0,3}, {-225,-55,3}, {-225,-55,-3} };
static Ppoint3 face5[5] =
{ {-225,-55,-3}, {-225,55,3}, {425,-55,3}, {425,-55,-3}, {-225,-55,-3} };
static Ppoint3 face6[5] =
{ {-225,0,-3}, {-225,0,3}, {425,0,3}, {425,0,-3}, {-225,0,-3} };

/* Points for Spring supports (hatched) */

static Ppoint3 Top\_Spring\_Sup[4] =

Appendix 1  Orbit Animation Program BR_ANIM.C  162
static Ppoint3 Side_Spring1_Sup[4] =
static Ppoint3 Side_Spring2_Sup[4] =

/* Points for spring ends */
static Ppoint3 Top_Spring[2] =
    { {0.1,1.0}, {0.95,0.0} };
static Ppoint3 Side_Spring1[2] =
    { {-0.3,-0.05,-3.25}, {-0.3,-3.325} };
static Ppoint3 Side_Spring2[2] =
    { {-0.3,-0.05,-3.25}, {-0.3,-3.325} };

static Ppoint3 X_AXES[2] =
    { {-0.25, 0.172, 0.00}, {0.475, 0.172, 0.00} };
static Ppoint3 Y_AXES[2] =
    { {-0.25, 0.172, 0.00}, {-0.25, 0.672, 0.00} };
static Ppoint3 Z_AXES[2] =
    { {-0.25, 0.172, 0.00}, {-0.25, 0.172, 0.50} };

/* Co-efficients of cosine terms for the first 40 harmonics of the displacement vector */
static double Disp_Vec_Cos[41][6] =
    { { .0030989, .0124486, .0000286, .0034858, .0006226, .0003075 },
      { .0000760, .0000690, .0000025, .0006327, .0002660, .0001702 },
      { -.0014612, -.0036527, -.0000195, -.0010206, .0005168, .0001484 },
      { -.0000275, -.0000333, -.0000023, -.0000006, .0002068, .0001323 },
      { -.0015377, -.0006187, -.0000126, -.0001709, .0005139, .0002401 },
      { -.0000136, -.0000111, -.0000002, -.0001478, .0000311, .0000200 },
      { -.0000168, -.0028614, -.0000056, -.0008009, .0000163, .0000437 },
      { -.0000111, -.0000072, -.0000001, -.0001764, -.0000012, -.0000001 },
      { -.0002201, -.0016276, -.0000050, -.0004552, .0000859, .0000457 },
      { -.0000073, -.0000067, -.0000002, -.0000593, .0000262, .0000166 },
      { -.0002343, -.0007467, -.0000033, -.0002087, .0000835, .0000406 },
      { -.0000042, -.0000032, -.0000000, -.0000567, .0000049, .0000036 },
      { -.0000525, -.0008978, -.0000014, -.0002514, -.0000105, -.0000048 },
      { -.0000025, -.0000013, -.0000001, -.0000507, -.0000053, -.0000031 },
      { -.0000042, -.0003654, -.0000008, -.0001023, .0000043, .0000034 },
      { -.0000008, -.0000009, -.0000000, -.0000051, .0000042, .0000029 },
      { -.0000358, -.0000062, -.0000003, -.0000017, .0000012, .0000042 },
      { -.0000002, -.0000000, -.0000000, -.0000042, -.0000010, -.0000010 },
      { -.0000608, -.0001511, -.0000002, -.0000424, -.0000190, -.0000061 },
      { -.0000001, -.0000004, -.0000001, -.0000101, -.0000047, -.0000036 },
      { -.0000178, -.0000366, -.0000001, -.0000103, -.0000056, -.0000027 },
      { -.0000091, -.0000001, -.0000000, -.0000033, .0000014, .0000012 },
      { -.0000243, -.0000519, -.0000001, -.0000145, .0000077, .0000022 },
      { -.0000001, -.0000001, -.0000000, -.0000007, .0000005, .0000003 }
  };

Appendix I. Orbit Animation Program BR_ANIM.C
/* Co-efficients of sine terms for the first 40 harmonics of the displacement vector */

static double Disp_Vec_Sin[41][6] = {
    { .000000,  -.000000,  -.000000,  -.000000,  -.000000,  -.000000 },
    { .000029,  -.000097,  .000000,  -.000027,  -.000059,  -.000012 },
    { -.000003,  -.000001,  .000000,  -.000083,  -.000018,  -.000010 },
    { .000008,  -.000069,  -.000001,  -.000194,  .000003,  .000005 },
    { -.000003,  -.000003,  .000000,  -.000008,  .000014,  .000007 },
    { -.000023,  -.000105,  -.000002,  -.000030,  .000078,  .000028 },
    { -.000001,  -.000002,  .000000,  -.000009,  .000008,  .000016 },
    { -.000114,  -.000053,  .000000,  -.000148,  -.000034,  -.000006 },
    { .000000,  .000001,  .000000,  -.000038,  -.000014,  -.000007 },
    { .000087,  -.000177,  .000000,  -.000050,  -.000027,  -.000011 },
    { .000001,  .000000,  .000000,  .000017,  .000002,  .000003 },
    { -.000009,  -.000528,  .000000,  -.000148,  .000027,  .000004 },
    { .000000,  .000001,  .000000,  .000027,  .000002,  .000000 },
    { .000011,  .000019,  .000000,  .000005,  .000039,  .000014 },
    { -.000001,  .000000,  .000000,  -.000014,  -.000011,  -.000015 },
    { .000001,  .000003,  .000000,  .000010,  .000050,  .000012 },
    { .080000,  .000000,  .000000,  .000011,  .000001,  .000002 },
    { -.000007,  -.000298,  .000000,  .000084,  .000030,  .000008 }
};
void Open_Phigs_Ws(); /* Open PHIGS & the Workstation */
void Color_Table(); /* Load the Color Table */
void Create_Main_Menu(); /* Create Main Menu */
void Create_Anal_Menu(); /* Create Analysis menu */
void Pick_Main_Menu(); /* Pick Main Menu */
void Pick_Anal_Menu(); /* Pick Analysis Menu */
void Create_Geom_Menu(); /* Create Geometry Menu */

void Create_Compressor(); /* Create Compressor Frame */
void Repeat_Animation(); /* Animate Compressor */
void handle_input(); /* Pan around compressor frame */
void Close_Phigs_Ws(); /* Close PHIGS & Workstation */

#define GEOMETRY (Pint) 10
#define ANALYSIS (Pint) 11 /* Pick id's for geometry Menu */
#define EXIT (Pint) 12
#define PICK_ID (Pint) 100

main()
{
    Open_Phigs_Ws(); /* Open PHIGS & Workstation */
    Color_Table(); /* Load the color table */
    Create_Main_Menu(); /* Create Main Menu */
    Close_Phigs_Ws(); /* Close Phigs & the Workstation */
}

/* Function to set the view index */

static int
set_view( ws_id, index,vrp,vpn,vup, view_map)
Pint ws_id;
Pint index;
Ppoint3 *vrp;
Pvector3 *vpn;
Pvector3 *vup;
Pview_map3 *view_map;

Appendix I. Orbit Animation Program BR_ANIM.C 165
{  
Pint error;
Pview_rep3 view;

peval_view_ori_matrix3 (vrp,vpn,vup,&error, view_ori_matrix);
if(error) return error;

peval_view_map_matrix3 (view_map, &error, view.map_matrix);
if(error) return error;

view.xy_clip = PIND_CLIP;
view.front_clip = view.back_clip = PIND_CLIP;
view.clip_limit = view_map->proj_vp;
pset_view_rep3 (ws_id, index, &view);
return 0;
}

/* Projection viewport points definition */

static Plimit3 vports[] = {
  /* x_min, x_max, y_min, y_max, z_min, z_max */
  { 0.00, 0.75, 0, 1.00, 0, 1 },
  { 0.75, 1.00, 0, 0.33, 0, 1 },
  { 0.75, 1.00, 0.33, 0.66, 0, 1 },
  { 0.75, 1.00, 0.66, 1.0, 0, 1 }
};

/* Port definitions for boundaries */

static Plimit3 ports[] = {
  /* x_min, x_max, y_min, y_max, z_min, z_max */
  { -1.0, 1.0, 0.33, 1, 0, 1 },
  { 0.5, -1, 1.0, 0, 1 },
  { 0.5, 1.0, -1, -0.33, 0, 1 },
  { 0.5, 1.0, -0.33, 0.33, 0, 1 }
};

void Open_Phigs_Ws()
{

popen_phigs("",0);

GPSS (1);
GPCRWS (WORKSTATION_ID,1,1,"","X",0);
GPASSW (WORKSTATION_ID,1);
}

Appendix I. Orbit Animation Program BR_ANIM.C
void Close_Phigs_Ws()
{
    pclose_ws (WORKSTATION_ID);
    pclose_phigs();
}

void Create_Main_Menu()
{
    Point_list contour[1];
    Point_list_list set;
    Point text;
    Pint names[1];
    Pint_list name_list;
    char junk[2];

    popca_struct(MAIN_MENU);
    /* Build the contour list */

    contour[0].num_points = 4;
    contour[0].points = outer_border;
    set.num_point_lists = 1; set.point_lists = contour;

    name_list.ints = names;
    name_list.num_ints = 1;

    names[0] = MAIN_MENU;    /* Create Names Set */
    padd_names_set (&name_list);

    pset_edge_flag( PEDGE_ON );    /* Set edge flag on */
    pset_int_style (PSTYLE_SOLID );    /* Set interior style solid */
    pset_int_colr_ind( (Pint) 7);    /* Set interior color index */
    pfill_area_set (&set);    /* Fill area set */
    pset_char_expan ( (Pfloat).42);    /* Set character expansion */
    pset_text_prec (PPREC_STROKE);    /* Set text precision */
    pset_text_font ( (Pint) 5);    /* Set text font */
    pset_char_hl ( (Pfloat) .07);    /* Set character height */
    pset_char_space ( (Pfloat) .75);    /* Set character space */
    pset_text_colr_ind( (Pint) 6);    /* Set text color index */
    text.x = -.3256;    /* Text coordinates */
    text.y = .9135;
    ptext(&text, "MAIN MENU");

    pset_edge_flag( PEDGE_ON );
    pset_int_style (PSTYLE_SOLID );
    pset_int_colr_ind( (Pint) 43);
    contour[0].num_points = 4;
    contour[0].points = Main_menu_border;
    pfill_area_set (&set);
    pset_char_expan ( (Pfloat) 1.42);
pset_text_prec (PPREC_STROKE);
pset_text_font ( (Pint) 8);
pset_char_hit ( (Pfloat) .05);
pset_char_space ( (Pfloat) .48);
text.x = -.95;
text.y = .825;
pset_text_colr_ind ( (Pint) 29);

pset_pick_id (GEOMETRY);          /* Set pick id */
pset_edge_flag (PEDGE_ON);
pfill_area_set (&inblst1);
ptext(&text, "1. GEOMETRY");


text.x = -.25;

pset_pick_id (ANALYSIS);          /* Set pick id */
pset_edge_flag (PEDGE_ON);
pfill_area_set (&inblst2);
ptext(&text, "2. ANALYSIS");


text.x = .5;

pset_pick_id (EXIT);              /* Set pick id */
pset_edge_flag (PEDGE_ON);
pfill_area_set (&inblst3);
ptext(&text, "3. EXIT");

pclose_struct();

popen_struct (PICK_ID);           /* open structure */
names[0] = PICK_ID;
name_list.num_ints = 1;
padd_names_set (&name_list);
pexec_struct (MAIN_MENU);
pclose_struct();                  /* close structure */

ppost_struct( WORKSTATION_ID, PICK_ID, (Pfloat) 1.0);/* post structure */
pupd_ws (WORKSTATION_ID,PFLAG_PERFORM); /* update workstation */

Pick_Main_Menu();              /* pick main menu */

}
#define DEVICE (Pint) 1 /* device number */
#define TIMEOUT (Pfloat) 30.0 /* timeout for pick event(s) */

void Pick_Main_Menu()
{

  PfILTER  filter;  /* pick filter */
Pint inclusion[1],ws_id,device_num;
Pin_status status; /* pick status*/
PPick_path path;  /* pick path */
PPick_path_elem path_elems[2];
int i;
Pin_class class;    /* pick class */

/* Set the pick filter */
filter.incl_set.num_ints = 1;
filter.incl_set.ints = inclusion;
inclusion[0] = PICK_ID;

filter.excl_set.num_ints = 0;
filter.excl_set.ints = (Pint *)NULL;

/* set the pick filter */
pset_pick_filter( WORKSTATION_ID, DEVICE, &filter );

path.path_list = path_elems;

/* set the pick device in request mode */

  pset_pick_mode (WORKSTATION_ID,(Pint) 1, POP_REQ, PSWITCH_ECHO);

    preq_pick (WORKSTATION_ID, DEVICE, (Pint) 2, &status, &path);

pupd_ws (WORKSTATION_ID, PFLAG_PERFORM );

if ( status == PIN_STATUS_OK )
{

  /* options for pick */

    switch (path.path_list[1].pick_id )
    {

Appendix I  Orbit Animation Program BR_ANIM.C
case GEOMETRY: {punpost_struct(WORKSTATION_ID,PICK_ID);
    punpost_struct (WORKSTATION_ID, GEOM_MENU);
    Create_Geom_Menu(); break;}
case ANALYSIS: {punpost_struct (WORKSTATION_ID, PICK_ID);
    Create_Anal_Menu();
    break;}
case EXIT: i = 0; break;
}
void Create_Geom_Menu()
{
    Ppoint_list contour[1];
    Ppoint_list_list set;
    Pint names[1],pet,ws_type,error;
    Pint_list name_list;
    Plimit echo_area;
    Ppoint text;
    Pstring_data G_record; /* data record */
    Pint status;
    Pdisp_space_size dc_info;
    float crank_rad1,conn_rod_len1,piston_dia1,offset1;
    float temp1,temap2,temp3,temp4;
    
    char junk[2], init_string[25], cr[10], crl[10], pdia[10], off[10];
    char tmp1[10], tmp2[10], tmp3[10], tmp4[10];
    punpost_struct(WORKSTATION_ID, GEOM_MENU);
    pupd_ws(WORKSTATION_ID, PFLAG_PERFORM);
    popen_struct(GEOM_MENU);
    
    name_list.ints = names;
    name_list.num_ints = 1;
    names[0] = GEOM_MENU;
    padd_names_set(&name_list);
    
    contour[0].num_points = 4;
    contour[0].points = outer_border;
    set.num_point_lists = 1; set.point_lists = contour;
    pset_edge_flag( PEDGE_ON );
    pset_int_style( PSTYLE_SOLID ),
    pset_int_colr_ind( (Pint)7);
    pfil_area_set(&set);
    
    pset_char_expan( (Pfloat) .42);
    pset_text_prec (PPREC_STROKE);
    pset_text_font ( (Pint) 5);
    pset_char_hi((Pfloat) .07);
    pset_char_space ((Pfloat) .75);
    pset_text_colr_ind( (Pint) 6);
}

Appendix I. Orbit Animation Program BR_ANIM.C 170
text.x = -.53256;
text.y = .9135;
ptext(&text, "GEOMETRY MENU");

pset_edge_flag( PEDGE_ON );
pset_int_style( PSTYLE_SOLID );
pset_int_colr_ind( (Pint) 31);
contour[0].num_points = 4;
contour[9].points = Geom_menu_border;
pfill_area_set (&set);

pset_char_expan ( (Pfloat) 1.42);
pset_text_prec (PPREC_STROKE);
pset_text_font ( (Pint) 16);
pset_char_hl( (Pfloat) .03);
pset_char_space( (Pfloat) .48);
text.x = -.95;
text.y = .68875; /* Create Geometry Menu Text in */
pset_text_colr_ind( (Pint) 12);
ptext(&text, "Displacement Scale (1)"); /* Text coordinates */
text.x = -.95;
text.y = .325;
ptext(&text, "Scale for Rotation (2)");

text.x = -.95;
text.y = -.03875;
ptext(&text, "No. of increments (100)");

text.x = -.95;
text.y = -.4025;
ptext(&text, "No. of repeats (5)");

ppost_struct( WORKSTATION_ID, GEOM_MENU, (Pfloat) 1.0);
pupd_ws (WORKSTATION_ID, PFLAG_PERFORM);

ws_type = 1;

strcpy (init_string, "");
echo_area.x_min = .1982267100;
echo_area.x_max = .30; /* echo area in device coordiantes */
echo_area.y_min = .24;
echo_area.y_max = .250;

G_record.in_buf_size = 10; /* buffer size */
G_record.init_pos = 1; /* initial position of record */

pinit_string (WORKSTATION_ID, DEVICE, &init_string, /* Initialise string device */
(Pint) 1, &echo_area, &G_record);
preq_string (WORKSTATION_ID, DEVICE, &status, tmp1); /* Request string device */
temp1 = atof(tmp1); /* char to float */

if (temp1 != 0) disp_scale1 = temp1;

text.x = .404554348;
text.y = .68875;
pset_text_colr_ind((Pint) 4);
ptext(&text, &tmp1);
pupil_ws (WORKSTATION_ID, PFLAG_PERFORM);

strcpy (init_string, "\n");

echo_area.x_min = .1982267100;
echo_area.x_max = .300;
echo_area.y_min = .188304959;
echo_area.y_max = .21;

G_record.in_buf_size = 10,
G_record.init_pos = 1;

pinit_string (WORKSTATION_ID, DEVICE, &init_string,
(Pint) 1, &echo_area, &G_record);

preq_string (WORKSTATION_ID, DEVICE, &status, tmp2);

temp2 = atof(tmp2);

if (temp2 != 0) disp_scale2 = temp2;

text.x = .404554348;
text.y = .325;
pset_text_colr_ind((Pint) 4);
ptext(&text, &tmp2);
pupil_ws (WORKSTATION_ID, PFLAG_PERFORM);

strcpy (init_string, "\n");

echo_area.x_min = .1982267100;
echo_area.x_max = .300;
echo_area.y_min = .136609919;
echo_area.y_max = .16;

pinit_string (WORKSTATION_ID, DEVICE, &init_string,
(Pint) 1, &echo_area, &G_record);

preq_string (WORKSTATION_ID, DEVICE, &status, tmp3);

temp3 = atof(tmp3);

if (temp3 != 0) no_increments = temp3;

text.x = .404554348;
text.y = -.03875;

Appendix I. Orbit Animation Program BR_ANIM.C 172
pset_text_colr_ind( (Pint) 4);
ptext(&text, &tmp3);
pupd_ws (WORKSTATION_ID, PFLAG_PERFORM);

strcpy (init_string, """);

echo_area.x_min = .1982267100;
echo_area.x_max = .300;
echo_area.y_min = .084914877;
echo_area.y_max = .12300;

pinit_string (WORKSTATION_ID, DEVICE, &init_string,
(Pint) 1, &echo_area, &G_record);

preq_string (WORKSTATION_ID, DEVICE, &status, tmp4);

temp4 = atof(tmp4);

if (temp4 != 0) no_repeats = temp4;

text.x = .404554348;
text.y = -.4025;

pset_text_colr_ind( (Pint) 4);
ptext(&text, &tmp4);
pupd_ws (WORKSTATION_ID, PFLAG_PERFORM);

if (status == PIN_STATUS_OK)
{
printf ("Displacement Scaling Factor = %3.1f\n", disp_scale1);
printf ("Rotational Scaling Factor = %3.1f\n", disp_scale2);
printf ("Number of time increments = %3d\n", no_increments);
printf ("Number of cycles to repeat = %3.1f\n", no_repeats);
pclose_struct();
}

punpost_struct (WORKSTATION_ID, GEOM_MENU);       /* unpost geometry menu */
Create_Main_Menu();
}

#define ANIMATION (Pint) 31
#define ANALRET (Pint) 32

void Create_Anal_Menu ()
{

Pint names[1];
Pint_list name_list;
Ppoint text;
char junk[2];
Ppoint_list_list set;

Appendix I. Orbit Animation Program BR_ANIM.C
Ppoint_list contour[1];

popen_struct(ANAL_MENU);

name_list.ints = names;
name_list.num_ints = 1;
names[0] = ANAL_MENU;
padd_names_set (&name_list);

contour[0].num_points = 4;
contour[0].points = outer_border;
set.num_point_lists = 1; set.point_lists = contour;
pset_edge_flag(PEDGE_ON);
pset_int_style(PSTYLE_SOLID);
pset_int_colr_ind((Pint) 7);
pfill_area_set (&set);
pset_char_expan((Pfloat) .42);
pset_text_prec(PPREC_STROKE);
pset_text_font((Pint) 5);
pset_char_hlt((Pfloat) .07);
pset_char_space((Pfloat) .75);
pset_text_colr_ind((Pint) 6);
text.x = -.53256;
text.y = .9135;
ptext(&text, "ANALYSIS MENU");

pset_edge_flag(PEDGE_ON);
pset_int_style(PSTYLE_SOLID);
pset_int_colr_ind((Pint) 3);
contour[0].num_points = 4;
contour[0].points = Main_menu_border;
pfill_area_set (&set);
pset_char_expan((Pfloat) 1.42);
pset_text_prec(PPREC_STROKE);
pset_text_font((Pint) 8);
pset_char_hlt((Pfloat) .03);
pset_char_space((Pfloat) .48);
text.y = .825;
pset_text_colr_ind((Pint) 4);

text.x = -.8;

pset_pick_id(ANIMATION); /* set pick id */
pfill_area_set (&anal1);
ptext(&text, "1. ANIMATION");

text.x = .3;

pset_pick_id(ANAL_RET); /* set pick id */
pfill_area_set (&anal2);
ptext(&text, "2. RETURN");
pcloseStruct();

pnumStruct(WORKSTATION_ID, ANAL_MENU, (Pf float) 1.0);
pupd_ws(WORKSTATION_ID, PFLAG_PERFORM);

Pick_Anal_Menu();

}
#define SPRING (Pint) 50 /* structure id for spring & supports */
#define WEB (Pint) 51 /* structure id for web */
#define ROTOR (Pint) 52 /* structure id for rotor */
#define CCASE (Pint) 53 /* structure id for crank-case */
#define AXES (Pint) 54
#define SYSTEM (Pint) 55 /* structure id for system */
#define VIEW1 (Pint) 90 /* structure id for animation */
#define VIEW2 (Pint) 91 /* structure id for vel curve */

void Pick_Anal_Menu ()
{
    Pfilter filter;
Pint inclusion[1], ws_id, device_num;
Pin_status status;
Ppick_path path;
Ppick_path_elem path elems[1];
int i;
Pin_class class;

    /* Set the pick filter */
    filter.incl_set.num_ints = 1;
    filter.incl_set ints = inclusion;
    inclusion[0] = ANAL_MENU;
    filter.excl_set.num_ints = 0;
    filter.excl_set ints = (Pint *) NULL;

    pset_pick_filter(WORKSTATION_ID, DEVICE, &filter);

    path. path_list = path elems;
    /* set pick device in request mode */
    pset_pick_mode(WORKSTATION_ID, (Pint) 1, POP_REQ, PSWITCH_ECHO);
    preq_pick(WORKSTATION_ID, DEVICE, (Pint) 1, &status, &path);

    /* check if status is OK */
    if ( status == PIN_STATUS_OK )

{
punpost_struct(WORKSTATION_ID, ANAL_MENU);
switch (path.path_list[0].pick_id)
{
  case ANIMATION: 
    Create_Compressor();
    punpost_struct(WORKSTATION_ID, VIEW1);
    Create_Anal_Menu();
    break;
  }
  case ANAL_RET: 
    {
      punpost_struct(WORKSTATION_ID, GEOM_MENU);
      Create_Main_Menu(); break;
    }
}
}

#define PAN_SCALE (M_PI/500) /* scale for panning */
#define MODEL (Pint) 61 /* structure id for pan model */
#define PAN_MENU (Pint) 99 /* structure id for pan menu */
#define PANNING   (Pint) 98 /* pick id for starting pan */
#define PANEX     (Pint) 97 /* pick id for exiting pan */
#define PAN_RET    (Pint) 96 /* pick id for returning to analysis menu */
#define NUM_INCREMENT  40 /* number of increments */
#define NUM_CIRCLE_FACETS 32 /* number of points for a circle */

void Create_Compressor()
{

  Ppoint_list_list3 sets;
Ppoint_list3 web, rot1, rot2, rotsid[6], contours[8];
Ppoint3 rot1_points[7], rot2_points[7];
Ppoint3 rts1[4], rts2[4], rts3[4], rts4[4], rts5[4], rts6[4];
Ppoint3 cyl1[41], cyl2[41];
Ppoint_list3 point_list3;

  int i, j, k;
  char junk[2];
  float alpha;
  double delta, ang;

  Pint names[1];
Pint_list name_list;
Ppoint3 text3;

Appendix I. Orbit Animation Program BR_ANIM.C

  176
Pvec3 text_dir[2];
Ppoint text;
Ppoint_list_list set;

Pfloat angle = 0.0, alpha1, alpha2, incr = ((2*M_PI) / NUM_INCREMENTS), dist = 0.0;
Ppoint3 fixed_pt;
Pmatrix3 xform, xform1, global, translate, sc;
Point err;
Pvec3 shift, scale;
Ppoint_list contour[1];

int lastx, lasty, doae = 0;
float theta = M_PI/4, phi = 0.0706;
float x, y;
PPoint locator_position; /* locator position */
int num_moves;
Point ws_id, device_num, view;
Ppick_path path;
Pin_class class;
Pin_status status;
Pupd_st upd_st;
Point error, index;
Pfilter filter;
Point inclusion[1];
Ppick_path_elem path elems[1];
Ppoint_list point_list;
Ppoint pts[5];
Ppoint3 anno_text_reftp;
Pvec3 anno_offset;

/* Web point definitions */
web.points = web_points;
web.num_points = 5;

/* Rotor point definitions */
rot1.points = rot1_points;
rot2.points = rot2_points;
rot1.num_points = rot2.num_points = 7;

rotside[0].points = rts1;
rotside[0].num_points = 4;
rotside[1].points = rts2;
rotside[1].num_points = 4;
rotside[2].points = rts3;
rotside[2].num_points = 4;
rotside[3].points = rts4;
rotside[3].num_points = 4;
rotside[4].points = rts5;
rotside[4].num_points = 4;
rotside[5].points = rts6;
rotside[5].num_points = 4;

/* Crank case definitions */

sets.num_point_lists = 6;
sets.point_lists = contours;

contours[0].num_points = 5;
contours[0].points = face1;
contours[1].num_points = 5;
contours[1].points = face2;
contours[2].num_points = 5;
contours[2].points = face3;
contours[3].num_points = 5;
contours[3].points = face4;
contours[4].num_points = 5;
contours[4].points = face5;
contours[5].num_points = 5;
contours[5].points = face6;
contours[6].num_points = 41;
contours[6].points = cyl1;
contours[7].num_points = 41;
contours[7].points = cyl2;

/* Calculation of points */

/* Rotor top & bottom (Hexagonal) */

angle = 0; incr = (2 * M_PI) / 6;
for (i = 0; i < 6; i++)
{
    rot1_points[i].x = rot2_points[i].x = 0.30*cos(angle);
    rot1_points[i].y = 0.0; rot2_points[i].y = .95;
    rot1_points[i].z = rot2_points[i].z = 0.30*sin(angle);
    angle = angle + incr;
}

rot1_points[6].x = rot2_points[6].x = rot1_points[0].x;
rot1_points[6].z = rot2_points[6].z = rot1_points[0].z;
rot1_points[6].y = rot1_points[0].y;
rot2_points[6].y = rot2_points[0].y;

/* Points for rotor side (Hexagonal) */

rts1[0].x = rot1_points[0].x;
rts1[0].y = rot1_points[0].y;
rts1[0].z = rot1_points[0].z;

rts1[1].x = rot1_points[1].x;
rts1[1].y = rot1_points[1].y;
rts1[1].z = rot1_points[1].z;

rts1[2].x = rot2_points[1].x;
rts1[2].y = rot2_points[1].y;
rts1[2].z = rot2_points[1].z;

rts1[3].x = rot2_points[0].x;
rts1[3].y = rot2_points[0].y;
rts1[3].z = rot2_points[0].z;

rts2[0].x = rot1_points[1].x;
rts2[0].y = rot1_points[1].y;
rts2[0].z = rot1_points[1].z;

rts2[1].x = rot1_points[2].x;
rts2[1].y = rot1_points[2].y;
rts2[1].z = rot1_points[2].z;

rts2[2].x = rot2_points[2].x;
rts2[2].y = rot2_points[2].y;
rts2[2].z = rot2_points[2].z;

rts2[3].x = rot2_points[1].x;
rts2[3].y = rot2_points[1].y;
rts2[3].z = rot2_points[1].z;

rts3[0].x = rot1_points[2].x;
rts3[0].y = rot1_points[2].y;
rts3[0].z = rot1_points[2].z;

rts3[1].x = rot1_points[3].x;
rts3[1].y = rot1_points[3].y;
rts3[1].z = rot1_points[3].z;

rts3[2].x = rot2_points[3].x;
rts3[2].y = rot2_points[3].y;
rts3[2].z = rot2_points[3].z;

rts3[3].x = rot2_points[2].x;
rts3[3].y = rot2_points[2].y;
rts3[3].z = rot2_points[2].z;

rts4[0].x = rot1_points[3].x;
rts4[0].y = rot1_points[3].y;
rts4[0].z = rot1_points[3].z;

rts4[1].x = rot1_points[4].x;
rts4[1].y = rot1_points[4].y;
rts4[1].z = rot1_points[4].z;

rts4[2].x = rot2_points[4].x;
rts4[2].y = rot2_points[4].y;
rts4[2].z = rot2_points[4].z;

rts4[3].x = rot2_points[3].x;
rts4[3].y = rot2_points[3].y;
rts4[3].z = rot2_points[3].z;

rts5[0].x = rot1_points[4].x;
rts5[0].y = rot1_points[4].y;
rts5[0].z = rot1_points[4].z;

rts5[1].x = rot1_points[5].x;
rts5[1].y = rot1_points[5].y;
rts5[1].z = rot1_points[5].z;

rts5[2].x = rot2_points[5].x;
rts5[2].y = rot2_points[5].y;
rts5[2].z = rot2_points[5].z;

rts5[3].x = rot2_points[4].x;
rts5[3].y = rot2_points[4].y;
rts5[3].z = rot2_points[4].z;

rts6[0].x = rot1_points[5].x;
rts6[0].y = rot1_points[5].y;
rts6[0].z = rot1_points[5].z;

rts6[1].x = rot1_points[6].x;
rts6[1].y = rot1_points[6].y;
rts6[1].z = rot1_points[6].z;

rts6[2].x = rot2_points[6].x;
rts6[2].y = rot2_points[6].y;
rts6[2].z = rot2_points[6].z;

rts6[3].x = rot2_points[5].x;
rts6[3].y = rot2_points[5].y;
rts6[3].z = rot2_points[5].z;
/* Crank case cylinder points */

ang = 0.0;
delta = 2*M_PI/40;

for (i = 0; i < 40; i++, aag+=delta)
{
    cyl1[i].x = .425;
    cyl1[i].y = -.15 + 0.1*sin(ang);
    cyl1[i].z = 0.0 + 0.1*cos(ang);

    cyl2[i].x = .425;
    cyl2[i].y = -.375 + 0.1*sin(ang);
    cyl2[i].z = 0.0 + 0.1*cos(ang);
}

cyl1[40].x = cyl1[0].x;
cyl1[40].y = cyl1[0].y;
cyl1[40].z = cyl1[0].z;
cyl2[40].x = cyl2[0].x;
cyl2[40].y = cyl2[0].y;
cyl2[40].z = cyl2[0].z;

popen_struct(ROTOR);
pset_int_colr_ind ((Pint) 7);        /* Create grey colored rotor */
pset_edge_flag (PEDGE_ON );
pset_edge_colr_ind ((Pint) 5);
pset_line_colr_ind ((Pint) 7);
ppolyline3 (&rotside[0]);
ppolyline3 (&rotside[1]);
ppolyline3 (&rotside[2]);
ppolyline3 (&rotside[3]);
ppolyline3 (&rotside[4]);
ppolyline3 (&rotside[5]);
pset_int_style (PSTYLE_SOLID);

ppolyline3 (&rot1);
ppolyline3 (&rot2);
pclose_struct();

popen_struct (CCASE);
pset_int_style (PSTYLE_SOLID);
pset_edge_colr_ind ((Pint) 5);
pset_edge_flag (PEDGE_ON );
pset_int_colr_ind ((Pint) 6);        /* Create sky blue crank case */
pset_line_colr_ind (Pint 7);
polyline3 (&contours[6]);
polyline3 (&contours[7]);
polyline3 (&contours[0]);
polyline3 (&contours[1]);
polyline3 (&contours[2]);
polyline3 (&contours[3]);
polyline3 (&contours[4]);
polyline3 (&contours[5]);
pclose_struct();

popen_struct (WEB);
pset_int_style (PSTYLE_SOLID);
pset_line_colr_ind (Pint 25);
pset_edge_flag (PEDGE_ON); /* Create red colored web */
polyline3 (&web);
pclose_struct();

popen_struct (SPRING);
pset_line_colr_ind (Pint 3);
pset_linetype (PLINE_DASH_DOT);
point_list3.num_points = 2;
point_list3.points = Top_Spring;
polyline3 (&point_list3);
point_list3.points = Side_Spring1;
polyline3 (&point_list3);
point_list3.points = Side_Spring2;
polyline3 (&point_list3);
pset_int_style (PSTYLE_HATCH);
pset_int_colr_ind (Pint 3);
pset_back_int_style (PSTYLE_SOLID);
pset_edge_flag (PEDGE_ON); /* Create green colored springs */
point_list3.num_points = 4;
point_list3.points = Top_Spring_Sup;
pfill_area3 (&point_list3);
point_list3.points = Side_Spring1_Sup;
pfill_area3 (&point_list3);
point_list3.points = Side_Spring2_Sup;
pfill_area3 (&point_list3);
pclose_struct();

popen_struct (AXES);
pset_edge_flag (PEDGE_ON);
pset_line_colr_ind ((Pint 1));
pset_linen_type (PLINE_SOLID);
point_list3.num_points = 2;
point_list3.points = X AXES;
ppolyline3 (&point_list3);
point_list3.points = Y AXES;
ppolyline3 (&point_list3);
point_list3.points = Z AXES;
ppolyline3 (&point_list3);
pset_char_expan ((Pfloat) .42);
pset_text_prec (PPREC_STROKE);
pset_text_font ( (Pint) 6);
pset_anno_char_ht((Pfloat) .05);
pset_char_space ( (Pfloat) .75);
pset_text_colr_ind((Pint) 2);
anno_text_refpt.x = -0.025;
anno_text_refpt.y = 0.172;
anno_text_refpt.z = 0.0 + 0.55;
anno_offset.delta_x = 0.0;
anno_offset.delta_y = 0.0;
anno_offset.delta_z = 0.0;
panno_text_rel3(&anno_text_refpt,&anno_offset,"X");
anno_text_refpt.x = -0.025 + 0.55;
anno_text_refpt.y = 0.172;
anno_text_refpt.z = 0.0;
panno_text_rel3(&anno_text_refpt,&anno_offset,"Y");
anno_text_refpt.x = -0.025;
anno_text_refpt.y = 0.172 + 0.55;
anno_text_refpt.z = 0.0;
panno_text_rel3(&anno_text_refpt,&anno_offset,"Z");

pclose_struct();

popen_struct (SYSTEM);
/* Compute the global scaling transform */
scale.delta_x = 1.0; scale.delta_z = 1.0;
scale.delta_y = 0.85;
pscale3 (&scale, &err, sc);

/* Compute the global translating transform */
shift.delta_x = 0.0; shift.delta_y = -0.3; shift.delta_z = 0.0;
ptranslate3 (&shift, &err, translate);

/* Compose them and create the Set Global Transform element */
pcompose_matrix3 (translate, sc, &err, global);
pset_global_tran3 (global);

Appendix I.  Orbit Animation Program BR_ANIM.C
GPROTX ((Pfloat) 0, xform);
pset_local_tran3 (xform, PTYPE_REPLACE);

GPROTY ((Pfloat) 0, xform);
pset_local_tran3 (xform, PTYPE_POSTCONCAT);

GPROTZ ((Pfloat) 0, xform);
pset_local_tran3 (xform, PTYPE_POSTCONCAT);

shift.delta_x = 0; shift.delta_y = 0; shift.delta_z = 0;
ptranslate3 (&shift, &err, xform);
pset_local_tran3 (xform, PTYPE_POSTCONCAT);

pexec_struct(WEB);

pexec_struct(ROTOR);

pexec_struct(CCASE);

pexec_struct(SPRING);

pexec_struct(AXES);

/* Calculate the view orientation matrix */

view_ref_pt.x = 0;
view_ref_pt.y = 0;
view_ref_pt.z = 0;
view_up_vec.delta_x = 0;
view_up_vec.delta_y = 1;
view_up_vec.delta_z = 0;
view_plane_normal.delta_x = 1;
view_plane_normal.delta_y = 0.3;
view_plane_normal.delta_z = -1;
peval_view_ori_matrix3(&view_ref_pt, &view_plane_normal, &view_up_vec, &error, view_rep1_ori_matrix);
pclose_struct();

/* Calculate the view mapping matrix */

view_map.proj_type = PTYPE_PARAL;
view_map.view_plane = 0.0;
view_map.proj_ref_point.x = 0;
view_map.proj_ref_point.y = 0;
view_map.proj_ref_point.z = 5;

view_map.win.x_min = -1; view_map.win.x_max = 1;
view_map.win.y_min = -1; view_map.win.y_max = 1;
view_map.back_plane = -1; view_map.front_plane = 1;

Appendix I  Orbit Animation Program BR_ANIM.C  184
view_map.proj_vp = vports[0];

peval_view_map_matrix3 (&view_map, &error, view_rep1.map_matrix);

/* store the view in the view table as view 1 */

view_rep1.clip_limit = view_map.proj_vp;
view_rep1.xy_clip = PND_CLIP;
view_rep1.front_clip = view_rep1.back_clip = PND_CLIP;

pset_view_rep3 (WORKSTATION_ID, (Pint) 1, &view_rep1);

/* All views are set according to the co-ordinate system used in the original
Bristol H-25 A Force Analysis i.e. X axis || lel to cylinder face, Y || lel to cylinder axis
and Z || lel to the crankshaft axis */

/* View as seen from X axis - view 2 */

view_plane_normal.delta_x = 0;
view_plane_normal.delta_y = 0;
view_plane_normal.delta_z = 1;

view_map.proj_vp = vports[3];
set_view (WORKSTATION_ID, (Pint) 2, &view_ref_pt, &view_plane_normal, &view_up_vec,
&view_map);

/* View as seen from Y axis - view 3 */

view_plane_normal.delta_x = 1;
view_plane_normal.delta_y = 0;
view_plane_normal.delta_z = 0;

view_map.proj_vp = vports[2];
set_view (WORKSTATION_ID, (Pint) 3, &view_ref_pt, &view_plane_normal, &view_up_vec,
&view_map);

/* View as seen from Z axis - view 4 */

view_up_vec.delta_x = 1;
view_up_vec.delta_y = 0;
view_up_vec.delta_z = 0;

view_plane_normal.delta_x = 0;
view_plane_normal.delta_y = 1;

Appendix I. Orbit Animation Program BR_ANIM.C
view_plane_normal.delta_z = 0;

view_map.proj_vp = vports[1];
set_view (WORKSTATION_ID, (Pint) 4, &view_ref_pt, &view_plane_normal, &view_up_vec,
&view_map);

popen_struct(PAN_MENU);
/* Build the contour list */

contour[0].num_points = 4;
contour[0].points = Pan_menu_border;
set.num_point_lists = 1; set.point_lists = contour;

name_list.ints = names;
name_list.num_ints = 1;

names[0] = PAN_MENU;
padd_names_set (&name_list);

pset_edge_flag(PEDGE_ON);
pset_int_style(PSTYLE_SOLID);
pset_int_colr_ind((Pint) 31);
pset_pick_id(MODEL);

pset_char_expan((Pfloat) .42);
pset_text_prec(PPRECROKESTROKE);
pset_text_font((Pint) 5);
pset_char_h((Pfloat) .07);
pset_char_space((Pfloat) .75);
pset_text_colr_ind((Pint) 2);
text.x = -.70;
text.y = .9135;
ptext((&text, "BRISTOL H-25A"));

contour[0].num_points = 4;
contour[0].points = outer_border1;

set.num_point_lists = 1; set.point_lists = contour;

pset_pick_id(PAN_RET);
pfill_area_set (&set);

pset_edge_flag(PEDGE_ON);
pset_int_style(PSTYLE_SOLID);
pset_int_colr_ind((Pint) 7);
pfill_area_set (&set);

Appendix I. Orbit Animation Program BR_ANIM.C
pset_char_expan ((Pfloat) .42);
pset_text_prc ((PPREC_STROKE));
pset_text_font ((Pint) 5);
pset_char_hl((Pfloat) .07);
pset_char_space ((Pfloat) .75);
pset_text_colr_ind((Pint) 2);
text.x = .02;
text.y = .75135;
ptext(&text, "RETURN");

close_struct();

popen_struct (VIEW1);
names[0] = VIEW1;
name_list.num_ints = 1;
padd_names_set (&name_list);
pxexec_struct (PAN_MENU);
pset_view_ind ((Pint) 1);
pxexec_struct (SYSTEM);
pset_view_ind ((Pint) 2);
pxexec_struct (SYSTEM);
pset_view_ind ((Pint) 3);
pxexec_struct (SYSTEM);
pset_view_ind ((Pint) 4);
pxexec_struct (SYSTEM);

/* Draw the view borders */
pset_view_ind ((Pint) 0);
point_list.num_points = 5;
point_list.points = pts;
for (i = 0; i < 4; i++) {
    pts[0].x = ports[i].x_min; pts[0].y = ports[i].y_min;
    pts[1].x = ports[i].x_max; pts[1].y = ports[i].y_min;
    pts[2].x = ports[i].x_max; pts[2].y = ports[i].y_max;
    pts[3].x = ports[i].x_min; pts[3].y = ports[i].y_max;
    pts[4].x = ports[i].x_min; pts[4].y = ports[i].y_min;
    pppolyline (&point_list);
}
close_struct();

ppost_struct (WORKSTATION_ID, VIEW1, (Pfloat) 1.0);
Repeat_Animation();
handle_input();
sleep(20);
}

void Repeat_Animation()

Appendix I.  Orbit Animation Program BR_ANIM.C
{ 
Pint ws_type;
Pfloat angle = 0.0, alpha1, alpha2, incr = ((2*M_PI) / NUM_INCREMENTs), dist = 0.0;
Ppoint3 fixed_pt;
matrix3 xform, xform1, sc, translate, global;
Pint err;
int i, iter;
vec3 shift, scale;
double omega, t, delta, x, y, z, th_x, th_y, th_z, no_time_instants;
int count;
double Time_Resp_Vector[150][6];
FILE *fp;

pset_disp_apd_st (WORKSTATION_ID, PREFER_ASAP, PMODE_UQUM);
alpha2 = alpha1 = 0.0;
pset_edit_mode ( PEDIT_REPLACE );

fp = fopen("BRISTOL.OUT","w");

omega = 361.28;
no_time_instants = no_increments/4.0;
delta = (0.01739)/no_time_instants; /* Use a scale factor of 10 */

for (t = 0, count = 1; t <= 0.01739; t = t + delta, count++)
{
    th_x = th_y = th_z = x = y = z = 0.0;
    for (i=1; i <= 40; i++)
    {
        th_x = th_x + disp_scale2 * (Disp_Vec_Cos[i][3] * cos(i) * omega * t)
            + Disp_Vec_Sin[i][3] * sin(i) * omega * t); /* th_x = +tx */
        th_y = th_y - disp_scale2 * (Disp_Vec_Cos[i][5] * cos(i) * omega * t)
            + Disp_Vec_Sin[i][5] * sin(i) * omega * t); /* th_y = -tz */
        th_z = th_z - disp_scale2 * (Disp_Vec_Cos[i][4] * cos(i) * omega * t)
            + Disp_Vec_Sin[i][4] * sin(i) * omega * t); /* th_z = -ty */

        x = x - disp_scale1 * (Disp_Vec_Cos[i][0] * cos(i) * omega * t)
            + Disp_Vec_Sin[i][0] * sin(i) * omega * t); /* x = -x */
        y = y + disp_scale1 * (Disp_Vec_Cos[i][2] * cos(i) * omega * t)
            + Disp_Vec_Sin[i][2] * sin(i) * omega * t); /* y = z */
        z = z + disp_scale1 * (Disp_Vec_Cos[i][1] * cos(i) * omega * t)
            + Disp_Vec_Sin[i][1] * sin(i) * omega * t); /* z = y */
    }
    Time_Resp_Vector[count][0] = x;
    Time_Resp_Vector[count][1] = y;
    Time_Resp_Vector[count][2] = z;
    Time_Resp_Vector[count][3] = th_x;
    Time_Resp_Vector[count][4] = th_y;
}

Appendix I. Orbit Animation Program BR_ANIM.C
Time_Resp_Vec[count][5] = th_z;

/* Write the displacement in one time period to the file BRISTOL.OUT */

fprintf(fp, "%15.6f, %15.6f, %15.6f, %15.6f, %15.6f, %15.6f
",
-Time_Resp_Vec[count][0],
+Time_Resp_Vec[count][2],
+Time_Resp_Vec[count][1],
+Time_Resp_Vec[count][3],
-Time_Resp_Vec[count][5],
-Time_Resp_Vec[count][4]);

}

fclose(fp);

for (iter = 1; iter <= no_repeats; iter++) /* Begin Animation */
{
    printf("Begin cycle # %2d\n", iter);
    for (count = 1; count <= no_time_instants; count++)
    {
        popen_struct (SYSTEM);

        pset_elem_ptr ((Pint) 2);
        prolate_x(Time_Resp_Vec[count][0], &err, xform);
        pset_local_tran3 (xform, PTYPE_REPLACE);

        pset_elem_ptr ((Pint) 3);
        prolate_y(Time_Resp_Vec[count][1], &err, xform);
        pset_local_tran3 (xform, PTYPE_POSTCONCAT);

        pset_elem_ptr ((Pint) 4);
        prolate_z(Time_Resp_Vec[count][2], &err, xform);
        pset_local_tran3 (xform, PTYPE_POSTCONCAT);

        shift.delta_x = Time_Resp_Vec[count][3];
        shift.delta_y = Time_Resp_Vec[count][4];
        shift.delta_z = Time_Resp_Vec[count][5];

        pset_elem_ptr ((Pint) 5);
        ptranslate3 (&shift, &err, xform);
        pset_local_tran3 (xform, PTYPE_POSTCONCAT);

        pclose_struct();

        ppost_struct(WORKSTATION_ID, VIEW1, (Pfloat)1.0);
        push_ws (WORKSTATION_ID, PFLAG_PERFORM);
    }
    printf("End cycle # %2d\n", iter);
}

Appendix I. Orbit Animation Program BR_ANIM.C
#define PAN_SCALE  (M_PI/500)

static void
handle_input()
{
    int  lastx, lasty, done = 0;
    float theta = M_PI/4, phi = 0.0706;
    float x, y;
    Ppoint locator_position;
    int  num_moves;
    Pint  ws_id, device_num, view;
    Ppick_path path;
    Pin_class class;
    Pin_status status;
    Pview_rep3 cur_view, req_view;
    Pupdate st upd_st;
    Pint  error, index;
    Pfilter filter;
    Pint  inclusion[1];
    Ppick_path_elem path_elems[1];

    theta = 3.14/4.0;
    phi = 0.0706;
    lastx = 0;
    lasty = 0;

    pset_loc_mode(WORKSTATION_ID, (Pint) 1, POP_SAMPLE, PSWITCH_ECHO);
    /* pset_disp_upd_st(WORKSTATION_ID, PREFER_WAIT, PMODE_NIVE); */
    pset_edit_mode(PEDIT_REPLACE);

    filter.incl_set.num_ints = 1;
    filter.incl_set.ints = inclusion;
    inclusion[0] = VIEW1;

    filter.excl_set.num_ints = 0;
    filter.excl_set.ints = (Pint *)NULL;

    pset_pick_mode (WORKSTATION_ID, DEVICE, POP_EVENT, PSWITCH_ECHO);
    pset_pick_filter( WORKSTATION_ID, DEVICE, &filter );

    path.path_list = path_elems;
    done = 5;
    while (done == 5 || done == 0 )
    {
        GPAWEV ((Pint) 0, &ws_id, &class, &device_num);
        /* printf("class = %d\n", class); */
        if (class != PIN_NONE)
            pget_pick ((Pint) 2, &status, &path);

        if ( status == PIN_STATUS_OK )

        {

Appendix I. Orbit Animation Program BR_ANIM.C 190
switch (path.path_list[1].pick_id )
{
  case PANNING: { done = 0; 
    break;}
  case PANEX: { done = 5; 
    break;}
  case PAN_RET: { done = 1; 
    break;}
}

} } 

if ( done == 0 ){
  psample_loc (WORKSTATION_ID, (Pint) 1, &view, &locator_position);
  if (200*locator_position.x != lastx  
      || 200*locator_position.y != lasty ) {
    x = locator_position.x*200;
    y = locator_position.y*200;

    theta += PAN_SCALE * (x - lastx);
    phi += PAN_SCALE * (lasty - y);

    /* Reset the view plane normal. */
    view_plane_normal.delta_x = cos( phi ) * sin( theta );
    view_plane_normal.delta_y = sin( phi );
    view_plane_normal.delta_z = cos( phi ) * cos( theta );

    /* Calculate the new orientation matrix. */
    peval_view_ori_matrix3( &view_ref_pt, &view_plane_normal,  
                           &view_up_vec, &error, view_rep1.ori_matrix );

    /* Set the view if orientation is okay. */
    pset_view_rep3( WORKSTATION_ID, (Pint) 1, &view_rep1 );

    /* Update the workstation */
    predraw_all_structs( WORKSTATION_ID, PFLAG_ALWAYS );

    lastx = x;
    lasty = y;
} } }
/* *********************************************** */
/* */
/* Color Table */
/* *********************************************** */

void Color_Table()
{

    /* DEFINE VARIABLES TO BE USED */

    int index = 0;  /* start entries at this location */
    int number = 50;  /* number of colors to be defined */
    float colors[152];  /* color table RGB values */

    /* set up the color table */

    colors[0] = 0.0;  colors[1] = 0.0;  colors[2] = 0.0;  /* 0 = black */
    colors[6] = 1.0;  colors[7] = 0.0;  colors[8] = 0.0;  /* 2 = red */
    colors[9] = 0.0;  colors[10] = 1.0;  colors[11] = 0.0;  /* 3 = green */
    colors[12] = 0.0;  colors[13] = 0.0;  colors[14] = 1.0;  /* 4 = blue */
    colors[15] = 1.0;  colors[16] = 1.0;  colors[17] = 0.0;  /* 5 = yellow */
    colors[18] = 1.0;  colors[19] = 0.0;  colors[20] = 1.0;  /* 6 = magenta */
    colors[21] = 1.0;  colors[22] = 1.0;  colors[23] = 1.0;  /* 7 = cyan */
    colors[24] = 0.9176471;  colors[25] = 0.9176471;  colors[26] = 0.678431;  /* 8 = brown */
    colors[27] = 0.6;  colors[28] = 0.5;  colors[29] = 1.0;  /* 9 = purple */
    colors[30] = 1.0;  colors[31] = 0.6;  colors[32] = 0.0;  /* 10 = orange */
    colors[33] = 0.8;  colors[34] = 0.8;  colors[35] = 0.8;  /* 11 = white */
    colors[36] = 0.8;  colors[37] = 0.0;  colors[38] = 0.0;  /* 12 = red */
    colors[39] = 0.0;  colors[40] = 0.8;  colors[41] = 0.0;  /* 13 = green */
    colors[42] = 0.0;  colors[43] = 0.0;  colors[44] = 0.8;  /* 14 = blue */
    colors[45] = 0.8;  colors[46] = 0.8;  colors[47] = 0.0;  /* 15 = magenta */
    colors[48] = 0.8;  colors[49] = 0.0;  colors[50] = 0.8;  /* 16 = cyan */
    colors[51] = 0.0;  colors[52] = 0.8;  colors[53] = 0.8;  /* 17 = yellow */
    colors[54] = 0.8;  colors[55] = 0.0;  colors[56] = 0.4;  /* 18 = brown */
    colors[57] = 0.5;  colors[58] = 0.4;  colors[59] = 0.8;  /* 19 = purple */
    colors[60] = 0.8;  colors[61] = 0.5;  colors[62] = 0.0;  /* 20 = orange */
    colors[63] = 0.6;  colors[64] = 0.6;  colors[65] = 0.6;  /* 21 = white */
    colors[66] = 0.6;  colors[67] = 0.0;  colors[68] = 0.0;  /* 22 = red */
    colors[69] = 0.0;  colors[70] = 0.6;  colors[71] = 0.0;  /* 23 = green */
    colors[72] = 0.0;  colors[73] = 0.0;  colors[74] = 0.6;  /* 24 = blue */
    colors[75] = 0.6;  colors[76] = 0.6;  colors[77] = 0.0;  /* 25 = yellow */
    colors[78] = 0.6;  colors[79] = 0.0;  colors[80] = 0.6;  /* 26 = magenta */
    colors[81] = 0.0;  colors[82] = 0.6;  colors[83] = 0.6;  /* 27 = cyan */
    colors[84] = 0.6;  colors[85] = 0.4;  colors[86] = 0.3;  /* 28 = brown */
    colors[87] = 0.4;  colors[88] = 0.3;  colors[89] = 0.6;  /* 29 = purple */
    colors[90] = 0.6;  colors[91] = 0.3;  colors[92] = 0.0;  /* 30 = orange */
    colors[93] = 0.7;  colors[94] = 0.7;  colors[95] = 0.7;  /* 31 = green */
    colors[96] = 0.9;  colors[97] = 0.4;  colors[98] = 0.4;  /* 32 = blue */
    colors[99] = 0.4;  colors[100] = 0.9;  colors[101] = 0.4;  /* 33 = cyan */
    colors[102] = 0.4;  colors[103] = 0.6;  colors[104] = 0.9;  /* 34 = magenta */
    colors[105] = 0.9;  colors[106] = 0.9;  colors[107] = 0.4;  /* 35 = yellow */
    colors[108] = 0.9;  colors[109] = 0.4;  colors[110] = 0.9;  /* 36 = brown */
    colors[111] = 0.4;  colors[112] = 0.9;  colors[113] = 0.9;  /* 37 = cyan */
    colors[114] = 0.4;  colors[115] = 0.2;  colors[116] = 0.2;  /* 38 = yellow */
    colors[117] = 0.3;  colors[118] = 0.2;  colors[119] = 0.5;  /* 39 = purple */

Appendix I. Orbit Animation Program BR_ANIM.C

192
colors[120] = 0.4; colors[121] = 0.2; colors[122] = 0.0; /* 40 = org4 */
colors[123] = 0.3; colors[124] = 0.3; colors[125] = 0.3; /* 41 = wht */
colors[126] = 0.8; colors[127] = 0.2; colors[128] = 0.4; /* 42 = red */
colors[129] = 0.2; colors[130] = 0.8; colors[131] = 0.4; /* 43 = grn */
colors[132] = 0.3; colors[133] = 0.3; colors[134] = 0.8; /* 44 = blu */
colors[135] = 0.7; colors[136] = 0.8; colors[137] = 0.2; /* 45 = yello */
colors[138] = 0.9; colors[139] = 0.1; colors[140] = 0.8; /* 46 = mag */
colors[141] = 0.1; colors[142] = 0.8; colors[143] = 0.8; /* 47 = cyan */
colors[144] = 0.8; colors[145] = 0.7; colors[146] = 0.2; /* 48 = brn */
colors[147] = 0.6; colors[148] = 0.3; colors[149] = 0.8; /* 49 = ppl */
colors[150] = 0.9; colors[151] = 0.7; colors[152] = 0.3; /* 50 = org */
/** load the color table **/
GPCR (WORKSTATION_ID, index, number, colors);
}
APPENDIX J. Experimental determination of spring stiffnesses

The experimental determination of the natural frequencies required modification of a standard production shell to allow access to the rigid body. Ten holes were cut at various locations on the shell, which while reducing the stiffness of the shell, would also result in a closer model of a production compressor than remounting the rigid body on a test stand. The shell was mounted on a steel and concrete inertial block without its usual rubber grommet isolators.

Through one of the ten holes, a piezoelectric accelerometer was mounted and through seven others an impact hammer with a piezoelectric force gage was used to excite the rigid body, in suspension. A soft rubber tip was used to excite the structure over a low frequency range (approximately 0 to 100 Hz).

A sample FRF is presented in Fig. 48, along with its input (force) and output (mobility) auto spectrums. The eigenvalue frequencies were obtained with an accuracy of 0.15 Hz.
Figure 48. A sample FRF for the suspension
References


5. Soong, K., Thompson, B.S., "An Experimental Investigation of the Dynamic Response of a Mechanical System with Bearing Clearance",


Vita

Ramakant P. Arcot, the third child of Dr. and Mrs. Arcot Devaraj Kothanda Pani, was born on August 4, 1969 in Bombay, India. He was industrious in his studies, and was the recipient of State and National Merit Scholarships for his undergraduate studies. He earned his Bachelor of Science degree in Mechanical Engineering from Osmania University, India. Mr. Arcot developed his interest in academic research and decided to pursue a Master of Science in Mechanical Engineering, and joined the M.S. program in the Department of Mechanical Engineering at Virginia Polytechnic Institute and State University in Fall 1991.