Dynamics of a Simple Rotor System

Patrick McComb

SRDD

12/9/2013
Introduction and Theory

Rotating machinery can be found in many applications from small pumps and compressors to large gas turbine generators and aircraft engines. These rotating machines have many different components each of which has to be designed to withstand different operating conditions. In a jet engine application, static structures have to maintain the stiffness and rigidity of the machine as they operate in a variety of different environments and maneuvers. However, a typical gas turbine engine has at least one rotating shaft that contains bladed disks which are responsible for either compressing air or extracting energy from a combination of air and fuel exhaust. The rotor system produces the power or thrust for these systems. Figure 1 is an example of a gas turbine engine rotor section, with a large multi-stage compressor connected by a shaft to a multistage turbine.

![Figure 1. Gas turbine engine compressor and turbine [1]](image)

The study of the dynamic response of the rotating components is called rotordynamics. Rotor systems can vibrate longitudinally, which is an extension or compression of a rotor. They can also vibrate torsionally, which is an oscillation of the twist of a rotor. The most important and difficult vibrations to understand are lateral vibrations of a rotor system. Unlike a simple
beam, the rotation of the rotor system and gyroscopic moments create a lateral motion that can be visualized as an orbit. Understanding the lateral vibrations of a rotor system is extremely important for stable and safe operation. It is important to understand and avoid critical speeds during operation. It can also be important in jet engine applications to assure proper deflections and corresponding clearances are held.

The behavior of a rotor system looks very much like other vibration problems. A vibrating beam system can be described by the equation

\[
[M][\ddot{x}] + [C][\dot{x}] + [K][x] = F
\]

The rotor system can be described by the following differential equation in a stationary reference frame.

\[
[M][\ddot{x}] + ([C] + [C_{gyr}])[\dot{x}] + [K][x] = F
\]

When used in the Finite Element Method, \([M]\) represents the mass matrix created for each discrete element. Similarly, \([C]\) represents the damping matrix, \([C_{gyr}]\) represents the gyroscopic matrix, and \([K]\) represents the stiffness matrix. The additional gyroscopic term shows how the spin speed of the rotor will directly affect both the natural frequencies of the rotor vibration and the mode shape or orbit of the system. The system of equations can be solved as an eigenvalue problem; the corresponding eigenvalues represent the natural frequencies of the system and the eigenvectors describe the mode shape and orbit of the system. It is important in the solution of the system to understand the shape and rotational direction of the modes. For isotropic bearings, the rotor will move in a circular orbit, however, if the bearings are not isotropic the orbits will be elliptical. The direction of rotation depends on the phase difference between the two directions of lateral motion. A mode is considered forward whirling
if it is rotating in the same direction as the rotor rotation and a backward whirling if it is rotating in the opposite direction. Modes typically come in pairs, with the forward whirling modes increasing in frequency with rotor speed, while the backward whirling modes decrease in frequency with rotor speed. A Campbell diagram is a very useful tool for understanding the interaction between rotor rotating speed and natural frequencies and an example can be found in Figure 2.

![Campbell Diagram](image)

**Figure 2. Sample Campbell Diagram**

Here 4 modes are plotted to show the stiffening and softening affects that the rotor rotation has on each mode.

This project will show two ways to perform dynamic analysis on a shaft and two disk rotor model. Both methods use the Finite Element Method; this first method is academic software that uses Matlab, while the second is using an ANSYS FEM. The results describe the frequencies and shapes of a simple rotor vibration.
Modeling and Methodology

A simple two disk rotor was modeled to demonstrate a dynamic analysis of a rotor system. Rotor models typically consist of a shaft, an orientation of several disks, and bearings to support the structure. In this analysis the bearings were assumed to be isotropic with a stiffness of 1 MN/m at either end of the rotor. The rotor modeled has a 1.5 meter long shaft with a .05 meter diameter. The two disks are modeled equally spaced from either end at .5 meters and 1.0 meters along the shaft. The first disk had a diameter of .28 meters and a thickness of .07 meters, while the second disk had a diameter of .35 meters and also a thickness of .07 meters. The rotor is assumed to be made from steel with $E=211 \text{ GN/m}^2$ and $\rho=7810 \text{ kg/m}^3$. The analysis was run from 0 to 4000 RPM.

Matlab

This rotor was model using two different methodologies. The first model was produced using software provided with *Dynamics of Rotating Machines* (Friswell et al. [3]). This software produces a simple Finite Element Model that is solved using Matlab. Figure 3 is a representation of the Friswell model in Matlab.

![Figure 3. Matlab model of rotor system](image)

The Matlab model consists of 7 nodes and 6 elements, which are Timoshenko elements with gyroscopic effects included. This model is simply run in Matlab and produces frequencies, mode
shapes, and orbits from 0 to 4000 RPM. The code for this model and analysis can be found in Appendix A.

**ANSYS**

The rotor was also modeled using ANSYS finite element software. The model has 16 nodes with 15 Beam188 elements equal spaced every .1 meters along the shaft. The disks were modeled as Mass21 elements at node 6 and node 11, and the moments of inertia and masses were calculated for each disk and input as real constants for each element. The bearings are modeled at the beginning and end of the shaft as Combi214 elements. The stiffness and damping constants in the lateral directions are input as real constants.

Each node is constrained in the longitudinal (x) direction, while the bearing nodes on either end are constrained in all degrees of freedom. The model was run rotating about the x axis with the Coriolis Effect on from 0 to 4000 RPM. Figure 4 shows the ANSYS model, which was solved using the Block-Lanczos mode extraction method. An input file for this model can be found in Appendix B.

![Figure 4. ANSYS rotor system model](image-url)
Results and Discussion

Both models produce frequencies, mode shapes and orbits for the first four modes of the system. Figure 5 shows the Campbell diagram for the rotor model from 0 to 4000 RPM from the Matlab model.

The Campbell diagram shows the backward whirling modes in green and the forward whirling modes in red. One can also see that the forward whirling modes increase in frequency with increasing rotor speed, while backward whirling modes decrease frequency with increasing rotor speed. Figures 5 and 6 show the mode shapes and orbits respectively as well as the frequencies at 4000 RPM.
The mode shapes in figure 6 show the relative shape, which is circular, but the amplitudes are arbitrary. The modes are all circular as a result of the isotropic bearings and would be elliptical if the bearing were not isotropic. One may also notice that the odd modes are backwards whirling and the even modes are forward whirling and that the modes 1 & 2 and modes 3 & 4 are pairs of similar modes.

Figure 8 shows the Campbell Diagram produced from the ANSYS model using the PLCAMP command from the rotor dynamics menu.
This Campbell diagram shows the backward whirling modes in purple and blue, and the forward whirling modes in pink and red. The diagram is consistent with Matlab with forward whirling modes increasing in frequency with increasing rotor speed, while backward whirling modes decrease frequency with increasing rotor speed. Figures 9 and 10 shows the mode shapes and orbits using the PLORB command in for the first 4 modes in ANSYS.
Modes 1 and 2 are first lateral modes with circular orbits, one forwards one backwards, and are the same as those produced by Matlab. Modes 3 and 4 are second lateral modes also with circular orbits and with the center of the rotor motionless just like the Matlab orbits. Overall, the Campbell diagrams, mode shapes, and orbits of both the Matlab model and the ANSYS beam model are in agreement. Table 1 shows that the first 4 natural frequencies are in good agreement and are within 1-2% of each other.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Matlab-Friswell (Hz)</th>
<th>ANSYS (Hz)</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td>13.5</td>
<td>13.3</td>
<td>1.50%</td>
</tr>
<tr>
<td>Mode 2</td>
<td>13.97</td>
<td>13.8</td>
<td>1.23%</td>
</tr>
<tr>
<td>Mode 3</td>
<td>40.1</td>
<td>40.7</td>
<td>-1.47%</td>
</tr>
<tr>
<td>Mode 4</td>
<td>46.91</td>
<td>47.9</td>
<td>-2.07%</td>
</tr>
</tbody>
</table>

Figure 10. Modes 3 and 4 Orbits
Conclusion

The academic Matlab code and commercial FEM software ANSYS are good tools for understanding the dynamics of rotating systems. Although the Matlab software was sufficient for this simple system, ANSYS would be the tool of choice for modeling more complex systems. ANSYS has a rotor dynamics menu that allows for automatic post processing of Campbell diagrams and orbits. The ability to use the animation features in ANSYS are also really good, especially for understanding the direction and shape of the rotor whirling.

Understanding the dynamics of a rotor system is critical for safe and efficient operation. The dynamics of a rotor system are similar to non-rotating systems, with the increased complexity of gyroscopic moments acting on the system. Both simple and more complex rotor systems can be modeled numerically using the Finite Element Method to understand their frequencies and orbit characteristics.
References


Appendix A: Friswell Software - Matlab Code

%Sample two disk rotor

clear
format short e
close all

set(0,'defaultaxesfontsize',12)
set(0,'defaultaxesfontname','Times New Roman')
set(0,'defaulttextfontsize',12)
set(0,'defaulttextfontname','Times New Roman')

% set the material parameters
E = 211e9;
G = 81.2e9;
rho = 7810;
damping_factor = 0; % no damping in shaft

% Consider the model with 6 equal length elements
% Shaft is 1.5m long
model.node = [1 0.0; 2 0.25; 3 0.5; 4 0.75; 5 1.0; 6 1.25; 7 1.50];

% Assume shaft type 2 - Timoshenko with gyroscopic effects included
% Solid shaft with 50mm outside diameter
shaft_od = 0.05;
shaft_id = 0;
model.shaft = [2 1 2 shaft_od shaft_id rho E G damping_factor; ...
    2 2 3 shaft_od shaft_id rho E G damping_factor; ...
    2 3 4 shaft_od shaft_id rho E G damping_factor; ...
    2 4 5 shaft_od shaft_id rho E G damping_factor; ...
    2 5 6 shaft_od shaft_id rho E G damping_factor; ...
    2 6 7 shaft_od shaft_id rho E G damping_factor];

% Disk 1 at node 3 has diameter of 280mm and thickness of 70mm
% Disk 2 at node 5 has diameter of 350mm and thickness of 70mm
% Note inside diameter of disk is assumed to be the outside diameter of the
% shaft
disk1_od = 0.28;
disk2_od = 0.35;
disk_thick = 0.07;
model.disc = [1 3 rho disk_thick disk1_od shaft_od; ...
    1 5 rho disk_thick disk2_od shaft_od];

% constant stiffness short isotropic bearing (1NM/m) with no damping
% bearings at the ends - nodes 1 and 7
bear_stiff = 1e6;
model.bearing = [3 1 bear_stiff bear_stiff 0 0; ...
    3 7 bear_stiff bear_stiff 0 0];

% draw the rotor
figure(1), clf
picrotor(model)

% plot the Campbell diagram and root locus

Rotor_Spd_rpm = 0:100:4500.0;
Rotor_Spd = 2*pi*Rotor_Spd_rpm/60;  % convert to rad/s
[eigenvalues,eigenvectors,kappa] = chr_root(model,Rotor_Spd);

figure(2)
NX = 2;
damped_NF = 1;  % plot damped natural frequencies
plotcamp(Rotor_Spd,eigenvalues,NX,damped_NF,kappa)

figure(3)
plotloci(Rotor_Spd,eigenvalues,NX)

% plot the modes at a given speed

Rotor_Spd_rpm = 4000;
Rotor_Spd = 2*pi*Rotor_Spd_rpm/60;  % convert to rad/s
[eigenvalues,eigenvectors,kappa] = chr_root(model,Rotor_Spd);

figure(4)
subplot(221)
plotmode(model,eigenvectors(:,1),eigenvalues(1))
subplot(222)
plotmode(model,eigenvectors(:,3),eigenvalues(3))
subplot(223)
plotmode(model,eigenvectors(:,5),eigenvalues(5))
subplot(224)
plotmode(model,eigenvectors(:,7),eigenvalues(7))

% plot orbits

figure(5)
outputnode = [3 5];
axes('position',[0.2 0.53 0.2 0.2 ])
plotorbit(eigenvectors(:,1),outputnode,'Mode 1',eigenvalues(1))
axes('position',[0.39 0.53 0.2 0.2 ])
plotorbit(eigenvectors(:,3),outputnode,'Mode 2',eigenvalues(3))
axes('position',[0.58 0.53 0.2 0.2 ])
plotorbit(eigenvectors(:,5),outputnode,'Mode 3',eigenvalues(5))
axes('position',[0.2 0.25 0.2 0.2 ])
plotorbit(eigenvectors(:,7),outputnode,'Mode 4',eigenvalues(7))
axes('position',[0.39 0.25 0.2 0.2 ])
plotorbit(eigenvectors(:,9),outputnode,'Mode 5',eigenvalues(9))
axes('position',[0.58 0.25 0.2 0.2 ])
plotorbit(eigenvectors(:,11),outputnode,'Mode 6',eigenvalues(11))
Appendix B: ANSYS Input file

finish
/clear,all
/filename,simple_jeffcott
/prep7

!Geometry
L=1.5
SD=.05
SR=SD/2
DL=.28
DR=.35
HL=.07
HR=.07

!Material Properties
E=2.11E11
PR=.3
Density=7810
RR=DR/2
RL=DL/2
piv=acos(-1)

!Left Disk
mL=piv*RL**2*HL*Density
IdL=(mL*{(3*RL**2)+(HL**2)})/12
IpL=(mL*RL**2)/2

!Right Disk
mR=piv*RR**2*HR*Density
IdR=(mR*{(3*RR**2)+(HR**2)})/12
IpR=(mR*RR**2)/2

!Bearings
kzz=1e6
kyy=1E6
czz=0
cyy=0

!Rotational Velocity
Start_rpm = 0
End_rpm = 4000
Inc = 500
Nmodes = 4

!Shaft
ET,1,Beam188
KEYOPT,1,3,2
MP,EX,1,E
MP,PRXY,1,PR
MP,DENS,Density
!section
SECTYPE, 1, BEAM, CSOLID
SECDATA, SR

! Shaft model
N, 1, 0
N, 2, 0.1
N, 3, 0.2
N, 4, 0.3
N, 5, 0.4
N, 6, 0.5
N, 7, 0.6
N, 8, 0.7
N, 9, 0.8
N, 10, 0.9
N, 11, 1.0
N, 12, 1.1
N, 13, 1.2
N, 14, 1.3
N, 15, 1.4
N, 16, 1.5

TYPE, 1
MAT, 1
secnum, 1
*DO, I, 1, 15
   E, I, I+1
*ENDDO

! Disk Left
ET, 2, MASS21
R, 2, mL, mL, mL, IpL, IdL, IdL
TYPE, 2
REAL, 2
E, 6

! Disk Right
ET, 2, MASS21
R, 2, mR, mR, mR, IpR, IdR, IdR
TYPE, 2
REAL, 2
E, 11

! Bearing
N, 17, 0
N, 18, 1.5
ET, 3, 214
KEYOPT, 3, 2, 1
R, 3, kzz, kyy, czz, cyy
TYPE, 3
REAL, 3
E,17,1
E,18,16

!BCs
/SOLU
D,ALL,UX
D,ALL,ROTX
D,17,ALL
D,18,ALL
allsel,all,all

/SOLU
esel,,ename,,188
esel,a,ename,,21
cm,rot_prt,elem
esel,all

!rotational velocity/coriolis
antype,modal
modopt,qrdamp,Nmodes,,on
mxpand,Nmodes
coriolis,on,,on
*DO,J,Start_rpm,End_rpm,Inc
spinRPM = J
spinRds = spinRPM*piv/30
cmomega,rot_prt,spinRds
SOLVE
*ENDDO
FINISH

/POST1
SET,LIST
PLCAMP,0,1,RPM,0,rot_prt,0
PRCAMP,0,1,RPM,0,rot_prt,0