CO-SIMULATION OF ACTIVE MAGNETIC BEARING ROTORS USING ADAMS, MSC APEX, AND SIMULINK

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Mechanical Engineering

by

Shea Charkowsky

August 2023

COMMITTEE MEMBERSHIP

TITLE:	Co-Simulation of Active Magnetic Bearing Rotors Using Adams, MSC Apex, and Simulink
AUTHOR:	Shea Charkowsky
DATE SUBMITTED:	August 2023
COMMITTEE CHAIR:	Xi (Julia) Wu, Ph.D. Professor of Mechanical Engineering
COMMITTEE MEMBER:	Andrew Davol, Ph.D. Professor of Mechanical Engineering
COMMITTEE MEMBER:	Hemanth (HP) Porumamilla, Ph.D. Associate Professor of Mechanical Engineering

ABSTRACT

Co-Simulation of Active Magnetic Bearing Rotors

Using Adams, MSC Apex, and Simulink

Shea Charkowsky

Apex-Adams-Simulink co-simulation is applied to active magnetic bearing (AMB) rotors, demonstrating results unshown in literature, including continuous frequency response, and unavailable in known rotor dynamics software, including touchdown bearing impact. AMBs levitate rotors without contact, so they involve no friction, wear, lubrication, pollution, or shaft speed limits and are thus valuable for large, high-speed applications. Modeling of such rotors, necessary for safety and performance, simultaneously requires flexible body dynamics and advanced control design, but simulation programs tend to specialize in only one or the other. The co-simulation method combines multiple such programs—MSC Apex (finite element modeling), Adams (multibody dynamics), and Simulink (graphical control design) with MATLAB (visualization tools)—expanding the design space for AMB rotor modeling beyond that of available commercial software. In this work, accuracy of co-simulation to theory is validated through a rigid AMB rotor and a hanging disk on a steel wire, and new results are shown for a flexible anisotropic rotor and a simplified touchdown bearing impact test.

ACKNOWLEDGMENTS

Solar Turbines, *for funding rotor dynamics research at Cal Poly, including this work.*

Dr. Wu, *for encouraging me to explore beyond what I ever thought possible.*

Dr. Davol and Dr. HP, *for taking the time to review this work.*

Cameron Naugle and Michael Mullen, *on whose prior research this work is founded.*

Jaret Wedow, *whose tutorials were indispensable for learning Adams co-simulation.*

Luis Herrera and Arturo Flores, *for their adjacent research.*

My family, *for making this effort possible through their constant love and support.*

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Active magnetic bearings (AMBs) support a rotor by levitating it in space with electromagnets, whose strengths are varied in real time by a microcontroller based on sensed rotor position and a suitable control law (Figure 1.1). This setup eliminates friction, wear, and lubrication, allows high shaft speeds, produces no pollutants, enables operation in extreme environments, and provides active vibration control [1], [2]. These benefits and others make AMBs desirable for many rotating machinery applications, especially those with high operating speeds. In such cases, since safety concerns are significant, AMBs rotors should be simulated prior to experimental testing, but the nature of AMBs presents unique challenges to such analysis.

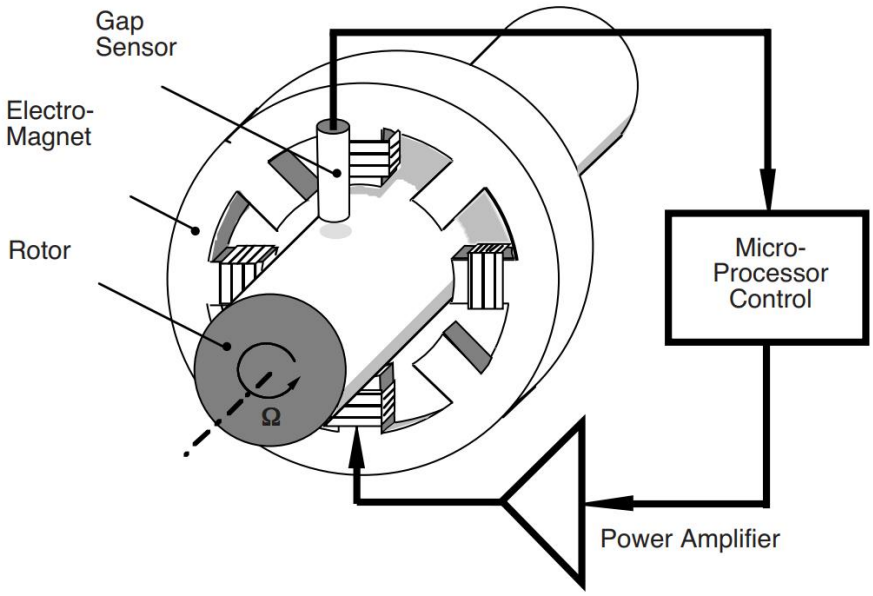


Figure 1.1. An active magnetic bearing (Figure 1.1 of [1]).

First, AMB systems are highly nonlinear and unstable, so active control is required to suspend the rotor. Basic algorithms such as PID, while useful in some cases, are often inadequate for large-scale practical systems [1]. Literature contains hundreds of articles on advanced AMB control schemes; various approaches can be found in [3], [4], [5], [6]. Second, AMB rotors are often run at very high speeds, sometimes exceeding multiple bending frequencies of the shaft [2]. At these speeds, flexible shaft behavior is vastly different from rigid behavior, so this flexibility must also be modeled to prevent miscalculation of resonance or destabilization by the controller [1].

Software is scarce that can perform both tasks at once. MADYN 2000 by DELTA JS is among the most comprehensive programs in this aspect, modeling flexible rotors while offering a variety of control scheme options for AMBs. It also allows the export of the flexible rotor as a state space plant that can be implemented in MATLAB [7]. However, export of the model as state space matrices makes some areas of analysis impossible. For example, AMB rotors often have a set of *touchdown bearings*, mechanical bearings on which the rotor will land in the event of control system failure, an important interaction to simulate. This type of contact involves rapid transitions between freefall, normal and tangential impulses, and sliding friction, well beyond what a single set of state space matrices can accommodate. Treatments of this topic reveal its complexity [1].

Instead of searching for a single application that can perform all required analysis, we turn to general purpose software that specializes in each of its aspects.

Two of these programs are Adams by Hexagon and Simulink by MathWorks; the first is a physics simulator that can model complex dynamics and interactions, and the second is a block diagram programming application useful for advanced control design. Like MADYN 2000, Adams can export models for MATLAB, but the Adams plant export retains the nonlinear and time-varying interactions of the system, even when implemented in a Simulink control scheme. This technique is known as Adams-Simulink *co-simulation*, which combines the full dynamics modeling of Adams with the full control design space of Simulink. Some examples in use are [8], [9], [10], [11].

As for Adams co-simulation of magnetic bearing rotors, literature research by the author returned remarkably few results, so few that all of them can be listed: [12], [13], [14], [15], [16]. Zhang et al. [12] levitate a non-spinning rotor with PID control. Li et al. [13] show stability of a non-spinning rotor under force impulses, with nonlinear magnetic forces calculated by ANSYS. Lee et al. [14] show stability of a flexible rotor at 0 RPM, 3000 RPM, and 30000 RPM. In the latter two cases, the shaft experiences small vibrations synchronous with spin frequency. Rotor flexibility is modeled with FEA capabilities of ANSYS, and the magnetic bearings are approximated as linear spring-like elements. Liu and Shi [15] also approximate the magnetic bearings of their rotor as spring-like elements but obtain their stiffness parameters from simulations in Maxwell 3D. They show correction of initial displacement in a non-spinning rotor using PID control with self-varying parameters. Finally, Ouyang et al. [16] control a spinning rotor with a modal decoupling controller at 20000 RPM, 25000 RPM, and 30000 RPM, showing displacement correction, disturbance rejection, and control of vibration due to mass imbalance.

Some rotor dynamics results are notably missing from this list, most arising from the fact that these articles do not simulate the rotor over a continuous shaft speed range. None show the rotor's *frequency response*, which plots vibration amplitude and phase with respect to shaft speed. This information makes resonant frequencies clear and ensures that the system will remain stable during acceleration and deceleration of the rotor. None show information about the shaft's *orbits*, the shape of their displacement paths, useful for fault identification in practical systems. These are two of many performance measures and rotor types that have yet to be explored in co-simulation.

The goals of this work then are the following:

1. Validate the accuracy of Adams-Simulink co-simulation of AMB rotors.
2. Show that co-simulation can model AMB rotors in ways that other software cannot, making it worth new research.
3. Expand on literature, performing simulations and analysis that have not yet been demonstrated.

In doing so, we rely heavily on a few resources. Most theoretical derivations closely follow the work performed by Schweitzer and Maslen et al. in *Magnetic Bearings* [1]. Virtual models of practical prototypes are inspired by [1], by Genta's *Dynamics of Rotating Systems* [17], and by the thesis of Naugle [18]. Visualizations are inspired by those of Naugle and of Mullen in their theses [18] [19] and related publications [20], and code provided by Mullen is used in the creation of some figures.

The main body of this work is presented in four chapters: Chapters 2 and 3 accomplish our first goal, validation of co-simulation, while Chapters 4 and 5 accomplish the second and third goals, showing new results. In Chapter 2, we study a rigid AMB rotor, establishing theory, co-simulating unbalance response, and demonstrating a match. In Chapter 3, we validate flexible body dynamics and the gyroscopic effect with a spinning disk hanging on a thin wire. In Chapter 4, we co-simulate an anisotropic flexible AMB rotor, presenting several useful diagrams of its unbalance response. In Chapter 5, we perform a simplified touchdown bearing interaction, showing that co-simulation can model solid body contact. We conclude with Chapter 6, summarizing results and suggesting future work.

Chapter 2

UNBALANCE RESPONSE OF AN AMB-SUPPORTED RIGID ROTOR

In this chapter, a rigid unbalanced rotor is run in an Adams-Simulink co-simulation and in a nonlinear pure Simulink model. The resulting frequency responses are practically identical and agree with theoretical calculations based on linearized magnetic forces, validating co-simulation of rigid AMB rotors.

The control of a rotor by active magnetic bearings (AMBs) is best understood by first examining the rigid case, where equations of motion are still tractable and results have clear conceptual interpretation. In this chapter, we develop the dynamics of a rigid rotor suspended by AMBs, create a pure Simulink theoretical model, and demonstrate that an Adams co-simulation produces vanishingly identical results.

To do so, we will take the following conceptual steps:

1. Since a rotor is made of metal, it cannot be repelled by an electromagnet, only attracted. Because of this, magnets are often arranged in opposed pairs so that force can be applied in both directions.
2. If an opposed pair provides 1D control, two perpendicular pairs provide 2D control; this arrangement is the basis of a conventional AMB.
3. The two ends of a rigid rotor can each be supported radially by an AMB, so two AMBs provide 3D control of a full rotor. (Only a thrust bearing, simply another opposed pair, is needed to control axial motion.)

These three bearings, shown notionally in Figure 2.1, are sufficient to suspend a rotor in most applications. In Section 2.1, we examine the 1D problem of suspension between an opposed pair of magnets and demonstrate a Simulink example. In Section 2.2, we apply these concepts to a 2D model of a complete AMB. In Section 2.3, we expand to the 3D problem of a rotor suspended by two AMBs, developing open-loop equations of motion in state-space form. In Section 2.4, we apply a simple control scheme and generate a closed-loop state space. Finally, in Section 2.5, we use the closed-loop state space in pure Simulink to validate an Adams co-simulation of the same rotor.

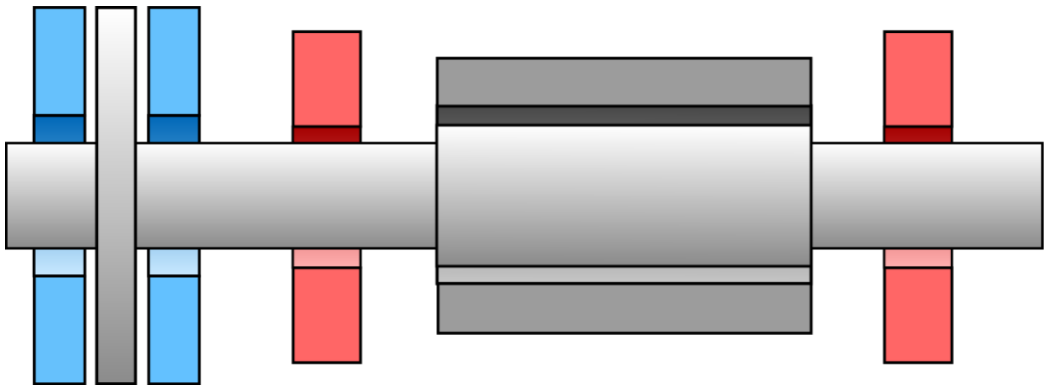


Figure 2.1. **A conventional AMB rotor**, with two radial AMBs (red), one thrust AMB (blue), and a driving motor (gray).

2.1. 1D dynamics: The magnet pair

The greatest challenge of magnetic levitation is that the force of an electromagnet is inherently both nonlinear and unstable. Consider the basic single-pole electromagnet

shown in Figure 2.2. Given certain assumptions[1] applicable to our case, the force that is applies on a metal shaft is given by

$$f = k\frac{i^2}{s^2},$$

where the *magnetic coupling constant* in henry-meters (H-m) is $k = \frac{1}{8}\mu_0 n^2 A$, and μ_0 is the magnetic permeability of free space, n is the number of wire coil turns of the pole, and A is the projected magnet pole area [21] [1]. We see from the expression that as the rotor approaches the magnet, the force increases toward the magnet, and as the rotor leaves, the force weakens, indicating system instability. Additionally, the direction of current does not change the direction of force since both terms are squared, making the force attractive only. Furthermore, these quantities are nonlinearly related, in this case rendering analytical solutions practically impossible.

However, these problems can be mitigated. The force directionality issue is solved by adding another magnet opposite the first, and if we assume that the gap s and current i remain close to some chosen values s_0 and i_0, the dynamics can be approximated by a linear relationship. With these, analysis of the two-magnet system becomes possible, which we perform next.

[1] A magnetic loop containing the iron and air gap has a constant cross-section over which magnetic flux is uniform and perpendicular; magnetization is negligible; we do not approach saturation flux density; air gap is very narrow relative to its area; air gap is approximately planar [1].

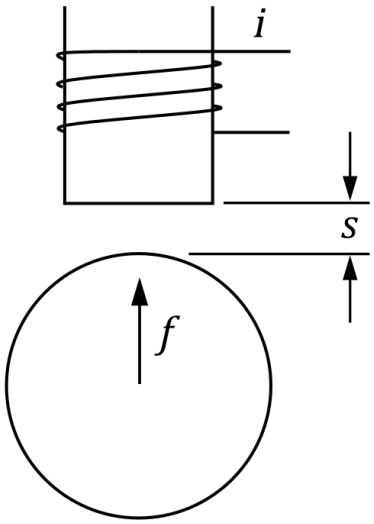


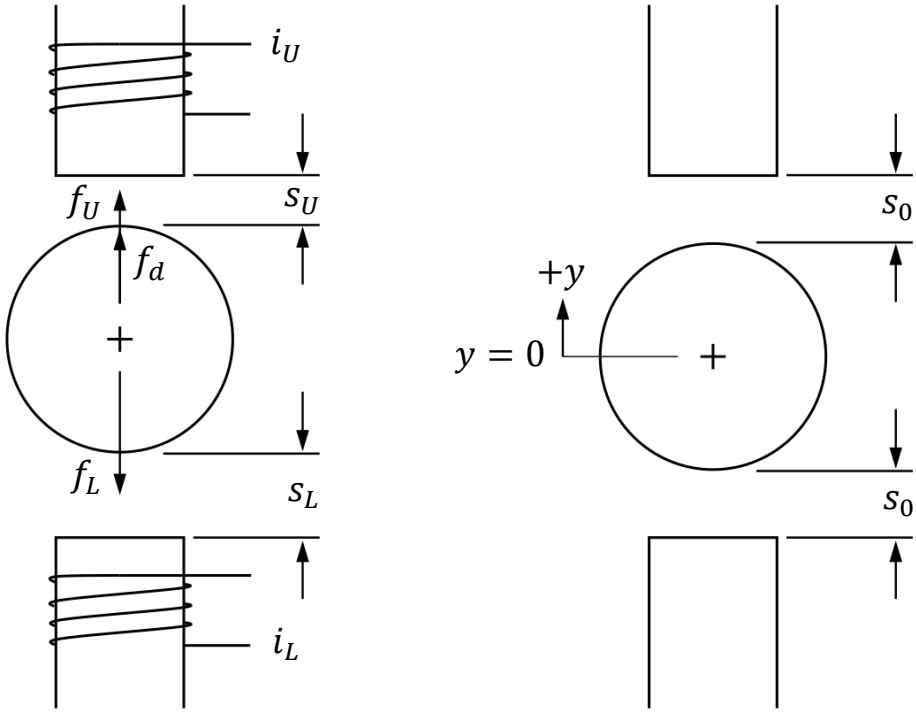
Figure 2.2. Force of a single electromagnet.

2.1.1. Magnet pair setup and equation of motion

For the reasons described above, the basic functional subunit of most magnetic bearing systems is not the single magnet but the opposed magnet pair, shown in Figure 2.3a. The upper and lower magnets each apply a force on the rotor, so the net vertical force on it, including any external disturbance forces f_d, is

$$f_{net} = f_U - f_L + f_d = k\frac{i_U^2}{s_U^2} - k\frac{i_L^2}{s_L^2} + f_d.$$

To obtain an equation of motion (EOM) for this system, we first establish a coordinate system where *y* points upward, and the origin is at the midpoint between the magnets. When the rotor's geometric center is at this location ($y = 0$), the air gaps are the same size, which we denote the *nominal gap*, s_0 (Figure 2.3b).



(a) (b)

Figure 2.3. The magnet pair.

This system as shown has two inputs (i_U and i_L) but only one output (y). This is often reduced to single-input-single-output (SISO) by applying current in *differential driving mode* (Figure 2.4). In this setup, a constant *bias current* i_0 is given to both magnets while a variable *differential current* i_y is added to the upper magnet and subtracted from the lower. Thus, the total current to the upper magnet is $i_U = i_0 + i_y$ and that to the lower is $i_L = i_0 - i_y$. In this way, if i_y is increased, the upper magnet strengthens and the lower weakens, adding upward net force, and vice versa if i_y is decreased. This yields a SISO system with input i_y and output y.

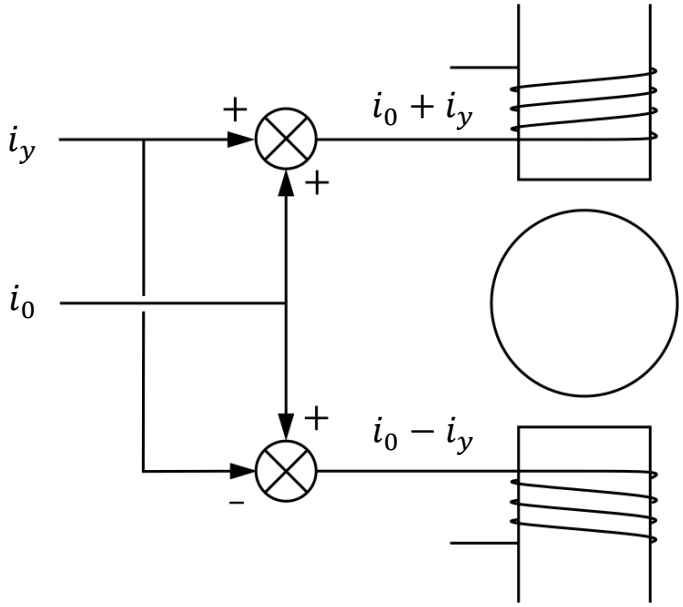


Figure 2.4. Current flow in differential driving mode.

From this, the EOM quickly follows. Newton's 2nd Law and the net force expression yield

$$m\ddot{y} = f_{net} = k\left(\frac{i_U^2}{s_U^2} - \frac{i_L^2}{s_L^2}\right) + f_d.$$

From the differential current setup, $i_U = i_0 + i_y$ and $i_L = i_0 - i_y$, and from our coordinates, $s_U = s_0 - y$ and $s_L = s_0 + y$, so the EOM is

$$m\ddot{y} = k\left(\frac{(i_0 + i_y)^2}{(s_0 - y)^2} - \frac{(i_0 - i_y)^2}{(s_0 + y)^2}\right) + f_d.$$

This equation relates the input current i_y to the output y. In this form, the strong nonlinearity of the system can be seen. To make analysis and control more manageable, we will linearize this equation, as is often done in industry.

2.1.2. Linearization of net force

Despite the nonlinearity of this system, it lends itself well to *linearization*, i.e. approximation by a linear relationship. To do this, we can linearize the force applied by each magnet. Taylor expanding our original general relationship $f = ki^2/s^2$ about $s = s_0$ (nominal rotor position) and $i = i_0$ (zero differential current), we find that

$$\begin{aligned} f = k\frac{i^2}{s^2} &\approx k\frac{i_0^2}{s_0^2} + (i - i_0)\frac{\partial}{\partial i}\left(k\frac{i^2}{s^2}\right)\Big|_{i=i_0, s=s_0} + (s - s_0)\frac{\partial}{\partial s}\left(k\frac{i^2}{s^2}\right)\Big|_{i=i_0, s=s_0} \\ &= k\frac{i_0^2}{s_0^2} + (i - i_0)\left(2k\frac{i_0}{s_0^2}\right) + (s - s_0)\left(-2k\frac{i_0^2}{s_0^3}\right) \\ &= k\frac{i_0^2}{s_0^2} + 2k\frac{i_0}{s_0^2}i - 2k\frac{i_0^2}{s_0^3}s. \end{aligned}$$

For the upper magnet, we substitute for i and s to find that $f_U = k\frac{i_0^2}{s_0^2} + 2k\frac{i_0}{s_0^2}(i_0 + i_y) - 2k\frac{i_0^2}{s_0^3}(s_0 - y)$, and for the lower magnet, we have $f_L = k\frac{i_0^2}{s_0^2} + 2k\frac{i_0}{s_0^2}(i_0 - i_y) - 2k\frac{i_0^2}{s_0^3}(s_0 + y)$, so the linearized net force expression simplifies to

$$\begin{aligned} f_{net} &= f_U - f_L + f_d \\ &= \left[k\frac{i_0^2}{s_0^2} + 2k\frac{i_0}{s_0^2}(i_0 + i_y) - 2k\frac{i_0^2}{s_0^3}(s_0 - y)\right] \\ &\qquad - \left[k\frac{i_0^2}{s_0^2} + 2k\frac{i_0}{s_0^2}(i_0 - i_y) - 2k\frac{i_0^2}{s_0^3}(s_0 + y)\right] + f_d \\ &= \left(4k\frac{i_0}{s_0^2}\right)i_y - \left(-4k\frac{i_0^2}{s_0^3}\right)y + f_d \\ &= k_i i_y - k_s y + f_d, \end{aligned}$$

where $k_i = \frac{4ki_0}{s_0^2}$ is the *force-current factor* and $k_s = -\frac{4ki_0^2}{s_0^3}$ is the *force-displacement factor*.[2] (Note that k_s takes a negative value, so if y increases, then $-k_s y$ also increases and contributes to upward force.) This new expression yields the linearized EOM

$$m\ddot{y} + k_s y = k_i i_y + f_d,$$

which is much easier to analyze and control.

2.1.3. PD control with a magnet pair in Simulink

The linearized EOM enables us to design a controller for suspending a mass in a magnet pair. This section provides an example, which will prepare us for full rotor control in Section 2.5. Magnetic bearing systems often incorporate current control, where current values are the control signal, and a separate electrical control loop applies voltage to ensure that the true current values are close to desired; the alternative is voltage control, where voltage itself is the control signal. We apply current control here and assume that the electrical control loop is fast enough to have negligible impact on the system's response.

One of the most basic control algorithms is *proportional-integral-derivative* (PID) control, where current is modified such that the system opposes the rotor's position error, time-accumulated position error, and velocity. Mathematically,

[2] These factors, defined as the change in net magnetic force per change in current or displacement, take on different expressions depending on the magnet arrangement. They will be different in Section 2.2.

$$i_y = -Py - I\int y\,dt - D\dot{y},$$

where P, I, and D are the *proportional*, *integral*, and *derivative gains* respectively, all positive scalars. The effect of integral control is to oppose the static load of gravity, so we can neglect both and apply *proportional-derivative* (PD) control, $i_y = -Py - D\dot{y}$, to a system that has no gravity. Plugging this into the linearized EOM,

$$m\ddot{y} + k_s y = k_i(-Py - D\dot{y}) + f_d,$$

and this rearranges to

$$m\ddot{y} + k_i D\dot{y} + (k_s + k_i P)y = f_d.$$

The equation can then be written in state-space form:

$$\begin{bmatrix}\dot{y}\\ \ddot{y}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\dfrac{k_s + k_i P}{m} & -\dfrac{k_i D}{m}\end{bmatrix}\begin{bmatrix}y\\ \dot{y}\end{bmatrix} + \begin{bmatrix}0\\ \dfrac{1}{m}\end{bmatrix} f_d,$$

or in abbreviated form,

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} + \mathbf{B}\mathrm{f_d}.$$

Response characteristics can be predicted from the *root locus* of the state space, which is a plot in the complex plane depicting the natural vibrations of the system according to their frequency (imaginary part) and decay (real part). The roots plotted on the root locus are the eigenvalues of the state matrix **A**. They must all be in the left half-plane for system stability, and decay is faster the farther the roots are to the left. As for the vertical direction, natural frequency increases with imaginary part

magnitude. Since the root locus depends on the PD gains, it can be used to inform the selection of P and D values.

To demonstrate this, we will use an example magnet pair system whose parameters are given in Table 2.1. The choice for the bias current i_0 can be arbitrary but should be high enough to allow for expected dynamic forces and low enough to avoid magnetic saturation and unnecessary power usage.

Table 2.1. Parameters for magnet pair control example.

Quantity	Value	Units
m	1	kg
s_0	0.0005	m
n	200	coils
A	0.001	m^2
i_0	0.6247	A

Plotting the eigenvalues of $\mathbf{A}$ for $P = 1500$ A/m as D varies from 0 to 6 A/(m/s) yields the root locus shown in Figure 2.5, colored for clarity. Since there are two states, this is a second-order system, with two roots. When $D = 0$, both roots lie on the imaginary axis, indicating oscillation with no decay; this makes sense because there is no source of damping in this case, so any disturbance will excite steady harmonic motion. As D increases from here, the roots have both a real part and imaginary part, indicating oscillations with decreasing frequency and increasing decay rate. When $D \approx 4$ A/(m/s), the two roots coincide at the point of critical damping, where decay is fastest and oscillation vanishes. As D further increases, there

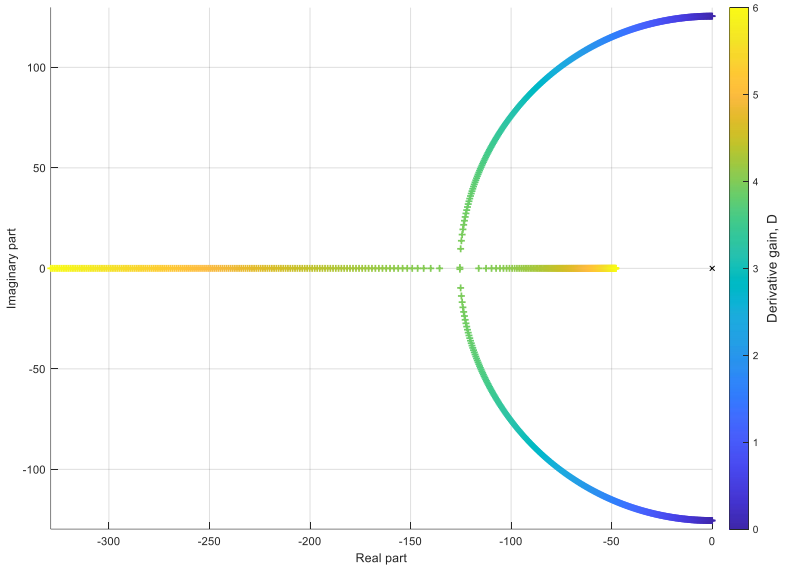


Figure 2.5. Root locus of magnet pair, $P = 1500$, D varies from 0 to 6.

is still no oscillation because the roots have no imaginary part, but since one root is closer to the origin, decay is slower than at the point of critical damping.

We now create a Simulink model representing the magnet pair system, shown in Figure 2.6. Progressing from left to right, the model first determines the displacement of the shaft, calculates the PD control signal, saturates that signal to the bias current magnitude, and adds that as a differential current to the upper and lower magnet; the nonlinear magnetic forces are then calculated by MATLAB function blocks, disturbances forces are added, and then Newton's 2nd Law yields the new shaft position that is fed back to previous portions of the loop. The responses of an upward impulse (50 N for 1 ms) for various D match the predictions from the root locus (Figure 2.7), showing respectively steady harmonic motion, damped oscillation, fast return with no oscillation, and slower return with no oscillation.

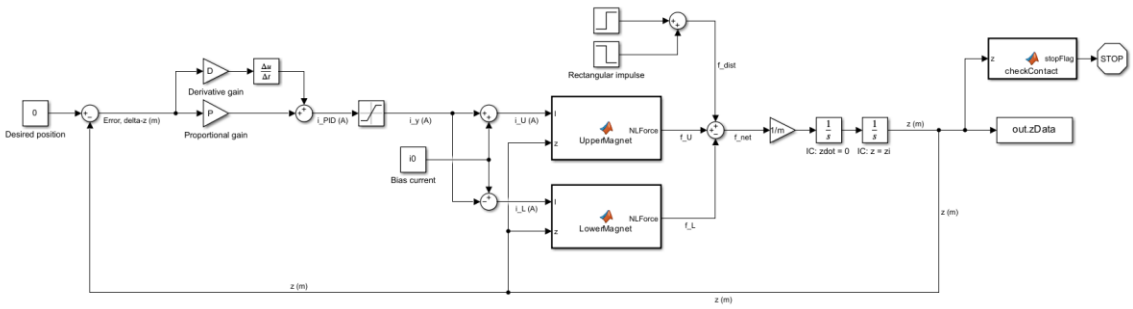


Figure 2.6. Magnet pair Simulink model.

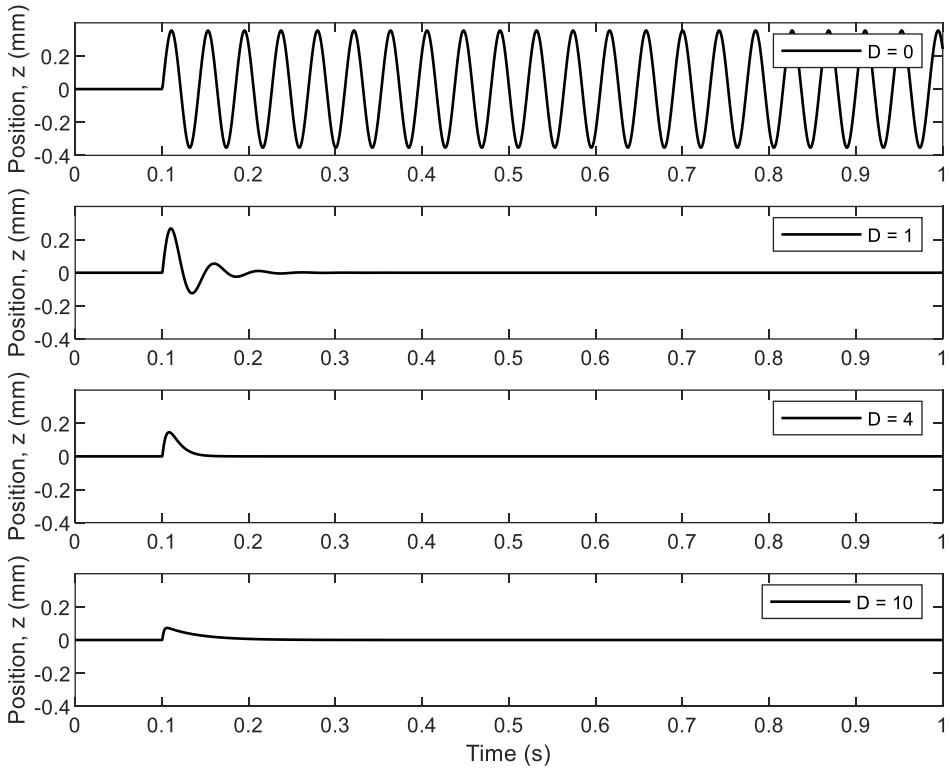


Figure 2.7. Magnet pair impulse response, $P = 1500$, $D = 0$, 1, 4, and 10.

We can also apply a *sine sweep* disturbance, which is an oscillating external force that gradually increases (or decreases) in frequency. This type of disturbance is often encountered in rotating machinery, since rotor imbalances cause forces with

frequencies matching shaft spin speed, which must be ramped up and down before and after operation. If the ramp-up is slow, it also approximates the rotor's steady state response at each frequency encountered. In this example, we will use the parameters in Table 2.2 to demonstrate this. To model a rotor-like unbalance response, we apply a disturbance force f_d with linearly increasing frequency and magnitude proportional to the square of the frequency. Mathematically,

$$f_d = K_d \omega^2 \cos(\omega t),$$

where K_d is constant gain and the disturbance frequency is given by $\omega(t) = \alpha t$, where α is a constant ramp rate in radians per second per second. In a real rotor, K_d is equal to the product of the equivalent mass imbalance m_u and its radius r_u from the center of the shaft.

Table 2.2. Disturbance and control parameters for magnet pair sine sweep test.

Quantity	Value	Units
K_d	1×10^{-5}	kg-m
α	1	rad/s^2
P	1500	A/m
D	0.2	A/(m/s)

The most common visualization of this frequency response is a *Bode plot* (BOH-dee), which displays the amplitude and phase of the response with respect to disturbance frequency. At each of the system's natural frequencies, there tends to be a peak in the amplitude plot due to resonance and a 180-degree shift in the phase plot.

In this case, the natural frequencies are the imaginary parts of the system's roots (eigenvalues of the state matrix), which solve

$$\det(\lambda \mathbf{I} - \mathbf{A}) = (\lambda - 0)\left(\lambda + \frac{2k_i D}{m}\right) - \left(\frac{2k_s + 2k_i P}{m}\right)(-1)$$

$$= \lambda^2 + \frac{2k_i D}{m}\lambda + \frac{2k_s + 2k_i P}{m} = 0.$$

The quadratic formula yields two λ whose imaginary parts have magnitude 125.3 rad/s, or 19.9 Hz.

The simulation was run, and a Bode plot for vibration in the x-direction was created from relative displacement maxima (using MATLAB's *findpeaks* function) and corresponding shaft angular positions (Figure 2.8). The noise in the phase data is caused by discrete points failing to exactly fall on amplitude peaks, which causes slight errors in the phase values, and the initial phase drop is caused by peak detection of minuscule numerical artifacts. We see that the amplitude peak at 20.3 Hz matches well. It is slightly above the theoretical value because the disturbance force increases in magnitude with frequency, which would move the peak to the right. The phase also behaves as expected, shifting by 180° across the natural frequency.

These methods applied to a magnet pair are similar to those we will use for a complete rigid rotor system. With these concepts and tools established, we can now study the full AMB as we progress toward modeling the full rigid rotor.

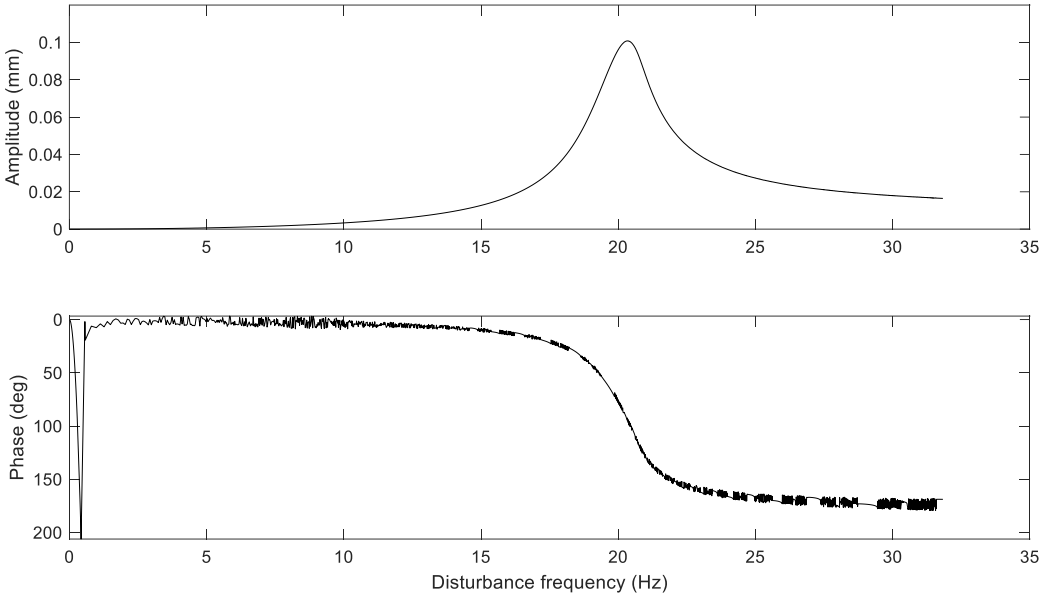


Figure 2.8. Bode plot of magnet pair sine sweep response.

2.2. 2D dynamics: The eight-pole bearing

After understanding the magnet pair, the idea of a full AMB quickly follows: A conventional AMB is not much more than two perpendicular magnet pairs, which control the rotor in the x- and y-directions, as shown notionally in Figure 2.9(a). There are two key differences, however. First, in most AMBs, each magnet is made of two separated poles, so the entire bearing has eight evenly spaced poles in total (these are known as *heteropolar* AMBs). Second, the pairs tend to be arranged so that their axes are 45-degrees from vertical for the convenience of symmetry with gravity, and coordinates are also aligned as such for mathematical convenience. The more realistic setup can be seen in Figure 2.9(b).

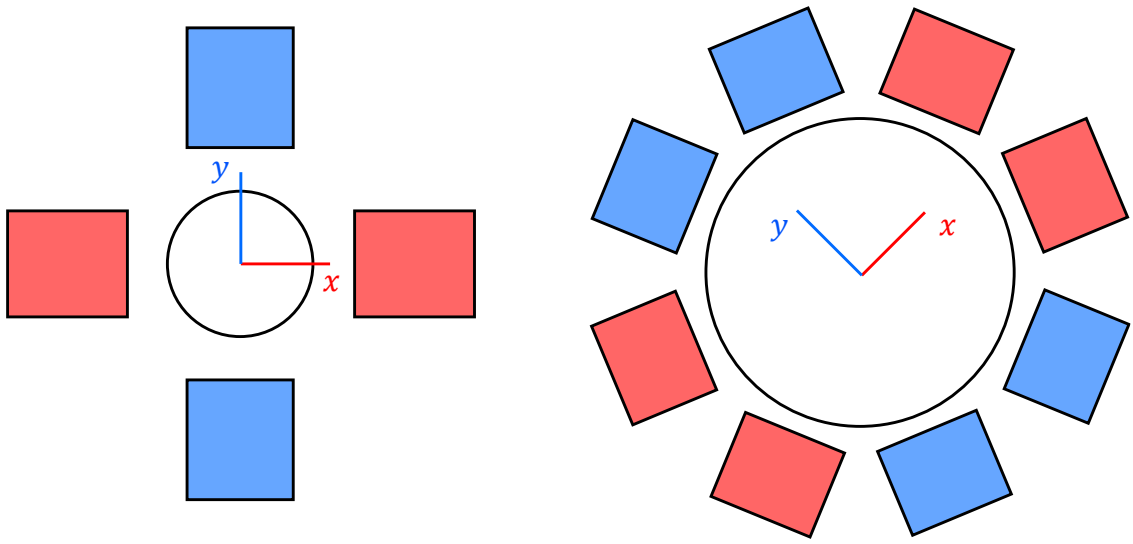


Figure 2.9. Notional (a) and realistic (b) conventional AMB configurations.

Since the poles are not perpendicular, the forces caused by the pairs are not truly independent; for example, the magnet pair aligned with the x-axis could apply some force in the y-direction if the bearing is closer to one pole than the other. However, to simplify our analysis, we will neglect this fact and treat the pairs as two independent 1D problems. In the y-direction then, we have the setup in Figure 2.10.

For the upper poles, if we additionally assume that the gap size can be approximated as $s_0 - y$ like in Section 2.1, we have two forces of magnitude

$$\left|\vec{f}_{U,pole}\right| = k\frac{i^2}{s^2} = k\frac{\left(i_0 + i_y\right)^2}{(s_0 - y)^2}.$$

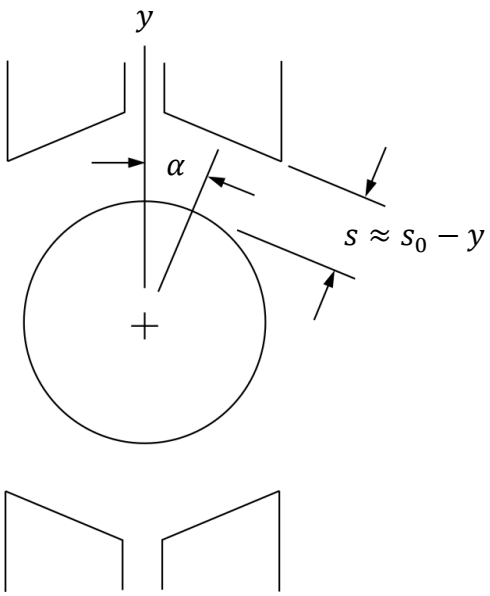


Figure 2.10. The four-pole pair, taken as a 1D problem.

Since each is at an angle α from the y-axis, the net vertical force provided by those two poles together is

$$f_{U,y} = 2k\frac{\left(i_0 + i_y\right)^2}{(s_0 - y)^2}\cos\alpha.$$

Similarly, the lower force is given by

$$f_{L,y} = 2k\frac{\left(i_0 - i_y\right)^2}{(s_0 + y)^2}\cos\alpha.$$

Therefore, the total force in the vertical direction is

$$f_y = 2k\left(\frac{\left(i_0 + i_y\right)^2}{(s_0 - y)^2} - \frac{\left(i_0 - i_y\right)^2}{(s_0 + y)^2}\right)\cos\alpha,$$

and likewise in the x-direction,

$$f_x = 2k\left(\frac{(i_0 + i_x)^2}{(s_0 - x)^2} - \frac{(i_0 - i_x)^2}{(s_0 + x)^2}\right)\cos\alpha.$$

As in the previous section, we would like linear expressions of these forces to aid in theoretical predictions. Examining y first, we linearize about $i_y = 0$ and $y = 0$. To obtain the first-order Taylor expansion, we need the following three quantities:

$$f_y\big|_{i_y=0,y=0} = 2k\left(\frac{(i_0 + i_y)^2}{(s_0 - y)^2} - \frac{(i_0 - i_y)^2}{(s_0 + y)^2}\right)\cos\alpha\Bigg|_{i_y=0,y=0} = 0$$

$$\frac{\partial f_y}{\partial i_y}\Bigg|_{i_y=0,y=0} = 2k\left(2\frac{(i_0 + i_y)}{(s_0 - y)^2} - 2\frac{-(i_0 - i_y)}{(s_0 + y)^2}\right)\cos\alpha\Bigg|_{i_y=0,y=0} = \frac{8ki_0}{s_0^2}\cos\alpha$$

$$\frac{\partial f_y}{\partial y}\Bigg|_{i_y=0,y=0} = 2k\left(-(-2)\frac{(i_0 + i_y)^2}{(s_0 - y)^3} - (-2)\frac{(i_0 - i_y)^2}{(s_0 + y)^3}\right)\cos\alpha\Bigg|_{i_y=0,y=0} = \frac{8ki_0^2}{s_0^3}\cos\alpha$$

The linearized form of the AMB force is then

$$\begin{aligned} f_y &\approx f_y\big|_{i_y=0,y=0} + (i_y - 0)\frac{\partial f_y}{\partial i_y}\Bigg|_{i_y=0,y=0} + (y - 0)\frac{\partial f_y}{\partial y}\Bigg|_{i_y=0,y=0} \\ &= 0 + i_y\left(\frac{8ki_0}{s_0^2}\cos\alpha\right) + y\left(\frac{8ki_0^2}{s_0^3}\cos\alpha\right) \\ &= k_i i_y - k_s y \end{aligned}$$

where $k_i = \frac{8ki_0}{s_0^2}\cos\alpha$, $k_s = -\frac{8ki_0^2}{s_0^3}\cos\alpha$, and $\alpha = 22.5°$ for eight evenly spaced poles. By symmetry, we have in the x-direction

$$f_x = k_i i_x - k_s x.$$

This completes our bearing analysis; we will not simulate a single bearing alone, but the above result is sufficient to move forward to the dynamics of the 3D rigid rotor.

2.3. 3D dynamics: The two-bearing rotor

As we increase complexity to a complete rotor, the concept is like that of any other bearing type: Two AMBs support the shaft in space, just as mechanical bearings do. With a 3D system now, we must keep track of more variables. The rotor itself can translate and tilt in both x and y, so this system now has 4 DOFs (we ignore the fifth, axial translation, since it is independent of radial motion and controllable with a simple bias pair), and the two bearings have two current inputs each. Thus, we have a 4-state system with 4 inputs, not counting disturbances that may be modeled.

We will begin with general rotor equations of motion from Krämer [22], show that they yield the equations given in Schweitzer and Maslen [1], and then follow Schweitzer and Maslen's analysis to achieve a state space representation of the rotor's open-loop dynamics.

2.3.1. Force expression in matrix form

Given bearings A and B, we can first write the four force equations (two directions for each bearing) and collect them into the matrix equation

$$\begin{bmatrix} f_{xA} \\ f_{xB} \\ f_{yA} \\ f_{yB} \end{bmatrix} = -\begin{bmatrix} k_{sA} & & & \\ & k_{sB} & & \\ & & k_{sA} & \\ & & & k_{sB} \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ y_A \\ y_B \end{bmatrix} + \begin{bmatrix} k_{iA} & & & \\ & k_{iB} & & \\ & & k_{iA} & \\ & & & k_{iB} \end{bmatrix} \begin{bmatrix} i_{xA} \\ i_{xB} \\ i_{yA} \\ i_{yB} \end{bmatrix},$$

where the force-current and force-displacement factors are defined above in Section 2.2. By defining symbols for the above vectors and matrices, we abbreviate this to

$$\mathbf{f}_\mathrm{b} = -\mathbf{K}_\mathrm{s}\mathbf{q}_\mathrm{b} + \mathbf{K}_\mathrm{i}\mathbf{i},$$

where $\mathbf{q}_\mathrm{b}$ is the vector of *bearing* coordinates of the shaft. This is Equation 8.2 of [1].

2.3.2. Open-loop equations of motion

We can now examine the dynamics of the overall system to eventually substitute the force expression. We begin with the general rotor setup shown in Figure 2.11.

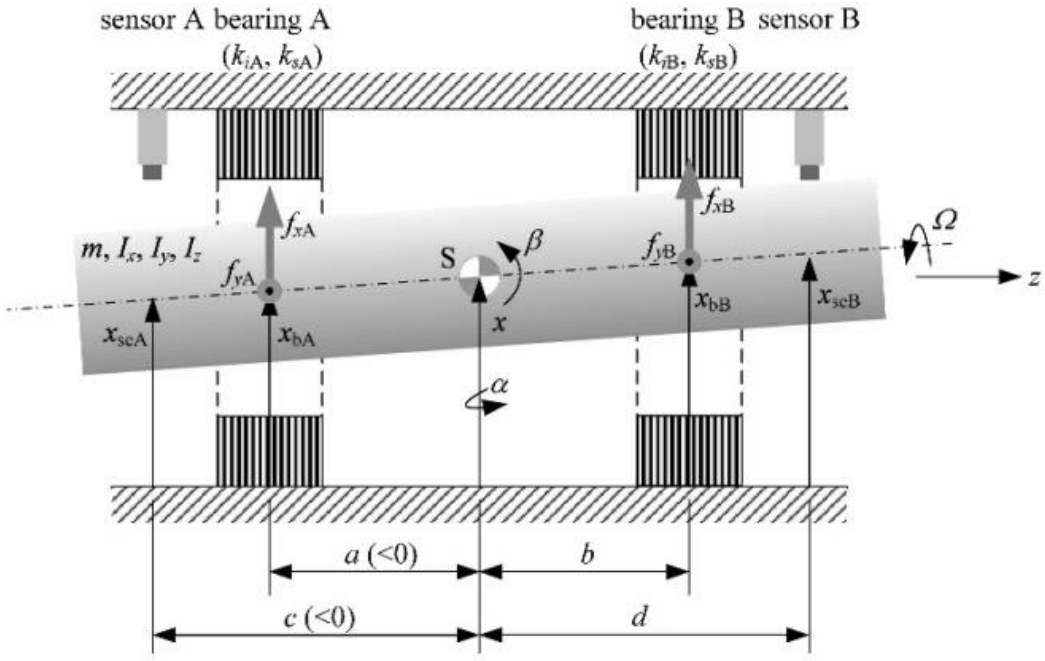


Figure 2.11. Rotor layout, coordinates, and nomenclature (Figure 8.1 of [1]).

From Krämer [22], we have the following rotor dynamics equations of motion after eliminating mechanical damping (since there is no contact in this system) and mechanical stiffness (since the rotor is rigid):

$$\begin{aligned} m\ddot{x} &= f_x \\ m\ddot{y} &= f_y \\ I_x\ddot{\alpha} + I_z\Omega\dot{\beta} &= T_x \\ I_y\ddot{\beta} - I_z\Omega\dot{\alpha} &= T_y, \end{aligned}$$

where I is moment of inertia and Ω is shaft speed. Replacing the force and moment terms with magnetic bearing forces, from Newton's 2nd Law we have $f_x = f_{xA} + f_{xB}$ and $f_y = f_{yA} + f_{yB}$, and for the moments, we have

$$T_x\hat{\imath} = a\hat{k} \times f_{yA}\hat{\jmath} + b\hat{k} \times f_{yB}\hat{\jmath} = \left(-af_{yA} - bf_{yB}\right)\hat{\imath}$$

$$T_y\hat{\jmath} = a\hat{k} \times f_{xA}\hat{\imath} + b\hat{k} \times f_{xB}\hat{\imath} = (af_{xA} + bf_{xB})\hat{\jmath}.$$

Thus, the equations of motion become

$$\begin{aligned} m\ddot{x} &= f_{xA} + f_{xB} \\ m\ddot{y} &= f_{yA} + f_{yB} \\ I_x\ddot{\alpha} + I_z\Omega\dot{\beta} &= -af_{yA} - bf_{yB} \\ I_y\ddot{\beta} - I_z\Omega\dot{\alpha} &= af_{xA} + bf_{xB}. \end{aligned}$$

To match the equations given in [1], we can multiply the third equation by -1, rearrange their order, and consolidate them into the matrix form

$$\begin{bmatrix} I_y & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & I_x & 0 \\ 0 & 0 & 0 & m \end{bmatrix} \begin{bmatrix} \ddot{\beta} \\ \ddot{x} \\ -\ddot{\alpha} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 & I_z\Omega & 0 \\ 0 & 0 & 0 & 0 \\ -I_z\Omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\beta} \\ \dot{x} \\ -\dot{\alpha} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} a & b & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_{xA} \\ f_{xB} \\ f_{yA} \\ f_{yB} \end{bmatrix},$$

which we will abbreviate to

$$\mathbf{M}\ddot{\mathbf{q}}_\mathrm{c} + \mathbf{G}\dot{\mathbf{q}}_\mathrm{c} = \mathbf{B}\mathbf{f}_\mathrm{b},$$

where $\mathbf{q}_\mathrm{c}$ is the vector of *center of gravity* (CG) coordinates of the shaft. This matrix equation is Equation 8.1a in [1]. Substituting the force expression from the previous section yields

$$\mathbf{M}\ddot{\mathbf{q}}_\mathrm{c} + \mathbf{G}\dot{\mathbf{q}}_\mathrm{c} = \mathbf{B}(-\mathbf{K}_\mathrm{s}\mathbf{q}_\mathrm{b} + \mathbf{K}_\mathrm{i}\mathbf{i}).$$

Now, we have here the shaft position vectors in both CG and bearing coordinates (i.e. the equation includes both $\mathbf{q}_\mathrm{c}$ and $\mathbf{q}_\mathrm{b}$), but we would like to write the equation using one system only. We see from Figure 2.11 that, assuming small angles,

$$\mathbf{q}_\mathrm{b} = \begin{bmatrix} x_A \\ x_B \\ y_A \\ y_B \end{bmatrix} = \begin{bmatrix} a & 1 & 0 & 0 \\ b & 1 & 0 & 0 \\ 0 & 0 & a & 1 \\ 0 & 0 & b & 1 \end{bmatrix} \begin{bmatrix} \beta \\ x \\ -\alpha \\ y \end{bmatrix} = \mathbf{B}^\mathrm{T}\mathbf{q}_\mathrm{c},$$

so we can substitute this for $\mathbf{q}_\mathrm{b}$ and manipulate to achieve the 2nd-order linear ODE

$$\mathbf{M}\ddot{\mathbf{q}}_\mathrm{c} + \mathbf{G}\dot{\mathbf{q}}_\mathrm{c} + \mathbf{B}\mathbf{K}_\mathrm{s}\mathbf{B}^\mathrm{T}\mathbf{q}_\mathrm{c} = \mathbf{B}\mathbf{K}_\mathrm{i}\mathbf{i}.$$

We have thus derived Equation 8.6 of [1] from the equations of [22].

2.3.3. Open-loop state space

We are now prepared to generate a state space representation. Since it will be useful to have an expression for the highest derivative of $\mathbf{q}$, we can rearrange our EOM by moving terms and then left-multiplying by the inverse of $\mathbf{M}$, obtaining

$$\ddot{\mathbf{q}}_{\mathrm{c}} = -\mathbf{M}^{-1}\mathbf{G}\dot{\mathbf{q}}_{\mathrm{c}} - \mathbf{M}^{-1}\mathbf{B}\mathbf{K}_{\mathrm{s}}\mathbf{B}^{\mathrm{T}}\mathbf{q}_{\mathrm{c}} + \mathbf{M}^{-1}\mathbf{B}\mathbf{K}_{\mathrm{i}}\mathbf{i}.$$

This can be written as the following state equation, where hereafter $\mathbf{q}$ represents $\mathbf{q}_{\mathrm{c}}$:

$$\begin{bmatrix}\dot{\mathbf{q}}\\ \ddot{\mathbf{q}}\end{bmatrix} = \begin{bmatrix}\mathbf{0}_{4\times4} & \mathbf{I}_{4\times4}\\ -\mathbf{M}^{-1}\mathbf{B}\mathbf{K}_{\mathrm{s}}\mathbf{B}^{\mathrm{T}} & -\mathbf{M}^{-1}\mathbf{G}\end{bmatrix}\begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{4\times4}\\ \mathbf{M}^{-1}\mathbf{B}\mathbf{K}_{\mathrm{i}}\end{bmatrix}\mathbf{i}.$$

The outputs are the measurements from the sensors, which can be determined from CG coordinates very similarly to the bearing coordinates. With this, we have a new matrix, $\mathbf{C}$:

$$\mathbf{q}_{\mathrm{s}} = \begin{bmatrix}x_{seA}\\ x_{seB}\\ y_{seA}\\ y_{seB}\end{bmatrix} = \begin{bmatrix}c & 1 & 0 & 0\\ d & 1 & 0 & 0\\ 0 & 0 & c & 1\\ 0 & 0 & d & 1\end{bmatrix}\begin{bmatrix}\beta\\ x\\ -\alpha\\ y\end{bmatrix} = \mathbf{C}\mathbf{q}.$$

Thus, the linear open-loop state space is

$$\dot{\mathbf{x}} = \mathbf{A}_{\mathrm{OL}}\mathbf{x} + \mathbf{B}_{\mathrm{OL}}\mathbf{i}$$

$$\mathbf{y} = \mathbf{C}_{\mathrm{OL}}\mathbf{x} + \mathbf{D}_{\mathrm{OL}}\mathbf{i}$$

$$\mathbf{x} = \begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix} \qquad \mathbf{y} = \mathbf{q}_{\mathrm{s}}$$

$$\mathbf{A}_{\mathrm{OL}} = \begin{bmatrix}\mathbf{0}_{4\times4} & \mathbf{I}_{4\times4}\\ -\mathbf{M}^{-1}\mathbf{B}\mathbf{K}_{\mathrm{s}}\mathbf{B}^{\mathrm{T}} & -\mathbf{M}^{-1}\mathbf{G}\end{bmatrix} \quad \mathbf{B}_{\mathrm{OL}} = \begin{bmatrix}\mathbf{0}_{4\times4}\\ \mathbf{M}^{-1}\mathbf{B}\mathbf{K}_{\mathrm{i}}\end{bmatrix} \quad \mathbf{C}_{\mathrm{OL}} = [\mathbf{C} \quad \mathbf{0}_{4\times4}] \quad \mathbf{D}_{\mathrm{OL}} = [\mathbf{0}_{4\times4}].$$

2.3.4. Augmented state space for disturbance forces

The above state space describes the motion of the system but provides no opportunity for external disturbance forces and moments. When simulating, we would like to be able to inject forces to model rotor imbalance or other perturbations. Mathematically, we desire a state space which models the EOM

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{G}\dot{\mathbf{q}} + \mathbf{B}\mathbf{K}_\mathrm{s}\mathbf{B}^\mathrm{T}\mathbf{q} = \mathbf{B}\mathbf{K}_\mathrm{i}\mathbf{i} + \mathbf{f}_\mathrm{d},$$

where the disturbances applied at the shaft CG are

$$\mathbf{f}_\mathrm{d} = \begin{bmatrix} T_{y,d} \\ f_{x,d} \\ -T_{x,d} \\ f_{y,d} \end{bmatrix}.$$

After we left-multiply by the inverse of the mass matrix and rearrange as we did before, we find

$$\ddot{\mathbf{q}} = -\mathbf{M}^{-1}\mathbf{G}\dot{\mathbf{q}} - \mathbf{M}^{-1}\mathbf{B}\mathbf{K}_\mathrm{s}\mathbf{B}^\mathrm{T}\mathbf{q} + \mathbf{M}^{-1}\mathbf{B}\mathbf{K}_\mathrm{i}\mathbf{i} + \mathbf{M}^{-1}\mathbf{f}_\mathrm{d},$$

and the resulting augmented state space is

$$\dot{\mathbf{x}} = \mathbf{A}_\mathrm{OL}\mathbf{x} + \mathbf{B}_\mathrm{OL}\mathbf{i} + \mathbf{E}_\mathrm{OL}\mathbf{f}_\mathrm{d}$$

$$\mathbf{y} = \mathbf{C}_\mathrm{OL}\mathbf{x} + \mathbf{D}_\mathrm{OL}\mathbf{i}$$

where $\mathbf{E}_\mathrm{OL} = \begin{bmatrix} \mathbf{0}_{4\times4} \\ \mathbf{M}^{-1} \end{bmatrix}$, and the other matrices are the same as before.

2.4. Control of the 3D rotor

Having derived the state space in the previous section, we are ready to apply a control algorithm, which changes nothing about the open-loop state space except the values or expressions taken by the currents in the vector **i**. In this section, we will apply the simplest control algorithm, *decentralized control*, in which each bias pair is operated by an independent PID controller. We will see that the current vector **i** will take on a simple expression of the other variables of the system, which we can substitute and manipulate into a closed-loop state space from which we can predict the system's stability and natural frequencies.

2.4.1. Closed-loop state space for decentralized PD control

In decentralized control, all four bias current pairs (two per bearing) are operated by independent PID controllers (Figure 2.12). For simplicity, we will assume as before that the integral control has accounted for gravity, so we can therefore examine the dynamics due to pure PD control.

Like in the two-magnet example from Section 2.1, a PD control scheme applies current given by a linear combination of shaft position and velocity:

$$\begin{bmatrix} i_{xA} \\ i_{xB} \\ i_{yA} \\ i_{yB} \end{bmatrix} = -\begin{bmatrix} P_A & 0 & 0 & 0 \\ 0 & P_B & 0 & 0 \\ 0 & 0 & P_A & 0 \\ 0 & 0 & 0 & P_B \end{bmatrix}\begin{bmatrix} x_{seA} \\ x_{seB} \\ y_{seA} \\ y_{seB} \end{bmatrix} - \begin{bmatrix} D_A & 0 & 0 & 0 \\ 0 & D_B & 0 & 0 \\ 0 & 0 & D_A & 0 \\ 0 & 0 & 0 & D_B \end{bmatrix}\begin{bmatrix} \dot{x}_{seA} \\ \dot{x}_{seB} \\ \dot{y}_{seA} \\ \dot{y}_{seB} \end{bmatrix},$$

or in abbreviated form, $\mathbf{i} = -\mathbf{P}\mathbf{q}_\mathbf{s} - \mathbf{D}\dot{\mathbf{q}}_\mathbf{s}$.

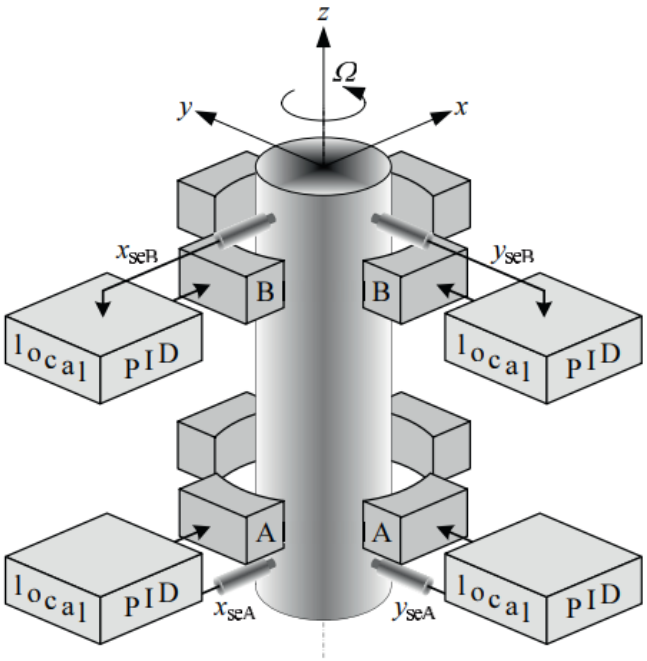


Figure 2.12. Decentralized PID control structure (Figure 8.2 of [1]).

We already know that $\mathbf{q}_s = \mathbf{Cq}$. Substituting these into the main EOM yields

$$\mathbf{M\ddot{q}} + \mathbf{G\dot{q}} + \mathbf{BK}_s\mathbf{B}^T\mathbf{q} = \mathbf{BK}_i(-\mathbf{PCq} - \mathbf{DC\dot{q}}),$$

which can be rearranged to

$$\mathbf{M\ddot{q}} + (\mathbf{G} + \mathbf{D}_{CL})\mathbf{\dot{q}} + (\mathbf{K}_{sS} + \mathbf{K}_{CL})\mathbf{q} = \mathbf{0}_{4\times4},$$

$$\mathbf{K}_{CL} = \mathbf{BK}_i\mathbf{PC} \qquad \mathbf{D}_{CL} = \mathbf{BK}_i\mathbf{DC} \qquad \mathbf{K}_{sS} = \mathbf{BK}_s\mathbf{B}^T.$$

Like for the open-loop model, we can construct a closed-loop state space. Replacing matrix coefficients, we find the following model, whose $\mathbf{A}_{CL}$ is Equation 8.12a of [1]:

$$\dot{\mathbf{x}} = \mathbf{A}_{\mathrm{CL}}\mathbf{x}$$

$$\mathbf{y} = \mathbf{C}_{\mathrm{CL}}\mathbf{x}$$

$$\mathbf{A}_{\mathrm{CL}} = \begin{bmatrix} \mathbf{0}_{4\times4} & \mathbf{I}_{4\times4} \\ -\mathbf{M}^{-1}(\mathbf{K}_{\mathrm{sS}} + \mathbf{K}_{\mathrm{CL}}) & -\mathbf{M}^{-1}(\mathbf{G} + \mathbf{D}_{\mathrm{CL}}) \end{bmatrix} \qquad \mathbf{C}_{\mathrm{CL}} = [\mathbf{C} \quad \mathbf{0}_{4\times4}].$$

We can then calculate the eigenvalues of $\mathbf{A}_{\mathrm{CL}}$ to predict the system's response.

2.5. Unbalance response in pure Simulink and co-simulation

This section presents an example rigid AMB rotor to which we will apply the above analysis. We will create a simulation of the example rotor in Adams-Simulink co-simulation, use the closed-loop state space to construct a pure theoretical model, and then use $\mathbf{A}_{\mathrm{CL}}$ to validate the results of both. The methods used here will closely follow the simulation example from Section 2.1.

2.5.1. Setup and theoretical analysis

The example rotor system is a cylindrical shaft with an offset disk, shown in Figure 2.13. The rotor's mass is made larger to yield lower resonant speeds and therefore reduce necessary runtime for a rotor ramp up, and the shaft is thickened so that the shaft CG is offset from the unbalanced disk and tilting modes will be more visible. A material density of 7801 kg/m^3, the default density of steel in Adams, defines the rotor's mass, moments of inertia, and CG location.

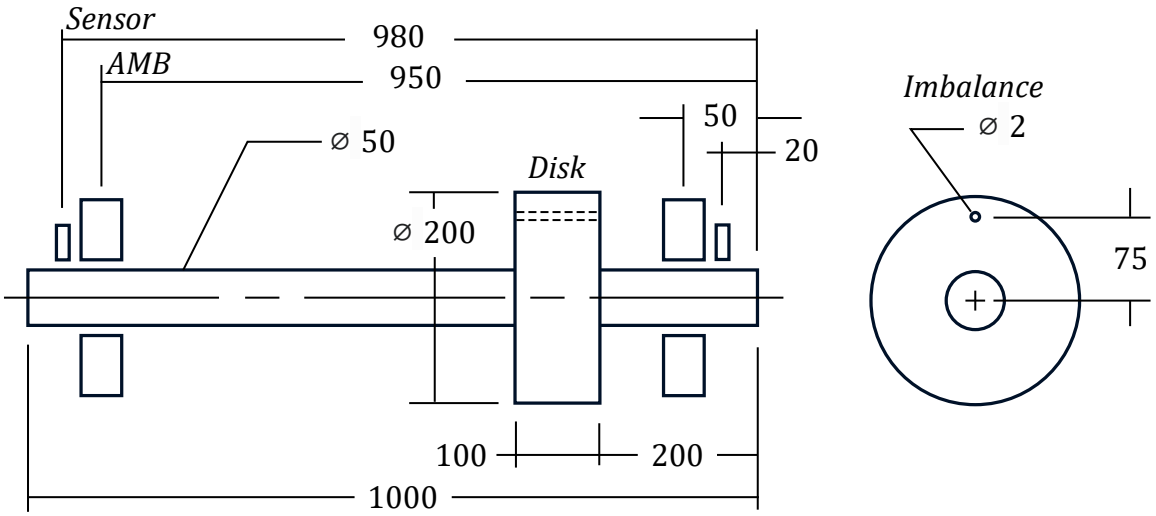


Figure 2.13. Co-simulation rigid rotor dimensions (mm).

The gains in this example are chosen to keep resonant speeds small, and damping is low so that amplitude peaks could be clearly seen; it is of course not good practice to allow such resonance in real applications, but we do so here simply to verify the model with theory. The quantities N_A and N_B are the filter coefficients for the PID controllers, and T_{drive} is the drive torque used to ramp up the speed of the shaft for the simulation. The drive torque is kept small to keep ramp rate relatively low; if the shaft speed passes through the natural frequency too quickly, inertial effects distort the peak. The resulting system parameters for this example are provided in Table 2.3.

From these values, $\mathbf{A}_{\mathrm{CL}}$ and its eigenvalues can be calculated in MATLAB and plotted on the complex plane, which yields the root locus in Figure 2.14. All poles have negative real part, indicating exponential decay and thus stability. The poles closer to the origin have multiplicity 2 and do not change with shaft speed. According to [1], these represent modes of parallel motion, which are not affected by the gyroscopic

Table 2.3. Rotor and control parameters for pure and co-simulation comparison.

Quantity	Value	Units
ρ	7801	kg/m^3
m	38.29	kg
$I_x = I_y$	1.9334	kg m^2
I_z	0.1268	kg m^2
a	-0.3	m
b	0.6	m
c	-0.33	m
d	-0.63	m
$s_{0A} = s_{0B}$	0.0005	m
$n_A = n_B$	200	coils
$A_A = A_B$	0.002	m^2
$i_{0A} = i_{0B}$	2.7336	A
$P_A = P_B$	6.5×10^3	A/m
$D_A = D_B$	0.25	A/(m/s)
$N_A = N_B$	5000	s^{-1}
T_{drive}	1.25	N-m

effect and therefore will not change with angular velocity. The other poles begin at the middle of the segments where the colors meet and then separate as Ω increases. These are the conical modes of the system, since the tilting of the shaft would sweep a double cone in space, and these do depend on angular velocity.

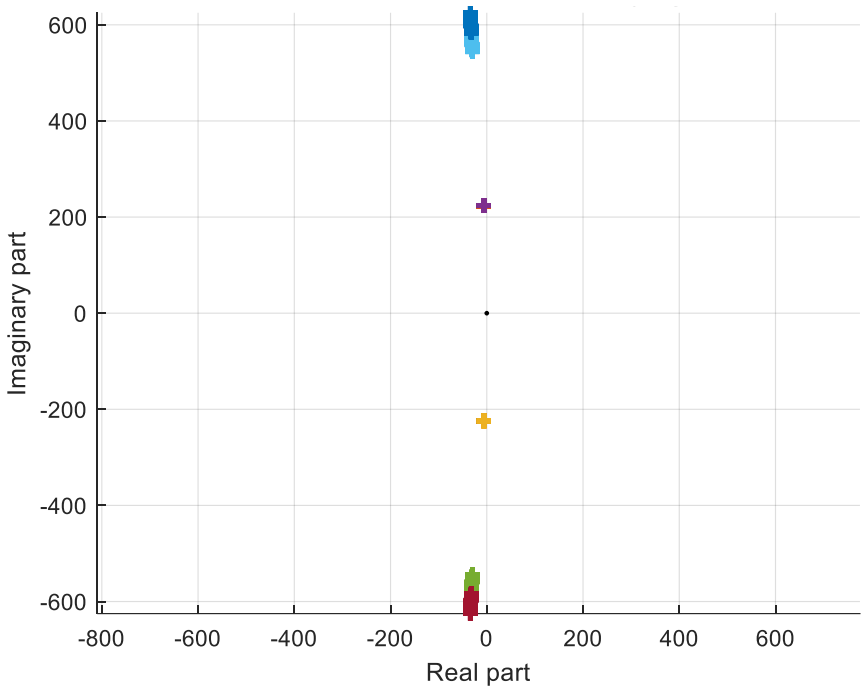


Figure 2.14. Closed-loop root locus of example rigid rotor as shaft speed varies.

An even more useful analysis tool in our case is the Campbell diagram, which plots natural frequencies ω_n with respect to Ω (Figure 2.15). The imaginary parts of the eigenvalues are the natural frequencies of the closed-loop system. When we plot these, we see two frequencies that do not change with speed and the remainder that separate with increasing speed. On top of these curves, we plot the lines $\omega_n = \pm \Omega$. Where these lines intersect the natural frequency curves are where shaft spin speed equals the natural frequency; if there is any unbalance in the rotor, these are the operating points at which the system will resonate. These are termed the *critical speeds* of the system. In this case, critical speeds lie near 36 Hz, 90 Hz, and 96 Hz. We expect the largest parallel shaft motion at 36 Hz and the largest wobbling of the shaft near 90-96 Hz.

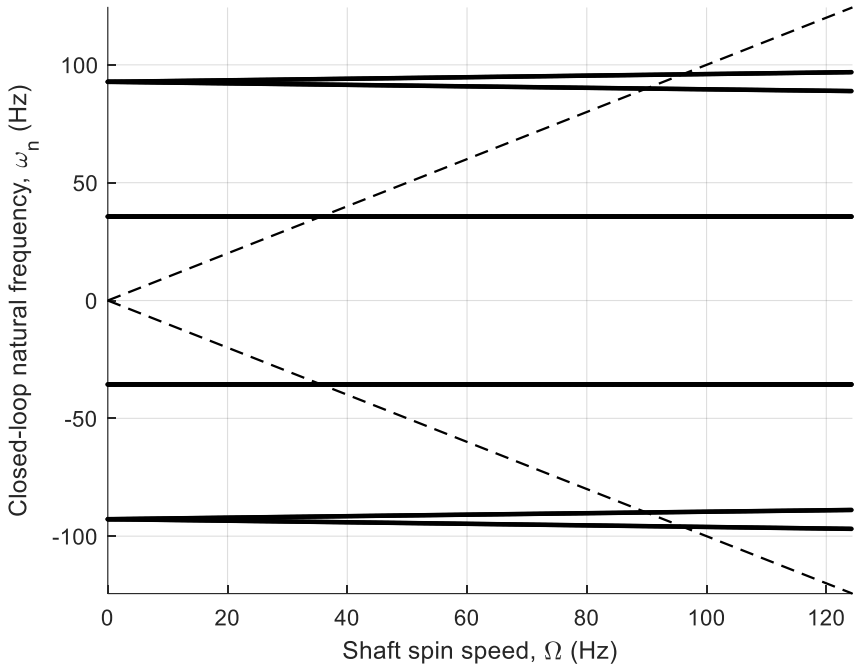


Figure 2.15. Closed-loop Campbell diagram of example rigid rotor.

2.5.2. Pure Simulink model

With theorical predictions complete, we move on to create a pure simulation of the rotor's unbalance response. The general closed-loop feedback architecture shown in Figure 2.16 will be used, with the control law and plant constructed according to the models that we have developed. The simulated system will be built in Simulink.

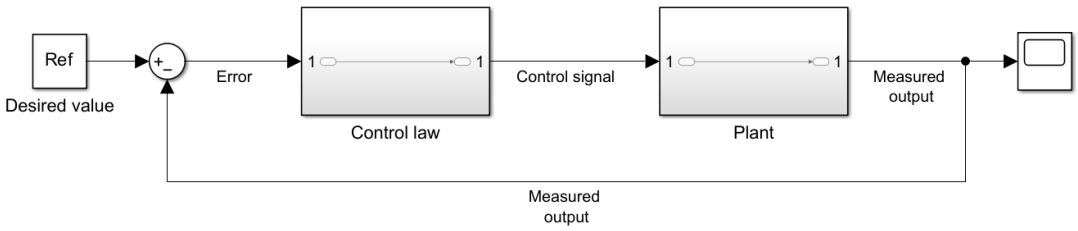


Figure 2.16. General closed-loop feedback architecture.

2.5.2.1. Control law

The control law is a PD controller applied to the four differential currents (Figure 2.17). Since the magnet pairs within a given AMB follow the same control law by symmetry, the two control signals can be simultaneously calculated elementwise using a Simulink PID block.

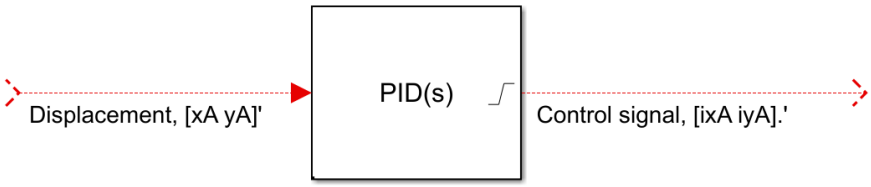


Figure 2.17. Control law applied at Bearing A.

The differential current magnitude must be saturated to that of the bias current to prevent a change in current direction in any electromagnet; if such a change occurs, the controller will expect the magnet to begin repelling the shaft while in fact it still attracts, destabilizing the system. Saturation can be added in the "Saturation" tab of the PID block. Additionally, the control signal must be filtered to prevent the derivative control from amplifying jumps in the Adams co-simulation output; filtering is included in the PID block and governed by the filter coefficient N.

2.5.2.2. Nonlinear plant

In previous sections, we linearized magnetic force to perform theoretical calculations, but we need not do so for the simulation itself. To keep the model more realistic, we can retain nonlinearity in the plant and directly calculate magnetic forces. Recall from

Section 2.2 that the forces applied by each eight-pole AMB can be approximated as the following, where *x* and *y* here are shaft displacements in local *bearing* coordinates:

$$f_y = 2k\left(\frac{\left(i_0 + \Delta i_y\right)^2}{(s_0 - y)^2} - \frac{\left(i_0 - \Delta i_y\right)^2}{(s_0 + y)^2}\right)\cos\alpha$$

$$f_x = 2k\left(\frac{(i_0 + \Delta i_x)^2}{(s_0 - x)^2} - \frac{(i_0 - \Delta i_x)^2}{(s_0 + x)^2}\right)\cos\alpha.$$

These can be inserted directly into the force vector, $\mathbf{f}_\mathrm{b} = [f_{xA} \quad f_{xB} \quad f_{yA} \quad f_{yB}]^\mathrm{T}$**.** Since this vector is in bearing coordinates but the plant calculates shaft position in *CG* coordinates ($\mathbf{q}_\mathrm{c}$), we must use $\mathbf{q}_\mathrm{b} = \mathbf{B}^\mathrm{T}\mathbf{q}_\mathrm{c}$ as derived in Section 2.3. With this information, $\mathbf{f}_\mathrm{b}$ can be calculated as a function of the bearing coordinates $\mathbf{q}_\mathrm{b}$ and magnet currents $\mathbf{i}$, and our EOM can be written

$$\mathbf{M}\ddot{\mathbf{q}}_\mathrm{c} + \mathbf{G}\dot{\mathbf{q}}_\mathrm{c} = \mathbf{B}\mathbf{f}_\mathrm{b}(\mathbf{q}_\mathrm{b}, \mathbf{i}) + \mathbf{f}_\mathrm{d},$$

or after algebra,

$$\ddot{\mathbf{q}}_\mathrm{c} = -\mathbf{M}^{-1}\mathbf{G}\dot{\mathbf{q}}_\mathrm{c} + \mathbf{M}^{-1}\mathbf{B}\mathbf{f}_\mathrm{b}(\mathbf{q}_\mathrm{b}, \mathbf{i}) + \mathbf{M}^{-1}\mathbf{f}_\mathrm{d}.$$

Implemented in Simulink (Figure 2.18), the nonlinear forces are calculated by a MATLAB function block called *NLbearingForce*, and the above matrix equation is calculated in the *nonlinearEOM* block. The output equation yielding $\mathbf{q}_\mathrm{s}$ is calculated outside of the MATLAB function block since an algebraic loop is formed if the output is calculated before integration. Added to this subsystem is another MATLAB function block, *checkContact*, that stops the simulation if the shaft contacts the bearing (i.e. if

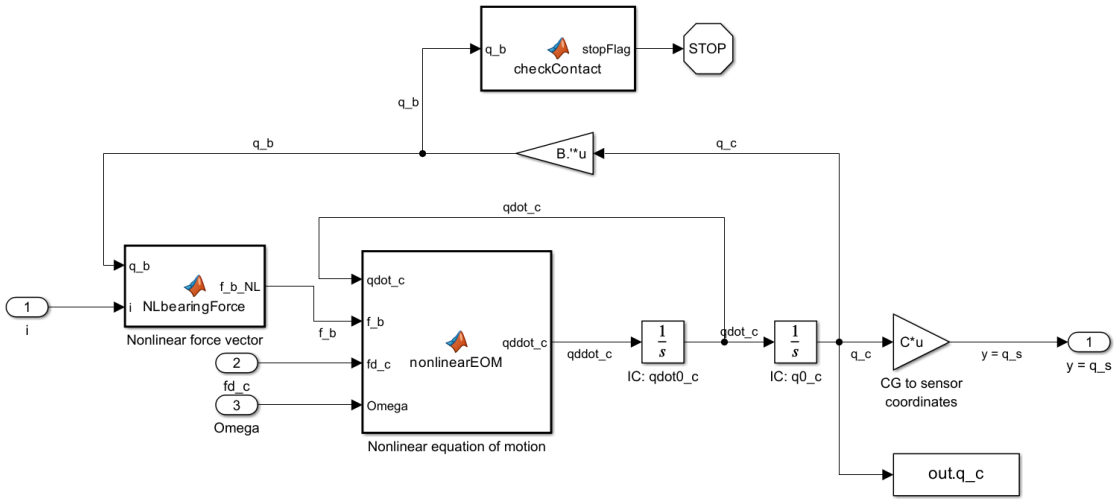


Figure 2.18. Nonlinear plant Simulink subsystem.

the displacement magnitude is greater than or equal to s_0), which would cause a division by zero in the magnetic force calculation. The initial conditions on the integrator blocks are 4x1 vectors of initial velocities and positions, in our case all zeros. The inputs to the overall plant are the control signal currents $\mathbf{i}$, the vector of disturbance forces and moments $\mathbf{f}_d$, and the shaft spin speed Ω on which the state space depends. The latter two are determined by the subsystem shown in Figure 2.19, whose input is a constant torque that generates a linearly increasing shaft speed. The

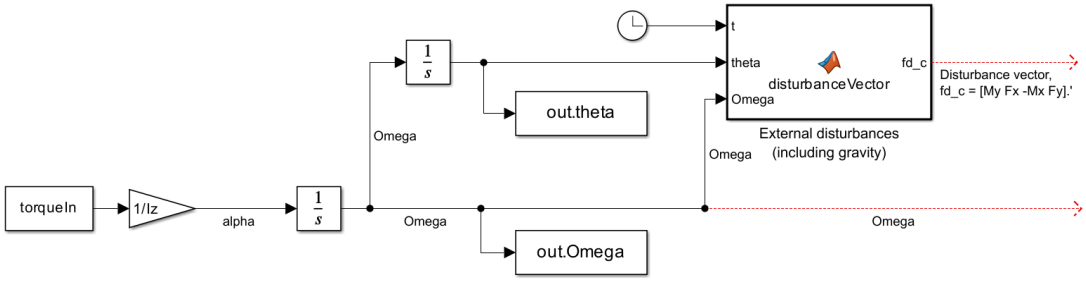


Figure 2.19. Disturbance and shaft speed inputs to plant.

disturbanceVector block calculates the forces caused by a mass imbalance at an axial location z_u from the CG, where z_u is positive if the vector pointing from the CG to the unbalance is in the positive z-direction. A mass imbalance on a spinning rotor applies a disturbance force of magnitude $m_u r_u \Omega^2$ in the direction $\theta + \phi_u$, where m_u is the mass of the unbalance, r_u is the *eccentricity* of the unbalance (i.e. its radius from the shaft's geometric center), θ is the angular position of the shaft, and ϕ_u is the angular position of the unbalance with respect to the shaft. Its expression in full is the following [3] :

$$\vec{f_u} = f_{u,x}\hat{\imath} + f_{u,y}\hat{\jmath} = m_u r_u \Omega^2 \cos(\theta + \phi_u)\,\hat{\imath} + m_u r_u \Omega^2 \sin(\theta + \phi_u)\hat{\jmath}.$$

The moments applied to the shaft CG are then

$$T_{u,x}\hat{\imath} + T_{u,y}\hat{\jmath} = z_u\hat{k} \times \left(f_{u,x}\hat{\imath} + f_{u,y}\hat{\jmath}\right) = z_u f_{u,x}\hat{\jmath} - z_u f_{u,y}\hat{\imath},$$

so we have $T_{u,x} = -z_u f_{u,y}$ and $T_{u,y} = z_u f_{u,x}$. We can then write the code governing the input disturbance as shown in Figure 2.20.

Gravity is neglected in the above calculation, assumed to have been accounted for by integral control, but it is important to note that since our coordinate axes are rotated, gravity would *not* point in the negative y-direction if included.

[3] This expression only accounts for radial forces due to unbalance, but tangential forces may exist if the shaft spin speed changes quickly; we assume here that the speed ramp rate is small enough to neglect the latter.

```
function fd_c = disturbanceVector(t,theta,Omega,...        % Inputs
                                  m,g,mru,zu,phaseu_deg)   % Parameters

% Unbalance forces
fu_x = mru*Omega^2*cos(theta + phaseu_deg*pi/180);
fu_y = mru*Omega^2*sin(theta + phaseu_deg*pi/180);

% Forces (N) and moments (N-m) in CG coordinates
fx = fu_x;
fy = fu_y;
Tx = -zu*fu_y;
Ty = zu*fu_x;

fd_c = [Ty fx -Tx fy].';        % Force vector in CG coordinates

end
```

Figure 2.20. Code for the *disturbanceVector* MATLAB function block.

2.5.2.3. Full simulated model

Together, the control laws and plant are combined into the model given in Figure 2.21, with subsystems grouped into colored regions. One PID controller for each bearing, the nonlinear plant, and the disturbance input to the plant can be seen. The visual complexity from multiplexors is superficial; they are present only for proper data routing. With this and a MATLAB script that defines all necessary input parameters, the pure Simulink model is complete.

Instead of showing the pure Simulink results now, we will create the Adams co-simulation model first and then show the results of both simulations together, comparing them with theoretical predictions.

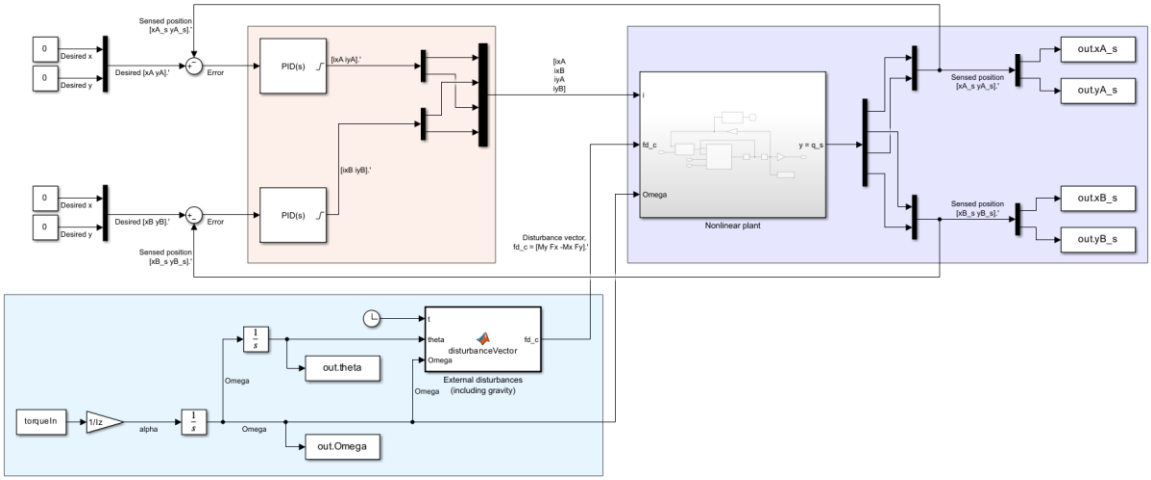


Figure 2.21. **Pure Simulink model of example rigid rotor**, with controller (red), plant (purple), and disturbance input (cyan).

2.5.3. Adams co-simulation model

To complete this chapter, we would like to verify that an Adams co-simulation provides the same output as a model built on theory. Only the plant will change in this case; the control system and magnetic force calculation will be identical to that of the pure Simulink model.

The Adams model for this example rotor (Figure 2.22) was constructed in the student version of Adams View 2022.1 following the general steps below:

1. Create the main shaft extending from the origin into the positive z-direction.
2. Create the disk, with a hole to accommodate the shaft.
3. Attach the disk to the shaft using a fixed joint.
4. Create markers along the shaft at the bearing and sensor locations.
5. Attach forces at the bearing locations to be used as net force inputs.

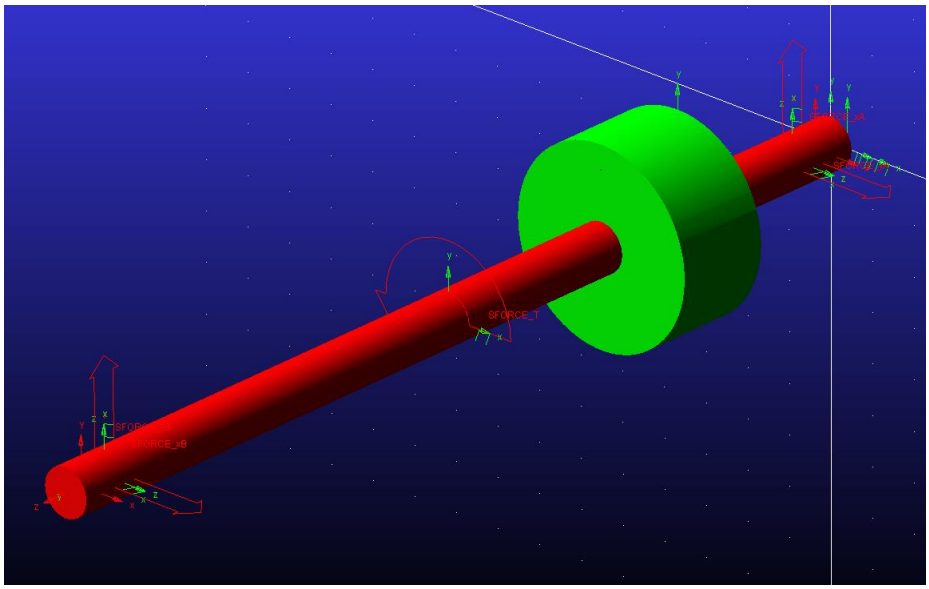


Figure 2.22. Adams model for pure and co-simulation comparison.

6. Attach a torque at the shaft CG marker to be used as motor torque input.
7. Create state variables, 16 in total:
    a. Four to measure x- and y-displacements of the bearing markers
    b. Four to measure x- and y-displacements of the sensor markers
    c. Two to measure the x- and y-displacement of the disk center
    d. One to measure the angular velocity of the shaft about the origin in the z-direction
    e. Four to be used as inputs for the bearing forces from Step 5
    f. One to be used as input for the motor torque from Step 6
8. Assign the values of the input state variables as the force and torque functions.
9. Create a cylinder and use it to cut the hole in the disk. This step should be performed after Step 7c so that disk displacement is measured with respect to its geometric center, not its off-center CG.

With the model complete, the Adams plant export for Simulink can be created:

1. Navigate to the “Plugins” tab, the “Controls” plugin, and “Plant Export.” Fill in the appropriate fields of the window (Figure 2.23). Press “OK,” which generates several files in the working directory of the Adams model.
2. One of the generated files is a MATLAB .m file. Run the file, which fills the workspace with appropriate variables and paths necessary to run the co-simulation.
3. Run the command “adams_sys” in the command window, which opens a Simulink window containing an orange block of the Adams plant (Figure 2.24a). The red blocks will not be used.
4. Double-click the orange block, then double-click the red “MSC Software” block (Figure 2.24b), then navigate to “Simulation Mode” in the window that appears, and change it to “continuous” (Figure 2.25).

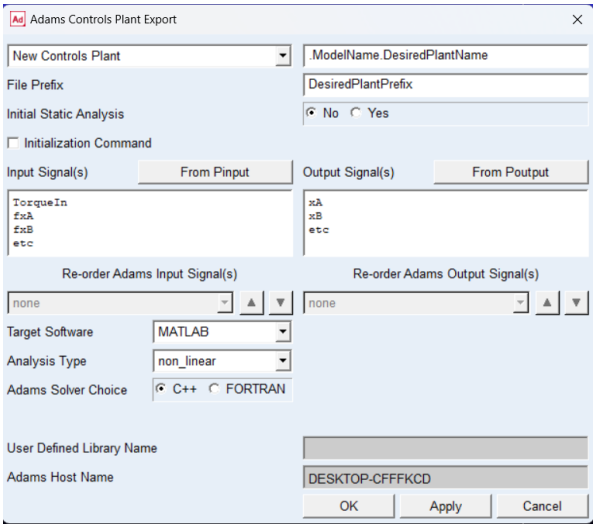
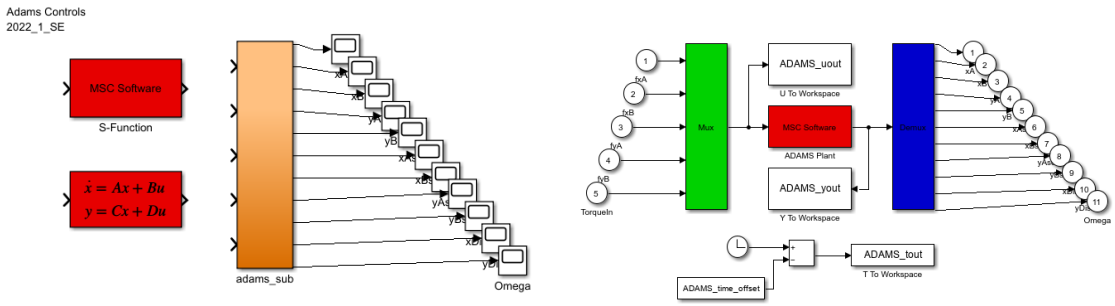


Figure 2.23. Plant export window in Adams.



(a) (b)

Figure 2.24. Adams plant block, colored orange (a) and its contents (b).

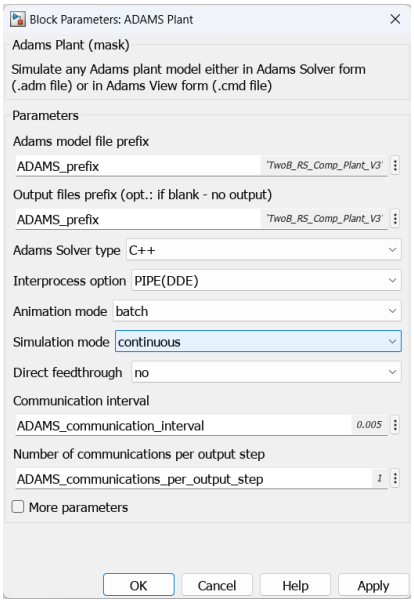


Figure 2.25. **Contents of "MSC Software" block from Figure 2.24b**, with continuous simulation mode selected.

The block can now be implemented into the Simulink model. Again, only the plant will change, becoming the subsystem in Figure 2.26. In this case, the unbalance forces are modeled within the Adams plant, so we no longer need the disturbance calculation; only an input torque is necessary. The x- and y-positions of the disk and

the shaft spin speed are included as outputs, along with sensor and bearing coordinate positions. The full Adams co-simulation model is shown in Figure 2.27.

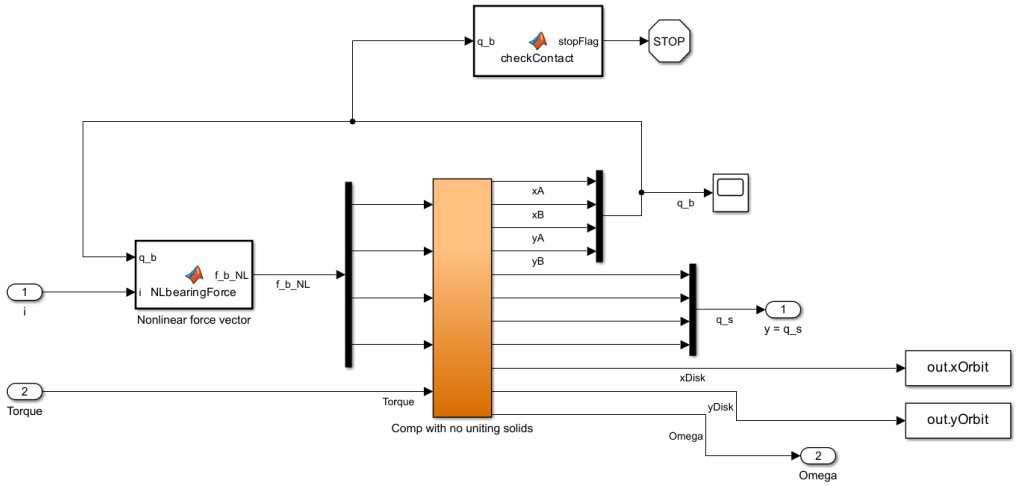


Figure 2.26. Implementation of the Adams block in Simulink.

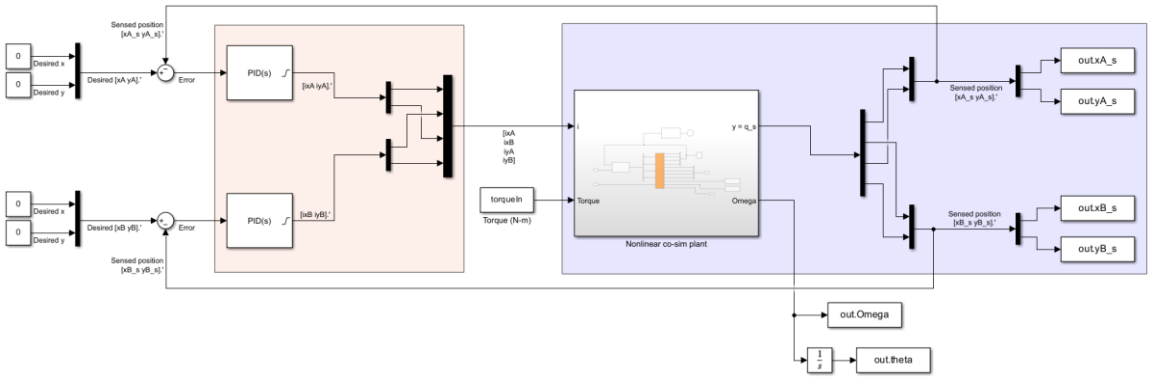


Figure 2.27. **Adams co-simulation model of the example rigid rotor**, with controller (red) and plant (purple).

2.5.4. Comparison of results

Both the pure Simulink and Adams co-simulation models were run with a simulation time of 80 seconds, which with the selected input torque yielded a speed range of

roughly 0 to 125 Hz. Figure 2.28 provides a Bode plot of the x-displacement at both bearings overlaid. The pure and co-simulation results are practically identical, and the peaks seen near 36 Hz and 96 Hz agree well with the predicted critical speeds from theoretical analysis. Additionally, the phase at the first resonance is the same at both sensor planes, indicating parallel shaft motion, and the phase is separated by 180° at the second resonance, indicating conical shaft motion or wobbling of the shaft, confirming Schweitzer and Maslen's predictions [1].

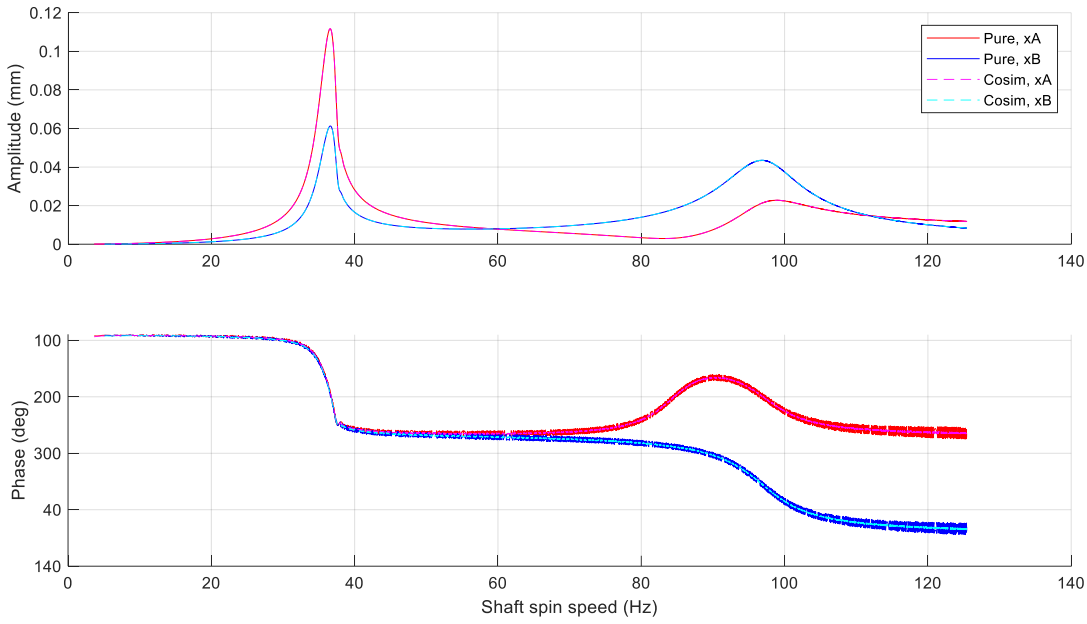


Figure 2.28. **Bode plot of pure Simulink and co-simulation unbalance response**, as measured at both sensor planes in the x-direction.

These results show that both the dynamics of Adams and AMB control in Simulink are accurate to theoretical predictions, validating co-simulation in the rigid body case.

Chapter 3

WHIRLING MODES OF A THIN WIRE HANGING ROTOR

This chapter studies a spinning disk hanging on a steel wire, which was an experimental setup shown in a DVD accompanying a rotor dynamics textbook [17]. *In Adams simulation, the disk's whirl shapes match those of the experiment, and their frequencies match theory, demonstrating accurate modeling of flexibly body dynamics and the gyroscopic effect.*

In the previous chapter, we demonstrated the validity of a rigid rotor co-simulation in Adams and Simulink. While this is useful, the rigid assumption is inappropriate when spin speeds approach or exceed the shaft's bending frequencies. In these situations, resonance becomes dominated not by the control behavior but by the bending of the shaft itself, changing natural frequencies and mode shapes drastically. Shaft bending also means that displacement at the AMBs may be different from that measured by the sensors due to their difference in axial position. These challenges and others make it necessary to know the bending behavior of the entire shaft to accurately model the controlled system. Thus, a proper AMB rotor simulation must model shaft flexibility, which the Adams co-simulation process can achieve.

In a DVD accompanying *Dynamics of Rotating Systems* by Giancarlo Genta [17], a disk hanging on a steel wire was used to visually demonstrate the gyroscopic effect in a flexible rotor and verify theoretical predictions of natural frequencies (Figure 3.1). Using a similar setup in Adams, along with FEA modeling in MSC Apex, we

demonstrate accurate modeling of flexible rotor dynamics and gyroscopic effect, further validating the applicability of the co-simulation process to AMB rotors.

Figure 3.1. Experimental thin wire rotor undergoing gyroscopic motion [17].

3.1. Modeling flexible rotors

The motion of a linear dynamic system can be represented by a sum of its *modes*, which in a mechanical system are the displacement shapes that would result in simple harmonic vibration. A system has as many modes as *degrees of freedom* (DOFs), the number of parameters required to completely specify its position, which is also the number of equations of motion (EOMs). In the previous chapter, the AMB-supported

rigid rotor had 4 EOMs, so it had only 4 DOFs and 4 modes, neglecting mirror symmetry. This system of equations was still compact enough to hand-derive.

A flexible body, on the other hand, can be displaced in infinitely many ways, so modeling its displacement requires infinitely many mode shapes, DOFs, and EOMs. This is unfeasible, but approximation methods can model flexible response with sufficient accuracy. *Finite element analysis (FEA)* is the most common of these, in which a flexible body is represented by many smaller elements of simple properties, yielding finite DOFs. For rotor shafts, *beam elements* are used most often, which are space curves that can accommodate shear forces and bending moments at the two ends (but not axial forces or torques). The displacement within the element is usually interpolated by a cubic polynomial with respect to position along the element's length. In our analysis, we assume that the steel wire acts as a set of *Euler-Bernoulli beam elements*, which do not model shear deformation; this assumption is usually appropriate when element length is much larger than the diameter. The alternative is *Timoshenko beam elements*, which do account for shear deformation but whose mathematics are more involved.

Since there are usually multiple DOFs per element, the equation set is still large, making analysis difficult. While many theoretical treatments exist, including for Euler-Bernoulli beam elements in [1] and Timoshenko beam elements in [18], implementation is complicated, time-consuming, and error-prone. Instead of creating our own FEA model, our goal in this work is to utilize general-purpose commercial software.

MSC Apex is one of these, developed by Hexagon AB, the company that created Adams. Flexible bodies can be created and turned into finite element models, which can be exported in a *modal neutral file* (MNF) that Adams can import for dynamic simulation. While other FEA programs can also export MNF files, Apex is of particular interest because both Apex and Adams are Hexagon-owned.[4] By using Apex, we can incorporate a flexible shaft into the rotor model in Adams. In this thin wire rotor case, there are no magnetic bearings and thus no need for co-simulation, so Adams alone will be used at the simulation stage.

3.2. Whirl modes of a single-disk flexible rotor

Whirl is the lateral oscillation of a spinning rotor, usually excited by rotor imbalance. For a spinning rotor on isotropic bearings, mode shapes are circular whirl patterns, and for a rotor with a single disk on a massless flexible shaft, there are 4 such modes (Figure 3.2). For an overhung rotor, or for a hanging rotor like the one in this demonstration, they can be described as follows:

1. Rapid wobbling of the disk in the direction of shaft rotation—the *2nd forward whirling mode*
2. Tire-swing-like motion of moderate speed in the direction of shaft rotation—the *1st forward whirling mode*

[4] Hexagon has recently developed Adams Modeler, which combines Adams and Apex into a single application. However, some features (e.g. beam elements) are not yet available in Modeler, so we use the separate programs in this research, but Modeler may be a more efficient tool in the future.

3. Slow tilting of the disk opposite the direction of shaft rotation—the *1st backward whirling mode*
4. Buffing-wheel-like motion of moderate speed opposite the direction of shaft rotation—the *2nd backward whirling mode*.

1st and 2nd indicate the relative frequency of each mode, 2nd being higher. The video of the thin wire rotor shows these modes being manually excited by hand one at a time. We would like to show that the whirling mode shapes of a simulated rotor in Adams match those of Genta's apparatus and that the whirling mode frequencies match predictions from rotor dynamics theory. If these are true, then we will have further evidence that Adams's flexible body and gyroscopic dynamics are appropriate for flexible rotor dynamics simulations.

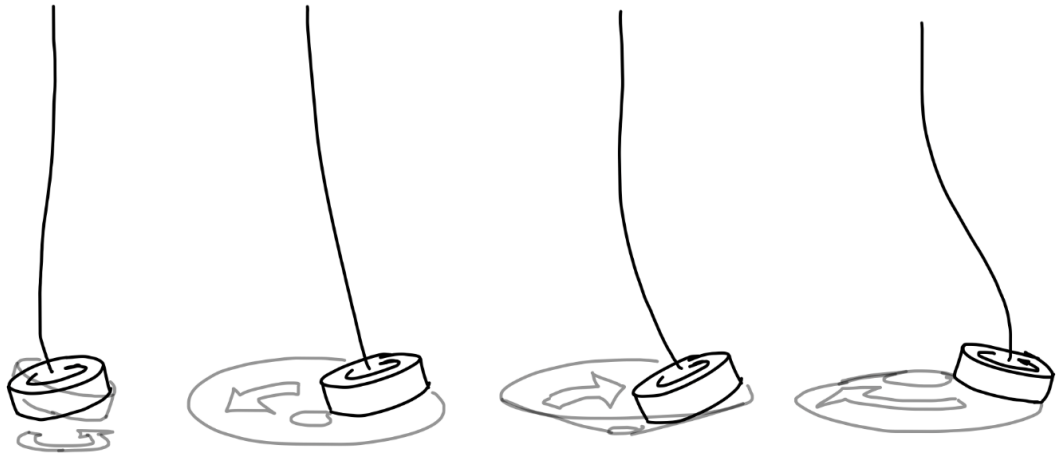


Figure 3.2. **Whirling modes of an overhung or hanging rotor**, 2nd and 1st forward and 1st and 2nd backward.

3.3. Theoretical analysis

While understanding the full motion of the wire requires FEA, if we are concerned only about the disk motion and assume a massless wire, the problem reduces to 4 DOFs, and if we assume linearity, we can return to Kramer's equations of motion [22]. Neglecting damping but retaining the stiffness terms now, we have

$$
\begin{aligned}
m\ddot{x} \qquad & + k_{xx}x + k_{x\beta}\beta = f_x \\
m\ddot{y} \qquad & + k_{yy}y + k_{y\alpha}\alpha = f_y \\
I_x\ddot{\alpha} + I_z\Omega\dot{\beta} & + k_{\alpha y}y + k_{\alpha\alpha}\alpha = T_x \\
I_y\ddot{\beta} - I_z\Omega\dot{\alpha} & + k_{\beta x}x + k_{\beta\beta}\beta = T_y.
\end{aligned}
$$

Here, α and β represent the angles of the disk, which are specified by Euler angles according to Appendix B, and each stiffness coefficient k_{ij} represents the force or torque in *i* that contributes to a unit displacement in *j*. Unlike in the previous analysis, the stiffness terms have been retained to model an approximately linear relationship between disk displacement and restoring force. However, the process of developing a Campbell diagram from the state representation will be the same as in Section 2.5.1. Thus, we only need to determine the values of the stiffness coefficients k_{ij}. Now, if the system is static, then the derivative terms vanish, and we have

$$
\begin{aligned}
k_{xx}x + k_{x\beta}\beta &= f_x \\
k_{yy}y + k_{y\alpha}\alpha &= f_y \\
k_{\alpha y}y + k_{\alpha\alpha}\alpha &= T_x \\
k_{\beta x}x + k_{\beta\beta}\beta &= T_y.
\end{aligned}
$$

Without the gyroscopic term, the first and last equations are uncoupled from the middle two. Let us examine the middle two first, which represent motion in the yz-plane. They can be written

$$\begin{bmatrix} f_y \\ T_x \end{bmatrix} = \begin{bmatrix} k_{yy} & k_{y\alpha} \\ k_{\alpha y} & k_{\alpha\alpha} \end{bmatrix} \begin{bmatrix} y \\ \alpha \end{bmatrix}.$$

Now, the matrix of k values is a 2x2 stiffness matrix, but we can define a *flexibility matrix* that is the inverse of the stiffness matrix (assuming invertibility), allowing us to rewrite the above as the following:

$$\begin{bmatrix} y \\ \alpha \end{bmatrix} = \begin{bmatrix} c_{yy} & c_{y\alpha} \\ c_{\alpha y} & c_{\alpha\alpha} \end{bmatrix} \begin{bmatrix} f_y \\ T_x \end{bmatrix}.$$

With this equation, if force and torque on the disk are known at static steady state, then its displacements are known. In particular, when $T_x = 0$, we see that $c_{yy} = y/f_y$ and $c_{\alpha y} = \alpha/f_y$, and when $f_y = 0$, we see that $c_{y\alpha} = y/T_x$ and $c_{\alpha\alpha} = \alpha/T_x$. Therefore, if we apply an f_y with no torque and measure y and α, and then do the same with T_x and no force, we can easily calculate $c_{yy}, c_{y\alpha}, c_{\alpha y}$, and $c_{\alpha\alpha}$ and thus the values of $k_{yy}, k_{y\alpha}, k_{\alpha y}$, and $k_{\alpha\alpha}$. This process is simplified by Maxwell's reciprocal theorem, which states that $k_{y\alpha} = k_{\alpha y}$ and equivalently $c_{y\alpha} = c_{\alpha y}$.

These k are all in the yz-plane. In the perpendicular xz-plane, we expect the magnitudes of $k_{xx}, k_{x\beta}, k_{\beta x}$, and $k_{\beta\beta}$ to be the same, but they may have different signs. In Figure 3.3, we see that while positive forces and torques in y and α cause positive displacements, a positive force in x causes a positive displacement in x but a *negative*

displacement in β, and a positive torque in α causes a positive displacement in β but a *negative* displacement in *x*. Therefore,

$$k_{xx} = k_{yy}, \qquad k_{\beta x} = -k_{\alpha y}, \qquad k_{x\beta} = -k_{y\alpha}, \qquad k_{\beta\beta} = k_{\alpha\alpha},$$

and the EOMs become

$$\begin{aligned} m\ddot{x} \qquad\quad &+ k_{yy}x - k_{y\alpha}\beta = f_x \\ m\ddot{y} \qquad\quad &+ k_{yy}y + k_{y\alpha}\alpha = f_y \\ I_x\ddot{\alpha} + I_z\Omega\dot{\beta} &+ k_{\alpha y}y + k_{\alpha\alpha}\alpha = T_x \\ I_y\ddot{\beta} - I_z\Omega\dot{\alpha} &- k_{\alpha y}x + k_{\alpha\alpha}\beta = T_y. \end{aligned}$$

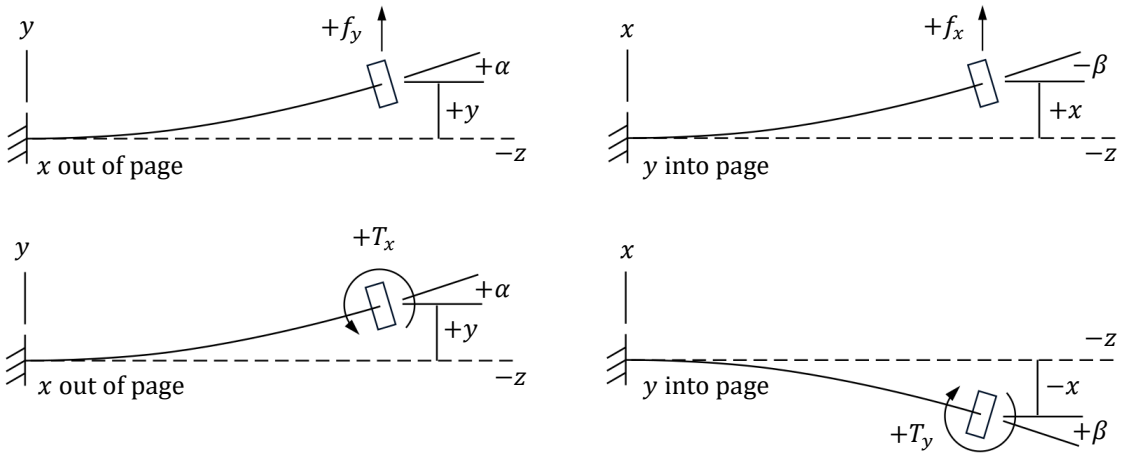


Figure 3.3. Signs of displacements due to forces and torques in yz- and xz-planes.

Following steps similar to Section 2.3.2, we can write the EOMs in the matrix form

$$\mathbf{M\ddot{q}} + \mathbf{G\dot{q}} + \mathbf{Kq} = \mathbf{f},$$

where $\mathbf{q} = [\beta \quad x \quad -\alpha \quad y]^{\mathrm{T}}$ are the coordinates of the disk, $\mathbf{f} = [T_y \quad f_x \quad -T_y \quad f_y]^{\mathrm{T}}$ are the external disturbance forces at the disk CG, **M** and **G** are as previously defined in Section 2.3.2, and

$$\mathbf{K} = \begin{bmatrix} k_{\alpha\alpha} & -k_{\alpha y} & 0 & 0 \\ -k_{y\alpha} & k_{yy} & 0 & 0 \\ 0 & 0 & k_{\alpha\alpha} & -k_{\alpha y} \\ 0 & 0 & -k_{y\alpha} & k_{yy} \end{bmatrix}.$$

The state matrix is then

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{4\times4} & \mathbf{I}_{4\times4} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{G} \end{bmatrix}.$$

Since the matrix **G** depends on Ω, we can again find the eigenvalues of **A** with respect to Ω and plot the rotor's Campbell diagram. Thus, if we only measure $k_{yy}, k_{y\alpha}, k_{\alpha y}$, and $k_{\alpha\alpha}$, we will have sufficient data to validate the Adams simulation.

3.4. Model setup in Apex and Adams

With theory established, we begin the thin wire rotor example. Genta [17] did not list all parameters of rotor example in the DVD demonstration, so we will create a new similar one. An aluminum disk is attached to the end of a thin, vertical steel wire (Figure 3.4, Table 3.1). The mass properties of the wire are unused except for density. The disk is treated in theory as a lumped mass with rotational inertia, which avoids gravity-induced moments on the disk and simplifies the analysis.

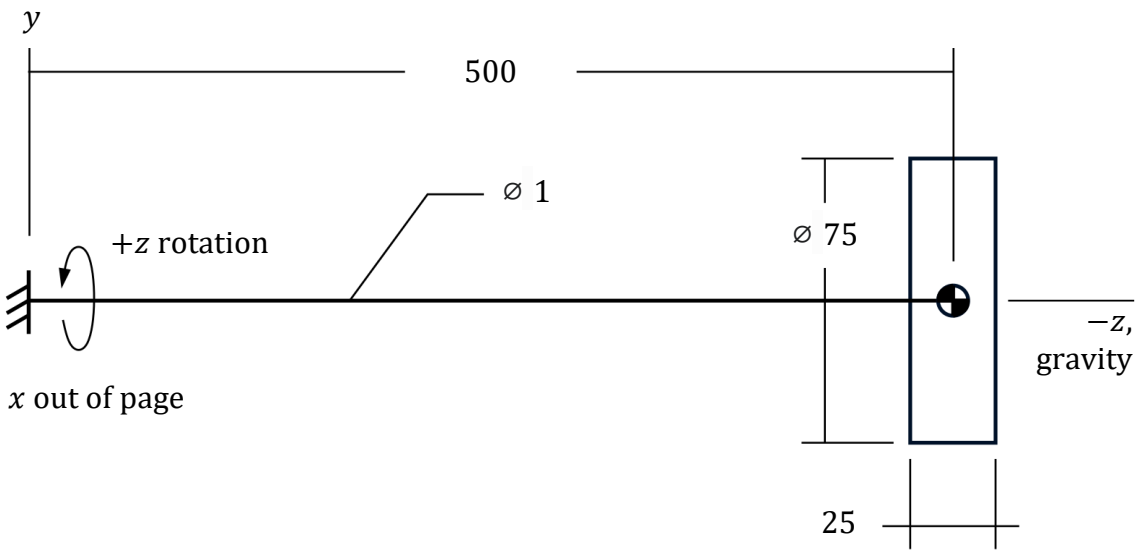


Figure 3.4. Thin wire rotor dimensions (mm), with gravity pointing right.

Table 3.1. Thin wire rotor parameters.

Quantity	Disk	Wire	Units
Length/thickness	25	500	mm
Diameter	7.5	1	mm
Density	2740	7801	kg/m^3
Mass	0.3026	—	kg
Polar MOI	2.1278×10^{-4}	—	$kg\ m^2$
Diametral MOI	1.2215×10^{-4}	—	$kg\ m^2$

To run this simulation, we first create the wire in Apex, run FEA analysis to produce its MNF file, use the MNF to create the rotor in Adams, and then simulate the rotor to determine its whirl frequencies. After creating the MNF, Apex is no longer needed, and all analysis can be performed in Adams alone.

Since the wire is very thin, we use beam elements to model the system and assume that axial displacements are negligible. For this test, we use 20 elements each of length 25 mm. While Apex models shear deformation in beam elements by default, this high length-to-width ratio means that they should act like Euler-Bernoulli elements. The full procedure for creating the wire and MNF using Apex is provided in Appendix C. The result is Figure 3.5.

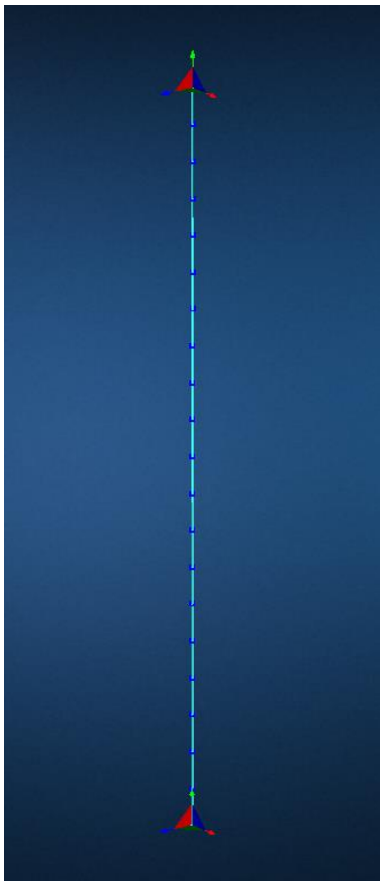


Figure 3.5. Thin wire created in Apex.

In Adams, importing an MNF replaces an existing body with an FEA model created in Apex. We first import the wire this way, and then we create the disk and fix its CG at the end of the wire. While this does imply interference between the wire

and disk, this choice aligns with our lumped mass assumption. A rotational joint is attached at the top of the wire to act as the motor axis, and measures are applied so that simulation data can eventually be plotted. The procedure for constructing the rotor in Adams is provided in Appendix C, and the result is Figure 3.6. This completes the thin wire rotor model, which we can now simulate.

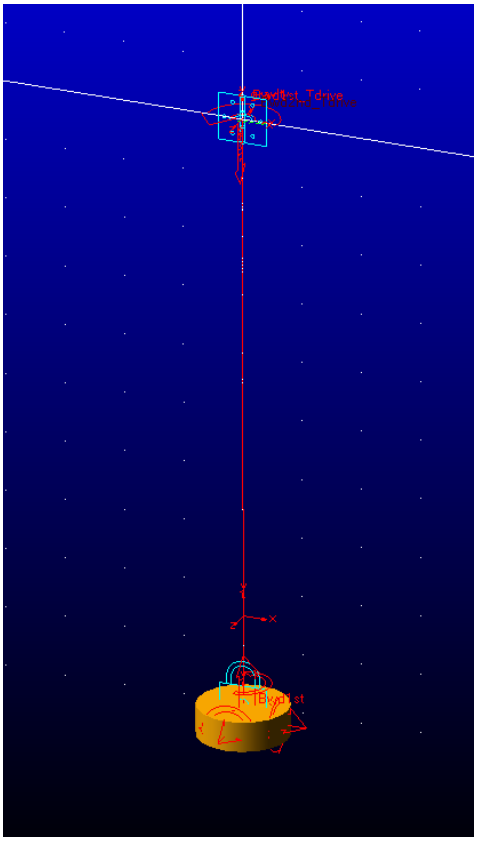


Figure 3.6. Thin wire rotor created in Adams.

3.5. Simulation and results

Before whirl simulation, it is necessary to determine the elements of the stiffness matrix by applying forces and torques and measuring displacement at static equilibrium. First, a constant test force f_y is applied horizontally at the disk center of

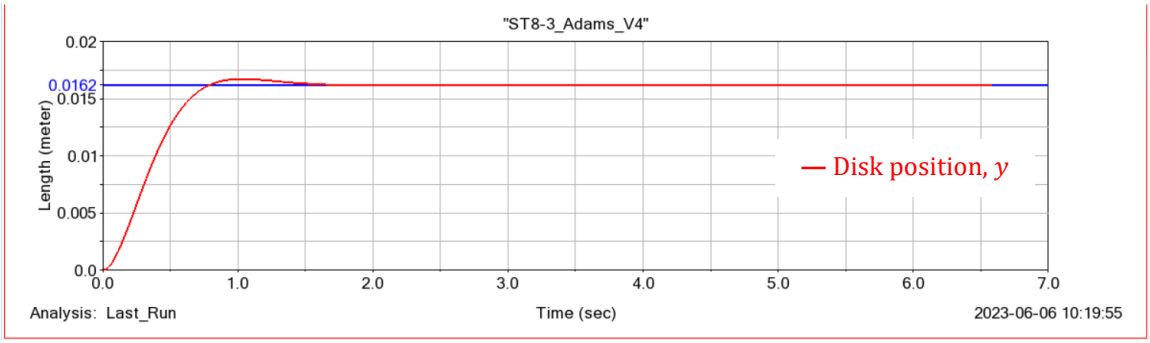
mass, and to prevent indefinite oscillation, a force proportional to the disk's velocity is subtracted from f_y so that the disk position will settle quickly. In Adams, the function expression is 0.1 – 2*VY(DISK.cm), where VY measures linear velocity. The resultant displacements are shown in Figure 3.7. Since $f_y = 0.1$ N gives $y = 0.0162$ m and $\alpha = 0.0452$ rad, we know that $c_{yy} = 0.162$ and $c_{\alpha y} = 0.452$. By Maxwell's reciprocal theorem, $c_{y\alpha} = 0.452$ also.

Next, we apply a torque T_x on the disk given by 0.01 – 0.05*WX(DISK.cm), where WX measures angular velocity; we also retain the force term –VY(DISK.cm) to damp out translation. The angular displacement due to this is shown in Figure 3.8, and since the value is 0.161 rad and the torque is 0.01 N, we know that $c_{\alpha\alpha} = 16.1$. Inverting the flexibility matrix then yields

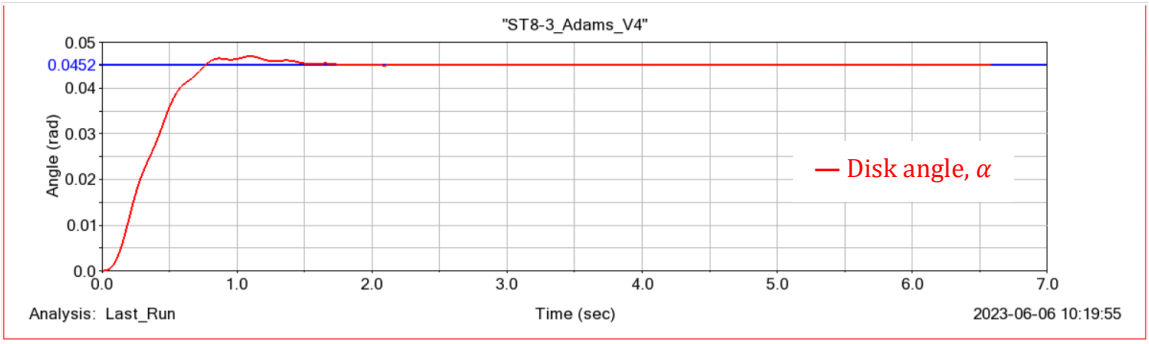
$$\begin{bmatrix} k_{yy} & k_{y\alpha} \\ k_{\alpha y} & k_{\alpha\alpha} \end{bmatrix} = \begin{bmatrix} 6.698 & -0.188 \\ -0.188 & 0.0674 \end{bmatrix},$$

which are the stiffness values needed for the theoretical Campbell diagram.

Each simulation test is run with a desired initial angular velocity and a set of impulse forces and torques designed to excite a particular mode, and then the resulting whirl frequency is measured and plotted on the theoretical Campbell diagram. During initial testing of simulation accuracy, results appeared to converge when the simulation time step was 0.001 s or smaller, so a 0.001 s time step was used for all simulations.



(a)



(b)

Figure 3.7. **Lateral** (a) **and angular** (b) **displacements resulting from a 0.1 N constant force.**

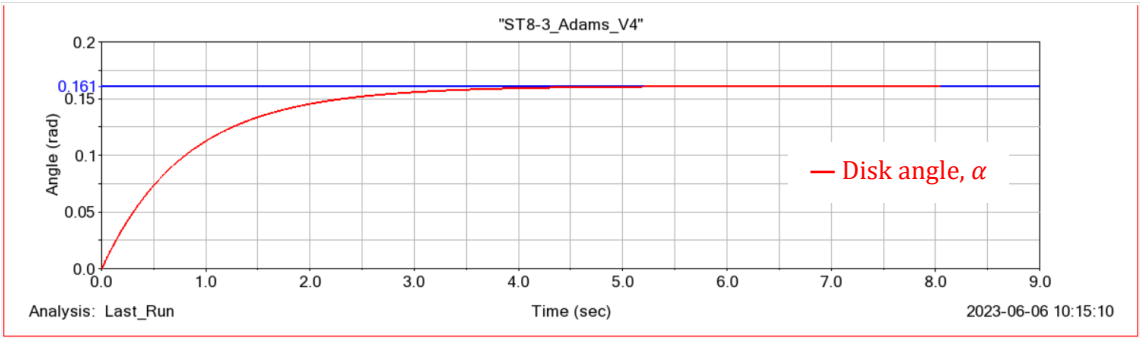


Figure 3.8. Angular displacement resulting from a 0.01 N-m constant torque.

To walk through an example of this process, consider the test performed for the 2nd backward mode (buffer wheel motion) at an initial velocity of 1000 RPM. In the video, the demonstrator grabbed the spinning wire by hand and gently moved it

in a circular motion until the mode could be clearly seen. We imitate this by applying a force at a shaft node close to but above the disk, having an impulse along one horizontal axis followed by one along the other. The timing and magnitude of the impulses are refined by trial-and-error, informed in part by the predicted whirl frequency and post-simulation displacement measurements. Eventually, a suitable set of force inputs in Adams is found to be

Force in y-direction: STEP(time, 0.1, 0, 0.5, 0.1) – STEP(time, 0.5, 0, 0.9, 0.1)

Force in x-direction: STEP(time, 0.5, 0, 0.9, 0.1) – STEP(time, 0.9, 0, 1.3, 0.1),

where STEP(time, t0, h0, t1, h1) is an approximated step function from h0 (force in N) at t0 (time in s) to h1 at time t1. Inputs with respect to time are given in Figure 3.9. With this input, the mode shape was stable and matched the whirl type; consecutive snapshots of this rotor simulation over one whirl cycle are shown in Figure 3.10.

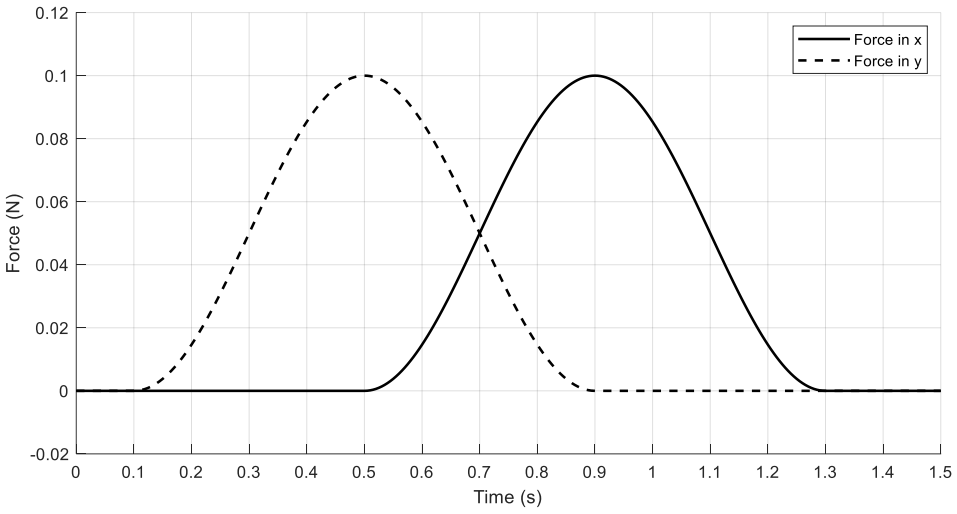


Figure 3.9. Impulses on the wire to excite 2nd backward mode at 1000 RPM.

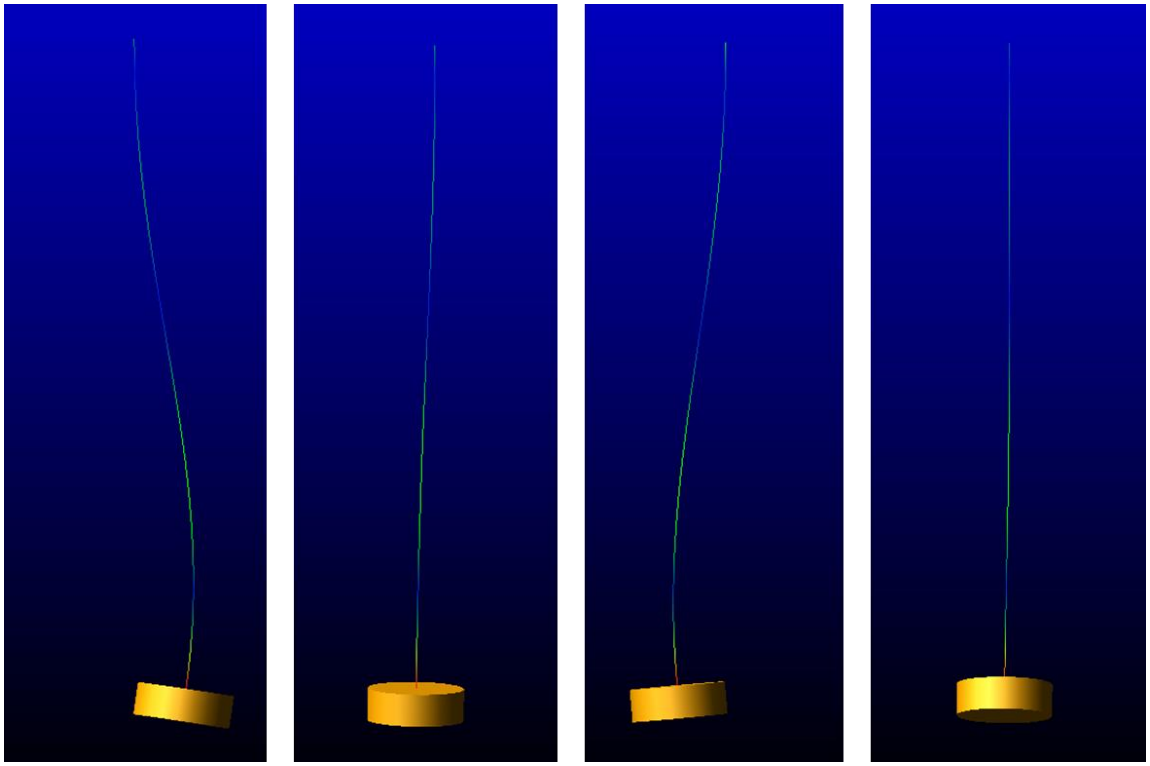


Figure 3.10. 2nd backward whirl motion over one cycle, four snapshots.

Having found the mode, we plot the displacement of the disk with respect to time in the simulation post-processor and determine whirl frequency from the period of the harmonic motion (Figure 3.11). In this example, since the most stable and visible displacements are the lateral motions, we plot and measure from those; other whirling motions, such as the 1st backward mode, are clearer in angular displacement. The period of motion averaged over four cycles is 1.28 s, which corresponds to a frequency of 0.78 Hz, or 46.8 RPM. This data point can now be plotted on the theoretical Campbell diagram.

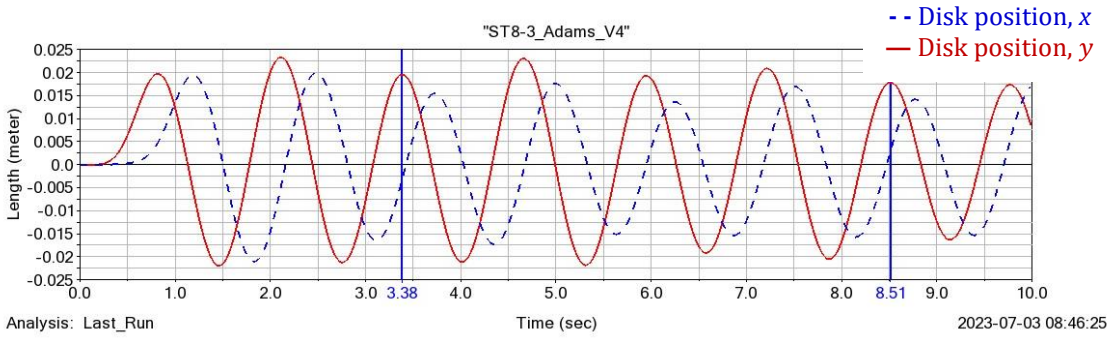


Figure 3.11. Lateral displacement under 2nd backward whirl at 1000 RPM.

This process is performed for several combinations of shaft speed and mode types, yielding Figure 3.12. The Campbell diagram does not distinguish whirl direction, so the curves that truly correspond with measured data have been highlighted. Other methods involving complex representations of disk position, such as that given in [17], yield only the highlighted curves.

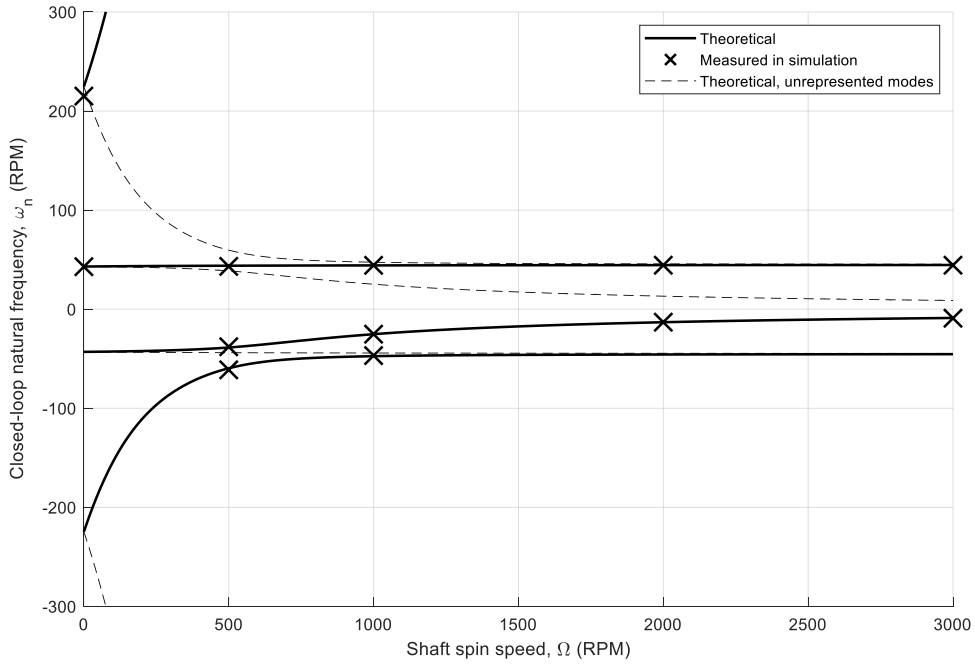


Figure 3.12. Campbell diagram with measured whirl frequencies from simulation.

We see a close match between the theoretical whirl frequencies and those measured in Adams. The largest difference is the overestimation of the 2nd forward mode at zero shaft spin, which can be explained by the fact that our theory neglects wire inertia but Adams does not. The backward modes (those below the horizontal axis) have no measurements at some locations because those modes have very similar behavior to the corresponding forward mode, and the 2nd forward mode is only measured at no spin because that vibration is otherwise almost undetectable in practical applications [17].

These results show that flexible body dynamics and the gyroscopic effect are modeled as expected in Adams for rotating machinery applications.[5] This analysis and the previous chapter enable us to simulate more complex systems with confidence in the accuracy of results.

[5] Some may wonder whether the thin wire rotor test is sufficient to draw this conclusion for shafts of higher material stiffness, or *EI* value. From Genta's analysis [17], the 2x2 matrix K of this rotor is the sum of wire elasticity K_{shaft} (which depends on *EI*) and stiffening due to gravity K_{grav} (which does not). In this example, K_{shaft} is relatively small, but for larger diameters, K_{shaft} will have a larger contribution, and for horizontal shafts, there is simply no K_{grav} term and $K = K_{shaft}$. However, K still has the same form in all cases, so we expect similar whirl behavior, even if frequencies are different and shapes harder to detect.

Chapter 4

VISUALIZATIONS OF A FLEXIBLE ANISOTROPIC AMB ROTOR

A flexible anisotropic rotor on AMBs is run, a first for Adams co-simulation. Visualizations of the rotor's unbalance response, programmatically generated during the simulation, include a Bode diagram, 2D and 3D orbit plots, and a full spectrum cascade plot, none of which have been shown in a co-simulated model.

Up to this point, our concern has been to validate the co-simulation process without access to experimental data. We did so by deriving theoretical models and simulations and showing that they matched, demonstrating proper modeling of rigid and flexible body dynamics, nonlinear magnetic forces, control algorithm implementation, and the gyroscopic effect. Now that these have been shown, we can free ourselves to explore the capabilities of Adams co-simulation through examples that are beyond our scope to numerically validate, namely a flexible anisotropic rotor on AMBs.

In Section 4.1, we investigate the theory of *anisotropic* rotors, i.e. those whose parameters vary along different radial directions and whose whirl behavior is different from that of isotropic rotors. In Section 4.2, we describe the rotor setup used as the example for this chapter, which has AMB control gains that are different in the x- and y-directions, forming an anisotropic system. In Section 4.3, the rotor is simulated, showing elliptical *orbits* (whirl shapes) both *forward* and *backward* with

respect to shaft spin direction. Various 2D and 3D plots are programmatically generated for this example, including shaft orbit plots and *full spectrum plots*.

This chapter presents several results demonstrated nowhere in literature among articles using Adams co-simulation to model magnetic bearing rotors, to the best of the author's research. In particular, no literature shows:

1. *The use of Apex to model a flexible shaft*.
2. *Rotor response at more than a single shaft speed at a time.* This work demonstrates response over a continuous range of speeds.
3. *Frequency response*. A ramp test enables us to generate the rotor's Bode plot.
4. *Anisotropic rotor behavior.* We show the response of a rotor whose parameters differ by radial direction, yielding both forward and backward whirl.
5. *Shaft orbits.* We use MATLAB to plot orbit shapes in both 2D and 3D.
6. *Full spectrum plots.* No other article shows whirl decomposition into its forward and backward frequencies.
7. *Cascade plots.* We show a 3D plot of whirl spectra with respect to shaft speed.

4.1. Anisotropic rotor theory

Until now, we have dealt entirely with *isotropic* rotors, which have radial symmetry. When parameters along some radial axes differ, such as when the support structure has different stiffness vertically and horizontally, the rotor is *anisotropic*. In this case, new whirl patterns and natural frequencies emerge, calling for new analysis and visualization tools.

4.1.1. Orbit concepts

The shape of a shaft's whirl path is called its *orbit*. An isotropic rotor always has a circular orbit, but an anisotropic single-disk rotor can have circular, elliptical, or straight-line orbits. In addition, anisotropic rotor whirl can be in the direction of shaft spin or opposite the direction of shaft spin, respectively known as *forward* or *backward whirl* (Figure 4.1). Orbit plots include a gap and a dot indicating whirl direction. The direction of whirl is such that the rotor hits the gap immediately before the dot; this is known as the "blank/bright" sequence, using nomenclature from Bently et al. [23]. The dot corresponds with zero angular displacement of the shaft. If whirl is *synchronous*, i.e. at the same rate as the shaft speed, the dot will fall in the same spot on the orbit when multiple cycles are plotted; if whirl is *nonsynchronous*, then the dots will appear in multiple locations around the orbit (Figure 4.2). Orbit plots will be useful in visualizing the co-simulated anisotropic rotor response at various shaft spin speeds.

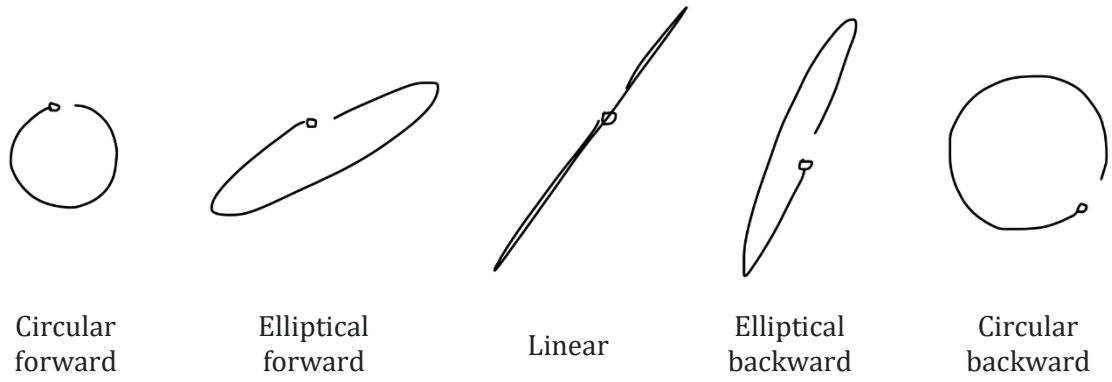
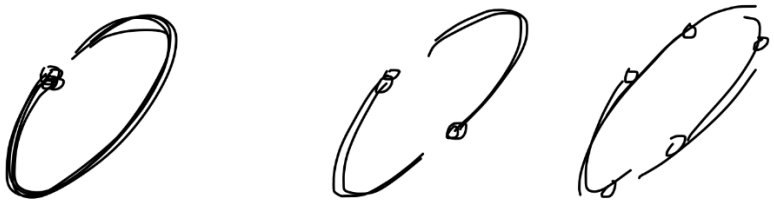


Figure 4.1. Orbit shapes for a shaft rotating CCW.



Synchronous Nonsynchronous

Figure 4.2. Synchronous and nonsynchronous orbits.

A perfectly elliptical orbit can be represented by parametric functions $x(t)$ and $y(t)$, both harmonic motions. Rotor data exists in this form if separate sensors measure x- and y-displacements, a common industry practice. Consider the displacement curves in Figure 4.3, and let us assume that the shaft rotates counterclockwise (CCW) when looking from the positive z-axis to the origin. In example (a), the rotor reaches a peak in x close to and before the next peak in y, so the shaft must be revolving CCW about the origin, which is forward whirl. In example (b), y leads x instead, so whirl is CW, which is backward whirl. In example (c), maxima coincide, so whirl is linear; this also occurs when maxima coincide with minima. In this way, the phase difference between x- and y-displacement dictates whirl direction.

As an alternate method of determining whirl direction, it can be shown that any elliptical whirl can be decomposed into a sum of two circular whirl motions, one forward and one backward. Figure 4.4 shows the whirl decompositions of the whirl patterns given in Figure 4.3. We see that the forward whirl component of example (a) is larger than the backward component, so the overall whirl is forward, and vice versa for example (b). The magnitudes are identical in the linear whirl in example (c).

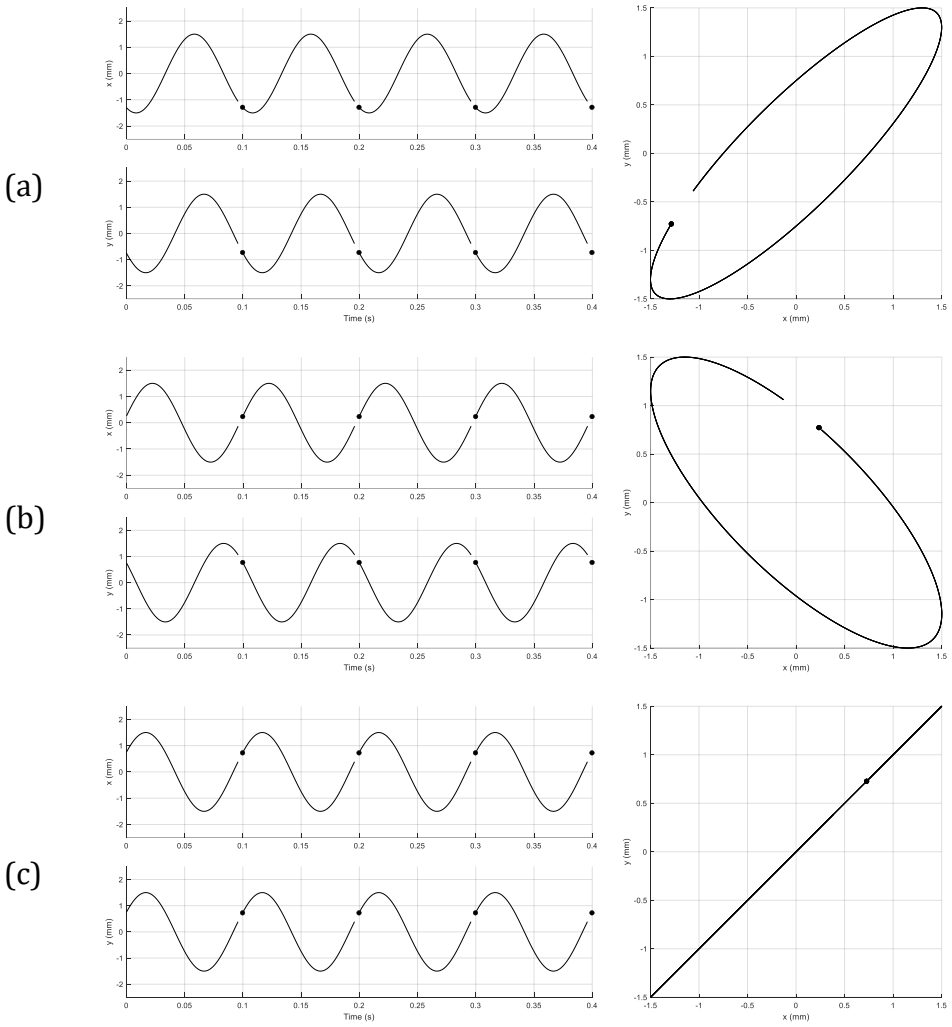


Figure 4.3. Whirl examples with x- and y-displacements (CCW shaft rotation).

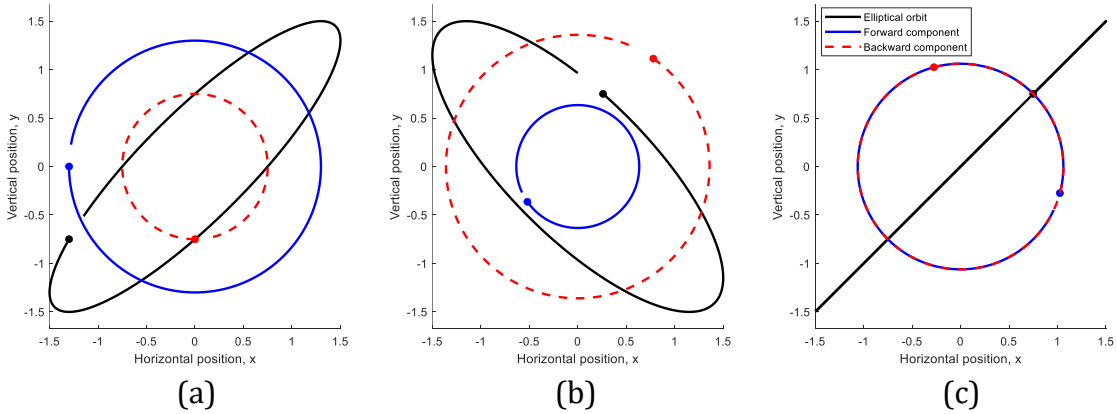


Figure 4.4. Whirl decompositions of the examples in Figure 4.3.

4.1.2. Orbit mathematics

Let us examine the mathematics behind the above results, adapted from Maurice Adams [24]. We can express the parametric expressions of position as the following:

$$x(t) = X\cos(\omega t + \phi_x)$$
$$y(t) = Y\sin\big(\omega t + \phi_y\big),$$

where X and Y are amplitudes, ω is the whirl frequency, and ϕ_x and ϕ_y are phases. Rewriting the trig functions as sums of complex exponentials, we have

$$x(t) = X\left[\frac{1}{2}\left(e^{j(\omega t+\phi_x)} + e^{-j(\omega t+\phi_x)}\right)\right]$$
$$y(t) = Y\left[-\frac{1}{2}j\left(e^{j(\omega t+\phi_x)} - e^{-j(\omega t+\phi_x)}\right)\right].$$

A complex position function $r(t) = x(t) + jy(t)$ can be formed from the above:

$$r(t) = X\left[\frac{1}{2}\left(e^{j(\omega t+\phi_x)} + e^{-j(\omega t+\phi_x)}\right)\right] + jY\left[-\frac{1}{2}j\left(e^{j(\omega t+\phi_x)} - e^{-j(\omega t+\phi_x)}\right)\right]$$
$$= X\left[\frac{1}{2}\left(e^{j(\omega t+\phi_x)} + e^{-j(\omega t+\phi_x)}\right)\right] + Y\left[\frac{1}{2}\left(e^{j(\omega t+\phi_x)} - e^{-j(\omega t+\phi_x)}\right)\right].$$

We can then perform algebra to collect two terms, one with $e^{j\omega t}$ and the other $e^{-j\omega t}$:

$$r(t) = \frac{1}{2}\left(Xe^{j\phi_x} + Ye^{j\phi_y}\right)e^{j\omega t} + \frac{1}{2}\left(Xe^{-j\phi_x} - Ye^{-j\phi_y}\right)e^{-j\omega t}.$$

This form already shows successful whirl decomposition; the first term is a complex constant times the forward circular whirl $e^{j\omega t}$, and the second is another constant times the backward circular whirl $e^{-j\omega t}$. However, we will continue with

manipulation to obtain clearer parameters. The exponentials can be split into cosines and sines according to Euler's formula and simplified by odd and even identities, giving us the following abbreviated expressions:

$$e^{j\phi_x} = \cos\phi_x + j\sin\phi_x = c_x + js_x$$

$$e^{j\phi_y} = \cos\phi_y + j\sin\phi_y = c_y + js_y$$

$$e^{-j\phi_x} = \cos\phi_x - j\sin\phi_x = c_x - js_x$$

$$e^{-j\phi_y} = \cos\phi_y - j\sin\phi_y = c_y - js_y.$$

Substituting and collecting real and imaginary parts,

$$r(t) = \frac{1}{2}\left[X(c_x + js_x) + Y\left(c_y + js_y\right)\right]e^{j\omega t} + \frac{1}{2}\left[X(c_x - js_x) - Y\left(c_y - js_y\right)\right]e^{-j\omega t},$$

$$= \frac{1}{2}\left[Xc_x + Yc_y + j(Xs_x + Ys_y)\right]e^{j\omega t} + \frac{1}{2}\left[Xc_x - Yc_y + j(-Xs_x + Ys_y)\right]e^{-j\omega t}.$$

Now, a complex number of the form $a + jb$ can be rewritten as $Re^{j\beta}$, where $R = \sqrt{a^2 + b^2}$ and $\beta = \tan^{-1}(b/a)$, and the inverse tangent must account for all four quadrants. If we do this to the coefficients of the two terms of $r(t)$, they become $Re^{j\beta}e^{\pm j\omega t} = Re^{j(\pm\omega t + \beta)}$, which is circular motion of radius R, frequency $\pm\omega$, and phase β. Performing this here, we see that the whirl decomposition in terms of variables from the parametric form is

$$r(t) = R_f e^{j(\omega t + \beta_f)} + R_b e^{j(-\omega t + \beta_b)},$$

where constant variables are given by

$$R_f = \frac{1}{2}\sqrt{\left(Xc_x + Yc_y\right)^2 + \left(Xs_x + Ys_y\right)^2} \qquad \beta_f = \tan^{-1}\left(\frac{Xs_x + Ys_y}{Xc_x + Yc_y}\right)$$

$$R_b = \frac{1}{2}\sqrt{\left(Xc_x - Yc_y\right)^2 + \left(-Xs_x + Ys_y\right)^2} \qquad \beta_b = \tan^{-1}\left(\frac{-Xs_x + Ys_y}{Xc_x - Yc_y}\right)$$

and $c_{x/y} = \cos\phi_{x/y}$ and $s_{x/y} = \sin\phi_{x/y}$. These formulas generated the whirl decompositions in Figure 4.4.

To connect to the previous conceptual analysis, we saw that differences in phase and differences in whirl component magnitude both dictate whirl direction. These two quantities should therefore be mathematically related. Consider the latter, $R_f - R_b$. We saw from Figure 4.4 that if this quantity is positive then whirl is forward, if negative then backward, and if zero then linear. However, since radius is always positive, this statement is also true of $R_f^2 - R_b^2$, a more mathematically convenient expression. Let us compute it and simplify:

$$\begin{aligned}
R_f^2 - R_b^2 &= \left(\frac{1}{2}\sqrt{\left(Xc_x + Yc_y\right)^2 + \left(Xs_x + Ys_y\right)^2}\right)^2 \\
&\qquad - \left(\frac{1}{2}\sqrt{\left(Xc_x - Yc_y\right)^2 + \left(-Xs_x + Ys_y\right)^2}\right)^2 \\
&= \frac{1}{4}\left[\left(Xc_x + Yc_y\right)^2 + \left(Xs_x + Ys_y\right)^2 - \left(Xc_x - Yc_y\right)^2 - \left(-Xs_x + Ys_y\right)^2\right] \\
&= \frac{1}{4}\left(\left[\left(Xc_x + Yc_y\right)^2 - \left(Xc_x - Yc_y\right)^2\right] + \left[\left(Xs_x + Ys_y\right)^2 - \left(-Xs_x + Ys_y\right)^2\right]\right) \\
&= \frac{1}{4}\left(\left[\left(Xc_x + Yc_y + Xc_x - Yc_y\right)\left(Xc_x + Yc_y - Xc_x + Yc_y\right)\right]\right. \\
&\qquad \left. + \left[\left(Xs_x + Ys_y - Xs_x + Ys_y\right)\left(Xs_x + Ys_y + Xs_x - Ys_y\right)\right]\right)
\end{aligned}$$

$$= \frac{1}{4}\left[(2Xc_x)\left(2Yc_y\right) + \left(2Ys_y\right)(2Xs_x)\right]$$

$$= XY\left(\cos\phi_x \cos\phi_y - \sin\phi_x \sin\phi_y\right)$$

$$= XY\cos(\phi_x - \phi_y).$$

We see then that $R_f^2 - R_b^2 = XY\cos\left(\phi_x - \phi_y\right)$. Since amplitudes X and Y are always positive, $R_f^2 - R_b^2$ has the same sign as $\cos\left(\phi_x - \phi_y\right)$, whose sign is in turn dictated by $\phi_x - \phi_y$. Thus, the magnitudes of the whirl components and the x- and y-phases are in fact mathematically dependent. In the latter case, whirl is forward when ϕ_x and ϕ_y are less than 90° apart, backward when they are more than 90° apart, and linear when they are exactly 90° apart. We will see these relationships appear in the simulated anisotropic rotor response.

4.1.3. Dependence of orbit shape on shaft speed

The type of orbit experienced by an unbalanced anisotropic rotor depends on shaft speed. Recall that the type of system we are analyzing will experience an amplitude peak at its natural frequency, coinciding with a 180° shift in phase. In this case, there will be such a peak in both the x- and y-directions, but since the rotor is anisotropic, the peaks will not coincide at the same frequency. In the example Bode plot in Figure 4.5, this causes the phase between the two directions to separate, eventually surpassing a 90° difference after the first peak, remaining in backward whirl for a short region, and then returning to forward whirl before the second peak. In other cases however, if damping is high or anisotropy is not strong, the phases might not separate past 90°, and no backward whirl will be observed.

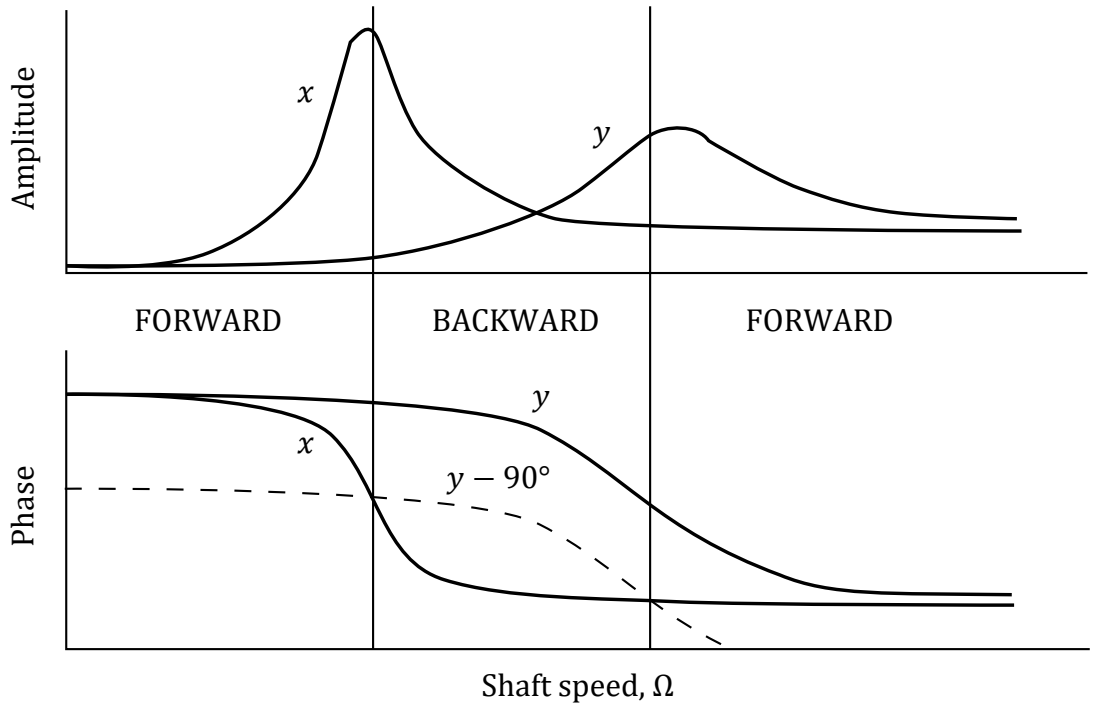


Figure 4.5. Example Bode plot of an anisotropic rotor.

In a video by Genta [17] (from the same DVD referenced in Chapter 3), a practical rotor is run on anisotropic AMBs, showing this type of whirl (Figure 4.6). Genta presents a 3D diagram plotting orbit shape with respect to shaft speed, colored by whirl direction. From the bottom up, the rotor begins in forward whirl (red), reaches a peak in y, enters backward whirl (green), returns to forward whirl, and then reaches a peak in x. We expect to see similar behavior in our anisotropic rotor.

4.2. Rotor setup

The rotor in this chapter is based loosely off the dimensions of the RK4 rotor kit model by Bently Nevada. The rotor that we will use has a main shaft body, two journals for AMBs, and a disk between the journals, all steel (Figure 4.7). It is symmetric about its midpoint. Magnetic bearing parameters are provided in Table 4.1.

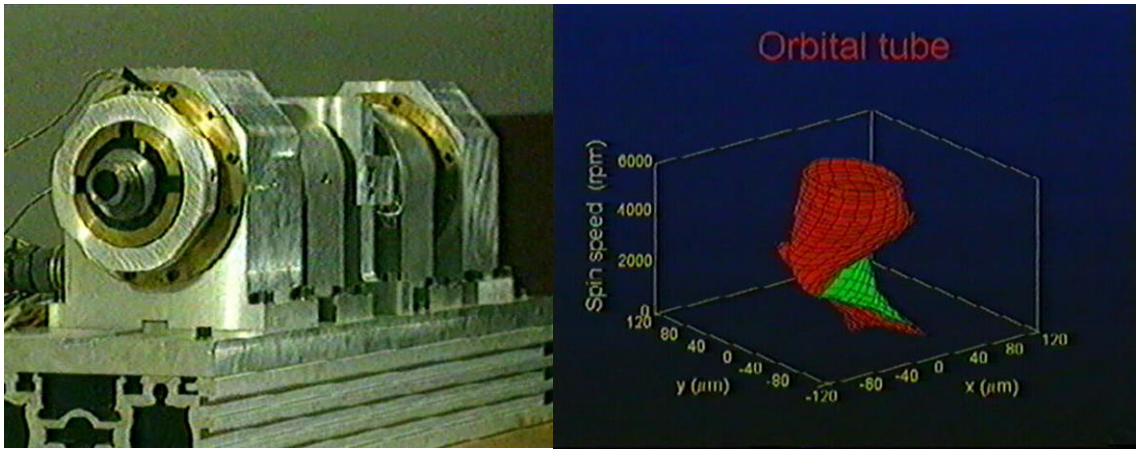


Figure 4.6. An experimental anisotropic AMB rotor and 3D orbit plot [17].

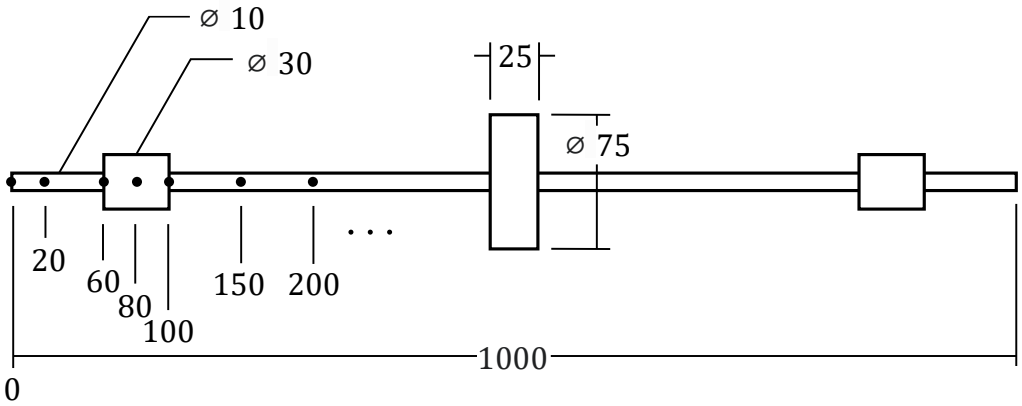


Figure 4.7. Flexible AMB rotor dimensions (mm), symmetric about midline.

Table 4.1. Flexible AMB rotor parameters.

Quantity	Value	Units
ρ	7801	kg/m^3
E	2.07×10^5	MPa
ν	0.29	—
m_u	1	g
r_u	0.03	m
s_0	0.0005	m
k	3.77×10^{-5}	H-m
i_0	0.35	A

In Apex, the finite element mesh contains nodes at 2 cm, 6 cm, 8 cm, and 10 cm from each end of the shaft, and then every 5 cm elsewhere. The MNF export requires an upper limit on modes calculated; this is set to 20. In Adams, the disk is a cylinder with a hole the diameter of the main shaft. The unbalance mass is a sphere fixed to the disk's interior, with mass specified by user input. AMB force inputs are applied at the center of the journals, i.e. at the nodes 8 cm from each end of the shaft.

For these tests, we will still use decentralized PD control as described in Chapter 2, but the Simulink model is modified as follows (Figure 4.8):

1. The rigid rotor control system was built with the assumption that the bearings would be symmetric, but we would like to create anisotropy by applying different PD control gains to the axes of a given bearing. To do this, we modify the Simulink model, implementing one PD controller for each axis instead of one for each bearing. Then we can implement the control parameters provided in Table 4.2.
2. The rigid rotor model had *non-collocated* sensors, i.e. those not at the same z-location as the bearings, but in this simulation, we will simplify our analysis by assuming that the sensors are *collocated* with the bearings. This is done by feeding the displacement at the bearings $\mathbf{q}_b$ into the sensed position $\mathbf{q}_s$ in the Simulink model.
3. Due to the nature of flexible rotor simulation, the co-simulation tests are computationally intensive and require long run times. To save time in simulation, we apply a high initial torque until the rotor reaches a speed

somewhat close to the expected first natural frequency and then cut it to a much smaller final value to ramp the rotor slowly through frequencies of interest. This is accomplished by a MATLAB function block that specifies the value of the torque based on the shaft speed, which is fed as an input. In this test, the torque was switched from 0.05 N-m to 0.001 N-m after passing 300 RPM.

Table 4.2. Control parameters for the anisotropic rotor.

Control parameter	x-direction	y-direction
$P_A = P_B$	7600	8100
$D_A = D_B$	0.05	0.3
$N_A = N_B$	1000	1000

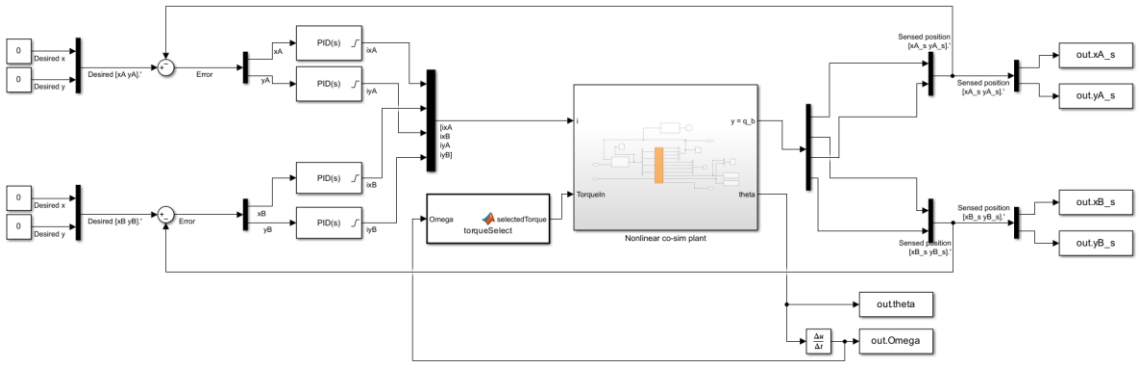


Figure 4.8. Simulink model for the anisotropic rotor.

4.3. Results and visualization tools

A benefit of the co-simulation process is that since the model is run in Simulink, results can be seamlessly generated and analyzed in MATLAB script code. This

section presents several useful diagrams which have been used to visualize the results of this anisotropic rotor test. These diagrams were generated programmatically when running the simulation, with user input only to specify desired parameters at which to plot.

4.3.1. Bode plot and rigid rotor Campbell diagram

We already saw the Bode plot in other sections, but we include one here to provide context for subsequent visualizations of shaft response (Figure 4.9). As expected, the peaks in the x- and y-directions are not at the same frequency. Examining the Campbell diagram for the equivalent rigid rotor (Figure 4.10), we see that the peak in x near 500 RPM is due to the control system resonance, since the damping is low in that direction and we are still well below the shaft's bending frequency of about 826 RPM, which was measured in Adams by attaching spherical (3D pin) joints at the AMB locations, providing an impulse at the disk, and measuring the frequency of the resulting oscillation. The peak in y is due to the shaft bending.

When plotting phase, to match the convention that $y(t)$ is defined in terms of sine instead of cosine, we need to shift the y-phase by 90 degrees so that it matches that of x for low shaft speeds. We see that in each direction, the phase falls by 180 degrees at its peak, as expected. However, since the peaks do not coincide, the phase in x shifts before the phase in y, causing them to separate beyond 90°. As discussed in Section 6, this indicates the presence of backward whirl between roughly 500 RPM and 750 RPM. We will see this more clearly in subsequent diagrams.

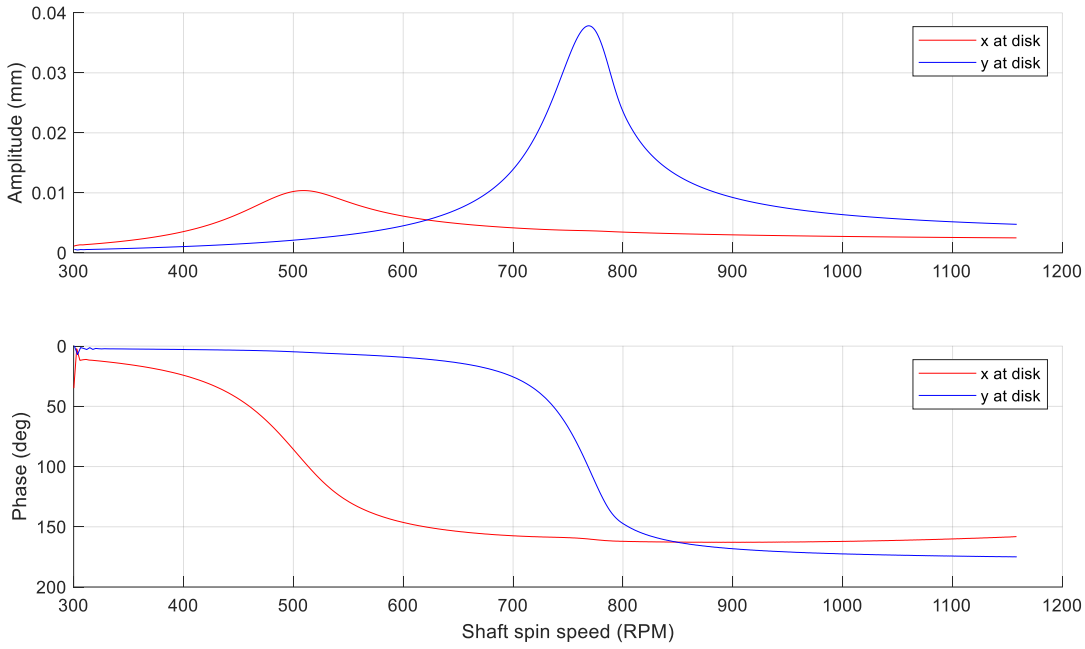


Figure 4.9. Bode plot of anisotropic rotor response.

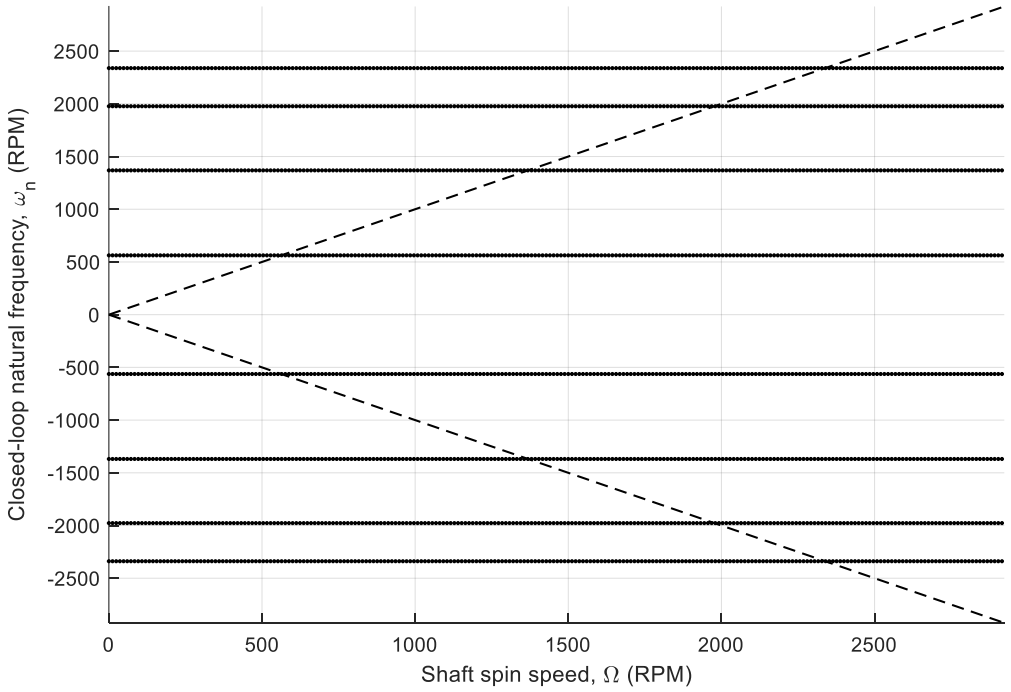


Figure 4.10. Campbell diagram of equivalent rigid anisotropic rotor.

4.3.2. 3D orbit plot

Our simulation gives us access to the disk's position at any point in time. If we plot this with respect to the current shaft speed, we can obtain a 3D orbit plot like that which we saw in Genta's results. The 3D orbit plot for this rotor is shown in Figure 4.11. To show the shape of the plot, we color it based on the angle of its slope. The peaks in x and y can be clearly seen, and by rotating the figure (not shown here), it is easy to identify where the orbit pinches to a straight line. In the view given, these locations can be faintly seen close to the peak in x and just before the peak in y.

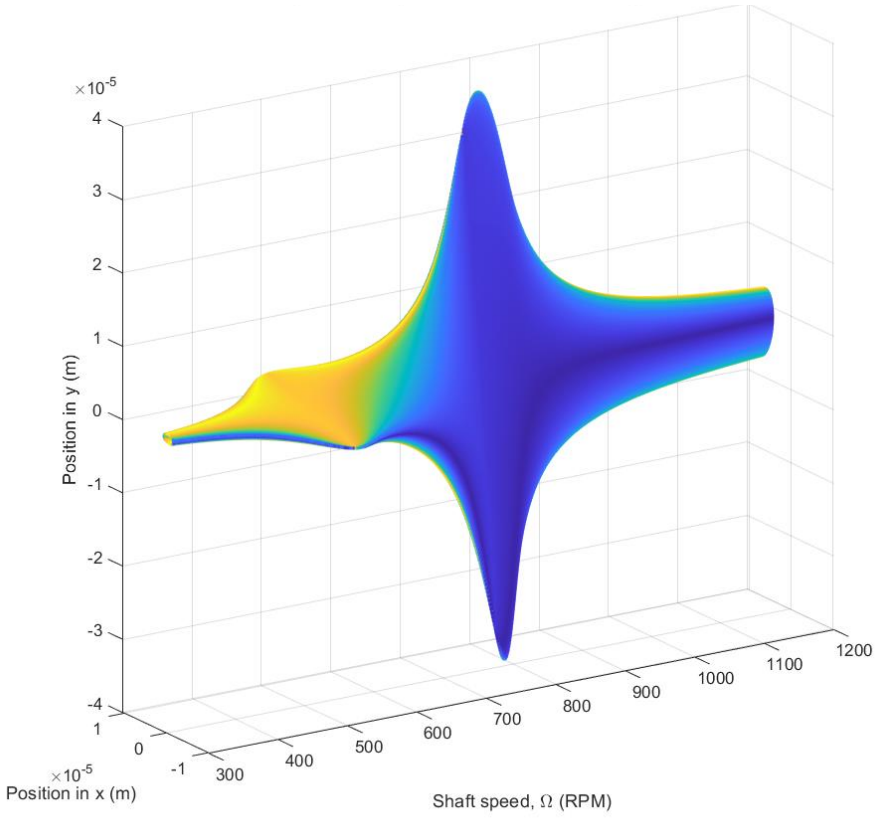


Figure 4.11. 3D orbit plot of anisotropic rotor response.

4.3.3. 2D orbit series

This 3D orbit plot, while useful, lacks some information. We would not have been able to identify whirl direction by this plot alone, nor could we identify the phase of the motion. 2D orbit plots provide this information visually.

Knowing that linear whirl occurs near 500 and 750 RPM, we can plot several successive orbits in a series to show how its shape and orientation change across the transitions. The two transitions are shown Figure 4.12 and Figure 4.13. These plots clearly show whirl direction using the blank/bright visualization, as well as the shape of the plot during the transition.

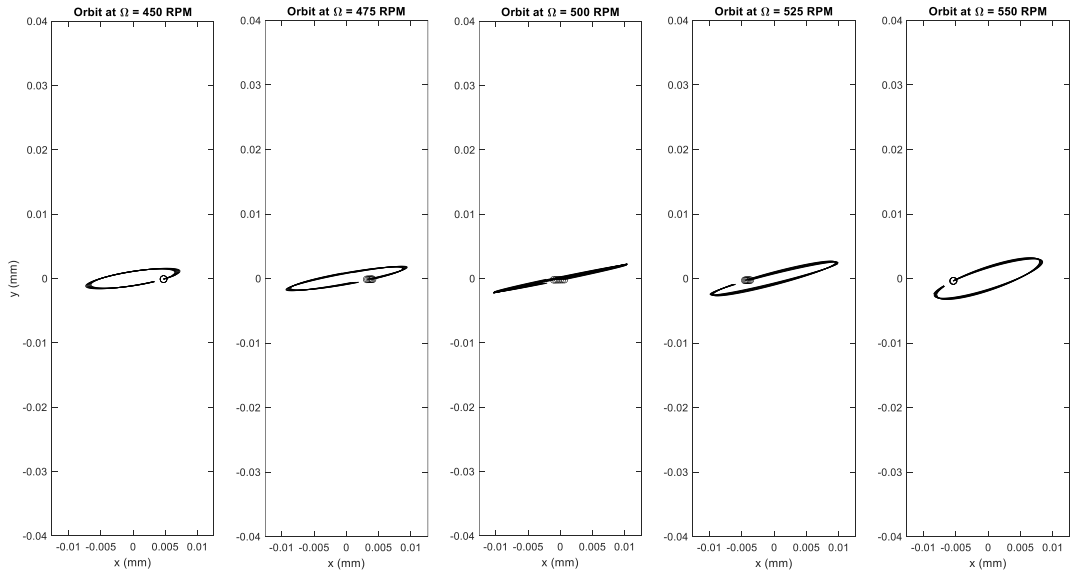


Figure 4.12. Orbits across forward-to-backward transition, 450 to 550 RPM.

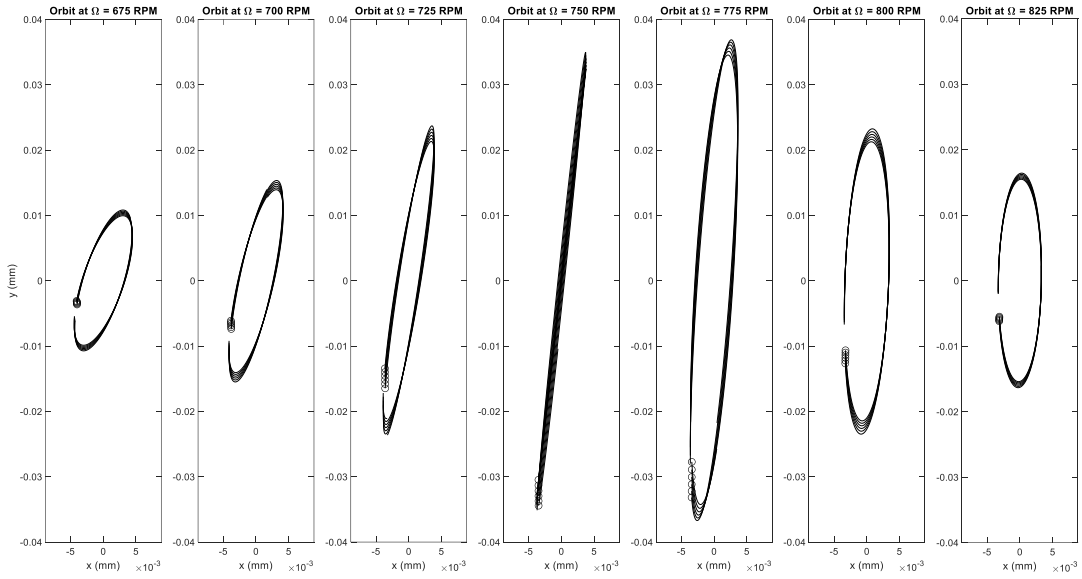


Figure 4.13. Orbits across backward-to-forward transition, 675 to 825 RPM.

4.3.4. Full spectrum cascade plot

Another visualization tool employed in industry is the *full spectrum plot*, which visualizes orbit in the frequency domain, displaying underlying circular whirl amplitudes with respect to their frequency, both positive (forward) and negative (backward). It is analogous to the Fourier series of a single-variable function. The full spectrum results from taking the Fourier transform of the complex position data $x + jy$, along with some additional processing; mathematical details can be found in Mullen [19]. This type of plot quickly visualizes underlying whirl behavior that would not be evident in time domain data or in the more common *half spectrum plot*, which is the Fourier transform of an individual displacement direction and contains only positive frequencies. This makes the full spectrum plot a powerful tool for fault diagnosis and performance improvement [23].

In this work, a MATLAB function written by Mullen was used to generate full spectrum plots given time domain x- and y-displacement data [19]. Full spectrum plots for data modeling the examples in Figure 4.3 are given in Figure 4.14.

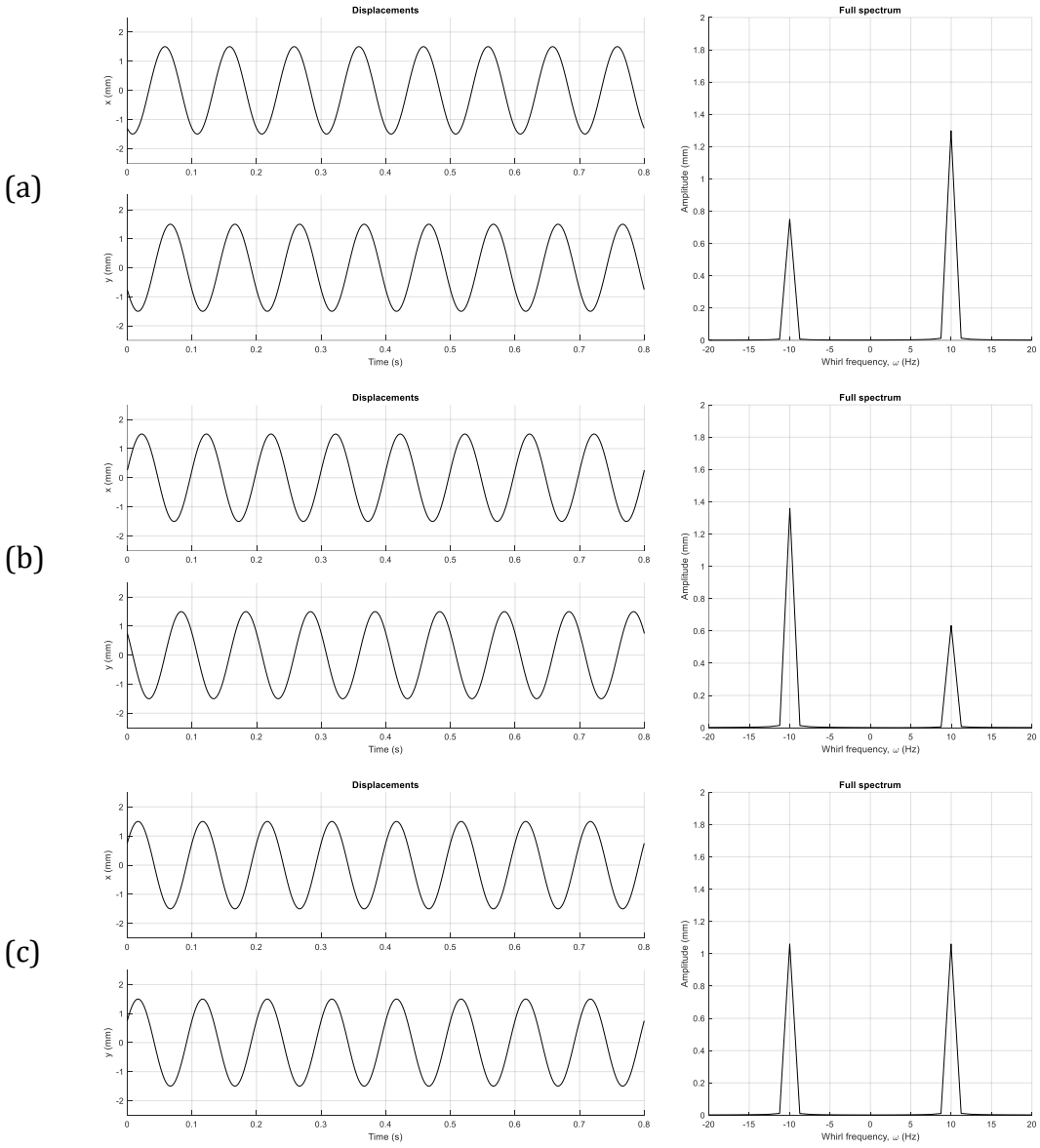


Figure 4.14. Full spectrum plots of whirl examples from Figure 4.3.

In each case, the two peaks correspond with the amplitudes of forward and backward whirl components shown in Figure 4.4, the forward being a positive

frequency and the backward a negative frequency. The peaks of the full spectrum plot can therefore quickly indicate whether whirl is forward, backward, or linear overall.

Performing this at many shaft speeds and plotting them together yields the complete cascade plot seen in Figure 4.15. Waviness along the cascade plot's ridge is due to discrete frequency values failing to exactly align with peak values. The MATLAB script that selects data segments and calls Mullen's function also determines whirl direction by measuring the peak amplitudes in positive and negative frequency and coloring the full spectra accordingly. We see that backward whirl occurs between roughly 500 and 750 RPM, matching our previous data, and the peaks with roughly linear whirl also correspond in frequency and amplitude (once their sum is taken).

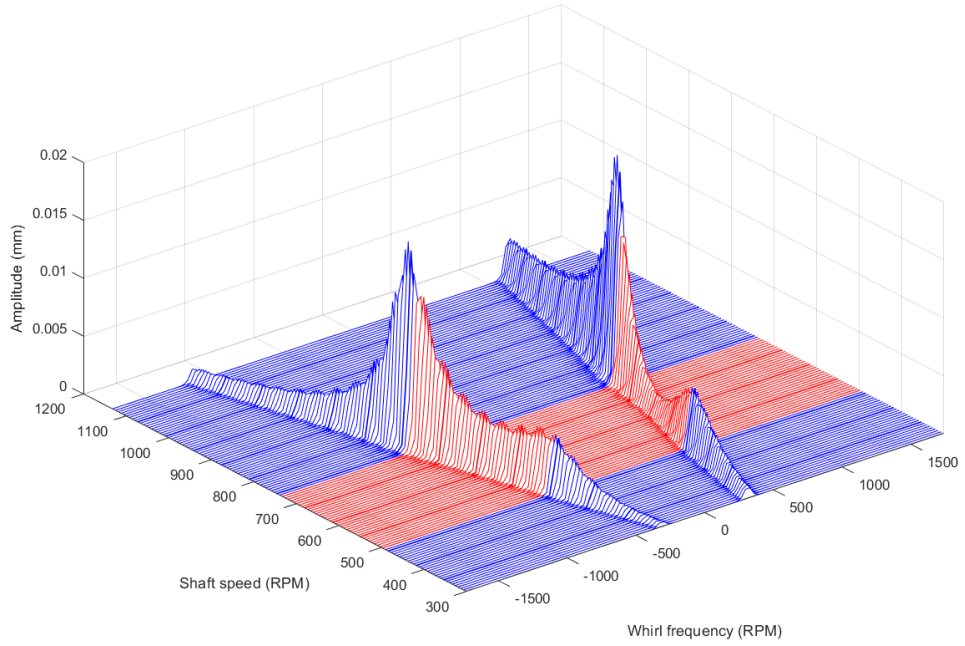


Figure 4.15. **Full spectrum cascade plot of flexible anisotropic rotor response**, with blue indicating forward whirl and red indicating backward.

Since few commercial software programs provide full spectrum plots, co-simulation shows its benefit here: Because models are run in Simulink, MATLAB scripts can programmatically analyze and visualize simulation data, allowing the user to produce plots according to their goals without needing to export data after a simulation run or be limited to a given program's visualization tools. While initial effort may be required to write the plots' scripts, doing so yields opportunity and control beyond that provided by currently available rotor dynamics programs.

Chapter 5

A BRIEF DEMONSTRATION OF TOUCHDOWN BEARING IMPACT

A dynamic co-simulation of a spinning rotor falling onto its protective touchdown bearing is shown in a simplified test, demonstrating solid body contact modeling, which current commercial rotor dynamics programs do not feature.

Important in AMB rotor design are *touchdown bearings*, which provide emergency containment of the shaft in the case of AMB failure or excessive whirl. While analysis of such systems is beyond the scope of this work, a quick test was performed to show that co-simulation can indeed model solid body contact, a feature nonexistent in current rotor dynamics software packages, to the author's knowledge.

In the model, a rigid cylinder represents a shaft, and a rigid cylindrical tube fixed in space models a touchdown bearing. Dimensions are provided in Figure 5.1, and the model is shown in Figure 5.2. A planar joint constrains the shaft center axially so that it does not fall out of the bearing during the simulation. The contact is created in Adams using the "Create a contact" tool under "Forces." Contact force parameters in Table 5.1 were chosen by trial-and-error to provide a visually interesting and intuitive impact response, as opposed to one that is accurate to reality. The shaft was given an initial speed of 2000 RPM.

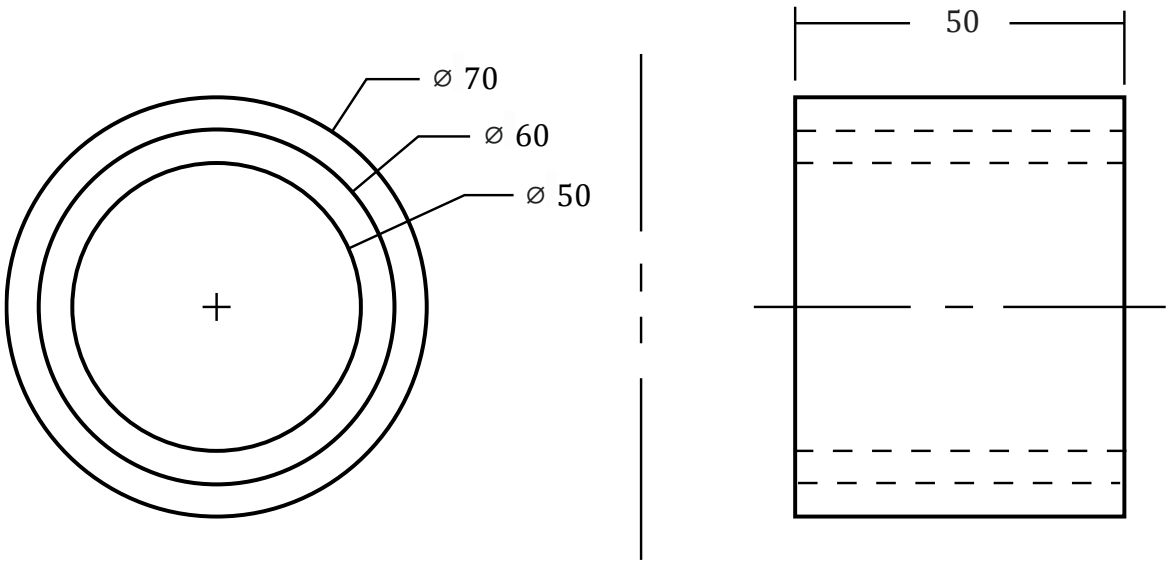


Figure 5.1. Dimensions of shaft and touchdown bearing (mm).

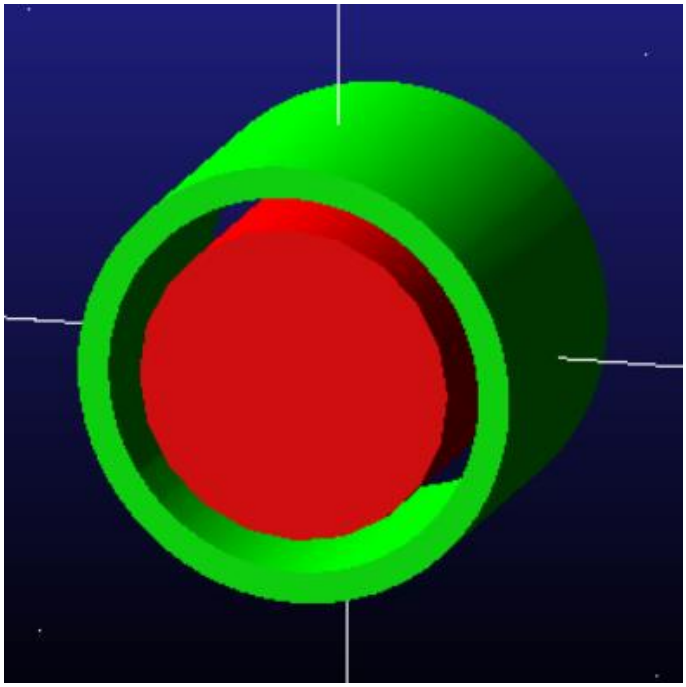


Figure 5.2. Shaft and touchdown bearing Adams model.

Table 5.1. Contact force parameters (*value differs from default).

Quantity	Value
Normal force	Impact
Stiffness*	1×10^{12}
Force exponent	2.2
Damping*	200
Penetration depth	1×10^{-4}
Friction force*	Coulomb
Coulomb friction	On
Static coefficient	0.3
Dynamic coefficient	0.1
Stiction transition vel.	0.1
Friction transition vel.	1.0

The simulation results show the path of the shaft CG as it rebounds several times and then begins slipping against the bearing surface (Figure 5.3). The dotted circle in the figure represents the clearance radius; when the CG path contacts this radius, we know that the shaft is contacting the bearing. Normal impulse forces show expected behavior; there is energy loss at each impact, as seen by the gradual decrease in bounce height. The model also includes frictional impulses; the rotor begins with 2000 RPM CCW rotation, and the first bounce shows an expected rebound to the left. Friction continues to be modeled once the shaft path nears rolling with slip, since the shaft trajectory along the clearance edge continues to favor the left.

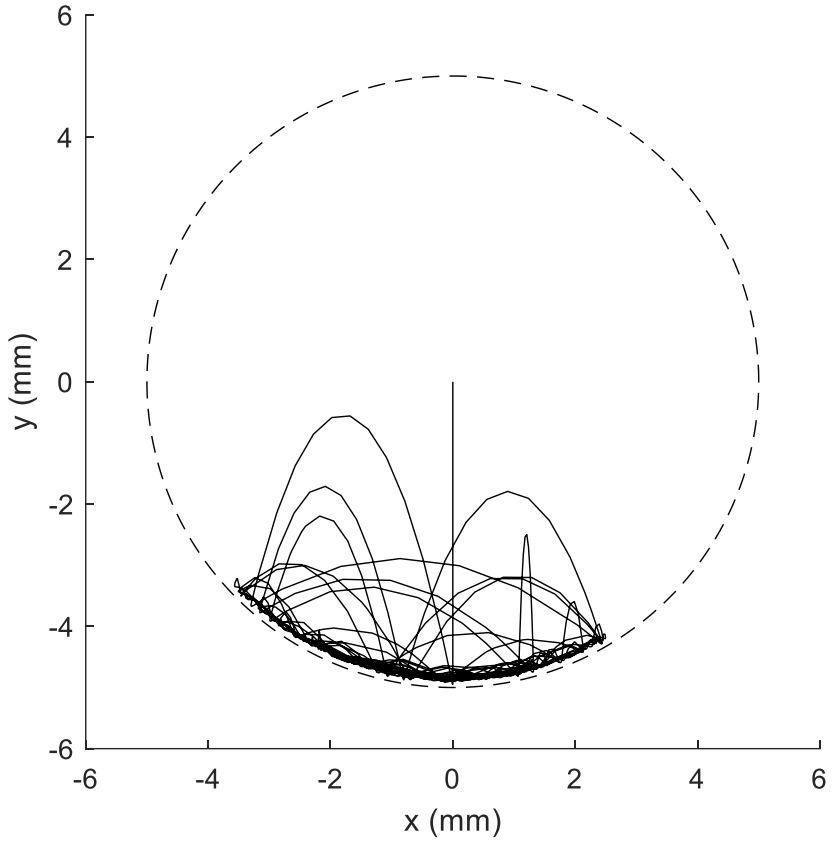


Figure 5.3. Simulated path of shaft center, with clearance radius shown.

This test shows promise for being capable of more realistic touchdown bearing simulations. If previous methods are included, a rotor created in Apex with FEA modeling could be modeled with both AMB control and impact testing within the same simulation environment, or even simultaneously in the same run. Since co-simulation does not restrict rotor models to state-space matrices, complex interactions like this one can be performed with little effort, especially compared to the difficult mathematics and programming that would be required to create such a simulation from first principles.

Chapter 6

CONCLUSION

The goals of this work were to demonstrate the accuracy and capability of Apex-Adams-Simulink co-simulation in modeling active magnetic bearing (AMB) rotors. This use of general-purpose commercial software combines advanced dynamics simulation (Adams) and finite element modeling (Apex) with advanced control design (Simulink) integrated seamlessly with visualization tools (MATLAB). No other known single piece of commercial software can perform both the dynamics and control necessary for some AMB rotor applications, except for hand-derived, manually programmed physics models and controllers, which are tedious to construct and prone to error. This co-simulation process enables wide-ranging modeling and visualization opportunities while relying on established computation methods and the convenience of familiar interfaces.

In this work, we first verified co-simulation's ability to model rotors. Without the resources to experimentally verify simulations, we focused on models that could be compared with well-known results from rotor theory. We showed a match for co-simulated AMB forces and rigid body dynamics in an unbalanced rigid AMB rotor (Chapter 2) and for the gyroscopic effect and flexible body dynamics in a spinning disk hanging on a steel wire (Chapter 3). The success of these tests allowed us to simulate more complex systems, one being a flexible anisotropic AMB rotor, for which we demonstrated several visualization tools generated programmatically in MATLAB (Chapter 4), and the other being a brief simulation of touchdown bearing impact,

showing the modeling of contact and subsequent bouncing and slipping motion (Chapter 5).

Several AMB rotor results have been shown in Adams co-simulation for the first time, including data over continuous shaft speed variation, frequency response, the use of Apex for finite element modeling, anisotropic rotor response, orbit shapes, forward and backward whirl, full spectrum analysis, cascade plots, and solid body contact. The combination of theoretical verification with the many new results listed above show promise that this method could be used for simulations even more complex and realistic than these, some of which are suggested below.

6.1. Future work

There are two immediate ways to better establish the work performed here:

1. *Experimental verification.* True validation of the software's capability should be based on comparison with practical models, which we did not have the resources to achieve. As of the completion of this report, a single magnetic bearing designed for a Bently Nevada RK4 rotor kit, originally the work of a senior project from California Polytechnic State University San Luis Obispo [25], is being updated for experimental testing by an adjacent graduate student researcher to the author. However, since this setup still requires mechanical contact, further practical design should include a fully suspended rotor with two AMBs.

2. *Modeling of mechatronics components*. This work assumed that appropriate control currents could be immediately injected into the AMBs, but real systems present several obstacles to this. Inductance of coil windings prevent rapid current changes, so a voltage-current control loop is usually a part of the control system. There are inevitable time delays in the mechatronics components, and sensors and power amplifiers have gains and involve noise. A more realistic simulation would include these, which is possible in Simulink.

Along with these improvements, other rotor models can be studied:

1. *Realistic touchdown bearing simulations*. We demonstrated the capability of contact modeling in co-simulation, but its accuracy to theory and experiment was beyond our scope. Another adjacent graduate student is currently pursuing this topic.
2. *Axially asymmetric rotors with multiple or distributed loads*. Our anisotropic rotor model was symmetric about its midpoint and contained a single heavy disk on a relatively light shaft. A more general rotor will be asymmetric, include many loads, and have distributed mass along its length, yielding responses at a level of complexity better matching those of industrial AMB rotors.
3. *Use of practical rotors CAD models*. The rotors in this work were made from simple cylinders and were easy to construct in Adams and Apex, but both applications can import and analyze models from CAD. Doing so could enable

even greater accuracy to real systems if components and support structures are modeled with true geometry.

This list is certainly not exhaustive, and as we have seen, it is also far from reaching the limits of the co-simulation process. If co-simulation proves itself in the ways that we anticipate, it will provide unprecedented modeling capabilities to future AMB rotor development.

REFERENCES

[1] G. Schweitzer and E. Maslen, *Magnetic Bearings*, Springer-Verlag Berlin Heidelberg, 2009.

[2] Ł. Breńkacz, Ł. Witanowski, M. Drosińska-Komor and N. Szewczuk-Krypa, "Research and applications of active bearings: A state-of-the-art review," *Mechanical Systems and Signal Processing,* vol. 151, 2021.

[3] Y. Liu, S. Ming, S. Zhao, J. Han and Y. Ma, "Research on automatic balance control of active magnetic bearing-rigid rotor system," *Shock and Vibration,* vol. 2019, 2019.

[4] M. Hutterer and M. Schrödl, "Modeling and mu-synthesis control of a flexible rotor stabilized by active magnetic bearings including current free control," *Journal of Sound and Vibration,* vol. 546, 2023.

[5] L. Di and Z. Lin, "Control of a flexible rotor active magnetic bearing test rig: a characteristic model based all-coefficient adaptive control approach," *Control Theory and Technology,* vol. 12, 2014.

[6] A. Noshadi, J. Shi, W. S. Lee, P. Shi and A. Kalam, "Robust control of an active magnetic bearing system using H∞ and disturbance observer-based control," *Journal of Vibration and Control,* vol. 23, no. 11, pp. 1857-1870, 2015.

[7] DELTA JS AG, "MADYN 2000 for Rotordynamics: Magnetic Bearings," [Online]. Available: https://www.delta-js.ch/en/software/madyn-2000-for-rotordynamics/magnetic-bearings/. [Accessed July 2023].

[8] Y. Zhao, S. Li, L. Zhao, X. Guo, Q. Wang and Y. Wen, "Sliding-mode control of permanent magnetic spherical motor based on co-simulation platform," in *IEEE Conference on Industrial Electronics and Applications*, 2016.

[9] M. Lashin, M. Fanni, A. M. Mohamed and T. Miyashita, "Dynamic Modeling and Inverse Optimal PID with Feed-forward Control in H∞ Framework for a Novel 3D Pantograph Manipulator," *International Journal of Control, Automation and Systems,* vol. 26, pp. 39-54, 2018.

[10] S. Jäger and S. Vogel, "Validation of a squeeze-film-damper test rig by using multibody cosimulation," *Multibody System Dynamics,* vol. 34, pp. 243-257, 2014.

[11] B.-S. Kim, J.-K. Park and D. Kim, "Integrated dynamic simulation of a magnetic bearing stage for particle free manufacturing process (linear magnetic bearings)," in *6th International Conference on Modeling, Simulation, and Applied Optimization*, Istanbul, Turkey, 2015.

[12] J. Zhang, Y. Hu, H. Wu and N. Wang, "Co-simulation of magnetic suspended rotor system: Research and application," in *International Conference on Mechatronics and Automation*, Changchun, China, 2009.

[13] H. Li, G. Cao, H. Qiu and J. Pan, "Nonlinear modeling and co-simulation with MATLAB, ADAMS and ANSYS for active magnetic bearings," in *Twelfth International Symposium on Magnetic Bearings*, Wuhan, China, 2010.

[14] K.-C. Lee, D.-K. Hong, Y.-H. Jeong, C.-Y. Kim and M.-C. Lee, "Dynamic simulation of radial active magnetic bearing system for high speed rotor using ADAMS and MATLAB co-simulation," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Seoul, Korea, 2012.

[15] S. Liu and J. Shi, "Co-simulation of magnetic bearing system based on Adams and MATLAB," in *6th International Conference on Machinery, Materials, Environment, Biotechnology and Computer*, Tianjin, China, 2016.

[16] H. Ouyang, C. Yu, G. Zhang, L. Mei, X. Deng and D. Wang, "Vibration suppression for flywheel energy storage system using modal decoupling control," *Advances in Mechanical Engineering,* vol. 10, no. 3, 2018.

[17] G. Genta, *Dynamics of Rotating Systems*, Springer Science+Business Media, Inc., 2005.

[18] C. Naugle, "Rotordynamic analysis of theoretical models and experimental systems," California Polytechnic State Univ., San Luis Obispo, CA, USA, 2018.

[19] M. Mullen, "Data acquisition, analysis, and modeling of rotordynamic systems," California Polytechnic State Univ., San Luis Obispo, CA, USA, 2020.

[20] X. Wu, C. Naugle and J. Meagher, "A Full Spectrum Analysis Methodology Applied to an Anisotropic Overhung Rotor," *Journal of Applied Mechanical Engineering,* vol. 05, 2016.

[21] R. Valle, F. Neves, R. J. de Andrade and R. M. Stephan, "Electromagnetic levitation of a disc," *IEEE Transations on Education,* vol. 55, no. 2, pp. 248-254, 2011.

[22] E. Krämer, *Dynamics of Rotors and Foundations*, New York: Springer-Verlag Berlin Heidelberg, 1993.

[23] D. E. Bently, C. T. Hatch and B. Grissom, *Fundamentals of Rotating Machinery Diagnostics*, Bently Pressurized Bearing Company, 2002.

[24] M. L. J. Adams, *Rotating Machinery Vibration: From Analysis to Troubleshooting*, CRC Press, 2009.

[25] S. Fowler and G. Olson, "Magnetic bearing for Bently Nevada RK4 rotor kits," California Polytechnic State Univ., San Luis Obispo, CA, USA, 2016.

[26] Wikipedia, "Euler angles," [Online]. Available: https://en.wikipedia.org/wiki/Euler_angles.

[27] X. Qiao and X. Tang, "The stability of magnetic levitation milling system based on modal decoupling control," *Shock and Vibration,* vol. 2020, 2020.

APPENDIX A: LIST OF VARIABLES

This section defines the subscripts and variables used throughout this work.

Symbol	Description
$\square_A$	...of bearing A
$\square_B$	...of bearing B
$\square_d$	disturbance...
$\square_L$	...of lower magnet
$\square_U$	...of upper magnet
$\square_x$	...in x x-component of...
$\square_y$	...in y y-component of...
$\mathbf{0}_{4\times4}$	4x4 matrix of zeros
a	signed distance from shaft CG to bearing A
A	magnet pole area
$\mathbf{A}$	state matrix
$\mathbf{A}_{\mathrm{CL}}$	state matrix, closed-loop
$\mathbf{A}_{\mathrm{OL}}$	state matrix, open-loop
α	angle from coord. axis to pole shaft tilt about *x*
b	signed distance from shaft CG to bearing B
$\mathbf{B}$	transformation matrix from bearing to CG coordinates
$\mathbf{B}_{\mathrm{OL}}$	input matrix, open-loop
β	shaft tilt about *y* phase of whirl component
β_b	phase of backward whirl component
β_f	phase of forward whirl component
c	signed distance from shaft CG to sensor A
c_{ij}	flexibility coefficient for displacement in *i* due to force or torque in *j*
c_x	shorthand for $\cos(\phi_x)$
c_y	shorthand for $\cos(\phi_y)$
$\mathbf{C}_{\mathrm{CL}}$	output matrix of closed-loop state space
$\mathbf{C}_{\mathrm{OL}}$	output matrix of open-loop state space

d	signed distance from shaft CG to sensor B
$\mathbf{D}$	feedthrough matrix
$\mathbf{D}_{\mathrm{CL}}$	damping matrix of closed-loop state space
$\mathbf{D}_{\mathrm{OL}}$	feedthrough matrix of open-loop state space
E	modulus of elasticity
$\mathbf{E}_{\mathrm{OL}}$	disturbance input matrix, open-loop
f	force
$\mathbf{f}_{\mathrm{b}}$	force vector in bearing coordinates
f_d	disturbance force
$\mathbf{f}_{\mathrm{d}}$	disturbance force vector (in CG coordinates)
f_{net}	net force
$\mathbf{G}$	gyroscopic matrix
i	current
i_0	bias current
i_x	differential current in x
i_y	differential current in y
I	moment of inertia integral control gain
$\mathbf{I}_{4\times4}$	4x4 identity matrix
j	unit imaginary number
k	magnetic coupling constant, in henry-meters
$\mathbf{K}$	stiffness matrix
$\mathbf{K}_{\mathrm{CL}}$	stiffness matrix of closed-loop state space
K_d	unbalance gain equivalent to $m_u r_u$
k_i	force-current factor
$\mathbf{K}_{\mathrm{i}}$	matrix of force-current factors
k_{ij}	stiffness coefficient for displacement in *i* due to force or torque in *j*
k_s	force-displacement factor
$\mathbf{K}_{\mathrm{s}}$	matrix of force-displacement factors
$\mathbf{K}_{\mathrm{sS}}$	transformed matrix of force-displacement factors, $\mathbf{B}\mathbf{K}_{\mathrm{s}}\mathbf{B}^{\mathrm{T}}$
λ	eigenvalue
m	mass of rotor
m_u	unbalance mass
$\mathbf{M}$	mass matrix
μ_0	vacuum permeability
n	number of coil turns
N	filter coefficient
ν	Poisson ratio

ω	vibration or whirl frequency
ω_n	natural frequency
Ω	shaft spin speed
P	proportional control gain
ϕ	phase
ϕ_u	angular position of imbalance relative to shaft
$\mathbf{q}$	shorthand for $\mathbf{q}_\mathrm{c}$
$\mathbf{q}_\mathrm{b}$	position vector in bearing coordinates
$\mathbf{q}_\mathrm{c}$	position vector in CG coordinates
$\mathbf{q}_\mathrm{s}$	position vector in sensor coordinates
r	position in complex representation $x + jy$
r_u	radial position of unbalance mass
R	radius of whirl component
R_b	radius of backward whirl component
R_f	radius of forward whirl component
ρ	density
s	air gap size between rotor and magnet
s_0	nominal air gap when rotor is centered
s_x	shorthand for $\sin \phi_x$
s_y	shorthand for $\sin \phi_y$
T_{drive}	drive torque applied to shaft
T_x	moment in x
T_y	moment in y
θ	angular position of shaft
x	position in the horizontal radial direction
$\mathbf{x}$	state vector
X	amplitude of whirl in x
x_{se}	shaft position at sensor in x
y	position in the vertical radial direction
$\boldsymbol{y}$	output vector state vector for magnet pair theory and simulation
Y	amplitude of whirl in y
y_{se}	shaft position at sensor in y
z	position in the axial direction
z_u	signed distance from shaft CG to unbalance mass

APPENDIX B: EULER ANGLES

This appendix describes the Euler angles used to measure the disk's tilt. In short, α is the first angle of the x-y'-z" (body 1-2-3) sequence, and β is the first angle of the y-x'-z" (body 2-1-3) sequence. Theory and notation are explained below.

Euler angles describe an object's orientation in 3D space. Three such angles can specify its orientation with respect to a global coordinate system. Most often, Euler angles are expressed as a sequence of three rotations by which the object's local coordinates are rotated from the global system's orientation to a desired position.

In Figure B.1 from [26], a common sequence is shown: the **z-x'-z" sequence**. The global coordinate system is lowercase xyz, while the desired local coordinate system is uppercase XYZ. Beginning with XYZ aligning with xyz, the z-x'-z" sequence is performed as follows:

1. **z** indicates that the first rotation occurs about the global z-axis, so the XYZ system is rotated by angle α about that axis. The new position of the XYZ system is x'y'z'.
2. **x'** indicates that the second rotation occurs about the x'-axis of the current x'y'z' system. After rotating by angle β, the new position is x"y"z".
3. **z"** indicates that the third rotation occurs about the z"-axis of the current x"y"z" system. Rotating by γ completes the rotation sequence.

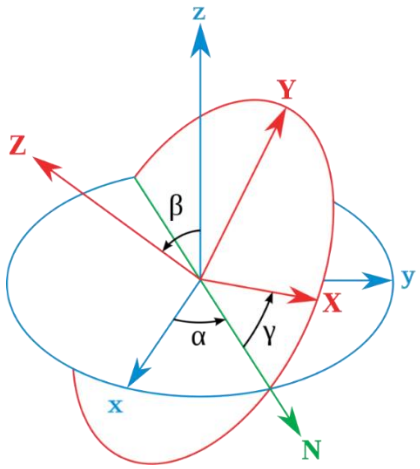


Figure B.1. The z-x'-z" Euler angle rotation sequence [26].

The sequence can be modified by choosing different axes of rotation than these for each step. So-called *proper Euler* angles use the same axis for the first and third rotations, but *Tait-Bryant* angles use one of each. A widely used example of the latter is the *yaw-pitch-roll* convention for aircraft: **z-y'-x"**. The first rotation about **z** is the aircraft's heading (yaw), the second rotation about **y'** is its elevation (pitch), and the final rotation about **x"** is its bank (roll). Figure B.2 shows this convention. In the image, X is the direction of the aircraft's flight, and Z is upward with respect to the aircraft, i.e. the left wing points toward Y. The rotations in order are ψ, θ, and ϕ.

In rotating machinery, flexibility in the rotor can cause angular deflections, which are important to characterize, particularly for disk-like components. The convention is to measure the orientation of a rotor disk with respect to a global coordinate system where the z-axis is the shaft's axis of symmetry; a rotation about the x-axis is then denoted α and that about the y-axis is β, as seen in Figure B.3 (note that these are not the same angles as α and β in Figure B.1).

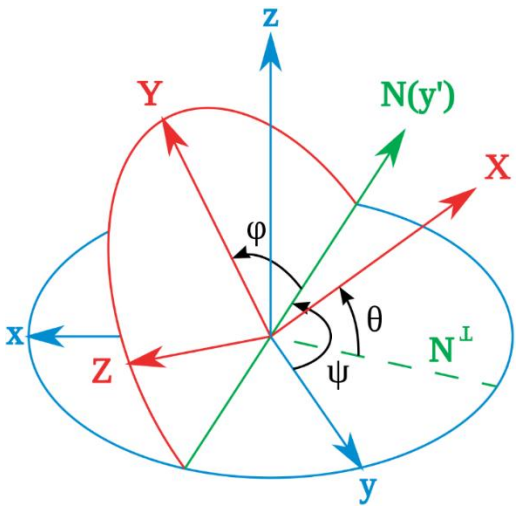


Figure B.2. The yaw-pitch-roll rotation sequence, z-y'-x" [26].

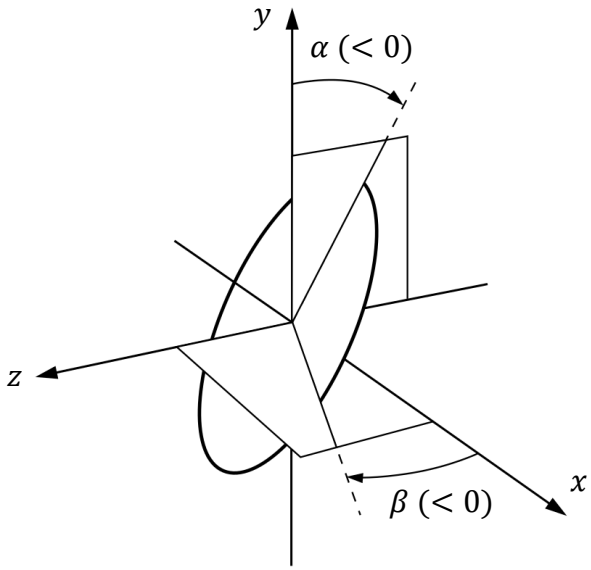


Figure B.3. Euler angles α and β of a rotor disk for a shaft directed along z.

Based on all this, as an Euler angle equivalent for α, we use the first angle of the x-y'-z" sequence, and for β, we use the first angle of the y-x'-z" sequence. In Adams, these are denoted "Body 1-2-3, first rotation" and "Body 2-1-3, first rotation" respectively.

APPENDIX C: STEPS TO CREATE THIN WIRE ROTOR IN MSC APEX AND ADAMS

These are the steps used to create the thin wire rotor model in MSC Apex and Adams.

The process uses Apex first and Adams second.

Model setup in MSC Apex

Assume single left-click for items without an explicit instruction.

1. Open Apex.
2. Save model with desired name and directory location.
3. *Geometry Create Tools* in the right icon ribbon.
    a. *Polyline*.
    b. Click on xz- or yz-plane.
    c. Draw a line from the origin to -500 mm along global z-axis by clicking its endpoints. It is also possible to edit the dimension of a line immediately after it is made or by using the *Edit* tool in *Geometry Create Tools*.
    d. Exit tool by pressing the middle mouse button (MMB) or the *Close* icon (red door).
4. *Meshing* in the right icon ribbon.
    a. *Curve mesh*.
    b. *Mesh size*: 25 mm
    c. Click on the part (the polyline). Nodes should appear along its length.
    d. Exit tool.
5. *Attribution* in the right icon ribbon.
    a. *Interface.*
    b. Click both endpoint nodes to add interfaces (wait for Apex to load after placing first interface point).
    c. Exit tool.
6. *Beams* in the right icon ribbon.
    a. *New*.
    b. *Solid circle* (default selection).
    c. *Radius*: 0.5 mm.
    d. *Close* icon to exit the editing of beam profile.
    e. *Create Beam Span*.
    f. Click on the part to add the beam span.
    g. Exit tool.

7. *Materials* in the right icon ribbon.

a. *Create Material*, rename if desired (e.g. "Steel (Adams default)").
b. Click the plus sign by *Constitutive Model* and check *Mass.*
    i. *Elasticity*
        1. *Elastic modulus*: 2.07e11 Pa (specify correct units)
        2. *Poisson ratio*: 0.29
    ii. *Mass*
        1. *rho*: 7801 kg/m^3
c. *Assign Material*.
d. Click on the part.
e. Exit tool.

8. In the Model Browser, right-click on the name of the part.
    a. *Place in Analysis Scene*.
    b. In the ribbon near the bottom center of the screen, change Environment Type to *Normal Modes.*
9. *Generate a Simulation Scenario* in the bottom center ribbon.
10. In the Model Browser, right-click on *Output Requests.*
    a. *Properties*.
    b. Navigate to the *MNF Export* tab.
    c. Check *Flex body output*.
    d. *Browse*, then specify desired MNF file location and name.
11. Right click on *Simulation Settings*.
    a. *Properties*.
    b. *Upper limit extraction mode number*: 20. For this setup, requesting more than 30 modes will cause the MNF generation to fail. Note that the first 6 modes will be rigid body modes.
12. *Run the Analysis Immediately* to generate the MNF file.

Model setup in Adams, after MNF creation

1. Open Adams.
2. A window should have appeared that gives an option for *New Model* or *Existing Model.* If so, click *New Model*, specify a desired model name and directory location, and then click *OK*.
    a. If not, the application may have opened to a blank workspace labeled "No model" in the upper left corner. If this is the case, click the icon in the upper ribbon that appears as a sheet of paper with a green plus (*Create a new model*) and give the model a desired name. Then go to *File*, *Save database as...*, right-click on the field, *Browse*, and specify a desired file name and location.
3. *Settings* in the upper ribbon
    a. *Units*: Change to base SI units for all quantities, including angles to radians
    b. *Gravity*: Click *-Z** (9.80665 m/s^2 in the -Z direction)
    c. *Icon size*: 1e-2 or 2e-2 (optional)

4. In the upper ribbon of tabs, *Bodies*, *Sphere.*

a. NOTE: This solid will be replaced by the MNF, so its size and type do not matter.
b. Click the origin first, then click anywhere other than the origin.
c. Rename the part "SHAFT" by changing its name in the Model Browser (the left ribbon) under *Bodies* and then pressing enter, OR by right-clicking on the part, clicking the current part name, and clicking *Rename*.

5. Right-click on "SHAFT" in the Model Browser.
   a. *Make Flexible.*
   b. *Import.*
   c. Right-click on the blank field after "MNF File" in the drop-down menu, *Browse,* select the MNF file to import, *OK*.
6. Expand "SHAFT_flex" in the Model Browser.
   a. There should be two "INT_NODE…" markers. Rename the one at the origin "TOPNODE," and rename the other one "BOTTOMNODE."
7. *Bodies*, *Cylinder.*
   a. Check both *Length* and *Radius.*
   b. *Length:* 2.5 cm
   c. *Radius:* 3.75 cm
   d. Click BOTTOMNODE, then click any other point directly below (in -Z).
   e. Rename the part "DISK."
   f. Double-click on DISK (OR right-click on DISK in Model Browser, *Modify*).
      i. *Material Type*: aluminum
8. In the upper ribbon of icons, right-click on the one that appears as a window with a gold square and a curved blue arrow.
   a. Click on the *Position: Move* tool, which has two gold squares with a blue arrow above them.
   b. Click the DISK, then the marker DISK.cm, then BOTTOMNODE. This moves the disk so that its center of mass is at the BOTTOMNODE.
9. In upper ribbon of tabs, *Connectors, Create a Fixed Joint*.
   a. Click SHAFT_flex, DISK, and then BOTTOMNODE.
10. *Connectors, Revolute joint.*
    a. In the second drop-down, select *Pick geometry feature*.
    b. Click SHAFT_flex, the ground (i.e. anywhere in the background), TOPNODE, and then any point or vector along the z-axis. This should create a cyan hinge icon at the origin with the hinge axis oriented vertically.
11. Double-click on SHAFT_flex.
    a. In the drop-down under *Inertia Modeling*, select *Full Coupling*, then *OK*.
12. *Bodies, Marker* (icon is a small set of coordinate axes).
    a. Place the marker at DISK.cm or BOTTOMNODE.
    b. Rename it "DISKPOS."

13. In the upper ribbon of tabs, *Design Exploration, Orientation measure* (gyroscope

icon).
    a. Leave *Measure name* as is for now.
    b. *Characteristic*: Body 1-2-3
    c. *Component*: First rotation
    d. *To Marker*: DISKPOS
    e. Leave *From Marker* blank.
    f. Uncheck *Create strip chart.*
    g. In the Model Browser, expand *Measures* and rename this one "phix."
14. Repeat the above but with the following changes:
    a. *Characteristic:* Body 2-1-3
    b. Rename this one "phiy."

This completes the model setup.

Example simulation

To run simulations, the following are example steps that can be performed to obtain the output in Figure 3.11, the 2nd backward whirling mode at 1000 RPM shaft speed:

1. In the upper ribbon of tabs, *Forces, General force vector*.
    a. Click DISK, then ground, then the fourth node from the disk end of the wire, the one located at coordinate (0, 0, -0.425) m.
    b. Under *Forces* in the Model Browser, rename it "Bwd2nd_1000RPM" or as desired.
    c. Double-click on the force.
        i. *Y Force*: STEP(time, 0.1, 0, 0.5, 0.1) – STEP(time, 0.5, 0, 0.9, 0.1)
        ii. *X Force*: STEP(time, 0.5, 0, 0.9, 0.1) – STEP(time, 0.9, 0, 1.3, 0.1)
        iii. Put 0 for the other four fields.
        iv. The above steps can also be performed with assistance from the Function Builder, which can be accessed by the button to the right of the field or by a right-click on the field.
2. Double-click on DISK.
    a. In *Category*, select *Velocity initial conditions*.
    b. Angular velocity about *Marker*; type "TOPNODE."
    c. Check *Z-axis*, then type 1000*2*PI/60, which is 1000 RPM in rad/s.
3. Double-click on SHAFT_flex.
    a. *Velocity ICs.*
    b. Repeat parts b and c of previous step.
4. In the upper ribbon of tabs, *Simulation*, then click the gear icon.
    a. Change the second drop down to *Step Size,* then type 1e-3.
    b. *End time*: 10 (in seconds).
    c. Click the green play button to run the simulation.
    d. Note that the stop button pauses the simulation and can be pressed while the simulation is running.
5. When the simulation is over, click *Plotting* (bottom-right icon with a graph), OR

in the upper ribbon of tabs, *Results*, *Post-processor*.
   a. Near the bottom, set *Source* to *Objects*.
   b. Select *body, DISK, CM_Position,* both *X* and *Y*, and then *Add curves*.
6. *File, Print*.
   a. Select *File*, right-click on the field, *Browse*, specify desired name and location, *OK*.

This completes the generation of the model and whirl figure in the report.

Activating and deactivating entities

Activating and deactivating entities is also useful, in this case to implement multiple sets of forces but only use one at a time. To do so:

1. Right-click on the force in the Model Browser, *(De)activate*.
   a. Check or uncheck *Objects Active*, which should also check or uncheck *Object's Dependents Active.*