Article

Speed-Dependent Eigenmodes for Efficient Simulation of Transverse Rotor Vibration

Jocelyn Kluger 1,*, Lynn Crevier 2 and Martin Udengaard 1

1 MathWorks, Natick, MA 01760, USA
2 aPriori Technologies, Concord, MA 01742, USA
* Correspondence: jkluger@mathworks.com

Abstract: Accurate, computationally efficient simulations enable engineers to design high-performing, cost-efficient, lightweight machines that can leverage models of predictive controls and digital twin predictive maintenance schedules. This study demonstrates a new speed-dependent eigenmode method for accurately and efficiently simulating shaft transverse vibrations. The method involves first independently computing shaft eigenmodes over a range of operating speeds, then correlating the eigenmodes across the different speeds during compilation, and finally adjusting modal properties gradually in accordance with a lookup method during simulation. The new method offers several distinct advantages over the traditional static eigenmodes and Craig-Bampton methods. The new method maintains accuracy over a large range of shaft rotation speeds whereas the static eigenmodes method does not. The new method typically requires fewer modal degrees of freedom than the Craig-Bampton method. Whereas the Craig-Bampton method is limited to modeling changes at the boundaries, the new method is suitable for modeling changing body properties as well as boundary-based changes. For this paper, a fluid-bearing-supported 10 MW direct-drive wind turbine drive shaft is tested virtually in a simulation model developed in Simscape™ Driveline™. Using the simulation statistics, this study compares the accuracy and computational efficiency of the speed-dependent eigenmode method to the traditional finite lumped element, static eigenmode, and Craig–Bampton methods. This paper shows that the new method simulates the chosen system 5 times faster than the traditional lumped mass method and 2.4 times faster than the Craig-Bampton method.

Keywords: modal analysis; model order reduction; real-time simulation; rotor dynamics; vibration

1. Introduction

Rotors play crucial energy transfer roles in machines used for energy generation, transportation, and other industrial applications. Since the rotor and its loads are never perfectly symmetrical about the rotor axis, the rotational motion is always accompanied by some transverse vibration. This vibration can contribute to lower quality power output, machine downtime due to failures, and a reduced machine lifetime. If the vibration is excessive, catastrophic failure occurs [1].

1.1. The Role of Rotor Dynamics Modeling

Modeling rotor dynamics throughout the machine design process and deployment can improve machine performance. Designers need to keep rotating machinery vibration within an acceptable tolerance [2]. Modeling the vibration response of rotors with different structural and control parameters can help optimize the machine performance and cost. During the design verification stage for complex machines, simulations of many different load cases help build confidence that a design will operate successfully throughout its lifetime. Simulating the machine response in many scenarios is lower cost and less time-consuming than what is possible using a hardware prototype. Simulation often plays a key role before creating a hardware prototype [3].
The ability to simulate physical systems in real time is a key stage of model-based design for efficiently and robustly developing control systems, software, and hardware. If the plant model can run in real time, then it can be used in place of a hardware prototype to test a processor and production code. Simulation permits testing in conditions that would damage equipment or personnel and before a hardware prototype exists. Moreover, simulation reduces costs and enables non-stop testing. In addition to the development process, real-time simulation can be used in the final product, such as in human-in-the-loop simulators for training, model-predictive controllers, and digital twins for predictive maintenance [4].

1.2. Rotor Dynamics Modeling Challenges

Traditional rotor dynamics models require tradeoffs between simulation fidelity and computational cost, as illustrated in Figure 1. Rotor dynamics behavior is complex. It involves elastic deformation with multiple resonant frequencies; asymmetric, speed-dependent bearing stiffness and damping; and disk gyroscopic effects. These effects can lead to instabilities that make modeling them both challenging and important [5].

![Figure 1. Non-reduced rotor dynamics models typically have a simulation performance tradeoff between the computational efficiency and model detail. Finite element analysis (FEA) tends to be at the upper end of the curve. A good reduced-order model (ROM) decreases the computational cost without decreasing the detail, which can affect the simulation accuracy.](image)

Model order reduction methods for rotors enable detailed and accurate simulations at a lower computational cost, which can accelerate and enhance the design process or permit real-time simulation. Rotor model order reduction methods are often used to design shaft structures and controllers, analyze them, and optimize their sensitivity. Key goals for any model reduction method include linear independence, completeness, high accuracy, low computational expense, simplicity, and automation. There are several highly common model reduction methods for rotors, each of which achieve some but not all of these goals [5].

Guyan reduction (or static reduction) partitions the system into primary and secondary degrees of freedom (DOFs). External loads excite the primary DOFs that, in turn, excite the secondary DOFs. This method is accurate only when the stiffness properties dominate the inertia properties [5,6].

Real modal analysis uses eigenanalysis to reduce and decouple the rotor equations of motion. Complex modal analysis that retains damping in the mode computation is not commonly used because it is overly complex and frequency-dependent. Typically, only modes less than twice the excitation frequency are retained. This method can reduce the computational cost of up to tens of thousands of DOFs by a factor of 10–100. However, for rotor systems that include fluid film bearings and other speed-dependent shaft properties, this traditional eigenanalysis can be inaccurate for some shaft rotation speeds, so it often cannot be used for transient system analysis [5].
Component mode synthesis is one of the most common model order reduction methods for rotor dynamics because it can accurately model speed-dependent effects. As with Guyan reduction, component mode synthesis partitions a system into substructures and then further partitions the substructures into primary and secondary DOFs. The primary DOFs interface with the other substructures. Static modes relate the primary DOFs to the secondary DOFs. By using assumed states for the primary DOFs, eigenanalysis reduces the secondary DOFs to inner modal DOFs. To compute the inner modes, some variations of component mode synthesis assume that the primary DOFs are free whereas the popular Craig–Bampton method assumes that they are stationary [6,7].

Component mode synthesis is well-suited for modeling the damping and time-varying properties of rotor supports because the inner mode properties are independent of the supports acting on the interface DOFs. However, in component mode synthesis the interface modes and inner modes are not independent and this DOF dependency complicates conceptual analysis of the system. Furthermore, mixing dynamics that vary both quickly and slowly results in a numerically stiff system that limits the computational efficiency of a model, and in component mode synthesis, bearing support DOFs tend to be stiffer than the modal DOFs [5].

This paper describes a new speed-dependent eigenmode method that maintains accuracy when shaft properties are speed-dependent during varying rotation speed in a simulation. The method involves first choosing several shaft operating speeds and independently computing eigenmodes of the shaft-bearing system at each speed. Then, the method correlates the eigenmodes across the different speeds to generate lookup tables of modal properties that gradually vary with the shaft speed. Finally, during simulation, a lookup method gradually adjusts modal properties as the shaft speed changes. We hypothesize that, for systems with certain characteristics, the new method will have better numerical performance than the Craig–Bampton method because incorporating support properties into the modes eliminates interface DOFs and decreases system numerical stiffness. We expect that the new method will have a simulation performance penalty compared to the traditional static eigenmode method because of the cost of lookup table computations. However, compared to the static eigenmode method, the new method maintains accuracy during transient simulations because it updates parameters that depend on transient variables.

1.3. Paper Organization

The main goals of this work are to describe the new speed-dependent eigenmode method for modeling transverse vibrations and to compare its performance to that of the non-reduced lumped mass model, static eigenmode reduction method, and Craig–Bampton reduction method in a simulation with variable shaft speed. Section 2 summarizes the four modeling approaches. Section 3 describes a wind turbine driveline simulation model implemented in Simscape™ Driveline™ that is used for the model performance comparison [8]. Section 4 describes and discusses the results of the case study. Section 5 summarizes key conclusions and future steps.

2. Modeling Methods

This section describes four methods for modeling the shaft bending vibrations: the lumped mass model, the Craig–Bampton method, the static eigenmodes method, and the new speed-dependent eigenmodes method. The last three methods are model order reduction methods performed on the lumped mass model. The lumped mass model considers shaft bending caused by a force unbalance that excites the shaft at a multiple (pole) of the rotation. The shaft diameter varies along the shaft axis. The shaft is supported by fluid film bearings that exhibit different linearized stiffness and damping properties as the shaft speed changes.

In this study, all four methods are implemented in a Simscape™ model in the Simulink simulation environment [9,10]. The lumped mass model, static eigenmode method, and
speed-dependent eigenmode method are implemented in the Simscape Driveline Flexible Shaft block [11]. The Craig–Bampton method is implemented using a custom Simscape component. A link to the models and scripts used to simulate the system is provided in [12].

2.1. Lumped Mass Model

This section summarizes the well-known lumped mass model for shaft bending [2,11]. The model divides the shaft into flexible linear elements with mass concentrated at nodes on either end. Each node has four DOFs: translational and rotational motions in both the $x$ and $y$ directions perpendicular to the shaft axis, $z$. That is, the DOFs in each node are:

$$\vec{x}_i = \begin{bmatrix} x_i \\ y_i \\ \theta_i \\ \phi_i \end{bmatrix}.$$  

The lumped mass equation of motion is:

$$M \ddot{\vec{x}} + (B + G_{Disk} \Omega) \dot{\vec{x}} + (K + G_{Disk} \Omega) \vec{x} = \vec{f},$$  

where $\Omega$ is the shaft speed.

The mass matrix of the $i$th flexible element, which affects the $i$th and $(i + 1)$th nodes, is:

$$M_i = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_d & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{m}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{m}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_d & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_d \end{bmatrix},$$  

where $m$ is the mass of the shaft element and $I_d$ is the diametric mass moment of inertia of half the shaft element about the $x$ and $y$ axes.

The mass matrix of a rigid disk attached to the $i$th node is modeled as:

$$M_{Disk,i} = \begin{bmatrix} m_{Disk,i} & 0 & 0 & 0 \\ 0 & m_{Disk,i} & 0 & 0 \\ 0 & 0 & I_{d,Disk,i} & 0 \\ 0 & 0 & 0 & I_{d,Disk,i} \end{bmatrix},$$  

where $m_{Disk,i}$ is the disk mass and $I_{d,Disk,i}$ is the disk diametric mass moment of inertia about the $x$ and $y$ axes.

The polar mass moment of inertia of a rigid disk attached to the $i$th node is modeled as:

$$G_{Disk,i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{p,Disk,i} \\ 0 & 0 & I_{p,Disk,i} & 0 \\ I_{p,Disk,i} & 0 & 0 & 0 \end{bmatrix}.$$
The stiffness matrix of the $i$th flexible element is modeled as an Euler-Bernoulli beam:

$$K_i = \frac{2EI}{l^3} \begin{bmatrix}
6 & 0 & 0 & 3l & -6 & 0 & 0 & 3l \\
0 & 6 & -3l & 0 & 0 & -6 & -3l & 0 \\
0 & -3l & 2l^2 & 0 & 0 & 3l & l^2 & 0 \\
3l & 0 & 0 & 2l^2 & -3l & 0 & 0 & l^2 \\
-6 & 0 & 0 & -3l & 6 & 0 & 0 & -3l \\
0 & -6 & 3l & 0 & 0 & 6 & 3l & 0 \\
0 & -3l & l^2 & 0 & 0 & 3l & 2l^2 & 0 \\
3l & 0 & 0 & l^2 & -3l & 0 & 0 & 2l^2
\end{bmatrix}, \quad (6)$$

where $EI$ is shaft rigidity and $l$ is the element length along the shaft axis. The described model neglects stress stiffening and spin softening due to centrifugal forces, which are negligible because the shaft spins at low speeds. We may incorporate these effects into the models as part of future work.

Fluid film bearings that support the shaft at the $i$th node are modeled as linearized stiffness and damping coefficients that vary with the shaft speed, $\Omega$. The linearized coefficients are valid for small shaft amplitude motions about the equilibrium position. Details on the linearization of fluid film bearings are provided in [13]. A bearing at the $i$th node is characterized by a stiffness matrix, $K_{\text{Support},i}$, and damping matrix, $B_{\text{Support},i}$:

$$K_{\text{Support},i}(\Omega) = \begin{bmatrix}
K_{xx}(\Omega) & K_{xy}(\Omega) & 0 & 0 \\
K_{yx}(\Omega) & K_{yy}(\Omega) & 0 & 0 \\
0 & 0 & K_{\Theta\Theta}(\Omega) & 0 \\
0 & 0 & 0 & K_{\Phi\Phi}(\Omega)
\end{bmatrix}, \quad (7)$$

$$B_{\text{Support},i}(\Omega) = \begin{bmatrix}
B_{xx}(\Omega) & B_{xy}(\Omega) & 0 & 0 \\
B_{yx}(\Omega) & B_{yy}(\Omega) & 0 & 0 \\
0 & 0 & B_{\Theta\Theta}(\Omega) & 0 \\
0 & 0 & 0 & B_{\Phi\Phi}(\Omega)
\end{bmatrix}. \quad (8)$$

Additional damping is modeled proportionally to the system mass and stiffness through coefficients $\alpha$ and $\beta$, respectively, so that the total system damping is:

$$B = \alpha M + \beta K(\Omega) + B_{\text{Support}}(\Omega). \quad (9)$$

Damping proportional to stiffness typically represents structural hysteretic damping [14]. In the case study of Section 3, the main contribution to damping in the rotor is generally due to the supports, while the small proportional damping contributes to model numeric stability. The model neglects the internal rotational damping that is typically caused by material stress–strain hysteresis and sliding friction between rotor components because the shaft speed remains well below the whirling mode frequencies [14]. We may add internal rotational damping effects as part of future work.

A static unbalanced magnetic force that acts on the $i$th node to excite bending in the $x$ and $y$ directions is:

$$\vec{f}_i = \begin{bmatrix}
f_i \cos \left( p\omega t + \xi_{\text{Offset}} \right) \\
f_i \sin \left( p\omega t + \xi_{\text{Offset}} \right)
\end{bmatrix}, \quad (10)$$

where $f_i$ is the magnitude of the unbalanced force of a permanent magnetic generator, $p$ is the number magnetic pole pairs, and $\xi_{\text{Offset}}$ is a phase offset [15,16].
2.2. Static Eigenmode Reduced Model

Standard eigenmode model order reduction reduces the number of simulated DOFs for the lumped mass model of Section 2.1. For rotating shafts, the standard eigenmode method chooses a nominal shaft speed, $\Omega_{\text{Nominal}}$, for calculating the modal properties. The method neglects the effects of modal properties changing with the shaft speed. Therefore, if the system properties change significantly as the shaft rotation speed changes, then the simulation results are inaccurate at rotation speeds that differ from the nominal speed. The properties of some fluid film bearings change significantly depending on the shaft rotation speed.

For a selected operating speed, $\Omega_{\text{Nominal}}$, the undamped eigenmodes are computed according to the extended modal reduction method for non-symmetric mass and stiffness matrices [17]. We use a traditional eigenproblem solver implemented in the MATLAB® `eigs` function [18,19]. The `eigs` function takes the form:

$$\begin{align*}
[R_{\text{static}}, D_{\text{static}}] &= \text{eigs}(K(\Omega_{\text{Nominal}}), M, \text{numModes}). \\
[L_{\text{static}}, D_{\text{static}}] &= \text{eigs}(K^T(\Omega_{\text{Nominal}}), M^T, \text{numModes})
\end{align*}$$

(11a) (11b)

The MATLAB `eigs` function outputs the right eigenvectors in the columns of $R_{\text{static}}$, the left eigenvectors in the columns of $L_{\text{static}}$, and the corresponding eigenvalues in the diagonal matrix, $D_{\text{static}} = \omega_{\text{static}}^2$. The integer input `numModes` limits the results to the `numModes` eigenmodes with the lowest frequencies.

Following the traditional eigenmode method, the governing equation for shaft bending, Equation (2), can be reduced from $4 \times \text{numNodes}$ DOFs to `numModes` DOFs. In the case study of Section 3, the eigenmode model order reduction is applied to the lumped mass model with 50 nodes (200 DOFs per Equation (1)), so $K$ and $M$ are matrices with sizes of 200-by-200. The eigenvector matrices $R_{\text{static}}$ and $L_{\text{static}}$ have sizes of 200-by-`numModes`, where the number of retained modes `numModes` ranges from 8 to 64, as shown in Table A1. The reduced equation of motion has the form:

$$M_{\text{Modal}} \ddot{\vec{x}}_{\text{Modal}} + (B_{\text{Modal}} + G_{\text{Modal}} \Omega) \vec{x}_{\text{Modal}} + (K_{\text{Modal}} + G_{\text{Modal}} \Omega) \vec{x}_{\text{Modal}} = \vec{f}_{\text{Modal}},$$

(12)

where

$$\begin{align*}
\vec{x}_{\text{Modal}} &= R_{\text{static}} \vec{x}, \\
M_{\text{Modal}} &= L_{\text{static}}^T M R_{\text{static}}, \\
K_{\text{Modal}} &= L_{\text{static}}^T K(\Omega_{\text{Nominal}}) R_{\text{static}}, \\
G_{\text{Modal}} &= L_{\text{static}}^T G_{\text{Disk}} R_{\text{static}}, \\
B_{\text{Modal}} &= L_{\text{static}}^T B(\Omega_{\text{Nominal}}) R_{\text{static}}, \\
\vec{f}_{\text{Modal}} &= L_{\text{static}}^T \vec{f}.
\end{align*}$$

2.3. Speed-Dependent Eigenmode Reduced Model

In this paper, we consider an extension of the standard eigenmode model order reduction method to increase simulation accuracy while incurring a moderate computational cost for rotational shafts with speed-dependent properties. This extended method involves choosing several reference rotational velocities, $\Omega_{\text{Reference}}$, for calculating the modal properties. The method correlates modes across neighboring reference rotational velocities by comparing and maximizing correlated mode shape similarity. Then, the method uses the mode shapes to generate modal property lookup tables. As the mode shapes gradually change, the modal properties gradually change. During simulation, as the shaft speed changes, the lookup table interpolation adjusts the modal properties based on the shaft speed.
The speed-dependent eigenmode method is similar to the traditional eigenmode method in that it starts by computing the undamped modal properties. However, the speed-dependent eigenmode method independently computes the modal properties at multiple reference shaft speeds, $\Omega_{\text{Reference},i}$, rather than at a single speed. The case study of Section 3 uses reference speeds of $[1, 2.5, 5, 7.5, 10]$ RPM to compute the eigenmodes:

$$ [R_{SD,i}, D_{SD,i}] = eigs \left( K \left( \Omega_{\text{Reference},i} \right), M, \text{numModes} \right). \quad (19a) $$

$$ [L_{SD,i}, D_{SD,i}] = eigs \left( K^T \left( \Omega_{\text{Reference},i} \right), M^T, \text{numModes} \right). \quad (19b) $$

The MATLAB $eigs$ function outputs the right eigenvectors in each column of $R_{SD,i}$, the left eigenvectors in each column of $L_{SD,i}$, and the corresponding eigenvalues in $D_{SD,i} = \omega^2_{SD,i}$.

The method computes the modal properties at each shaft speed $i$ according to:

$$ \vec{x}_{\text{Modal},i} = R_{SD,i} \vec{x}, \quad (20) $$

$$ M_{\text{Modal},i} = L_{SD,i}^T M R_{SD,i}, \quad (21) $$

$$ K_{\text{Modal},i} = L_{SD,i}^T K \left( \Omega_{\text{Reference},i} \right) R_{SD,i}, \quad (22) $$

$$ G_{\text{Modal},i} = L_{SD,i}^T G_{\text{Disk}} R_{SD,i}, \quad (23) $$

$$ B_{\text{Modal},i} = L_{SD,i}^T B \left( \Omega_{\text{Reference},i} \right) R_{SD,i}, \quad (24) $$

$$ \vec{f}_{\text{Modal},i} = L_{SD,i}^T \vec{f}. \quad (25) $$

Simscape Driveline uses an automated approach to sort modes at the different reference speeds. The core concept of the sorting method is a modal assurance criterion similar to the ones described in Mogenier et al. [20]. This approach is robust to mode frequency crossing behavior, which is when the modal frequencies of two distinct modes swap order at neighboring reference shaft speeds. Section 3.2 describes a Campbell diagram generated using the speed-dependent eigenmode method and shows that the algorithm captured multiple frequencies crossing each other.

Once the modes at different shaft speeds are correlated with each other and sorted, the corresponding modal properties are stored in three-dimensional tables, where each two-dimensional sheet within the table corresponds to the modal properties at each reference shaft speed. At each time step of the simulation, the 2D modal property matrices are updated based on lookup table linear interpolation of the 3D tables. Therefore, as shaft speed varies during a simulation, the modal properties remain accurate.

The final speed-dependent modal governing equation is:

$$ \ddot{\vec{x}}_{\text{Modal}} + M_{\text{Modal}}^{-1}(B_{\text{Modal}} + G_{\text{Modal}} \Omega) \dot{\vec{x}}_{\text{Modal}} + M_{\text{Modal}}^{-1}(K_{\text{Modal}} + G_{\text{Modal}} \Omega) \vec{x}_{\text{Modal}} = M_{\text{Modal}}^{-1} \vec{f}_{\text{Modal}} \quad (26) $$

where the terms are normalized by the modal mass matrix to reduce the number of lookup table computations.

During simulation, the four modal properties, $M_{\text{Modal}}^{-1} B_{\text{Modal}}, M_{\text{Modal}}^{-1} G_{\text{Modal}}, M_{\text{Modal}}^{-1} K_{\text{Modal}}$, and $M_{\text{Modal}}^{-1} \vec{f}_{\text{Modal}}$ are computed according to the aforementioned lookup table method:

$$ C(t) = \text{tablelookup} \left( \Omega_{\text{Reference}}, C_{\text{Reference}}, \Omega \right), \quad (27) $$

where $C$ represents the four modal properties listed above and $\Omega$ is the transient shaft speed during simulation.
2.4. Craig–Bampton Reduced Model

In this paper, we implement the standard Craig–Bampton reduction method as a performance benchmark for the speed-dependent eigenmode method of Section 2.3 [6,7]. As with the eigenmode methods, we apply the Craig–Bampton method to the shaft lumped mass model described in Section 2.1.

Following the traditional Craig–Bampton method, the partitioning matrix, \( T_{\text{part}} \), partitions the DOFs into primary nodes, which interface with the supports and external forcing, and secondary nodes. Each primary node contains four DOFs as described in Equation (1). Mathematically, the partitioning process is:

\[
\vec{x}_{\text{part}} = T_{\text{part}} \vec{x}, \tag{28}
\]

where

\[
\vec{x}_{\text{part}} = \begin{bmatrix} \vec{x}_{\text{Primary}} \\ \vec{x}_{\text{Secondary}} \end{bmatrix}. \tag{29}
\]

The partitioned equation of motion matrices are:

\[
M_{\text{part}} = T_{\text{part}} M_{\text{part}}', \tag{30}
\]

\[
K_{\text{part}} = T_{\text{part}} K_{\text{part}}', \tag{31}
\]

\[
C_{\text{part}} = T_{\text{part}} C_{\text{part}}', \tag{32}
\]

\[
B_{\text{part}} = T_{\text{part}} B_{\text{part}}', \tag{33}
\]

\[
\vec{f}_{\text{part}} = T_{\text{part}} \vec{f}. \tag{34}
\]

The partitioned stiffness matrix has quadrants corresponding to the primary and secondary DOFs:

\[
K_{\text{part}} = \begin{bmatrix} K_{pp} & K_{ps} \\ K_{sp} & K_{ss} \end{bmatrix}. \tag{35}
\]

The static reduction matrix is obtained from the partitioned stiffness matrix:

\[
S_{CB} = \begin{bmatrix} I_{pp} \\ -\text{inv}(K_{ss})K_{sp} \end{bmatrix}. \tag{36}
\]

The columns of the static reduction matrix are the Craig–Bampton static modes. In each static mode, one primary (interface) DOF undergoes a unit displacement. The condensation matrix, \(-\text{inv}(K_{ss})K_{sp}\), relates the secondary (inner) DOFs to each primary DOF.

In addition to the static modes, the Craig–Bampton method computes the eigen-modes of the secondary (inner) DOFs. This study uses the MATLAB \textit{eigs} function for the eigenmode computation according to:

\[
[V_s, \sim] = \text{eigs}(K_{ss}, M_{ss}, \text{numInnerModes}), \tag{37}
\]

where \text{numInnerModes} is an integer limit to the number of retained inner modes. The inner modes have stationary boundaries that correspond to the primary DOFs. The inner modes are the columns of the dynamic transformation matrix:

\[
D_{CB} = \begin{bmatrix} 0_{pp} \\ V_s \end{bmatrix}. \tag{38}
\]

The final Craig–Bampton transformation matrix is:

\[
T_{CB} = [S_{CB} D_{CB}] \tag{39}
\]
The final Craig–Bampton modal properties are:

\[ M_{CB} = T_{CB}M_{part}T_{CB}^T, \]  
(40)

\[ K_{CB} = T_{CB}K_{part}T_{CB}^T, \]  
(41)

\[ G_{CB} = T_{CB}G_{part}T_{CB}^T, \]  
(42)

\[ B_{CB} = T_{CB}B_{part}T_{CB}^T, \]  
(43)

\[ \dot{f}_{CB} = T_{CB}\dot{f}. \]  
(44)

The final Craig–Bampton governing equation is:

\[ \ddot{x}_{CB} + M_{CB}^{-1}(B_{CB} + G_{CB}\Omega)\dot{x}_{CB} + M_{CB}^{-1}\left(K_{CB} + G_{CB}\Omega\right)x_{CB} = M_{CB}^{-1}\dot{f}_{CB}. \]  
(45)

The system forcing accounts for both the static unbalanced magnetic force given by Equation (10) and the support forcing. That is, \( K_{CB} \) and \( B_{CB} \) do not account for the support loads. The support loads acting at the primary nodes \( i \) are:

\[ \dot{f}_i = -K_{\text{Support},i}(\Omega)\dot{x}_{\text{Support},i} - B_{\text{Support},i}(\Omega)\dot{x}_{\text{Support},i}, \]  
(46)

where lookup table interpolation computes the support stiffness and damping at the varying shaft speed, according to:

\[ K_{\text{Support},i}(\Omega) = \text{tablelookup}\left(\Omega_{\text{Reference}}, K_{\text{Support},i\text{Reference}}, \Omega\right). \]  
(47)

The Craig–Bampton transformation reduces the original \( 4 \times \text{numNodes} \) DOFs into \( 4P + S \) DOFs, where \( P \) is the number of interface (primary) nodes and \( S \) is the number of retained inner (secondary) modes. The number of inner modes required for simulation convergence generally depends on the system stiffness, excitation frequency, and location of transverse excitation. Systems with weaker shaft stiffnesses, higher excitation frequencies, and transverse excitations located at mode shape displacement peaks typically require more inner modes.

3. Case Study: Wind Turbine Driveline Model

To benchmark the different modeling methods, a fluid-bearing-supported 10 MW direct-drive wind turbine drive shaft with transverse vibrations is tested virtually in a simulation model developed in Simscape Driveline. The wind industry is generally moving towards using direct-drive generators in large offshore wind turbines because direct-drive generators likely offer better availability and efficiency and have reduced operation and maintenance requirements compared to geared wind turbines [21].

In wind turbines, rotor vibration contributes to main bearing failure, the rate of which can be as high as 30% over a 20-year wind turbine lifetime. Reliability of the main bearings is often considered one of the most important mechanical reliability challenges in wind turbines, after gearbox reliability [22].

Wind turbine driveshafts are subject to complex sources of transverse vibration. Although rotors spin at low-rated speeds ranging from about 8–15 rpm, there are many sources of higher harmonic transverse vibration excitation. As blades pass over the tower, their frequency causes a third harmonic excitation. Gearbox and generator unbalanced magnetic forces can generate excitations ranging from approximately 80–2000 times the rotor speed [23]. In this paper, the source of transverse vibration is a linearized unbalanced magnetic force corresponding to a generator with 200 poles [24,25].
Fluid film bearings offer several advantages over ball or roller bearings. Most importantly, when designed properly, they can have very low friction and nearly infinite life. Tolerance-related manufacturing issues due to increasing wind turbine size negatively impact the reliability of ball and roller bearing systems. Reliability tends to also be negatively affected by larger drive shaft diameters as the required number of rolling elements increases. While repairs for traditional main bearings often require some disassembly of the rotor, it is possible to design plain journal bearings or other fluid film bearings to avoid disassembly for repairs. Additionally, when designed properly, fluid film bearings can dampen vibration [26,27].

From a modeling standpoint, the speed-dependent properties of fluid film bearings make them more challenging to model. While this paper considers the modeling challenges of rotors supported by plain journal bearings, ball and roller bearings with lubrication face similar modeling challenges because they also have speed-dependent stiffness and damping properties [28].

3.1. System Concept

Figure 2 shows a schematic of the wind turbine and Table 1 lists the parameters. The wind turbine rotor model consists of three 99 m rigid blades mounted on a hub. We assume that the blades are rigid since the goal of the paper is to test the computational performance of the different shaft models while accounting for the large inertia effects of the rotor. Future work may involve modeling flexible blades in addition to the flexible shaft. The flexible shaft has a length of 10 m. Its front end has a diameter of 1.5 m that linearly decreases to 1.25 m at its rear end. Four fluid film bearings support the shaft at axial positions of 0, 5, 8, and 10 m from the front end. This model assumes that the bearing casings are stationary. A generator with 200 poles is located at the shaft midpoint. Unbalanced magnetic pull due to rotor sag causes a transverse force with a maximum amplitude of 20 kN.
Figure 3 shows the wind turbine simulation model developed in Simulink and Simscape Driveline. This model is available at [12]. Simscape blocks model the torque generated by wind acting on the rotor, the rotor and hub inertia, the shaft with transverse vibrations, the generator torque, and a disc brake. The blade torque block computes the wind turbine torque using a lookup table of power coefficient versus tip-speed ratio and blade pitch. Blade pitch is modeled quasi-statically, responding instantaneously to the pitch controller command.

The controllers in the model are based on the well-known National Renewable Energy Laboratory 5 MW Reference offshore wind turbine controller scaled up to 10 MW [29,30]. Generator torque and blade pitch controllers are modeled using Simulink blocks while a simple brake controller is modeled using Simscape blocks.

The generator controller uses shaft speed and wind speed as inputs. It has four main control regimes: a start-up region when no torque is applied, an optimized control region that tries to maintain an optimal tip-speed ratio for maximum power, a rated power region when the generator provides the rated power, and a cut-out region where the generator turns off during wind speeds that exceed 25 m/s.

When the wind turbine is in the rated power region, a blade pitch controller adjusts the collective blade pitch to maintain the rated turbine speed. The blade pitch controller uses scheduled proportional and integral gains to reduce the shaft speed error. When the
wind speed exceeds 25 m/s, the blades are feathered to 90 degrees and the disc brake is engaged to avoid overloading the generator.

Figure 4 shows the mechanical power, collective blade pitch, and shaft rotation speed in response to the simple wind speed scenario used in the case study of Section 3.3.

Table 1. 10 MW direct drive wind turbine parameters. These parameters are based on [24,31,32].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind turbine electric power rating</td>
<td>10 MW</td>
</tr>
<tr>
<td>Wind turbine power efficiency</td>
<td>94%</td>
</tr>
<tr>
<td>Wind turbine radius</td>
<td>99 m</td>
</tr>
<tr>
<td>Mass of blades, $m_{Blades}$</td>
<td>228 tons</td>
</tr>
<tr>
<td>Diametric moment of inertia of blades, $I_{D,Blades}$</td>
<td>$3.4 \times 10^7$ kg $m^2$</td>
</tr>
<tr>
<td>Polar moment of inertia of blades, $I_{P,Blades}$</td>
<td>$6.8 \times 10^7$ kg $m^2$</td>
</tr>
<tr>
<td>Mass of hub, $m_{Hub}$</td>
<td>106 tons</td>
</tr>
<tr>
<td>Diametric moment of inertia of hub, $I_{D,Hub}$</td>
<td>$9.29 \times 10^6$ kg $m^2$</td>
</tr>
<tr>
<td>Polar moment of inertia of hub, $I_{P,Hub}$</td>
<td>$1.68 \times 10^6$ kg $m^2$</td>
</tr>
<tr>
<td>Generator axal location, $z_{Generator}$</td>
<td>6 m</td>
</tr>
<tr>
<td>Mass of generator rotor, $m_{Generator}$</td>
<td>137 tons</td>
</tr>
<tr>
<td>Diametric moment of inertia of generator rotor, $I_{D,Generator}$</td>
<td>$8.5 \times 10^6$ kg $m^2$</td>
</tr>
<tr>
<td>Polar moment of inertia of generator rotor, $I_{P,Generator}$</td>
<td>$5.8 \times 10^6$ kg $m^2$</td>
</tr>
<tr>
<td>Rated shaft speed</td>
<td>8.6 RPM</td>
</tr>
<tr>
<td>Number of poles of generator, $p$</td>
<td>200</td>
</tr>
<tr>
<td>Generator maximum static unbalance magnitude, $f$</td>
<td>20 kN</td>
</tr>
<tr>
<td>Shaft length</td>
<td>10 m</td>
</tr>
<tr>
<td>Shaft maximum outer diameter</td>
<td>1.5 m</td>
</tr>
<tr>
<td>Shaft minimum outer diameter</td>
<td>1.25 m</td>
</tr>
<tr>
<td>Shaft maximum radial thickness</td>
<td>0.625 m</td>
</tr>
<tr>
<td>Shaft minimum radial thickness</td>
<td>0.375 m</td>
</tr>
<tr>
<td>Shaft density</td>
<td>7800 kg/m$^3$</td>
</tr>
<tr>
<td>Shaft elastic modulus</td>
<td>200 GPa</td>
</tr>
<tr>
<td>System damping proportional to mass, $\alpha$</td>
<td>10</td>
</tr>
<tr>
<td>System damping proportional to stiffness, $\beta$</td>
<td>$1 \times 10^{-9}$</td>
</tr>
<tr>
<td>Bearing axial locations, $z_{Support}$</td>
<td>[0, 5, 8, 10] m</td>
</tr>
<tr>
<td>Bearing fluid density</td>
<td>1077 kg/m$^3$</td>
</tr>
<tr>
<td>Bearing fluid viscosity</td>
<td>$3.2 \times 10^{-4}$ m$^2$/s</td>
</tr>
<tr>
<td>Bearing axial length</td>
<td>30 cm</td>
</tr>
<tr>
<td>Bearing radius</td>
<td>0.6875 m</td>
</tr>
<tr>
<td>Bearing clearance</td>
<td>0.4 mm</td>
</tr>
</tbody>
</table>

3.2. Rotordynamics Characteristics

Figure 5 shows the bearing properties versus rotation speed. These stiffness and damping properties represent a bearing with a radius of 0.6875 m, clearance of 0.4 mm, and Sommerfeld numbers over the relatively low range of $2 \times 10^{-3}$ to $2 \times 10^{-2}$. Low Sommerfeld numbers are typical for bearings with low speed and high radial load. When generating these bearing properties, we decrease the bearing damping compared to a typical journal bearing because of the current limitation that Simscape Driveline computes only undamped eigenmodes. Decreasing the support damping permits closer convergence between the lumped mass model and the speed-dependent eigenmode model. While computation of the mode shapes assumes undamped modes (Equations (11) and (19)), the modal properties account for support damping (Equations (17) and (24)). Future work will consider extending the model to account for complex mode shapes. As is typical for fluid film bearings, the vertical stiffness, subject to the shaft gravitational load, has the higher stiffness. For more detailed information, see [13].
Table 1. 10 MW direct drive wind turbine parameters. These parameters are based on [24,31,32].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind turbine electric power rating</td>
<td>10 MW</td>
</tr>
<tr>
<td>Wind turbine power efficiency</td>
<td>94%</td>
</tr>
<tr>
<td>Wind turbine radius</td>
<td>99 m</td>
</tr>
<tr>
<td>Mass of blades, $m_{\text{Blades}}$</td>
<td>228 tons</td>
</tr>
<tr>
<td>Diametric moment of inertia of blades, $I_{\text{D,Blades}}$</td>
<td>$3.4 \times 10^7$ kg·m²</td>
</tr>
<tr>
<td>Polar moment of inertia of blades, $I_{\text{P,Blades}}$</td>
<td>$6.8 \times 10^7$ kg·m²</td>
</tr>
<tr>
<td>Mass of hub, $m_{\text{Hub}}$</td>
<td>106 tons</td>
</tr>
<tr>
<td>Diametric moment of inertia of hub, $I_{\text{D,Hub}}$</td>
<td>$9.29 \times 10^6$ kg·m²</td>
</tr>
<tr>
<td>Polar moment of inertia of hub, $I_{\text{P,Hub}}$</td>
<td>$1.68 \times 10^6$ kg·m²</td>
</tr>
<tr>
<td>Generator axal location, $z_{\text{Generator}}$</td>
<td>6 m</td>
</tr>
<tr>
<td>Mass of generator rotor, $m_{\text{Generator}}$</td>
<td>137 tons</td>
</tr>
<tr>
<td>Diametric moment of inertia of generator rotor, $I_{\text{D,Generator}}$</td>
<td>$8.5 \times 10^6$ kg·m²</td>
</tr>
<tr>
<td>Polar moment of inertia of generator rotor, $I_{\text{P,Generator}}$</td>
<td>$5.8 \times 10^6$ kg·m²</td>
</tr>
<tr>
<td>Rated shaft speed</td>
<td>8.6 RPM</td>
</tr>
<tr>
<td>Number of poles of generator</td>
<td>200</td>
</tr>
<tr>
<td>Generator maximum static unbalance magnitude, $f_{\text{Generator}}$</td>
<td>20 kN</td>
</tr>
<tr>
<td>Shaft length</td>
<td>10 m</td>
</tr>
<tr>
<td>Shaft maximum outer diameter</td>
<td>1.5 m</td>
</tr>
<tr>
<td>Shaft minimum outer diameter</td>
<td>1.25 m</td>
</tr>
<tr>
<td>Shaft maximum radial thickness</td>
<td>0.625 m</td>
</tr>
<tr>
<td>Shaft minimum radial thickness</td>
<td>0.375 m</td>
</tr>
<tr>
<td>Shaft density</td>
<td>7800 kg/m³</td>
</tr>
<tr>
<td>Shaft elastic modulus</td>
<td>200 GPa</td>
</tr>
<tr>
<td>System damping proportional to mass, $\alpha$</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 4. Wind speed and shaft speed scenario used for the case study in Section 3.3.

Figure 6 shows the Campbell diagram for the 20 lowest eigenmodes of the system generated using the speed-dependent eigenmode reduced model. To avoid static unbalance instabilities, none of the whirling frequencies match the shaft speed. Several of the modes pass through the 200th harmonic of the shaft rotation speed, which corresponds to the generator magnetic unbalance. These include modes 1 and 2 at a shaft speed of 0.5 RPM, mode 4 at a shaft speed of 5.75 RPM, and mode 5 at a shaft speed of 6.5 RPM. Therefore, to avoid excess vibration in this system, it is best for the controller to quickly pass through these shaft rotation speeds. The frequencies of several modes cross each other when the shaft speed passes through 2 RPM. To produce an accurate response prediction, the Simscape Driveline algorithm for sorting modes during compile time that is described in Section 2.3 can capture this behavior.
Figure 5. Speed-dependent properties of the fluid film bearings. Figure 7 shows the shapes of the eight lowest frequency eigenmodes at the rated shaft speed of 8.6 RPM. The mode shapes do not significantly change with rotation speed even though some mode frequencies do because the bearings are relatively stiff. Modes 1, 3, 5, and 6 have large amplitude in the X direction and small amplitudes in the Y direction, while modes 2, 4, 7, and 8 have large amplitudes in the Y direction and small amplitudes in the X direction. These modes are relatively planar because the bearing cross-coupling stiffnesses are relatively small.
Figure 6. Campbell diagram. Several modes show frequency-crossing events. The black dashed line corresponds to 200 p excitation. This figure was generated using the speed-dependent eigenmode method described in Section 2.3. This figure shows how the algorithm captured multiple frequencies crossing each other.

3.3. Case Study Scenario

The case study uses the wind speed scenario shown in Figure 4. The scenario consists of the wind ramping up from 0 m/s to 20 m/s over 30 s, holding steady at 20 m/s for 30 s, ramping back down to 0 m/s over 30 s, and then remaining at 0 m/s for the final 10 s. In response to this wind speed, the wind turbine shaft accelerates from rest until it reaches a peak speed of 10.9 RPM and then gradually decelerates to maintain a constant power.

To fully exercise the lookup-table-based algorithm, we choose this scenario, wherein the simulation passes through a large range of shaft speeds. To ensure that the measured simulation execution times are long enough that compilation start-up effects are not significant, and that data can still be gathered relatively quickly, we choose a simulated time of 100 s. A more typical wind turbine simulation might span an hour or more of simulated time and have more wind speed and shaft speed fluctuations. Additionally, as mentioned in the introduction, a typical wind turbine simulation will account for different sources of shaft bending excitation with different harmonics. This simplified scenario permits the simulation performance to be dominated by the vibration model performance.

3.4. Numeric Methodology

The simulation data are from an Intel® Xeon® W-2133 CPU @ 3.60GHz x64-based processor. For each modeling method and chosen number of DOFs, a MATLAB script runs the simulation multiple times and averages the execution time.

The Simulink model uses the ode23t solver with an absolute tolerance of $1 \times 10^{-3}$. This model uses variable nominal value scaling to improve the simulation speed for each modeling method. Nominal values scale the variables seen by the solver. When the solver sees variables with scaled magnitudes on the order of 1, the simulation can have improved
robustness and execution speed. We use the Simscape Results Explorer and Simscape Variable Scaling Analyzer to determine the magnitudes of each shaft DOF and its time-derivative. For each modeling method, we set all of the shaft DOFs to have nominal values equal to the peak value of the DOF with the largest observed response.

Figure 7. Eight lowest eigenmodes at a shaft speed of 8.6 RPM. The x- and y-axes have different scales depending on the nature of each mode shape.
4. Results and Discussion

This section compares the performance of the four different methods for simulating the wind turbine shaft transverse vibrations. These results correspond to the wind speed and shaft speed scenario shown in Figure 4. Figure 8 plots the percent error in the maximum horizontal and vertical forces acting on the second bearing versus the run/sim time ratio. The run/sim time ratio is the ratio of the execution time to the simulated time of 100 s. Figure 8 is based on simulations with transient shaft speeds. Figure 9 plots a snapshot of the transient time series for each simulation method. The performance trends for the chosen variables are representative of the performance trends for other variables.

Figure 8. Comparison of the % error and run/sim time ratios of the different simulation modeling methods. For the % error, the reference results are the forces computed by the lumped mass model simulation with 200 DOFs (50 nodes). The reference results are 2858 N for the horizontal bearing force and 3175 N for the vertical bearing. For the run/sim time ratio, the simulated time is 100 s. For each curve, the markers represent different DOFs: 4 × (number of lumped elements) or number of modes. The circles indicate the most computationally efficient simulation that converges within 2% of the reference result. Increasing the number of DOFs generally increases the simulation accuracy (as shown by curves converging to an error of 0) while increasing the run time.
As the most computationally efficient simulation, it does not converge to an accurate result, regardless of how many eigenmodes are added. The static eigenmode properties are based on a nominal shaft speed of 4.3 RPM, so the response of the model is inaccurate at the instants of peak bearing force, which occur at shaft speeds greater than 7 RPM. We expect accuracy errors for the static eigenmode method because the simulation has varying shaft speeds and the model properties are sensitive to the shaft speed.
Therefore, these static eigenmode results serve as a lower-bound reference on the run/sim time ratio of the other methods rather than providing a meaningful % error comparison.

The Craig–Bampton method generally requires more DOFs than the speed-dependent eigenmode method because the Craig–Bampton method requires four static modes for each external load and support interface in addition to the inner modes that capture flexible dynamics. Since the system in this case study has four support nodes and one forced node, the Craig–Bampton system requires 20 static modes (at least 20 DOFs). The eigenmode approaches capture external load, support, and flexibility dynamics all within each mode. The Craig–Bampton static modes, which are coupled with the inner modes, tend to have different stiffnesses than the inner modes, and this makes it a more challenging problem for the numeric solver. An advantage of the Craig–Bampton approach for this system is that it requires fewer inner modes to converge to the accurate solution. More Craig–Bampton inner modes are required when a shaft is more flexible.

The speed-dependent eigenmode method requires a sufficient number of modes to converge. Some of the modes have natural frequencies that are much higher than the excitation frequency, but the small excitation of these modes still has a non-negligible effect on the force since higher modes with higher bending tend to impart higher forces on the bearings.

For all modeling methods, the vertical force requires more DOFs than the horizontal force to converge to the reference value. This is related to how the system has higher stiffness in the vertical direction.

Previous practical rotating shaft model order reduction methods have focused on component mode synthesis approaches such as the Craig–Bampton method. The hypothesis of this study was that the additional cost of lookup table interpolation would be more than offset by the computational efficiency of the new method because it generally requires fewer DOFs (none of which are coupled) than the Craig–Bampton method. The results of the wind turbine driveline case study confirm this hypothesis. As mentioned in the introduction, detailed and computationally efficient rotor dynamics models can accelerate and strengthen conceptual design exploration, parameter optimization, and system design verification and validation. Detailed, computationally efficient models can also enable real-time simulation for controller hardware testing and predictive maintenance, which can reduce the risk of failures and improve machine performance. These computational benefits become more significant when shaft models are based on higher-cost finite element analysis approaches, and simulations require longer simulated scenarios.

In addition to more computationally efficient simulation, the uncoupled DOFs of the speed-dependent eigenmode method make it easier to fundamentally understand the system behavior and make design decisions.

The speed-dependent eigenmode approach has a few limitations. The method considers the entire shaft-bearing-disk system when computing the eigenmodes. This approach is more restrictive than the Craig–Bampton method for interfacing with other machine subcomponents. Additionally, the speed-dependent eigenmode method in Simscape Driveline currently restricts the bearing casing to be stationary. A limitation of this study is that the eigenmode methods consider only undamped vibration mode shapes. In order to ensure that the eigenmode results could sufficiently converge with the reference lumped mass method, we included only light damping in the system. Future extensions could loosen these restrictions to permit additional external excitations and parameters of the system.

In the future, this approach could be expanded to complex modes and additional gradually changing variables and systems. Unlike the Craig–Bampton approach, the speed-dependent eigenmode approach is well-suited for capturing the behavior of changing body characteristics in addition to changing boundary characteristics.
5. Conclusions

In this paper, we show that a speed-dependent eigenmode method is capable of accurately simulating the transient transverse vibrations of a wind turbine driveline with a run to simulation time ratio of 0.14 for excitation frequencies that range from 0 Hz to 28 Hz. The speed-dependent eigenmode method simulates 2.4 times faster than the Craig–Bampton method. A limitation of this Craig–Bampton method performance is that it requires four DOFs for every interface. A limitation of the speed-dependent eigenmode method performance is that it requires lookup table computations for the varying modal properties during every simulation time step. Accordingly, we observe that the speed-dependent eigenmode method computational performance is generally better than that of the Craig–Bampton method when characterizing the system behavior requires fewer eigenmodes and more Craig–Bampton interface nodes for an accurate simulation. For future work in this space, we will consider extending the method to model complex modes and more detailed finite element analysis models in the Simscape Driveline simulation model. The authors encourage readers to adopt the model at [12] for their own studies and contact J.K. with any questions.

Author Contributions: Conceptualization, J.K., L.C. and M.U.; methodology, J.K. and M.U.; software, J.K.; validation, J.K.; formal analysis, J.K.; investigation, J.K.; writing—original draft preparation, J.K.; writing—review and editing, J.K., L.C. and M.U.; supervision, M.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MathWorks.

Data Availability Statement: The publicly available Simulink model used to generate the data in this study can be found here: https://www.mathworks.com/matlabcentral/fileexchange/120033-wind-turbine-driveline-with-vibrations-in-simscape (accessed on 15 October 2022) [12].

Acknowledgments: We would like to thank Steve Miller for useful discussions on how to clearly present the information and the MathWorks Advanced Research & Technology Office for their support.

Conflicts of Interest: Jocelyn Kluger and Martin Udengaard are employed by MathWorks. Lynn Crevier was formerly employed by MathWorks. All three authors developed the flexible shaft feature that is part of the Simscape Driveline product.

Appendix A

Table A1 provides additional details on the performance metrics of the different simulation methods. These results correspond to the results shown in Figure 8.
Table A1. Performance metrics of the different modeling methods for computing the maximum horizontal force (%ErrX) and vertical force (%ErrY) in the second bearing. # DOF is the number of degrees of freedom in the models: four times the number of lumped elements for the lumped mass model and number of modes for the other three models. $T_{\text{Exec}}$ is the execution time in seconds for a 100 s simulation time. %ErrX and %ErrY are the errors of the predicted forces relative to the lumped mass model simulation with 200 DOFs (50 nodes), the reference result. The bold highlighted results are the simulation for each method with the highest computational efficiency that converges within 2% of the reference result. For further reference, the execution time is 0.08 s when shaft flexibility is disabled.

<table>
<thead>
<tr>
<th># DOF</th>
<th>$T_{\text{Exec}}$</th>
<th>%ErrX</th>
<th>%ErrY</th>
<th># DOF</th>
<th>$T_{\text{Exec}}$</th>
<th>%ErrX</th>
<th>%ErrY</th>
<th># DOF</th>
<th>$T_{\text{Exec}}$</th>
<th>%ErrX</th>
<th>%ErrY</th>
<th># DOF</th>
<th>$T_{\text{Exec}}$</th>
<th>%ErrX</th>
<th>%ErrY</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>45.6</td>
<td>0.4</td>
<td>−3.0</td>
<td>8</td>
<td>4.4</td>
<td>−20.6</td>
<td>−8.2</td>
<td>8</td>
<td>6.2</td>
<td>15.8</td>
<td>−6.9</td>
<td>20</td>
<td>32.7</td>
<td>−0.0003</td>
<td>−0.13</td>
</tr>
<tr>
<td>80</td>
<td>69.2</td>
<td>0.1</td>
<td>−0.5</td>
<td>12</td>
<td>5.0</td>
<td>−14.7</td>
<td>−7.3</td>
<td>12</td>
<td>8.6</td>
<td>9.1</td>
<td>−0.04</td>
<td>24</td>
<td>35.9</td>
<td>−0.04</td>
<td>−0.07</td>
</tr>
<tr>
<td>120</td>
<td>130.2</td>
<td>0.05</td>
<td>−0.1</td>
<td>16</td>
<td>5.4</td>
<td>−15.7</td>
<td>−8.2</td>
<td>16</td>
<td>10.2</td>
<td>3.9</td>
<td>−0.7</td>
<td>28</td>
<td>39.4</td>
<td>0.01</td>
<td>−0.21</td>
</tr>
<tr>
<td>160</td>
<td>216.2</td>
<td>0.002</td>
<td>−0.2</td>
<td>20</td>
<td>6.1</td>
<td>−16.2</td>
<td>−9.1</td>
<td>20</td>
<td>13.7</td>
<td>1.36</td>
<td>−0.3</td>
<td>32</td>
<td>51.8</td>
<td>−0.005</td>
<td>0.08</td>
</tr>
<tr>
<td>200</td>
<td>341.2</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>7.0</td>
<td>−16.4</td>
<td>−9.5</td>
<td>24</td>
<td>19.1</td>
<td>0.41</td>
<td>−0.2</td>
<td>36</td>
<td>55.1</td>
<td>−0.0006</td>
<td>0.06</td>
</tr>
<tr>
<td>32</td>
<td>8.9</td>
<td>−16.4</td>
<td>−9.8</td>
<td>32</td>
<td>28.1</td>
<td>0.20</td>
<td>−0.4</td>
<td>44</td>
<td>63.9</td>
<td>0.01</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>11.3</td>
<td>−16.4</td>
<td>−9.8</td>
<td>40</td>
<td>40.0</td>
<td>0.17</td>
<td>−0.5</td>
<td>52</td>
<td>74.3</td>
<td>−0.006</td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>14.4</td>
<td>−16.4</td>
<td>−9.8</td>
<td>48</td>
<td>53.1</td>
<td>0.10</td>
<td>−0.06</td>
<td>60</td>
<td>87.5</td>
<td>0.007</td>
<td>0.07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>17.9</td>
<td>−16.4</td>
<td>−9.8</td>
<td>56</td>
<td>63.3</td>
<td>0.08</td>
<td>−0.05</td>
<td>68</td>
<td>126.9</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>22.5</td>
<td>−16.4</td>
<td>−9.8</td>
<td>64</td>
<td>85.2</td>
<td>0.06</td>
<td>−0.05</td>
<td>68</td>
<td>126.9</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
