User's guide to MBC3: Multi-blade coordinate transformation code for 3-bladed wind turbine

G. S. Bir
National Renewable Energy Laboratory

Follow this and additional works at: https://digitalscholarship.unlv.edu/renew_pubs
Part of the Mechanical Engineering Commons, Oil, Gas, and Energy Commons, and the Power and Energy Commons

Repository Citation
Available at: https://digitalscholarship.unlv.edu/renew_pubs/57

This Technical Report is brought to you for free and open access by the Energy at Digital Scholarship@UNLV. It has been accepted for inclusion in Publications (E) by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

G.S. Bir

G.S. Bir

Prepared under Task No. WER9.3535
About This Guide

The dynamics of wind turbine rotor blades are conventionally expressed in rotating frames attached to the individual blades. The tower-nacelle subsystem though, sees the combined effect of all rotor blades, not the individual blades. Also, the rotor responds as a whole to excitations such as aerodynamic gusts, control inputs, and tower-nacelle motion—all of which occur in a nonrotating frame. Multi-blade coordinate transformation (MBC) helps integrate the dynamics of individual blades and express them in a fixed (nonrotating) frame. MBC involves two steps: transforming the rotating degrees of freedom and transforming the equations of motion. Reference 1 details the MBC operation. This guide summarizes the MBC concept and underlying transformations.

This guide also explains how to use MBC3, a MATLAB-based script we developed to perform multi-blade coordinate transformation of system matrices for three-bladed wind turbines. In its current form, MBC3 can be applied to system matrices generated by FAST.2 The system matrices can have arbitrary—but physically consistent—sizes. These matrices can also optionally include control, disturbance, output, and feed-forward matrices. In addition, the associated state, control, and disturbance vectors can have states that are partly referred to the fixed frame and partly to the rotating frame. A key feature of MBC3 is that it can handle variable-speed operation and turbines with dissimilar blades. Depending on the analysis objective, users can generate system matrices in either the first-order (state-space) form or the second-order (physical-domain) form. MBC3 can handle both the first- and second-order system matrices. These matrices in general govern the aero-hydro-servo-elastic behavior of a wind turbine. After multi-blade coordinate transformation, MBC3 also performs eigenanalysis, and the results are presented in different formats to aid modal and stability analysis. MBC3 automatically detects whether optional matrices (e.g., control, disturbance, output, and feed-forward matrices) are present and performs multi-blade coordinate transformation accordingly.

Multi-blade coordinate transformation has several applications,3 but it is essential to modal and stability analyses. Commonly, wind turbine researchers first compute the periodic state-space matrix, time average it over the rotor rotational period, and then apply conventional eigenanalysis to compute modal and stability characteristics. Direct averaging, however, eliminates all periodic terms that contribute to system dynamics, producing errors. Although averaging itself is not always a bad approach, it must follow MBC. Reference 1 contains sample results that illustrate this point. The main advantages of MBC are capturing cumulative dynamics of the rotor blades and interaction of the rotor with the tower-nacelle subsystem, well-conditioning the system matrices by eliminating nonessential periodicity, and performing filtering operation.1

This guide gives step-by-step instructions on how to load system matrices from FAST to MATLAB, how to perform multi-blade coordinate transformation on these matrices, and how to interpret the transformed matrices. The guide also describes the inputs to MBC3. The script runs quickly (usually in a few seconds). In the future, we will release two versions of MBC3, one written in MATLAB and the other in Fortran, with clearly
defined interfaces. This will allow users to directly integrate MBC3 with FAST or any other aeroelastic code.

**Acknowledgments**

Thanks to Sandy Butterfield for motivating the development of MBC3. Thanks also to Bonnie Jonkman for her help with writing the code certification script and to Marshall Buhl for his help with uploading the MBC files to our Web site. Thanks to Jason Jonkman for providing the GetMats.m script that loads FAST output data into the MATLAB workspace. DOE supported this work under contract number DE-AC36-08-GO28308.
## Document Revision Record

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Oct. 13, 2008</td>
<td>First publication. Supports MBC3 v1.00.00.</td>
</tr>
<tr>
<td>1.10</td>
<td>May 18, 2008</td>
<td>Second publication. Supports MBC3 v1.01.00, which includes control-specific upgrades and generates descriptions of MBC-transformed states.</td>
</tr>
</tbody>
</table>

# Table of Contents

1. Introduction..........................................................................................................................1
2. MBC Concept and Associated Transformation.................................................................3
3. Transformation of System Matrices......................................................................................5
   3.1 Second-Order System Matrices ...................................................................................5
   3.2 First-Order System Matrices........................................................................................10
4. Input Description ................................................................................................................13
5. Executing MBC3 ..................................................................................................................16
6. Output Description .............................................................................................................18
7. Downloading and Verifying MBC-Specific Files ...............................................................22
8. Future Plans .......................................................................................................................23
References...............................................................................................................................24
1. Introduction

The dynamics of wind turbine rotor blades are generally expressed in rotating frames attached to the individual blades. The tower-nacelle subsystem sees the combined effect of all rotor blades, not the individual blades. Also, the rotor responds as a whole to excitations such as aerodynamic gusts, control inputs, and tower-nacelle motion—all of which occur in a nonrotating frame. Multi-blade coordinate transformation (MBC) helps integrate the dynamics of individual blades and express it in a fixed (nonrotating) frame.

MBC offers several benefits. First, it properly models the dynamic interaction between the nonrotating tower-nacelle and the spinning rotor. Second, it offers physical insight into rotor dynamics and how the rotor interacts with fixed-system entities such as wind, controls, and the tower-nacelle subsystem. Third, MBC filters out all periodic terms except those that are integral multiples of $\Omega N$, where $\Omega$ is the rotor angular speed and $N$ is the number of rotor blades. A wind turbine system is basically a periodic system; equations governing its dynamics show periodic parameters that arise because of the periodic interaction between the rotating subsystem (rotor) and the nonrotating entities (tower, nacelle, wind, controls, and gravity). The blade equations usually contain all harmonics. MBC behaves as a filter, allowing the fixed system to feel only specific harmonics of blade motion. This filtering action also renders the system equations numerically well-conditioned, retaining only those periodic terms that are actually seen in a fixed reference frame.

MBC is widely used in the helicopter field. Miller used it to analyze flap-motion related stability and control. Coleman and Feingold used it to analyze rotor in-plane motion (lag motion); theirs was the first successful attempt to understand the helicopter ground resonance problem, which had eluded earlier researchers. These efforts, however, applied MBC only in a heuristic fashion. Later, Johnson delineated a systematic mathematical basis and presented a thorough exposition of MBC. Using this mathematical basis, Bir et al. developed a numerical MBC approach that could be applied to a general helicopter system governed by arbitrary degrees of freedom; they used the approach to analyze the stability and response of several helicopters.

Because MBC offers so many benefits, it is receiving attention in the wind turbine field. Bir, Wright, and Butterfield included MBC in a stability analysis scheme and predicted an instability caused by coalescence between the rotor in-plane and the tower motions. They also showed how MBC gives physical insight into the rotor in-plane motion. In the turbine field, Malcolm appears to be the first to derive a mathematical form of the turbine equations after applying the MBC. His prime motivation was to relate the inflow characteristics with the turbine response and to extract linearized models from general aeroelastic codes such as ADAMS. McCoy extended this effort to obtain linear time-invariant system equations required in the standard control design approaches. Hansen used MBC for improved modal dynamics to avoid stall-induced vibrations and later combined it with an eigenvalue approach to study the aeroelastic stability characteristics of a three-bladed turbine. Riziotis et al. applied MBC to analyze stability of two three-bladed turbines: one stall regulated and the other pitch regulated. Bir and Jonkman
used MBC in conjunction with FAST to study aeroelastic characteristics of a 5 MW turbine in both land-based and offshore configurations.¹⁵

All attempts at MBC thus far, in both the helicopter and the wind turbine fields, assume that the rotor speed is constant and the rotor blades are similar. A modern wind turbine is rarely constant speed. Also, turbines may not have identical blades, structurally or aerodynamically. We need an MBC approach that overcomes these limitations.

This report describes a new MBC scheme that is applicable to a variable-speed turbine, which may also have dissimilar blades. The scheme also covers control, disturbance, output, and feed-forward matrices, which have been ignored to date. Depending on the analysis objective, wind turbine researchers may generate system matrices in either the first-order (state-space) form or the second-order (physical-domain) form. We develop MBC relations for both these forms. In the literature, MBC is also referred to as the Fourier coordinate transformation (FCT) and as the Coleman transformation.

Although MBC has several applications, it is essential to and mostly used for modal and stability analyses and state-space-based controls design. Commonly, wind turbine researchers first compute the periodic state-space matrix, time average it over the rotor rotational period, and then apply conventional eigenanalysis to compute modal and stability characteristics. Averaging, however, eliminates all periodic terms that contribute to system dynamics and can yield erroneous results. Although averaging itself is not a bad approach, it must follow MBC.¹ Theoretically, a Floquet analysis—not averaging—must follow MBC. In normal cases, however, MBC eliminates the dominant periodic terms; the remaining periodic terms (\(N_b/\text{rev}, 2N_b/\text{rev}, \ldots\), etc., for an \(N_b\)-bladed turbine) usually have a negligible magnitude and, therefore, direct averaging of MBC-transformed matrices suffices.¹¹

In this guide, Section 2 summarizes the MBC concept and how it relates the blade coordinates to rotor coordinates. Section 3 shows how we transform the system matrices from rotating to nonrotating (fixed) coordinates, which is the objective of MBC. We develop transformation relations for both first- and second-order matrices. Section 4 describes the inputs required by MBC3, and Section 5 explains how MBC3 is executed. Section 6 describes the MBC3-generated outputs; these include MBC-transformed system matrices and eigenanalysis results. Section 7 lists the MATLAB scripts required for MBC; we also give our Web site for access to these scripts, along with sample I/O files. Section 8 summarizes future plans.
2. MBC Concept and Associated Transformation

The turbine equations (i.e., the coupled tower-nacelle-rotor equations) are generally derived using mixed degrees of freedom. Some degrees of freedom may be in the rotating frame and others in the nonrotating frame. This can be desirable. For example, in some simulation studies, we might be interested in studying the tower response in the ground-fixed (nonrotating) frame and the blades’ response in their respective rotating frames. Often, however, we are interested in understanding the coupled behavior of the tower-nacelle-rotor system. In such cases, we want to express the full system behavior in a fixed frame. MBC helps us achieve this through a rotating-frame to nonrotating-frame coordinate transformation.

Consider a rotor with $N_b$ blades spaced equally around the rotor azimuth. The azimuth location of $b$th blade is given by

$$\psi_b = \psi + (b-1)\frac{2\pi}{N_b}$$  \hspace{1cm} (1)

where $\Psi$ is the azimuth of the first (reference) blade (see Figure 1 in which $N_b = 3$). We assume that $\psi = 0$ implies the first blade is vertically up.

Let $q_b$ be a particular rotating degree of freedom for the $b$th blade. The MBC is a linear transformation that relates the rotating degrees of freedom to new degrees of freedom, defined as follows:\(^3\):

$$q_0 = \frac{1}{N_b} \sum_{b=1}^{N_b} q_b$$

$$q_{nc} = \frac{2}{N_b} \sum_{b=1}^{N_b} q_b \cos n\psi_b$$

$$q_{ns} = \frac{2}{N_b} \sum_{b=1}^{N_b} q_b \sin n\psi_b$$

$$q_{N/2} = \frac{1}{N_b} \sum_{b=1}^{N_b} q_b (-1)^b$$  \hspace{1cm} (2)

**Figure 1. Rotor reference frame**

Mathematically, the new degrees of freedom are the projections of the blade degrees of freedom onto the nonrotating (fixed) frame, which is the frame ($X_{NR}$, $Y_{NR}$, $Z_{NR}$) shown in Figure 1. The axis $X_{NR}$ is directed along the shaft axis, and $Z_{NR}$ lies in the vertical plane. In the literature, $q_0, q_{nc}, q_{ns}, q_{N/2}$ are called nonrotating degrees of freedom; we call these rotor coordinates because they express the cumulative behavior of all rotor blades (not individual blades) in the fixed frame. The physical interpretation of each rotor coordinate depends on the degree of freedom to which it refers. For example, if $q_b$ is a flap degree of
freedom, then \( q_0 \) is the rotor coning, \( q_{1c} \) is the rotor tip-path-plane fore-aft tilt about an axis that is horizontal and normal to the rotor shaft, and \( q_{1s} \) is the rotor tip-path-plane side-side tilt about an axis that is vertical and normal to the rotor shaft. If \( q_b \) is a lag degree of freedom, then \( q_0 \) is the rotor collective lag, \( q_{1c} \) is the horizontal displacement of the rotor center-of-mass in the rotor plane, and \( q_{1s} \) is the vertical displacement of the rotor center-of-mass in the rotor plane. Other degrees of freedom can be interpreted using Equation 2 and drawing supporting sketches. The rotor modes corresponding to \( q_{nc} \) and \( q_{ns} \) \((n > 1)\) and \( q_{Nh/2} \) are called reactionless modes because they do not cause any transference of moments or forces from the rotor to the hub (fixed frame). The value of \( n \) goes from 1 to \((N_b - 1)/2\) if \( N_b \) is odd, and from 1 to \((N_b - 2)/2\) if \( N_b \) is even. The \( q_{Nh/2} \) mode, called the differential mode, exists only if the number of blades is even.

Most of the wind turbines in the world have three blades. Therefore, we assume \( N_b = 3 \) in this report. The formulation we developed here for \( N_b = 3 \), however, is general and can be readily extended for more blades. Setting \( N_b = 3 \) in Equation 2, we obtain

\[
q_0 = \frac{1}{3} \sum_{b=1}^{3} q_b \\
q_c = q_{1c} = \frac{2}{3} \sum_{b=1}^{3} q_b \cos \psi_b \\
q_s = q_{1s} = \frac{2}{3} \sum_{b=1}^{3} q_b \sin \psi_b
\]  

(3)

We call \( q_{1c} \) the cosine-cyclic mode and \( q_{1s} \) the sine-cyclic mode. These two cyclic modes, together with the coning mode, \( q_0 \), lead to coupling of the rotor with the rest of the turbine. Equation 3 determines the rotor coordinates given the blade coordinates. The inverse transformation, which yields the blade coordinate given the rotor coordinates, is

\[
q_b = q_0 + q_c \cos \psi_b + q_s \sin \psi_b; \quad b = 1, 2, 3.
\]  

(4)
3. Transformation of System Matrices

Most aeroelastic codes generate linear equations in the second-order form and, if required (for controls, for example), transform these to a first-order form. We derive MBC relations for both the second- and first-order systems.

3.1 Second-Order System Matrices

Second-order system equations can be written as

\[ M \ddot{X} + C \dot{X} + KX = Fu + F_d w \]  \hspace{1cm} (5a)

and the associated output equation as

\[ Y = C_x \ddot{X} + C_x \dot{X} + Du + D_x w \]  \hspace{1cm} (5b)

where \( M, C, K, F, \) and \( F_d \) are the mass, damping, stiffness, control, and disturbance matrices, respectively. The \( u \) and \( w \) are, respectively, the control and disturbance vectors. The size of the \( M, C, \) and \( K \) matrices is \( n \times n \), where \( n \) is the number of system degrees of freedom. The sizes of the \( u, w, \) and \( Y \) vectors are \( n_c, n_w, \) and \( n_o \), respectively, where \( n_c \) is the number of control variables, \( n_w \) is the number of disturbance states, and \( n_o \) is the number of outputs. Note that the \( C \) matrix contains gyroscopic terms in addition to the structural damping terms, even though we designate it as a damping matrix. Similarly, \( K \) contains centrifugal terms in addition to the structural and aerodynamic stiffness terms. Also, each of these matrices contains direct and cross-coupling terms. For example, \( M \) contains direct blade inertias as well as blade-tower coupling inertias. The formulation we develop does not, however, require the user to explicitly partition any of the matrices into submatrices delineating direct and cross-coupling terms. The partitioning is implicit in the definition of the coordinates vector \( X \) (also called the physical vector in the literature). Assume that \( X \) is expressed as

\[
X = \begin{bmatrix}
X_F \\
q_1^1 \\
q_2 \\
q_3 \\
\vdots \\
q_i \\
q_2^l \\
q_3^l \\
\vdots \\
q_i^m \\
q_2^m \\
q_3^m
\end{bmatrix}
\]  \hspace{1cm} (6)
where $X_F$ is a $nF \times 1$ column vector representing the $nF$ fixed-frame-referenced degrees of freedom, $q^{(j)}_i$ is the $j$th rotating degree of freedom for the $b$th blade, and $m$ is the number of rotating degrees of freedom for each blade. The length $n$ of vector $X$, then equals $nF + 3m$, the total number of degrees of freedom for the full system. Most often, aeroelastic codes such as FAST assume that the physical vector has the form of Equation 6. If not, we can always perform simple rows and columns exchange to transform system equations (Equation 5) such that the physical vector $X$ assumes the form of Equation 6.

Two methods transform Equation 5 to the nonrotating frame: the operational method and the substitution method. In the operational method, we apply the following summation operators to the rotating-frame equations of motion:

$$\frac{1}{3} \sum_{b=1}^{3} (rot\_eqns)$$

$$\frac{2}{3} \sum_{b=1}^{3} (rot\_eqns) \cos \psi_b$$

$$\frac{2}{3} \sum_{b=1}^{3} (rot\_eqns) \sin \psi_b$$

where $rot\_eqns$ refers to the rotating-frame equations of motion. Note the similarity of summation operators in Equations 3 and 7. To accomplish these operations, we first need the $rot\_eqns$ to be available in an analytical form, explicitly showing all periodic terms. The periodic terms involve trigonometric functions of $\sin(k \psi_b)$ and $\cos(k \psi_b)$ terms, where $k$ in general can have any integer value, depending on the system. First, we must express all products of trigonometric functions as sums of trigonometric functions. Next, we must perform cumbersome operations to ensure that the value of each $n$ in these trigonometric functions satisfies the requirements of Equation 3. Finally, we use transformation (Equation 3) to convert the $rot\_eqns$ to the nonrotating frame. The operational method is thus quite tedious and applicable only if we have equations of motion available in an analytical form.

All aeroelastic codes generate equations of motion in a numerical form and are therefore not amenable to the operational method. We must instead use the substitutional method. In this method, we substitute the rotational degrees of freedom with the rotor coordinate using Equation 4. If we do this, the three rotational degrees of freedom $(q^1_j, q^2_j, q^3_j)$, corresponding to the $j$th degree of freedom for each of the three blades, can be transformed to nonrotating coordinates $(q^0_j, q^c_j, q^s_j)$ using the following relation:
\[
\begin{align*}
\begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix} &= \tilde{t} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \\
\end{align*}
\tag{8}
\]

where
\[
\tilde{t} = \begin{bmatrix} 1 & \cos \psi_1 & \sin \psi_1 \\ 1 & \cos \psi_2 & \sin \psi_2 \\ 1 & \cos \psi_3 & \sin \psi_3 \end{bmatrix}
\tag{9}
\]

Using Equation 8, the full-system, rotating-frame, degrees-of-freedom vector, \(X\), is expressed in terms of the nonrotating-frame, degrees-of-freedom vector, \(X_{NR}\), as follows
\[
X = T_1 X_{NR}
\tag{10}
\]

where \(T_1\) is the block diagonal matrix:
\[
T_1 = \begin{bmatrix}
I_{nF \times nF} & & \\
& \tilde{t} & \\
& & \tilde{t} \\
& & & \cdots \tilde{t} \\
& & & & I_{(nF+3m) \times (nF+3m)}
\end{bmatrix}
\tag{11}
\]

and
\[
X_{NR} = \begin{bmatrix}
X_F \\
q_0 \\
q_c^1 \\
q_s^1 \\
\vdots \\
q_0^j \\
q_c^j \\
q_s^j \\
\vdots \\
q_0^m \\
q_c^m \\
q_s^m
\end{bmatrix}
\]  

Taking the first and second time derivatives of the two sides of Equation \(10\), we obtain

\[
\begin{align*}
\ddot{X} &= T_1 \dot{X}_{NR} + \Omega T_2 \dot{X}_{NR} \\
\ddot{X} &= T_1 \ddot{X}_{NR} + 2\Omega T_2 \dot{X}_{NR} + (\Omega^2 T_3 + \Omega \dot{T}_2) X_{NR}
\end{align*}
\]  

where \(\Omega\) is the rotor angular velocity and \(\dot{\Omega}\) is the rotor angular acceleration. The \(T_2\) and \(T_3\) transformation matrices are given by

\[
T_2 = \begin{bmatrix}
0_{nF \times nF} & \tilde{t}_2 \\
\vdots & \vdots \\
\tilde{t}_2 & \tilde{t}_2 \\
& \vdots \\
& \tilde{t}_2_{(nF+3m)\times(nF+3m)}
\end{bmatrix}
\]  

\[
T_3 = \begin{bmatrix}
0_{nF \times nF} & \tilde{t}_3 \\
\vdots & \vdots \\
\tilde{t}_3 & \tilde{t}_3 \\
& \vdots \\
& \tilde{t}_3_{(nF+3m)\times(nF+3m)}
\end{bmatrix}
\]
where
\[
\tilde{t}_2 = \begin{bmatrix}
0 & -\sin \psi_1 & \cos \psi_1 \\
0 & -\sin \psi_2 & \cos \psi_2 \\
0 & -\sin \psi_3 & \cos \psi_3
\end{bmatrix}
\quad \text{and} \quad
\tilde{t}_3 = \begin{bmatrix}
0 & -\cos \psi_1 & -\sin \psi_1 \\
0 & -\cos \psi_2 & -\sin \psi_2 \\
0 & -\cos \psi_3 & -\sin \psi_3
\end{bmatrix}
\quad (16)
\]

Substituting for \( X \) and its time derivatives from Equations 10 and 13 in Equations 5a and 5b, we obtain the system equations of motion in the nonrotating frame as
\[
M_{NR} \ddot{X}_{NR} + C_{NR} \dot{X}_{NR} + K_{NR} X_{NR} = F_{NR} u_{NR} + F_{dNR} w
\quad (17)
\]
and
\[
Y_{NR} = C_{vNR} \dot{X}_{NR} + C_{dNR} X_{NR} + D_{NR} u_{NR} + D_{dNR} w
\quad (18)
\]
The subscript \( NR \) identifies the associated quantity to be in the nonrotating frame. The various matrices in Equations 17 and 18 are as follows:
\[
M_{NR} = MT_1 \\
C_{NR} = 2\Omega MT_2 + CT_1 \\
K_{NR} = \Omega^2 MT_3 + \Omega MT_2 + \Omega CT_2 + KT_1 \\
F_{NR} = FT_{1c} \\
F_{dNR} = F_d \\
C_{vNR} = T_{1o}^{-1} C_v T_1 \\
C_{dNR} = T_{1o}^{-1} (\Omega C_v T_2 + C_v T_3) \\
D_{NR} = T_{1o}^{-1} DT_{1c} \\
D_{dNR} = T_{1o}^{-1} D_d
\quad (19)
\]
Note that the variable-speed operation affects only the stiffness matrix. Also, the disturbance matrix stays unchanged because the disturbance \( w \) is already in the nonrotating frame. If \( F_c \) and \( m_c \) are the number of controls in the fixed and rotating frames, respectively, then the control vectors \( u \) and \( u_{NR} \) are related as follows:
\[
u = T_{1c}^t u_{NR}
\quad (20)
\]
where
\[
T_{1c} = \begin{bmatrix}
I_{F_c \times F_c}
\vdots
\vdots
\end{bmatrix}_{\left( F_c + 3 \times m_c \right) \times \left( F_c + 3 \times m_c \right)}
\]

(21)

Similarly,

\[
Y = T_{1o} Y_{NR}
\]

(22)

where

\[
T_{1o} = \begin{bmatrix}
I_{F_o \times F_o}
\vdots
\vdots
\end{bmatrix}_{\left( F_o + 3 \times m_o \right) \times \left( F_o + 3 \times m_o \right)}
\]

(23)

and where \( F_o \) and \( m_o \) are the number of outputs in the fixed and rotating frames, respectively. Note that \( F_c + 3 \times m_c \) equals \( n_c \), the total number of control variables. Similarly, \( F_o + 3 \times m_o \) equals \( n_o \), the total number of disturbance states. In summary, after applying MBC, the rotating-frame, second-order equations (Equations 5a and 5b) are transformed to the nonrotating-frame equations (Equations 17 and 18).

### 3.2 First-Order System Matrices

First-order system equations are generally expressed as

\[
\dot{z} = Az + Bu + B_d w
\]

(24)

where \( z \) is the state-space vector. The \( A, B, \) and \( B_d \) are, respectively, the system dynamics, control, and disturbance matrices. The \( u \) and \( w \) are the control and disturbance vectors, which are the same vectors that appear in Equation 5a. The associated output equation is

\[
Y = Cz + Du + D_d w
\]

(25)

The size of the \( A \) matrix is \( N \times N \), where \( N \) is the number of system states. The sizes of the \( u, w, \) and \( Y \) vectors are \( n_c, n_w, \) and \( n_o \), respectively, where \( n_c \) is the number of control variables, \( n_w \) is the number of disturbance states, and \( n_o \) is the number of outputs. From
this information, we can readily infer the sizes of the remaining matrices (listed in Table 1). The state-space vector, $z$, is related to the physical vector, $X$, as follows:

$$ z = \begin{bmatrix} X \\ \dot{X} \end{bmatrix} \quad (26) $$

Using Equations 26, 8–16, and 20–23, Equations 24 and 25 are transformed to the nonrotating frame as follows:

$$ \dot{z}_{NR} = A_{NR}z_{NR} + B_{NR}u_{NR} + B_{d_{NR}}w \quad (27) $$

and

$$ Y_{NR} = C_{NR}z_{NR} + D_{NR}u_{NR} + D_{d_{NR}}w \quad (28) $$

The subscript $NR$ identifies the associated quantity as being in the nonrotating frame. The various matrices in Equations 27 and 28 are as follows:

$$ A_{NR} = \begin{bmatrix} T_1^{-1} & 0 \\ 0 & T_1^{-1} \end{bmatrix} \begin{bmatrix} A & T_1 \\ \Omega T_2 & T_1 \end{bmatrix} - \begin{bmatrix} \Omega T_2 & 0 \\ \Omega^2 T_3 + \dot{\Omega} T_2 & 2\Omega T_2 \end{bmatrix} \quad (29) $$

$$ B_{NR} = \begin{bmatrix} T_1^{-1} & 0 \\ 0 & T_1^{-1} \end{bmatrix} B T_{1C} \quad (30) $$

$$ C_{NR} = T_{10}^{-1} \begin{bmatrix} C_d T_1 + \Omega C_r T_2 \\ C_r T_1 \end{bmatrix} ; \quad C = \begin{bmatrix} C_d \\ C_r \end{bmatrix} \quad (31) $$

$$ D_{NR} = T_{10}^{-1} D T_{1C} \quad (32) $$

$$ D_{d_{NR}} = T_{10}^{-1} D_d \quad (33) $$
In summary, after applying MBC, the rotating-frame first-order equations (Equations 24 and 25) are transformed to the nonrotating-frame equations (Equations 27 and 28). Also, note that the dynamics matrix, $A$, is influenced by the variable rotor speed.
4. Input Description

MBC3 is a postprocessor for FAST; its inputs come from a FAST output file. Typically, for a given wind turbine model, the user first runs FAST with the option to generate state-space matrices. These matrices and other ancillary data are stored in a FAST output file named ".lin", where "*" represents the name (without extension) of the input file the user supplies to FAST. For example, if "test01.fst" is the name of the user input file to FAST with an option to generate a linear model, then FAST would generate an output file named "test01.lin" (FAST automatically appends the suffix "lin"). This file contains system matrices at each of the \( N_\psi \) azimuth steps; the user specifies \( N_\psi \) via the parameter \( NAzimStep \) in the FAST input file. The FAST output file also contains the vector \( \Psi \), which is the vector of reference-blade azimuth positions at the \( N_\psi \) steps. The \( \psi \) is the reference azimuth position defined in Equation 1. The user loads the *.lin file into MATLAB using the \( \text{GetMats} \) command (see Section 5). The loaded file includes data items that MBC3 and CCE (discussed in the Section 5) use as inputs. Table 1 lists these inputs and their descriptions.

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Description</th>
<th>See Equation</th>
<th>Symbol</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>( NA_{\text{ctvDOF}} )</td>
<td>Number of degrees of freedom</td>
<td>5a</td>
<td>( n )</td>
<td>scalar</td>
</tr>
<tr>
<td>( N )</td>
<td>Number of states</td>
<td>24</td>
<td>( N )</td>
<td>scalar</td>
</tr>
<tr>
<td>( N_{\text{Inputs}} )</td>
<td>Number of control inputs</td>
<td>34</td>
<td>( N_c )</td>
<td>scalar</td>
</tr>
<tr>
<td>( N_{\text{Disturbs}} )</td>
<td>Number of disturbance states</td>
<td>24</td>
<td>( n_w )</td>
<td>scalar</td>
</tr>
<tr>
<td>( N_{\text{Outs}} )</td>
<td>Number of outputs</td>
<td>35</td>
<td>( N_o )</td>
<td>scalar</td>
</tr>
<tr>
<td>( NA\text{zimStep} )</td>
<td>Number of azimuth steps at which system matrices are input</td>
<td>—</td>
<td>( N_\psi )</td>
<td>scalar</td>
</tr>
<tr>
<td>( \text{Azimuth} )</td>
<td>Vector of azimuth positions, in degrees, of the reference blade (blade 1) with respect to the ( Z_{NR} ) axis shown in Figure 1. System matrices are input at each of these azimuth positions.</td>
<td>—</td>
<td>( \Psi )</td>
<td>( N_\psi )</td>
</tr>
<tr>
<td>( \Omega )</td>
<td>Vector of rotor speeds at specified azimuths (rad/s)</td>
<td>19</td>
<td>( \Omega )</td>
<td>( N_\psi )</td>
</tr>
<tr>
<td>( \Omega_{\text{Dot}} )</td>
<td>Vector of rotor accelerations at specified azimuths (rad/s(^2))</td>
<td>19</td>
<td>( \dot{\Omega} )</td>
<td>( N_\psi )</td>
</tr>
<tr>
<td>( A_{\text{Mat}} )</td>
<td>First-order system matrices at ( N_\psi ) azimuth locations</td>
<td>24</td>
<td>( A )</td>
<td>( N \times N \times N_\psi )</td>
</tr>
<tr>
<td>( B_{\text{Mat}} )</td>
<td>First-order control matrices at ( N_\psi ) azimuth locations</td>
<td>24</td>
<td>( B )</td>
<td>( N \times N_c \times N_\psi )</td>
</tr>
<tr>
<td>( B_{d_{\text{Mat}}} )</td>
<td>First-order disturbance matrices at ( N_\psi ) azimuth locations</td>
<td>24</td>
<td>( B_d )</td>
<td>( N \times N_w \times N_\psi )</td>
</tr>
<tr>
<td>( C_{\text{Mat}} )</td>
<td>First-order output matrices associated with states at ( N_\psi ) azimuth locations</td>
<td>25</td>
<td>( C )</td>
<td>( N_o \times N \times N_\psi )</td>
</tr>
<tr>
<td>( D_{\text{Mat}} )</td>
<td>Output matrices associated with controls at ( N_\psi ) azimuth locations (same for first- and second-order systems)</td>
<td>5b &amp; 25</td>
<td>( D )</td>
<td>( N_o \times N_c \times N_\psi )</td>
</tr>
<tr>
<td>( D_{d_{\text{Mat}}} )</td>
<td>Output matrices associated with disturbances at ( N_\psi ) azimuth locations (same for first- and second-order systems)</td>
<td>5b &amp; 25</td>
<td>( D_d )</td>
<td>( N \times N_w \times N_\psi )</td>
</tr>
<tr>
<td><strong>MassMat</strong></td>
<td>Second-order system mass matrices at $N_\Psi$ azimuth locations</td>
<td>5a</td>
<td>$M$</td>
<td>$n \times n \times N_\Psi$</td>
</tr>
<tr>
<td><strong>DampMat</strong></td>
<td>Second-order system damping matrices at $N_\Psi$ azimuth locations</td>
<td>5a</td>
<td>$C$</td>
<td>$n \times n \times N_\Psi$</td>
</tr>
<tr>
<td><strong>StiffMat</strong></td>
<td>Second-order system stiffness matrices at $N_\Psi$ azimuth locations</td>
<td>5a</td>
<td>$K$</td>
<td>$n \times n \times N_\Psi$</td>
</tr>
<tr>
<td><strong>FMat</strong></td>
<td>Second-order control matrices at $N_\Psi$ azimuth locations</td>
<td>5a</td>
<td>$F$</td>
<td>$n \times N_c \times N_\Psi$</td>
</tr>
<tr>
<td><strong>FdMat</strong></td>
<td>Second-order system disturbance matrices</td>
<td>5a</td>
<td>$F_d$</td>
<td>$n \times N_w \times N_\Psi$</td>
</tr>
<tr>
<td><strong>VelCMat</strong></td>
<td>Second-order output matrices associated with velocities at $N_\Psi$ azimuth locations</td>
<td>5b</td>
<td>$C_v$</td>
<td>$N_v \times n \times N_\Psi$</td>
</tr>
<tr>
<td><strong>DspCMat</strong></td>
<td>Second-order output matrices associated with displacements at $N_\Psi$ azimuth locations</td>
<td>5b</td>
<td>$C_d$</td>
<td>$N_v \times n \times N_\Psi$</td>
</tr>
<tr>
<td><strong>RotTriplet-IndicesStates</strong></td>
<td>The $i^{th}$ row in this matrix identifies the indices of the $i^{th}$ rotating-frame degrees of freedom associated with blades 1, 2, and 3</td>
<td>—</td>
<td>—</td>
<td>$m \times 3$</td>
</tr>
<tr>
<td><strong>RotTripletIndicesCntrlInpt</strong></td>
<td>The $i^{th}$ row in this matrix identifies the indices of the $i^{th}$ rotating-frame control associated with blades 1, 2, and 3</td>
<td>—</td>
<td>—</td>
<td>$m_c \times 3$</td>
</tr>
<tr>
<td><strong>RotTripletIndicesOutput</strong></td>
<td>The $i^{th}$ row in this matrix identifies the indices of the $i^{th}$ rotating-frame output associated with blades 1, 2, and 3</td>
<td>—</td>
<td>—</td>
<td>$m \times 3$</td>
</tr>
<tr>
<td><strong>xdop</strong></td>
<td>Matrix of operating-point state-derivative vectors at $N_\Psi$ azimuth locations</td>
<td>—</td>
<td>—</td>
<td>$N \times N_\Psi$</td>
</tr>
<tr>
<td><strong>xop</strong></td>
<td>Matrix of operating-point state vectors at $N_\Psi$ azimuth locations</td>
<td>—</td>
<td>—</td>
<td>$N \times N_\Psi$</td>
</tr>
<tr>
<td><strong>qd2op</strong></td>
<td>Matrix of operating-point acceleration vectors at $N_\Psi$ azimuth locations</td>
<td>—</td>
<td>—</td>
<td>$n \times N_\Psi$</td>
</tr>
<tr>
<td><strong>qdop</strong></td>
<td>Matrix of operating-point velocity vectors at $N_\Psi$ azimuth locations</td>
<td>—</td>
<td>—</td>
<td>$n \times N_\Psi$</td>
</tr>
<tr>
<td><strong>qop</strong></td>
<td>Matrix of operating-point displacement vectors at $N_\Psi$ azimuth locations</td>
<td>—</td>
<td>—</td>
<td>$n \times N_\Psi$</td>
</tr>
<tr>
<td><strong>yop</strong></td>
<td>Matrix of operating-point output vectors at $N_\Psi$ azimuth locations</td>
<td>—</td>
<td>—</td>
<td>$N_v \times N_\Psi$</td>
</tr>
<tr>
<td><strong>DescStates</strong></td>
<td>Description of degrees-of-freedom vector</td>
<td>—</td>
<td>—</td>
<td>$n \times 1$</td>
</tr>
</tbody>
</table>

The first column in Table 1 lists the names of the inputs required by MBC3. The user does not explicitly input any of the parameters; these are loaded automatically into MATLAB via the command GetMats. The second column gives brief descriptions of the input parameters. Entries in the third and fourth columns refer to equation numbers and symbols used in this guide for the various parameters. A dash in these columns implies that corresponding parameter is not used in the mathematical derivation and, therefore, has no associated symbol or equation number. The last column lists the sizes of these parameters. The MBC3 code automatically identifies the sizes of the various matrices by and checks to ensure compatibility with inputs $n$ and $N$. As described in Section 4, $n$ equals $nF + 3m$, where $nF$ is the degrees of freedom referenced to the fixed frame and $m$ is the number of rotating degrees of freedom for each blade. $N_c$ is the total number of control parameters.
\[ N_c = F_c + 3m_c \quad (34) \]

where \( F_c \) is number of controls referenced to the fixed frame and \( m_c \) is the number of controls referenced to each rotating-blade frame. \( N_o \) is the total number of outputs:

\[ N_o = F_o + 3m_o \quad (35) \]

where \( F_o \) is number of outputs referenced to the fixed frame and \( m_o \) is the number of outputs referenced to each rotating-blade frame.

The matrix \( yop \) is generated by FAST at user request; the \( j^{th} \) column in this matrix represents the output vector at the \( j^{th} \) azimuth location. This column vector is transformed as follows:

\[ Y_{NR} = T_{1o}^{-1} Y_{op} \quad (36) \]

Note that all inputs to MBC3 (first column of Table 1) must come from the FAST-generated *.lin output file. In general, however, some inputs to MBC3 may not be available in the FAST output file. For example, a user can execute FAST with options to generate only first-order matrices and exclude disturbances. In this case, second-order matrices as well as disturbance-related matrices (BdMat and DdMat) will not appear in the FAST output file. MBC3 automatically detects which optional matrices (e.g., control, disturbance, output, and feed-forward matrices) are present and which are not and performs multi-blade coordinate transformation accordingly.

In addition, note that FAST always generates the first-order system matrix, \( AMat \), even if the user requests only a second-order system analysis. Therefore, the \textit{GetMats} command always loads this matrix into the MATLAB workspace.
5. Executing MBC3

To execute MBC3, follow the steps outlined in this section.

**Step 1:** Open MATLAB (ensure that paths have been set to directories where MBC3-related scripts and FAST output files reside). Issue this command:

```
GetMats
```

The following message will appear on the workspace screen, followed by a prompt sign:

```
ENTER:
```

Name of FAST linearization output file to process (omit the .lin extension when entering the name)

At the prompt sign, type the name of the FAST output file without the extension .lin. For example, if `test01.lin` is the name of the FAST output file, type `test01`

To confirm that the inputs listed in Table 1 have been loaded successfully into MATLAB, type `who` and check the parameters that appear in the workspace.

**Step 2:** Issue this command

```
mbc3
```

If the MBC execution is successful, the following message will appear on the workspace screen:

**Multi-Blade Coordinate transformation completed**

Type `who` to confirm that the MBC-related outputs, listed in Table 2 and described in the Section 6, have been generated.

Next, if you are interested in azimuth-averaged system matrices and the system eigenanalysis results, type

```
cce
```

Note that azimuth-averaging of system matrices zeroes out all periodic terms, which is equivalent to retaining just the constant-coefficient terms in the system equations. As a result, the literature often refers to this operation as the constant-coefficient equations
(CCE) approximation. After you have issued the `cce` command and the eigenanalysis of the constant-coefficient approximation is successful, the following message will appear:

**cce.m completed**

Note that `cce` performs eigenanalysis only on the first-order system matrix, $A$. Type `who` to confirm if the CCE-related outputs, listed in Table 3, are generated.

After you have loaded FAST-generated variables into MATLAB (using the `GetMats` command), you have three options to issue further commands: execute only `mbc3`, execute only `cce`, or execute both `mbc3` and `cce` (the sequence is immaterial). If you choose the last option, outputs generated from each command, together with the original inputs, will appear in the MATLAB workspace. Section 6 describes the outputs generated by `mbc3` and `cce` commands.
6. Output Description

The command \textit{mbc3} generates two sets of outputs: MBC-transformed matrices, and eigen-results associated with the azimuth-averaged MBC-transformed system matrices. Table 2 lists these outputs.

<table>
<thead>
<tr>
<th>Output Parameter</th>
<th>Description</th>
<th>Symbol</th>
<th>See Equation</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{MBC}_A</td>
<td>MBC-transformed first-order system matrices at (N_\psi) azimuth locations</td>
<td>(A_{NR})</td>
<td>27</td>
<td>(N \times N \times N_\psi)</td>
</tr>
<tr>
<td>\textit{MBC}_B</td>
<td>MBC-transformed first-order control matrices at (N_\psi) azimuth locations</td>
<td>(B_{NR})</td>
<td>27</td>
<td>(N \times N_c \times N_\psi)</td>
</tr>
<tr>
<td>\textit{MBC}_Bd</td>
<td>MBC-transformed first-order disturbance matrices at (N_\psi) azimuth locations</td>
<td>(B_{dNR})</td>
<td>27</td>
<td>(N \times N_w \times N_\psi)</td>
</tr>
<tr>
<td>\textit{MBC}_C</td>
<td>MBC-transformed output matrices, associated with states, at (N_\psi) azimuth locations</td>
<td>(C_{NR})</td>
<td>28</td>
<td>(N_o \times N \times N_\psi)</td>
</tr>
<tr>
<td>\textit{MBC}_D</td>
<td>MBC-transformed output matrices, associated with controls, at (N_\psi) azimuth locations</td>
<td>(D_{NR})</td>
<td>28</td>
<td>(N_o \times N_c \times N_\psi)</td>
</tr>
<tr>
<td>\textit{MBC}_Dd</td>
<td>MBC-transformed output matrices, associated with disturbances, at (N_\psi) azimuth locations</td>
<td>(D_{dNR})</td>
<td>28</td>
<td>(N_o \times N_w \times N_\psi)</td>
</tr>
<tr>
<td>\textit{MBC}_AvgA</td>
<td>MBC-transformed first-order system matrix (azimuth-averaged)</td>
<td>—</td>
<td>—</td>
<td>(N \times N)</td>
</tr>
<tr>
<td>\textit{MBC}_AvgB</td>
<td>MBC-transformed first-order control matrix (azimuth-averaged)</td>
<td>—</td>
<td>—</td>
<td>(N \times N_c)</td>
</tr>
<tr>
<td>\textit{MBC}_AvgBd</td>
<td>MBC-transformed first-order disturbance matrix (azimuth-averaged)</td>
<td>—</td>
<td>—</td>
<td>(N \times N_w)</td>
</tr>
<tr>
<td>\textit{MBC}_AvgC</td>
<td>MBC-transformed first-order output matrix associated with states (azimuth-averaged)</td>
<td>—</td>
<td>—</td>
<td>(N_o \times N)</td>
</tr>
<tr>
<td>\textit{MBC}_AvgD</td>
<td>MBC-transformed first-order output matrix associated with controls (azimuth-averaged)</td>
<td>—</td>
<td>—</td>
<td>(N_o \times N_c)</td>
</tr>
<tr>
<td>\textit{MBC}_AvgDd</td>
<td>MBC-transformed first-order output matrix associated with disturbances (azimuth-averaged)</td>
<td>—</td>
<td>—</td>
<td>(N_o \times N_w)</td>
</tr>
<tr>
<td>\textit{MBC}_M</td>
<td>MBC-transformed mass matrix at (N_\psi) azimuth locations</td>
<td>(M_{NR})</td>
<td>17</td>
<td>(n \times n \times N_\psi)</td>
</tr>
<tr>
<td>\textit{MBC}_Dmp</td>
<td>MBC-transformed</td>
<td>(C_{NR})</td>
<td>17</td>
<td>(n \times n \times N_\psi)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Transformed</td>
<td>Size</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------</td>
<td>-----------------------</td>
<td></td>
</tr>
<tr>
<td>$MBC_K$</td>
<td>MBC-transformed stiffness matrix at $N_\psi$ azimuth locations</td>
<td>$K_{NR}$</td>
<td>$n \times n \times N_\psi$</td>
<td></td>
</tr>
<tr>
<td>$MBC_F$</td>
<td>MBC-transformed second-order control matrices at $N_\psi$ azimuth locations</td>
<td>$F_{NR}$</td>
<td>$n \times N_c \times N_\psi$</td>
<td></td>
</tr>
<tr>
<td>$MBC_Fd$</td>
<td>MBC-transformed second-order disturbance matrices at $N_\psi$ azimuth locations</td>
<td>$F_{dNR}$</td>
<td>$n \times N_w \times N_\psi$</td>
<td></td>
</tr>
<tr>
<td>$MBC_Vc$</td>
<td>MBC-transformed second-order output matrices associated with velocities at $N_\psi$ azimuth locations</td>
<td>$C_{vNR}$</td>
<td>$N_o \times n \times N_\psi$</td>
<td></td>
</tr>
<tr>
<td>$MBC_Dc$</td>
<td>MBC-transformed second-order output matrices associated with displacements at $N_\psi$ azimuth locations</td>
<td>$C_{dNR}$</td>
<td>$N_o \times n \times N_\psi$</td>
<td></td>
</tr>
<tr>
<td>$MBC_Yop$</td>
<td>MBC-transformed matrix of operating-point output vectors at $N_\psi$ azimuth locations</td>
<td>—</td>
<td>$N_o \times N_\psi$</td>
<td></td>
</tr>
<tr>
<td>$MBC_AvgYop$</td>
<td>MBC-transformed operating-point output vector (azimuth-averaged)</td>
<td>—</td>
<td>$N_o \times 1$</td>
<td></td>
</tr>
<tr>
<td>$MBC_DampedFrequency$</td>
<td>MBC-transformed and azimuth-averaged system damped frequencies in rad/s</td>
<td>—</td>
<td>$M \times 1$</td>
<td></td>
</tr>
<tr>
<td>$MBC_DampedFrequencyHz$</td>
<td>MBC-transformed and azimuth-averaged system damped frequencies in Hz</td>
<td>—</td>
<td>$M \times 1$</td>
<td></td>
</tr>
<tr>
<td>$MBC_DampingRatio$</td>
<td>MBC-transformed and azimuth-averaged system damping ratios</td>
<td>—</td>
<td>$M \times 1$</td>
<td></td>
</tr>
<tr>
<td>$MBC_DecrementRate$</td>
<td>MBC-transformed and azimuth-averaged system decrement rates</td>
<td>—</td>
<td>$M \times 1$</td>
<td></td>
</tr>
<tr>
<td>$MBC_EigenVects$</td>
<td>MBC-transformed and azimuth-averaged system eigenvectors</td>
<td>—</td>
<td>$N \times N$</td>
<td></td>
</tr>
<tr>
<td>$MBC_Evals$</td>
<td>MBC-transformed and azimuth-averaged system eigenvalues</td>
<td>—</td>
<td>$N \times 1$</td>
<td></td>
</tr>
<tr>
<td>$MBC_ModeShapeMagnitude$</td>
<td>MBC-transformed and azimuth-averaged system modal amplitudes</td>
<td>—</td>
<td>$n \times M$</td>
<td></td>
</tr>
<tr>
<td>$MBC_ModeShapePhaseDeg$</td>
<td>MBC-transformed and azimuth-averaged modal phases (deg)</td>
<td>—</td>
<td>$n \times M$</td>
<td></td>
</tr>
<tr>
<td>$MBC_NaturalFrequency$</td>
<td>MBC-transformed and azimuth-averaged natural frequencies in rad/s</td>
<td>—</td>
<td>$M \times 1$</td>
<td></td>
</tr>
<tr>
<td>$MBC_NaturalFrequencyHz$</td>
<td>MBC-transformed and azimuth-averaged natural frequencies in Hz</td>
<td>—</td>
<td>$M \times 1$</td>
<td></td>
</tr>
<tr>
<td>$MBC_{reduced_size_eigenVals}$</td>
<td>MBC-transformed and reduced-size system eigenvalues</td>
<td>—</td>
<td>$M \times 1$</td>
<td></td>
</tr>
</tbody>
</table>
Symbols in the fifth column identify sizes of the MBC3-generated outputs. Section 4 defines all symbols except \( M \). Before we define \( M \), note that eigenvalues of a real system matrix appear as real or as complex conjugate values. A pair of complex conjugate eigenvalues refers to the same mode. Only one of these conjugate eigenvalues, say the eigenvalue with the positive imaginary part, suffices to yield modal frequency and damping; the other eigenvalue, the one with the negative imaginary part, does not give any additional information. Accordingly, we output eigen-results corresponding to only those eigenvalues that are either real or have positive imaginary parts.

Now, we are ready to define \( M \); it equals \( N – m_n \), where \( m_n \) is the number of eigenvalues with negative imaginary parts. Thus, elements of vectors \( MBC_{DampedFrequency} \), \( MBC_{DampedFrequencyHz} \), \( MBC_{DampingRatio} \), \( MBC_{DecrementRate} \), \( MBC_{NaturalFrequency} \), \( MBC_{NaturalFrequencyHz} \), and \( MBC_{eigenVals} \) correspond to those eigenvalues whose imaginary parts are either zero or positive.

The command \( cce \), as described in Section 5, generates azimuth-averaged matrices for first- and second-order systems and also obtains an eigensolution of the system matrix \( A \). Table 3 lists these outputs.

### Table 3. CCE-Generated Outputs

<table>
<thead>
<tr>
<th>Output Parameter</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvgAMat</td>
<td>Azimuth-averaged system matrix</td>
<td>( N \times N )</td>
</tr>
<tr>
<td>AvgBMat</td>
<td>Azimuth-averaged first-order control matrix</td>
<td>( N \times N_c )</td>
</tr>
<tr>
<td>AvgBdMat</td>
<td>Azimuth-averaged first-order disturbance matrix</td>
<td>( N \times N_w )</td>
</tr>
<tr>
<td>AvgCMat</td>
<td>Azimuth-averaged first-order output matrix associated with states</td>
<td>( N_c \times N )</td>
</tr>
<tr>
<td>AvgDMat</td>
<td>Azimuth-averaged first-order output matrix associated with controls</td>
<td>( N_c \times N_c )</td>
</tr>
<tr>
<td>AvgDdMat</td>
<td>Azimuth-averaged output matrix associated with disturbances</td>
<td>( N_c \times N_w )</td>
</tr>
<tr>
<td>Avgxdop</td>
<td>Azimuth-averaged vector of the time-derivatives of the states</td>
<td>( N \times 1 )</td>
</tr>
<tr>
<td>Avgxop</td>
<td>Azimuth-averaged vector of the system states</td>
<td>( N \times 1 )</td>
</tr>
<tr>
<td>AvgMassMat</td>
<td>Azimuth-averaged mass matrix</td>
<td>( n \times n )</td>
</tr>
<tr>
<td>AvgDampMat</td>
<td>Azimuth-averaged damping matrix</td>
<td>( n \times n )</td>
</tr>
<tr>
<td>AvgStiffMat</td>
<td>Azimuth-averaged stiffness matrix</td>
<td>( n \times n )</td>
</tr>
<tr>
<td>AvgFMat</td>
<td>Azimuth-averaged second-order control matrix</td>
<td>( n \times N_c )</td>
</tr>
<tr>
<td>AvgFdMat</td>
<td>Azimuth-averaged second-order disturbance matrix</td>
<td>( n \times N_w )</td>
</tr>
<tr>
<td>AvgVeilCMat</td>
<td>Azimuth-averaged second-order output matrix associated</td>
<td>( N_c \times n )</td>
</tr>
</tbody>
</table>
The decrement rates listed in Tables 2 and 3 are simply the negatives of the real parts of the associated eigenvalues. We recommend using these rates when presenting stability analysis results. Alternately, damping ratios can be used. Problems can arise, however, if an eigenvalue has a zero imaginary part (i.e., it represents a rigid-body mode). In this case, the damping ratio would be either infinity or indeterminate.
7. Downloading and Verifying MBC-Specific Files

The MATLAB scripts required to carry out the MBC operation are `mbc3.m`, `eianalysis.m`, `get_new_seq.m`, and `row_col_xtion.m`. The `mbc3.m` is the main script and calls the other three. The script `GetMats.m` is required to load select contents of the FAST-generated linear-data output file to the MATLAB workspace. The `cce.m` script is not required for MBC. We include it, however, to allow interested users to obtain eigen-results of an azimuth-averaged system matrix. The `cce.m` script calls `eianalysis.m` to perform eigenanalysis. Visit our Web site at http://wind.nrel.gov/designcodes/postprocessors/ to download these scripts, along with sample I/O files and this guide.

If the download is successful, you will see the main folder named `MBC` and two subfolders named `Source` and `CertTest`. The main folder contains the user’s manual. The `Source` subfolder contains the MATLAB scripts required for MBC and CCE operations. The `CertTest` subfolder contains a test input file named `test01.lin` and a subfolder named `TestFiles`. The file `test01.lin` was generated by FAST with options to output first-order state matrices, including control, disturbance, and output matrices. You can open this file to examine these matrices and supporting information describing state vectors, number of azimuth steps, and so on (for a description of the contents of this file, see the FAST user’s manual, which is also available on our Web site). Currently, we have only one test input file, but we plan to add more in the future. The subfolder `TestFiles` contains the output file `Filetest01.mat`, which is a MATLAB binary file generated using the sequence of commands `GetMats`, `mbc3`, and `cce` described in Section 5. You can load this file into MATLAB and examine its contents if you like (Tables 1–3 describe these contents).

To verify that you have successfully loaded the required files, set the path name to the `CertTest` subfolder in your MATLAB workspace. Next, issue the command `CertTest`. This command performs MBC and CCE operations and compares the resulting outputs with those in the `Filetest01.mat` output file. The results of this comparison are displayed in the MATLAB workspace. No difference displayed for all variables implies successful verification. Now you can use FAST to generate state matrices for a turbine of your interest and follow the steps described in Section 5 to perform MBC or CCE or both.
8. Future Plans

Though the MBC formulation we developed assumes a three-bladed turbine, it is general enough to be readily extended to a turbine with more than three blades. In the future, depending on user demand, we will update both the theory basis and the scripts to accommodate rotors with three or more blades. We will exclude two-bladed rotors because, even though an MBC-like transformation is theoretically possible for a two-bladed rotor (yielding coning and teeter rotor modes), such a transformation does not result in a filtering action similar to the one that the conventional MBC provides.

As mentioned earlier, the current MBC scripts work only with FAST. The future versions will accept data (e.g., state matrices) from any rotary-system aeroelastic code. We will also write a FORTRAN-based MBC utility and integrate it into FAST. This will allow a user to invoke MBC operation directly from the FAST input file.

Finally, we will update the theory basis to explain the filtering operation of MBC and include additional I/O files to demonstrate this operation. We will also include results to show how MBC can be applied to rotors with dissimilar blades.
References


24
This guide explains how to use MBC3, a MATLAB-based script NREL developed to perform multi-blade coordinate transformation of system matrices for three-bladed wind turbines. In its current form, MBC3 can be applied to system matrices generated by FAST2.