

User's Guide to MBC3

(Multi-blade Coordinate Transformation Utility for 3-Bladed Wind Turbines)

Gunjit S. Bir

October 2008

National Renewable Energy Laboratory,
1617, Cole Blvd, Golden, CO 80401

About this Guide

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. This is because 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: transformation of the rotating degrees of freedom, and transformation of the equations of motion. Ref [1] details the MBC operation. However, a summary of the MBC concept and underlying transformations is provided in this manual.

This guide describes usage of MBC3, a MATLAB-based script we have developed to perform multi-blade coordinate transformation of wind turbine system matrices. In its current form, MBC3 is applicable to system matrices generated by FAST¹ only. The system matrices may have arbitrary—but physically consistent—sizes. These matrices may also optionally include control, disturbance, output, and feed-forward matrices. Also, the associated state, control, and disturbance vectors may 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, a user may generate system matrices either in 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. Following multi-blade coordinate transformation, MBC3 also performs eigenanalysis, results of which are presented in different formats to aid modal and stability analysis. MBC3 automatically detects if 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 applications³, but 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, thereby producing errors. While averaging itself is not always a bad approach, it must follow MBC. Reference [1] provides sample results that illustrate this point. Main advantages of MBC are: capturing cumulative dynamics of the rotor blades and its interaction with the tower-nacelle subsystem, well-conditioning of system matrices by eliminating non-essential periodicity, and filtering operation¹.

This guide provides 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 fast (usually in a few seconds). If this guide does not help resolve a problem you may have with script execution, contact us. In future, we will upgrade MBC3 so it could be directly integrated with FAST or any other aeroelastic code.

DOCUMENT REVISION RECORD

Revision	Date	Description
1.00	13-Oct-2008	First publication. Supports MBC3 v1.00.00.

DRAFT

Contents

About this Guide	ii
Contents	ii
1 Introduction.....	3
2 MBC Concept and Associated Transformation	4
3 Transformation of System Matrices.....	5
3.1 Second-Order System Matrices	5
3.2 First-Order System Matrices	9
4 Input Description	10
5 Executing MBC3	12
6 Output Description.....	13
7 Downloading and Verifying MBC-Specific Files	15
8 Future Plans	16
Acknowledgments.....	16
References.....	17

DRAFT

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. This is because 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 tower-nacelle subsystem. Third, MBC filters out all periodic terms except those which are integral multiples of ΩN , where Ω 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, which 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 only specific harmonics of blade motion to be felt by the fixed system. This filtering action also renders the system equations numerically well-conditioned; all nonessential periodic terms are eliminated. MBC is widely used in the helicopter field. Miller⁴ used it to analyze the flap-motion related stability and control. Coleman and Feingold⁵ used it to analyze the rotor in-plane motion (lag motion); it was the first successful attempt to understand the helicopter ground resonance problem, which had been eluding the earlier researchers. However, these efforts applied the MBC only in a heuristic fashion. Hohenemser and Yin⁶ provided the first mathematically sound basis. Later, Johnson³ provided a systematic mathematical basis and thorough exposition of the MBC. Using this mathematical basis, Bir et al⁷ developed a numerical MBC approach that could be applied to a general helicopter system governed by arbitrary degrees of freedom; the approach was used for stability and response analysis of several helicopters.

Because MBC offers so many benefits, it is receiving attention in the wind turbine field. Bir and Butterfield⁸ 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 provides a physical insight into the rotor in-plane motion. In the turbine field, Malcolm⁹ appears to be the first to provide a mathematical form of the turbine equations following application of 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 ADAMS¹⁰. McCoy¹¹ extended this effort to obtain linear time-invariant system equations required in the standard control design approaches. Hansen¹² used MBC for improved modal dynamics to avoid stall-induced vibrations and later combined it with an eigenvalues approach to study the aeroelastic stability characteristics of a three-bladed turbine¹³. Riziotis et al¹⁴ 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 configurations¹⁵.

All attempts at MBC thus far, both in the helicopter and wind turbine fields, have assumed the rotor speed to be constant and the rotor blades to be 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 the two limitations.

This paper provides 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 either in the first-order (state-space) form or the second-order (physical-domain) form. We develop MBC relations for both these forms. In literature, MBC is also referred to as the Fourier coordinate transformation (FCT) and as the Coleman transformation.

Though MBC has several applications, it is essential to and mostly used for 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. The averaging, however, eliminates all periodic terms that contribute to system dynamics and can cause erroneous results. While averaging itself is not a bad approach, it must follow MBC¹. Theoretically, a Floquet analysis—not averaging—must follow MBC. However, in normal cases, MBC eliminates the dominant periodic terms; the remaining periodic terms (N/rev , $2N/\text{rev}$, ..., etc. for an N -

bladed turbine) usually have a negligible magnitude and, therefore, direct averaging of MBC-transformed matrices suffices.

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 and identifies our website from which these scripts, along with sample I/O files, may be accessed. Section 8, the concluding section, summarizes the 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 of which may be in the rotating frame and the other in the nonrotating frame. This is sometimes desirable. For example, in some simulations studies, we may 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 tower-nacelle-rotor system. In such cases, it is desirable 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 blades that are spaced equally around the rotor azimuth. In such a case, the azimuth location of b th blade is given by

$$\psi_b = \psi + (b-1) \frac{2\pi}{N} \quad (1)$$

where ψ is the azimuth of the first (reference) blade (see Figure 1 in which N , the number of blades, is 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 follows³

$$\begin{aligned} q_0 &= \frac{1}{N} \sum_{b=1}^N q_b \\ q_{nc} &= \frac{2}{N} \sum_{b=1}^N q_b \cos n\psi_b \\ q_{ns} &= \frac{2}{N} \sum_{b=1}^N q_b \sin n\psi_b \\ q_{N/2} &= \frac{1}{N} \sum_{b=1}^N q_b (-1)^b \end{aligned} \quad (2)$$

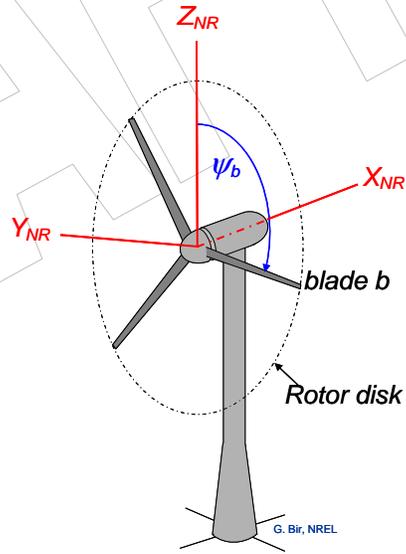


Figure 1: Rotor nonrotating frame (X_{NR}, Y_{NR}, Z_{NR})

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 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 (and not individual blades) in the fixed frame. The physical interpretation of each rotor coordinate depends on the degree of freedom it refers to. 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 plane⁸. Other degrees of freedom may be interpreted using equations (2) and drawing supporting sketches. The rotor modes corresponding to q_{nc} and q_{ns} ($n>1$) and $q_{N/2}$ 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 $(N-1)/2$ if N is odd, and from 1 to $(N-2)/2$ if N is even. The $q_{N/2}$ mode, called the differential mode, exists only if the number of blades is even³.

Most of the wind turbines in the world are three-bladed. Therefore, we will assume N to be 3 in the rest of this paper. The formulation developed in the paper for $N=3$, however, is general and can be readily extended for more blades. Setting $N=3$ in equations (2), we obtain

$$\begin{aligned} q_0 &= \frac{1}{3} \sum_{b=1}^3 q_b \\ q_c &= q_{1c} = \frac{2}{3} \sum_{b=1}^3 q_b \cos \psi_b \\ q_s &= q_{1s} = \frac{2}{3} \sum_{b=1}^3 q_b \sin \psi_b \end{aligned} \quad (3)$$

We will 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. Equations (3) determine the rotor coordinates, given the blade coordinates. The inverse transformation, yielding the blade coordinate given the rotor coordinates, is

$$q_b = q_0 + q_c \cos \psi_b + q_s \sin \psi_b; \quad b = 1, 2, 3 \quad (4)$$

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 may be written as

$$M\ddot{X} + C\dot{X} + KX = Fu + F_d w \quad (5a)$$

and the associated output equations as

$$Y = C_v \dot{X} + C_d X + Du + D_d w \quad (5b)$$

where M , C , K , F , and F_d are respectively the mass, damping, stiffness, control, and disturbance matrices. 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 size of the u , w , and Y vectors is and 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. However, the formulation we develop does not require a user to explicitly partition any of the matrices into sub-matrices delineating direct and cross-coupling terms. The partitioning is implicit in the definition of the coordinates vector X (also called the physical vector in literature). Assume that X is expressed as

$$X = \begin{Bmatrix} X_F \\ q_1^1 \\ q_2^1 \\ q_3^1 \\ \vdots \\ q_1^j \\ q_2^j \\ q_3^j \\ \vdots \\ q_1^m \\ q_2^m \\ q_3^m \end{Bmatrix} \quad (6)$$

where X_F is a $nF \times 1$ column vector representing the nF fixed-frame-referenced degrees of freedom and q_b^j is the j^{th} rotating degree of freedom for the b^{th} blade. As is evident from (6), m is the number of rotating degrees of freedom for each blade. Thus the length n of vector X thus equals $nF+3m$, the total number of degrees of freedom for the full system. Most often, aeroelastic codes, e.g. FAST, assume that the physical vector has the form (6). If not, we can always perform simple rows and columns exchange to transform system equations (5) such that the physical vector X assumes the form (6).

There are two methods to transform equations (5) to the nonrotating frame: the operational method and the substitution method³. In the operational method, we apply the following summation operators to the rotating-frame equations of motion:

$$\begin{aligned} & \frac{1}{3} \sum_{b=1}^3 (rot_eqns) \\ & \frac{2}{3} \sum_{b=1}^3 (rot_eqns) \cos \psi_b \\ & \frac{2}{3} \sum_{b=1}^3 (rot_eqns) \sin \psi_b \end{aligned} \quad (7)$$

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 that the *rot-eqns* are available in an analytical form showing all periodic terms explicitly; 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 equations (3). Finally, we use transformation (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, therefore, are 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 so, 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, may be transformed to nonrotating coordinates (q_0^j, q_c^j, q_s^j) using the following relation:

$$\begin{Bmatrix} q_1^j \\ q_2^j \\ q_3^j \end{Bmatrix} = \tilde{t} \begin{Bmatrix} q_0^j \\ q_c^j \\ q_s^j \end{Bmatrix} \quad (8)$$

where

$$\tilde{t} = \begin{bmatrix} 1 & \cos \psi_1 & \sin \psi_1 \\ 1 & \cos \psi_2 & \sin \psi_2 \\ 1 & \cos \psi_3 & \sin \psi_3 \end{bmatrix} \quad (9)$$

Using (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

$$X = T_1 X_{NR} \quad (10)$$

where T_1 is the block diagonal matrix:

$$T_1 = \begin{bmatrix} I_{nF \times nF} & & & & \\ & \tilde{t} & & & \\ & & \tilde{t} & & \\ & & & \ddots & \\ & & & & \tilde{t} \end{bmatrix}_{(nF+3m) \times (nF+3m)} \quad (11)$$

and

$$X_{NR} = \begin{Bmatrix} X_F \\ q_0^1 \\ q_c^1 \\ q_s^1 \\ \vdots \\ q_0^j \\ q_c^j \\ q_s^j \\ \vdots \\ q_0^m \\ q_c^m \\ q_s^m \end{Bmatrix} \quad (12)$$

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

$$\begin{aligned} \dot{X} &= T_1 \dot{X}_{NR} + \Omega T_2 X_{NR} \\ \ddot{X} &= T_1 \ddot{X}_{NR} + 2\Omega T_2 \dot{X}_{NR} + (\Omega^2 T_3 + \dot{\Omega} T_2) X_{NR} \end{aligned} \quad (13)$$

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

$$T_2 = \begin{bmatrix} 0_{nF \times nF} & & & & \\ & \tilde{t}_2 & & & \\ & & \tilde{t}_2 & & \\ & & & \ddots & \\ & & & & \tilde{t}_2 \end{bmatrix}_{(nF+3m) \times (nF+3m)} \quad (14)$$

$$T_3 = \begin{bmatrix} 0_{nF \times nF} & & & & \\ & \tilde{t}_3 & & & \\ & & \tilde{t}_3 & & \\ & & & \ddots & \\ & & & & \tilde{t}_3 \end{bmatrix}_{(nF+3m) \times (nF+3m)} \quad (15)$$

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 appearing in equations (17) and (18) are as follows

$$\begin{aligned} M_{NR} &= MT_1 \\ C_{NR} &= 2\Omega MT_2 + CT_1 \\ K_{NR} &= \Omega^2 MT_3 + \dot{\Omega} MT_2 + \Omega CT_2 + KT_1 \\ F_{NR} &= FT_{1c} \\ F_{dNR} &= F_d \\ C_{vNR} &= T_{1o}^{-1} C_v T_1 \\ C_{dNR} &= T_{1o}^{-1} (\Omega C_v T_2 + C_d T_1) \\ D_{NR} &= T_{1o}^{-1} DT_{1c} \\ D_{dNR} &= T_{1o}^{-1} D_d \end{aligned} \quad (19)$$

Note that the variable-speed operation affects only the stiffness matrix. Also, the disturbance matrix stays unchanged; this is 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:

$$u = T_{1c} u_{NR} \quad (20)$$

where

$$T_{1c} = \begin{bmatrix} I_{F_c \times F_c} & & & & \\ & \tilde{t} & & & \\ & & \tilde{t} & & \\ & & & \ddots & \\ & & & & \tilde{t} \end{bmatrix}_{(F_c+3mc) \times (F_c+3mc)} \quad (21)$$

Similarly,

$$Y = T_{1o} Y_{NR} \quad (22)$$

where

$$T_{1o} = \begin{bmatrix} I_{F_o \times F_o} & & & & \\ & \tilde{t} & & & \\ & & \tilde{t} & & \\ & & & \ddots & \\ & & & & \tilde{t} \end{bmatrix}_{(F_o+3mo) \times (F_o+3mo)} \quad (23)$$

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

3.2 First-Order System Matrices

First-order system equations are generally expressed as

$$\dot{z} = Az + Bu + B_d w \quad (24)$$

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. The associated output equations are

$$Y = Cz + Du + D_d w \quad (25)$$

The size of the A matrix is $N \times N$, where N is the number of system states. The size of the u , w , and Y vectors is and 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, the sizes of the remaining matrices can be readily inferred. These sizes are listed later in the guide (see 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:

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

and

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

The subscript NR identifies the associated quantity to be in the nonrotating frame. The various matrices appearing in equations (27) and (28) are as follows

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

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

$$C_{NR} = T_{10}^{-1} [C_d T_1 + \Omega C_v T_2 \quad C_v T_1]; \quad C = [C_d \quad C_v] \quad (31)$$

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

$$D_{dNR} = T_{10}^{-1} D_d \quad (33)$$

In summary, following the application of MBC, the rotating-frame first-order equations (24) and (25) are transformed to the nonrotating-frame equations (27) and (28). Also, note that the dynamics matrix, A , is influenced by the variable rotor speed.

4 Input Description

MBC3 is a post-processor for FAST; its inputs come from a FAST output file. Typically, for a given wind turbine model, a 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 supplied by the user to FAST. Thus if *test01.fst* is the name of the user input file to FAST with an option to generate linear model, then FAST would generate an output file named *test01.lin* (suffix *‘.lin’* is automatically appended by FAST). This file contains system matrices at each of the N_ψ azimuth steps; N_ψ is specified by the user via the parameter *NAzimStep* in the FAST input file. The FAST output file also contains the vector Ψ , which is the vector of reference-blade azimuth positions at the N_ψ steps. The reference azimuth, ψ , is defined via Equation (1). The user then loads the **.lin* file into MATLAB using the *GetMats* command (see next section). The loaded file includes data items that MBC3 uses as inputs. Table 1 lists these inputs and their descriptions.

Table 1. Inputs to MBC3

Input parameter	Description	See Eqn.	Symbol	Size
<i>NActvDOF</i>	Number of degrees of freedom	5a	n	<i>scalar</i>
N	Number of states	24	N	<i>scalar</i>
<i>NAzimStep</i>	Number of azimuth steps at which system matrices are input	-	N_ψ	<i>scalar</i>
<i>Azimuth</i>	Vector of azimuth positions, in degrees, of the reference blade (i.e. blade 1) at which system matrices are input	-	Ψ	N_ψ
<i>Omega</i>	Vector of rotor speeds at specified azimuths (rad/sec)	19	Ω	N_ψ
<i>OmegaDot</i>	Vector of rotor accelerations at specified azimuths (rad/sec ²)	19	$\dot{\Omega}$	N_ψ
<i>AMat</i>	First-order system matrices at N_ψ azimuth locations	24	A	$N \times N \times N_\psi$
<i>BMat</i>	First-order control matrices at N_ψ azimuth locations	24	B	$N \times N_c \times N_\psi$

<i>BdMat</i>	First-order disturbance matrices at N_ψ azimuth locations	24	B_d	$N \times N_w \times N_\psi$
<i>CMat</i>	First-order output matrices associated with states at N_ψ azimuth locations	25	C	$N_o \times N_c \times N_\psi$
<i>DMat</i>	Output matrices associated with controls at N_ψ azimuth locations (same for first- and second-order systems)	5b & 25	D	$N_o \times N_c \times N_\psi$
<i>DdMat</i>	Output matrices associated with disturbances at N_ψ azimuth locations (same for first- and second-order systems)	5b & 25	D_d	$N_o \times N_w \times N_\psi$
<i>MassMat</i>	Second-order system mass matrices at N_ψ azimuth locations	5a	M	$n \times n \times N_\psi$
<i>DampMat</i>	Second-order system damping matrices at N_ψ azimuth locations	5a	C	$n \times n \times N_\psi$
<i>StffMat</i>	Second-order system stiffness matrices at N_ψ azimuth locations	5a	K	$n \times n \times N_\psi$
<i>FMat</i>	Second -order control matrices at N_ψ azimuth locations	5a	F	$n \times N_c \times N_\psi$
<i>FdMat</i>	Second-order system disturbance matrices	5a	F_d	$n \times N_w \times N_\psi$
<i>VelCMat</i>	Second -order output matrices associated with velocities at N_ψ azimuth locations	5b	C_v	$N_o \times n \times N_\psi$
<i>DspCMat</i>	Second -order output matrices associated with displacements at N_ψ azimuth locations	5b	C_d	$N_o \times n \times N_\psi$
<i>RotTriplet-IndicesStates</i>	The i^{th} row in this matrix identifies the i^{th} rotating - frame dof associated with blades 1, 2, and 3	--	--	$m \times 3$
<i>RotTripletIndicesCntrlInpt</i>	The i^{th} row in this matrix identifies the i^{th} rotating- frame control associated with blades 1, 2, and 3	--	--	$m_c \times 3$
<i>RotTripletIndicesOutput</i>	The i^{th} row in this matrix identifies the i^{th} rotating- frame output associated with blades 1, 2, and 3	--	--	$m_o \times 3$

In the above Table, the 1st column lists the names of the inputs required by MBC3. It should be emphasized that the user does not explicitly input any of the parameters; these are loaded automatically into MATLAB via the command *GetMats*. The 2nd column provides a brief description of the input parameters. Entries in the 3rd and 4th columns refer to equation numbers and symbols used in this guide for the various parameters. The last column lists the sizes of these parameters. The sizes of the various matrices are automatically identified by the MBC3 code and checked 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, which equals F_c+mc , where F_c is number of controls referenced to the fixed frame and mc is the number of controls referenced to each rotating-blade frame. N_o is the total number of outputs, which equals F_o+mo , where F_o is number of outputs referenced to the fixed frame and mo is the number of outputs referenced to each rotating-blade frame.

Note that all inputs to MBC3, listed in the 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 may execute FAST with options to generate only first-

order matrices and exclude disturbances. In such a 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 or not and performs multi-blade coordinate transformation accordingly.

The user should also note that FAST always generates the first-order system matrix, *AMat*, even if only a second-order system analysis is requested.

5 Executing MBC3

As described in the previous section, inputs to MBC3 come from FAST output. A 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 supplied by the user to FAST. Thus if *test01.fst* is the name of the user input file to FAST with an option to generate linear model, then FAST would generate an output file named *test01.lin* (suffix *.lin* is automatically appended by FAST). The user then performs the following simple steps:

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

GetMats

The following message will appear in 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 that without the extension *.lin*. Thus, if *test01.lin* is the name of the FAST output file, you would type

test01

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

Step 2: Issue the command:

mbc3

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

Multi-Blade Coordinate transformation completed

Type *who* to confirm that the MBC-related outputs, described in the next section, have been generated.

Next, should you be interested in azimuth-averaged first-order system matrix and associated eigenanalysis results, type

cce

Note that azimuth-averaging of system matrices zeros out all periodic terms; which is equivalent to retaining just the constant-coefficient terms in the system equations. In literature, therefore, this operation is often referred to as the constant-coefficient equations (CCE) approximation. After you have issued the *cce* command and the constant-coefficient-approximation eigenanalysis 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.

Note that 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 be present in the MATLAB workspace. The next section describes the outputs generated by *mbc3* and *cce* commands..

6 Output Description

The command *mbc3*, described in the previous section, generates two sets of outputs: a) MBC-transformed matrices and b) eigen-results associated with the azimuth-averaged MBC-transformed system matrices. These outputs are listed in Table 2.

Table 2. MBC3-Generated Outputs

Output Parameter	Description	Symbol	See Eqn.	Size
<i>MBC_A</i>	MBC-transformed first-order system matrices at N_ψ azimuth locations	A_{NR}	27	$N \times N \times N_\psi$
<i>MBC_AvgA</i>	MBC-transformed first-order system matrix (azimuth-averaged)	--	--	$N \times N$
<i>MBC_B</i>	MBC-transformed first-order control matrices at N_ψ azimuth locations	B_{NR}	27	$N \times N_c \times N_\psi$
<i>MBC_Bd</i>	MBC-transformed first-order disturbance matrices at N_ψ azimuth locations	B_{dNR}	27	$N \times N_w \times N_\psi$
<i>MBC_C</i>	MBC-transformed output matrices, associated with states, at N_ψ azimuth locations	C_{NR}	28	$N_o \times N \times N_\psi$
<i>MBC_D</i>	MBC-transformed output matrices, associated with controls, at N_ψ azimuth locations	D_{NR}	28	$N_o \times N_c \times N_\psi$
<i>MBC_Dd</i>	MBC-transformed output matrices, associated with disturbances, at N_ψ azimuth locations	D_{dNR}	28	$N_o \times N_w \times N_\psi$
<i>MBC_M</i>	MBC-transformed mass matrix at N_ψ azimuth locations	M_{NR}	17	$n \times n \times N_\psi$
<i>MBC_Dmp</i>	MBC-transformed damping/gyroscopic	C_{NR}	17	$n \times n \times N_\psi$

	matrix at N_ψ azimuth locations			
<i>MBC_K</i>	MBC-transformed stiffness matrix at N_ψ azimuth locations	K_{NR}	17	$n \times n \times N_\psi$
<i>MBC_F</i>	MBC-transformed second-order control matrices at N_ψ azimuth locations	F_{NR}	17	$n \times N_c \times N_\psi$
<i>MBC_Fd</i>	MBC-transformed second-order disturbance matrices at N_ψ azimuth locations	F_{dNR}	17	$n \times N_w \times N_\psi$
<i>MBC_Vc</i>	Second -order output matrices associated with velocities at N_ψ azimuth locations	C_{vNR}	18	$N_o \times n \times N_\psi$
<i>MBC_Dc</i>	Second -order output matrices associated with displacements at N_ψ azimuth locations	C_{dNR}	18	$N_o \times n \times N_\psi$
<i>MBC_DampedFrequency</i>	MBC-transformed and azimuth-averaged system damped frequencies in rad/sec	--	--	$M \times 1$
<i>MBC_DampedFrequencyHz</i>	MBC-transformed and azimuth-averaged system damped frequencies in Hz	--	--	$M \times 1$
<i>MBC_DampingRatio</i>	MBC-transformed and azimuth-averaged system damping ratios	--	--	$M \times 1$
<i>MBC_DecrementRate</i>	MBC-transformed and azimuth-averaged system decrement rates	--	--	$M \times 1$
<i>MBC_EigenVects</i>	MBC-transformed and azimuth-averaged system eigenvectors	--	--	$N \times N$
<i>MBC_Evals</i>	MBC-transformed and azimuth-averaged system eigenvalues	--	--	$N \times 1$
<i>MBC_ModeShapeMagnitude</i>	MBC-transformed and azimuth-averaged system modal amplitudes	--	--	$n \times M$
<i>MBC_ModeShapePhaseDeg</i>	MBC-transformed and azimuth-averaged modal phases (deg)	--	--	$n \times M$
<i>MBC_NaturalFrequency</i>	MBC-transformed and azimuth-averaged natural frequencies in rad/sec	--	--	$M \times 1$
<i>MBC_NaturalFrequencyHz</i>	MBC-transformed and azimuth-averaged natural frequencies in Hz	--	--	$M \times 1$
<i>cc</i>	MBC-transformed and azimuth-averaged system eigenvalues (showing complex conjugate eigenvalues only with non-negative imaginary parts)	--	--	$M \times 1$
<i>MBC_eigenVects</i>	MBC-transformed and azimuth-averaged system eigenvectors	--	--	$n \times M$

	associated with $MBC_eigenVals$			
--	----------------------------------	--	--	--

Symbols in the fifth column identify sizes of the MBC3-generated outputs. All symbols, except M , are defined in Section 4. 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 provide any additional information. Accordingly, we output eigen-results corresponding to only those eigenvalues which 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 the previous section, generates azimuth-averaged system matrix A and its eigen-results. Table 3 lists these outputs.

Table 3. CCE-Generated Outputs

Output Parameter	Description	Size
<i>AvgEigenvectors</i>	Eigenvectors of azimuth-averaged system matrix	$N \times N$
<i>AvgEvals</i>	Eigenvalues of azimuth-averaged system matrix	$N \times 1$
<i>AvgEigenVects</i>	Eigenvectors of azimuth-averaged system matrix	$n \times M$
<i>AvgEigenVals</i>	Eigenvalues of azimuth-averaged system matrix	$M \times 1$
<i>AvgDampedFrequency</i>	Damped frequencies of azimuth-averaged system matrix (rad/s)	$M \times 1$
<i>AvgDampedFrequencyHz</i>	Damped frequencies of azimuth-averaged system matrix (Hz)	$M \times 1$
<i>AvgNaturalFrequency</i>	Natural frequencies of azimuth-averaged system matrix (rad/s)	$M \times 1$
<i>AvgNaturalFrequencyHz</i>	Natural frequencies of azimuth-averaged system matrix (Hz)	$M \times 1$
<i>AvgDecrementRate</i>	Decrement rates of azimuth-averaged system matrix	$M \times 1$
<i>AvgDampingRatio</i>	Damping ratios of azimuth-averaged system matrix	$M \times 1$
<i>AvgModeShapeMagnitude</i>	Modal amplitudes of azimuth-averaged system matrix	$n \times M$
<i>AvgModeShapePhaseDeg</i>	Modal phases of azimuth-averaged system matrix (deg)	$n \times M$

The decrement rates, listed in Tables 2 and 3, are simply the negatives of the real parts of the associated eigenvalues. We recommend that these rates be used while presenting stability analysis results. Alternately, damping ratios may be used. However, problem can arise if an eigenvalue has zero imaginary part (i.e., it represents a rigid-body mode); in such a case, the damping ratio would be either infinity or indeterminate.

7 Downloading and Verifying MBC-Specific Files

MATLABS 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. It is, however,

included to allow an interested user to obtain eigen-results of an azimuth-averaged system matrix. The *cce.m* script calls *eianalysis.m* to perform eigenanalysis. These scripts, along with sample I/O files and this manual, may be downloaded from our website:

<http://wind.nrel.gov/designcodes/postprocessors/>

If the download is successful, you should 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 1st-order state matrices including control, disturbance and output matrices. You may open this file to examine these matrices and supporting information describing state vectors, number of azimuth steps, etc (for a description of the contents of this file, see FAST user's manual, which is available on our website). Currently, we have only one test input file; we will include 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. If you are curious, you may load this file into MATLAB and examine its contents (Tables 1-3 provide a description of these contents).

To verify that you have successfully loaded the required files, set the pathname to the *CertTest* subfolder in your MATLAB workspace. Next issue the command: *CertTest*. This command performs MBC and CCE operations and compares the outputs resulting from these operations with those in the provided *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 may use FAST to generate state matrices for a turbine of your interest and follow steps described in Section 5 to perform MBC or CCE or both.

8 Future Plans

Though the MBC formulation developed in this paper assumes a three-bladed turbine, the formulation is general enough to be readily extendable to a turbine with more than three blades. In future, depending on demand by the users, we will update both the theory basis and the scripts to accommodate rotors with three or more blades. Note that we will exclude two-bladed rotors because, even though an MBC-like transformation is theoretically possible for a two-bladed rotor; such a transformation will not provide any of the benefits listed in Section 1.

As mentioned earlier, the current MBC scripts work with FAST only. The future versions will accept data (state matrices, etc) 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 application of MBC to rotor with dissimilar blades.

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 website. DOE supported this work under contract number DE-AC36-83CH10093.

References

1. Bir, G. S., "Multi-Blade Coordinate Transformation and Its Application to Wind Turbine Analysis," Proceedings of the AIAA/ASME Wind Energy Symposium, Reno, January 2008.
2. Jonkman, J.M.; Buhl Jr., M.L. (2005). *FAST User's Guide*. NREL/EL-500-29798. Golden, CO: National Renewable Energy Laboratory.
3. Johnson, W., *Helicopter Theory*. Princeton University Press, New Jersey, 1980.
4. Miller, R. H., "Helicopter control and stability in hovering flight," *JAS* , 15:8, August 1948.
5. Coleman, R. P. and Feingold, A. M., "Theory of self-excited mechanical oscillations of helicopter rotors with hinged blades," NACA Technical Report TR 1351, 1958.
6. Hohenemser, K. H. and Yin, S. K., "Some applications of the method of multiblade coordinates," *Journal of the American Helicopter Society* , Vol. 17, No. 3, July 1972.
7. Bir, G., Chopra, I., et al. "University of Maryland Advanced Rotor Code (UMARC) Theory Manual," Technical Report UM-AERO 94-18, Center for Rotorcraft Education and Research, University of Maryland, College Park, July 1994.
8. Bir, G.S., Wright, A.D. and Butterfield, S., "Stability Analysis of Variable-Speed Wind Turbines", Proceedings of the 1997 ASME Wind Energy Symposium, Reno, January 6-9, 1997.
9. Malcolm, D. J., "Modal Response of 3-Bladed Wind turbines," *Journal of Solar Engineering*, Vol. 124, No. 4, Nov. 2002, pp. 372-377.
10. Elliott, A.S.; McConville, J.B. (1989). "Application of a General-Purpose Mechanical Systems Analysis Code to Rotorcraft Dynamics Problems." Presented at the American Helicopter Society National Specialists' Meeting on Rotorcraft Dynamics, 1989.
11. McCoy, T. J., "Wind turbine ADAMS model linearization including rotational and aerodynamic effects," 2004 ASME Wind Energy Symposium, Jan. 2004, pp. 224-233.
12. Hansen, M. H., "Improved modal dynamics of wind turbines to avoid stall-induced vibrations," *Wind Energy*, Vol. 6, Issue 2, 2003, pp.179-195.
13. Hansen, M. H., "Stability analysis of three-bladed turbines using an eigenvalue approach," 2004 ASME Wind Energy Symposium, Jan. 2004, pp. 192-202.
14. Riziotis, V. A., Voutsinas, S. G., Politis, E. S., and Chaviaropoulos, P. K., "Aeroelastic stability of wind turbines: The problem, the methods, and the issues," *Wind Energy*, Vol. 7, Issue 4, 2004, pp.373-392.
15. Bir, G. and Jonkman, J., "Aeroelastic instabilities of large offshore and onshore wind turbines," *Journal of Physics: Conference Series*, 2007.