CONTRACTOR REPORT

SAND98–0668
Unlimited Release
UC–1213

An Approach to the Development and Analysis of Wind Turbine Control Algorithms

Kung Chris Wu
Santa Clara University
Department of Mechanical Engineering
Santa Clara, CA 95053

Prepared by Sandia National Laboratories Albuquerque, New Mexico 87185
and Livermore, California 94550 for the United States Department of Energy
under Contract DE-AC04-94AL85000

Printed March 1998

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

Distribution
Category UC-1213

SAND 98-0668
Unlimited Release
Printed March 1998

An Approach to the Development and Analysis of Wind Turbine Control Algorithms

Kung Chris Wu
Santa Clara University
Department of Mechanical Engineering
Santa Clara, CA 95053

Sandia Contract AS-0985

ABSTRACT

The objective of this project is to develop the capability of symbolically generating an analytical model of a wind turbine for studies of control systems. This report focuses on a theoretical formulation of the symbolic equations of motion (EOMs) modeler for horizontal-axis wind turbines. In addition to the power train dynamics, a generic 7-axis rotor assembly is used as the base model from which the EOMs of various turbine configurations can be derived. A systematic approach to generate the EOMs is presented using d'Alembert's principle and Lagrangian dynamics. A Matlab M file was implemented to generate the EOMs of a two-bladed, free-yaw wind turbine. The EOMs will be compared in the future to those of a similar wind turbine modeled with the Yawdyn code for verification. This project was sponsored by Sandia National Laboratories as part of the Adaptive Structures and Control Task. This is the final report of Sandia Contract AS-0985.

Acknowledgments

The author wishes to thank Dr. P.S. Veers for his technical guidance throughout this contract, and his patience and help in preparing this report. This project was executed at Santa Clara University with funding from Sandia National Laboratories under contract number AS-0985.

Table of Contents

1.0 Introduction

Wind turbine control systems for power regulation and structural load mitigation are commonly treated as two decoupled and distinct problems. Much of the literature on wind turbine controls makes use of simplified turbine models to avoid the complexity of turbine dynamics. Turbine models without power train dynamics, for example, are typically used to deal with aero-elastic control systems [Block and Gilliatt 1997]. The design of variable-speed control algorithms, on the other hand, is generally done using only power train models [Thiringer and Linders 1993, Leithead et al. 1994a, Cardenas et al. 1996]. Consequently, the performance of control algorithms designed with "improper" models is questionable. Without an adequate dynamics model, it is not surprising to see that a "properly" designed control algorithm can actually be the major contributor to excessive fatigue loads in the turbine.

In the commercial sector, it is a common practice to design and analyze wind turbine control algorithms by coding proprietary programs. Unless collaborated efforts are organized, modeling efforts for turbines of similar configurations are repeated at different companies. In addition, repeated modeling efforts are needed to explore different design options for a new or existing turbine. For instance, to evaluate the benefits versus disadvantages of adopting a full span pitch or a partial span pitch (aileron) controller for a particular wind turbine, two programs have to be coded for computer analysis. This problem can be illustrated by NREL's recent effort to modify the FAST code to integrate the dynamics of an aileron control system into the structure code [Wright 1995, Stuart 1996]. Modeling wind turbines and verification of these models demand tremendous amounts of resources from the wind turbine industry. Consequently, simplified EOMs, e.g., only the EOMs of the subsystems to be studied, are generally used, and they might suffer from loss of accuracy and sometimes yield unreliable designs.

From a control engineering point of view, many advanced control methodologies (such as the robust adaptive controllers and neural-network fuzzy logic controllers) might provide better performance for wind turbine applications than the commonly used proportional-integral-derivative (PID) controllers [Bongers and Dijkstra 1992, Wu and De La Guardia 1996]. Many of the structural instabilities and load problems of contemporary wind turbines might be alleviated, or even avoided, if a different control methodology were used. For example, some European-designed constant-speed wind turbines (for 50Hz line frequency) installed in US wind farms (with 60Hz line frequency) might perform much better, in terms of fatigue life and power production, if they were operated as variable-

speed wind turbines. For variable-speed wind turbines and a new generation of continuously controlled machines, the control problems become multi-variable and much more complex than those of constant-speed wind turbines. Instability caused by improper use of multi-variable controllers might result in catastrophic failures. Careful study and full understanding of turbine control systems are crucial to the economics and reliability of these machines.

Two structural codes, ADAMS/WT and BLADED [Elliott 1996, Garrad 1996], for wind turbine design and analysis are capable of integrating limited control laws into their computer models. However, both codes provide only time domain simulations of the performance of the controllers. They do not offer linear analysis tools, such as transfer function, root locus, eigenvector analysis, stability margins, and power spectrum analysis, etc., that are critical to understanding the interactions between control systems and loads in wind turbines. Consequently, these codes are not adequate for control system design.

The objective of this project is to develop the capability of generating symbolically the analytical model of a wind turbine for the development and analysis of wind turbine control algorithms. In addition to the power train dynamics, a 7-axis rotor assembly is used in this report as the base model from which the EOMs of various turbine configurations can be derived. The future goal of this project is to develop a turnkey wind turbine modeler or an add-on package to a commercial software package like Matlab. The wind turbine modeling software will include a wind turbine configuration builder, a symbolic EOM builder, and interfaces to other engineering software. Users can create and modify interactively a wind turbine model by editing the parameter files describing the turbine configurations. The turbine modeler will then construct automatically the closed-form EOMs of the machine. The symbolic EOM builder generates explicit nonlinear differential equations rather than linear state matrices. Linear state matrices can be generated from the nonlinear EOMs analytically or numerically using Taylor series expansion and perturbation methods [Balafoutis 1991, Siljak 1969, Ogata 1992]. The closed-form EOMs can be extremely informative when studying wind turbine performance at unusual operating conditions, such as the system's behavior when the turbine is at a high yaw angle and yaw rate (where the inertial forces might dominate the turbine performance). Understanding control stability and reliability at extreme operating conditions is critical to the approval of wind turbine certifications, e.g., high yaw operations are part of the International Electrotechnical Commission standard test cases. A Matlab M file was implemented to generate the EOMs of a free-yaw, two-bladed, teetered-rotor wind turbine.

This report focuses on the mathematical formulation of the turbine modeler. It is organized into six chapters. Chapter 2 discusses the basic components of a horizontal axis wind turbine (HAWT) and how various turbine configurations with different control options can be constructed from the basic components. Chapters 3 and 4 discuss the mathematical formulation of the EOM builder. The EOMs of a rotating blade and a two-bladed teetered-rotor wind turbine are generated using the mathematical formulation. Chapter 5 presents the computer programs and the EOMs of these two examples. A summary and conclusions are given in the final chapter.

2.0 Constructing a Horizontal Axis Wind Turbine (HAWT) Model

To accommodate different machine configurations, the wind turbine modeler should be designed in a modular manner. A library of turbine components should be provided in the modeler. Wind turbines can be easily modeled by mixing components from the component library. Figure 1 depicts the block diagram of a HAWT model that includes a tower, rotor, power train, blades, aerodynamics forces, and various control modules. The arrows and their directions in the block diagram represent the inputs and outputs of each component module. The tower and the blades can be modeled either as continuously flexible beams or a collection of discrete elements depending on the modeling requirements. The term "rotor" is used in this report to represent collectively the nacelle assembly, hub, and teeter mechanism. The rotor is typically much stiffer than the tower and the blades, so it is modeled as a collection of discrete rigid bodies in this study. The power train consists of a combination of low- and high-speed shafts, gear boxes, brakes, and generators, depending on its configuration. Mechanical brakes at the low- and high-speed shafts can be modeled as external torques applied to the azimuth axis at the rotor side and the generator side, respectively. Equal but opposite moments will also be added to the component, e.g., the nacelle, where the brakes are mounted. Figure 2 shows a HAWT model consisting of nine basic components: namely, the tower (1), bed plate (2), nacelle (3), low-speed shaft (4), gearbox (5), high speed shaft (6), generator (7), hub (8), and blades (9). Notice that these turbine components can be viewed as components connected in series as an open-chain kinematic mechanism, see Figure 3, and only rotational motion is observed between any two adjacent components. To facilitate a systematic approach that can be programmed to generate the EOMs symbolically, all components in a wind turbine model are constrained to have no more than one degree of freedom (1-DOF) in motion with respect to the component upon which it is mounted (called the lower neighbor). For a turbine component having more than 1-DOF relative to its lower neighbor, it is modeled as a collection of several 1-DOF components. Examples are the wind turbine towers and blades which can pitch, yaw, and roll with respect to their lower neighbors, i.e., the ground and hubs, respectively.

2.1 The Tower

A wind turbine tower can be modeled as a continuously flexible beam or as multiple sections of discrete rigid bodies jointed by springs and dampers. The spring and damper represent the structural stiffness and damping of the tower, respectively. The mathematical formulation discussed in chapters 3 and 4 supports a discrete tower model. A continuous

flexible tower model can be found in the book by Junkins and Kim [1993]. The aerodynamic forces, applied directly to the tower and transmitted from the rotor, and the inertial forces due to turbine motion are considered as external forces driving the tower motion, i.e., the inputs to the tower module as shown in Figure 1. The position and velocity at the top of the tower, to where the positions of the rest of the turbine components are referenced, are thus the outputs of the tower model.

A discrete tower model consists of a number of rigid bodies connected in sequence. Each tower segment can have 3 rotational DOF in motion with respect to its lower neighbor. To comply with the 1-DOF relative motion between any two adjacent turbine components as discussed previously, a 3-DOF tower segment is represented by three 1-DOF rotational joints (see Figure 4). The rotation axes of the three mutually perpendicular joints intersect at a point. It is the center of rotation of the combined "3- DOF" joint. Notice that if three 1-DOF joints are used to simulate a 3-DOF joint, the rotation angles of the 1-DOF joints must be limited to small angles, typically less than 15 degrees, to ensure the accuracy of the model. Because of this small angle limitation, a softer tower must be modeled with a larger number of rigid tower segments for better modeling accuracy. Another factor determining the number of rigid tower segments is the number of mode shapes that need to be included in the tower model. Nevertheless, details of the tower model are outside the scope of this study.

2.2 The Power Train

The power train of a wind turbine consists of a combination of low and high speed shafts, brakes, gearboxes, and generators. Since the brakes are modeled as external torques instead of physical devices in this study, they are discussed in Chapter 4. Some wind turbines utilize coupling devices to provide compliance to sharp impacts in the drive train. However, coupling devices are not commonly used in utility-scale wind turbines because of their poor efficiencies. They are not discussed in this report.

2.2.1 The Drive Train Dynamics

The drive train dynamics relate the generator torque to the mechanical torque produced by the wind. The drive train in a wind turbine can be modeled either as a solid drive train or as two flexible shafts joined by the gearbox, depending on the application. For a constant-speed wind turbine, the solid drive train is an appropriate assumption. However, flexible shafts are necessary in the analysis of variable-speed operations [Hinrichsen and

Nolan 1982, Leithead et al. 1993]. To simplify the drive train dynamics, gears and shafts in the gearbox are assumed to be rigid bodies. Their mass moment of inertia and damping are lumped into that of the low-speed shaft [Palm 1986].

A. First-order Drive Train Model

The solid drive train assumption yields a simple first-order drive train model. The dynamics of the model in the Laplace domain are given by,

$$\frac{T_m}{n} - T_g = s \times \left(\frac{I_l}{n^2} + I_h \right) \times \omega_g \quad , \tag{1}$$

where

T_m is the mechanical torque applied to the low-speed shaft by the rotating rotor,

T_g is the electrical torque applied to the high-speed shaft by the generator,

n is the gearbox ratio,

s is the Laplace operator,

I_l and I_h are the mass moment of inertia of the low- and high-speed shaft about the rotation axis, respectively, and

ω_g is the generator speed. The rotor speed is ω_g /n.

B. Flexible Drive Train Model

With flexible shafts, the drive train dynamics become a third-order system. The EOMs can be written as, (see Figure 5)

$$\begin{Bmatrix} T_m \\ T_g \\ 0 \end{Bmatrix} = \begin{bmatrix} I_l s + b_l + \frac{k_l}{s} & -\frac{k_l}{s} & 0 \\ 0 & \frac{n k_h}{s} & -(I_h s + b_h + \frac{k_h}{s}) \\ \frac{k_l}{s} & -\frac{1}{s}(k_l + n^2 k_h) & \frac{k_h n}{s} \end{bmatrix} \begin{Bmatrix} \omega_l \\ \omega_h \\ \omega_g \end{Bmatrix} \quad , \tag{2}$$

where

k_l and k_h are the torsional stiffness of the low- and high-speed shafts, respectively

b_l and b_h are the viscous friction coefficients of the low- and high-speed shafts, respectively

ω_l is the rotor speed,

ω_h is the angular velocity of the low-speed shaft at the gearbox side, and
ω_g is the generator speed.

2.2.2 Induction Generators

There are many articles discussing ways to model generator behavior [Fitzgerald et al. 1990, Leithead and Connor 1994b, Zhang 1994]. Three induction-generator models, a steady- state, a first-order, and a fourth-order model, are summarized in this report. All models are for squirrel-type induction motors whose torque is a function of slip. The slip S relates the generator speed ω_g to the grid frequency ω_0, i.e.,

$$S = 1 - \frac{\omega_g}{\omega_0} \quad . \tag{3}$$

The utility grid is assumed to be a strong network in deriving these models. Since the generator dynamics are dictated by the drive-train dynamics [Hinrichsen and Nolan 1982, Wu 1997] in a strong network, a low-order generator model is adequate for quasi-static turbine analysis and control purposes. However, high-order generator models are necessary in the analysis of startup and shutdown operations, grid failure, power quality issues, and advanced generator features, such as active slip controls.

A. Steady-State Model

The steady-state model based upon the Thevenin equivalent circuit is the most commonly seen generator model [Fitzgerald et al., 1990]. This model yields the following algebraic equation describing the generator torque as a function of slip:

$$T_g = -\frac{p}{2\omega_0} \frac{3X_m^2 R_2 S V^2}{\left[R_1 R_2 + S(X_m^2 - X_{11}X_{22})\right]^2 + (R_2 X_{11} + S R_1 X_{22})} \quad , \tag{4}$$

where
p is the number of generator poles,
$X_{11} = X_{L1} + X_m$,
$X_{22} = X_{L2} + X_m$,
ω_0 is the grid frequency in rad/sec,
R_1 and X_{L1} are the stator resistance and leakage reactance, respectively

R_2 and X_{L2} are the rotor resistance and leakage reactance, respectively
X_m is the magnetizing reactance,
S is the generator slip, (Equation 3) and
V is the line voltage at the generator bus in phasor notation,

This method does not take into account any non-linearities in the generator and contains no dynamic information, e.g., the electromagnetic transients.

B. First-Order Model

The dynamic model for a first-order approximation of the generator torque is given by the equation

$$sT_g = \frac{1}{\tau}\left[T_g + D_e(\omega_g - \frac{\omega_0}{p})\right] \quad , \qquad (5)$$

where
τ is the generator time constant,
D_e is the slope of the generator torque/speed curve. It is determined from the static state model, i.e., Equation 4, and
s is the Laplace operator.

The first-order model contains some dynamic information about the generator, but assumes a linear relationship between the slip and the generator torque. If the drive train is assumed to be a solid shaft, this model alone does not provide much meaningful information compared to the steady-state model. However, if a flexible drive train model is used, this first- order approximation simulates some dynamic interactions, such as resonant frequencies, between the drive train and the generator. It is recommended that it be used with the flexible drive train model.

C. Electromagnetic Transient Model

To include electromagnetic transients in the power train model, the generator discussed in Zhang's report [Zhang 1994] is adopted in this study. The model bases its theory on Park's equation, which translates the three-phase voltage and current references and rotating components into a stationary quadrature and direct-axis frame, i.e., the d-q frame. That is,

$$\begin{bmatrix} v_{q1} \\ v_{d1} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{-\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \end{bmatrix} \begin{bmatrix} v_{a1} \\ v_{b1} \\ v_{c1} \end{bmatrix} \quad , \tag{6}$$

where

v_{q1} and v_{d1} are the line voltages in the q-d frame, and

v_{a1}, v_{b1}, and v_{c1} are the 3-phase line to line voltages in phasor notation.

The derived equations for the generator torque are fourth order and reduce to the following form,

$$\begin{aligned}
\frac{d\psi_{q1}}{dt} &= x'_1 = \omega_0 v_{q1} + \left[\left(\frac{\omega_0 R_1}{X_{L1}} \right) \left(\frac{X_{qd}}{X_{L1}} - 1 \right) \right] x_1 + \left(\frac{\omega_0 R_1 X_{qd}}{X_{L1} X_{L2}} \right) x_3 \\
\frac{d\psi_{d1}}{dt} &= x'_2 = \omega_0 v_{d1} + \left[\left(\frac{\omega_0 R_1}{X_{L1}} \right) \left(\frac{X_{qd}}{X_{L1}} - 1 \right) \right] x_2 + \left(\frac{\omega_0 R_1 X_{qd}}{X_{L1} X_{L2}} \right) x_4 \\
\frac{d\psi_{q2}}{dt} &= x'_3 = x_4 \omega_e + \left(\frac{\omega_0 R_2 X_{qd}}{X_{L1} X_{L2}} \right) x_1 + \left[\left(\frac{\omega_0 R_2 X_{qd}}{X_{L2}} \right) \left(\frac{X_{qd}}{X_{L2}} - 1 \right) \right] x_3 \\
\frac{d\psi_{d_2}}{dt} &= x'_4 = -x_3 \omega_e + \left(\frac{\omega_0 R_2 X_{qd}}{X_{L1} X_{L2}} \right) x_2 + \left[\left(\frac{\omega_0 R_2 X_{qd}}{X_{L2}} \right) \left(\frac{X_{qd}}{X_{L2}} - 1 \right) \right] x_4
\end{aligned} \tag{7}$$

where

ψ_{q1}, ψ_{d1}, ψ_{q2}, ψ_{d2}, are the flux linkages per second of the q-d stator and rotor windings

$$X_{qd} = \left[\frac{X_{L1} X_{L2} X_m}{X_{L1} X_m + X_{L2} X_m + X_{L1} X_{L2}} \right] \tag{8}$$

and

$$\omega_e = \frac{\omega_g p}{2} \tag{9}$$

where ω_e is the electrical speed of the generator and ω_g is the mechanical speed calculated from Equation 1 or 2.

The resulting generator torque T_g needed in Equations 1 and 2 is given by

$$T_g = \frac{3}{2}\frac{p}{2\omega_0}(\psi_{q1}i_{d1} - \psi_{d1}i_{q1}) \quad , \tag{10}$$

where i_{d1} and i_{q1} are the q-d currents in the stator winding and are given by

$$\begin{aligned} i_{q1} &= \frac{\psi_{q1}}{X_{L1}} - \frac{X_{qd}\psi_{q1}}{X_{L1}^2} - \frac{X_{qd}\psi_{q2}}{X_{L1}X_{L2}} \\ i_{d1} &= \frac{\psi_{d1}}{X_{L1}} - \frac{X_{qd}\psi_{d1}}{X_{L1}^2} - \frac{X_{qd}\psi_{d2}}{X_{L1}X_{L2}} \end{aligned} \quad . \tag{11}$$

The generator, modeled in Equation 4, 5 or 10, together with the drive train, modeled in Equation 1 or 2, form the power train model for a constant-speed wind turbine. The power output to the utility grid is given by

$$P_g = T_g\omega_g \quad . \tag{12}$$

For a power train with an electromagnetic transient generator model, the power output to the utility grid is given by,

$$P_g = -\frac{3}{2}(v_{q1}i_{q1} + v_{d1}i_{d1}) \quad . \tag{13}$$

If the generator efficiency is known, the power output can be adjusted accordingly using a lookup table.

2.2.3 Power Electronics for Variable-Speed Wind Turbines

When a power electronics (PE) unit is used to regulate the generator torque (and the power output), the generator and the PE together can be considered as a torque actuator. In addition, the response of the torque actuator is much faster than that of the drive train, typically an order of magnitude. The torque actuator can be modeled as an ideal one,

which applies the desired (or commanded) torque to the high-speed shaft precisely and instantaneously, i.e., (see Figure 1)

$$T_r = T_g \quad . \tag{14}$$

2.2.4 Geometric Location of the Power Train

Some wind turbine configurations incorporate a rotor offset (a non-zero *c* distance as shown in Figure 2) and an uptilt angle (a constant angle about the z_3 axis) to aerodynamically balance the rotor assembly and increase the blade-tip and tower clearance. The *a* and *b* parameters, which are measured along the x_1 and z_1 axes, respectively, locate the tilt axis z_3 of the bed plate. Notice that the x_1-z_1 frame is located at the top of the tower, whose instantaneous position and velocity are determined from the tower model. Figure 2 shows positive *a* and *b* values. The geometry parameter *c* represents the offset of the low-speed shaft from the yaw axis. It is measured along the y_1 axis. Figure 2 shows a positive *c* value. To model a rotor without an offset, *c* is set to zero. Parameter *d* represents a vertical offset of the low-speed shaft from the tilt axis. It is measured along the z_1 axis. Figure 2 shows a positive *d* value. The sum of the *b* and *d* values represents the clearance distance between the low-speed shaft and the base of the nacelle. For a wind turbine with a constant tilt angle, the sense of the rotation is measured about the z_3 axis by following the right-hand rule. A zero tilt angle places the low-speed shaft parallel to the nacelle. A positive tilt angle represents an up-tilt configuration, and a negative value represents a down-tilt configuration.

2.3. The Rotor Dynamics

The rotor model, see Figure 1, studied in this report includes the nacelle assembly, hub, and teeter mechanism. It does not include blades here because blades are considered as independent wind turbine components in this report. The rotor dynamics are the most complex among all wind turbine subsystems because the rotor can have many DOFs relative to its adjacent components. It relates the aerodynamics forces to the motion and loads in the rotor. The rotor studied in this report includes seven rotation axes, as shown in Figure 2. The nacelle has yaw and tilt motion relative to the tower; the hub has azimuth and teeter motion relative to the nacelle; and the blade root of each blade can have three DOF relative to the hub. The following sections describe the geometric parameters of the rotor assembly. Derivations of the EOMs of the rotor are discussed in Chapters 3 and 4. A

detailed drawing of the rotor assembly is given in Figure 6.

2.3.1 The Bed Plate and the Nacelle

The bed plate can yaw about the z_2 axis with respect to the tower, and the nacelle can tilt (or pitch) about the z_3 axis with respect to the bed plate, see Figure 6. Notice that the bed plate and the nacelle are generally considered as one component (called nacelle) in most wind turbine models that allow 2-DOF components. However, to conform to the constraints of one DOF per component in our models, the bed plate is introduced to accommodate the yaw motion, and the nacelle that houses the drive train allows tilt motion with respect to the bed plate.

2.3.2 The Hub, Teeter Axis, Overhang, and Undersling

For a two-bladed rotor, the hub assembly may have one rotational DOF, i.e., the teeter motion, about the z_5 axis with respect to the low speed shaft. The teeter motion allows the blades to flap and/or pitch to remove or reduce gyroscopic forces and bending moments exerted on the low speed shaft. The angle between the teeter axis and the line perpendicular to the spanwise axis of the blades is known as the δ_3 angle (see Figure 7). The δ_3 angle is measured about the z_4 axis in the right-hand sense to align the teeter (z_5) axis to the x_4 axis, i.e., parallel to z_0 as in the configuration shown in Figure 6. Figure 6 shows a negative δ_3 angle, and Figure 7 shows a positive δ_3 angle. Notice that the direction of z_4 is defined so that the rotation of the rotor follows the right-hand rule. When $\delta_3 = 0$, the teeter motion causes the blades to flap in and out of the rotation plane. On the other hand, the teeter motion causes the blades to pitch when $\delta_3 = 90°$. If δ_3 is between 0 and 90°, the teeter motion causes the blades to flap and pitch simultaneously.

The location of the teeter pin (or the intersection point of the teeter axis and the low-speed shaft) is determined by the overhang and undersling distances. The terms overhang and undersling do not appear to have clear definitions in the wind turbine industry, regardless of their popularity in the literature. In this study, the term overhang *(e* shown in Figure 6*)* is used as a measure of the distance along the low speed-shaft (Z_4 axis) from the point T immediately below the tilt axis to the intersection point *s* of the span axes of the blades at a zero coning angle. In general, a long overhang gives more geometric clearance between the blade-tip and the tower. It also introduces a greater bending and yaw moment to the low-speed shaft and tower due to gravity and aerodynamic forces.

Although it is not a standard, undersling is defined as the distance from the intersection point of the span axes of the blades (as the *s* point in Figure 6) to the teeter pin in some literatures. In practice, the intersection point dynamically changes due to the effect of varying aerodynamic blade loads. The *s'* point in Figure 6 illustrates the intersection point at some pre-cone angle. For precision, the term undersling *(f)* is used here as a measure of the distance along the low-speed shaft from the point *s*, i.e., the intersection point of the span axes of the blades at a zero coning angle, to the teeter pin. Undersling can have either a positive or negative value (measured along the z_4 axis), depending on a downwind or upwind configuration. Figure 6 shows a positive *f* value. An aerodynamically balanced rotor will place the teeter pin coincident with the center of gravity of the rotor (CGR) assembly to reduce the effect of the Coriolis force transmitted to the low-speed shaft. Since the CGR dynamically shifts due to the flexibility of the blades during operation, a good rotor design will place the teeter pin in the center of the region where the CGR is dynamically moving.

The points at which the blades are attached to the hub are located by the parameter *g* measured from the point S to the blade root. The parameter *g* is also known as the hub radius and always has a positive value.

2.4 Blades

Similar to the tower model, a blade can be modeled as a continuously flexible beam or a collection of rigid segments jointed in sequence. A blade has 3-DOF, i.e., pitch, flapping, and lead-lag bending at the root with respect to the hub. Like a flexible tower, the 3-DOF blade root is modeled as three mutually perpendicular 1-DOF joints intersecting at the blade root point, see Figure 6. The blade feather has 1 DOF, i.e., the pitch motion about the z_6 axis with respect to the hub. The blade coning has 1 DOF, i.e., the coning motion about the z_7 axis with respect to the feather. The blade torsion has 1 DOF, i.e., the edgewise lead-lag bending motion about the z_8 axis with respect to the blade coning. Springs and dampers can be attached to any of the three axes to model the structural stiffness and damping at the blade roots. In this study, both the blades and tower are modeled in separate modules from the rotor assembly. The rotor model outputs the positions and velocities at the blade roots to the blade models, which return the aerodynamics forces at the blade roots, or at prescribed geometric locations measured from the blade roots. Interactions between the blade models and the rotor model are explored more in Chapter 4.

2.5 Deviations From the Example Model in Other Wind Turbine Configurations

The relative motion of the turbine components with respect to their neighboring components can be inhibited by locking the corresponding axes to model wind turbines of different configurations. For example, by locking the tilt axis, teeter axis, coning axis, and pitch axis, a typical 2-bladed or 3-bladed rigid-rotor, stall-regulated, free-yaw wind turbine can be modeled. A teetered-rotor configuration can be obtained from the same model by freeing the motion about the teeter axis. Freeing the pitch axis simulates full or partial pitch motion. Although the example model in Figure 2 consists of only nine basic components, combinations of these basic components yield virtually unlimited turbine configurations.

2.6 Driving Forces

Aerodynamic, actuator (or control), and external forces can be added to any of the rotation axes described above to model springs, dampers, brakes, and viscous and coulomb friction. For example, by adding spring and damping forces to the blade axes, constrained motion of the blades is modeled to simulate coning, in-plane and out-of-plane bending. Active controlled wind turbine configurations can be modeled by adding control dynamics to any of the rotation axes. For example, an active yaw or pitch-controlled wind turbine can be studied by adding control dynamics to the yaw or pitch axis. By adding control dynamics to tilt and/or coning axes, advanced wind turbine configurations with active up/down tilt and coning control systems can be simulated. The controllers incorporated in the model can be referenced collectively, i.e., same pitch angles for all the blades, or independently to yield optimal modeling flexibility of the turbine modeler.

3.0 Kinematics of the Rotor Assembly of a HAWT

3.1 Kinematics of Structural Components

By placing the kinematic constraints of 1 DOF between any pair of adjacent components, the wind turbine model described in Section 2 can be viewed as an open-chain kinematic linkage where the tower is rigidly attached to earth (or the inertial frame) and the upper bodies of the linkage (or the blades) are free to move in space, as illustrated in Figure 3. For the remainder of this report, the term "link" is used as a generic name for any one of the turbine components. With this arrangement the kinematic relationship of the neighboring bodies can be systematically established by assigning a body-attached orthonormal coordinate system to each of the links of the wind turbine using the Denavit-Hartenberg (D-H) convention [Denavit and Hartenberg 1955]. This convention allows the spatial descriptions, i.e., kinematics, of the wind turbine to be established by following a mechanical procedure that is independent of turbine configurations.

Each set of coordinates, (x_i, y_i, z_i) denoted by $\{i\}$, or called Frame $\{i\}$, is rigidly attached to the body (i) and rotates along with the body, and therefore is called a "body-attached" coordinate system. The z-axes of the frames are always aligned with the rotation axes of the joints (see Figure 6). If the link is unidirectional such as the blades, the direction of the link rotation is determined by the z-axis in the right hand sense. After establishing the Cartesian frames, four parameters (α_i , b_i , θ_i , d_i), namely, the twist angle, the link length, the rotation angle, and the link offset, respectively, are determined for each body frame $\{i\}$. Definitions of these parameters are given as follows,

1. α_i is the angle of rotation about x_{i-1} in the right-hand sense to align z_{i-1} to z_i. Notice that x_{i-1}-z_{i-1} frame, or $\{i-1\}$ frame, is assigned to link i-1, the lower neighbor of link i.
2. b_i is the distance along x_{i-1} ,from the joint axis z_{i-1}, to joint axis z_i.
3. θ_i is the angle of rotation about z_i in the right-hand sense to align x_{i-1} to x_i,
4. d_i is the distance along z_i from the intersection point of x_{i-1} and z_i to the origin of the frame $\{i\}$.

Details and examples of the link parameters can be found in many kinematics and robotics books [Craig 1989, Balafoutis and Patel 1991]. The D-H link parameters for the two-bladed, teetered rotor HAWT shown in Figure 6 are given in Table 1.

The configuration, or spatial transformation, matrix describing the kinematic relationship between link i and its lower neighbor (link i-1) is given by

$$^{i-1}C_i = {}^{i-1}R_i \; {}^{i}C_i^* + {}^{i-1}P_i \quad , \tag{15}$$

where

$^{i-1}C_i$ is a 3x1 position vector of a point C fixed on link i (denoted by the lower right subscript in our notation) in a 3-D space measured and expressed with respect to Frame {i-1} (denoted by the upper left superscript). That is, it is a vector from the origin of Frame {i-1} to the point C on link i, expressed with respect to Frame {i-1},

$$^{i-1}R_i = \begin{vmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) \end{vmatrix} \quad . \tag{16}$$

$^{i-1}C^*_I$ is a position vector of a point C fixed in link i measured in Frame {i}, expressed with respect to Frame {i-1}. That is, it is a vector from the origin of Frame {i} to the point C on link i. It is a constant vector since Frame {i} is fixed to link i. Notice that the upper right superscript * is used for the vectors that are measured with respect to their own body frames.

Let

$$^{i-1}P_i = [\, b_i \;\; -d_i \sin(\alpha i) \;\; d_i \cos(\alpha_i) \,]^t \tag{17}$$

be the position vector of the origin of Frame {i} measured and expressed with respect to Frame {i-1}. Notice that the symbol C is used for an arbitrary point on a link, e.g., the center of mass, and the symbol P is used for the origin of a body frame in our notation (see Figure 8).

After the user specifies the geometric configurations, (i.e., the values of the parameters b_i , d_i, δ_3, and τ_i), and the DOFs of a turbine, (i.e., specifies whether θ_i is a constant or a variable), kinematic relationships among the turbine components can be derived systematically by substituting the D-H link parameters as shown in Table 1 into Equations 15 through 17.

Table 1 D-H link parameters for the HAWT model in Figure 6.

Link	α_i	b_i	θ_i	d_i	Link Name (Remark)
1					Reference frame at the tower top
2	0	0	q_2	b	Yaw
3	90°	a	$-90°+q_3$	$-c$	Tilt
4	-90°	d	$-90-\delta_3+q_4$	$e+f$	Azimuth
5	90°	0	$-90°+q_5$	0	Teeter
6	$90°-\delta_3$	f	$180°+q_6+\tau_1$	$-g$	Pitch + blade twist τ_1 at root (for blade 1)
7	-90°	0	$90°+q_7+q_0$	0	Coning (for blade 1), q_0 = precone
8	-90°	0	$90°+q_8$	0	Bending (for blade 1)
6'	$-90°-\delta_3$	f	$180°+q_{6'}+\tau_2$	$-g$	Pitch + blade twist τ_2 at root (for blade 2)
7'	-90°	0	$90°+q_{7'}+q_0$	0	Coning (for blade 2), q_0 = precone
8'	-90°	0	$90°+q_{8'}$	0	Bending (for blade 2)

3.2 Linear Velocity and Acceleration

The angular velocity ω_i of a link i with respect to the inertial frame can be expressed as a sum of the relative angular velocities of all lower links. That is,

$$\omega_i = \sum_{j=1}^{i} {}^0R_j \, {}^jq'_j = \sum_{j=1}^{i} q'_j \quad , \tag{18}$$

where

${}^jq'_j = [0\ 0\ q'_j]^t$, is the angular velocity of link j about joint axis j,

${}^0R_j = {}^0R_1 \, {}^1R_2 \, {}^2R_3 \, {}^3R_4 \cdots {}^{j-1}R_j$, and

$q'_j \triangleq {}^0q'_j = {}^0R_j \, {}^jq'_j$.

Notice that the upper-left superscript is omitted if it is 0 to simplify the notation. So z_j represents the unit vector of Frame {j} expressed with respect to the inertial frame. Let r_i be the position vector of the center of mass of link i (measured and expressed with respect to the inertial frame, see Figure 8). The position vector can be written as

$$r_i = \sum_{j=1}^{i} {}^0R_{j-1}\,{}^{j-1}P_j + {}^0R_i\,{}^iC_i^* = \sum_{j=1}^{i} P_j^* + C_i^* \quad , \tag{19}$$

where

$P_j^* = {}^0P_j^* = {}^0R_{j-1}\,{}^{j-1}P_j$, is the position vector of the origin of Frame {j} measured in Frame {j-1} expressed with respect to the inertial frame, and

$C_i^* = {}^0C_i^* = {}^0R_i\,{}^iC_i^*$, is the position vector of the center of mass of body i measured in Frame {i} expressed with respect to the inertial frame.

The linear velocity and acceleration of link i at its center of mass, observed in the inertial frame, can then be derived by taking first and second time derivatives of Equation 19.

$$v_i = \sum_{j=2}^{i} \left(\omega_{j-1} * P_j^*\right) + \omega_i * C_i^* \tag{20}$$

$$\begin{aligned} a_i = & \sum_{j=2}^{i} \left(Q''_{j-1} * P_j^* \right) \\ & + \sum_{j=3}^{i} \left[\sum_{k=2}^{j-1} \left(\omega_{k-1} * q'_k \right) * P_j^* \right] \\ & + \sum_{j=2}^{i} \left[\omega_{j-1} * \left(\omega_{j-1} * P_j^* \right) \right] \\ & + Q''_i * C_i^* \\ & + \sum_{j=2}^{i} \left(\omega_{j-1} * q'_j \right) * C_i^* \\ & + \omega_i * \left(\omega_i * C_i^* \right) \end{aligned} \tag{21}$$

where

$$Q''_i = \sum_{j=1}^{i} q''_j \tag{22}$$

and $q''_j \triangleq {}^0q''_j = {}^0R_j\,{}^jq''_j$, ${}^jq''_j = [0\ 0\ q''_j]^t$ is the angular acceleration of link j about joint axis j.

Although Equation 21 seems complicated, it is computationally efficient (using a linear algebra software package) and highly structured. The first and fourth terms in Equation 21 represent the linear acceleration components due to the angular acceleration of the links lower than i. The second and fifth terms are the linear acceleration components due to the centrifugal acceleration of the lower components. The third and sixth terms correspond to the Coriolis acceleration due to the linear motion of the lower links.

4.0 Dynamics of the Rotor Assembly of a HAWT

4.1 Structural Dynamics

In deriving the EOMs of a HAWT, the generalized d'Alembert (GD) formulation is applied. Recall the well-known Lagrange equation

$$La[L]_s = \frac{d}{dt}\left(\frac{\partial L}{\partial q'_s}\right) - \frac{\partial L}{\partial q_s} = \tau_s , \qquad s = 1,2,3,\ldots,n \tag{23}$$

where

$La(L)_s$ denotes the Lagrangian operator applied to a system L with respect to a generalized coordinate s,

L = Lagrangian function of the system L = total kinetic energy K - total potential energy V,

q_s = generalized coordinate at the joint s where the system is free to move

q'_s = first time derivative of the generalized coordinate, q_s

τ_s = effective force (or torque) applied to the system at joint s.

The d'Alembert principle [Ogata 1992] states that the external forces [or torques (τ)] applied to a multi-body system must be equal to the inertial forces [or torques (M)] of the system, measured with respect to the independent axes describing the degrees of freedom of the system, for the system to be in dynamic equilibrium. That is

$$\tau_s = M_s , \quad s = 1,2, \ldots ,n \quad . \tag{24}$$

The wind turbine components considered in this study only have rotational motion relative to their lower neighbors. Only the resultant torques at the joints will be considered for the EOMs of the system. The external torques considered in this study are the torques caused by the aerodynamic forces and moments T_{ae}, spring and damper torques T_{sp} and T_{da}, respectively, actuator torques T_{ac}, and torques transmitted from the upper links T_{fr}. Notice that the energy-stored springs and the energy-dissipating dampers and brakes are considered as external torques in this formulation. These mechanisms apply equal but opposite torques to two links of the turbine where they are mounted. These torques will be discussed more in detail later. The inertial torques are those caused by translational and rotational motion as results of external forces, and motion caused by gravitation force on

the individual turbine components, denoted by M_t , M_r , and M_g, respectively. Equation 24 can be rewritten as

$$M_{t_s} + M_{r_s} + M_{g_s} = T_{ae_s} + T_{sp_s} + T_{da_s} + T_{ac_s} + T_{fr_s}, \quad s = 1,2,3,\ldots,n \tag{25}$$

where the letter s represents the generalized coordinates describing the DOF of the system. Recall that

$$L = K - V = K_t + K_r - V \quad , \tag{26}$$

where

K_t is the total kinetic energy due to the transnational effect, i.e., the linear translation of all of the turbine components, and

K_r is the total kinetic energy due to the rotational effect, i.e., the angular rotation of all of the turbine components.

Substituting Equation 26 into Equation 23,

$$La[L]_s = La[K_t]_s + La[K_r]_s - La[V]_s = \tau_s \quad . \tag{27}$$

If Equation 25 is compared with Equation 27, one obtains

$$\tau_s = \left(T_{ae_s} + T_{sp_s} + T_{da_s} + T_{ac_s} + T_{fr_s}\right)_{effective}, \quad s = 1,2,3,\ldots,n \tag{28}$$

$$\begin{aligned} M_{t_s} &= La[K_t]_s \\ M_{r_s} &= La[K_r]_s \\ M_{g_s} &= -La[V]_s \quad . \end{aligned}$$

Substituting Equation 28 into Equation 25,

$$M_{t_s} + M_{r_s} + M_{g_s} = \tau_s \quad . \tag{29}$$

Equations 25 and 28 are two important relations that form the mathematical basis of the automatic EOM builder.

4.2 Inertial Torques

4.2.1 Inertial Torque Due to Translational Effect M_t

The kinetic energy of wind turbine component i, i=1,2,3, ..., n, with mass m_i, due to its translational motion can be expressed as:

$$K_{t_i} = \frac{1}{2} m_i \, v_i \cdot v_i \quad , \tag{30}$$

where v_i is the linear velocity of component i at its center of mass, see Equation 20. The total kinetic energy of the wind turbine due to translational motion can be written as a summation of the individual components. That is,

$$K_t = \frac{1}{2} \sum_{i=1}^{n} m_i \, v_i \cdot v_i \quad . \tag{31}$$

If the Lagrangian operator is applied to the above equation with respect to generalized coordinate s (see Equation 23), one obtains

$$M_{t_s} = \sum_{i=s}^{n} \left\{ m_i a_i \cdot \left[z_s * \left(r_i - P_s \right) \right] \right\} \quad , \tag{32}$$

where
(·) and (*) are the dot product and cross product of two vectors, respectively
$P_s = {}^0P_s = {}^0P_1 + {}^0R_1 \, {}^1P_2 + \cdots + {}^0R_{s-1} \, {}^{s-1}P_s$ is the position vector of the origin of Frame {s} measured and expressed with respect to the inertial frame, and
z_s is a unit vector along the generalized coordinate s measured with respect to the inertial frame.

4.2.2 Inertial Torque Due to Rotational Effect M_r

The rotational kinetic energy of wind turbine component i, with mass moment of inertia I_i about its center of mass expressed with respect to Frame {i}, is given by

$$k_{r_i} = \frac{1}{2}\left({}^{i}R_0 \omega_i \right)^t \; I_i \left({}^{i}R_0 \omega_i \right) \quad , \tag{33}$$

where ${}^{i}R_0 = ({}^{0}R_i)^t$.

The total kinetic energy of the wind turbine due to the rotational effect can then be written as a summation of the individual components. That is,

$$k_r = \frac{1}{2}\sum_{i=1}^{n} \left({}^{i}R_0 \omega_i \right)^t \; I_i \left({}^{i}R_0 \omega_i \right) \quad . \tag{34}$$

Substituting Equation 18 into Equation 34 and applying the Lagrangian operator to the resulting equation with respect to a generalized coordinate s leads to

$$\begin{aligned} M_{r_s} = \sum_{i=s}^{n} & \left\{ \left({}^{i}R_0 z_s \right)^t I_i \left({}^{i}R_0 Q_i'' \right) \right. \\ & + \left({}^{i}R_0 z_s \right)^t I_i \left\{ \sum_{j=1}^{i} {}^{i}R_0 \left[q'_j * (\omega_i - \omega_j) \right] \right\} \\ & \left. + {}^{i}R_0 \left(z_s * \omega_i \right)^t \; I_i \left({}^{i}R_0 \omega_i \right) \right\} \; , \quad j \le s \le i \end{aligned} \tag{35}$$

Notice that the first, second, and third terms in the above equation represent the inertial torques due to angular accelerations, centrifugal forces, and Coriolis forces of the rotating links, respectively.

4.2.3 Inertial Torque Due to Gravitation M_g

The total potential energy of a wind turbine can be expressed as

$$V = \sum_{i=1}^{n} -m_i \; g \cdot r_i \quad , \tag{36}$$

where g is the gravitational acceleration vector, e.g., $[0\ 0\ 9.81\ m/sec^2]^t$.

Applying the Lagrange operation to the above equation with respect to the generalized

coordinate s, leads to

$$M_{g_s} = \sum_{i=s}^{n} m_i \, g \cdot \left[z_s * \left(r_i - P_s \right) \right] \quad . \tag{37}$$

Details of the derivation of Equation 37 are outside the scope of this report.

4.3 External Loads and Torques

4.3.1Direct Torques Applied to links

A. Spring and Damping Torques

As shown in the free-body diagram depicted in Figure 9, the torques exerted on link s by a torsional spring and damper can be expressed as

$$T_{sp_s} = -k_s q_s z_s + k_{s+1} q_{s+1} z_{s+1} \tag{38}$$

$$T_{da_s} = -b_s q'_s z_s + b_{s+1} q'_{s+1} z_{s+1} \quad , \tag{39}$$

where
k_s is the torsional spring constant of the spring at joint s
b_s is the damping coefficient of the damper at joint s.

The first terms in Equations 38 and 39 are the spring and damping torques, respectively, due to the spring and damper connecting links s-1 and s. The second terms in Equations 38 and 39 are the spring and damping torques, respectively, due to the spring and damper connecting links s and s+1.

B. Actuator Torque (Controller torque)

Externally applied loads or torques can be modeled as

$$T_{ac_i} = f(q, q') z_i \quad , \tag{40}$$

where f represents an external forcing function or a function of the states of the wind

turbine defined by users, e.g., a braking force can be a function of the shaft speed. It can represent a mechanical brake or the dynamics of an active control system. If *f* represents a brake force, an equal but opposite moment will be applied to the link where the brake is mounted. If *f* is an actuator moment, the internally reacted actuator forces will be neglected, i.e., the reacted actuator forces are assumed to be absorbed by the structure.

C. Aerodynamic Forces and Moments

As shown in Figure 10, the resulting moment of the aerodynamic force and moment directly exerted on link s at the rotation axis can be calculated as

$$T_{ae_s} = M_{ae_s} + C_s^* \times F_s \quad , \tag{41}$$

where M_{ae_s} and F_s are the aerodynamic moment and force exerted on link s, respectively, (calculated using an external aerodynamic subroutine). The aerodynamic moments and forces are assumed to be point moments and forces acting through the center of gravity of each link.

4.3.2 Forces Transmitted From Upper Links to Lower Components

External forces, like the aerodynamic force F_i, exerted on link i will be partially consumed by the link, by converting to kinetic energy through link rotation. The rest of the forces are transmitted to its lower neighbors. Let

$$\zeta_n = F_n - (1 - \mu_n)\left[F_n - (F_n \cdot n_n)n_n - (F_n \cdot z_n)z_n\right] \tag{42}$$

be the force transmitted from the uppermost link, n, to its lower neighbor n-1, where n_n is a unit vector parallel to the projection of the vector C^*_n on the x-y plane of Frame {n}. That is

$$n_n = \frac{\left[C_n^* - (C_n^* \cdot z_n)z_n\right]}{\left|\left[C_n^* - (C_n^* \cdot z_n)z_n\right]\right|} \tag{43}$$

and μ_n is the joint coefficient of joint n. The joint is free if μ_n=0 and the joint is locked (e.g., mechanically locked) if μ_n=1.

In practice, some of the torque generated from external forces are consumed by friction in the joints. The frictional torque in a joint can be modeled as external torques as discussed in Section 4.3.1B. For convenience, the joint coefficients can be used as an alternative way to include friction loss in the model. Instead of using zero for the joint coefficient of a free joint, any value between zero and one can be used to model the friction loss at the joint. Nevertheless, one should be aware of the fact that joint coefficients are not friction coefficients. It is proposed here merely for mathematical convenience.

Similar to Equation 42, let

$$\begin{aligned}\zeta_{n-1} = F_{n-1} - (1-\mu_{n-1})\left[F_{n-1} - (F_{n-1}\cdot n_{n-1})n_{n-1} - (F_{n-1}\cdot z_{n-1})z_{n-1}\right] \\ + \zeta_n - (1-\mu_{n-1})\left[\zeta_n - (\zeta_n\cdot\phi_{n-1})\phi_{n-1} - (\zeta_n\cdot z_{n-1})z_{n-1}\right]\end{aligned} \tag{44}$$

be the force transmitted from link n-1 to its lower neighbor n-2, where ϕ_{n-1} is the unit vector parallel to the projected vector of P^*_n on the x-y plane in Frame {n-1}. That is

$$\phi_{n-1} = \frac{\left[P_n^* - (P_n^*\cdot z_{n-1})z_{n-1}\right]}{\left|\left[P_n^* - (P_n^*\cdot z_{n-1})z_{n-1}\right]\right|} \tag{45}$$

By induction, the force transmitted to link s-1 from link s can be written as

$$\zeta_s = (F_s + \zeta_{s+1}) - (1-\mu_s)\{(F_s + \zeta_{s+1}) - (F_s\cdot n_s)n_s - (\zeta_{s+1}\cdot\phi_s)\phi_s - [(F_s + \zeta_{s+1})\cdot z_s]z_s\} \tag{46}$$

In a typical wind turbine model, aerodynamic forces are calculated and applied to the blades. Equation (46) can be used to calculate the aerodynamic forces transmitted from the blades to the rest of the turbine components.

4.3.3 Moments Transmitted From Upper Links to Lower Component and the Effective Torques

Like aerodynamic forces, moments exerted on link i will be partially converted to kinetic energy by link i rotation. The rest of the moment is transmitted to its lower neighbor i-1. Let T_n be the total direct moment exerted on the last link n, i.e., the outermost link n, excluding the externally applied torque. That is,

$$T_n = T_{sp_n} + T_{da_n} + T_{ae_n} \quad . \tag{47}$$

The effective torque that causes the link to rotate about its axis is

$$\tau_n = T_n \cdot z_n + T_{ac_n} \quad , \tag{48}$$

where z_n is a unit vector along the direction of the joint axis n. The portion of the moment transmitted to its lower neighbor n-1 is

$$T'_n = T_n - (1-\mu_n)[(T_n \cdot z_n) z_n] \quad . \tag{49}$$

The total moment, excluding the actuator moment, exerted on link n-1 is

$$T_{n-1} = T'_n + P_n^* \times \zeta_n + T_{sp_{n-1}} + T_{da_{n-1}} + T_{ae_{n-1}} \quad . \tag{50}$$

The effective torque at joint axis n-1 is

$$\tau_{n-1} = T_{n-1} \cdot z_{n-1} + T_{ac_{n-1}} \tag{51}$$

and

$$T'_{n-1} = T_{n-1} - (1-\mu_{n-1})[(T_{n-1} \cdot z_{n-1}) z_{n-1}] \quad . \tag{52}$$

By induction, the total moment exerted on link s and its effective torque can be derived as

$$T_s = T'_{s+1} + P_{s+1}^* \times \zeta_{s+1} + T_{sp_s} + T_{da_s} + T_{ae_s} \tag{53}$$

$$\tau_s = T_s \cdot z_s + T_{ac_s} \tag{54}$$

$$T'_s = T_s - (1-\mu_s)[(T_s \cdot z_s) z_s] \quad . \tag{55}$$

Equation (53) calculated the aerodynamic torques transmitted from the blades to the rest of the turbine components.

4.4 EOM Builder and a Recursive Algorithm to Calculate the Effective Torque

The equations derived in the above sections are totally generic to the turbine structures. The inertial components of the EOM, i.e., the left-hand side of Equation 26, of a wind turbine can be derived symbolically using Equations 25, 32, 35, and 37. The effective torque applied to each rotation axis will be systematically calculated in real time using a recursive algorithm as stated below.

(1) Call an external aerodynamic subroutine (e.g., the Utah Aerodyn subroutine [Hansen 1995]) to calculate the aerodynamic forces F_i and moments Mae_s directly exerted on each of the blade segments, i=1,2, ···, n. Notice that the aerodynamic forces exerted on the nacelle and tower can also be included in the calculations.

(2) Calculate ζ_i, for i = n, n-1, n-2, ..., 2, in reversed order using equations 43, 45 and 46. In the case where an aileron device is attached to the end of the blade, $\zeta_{n+1} = m_{aileron} \cdot g$. That is, ζ_{n+1} equals the weight of the aileron. Otherwise, $\zeta_{n+1} = 0$.

(3) Calculate T_i, τ_i, T'_i, i=n, n-1, n-2,..., 1, in reverse order using Equations 53 through 55. Notice that $T'_{n+1}=0$.

The above algorithms were used to derive the EOMs of two simple wind turbines in the following chapter.

5.0 Phase I Test Cases

One of the most important tasks in developing a wind turbine modeler is to verify its theoretical formulations. The formulations discussed in chapters 3 and 4 are verified in two phases. The first phase focuses on the verifications of "text book" cases, i.e., simple wind turbines whose EOMs can be found in the text books. The second phase checks the EOMs against commercial wind turbine models and experimental data. This chapter discusses two test cases as part of the Phase I verification test cases.

5.1 A Single Rotating Blade

The first test case is a single rotating blade as depicted in Figure 11. The single-blade wind turbine is assumed to be a rigid blade mounted on a rigid tower with a fixed yaw angle. The rotor has zero tilt angle, zero rotor offset, and no teeter axis, i.e.,

$$a = b = c = d = e = f = 0 \text{ and } \delta_3 = 0 \quad . \tag{56}$$

Since there is no yaw motion, the rotor length e and the overhang distance f can both be set to zero without changing the dynamics of the turbine. The blade is assumed to be a rigid blade but flexible at the root point about the in-plane bending axis relative to the hub. This flexible blade-root configuration can be easily accommodated by locking the pitch and coning axes of the generic wind turbine model as shown in Figure 6. This unrealistic turbine configuration was chosen as the first test case because it resembles a double pendulum problem whose EOMs can be found in many books. This turbine model can be derived from the base model described in Chapter 2 by locking all rotation axes except for the azimuth and the in-plane bending axes, see Figure 6. That is,

$$q_2 = 0,\ q_3 = 0,\ q_4 = \theta_1,\ q_5 = 0,\ q_6 = 0,\ q_7 = 0,\ q_8 = \theta_2 \quad , \tag{57}$$

where θ_1 and θ_2 are two independent variables representing the azimuth angle and the in-plane bending angle, respectively. The mass of the hub and rotor are denoted by m_1 and m_2. The hub radius is L_1 and the blade length is L_2. The shapes of the hub and the blade are both assumed to be slender rods. Aerodynamic forces and moment are symbolically represented by the vectors $Fae_8=[fx_2\ fy_2\ fz_2]^t$ and $Mae_8=[tx_2\ ty_2\ tz_2]^t$. The actual values of the aerodynamic forces and moments are calculated from an external subroutine in real time. They are attached to the blade at its center of gravity, i.e., $L_2/2$ ft measured from the blade root. The azimuth axis is assumed to be frictionless, and the in-plane bending axis is

restrained by a torsional spring and damper simulating the stiffness and damping at the blade root.

Appendix A includes a Matlab M file that was implemented to generate the EOMs specifically for this example. The EOMs generated by this problem are included in Appendix B. The EOMs were verified and agree with the EOMs of a double pendulum discussed in [Shabana 1994, Murray et al. 1994]. EOM 1 in Appendix B represents the EOM about the azimuth axis, and EOM 2 is the EOM about the in-pane bending axis.

5.2 A Down-Wind, Free-Yaw, Two-Bladed, Teetered-Rotor Wind Turbine

The second test case is a free-yaw, two-bladed, teetered-rotor wind turbine as shown in Figure 12. The blades and the tower are both assumed to be rigid. The turbine is stall regulated, and has no flexibility at the blade roots. It has zero delta-3 angle and zero overhang distance. Each blade has 7 degrees of precone angle, down wind. From the base model, this wind turbine configuration can be obtained by setting the joint variables as follows,

$$q_2 = q_2,\ q_3 = 0,\ q_4 = q_4,\ q_5 = q_5,\ q_6 = 0,\ q_7 = -7^0,\ q_8 = 0 \quad , \tag{57}$$

where q_2, q_4, q_5 denote the yaw angles, azimuth angles, and teeter angles, respectively.

The rotor has zero offset and uptilt angle. The shaft length is assumed to be 6.97 ft. It has no overhang, and the hub radius is 3 ft. Since the blades are preconed, the center of gravity of the rotor assembly is assumed to be L_5 ft downwind along the x_5 axis, see Figure 6. The aerodynamic forces and moments are attached to the points L_7 ft along the span axis of each blade measured from the blade root. A teeter spring and a damper are attached to the teeter pin to restrain the teeter motion. K_5 and b_5 are used to denote the spring and damping coefficients, respectively.

Appendix C contains the Matlab M file that was implemented to generate the EOMs for this test case. The EOMs generated by this program are included in Appendix D. EOM 1 in Appendix D represents the yaw equation. EOM 2 is the azimuth equation, and EOM 3 is the teeter equation. The EOMs do not agree with the EOMs of a similar turbine model discussed in the NREL report [Hansen 1992] in a term by term comparison. Nevertheless, the disagreements in EOMs do not necessarily imply that the EOMs in Appendix D are incorrect. The turbine model discussed in Hansen's report uses different coordinate

systems and some simplification assumptions. For example; the generator speed in Hansen's model is a constant and the center of gravity of the teetered-rotor is always aligned with the azimuth axis. The EOMs in Appendix D do not use these assumptions. A possible way to verify the EOMs is to run time simulation on measured wind data and compare the simulation results against measured field data.

Although the accuracy of the EOMs in Appendix D is yet to be verified, the two test cases have demonstrated the potential of the EOM builder as proposed in this report. Using the systematic procedures, a complicated wind turbine model can be easily generated. In addition, a reliable free yaw wind turbine model can be easily derived from a verified fixed yaw wind turbine model by simply changing the yaw joint coefficient from one to zero. The behavior of the free yaw machine can then be studied analytically without any hardware construction cost. A summary of the problems encountered and lessons learned from implementations of these two test cases are discussed in the following chapter.

6.0 Summary and Conclusions

Not only can a properly designed wind turbine control system produce higher power output, but it also can reduce structural loads. To accomplish this, a reasonable dynamics model must be included in the control studies. This project proposes a modular wind turbine modeler. The modeler automatically constructs analytical wind turbine models from user input files that describe the turbine configurations. A basic turbine model consists of modules defining the tower, rotor, drive train, generator, power electronics, and blade dynamics. Chapter 2 of this report discusses the dynamics of the power train components. A mathematical formulation that attempts to provide a systematic way to derive the EOMs describing the dynamics of the rotor assembly is presented in Chapters 3 and 4. Two test cases that served as preliminary checks for the theoretical formulation are discussed in Chapter 5. The formulation generates explicit nonlinear differential equations. The closed form EOMs can be extremely informative when studying wind turbine performance at unusual operating conditions. Individual terms can be isolated from the equations to gain insight on the turbine's dynamic behavior. This mathematical formulation has been tested to derive the EOMs of a two-bladed, free-yaw wind turbine; test case 1.2 discussed in Chapter 5, which is modeled in the Yawdyn code for future verification. Further verifications will also be conducted for industrial applications. With the EOMs linearized at an operating point, control designs can be studied using commercial control software like the Matlab control toolbox.

There are a few issues that need to be addressed in future studies. The first relates to the software implementation. The Matlab M files listed in Appendix A and C were initially implemented for Matlab Symbolic Toolbox Version 1.2. The Matlab Symbolic Toolbox was chosen because the EOMs generated can be easily integrated into other Matlab toolboxes for control studies. However, it was found that the 1.2 version was totally inadequate to handle systems of complex dynamics. After a lengthy negotiation with Mathworks, manufacturer of Matlab, a beta version of Symbolic Toolbox 2.0 was obtained in May 1997. The production unit was released in late August. The M files were then rewritten for the new version of the Symbolic Toolbox because the new version is incompatible with the early ones. After taking all that trouble, the results were still discouraging. The Symbolic Toolbox was installed on two Pentiums (75MHz and 166MHz, respectively, 40Mbytes of memory in each PC). Test case 1.1, see Chapter 5, will run on either one of the computers and generates correct results within a few seconds. Both machines failed to run test case 1.2. After dividing the M file of test case 1.2 into three sections and executing each section sequentially, results can be generated on the

166MHz Pentium in about 10 minutes but not on the 75MHz Pentium. In addition to hardware dependency, the Symbolic Toolbox does not provide functions to collect terms with common factors. This functional deficiency might result in very large EOMs that are difficult to simplify manually by users. This problem can be illustrated by the EOMs of test case 1.2 listed in Appendix D. A better symbolic math package should be investigated.

The second issue concerns the linearlization procedure. If the resulting nonlinear EOMs are linearized at the "zero" nominal point, i.e., all nominal values of the states equal to zero, then the linearization procedure is a straightforward process. However, if linearization about a non-zero nominal point is desired, say, at a certain wind speed, the nominal values of the other states need to be determined. This implies that a numerical procedure must be integrated into the turbine modeler to solve for the state values at the nominal operating points, otherwise they have to be given by the users.

The last issue, but not the least one, is the need for the integration of ordinary differential equations, which describe rigid-body motion such as the rotor and power train dynamics, and partial differential equations, which describe continuously flexible components, such as the tower and blades. Models of continuous flexible components were not discussed in this report. A good reference for this subject is the book by Junkins [Junkins and Kim 1993]. It is expected that the Matlab Simulink software can integrate these two types of models in one system and linearize the system at a nominal point specified by the user. However, until it is done, many intricate problems are anticipated.

References

Balafoutis, C.A. and Patel, R.V. [1991], *Dynamic Analysis of Robot Manipulators: A Cartesian Tensor Approach*, Kluwer Academic Publishers, Boston.

Block, J. and Gilliatt, H. [1997], "Active Control of an Aeroelastic Structure," *Proceedings of 35th Aerospace Sciences Meeting and Exhibit*, Reno, NV, AIAA 97-0016.

Bongers, P.M.M. and Dijkstra, S. [1992], "Control of Wind Turbine Systems Aimed at Load Reduction," *Proceedings of American Control Conference*, Chicago, Illinois, USA, 1710-1714.

Cardenas Dobson, R., Asher, G.M., and Asher G. [1996], "Torque Observer for the Control of Variable Speed Wind Turbines Operating Below Rated Wind Speed," *Wind Engineering*, Vol. 20, No. 4, 259-283.

Craig, J.J. [1989], *Introduction to Robotics*, 2nd Edition, Addison-Wesley Publishing Company, New York.

Denavit, J. and Hertenberg, R.S. [1955], "A Kinematic Notation for Lower-Pair Mechanism Based on Matrices," *J. App. Mech.*, Vol. 77, 215-221.

Elliott, A. And Depauw, T., "ADAMS/WT Advanced Developement," *Proceedings of Wind Power 1996*, American Wind Energy Association, Denver, Colorado, 613-622.

Fitzgerald, A.E., Kingsley, C. and Umans, S.D. [1990], *Electric Machinery*, Fifth edition, McGraw Hill, New York, USA.

Garrad, A.D. [1996], Oral Presentation of BLADED code, *1996* American Wind Energy Association Conference, Garrad Hassan and Partners, The Coach House, Folleigh Lane, Long Ashton, Bristol BS18 9JB, Great Britain.

Hansen, A.C. [1992], *Yaw Dynamics of Horizontal Axis Wind Turbines*, National Renewable Energy Laboratory Report NREL/TP-442-4822, NREL, Colorado, USA.

Hansen, A.C. [1995], *User's Guide: YawDyn and AeroDyn for ADAMS*, Version 9.3, National Renewable Energy Laboratory Report, Subcontract No. XF-1-11009-2.

Hinrichsen, E.N. and Nolan, P.J. [1982], "Dynamics and Stability of Wind Turbine Generators," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-101, No. 8, Aug. 1982, pp. 2640-2648.

Junkins, J.L. and Kim, Y. [1993], *Introduction to Dynamics and Control of Flexible Structures*, AIAA Education Series, AIAA, Inc., Washington, DC, 1993.

Leithead, W.E., Rogers, M.C.M., and Agius, P.R.D. [1993], "Dynamic Characteristics of the Drive-train: Cause and Effects," *Proceedings of 15th British Wind Energy Association Conference*, 1993, 211-216.

Leithead, W.E., Rogers, M.C.M., Connor, B., Pierik, J.T.E., van Engelen, T.G., and Reilly, J.O. [1994a], "Design of a Controller for a Test-Rig for a Variable Speed Wind Turbine," *Proceedings of the Third IEEE Conference on Control Applications*, 1994, 239-244.

Leithead, W.E. and Connor, B. [1994b], "Control of a Variable Speed Wind Turbine with Induction Generator," *Proceedings of Control '94 Conference*, University of Warwick, March 21-24, 1994.

Lee, C.S.G. [1985], "Robot Arm Kinematics and Dynamics," in *Advances in Automation and Robotics: Theory and Applications* (G. N. Daridis, ed.), JAI Press, Conn., 21-63.

Murray, R.M., Li, Z. and Sastry, S.S. [1994], *A Mathematical Introduction to Robotic*

Manipulation, CRC Press, Boca Raton, Florida, USA.
Ogata, K. [1992], *System Dynamics*, 2nd edition, Prentice-Hall, New Jersey.
Palm, W.J. [1986], *Control Systems Engineering*, John Wiley & Sons, New York, USA.
Rao, S.S. [1990], *Mechanical Vibration*, 2nd edition, Addison-Wesley Publishing Company, New York.
Shabana, A.A. [1994], *Computational Dynamics*, John Wiley & Sons, New York, USA.
Siljak, D. [1969], *Nonlinear Systems*, John Wiley & Sons, 1969, 299-301.
Stuart, J., Wright, A.D. and Butterfield, C.P. [1996], "Considerations for an Integrated Wind Turbine Controls Capability at the National Wind Turbine Center: An Aileron Control Case Study for Power Regulation and Load Mitigation," *Proceedings of Wind Power 1996*, American Wind Energy Association, Denver, Colorado, 601-612.
Thiringer, T. and Linders, J. [1993], "Control by Variable Rotor Speed of a Fixed Pitch Wind Turbine Operating in a Wind Speed Range," *IEEE Transactions on Energy Conversion*, Vol. 8, No. 3, Sep., 1993, 520-526.
Wright, A.D., Bir, G.S. and Butterfield, C.P. [1995], "An Investigation of Method for Reducing Blade and System Loads for Two-Bladed Teetering-Hub Horizontal-Axis Wind Turbines," *Proceedings of Wind Energy 1995*, ASME SED-Vol. 16, 253.
Wu, K.C. and De La Guardia, R. [1996], "The Effects of Controls on Fatigue Loads in two-Bladed Teetered Rotor Wind Turbines," *Proceedings of Wind Energy 1995*, ASME SED-Vol. 16.
Wu, K.C. and Smith, T. [1997], Modeling and Control of Variable Speed Wind Turbines Operating Below Rated Wind Speed, Final research report to be prepared for National Renewable Energy Laboratory, Dec., 1997.
Zhang, C. [1994], *Dynamics Simulation of Induction Generators Driven by Wind Turbine Rotor*, Research Report No. WE001/94., The AMSET Centre, De Montfort University, Leicester, UK, 1994.

Appendix A. Matlab M File for Test Case 1.1 (phase 1, case 1)

```
% Test Case 1.1 - A single-blade rigid rotor with a free rotational
%                 degree of freedom about the azimuth axis, i.e.,
%                 mu4 = k4 = b4 = 0, and a constrained degree of
%                 freedom about the in-plane bending axis, i.e.,
%                 k8 and b8 are non zeros.
%
% User's input

% define basic kinematic parameters, i.e., the D-H paramters

% L1, L2, theta1, and theta2 are used as the symbolic variables
% reprenting the hub length, blade length, azimuth angle,
% and in-plane bending angle, respectively.

syms L1 L2 theta1 theta2;
q2 = 0; q3 = 0; q4 = theta1; q5 = 0; q6 = 0; q7 = 0; q8 = theta2;
a = 0; b = 0; c = 0; d = 0; e = 0; f = 0;
g = L1;
delta3 = 0;

% define the center of gravity (cg)  of each mass w.r.t.
% its local frame

c44p = [L1/2;0;0];
c88p = [0;-L2/2;0];

% define the mass of the hub and the rotor

syms m1 m2;

% define the inertial of each link w.r.t. it local frame
I44 =[0,          0,          0; ...
      0, m1*L1^2/12,          0; ...
      0,          0, m1*L1^2/12];

I88 =[m2*L2^2/12, 0,          0; ...
                0, 0,          0; ...
                0, 0, m2*L2^2/12];

% define the aerodynamic forces and moments w.r.t.
% the inertial frame

syms fx2 fy2 fz2 tx2 ty2 tz2;
```

```
Fae4 = [0;0;0];

Mae4 = [0;0;0];

Fae8 = [fx2;fy2;fz2];

Mae8 = [tx2;ty2;tz2];

% define actuation torques

Tac4 = 0;

Tac8 = 0;

% define the spring and damping coefficients for each axis

syms k2 b2;
k4 = 0; b4 = 0;
k8 = k2; b8 = b2;

% define the viscous friction coefficients of the joints

mu4 = 0; mu8 = 0;


% build basic D-H matroces

T12 = dhmat(0,0,q2,b);
T23 = dhmat(pi/2,a,-pi/2+q3,-c);
T34 = dhmat(-pi/2,d,-pi/2-delta3+q4,e+f);
T45 = dhmat(pi/2,0,-pi/2+q5,0);
T56 = dhmat(pi/2-delta3,f,pi+q6,-g);
T67 = dhmat(-pi/2,0,pi/2+q7,0);
T78 = dhmat(-pi/2,0,pi/2+q8,0);

% build combined D-H matrices

T2 = T12;
T2 = simple(T2);  % simplify T2 if possible
T24 = T23 * T34;
T24 = simple(T24);
T4 = T2 * T24;
T4 = simple(T4);
T46 = T45 * T56;
T46 = simple(T46);
```

```
T68 = T67 * T78;
T68 = simple(T68);
T48 = T46 * T68;
T48 = simple(T48);
T8 = T4 * T48;
T8 = simple(T8);

% change variable indices from 4,8 to 1,2
T1 = T4;
T2 = T8;
T12 = T48;

I11 = I44;
I22 = I88;

Fae1 = Fae4; Mae1 = Mae4;
Fae2 = Fae8; Mae2 = Mae8;

Tac1 = Tac4; Tac2 = Tac8;

q1 = q4; q2 = q8;
k1 = k4; b1 = b4;
k2 = k8; b2 = b8;

c11p = c44p;
c22p = c88p;

% extracts rotation matrices, z-vectors, and position vectors
% of the origins

R1 = hom2rot(T1);
R12 = hom2rot(T12);
R2 = hom2rot(T2);

z1 = wtextc(T1,3);
z2 = wtextc(T2,3);


P1p = wtextc(T1,4);
P12p = wtextc(T12,4);
P2p = R1 * P12p;

P1 = P1p;
P2 = wtextc(T2,4);
```

```
% calculate the position vector of the cg w.r.t. the
% inertial frame Equation 5

C1p = R1 * c44p;
C2p = R2 * c88p;

r1 = P1p + C1p;          % r1 = p1 + c1p
r2 = P1p + P2p + C2p;    % r2 = p2 + c2p

% define the angular velocities and accelerations of each joint
% w.r.t. the initial frame, i.e., observed in the local frame
% but expressed w.r.t. the initial frame

q1p = sym('q1p') * z1; % angular velocity at the azimuth axis
q2p = sym('q2p') * z2; % angular velocity at the in-plane bending axis
q1pp = sym('q1pp') * z1; % angular acceleration at the azimuth axis
q2pp = sym('q2pp') * z2; % angular acceleration at the in-plane bending axis

% compute the angular velocity of each link w.r.t. the inertial
% frame using Equation 4, i.e., observed and expressed w.r.t.
% the inertial frame

Q1p = q1p;
Q2p = q1p + q2p;

% compute the linear acccelerations at the cg of each link
% using Equation 5, i.e., observed and expressed w.r.t. the
% inertial frame

Q1pp = q1pp;
Q2pp = q1pp + q2pp;

t4 = xproduct(Q1pp,C1p);
t6 = xproduct(Q1p, xproduct(Q1p,C1p));
a1 = t4 + t6;
a1 = simple(a1);

t1 = xproduct(Q1pp,P2p);
t3 = xproduct(Q1p, xproduct(Q1p,P2p));
t4 = xproduct(Q2pp,C2p);
t5 = xproduct( xproduct(Q1p,q2p),C2p);
t6 = xproduct( Q2p, xproduct (Q2p, C2p));
a2 = t1 + t3 + t4 + t5 + t6;
a2 = simple(a2);
```

```
% compute the inertial due to the translation motion applied
% to joint 1 using Eqation 16, i.e., s = 1

t1 = (m1*a1)' * xproduct(z1 , (r1 - P1));
t2 = (m2*a2)' * xproduct(z1 , (r2 - P1));
Mt1 = t1 + t2;
Mt1 = simple(Mt1);

% compute the inertial due to the translation motion applied
% to joint 2, i.e., s = 2

Mt2 = (m2*a2)' * xproduct(z2 , (r2 - P2));
Mt2 = simple(Mt2);

% compute the inertial due to the rotation motion applied
% to joint 1 using Eqation 19, i.e. s = 1

% i=1
t1 = (R1' * z1)' * I11 * (R1' * Q1pp);
t3 = (R1' * xproduct(z1,Q1p))' * I11 * (R1' * Q1p);
Mr1 = t1 + t3;
% i=2
t1 = (R2' * z1)' * I22 * (R2' * Q2pp);
t2 = (R2' * z1)' * I22 * (R2' * xproduct(q1p,Q2p-Q1p));
t3 = (R2' * xproduct(z1,Q2p))' * I22 * (R2' * Q2p);
Mr1 = Mr1 + t1 + t2 + t3;
Mr1 = simple(Mr1);

% compute the inertial due to the rotation motion applied
% to joint 2, i.e., s = 2

% i=2
t1 = (R2' * z2)' * I22 * (R2' * Q2pp);
t2 = (R2' * z2)' * I22 * (R2' * xproduct(q1p,Q2p-Q1p));
t3 = (R2' * xproduct(z2,Q2p))' * I22 *(R2' * Q2p);
Mr2 = t1 + t2 + t3;
Mr2 = simple(Mr2);

% compute the inertial due to the gravitation applied
% to joint 1 using Eqation 21, i.e., s = 1

% i=1
syms g; % define g as the gravitation constant, a scalar
% whose direction is along the negative y direction
% in this example
```

```
t1 = m1 * [0; 0; -g]' * xproduct(z1,(r1-P1));
% i=2
t2 = m2 * [0; 0; -g]' * xproduct(z1,(r2-P1));
Mg1 = t1 + t2;
Mg1 = simple(Mg1);

% compute the inertial due to the gravitation applied
% to joint 2 using Eqation 21, i.e., s = 2

%i=2;
Mg2 = m2 * [0; 0; -g]' * xproduct(z2,(r2-P2));
Mg2 = simple(Mg2);

% compute the spring and damper torque applied to each joint
% define the spring constants and damping coefficents
% of each joints using Equation 23

Tsp1 = -k1 * theta1 * z1 + k2 * theta2 * z2;
Tsp2 = -k2 * theta2 * z2;
Tda1 = -b1 * q1p + b2 * q2p;
Tda2 = -b2 * q2p;

% compute the actuator torques applied to each joint
% The actuator torques equal to zero in this example

% compute the projection vectors on the x-y plane

[num1,den1,n1] = projpp(C1p , z1);

[num2,den2,n2] = projpp(C2p , z2);

[num3,den3,phi1] = projpp(P2p, z1);

% compute the direct aerodynamic torque applied to each link

Tae2 = Mae2 + xproduct(C2p, Fae2);

Tae1 = Mae1 + xproduct(C1p, Fae1);

% define the viscous friction coefficients of the joints

mu1 = 0;

mu2 = 0;
```

```
% compute the total external torque applied to the

% outmost link, i.e., s = 2

T2 = Tsp2 + Tda2 + Tae2;

% compute the effective torque at the joint

tau2 = T2' * z2 + Tac2;

% compute the force transmitted from link s to s-1

% s = 2, i.e., force applied to link 1 by link 2

zeta2 = Fae2 - (1-mu2)*(Fae2- (Fae2*n2)*n2 - (Fae2*z2)*z2);

% compute the torque transmitted from link s to s-1

T2p = T2 - (1-mu2)*((T2' * z2)*z2);

% compute the total external torque applied to the

% inner links, i.e., s < n

T1 = T2p + xproduct(P2p,zeta2) + Tsp1 + Tda1 + Tae1;

% compute the effective torque at the joint

tau1 = T1' * z1 + Tac1;

% construct the EOMs using Equaiton 13

LEOM1 = Mt1 + Mr1 + Mg1;
LEOM1 = simple (LEOM1);
REOM1 = simple (tau1);

LEOM2 = Mt2 + Mr2 + Mg2;
LEOM2 = simple (LEOM2);
REOM2 = simple (tau2);

% store the EOMs in a file

delete('oneblade.eom');

diary oneblade.eom;
```

```
display('EOMs of Example 1')
display('Inertial torques (LHS) of EOM 1');
pretty(LEOM1)
display('External torques (RHS) of EOM 1');
pretty(REOM1)
display('Inertial torques (LHS) of EOM 2');
pretty(LEOM2)
display('External torques (RHS) of EOM 2');
pretty(REOM2)
diary off;
```

Appendix B. EOMs for Test Case 1.1

Inertial torques (LHS) of EOM 1

$$m2\ q1pp\ L1^2 + 1/3\ m1\ L1^2\ q1pp - 1/3\ m2\ L2^2\ q2pp + 1/3\ m2\ L2^2\ q1pp$$

$$+ 1/2\ m2\ L2\ q2p^2\ L1 \sin(theta2) - m2\ L2\ q1p\ q2p\ L1 \sin(theta2)$$

$$- 1/2\ m2\ L2\ q2pp\ L1 \cos(theta2) + m2\ q1pp\ L1\ L2 \cos(theta2)$$

$$+ 1/2\ g\ m1 \cos(theta1)\ L1 + g\ m2 \cos(theta1)\ L1$$

$$+ 1/2\ g\ m2 \cos(theta1 - theta2)\ L2$$

External torques (RHS) of EOM 1

$$- 1/2\ L1\ fy2 \sin(theta1 - 2\ theta2) + 1/2 \sin(theta1)\ L1\ fy2$$

$$+ 1/2\ L1\ fz2 \cos(theta1 - 2\ theta2) - 1/2 \cos(theta1)\ L1\ fz2 - k2\ theta2$$

$$- b2\ q2p$$

Inertial torques (LHS) of EOM 2

$$- 1/6\ m2\ L2\ (3\ q1pp\ L1 \cos(theta2) - 3\ q1p^2\ L1 \sin(theta2) + 2\ L2\ q1pp$$

$$- 2\ L2\ q2pp + 3\ g \cos(theta1 - theta2))$$

External torques (RHS) of EOM 2

$$-k2\ theta2 - b2\ q2p - tx2 - 1/2 \sin(theta1 - theta2)\ L2\ fy2$$

$$+ 1/2 \cos(theta1 - theta2)\ L2\ fz2$$

Appendix C. Matlab M File for Test Case 1.2 (phase 1, case 2)

```
% Test Case 1.2 - A Simplified ERS-80 Rotor Model

% User's input

% q2, q4, and q5 are used as the symbolic variables
% reprenting the yaw, azimuth, and teeter angles, respectively.

syms q2 q4 q5;
q3 = 0;  q6 = 0; q6p = 0;
q7 = - 7 * pi / 360; % 7 degrees precone downwind
a = 0; b = 0; c = 0; d = 0;
e = 6.97; % ft
f = 0;
g = 3; %ft
delta3 = 0;

% define the center of gravity (cg)  of each mass w.r.t.
% its local frame

syms L5 L7;

c22p = [0;0;0];
c44p = [0;0;0];
c55p = [-L5;0;0]; % L5 ft downwind
c77p = [L7;0;0]; % L7 ft outward along the span axis

% define the mass of the nacelle, drive train and the rotor, respectively

syms m2 m5;
m4 = 0;

% define the inertial of each link w.r.t. it local frame
syms I2xx I2yy I2zz;
syms I5xx I5yy I5zz;

I22 =[I2xx,      0,      0; ...
      0,      I2yy,      0; ...
      0,        0,    I2zz];

I44 =[0,       0,       0; ...
      0,       0,       0; ...
      0,       0,       0];
```

```
I55 =[I5xx,      0,      0; ...
      0,     I5yy,      0; ...
      0,      0,    I5zz];

% define the aerodynamic forces and moments w.r.t.
% the inertial frame

syms fx7 fy7 fz7 tx7 ty7 tz7;
syms fx7p fy7p fz7p tx7p ty7p tz7p;


Fae2 = [0;0;0];

Mae2 = [0;0;0];

Fae4 = [0;0;0];

Mae4 = [0;0;0];

Fae5 = [0;0;0];

Mae5 = [0;0;0];

Fae7 = [fx7;fy7;fz7];

Mae7 = [tx7;ty7;tz7];

Fae7p = [fx7p;fy7p;fz7p];

Mae7p = [tx7p;ty7p;tz7p];

% define actuation torques

Tac2 = 0;

Tac4 = sym('Tp'); % Torque transmitted from the power train

Tac5 = 0;

% define the spring and damping coefficients for each axis

k2 = 0; b2 = 0;
k4 = 0; b4 = 0;
syms k5 b5;
```

```
% define the viscous friction coefficients of the joints

mu1 = 0; mu2 = 0; mu3 = 0;

% define the angular velocities and accelerations of each joint
% For constant speed operation q4p = a constant
% q4pp = 0;

syms q2p q4p q5p q2pp q4pp q5pp


% build basic D-H matroces

T12 = dhmat(0,0,q2,b);
T23 = dhmat(pi/2,a,-pi/2+q3,-c);
T34 = dhmat(-pi/2,d,-pi/2-delta3+q4,e+f);
T45 = dhmat(pi/2,0,-pi/2+q5,0);
T56 = dhmat(pi/2-delta3,f,pi+q6,-g);
T67 = dhmat(-pi/2,0,pi/2+q7,0);
T56p = dhmat(-pi/2-delta3,f,pi+q6,-g);
T6p7p = dhmat(-pi/2,0,pi/2+q7,0);

% build combined D-H matrices

T2 = T12;
T2 = simple(T2);  % simplify T2 if possible
T24 = T23 * T34;
T24 = simple(T24);
T4 = T2 * T24;
T4 = simple(T4);
T5 = T4 * T45;
T5 = simple(T5);
T57 = T56 * T67;
T57 = simple(T57);
T7 = T5 * T57;
T7= simple(T7);
T57p = T56p * T6p7p;
T57p = simple(T57p);
T7p = T5 * T57p;
T7p = simple(T7p);

% change variable indices from 2,4,5,7 to 1,2,3,4
T1 = T2;
T12 = T24;
T2 = T4;
```

```
T23 = T45;
T3 = T5;
T34 = T57;
T4 = T7;
T34p = T57p;
T4p = T7p;

m1 = m2; m2 = m4; m3 = m5;
I11 = I22; I22 = I44; I33 = I55;
Fae1 = Fae2; Mae1 = Mae2;
Fae2 = Fae4; Mae2 = Mae4;
Fae3 = Fae5; Mae3 = Mae5;
Fae4 = Fae7; Mae4 = Mae7;
Fae4p = Fae7p; Mae4p = Mae7p;
Tac1 = Tac2; Tac2 = Tac4; Tac3 = Tac5;

q1 = q2; q2 = q4; q3 = q5;
k1 = k2; b1 = b2;
k2 = k4; b2 = b4;
k3 = k5; b3 = b5;

% extracts rotation matrices, z-vectors, and position vectors
% of the origins

R1 = hom2rot(T1);
R12 = hom2rot(T12);
R2 = hom2rot(T2);
R23 = hom2rot(T23);
R3 = hom2rot(T3);
R34 = hom2rot(T34);
R34p = hom2rot(T34p);
R4 = hom2rot(T4);
R4p = hom2rot(T4p);



z1 = wtextc(T1,3);
z2 = wtextc(T2,3);
z3 = wtextc(T3,3);
z4 = wtextc(T4,3);
z4p = wtextc(T4p,3);


P1p = wtextc(T1,4);
P12p = wtextc(T12,4);
```

```
P2p = R1 * P12p;
P23p = wtextc(T23,4);
P3p = R2 * P23p;
P34p = wtextc(T34,4);
P4p = R3 * P34p;
P34pp = wtextc(T34p,4);
P4pp = R3 * P34pp;


P1 = P1p;
P2 = wtextc(T2,4);
P3 = wtextc(T3,4);
P4 = wtextc(T4,4);
P4p = wtextc(T4p,4);

% calculate the position vector of the cg w.r.t. the
% inertial frame Equation 5

C1p = R1 * c22p;
C2p = R2 * c44p;
C3p = R3 * c55p;
C4p = R4 * c77p;
C4pp = R4p * c77p;

r1 = P1p + C1p;          % r1 = p1 + c1p
r2 = P1p + P2p + C2p;    % r2 = p2 + c2p
r3 = P1p + P2p + P3p + C3p; % r3 = p3 + c3p
% aerodynamic forces attach on Blade 1
r4 = P1p + P2p + P3p + P4p + C4p;
% aerodynamic forces attach on Blade 2
r4p = P1p + P2p + P3p + P4pp + C4pp;


% define the angular velocities and accelerations of each joint
% w.r.t. the initial frame, i.e., observed in the local frame
% but expressed w.r.t. the initial frame

q1p = q2p * z1; % angular velocity at the yaw axis
q2p = q4p * z2; % angular velocity at the azimuth axis
q3p = q5p * z3; % angular velocity at the teeter axis
q1pp = q2pp* z1; % angular acceleration at the yaw axis
q2pp = q4pp* z2; % angular acceleration at the azimuth axis
q3pp = q5pp* z3; % angular acceleration at the teeter axis
```

```
% compute the angular velocity of each link w.r.t. the inertial
% frame using Equation 4, i.e., observed and expressed w.r.t.
% the inertial frame

Q1p = q1p;
Q2p = q1p + q2p;
Q3p = q1p + q2p + q3p;

% compute the linear acccelerations at the cg of each link
% using Equation 7, i.e., observed and expressed w.r.t. the
% inertial frame

Q1pp = q1pp;
Q2pp = q1pp + q2pp;
Q3pp = q1pp + q2pp + q3pp;

% i =1
t4 = xproduct(Q1pp,C1p);
t6 = xproduct(Q1p, xproduct(Q1p,C1p));
a1 = t4 + t6;
a1 = simple(a1);
%i=2
t1 = xproduct(Q1pp,P2p);
t3 = xproduct(Q1p, xproduct(Q1p,P2p));
t4 = xproduct(Q2pp,C2p);
t5 = xproduct(xproduct(Q1p,q2p),C2p);
t6 = xproduct(Q2p, xproduct (Q2p, C2p));
a2 = t1 + t3 + t4 + t5 + t6;
a2 = simple(a2);
%i=3
t1 = xproduct(Q1pp,P2p) + xproduct(Q2pp,P3p);
t2 = xproduct(xproduct(Q1p,q2p),P3p);
t3 = xproduct(Q1p, xproduct(Q1p,P2p)) ...
  + xproduct(Q2p, xproduct(Q2p,P3p));
t4 = xproduct(Q3pp,C3p);
t5 = xproduct(xproduct(Q1p,q2p)+xproduct(Q2p,q3p),C3p);
t6 = xproduct(Q3p, xproduct (Q3p, C3p));
a3 = t1 + t2 + t3 + t4 + t5 + t6;
a3 = simple(a3);

% compute the inertial due to the translation motion applied
% to the yaw axis using Eqation 16, i.e., s = 1

t1 = (m1*a1)' * xproduct(z1 , (r1 - P1));
t2 = (m2*a2)' * xproduct(z1 , (r2 - P1));
```

```
t3 = (m3*a3)' * xproduct(z1 , (r3 - P1));
Mt1 = t1 + t2 + t3;
Mt1 = simple(Mt1);

% compute the inertial due to the translation motion applied
% to the azimuth axis, i.e., s = 2

t2 = (m2*a2)' * xproduct(z2 , (r2 - P2));
t3 = (m3*a3)' * xproduct(z2 , (r3 - P2));
Mt2 = t2 + t3;
Mt2 = simple(Mt2);

% compute the inertial due to the translation motion applied
% to the teeter axis, i.e., s = 3

t3 = (m3*a3)' * xproduct(z3 , (r3 - P3));
Mt3 = t3;
Mt3 = simple(Mt3);

% compute the inertial due to the rotation motion applied
% to the yaw axis using Eqation 19, i.e. s = 1

% i=1
t1 = (R1' * z1)' * I11 * (R1' * Q1pp);
t3 = (R1' * xproduct(z1,Q1p))' * I11 * (R1' * Q1p);
Mr1 = t1 + t3;
% i=2
t1 = (R2' * z1)' * I22 * (R2' * Q2pp);
t2 = (R2' * z1)' * I22 * (R2' * xproduct(q1p,Q2p-Q1p));
t3 = (R2' * xproduct(z1,Q2p))' * I22 * (R2' * Q2p);
Mr1 = Mr1 + t1 + t2 + t3;
% i =3
t1 = (R3' * z1)' * I33 * (R3' * Q3pp);
t2 = (R3' * z1)' * I33 * (R3' * (xproduct(q1p,Q3p-Q1p) ...
  +xproduct(q2p,Q3p-Q2p)));
t3 = (R3' * xproduct(z1,Q3p))' * I33 * (R3' * Q3p);
Mr1 = Mr1 + t1 + t2 + t3;

Mr1 = simple(Mr1);

% compute the inertial due to the rotation motion applied
% to the azimuth axis, i.e., s = 2
% i=2

t1 = (R2' * z2)' * I22 * (R2' * Q2pp);
```

```
t2 = (R2' * z2)' * I22 * (R2' * xproduct(q1p,Q2p-Q1p));
t3 = (R2' * xproduct(z2,Q2p))' * I22 * (R2' * Q2p);
Mr2 = t1 + t2 + t3;

% i=3
t1 = (R3' * z2)' * I33 * (R3' * Q3pp);
t2 = (R3' * z2)' * I33 * (R3' * (xproduct(q1p,Q3p-Q1p)+ ...
  xproduct(q2p,Q3p-Q2p)));
t3 = (R3' * xproduct(z2,Q3p))' * I33 * (R3' * Q3p);
Mr2 = Mr2 + t1 + t2 + t3;
Mr2 = simple(Mr2);

% compute the inertial due to the rotation motion applied
% to the teeter axis, i.e., s = 3

% i=3
t1 = (R3' * z3)' * I33 * (R3' * Q3pp);
t2 = (R3' * z3)' * I33 * (R3' * (xproduct(q1p,Q3p-Q1p) + ...
  xproduct(q2p,Q3p-Q2p)));
t3 = (R3' * xproduct(z3,Q3p))' * I33 * (R3' * Q3p);
Mr3 = t1 + t2 + t3;
Mr3 = simple(Mr3);

% compute the inertial due to the gravitation applied
% to the yaw axis using Eqation 21, i.e., s = 1

g = [0 ; 0; sym('-g')];
% define g as the gravitation vector, whose direction is
% along the negative z direction in this example

% i=1
t1 = m1 * g' * xproduct(z1,(r1-P1));
% i=2
t2 = m2 * g' * xproduct(z1,(r2-P1));
% i=3
t3 = m3 * g' * xproduct(z1,(r3-P1));
Mg1 = t1 + t2 + t3;
Mg1 = simple(Mg1);


% compute the inertial due to the gravitation applied
% to the azimuth axis using Eqation 21, i.e., s = 2

%i=2;
t2 = m2 * g' * xproduct(z2,(r2-P2));
```

```
%i=3;
t3 = m3 * g' * xproduct(z2,(r3-P2));
Mg2 = t2 + t3;
Mg2 = simple(Mg2);

% compute the inertial due to the gravitation applied
% to the teeter axis using Eqation 21, i.e., s = 3

%i=3;
Mg3 = m3 * g' * xproduct(z3,(r3-P3));
Mg3 = simple(Mg3);


% compute the spring and damper torque applied to each
% axis using equations 22 and 23

Tsp1 = -k1 * q1 * z1 + k2 * q2 * z2;
Tsp2 = -k2 * q2 * z2 + k3 * q3 * z3;
Tsp3 = -k3 * q3 * z3;

Tda1 = -b1 * q1p + b2 * q2p;
Tda2 = -b2 * q2p + b3 * q3p;
Tda3 = -b3 * q3p;

% compute the projection vector n on the x-y plane
% using equation 27 and 29

[num1,den1,n1] = projpp(C1p , z1);

[num2,den2,n2] = projpp(C2p , z2);

[num3,den3,n3] = projpp(C3p , z3);

[num4,den4,phi1] = projpp(P2p, z1);

[num5,den5,phi2] = projpp(P3p, z2);

[num6,den6,phi3] = projpp(P4p, z3);

[num5,den5,phi3p] = projpp(P4pp, z3);

% compute the direct aerodynamic torques applied to
% rotation axis using Equaiton 25

Tae4 = Mae4 + xproduct(C4p, Fae4);
```

```
Tae4p = Mae4p + xproduct(C4pp, Fae4p);

Tae3 = Mae3 + xproduct(C3p, Fae2);

Tae2 = Mae2 + xproduct(C2p, Fae2);

Tae1 = Mae1 + xproduct(C1p, Fae1);

% compute the aerodynamic tprque transmitted to the
% teeter axis using Equation 30, note: mu4 = 1.

Ttr4  = Tae4;
Ttr4p = Tae4p;
zeta4 = Fae4;
zeta4p = Fae4p;

% compute the total external torque applied to the
% teeter axis,i.e. s = 3, using Equation 37.

Tsum3 = Ttr4 + xproduct(P4p,zeta4) + Ttr4p + xproduct(P4pp,zeta4p) ...
  + Tsp3 + Tda3 + Tae3;

% compute the effective torque at the teeter axis

tau3 = Tsum3' * z3 + Tac3;

% compute the force transmitted from the rotor to
% the power train, i.e., s = 3, using Equation 30

t1 = Fae3 - (1-mu3) * (Fae3 - (Fae3'*n3)*n3 - (Fae3'*z3)*z3);
t2 = zeta4 - (1-mu3) * (zeta4 - (zeta4'*n3)*n3 - (zeta4'*phi3)*phi3);
t3 = zeta4p - (1-mu3) * (zeta4p - (zeta4p'*n3)*n3 - (zeta4p'*phi3p)*phi3p);

zeta3 = t1 + t2 + t3;

% compute the torque transmitted from the rotor to
% the power train, i.e., s = 3, using Equation 39

Ttr3 = Tsum3 - (1-mu3)*((Tsum3' * z3)*z3);

% compute the total external torque applied to the
% azimuth axis,i.e. s = 2, using Equation 37.

Tsum2 = Ttr3 + xproduct(P3p,zeta3) + Tsp2 + Tda2 + Tae2;
```

```
% compute the effective torque at the teeter axis

tau2 = Tsum2' * z2 + Tac2;

% compute the force transmitted from the power train to
% the nacelle, i.e., s = 2, using Equation 30

zeta2 = (Fae2 + zeta3) - (1-mu2) * ...
  ( ...
  (Fae2+zeta3) - (Fae2'*n2)*n2 - ...
  (zeta3'*phi2)*phi2 - ((Fae2 + zeta3)'*z2)*z2 ...
  );

% compute the torque transmitted from the power train to
% the nacelle, i.e., s = 2, using Equation 39

Ttr2 = Tsum2 - (1-mu2)*((Tsum2' * z2)*z2);

% compute the total external torque applied to the
% yaw axis,i.e. s = 1, using Equation 37.

Tsum1 = Ttr2 + xproduct(P2p,zeta2) + Tsp1 + Tda1 + Tae1;

% compute the effective torque at the teeter axis

tau1 = Tsum1' * z1 + Tac1;

% compute the force transmitted from the nacelle to
% the tower, i.e., s = 1, using Equation 30

zeta1 = (Fae1 + zeta2) - (1-mu1) * ...
  ( ...
  (Fae1+zeta2) - (Fae1'*n1)*n1 - ...
  (zeta2'*phi1)*phi1 - ((Fae1 + zeta2)'*z1)*z1 ...
  );

% compute the torque transmitted from the necelle to
% the tower, i.e., s = 1, using Equation 39

Ttr1 = Tsum1 - (1-mu1)*((Tsum1' * z1)*z1);

% construct the EOMs using Equaiton 13

LEOM1 = Mt1 + Mr1 + Mg1;
REOM1 = tau1;
```

```
LEOM2 = Mt2 + Mr2 + Mg2;
REOM2 = tau2;


LEOM3 = Mt3 + Mr3 + Mg3;
REOM3 = tau3;

% store the EOMs in a file

delete('case12.eom');

diary case12.eom;
display('EOMs of Test Case 1.2')

display('Inertial torques (LHS) of EOM 1');
pretty(LEOM1)
display('External torques (RHS) of EOM 1');
pretty(REOM1)
display('Inertial torques (LHS) of EOM 2');
pretty(LEOM2)
display('External torques (RHS) of EOM 2');
pretty(REOM2)
display('Inertial torques (LHS) of EOM 3');
pretty(LEOM3)
display('External torques (RHS) of EOM 3');
pretty(REOM3)

diary off;
```

Appendix D. EOMs for Test Case 1.2

Inertial torques (LHS) of EOM 1

$$
\begin{aligned}
&\left(I2zz + \frac{485809}{10000}\,m5\right) q2pp - m5\, q4p^2\, L5^2 \cos(q4) \sin(q5) \cos(q5) \\
&- m5 \sin(q4) \sin(q5) L5^2\, q4pp \cos(q5) - I5xx\, q2pp \cos(q5)^2 \\
&- I5xx\, q2pp \cos(q4)^2 + \cos(q5)^2\, I5yy\, q2pp + \cos(q4)\, I5zz\, q5pp \\
&+ \cos(q4)^2\, I5zz\, q2pp + m5\, L5^2\, q5pp \cos(q4) + m5\, L5^2\, q2pp \cos(q5)^2 \\
&+ m5\, L5^2\, q2pp \cos(q4)^2 - m5\, L5^2\, q2pp \cos(q4)^2 \cos(q5)^2 \\
&+ I5yy\, q5p \sin(q4)\, q4p + \sin(q4) \sin(q5)\, I5xx \cos(q5)\, q4pp \\
&- I5zz\, q4p\, q5p \sin(q4) + I5xx\, q2pp + I5xx\, q2pp \cos(q4)^2 \cos(q5)^2 \\
&- \sin(q4) \cos(q5)\, I5yy \sin(q5)\, q4pp - \cos(q5)^2\, I5yy\, q2pp \cos(q4)^2 \\
&- I5xx\, q4p\, q5p \sin(q4) + I5xx \cos(q4) \sin(q5)\, q4p^2 \cos(q5) \\
&- I5yy \cos(q4) \sin(q5)\, q4p^2 \cos(q5) - \frac{697}{100}\, m5\, q5p^2 \cos(q4) \sin(q5)\, L5 \\
&- 2\, m5\, L5^2\, q4p\, q5p \sin(q4) \cos(q5)^2 - \frac{697}{100}\, m5 \sin(q4) \sin(q5)\, L5\, q4pp \\
&- \frac{697}{50}\, m5\, L5\, q4p\, q5p \sin(q4) \cos(q5) - 2\, m5 \sin(q5)\, L5^2\, q2p\, q5p \cos(q5) \\
&- \frac{697}{50}\, m5 \sin(q5)\, L5\, q2p\, q5p + \frac{697}{100}\, m5\, L5\, q5pp \cos(q4) \cos(q5) \\
&- \frac{697}{100}\, m5\, q4p^2\, L5 \cos(q4) \sin(q5) + \frac{697}{50}\, m5\, L5\, q2pp \cos(q5)
\end{aligned}
$$

```
                                  2                  2
+ 2 m5 q2p L5  q5p cos(q4)  cos(q5) sin(q5)

                        2
- 2 m5 sin(q4) L5  q2p q4p cos(q4)

                        2                          2
+ 2 m5 sin(q4) L5  q2p q4p cos(q4) cos(q5)

                                        2
- 2 I5yy q5p sin(q4) q4p cos(q5)

                                                  2
- 2 sin(q5) I5xx cos(q5) q2p q5p cos(q4)

+ 2 sin(q5) I5xx cos(q5) q2p q5p + 2 sin(q4) I5xx cos(q4) q2p q4p

                                                  2
- 2 sin(q4) I5xx cos(q4) q2p q4p cos(q5)

                        2
+ 2 sin(q4) cos(q5)  I5yy cos(q4) q2p q4p

- 2 cos(q5) I5yy sin(q5) q2p q5p

                                                  2
+ 2 cos(q5) I5yy sin(q5) q2p q5p cos(q4)

                                                      2
- 2 cos(q4) I5zz sin(q4) q2p q4p + 2 I5xx cos(q5)  q5p sin(q4) q4p
```

External torques (RHS) of EOM 1

$$
\begin{aligned}
&tz7 - (-\%6 + \%5)\ L7\ fx7 + (-\%4 + \%3)\ L7\ fy7 \\
&+ \left(-3\ \%2 + 3\ \sin(q2)\ \sin(q5) - \frac{697}{100}\ \sin(q2)\right)\ fx7 \\
&+ \left(-3\ \%1 - 3\ \cos(q2)\ \sin(q5) + \frac{697}{100}\ \cos(q2)\right)\ fy7 + tz7p \\
&- (-\%6 - \%5)\ L7\ fx7p + (-\%4 - \%3)\ L7\ fy7p \\
&+ (-3\ \%2 + 3\ \sin(q2)\ \sin(q5))\ fx7p + (-3\ \%1 - 3\ \cos(q2)\ \sin(q5))\ fy7p - \Big| \\
&\Big| tx7 - (\%8 - \%7)\ L7\ fy7 + (-\%6 + \%5)\ L7\ fz7 - 3\ \sin(q4)\ \cos(q5)\ fy7 \\
&+ \left(3\ \%2 - 3\ \sin(q2)\ \sin(q5) + \frac{697}{100}\ \sin(q2)\right)\ fz7 + tx7p - (\%8 + \%7)\ L7\ fy7p \\
&+ (-\%6 - \%5)\ L7\ fz7p - 3\ \sin(q4)\ \cos(q5)\ fy7p \\
&+ (3\ \%2 - 3\ \sin(q2)\ \sin(q5))\ fz7p - k5\ q5\ \sin(q2)\ \sin(q4) \\
&- b5\ q5p\ \sin(q2)\ \sin(q4)\Big|\ \sin(q2)\ \sin(q4) - \Big|ty7 + (\%8 - \%7)\ L7\ fx7 \\
&- (-\%4 + \%3)\ L7\ fz7 + 3\ \sin(q4)\ \cos(q5)\ fx7 \\
&+ \left(3\ \%1 + 3\ \cos(q2)\ \sin(q5) - \frac{697}{100}\ \cos(q2)\right)\ fz7 + ty7p + (\%8 + \%7)\ L7\ fx7p \\
&- (-\%4 - \%3)\ L7\ fz7p + 3\ \sin(q4)\ \cos(q5)\ fx7p \\
&+ (3\ \%1 + 3\ \cos(q2)\ \sin(q5))\ fz7p + k5\ q5\ \cos(q2)\ \sin(q4) \\
&+ b5\ q5p\ \cos(q2)\ \sin(q4)\Big|\ \cos(q2)\ \sin(q4) + \Big|tz7 - (-\%6 + \%5)\ L7\ fx7 \\
&+ (-\%4 + \%3)\ L7\ fy7 + \left|-3\ \%2 + 3\ \sin(q2)\ \sin(q5) - \frac{697}{100}\ \sin(q2)\right|\ fx7
\end{aligned}
$$

```
      /                                  697          \
    + |-3 %1 - 3 cos(q2) sin(q5) + ------ cos(q2)| fy7 + tz7p
      \                                  100         /

    - (-%6 - %5) L7 fx7p + (-%4 - %3) L7 fy7p

    + (-3 %2 + 3 sin(q2) sin(q5)) fx7p + (-3 %1 - 3 cos(q2) sin(q5)) fy7p

                                      \          \
    - k5 q5 cos(q4) - b5 q5p cos(q4)| cos(q4)| cos(q4)
                                      /          /
```

$$\%1 := \sin(q2)\cos(q4)\cos(q5)$$

$$\%2 := \cos(q2)\cos(q4)\cos(q5)$$

$$\%3 := (\%1 + \cos(q2)\sin(q5))\sin\left(\frac{173}{360}\pi\right)$$

$$\%4 := (\sin(q2)\cos(q4)\sin(q5) - \cos(q2)\cos(q5))\cos\left(\frac{173}{360}\pi\right)$$

$$\%5 := (-\%2 + \sin(q2)\sin(q5))\sin\left(\frac{173}{360}\pi\right)$$

$$\%6 := (-\cos(q2)\cos(q4)\sin(q5) - \sin(q2)\cos(q5))\cos\left(\frac{173}{360}\pi\right)$$

$$\%7 := \sin(q4)\cos(q5)\sin\left(\frac{173}{360}\pi\right)$$

$$\%8 := \sin(q4)\sin(q5)\cos\left(\frac{173}{360}\pi\right)$$

Inertial torques (LHS) of EOM 2

$$
\begin{aligned}
&-m5\,L5^2\,q4pp\,\cos(q5)^2 - m5\,\sin(q4)\,\sin(q5)\,L5^2\,q2pp\,\cos(q5) \\
&- m5\,\sin(q4)\,L5^2\,q2p^2\,\cos(q4)\,\cos(q5)^2 + m5\,\sin(q4)\,L5^2\,q2p^2\,\cos(q4) \\
&- I5yy\,\cos(q5)\,\sin(q4)\,\sin(q5)\,q2pp + I5yy\,q2p\,q5p\,\sin(q4) \\
&- I5yy\,\cos(q4)\,\cos(q5)^2\,q2p^2\,\sin(q4) + I5xx\,\cos(q5)\,\sin(q4)\,\sin(q5)\,q2pp \\
&+ I5zz\,q2p\,q5p\,\sin(q4) + I5zz\,\sin(q4)\,q2p^2\,\cos(q4) \\
&- I5xx\,\cos(q4)\,q2p^2\,\sin(q4) + I5xx\,\cos(q4)\,q2p^2\,\sin(q4)\,\cos(q5)^2 \\
&- I5xx\,q5p\,q2p\,\sin(q4) + 2\,m5\,\sin(q5)\,L5^2\,q4p\,q5p\,\cos(q5) \\
&- 2\,m5\,\sin(q4)\,L5^2\,q2p\,q5p\,\cos(q5)^2 + 2\,m5\,\sin(q4)\,L5^2\,q2p\,q5p \\
&- \frac{697}{100}\,m5\,\sin(q4)\,\sin(q5)\,L5\,q2pp + 2\,I5yy\,\cos(q5)\,\sin(q5)\,q4p\,q5p \\
&- 2\,I5yy\,q2p\,q5p\,\sin(q4)\,\cos(q5)^2 + 2\,I5xx\,\cos(q5)^2\,q2p\,q5p\,\sin(q4) \\
&- 2\,I5xx\,\cos(q5)\,\sin(q5)\,q4p\,q5p + I5yy\,q4pp + m5\,L5^2\,q4pp \\
&- I5yy\,q4pp\,\cos(q5)^2 + I5xx\,\cos(q5)^2\,q4pp - m5\,g\,L5\,\cos(q4)\,\sin(q5)
\end{aligned}
$$

External torques (RHS) of EOM 2

$$
\begin{aligned}
&\Big/ \\
&\Big| tx7 - (\%8 - \%7)\ L7\ fy7 + (-\%6 + \%5)\ L7\ fz7 - 3 \sin(q4) \cos(q5)\ fy7 + \%10 \\
&\Big\backslash \\
&\quad + tx7p - (\%8 + \%7)\ L7\ fy7p + (-\%6 - \%5)\ L7\ fz7p - 3 \sin(q4) \cos(q5)\ fy7p \\
&\quad + (3\ \%2 - 3 \sin(q2) \sin(q5))\ fz7p - \Big/ \Big| (tx7 - (\%8 - \%7)\ L7\ fy7 \\
&\quad + (-\%6 + \%5)\ L7\ fz7 - 3 \sin(q4) \cos(q5)\ fy7 + \%10 + tx7p \\
&\quad - (\%8 + \%7)\ L7\ fy7p + (-\%6 - \%5)\ L7\ fz7p - 3 \sin(q4) \cos(q5)\ fy7p \\
&\quad + (3\ \%2 - 3 \sin(q2) \sin(q5))\ fz7p - k5\ q5 \sin(q2) \sin(q4) \\
&\quad - b5\ q5p \sin(q2) \sin(q4))\ \sin(q2) \sin(q4) - (ty7 + (\%8 - \%7)\ L7\ fx7 \\
&\quad - (-\%4 + \%3)\ L7\ fz7 + 3 \sin(q4) \cos(q5)\ fx7 + \%9 + ty7p \\
&\quad + (\%8 + \%7)\ L7\ fx7p - (-\%4 - \%3)\ L7\ fz7p + 3 \sin(q4) \cos(q5)\ fx7p \\
&\quad + (3\ \%1 + 3 \cos(q2) \sin(q5))\ fz7p + k5\ q5 \cos(q2) \sin(q4) \\
&\quad + b5\ q5p \cos(q2) \sin(q4))\ \cos(q2) \sin(q4) + \Big/ \Big| tz7 - (-\%6 + \%5)\ L7\ fx7 \\
&\quad + (-\%4 + \%3)\ L7\ fy7 + \Big/ \Big| -3\ \%2 + 3 \sin(q2) \sin(q5) - \frac{697}{100} \sin(q2) \Big| \ fx7 \\
&\quad + \Big/ \Big| -3\ \%1 - 3 \cos(q2) \sin(q5) + \frac{697}{100} \cos(q2) \Big| \ fy7 + tz7p \\
&\quad - (-\%6 - \%5)\ L7\ fx7p + (-\%4 - \%3)\ L7\ fy7p \\
&\quad + (-3\ \%2 + 3 \sin(q2) \sin(q5))\ fx7p + (-3\ \%1 - 3 \cos(q2) \sin(q5))\ fy7p \\
&\quad - k5\ q5 \cos(q4) - b5\ q5p \cos(q4) \Big| \ \cos(q4) \Big| \ \sin(q2) \sin(q4) \Big| \ \cos(q2) + \Big| \\
&\quad ty7 + (\%8 - \%7)\ L7\ fx7 - (-\%4 + \%3)\ L7\ fz7 + 3 \sin(q4) \cos(q5)\ fx7 + \%9 \\
&\quad + ty7p + (\%8 + \%7)\ L7\ fx7p - (-\%4 - \%3)\ L7\ fz7p + 3 \sin(q4) \cos(q5)\ fx7p \\
&\quad + (3\ \%1 + 3 \cos(q2) \sin(q5))\ fz7p + \Big/ \Big| (tx7 - (\%8 - \%7)\ L7\ fy7
\end{aligned}
$$

```
      + (-%6 + %5) L7 fz7 - 3 sin(q4) cos(q5) fy7 + %10 + tx7p

      - (%8 + %7) L7 fy7p + (-%6 - %5) L7 fz7p - 3 sin(q4) cos(q5) fy7p

      + (3 %2 - 3 sin(q2) sin(q5)) fz7p - k5 q5 sin(q2) sin(q4)

      - b5 q5p sin(q2) sin(q4)) sin(q2) sin(q4) - (ty7 + (%8 - %7) L7 fx7

      - (-%4 + %3) L7 fz7 + 3 sin(q4) cos(q5) fx7 + %9 + ty7p

      + (%8 + %7) L7 fx7p - (-%4 - %3) L7 fz7p + 3 sin(q4) cos(q5) fx7p

      + (3 %1 + 3 cos(q2) sin(q5)) fz7p + k5 q5 cos(q2) sin(q4)

                                                                  /
      + b5 q5p cos(q2) sin(q4)) cos(q2) sin(q4) + |tz7 - (-%6 + %5) L7 fx7
                                                  \

                                                                  /
      + (-%4 + %3) L7 fy7 + |-3 %2 + 3 sin(q2) sin(q5) - 697/100 sin(q2)| fx7
                            \                                             /

        /                                                   \
      + |-3 %1 - 3 cos(q2) sin(q5) + 697/100 cos(q2)| fy7 + tz7p
        \                                           /

      - (-%6 - %5) L7 fx7p + (-%4 - %3) L7 fy7p

      + (-3 %2 + 3 sin(q2) sin(q5)) fx7p + (-3 %1 - 3 cos(q2) sin(q5)) fy7p

                                                  \          \          \
      - k5 q5 cos(q4) - b5 q5p cos(q4)| cos(q4)| cos(q2) sin(q4)| sin(q2) + Tp
                                      /         /              /

%1 := sin(q2) cos(q4) cos(q5)

%2 := cos(q2) cos(q4) cos(q5)

                                    173
%3 := (%1 + cos(q2) sin(q5)) sin(--- pi)
                                    360

                                                          173
%4 := (sin(q2) cos(q4) sin(q5) - cos(q2) cos(q5)) cos(--- pi)
                                                          360

                                     173
%5 := (-%2 + sin(q2) sin(q5)) sin(--- pi)
                                     360

                                                           173
%6 := (-cos(q2) cos(q4) sin(q5) - sin(q2) cos(q5)) cos(--- pi)
                                                           360
```

$$\%7 := \sin(q4)\cos(q5)\sin\left(\frac{173}{360}\pi\right)$$

$$\%8 := \sin(q4)\sin(q5)\cos\left(\frac{173}{360}\pi\right)$$

$$\%9 := \left(3\,\%1 + 3\cos(q2)\sin(q5) - \frac{697}{100}\cos(q2)\right) fz7$$

$$\%10 := \left(3\,\%2 - 3\sin(q2)\sin(q5) + \frac{697}{100}\sin(q2)\right) fz7$$

Inertial torques (LHS) of EOM 3

$$
\begin{aligned}
&2\ m5\ L5^2\ q2p\ q4p\ \sin(q4)\ \cos(q5)^2 - 2\ m5\ L5^2\ q2p\ q4p\ \sin(q4) + m5\ L5^2\ q5pp \\
&+ \frac{697}{100}\ m5\ q2p^2\ \sin(q5)\ L5 + m5\ L5^2\ q2pp\ \cos(q4) \\
&- m5\ L5^2\ q2p^2\ \cos(q5)\ \sin(q5)\ \cos(q4)^2 + m5\ L5^2\ q2p^2\ \cos(q5)\ \sin(q5) \\
&- m5\ q4p^2\ \sin(q5)\ L5^2\ \cos(q5) + \frac{697}{100}\ m5\ q2pp\ \cos(q4)\ L5\ \cos(q5) \\
&- m5\ g\ \sin(q4)\ L5\ \cos(q5) - I5yy\ \sin(q5)\ q4p^2\ \cos(q5) \\
&+ I5yy\ \cos(q5)\ q2p^2\ \sin(q5) - I5yy\ \cos(q5)\ q2p^2\ \sin(q5)\ \cos(q4)^2 \\
&+ 2\ I5yy\ \cos(q5)^2\ q4p\ q2p\ \sin(q4) - I5yy\ \sin(q4)\ q2p\ q4p \\
&+ I5xx\ \sin(q5)\ q4p^2\ \cos(q5) - I5zz\ q2p\ q4p\ \sin(q4) + I5xx\ q4p\ q2p\ \sin(q4) \\
&- 2\ I5xx\ q4p\ q2p\ \sin(q4)\ \cos(q5)^2 - I5xx\ \cos(q5)\ q2p^2\ \sin(q5) \\
&+ I5xx\ \cos(q5)\ q2p^2\ \sin(q5)\ \cos(q4)^2 + I5zz\ q5pp + I5zz\ \cos(q4)\ q2pp
\end{aligned}
$$

External torques (RHS) of EOM 3

$$
\begin{aligned}
&\Big| tx7 - (\%8 - \%7)\ L7\ fy7 + (-\%6 + \%5)\ L7\ fz7 - 3\ \sin(q4)\ \cos(q5)\ fy7 \\
&\quad + \Big| 3\ \%2 - 3\ \sin(q2)\ \sin(q5) + \frac{697}{100}\ \sin(q2) \Big|\ fz7 + tx7p - (\%8 + \%7)\ L7\ fy7p \\
&\quad + (-\%6 - \%5)\ L7\ fz7p - 3\ \sin(q4)\ \cos(q5)\ fy7p \\
&\quad + (3\ \%2 - 3\ \sin(q2)\ \sin(q5))\ fz7p - k5\ q5\ \sin(q2)\ \sin(q4) \\
&\quad - b5\ q5p\ \sin(q2)\ \sin(q4) \Big|\ \sin(q2)\ \sin(q4) - \Big| ty7 + (\%8 - \%7)\ L7\ fx7 \\
&\quad - (-\%4 + \%3)\ L7\ fz7 + 3\ \sin(q4)\ \cos(q5)\ fx7 \\
&\quad + \Big| 3\ \%1 + 3\ \cos(q2)\ \sin(q5) - \frac{697}{100}\ \cos(q2) \Big|\ fz7 + ty7p + (\%8 + \%7)\ L7\ fx7p \\
&\quad - (-\%4 - \%3)\ L7\ fz7p + 3\ \sin(q4)\ \cos(q5)\ fx7p \\
&\quad + (3\ \%1 + 3\ \cos(q2)\ \sin(q5))\ fz7p + k5\ q5\ \cos(q2)\ \sin(q4) \\
&\quad + b5\ q5p\ \cos(q2)\ \sin(q4) \Big|\ \cos(q2)\ \sin(q4) + \Big| tz7 - (-\%6 + \%5)\ L7\ fx7 \\
&\quad + (-\%4 + \%3)\ L7\ fy7 + \Big| -3\ \%2 + 3\ \sin(q2)\ \sin(q5) - \frac{697}{100}\ \sin(q2) \Big|\ fx7 \\
&\quad + \Big| -3\ \%1 - 3\ \cos(q2)\ \sin(q5) + \frac{697}{100}\ \cos(q2) \Big|\ fy7 + tz7p \\
&\quad - (-\%6 - \%5)\ L7\ fx7p + (-\%4 - \%3)\ L7\ fy7p \\
&\quad + (-3\ \%2 + 3\ \sin(q2)\ \sin(q5))\ fx7p + (-3\ \%1 - 3\ \cos(q2)\ \sin(q5))\ fy7p \\
&\quad - k5\ q5\ \cos(q4) - b5\ q5p\ \cos(q4) \Big|\ \cos(q4)
\end{aligned}
$$

```
%1 := sin(q2) cos(q4) cos(q5)

%2 := cos(q2) cos(q4) cos(q5)

                                        173
%3 := (%1 + cos(q2) sin(q5)) sin(--- pi)
                                        360

                                                                  173
%4 := (sin(q2) cos(q4) sin(q5) - cos(q2) cos(q5)) cos(--- pi)
                                                                  360

                                        173
%5 := (-%2 + sin(q2) sin(q5)) sin(--- pi)
                                        360

                                                                   173
%6 := (-cos(q2) cos(q4) sin(q5) - sin(q2) cos(q5)) cos(--- pi)
                                                                   360

                                  173
%7 := sin(q4) cos(q5) sin(--- pi)
                                  360

                                  173
%8 := sin(q4) sin(q5) cos(--- pi)
                                  360
```

Appendix E. Matlab M Files Used in Test Cases 1.1 and 1.2

```
function h = dhmat(alpha,a,theta,d)

% dhmat(alpha,a,theta,d)
%  - This function generates a symbolic 4x4 configuration
%    matrix relating the configuration of the i-th frame to
%    the (i-1)-th frame in the homogeneous coordinate system
%    using the Denavit-Hartenberg notation.
%
% Input:  alpha  - twist angle, in radian (v),
%         a      - link length (u),
%         theta  - joint angle, in radian (t),
%         d      - link offset (w).
%
% Kung Chris Wu
% Symbolic toolbox 2.0

v=sym(alpha);
u=sym(a);
t=sym(theta);
w=sym(d);
h =[       cos(t),         -sin(t),       0,          u;...
    sin(t)*cos(v), cos(t)*cos(v), -sin(v), -sin(v)*w;...
    sin(t)*sin(v), cos(t)*sin(v), cos(v),  cos(v)*w;...
                0,             0,       0,          1];

function R = hom2rot(T)

% hom2rot(T) - This function extracts the symbolic 3x3 rotation
%              matrix R from a 4x4 matrix T in the homogeneous
%              coordinate system.
%
% Input: T - a symbolic 4x4 matrix
%
% Kung Chris Wu
% Symbolic Toolbox v2.0

for i = 1:3
  for j = 1:3
    R(i,j) = T(i,j);
  end
```

```
end


function w = wtextc(T,n)

% wtextc(T,n) -   This function extracts the first 3 elements of the
%                 n-th column from the input matrix T.
%
% inputs:  T      a 3x3 rotation or a 4x4 configuration
%                 matrix,
%          n      column index
%
% output:  w      a 3x1 vector.
%
% Kung Chris Wu
% Symbolic Toolbox v2.0

for i = 1:3
  w(i,1) = T(i,n);
end


function v = xproduct(a,b)

% xproduct(a,b) - This function computes the crorss product
%                 of two 3x1 vectors.
%
% Input: a,b - symbolic 3x1 vectors
%
% Kung Chris Wu
% Symbolic Toolbox v2.0


S = [0, -a(3,1), a(2,1);a(3,1),0,-a(1,1);-a(2,1),a(1,1),0];
v = S * b;
```

```
function [num,den,n] = projpp(c,z)

% projpp(c,z) -    this function finds the projection of the vector
%                  c on the plane (x-y plane) perpendicular to the
%                  vector z
%
% It returns a unit vector.
%
% Kung Chris Wu
c = sym(c);
z = sym(z);
num = c - (c' * z) * z;
num = simple(num);
den = num' * num;
den = simple(den);
den = sqrt(den);
if (num == sym([0;0;0]) )
  n = num;
else
  n = num ./ den;
  n = simple(n);
end
```

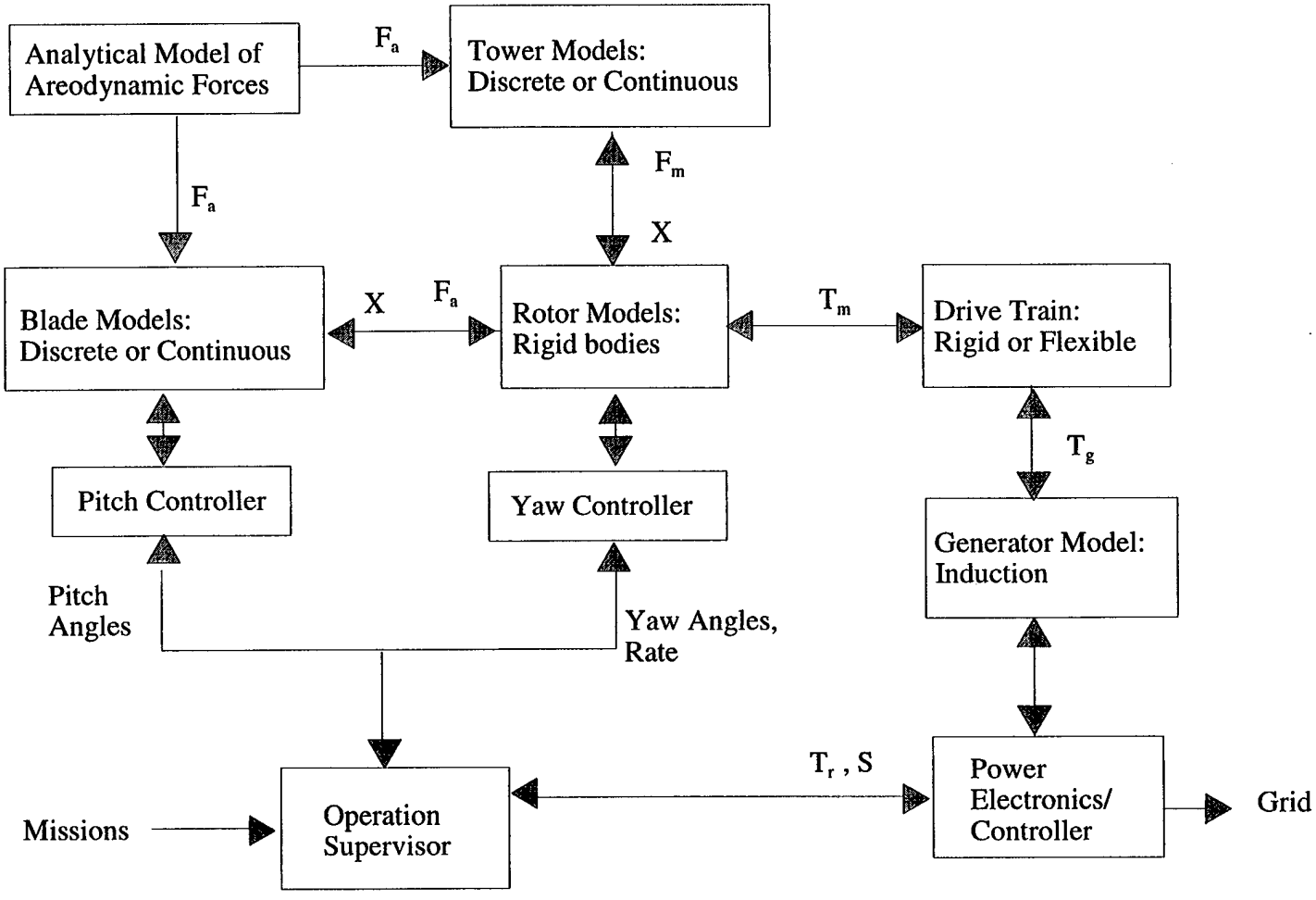

Figure 1. Block diagram of a HAWT with pitch and yaw controllers.

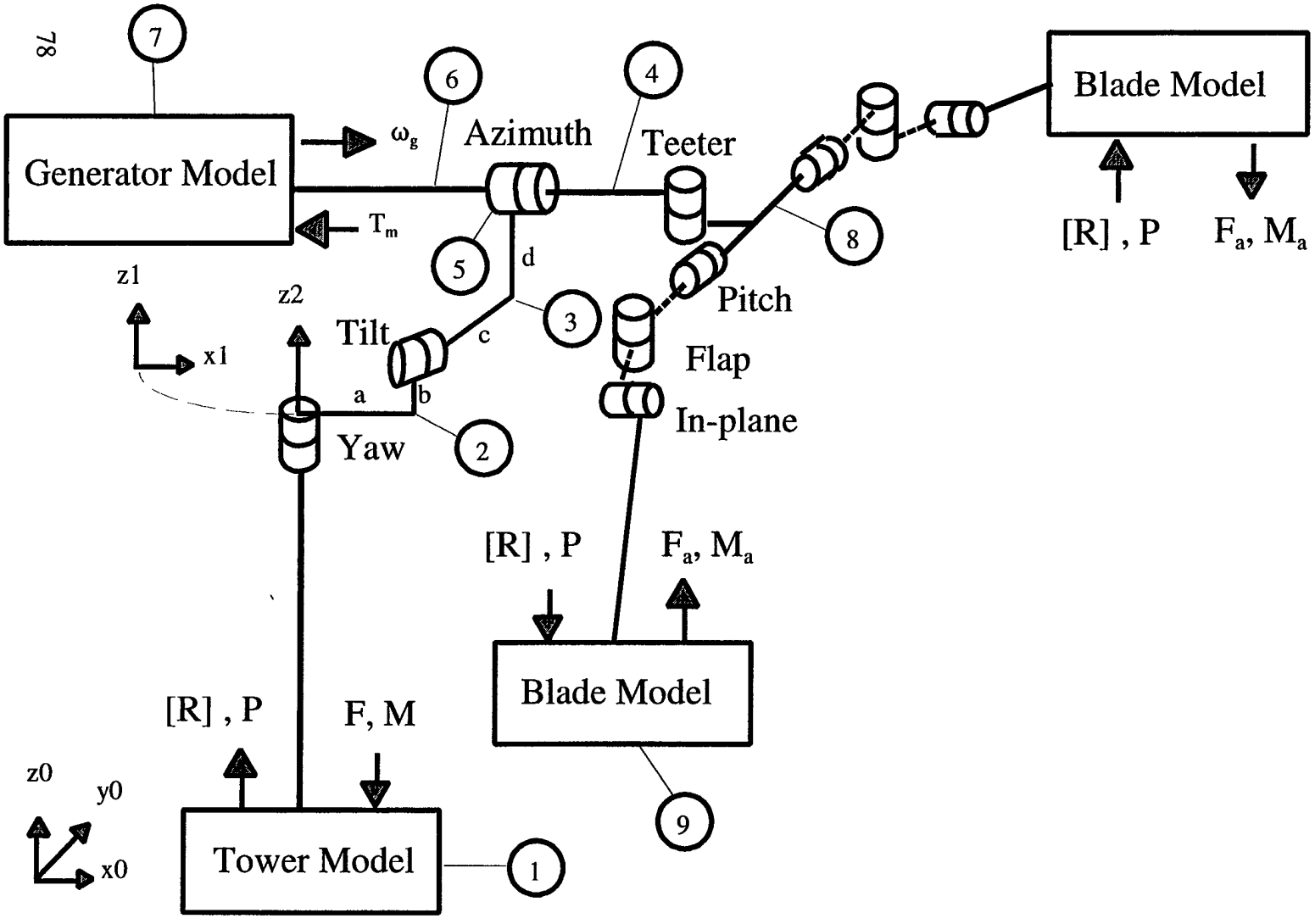

Figure 2. A HAWT model with nine basic components: namely, the tower 1, the bed plate 2, the nacelle 3, the low speed shaft 4, The gearbox 5, the high speed shaft 6, the generator model 7, the teeter assembly 8, and the blades 9.

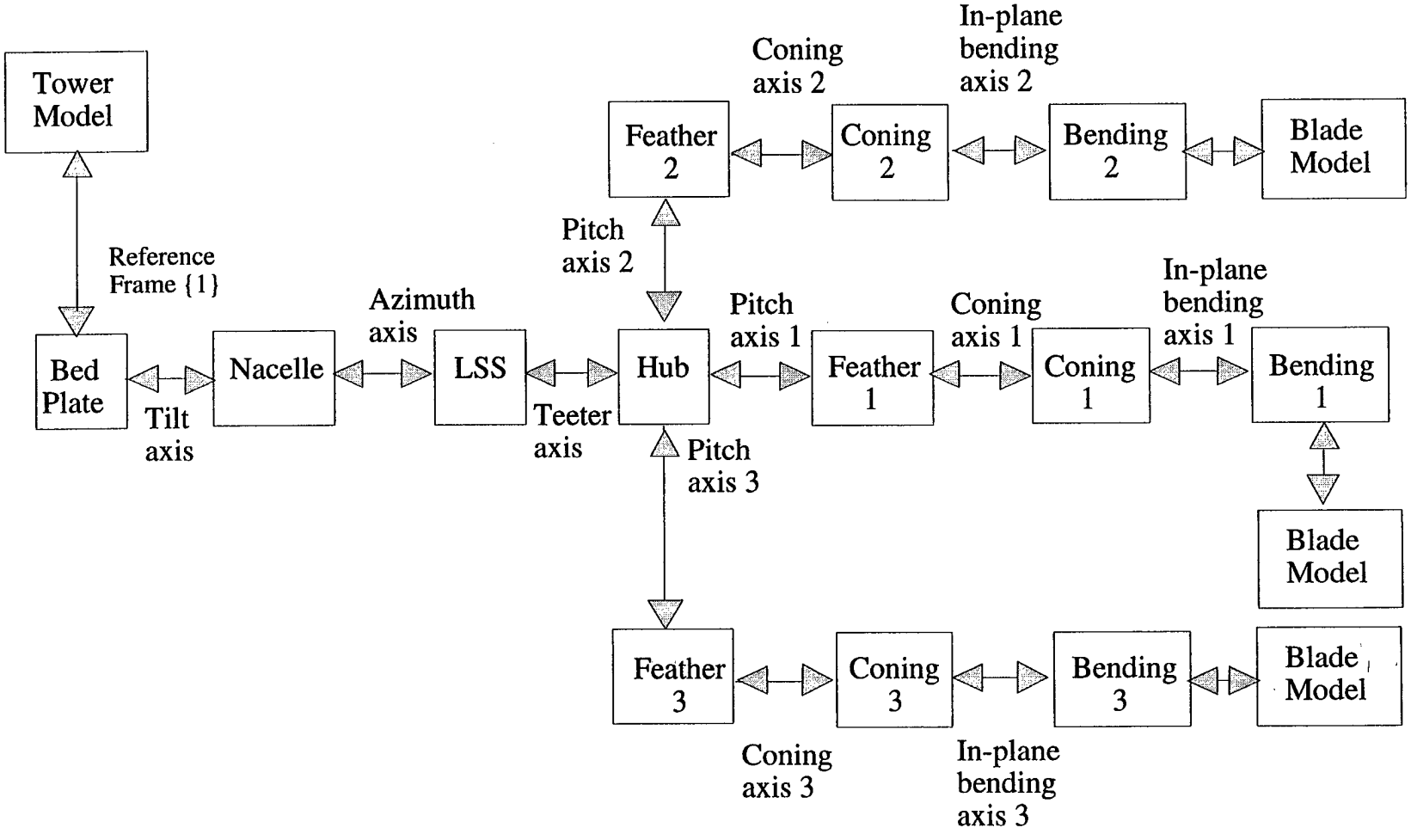

Figure 3. Block diagram of a HAWT when viewed as an open-chain kinematic linkage.

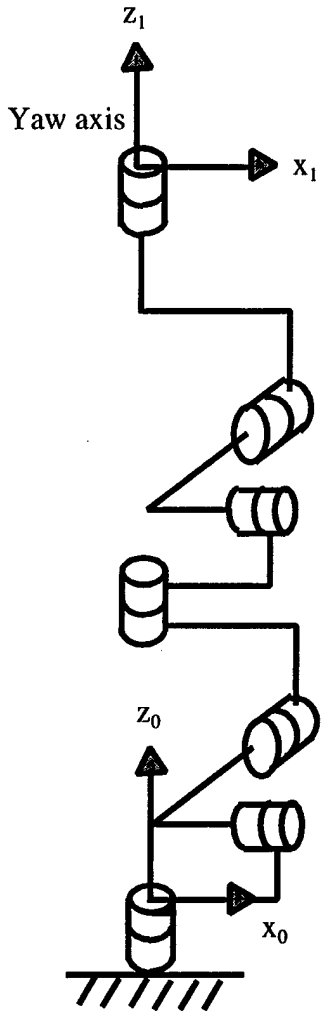

Figure 4. A flexible tower modeled as two rigid segments of rigid bodies. Each rigid body has 3-DOF with respect to its lower neighbor.

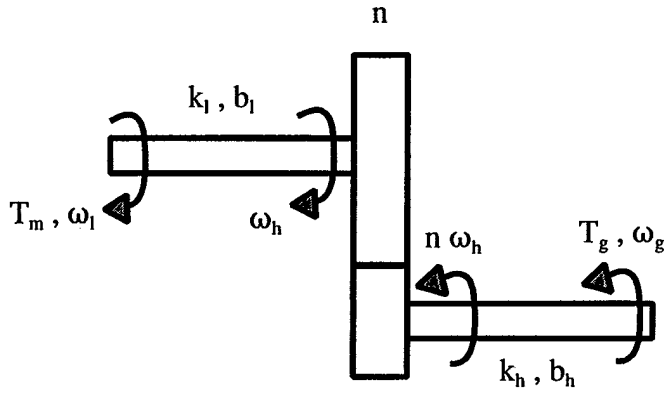

Figure 5. A flexible drive train model.

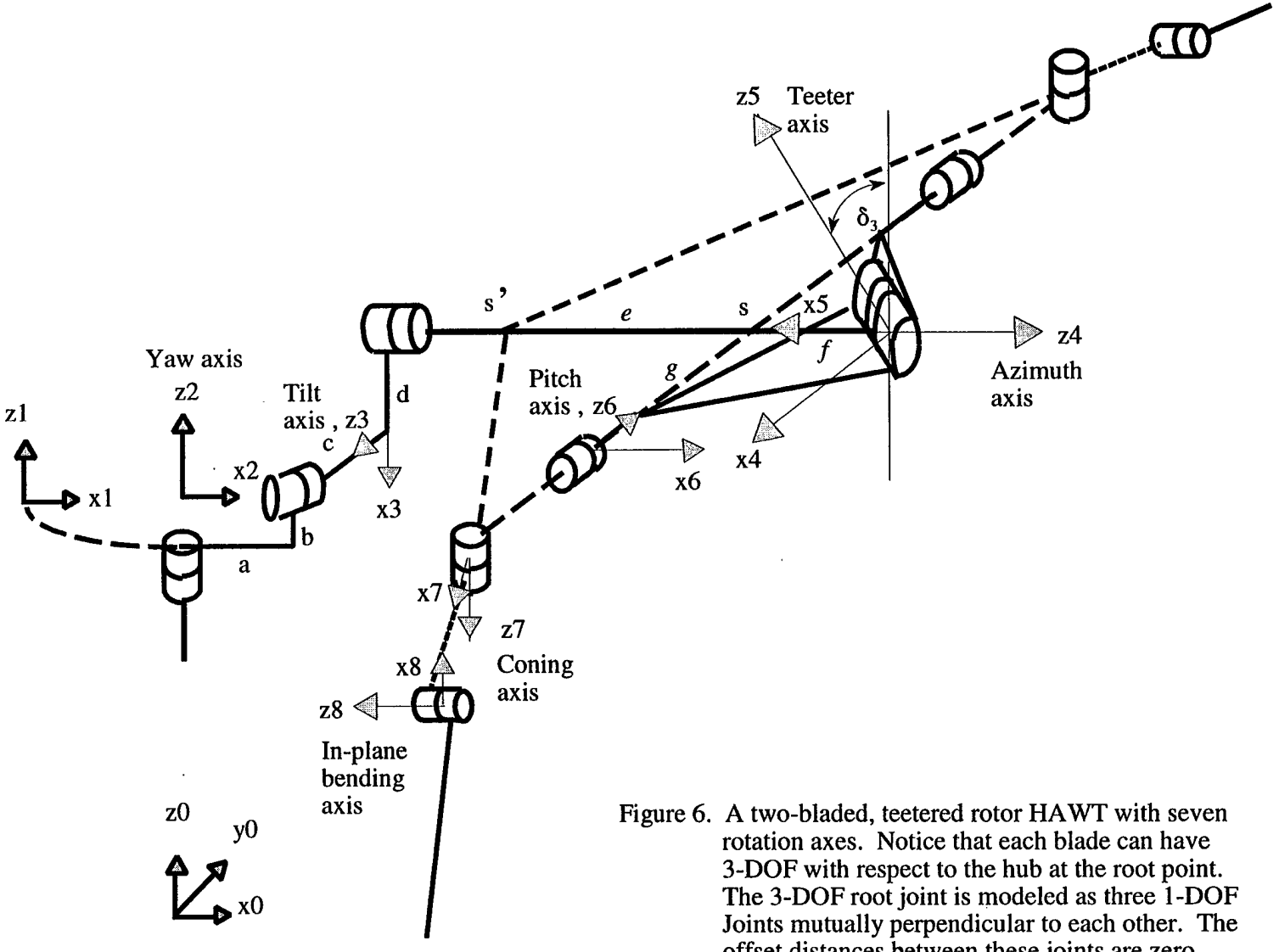

Figure 6. A two-bladed, teetered rotor HAWT with seven rotation axes. Notice that each blade can have 3-DOF with respect to the hub at the root point. The 3-DOF root joint is modeled as three 1-DOF Joints mutually perpendicular to each other. The offset distances between these joints are zero.

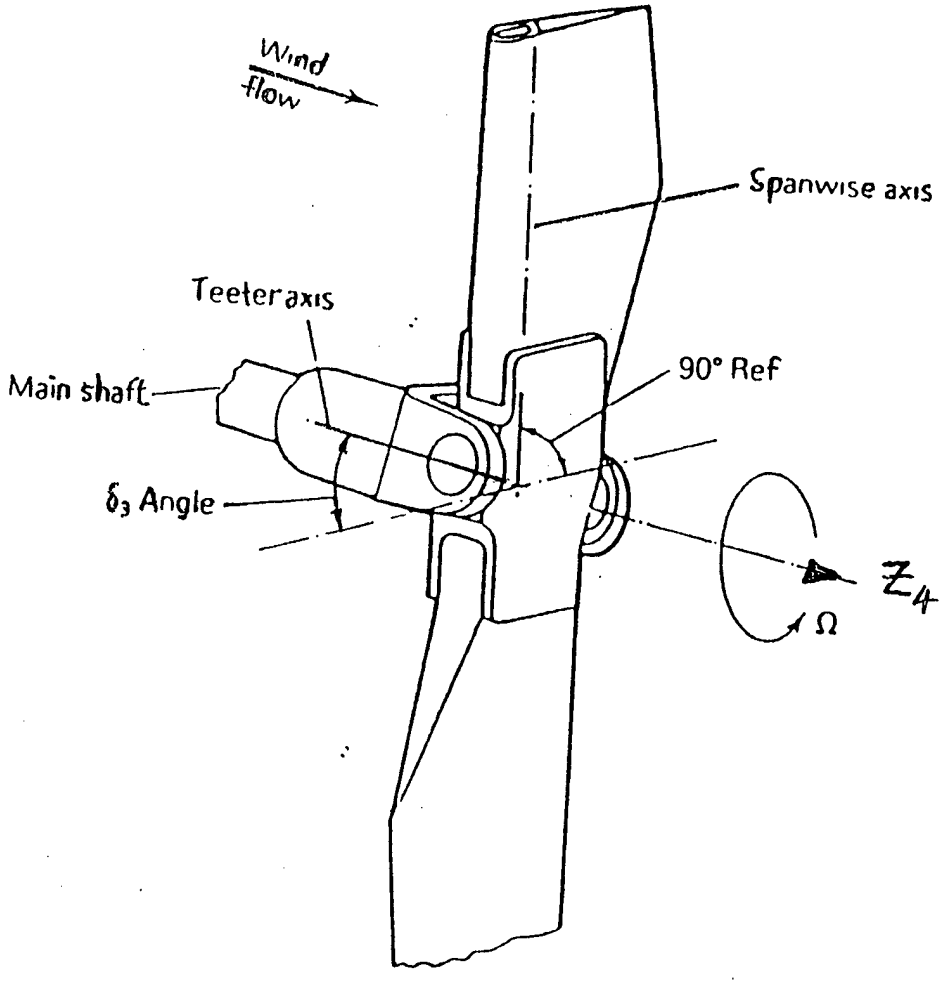

Figure 7. Teetered rotor showing positive δ_3 angle.

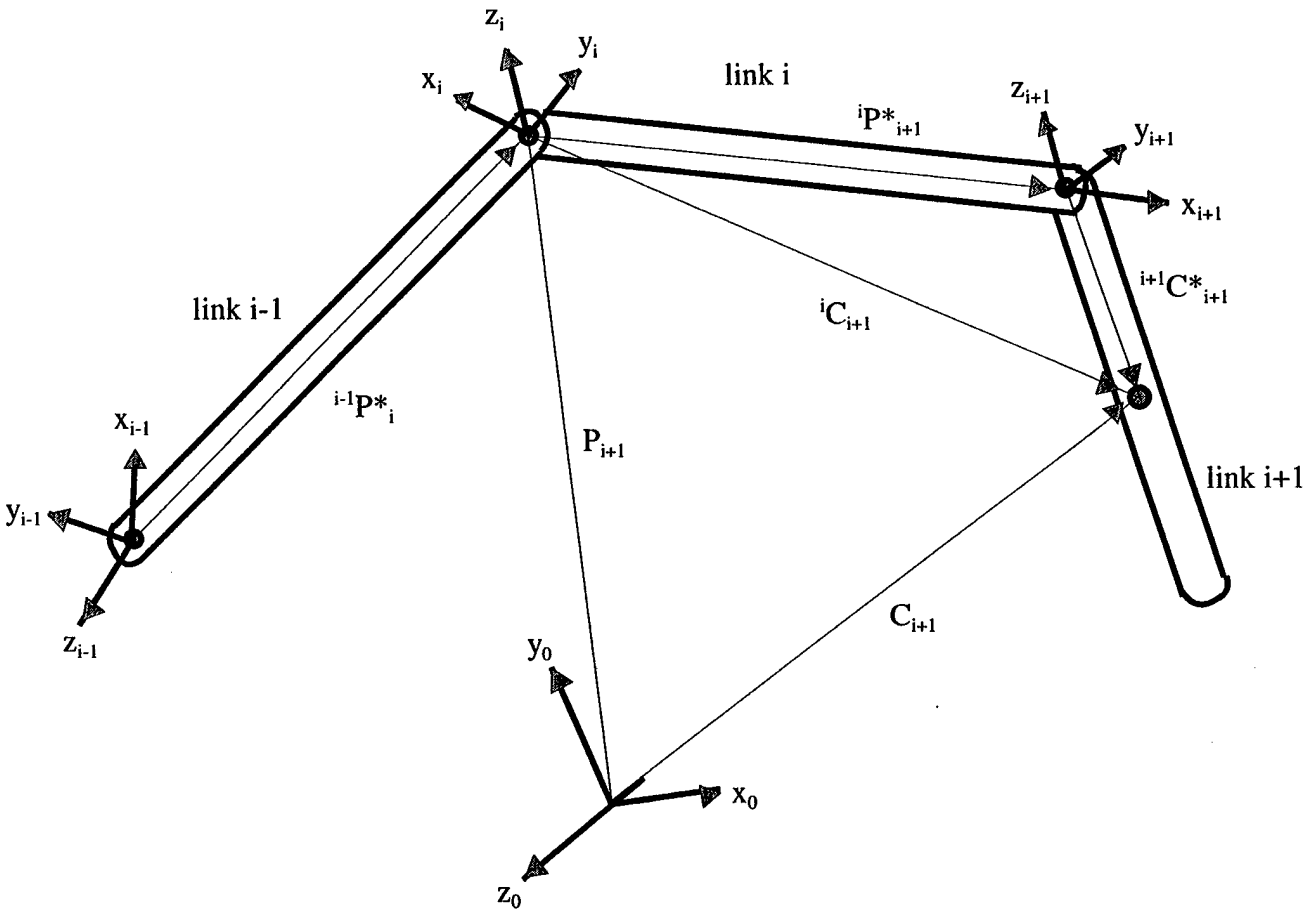

Figure 8. This figure illustrates the notation used in Chapter 3. The right subscript indicates which link the point belongs. Right and left superscripts are used to indicate the frame used to measure and express the position vectors, respectively. For example, C_{i+1} and $^iC_{i+1}$ are position vectors for same point C on the link i+1. C_{i+1} is measured and expressed with respect to the inertial frame. $^iC_{i+1}$ is measured and expressed with the {i}-th frame. The * indicates a point measured within its own coordinate system, but can be expressed with respect to any other frame. For example, C^*_{i+1} is the same vector as $^{i+1}C^*_{i+1}$, but expressed w.r.t. the inertial frame, i.e., $C^*_{i+1} = {}^0R_{i+1}\ {}^{i+1}C^*_{i+1}$.

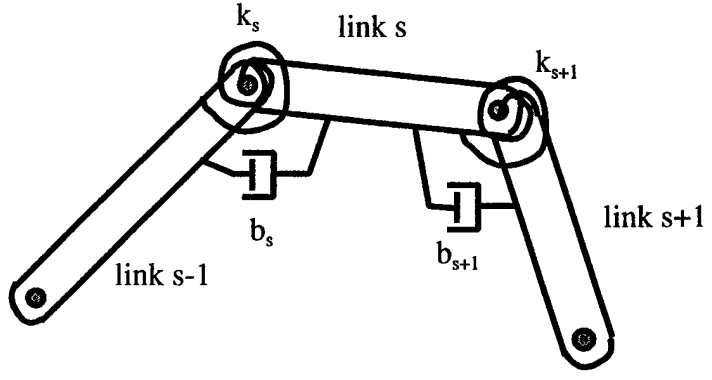

Figure 9. Springs and dampers are used to model structural stiffness and damping.

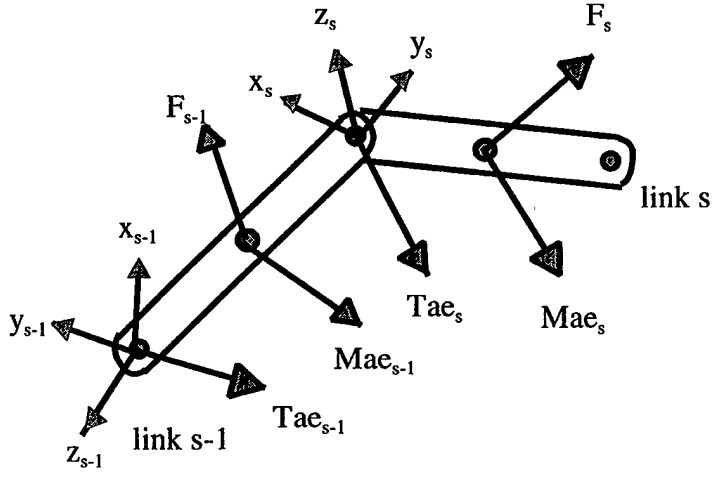

Figure 10. Aerodynamic forces and moments are assumed to be point forces acting through the center of mass of each link. Notice that the resulting torques at the joints are not necessarily parallel to the joint axis.

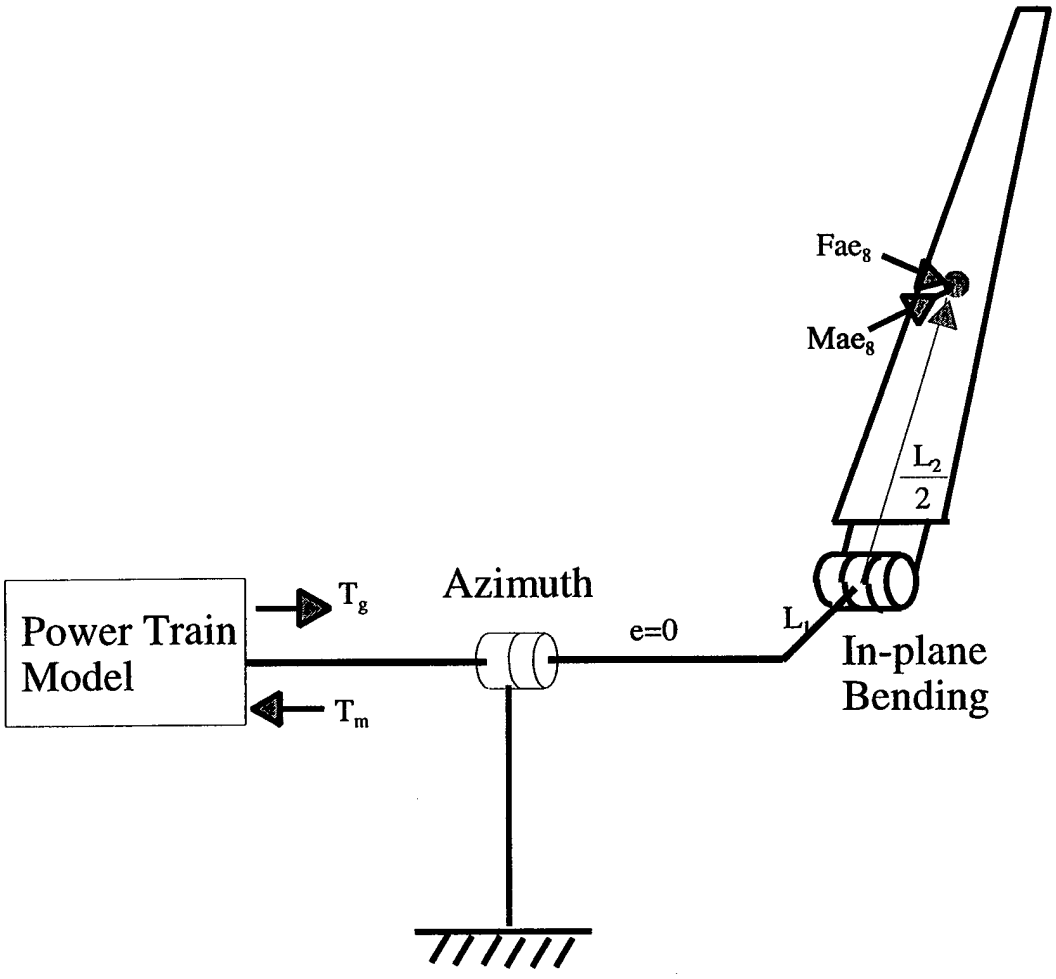

Figure 11. A single rotating blade with in-plane bending flexibility.

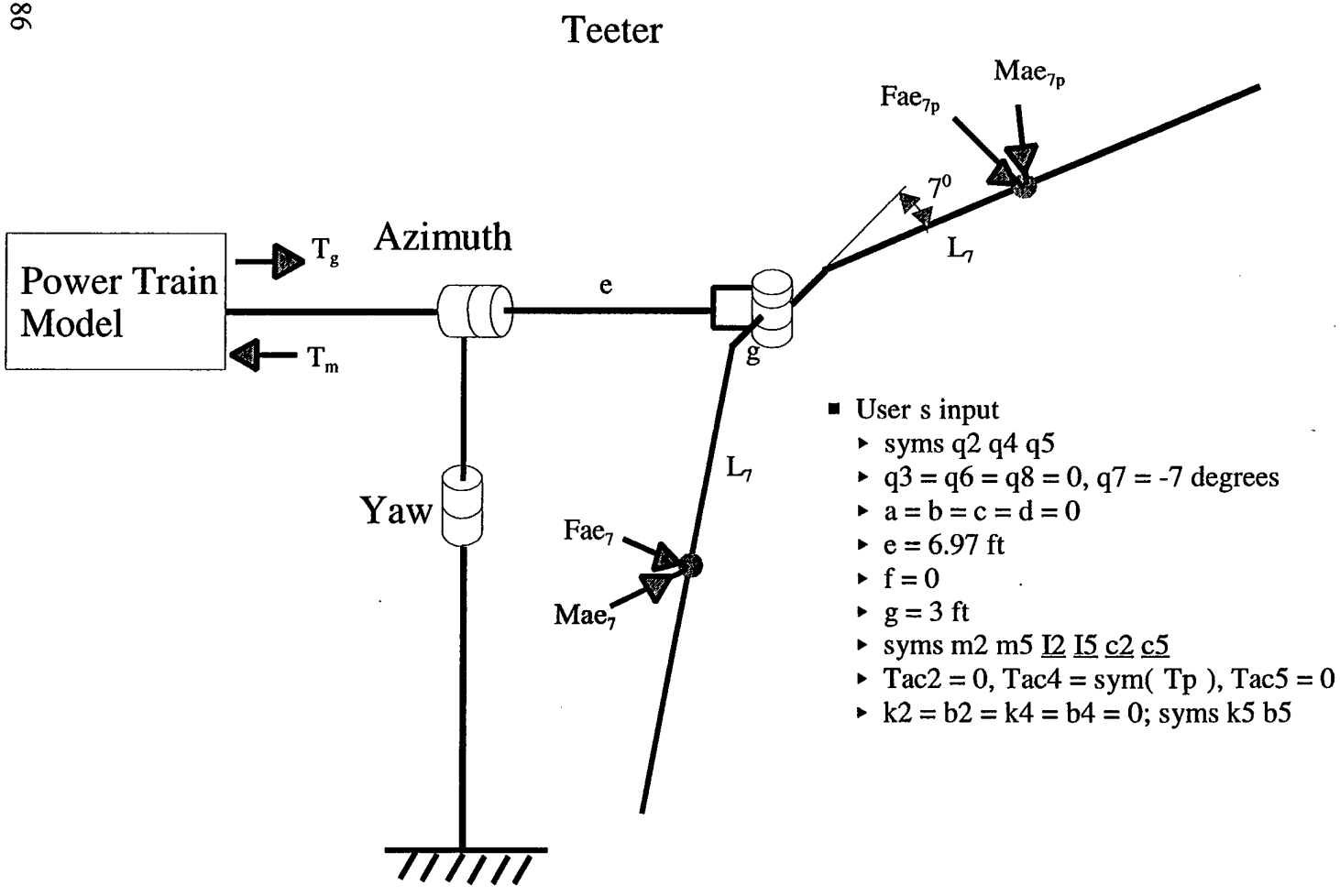

Figure 12. A downwind, free-yaw, two-bladed, teetered-rotor wind turbine with rigid tower and blades.

Report Number (14) SAND--98-0668

Publ. Date (11) 199803

Sponsor Code (18) DOE/EE, XF

JC Category (19) UC-1213, DOE/ER

DOE