Short Course on Wind Turbine Modeling and Control – Part I: Aero–servo–elastic Modeling –

C.L. Bottasso
Politecnico di Milano
Milano, Italy

Korea Institute of Machinery and Materials & Kangwon National University
October 18–19, 2007
Overview of a Comprehensive Multibody Aero–servo–elastic Code
Aero–servo–elastic Models

FEM multibody code, extensively validated for rotorcraft applications:

(Bauchau, Bottasso, Nikishkov, *MCM* 2001)

Wind–energy version:

*CpLambda* (Code for *Performance*, *Loads* and *Aeroelasticity* by *Multi–Body Dynamic Analysis*)
Aero–servo–elastic Models

Classical modeling philosophy:
Ad-hoc codes developed in-house by manufacturers, tailored to specific configurations (e.g., horizontal axis, three bladed, etc.)

FEM multibody approach to wind turbine modeling:
- Turbine is viewed as a complex flexible mechanism
- Model novel configurations of arbitrary topology by assembling basic components chosen from an extensive library of elements

Advantages:
- Simulation software tools are modular and expandable
- Applicable to configurations with arbitrary topologies, including those not yet foreseen
Aero–servo–elastic Models

Topological view of possible wind turbine multibody models:

- Rigid body
- Beam
- Revolute joint
- Actuator
- Boundary condition

Wind Turbine Modeling

Direct-drive three bladed

Blade

Torque actuator

Generator

Nacelle inertia

Pitch actuator

Nacelle stiffness

Yaw actuator

Tower

Elastic shaft

Elastic foundation

Teetering hinge

Mechanical losses

Friction

Direct-drive teetering

Releasable ice accretion

Aero-servo-elastic Models

Tower
**Aero-servo-elastic Models**

**CpLambda structural element library:**

- **Beams:**
  - Geometrically exact, composite-ready beams
  - Curved and twisted NURBS reference lines
  - Fully populated 6x6 stiffness (aeroelastic couplings)

- **Joints:**
  - Enforced by Lagrange multipliers (DAE formulation)
  - Spring, damper, backlash and friction in all joints
  - Flexible joints (contact beam-cylindrical, prismatic, screw)
  - Unilateral joints (contact-impact analysis)

- **Actuators:** first and second order linear and rotational models, refined actuator models

- **Sensors** and control elements
Aero–servo–elastic Models

- **Aerodynamic model:**
  - Lifting lines (two-dimensional strip theory)
  - Tip losses, radial & unsteady flow, dynamic stall
  - Inflow models (Dynamic Pitt–Peters & Peters–He)
  - Generic interface to external CFD or free wake
  - Tower shadow
  - **Wind models** (according to IEC 61400–1):
    - Deterministic gusts (EOG1, ECG)
    - 3D stochastic turbulent wind
    - Wind shear (exponential and logarithmic)

- **Analysis types:**
  - Static analysis
  - Eigenanalysis
  - Dynamic response analysis
  - Stability analysis (implicit Floquet or by excitation)
Aero–servo–elastic Models

**CpLambda** time integrations schemes:

- **Geometric integrators for DAEs:**
  - Exact treatment of geometric non-linearities
  - Exact satisfaction of constraints (no drift)
  - Scaling for improved numerical conditioning (Bottasso et al. 2007)
  - Non-linear unconditional stability (Bottasso et al. 2003):
    bound on total energy of deformable bodies + vanishing of work of constraint forces + conservation of momenta

- **Energy preserving/decaying scheme:**
  - High frequency modes artifacts of discretization
  - Energy decaying scheme damps unresolved modes
  - Improved robustness for large non-linear FEM models

\[ E_D \geq 0 \]
Other Wind Turbine Model Types

**Reduced aero–servo–elastic models:**

- Few degrees of freedom, capture gross to-be-controlled solution scales
- Flexibility neglected or modeled as
  - Equivalent hinge–spring systems or
  - A few carefully chosen modes
- Simplified aerodynamics
- Useful for model-based control, state observers, etc.
Reduced Model for Model-Based Controllers

Example: 6 state – 2 input collective-pitch-only model

Equations:
- Drive-train shaft dynamics
- Elastic tower fore-aft motion
- Blade pitch actuator dynamics
- Electrical generator dynamics

States: $d, \dot{d}, \Omega, \beta_e, \dot{\beta}_e, T_{el_c}$

Inputs: $\beta_c, T_{el_c}$
Reduced Model for Model-Based Controllers

Equations of motion:

\[
(J_R + J_G)\dot{\Omega} + T_l(\Omega) + T_{el_e} - T_a(\Omega, \beta_e, V_w - \dot{d}, V_m) = 0 \\
M_T\ddot{d} + C_T\dot{d} + K_Td - F_a(\Omega, \beta_e, V_w - \dot{d}, V_m) = 0 \\
\ddot{\beta}_e + 2\xi\omega\dot{\beta}_e + \omega^2(\beta_e - \beta_c) = 0 \\
\dot{T}_{ele} + \frac{1}{\tau}(T_{ele} - T_{ele}) = 0
\]

- Tip speed ratio: \( \lambda = \frac{\Omega R}{(V_w - \dot{d})} \)
- Wind: \( V_w = V_m + V_t \) (mean wind + turbulence)
Reduced Model for Model-Based Controllers

Rotor force and moment coefficients:

\[ T_a = \frac{1}{2} \rho \pi R^3 \frac{C_{P_e}(\lambda, \beta_e, V_m)}{\lambda} (V_w - \dot{d})^2 \]

\[ F_a = \frac{1}{2} \rho \pi R^2 C_{F_e}(\lambda, \beta_e, V_m) (V_w - \dot{d})^2 \]

- \( C_{F_e}(\lambda, \beta_e, V_m), C_{P_e}(\lambda, \beta_e, V_m) \) computed off-line with \( \text{CpLambda} \) aero-servo-elastic model, averaging periodic response over one rotor rev

- Stored in look-up tables

Dependence of \( C_{F_e}(\lambda, \beta_e, V_m) \) and \( C_{P_e}(\lambda, \beta_e, V_m) \) on mean wind \( V_m \) accounts for deformability of tower and blades under high winds:
Reduced Model for Model-Based Controllers

Example: 9 state – 4 input individual-pitch model

Equivalent stiffness through springs (no beam elements)

States:
- 3 flap angles
- Rotor azimuth
- Shaft torsion
- 3 tower angles (fore-aft, side-side, torsion)
- Yaw angle
(and their rates)

Inputs: $\beta_{c1}, \beta_{c2}, \beta_{c3}, T_{el c}$
Virtual Testing Environment

Simulation of wind turbine operations in a high fidelity environment:

- Compute extreme loads due to gusts
- Evaluate fatigue damage due to turbulence
- Evaluate response spectra
- Judge performance and suitability of control laws
- Simulate failures and off-design conditions
- Etc.
Virtual Testing Environment

Sensor models

Measurement noise

Virtual plant

Cp L

aero-servo-
elastic model

Wind generator

Process noise

Supervisor

Choice of operating condition:
• Start up
• Power production
• Normal shut-down
• Emergency shut-down
• ...

Kalman filtering

Wind & tower/blade state estimation

Controller

Feedback controller

• PID
• MIMO LQR
• RAPC
• ...

Adaptive reduced model
Virtual Testing Environment

Detailed FEM analysis of sub-components

- Supervisor
  - Choice of operating condition:
    - Start up
    - Power production
    - Normal shut-down
    - Emergency shut-down
    - …

- Feedback controller
  - PID
  - MIMO LQR
  - RAPC
  - …
  - Adaptive reduced model

- Kalman filtering
  - Wind & tower/blade state estimation

Gather loads from multibody model analysis

- Apply equivalent load system on sub-component
Structural Dynamics Models
Coordinates for Multibody Systems

Example: the four bar mechanism (from Geradin & Cardona 2000).

Gruebler (or Kutzbach) formula:

\[ n_{dof} = \frac{n_{sd}(n_{sd} + 1)}{2} N - \sum_{i=1}^{n_j} c_i \]

- \( n_{sd} \) number of space dimensions,
- \( N \) number of bodies,
- \( n_j \) number of joints,
- \( c_i \) dofs removed by a joint

· This system has one single dof: 

\[ \frac{2(2 + 1)}{2} - 3 - \sum_{i=1}^{4} 2 = 1 \]
Minimal coordinates:

Pick $\theta_1$ as free coordinate:

$$\theta_2 = 2\pi - \theta_1 - \theta_3 - \theta_4$$

$$\cos \theta_3 = -\frac{l_2^2 + l_3^2 + 2l_1l_4 \cos \theta_1}{l_1^2 + l_4^2 - 2l_2l_3}$$

$$\sin(\theta_4 + \tan^{-1} \frac{l_4 - l_1 \cos \theta_1}{l_1 \sin \theta_1}) = \frac{l_3 - l_2 \cos \theta_3}{\sqrt{(l_4 - l_1 \cos \theta_1)^2 + (l_1 \sin \theta_1)^2}}$$
Coordinates for Multibody Systems

**Minimal coordinates:**

+ Pros:
  - Minimum number of unknowns
  - No constraint conditions, e.g. through Lagrange multipliers (therefore, the governing equations for dynamics are ODEs, and not DAEs)

- Cons:
  - Complicated, highly non-linear resulting expressions
  - Unsuitable for flexible systems
  - Difficult to generalize to arbitrary topologies
Coordinates for Multibody Systems

Lagrangian coordinates (relative, recursive):

Write two constraints to impose loop closure:

\[ l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) = l_4 \]
\[ l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) = 0 \]
Coordinates for Multibody Systems

Lagrangian coordinates (relative, recursive):

+ Pros:
  - For open loops, leads to minimum number of unknowns
  - Particularly efficient for tree topologies, e.g. robotics
    (important for real-time applications)

- Cons:
  - Complicated, highly non-linear resulting expressions
  - Not well suited for flexible systems, especially when based on
    the floating frame approach
Denavit–Hartenberg formulation of Lagrangian coordinates

Systematic approach for three–dimensional kinematic chains, based on homogeneous coordinates:

\[
d_Q = d_P + d_{QP} = d_P + R \tilde{d}_{QP}
\]

This coordinate transformation can be expressed through a 4x4 matrix:

\[
d_{Q,4} = R_4 \tilde{d}_{QP,4} = \begin{bmatrix} R & d_P \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{d}_{QP} \\ 1 \end{bmatrix}
\]

\[
R_4^{-1} = \begin{bmatrix} R^T & -R^T d_P \\ 0 & 1 \end{bmatrix}
\]
Coordinates for Multibody Systems

Denavit–Hartenberg formulation of Lagrangian coordinates

Homogeneous coordinate transformation from frame $i - 1$ to frame $i$:

$$x^i_4 = R^{i,i-1}_4 x^{i-1}_4$$

This transformation can be used in a recursive manner to describe an arbitrary kinematic chain.
Denavit–Hartenberg formulation of Lagrangian coordinates

Each transformation is described by four parameters (two rotations $\alpha$ and $\theta$, and two translations $a$ and $d$).
Coordinates for Multibody Systems

Rotation $\alpha$ about $x^{i-1}$:

\[
R^{4}_4 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos\alpha & -\sin\alpha & 0 \\
0 & \sin\alpha & \cos\alpha & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Translation $a$ along $x^{i-1}$:

\[
R^{3}_4 = \begin{bmatrix}
1 & 0 & 0 & a \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Rotation $\theta$ about $\hat{z}^i$:

\[
R^{2}_4 = \begin{bmatrix}
\cos\theta & -\sin\theta & 0 & 0 \\
\sin\theta & \cos\theta & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Translation $d$ along $\hat{z}^i$:

\[
R^{1}_4 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & d \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

The link-to-link Denavit–Hartenberg transformation is:

\[
R^{i,i-1}_4 = R^{4}_4 R^{3}_4 R^{2}_4 R^{1}_4
\]
Coordinates for Multibody Systems

Lagrangian coordinates and floating frames: motion is split into gross rigid body motion of a floating frame and small elastic deformation about the frame

+ Pros:
  - Well suited for modal–based elasticity

– Cons:
  - Complicated expressions of the kinetic energy
  - Geometric stiffening is not modelled
Component Mode Synthesis Method

Not yet covered in these notes
Coordinates for Multibody Systems

Cartesian coordinates:

Write eight constraints to assemble the system:

\[
\begin{align*}
    x_1 - \frac{l_1}{2} \cos \theta_1 &= 0 \\
    x_1 + \frac{l_1}{2} \cos \theta_1 &= x_2 - \frac{l_2}{2} \cos \theta_2 \\
    x_2 + \frac{l_2}{2} \cos \theta_2 &= x_3 - \frac{l_3}{2} \cos \theta_3 \\
    x_3 + \frac{l_3}{2} \cos \theta_3 &= 0 \\
    y_1 - \frac{l_1}{2} \sin \theta_1 &= 0 \\
    y_1 + \frac{l_1}{2} \sin \theta_1 &= y_2 - \frac{l_2}{2} \sin \theta_2 \\
    y_2 + \frac{l_2}{2} \sin \theta_2 &= y_3 - \frac{l_3}{2} \sin \theta_3 \\
    y_3 + \frac{l_3}{2} \sin \theta_3 &= 0
\end{align*}
\]
Coordinates for Multibody Systems

Cartesian coordinates with geometrically exact formulations:

The strain energy is $V = V(\varepsilon)$ where $\varepsilon$ are strain measures that are unaffected by arbitrarily large rigid body motions, $V(\varepsilon) = V(Re)$

+ Pros:
  - Exact non-linear geometry
  - Simple expression of the strain energy
Geometrically Exact Beams

Beam kinematics:

- $k = \text{axial}(\alpha' \alpha^T)$ Sectional curvature
- $d, R$
- $d + u_0, \alpha = RR_0$
- $\omega = \text{axial}(\dot{\alpha}\alpha^T)$ Angular velocity
- $v = \dot{d}$ Linear velocity
- $u'_0 + d'$ Tangent to reference line

Sections remain planar after deformation, but not necessarily normal to reference line

Change of components for generic vector $\alpha$

$\alpha^S = \alpha a^B$
Geometrically Exact Beams

**Kinetic energy:**

\[ K = \frac{1}{2} \int_{0}^{L} \mathbf{w} \cdot \mathbf{p} \, ds \]

Generalized velocities and momenta:

\[ \mathbf{w} = (v, \omega) \quad \mathbf{p} = (l, h) \quad \mathbf{p} = M \mathbf{w} \]

6x6 inertial tensor \( M \) (constant in sectional frame \( B \))

**Remarks:**

- Rotary inertia effects included
- Allows offset of sectional mass center and beam reference line
Geometrically Exact Beams

Strain energy: $$V = \frac{1}{2} \int_0^L e \cdot f \, ds$$

Strains and sectional stress resultants:
$$e = (u'_0 + d' - Ru'_0, k - Rk_0) \quad f = (s, m) \quad f = Ke$$

6x6 stiffness matrix $K$ (constant in sectional frame $B$)

Remarks:
- Geometrically exact measures (arbitrarily large rotations and displacements, rigid motion invariance), but small strains
- Fully populated stiffness allows for modeling of elastic couplings arising from use of composite materials
- Offset of tension center from shear center and beam reference line

Stiffness matrix $K$ can be obtained by solving 2D FEM sectional problem (see Giavotto, Borri, Hodges)
Geometrically Exact Beams

Hamilton’s Principle:
\[ \int_{t_i}^{t_f} \int_0^L (\delta v \cdot p - \delta e \cdot f + \delta W_e) \, ds \, dt = 0 \]

with \( \delta W_e = \) work of external forces \( f_e \)

Equations of motion:
\[ \frac{\text{d}}{\text{d}t}(Ap^B) + U(\dot{d})Ap^B - \frac{\text{d}}{\text{d}s}(Af^B) - U(u'_0 + u')Af^B = f_e^S \]

where
\[ A = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \quad U(\bullet) = \begin{bmatrix} 0 & 0 \\ 0 & \bullet \times \end{bmatrix} \]

- Discretize in space using iso-parametric shape functions and reduced integration
- Discretize in time using suitable integration scheme (HHT, modified-\( \alpha \), energy preserving-decaying, etc.)
Classification of Constraints

**Holonomic constraints** (most common in mechanics):
- Scleronomic: \( c(q) = 0, \quad c : \mathbb{R}^n \to \mathbb{R}^m \)
- Rheonomic: \( c(q, t) = 0, \quad c : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^m \)

**Non-holonomic constraints**:
- Scleronomic: \( c(\dot{q}, q) = 0, \quad c : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m \)
- Rheonomic: \( c(\dot{q}, q, t) = 0, \quad c : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^m \)

In practical applications, non-holonomic constraints are typically linear in the velocities, i.e.
\[
A(q)\dot{q} + a(q, t) = 0
\]
Constrained Algebraic Problems

We consider the **minimization** of the functional $\mathcal{F}(q)$ subjected to the **equality constraints** $c(q) = 0$:

$$\min_q \mathcal{F}(q) \quad \text{s.t.: } c(q) = 0$$

with $q \in \mathbb{R}^n$, $c : \mathbb{R}^n \to \mathbb{R}^m$ and $m < n$

There are several possible solution techniques:

- Reduction to an independent set of unknowns
- The method of Lagrange multipliers
- The penalty method
- The augmented Lagrangian method
Constrained Algebraic Problems

**Reduction to an independent set of unknowns** (coordinate partitioning).

The virtual variations of unknown parameters $q$ are constrained as

$$\delta c = A^T \delta q = 0$$

where

$$A_{ij} = \frac{\partial c_j}{\partial q_i}$$

Partition $\delta q$ in independent ($\delta q_I \in \mathbb{R}^{n-m}$) and dependent ($\delta q_D \in \mathbb{R}^m$) sets, i.e.

$$\delta q = (\delta q_I, \delta q_D) \quad A_I^T \delta q_I + A_D^T \delta q_D = 0$$

so that $A_D$ is non-singular

The dependent set can now be expressed as a function of the independent set:

$$\delta q_D = -A_D^{-T} A_I^T \delta q_I$$
Constrained Algebraic Problems

The virtual variations of the unknown parameters can now be expressed through the virtual variations of the independent set:

\[
\begin{bmatrix}
I \\
-A^T_D A^T_f
\end{bmatrix}
\delta q_l = F \delta q_l
\]

Stationarity of the functional \( \mathcal{F}(q) \) can now be written

\[
\delta \mathcal{F} = \delta q \cdot f = \delta q_l \cdot F^T f = 0
\]

where \( f = \frac{\partial \mathcal{F}}{\partial q} \)

Since the \( \delta q_l \) are independent and free (unconstrained), the solving equations are

\[
F^T f = 0
\]

+ Pros: minimal set of equations and unknowns

- Cons: need to identify non-singular block \( A_D \)
Constrained Algebraic Problems

The method of Lagrange multipliers

The functional $\mathcal{F}(q)$ is augmented as

$$\mathcal{F}'(q, \lambda) = \mathcal{F}(q) + c \cdot \lambda$$

where $\lambda \in \mathbb{R}^m$ are the Lagrange multipliers

Stationarity of $\mathcal{F}'(q, \lambda)$ yields

$$\delta \mathcal{F}' = \delta q \cdot f + \delta q \cdot A \lambda + \delta \lambda \cdot c = 0$$

and the resulting solving equations are

$$\begin{cases} 
  f + A \lambda = 0 \\
  c = 0
\end{cases}$$

Linearization of the equations for solving with Newton method:

$$\begin{bmatrix}
  B^T + (A \lambda)_q & A \\
  A^T & 0
\end{bmatrix} \begin{bmatrix}
  \delta q \\
  \delta \lambda
\end{bmatrix} = - \begin{bmatrix}
  f + A \lambda \\
  c
\end{bmatrix}$$
Constrained Algebraic Problems

The penalty method

The functional $\mathcal{F}(q)$ is augmented as

$$\mathcal{F}'(q) = \mathcal{F}(q) + \frac{1}{2} p c \cdot c$$

where $p \in \mathbb{R}$ is a large number (penalty parameter).

Stationarity of $\mathcal{F}'(q)$ yields

$$\delta \mathcal{F}' = \delta q \cdot f + p \delta q \cdot Ac = 0$$

and the resulting solving equations are

$$f + p Ac = 0$$

Note: exact solution is recovered only for $p \to \infty$

However, large $p$ implies ill conditioning

Linearization of the equations for solving with Newton method:

$$\left( B^T + p AA^T \right) \delta q = -(f + p Ac)$$
Constrained Algebraic Problems

The **augmented Lagrangian method**

The functional $\mathcal{F}(q)$ is augmented as

$$\mathcal{F}'(q, \lambda) = \mathcal{F}(q) + c \cdot s \lambda + \frac{1}{2} p c \cdot c$$

where $\lambda \in \mathbb{R}^m$ are the Lagrange multipliers, $s$ is a scaling factor for the Lagrange multipliers, and $p \in \mathbb{R}$ is a penalty–like term that however does not need to be large.

Stationarity of $\mathcal{F}'(q, \lambda)$ yields

$$\delta \mathcal{F}' = \delta q \cdot f + \delta q \cdot A s \lambda + \delta \lambda \cdot s c + p \delta q \cdot A c = 0$$

and the resulting solving equations are

$$f + A s \lambda + p A c = 0$$

$$s c = 0$$
Constrained Algebraic Problems

The **augmented Lagrangian method** (cont.)

Linearization of the equations for solving with Newton method yields:

\[
\begin{bmatrix}
B^T + s(A\lambda)_q + p(Ac)_q + pAA^T sA & sA^T \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta q \\
\delta \lambda
\end{bmatrix}
= - \begin{bmatrix}
f + A(s\lambda + p c) \\
s c
\end{bmatrix}
\]

The term \(pAA^T\) improves the conditioning of the Jacobian, leading to better convergence.

The augmented Lagrangian method is then a combination of the method of Lagrange multipliers with scaling and the penalty method.

The role of the penalty parameter is to **improve conditioning**, not to enforce the constraint conditions.
Constrained Dynamic Problems

The same solution techniques developed for algebraic problems can be used also for **dynamic problems**:

- Reduction to an independent set of unknowns: not recommended, identification of independent set not trivial and might vary with time; complex dense set of solving equations
- The method of Lagrange multipliers: sparse set of equations, ideal for FEM; presence of algebraic variables (multipliers) leads to DAEs and needs special numerical methods
- The penalty method: possible severe ill-conditioning
- The augmented Lagrangian method: same as Lagrangian, but with improved conditioning
Constrained Dynamic Problems

For dynamic unconstrained problems, the governing equations are obtained through Hamilton’s Principle:

\[ \delta \mathcal{F}(q, \dot{q}) = \int_T (\delta L + \delta W) \, dt = 0 \]

where \( L \) is the Lagrangian

\[ L = K - V \]

\( K \) is the kinetic energy, \( V \) is the potential energy, and \( \delta W \) is the virtual work of the external forces

\[ \delta W = \delta q \cdot f_e \]

To treat constrained problems, the functional \( \mathcal{F}(q, \dot{q}) \) is augmented as in the algebraic case.
The **method of Lagrange multipliers**

For holonomic constraints, the functional \( F(q, \dot{q}) \) is augmented as

\[
\delta F'(q, \dot{q}) = \int_T (\delta L + \delta W + \delta (c \cdot \lambda)) \, dt = 0
\]

which yields the governing equations for holonomically constrained multibody systems in first order form

\[
\begin{aligned}
\frac{d}{dt}(Mw) + f + A\lambda - f_e &= 0 \\
N\dot{q} - w &= 0 \\
c &= 0
\end{aligned}
\]

Generalized coordinates \( q = (d, r) \) (\( d \) : displacements, \( r \) : rotations),

Generalized velocities \( w = (v, \omega) \)

Mass matrix \( M \) and

\[
N = \begin{bmatrix}
I & 0 \\
0 & S
\end{bmatrix}
\]

(Recall, angular velocities are not time rates of rotational parameters)
Theorem of total mechanical energy

Differentiating the constraint \( c = 0 \) with respect to time yields:

\[
\dot{c} = A^T w + c_t = 0 \quad (c_t = 0 \text{ for scleronomic constraints})
\]

Dot multiplying the equations of equilibrium by the velocity, we have:

\[
w \cdot \left( \frac{d(Mw)}{dt} + f + A\lambda - f_e \right) = \dot{E} + \lambda \cdot A^T w - f_e \cdot w
\]

\[
= \dot{E} - \lambda \cdot c_t - f_e \cdot w
\]

\[
= \dot{E} - P_c - P_e
\]

\[
= 0
\]

\( P_c \): power of rheonomic constraints; \( P_e \): power of external forces.

Time rate of change of total mechanical energy is equal to the power of the external forces plus the power of the rheonomic constraints

- Ideal scleronomic joints do not absorb nor produce power
Joint Models

**Joints** (or pairs) connect two bodies allowing some kind of relative motion.

- **Lower pairs**: surface contact between elements
  - They are the simplest, but most useful and common in applications
  - There are only 6 possible surface contact joints

- **Higher pairs**: point or line contact between elements
  Occasionally, can be expressed as combinations of lower pairs
Lower Pairs

\[ S \]
Inertial frame

\[ S^A = (e^A_1, e^A_2, e^A_3) \]
Body A frame

\[ S^B = (e^B_1, e^B_2, e^B_3) \]
Body B frame

Allowed relative displacements and rotations for the 6 lower pairs:

<table>
<thead>
<tr>
<th>Joint type</th>
<th>Relative displacements</th>
<th>Relative rotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revolute</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Prismatic</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Screw</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Planar</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Spherical</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Lower Pairs

Allowed relative displacements and rotations:

- **Revolute**
  - $e_1^A$, $e_1^B$
  - $e_2^A$, $e_2^B$
  - $e_3^A$, $e_3^B$
  - $\theta_3$

- **Prismatic**
  - $e_1^A$, $e_1^B$
  - $e_2^A$, $e_2^B$
  - $e_3^A$, $e_3^B$
  - $d_3$

- **Screw**
  - $e_1^A$, $e_1^B$
  - $e_2^A$, $e_2^B$
  - $e_3^A$, $e_3^B$
  - $d_3 = p \theta_3$

- **Cylindrical**
  - $e_1^A$, $e_1^B$
  - $e_2^A$, $e_2^B$
  - $e_3^A$, $e_3^B$
  - $d_3$
  - $\theta_3$
Lower Pairs

Allowed relative displacements and rotations:

Planar

Spherical
Lower Pairs

When \( d_1 = d_2 = d_3 = 0 \), we have \( d^A = d^B \)

Use Boolean identification of translational dofs of bodies A and B

All other constraints can be written using one of these equations:

\[
e^A_i \cdot (d^A - d^B) - d_i = 0
\]

\[
\cos \theta_i (e^A_j \cdot e^B_k) - \sin \theta_i (e^A_k \cdot e^B_k) = 0
\]

Note:

\[
e^A_j \cdot e^B_k = \cos(\theta_i + \pi/2) = \sin \theta_i
\]

\[
e^A_k \cdot e^B_k = \cos \theta_i
\]
Other Joints

**Sliding joints** account for the contact conditions between a rigid and a flexible body:

- Sliding cylindrical
- Sliding Prismatic
- Sliding screw

- Point on curve
- Point on surface
Other Joints

**Contact conditions** between a rigid and a flexible body.

**Beam:**
\[ d^A = d^B(\xi) \quad 0 \leq \xi \leq 1 \]
\[ d^B(\xi) = \sum_{i=1}^{n_{\text{nodes}}} h_i(\xi) d^B_i \]

where \( h_i(\xi) \) are the shape functions, and \( d^B_i \) the positions of the beam nodes. \( \xi \) is the unknown point of contact.

**Shell:**
\[ d^A = d^B(\xi, \eta) \quad 0 \leq \xi \leq 1 \quad 0 \leq \eta \leq 1 \]
\[ d^B(\xi, \eta) = \sum_{i=1}^{n_{\text{nodes}}} h_i(\xi, \eta) d^B_i \]

Rotational constraints are enforced as for the lower pairs.
Temporal Integration

Issues in the selection of appropriate numerical integration schemes:

- Explicit vs. implicit
  - Low frequency aero-elastic temporal scales
  - Possible high frequency scales in actuator models
  - Contact/impact phenomena and the need for time refinement
  - Order (2nd or higher)
- Full FEM or modal based approaches, and the need for high frequency damping
- DAE multibody formulations:
  - Enforcement of constraints for index-3 DAE
  - Stiffness and need for high frequency damping
  - Ill conditioning and Lagrange multiplier scaling

No general treatment on these issues in these notes
Temporal Integration

**Unresolved scales** in FEM and temporal processes

All numerical processes are based on the selection of a grid

The sole fact of selecting a grid, implies that some scales will not be accurately resolved (Nyquist Theorem)

Some of the unresolved scales might (will) appear as abnormally amplified in the computed solution.

This is a universal limitation of all numerical schemes
Temporal Integration

**Example:** integration failure of flexible constrained mechanical system due to pollution from high frequency modes (artifacts of spatial discretization process)

Non-linearities provide a mechanism to excite the higher non-physical modes.
Temporal Integration

Properties of constrained mechanical systems

Classical approach: derive the equations and apply an off-the-shelf general-purpose DAE integrator

Pros: easy

Cons: the integrator knows nothing about the problem being solved
- Invariants are not preserved, only linear notions of stability
- Lack of robustness, failure for particularly difficult problems
Temporal Integration

**Geometric integration theory**

Design (backward–engineer) integrators that incorporate specific knowledge of the equations being solved:

- Exact treatment of geometric non–linearities
- Exact satisfaction of the constraints (no drift)
- Non–linear unconditional stability and preservation of invariants:
  - Bound on total energy of deformable bodies + Vanishing of the work of constraint forces + Conservation of momenta

The numerical procedure inherits qualitative features of the true solution


\[ T_f = T_i \quad (T = K + V) \]
**Temporal Integration**


\[ T_f = T_i - T_D \quad T_D \geq 0 \text{ dissipated total energy} \]

- Unconditional stability in the non-linear regime from the bound on the total energy
- Mechanism for controlling the unresolved frequencies

For a single degree of freedom linear oscillator model problem:

- \( \alpha = 1 \), asymptotic annihilation
- \( \alpha = 0 \), energy preserving scheme recovered

\[ (K_f + V_f) - (K_i + V_i) + \alpha e^2 = 0. \]

\[ \rho_\infty = \frac{(1 - \alpha)}{(1 + \alpha)} \]
Temporal Integration

Energy Decaying scheme (solid triangles) vs. modified-\(\alpha\) (Chung & Hulbert, 1995) (empty triangles)

Energy Decaying scheme accuracy:
- Linear model problem: fourth order \(\alpha = 0\), third order \(\alpha = 1\)
- In general, between second and third order
Passive actuator models

- First order actuator:
\[ \dot{y}(t) + \frac{1}{\tau}(y(t) - u(t)) = 0 \]

- Second order actuator:
\[ \ddot{y}(t) + 2\xi\omega\dot{y}(t) + \omega^2(y(t) - u(t)) = 0 \]

where:
- \( \tau \) = time constant
- \( \omega \) = natural frequency
- \( \xi \) = damping ratio
- \( u \) = input
- \( y \) = output
Refined actuator models

Example, pitch actuator:
- Compute bearing internal reaction from blade inertial and aerodynamic loads
- Compute bearing friction based on reaction (bending, axial, etc.)
- Given pitch demand, actuator controller (e.g. PID) computes necessary torque
- Apply torque limits
- Apply limited torque about blade pitch axis
Actuator Models

Basic hydraulic elements: building blocks for more complex hydraulic components

1) Hydraulic chamber

2) Hydraulic orifice

3) Pressure relief valve:
**Actuator Models**

**Derived hydraulic elements:**

Hydraulic linear actuator
(2 chambers + 2 orifices):

Simple hydraulic damper
(2 chambers + 1 orifice):
Actuator Models

Non-linear formulation of hydraulic elements must be fully compatible with other multibody components

- Structural response: dictated by frequency content of structural elements
- Hydraulic response: dictated by bulk modulus of the fluid (very stiff equations, very small time steps)
- Decoupling strategy with sub-cycling for efficiency:
  - Structural steps (implicit EDS)
  - Hydraulic steps (explicit RK)
Fluid Dynamics Models
Rotor Aerodynamic Models

Aerodynamic models:

- **Lifting lines** based on two-dimensional strip theory:
  - Associable with any beam in the model
  - Lifting line and beam reference line of the same blade can be different, for maximum generality
  - Curved and twisted lifting lines, described by NURBS curves
  - Airfoil aerodynamic characteristics stored in table look-up form
  - An arbitrary number of airfoils can be associated with any lifting line
  - Tip losses, radial flow, unsteady correction, dynamic stall

- **Inflow models:**
  - Blade element momentum (BEM) theory
  - Dynamic Pitt–Peters model and Peters–He wake models

- **Generic interface** to external CFD or free wake codes
Example of servo/structural and aerodynamic model of rotor:

- Span-varying structural properties
- Command input from controller
- Beam internal force sensor
- Pitch hinge
- Aerodynamic force sensor
- Inflow model couples lifting lines
- Turbulent wind grid
- Tower shadow and wind shear
- Turbulent wind component
- 1st or 2nd order actuator
- Angular speed sensor
- Curved and twisted beam reference line
- Curved and twisted lifting line
Lifting Line Kinematics

- Interpolate velocities \( \mathbf{v}_P \), \( \mathbf{\omega} \) at point \( P \) on structural reference line from FEM solution
- Compute velocity at air–station \( A \)

\[
\mathbf{v}_A = \mathbf{v}_P + \mathbf{\omega} \times \mathbf{r}_{PA}
\]
Lifting Line Kinematics

- Compute local flow velocity at air-station

\[ \mathbf{v} = \mathbf{v}_W + \mathbf{v}_I - \mathbf{v}_A \]
Lifting Line Kinematics

- Compute local angle of attack and Reynolds number

\[ \tan \alpha = \frac{v}{u} \]

\[ Re = \frac{\rho c U_T}{\nu} \]
Lifting Line Loads

- Compute lift, drag and moment at air-station using experimental look-up tables

\[ C_L = C_L(\alpha, Re) \quad C_D = C_D(\alpha, Re) \quad C_M = C_M(\alpha, Re) \]
Lifting Line Loads

- Correct for unsteady effects (e.g. using thin airfoil theory)

\[ C_L = C_L + C_{L,\alpha} \left( \frac{c\dot{\alpha}}{4U_T^2} + \frac{c\dot{\alpha}}{4U_T} + \frac{c^2\ddot{\alpha}}{16} \right) \]

\[ C_M = C_M - \frac{C_{L,\alpha}}{4} \left( \frac{c\dot{\alpha}}{4U_T^2} + \frac{c\dot{\alpha}}{2U_T} + \frac{3c^2\ddot{\alpha}}{32} \right) \]

- Correct for dynamic stall (e.g. ONERA, Leishman–Beddoes)

\[ C_L = C_L + \Delta C_{L,\text{dyn stall}} \]
\[ C_D = C_D + \Delta C_{D,\text{dyn stall}} \]
\[ C_M = C_M + \Delta C_{M,\text{dyn stall}} \]

- Correct for tip \( F_t \) (e.g. Prandtl) and hub losses \( F_h \)

\[ C_L = F_t F_h C_L \quad C_D = F_t F_h C_D \quad C_M = F_t F_h C_M \]
\[ F_t = F_t \left( \frac{r}{R} \right) \quad F_h = F_h \left( \frac{r}{R} \right) \]
Lifting Line Loads

- Air-load components:
  
  \[ F_x = \frac{1}{2} \rho c U_T (v C_L + u C_{Dp}) + \frac{1}{2} \rho c U u C_f \]
  
  \[ F_y = \frac{1}{2} \rho c U_T (u C_L + v C_{Dp}) + \frac{1}{2} \rho c U v C_f \]
  
  \[ F_z = \frac{1}{2} \rho c U w C_f \]
  
  \[ M_z = \frac{1}{2} \rho c^2 U_T^2 C_M \]

Pressure drag: \( C_{Dp} = C_D - C_f \)   
Skin friction drag: \( C_f \approx 0.006 \)

- Transport load resultant to beam reference line:
  
  \[ \mathbf{s} = (F_x, F_y, F_z)^T \]
  
  \[ \mathbf{m}_P = \mathbf{m}_A + \mathbf{s} \times \mathbf{r}_{PA} \]
  
  \[ \mathbf{m}_A = (0, 0, M_z)^T \]
Inflow Models

One dimensional annular stream tube theory:

Axial thrust: \[ dT = \dot{m}(V - V_D) = dA(p_0 - p_1) \]

Bernoulli’s Th.: \[ p + \frac{1}{2} \rho V^2 = p_0 + \frac{1}{2} \rho (V - v_i)^2 \]

\[ p_1 + \frac{1}{2} \rho (V - v_i)^2 = p + \frac{1}{2} \rho V_D^2 \]

Results:

\[ dT = \rho V^2 4a(1 - a) \pi rdr \quad V_D = V(1 - 2a) \quad a \leq 1/2 \]
Inflow Models

Assuming uniform inflow across the rotor disk:

Thrust: \[ T = 2\rho AV^2 a(1 - a) \quad C_T = \frac{T}{1/2\rho AV^2} = 4a(1 - a) \]

\[ C_{T_{\text{max}}} = 1 \text{ for } a = 1/2 \]

Power: \[ P = Q\Omega = T(V - v_i) \quad C_P = \frac{P}{1/2\rho AV^3} = 4a(1 - a)^2 \]

\[ C_{P_{\text{max}}} = 16/27 \approx 0.593 \text{ for } a = 1/3 \]
Inflow Models

One dimensional annular stream tube theory with wake swirl:

Rotor torque: \[ dQ = d\dot{m}(\omega r)r \]

\[ = (\rho(V - v_i)2\pi r dr)(\omega r)r \]

\[ = \rho V \Omega 4a'(1 - a)\pi r^3 dr \]
Inflow Models

Combined lifting line – one dimensional annular stream tube theory with wake swirl (Blade Element Momentum, BEM):

- Compute annular thrust $dT$ and torque $dQ$ from lifting lines
- Solve for axial and angular induction factors
  \[
  \rho V^2 4a(1 - a)\pi r dr = dT
  \]
  \[
  \rho V\Omega 4a'(1 - a)\pi r^3 dr = dQ
  \]
- Compute local inflow: $v_I = aVn + a'\Omega rt$
- Iterate until convergence
Inflow Models

- **Finite-state dynamic inflow models**
  - Pitt Peters 3-state model
    \[ \mathbf{v}_I = (v_0 + v_s \sin \psi + v_c \cos \psi) \mathbf{n} \]
    \[
    M \frac{d}{d(Vt/R)} \left\{ \begin{array}{c} v_0 \\ v_s \\ v_c \end{array} \right\} + L^{-1} \left\{ \begin{array}{c} v_0 \\ v_s \\ v_c \end{array} \right\} = \left\{ \begin{array}{c} C_T \\ C_l \\ C_m \end{array} \right\}
    \]
  - Peters He \( n \)-state model (hierarchical, contains Pitt Peters)

- **Free wake models**
CFD Coupling

- Compute geometric configuration and velocity at air-station
- Move and deform fluid dynamic mesh according to geometry
- Apply boundary conditions according to motion
- Solve flow problem
- Integrate pressures at air-station local span to get air-loads
- Transport air-loads to beam reference line
Wind Models

Aerodynamic model (continued):

- **Wind models** (according to IEC 61400–1):
  - **Deterministic gusts** (EOG1, ECG)
  - **3D stochastic turbulent wind**:
    - Pre-computed before the beginning of the simulation for an assigned duration of time and for a user-specified two-dimensional grid of points
    - Steady mean wind + stochastic component based on von Karman and Kaimal turbulence
  - **Wind shear**: exponential and logarithmic models

- **Tower shadow**: potential flow model for a conical tower, downwind empirical Powles model, or an interpolation of the two
Wind Models

Wind speed distribution

Probability distribution function, used to describe the distribution of wind speeds over an extended period of time

Weibull distribution

\[ F_w = 1 - \exp \left( -\left( \frac{V_0}{C} \right)^k \right) \]
\[ f_w = -k \frac{V_0^{k-1}}{C^k} \exp \left( -\left( \frac{V_0}{C} \right)^k \right) \]

\( F_w(V_0) \) = cumulative probability function, i.e. probability that \( V < V_0 \)

\( f_w(V_0) \) = probability density function

\( V_0 \) = wind speed

\( C \) = scale parameter \( (C = c\bar{V} \text{, } c = 1/\Gamma(1 + 1/k)) \)

\( k \) = shape parameter

Example: annual energy yield given power-speed curve \( P(V_0) \)

\[ E = Y \int_{V_{\text{in}}}^{V_{\text{out}}} P(V_0) f_w dV_0 \quad Y = \text{year length} \]
Wind Models

Wind profile – wind shear law

Logarithmic or power law profiles

\[ V(z) = V(z_r) \frac{\ln(z/z_0)}{\ln(z_r/z_0)} \quad \text{for fitting the profile} \]

\[ V(z) = V(z_r) \left( \frac{z}{z_r} \right)^\alpha \]

- \( V(z) \) = wind speed at height \( z \)
- \( z \) = height above ground
- \( z_r \) = reference height above ground
- \( z_0 \) = roughness length
- \( \alpha \) = wind shear (power law) exponent

Normal Wind Profile (NWP):

\[ V(z) = V_{\text{hub}} \left( \frac{z}{z_{\text{hub}}} \right)^{0.2} \]
Wind Models

**Extreme wind conditions**

The extreme wind conditions include wind shear events, as well as peak wind speeds due to storms and rapid changes in wind speed and direction.

**Extreme wind speed model (EWM)**

\[ V_{e50}(z) = 1.4V_{\text{ref}} \left( \frac{z}{z_{\text{hub}}} \right)^{0.11} \]

\[ V_{e1}(z) = 0.8V_{e50}(z) \]

\[ V_{\text{ref}} = \text{reference wind speed used for defining wind turbine classes} \]
风力机建模

极端操作风速（EOG）

\[
V(z, t) = \begin{cases} 
V(z) - 0.37V_{\text{gust}} \sin(3\pi t/T)(1 - \cos(2\pi t/T)) & \text{for } 0 \leq t \leq T \\
V(z) & \text{otherwise}
\end{cases}
\]

\[
V_{\text{gust}} = \min \left(1.35(V_e - V_{\text{hub}}, 3.3(\sigma_1/(1 + 0.1D/\Lambda_1))))\right)
\]

\[
\sigma_1 = \text{湍流标准偏差}
\]

\[
\lambda_1 = \text{湍流尺度参数}
\]

\[
D = \text{叶轮直径}
\]

\[
T = \text{风速持续时间 } (T = 10.5 \text{ 秒})
\]

图示：极端操作风速
Wind Models

Extreme Direction Change (EDC)

\[ \theta(t) = \begin{cases} 
0 \text{ deg} & \text{for } t < 0 \\
\pm 0.5 \theta_e (1 - \cos(\pi t/T)) & \text{for } 0 \leq t \leq T \\
\theta_e & \text{otherwise} 
\end{cases} \]

\[ \theta_e = \pm 4 \arctan(\sigma_1 / (V_{\text{hub}} (1 + 0.1(D/\Lambda_1)))) \]

Duration: \( T = 6 \text{ s} \)
Extreme coherent gust with direction change (ECD)

\[ V(z, t) = \begin{cases} 
V(z) & \text{for } t < 0 \\
V(z) - 0.5V_{cg}(1 - \cos(\pi t/T)) & \text{for } 0 \leq t \leq T \\
V(z) + V_{cg} & \text{for } t > T 
\end{cases} \]

\[ V_{cg} = 15 \text{ m/s} \]

\[ \theta(t) = \begin{cases} 
0 \text{ deg} & \text{for } t < 0 \\
\pm 0.5 \theta_{cg}(1 - \cos(\pi t/T)) & \text{for } 0 \leq t \leq T \\
\pm \theta_{cg} & \text{for } t > T 
\end{cases} \]

\[ \theta_{cg} = \begin{cases} 
180 \text{ deg} & \text{for } V_{hub} < 4 \text{ m/s} \\
720 \text{ deg}/V_{hub} & \text{for } 4 \text{ m/s} \leq V_{hub} \leq V_{ref} 
\end{cases} \]
Wind Models

Extreme wind shear (EWS)

Vertical shear

\[
V(z, t) = \begin{cases} 
V_{\text{hub}} \left( \frac{z}{z_{\text{hub}}} \right)^\alpha \pm \left( \frac{z-z_{\text{hub}}}{D} \right) \left( 2.5 + 0.2\beta \sigma_1 \left( \frac{D}{\Lambda_1} \right)^{0.25} \right) (1 - \cos(\pi t/T)) & \text{for } 0 \leq t \leq T \\
V_{\text{hub}} \left( \frac{z}{z_{\text{hub}}} \right)^\alpha & \text{otherwise}
\end{cases}
\]

Horizontal shear

\[
V(y, z, t) = \begin{cases} 
V_{\text{hub}} \left( \frac{z}{z_{\text{hub}}} \right)^\alpha \pm \left( \frac{y}{D} \right) \left( 2.5 + 0.2\beta \sigma_1 \left( \frac{D}{\Lambda_1} \right)^{0.25} \right) (1 - \cos(\pi t/T)) & \text{for } 0 \leq t \leq T \\
V_{\text{hub}} \left( \frac{z}{z_{\text{hub}}} \right)^\alpha & \text{otherwise}
\end{cases}
\]

\[\alpha = 0.2\]
\[\beta = 6.4\]
\[T = 12 \text{ s}\]
Wind Models

Normal Turbulence Model (NTM)

Turbulence standard deviation:

$$\sigma_1 = I_{\text{ref}} (0.75V_{\text{hub}} + b) \quad b = 5.6 \text{ m/s}$$

$I_{\text{ref}}$ = expected value of turbulence intensity at 15 m/s

$V_{\text{ref}}$ = reference wind speed averaged over ten minutes

$V_{\text{hub}}$ = wind speed at hub height averaged over ten minutes

<table>
<thead>
<tr>
<th>Wind Turbine Class</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{\text{ref}}$ (m/s)</td>
<td>50</td>
<td>42.5</td>
<td>37.5</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>$I_{\text{ref}}$ (-)</td>
<td>0.16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$I_{\text{ref}}$ (-)</td>
<td>0.14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>$I_{\text{ref}}$ (-)</td>
<td>0.12</td>
<td>Values specified by designer</td>
<td></td>
</tr>
</tbody>
</table>
Normal Turbulence Model (NTM)

According to IEC–61400–1, Ed.3, the following turbulence models may be used for design load calculations:

1. Kaimal spectral and exponential coherence model
2. Mann uniform shear model

Kaimal spectral and exponential coherence model with 13m/s mean wind speed:

- Turbulence intensity $s_1 = 5\%$
- Turbulence intensity $s_1 = 16\%$ (wind turbine Class A)
Wind Models

**Tower shadow model**
Accounts for wind distortion due to tower

\[ V_s(x, y, z) = A(x, y, z)V \]

- Potential flow model (upwind turbine)
  \[ A = 1 + \frac{D(z)^2(y^2 - x^2)}{4(y^2 + x^2)^2} \]
- Empirical model of Powles (downwind turbine)
  \[ A = 1 - \Delta \cos^2\left(\frac{\pi x}{WD}\right) \]

**Blending function**

\[ A_b = A(0.5 - \cos(\psi) + (0.5 + \cos \psi)) \]
Other Important Topics not yet Covered in These Notes

- Upwind turbine wake models
- Current and wave models for off-shore applications
Aero–servo–elasticity, Loads and Fatigue
**Analysis Types**

**Static analysis:**
- Yields deformed structural configuration under steady loads:
  - Prescribed external forces
  - Inertial loads due to rotation of elements of the model at constant prescribed angular velocity
  - Steady aerodynamic loads
- It is an approximation: gravity neglected (1P) and aerodynamic loads are not steady (even in non turbulent wind) because of
  - Wind shear
  - Tower shadow
  - Non axial flow (rotor up-tilt, yaw)
- Useful for:
  - Finding approximate deflected configuration for eigenanalysis
  - Computing initial conditions for a subsequent dynamic analysis, reducing transients to arrive to trimmed periodic response
Analysis Types

**Eigenanalysis:**
- Yields natural frequencies and eigenmodes
- Analysis of isolated undeflected subcomponents (tower, blade, rotor)
- Analysis of complete model for varying angular and wind speed
  - Compute deflected configuration
  - Remove aerodynamic loads
  - Perform eigenanalysis in a vacuum about deflected configuration

1st rotor collective flap  2nd rotor collective flap
### Analysis Types

**The rotor as a filter**

Periodic trimmed condition in non-turbulent wind: all blades have same motion and loads

**Fore–aft tower force:**

- Rotating frame: complex Fourier series expansion of blade shear
  \[ T^m = \sum_{n=-\infty}^{\infty} T_n e^{in\psi_m} \]
  
  - Non–rotating frame: total fore–aft force on tower
  \[ T = \sum_{m=1}^{B} T^m = B \sum_{p=-\infty}^{\infty} T_p B e^{ipB\psi} \]

Only \( pB/\text{rev} \) harmonics are transmitted to the tower

\[
\psi_m = \psi + m\Delta\psi \\
m = 1, \ldots, B \\
\Delta\psi = \frac{2\pi}{B} \\
\]
Analysis Types

**Rotor torque:**

Rotating frame: complex Fourier series expansion of in-plane blade shear and bending moment

\[
D^m = \sum_{n=-\infty}^{\infty} D_n e^{i n \psi_m}
\]

\[
Q^m = \sum_{n=-\infty}^{\infty} Q_n e^{i n \psi_m}
\]

- Non-rotating frame: total torque

\[
Q = \sum_{m=1}^{B} (Q^m + R_{\text{root}} D^m) = B \sum_{p=-\infty}^{\infty} (Q_{pB} + R_{\text{root}} D_{pB}) e^{i pB \psi}
\]

Transmission of the sole $pB/\text{rev}$ harmonics
Analysis Types

Side–side tower force:

Rotating frame: complex Fourier series expansion of in–plane blade shear and axial force

\[ D^m = \sum_{n=-\infty}^{\infty} D_n e^{in\psi_m} \quad N^m = \sum_{n=-\infty}^{\infty} N_n e^{in\psi_m} \]

- Non–rotating frame: total side force

\[ S = \sum_{m=1}^{B} (N^m \sin \psi_m + D^m \cos \psi_m) \]

\[ S = \frac{B}{2} \sum_{p=-\infty}^{\infty} \left( \frac{1}{i} N_{pB-1} - \frac{1}{i} N_{pB+1} + D_{pB-1} + D_{pB+1} \right) e^{ipB\psi} \]

\( pB/\text{rev} \) harmonics caused by rotating \( pB\pm1/\text{rev} \) harmonics
Analysis Types

Campbell diagram

$1^{st}$ rotor collective

$2^{nd}$ tower side-side

$2^{nd}$ tower fore-aft

$2^{nd}$ blade flapwise

$1^{st}$ blade edgewise

$1^{st}$ blade flapwise

$1^{st}$ rotor collective

$2^{nd}$ tower side-side

$2^{nd}$ tower fore-aft

$2^{nd}$ blade flapwise

$1^{st}$ blade edgewise

$1^{st}$ blade flapwise

$1^{st}$ rotor collective

Operating regime
Dynamic response analysis:
- Implicit or explicit time marching, depending on formulation
- Time step size chosen based on stability and accuracy
- Automatic time step selection
- Restart capability from a previous static or dynamic analysis

Useful productivity tools:
- Automated run of parametric simulations
- Automated output file naming and segregation
- Search for peaks and ultimate load cases

EOG1 with electrical network loss
# Design Load Cases

<table>
<thead>
<tr>
<th>Design situation</th>
<th>Wind condition</th>
<th>Other conditions</th>
<th>Type of analysis</th>
<th>Partial safety factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Power production</td>
<td>NTM ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td>For extrapolation of extreme events</td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>NTM ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td></td>
<td>F</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>ETM ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td></td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>ECD ( V_{\text{hub}} = V_r \pm 2\text{m/s}, V_r, V_r \pm 2\text{m/s} )</td>
<td></td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>EWS ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td></td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td>2) Power production plus occurrence of fault</td>
<td>NTM ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td>Control system fault or loss of electrical network</td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>NTM ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td>Protection system or preceding internal electrical fault</td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>EOG ( V_{\text{hub}} = V_r \pm 2\text{m/s} ) ( V_{\text{out}} )</td>
<td>External or internal electrical fault including loss of electrical network</td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>NTM ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td>Control, protection, or electrical system faults including loss of electrical network</td>
<td>F</td>
<td>*</td>
</tr>
<tr>
<td>3) Start up</td>
<td>NWP ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td></td>
<td>F</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>EOG ( V_{\text{hub}} = V_r \pm 2\text{m/s}, V_{\text{out}} )</td>
<td></td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>EDC ( V_{\text{hub}} = V_r \pm 2\text{m/s} ) ( V_{\text{out}} )</td>
<td></td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td>4) Normal shut down</td>
<td>NWP ( V_{\text{in}} &lt; V_{\text{hub}} &lt; V_{\text{out}} )</td>
<td></td>
<td>F</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>EOG ( V_{\text{hub}} = V_r \pm 2\text{m/s}, V_{\text{out}} )</td>
<td></td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td>5) Emergency shut down</td>
<td>NTM ( V_{\text{hub}} = V_r \pm 2\text{m/s} ) ( V_{\text{out}} )</td>
<td></td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td>6) Parked (standing still or idling)</td>
<td>EWM 50 year recur. period</td>
<td>Loss of electrical network conn.</td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>EWM 50 year recur. period</td>
<td>Extreme yaw misalignment</td>
<td>U</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>NTM ( V_{\text{hub}} &lt; 0.7 V_{\text{ref}} )</td>
<td></td>
<td>F</td>
<td>*</td>
</tr>
<tr>
<td>7) Parked and fault conditions</td>
<td>EWM 1 year recur. period</td>
<td></td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td>8) Transport, assembly, maint. and repair</td>
<td>To be stated by the manufacturer</td>
<td></td>
<td>U</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>EWM 1 year recur. period</td>
<td></td>
<td>U</td>
<td>A</td>
</tr>
</tbody>
</table>

**Legend**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECD</td>
<td>Extreme coherent gust with direction change</td>
</tr>
<tr>
<td>EDC</td>
<td>Extreme wind direction change</td>
</tr>
<tr>
<td>EOG</td>
<td>Extreme operating gust</td>
</tr>
<tr>
<td>ETM</td>
<td>Extreme turbulence model</td>
</tr>
<tr>
<td>EWM</td>
<td>Extreme wind speed model</td>
</tr>
<tr>
<td>EWS</td>
<td>Extreme wind shear</td>
</tr>
<tr>
<td>NWP</td>
<td>Normal wind profile model</td>
</tr>
<tr>
<td>NTM</td>
<td>Normal turbulence model</td>
</tr>
<tr>
<td>U</td>
<td>Ultimate</td>
</tr>
<tr>
<td>F</td>
<td>Fatigue</td>
</tr>
<tr>
<td>N</td>
<td>Normal</td>
</tr>
<tr>
<td>A</td>
<td>Abnormal</td>
</tr>
<tr>
<td>*</td>
<td>Partial safety for fatigue</td>
</tr>
<tr>
<td>T</td>
<td>Transport</td>
</tr>
</tbody>
</table>
Analysis Types

Data output, post-processing, visualization:
- Sensor measured data output in ASCII data files at each time step (useful for further post-processing in MATLAB or other external software)
- Post-processing tool to produce time-history plots of any quantities of interest in user-specified reference frames
- Graphic tool for model animation of eigenmodes and dynamic responses

Example of animation of wind turbine model with visualization of aerodynamic loads at airstations
Wind Turbine Modeling

Analysis Types

Spectral analysis

- Vibratory phenomenon during regular operation in turbulent wind
- Fast Fourier transform of edgewise blade root bending moment

* Numerical values removed to protect reserved data
**Analysis Types**

**Fatigue analysis**

- Perform simulation in turbulent wind (600 sec)
- Reduce time histories of stochastic loads into simple cyclic loads using **rainflow analysis**
- For each stress level, calculate degree of cumulative damage using **Wöhler S–N curve**
- Combine individual contributions using Palmgren–Miner linear damage rule:

\[
\sum_{i=1}^{k} \frac{n_i}{N_i} = 1 \quad \text{Failure condition}
\]

where

- \( k = \) number of different stress ranges \( R_i \) (bins)
- \( n_i(R_i) = \) number of cycles occurred in the range \( R_i \)
- \( N_i(R_i) = \) number of cycles to failure at the stress level \( R_i \)
  from S–N curve
**Analysis Types**

**Rainflow analysis**

- Input: stress time history
- Output: number of stress cycles binned in terms of amplitude ranges and mean values

- Tower root fore-aft bending moment rainflow counting matrix
- Blade root edge-wise bending moment rainflow counting matrix
- Blade root flap-wise bending moment rainflow counting matrix
Analysis Types

**Stability analysis**: not yet covered in these notes

- Floquet theory
- Autoregressive models
- Proper Orthogonal Decomposition
- ...
Drive Train Modeling
Drive Train Modeling

Topological view of possible drive train multibody model:

- Couplers modeled as flexible joints with internal equivalent springs and dampers
- Bearings with internal friction models depending on internal reaction components
- Shafts modeled with beam elements
- Rigid bodies to account for inertia not included in beam elements
- Nacelle frame modeled as an equivalent beam
Drive Train Modeling

*Contact/impact* analysis in multibody dynamics

Possible applications in wind turbine analysis:
- Gear meshing, dynamic loads, wear, backlash in
  - Gearboxes
  - Yaw drive systems
- Bearings
- Clutches
- Brakes
- Teetering restraints
- Rotor out-of-the-wind yawing or tilting mechanisms
- ...
Contact/Impact in MBD

The different phases of contact:

- **Approach**
- **Compression**
- **Expansion**

Complementarity property of contact:

\[ \mathbf{r} \cdot \mathbf{F}_N = 0 \iff \mathbf{r} = 0, \mathbf{F}_N \neq 0 \text{ or } \mathbf{r} \neq 0, \mathbf{F}_N = 0 \]

The situation is more complicated if friction is present, but the complementarity property still holds.
Locally deformable models:

- Typically FEM based, discretize the equilibrium and constitutive equations in the contact zone
- Yield details of stresses and strains in the contact area
- Also required if the geometry of contact changes profoundly throughout the simulation (e.g. crash) or when details of deformation influence the global dynamics
- General and flexible
- Expensive and potentially difficult to use (need to provide numerous local data which might be hard to predict)
Locally undeformable models:

- The curve shape does not change throughout the simulation
- The candidate contact points are determined based on geometric conditions
- Equivalent local models try to capture the effects of deformability (coefficients of restitution, laws based on local interpenetration (approach), etc.)
- Cheap and simpler to use than deformable models
- Useful when the details of local deformation are not important for the determination of the global system dynamics (typical of most MBD applications)

Local Models
Unilateral Contact Modeling in MBD

We concentrate in the following on the locally undeformable model.

Ingredients:
- **NURBS parameterization** of contacting parts for representing arbitrary geometries.
- **Minimum distance problem** gives the position of the candidate contact points and the relative distance \( q \) (holonomic constraints).

A contact model defines the constitutive law of the interaction forces; two main options: **impulsive** and **continuous models**.
Kinematics of Contact

2D case: regular curves \( u_i : A_i \to \mathbb{R}^2 \) defined by the mapping

\[
C_i := \{ u \in \mathbb{R}^2 : u \in u_i(\xi_i) \forall \xi_i \in A_i \subset \mathbb{R} \}, \quad i = 1, 2
\]

where \( \xi_i \) is the curvilinear coordinate of the curve.

The candidate contact points are determined by the two non-linear holonomic constraints:

\[
\begin{align*}
\mathbf{n}_1(\xi_1) \cdot \mathbf{t}_2(\xi_2) &= 0 \\
\mathbf{r}(\xi_1, \xi_2) \cdot \mathbf{t}_2(\xi_2) &= 0
\end{align*}
\]

The minimum relative distance \( q \) is then:

\[ q = \mathbf{r} \cdot \mathbf{n}_1 \]
Kinematics of Contact

3D case: regular surfaces \( u_i : A_i \mapsto \mathbb{R}^3 \) defined by the mapping

The candidate contact points are determined by the four non-linear holonomic constraints:

\[
\begin{align*}
\mathbf{n}_1(\xi) \cdot \mathbf{s}_2(\xi) &= 0 \\
\mathbf{n}_1(\xi) \cdot \mathbf{t}_2(\xi) &= 0 \\
\mathbf{r}(\xi_1, \xi_2) \cdot \mathbf{s}_2(\xi_2) &= 0 \\
\mathbf{r}(\xi_1, \xi_2) \cdot \mathbf{t}_2(\xi_2) &= 0
\end{align*}
\]

The minimum relative distance \( q \) is then:

\[ q = \mathbf{r} \cdot \mathbf{n}_1 \]
Kinematics of Contact

Normals, tangents and distances appearing in the minimum distance problem need to be **linearized** (solution based on some form of Newton method). E.g., normal:

\[
\delta n = \delta R n^* + R \delta n^* = \partial_d \times R n^* + R g_n d\xi
\]

\[
= -n \times \partial_d + R g_n d\xi
\]

\[
= -n \times S d\psi + R g_n d\xi \quad [\text{with } \partial_d = S d\psi]
\]

When bodies are in contact, this problem yields the approach \( a \):

\[
q > 0 \quad a = -q
\]

\[
q < 0, \quad a = -q
\]
Kinematics of Contact

The complete set of constraints is then:

\[
\begin{align*}
C_1 &= u_1 - d_1 - z_1 = 0 \\
C_2 &= u_2 - d_2 - z_2 = 0 \\
C_3 &= n_1 \cdot t_2 = 0 \\
C_4 &= (u_2 - u_1) \cdot t_2 = 0 \\
C_5 &= q - (u_2 - u_1) \cdot n_1 = 0
\end{align*}
\]

The constraints are enforced using the **Lagrange multiplier** technique:

\[
W_C = \sum_{i=1}^{5} \lambda_i C_i \quad \text{Variations of } W_C \text{ yield: } \delta W_C = \sum_{i=1}^{5} \delta \lambda_i C_i + \sum_{i=1}^{5} \lambda_i \delta C_i
\]
Kinematics of Contact

Relative velocities are computed as:

\[ \mathbf{u} = \mathbf{d} + \mathbf{z} \quad \dot{\mathbf{u}} = \dot{\mathbf{d}} + \dot{\mathbf{z}} \]
\[ = \dot{\mathbf{d}} + \mathbf{R} \mathbf{z}^* \]
\[ = \dot{\mathbf{d}} + \mathbf{\omega} \times \mathbf{z} \]

holding \( \xi \) fixed. Then

\[ \mathbf{v}_R = \dot{\mathbf{u}}_1 - \dot{\mathbf{u}}_2 \]
\[ = \dot{\mathbf{d}}_1 - \dot{\mathbf{d}}_2 - (\mathbf{z}_1 \times \mathbf{\omega}_1 - \mathbf{z}_2 \times \mathbf{\omega}_2) \]

The tangential velocity is then:

\[ \mathbf{v}_T = \mathbf{t}_1 \cdot \mathbf{v}_R \]
\[ = \mathbf{t}_1 \cdot \dot{\mathbf{u}}_1 + \mathbf{t}_2 \cdot \dot{\mathbf{u}}_2 \quad [\mathbf{t}_1 = -\mathbf{t}_2 \text{ at contact}] \]
\[ = \mathbf{t}_1 \cdot \dot{\mathbf{d}}_1 + \mathbf{t}_2 \cdot \dot{\mathbf{d}}_2 - (\mathbf{t}_1 \cdot \mathbf{z}_1 \times \mathbf{\omega}_1 + \mathbf{t}_2 \cdot \mathbf{z}_2 \times \mathbf{\omega}_2) \]

E.g.: Coulomb friction model

Sliding: \[ F_T = -\mu_k(\mathbf{v}_T) \text{sign}(\mathbf{v}_T) F_N \]

Sticking or rolling: \[ F_T \leq \mu_s F_N \]

Similarly for \( \mathbf{v}_N \)
Kinematics of Contact

A simple practical example:

Contact condition: \( q \geq 0 \)

Introduce slack \( r \): \( q - r^2 = 0 \)

Constraint potential: \( W_c = \lambda(q - r^2) \). Variations \( \delta(K + V + W_c) = 0 \)

yield \( m\dot{q}\dot{q} + mg\dot{q} + \delta\lambda(q - r^2) + \lambda\delta q - 2\lambda r \delta r = 0 \)

Resulting equations:

\[
\begin{align*}
\dot{m}\ddot{q} &= mg + \lambda & \text{Equilibrium} \\
\lambda r &= 0 & \text{Complementarity condition} \\
q - r^2 &= 0 & \text{Contact condition}
\end{align*}
\]

Two possible solutions: \( \lambda = 0, \quad r \neq 0, \quad \text{separation} \quad (q > 0) \)
\( \lambda \neq 0, \quad r = 0, \quad \text{contact} \quad (q = 0) \)


**Kinematics of Contact**

Integrate using e.g. the mid point rule:

\[
\begin{align*}
    m \frac{\dot{q}_f - \dot{q}_i}{\Delta t} &= mg + \lambda_m \\
    \frac{\dot{q}_f + \dot{q}_i}{2} &= q_f - q_i \\
    \lambda_m \frac{r_f + r_i}{2} &= 0 \\
    q_f - r_f^2 &= 0 \quad \text{(but also} \quad q_i - r_i^2 = 0) 
\end{align*}
\]

At contact: \( r_m = 0 \Rightarrow r_f = -r_i \Rightarrow q_f = q_i \Rightarrow \dot{q}_f = -\dot{q}_i \Rightarrow \lambda_m = -m \left( g + \frac{2\dot{q}_i}{\Delta t} \right) \)

A discrete version of the principle of impulse and momentum

Note that \( q_f = q_i \) and not \( q_f = 0 \).

Accurate determination of contact event will in general necessitate of **time refinement**
Kinematics of Contact

Handling **multiple contacts**:

To avoid solving minimum distance problems when not necessary, use cheap filters, e.g. bounding spheres or boxes:

$\forall (i, j), \text{ if } r_{ij} \leq R_i + R_j, \text{ minimum distance pb.}$

Need to provide linear initial guess algorithm
Interaction Models

**Interaction model**: constitutive law of the interaction forces

- **Impulsive models**:
  - Null duration of the event (impulsive nature) (can be tough numerically since impulses will excite higher non-physical modes of the MB system)
  - Two different flavors: Newton and Poisson
  - Restitution coefficients model the details of contact (local deformation, local geometry, hysteresis etc.)

- **Continuous models**:
  - Finite duration event
  - Contact force models based on local interpenetration
  - Time refinement (triggered by contact forces) automatically determines contact events with accuracy

Both can model **stick, slip, adhesion** of finite duration etc.
Impulsive Models

**Newton’s law.** Hypothesis: very short duration, frozen configuration during impact.

MBS equations of dynamic equilibrium:

\[
\begin{align*}
\dot{\mathbf{M}}\ddot{\mathbf{u}} &= \mathbf{h} + \mathbf{A}\lambda + \mathbf{A}_C\lambda_C \\
\mathbf{c} &= 0 \\
\mathbf{q}_C &= 0
\end{align*}
\]

with \(\dot{\mathbf{c}} = \mathbf{A}^T\dot{\mathbf{u}} + \mathbf{a}\) and the relative distances \(\mathbf{q}_C = [\ldots, q_i, \ldots]^T, i = 1, m\)

\(\dot{\mathbf{q}}_C = \mathbf{v}_N = \mathbf{A}_C^T\dot{\mathbf{u}} + \mathbf{a}_C\)

\(\mathbf{v}_N = [\ldots, v_{N_i}, \ldots]^T, i = 1, m\)

The relative normal velocities are:

\[
\mathbf{v}_N = \mathbf{n}_1 \cdot \mathbf{v}_R
\]

\[
= \mathbf{n}_1 \cdot \dot{\mathbf{u}}_1 + \mathbf{n}_2 \cdot \dot{\mathbf{u}}_2 \quad [\mathbf{n}_1 = -\mathbf{n}_2 \text{ at contact}]
\]

\[
= \mathbf{n}_1 \cdot \dot{\mathbf{d}}_1 + \mathbf{n}_2 \cdot \dot{\mathbf{d}}_2 - (\mathbf{n}_1 \cdot \mathbf{z}_1 \times \omega_1 + \mathbf{n}_2 \cdot \mathbf{z}_2 \times \omega_2)
\]
Impulsive Models

Impact begins at $t_i$ ends at $t_f$. Integrate over the interval

$$\lim_{t_f \to t_i} \int_{t_i}^{t_f} (M\ddot{u} - h - A\lambda - A_C\lambda_C)dt = 0$$

to get

$$M(\ddot{u}_f - \ddot{u}_i) - A_C\Lambda_C = 0$$

assuming a frozen configuration at impact, and where $\Lambda_C = \lim_{t_f \to t_i} \int_{t_i}^{t_f} \lambda_C dt$

The normal relative velocities at $t_f$ and $t_i$ are

$$v_{N_f} = A_C^T\ddot{u}_i + a_C$$
$$v_{N_i} = A_C^T\ddot{u}_f + a_C$$

So that we find

$$v_{N_f} - v_{N_i} = A_C^T(\ddot{u}_f - \ddot{u}_i) = A_C^T M^{-1} A_C \Lambda_C$$

The problem unknowns are $v_{N_f}, \Lambda_C$, and we need additional relations in order to solve the problem.
Impulsive Models

Newton’s relations between the pre and post impulse velocities:

\[ v_{N_f} = -\varepsilon v_{N_i}, \quad \varepsilon = \text{diag}(\varepsilon_i), \quad 0 \leq \varepsilon_i \leq 1 \]

Coefficients of restitution \( \varepsilon_r \)

\( \varepsilon_f = 0 \): completely inelastic shock; \( \varepsilon_f = 1 \): completely reversible event

With the help of Newton’s relations, we get

\[ -(I + \varepsilon) v_{N_i} = A_C^T M^{-1} A_C \Lambda_C \]

The contact impulses are

\[ \Lambda_C = (A_C^T M^{-1} A_C)^{-1} (I + \varepsilon) v_{N_i} \]

and the MBS velocities

\[ \dot{u}_f = \dot{u}_i - M^{-1} A_C \left( A_C^T M^{-1} A_C \right)^{-1} (I + \varepsilon) v_{N_i} \]

Since the configuration was frozen at impact, this information is enough to restart the computation after contact.
Impulsive Models

**Poisson’s law.** Hypothesis: very short duration, frozen configuration during impact.

Two impact phases: compression (begins at $t_i$, ends at $t_c$), expansion (begins at $t_c$, ends at $t_f$).

MBS equations of dynamic equilibrium, including friction:

\[
M \ddot{\mathbf{u}} = h + A \dot{\lambda} + \left[ A_{CN} A_{CT} \right] \begin{pmatrix} \lambda_{CN} \\ \lambda_{CT} \end{pmatrix}
\]

The normal and tangential relative velocities can be expressed in terms of $\dot{\mathbf{u}}$ as

\[
\begin{pmatrix} v_N \\ v_T \end{pmatrix} = \begin{pmatrix} A_{CN}^T \\ A_{CT}^T \end{pmatrix} \dot{\mathbf{u}} + \begin{pmatrix} a_{CN} \\ a_{CT} \end{pmatrix}
\]
Impulsive Models

**Compression phase.** Integrate over the interval

\[
\lim_{t_c \to t_i \to t_f} \int_{t_i}^{t_c} (\dot{\mathbf{m}} - \mathbf{h} - \mathbf{A} \dot{\lambda} - \left[ \mathbf{A}_{CN} \mathbf{A}_{CT} \right] \begin{cases} \lambda_{CN} \\ \lambda_{CT} \end{cases}) dt = 0
\]

to get

\[
\mathbf{M}(\dot{\mathbf{c}}_{\varepsilon} - \dot{\mathbf{f}}_{i}) - \left[ \mathbf{A}_{CN} \mathbf{A}_{CT} \right] \begin{cases} \Lambda^C_{CN} \\ \Lambda^C_{CT} \end{cases} = 0
\]

**Expansion phase.** Integrate over the interval

\[
\lim_{t_f \to t_c \to t_c} \int_{t_c}^{t_f} (\dot{\mathbf{m}} - \mathbf{h} - \mathbf{A} \dot{\lambda} - \left[ \mathbf{A}_{CN} \mathbf{A}_{CT} \right] \begin{cases} \lambda_{CN} \\ \lambda_{CT} \end{cases}) dt = 0
\]

to get

\[
\mathbf{M}(\dot{\mathbf{c}}_{\varepsilon} - \dot{\mathbf{f}}_{\varepsilon}) - \left[ \mathbf{A}_{CN} \mathbf{A}_{CT} \right] \begin{cases} \Lambda^E_{CN} \\ \Lambda^E_{CT} \end{cases} = 0
\]
Impulsive Models

We find:

\[
\begin{align*}
\begin{cases}
\mathbf{v}_{N_c} & - \mathbf{v}_{N_i} = \mathbf{A}^{T}_{CN} \mathbf{M}^{-1} \mathbf{A}_{CN} \mathbf{A}_{CT} \begin{cases} \Lambda^C_{CN} \\ \Lambda^C_{CT} \end{cases} \\
\mathbf{v}_{T_c} & - \mathbf{v}_{T_i} = \mathbf{A}^{T}_{CT} \mathbf{M}^{-1} \mathbf{A}_{CN} \mathbf{A}_{CT} \begin{cases} \Lambda^E_{CN} \\ \Lambda^E_{CT} \end{cases}
\end{cases}
\end{align*}
\]

The problem unknowns are

\[
\mathbf{v}_{N_f}, \mathbf{v}_{T_f}, \mathbf{v}_{N_c}, \mathbf{v}_{T_c}; \quad \Lambda^C_{CN}, \Lambda^C_{CT}, \Lambda^E_{CN}, \Lambda^E_{CT}
\]

and we need additional relations in order to solve the problem. Poisson’s relations between the compression and expansion impulses:

\[
\Lambda^E_{CN} = \epsilon \Lambda^C_{CN}, \quad \epsilon = \text{diag}(\epsilon_i), \quad 0 \leq \epsilon_i \leq 1
\]

Coefficients of restitution \( \epsilon_j \)
The complete set of conditions that makes the problem solvable is:

\[
\Lambda_{CN}^E = \varepsilon \Lambda_{CN}^C
\]

\[
v_{N_c} = 0
\]

\[
\begin{cases}
\Lambda_{CT_i}^C < \mu_i \Lambda_{CN_i}^C \Rightarrow v_{T_{ei}} = 0 \\
\Lambda_{CT_i}^C = -\mu_i \text{sign}(v_{T_{ei}}) \Lambda_{CN_i}^C
\end{cases}
\]

\[
\begin{cases}
\Lambda_{CT_i}^E < \mu_i \Lambda_{CN_i}^E \Rightarrow v_{T_{fi}} = 0 \\
\Lambda_{CT_i}^E = -\mu_i \text{sign}(v_{T_{fi}}) \Lambda_{CN_i}^E
\end{cases}
\]

The situation can be more complicated, e.g. if \(v_{N_i} < 0\) the Poisson impulse \(\Lambda_{CN_i}^E\) is **not strong enough** and bodies will not separate, i.e. we need to replace

\[
\Lambda_{CN_i}^E = \varepsilon_i \Lambda_{CN_i}^C
\]

with

\[
v_{N_{fi}} = 0
\]
Continuous Models

At contact, introduce the new variable \( a \) (approach):

\[
a = -q
\]

The **complementarity condition** is in this case:

\[
(a + q)F_N = 0
\]

which yields

\[
\begin{cases}
a + q > 0, & F_N = 0, \text{ separation} \\
a + q = 0, & F_N > 0, \text{ contact}
\end{cases}
\]

Need a **phenomenological law** that relates \( F_N \) to \( a \):

\[
F_N = F_N(a, \dot{a}, \ddot{a}, \ldots) \quad F_N = F_N^{\text{Elastic}} + F_N^{\text{Dissipative}}
\]
Continuous Models

Modeling the **elastic part** of the interaction forces $F_N^{\text{Elastic}}$:

- **Linear spring** (simplest possible model): elastic potential

\[
V = \frac{1}{2} k a^2
\]

- **Hertz model**: elastic potential

\[
V = \frac{2}{5} c a^{5/2}
\]

where

\[
c = \frac{4}{3 \left( \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2} \right)} \sqrt{\frac{R_1 R_2}{R_1 + R_2}}
\]

* $E$ = Young’s modulus
* $\nu$ = Poisson ratio
* $R$ = local radius of curvature
Continuous Models

Modeling the **dissipative part** of the interaction forces $F_{N,\text{Dissipative}}$:

**Linear damper** (simplest possible model):

$$F_N^D = c\dot{a}$$

Unfortunately, this model **does not make physical sense**. In fact locally

$$m\ddot{a} + c\dot{a} + k\dddot{a} = 0$$

At separation $a = 0$, $\dot{a} < 0$ and the interaction force is

$$c\dot{a} + k\dddot{a} < 0$$

and therefore adhesive!

Unphysical hysteresis loop:

![Diagram showing unphysical hysteresis loop with labels 'shock', 'adhesion', and axes for $F_N$ and $a$.]
An **improved model** with physical hysteresis:

\[ F_N^D = da^n \dot{a} \]

Characterization of parameter \( d \) based on restitution coefficients:

\[ \dot{a}_f = \varepsilon \dot{a}_i \quad \varepsilon = 1 - \alpha \dot{a}_i \]

The dissipated energy is

\[
\begin{align*}
\Delta E &= \frac{1}{2} m (\dot{a}_f^2 - \dot{a}_i^2) \\
&= \frac{1}{2} m \dot{a}_i^2 (2\alpha \dot{a}_i - \alpha^2 \dot{a}_i^2) \\
&\approx \alpha m \dot{a}_i^3
\end{align*}
\]

If the dissipated energy is small, the deformation energy is approx. equal to the initial kinetic energy:

\[
\frac{1}{2} m \dot{a}_i^2 \approx \int_0^{a_M} k a^n da \quad \Rightarrow \quad \dot{a}_i = \sqrt{\frac{2k}{m(n+1)} a_M^{(n+1)/2}} \quad \Delta E = \alpha \sqrt{\frac{(2k)^3}{m(n+1)^3}} a_M^{3(n+1)/2}
\]
Continuous Models

Analogously, for a generic time instant:

\[
\frac{1}{2}m\dot{a}^2 \approx \frac{1}{2}m\dot{a}_i^2 - \int_0^a ka^n \, da \quad \Rightarrow \quad \dot{a} = \sqrt{\frac{2k}{m(n+1)}} a_M^{n+1} - a^{n+1}
\]

We can now compute the dissipated energy, which gives

\[
\Delta E = 2 \int_0^{a_M} \dot{a}^n \, da = 2 \int_0^{a_M} da^n \dot{a} \, da = d \frac{2k}{m(n+1)} \frac{4}{3(n+1)} a_M^{3(n+1)/2}
\]

Comparing with the previously found expression

\[
\Delta E = \alpha \sqrt{\frac{(2k)^3}{m(n+1)^3}} a_M^{3(n+1)/2}
\]

we finally get \( d = \frac{3}{2} \alpha k \) so that the \textbf{improved model} is \( F_N^D = \frac{3}{2} \alpha ka^n \dot{a} \)
Example: the **revolute joint** (but readily applicable to any joint where a relative distance can be defined)

Modeling of a teetering restraint

Revolute joint constraint conditions:

\[
C_1 = e_3^A \cdot e_1^B = 0 \\
C_2 = e_3^A \cdot e_2^B = 0 \\
C_3 = e_1^A \cdot e_1^B \sin \phi + e_1^A \cdot e_2^B \cos \phi = 0
\]

Define a **relative distance** \( q \) even in this case:

\[
q_1 = R_1 (\phi_1 - \phi) \\
q_2 = R_2 (\phi_2 - \phi)
\]

and add a **contact model**
There are many other important issues in the modeling of unilateral constraints in MBD, e.g.:

- Analysis of wear in joints (with effects on noise, vibration, functionality, etc.)
- Joints with clearance (noise, vibration, etc.)
- Contacts with local non-convex geometry (segmentation, starting points, multiple contacts)
- Automatic time adaption and event determination
- Robust time integration schemes
Model Validation
Model Validation

This topic is not yet covered in these notes

- Measurements and turbine instrumentation for model validation
- Examples
- System identification
Scale Effects
Scale Effects

Applications:

- Understanding scaling effects in small–large wind turbines
- Scaling laws for wind tunnel models

Tools:

- Buckingham π–Theorem (dimensional analysis)
Dimensional Analysis: Buckingham π−Theorem

Buckingham π−Theorem:

*Given the governing equation for a physical system defined by* $n$ *physical variables, which are expressible in terms of* $k$ *independent fundamental quantities, one can construct an equivalent equation involving a set of* $m = n - k$ *dimensionless variables constructed from the original variables*

The dimensionless parameters provide relations which define the scaling laws
Dimensional Analysis: Buckingham $\pi$–Theorem

Consider the generic model:

$$f(p_1, \ldots, p_n, a_1, \ldots, a_k) = 0$$

The $i$–th physical parameter can be expressed in terms of $k$ independent fundamental quantities:

$$p_i = a_1^{d_{i1}} \cdots a_k^{d_{ik}}$$

$d_{i1}, \ldots, d_{ik} =$ dimensions of $p_i$ wrt fundamental quantities

The Dimensional Matrix $\mathcal{D} \in \mathbb{R}^k \times \mathbb{R}^n$ contains as elements $d_{ri}$ the $r$–th dimension of the $i$–th parameter.
Equivalent non-dimensional model:
\[ \phi(\pi_1, \ldots, \pi_m) = 0, \ m = n - k \]

j-th non-dimensional parameter:
\[ \pi_j = p_1^{m_1j} \cdot \ldots \cdot p_n^{m_nj} \]

The exponents are the components of matrix \( \mathcal{M} \in \mathbb{R}^n \times \mathbb{R}^m \) found from the solution of the following system of equations:
\[ \mathcal{D} \ \mathcal{M} = 0 \quad \text{i.e.:} \quad \mathcal{M} = \text{null}(\mathcal{D}) \]

Remarks:
- Set of \( m=n-k \) parameters is not unique
- Selection of proper/representative non-dimensional parameters needs to be guided by physical considerations based on the model under consideration
Consider two systems:

\[ P = \textbf{Physical system} \text{ (full scale)} \]
\[ M = \textbf{Model} \text{ (reduced scale)} \]

with governing equations:

\[
f(p_{1P}, \ldots, p_{nP}, a_{1P}, \ldots, a_{kP}) = 0
\]
\[
f(p_{1M}, \ldots, p_{nM}, a_{1M}, \ldots, a_{kM}) = 0
\]

and equivalent non–dimensional relations obtained through Buckingham \( \pi \)-theorem:

\[
\phi(\pi_{1P}, \ldots, \pi_{nP}) = 0
\]
\[
\phi(\pi_{1M}, \ldots, \pi_{nM}) = 0
\]
Similarity

P is similar to M if:

\[ \pi_{jP} = \pi_{jM}, \text{ for } j = 1, \ldots, m \]

which provides a set of scaling relations:

\[ p_{1P}^{m_{1j}} \cdots p_{nP}^{m_{nj}} = p_{1M}^{m_{1j}} \cdots p_{nM}^{m_{nj}} \]

Generic scaling relation for a model parameter:

\[ p_{1M} = p_{1P} \left( \frac{p_{2P}}{p_{2M}} \right)^{m_{2j}} \cdots \left( \frac{p_{nP}}{p_{nM}} \right)^{m_{nj}} \]
Typically, scaling is based on the assignment of a scaling parameter:

\[ n = \frac{l_M}{l_P} \rightarrow \text{Characteristic length of the model} \]

\[ \frac{l_P}{l_M} \rightarrow \text{Characteristic length of full scale system} \]

Accordingly, the general scaling relation can be expressed as:

\[ p_{1M} = p_{1P} n^{\alpha_1} \cdot \ldots \cdot n^{\alpha_n} \quad p_{1M} = p_{1P} n^{\alpha_1} \]

The scaling process reduces to a linear transformation of parameters:

\[ p_M = S p_P \]

Scaling matrix: \[ S = \text{diag}(n^{\alpha_1}, \ldots, n^{\alpha_n}) \]
Wind Turbine Dimensional Analysis

**Example:** 6 state – 4 input individual-pitch model, with stiffness modeled using equivalent springs

**States:**
- Flap angles $\theta_1, \theta_2, \theta_3$
- Rotor azimuth $\psi$
- Shaft torsion $\epsilon$
- Fore-aft angle $\alpha$
  (and their rates)

**Inputs:**
- Pitch angles $\beta_{c1}, \beta_{c2}, \beta_{c3}$
- Electrical torque $T_{elc}$

Readily generalized to more complex models
Wind Turbine Dimensional Analysis

Derive equations of motion, apply Buckingham \( \pi \)-theorem using as fundamental quantities

\[ \text{Mass, Length, Time} \]

Non-dimensional equations of motion:

\[ \phi(x, \dot{x}, \ddot{x}, u, \pi) = 0 \]

States: \( x = (\psi, \theta_i, \epsilon, \alpha)^T \)

Inputs: \( u = (\beta_i, \tilde{T}_{el_c})^T \)

\( \tilde{T}_{el_c} = \frac{T_{el_c}}{1/2 \rho ARV^2} \)

Non-dimensional parameters

\[ \pi = (Ma, Re, Fr, Lo, \lambda, \tau, \frac{\omega_\alpha}{\Omega}, \frac{\omega_{\theta_i}}{\omega_\alpha}, \frac{\omega_\epsilon}{\omega_\alpha})^T \]
**Wind Turbine Dimensional Analysis**

**Non-dimensional parameters** of physical relevance resulting from dimensional analysis:

- **Mach:** $Ma = V/a$  
  Effect typically negligible for WT

- **Reynolds:** $Re = \rho V c/\nu$  
  Inertial/viscous aerodynamic force ratio

- **Froude:** $Fr = V^2/gR$  
  Aerodynamic/gravitation force ratio

- **Lock:** $Lo = C_L \alpha \rho c R^4/J_\theta$  
  Aerodynamic/inertial force ratio

- **TSR:** $\lambda = \Omega R/V$

- **Non-dim. time:** $\tau = \Omega t$

- **Non-dim. tower freq.:** $\tilde{\omega}_\alpha = \omega_\alpha/\Omega$, $\omega_\alpha = \sqrt{K_\alpha/J_\alpha}$

- **Flapping freq. placement:** $\omega_{\theta_i}/\omega_\alpha$, $\omega_{\theta_i} = \sqrt{K_{\theta_i}/J_{\theta_i}}$

- **Shaft freq. placement:** $\omega_\epsilon/\omega_\alpha$, $\omega_\epsilon = \sqrt{K_\epsilon/J_\epsilon}$
Consider two wind turbines $M$ and $P$ with scale ratio

$$n = \frac{R_M}{R_P}$$

operating in the same wind

$$V_M = V_P$$

at the same TSR

$$\lambda_M = \lambda_P$$
## Size Effects on Wind Turbines

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Symbol</th>
<th>Scaling coefficient</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotor speed</td>
<td>$\Omega_M / \Omega_P$</td>
<td>$n^{-1}$</td>
<td></td>
</tr>
<tr>
<td>Reynolds</td>
<td>$Re_M / Re_P$</td>
<td>$n$</td>
<td></td>
</tr>
<tr>
<td>Froude</td>
<td>$Fr_M / Fr_P$</td>
<td>$n^{-1}$</td>
<td>Gravity important only for very large sizes</td>
</tr>
<tr>
<td>Lock</td>
<td>$Lo_M / Lo_P$</td>
<td>$n^0$</td>
<td>Assuming same density</td>
</tr>
<tr>
<td>Non–dim. freq.</td>
<td>$\tilde{\omega}<em>\alpha M / \tilde{\omega}</em>\alpha P$</td>
<td>$n^0$</td>
<td>Assuming same Young modulus (*)</td>
</tr>
<tr>
<td>Freq. placements</td>
<td>$(\omega_{\theta_i} / \omega_\alpha)<em>M / (\omega</em>{\theta_i} / \omega_\alpha)_P$</td>
<td>$n^0$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(\omega_\epsilon / \omega_\alpha)<em>M / (\omega</em>\epsilon / \omega_\alpha)_P$</td>
<td>$n^0$</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>$P_M / P_P$</td>
<td>$n^2$</td>
<td></td>
</tr>
<tr>
<td>Torque</td>
<td>$Q_M / Q_P$</td>
<td>$n^3$</td>
<td></td>
</tr>
<tr>
<td>Bending stress</td>
<td>$\sigma_M / \sigma_P$</td>
<td>$n^0$</td>
<td>Size indep. stress level</td>
</tr>
<tr>
<td>Shaft shear stress</td>
<td>$\tau_M / \tau_P$</td>
<td>$n^0$</td>
<td>Size indep. stress level</td>
</tr>
</tbody>
</table>

- **Aeroelastic effects unchanged** (Lock and frequencies), except for possible influence of Froude (only large wind turbines)
- **Stress level unchanged** (except for gravity induced loads)

(*) Using a more realistic beam–like natural frequency \( \omega = \sqrt{EJ/(mL^4)} \)
Scaling Laws for Wind Turbine Models

General scaling procedure:

- Given **scale factor**  \( n = R_M / R_P \)
- Find **velocity**  \( n_V = V_M / V_P \) and **time**  \( n_t = t_M / t_P \) scalings

Enforce:

- **Time:**
  \( \tau_M = \tau_P \)  \( (\Omega t)_M = (\Omega t)_P \)

- **Tip-speed-ratio:**
  \( \lambda_M = \lambda_P \)  \( (\Omega R/V)_M = (\Omega R/V)_P \)

- **Lock:**
  \( \text{Lo}_M = \text{Lo}_P \)  \( (C_{L\alpha} \rho c R^4 / J_\theta)_M = (C_{L\alpha} \rho c R^4 / J_\theta)_P \)

Remark: Lock can always be fixed with material density

\[ \rho_{mM} / \rho_{mP} = 1 \]
Scaling Laws for Wind Turbine Models

**Enforce** (continued):

- **Frequency placement:**
  \[
  \tilde{\omega}_\alpha^M = \tilde{\omega}_\alpha^P
  \]
  \[
  (EJ/(mL^4\Omega^4))^M = (EJ/(mL^4\Omega^4))^P
  \]
  \[
  (EJ)^M/(EJ)^P = n^6/n_t^2
  \]

  **Remark:** frequency placement can always be fixed with stiffness

- **Resulting errors:**
  - Reynolds
    \[
    \text{Re}_M/\text{Re}_P = n^2/n_t
    \]
  - Froude
    \[
    \text{Fr}_M/\text{Fr}_P = n/n_t^2
    \]
  - Mach
    \[
    \text{Ma}_M/\text{Ma}_P = n/n_t^2
    \]

**Important remark:** only unknown left is time scaling $n_t$
Scaling Laws for Wind Turbine Models

**Optimal scaling:** minimize Reynolds error (reduce airfoil aerodynamic differences) + scaled time acceleration (reduce active control frequency)

\[
\min \left( k^2 \frac{\text{Re}_M}{\text{Re}_P} + \frac{t_M}{t_P} \right) = \min \left( k^2 \frac{n^2}{n_t + n_t} \right)
\]

which gives mismatched Mach scaling \( n_V = 1/k \)

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Symbol</th>
<th>Scaling coefficient</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotor speed</td>
<td>( \Omega_M / \Omega_P )</td>
<td>( 1/(nk) )</td>
<td></td>
</tr>
<tr>
<td>Reynolds</td>
<td>( \text{Re}_M / \text{Re}_P )</td>
<td>( n/k )</td>
<td></td>
</tr>
<tr>
<td>Froude</td>
<td>( \text{Fr}_M / \text{Fr}_P )</td>
<td>( 1/(nk^2) )</td>
<td>Gravity important only for very large sizes</td>
</tr>
<tr>
<td>Lock</td>
<td>( \text{Lo}_M / \text{Lo}_P )</td>
<td>( n^0 )</td>
<td>Implies same density</td>
</tr>
<tr>
<td>Non-dim. freq.</td>
<td>( \tilde{\omega}_M / \tilde{\omega}_P )</td>
<td>( n^0 )</td>
<td>Implies bending stiffness ratio</td>
</tr>
<tr>
<td>Power</td>
<td>( P_M / P_P )</td>
<td>( n^2/k^3 )</td>
<td></td>
</tr>
<tr>
<td>Torque</td>
<td>( Q_M / Q_P )</td>
<td>( n^3/k^2 )</td>
<td></td>
</tr>
<tr>
<td>Material density</td>
<td>( \rho_{mM} / \rho_{mP} )</td>
<td>( n^0 )</td>
<td>Enforces Lock constraint</td>
</tr>
<tr>
<td>Bending stiffness</td>
<td>( (EJ)_M / (EJ)_P )</td>
<td>( n^4/k^2 )</td>
<td>Enforces freq. constr.</td>
</tr>
</tbody>
</table>
## Scaling Laws for Wind Turbine Models

**Example:** scaling a large wind turbine \( R_P \approx 30 \text{ m} \) to fit in a 4mx4m test section \( R_M \approx 1 \text{ m} \):

\[
n \approx 1/30 \quad k = 2
\]

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Symbol</th>
<th>Scaling coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotor speed</td>
<td>( \Omega_M/\Omega_P )</td>
<td>15</td>
</tr>
<tr>
<td>Reynolds</td>
<td>( \text{Re}_M/\text{Re}_P )</td>
<td>1/60</td>
</tr>
<tr>
<td>Froude</td>
<td>( \text{Fr}_M/\text{Fr}_P )</td>
<td>7.5</td>
</tr>
<tr>
<td>Lock</td>
<td>( \text{Lo}_M/\text{Lo}_P )</td>
<td>1</td>
</tr>
<tr>
<td>Non-dim. freq.</td>
<td>( \tilde{\omega}<em>\alpha_M/\tilde{\omega}</em>\alpha_P )</td>
<td>1</td>
</tr>
<tr>
<td>Power</td>
<td>( P_M/P_P )</td>
<td>1/57600</td>
</tr>
<tr>
<td>Torque</td>
<td>( Q_M/Q_P )</td>
<td>1/432e3</td>
</tr>
<tr>
<td>Material density</td>
<td>( \rho_m M/\rho_m P )</td>
<td>1</td>
</tr>
<tr>
<td>Bending stiffness</td>
<td>( (EJ)_M/(EJ)_P )</td>
<td>1/1296e4</td>
</tr>
</tbody>
</table>

- Aeroelastic effects unchanged (Lock and frequencies), except for possible influence of Froude (only large wind turbines)
- Moderate Reynolds mismatch, can be partially mitigated with transition strips or similar devices
- Higher required control frequencies than on full scale system, but manageable with sufficient computing power
Approach:

- Choose comparison metrics (e.g., fatigue damage index in turbulent winds, load peak values for gust response, etc.)
- Simulate response of scaled and full-scale models
- Compare responses upon back-scaling of scaled results using metrics
Testing of Scaling Laws

Example: performance comparison of two control laws on full scale and Mach–mismatched–scaled model

Turbulent 16 m/s wind
Re full scale $5.25 \times 10^6$
Re scaled $4.6 \times 10^5$

“Goodness” of one controller wrt the other is practically the same when tested on full scale and scaled model

Hence:
Scaled model is appropriate for conducting control law comparisons
References

On wind energy:

Wind Turbines – Part 1: Design Requirements, IEC 61400–1, 2005


On multibody dynamics:


References

On rotor aeroelasticity and aerodynamics:


On contact/impact:


On scaling laws:
