Energy storage and stabilization simulation of floating wind turbines

Martinus Aarmo

Magnus Nygaard Sivesind

Bachelor's thesis in Marine Technology

Bergen, Norway 2021

Høgskulen på Vestlandet

Energy storage and stabilization simulation on floating wind turbines

Martinus Aarmo

Magnus Nygaard Sivesind

Department of Mechanical- and Marine Engineering

Western Norway University of Applied Sciences

NO-5063 Bergen, Norway

IMM 2021-M32

Høgskulen på Vestlandet

Fakultet for Ingeniør- og Naturvitskap

Institutt for maskin- og marinfag

Inndalsveien 28

NO-5063 Bergen, Norge

Norsk tittel:	Energioppbevaring og Stabilitetssimulering av flytende vindmøller

Author(s), student number:	Martinus Aarmo, h581782
	Magnus Nygaard Sivesind, h578094
Study program:	Marine Technology
Date:	May 2021
Report number:	IMM YEAR-M32
Supervisor at HVL:	Thomas J. Impelluso
Assigned by:	HVL
Contact person:	Thomas J. Impelluso
Antall filer levert digitalt:	none

Preface

This bachelor thesis is written at the Department of Mechanical and Marine Engineering at Western University of Applied Scienes (WNUAS), in which the Marine Technology Program is located. The thesis is written in cooperation with Professor Thomas J. Impelluso, Professor Jan Michael Simon Bartl, and Professor David Lande-Sudall

Abstract

This project has a dual focus: to store energy extracted from floating wind turbines; and to stabilize such structures. To store the energy extracted, disks in vakuum are incorporated in the tower of the windmill. When the windmill extracts energy, this is then converted to the disks in the form of kinetic energy. The energy extracted can be extracted after the wind subsides. The stability factor comes in the form of the gyroscopic effect. The project will deploy the Moving Frame Method (MFM) to analyze the kinematics and kinetics of the system. The MFM exploits aspects of Lie Group Theory in place of vector-based dynamics. It leverages the work of Elie Cartan to model all moving bodies. Finally, it deploys a compact notation for both 3D and 2D notation. The research is an extension of past projects built on the principal of incorporating spinning disks to counter the instability in the system. The improvement comes in the form of including relevant forces acting on the system, implementation of an improved numerical integration scheme, accountment of mooring lines, and an approximation of simplified damping forces. The project defines the initial spin of the disks as a prescribed variable. Together with the prescribed variables, the Runge-Kutta method is applied for numerical integration of the equations of motion. The data is updated with an assumed correction for the rotation matrices that exploits Rodriguez' formula. Afterwards the simulation is performed by creating a port from Matlab to the Web Graphics Library and Three JS using Javascript.

Sammendrag

Dette prosjektet har ett dobbeltfokus: lagring av energi produsert av flytende vindmøller; og stabilisering av den flytende konstruksjonen via den gyroskopiske effekten. For lagring av energien innføres disker i vakum i søylen av vindmøllen. Når vindmøllen henter energi, vil denne videre konverteres til diskene i form av kinetisk energi. Målsettingen er å hente ut denne energien igjen så snart vinden har løyet. Som nevnt, vil den gyroskopiske effekten bidra med å stabilisere den flytende konstruksjonen. For prosjektet vil Moving Frame Method (MFM) benyttes for analyse av det totale systemets kinematikk og kinetikk. MFM utnytter aspekter av Lie Group Theory kontra vektor-basert dynamikk. I prosjektet benyttes tidligere arbeid utført av Elie Cartan for modelering av alle bevegelige masser. Det inkluderes i tillegg kompakt notasjon for både 2D og 3D.

Undersøkelsen er en videreføring av tidligere oppgaver utført ved HVL. Tidligere prosjekter er basert på bruk av spinnende disker for motvirkning av ustabilitet. Dette prosjektet vil bistå med samtlige essensielle faktorer for økning av prosjektets relevanse. Blandt disse er inklusjon av relevante krefter som påvirker systemet, implementering av forbedret metode for numerisk integrasjon, inklusjon av forankringsliner og tilnærming av forenklede viskøse dempekrefter. Prosjektet definerer initielle verdier for spin av diskene som en foreskrevet verdi. Runge-Kutta metoden benyttes for numerisk integrasjon av bevegelsesligningene. Dataene oppdateres med en antatt korreksjon for rotasjonsmatrisene som utnytter Rodriguez’ formel. I etterkant vil en simulering gjøres ved å danne en port fra MatLab til Web Graphics Library og Three JS ved bruk av Javascript.

Table of contents

Preface ... V

Abstract ... VII

Sammendrag ... IX

Nomenclature ... XIV

1. Introduction ... 1

2. Method ... 2

2.1 The moving frame method ... 2

2.2 general Principles of the MFM ... 2

2.2.1 Kinematics using SO(3) ... 2

2.2.2 Kinematics using SE(3) ... 4

2.3 Kinematics of the floating wind turbine ... 6

2.3.1 First Frame – The Turbine Tower ... 6

2.3.2 The second frame – The Turbine Rotor ... 8

2.3.3 The Third Frame – The Correcting Rotating Disk ... 11

2.4 Kinetics of the floating wind turbine ... 12

2.4.1 Constraint on the variation ... 13

2.4.2 Principle of Virtual Work ... 14

2.5 Equation of motion and numerical integration ... 14

2.5.1 Formation of the numerical integration equation ... 15

2.5.2 Updating the rotation matrix ... 16

2.5.3 Input Data ... 17

3. Results ... 18

3.1 Results and graphs ... 18

3.2 WebGL ... 20

4. Discussion ... 21

5. Conclusion ... 22

6. References ... 23

List of figures ... 24

Attachment 1 (Matlab code) ... 25

Attachment 2 (IMECE report) ... 41

Nomenclature

$[B]$:	Transformation matrix to generalized coordinates
$[C]$:	Transformation matrix for prescribed rates
$[D]$:	Combined angular velocity matrix
$\mathbf{e}$:	Unit basis vector
E:	Frame connection matrix
$\{F\}$:	Force and moment list
$\{F^*\}$:	Generalized force and moment list
g:	Gravitational acceleration
I_3:	3×3 Identity matrix
J:	3×3 Mass moment of inertia matrix
K:	Kinetic energy
m:	Mass
$[M]$:	Mass matrix
$[M^*]$:	Reduced mass matrix
[N]:	Non-linear velocity matrix
$[N^*]$:	Reduced non-linear velocity matrix
$q(t)$:	Generalized coordinates
$\{\dot{q}(t)\}$:	Generalized velocity
$\{\ddot{q}(t)\}$:	Generalized acceleration
$\{\dot{r}(t)\}$:	Generalized prescribed velocity
R:	3×3 Rotation matrix

$[T^*]$: Reduced velocity matrix for prescribed rates

ω: Angular velocity components

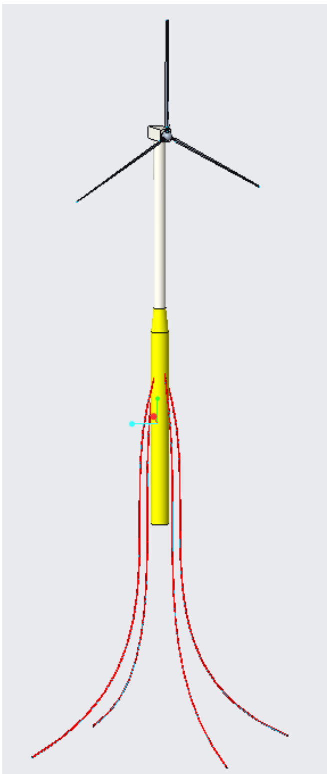
$\overleftrightarrow{\omega}$: Skew-symmetric angular velocity matrix

1. Introduction

Offshore wind energy has become highly relevant for the past years. According to recent reports, the industry is expected to grow even further in the years to come [1]. Research from such reports states that offshore wind has the capability of producing more than 420 000 TWh. This amount is more than 18 times the electricity demand of today [2].

Norway endeavors to be one of the most sustainable countries in the world. With a long coastline, Norway has a great opportunity to develop and invest in offshore solutions. While land-based wind turbines have been critiqued due to their impact on nature when being constructed and operated, offshore installations do not. They do however have an impact on marine life offshore. Still, despite this, there is a strong wind resource offshore and more space available. However, offshore wind turbines are a subject to harsher elements such as waves and stronger winds compared to onshore, and all this must be considered.

The water depth off the Norwegian cost is generally too deep for bottom fixed foundations. Therefore, floating wind turbines with a mooring system, appears to be a preferable solution. The challenge with this concept, however, is based on the stability of the floating wind turbines, wind- and wave induced forces. In this paper, the concept behind a self-stabilizing wind turbine is explored. This mechanism is based on the principle behind a gyroscopic mechanism. Finally, the extracted energy to drive the correcting rotors, is converted into kinetic energy in the tower of the turbine. Essentially, the disks will serendipitously store energy generating a stabilizing effect on the turbine itself. To carry out this work the solution adopts the moving frame method (MFM) in dynamics.



Figur 1: Model of monopile windturbine created in Creo.

2. Method

2.1 The moving frame method

The reader may find an introduction to an undergraduate and graduate Moving Frame Method, along with pedagogical assessement in Impelluso [3]. The following section summarizes salient elements of the MFM.

2.2 general Principles of the MFM

2.2.1 Kinematics using SO(3)

At the center of mass of each body (α) a time-dependent moving frame is placed:

$$\mathbf{e}^{(\alpha)}(t) = \left(\mathbf{e}_1^{(\alpha)}(t)\ \ \mathbf{e}_2^{(\alpha)}(t)\ \ \mathbf{e}_3^{(\alpha)}(t)\right) \tag{1}$$

In the previous, **e** is a unit vector, and the subscript denotes the Cartersian coordinate direction. Set $t = 0$ to define and deposit an inertial frame from a moving frame:

$$\mathbf{e}^I = \left(\mathbf{e}_1^I\ \ \mathbf{e}_2^I\ \ \mathbf{e}_3^I\right) = \left(\mathbf{e}_1^{(\alpha)}(0)\ \ \mathbf{e}_2^{(\alpha)}(0)\ \ \mathbf{e}_3^{(\alpha)}(0)\right) \tag{1}$$

Define the absolute position vector $\mathbf{r}_C^{(\alpha)}(t)$ of a frame as a translation from the inertial frame $\mathbf{e}^I$ using a compact notation:

$$\mathbf{r}_C^{(\alpha)}(t) = \mathbf{e}^I x_C^{(\alpha)}(t) = \mathbf{e}^I \left(x_{1C}^{(\alpha)}(t) x_{2C}^{(\alpha)}(t) x_{3C}^{(\alpha)}(t)\right)^T \tag{2}$$

In (3) $x_C^{(\alpha)}(t)$ is used to represent, in vertical form (transpose, above), the *absolute* coordinates of the distance to the center of mass of a body (subscript C), expressed in the inertial frame.

Assert the *relative* position vector of a frame ($\alpha + 1$) from another frame (α) by $\mathbf{s}_C{}^{(\alpha+1/\alpha)}(t)$. Express this relative translation in the α-frame:

$$\mathbf{s}_C{}^{(\alpha+1/\alpha)}(t) = \mathbf{e}^{(\alpha)}(t) s_C{}^{(\alpha+1/\alpha)}(t) \tag{3}$$

By adding the absolute position vector of the α-frame $\mathbf{r}_C^{(\alpha)}(t)$ and the relative position vector $\mathbf{s}_C{}^{(\alpha+1/\alpha)}(t)$, one obtains the absolute position vector of the ($\alpha + 1$) frame: frame, $\mathbf{r}_C^{(\alpha+1)}(t)$:

$$\mathbf{r}_C^{(\alpha+1)}(t) = \mathbf{r}_C^{(\alpha)}(t) + \mathbf{e}^{(\alpha)}(t) s_C^{(\alpha+1/\alpha)}(t) \tag{4}$$

Now, turn the attention to frame orientations. A rotation matrix is used, a member of the Special Orthogonal Group $R \in SO(3)$, to relate the orientation of a moving frame to an inertial frame:

$$\mathbf{e}^{(\alpha)}(t) = \mathbf{e}^{I} R^{(\alpha)}(t) \tag{5}$$

The relative rotation of a frame ($\alpha + 1$) from another frame (α) can be written as:

$$\mathbf{e}^{(\alpha+1)}(t) = \mathbf{e}^{(\alpha)}(t) R^{(\alpha+1/\alpha)}(t) \tag{6}$$

The orientation of body ($\alpha + 1$) can be expressed in the inertial frame by inserting equation (5) into (6) and exploiting the closure property of the SO(3) Group:

$$\mathbf{e}^{(\alpha+1)}(t) = \mathbf{e}^{I} R^{(\alpha)}(t) R^{(\alpha+1/\alpha)}(t) = \mathbf{e}^{I} R^{(\alpha+1)}(t) \tag{7}$$

As a property of SO(3), the inverse of a rotation matrix is the transpose:

$$\left(R^{(\alpha)}(t)\right)^{-1} = \left(R^{(\alpha)}(t)\right)^{T} \tag{8}$$

The time rate of frame rotation is (with time depdendent R):

$$\dot{\mathbf{e}}^{(\alpha)}(t) = \mathbf{e}^{I} \dot{R}^{(\alpha)}(t) \tag{9}$$

We use (9) in (6) to formulate the inertial frame in terms of the moving frame and then substitute the result into (10) to obtain:

$$\dot{\mathbf{e}}^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t) \left(R^{(\alpha)}(t)\right)^{T} \dot{R}^{(\alpha)}(t) \tag{10}$$

The time rate of frame rotation is now expressed in its own frame, satisfying the thoughts of Elie Cartan [4]. It can be shown (Lie Group Theory), that the matrix products in (11) produce a skew symmetric matrix. Thus, the skew-symmetric angular velocity matrix is defined. Note that this element is a member of the associated algebra, so(3):

$$\overleftrightarrow{\omega^{(\alpha)}(t)} = \left(R^{(\alpha)}(t)\right)^{T} \dot{R}^{(\alpha)}(t) = \begin{bmatrix} 0 & -\omega_3^{(\alpha)}(t) & \omega_2^{(\alpha)}(t) \\ \omega_3^{(\alpha)}(t) & 0 & -\omega_1^{(\alpha)}(t) \\ -\omega_2^{(\alpha)}(t) & \omega_1^{(\alpha)}(t) & 0 \end{bmatrix} \tag{11}$$

Equation (10) is rewritten as:

$$\dot{\mathbf{e}}^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t)\overleftrightarrow{\omega^{(\alpha)}(t)} \tag{12}$$

The skew-symmetric angular velocity matrix is isomorphic to the same frame to the angular velocity vector of that frame:

$$\boldsymbol{\omega}^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t)\begin{pmatrix} \omega_1^{(\alpha)}(t) \\ \omega_2^{(\alpha)}(t) \\ \omega_3^{(\alpha)}(t) \end{pmatrix} \tag{13}$$

In (14) above, unlike in planar dynamics, it can be observed that the basis frame is time dependent.

2.2.2 Kinematics using SE(3)

Before beginning, it's noted that the analysis of specifics of this windmill turbine could be conducted with the aforementioned work, alone—SO(3). However, in this work the approach chosen is SE(3)[5], as it is more readily extensible and is being used in the expansion of this work, currently underway. Here, an overview is presented.

One combines the rotational and translational data of a frame (α), in one structure. The 4×4 absolute *frame connection matrix* (a member of the Special Euclidean Group) is defined, E ∈ SE(3):

$$E^{(\alpha)}(t) = \begin{bmatrix} R^{(\alpha)}(t) & x_C^{(\alpha)}(t) \\ 0_3^T & 1 \end{bmatrix} \tag{14}$$

The *inertial frame connection* is defined. This consists of the frame and its position, represented as:

$$\begin{pmatrix} \mathbf{e}^I & \mathbf{0} \end{pmatrix} \tag{15}$$

Similarly, the *moving frame connection* is represented as:

$$\begin{pmatrix} \mathbf{e}^{(\alpha)}(t) & \mathbf{r}_C^{(\alpha)}(t) \end{pmatrix} \tag{16}$$

The structure in (17) contains both the frame and its position from the inertial frame. The inertial frame connection (16) and the moving frame connection (17) are related utilizing the *absolute* frame connection matrix (14):

$$\left(\mathbf{e}^{(\alpha)}(t)\ \ \mathbf{r}_C^{(\alpha)}(t)\right)=\left(\mathbf{e}^I\ \ \mathbf{0}\right)E^{(\alpha)}(t) \tag{17}$$

Moving to the relative forms, the *relative* frame connection matrix is defined as:

$$E^{(\alpha+1/\alpha)}(t)=\begin{bmatrix} R^{(\alpha+1/\alpha)}(t) & s_C^{(\alpha+1/\alpha)}(t) \\ 0_3^T & 1 \end{bmatrix} \tag{18}$$

Equation (18) is used to express the relative relationship between two moving frames, ($\alpha+1$) and (α):

$$\left(\mathbf{e}^{(\alpha+1)}(t)\ \ \mathbf{r}_C^{(\alpha+1)}(t)\right)=\left(\mathbf{e}^{(\alpha)}(t)\ \ \mathbf{r}_C^{(\alpha)}(t)\right)E^{(\alpha+1/\alpha)}(t) \tag{19}$$

Equation (20), with its defining element (19), recapitulates equations (5) and (7).

The absolute frame connection matrix of body ($\alpha+1$) can be found as the product of the absolute frame connection matrix of body (α) and the relative frame connection matrix that relates them (as a result of the closure property of the SE(3) group):

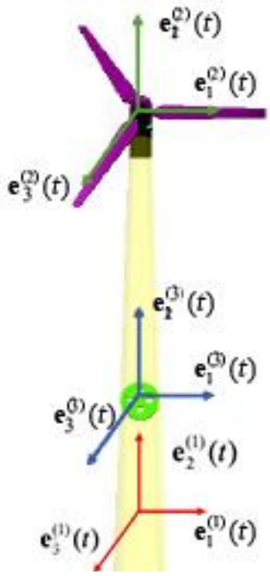
$$E^{(\alpha+1)}(t)=E^{(\alpha)}(t)E^{(\alpha+1/\alpha)}(t) \tag{20}$$

Instead of discussing the details behind the theory, one advances to implementing SE(3) theory in tutorial style, through an example for edification using a wind turbine.

2.3 Kinematics of the floating wind turbine

2.3.1 First Frame – The Turbine Tower

The first moving frame $\mathbf{e}^{(1)}(t)$ is placed at the center of mass of the tower of the turbine. At $t = 0$ one deposits an inertial frame from the first frame. In regard to the figure, the inertial frame isn't displayed, but is located at the exact location of the first moving frame.



Figur 2: Schematic of the Wind Turbine with three moving frames

$$\mathbf{e}^I \equiv \mathbf{e}^{(1)}(0) \tag{21}$$

The orientation of the first moving frame is expressed as follows:

$$\mathbf{e}^{(1)}(t) = \mathbf{e}^I R^{(1)}(t) \tag{23}$$

The elements of $R^{(1)}(t)$ will contain information about the pitch, yaw and roll of the turbine from an inertial configuration.

The displacement of the first moving frame is stated as:

$$\mathbf{r}_c^{(1)}(t) = \mathbf{e}^I x_c^{(1)}(t) \tag{24}$$

One may immediately apply Equation (15) and (18) in the form of a frame connection relationship. Then, continuing, the time rate of the frame connection is calculated of the same form (25b).

$$\begin{pmatrix}\mathbf{e}^I & \mathbf{0}\end{pmatrix} = \begin{pmatrix}\mathbf{e}^{(1)}(t) & \mathbf{r}_C^{(1)}(t)\end{pmatrix}\left(E^{(1)}(t)\right)^{-1} \tag{25a}$$

$$\left(\dot{\mathbf{e}}^{(1)}(t)\ \ \dot{\mathbf{r}}_C^{(1)}(t)\right) = \left(\mathbf{e}^I\ \ \mathbf{0}\right)\dot{E}^{(1)}(t) \tag{25b}$$

The time rate of the frame connection matrix $\dot{E}^{(1)}(t)$ is found by taking the time derivative of each data structure:

$$\dot{E}^{(1)}(t) = \begin{bmatrix} \dot{R}^{(1)}(t) & \dot{x}_c^{(1)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{26}$$

The inverse of the frame connection matrix, is expressed as (due to $E \in$ SE(3)):

$$\left(E^{(1)}(t)\right)^{-1} = \begin{bmatrix} \left(R^{(1)}(t)\right)^T & -\left(R^{(1)}(t)\right)^T x_C^{(1)}(t) \\ 0_3^T & 1 \end{bmatrix} \tag{27}$$

Equation (25) is used in (21) to formulate the inertial frame connection in terms of the moving frame connection to obtain:

$$\left(\dot{\mathbf{e}}^{(1)}(t)\quad \dot{\mathbf{r}}_C^{(1)}(t)\right) = \left(\mathbf{e}^{(1)}(t)\quad \mathbf{r}_C^{(1)}(t)\right)\left(E^{(1)}(t)\right)^{-1}\dot{E}^{(1)}(t) \tag{28}$$

The absolute time rate of frame connection matrix is defined for the first body, $\Omega^{(1)}$ as the product of $\left(E^{(1)}(t)\right)^{-1}$ and $\dot{E}^{(1)}(t)$. Note that $\Omega \in$ se(3) (the algebra associated with the SE(3) group):

$$\Omega^{(1)} = \left(E^{(1)}(t)\right)^{-1}\dot{E}^{(1)}(t) \tag{29}$$

As a result, in keeping with the view of Cartan (expressing the change of structure in terms of the same structure, as in Eqn. 13) equation (28) is rewritten as:

$$\left(\dot{\mathbf{e}}^{(1)}(t)\ \ \dot{\mathbf{r}}_C^{(1)}(t)\right) = \left(\mathbf{e}^{(1)}(t)\ \ \mathbf{r}_C^{(1)}(t)\right)\Omega^{(1)}(t) \tag{30}$$

$\Omega^{(1)}$ multiplied out in matrix from:

$$\Omega^{(1)} = \begin{bmatrix} \left(R^{(1)}(t)\right)^T \dot{R}^{(1)}(t) & \left(R^{(1)}(t)\right)^T \dot{x}_C^{(1)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{31}$$

By comparing the expression to (12), (31) is rewritten as:

$$\Omega^{(1)} = \begin{bmatrix} \overleftrightarrow{\omega^{(1)}(t)} & \left(R^{(1)}(t)\right)^T \dot{x}_C^{(1)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{32}$$

By expanding, parts of the system are extracted and recapitulated as:

$$\dot{\mathbf{e}}^{(1)}(t) = \mathbf{e}^{(1)}(t)\,\overleftrightarrow{\omega^{(1)}(t)} \tag{33}$$

The second equation extracted from equation (30) is:

$$\dot{\mathbf{r}}_C^{(1)}(t) = \mathbf{e}^{(1)}(t)\left(R^{(1)}(t)\right)^T \dot{x}_C^{(1)}(t) \tag{34}$$

Thus, one states the translational velocity as:

$$\dot{\mathbf{r}}_C^{(1)}(t) = \mathbf{e}^I \dot{x}_C^{(1)}(t) \tag{35}$$

This marks the point where the first body is properly assessed, and the equations of the first body are listed.

$$\dot{\mathbf{e}}^{(1)}(t) = \mathbf{e}^{(1)}(t)\overleftrightarrow{\omega^{(1)}(t)} \tag{36}$$

$$\dot{\mathbf{r}}_c^{(1)}(t) = \mathbf{e}^I \dot{x}_c^{(1)}(t) \tag{37}$$

2.3.2 The second frame – The Turbine Rotor

The second body in this analysis is the turbine rotor of the wind turbine, which branches off the first body. The frame is placed at the centre of mass of the blades. The relationship of the second frame connection from the first frame (turbine) connection is stated as:

$$(\mathbf{e}^{(2)}(t)\;\; \mathbf{r}_c^{(2)}(t)) \equiv \left(\mathbf{e}^{(1)}(t)\;\; \mathbf{r}_c^{(1)}(t)\right) E^{(2/1)}(t) \tag{38}$$

Or just the frame connection matrix as:

$$E^{(2/1)}(t) = \begin{bmatrix} R^{(2/1)}(t) & s_c^{(2/1)} \\ 0 & 1 \end{bmatrix} \tag{39}$$

This frame connection matrix could be expressed as the product of two, in which displacement and rotation are separated—this, only for the sake of edification. Thus, in the first matrix, below, the two values locate the center of mass of the windmill blades, from the center of mass of the turbine body. While it's prefered to reserve formal numbers to the solution process, the rule is broken here, for the sake of demonstration.

Progress in the 2-direction, $d^{(1)}$, and then out (along the nacelle) in the 3-direction, $h^{(1)}$. The boxed column in the first matrix, below demonstrates this translation.

$$E^{(2/1)}(t) = \begin{bmatrix} 1 & 0 & 0 & d^{(1)} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h^{(1)} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) & 0 & 0 \\ \sin(\theta(t)) & \cos(\theta(t)) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{40}$$

In the second matrix, above, set aside the conforming "1" in the lower right corner and the column above it. The remaining 3x3 marix represents a standard rotation about the local body's (the turbine's) 3-axis, for the rotation of the frame by a time dependent angle: θ.

With (40) and (26), the absolute frame connection matrix from the inertial frame is stated as:

$$(\mathbf{e}^{(2)}(t)\ \ \mathbf{r}_c^{(2)}(t)) \equiv \left(\mathbf{e}^I\ \ \mathbf{0}\right) E^{(2)}(t) \tag{41}$$

Where:

$$E^{(2)}(t) = \begin{bmatrix} R^{(1)}(t) & x_c^{(1)}(t) \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} R^{(2/1)}(t) & s_c^{(2/1)} \\ 0 & 1 \end{bmatrix} \tag{42}$$

Equation (42) is expanded to matrix form as:

$$E^{(2)}(t) = \begin{bmatrix} \left(R^{(1)}(t)R^{(2/1)}(t)\right) & \left(R^{(1)}(t)s_c^{(2/1)} + x_c^{(1)}(t)\right) \\ 0 & 1 \end{bmatrix} \tag{43}$$

Continuing, the inverse of the frame connection matrix is formed:

$$\left(E^{(2)}(t)\right)^{-1} = \begin{bmatrix} \left(R^{(2/1)}(t)\right)^T\left(R^{(1)}(t)\right)^T & -\left(\left(R^{(2/1)}(t)\right)^T s_c^{(B/1)} + \left(R^{(2)}(t)\right)^T x_c^{(1)}(t)\right) \\ 0 & 1 \end{bmatrix} \tag{44}$$

In the same manner as previously, the time rate of the frame connection matrix is developed by deriving each block of the matrix. Considering the location of the blades will not translate according to the main tower, one may cancel out the derived product of the position vector $S_c^{(2/1)}$:

$$\dot{E}^{(2)}(t) = \begin{bmatrix} \dot{R}^{(1)}(t)R^{(2/1)}(t) + R^{(1)}(t)\dot{R}^{(2/1)}(t) & \dot{R}^{(1)}(t)s_c^{(2/1)} + \dot{x}_c^{(1)}(t) \\ 0 & 0 \end{bmatrix} \tag{45}$$

The general form of the omega matrix is stated as:

$$\Omega^{(2)}(t) \equiv \left(E^{(2)}(t)\right)^{-1}\dot{E}^{(2)}(t) \tag{46}$$

In expanded notation:

$$\Omega^{(2)}(t) \equiv \begin{bmatrix} \left(\left(R^{(2/1)}(t)\right)^T \overleftrightarrow{\omega^{(1)}(t)} R^{(2/1)}(t) + \overleftrightarrow{\omega^{(2/1)}(t)}\right) & \left(R^{(2/1)}(t)\right)^T \left(\overleftrightarrow{\omega^{(1)}(t)} s_c^{(2/1)} + \left(R^{(1)}(t)\right)^T \dot{x}_c^{(1)}(t)\right) \\ 0 & 0 \end{bmatrix} \tag{47}$$

By comparing with the general definition of an omega matrix, one can extract the definition of $\overleftrightarrow{\omega^{(2)}(t)}$. The omega term, is extracted and formulated as (using aspects of the Lie Algebra):

$$\omega^{(2)}(t) = \left(R^{(2/1)}(t)\right)^T \omega^{(1)}(t) + \omega^{(2/1)}(t) \tag{48}$$

Continuing, having alluded to the blade-axis of rotation, (48) can be reformulated using:

- $e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$
- The spin rate being perscribed with a rate: $\dot{\zeta}$

Now, the omega vector is expressed as:

$$\omega^{(2)}(t) = \left(R^{(2/1)}(t)\right)^T \omega^{(1)}(t) + \dot{\zeta} e_i \tag{49}$$

For the translation expression, the term is modified by bringing the rotational matrix to the right-hand side. Hence, the expression for the translation is modified, accounting for the rotation and translation of the turbine tower:

$$\dot{x}_c^{(2)}(t) = R^{(1)}\left(\overleftrightarrow{\omega^{(1)}(t)} s_c^{(2/1)} + \left(R^{(1)}(t)\right)^T \dot{x}_c^{(1)}(t)\right) \tag{50}$$

Like the expression for omega, one will eventually require the omegas and the translation vectors at the end of each term (when one is to relate the generalized and Cartesian coordinates). The term is modified by switching the omega with a negative, and then transposing to negate the negative:

$$\dot{x}_c^{(2)}(t) = \left(R^{(1)} \left(\overleftrightarrow{s_c^{(2/1)}}\right)^T \omega^{(1)}(t) + \dot{x}_c^{(1)}(t) \right) \tag{51}$$

This concludes the extraction of equations for frame/body 2.

2.3.3 The Third Frame – The Correcting Rotating Disk

The location of the third body is fortuitous: it does not extend off the second frame, but, like the blades, off the turbine tower. Similarly compared to the 2nd moving body, the correcting rotor is also rotating. Since the equations describing the first moving body are developed in general terms (not specifying axis of rotation before the solution process), one can change the subscripts of the expressions for the second frame, and obtain the expressions for the third body, in general terms, with the rotor spin signified by $\dot{\psi}$. In future work, this is to be expanded to allow for internal mechanisms.

Hence, the equations for the translation- and omega-vectors are stated for the third body:

$$\dot{x}_c^{(3)}(t) = \dot{x}_c^{(1)}(t) + R^{(1)} \left(\overleftrightarrow{s_c^{(3/1)}}\right)^T \omega^{(1)}(t) \tag{52}$$

And:

$$\omega^{(3)}(t) = \left(R^{(3/1)}(t)\right)^T \omega^{(1)}(t) + e_3 \dot{\psi} \tag{53}$$

At this point, the kinematic expressions are fully developed, and one can turn to kinetics. First, however, the prescribed rotations for the wind turbine and correcting rotor are separated from the two generalized variables. With this compact matrix form, below, the expressions for all the relevant Cartesian variables necessary to conduct the minimization required of the Principle of Virtual Work are defined.

$$\begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \\ \dot{x}^{(2)}(t) \\ \omega^{(2)}(t) \\ \dot{x}^{(3)}(t) \\ \omega^{(3)}(t) \end{pmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \\ I & R^{(1)}\left(\overleftrightarrow{s_c^{(2/1)}}\right)^T \\ 0 & \left(R^{(2/1)}(t)\right)^T \\ I & R^{(1)}\left(\overleftrightarrow{s_c^{(3/1)}}\right)^T \\ 0 & \left(R^{(3/1)}(t)\right)^T \end{bmatrix} \begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ e_i & 0 \\ 0 & 0 \\ 0 & e_j \end{bmatrix} \begin{pmatrix} \dot{\zeta} \\ \dot{\psi} \end{pmatrix} \tag{54}$$

Below, (54) is stated in the form of block matrix forms, where the definitions of C and B are easily comparable to (54) above.

$$\left\{ \dot{X}(t) \right\} = \begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \\ \dot{x}^{(2)}(t) \\ \omega^{(2)}(t) \\ \dot{x}^{(3)}(t) \\ \omega^{(3)}(t) \end{pmatrix} = B \begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \end{pmatrix} + C \begin{pmatrix} \dot{\zeta} \\ \dot{\psi} \end{pmatrix} \tag{55}$$

Finally, it’s recast in the simplest terms with the definitions as:

$$\dot{X}(t) = B\dot{q}(t) + C\dot{r} \tag{56}$$

2.4 Kinetics of the floating wind turbine

Kinetic energy K includes translation energy and rotational energy regarding each center of mass.

$$K^{(\alpha)} = \frac{1}{2}\left\{ \left(\dot{x}_C^{(\alpha)}\right)^T m^{(\alpha)} \dot{x}_C^{(\alpha)} + \left(\omega^{(\alpha)}\right)^T J_C^{(\alpha)} \omega^{(\alpha)} \right\} \tag{57}$$

This is used in Hamilton’s Principle, reformed as the Principle of Virtual work with all work (conservative and non-conservative) on the right side.

$$\delta \int_{t_0}^{t_1} K^{(\alpha)}(t)\, dt = -\delta \int_{t_0}^{t_1} W^{(\alpha)}(t)\, dt \tag{58}$$

2.4.1 Constraint on the variation

The variation is taken as follows:

$$\delta \mathbf{r}_C^{(\alpha)} = \mathbf{e}^I \delta x_C^{(\alpha)} \tag{59}$$

The commutativity of mixed partials readily holds for translational velocity and one obtains:

$$\delta \dot{x}_C^{(\alpha)}(t) = \left(\frac{d}{dt} \delta x_C^{(\alpha)}(t) \right) \tag{60}$$

However, the variation of the angular velocity is restricted in 3D space. This was found by Murakami [10] and independently by Holm [14]. First, it's essential to define the following term:

$$\overleftrightarrow{\delta \pi^{(\alpha)}(t)} = \left(R^{(\alpha)}(t) \right)^T \delta R^{(\alpha)}(t) \tag{61}$$

Equation (61) does not exist in its unvaried form. It defines the virtual frame-rotation vector $\delta\boldsymbol{\pi}^{(\alpha)}$, in the same way as the angular velocity matrix defined the angular velocity:

$$\delta \boldsymbol{\pi}^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t) \delta \pi^{(\alpha)}(t) \tag{62}$$

By ensuring the commutativity of mixed partials (time and variation with regard to the directional derivative of the variation parameter), a restriction is formed. It is found that the variation of the angular velocity depends on the virtual frame rotation, referred to as restricted variation of virtual angular velocity:

$$\delta \omega^{(\alpha)} = \delta \dot{\pi} + \overleftrightarrow{\omega^{(\alpha)}} \delta \pi^{(\alpha)} \tag{63}$$

With the above, the virtual generalized displacments are established.

$$\{\delta \tilde{X}(t)\} = \begin{pmatrix} \delta x_C^{(1)}(t) \\ \delta \pi^{(1)}(t) \\ \delta x_C^{(2)}(t) \\ \delta \pi^{(2)}(t) \\ \delta x_C^{(3)}(t) \\ \delta \pi^{(3)}(t) \end{pmatrix} \tag{63a}$$

2.4.2 Principle of Virtual Work

Continuing, the coefficient matrix $[B(t)]$ which relates the generalized velocities in Eq.(54), also relates the generalized displacements $\{\delta \tilde{X}(t)\}$ and the essential virtual displacements $\{\delta q(t)\}$:

$$\{\delta\tilde{X}(t)\} = [B(t)]\{\delta q(t)\} \tag{64}$$

Proceeding with the virtual work done by the physical forces, where moments and $\overleftrightarrow{\delta\pi^{(\alpha)}(t)}$ are a conjugate pair:

$$\delta W = \{\delta \tilde{X}(t)\}^T \{F(t)\} \tag{65}$$

Before continuing, it is first stated one should use these terms in the Principle of Virtual Work: Moment vs. virtual rotation represent a natural pair. They are conjugate to the moment expressed with the body frame. Moment vs. virtual rotation is a natural pair: Hamilton's principle, which yields Euler's equation. Wittenburg [7] postulated the principle of virtual power to use the weighted form of Euler's equation by the virtual angular velocity.

Continuing, for a simple, first pass analysis, one allows for the following external forces: Wind Force on the blades and fixation force to hold the turbine in place (with cables attached at the turbine center of mass—obviating the need to apply cable moments). Buoyancy and gravity are ignored (all of which are unjustified and excessive—however, as stated this first pass paper is for edification). Finally, after multiplying by the B matrix (Eqn. 4 and 5), the following expressions are obtained, in compact form, for the generalized forces.

3 row Forces from cable fixation $\quad CableF^{(1)} \quad$ (66a)

3 row Windforces $\quad WindF^{(2)} \quad$ (66b)

$$F^* = \begin{bmatrix} CableF^{(1)} + WindF^{(2)} \\ \left(\overleftrightarrow{s_c^{(2/1)}}\left(R^{(1)}\right)^T\right) WindF^{(2)} \end{bmatrix} \tag{66c}$$

2.5 Equation of motion and numerical integration

By making all the substitutions and carrying out the calculus of variations, one obtains the following results:

2.5.1 Formation of the numerical integration equation

Related work for cranes on ships [9] and ROV motion [9] discusses all terms below, in detail. For now, however, it is alluded to the fact that the M matrix consists of alternating mass and moment of inertial in block diagonal form. Below, the mass entry for each body is stated.

$$[M]^{(i)} = \begin{bmatrix} m^{(i)} \underset{3x3}{I} & \underset{3x3}{0} \\ \underset{3x3}{0} & \underset{3x3}{J^{(i)}} \end{bmatrix} \tag{67a}$$

The D matrix allows for the modification of the variation of the angular velocities.

$$[D]^{(i)} = \begin{bmatrix} \underset{3x3}{0} & \underset{3x3}{0} \\ \underset{3x3}{0} & \underset{3x3}{\overleftrightarrow{\omega}^{(i)}} \end{bmatrix} \tag{67a}$$

With the previous set up, one can construct the minimization (Principle of Virtual Work) and extract the equations of motion in terms of generalized variables. The following equations present definitions, followed by the equation of motion:

$$\left[M^*(t)\right] \equiv \left[B(t)\right]^T \left[M\right]\left[B(t)\right] \tag{68a}$$

$$\left[N^*(t)\right] \equiv \left[B(t)\right]^T \left(\left[M(t)\right]\left[\dot{B}(t)\right] + \left[D(t)\right]\left[M\right]\left[B(t)\right]\right) \tag{68b}$$

$$\left\{F^*(t)\right\} = \left[B(t)\right]^T \left\{F(t)\right\} \tag{68c}$$

$$\left[T^*\right] = \left(B^T M\dot{C} + B^T DMC\right) \tag{68d}$$

$$\left[M^*(t)\right]\left\{\ddot{q}(t)\right\} + \left[N^*(t)\right]\left\{\dot{q}(t)\right\} = \left\{F^*(t)\right\} - \left[T^*\right]\left\{\dot{r}\right\} \tag{68e}$$

Numerical Integration

For the solution, the Runge Kutta method is utilized. However, it is known that that method updates both spatially and temporally. The method is reformulated in terms of:

$$\dot{y}(t) = \left(M^*(t)\right)^{-1}\left(F^*(t) - N^*(t)y(t) \ - T^*(t)\dot{r}\right) \tag{69}$$

Thus, in terms of the RK4 constants

$$k_1 = \left(M^*_{t_n}\right)^{-1}\left(F^*_{t_n} - N^*_{t_n} y_{t_n} - T^*_{t_n}\dot{r}\right) \tag{70a}$$

$$k_2 = \left(M^*_{t_n+\frac{dt}{2}}\right)^{-1}\left(F^*_{t_n+\frac{dt}{2}} - N^*_{t_n+\frac{dt}{2}}\left(y_{t_n} + \frac{k_1}{2}dt\right) - T^*_{t_n+\frac{dt}{2}}\dot{r}\right) \tag{70b}$$

$$k_3 = \left(M^*_{t_n+\frac{dt}{2}}\right)^{-1}\left(F^*_{t_n+\frac{dt}{2}} - N^*_{t_n+\frac{dt}{2}}\left(y_{t_n} + \frac{k_2}{2}dt\right) - T^*_{t_n+\frac{dt}{2}}\dot{r}\right) \tag{70c}$$

$$k_4 = \left(M^*_{t_n+dt}\right)^{-1}\left(F^*_{t_n+dt} - N^*_{t_n+dt}\left(y_{t_n} + k_3 dt\right) - T^*_{t_n+dt}\dot{r}\right) \tag{70d}$$

$$y_{n+1} = y_n + \frac{dt}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{70e}$$

However, in between each spatial update, one must first update the rotation matrix for the turbine, which is presented next.

2.5.2 Updating the rotation matrix

The rotation matrices for the two arms are standard, due to the derivation from revolute joints. However, one must know the rotation matrix of the turbine for several reasons. First, it is required in the updating of the B matrix. Second, it is required to apply the hydrodynamic forces, which will be included in a later iteration. Finally, it is needed for the visualization.

The rotation matrix of the wind turbine must be reconstructed from the angular velocity. The rotation matrix $R^{(1)}(t)$ is computed by solving the following equation:

$$\dot{R}^{(1)}(t) = R^{(1)}\overleftrightarrow{\omega^{(1)}(t)} \tag{71}$$

Assuming that $\overleftrightarrow{\omega^{(1)}(t)}$ is constant and is designated as $\overleftrightarrow{\omega}_0$. Then, with initial value $R(0)$, the solution is:

$$R^{(1)}(t) = R(0)\exp(t\overleftrightarrow{\omega_0}) \tag{72}$$

There does exist a known analytical, closed form solution to Eq. (71), but only for cases in which $\overleftrightarrow{\omega}_0$ is constant. It derives from the Cayley Hamilton Theorem and is known as the Rodriguez' rotation formula to obtain a series expansion of the exponential of a matrix.

$$R(t+\Delta t)=R(t)\left(\begin{array}{l} I_d+\dfrac{\overleftrightarrow{\omega_0(t+\Delta t)}}{\left\|\omega(t+\Delta t)\right\|}\sin(t\left\|\omega_0(t+\Delta t)\right\|) \\ \left(\dfrac{\overleftrightarrow{\omega_0(t+\Delta t)}}{\left\|\omega(t+\Delta t)\right\|}\right)^2(1-\cos(t\left\|\omega_0(t+\Delta t)\right\|)) \end{array}\right) \tag{73}$$

The difficulty is that one doesn't have a constant angular velocity matrix. However, it's possible to approximate its constancy in each time step of the numerical integration. In principle, one averages this over two-time steps using a central difference approximation. However, for ease of first pass coding, this rule is violated by assuming a constant value at the start of each time step:

$$\omega(t+\Delta t/2)\equiv(\omega(t)+\omega(t+\Delta t/2))/2 \tag{74}$$

To compile the system of variables, one gathers the cartesian and generalized coordinates:

2.5.3 Input Data

Trubine mass	$7.5x10^6 kg$
Hub height	$200m$
Tower Radius	$10m$
Rotor mass	$53,000Kg$
Rotor radius	$60m$
Rotor location from turbine CM	$90m$
Correcting rotor mass	$530,000Kg$
Correcting rotor radius	$10m$
Correcting rotor location from CM	0.0
Tip Speed Ratio	0.7
Wind density	$1.225kg/m^3$
Wind Speed (direction-3): U	$10m/\sec$
Moment of Inertia of Blades	$(0.50 \quad 0.50 \quad 0.90)10^9 kg-m$
Moment of Inertia of Turbine	$(34.0 \quad .075 \quad 35.0)10^9 kg-m$

Moment of Inertia of Rotor $(0.01 \quad 0.01 \quad 0.02)10^{9}\,kg-m$

The following equation is used for the angular velocity of the windmill blades, due to the wind speed.

$$\omega_{blade\_tip} = TSR * U / r \tag{75}$$

Continuing, the wind force, due to the wind speed, is approcimated using:

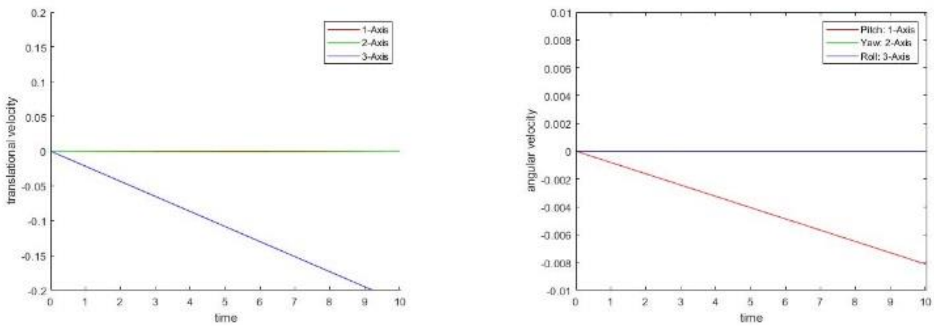
$$\omega_{blade\_tip} = C_D * 0.5 * Area_{Swept\_rotor} * U^2 \tag{76}$$

For the damping forces, equations based on the turbine tower angular velocity were formed. The equations were approximations and made to represent an estimate of the real-life variable. Problems occurred during the coding and including the damping calculations led to unexpected complications in the code. Considering that these forces wouldn't cause instability, but rather stability, they were removed. This way, the results would be more generic compared to a real-life situation, and the code remained stable.

3. Results

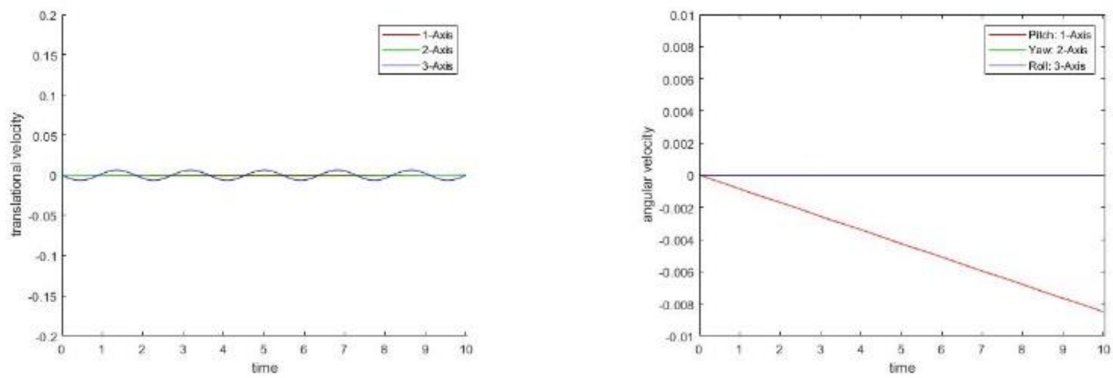
The results presented in this paragraph is based on the simulation interface formed according to the system of bodies presented in section 2.3-2.5. As mentioned, the disks implemented in the centre column of the structure are expected to act as an additional stabilizing factor.

3.1 Results and graphs



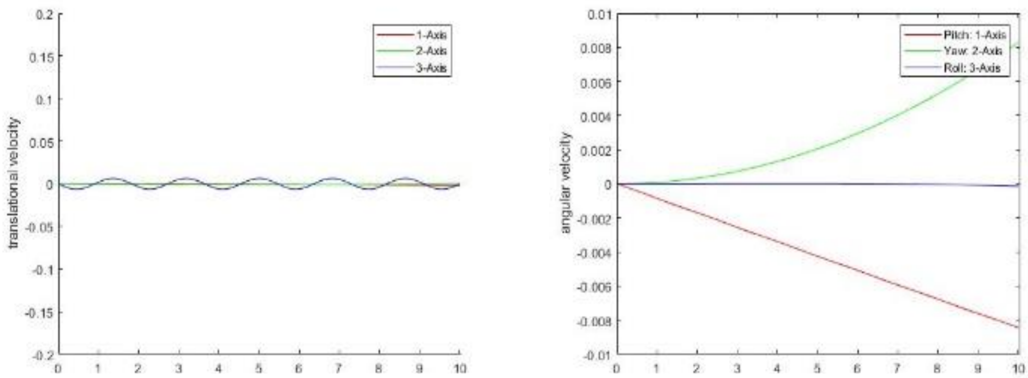
Figur 3: Translation rate (a), Angular Rate (b)

Figure 3 presents translational rates (a) and rotational rates (b) for blades locked, correcting rotor locked, turbine free to translate. Results show turbine translation and pitch. Furthermore in both cases, a negative linear change in the translational rate in third direction and a the rotation about the first axis. In both cases, the linear response is expected.



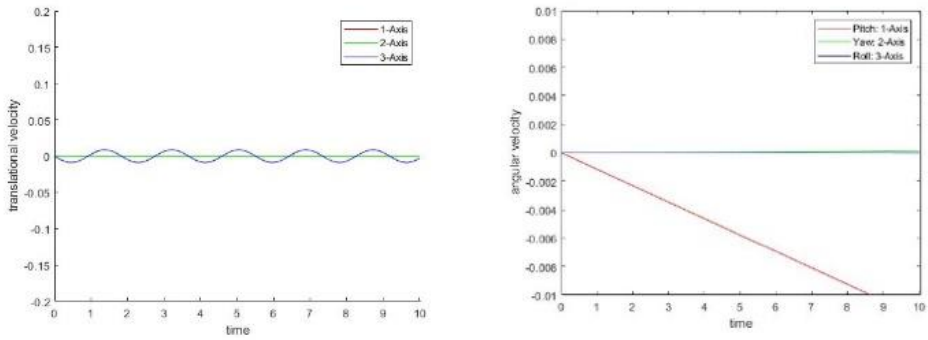
Figur 4: Translation rate (a), Angular Rate (b)

Figure 4 presents the results for: blades locked, correcting rotor locked, turbine fixed by cables. Translation minimized, pitch remains. Articially high tensile strength in cable: will enable future work to analyze fixation loads.



Figur 6: Translation rate (a), Angular Rate (b)

Figure 5 presents the results for rotating blades, correcting rotor locked. Fixation cables active. Blades induce Yaw.



Figur 5: Translation rate (a), Angular Rate (b)

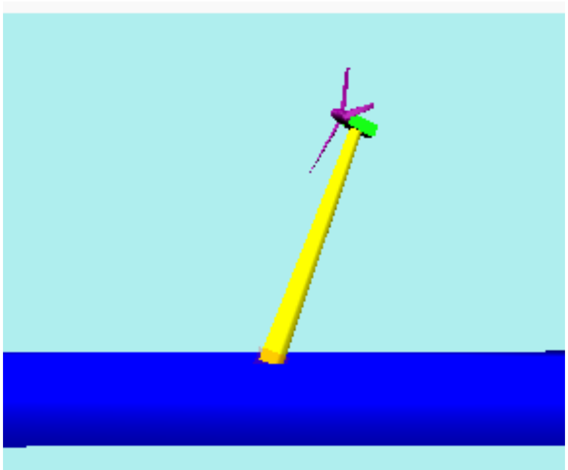
Figure 6 presents the results for rotating blades. Correcting rotor active and fixation cables active. Pitch remains, Yaw eliminated.

For the sake of the correcting rotor, the energy stored, considering the mass, and geometry was:

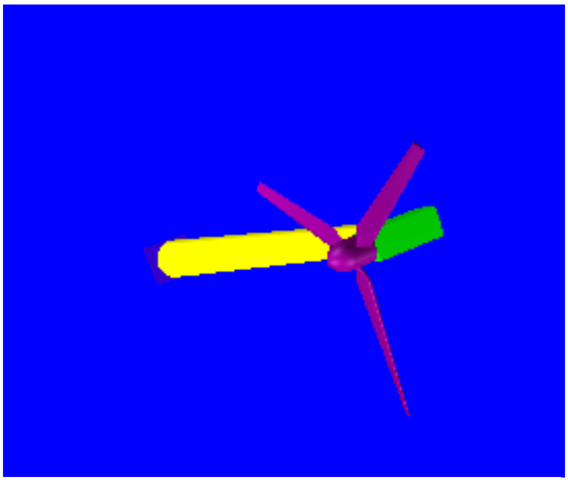
$$E_{Stored} = 2.2x10^{6} Joules$$

3.2 WebGL

WebGL (Web Graphics Library) is a JavaScript interface for rendering interactive 2D and 3D computer graphics. WebGL is compatible with most of the major web browsers such as Chrome, Firefox, Safari, and Opera. In addition, it is free of charge and can be used without the need for plugins. As with similar projects, the goal was to have a reader operate the system on their cell phones. However, the complexity of the Runge-Kutta method and the Rodriguez formula necessitated Matlab over Javascript to complete this first phase task. However, ThreeJS, client-side scripting language, cannot open files. Thus, bindings between Matlab and Javascript were forged. The user is unable to interact at this time, however, the results that would be seen is shown below. This feature is to be addressed for the next pass. For now, the pitch and yaw of the turbine is presented, without the correcting rotor. Figures 7 and 8 demonstrate the pitch and yaw in accordance with Figure 5. Figures 9 and 10 demonstrate the pitch and yaw in accordance with when the correcting rotor is active, figure 6.



Figur 8: Side View (pitch)



Figur 7: Top view (yaw)

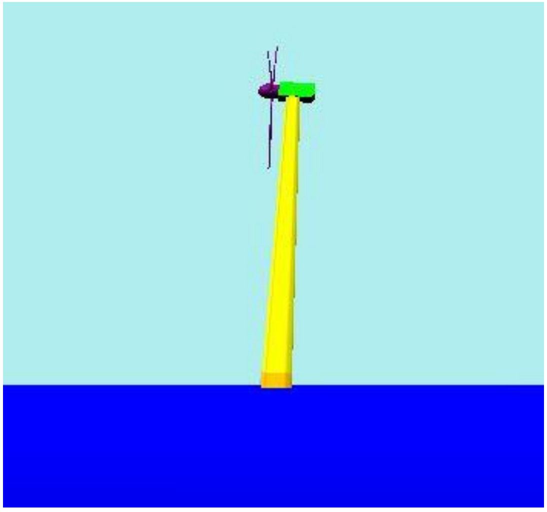


Figure 9: Side View (pitch)

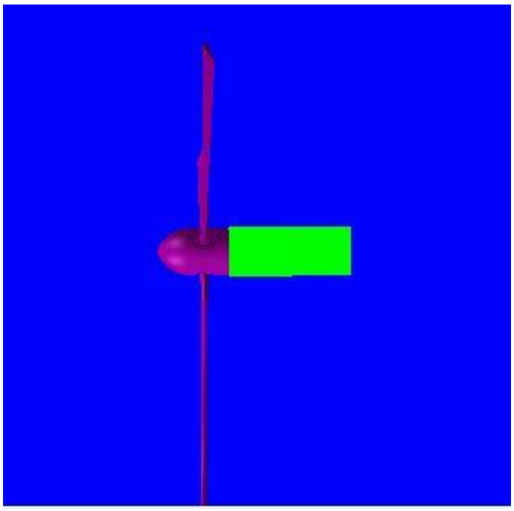


Figure 10: Top view (yaw)

4. Discussion

In contrast to the goal of this paper, these results seem too short for a full paper. Justifying, the inclusion in the IMECE conference. This adoption required significant time and energy, which wasn't expected beforehand. Hopefully, comments regarding the solicit advice and the guidance on the continuation of the work is given. For now the goal of this work was to demonstrate the pedagogical aspects behind the method, which was indeed fulfilled

Furthermore, while the results are fairly obvious, and expected, the general approach taken here will allow and open up for modelling offsets and additional links on the structure.

For now, it has been shown that a correction rotor can in fact stabilize the turbine while storing energy. Still, the ideal location and orientation of this correcting rotor has not been investigated. For now, the viscous effects of the water, the boyancy, and gravity, has not been taken into account. Despite these limitations, the work satisfies the proof of the concept—The primary goal of the first phase.

The results invite the question: could a stablizing rotor, facing in another direction, reduce the pitch while ensuring there is no yaw? In fact, can multiple disks be placed. One more issue is emphasized. Cosidering the work was done by undergraduate students, for the sake of edification, this demonstrates the simplicity of this method in dynamics.

5. Conclusion

The project acts as a foundation for future work on stability calculations for floating structures. It serves as proof that the installation of spinning discs for inducing the gyroscopic effect, in fact, can serve as an additional factor of stabilization for a floating structure. During the process, several questions and ideas for improvement have emerged. The location and placement of the spinning disks is one of these. Implementing a family of rotors in the tower of a wind turbine could allow for the usage of advanced A.I. technology to improve the location and orientation of these disks in real time. This way, one could achieve a fully automatic process allowing to get the best possible result. This process would be advanced but could possibly serve as a breakthrough in the fields of Marine Technology and renewable energy.

The project has sparked an interest in the undergraduate students, showing them that the knowledge gained within the field of dynamics can be so relevant in today's job market. As an experimental process, they want to come up with a way of efficiently accounting for the effects of gravity and buoyancy in MFM.

Although some factors of the project were too hard to complete, it is seen as a success. For the sake of producing a stable code, the viscous damping forces were ignored. This does indeed affect the results, but not in a negative manner, considering these forces would suffice in stabilizing the structure instead of inducing instability.

6. References

[1] H. Bendix, "forskning.no," 2016. [Online]. Available: https://forskning.no/alternativ-energi-havforskning-spor-en-forsker/spor-en-forsker-hvor-blir-det-av-bolgeenergien/378374. [Accessed: Jan. 11, 2019]

[2] N. Regjering, "Regjeringen.no," 2014. [Online]. Available: https://www.regjeringen.no/no/tema/klima-og-miljo/klima/innsiktsartikler-klima/gront-skifte/id2076832/. [Accessed: Jan. 11, 2019].

[8] T. Impelluso, "The moving frame method in dynamics: Reforming a curriculum and assessment," International Journal of Mechanical Engineering Education, pp. 158-191, 2018.

[3] Goldin, A., 2004, "Autonomous Gyroscopic Ocean-Wave Powered Generator: Invention of a New Energy Conversion Technology, "www.siemensfoundation.org/en/competition/2004-winners/aaron_goldin.html.

[4] Cartan, E., 1986, *On Manifolds with an Affine Connection and the Theory of General Relativity*, translated by A. Magnon and A. Ashtekar, Napoli, Italy, Bibiliopolis.

[5] H. Murakami, "A moving frame method for multibody dynamics using SE(3)," ASME 2015 International Mechanical Engineering Congress & Exposition, paper IMECE2015-51192., 2015.

[6] Holm, D. D., 2008, Geometric Mechanics, Part II: Rotating, Translating and Rolling, World Scientific, NJ.

[7] Wittenburg J, (2008) Dynamics of Multibody Systems, 2nd ed., Springer.

[8] Alexander Jacobsen Jardim, P., Tore Rein, J., Haveland, Ø., Rykkje, T. R., and Impelluso, T. J. (February 18, 2019). "Modeling Crane-Induced Ship Motion Using the Moving Frame Method." ASME. *J. Offshore Mech. Arct. Eng*. Oct. 2019; 141(5): 051103. https://doi.org/10.1115/1.4042536

[9] Austefjord, KO, Hestvik, MO, Larsen, LS, & Impelluso, TJ. "Modelling Subsea ROV Motion Using the Moving Frame Method." *Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition*. *Volume 4A: Dynamics, Vibration, and Control*. Pittsburgh, Pennsylvania, USA. November 9–15, 2018.. ASME. https://doi.org/10.1115/IMECE2018-86191

List of figures

Figur 1: Model of monopile windturbine created in Creo. 1

Figur 2: Schematic of the Wind Turbine with three moving frames 6

Figur 3: Translation rate (a), Angular Rate (b) 18

Figur 4: Translation rate (a), Angular Rate (b) 19

Figur 6: Translation rate (a), Angular Rate (b) 19

Figur 5: Translation rate (a), Angular Rate (b) 19

Figur 7: Side View (pitch).. 20

Figur 8: Top view (yaw) 20

Figur 9: Side View (pitch)..21

Figur 10: Top View (yaw)...21

Attachment 1 (Matlab code)

```
function WindmillRK()

%  These two lines clear the matlab memory and screen
clear
clc

% These are global variables that all functions can access
global Wind
global CableF
global CableM
global J1
global J2
global J3
global m1
global m2
global m3
global e2
global e3
global s21s
global s31s
global a2d
global a3d
global stiff
```

```
%Ignore this value of the spring stiffness for now.  It will have impact in
%only ONE spot and it will be obvious.
stiff = 0;

%Get to body 2 (blades) from CM of turbine): up 40 and out 2
s21 = zeros(3,1);
s21(1) = 0;
s21(2) = 95;
s21(3) = 1;
s21s= skew(s21);

%Get to body 3 (gyro) from CM of turbine): down 40 from CM
s31 = zeros(3,1);
s31(1) = 0;
s31(2) = 0;
s31(3) = 0;
s31s = skew(s31);

% The line below sets the wind force.  Direction not yet selected
% You could approximate the force from a wind speed function, but you can
% do that.
W = 10000;

% Set the mass of Turbine (1), blades (2), and gyro (3)
m1 = 50000;
```

```
m2 = m1 * 0.10;

m3 = m2 * 2;


h1 = 100;

r1 = 10;


r2 = 10;

r3 = 10;


% Set the angular velocity of blades (2) and gyro (3)

a2d = 0;

a3d = 0;


%set all time and stepping values

time = 0;

dt = 0.01;

time_end = 2;

steps = time_end/dt + 1;


% The two lines below state the LOCAL axis about which body 2 and body 3

% rotate.  Currently, they both face in the 3 direction

% By changing these numbers, we reorient the gyro.

% I am adhering to computer graphics: 2-axis is UP; 3 faces the wind

e2 = 3;

e3 = 3;
```

```
%Define array for J1 (turbine), J2 (blades), J3 (gyro)

J1 = zeros(3,3);

J2 = zeros(3,3);

J3 = zeros(3,3);


%Set the moment of inertia matrix for turbine which rises in 2-direction

J1(1,1) = m1 *(3*r1*r1 + h1) / 12.;

J1(2,2) = m1 * (r1 * r1 ) / 2.;

J1(3,3) = m1 *(3*r1*r1 + h1) / 12.;


%Set the moment of inertia matrix for blades which face in 3-direction

J2(1,1) = m2 * (r2 * r2) / 2.;

J2(2,2) = m2 * (r2 * r2) / 4.;

J2(3,3) = m2 * (r2 * r2) / 2.;


%Set the moment of inertia matrix for rotor which face in 3-direction

J3(1,1) = m3 * (r3 * r3) / 2.;

J3(2,2) = m3 * (r3 * r3) / 4.;

J3(3,3) = m3 * (r3 * r3) / 2.;


%Form the vector for the loads on the blades: the initial force is here.

%A time dependent function can always change this value.

Wind = zeros(3,1);

Wind(3) = -W;


CableF = zeros(3,1);
```

```
CableM = zeros(3,1);


%Form the inital rotation matrix, R1-dot and the minimal coordinates.
R1 = eye(3);
R1dot = zeros(3,3);
q     = zeros(6,1);


%The arrays below will hold the Cartesian coordinates for the turbine for
%each time step
ta  = zeros(steps,1);
x1d = zeros(steps,1);
x2d = zeros(steps,1);
x3d = zeros(steps,1);
x1  = zeros(steps,1);
x2  = zeros(steps,1);
x3  = zeros(steps,1);
w1  = zeros(steps,1);
w2  = zeros(steps,1);
w3  = zeros(steps,1);



for i = 2:steps


% We have R1 from the previous time step.  R1 is our pain in the ass.
% Because we get it from omega1.  We are going to ASSUME that we can use
% the rodriguez formula and ASSERT that R1 can be found IN THE MIDDLE of a
```

```
% step.  This means, that we will assume rotation STARTS from a PREVIOUS
% step.  So we will  obtain the rotation matrix for a prediction of an
% omega that is the average of the previous and the prediction.

    tn = time;

    % Like "stiffness" this line means nothing right now.
    % I should modify the displacement, inside of the RK routine.
    % There is no loss, now, for deciding the defelection from the
    % previous calculation.  Just save it.  Assume that x3 holds
    % the displacements in the 3 direction.
    disp = x3(i-1);


    %For RK4, we begin by using q as our first "p"rediction.
    %With that, and all information from previous time step, we can find
    %k1
    qp = q;
    k1 = getk(tn,       qp,        R1, R1dot, disp);

 % two things will now happen.  We will advance time by half a time step.
 % However, we will advance space (q contains displacement and rotation
 % rates) by k1*dt/2.  Since THOSE quantities are changed by the assumption
 % of a final projection, we must calculate all quantities in that time
 % window, that we can.
```

```
%Predict a new quantity
qp = q + k1 * 0.5 * dt;
%get the average in this time prediction window
qa = (q + qp )*0.5;
%extract the three omega terms for the windmill: pitch, yaw, roll.
o1 = qa(4:6,:);
%We will now assume that IN THIS TIME WINDOW, these numbers are
%constant, and we must use Rodriguez to find the roation matrix int
%this window.
%This o1 is for the entire duration of 0 to half*dt
xR1 = expMRodriguez(o1, dt*0.5);
%We "add" this rotation.  But do not forget that with regard to
%rotations, this means we post-multiply with this number as the new
%rotation mtrix

%Next, we "add" this rotation that happens DURING the time step.
%However the group theory posits that rotations ADD by POST
%multiplicatoin



R1temp = R1 * xR1;
%Use our formula to get the associated R1-dot:
%R1t * R1-dot = oskew
%R1-dot = R1 *oskew
R1dot = R1temp*skew(o1);
%Now we can compute k2
```

```
k2 = getk(tn + dt/2, qp,        R1, R1dot, disp);

%repeat the process with a half-step change in time and a change in
%space by the following
qp = q + k2 * 0.5 * dt;
qa = (q + qp )*0.5;
o1 = qa(4:6,:);
xR1 = expMRodriguez(o1, dt*0.5);
R1temp = R1 * xR1;
R1dot = R1temp*skew(o1);
k3 = getk(tn + dt/2, qp,        R1, R1dot, disp);

%Using K3, we return to a full time step.
qp = q + k3 *  dt;
qa = (q + qp )*0.5;
o1 = qa(4:6,:);
xR1 = expMRodriguez(o1, dt);
R1temp = R1 * xR1;
R1dot = R1temp*skew(o1);
k4 = getk(tn + dt,   qp,        R1, R1dot, disp);

%Now we make our FINAL approximation to the next time step.
qp = q + dt * (k1 + 2*k2 + 2*k3 + k4) / 6.;
%Once again, in that window, we must compute R1 and R1dot
qa = (q + qp )*0.5;
o1 = qa(4:6,:);
```

```
xR1 = expMRodriguez(o1, dt);

R1 = R1 * xR1;

R1dot = R1*skew(o1);


%Save qp as q for the next step

q = qp;


% pull out what we need

time = time + dt;

ta(i)  = time;

x1d(i) = q(1);

x2d(i) = q(2);

x3d(i) = q(3);

w1(i)  = q(4);

w2(i)  = q(5);

w3(i)  = q(6);


%Use simple forward difference for the postions.

x1(i) = x1(i-1) + x1d(i)*dt;

x2(i) = x2(i-1) + x2d(i)*dt;

x3(i) = x3(i-1) + x3d(i)*dt;
```

```
end

close all
figure(1)
hold on
plot(ta, x1, 'r')
hold on
plot(ta, x2, 'g')
hold on
plot(ta, x3, 'b')
hold off

figure(2)
plot(ta, w1, 'r')
hold on
plot(ta, w2, 'g')
hold on
plot(ta, w3, 'b')
hold off

end
```

```
function k = getk(time, q, R1, R1dot, disp)



% These are global variables that all functions can access

% They must be repeated in the same order.  This is stupid.


global Wind

global CableF

global CableM

global J1

global J2

global J3

global m1

global m2

global m3

global e2

global e3

global s21s

global s31s

global a2d

global a3d

global stiff



    %FIRST, for this time and approximation, we need the information
```

```
%about the two standard rotations of blades and rotor


%For this time, compute the angle of blade and gyro
a2 = a2d * time;
a3 = a3d * time;
%Now compute the cosine and sine of those bodies
ca2 = cos(a2);
sa2 = sin(a2);
ca3 = cos(a3);
sa3 = sin(a3);
%Now compute the relative rotation matrices
R21 = [ca2, -sa2, 0; sa2, ca2, 0; 0,0,1];
R31 = [ca3, -sa3, 0; sa3, ca3, 0; 0,0,1];
%Now compute the relative time derivative of rotation matrices
R21d= [-sa2, -ca2, 0; ca2, -sa2, 0; 0,0,0];
R31d= [-sa3, -ca3, 0; ca3, -sa3, 0; 0,0,0];


R21d = R21d * a2d;
R31d = R31d * a3d;


%pull out o1 for the windmill from q
o1 = q(4:6,:);
%skew it
o1s = skew(o1);


%Build o2
```

```
o2   = R21' * o1;
o2(e2) = o2(e2) + a2d;
o2s = skew(o2);

%Build o3
o3   = R31' * o1;
o3(e3) = o3(e3) + a3d;
o3s = skew(o3);

%Compute N12
N12 = m2 * R1dot * s21s' + m3 * R1dot * s31s';

%Compute N22
N1 = o1s * J1;
N2 = m2 * s21s * (R1'*R1dot) * s21s';
N3 = R21 *(J2 * R21d' + o2s * J2 * R21');
N4 = m3 * s31s * (R1'*R1dot) * s31s';
N5 = R31 *(J3 * R31d' + o3s * J3 * R31');
N22 = N1 + N2 + N3 + N4 + N5;


%Compute all components of M-star
Mstar(1:3, 1:3) = (m1 + m2 + m3) * eye(3);
Mstar(1:3, 4:6) = m2 * R1 * s21s' + m3 * R1 * s31s';
Mstar(4:6, 1:3) = m2 * s21s * R1' + m3 * s31s * R1';
M1 = J1;
```

```
M2 = m2 * (s21s * s21s');

M3 = R21 * J2 * R21';

M4 = m3 * (s31s * s31s');

M5 = R31 * J3 * R31';

Mstar(4:6,4:6) = M1 + M2 + M3 + M4 + M5;



%This reaction force below is from the cables acting like springs.

Rxn = -disp * stiff;

CableF(3)= Rxn;


%Compute all components of Magenta

FR1 = CableF + Wind - N12 * o1;

F1  = CableM;

F2  = s21s * R1' * Wind;

F3  = N22 * o1;

F4 = a2d * R21 * o2s * ( J2(:, e2));

F5 = a3d * R31 * o3s * ( J3(:, e3));

FR2 = F1 + F2 - F3 - F4 - F5;



FF = vertcat(FR1, FR2);


k =  Mstar \  FF;
```

```
end

function ms = skew(vec)
    ms = zeros(3,3);
    ms(1,2) = -vec(3);
    ms(1,3) =  vec(2);
    ms(2,3) = -vec(1);

    ms(2,1) =  -ms(1,2);
    ms(3,1) =  -ms(1,3);
    ms(3,2) =  -ms(2,3);
end

function R = expMRodriguez(o1, t)
```

```
w1 = o1(1);
w2 = o1(2);
w3 = o1(3);

    normw = sqrt(w1*w1 + w2*w2 + w3*w3);


    I = eye(3);

R = I;

    if(normw > 0.00000001)
  O = skew(o1);
  fs = sin(normw * t) / normw;
  fc = (1 - cos(normw * t)) / normw / normw;
  R = I + O * fs + O * O * fc;
end


end
```

Attachment 2 (IMECE report)

Comes on the next page.

Proceedings of the ASME 2021
International Mechanical Engineering Congress and Exposition
IMECE2021
November 1-5, 2021, Virtual, Online

DRAFT IMECE2021-72984

ENERGY STORAGE AND STABILIZATION SIMULATION OF FLOATING WIND TURBINES

Martinus K. Aarmo
Western Norway University of Applied Sciences
(HVL), Bergen, Norway
maaarmo@icloud.com

Magnus N. Sivesind
Western Norway University of Applied Sciences
(HVL), Bergen, Norway
magnusns99@gmail.com

Jan Michael Simon Bartl
Western Norway University of Applied Sciences
(HVL), Bergen, Norway
jan.bartl@hvl.no

David Lande-Sudall
Western Norway University of Applied Sciences
(HVL), Bergen, Norway
david.lande-sudall@hvl.no

Thomas J. Impelluso
Western Norway University of Applied Sciences
(HVL), Bergen, Norway
tjm@hvl.no

ABSTRACT

This project has a dual focus: to store energy extracted from floating wind turbines; and to stabilize such structures. The energy extracted can be delivered after the wind subsides. The gyroscopic effect will ensure stability. The project will deploy the Moving Frame Method (MFM) to analyze the kinematics and kinetics of the system. The MFM exploits aspects of Lie Group Theory in place of vector-based dynamics. It leverages the work of Elie Cartan to model all moving bodies. Finally, it deploys a compact notation for both 3D and 2D. The research is an extension of past projects built on the principal of incorporating spinning disks to counter the instability in the system. The improvement comes in the form of including relevant forces acting on the system, implementation of an improved numerical integration scheme, accountment of mooring lines, and an approximation of simplified damping forces. The project defines the initial spin of the disks as a prescribed variable. We then use the Runge-Kutta method for numerical integration of the equations of motion. We update the data with an assumed correction for the rotation matrices that exploits Rodriguez' formula. Afterwards we create a simulation by creating a port from Matlab to the Web Graphics Library and Three JS using Javascript.

NOMENCLATURE

$[B]$:	Transformation matrix to generalized coordinates
$[C]$:	Transformation matrix for prescribed rates
$[D]$:	Combined angular velocity matrix
$\mathbf{e}$:	Unit basis vector
E:	Frame connection matrix
$\{F\}$:	Force and moment list
$\{F^*\}$:	Generalized force and moment list
g:	Gravitational acceleration
I_3:	3×3 Identity matrix
J:	3×3 Mass moment of inertia matrix
K:	Kinetic energy
m:	Mass
$[M]$:	Mass matrix
$[M^*]$:	Reduced mass matrix
$[N]$:	Non-linear velocity matrix
$[N^*]$:	Reduced non-linear velocity matrix
$q(t)$:	Generalized coordinates
$\{\dot{q}(t)\}$:	Generalized velocity
$\{\ddot{q}(t)\}$:	Generalized acceleration
$\{\dot{r}(t)\}$:	Generalized prescribed velocity
R:	3×3 Rotation matrix
$[T^*]$:	Reduced velocity matrix for prescribed rates
ω:	Angular velocity components
$\overleftrightarrow{\omega}$:	Skew-symmetric angular velocity matrix

INTRODUCTION

Offshore wind energy has become highly relevant for the past years. According to recent reports, the industry is expected to grow even further in the years to come [1]. Research from such reports states that offshore wind has the capability of producing more than 420 000 TWh. This amount is more than 18 times the electricity demand of today [2].

Norway endeavors to be one of the most sustainable countries in the world. With a long coastline, Norway has a great opportunity to develop and invest in offshore solutions. While land-based wind turbines have been critiqued due to their impact on nature when being constructed and operated, offshore installations do not. They do however have an impact on marine life offshore. Still, despite this, there is a strong wind resource offshore and more space available. However, offshore wind turbines are a subject to harsher elements such as waves and stronger winds compared to onshore, and all this must be considered.

The water depth off the Norwegian cost is generally too deep for bottom fixed foundations. Therefore, floating wind turbines with a mooring system, appears to be a more preferable solution. The challenge with this concept, however, is based on the stability of the floating wind turbines, wind- and wave induced forces. In this paper, the concept behind a self-stabilizing wind turbine is explored. This mechanism is based on the principle behind a gyroscopic mechanism. Finally, the extracted energy to drive the correcting rotors, is converted into kinetic energy in the tower of the turbine. Essentially, the disks will serendipitously store energy generating a stabilizing effect on the turbine itself. To carry out this work the solution adopts the moving frame method (MFM) in dynamics.

THE MOVING FRAME METHOD

The reader may find an introduction to an undergraduate and graduate Moving Frame Method, along with pedagogical assessement in Impelluso [3]. The following section summarizes salient elements of the MFM.

GENERAL PRINCIPLES OF THE MFM

Kinematics using SO(3)

At the center of mass of each body (α) we place a time-dependent moving frame:

$$\mathbf{e}^{(\alpha)}(t) = \left(\mathbf{e}_1^{(\alpha)}(t)\;\; \mathbf{e}_2^{(\alpha)}(t)\;\; \mathbf{e}_3^{(\alpha)}(t)\right) \tag{1}$$

In the previous, **e** is a unit vector and the subscript denotes the Cartersian coordinate direction. Set $t = 0$ to define and deposit an inertial frame from a moving frame (as if peeling off a decal):

$$\mathbf{e}^{I} = \left(\mathbf{e}_1^{I}\;\; \mathbf{e}_2^{I}\;\; \mathbf{e}_3^{I}\right) = \left(\mathbf{e}_1^{(\alpha)}(0)\;\; \mathbf{e}_2^{(\alpha)}(0)\;\; \mathbf{e}_3^{(\alpha)}(0)\right) \tag{2}$$

Define the absolute position vector $\mathbf{r}_C^{(\alpha)}(t)$ of a frame as a translation from the inertial frame $\mathbf{e}^I$ using a compact notation:

$$\mathbf{r}_C^{(\alpha)}(t) = \mathbf{e}^I x_C^{(\alpha)}(t) = \mathbf{e}^I \left(x_{1C}^{(\alpha)}(t) x_{2C}^{(\alpha)}(t) x_{3C}^{(\alpha)}(t)\right)^T \tag{3}$$

In (3) we use $x_C^{(\alpha)}(t)$ to represent, in vertical form (transpose, above), the *absolute* coordinates of the distance to the center of mass of a body (subscript C), expressed in the inertial frame.

Assert the *relative* position vector of a frame $(\alpha + 1)$ from another frame (α) by $\mathbf{s}_C{}^{(\alpha+1/\alpha)}(t)$. Express this relative translation in the α-frame:

$$\mathbf{s}_C{}^{(\alpha+1/\alpha)}(t) = \mathbf{e}^{(\alpha)}(t) s_C{}^{(\alpha+1/\alpha)}(t) \tag{4}$$

By adding the absolute position vector of the α-frame $\mathbf{r}_C^{(\alpha)}(t)$ and the relative position vector $\mathbf{s}_C{}^{(\alpha+1/\alpha)}(t)$, we obtain the absolute position vector of the $(\alpha + 1)$ frame: frame, $\mathbf{r}_C^{(\alpha+1)}(t)$:

$$\mathbf{r}_C^{(\alpha+1)}(t) = \mathbf{r}_C^{(\alpha)}(t) + \mathbf{e}^{(\alpha)}(t) s_C^{(\alpha+1/\alpha)}(t) \tag{5}$$

Let us now turn our attention to frame orientations. We use a rotation matrix, a member of the Special Orthogonal Group $R \in SO(3)$, to relate the orientation of a moving frame to an inertial frame:

$$\mathbf{e}^{(\alpha)}(t) = \mathbf{e}^I R^{(\alpha)}(t) \tag{6}$$

The relative rotation of a frame $(\alpha + 1)$ from another frame (α) can be written as:

$$\mathbf{e}^{(\alpha+1)}(t) = \mathbf{e}^{(\alpha)}(t) R^{(\alpha+1/\alpha)}(t) \tag{7}$$

The orientation of body $(\alpha + 1)$ can be expressed in the inertial frame by inserting equation (6) into (7) and exploiting the closure property of the SO(3) Group:

$$\mathbf{e}^{(\alpha+1)}(t) = \mathbf{e}^I R^{(\alpha)}(t) R^{(\alpha+1/\alpha)}(t) = \mathbf{e}^I R^{(\alpha+1)}(t) \tag{8}$$

As a property of SO(3), the inverse of a rotation matrix is the transpose:

$$\left(R^{(\alpha)}(t)\right)^{-1} = \left(R^{(\alpha)}(t)\right)^T \tag{9}$$

The time rate of frame rotation is (with time depdendent R):

$$\dot{\mathbf{e}}^{(\alpha)}(t) = \mathbf{e}^I \dot{R}^{(\alpha)}(t) \tag{10}$$

We use (9) in (6) to formulate the inertial frame in terms of the moving frame and then substitute the result into (10) to obtain:

$$\dot{\mathbf{e}}^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t) \left(R^{(\alpha)}(t)\right)^T \dot{R}^{(\alpha)}(t) \tag{11}$$

The time rate of frame rotation is now expressed in its own frame, satisfying the thoughts of Elie Cartan [4].

It can be shown (Lie Group Theory), that the matrix products in (11) produce a skew symmetric matrix. We thus define the skew-symmetric angular velocity matrix. We note that this element is a member of the associated algebra, so(3):

$$\overleftrightarrow{\omega^{(\alpha)}(t)} = \left(R^{(\alpha)}(t)\right)^T \dot{R}^{(\alpha)}(t) = \begin{bmatrix} 0 & -\omega_3^{(\alpha)}(t) & \omega_2^{(\alpha)}(t) \\ \omega_3^{(\alpha)}(t) & 0 & -\omega_1^{(\alpha)}(t) \\ -\omega_2^{(\alpha)}(t) & \omega_1^{(\alpha)}(t) & 0 \end{bmatrix} \quad (12)$$

We rewrite equation (11) as:

$$\dot{\mathbf{e}}^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t)\overleftrightarrow{\omega^{(\alpha)}(t)} \quad (13)$$

The skew-symmetric angular velocity matrix is isomorphic to the same frame to the angular velocity vector of that frame:

$$\boldsymbol{\omega}^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t)\begin{pmatrix} \omega_1^{(\alpha)}(t) \\ \omega_2^{(\alpha)}(t) \\ \omega_3^{(\alpha)}(t) \end{pmatrix} \quad (14)$$

In (14) above, unlike in planar dynamics, we see the basis frame is time dependent.

Kinematics using SE(3)

Before we begin, we assert that the analysis of specifics of this windmill turbine could be conducted with the aforementioned work, alone—SO(3). However, we amend this approach with SE(3), [5] as it is more readily extensible and is being used in the expansion of this work, currently underway. We present here, only an overview.

We combine the rotational and translational data of a frame (α), in one structure. We define the 4×4 absolute *frame connection matrix* (a member of the Special Euclidean Group), $E \in SE(3)$:

$$E^{(\alpha)}(t) = \begin{bmatrix} R^{(\alpha)}(t) & x_C^{(\alpha)}(t) \\ 0_3^T & 1 \end{bmatrix} \quad (15)$$

We define an *inertial frame connection*. This consists of the frame and its position, represented as:

$$\begin{pmatrix} \mathbf{e}^I & \mathbf{0} \end{pmatrix} \quad (16)$$

Similarly, we represent the *moving frame connection* as:

$$\begin{pmatrix} \mathbf{e}^{(\alpha)}(t) & \mathbf{r}_C^{(\alpha)}(t) \end{pmatrix} \quad (17)$$

The structure in (17) contains both the frame and its position from the inertial frame. We relate the inertial frame connection (16) and the moving frame connection (17) by utilizing the *absolute* frame connection matrix (15):

$$\begin{pmatrix} \mathbf{e}^{(\alpha)}(t) & \mathbf{r}_C^{(\alpha)}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{e}^I & \mathbf{0} \end{pmatrix} E^{(\alpha)}(t) \quad (18)$$

Moving to the relative forms, the *relative* frame connection matrix is defined as:

$$E^{(\alpha+1/\alpha)}(t) = \begin{bmatrix} R^{(\alpha+1/\alpha)}(t) & s_C^{(\alpha+1/\alpha)}(t) \\ 0_3^T & 1 \end{bmatrix} \quad (19)$$

We use equation (19) to express the relative relationship between two moving frames, ($\alpha + 1$) and (α):

$$\begin{pmatrix} \mathbf{e}^{(\alpha+1)}(t) & \mathbf{r}_C^{(\alpha+1)}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{e}^{(\alpha)}(t) & \mathbf{r}_C^{(\alpha)}(t) \end{pmatrix} E^{(\alpha+1/\alpha)}(t) \quad (20)$$

Equation (20), with its defining element (19), recapitulates equations (5) and (7).

The absolute frame connection matrix of body (α + 1) can be found as the product of the absolute frame connection matrix of body (α) and the relative frame connection matrix that relates them (as a result of the closure property of the SE(3) group):

$$E^{(\alpha+1)}(t) = E^{(\alpha)}(t)E^{(\alpha+1/\alpha)}(t) \quad (21)$$

Rather than belabor details, we advance to implementing SE(3) theory in tutorial style, through an example for edification using a wind turbine.

KINEMATICS OF THE FLOATING WIND TURBINE

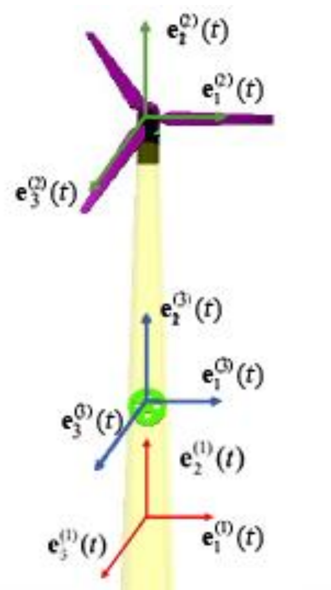


Figure 1 – Schematic of the Wind Turbine with three moving frames.

First Frame – The Turbine Tower

We place the first moving frame $\mathbf{e}^{(1)}(t)$ at the center of mass of the tower of the turbine. At $t = 0$ we deposit an inertial frame from the first frame (we do not show the inertial frame):

$$\mathbf{e}^I \equiv \mathbf{e}^{(1)}(0) \quad (22)$$

The orientation of the first moving frame is expressed as follows:

$$\mathbf{e}^{(1)}(t) = \mathbf{e}^{I} R^{(1)}(t) \tag{23}$$

The elements of $R^{(1)}(t)$ will contain information about the pitch, yaw and roll of the turbine from an inertial configuration.

The displacement of the first moving frame is stated as:

$$\mathbf{r}_c^{(1)}(t) = \mathbf{e}^{I} x_c^{(1)}(t) \tag{24}$$

We may now immediately apply Equation (15) and (18) in the form of a frame connection relationship (actually, its inverse in 25b). Then, next, continuing, we take the time rate of the frame connection of the same form (25b).

$$\left(\mathbf{e}^{I} \;\; \mathbf{0}\right) = \left(\mathbf{e}^{(1)}(t) \;\; \mathbf{r}_C^{(1)}(t)\right)\left(E^{(1)}(t)\right)^{-1} \tag{25a}$$

$$\left(\dot{\mathbf{e}}^{(1)}(t) \;\; \dot{\mathbf{r}}_C^{(1)}(t)\right) = \left(\mathbf{e}^{I} \;\; \mathbf{0}\right)\dot{E}^{(1)}(t) \tag{25b}$$

The time rate of the frame connection matrix $\dot{E}^{(1)}(t)$ is found by taking the time derivative of each data structure:

$$\dot{E}^{(1)}(t) = \begin{bmatrix} \dot{R}^{(1)}(t) & \dot{x}_c^{(1)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{26}$$

The inverse of the frame connection matrix, is expressed as (due to $E \in \mathrm{SE}(3)$):

$$\left(E^{(1)}(t)\right)^{-1} = \begin{bmatrix} \left(R^{(1)}(t)\right)^T & -\left(R^{(1)}(t)\right)^T x_C^{(1)}(t) \\ 0_3^T & 1 \end{bmatrix} \tag{27}$$

We use (25) in (21) to formulate the inertial frame connection in terms of the moving frame connection to obtain:

$$\left(\dot{\mathbf{e}}^{(1)}(t) \;\; \dot{\mathbf{r}}_C^{(1)}(t)\right) = \left(\mathbf{e}^{(1)}(t) \;\; \mathbf{r}_C^{(1)}(t)\right)\left(E^{(1)}(t)\right)^{-1}\dot{E}^{(1)}(t) \tag{28}$$

We define the absolute time rate of frame connection matrix for the first body, $\Omega^{(1)}$ as the product of $\left(E^{(1)}(t)\right)^{-1}$ and $\dot{E}^{(1)}(t)$. We note that $\Omega \in \mathrm{se}(3)$ (the algebra associated with the SE(3) group):

$$\Omega^{(1)} = \left(E^{(1)}(t)\right)^{-1}\dot{E}^{(1)}(t) \tag{29}$$

As a result, in keeping with the view of Cartan (expressing the change of structure in terms of the same structure, as in Eqn. 13) we can rewrite equation (28) as:

$$\left(\dot{\mathbf{e}}^{(1)}(t) \;\; \dot{\mathbf{r}}_C^{(1)}(t)\right) = \left(\mathbf{e}^{(1)}(t) \;\; \mathbf{r}_C^{(1)}(t)\right)\Omega^{(1)}(t) \tag{30}$$

$\Omega^{(1)}$ multiplied out in matrix from:

$$\Omega^{(1)} = \begin{bmatrix} \left(R^{(1)}(t)\right)^T \dot{R}^{(1)}(t) & \left(R^{(1)}(t)\right)^T \dot{x}_C^{(1)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{31}$$

By comparing the expression to (12), we can rewrite (31) as:

$$\Omega^{(1)} = \begin{bmatrix} \overleftrightarrow{\omega^{(1)}(t)} & \left(R^{(1)}(t)\right)^T \dot{x}_C^{(1)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{32}$$

By expanding, we can extract parts of the system and recapitulate:

$$\dot{\mathbf{e}}^{(1)}(t) = \mathbf{e}^{(1)}(t)\overleftrightarrow{\omega^{(1)}(t)} \tag{33}$$

The second equation extracted from equation (30) is:

$$\dot{\mathbf{r}}_C^{(1)}(t) = \mathbf{e}^{(1)}(t)\left(R^{(1)}(t)\right)^T \dot{x}_C^{(1)}(t) \tag{34}$$

Thus, state the translational velocity as:

$$\dot{\mathbf{r}}_C^{(1)}(t) = \mathbf{e}^{I}\dot{x}_C^{(1)}(t) \tag{35}$$

This marks the point where the first body is properly assessed, and the equations of the first body are listed.

$$\dot{\mathbf{e}}^{(1)}(t) = \mathbf{e}^{(1)}(t)\overleftrightarrow{\omega^{(1)}(t)} \tag{36}$$

$$\dot{\mathbf{r}}_c^{(1)}(t) = \mathbf{e}^{I}\dot{x}_c^{(1)}(t) \tag{37}$$

The second frame – The Turbine Rotor

The second body in this analysis is the turbine rotor of the wind turbine, is a branch off the first body. We placed a frame at the center of mass of the blades. The relationship of the second frame connection from the first frame (turbine) connection is:

$$(\mathbf{e}^{(2)}(t) \;\; \mathbf{r}_c^{(2)}(t)) \equiv \left(\mathbf{e}^{(1)}(t) \;\; \mathbf{r}_c^{(1)}(t)\right)E^{(2/1)}(t) \tag{38}$$

Or just the frame connection matrix as:

$$E^{(2/1)}(t) = \begin{bmatrix} R^{(2/1)}(t) & s_c^{(2/1)} \\ 0 & 1 \end{bmatrix} \tag{39}$$

This frame connection matrix could be expressed as the product of two, in which displacement and rotation are separated—this, only for the sake of edification. Thus, in the first matrix, below, the two values locate the center of mass of the windmill blades, from the center of mass of the turbine body. While we would prefer to reserve formal numbers to the solution process, we break that rule here, just to demonstrate.

Progress in the 2-direction, $d^{(1)}$, and then out (along the nacelle) in the 3 direction, $h^{(1)}$. The boxed column in the first matrix, below demonstrates this.

$$E^{(2/1)}(t) = \begin{bmatrix} 1 & 0 & 0 & d^{(1)} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h^{(1)} \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) & 0 & 0 \\ \sin(\theta(t)) & \cos(\theta(t)) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{40}$$

In the second matrix, above, set aside the conforming "1" in the lower right corner and the column above it. The remaining 3x3 marix represents a standard rotation about the local body's (the turbine's) 3-axis, for the rotation of the frame by a time dependent angle: θ.

With (40) and (26), we can state the absolute frame connection matrix from the inertial frame

$$(\mathbf{e}^{(2)}(t)\ \ \mathbf{r}_c^{(2)}(t)) \equiv \left(\mathbf{e}^I\ \ \mathbf{0}\right) E^{(2)}(t) \tag{41}$$

Where:

$$E^{(2)}(t) = \begin{bmatrix} R^{(1)}(t) & x_c^{(1)}(t) \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} R^{(2/1)}(t) & s_c^{(2/1)} \\ 0 & 1 \end{bmatrix} \tag{42}$$

We may expand (42) to matrix form as:

$$E^{(2)}(t) = \begin{bmatrix} \left(R^{(1)}(t) R^{(2/1)}(t)\right) & \left(R^{(1)}(t) s_c^{(2/1)} + x_c^{(1)}(t)\right) \\ 0 & 1 \end{bmatrix} \tag{43}$$

Continuing, the inverse of the frame connection matrix is formed:

$$\begin{gathered}\left(E^{(2)}(t)\right)^{-1} = \\ \begin{bmatrix} \left(R^{(2/1)}(t)\right)^T \left(R^{(1)}(t)\right)^T & -\left(\left(R^{(2/1)}(t)\right)^T s_c^{(B/1)} + \left(R^{(2)}(t)\right)^T x_c^{(1)}(t)\right) \\ 0 & 1 \end{bmatrix}\end{gathered} \tag{44}$$

In the same manner as previously, the time rate of the frame connection matrix is developed by deriving each block of the matrix. Considering the location of the blades will not translate according to the main tower, one can cancel out $\dot{s}_c^{(B/1)}$:

$$\dot{E}^{(2)}(t) = \begin{bmatrix} \dot{R}^{(1)}(t) R^{(2/1)}(t) + R^{(1)}(t) \dot{R}^{(2/1)}(t) & \dot{R}^{(1)}(t) s_c^{(2/1)} + \dot{x}_c^{(1)}(t) \\ 0 & 0 \end{bmatrix} \tag{45}$$

The general form of the omega matrix is stated as:

$$\Omega^{(2)}(t) \equiv \left(E^{(2)}(t)\right)^{-1} \dot{E}^{(2)}(t) \tag{46}$$

In expanded notation:

$$\Omega^{(2)}(t) \equiv \begin{bmatrix} \left(\left(R^{(2/1)}(t)\right)^T \overleftrightarrow{\omega^{(1)}(t)} R^{(2/1)}(t) + \overleftrightarrow{\omega^{(2/1)}(t)}\right) & \left(R^{(2/1)}(t)\right)^T \left(\overleftrightarrow{\omega^{(1)}(t)} s_c^{(2/1)} + \left(R^{(1)}(t)\right)^T \dot{x}_c^{(1)}(t)\right) \\ 0 & 0 \end{bmatrix} \tag{47}$$

By comparing with the general definition of an omega matrix, one can extract the definition of $\overleftrightarrow{\omega^{(2)}(t)}$. The omega term, is extracted and formulated as (using aspects of the Lie Algebra):

$$\omega^{(2)}(t) = \left(R^{(2/1)}(t)\right)^T \omega^{(1)}(t) + \omega^{(2/1)}(t) \tag{48}$$

Since we already alluded to the blade-axis of rotation, (48) can be reformulated using:

- $e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$
- The spin rate being perscribed with a rate: $\dot{\zeta}$

Now, the omega vector is expressed as:

$$\omega^{(2)}(t) = \left(R^{(2/1)}(t)\right)^T \omega^{(1)}(t) + \dot{\zeta} e_i \tag{49}$$

For the translation expression, the term is modified by bringing the rotational matrix to the right hand side: Hence the expression for the translation is modified, accounting for the rotation and translation of the turbine tower:

$$\dot{x}_c^{(2)}(t) = R^{(1)} \left(\overleftrightarrow{\omega^{(1)}(t)} s_c^{(2/1)} + \left(R^{(1)}(t)\right)^T \dot{x}_c^{(1)}(t)\right) \tag{50}$$

Like the expression for omega, one will eventually require the omegas and the translation vectors at the end of each term (when we relate generalized and Cartesian coordinates). The term is modified by switching the omega with a negative, and then transposing to negate the negative:

$$\dot{x}_c^{(2)}(t) = \left(R^{(1)} \left(\overleftrightarrow{s_c^{(2/1)}}\right)^T \omega^{(1)}(t) + \dot{x}_c^{(1)}(t)\right) \tag{51}$$

This concludes the extraction of equations for frame/body 2.

The Third Frame – The Correcting Rotating Disk

The location of the third body is fortuitous: it does not extend off the second frame, but, like the blades, off the turbine tower; it is also a rotating structure. This means we can rapidly obtain the necessary equations from those of the blades. In future work, we will expand this and allow for internal mechanisms.

One can extract the equations for the kinematics of the correcting rotor disk because the the turbine rotor rotates about the same axes, while displaced a certain distance from the CM. By considering this generality, one can change the subscripts of the expressions for the second frame, and obtain the expressions for the third body, in general terms, with the rotor spin signified by $\dot{\psi}$.

$$\dot{x}_c^{(3)}(t) = \dot{x}_c^{(1)}(t) + R^{(1)} \left(\overleftrightarrow{s_c^{(3/1)}}\right)^T \omega^{(1)}(t) \tag{52}$$

And:

$$\omega^{(3)}(t) = \left(R^{(3/1)}(t)\right)^T \omega^{(1)}(t) + e_3 \dot{\psi} \tag{53}$$

We have now obtained all kinematic expressions and could turn to kinetics. First, however, we separate the prescribed rotations for the wind turbine and correcting rotor, from the two generalized variables. With this compact matrix form, below, we have expressions for all the relevant Cartesian variables necessary to conduct the minimization required of the Principle of Virtual Work.

$$
\begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \\ \dot{x}^{(2)}(t) \\ \omega^{(2)}(t) \\ \dot{x}^{(3)}(t) \\ \omega^{(3)}(t) \end{pmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \\ I & R^{(1)}\left(\overleftrightarrow{s_c^{(2/1)}}\right)^T \\ 0 & \left(R^{(2/1)}(t)\right)^T \\ I & R^{(1)}\left(\overleftrightarrow{s_c^{(3/1)}}\right)^T \\ 0 & \left(R^{(3/1)}(t)\right)^T \end{bmatrix} \begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ e_i & 0 \\ 0 & 0 \\ 0 & e_j \end{bmatrix} \begin{pmatrix} \dot{\zeta} \\ \dot{\psi} \end{pmatrix} \tag{54}
$$

This stated, we find it easier to formulate the equation of motion using the following block matrix forms where the definitions of C and B are obvious, by relating them to (54), above.

$$
\{\dot{X}(t)\} = \begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \\ \dot{x}^{(2)}(t) \\ \omega^{(2)}(t) \\ \dot{x}^{(3)}(t) \\ \omega^{(3)}(t) \end{pmatrix} = B\begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \end{pmatrix} + C\begin{pmatrix} \dot{\zeta} \\ \dot{\psi} \end{pmatrix} \tag{55}
$$

Finally, we recast it most simply with obvious definitions as:

$$
\dot{X}(t) = B\dot{q}(t) + C\dot{r} \tag{56}
$$

KINETICS

Kinetic energy K includes translation energy and rotational energy regarding each center of mass.

$$
K^{(\alpha)} = \frac{1}{2}\left\{\left(\dot{x}_C^{(\alpha)}\right)^T m^{(\alpha)} \dot{x}_C^{(\alpha)} + \left(\omega^{(\alpha)}\right)^T J_C^{(\alpha)} \omega^{(\alpha)}\right\} \tag{57}
$$

We use this in Hamilton's Principle, reformed as the Principle of Virtual work with all work (conservative and non-conservative) on the right side.

$$
\delta\int_{t_0}^{t_1} K^{(\alpha)}(t)\,dt = -\delta\int_{t_0}^{t_1} W^{(\alpha)}(t)\,dt \tag{58}
$$

Constraint on the variation

We will need to take variations as follows:

$$
\delta\mathbf{r}_C^{(\alpha)} = \mathbf{e}^I \delta x_C^{(\alpha)} \tag{59}
$$

The commutativity of mixed partials readily holds for translational velocity and one obtains the obvious:

$$
\delta\dot{x}_C^{(\alpha)}(t) = \left(\frac{d}{dt}\delta x_C^{(\alpha)}(t)\right) \tag{60}
$$

However, the variation of the angular velocity is restricted in 3D space. This was found by Murakami [10] and independently by Holm [14]. First, we define the following term:

$$
\overleftrightarrow{\delta\pi^{(\alpha)}(t)} = \left(R^{(\alpha)}(t)\right)^T \delta R^{(\alpha)}(t) \tag{61}
$$

Equation (61) does not exist in its unvaried form. It defines the virtual frame-rotation vector $\delta\boldsymbol{\pi}^{(\alpha)}$, in the same way as the angular velocity matrix defined the angular velocity:

$$
\delta\boldsymbol{\pi}^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t)\delta\pi^{(\alpha)}(t) \tag{62}
$$

By ensuring the commutativity of mixed partials (time and variation with regard to the directional derivative of the variation parameter), we arrive at a restriction. We find that the variation of the angular velocity depends on the virtual frame rotation, referred to as restricted variation of virtual angular velocity:

$$
\delta\omega^{(\alpha)} = \delta\dot{\pi} + \overleftrightarrow{\omega^{(\alpha)}}\delta\pi^{(\alpha)} \tag{63}
$$

With the above, we establish the virtual generalized displacments.

$$
\{\delta\tilde{X}(t)\} = \begin{pmatrix} \delta x_C^{(1)}(t) \\ \delta\pi^{(1)}(t) \\ \delta x_C^{(2)}(t) \\ \delta\pi^{(2)}(t) \\ \delta x_C^{(3)}(t) \\ \delta\pi^{(3)}(t) \end{pmatrix}. \tag{63a}
$$

Principle of Virtual Work

Continuing, the coefficient matrix $[B(t)]$ which relates the generalized velocities in Eq.(54), also relates the generalized displacements $\{\delta\tilde{X}(t)\}$ and the essential virtual displacements $\{\delta q(t)\}$:

$$
\{\delta\tilde{X}(t)\} = [B(t)]\{\delta q(t)\} \tag{64}
$$

Proceeding with the virtual work done by the physical forces, where moments and $\overleftrightarrow{\delta\pi^{(\alpha)}}(t)$ are a conjugate pair:

$$
\delta W = \{\delta\tilde{X}(t)\}^T \{F(t)\} \tag{65}
$$

Before continuing, we first state we should use these terms in the Principle of Virtual Work: Moment vs. virtual rotation represent a natural pair. They are conjugate to the moment expressed with the body frame. Moment vs. virtual rotation is a natural pair: Hamilton's principle, which yields Euler's equation. Wittenburg [7] postulated the principle of virtual power to use the weighted form of Euler's equation by the virtual angular velocity.

Continuing, for a simple, first pass analysis, we allow for the following external forces: Wind Force on the blades and fixation force to hold the turbine in place (with cables attached at the turbine center of mass—obviating the need to apply cable moments). Buoyancy, gravity and damping are ignored (all of which are unjustified and excessive—however, as stated this first pass paper is for edification). Finally, after multiplying by the B matrix (Eqn. 4 and 5), we obtain the following expression, in compact form, for the generalized forces.

3 row Forces from cable fixation $CableF^{(1)}$ (66a)
3 row Windforces $WindF^{(2)}$ (66b)

$$F^* = \begin{bmatrix} CableF^{(1)} + WindF^{(2)} \\ \left(\overleftrightarrow{s_c^{(2/1)}}\left(R^{(1)}\right)^T\right) WindF^{(2)} \end{bmatrix} \tag{66c}$$

Equation of motion

By making all the substitutions and carrying out the calculus of variations, one obtains the following results:

Formation of the numerical integration equation

Related work for cranes on ships [9] and ROV motion [9] discusses all terms below, in detail. For now, however, we alluded to the fact that the M matrix consists of alternating mass and moment of inertial in block diagonal form. Below, we state the mass entry for each body.

$$\left[M\right]^{(i)} = \begin{bmatrix} m^{(i)} \underset{3x3}{I} & \underset{3x3}{0} \\ \underset{3x3}{0} & \underset{3x3}{J^{(i)}} \end{bmatrix} \tag{67a}$$

The D matrix allows for the modification of the variation of the angular velocities.

$$\left[D\right]^{(i)} = \begin{bmatrix} \underset{3x3}{0} & \underset{3x3}{0} \\ \underset{3x3}{0} & \underset{3x3}{\overleftrightarrow{\omega^{(i)}}} \end{bmatrix} \tag{67a}$$

With the previous set up, one can construct the minimization (Principle of Virtual Work) and extract the equations of motion in terms of generalized variables. The following equations present definitions, followed by the equation of motion:

$$[M*(t)] \equiv [B(t)]^T [M][B(t)] \tag{68a}$$

$$\left[N*(t)\right] \equiv \left[B(t)\right]^T \left(\left[M(t)\right]\left[\dot{B}(t)\right] + \left[D(t)\right]\left[M\right]\left[B(t)\right]\right) \tag{68b}$$

$$\left\{F*(t)\right\} = \left[B(t)\right]^T \left\{F(t)\right\} \tag{68c}$$

$$\left[T^*\right] = \left(B^T M\dot{C} + B^T DMC\right) \tag{68d}$$

$$\left[M*(t)\right]\left\{\ddot{q}(t)\right\} + \left[N*(t)\right]\left\{\dot{q}(t)\right\} = \left\{F*(t)\right\} - \left[T^*\right]\left\{\dot{r}\right\} \tag{68e}$$

Numerical Integration

For the solution we turn to the Runge Kutta method. However, it is known that that method updates both spatially and temporally. We reformulate the method in terms of

$$\dot{y}(t) = \left(M^*(t)\right)^{-1}\left(F^*(t) - N^*(t)y(t) \; - T^*(t)\dot{r}\right) \tag{69}$$

Thus, in terms of the RK4 constants

$$k_1 = \left(M^*_{t_n}\right)^{-1}\left(F^*_{t_n} - N^*_{t_n} y_{t_n} - T^*_{t_n}\dot{r}\right) \tag{70a}$$

$$k_2 = \left(M^*_{t_n + \frac{dt}{2}}\right)^{-1}\left(F^*_{t_n + \frac{dt}{2}} - N^*_{t_n + \frac{dt}{2}}\left(y_{t_n} + \frac{k_1}{2}dt\right) - T^*_{t_n + \frac{dt}{2}}\dot{r}\right) \tag{70b}$$

$$k_3 = \left(M^*_{t_n + \frac{dt}{2}}\right)^{-1}\left(F^*_{t_n + \frac{dt}{2}} - N^*_{t_n + \frac{dt}{2}}\left(y_{t_n} + \frac{k_2}{2}dt\right) - T^*_{t_n + \frac{dt}{2}}\dot{r}\right) \tag{70c}$$

$$k_4 = \left(M^*_{t_n + dt}\right)^{-1}\left(F^*_{t_n + dt} - N^*_{t_n + dt}\left(y_{t_n} + k_3 dt\right) - T^*_{t_n + dt}\dot{r}\right) \tag{70d}$$

$$y_{n+1} = y_n + \frac{dt}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{70e}$$

However, in between each spatial update, we must first update the rotation matrix for the turbine, which is presented next.

Updating the rotation matrix

The rotation matrices for the two arms are standard, due to the derivation from revolute joints. However, we must know the rotation matrix of the turbine for several reasons. First, it is required in the updating of the B matrix. Second, it is required to apply the hydrodynamic forces, which will be included in a later iteration. Finally, we need it for visualization.

We must reconstruct the rotation matrix of the wind turbine from the angular velocity. We must compute the rotation matrix $R^{(1)}(t)$ by solving the following equation:

$$\dot{R}^{(1)}(t) = R^{(1)}\overleftrightarrow{\omega^{(1)}(t)} \tag{71}$$

Let us assume for a moment that $\overleftrightarrow{\omega^{(1)}(t)}$ is constant and is designated as $\vec{\omega}_0$. Then, with initial value $R(0)$, the solution is:

$$R^{(1)}(t) = R(0)\exp(t\overleftrightarrow{\omega_0}) \tag{72}$$

There does exist a known analytical, closed form solution to Eq. (71), but only for cases in which $\vec{\omega}_0$ is constant. It derives from the Cayley Hamilton Theorem and is known as the Rodriguez' rotation formula to obtain a series expansion of the exponential of a matrix.

$$R(t+\Delta t) = R(t)\begin{pmatrix} I_d + \dfrac{\overleftrightarrow{\omega_0(t+\Delta t)}}{\left\|\omega(t+\Delta t)\right\|}\sin(t\left\|\omega_0(t+\Delta t)\right\|) \\ \left(\dfrac{\overleftrightarrow{\omega_0(t+\Delta t)}}{\left\|\omega(t+\Delta t)\right\|}\right)^2 (1-\cos(t\left\|\omega_0(t+\Delta t)\right\|)) \end{pmatrix} \tag{73}$$

The difficulty is that we do not have a constant angular velocity matrix. However, we can approximate its constancy in each time step of the numerical integration. In principle, one averages this over two-time steps using a central difference approximation. However, for ease of first pass coding, we violate this rule by assuming a constant value at the start of each time step:

$$\omega(t+\Delta t/2) \equiv (\omega(t)+\omega(t+\Delta t/2))/2 \tag{74}$$

To compile the system of variables, one gathers the cartesian and generalized coordinates:

INPUT DATA

Trubine mass	$7.5x10^6\,kg$
Hub height	$200m$
Tower Radius	$10m$
Rotor mass	$53,000Kg$
Rotor radius	$60m$
Rotor location from turbine CM	$90m$
Correcting rotor mass	$530,000Kg$
Correcting rotor radius	$10m$
Correcting rotor location from CM	0.0
Tip Speed Ratio	0.7
Wind density	$1.225kg/m^3$
Wind Speed (direction-3): U	$10m/\sec$
Moment of Inertia of Blades	$(0.50 \quad 0.50 \quad 0.90)10^9\,kg-m$
Moment of Inertia of Turbine	$(34.0 \quad .075 \quad 35.0)10^9\,kg-m$
Moment of Inertia of Rotor	$(0.01 \quad 0.01 \quad 0.02)10^9\,kg-m$

We used the following equation for the angular velocity of the windmill blades, due to the wind speed.

$$\omega_{blade\_tip} = TSR*U/r \tag{75}$$

We used the following equation for the wind force, due to the wind speed.

$$\omega_{blade\_tip} = C_D*0.5*Area_{Swept\_rotor}*U^2 \tag{76}$$

RESULTS

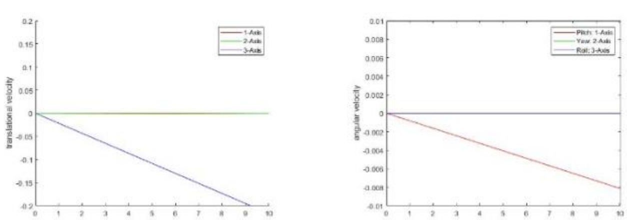
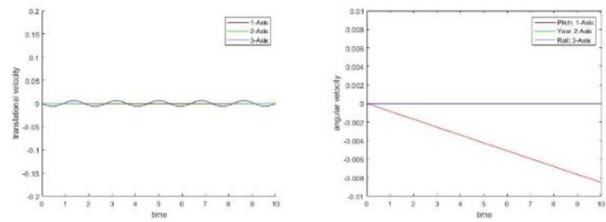


Figure 2: Translation rate (a), Angular Rate (b)

Figure 2 presents translatoinal rates (a) and rotational rates (b) for blades locked, correcting rotor locked, turbine free to translate. Results show turbine translation and pitch.



Furthermore in both cases, a negative linear change in the translational rate in third direction and a the rotation about the first axis. In both cases, the linear response is expected.

Figure 3: Translation rate (a), Angular Rate (b)

Figure 3 presents the results for: blades locked, correcting rotor locked, turbine fixed by cables. Translation minimized, pitch remains. Articially high tensile strength in cable: will enable future work to analyze fixation loads.

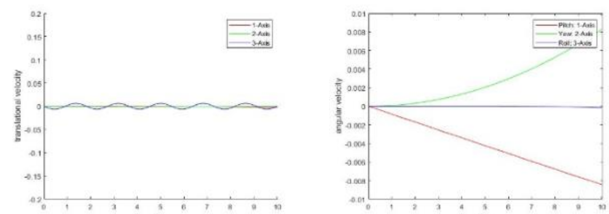


Figure 4: Translation rate (a), Angular Rate (b)

Figure 4 presents the results for rotating blades, correcting rotor locked. Fixation cables active. Blades induce Yaw.

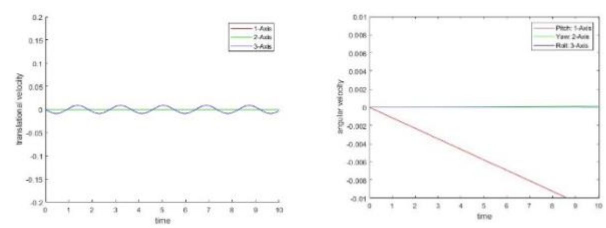


Figure 5: Translation rate (a), Angular Rate (b)

Figure 5 presents the results for rotating blades. Correcting rotor active and fixation cables active. Pitch remains, Yaw eliminated. For the sake of the correcting rotor, the energy stored, considering the mass, and geometry was:

$E_{Stored} = 2.2x10^6\,Joules$

WebGL

WebGL (Web Graphics Library) is a JavaScript interface for rendering interactive 2D and 3D computer graphics. WebGL is compatible with most of the major web browsers such as Chrome, Firefox, Safari, and Opera. In addition, it is free of cost and can be used without the need for plugins. As with similar projects, the goal was to have a reader operate the system on their cell phones. However, the complexity of the Runge-Kutta method and the Rodriguez formula necessitated Matlab over Javascript to complete this first phase task. However, ThreeJS, client side scripting language, cannot open files. Thus, we forged bindings between Matlab and Javascript. The user is unable to interact at this time, however, we show the final results that would be seen. We will address this in the next round. For now, we present the pitch and yaw of the turbine, without the correcting rotor. Figure 6 and 7 demonstrate the pitch and yaw in accordance with Figure 4.

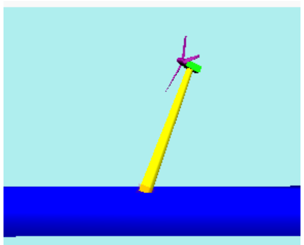


Figure 6: Side View (pitch)

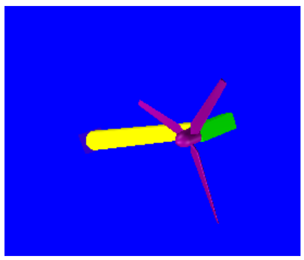


Figure 7: Top view (yaw)

DISCUSSION

Indeed, these results are short of a full paper, justifying, we hope, inclusion in the IMECE conference. We hope to solicit advice and guidance on the continuation of the work. For now, one goal of this work was to demonstrate pedagogical aspects of this method.

Furthermore, while the results are fairly obvious, and expected, the general appraoch taken here will allows us to model offsets and additional links on the structure.

We gave shown that the correcting rotor can stabilize the turbine and also store energy for later use. We have not studied the ideal location of the correcting rotor. We have not accounted for the added mass and viscous effects of the water. We did not account for buoyancy or gravity. Despite all these limitations, the work here satisfies the proof of concept—the primary goal for this first phase.

The results invite the question: could a stablizing rotor, facing in another direction, reduce the pitch while ensuring there is no yaw? In fact, can multiple disks be placed. We emphasize one more issue. This work was conducted by undergraduate students conducting their senior project. In itself, for the sake of edification, this demonstrates the simplicity of this method in dynamics.

FUTURE WORK

The team has secured funding to take the next steps. We will account for wave radiation, and mooring lines not placed at the turbine center of mass. The team will build a small scale model in the university wave tank to conduct experimental validation. We will incorporate a A.I. technology to auto-place the correting rotors (a family of them facing in different directions, to eliminate pitch and store energy).

The project has sparked an interest in the undergraduate students, showing them that the knowledge gained within the field of dynamics can be so relevant in today's job market. As an experimental process, they want to come up with a way of efficiently accounting for the effects of gravity and buoyancy.

REFERENCES

[1] H. Bendix, "forskning.no," 2016. [Online]. Available: https://forskning.no/alternativ-energi-havforskning-spor-en-forsker/spor-en-forsker-hvor-blir-det-av-bolgeenergien/378374. [Accessed: Jan. 11, 2019]

[2] N. Regjering, "Regjeringen.no," 2014. [Online]. Available: https://www.regjeringen.no/no/tema/klima-og-miljo/klima/innsiktsartikler-klima/gront-skifte/id2076832/. [Accessed: Jan. 11, 2019].

[8] T. Impelluso, "The moving frame method in dynamics: Reforming a curriculum and assessment," International Journal of Mechanical Engineering Education, pp. 158-191, 2018.

[3] Goldin, A., 2004, "Autonomous Gyroscopic Ocean-Wave Powered Generator: Invention of a New Energy Conversion Technology, "www.siemensfoundation.org/en/competition/2004-winners/aaron_goldin.html.

[4] Cartan, E., 1986, *On Manifolds with an Affine Connection and the Theory of General Relativity*, translated by A. Magnon and A. Ashtekar, Napoli, Italy, Bibiliopolis.

[5] H. Murakami, "A moving frame method for multibody dynamics using SE(3)," ASME 2015 International Mechanical Engineering Congress & Exposition, paper IMECE2015-51192., 2015.

[6] Holm, D. D., 2008, Geometric Mechanics, Part II: Rotating, Translating and Rolling, World Scientific, NJ.

[7] Wittenburg J, (2008) Dynamics of Multibody Systems, 2nd ed., Springer.

[8] Alexander Jacobsen Jardim, P., Tore Rein, J., Haveland, Ø., Rykkje, T. R., and Impelluso, T. J. (February 18, 2019). "Modeling Crane-Induced Ship Motion Using the Moving Frame Method." ASME. *J. Offshore Mech. Arct. Eng*. Oct. 2019; 141(5): 051103. https://doi.org/10.1115/1.4042536

[9] Austefjord, KO, Hestvik, MO, Larsen, LS, & Impelluso, TJ. "Modelling Subsea ROV Motion Using the Moving Frame Method." *Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition. Volume 4A: Dynamics, Vibration, and Control*. Pittsburgh, Pennsylvania, USA. November 9–15, 2018.. ASME. https://doi.org/10.1115/IMECE2018-86191