

Wind Energ. Sci. Discuss., referee comment RC2
<https://doi.org/10.5194/wes-2021-46-RC2>, 2021
© Author(s) 2021. This work is distributed under
the Creative Commons Attribution 4.0 License.

Comment on wes-2021-46

Anonymous Referee #2

Referee comment on "A symbolic framework to obtain mid-fidelity models of flexible multibody systems with application to horizontal-axis wind turbines" by Emmanuel Branlard and Jens Geisler, Wind Energ. Sci. Discuss.,
<https://doi.org/10.5194/wes-2021-46-RC2>, 2021

As the title states, this paper presents a symbolic framework for flexible multibody systems applied to HAWTs. Overall, the approach is clearly presented and shows excellent agreement to OpenFAST. Moreover, in addition to the paper, a companion open-source Python implementation is provided so readers can repeat the analysis and apply the approach to general systems. In contrast to the other reviewers comments, I would strongly encourage publication of this article. Novel items include a (1) clear and concise presentation of flexible multibody dynamics expressed in Kane's formulation, applicable to both nonlinear and linearized systems, in symbolic form and (2) an open-source Python implementation based on SymPy and PyDy. While the formulation will not replace widely used structural dynamics software for HAWTs such as OpenFAST, Bladed, HAWC2, and FLEX5, that is not the intent, as stated. While the approach could be used to simplify or enhance the ElastoDyn module of OpenFAST, more important applications, as stated, include frequency-domain analysis important in preliminary design, stability analysis, controls design, physics-based digital twins, etc. The approach does account for centrifugal stiffening and gravitational destiffening.

Please find a few specific comments below:

Section 2: The ElastoDyn module of OpenFAST uses the concept of "partial loads", which is extension of the "partial velocity" approach used by Kane's method. Partial loads simplify the formulation of the equations of motion into terms that are useful for load output calculations once the equations of motion are solved. I don't see this concept mentioned, but perhaps it would be an interesting extension of the described approach, if possible?

Page 1, Line 4: Change "facilite" to "facilitates".

Page 1, Line 6: Change "application" to "applications".

Page 3, Line 79: "masscenter" should be two words.

Page 4, Line 93: Change "is" to "are".

Page 5, Line 129: Change "reminder" to "remainder".

Page 6, Line 141: It would be useful to mention the contribution of the inertia term ($M \cdot \ddot{q}$) on the stiffness matrix, which can impact the linearized solution if the model is not in steady state, where otherwise $\ddot{q} = 0$).

Page 6, Line 158: "Shape functions of any order" are mentioned. Is this referring to

shape functions expressed as polynomials? Are other forms of shape functions permitted in the Python implementation?

Page 20, Line 467: Change "of" to "or".

Page 22, Line 508: Add "be" after "to".