Development of Morphing Wing Simulation Tool Using Finite Element Modeling

Staci Parker*

Texas A&M University, College Station, TX, 77843-3141

ABSTRACT

Birds have been inspiring man to flight for years and most recently have inspired a desire to enable an airplane’s wings to morph during flight. Birds and fish use morphing to control their flight path, and birds are also capable of using perching to land quickly in a small space. In accordance with the desire to mimic their movement in planes Texas A&M University has been researching how to control a morphing plane and this research was done in order to know structural impact. A MATLAB finite element model was developed to calculate the structural response of a morphing wing. The morphing airplane wing is modeled as a cantilevered beam, and this simulation has the ability to change inputs: modulus of elasticity, moment of inertia, area, length, and the grid of the wing. Results presented in the paper show that modeling the airplane wing as a beam causes difficulty and future simulations should look towards a frame analysis.

---

* Undergraduate Research Assistant, Flight Simulation Laboratory, Aerospace Engineering Department. Student Member AIAA. hdumpty@tamu.edu
I. Introduction

Both morphing wings and micro air vehicles have come to the forefront of aerospace research.

The morphing concept came about through one of the oldest motivations for flight – birds. Birds morph their wings in order to fly, fish morph their fins in order to navigate through the water, and one day planes will morph in order to fly more efficiently.

Some of the motivations in researching morphing wings include saving military customers money, increasing fuel efficiency in multiple flight phases, and landing planes through perching. Currently military customers use multiple planes for multiple tasks, but a morphing aircraft would allow the customer to consistently purchase one model of plane for all tasks (this would also enable them to train all pilots on a smaller number of planes). Planes are optimized for one phase of flight, the one they spend the most time in (typically cruise). See Fig. 1 for a comparison of fixed and morphing wing aircraft performance.

Figure 1: Plot comparison of fixed and morphing wing aircraft [1].

If morphing wing technology turned reality planes could change their configuration to generate the best aerodynamic shape for each phase in flight. Birds land by perching. They air brake and stop immediately. Planes need many feet of runway before they may come to a complete stop. If planes were able to land using perching, small planes could be used for surveillance, larger planes could take advantage of air craft carriers, and runways would not be needed for landing in remote portions of the world.

One of the difficulties in developing a morphing wing is material. We may be able to design and control a morphing wing aircraft, but can we develop structures that are able to perform the morphing and take the loading forces while remaining safe and reliable without high maintenance? This simulation tool tries to answer the structural response question through calculating stress and strain in the wing during a particular loading.

Texas A&M’s Aerospace Engineering department has developed a morphing airfoil wind tunnel model (Fig. 2).

Figure 2: Morphing airfoil

Other research in the morphing field is being conducted by Cornell University and NASA Langley. The research looks into using SMA actuators to cause the morphing in wings [6], and NASA Langley also has several ornithopters (flapping wing planes).

Micro Air Vehicles, or MAVs, are small planes with a wing span of less than 12 inches. These small planes are similar to Unmanned Air Vehicles (UAVs) in that they are piloted remotely, but differ drastically in size. Low weight and low air speed contribute to challenges with MAVs. Any small wind gust has a much larger effect on the small planes than it would on a larger, more stable plane. The Air Force, working with the University of Florida, has interests in both UAVs and MAVs. The university has developed a micro air vehicle that may be carried in a tube by soldiers before being put together for short term surveillance work.

The research covered in this paper, in combination with ongoing research at Texas A&M, hopes to combine both the morphing and micro vehicle technologies.

The simulation tool, written in MATLAB, aims to remain general in order to work for a variety of different wing grids. The simulation allows inputting
varying moments of inertia, moduli of elasticity, area, and length.

II. Research Objectives and Plan

The purpose of this research was to develop a FEM for the structural response of a morphing wing, have a simulation evaluate stress and strain in morphing wing, and have a simulation to interface with other aerodynamic simulations. Specifically, to work with active learning and aerodynamic simulations which are being developed at Texas A&M University by Amanda Lampton and Adam Niksche.

Before starting on developing the simulation, a literature search was performed (and continued to be updated throughout the project). A 2D program developed for an undergraduate class was taken as the starting point for this simulation and developed to add new functions and capabilities. The program used a simple input file to validate the code and the simulation will continue to be verified over time.

1. Literature search
2. Basic 2D program outputting displacements developed for AERO 306
3. Increase degrees-of-freedom
4. Increase variable inputs
5. Add loading
6. Add drag
7. Validation
8. Verification

III. Error Analysis

As with any simulation tool there are assumptions made when modeling the morphing wing as a beam. Error also exists in the values inputted for each element depending on how accurately those values are determined. Because the simulation is of a wing and not an entire plane, boundary conditions exist which are not realistic for an entire plane.

IV. Finite Element Model Brief

The simulation models the morphing wing as a 3D cantilevered beam with x, y, and z-axis and six degrees-of-freedom. The degrees-of-freedom may be altered to fit any scenario, but for the simulation to be correct, the interpolation formulas used to calculate a K matrix must be changed as well. The point loads outputted from the aerodynamic simulation developed by Adam Niksche are inputted into the structural simulation’s input file with each point load occurring at a specific node. The simulation reads the input file information and calculates the K and F matrices for each element before combing to form a global matrix. The boundary conditions are applied and the displacements are then calculated. From the displacements, the stress and strain in the beam may be determined.

While developing the simulation, it was determined that the inputs needed to run the code would be difficult to obtain and future simulations would need to address those difficulties in order for the simulation to be applicable to morphing wing structural analysis (see Appendix for example input file). Also, while non-morphing wings may be analyzed in a finite element model, morphing wings would require many different meshes. The addition of multiple meshes for one wing analysis means that inputting the mesh by hand will be too time consuming to be worthwhile and a new method in which the mesh is reconfigured by the program needs to be developed. While these difficulties exist, the finite element model simulation does still calculate the stress and strain in the wing.

V. Conclusions

This paper introduced why morphing is being considered and how the developed simulation will work with other simulations to advance the knowledge of controlling a morphing wing airplane. It was found that this particular model requires inputs of the exact mesh as well as the moments of inertia and modulus of elasticity. While this kind of information may be known or easily calculable for a beam structure, the materials and shape of a morphing wing have complexities which hinder the individual element moments of inertias from being known.

Based upon the results presented in this paper, it is concluded that a finite element model may work better for non-morphing wings than for morphing wings. Inputs required for a beam analysis would be difficult to find while a frame analysis may avoid such challenges, and the output from the aerodynamic code must be inputted into a new input file in order for the structural simulation to run.
VI. Recommendations

In the future a different programming language would shorten the run time of the simulation. The C++ language is a good candidate for this purpose. Also, for this particular application, a frame analysis may work better than a beam analysis considering all the needed inputs.

The final objective of interfacing with an aerodynamic simulation has not been fully reached. More work should allow the simulations to exchange data if desired.

Future work on the research would allow for the simulation to be verified over time and a real wing grid to be run on both the aerodynamic and structural simulations.

Acknowledgments

Thanks to Dr. John Valasek, the faculty advisor on this project; Amanda Lampton, the graduate student advisor on this project; Texas A&M Flight Simulation Laboratory; Cornell University for their sharing of information; Adam Niksche for his aerodynamic simulation; and Doctors Davis and Lagoudas, heads of the Nanotechnology and Material Systems Research Experience for Undergraduates.

This Research Experience for Undergraduates Site is sponsored by the National Science Foundation Grant No. 0453578, the Air Force Office of Scientific Research, U.S. Air Force, Department of Defense and NASA Cooperative Agreement No. NCC1-02038.

References


Appendix

Degrees of Freedom per node:
6
Number of Nodes:
6
Number of Elements:
5
Node #: X: Y: Z:
1   0 0 0
2   1 0 0
3   2 0 0
4   3 0 0
5   4 0 0
6   5 0 0
Element #: Youngs: Ixx: Iyy: Izz: A: Connectivity:
1     1E+07 1 1 1 1 1 2
2     1E+07 1 1 1 1 2 3
3     1E+07 1 1 1 1 3 4
4     1E+07 1 1 1 1 4 5
5     1E+07 1 1 1 1 5 6
Number of Constraints:
6
dof specified to be zero:
1 2 3 4 5 6
Number of Point loads:
10
Load#: dof: load:
1   9   2
2   15  2
3   21  2
4   27  2
5   33  2
6   11  1
7   17  1
8   23  1
9   29  1
10  35  1