Comprehensive Nonlinear Simulation with Robust Autonomous Control

G. Douglas Baldwin*
Baldwin Technology Company, LLC
Port Washington
New York 11050

Abstract

A mostly free, open source software tool suite for comprehensive analysis of a complex system's physical behavior is presented. This sophisticated tool suite, which captures the system-wide effects of multi-body dynamics, viscoelastic materials, and unsteady aerodynamics, is comprised of mostly industrial quality software and was developed specifically to advance the understanding of the Mono Tiltrotor (MTR) design. Virtually any rotorcraft design can be modeled and piloted in this computer simulation environment, and the piloted trajectories are then learned for autonomous control. The controller can robustly reproduce the learned trajectory for variants of the base model to include additional elements for a higher fidelity simulation and changed elements for trade studies. The model can be validated by applying the controller to a physical test aircraft and comparing its real flight dynamic behavior with the simulation predictions.

Introduction

Rotorcraft require a comprehensive analytical treatment to capture the combined effects of multi-body dynamics, viscoelastic materials, and unsteady aerodynamics. Closed source comprehensive analytical methods are commercially available, and these codes could be modified for unusual design features outside of their legacy rotorcraft design domain, but the details of such code changes would be proprietary to the software vendor. Open source software allows anyone to examine and improve as needed the analytical method.

The integrated suite presented in this paper exceeds the capabilities of traditional proprietary comprehensive codes in the following respects: using a graphical interface, the design engineer can interactively construct and pilot a visually and physically accurate three-dimensional model of virtually any configuration; and, piloted trajectories are learned and repeated for the engineer, facilitating examination of the effects of design changes for a particular trajectory.

The underlying analytical engine is an object oriented, open source, multibody dynamic simulation code with blade element aerodynamics and rotor inflow models. The simulator offers a menu of fundamental engineering elements including inertial bodies, viscoelastic beams and joints, and aerodynamic bodies and wings. A proprietary free-vortex wake software module enhances the multibody simulation by producing a coupled wake flow field that acts on all aerodynamic elements.

Motivation

Conceptual studies performed in 2004 concluded that the Mono Tiltrotor (MTR) design is half the size, 1/3rd the gross weight, and 1/3rd the fuel burn of legacy rotorcraft configurations for long range missions [1, 2]. Recent successful flight tests have shown the design to be technically feasible at a very small scale [3, 4], and preliminary designs at a 9400 pound gross weight scale have reinforced the conclusion of breakthrough performance [5, 6, 7]. Concept of operations (CONOPs) animated renderings of dimensionally accurate and kinematically correct full scale models illustrating the dynamic behavior observed with the small scale flight models have been produced [8].

At this point in the MTR's maturation, an irrefutable prediction of full scale dynamic behavior is desirable. Traditional comprehensive rotorcraft codes can handle the design's coaxial proprotor, but have difficulty with two other fundamental features of the MTR design. An enhanced comprehensive rotorcraft code that can handle aerodynamically deployed wing panels and a pitch axis suspended aerodynamic pod is needed. Rather than contracting with and becoming dependent upon a proprietary vendor to create specialized codes, it was decided to select and integrate general purpose tools.

A further motivation was to identify and integrate an autonomous controller that is capable of duplicating a learned trajectory both in the simulator and in flight test hardware, facilitating validation of simulation predictions against empirical observations.

---

* Managing Director. **doug@baldwintechnology.com**
Presented at the American Helicopter Society International, Specialists' Meeting, "Technologies to Support the Next Generation of Vertical Lift Aircraft and Beyond," Fort Worth, TX, 23-25 February, 2011.

Research Methodology

Blender and MBDyn are two open source tools that were integrated and reported in a prior conference paper [9]. Blender provides a three-dimensional graphical interface for creating, animating, and rendering artistic objects [10]. MBDyn from Politecnico di Milano is a validated multibody dynamic simulation tool with a script based model specification language and tabular numerical output [11]. The prior paper describes in detail the Blender and MBDyn open source integration software, wherein Blender objects take the attributes of MBDyn elements. The result is a graphical point-and-click environment for creating three-dimensional multibody models, executing the models either in batch or in realtime with joystick controlled pilot-in-the-loop, and rendering the dynamic model behavior in a three-dimensional artistic scene. MBDyn has been validated against rotorcraft wind tunnel data [12], and over 95% of MBDyn element types have been implemented so far in the Blender environment, including all element types needed for constructing a rotorcraft model.

This paper focuses on the further integration of an autonomous controller, and touches upon the integration of a free-vortex wake method for coupled aerodynamic simulation.

Autonomous Control

Apprenticeship Learning for autonomous helicopter control is a technique developed by Dr. Pieter Abbeel as a doctoral candidate at Stanford University [13] and further developed by him as an associate professor at the University of California at Berkeley [14]. His approach can be contrasted with classic linear control theory in the following ways.

Helicopters are nonlinear systems, but are classically approximated as linear systems about trimmed flight conditions. Rarely do these approximations comply with handling quality standards, so control feedback loops are engineered to transform the linear approximation into a target linear system that is compliant. These control feedback loops are space and time independent. In order to introduce space and time, an outer control loop is added that operates on the target linear system to achieve a specified trajectory.

In contrast, apprenticeship learning performs a direct mapping of a target trajectory to non-linear control inputs. The technique was first demonstrated on a helicopter performing extreme aerobatic maneuvers at the Stanford flight test range [15]. Learning can occur with just one flight demonstration, wherein the demonstrated trajectory is replicated. Learning can also occur for multiple repeated demonstrations that each approximate a desired trajectory, and the software will find the hidden desired trajectory. The apprenticeship learning software was ported to and integrated with the Blender and MBDyn suite.

Learning from a single demonstration.

Whether the aircraft is real or simulated, the engineering steps to achieving autonomy for a single demonstration are as follows:

1) Fly the aircraft with joystick, pilot-in-the-loop control to demonstrate a trajectory, while simultaneously logging the following data at 20Hz: position, velocity, acceleration, orientation, angular velocity, angular acceleration, and joystick inputs.

2) Model the aircraft's non-gravitational accelerations for each of its six degrees of freedom as a simple weighted sum of features (e.g. acceleration = c_1 + c_2*velocity + c_3*control; wherein c_x are unknown weights).

3) Regress the data of step 1 to calculate the best fit weights for the equations of step 2.

4) Compute and tabulate the differences between the calculated accelerations using the weighted equations of step 2 against the logged accelerations from step 1.

5) While re-flying the aircraft, use differential dynamic programming at 20Hz with a receding horizon of 50 timesteps (2.5 seconds) to predict the best control inputs for replicating the trajectory of step 1, based on the weighted equations of steps 2 and 3 and the tabulated bias values of step 4.

The above five steps are all automated in Blender and MBDyn so that the engineer simply pushes a button to switch from manual flight control demonstration to automated flight control of the learned trajectory. The core process is step 5 above, which is implemented in about 300 lines of open source C++ code.

The controller is theoretically more general than the above explanation indicates. It works generally for a system with measurable features (step 1 above) and approximate mathematical relationships (step 2 above). The specific mathematical relationships encoded within the Blender and MBDyn autonomous controller are as follows:

$$\ddot{x}=c_0\cdot\dot{x}$$
$$\ddot{y}=c_1+c_2\cdot\dot{y}$$
$$\ddot{z}=c_3+c_4\cdot\dot{z}+c_5\cdot collective$$
$$\dot{p}=c_6+c_7\cdot p+c_8\cdot lateral$$
$$\dot{q}=c_9+c_{10}\cdot q+c_{11}\cdot fore-aft$$
$$\dot{r}=c_{12}+c_{13}\cdot r+c_{14}\cdot pedal$$

...wherein the xyz-aircraft reference frame's positive directional orientation is forward, right wing, and down.

If the demonstration aircraft has a stability augmentation system, control inputs can be recorded either before or after stability augmentation. For real remote control aircraft, Dr. Abbeel usually records control inputs before any stability augmentation. Blender and MBDyn records control inputs after any stability augmentation. Apprenticeship Learning works in either case, with autonomous control inputs inserted at the location where they had been recorded.

Learning from multiple demonstrations.

When multiple repeated demonstrations each approximate a hidden desired trajectory, the steps to achieving autonomous control for the hidden trajectory are as listed above, with one additional step. Between steps 3 and 4, the multiple demonstrations are time warped and aligned using a least cost algorithm, then interpolated using a Kalman filter with smoothing. The time warping and interpolation process is automated in Blender and MBDyn, so that the engineer simply selects from a list of all prior demonstrations those to be interpolated.

Robust control

An autonomous controller learned from either a single demonstration or multiple demonstrations can control more than just the original MBDyn simulation model. Additional MBDyn elements can be included for increased model fidelity and complexity, and the controller can then be executed against this new model as a long running batch application which replicates the original trajectory. In the event that a series of incremental changes to the MBDyn model ultimately result in new system dynamics outside the bounds of the original learned controller, the new demonstration trajectories of intermediate models can be interpolated to produce suitable intermediate controllers.

Free-vortex wake method.

All the above integrated open source code provides a comprehensive suite for rotorcraft analysis, with one significant omission; MBDyn's aerodynamic elements do not generate a three-dimensional wake, and so aerodynamic coupling between aerodynamic elements is missing. An optional proprietary software module provides this missing capability.

The CHARM, (Comprehensive Hierarchical Aeromechanics Rotorcraft Model), Wake-Panel Module from Continuum Dynamics Incorporated [16] has been integrated into Blender and MBDyn. CHARM provides wake- and surface-induced velocities using a full-span, Constant Vorticity Contour (CVC), free-vortex rotor wake model. The module can also analyze lifting or non-lifting surfaces in the vicinity of the rotor (e.g., fuselage, wings, ground planes) in a coupled wake/body solution. All fixed topology MBDyn aerodynamic elements can contribute to the CHARM generated flow field solution, and all MBDyn aerodynamic elements are affected by this flow field solution.

Results and Discussion

The resulting comprehensive analytical suite is best appreciated by viewing demonstration videos and tutorial videos, by installing the suite on a Linux PC workstation, and by working through the example Blender and MBDyn models that accompany this paper. All videos and all software except for the CHARM module can be freely downloaded from the Internet [17, 18, 19, 20, 21].

The tutorials range from how to install the software and test the installation by creating a simple model that releases an inertial body into a gravity field, to the development of rather complex models such as a coaxial helicopter with dual swash plates, pitch links, and joystick pilot-in-the-loop control. The video tutorials are accompanied by a Blender data file containing MBDyn tutorial models to facilitate rapid learning, including models of an aeroelastic beam and of trimmed helicopter flight.

As will become clear from working with this software suite, it is ripe with potential. For example, the Apprenticeship Learning technique can be studied and further developed on a low-cost Linux workstation, without any need for an expensive flight test facility and expert pilots. As a result, a significant cost and logistics barrier to the proliferation and practical implementation of the Apprenticeship Learning technique has been removed. Furthermore, trajectories learned on the workstation can subsequently be loaded onto a real aircraft for MBDyn model validation. Ultimately, operational trajectories could be flown and refined on the workstation before being performed by real aircraft.

This software suite also helps solve a practical dilemma regarding the advancement of potential breakthrough aircraft configurations such as the MTR. The investment dollars needed to demonstrate a vertical lift aircraft innovation at full scale must be justified, and this usually requires modeling and simulation at full scale using a validated tool suite. The software tool suite presented in this paper enables MBDyn and CHARM model validation against the successful small scale MTR functional flight demonstrator, and the scaling up of this simulation model to larger configurations.

Limitations.

This very general comprehensive simulation suite has the co-requisite limitation that a user can unintentionally create a malformed model. As is true with native MBDyn, malformed models can fail to execute due to over-constraints, under-constraints, or other issues. The combination of Blender with MBDyn mitigates this limitation by automatically generating correct element specifications, rotation matrices, and position vectors from the enhanced 3-dimensional model geometry, and by facilitating interative model construction. A simple assembly of MBDyn elements can be visually and dynamically tested in isolation in one Blender scene, and then copied, pasted, duplicated, and attached to a more complex working MBDyn model in another Blender scene.

Another limitation is MBDyn processing speed which is usually constrained by the fastest available single processor core. Since pilot-in-the-loop control requires near realtime results, a relatively simple MBDyn model is usually needed. This limitation can be mitigated by first learning a demonstration trajectory for a simple MBDyn model, and then employing Apprenticeship Learning for autonomous control to repeat this trajectory on a complex MBDyn model in a long running batch process.

Also, surface contact is not generally offered in the MBDyn menu of fundamental engineering elements. This limitation is irrelevant for airborne flight simulation, but does affect takeoff and landing analysis particularly in regards to alighting gear. Point contact with a plane, however, can be modeled using MBDyn's very general implementation of viscoelastic joints, and an experimental wheel element is also available.

Another significant limitation is that CHARM does not accurately handle bluff body separated flow. This limitation is typical of most analytical aerodynamic codes, and is a practical necessity for near realtime performance. As a result, any MBDyn model features that could produce separated flow need to be identified and addressed outside of this integrated suite. More advanced and computationally efficient tools that correctly handle separated flow are in development by CDI and others [22], and could possibly be integrated with this comprehensive analytical suite in the future.

Finally, Blender executes Python programming language scripts in a single threaded processing environment. This matters because Blender spawns the MBDyn and Apprenticeship Learning processes through Python, and has very little control of these processes once they start. Under normal circumstances, the initiation and termination of these asynchronous processes is transparent to the user, but occasionally the MBDyn and Apprenticeship Learning processes must be manually terminated through a Linux console.

Opportunities.

MBDyn modal element support is not currently but could readily be integrated into this software suite. MBDyn permits topological connections to the nodes of a modal body, and Blender is capable of representing the nodes of modal bodies as the verticies of a mesh. The author has not yet needed nor developed this MBDyn feature, however the MBDyn user community could find it useful.

The most significant CHARM feature yet to be integrated into this analytical suite is its panel method fuselage model. Blender is capable of representing the panels of a fuselage as the faces of a mesh for full implementation of this CHARM feature. This feature is rather important to the author, and will likely be implemented soon.

Blender's fundamental strength is in the visual representation of dynamic three-dimensional objects. Blender has been employed to produce photo-realistic rendered animations of the MTR, and can be extended to present in graphical form the various tabular quantitative products of MBDyn and CHARM.

Conclusions

A mostly free, open source software tool suite for comprehensive analysis of a complex system's physical behavior was presented. This sophisticated tool suite, which captures the system-wide effects of multi-body dynamics, viscoelastic materials, and unsteady aerodynamics, integrates the capabilities of Blender, MBDyn, CHARM, and Apprenticeship Learning for Autonomous Helicopter Control, and was developed specifically to analyze the Mono Tiltrotor (MTR) design. Virtually any rotorcraft design can be modeled and piloted in this computer simulation environment, and the piloted trajectories are then learned for autonomous control. The controller can robustly reproduce the learned trajectory for variants of the base model to include additional elements for a higher fidelity simulation and changed elements for trade studies. The model can be validated by applying the controller to a physical test aircraft and comparing its real flight dynamic behavior with the simulation predictions.

Acknowledgments

This paper was supported by Office of Naval Research (ONR) contract N00014-09-C-0435, “Mono Tiltrotor (MTR) Flight Dynamics and Control.” The author is grateful to Dr. Pieter Abbeel of the University of California at Berkeley for releasing source code and providing integration advice for his Apprenticeship Learning method, and to Dr. Perangelo Masarati of Politecnico di Milano for creating new features in MBDyn and who with the assistance of Mr. Daniel A. Wachspress of Continuum Dynamics, Inc. integrated the CHARM Wake-Panel Module into MBDyn.

References

1. Leishman, J. G., Preator, R., Baldwin, G. D., *Conceptual Design Studies of a Mono Tiltrotor (MTR) Architecture*, U. S. Navy Contract Number: N00014-03-C-0531, 2004. http://www.baldwintechnology.com/FY04_MTR_Conceptual_Design_Report.pdf
2. Preator, R., Leishman, J. G., Baldwin, G. D., "Conceptual Design Studies of a Mono Tiltrotor Architecture", Proceedings of the 60th Annual National Forum of the American Helicopter Society, Baltimore, MD, June 7–10, 2004. http://www.baldwintechnology.com/MTR_AHS04.pdf
3. First full conversion video, October 2, 2010, http://www.baldwintechnology.com/video/conversion.mp4
4. Flight Test Video#1 http://www.youtube.com/watch?v=vBfW4mbrodA
5. Baldwin, G. D., *Mono Tiltrotor (MTR) Concept Evaluation*, U. S. Army Contract Number: W911W6-04-D-0004-0001, RDECOM TR 06-D-40, DTIC Accession Number ADB324612, 2006.
6. Baldwin, G. D., *Mono Tiltrotor Validation Activities*, U. S. Army Contract Number: W911W6-04-D-0004-0002, RDECOM TR 08-D-0069, July 2008.
7. Baldwin, G. D., "Mono Tiltrotor (MTR) Validation Activities", Proceedings of the 64th Annual National Forum of the American Helicopter Society, Montreal, Canada, April 29 – May 1, 2008. http://www.baldwintechnology.com/MTR_AHS_08.pdf
8. Cargo UAS CONOPS http://www.youtube.com/watch?v=kl1_GrCAalE
9. Baldwin, G. D., "Open Source Multibody Aeroelastic Modeling, Simulation, and Video Rendering", Proceedings of the 65th Annual National Forum of the American Helicopter Society, Grapevine, Texas, May 27-29, 2009. http://www.baldwintechnology.com/MTR_AHS_09.pdf
10. http://www.blender.org/
11. http://www.aero.polimi.it/mbdyn/
12. Shen, J., Masarati, P., Roget, B., Piatak, D. J., Nixon, M. W., Singleton, J. D., "Still-Inplane Tiltrotor Aeromechanics Investigation Using Two Multibody Analyses", Multibody Dynamics 2007, ECCOMAS Thematic Conference, Milano, Italy, 25–28 June 2007. http://research.nianet.org/~shenjw/webpage/milan07.pdf
13. Abbeel, "Apprenticeship Learning and Reinforcement Learning with Application to Robotic Control," Stanford University, Department of Computer Science, Aug. 2008. http://ai.stanford.edu/~pabbeel//thesis/thesis.pdf
14. Abbeel, P., Coates, A., and Ng, A. Y., 'Autonomous Helicopter Aerobatics through Apprenticeship Learning', International Journal of Robotics Research (IJRR), 2010. http://www.cs.berkeley.edu/~pabeel/papers/AbbeelCoatesNg_IJRR2010.pdf
15. Autonomous Helicopter: Stanford University AI Lab Website, http://heli.stanford.edu/
16. CHARM Real-Time Wake/Panel Module, http://www.continuum-dynamics.com/pr-smod.html
17. http://www.baldwintechnology.com/blender.html, http://www.baldwintechnology.com/deepdive.html
18. http://www.blender.org/download/get-blender/
19. http://www.aero.polimi.it/mbdyn/download/
20. http://sourceforge.net/projects/blenderandmbdyn/
21. http://www.baldwintechnology.com/blender
22. VorTran-M Wake Module http://www.continuum-dynamics.com/pr-vortran.html