CAMRAD II User Update

CII User Meeting

2 November 2026

Gifu, Japan



Aeronautic Concepts, LLC

- Acquired IP rights to CAMRADII from Johnson Aeronautics in January 2026
  - Partner with Johnson Aeronautics since 2016 to support CAMRADII user training & development
  - Wayne has continued involvement in support & development (partial retirement)
  - End user licensing, maintenance and support pricing remains unchanged
  - Continue to distribute CAMRADII via long time partner AMI Aero, LLC (www.amiaerollc.com)
    - Terrabyte continues to be distributor in Japan

- Partnership between Chris Silva (chris@aeroconcepts.us) & Jeff Sinsay (jeff@aeroconcepts.us)
  - 40+ years of rotorcraft aeromechanics design & analysis experience
  - Participated in development of multiple rotorcraft design software projects (e.g. NDARC, AIDEN, TORC)
  - Provide aeromechanics consulting in addition to CAMRADII support & development

Trends in Rotorcraft Aeromechanics

- Urban Air Mobility
  - Focus on multirotor configurations with hingeless and propeller-like hubs (high relative stiffness)
  - RPM control, 2-bladed rotors
  - Significant rotor-rotor, rotor-wing and rotor-fuselage aerodynamic interaction
  - Increased consideration of acoustic performance on design & operation
- Small UAS
  - Low Reynolds Number
  - RPM control, simple hubs
- Digital Engineering
  - Regular use of CFD/CSD coupling to analyze rotor loads and performance
  - Integrated design & optimization frameworks for new rotor development
  - GPUs bring high-performance computing capability to workstation class computers
  - Increasing demand for digital modeling in all-phases of design & development

Recent Published CAMRADII Usage

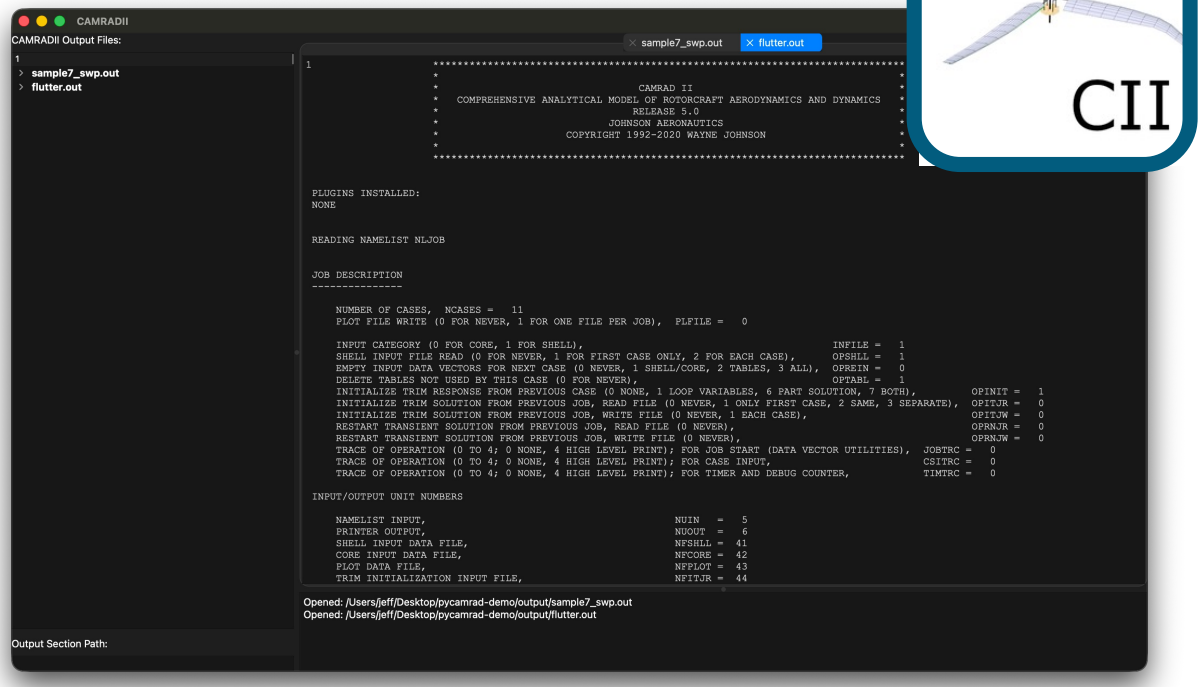
- Kreshock, et. al., "Hingeless Rotor Whirl Flutter Stability Predictions for TRAST," VFS Forum, 2026.
- Park, et. al., "Aerodynamic Optimization for Rotor Blade with Structural Design in Hover and Forward Flight," VFS Forum 2025.
- Ahn et. al., "Analysis of an electric-propulsion gimballed prop-rotor by CAMRADII," European Rotorcraft Forum, 2024.
- Wright et. al., "Comprehensive Analysis Aeroelastic Stability Predictions for the Mars Sample Recovery Helicopters and Ingenuity," VFS Decennial Aeromechanics Conference, 2024.
- Kottapalli, Silva, Boyd, "Applicability of Fly Neighborly Noise Recommendations to UAM Quadrotors Undergoing Steady Maneuvers," VFS Decennial Aeromechanics Conference, 2024.
- Wright, "Aeroeleastic Analysis of an Urban Air Mobility Rotor in Forward Edgewise Flight," VFS Aeromechanics Technical Meeting, 2022.

Future CII Development Focus

- Core CAMRADII technologies are stable
  - Computationally efficient for broad class of aeromechanics problems
  - Leverage high-fidelity aerodynamics through coupling (VVPM, URANS, etc)
- Enhancing user productivity
  - Integrated graphical environment for running jobs and interpreting results
  - Standardized, native python API to support toolchain integration (including CFD coupling)
  - Helper utilities to aide beam property development and airfoil table creation
- Modern computing architectures
  - GPU computing
  - Parallelization (across cases and within cases)

New CAMRADII User Interfaces

- Developing python package pycamrad and associated native GUI application
- Hope to have beta version for interested users soon
- pycamrad will enable python script based interactions
- GUI to provide integrated user workflow from beginning to end

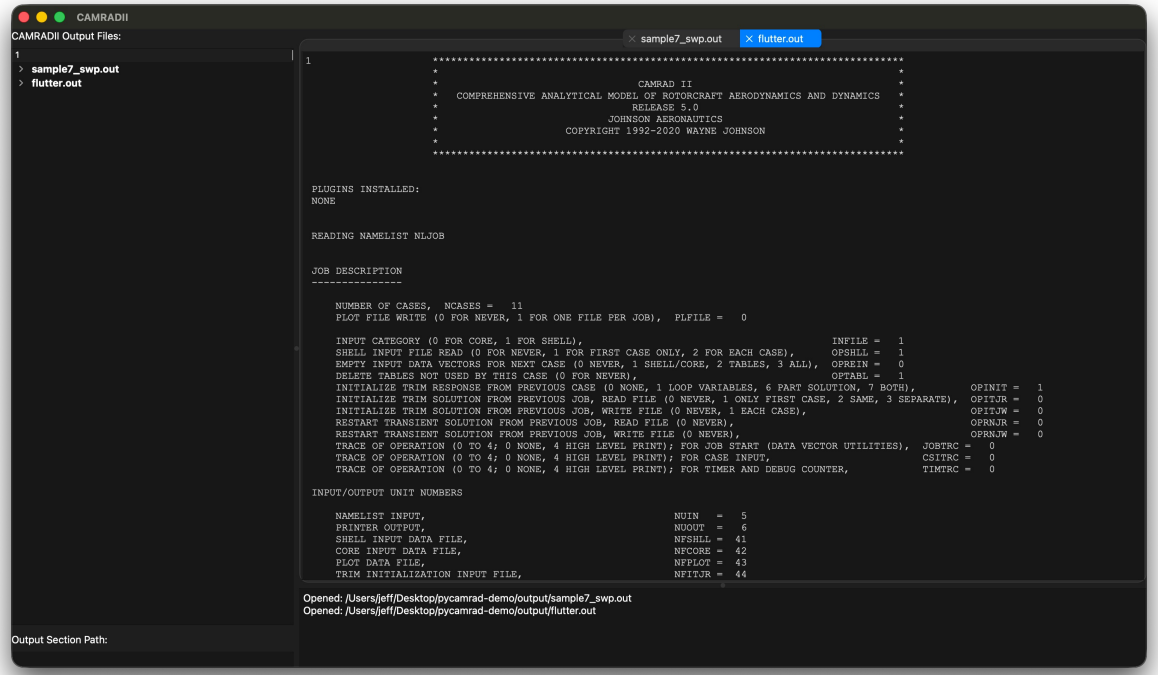


GUI Feature List

- Tree based output viewer
- Documentation viewer
- Plot data visualizer
- Import/export job structure
- Generate airfoil table and shell input
- Input visualization
  - Plot blade input parameters
  - Visualize 3D model
  - Core model diagram
- Input generation utilities
  - Beam modes
  - Airfoil 2D table generation utilities
- Analysis case execution
  - Case setup
  - Parameter sweeping
  - Case monitoring
  - Multithread parallelization
- Output visualization
  - Wake geometry
  - Solution movies (wake, forces, motion)
- Matching capability in pycamrad

*Currently have beta capability implemented

GUI Demo


CAMRADII
CAMRADII Output Files:
sample7_swp.out
flutter.out
sample7_swp.out
flutter.out
Output Section Path:
Opened: /Users/jeff/Desktop/pycamrad-demo/output/sample7_swp.out
Opened: /Users/jeff/Desktop/pycamrad-demo/output/flutter.out
CAMRAD II
COMPREHENSIVE ANALYTICAL MODEL OF ROTORCRAFT AERODYNAMICS AND DYNAMICS
RELEASE 5.0
JOHNSON AERONAUTICS
COPYRIGHT 1992-2020 WAYNE JOHNSON
PLUGINS INSTALLED:
NONE
READING NAMELIST NLJOB
JOB DESCRIPTION
NUMBER OF CASES, NCASES = 11
PLOT FILE WRITE (0 FOR NEVER, 1 FOR ONE FILE PER JOB), PLFILE = 0
INPUT CATEGORY (0 FOR CORE, 1 FOR SHELL), INFILE = 1
SHELL INPUT FILE READ (0 FOR NEVER, 1 FOR FIRST CASE ONLY, 2 FOR EACH CASE), OPSHLL = 1
EMPTY INPUT DATA VECTORS FOR NEXT CASE (0 NEVER, 1 SHELL/CORE, 2 TABLES, 3 ALL), OPREIN = 0
DELETE TABLES NOT USED BY THIS CASE (0 FOR NEVER), OPTABL = 1
INITIALIZE TRIM RESPONSE FROM PREVIOUS CASE (0 NONE, 1 LOOP VARIABLES, 6 PART SOLUTION, 7 BOTH), OPINIT = 1
INITIALIZE TRIM SOLUTION FROM PREVIOUS JOB, READ FILE (0 NEVER, 1 ONLY FIRST CASE, 2 SAME, 3 SEPARATE), OPITJR = 0
INITIALIZE TRIM SOLUTION FROM PREVIOUS JOB, WRITE FILE (0 NEVER, 1 EACH CASE), OPITJW = 0
RESTART TRANSIENT SOLUTION FROM PREVIOUS JOB, READ FILE (0 NEVER), OPRNJR = 0
RESTART TRANSIENT SOLUTION FROM PREVIOUS JOB, WRITE FILE (0 NEVER), OPRNJW = 0
TRACE OF OPERATION (0 TO 4; 0 NONE, 4 HIGH LEVEL PRINT); FOR JOB START (DATA VECTOR UTILITIES), JOBTRC = 0
TRACE OF OPERATION (0 TO 4; 0 NONE, 4 HIGH LEVEL PRINT); FOR CASE INPUT, CSITRC = 0
TRACE OF OPERATION (0 TO 4; 0 NONE, 4 HIGH LEVEL PRINT); FOR TIMER AND DEBUG COUNTER, TIMTRC = 0
INPUT/OUTPUT UNIT NUMBERS
NAMELIST INPUT, NUIN = 5
PRINTER OUTPUT, NUOUT = 6
SHELL INPUT DATA FILE, NFSHLL = 41
CORE INPUT DATA FILE, NFCORE = 42
PLOT DATA FILE, NFPLOT = 43
TRIM INITIALIZATION INPUT FILE, NFITJR = 44

pycamrad Demo

CamradOutput(filename): loads a CAMRADII output file into memory, cases are indexed like a list

For each individual case the following convenience functions are provided:

rotorcraft_perf(): returns the case's rotorcraft performance section in trim as a nested dictionary
rotor_perf(rotor_id): returns the case's rotor performance section in trim as a nested dictionary
part_output(resource_path): returns the output of a part in trim as a nested dictionary
extract_eigenvalues(): returns the eigenvalues from a case's flutter section
extract_modes(): returns the mode set data from a cases's flutter section

To get a list of parameter across all cases in a job (convenient for plotting):
get_param_from_job(job,resource_path)

GUI Feature List

- Tree based output viewer
- Documentation viewer
- Plot data visualizer
- Import/export job structure
- Generate airfoil table and shell input
- Input visualization
  - Plot blade input parameters
  - Visualize 3D model
  - Core model diagram
- Input generation utilities
  - Beam modes
  - Airfoil 2D table generation utilities
- Analysis case execution
  - Case setup
  - Parameter sweeping
  - Case monitoring
  - Multithread parallelization
- Output visualization
  - Wake geometry
  - Solution movies (wake, forces, motion)
- Matching capability in pycamrad

User Requests?