



AFRL-RQ-WP-TR-2013-0031

**STRUCTURAL TECHNOLOGY EVALUATION AND
ANALYSIS PROGRAM (STEAP)**

**Delivery Order 0049: Computational Prototyping of Micro Air
Vehicles**

Jeff Scott

General Dynamics Information Technology

Gary Clayman

Data Science Automation

JANUARY 2013

Final Report

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
AEROSPACE SYSTEMS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the USAF 88th Air Base Wing (88 ABW) Public Affairs Office (PAO) and is available to the general public, including foreign nationals.

Copies may be obtained from the Defense Technical Information Center (DTIC)
(<http://www.dtic.mil>).

AFRL-RQ-WP-TR-2013-0031 HAS BEEN REVIEWED AND IS APPROVED FOR
PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

PHILIP S. BERAN, Tech Advisor
Design and Analysis Branch
Aerospace Vehicles Division

//Signature//

THOMAS CO, Chief
Design and Analysis Branch
Aerospace Vehicles Division

//Signature//

FRANK WITZEMAN, Chief
Aerospace Vehicles Division
Aerospace Systems Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show “//Signature//” stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YY) January 2013		2. REPORT TYPE Final		3. DATES COVERED (From - To) 30 October 2008 – 21 December 2012	
4. TITLE AND SUBTITLE STRUCTURAL TECHNOLOGY EVALUATION AND ANALYSIS PROGRAM (STEAP) Delivery Order 0049: Computational Prototyping of Micro Air Vehicles				5a. CONTRACT NUMBER FA8650-04-D-3446-0049	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 62201F	
6. AUTHOR(S) Jeff Scott (General Dynamics Information Technology) Gary Clayman (Data Science Automation)				5d. PROJECT NUMBER 2403	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER Q12P	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES): General Dynamics Information Technology Data Science Automation 5100 Springfield Pike, Suite 509 375 Valley Brook Road Dayton, OH 45431 McMurray, PA 15317				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Aerospace Systems Directorate Wright-Patterson Air Force Base, OH 45433-7542 Air Force Materiel Command United States Air Force				10. SPONSORING/MONITORING AGENCY ACRONYM(S) AFRL/RQVC	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-RQ-WP-TR-2013-0030	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES PA Case Number: 88ABW-2013-1713; Clearance Date: 12 Apr 2013. Report contains color.					
14. ABSTRACT This research investigates the energy-efficient optimization of flapping wing micro aerial vehicle win geometry and corresponding optimal wing kinematics.					
15. SUBJECT TERMS clapping wing micro unmanned aerial vehicle (FWMUAV), multidisciplinary, multifidelity, model-based computational tool (M3CT), flapping sciences integration (FSI), multi-physics prototyping (MPP)					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT: SAR	18. NUMBER OF PAGES 58	19a. NAME OF RESPONSIBLE PERSON (Monitor) Philip S. Beran 19b. TELEPHONE NUMBER (Include Area Code) N/A
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
LIST OF FIGURES	ii
LIST OF TABLES	iii
1.0 SUMMARY	1
1.1 Flapping Wing MAV Sizing and Closed-loop Control Optimization in a Distributed Computing Environment	1
1.2 Simulink Engineering Description (Simulator) of a Flapping Wing MAV	1
2.0 FLAPPING WING MAV SIZING STUDY IN A DISTRIBUTED COMPUTING ENVIRONMENT	3
2.1 Introduction	3
2.2 Methods, Assumptions, and Procedures	4
2.2.1 Coordinate System Definition	4
2.2.2 Vehicle Geometry Definition	4
2.2.3 Kinematic Parameterization	6
2.2.4 Control System Modeling	8
2.2.5 Wind Gust Disturbance Modeling.....	9
2.2.6 Optimization.....	10
2.2.7 Distributed Computing Utilizing SORCER Framework.....	13
2.3 Open-Loop Results: Optimization using CONMIN of a Pinned MAV	16
2.4 Closed-Loop Results: Optimization using MMA of a Hovering MAV in Gust.....	19
2.4.1 Convergence.....	21
2.4.2 Results from Optimization Study 1: Analysis of Gust Vector Distributions	22
2.4.3 Results from Optimization Study 2: Analysis of Kinematics for Design and Control Parameters	25
2.5 Multidisciplinary Optimization Cost in the SORCER Environment	29
3.0 FLAPPING WING MAV SYSTEMS ENGINEERING DESCRIPTION AND MODELING USING FSI METHODS AND SIMULINK	31
3.1 Introduction	31
3.2 Methods, Assumptions, and Procedures	32
3.2.1 Coordinate System Definition	33
3.2.2 Vehicle Aerodynamics	33
3.2.2 Vehicle Mechanical and Electrical Engineering Description Modeling	35
3.2.4 Pilot in the Loop Simulation	40
3.3 Application of Simulink Model	41
4.0 CONCLUSIONS	43
5.0 REFERENCES	45

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
Figure 1: Flapping Wing Coordinate System	4
Figure 2: Wing Discretized into Ten Wing Sections	5
Figure 3: Flapping Wing Geometry Description	5
Figure 4: Wing Kinematic Angle Magnitude and Offset Description	6
Figure 5: Split-cycle Waveform ⁴ with Scaling Factor and Prescribed Amplitude ³	8
Figure 6: Lateral (left) and Spherical (right) Gust Vector Distributions	10
Figure 7: MMA-FWMAV Optimization Flow Chart for Multiple Gusts	12
Figure 8: M3CT and Provider Interaction with SORCER Framework	14
Figure 9: M ³ CT Service Provider Graphical User Interface	15
Figure 10: Wing Chord (left) and Thickness (right) Distribution Results	18
Figure 11: Optimized Steady-state Aerodynamic Power Required (left) and the Resultant Cycle-averaged Lift (right)	18
Figure 12: Optimized Output for Each Kinematic Motion	18
Figure 13: Optimized Wing Tip Trace for Aeroelastic Wing and Rigid Wing Model	19
Figure 14: Convergence History of the Design Objective (left), Constraint Behaviors, and Resulting Wing Planform (right) for (from top) Cases 1A, 1B, and 1C	23
Figure 15: Comparison of the Design Parameters: Chord Distribution (left) and the LQR Coefficient (right) for (from top) Cases 1A, 1B, and 1C	24
Figure 16: Gust Spectrum Change in Displacement Performance for Single Gust Vector Optimization and Spherical Gust Optimization	25
Figure 17: Convergence History of the Design Objective (left), Constraint Behaviors, and Resulting Wing Planform (right) for (from top) Cases 2A and 2B	27
Figure 18: Comparison of the Design Parameters: (from left to right) Chord Distribution, Design Kinematics, and LQR Coefficient for (from top) Cases 2A and 2B	28
Figure 19: Orbital Path (left) and Kinematic Perturbations (right) from a Lateral Gust Disturbance Imposed on the Solutions to Case 1 and Case 2	29
Figure 20: Linear Gust versus Nonlinear Gust for Small Disturbances	30
Figure 21: Simulink Ornithopter Model and Engineering Description	33
Figure 22: Ornithopter Coordinate System	33
Figure 23: Wing Discretization and Node Points for Aerodynamic Calculations	34
Figure 24: Vertical and Horizontal Stability and Control Surface Aerodynamic Geometry Description	35
Figure 25: Simulink Ornithopter Electromechanical Model	35
Figure 26: Simulink Mechanical Description of the Ornithopter Body	36
Figure 27: Simulink Mechanical Description of the Ornithopter Right Wing	36
Figure 28: SimMechanics Visualization of the Ornithopter	37
Figure 29: SimMechanics Implementation of the Ornithopter Wing Actuation	38
Figure 30: SimMechanics Model of the Follow Gear and Crank Rod Wing Drivers	38
Figure 31: SimMechanics Model of the Wing Connector Rod	39
Figure 32: SimMechanics Model of the Ornithopter Tail Assembly	39
Figure 33: SimMechanics Model of the Ornithopter Elevator Control Surface	40
Figure 34: FlightGear Simulation of Ornithopter over Water	41
Figure 35: FlightGear Simulation of Ornithopter over Land	41

LIST OF TABLES

<u>Table</u>	<u>Page</u>
Table 1: QS-Aeroelastic Optimization with CONMIN: Case Study Results	17
Table 2: Prescribed Dimensions Based on the Morphology of Manduca Sexta.....	20
Table 3: Design Parameter Initialization and Min/Max Side Constraints	21
Table 4: Summary of Results from Optimization Study 1	22
Table 5: Summary of Results from Optimization Study 2	26
Table 6: UA Ornithopter Specifications	42

1.0 SUMMARY

Research and development of Flapping Wing Micro Air Vehicle (FWMAV) technology relies heavily on the coalescing advancements contributed from various scientific disciplines. The tightly coupled multidisciplinary system of the FWMAV includes fluid dynamics, control theory, and structures; however, the coupling between these areas and the aerodynamics of the flapping wing are nebulous. Like other multi-physics, tightly coupled systems, much of the advances in related research conducted across the various pertinent disciplines are sequestered through lack of awareness or computational compatibility. The purpose of this research is to provide representative research methods that encourage broader development collaboration and the efficient re-utilization of existing computation and information capabilities to further technological development.

The report has been divided into two chapters based on the two main research objectives. The first objective is performed utilizing a distributed computing environment and the “Multidisciplinary, Multifidelity, Model Based Computational Tool” (M³CT) which serves as a graphical interface to the distributed computing environment. For the first objective, a physics-based model of a FWMAV aircraft was developed as part of the MAV Hover Flight Sciences Project through a task entitled “Flapping Sciences Integration” (FSI). This research, “MAV Multi-physics Prototyping” (MPP) applied the FSI tools to FWMAV Quantitative Technology Assessment (QTA). For the second objective, the MPP activity is refocused to enable FWMAV QTA with realistic FWMAV engineering descriptions augmented by physical data obtained from the NATO AVT Task Group 184, “Characterization of Bio-Inspired Micro Air Vehicle Dynamics.” Details related to the research cases are presented in the following sections of this report. A summary of the work performed for the objectives are given below:

1.1 Flapping Wing MAV Sizing and Closed-loop Control Optimization in a Distributed Computing Environment

This research investigates the energy-efficient optimization of flapping wing micro aerial vehicle wing geometry and corresponding optimal wing kinematics. The study is performed using service-oriented distributed computing framework controlled through the use of M³CT. The service-oriented framework is used to apply gradient-based optimization to a pinned flapping vehicle physics-based quasi-steady aerodynamics/aero elastic model and six-degree of freedom closed-loop control quasi-steady aerodynamics model. For the first case, the optimization study is performed with the simulated FWMAV with a fixed, non-translating body position. The kinematic (wing-stroke pattern) and geometric (wing shape) design variables are considered in a sequence of optimization problems with constraints placed on the flapping cycle average thrust. In the second case the optimization of a controlled flapping wing micro aerial vehicle for energy-efficient flight with a 1-cos gust model disturbance is considered. The kinematic (wing-stroke pattern), geometric (wing shape), wing beat frequency, and control (state penalty) design variables are considered in a sequence of optimization problems.

1.2 Simulink Engineering Description (Simulator) of a Flapping Wing MAV

Physics-based models of FWMAV aircraft have started to become available, but these models generally ignore certain vehicle components and their integration at the system level. To quantitatively assess MAV technology, a more detailed engineering description is needed.

Calibrating the models with data obtained by ground or flight test increases the accuracy of these engineering descriptions. A fairly unique source of system-level FWMAV data is the NATO AVT Task Group 184, "Characterization of Bio-Inspired Micro Air Vehicle Dynamics." This Task Group is conducting a broad range of ground and flight tests on different micro air vehicles to characterize their behavior, develop international terms by which FWMAVs are described, and to refine the experimental techniques by which this data is collected.

This research generates a multi-physics FWMAV model engineering description for which data collected by AVT-184 can be used to calibrate FWMAV description. For this particular research, the University of Arizona 25cm Ornithopter is used as the research FWMAV. The calibrated model data and its physical characteristics are used to develop a simulator of the ornithopter that can be used for FWMAV QTA, including the pilot interface. This work was carried out in collaboration with the University of Arizona (Professor Shkarayev, PI).

2.0 FLAPPING WING MAV SIZING STUDY IN A DISTRIBUTED COMPUTING ENVIRONMENT

Summary: The research work presented here evaluates the optimization of a flapping wing micro aerial vehicle for energy-efficient flight using a service-oriented framework in a distributed computing environment facilitated by the M³CT. The optimization is carried out for a vehicle in a pinned environment and a six-degree of freedom environment with closed-loop control. Kinematic and geometric optimization design parameters are considered in a sequence of optimization problems for both pinned and movable simulations while the additional flapping frequency and control (state penalty) design variables are considered for the latter 6DOF case. The service-oriented framework is applied to the coupling of a flapping vehicle physics-based model, a linear quadratic regulator control system, a wind gust model, and gradient-based optimizers. Optimization constraints are applied to maintain adequate vehicle lift conditions and restricted path displacement (in a gust disturbance) along with correlating peak control restrictions resulting from power exerted to maintain a fixed position during hover. Various optimization studies utilizing varying design parameters are evaluated with focus on wing shape optimization and evaluating tradeoff between prescribed periodic kinematic motion and governing the kinematic motion with closed-loop control.

2.1 Introduction

The research case studies presented in this report are examples of utilizing M³CT to perform multi-disciplinary analysis and optimization of a pinned or flying flapping wing MAV. The wing flapping kinematic parameters, wing chord lengths and their respective thicknesses are prescribed for a wing discretized into node elements. The design goal of both the pinned and the flying model is to minimize the cycle-averaged aerodynamic power required to produce a desired lift and control behavior. The optimization results presented in this paper have not been validated against physical models nor have the results been compared with optimization results utilizing more advanced aerodynamic computational methods. The aerodynamic models used in this study utilize a lower fidelity quasi-steady blade element method so that the aerodynamic terms can be projected onto the aero-elastic terms in the pinned model and the dynamic states for rigid wing closed-loop control.

The pinned aero-elastic model and CONMIN optimizer utilized in this research were originally developed in C language and Fortran while the closed-loop control models and the MMA optimization algorithm were developed and tested using Matlab® software and then compiled as stand-alone executable using Matlab Compiler™. The source code and executable modules were integrated into the service-oriented framework, “Service-ORiented Computing EnviRonment” (SORCER); a federated service-to-service meta-computing environment, which employs exertion-oriented programming¹. SORCER allows a myriad of engineering applications, such as stand-alone source and executable modules, to be published as service providers where they can be called upon as part of a federated service object-oriented architecture. To facilitate this research the M³CT was utilized as a graphical interface for generating service requests to the distributed computing environment. The M³CT serves as a tool in which to initialize, control, and monitor the progress of a study composed of multiple distributed physics based models, optimizers, and other engineering design and research applications.

2.2 Methods, Assumptions, and Procedures

2.2.1 Coordinate System Definition

The FWMAV wing and body movement, shown in Figure 1 is comprised of a fixed reference frame in the global coordinate system and the inertial frame of reference $I = \{IX, IY, IZ\}$. The vehicle body frame $R = \{R_X, R_Y, R_Z\}$, is obtained by rotating I with respect to the global frame. The positive direction of the body frame component R_Y extends in what is considered the vehicles normal forward flight while the positive R_X extends out the right wing. The wing frame, $W_n = \{W_{nX}, W_{nY}, W_{nZ}\}$, where n is the wing number, is rotated about R to obtain the flapping sweep angle, wing pitch (about W_{nX}), and elevation as represented by the three Euler angles, ϕ , η , and θ respectively².

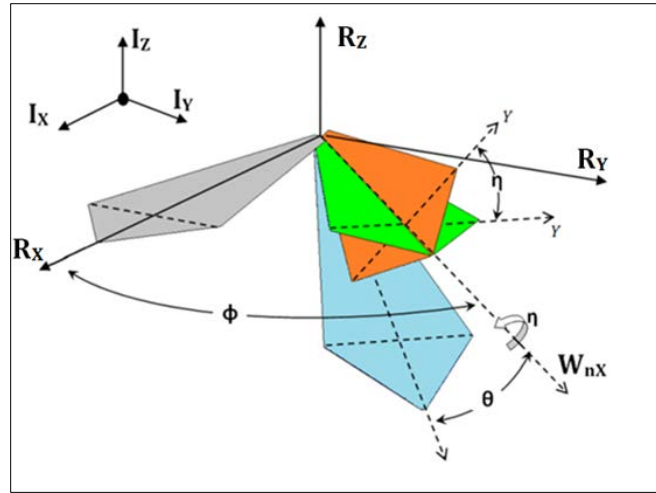


Figure 1: Flapping Wing Coordinate System

2.2.2 Vehicle Geometry Definition

The wing geometry is represented in terms of discretized wing sections described by the chord thickness and chord length at each section. Aligned along the center span line with the wing frame description in the previous section, the wing section parameters are used in performing the calculations in the quasi-steady blade element aerodynamic model. For the research cases presented ten nodes or wing sections were utilized as shown in Figure 2. This represents the complete wing geometric description utilized in the aero-elastic pinned analysis. The closed-loop control geometric description utilizes the same underlying geometric description in the aerodynamics analysis, but adds an abstract level of describing the wing geometry, as discussed below.

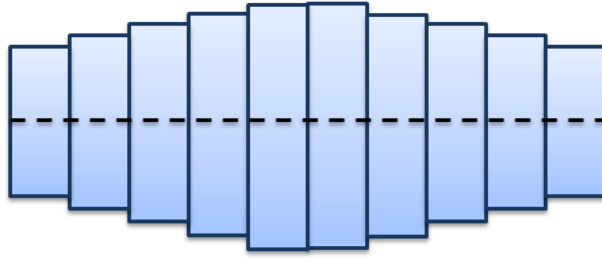


Figure 2: Wing Discretized into Ten Wing Sections

A diagram of the rigid wing geometry used in the closed-loop, six degree-of-freedom model is provided in Figure 3, where the wing shape derived in the work of Bhatia et al.³ is described using eight parameters: three chord section lengths, their respective chord thickness, span-break ratio (SBR), and the wing radius. The three chord sections lengths at the wing root, span-break position, and wing tip (denoted C_0 , C_s , and C_N , respectively) are orthogonal and reflected symmetrically across the center wingspan line, which extends co-linearly with the local x-axis (W_{nX}). A wing cross-section thickness is also assigned to each respective chord position; subsequently the thickness and chord lengths between prescribed chords can be determined from simple linear extrapolation. The SBR represents the position of the span-break chord with respect to the wing radius as measured from the wing root along the wingspan centerline. The resultant geometry is then divided into the 10-node element discretization described above.

A simple symmetrical ellipsoid, described by semi-major and semi-minor axis lengths, represents the vehicle body. Body mass is independent of the volume in this model; however the body mass is accounted for in the inertial terms and contributes to establishing lift requirements for hover. The position of the right flapping wing hinge is defined with respect to the body center of gravity and is independent of the body geometry; subsequently the left wing is positioned symmetrically based on the right wing position. Aerodynamic effects related to the body shape are neglected as well as collision detection between the body and the flapping wing. For this research, this is easily avoided by prescribing appropriate boundaries to the rigid wing kinematics. If an aeroelastic model were introduced this would certainly be of concern and it may be prudent to implement a kinematic boundary based on wing-body collision detections.

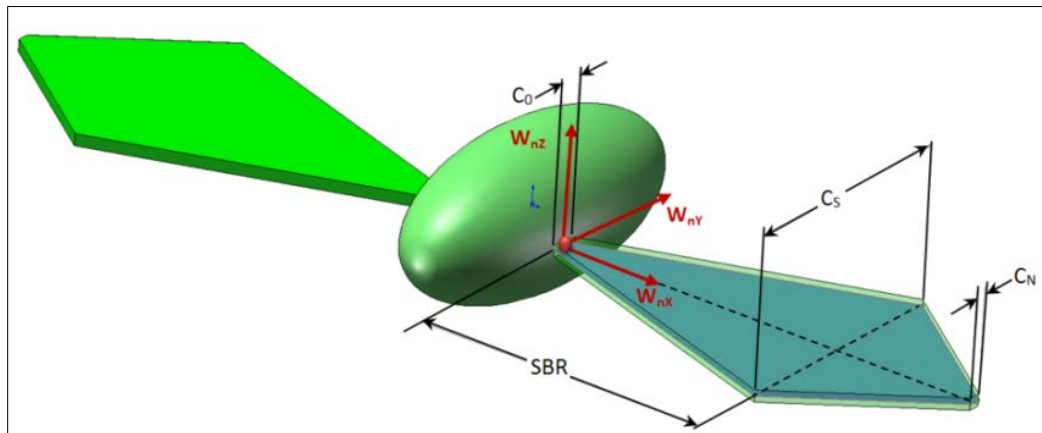


Figure 3: Flapping Wing Geometry Description

2.2.3 Kinematic Parameterization

The flapping wing model presented in this research utilizes a kinematic parameterization that facilitates the split-cycle control approach of Doman et al.⁴ The parameterization has a biological basis, as proposed by Berman and Wang² and used by Stanford et al.⁵ and was extended for smooth transition between flapping cycles by Bhatia et al.³ The three degree-of-freedom flapping motion is assumed to be given by the following Euler angles (see Figure 1) representing the flapping stroke angle (ϕ , sweep), wing elevation (θ , deviation), and pitch (η , feather) as parameterized, periodic functions of time:

$$\phi(t) = \phi_m(1 + A_\phi) \frac{\sin^{-1}(K_\phi \cos \beta)}{\sin^{-1} K_\phi} + \phi_o \quad (1)$$

$$\theta(t) = (1 + A_\theta \cos((\omega - \delta)t))(\theta_m \cos(N_\theta \beta + \theta_s) + \theta_o) \quad (2)$$

$$\eta(t) = \hat{\eta}_m \frac{\tanh(K_\eta \sin(\beta + \eta_s))}{\tanh K_\eta} + \hat{\eta}_o \quad (3)$$

The wing elevation frequency (2) deviates from the flapping frequency, ω , by a factor of N_θ . This deviation has a strong influence on the overall path of the wingtip; e.g., when $N_\theta = 1$, the wing tip follows an elliptic path, when $N_\theta = 2$, the wingtip follows a “figure-eight” path. The subscripts associated with each angle correspond to the angle obtained from the previous flapping cycle (*old*) or the angle magnitudes (*m*), phase shifts (*S*), and offset (*o*). The angle offset, is defined as the angle between the wing neutral line and the center of the angular arc length (magnitude) as shown with respect to ϕ in Figure 4 here the wing neutral line lies collinearly with the FWMAV body y-axis R_Y .

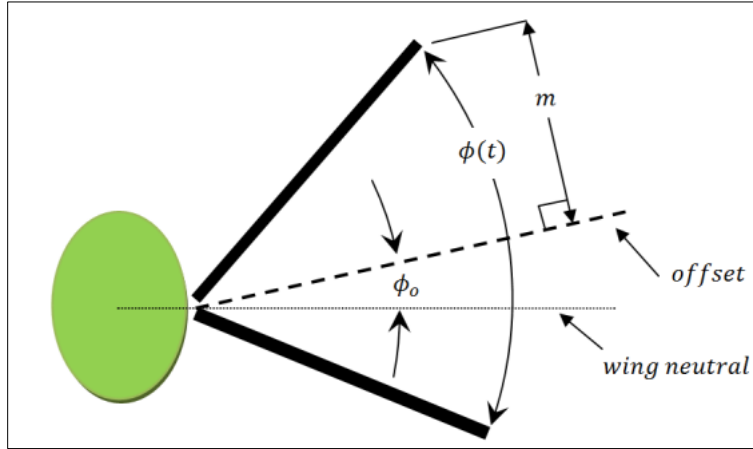


Figure 4: Wing Kinematic Angle Magnitude and Offset Description

The coefficients K_ϕ and K_η define their respective Euler angle waveform function between sinusoidal ($K = 0$) and a triangular function ($K_\phi = 1$) for the flapping stroke plane or a rectangular function ($K_\eta > 1$) for the wing pitch. The term δ represents the split-cycle control parameter, which adjusts the frequency between the up and down stroke, while maintaining a constant period of $2\pi/\omega$. The time-dependent coefficients $\hat{\eta}_m$, $\hat{\eta}_o$, β , and the amplitude scaling factors A_ϕ and A_θ are provided to ensure continuity of flapping motion across consecutive wing-beat cycles.

$$\hat{\eta}_m = \begin{cases} \eta_m \frac{(\omega - \delta)2t}{\pi} + \eta_{m,old} \left(1 - \frac{(\omega - \delta)2t}{\pi}\right) & 0 \leq t < \frac{\pi}{2(\omega - \delta)} \\ \eta_m & \frac{\pi}{2(\omega - \delta)} \leq t < \frac{2\pi}{\omega} \end{cases} \quad (4)$$

$$\hat{\eta}_o = \begin{cases} \eta_o \frac{(\omega - \delta)2t}{\pi} + \eta_{o,old} \left(1 - \frac{(\omega - \delta)2t}{\pi}\right) & 0 \leq t < \frac{\pi}{2(\omega - \delta)} \\ \eta_o & \frac{\pi}{2(\omega - \delta)} \leq t < \frac{2\pi}{\omega} \end{cases} \quad (5)$$

$$\beta = \begin{cases} (\omega - \delta)t & 0 \leq t < \frac{\pi}{\omega - \delta} \\ \frac{\omega(\omega - \delta)t - 2\pi\delta}{\omega - 2\delta} & \frac{\pi}{\omega - \delta} \leq t < \frac{2\pi}{\omega} \end{cases} \quad (6)$$

$$A_\phi = \begin{cases} \frac{\phi_{m,old} + \phi_{o,old} - \phi_o}{\phi_m} - 1 & 0 \leq t < \frac{\pi}{2(\omega - \delta)} \\ 0 & \frac{\pi}{2(\omega - \delta)} \leq t < \frac{2\pi}{\omega} \end{cases} \quad (7)$$

$$A_\theta = \begin{cases} \frac{\theta_{m,old} \cos \theta_s + \theta_{o,old}}{\theta_m \cos \theta_s + \theta_o} - 1 & 0 \leq t < \frac{\pi}{2(\omega - \delta)} \\ 0 & \frac{\pi}{2(\omega - \delta)} \leq t < \frac{2\pi}{\omega} \end{cases} \quad (8)$$

These angular prescriptions are for a single wing; a similar set of parameters describes the motion of the second wing. The resultant, prescribed kinematic parameters that fully define the time varying motion of the flapping wings are:

$$\mathbf{X}_{kin} = \{ \phi_m, \phi_o, K_\phi, \theta_o, \theta_m, \theta_s, \eta_o, \eta_m, \eta_s, K_\eta, \omega, \delta \}$$

As presented by Bhatia et al.³, the time dependent scaling factors are applied to the closed-loop control “flying” model at the beginning quarter of the cycle to ensure that the wing-beat offset values are reflected in the current cycle rather than the following cycle, as is the case in the original split-cycle method by Doman et al.⁴ Additionally, the scaling factor Bhatia et al.³

introduced incorporates the amplitude parameters ϕ_m and η_m for the wing flapping stroke plane and pitch respectively, as opposed to the originally fixed values of $\pi/2$ and $\pi/4$. The flapping stroke plane over a single wing-beat is shown in Figure 5, where the modified Bhatia et al.³, implementation is compared with the original split-cycle waveforms of Doman et al.⁴ for values of δ less than zero (left), equal to zero (center), and greater than zero (right). The first quarter cycle scaling factor plot depicts the transition from a flapping stroke magnitude of $\pi/3$ found in the previous cycle to the current flapping stroke magnitude of $\pi/2$. The remaining waveform, which includes the impeded upstroke and advanced down stroke, remains consistent between the two approaches. For this research, the split-cycle term δ is specified to vanish throughout the optimizations. Note that in the pinned model, the original split-cycle method presented by Doman et al.⁴ is used.

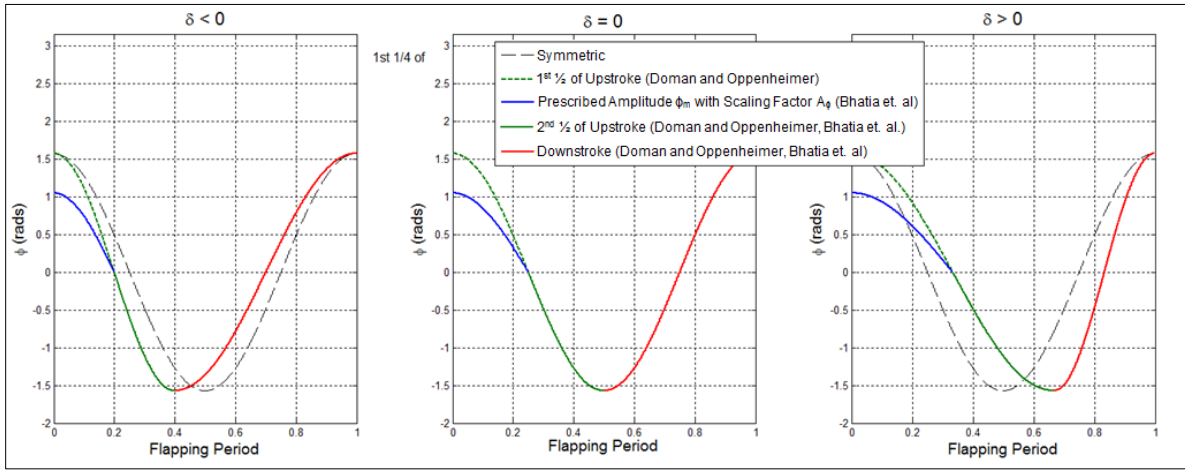


Figure 5: Split-cycle Waveform⁴ with Scaling Factor and Prescribed Amplitude³

2.2.4 Control System Modeling

For obtaining closed-loop control for stabilized orbital position in the presence of gust disturbances, the flapping wing model relies on a linear quadratic regulator (LQR) for calculation of an optimal controller gain matrix ($\mathbf{K}$) such that the feedback changes to the wing kinematics ($\mathbf{q}$) relate to changes to the current state vector ($\mathbf{x}$) by:

$$d\mathbf{q}^{t_0} = -\mathbf{K}d\mathbf{x}^{t_0} \quad (9)$$

This is accomplished first by modeling the FWMAV as a nonlinear periodic system using Floquet analysis to find the periodic trim. The periodic shooting method is then applied to obtain the trimmed periodic orbit, as described by Stanford et al.⁵. This orbit is then converted to a discrete linear representation of the system about the trim orbit, which also coincides with the period of the flapping cycle. As a result, the state of the FWMAV can be defined by its position (both linear and angular) and respective time derivatives related to both the inertial frame and body frame as shown in the position vector:

$$\mathbf{x}^{t_0} = \{x_I \quad y_I \quad z_I \quad \theta_x \quad \theta_y \quad \theta_z \quad \dot{x}_B \quad \dot{y}_B \quad \dot{z}_B \quad \dot{\theta}_{x,B} \quad \dot{\theta}_{y,B} \quad \dot{\theta}_{z,B}\}$$

As shown by Bhatia et al.³, given the position vector $\mathbf{x}$ (above) and the kinematic vector $\mathbf{q}$, the sensitivity of the states at the end of each flapping cycle with respect to the state at the

beginning of the flapping cycle can be expressed as a linearized discrete-time system of equations in the form of:

$$d\mathbf{x}^{T+t_0} = d\mathbf{x}^{t_0} \left[\frac{d\mathbf{x}}{d\mathbf{x}^{t_0}} \right]_{t=T+t_0} + d\mathbf{q}^{t_0} \left[\frac{d\mathbf{x}}{d\mathbf{q}^{t_0}} \right]_{t=-T+t_0} \quad (10)$$

The vector $d\mathbf{q}^{t_0-T}$ contains only the kinematic parameters from the previous cycle that influence the state variables from the current cycle. The kinematic parameters that may be used for trim are prescribed in the model setup. To establish an effective gain matrix, LQR theory attempts to identify those values for $\mathbf{K}$ that best minimize the linear quadratic cost:

$$\mathbf{K} = \rho \mathbf{Q} + \mathbf{B}^T \mathbf{P} \mathbf{B}^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} \quad (11)$$

In the equation above, $\mathbf{P}$ is obtained from the solution to the *discrete algebraic Riccati* equation⁶, while the matrices $\mathbf{A}$ and $\mathbf{B}$ are the system matrix and control coefficient matrices, respectively, of the discrete linear time-invariant system model. The coefficient ρ is of particular importance in this research, because it represents requital for control-cost in the form of state-penalty. Subsequently, as ρ increases, the emphasis on minimizing control cost increases; conversely, as ρ decreases, the emphasis on minimizing state cost decreases. More discussion on ρ and its impact related to optimization is provided later. It is from the state sensitivities described above that we also calculate the cycle-averaged aerodynamic power for the baseline kinematics, along with the power sensitivities related to the control inputs, which are also discussed in further detail later.

2.2.5 Wind Gust Disturbance Modeling

In the closed-loop “flying” model, we introduced a wind-gust model to provide a disturbance input to the closed-loop control system. To generate disturbances in the orbital flight of the FWMAV, this research utilizes the 1-cos linearized discrete gust model⁷ with gust vectors distributed over a spherical boundary enclosing the FWMAV. Each gust is defined by three quantities: direction (as unit vector), magnitude, and frequency. The gust disturbance can be analyzed as the result of a single vector or the statistical average over multiple gust vectors applied laterally, longitudinally, or spherically. For a linearized gust model, the FWMAV kinematic perturbations are linearly proportional to the gust magnitudes, since the response is the product of the forcing function. The resultant disturbance may be scaled according to the gust magnitude, eliminating the need for multiple gust velocity optimizations. To ensure complete capture of the entire gust cycle, the number of flapping cycles is calculated as a function of the gust frequency (f_G) as shown in (12).

$$N_{FlappingCycles} = \frac{\omega}{2\pi f_G} \quad (12)$$

From (12), it can be shown that lower gust frequencies will have a more persistent effect on a FWMAV displacement for an equal number of flapping cycles. In cases involving multiple linear gust vectors, nodes are equally distributed laterally or longitudinally over a semi-circle, as shown in the left of Figure 6. or over a semi-sphere as shown in the right of Figure 6. Equally distributing spherical coordinates can lead to tighter clusters of distributed vectors at the antipodal points representing the axis of rotation. To avoid this, the equal spherical distribution of nodes is calculated using the golden ratio; ensuring that the distance between all adjacent

nodes are the same. The resultant gust vectors are located at the node points (as shown by the arrows in the left of Figure 6) directed radially towards the center of the spherical orbit boundary. Taking advantage of the left/right symmetry of the FWMAV to eliminate redundancy, we omitted half of the spherical gusts corresponding to the local FWMAV body coordinate shown in Figure 1 ($R_Y < 0$).

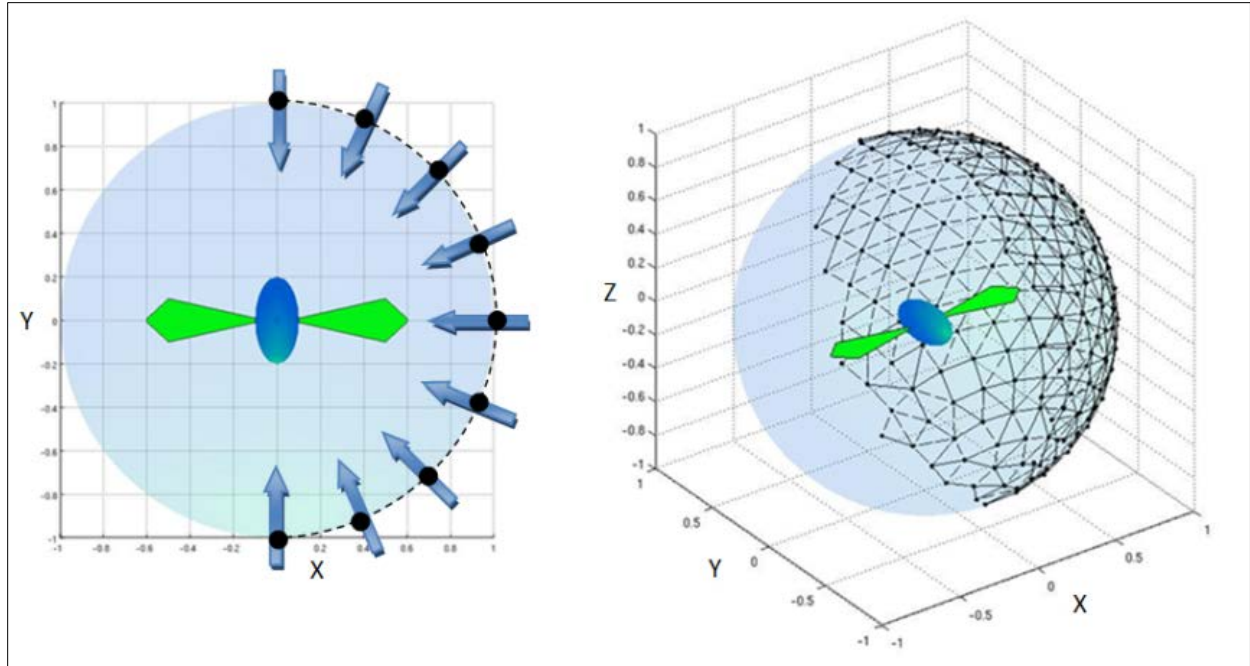


Figure 6: Lateral (left) and Spherical (right) Gust Vector Distributions

2.2.6 Optimization

For this research, two gradient-based optimization methods were utilized. The first method, CONMIN, was applied to the pinned model analysis and optimization. The CONMIN optimizer utilizes the Method of Feasible Directions to compute the minimization of constrained linear or nonlinear functions. Originally written in FORTRAN, CONMIN was developed in 1973 by Garret N. Vanderplaats at the Ames Research Center and U.S. Army Air Mobility Research and Development Laboratory. The objective function and constraint behavior are provided to CONMIN by the optimization problem in M³CT by way of the SORCER context object. The analytical constraint gradients are also provided through the context object, but may (as an alternative to the method used) be calculated by CONMIN using the built in finite-difference method.

For the second portion of this research involving the analysis of the closed-loop, FWMAV, the method of moving asymptotes (MMA) gradient-based optimization was utilized. The MMA method, originally presented by Svanberg⁸, is coupled with the flapping wing and wind gust models. The optimization approach includes single gust and multiple gust profiles for mean efficiency optimization. In each optimization case, minimization of the cycle-averaged power over the entire gust simulation is used as the objective function. The peak control power expended and the total flight orbit spherical displacement of the FWMAV are used as metrics for constraint behavior.

Given the generic design parameters $x=[x_1, \dots, x_{j=n}]^T$, implicit constraint behavior $f_i(x)$, objective function $f_o(x)$, and their gradients $\Delta f_i(x)$ and $\Delta f_o(x)$, the MMA obtains an optimal solution for the given design iteration from a convex, separable sub-function in which the implicit constraint behavior is replaced with an explicit approximation. The MMA is unique in that the explicit constraint behavior is obtained as a linearization of the implicit constraints based on upper (U_j) and lower (L_j) asymptotes for which $L_j < x_j < U_j$. The magnitudes of the asymptotes are modified according to the behavior of the problem gradients. As an example, if the optimization process is oscillatory, the asymptotes may be squeezed closer together; conversely the asymptotes may be relaxed if the optimization is too unproductive. A more detailed discussion related to the theory and applications related to the method of moving asymptotes can be found in Ref. 8.

As part of the philosophy for the optimization presented, emphasis was placed on obtaining a reduction in power, but not at the cost of “run-away” control authority; as was experienced in the closed-loop optimization. If one were to simply measure the cycle-averaged forces obtained from the baseline kinematic perturbations during trim and linearization, the additional forces incurred from the perturbations caused by control gain would be discounted. Therefore, the behavior constraint related to peak control cost was added to the closed-loop optimization problem; the control cost is calculated as the product of the linear aerodynamic power and changes in kinematic perturbations resulting from the controller gain. The constraint behavior related to the peak control cost is then a function of the cycle-averaged power and a constraint coefficient such that the peak control cost is less than kP , where the constraint coefficient k represents the ratio of total cycle-averaged power dedicated to changes in control input. The peak control cost is then monitored and captured by a peak detector over the entire gust simulation. The motivation behind constraining the peak control power stems from the practical power requirements for FWMAVs. In real-world applications, fuel cells have not only limited capacity, but also limited discharge rates. An optimal design solution requiring low peak power input inherits a wider design selection of available power sources.

The optimization workflow diagram is presented in Figure 7 for the closed-loop study using the MMA optimization method. The first step in the process is to define the initial conditions of the design parameters. The initial conditions are applied to the *objective function*, which includes the FWMAV trim and linearization, LQR design of the controller, and discrete linear gust simulation. From the objective function, the objective value and constraint behaviors are determined; the constraint behavior may be defined from a single gust disturbance or an average of multiple gust disturbances. The gradients of the objective value and side constraints are then calculated along with the design parameters, and are normalized with respect to the design parameter side constraints. The normalized information is provided to the MMA optimizer, which determines updated values of the design parameters. These new parameter values are un-normalized and presented to the FWMAV objective function. The entire process is repeated until the MMA optimizer declares a converged solution, a predefined maximum number of iterations have been met, or the user has decided the convergence is satisfactory and terminates the study.

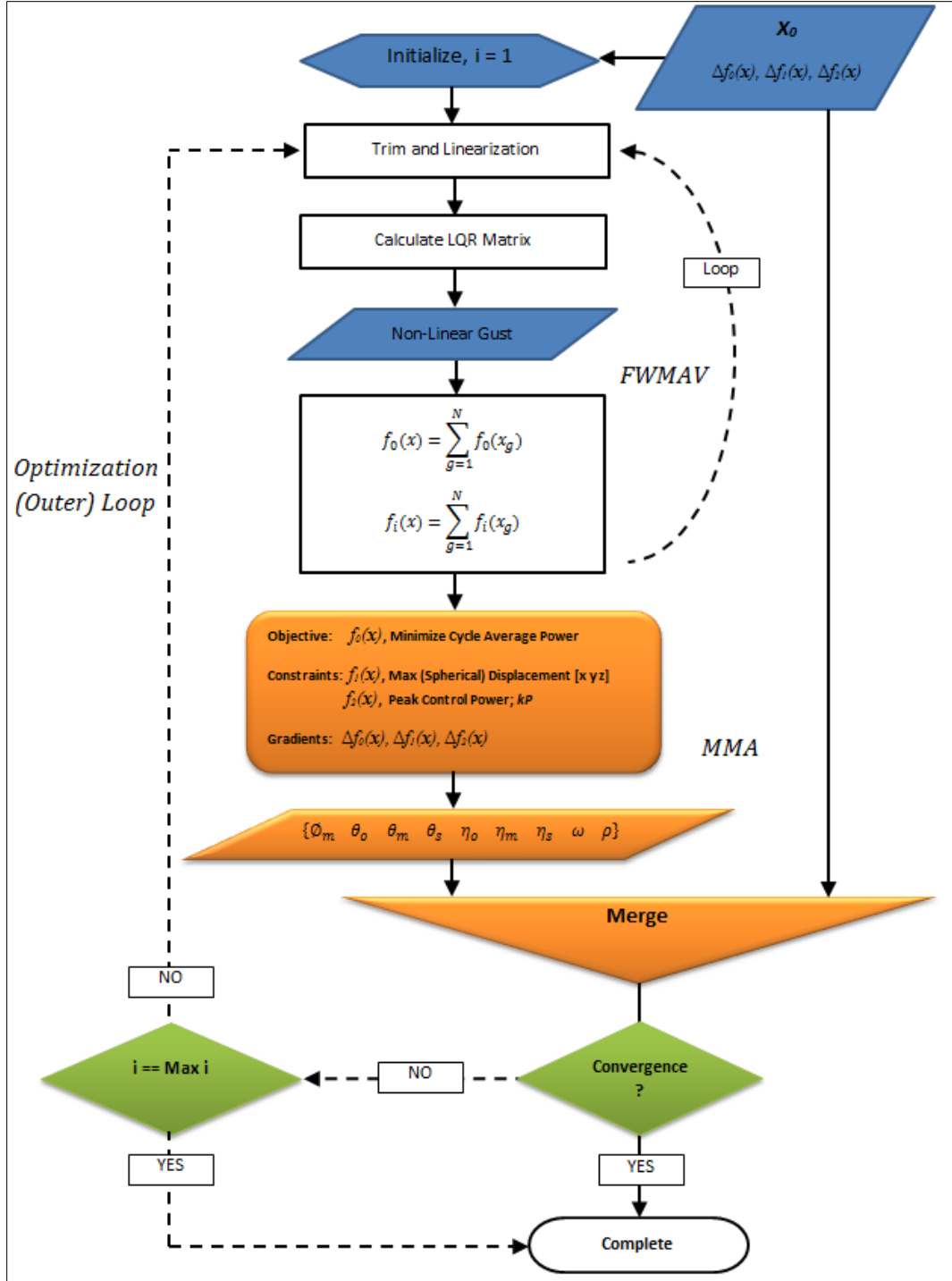


Figure 7: MMA-FWMAV Optimization Flow Chart for Multiple Gusts

Both the pinned and the closed-loop 6DOF models represent tightly coupled systems; through optimization, each change in geometric and kinematic parameter impact the other parameters. With varying levels of influence, each of these changes affects all other aspect of the design objective and constraints behavior. In the closed-loop control case, the LQR coefficient ρ was introduced as the primary control design parameter. Unlike other parameters,

which indirectly improve constraint behaviors, ρ has the unique ability to balance the opposing constraints related to displacement and peak control power. The LQR coefficient fundamentally tailors the LQR solution to the given design optimization.

Kinematic parameters assigned to the closed-loop model controller may be optimized with respect to their assigned baseline definition. The baseline defines how each particular kinematic parameter behaves in the absence of control authority (i.e., the pinned optimization case) and directly influences the formulation of the LQR gain matrix. For the purposes of this research, kinematic parameters assigned to the controller are excluded from the optimization. This was done to avoid conflict with the additional gradients resulting from the direct changes to the LQR gain matrix.

For the closed-loop FWMAV model, the optimization can be applied to the FWMAV subjected to a linear or nonlinear gust. Both methods establish closed-loop control via LQR theory, as presented above. For the linearized method, cycle-averaged forces and power are calculated based on the state sensitivities calculated from the discrete-time equation derived from the linearized periodic trim orbit. For the nonlinear method, cycle-averaged forces and power are derived from the perturbations calculated from applying the time-marching method over the nonlinear gust model.

2.2.7 Distributed Computing Utilizing SORCER Framework

This research was performed in a distributed computing environment using SORCER framework coupled with a M³CT. This approach was taken for two reasons: it mitigates the tedious process of managing multiple optimization studies and it accelerates the discovery of future solutions by encouraging the efficient reuse of existing models, applications, and configurations regardless of their native development environment.

SORCER provides an environment for which engineering models, applications, and data are made available across a potentially distributed, heterogeneous network of computing resources. The SORCER environment is rooted in the Jini service oriented architecture technology developed by Sun⁹. SORCER inherits from Jini a federated service-to-service metacomputing environment that utilizes explicit leases, distributed events, transactions, and discovery/join protocols that enable SORCER to regard service hosts as network peers⁹. SORCER diverges from the Jini network service management by focusing on exertion-oriented programming and providing the execution environment for these exertions⁹.

As a federated environment, SORCER permits a single service exertion (requestor) to organize a dynamic collection of collaborating service (providers) at runtime^{10,11}. Each provider deploys a particular service (e.g. MMA or FWMAV model) by publishing its proxy object to the collection of SORCER registries as shown in Figure 8; this proxy object serves as a discovery mechanism between the provider and requestor. It's through the registries that service requestors can dynamically explore and access each proxy object, given that the service availability is extended to that particular requestor by the provider; this exploration is performed without the requestor having any prior knowledge or dependency related to the provider's platform, architecture, implementation, or network location. SORCER allocates the necessary computational resources for each exertion (or request) at runtime based on the requirements presented by the requesting service and the federation of providers. This meta-processing allows

the aggregation of the service requestor and providers, which can utilize the protocols defined in their respective proxy objects to dynamically collaborate with one another directly so as to appear as one program operating on a single machine⁹.

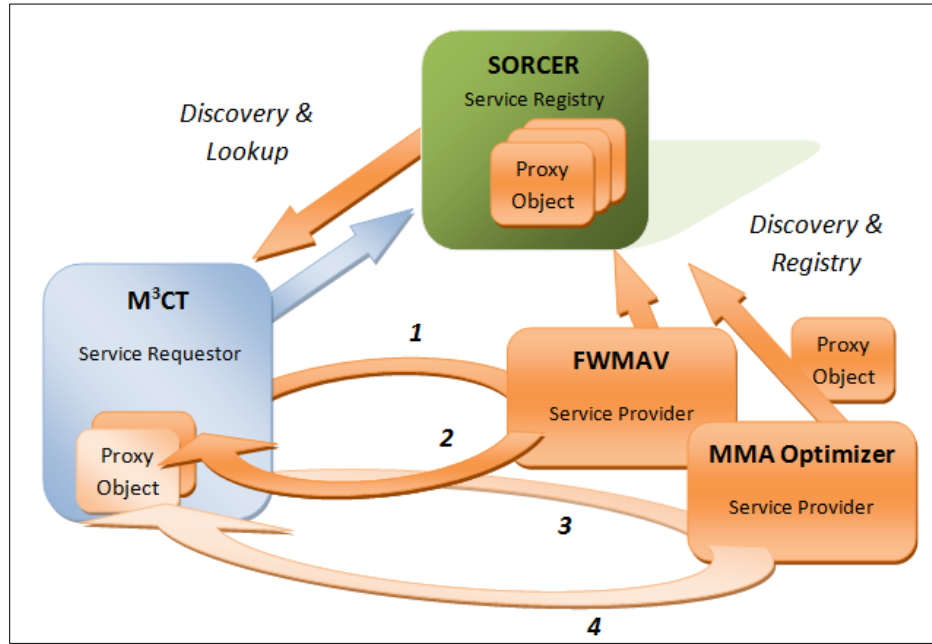


Figure 8: M3CT and Provider Interaction with SORCER Framework

In this research, the requesting service is the M³CT graphical user interface, while the FWMAV models and either the CONMIN or the MMA optimizer are implemented as service providers. The purpose of M³CT is to improve access to the SORCER environment from an end-user's perspective. This is accomplished by providing a graphical tool suite, like the one shown in Figure 9, for which studies can be deployed using quick and intuitive initialization methods, while providing autonomous monitoring for each case study. M³CT accomplishes this within SORCER, circumventing the computer science background typically required of the user to work in a distributed-computing environment. Performing a study in M3CT, using the closed-loop optimization research, using MMA as an example, would require that the user simply select the FWMAV models and MMA optimizer which have already been identified in SORCER by M3CT. Once the FWMAV and MMA modules are added to the workspace, the user simply defines the conditions for each model through the appropriate properties dialogue interfaces and then launches the study. M3CT manages the context between the FWMAV models and the MMA optimizer, while continuously monitoring the progress towards convergence. This process is depicted in Figure 8 above, in which M³CT invokes the FWMAV model through the network to generate the baseline results (step 1). A completion is signaled to M³CT from the FWMAV provider, which then processes the data (step 2) and invokes the MMA provider, providing the newly obtained data as the optimization conditions (step 3). The MMA provider then signals M³CT that it has completed, and M³CT retrieves the data from the MMA provider, which contains the next iteration optimization values to be passed to the FWMAV (step 1); this process is repeated until a predefined condition is met such as a convergence acknowledgment return from step 4.

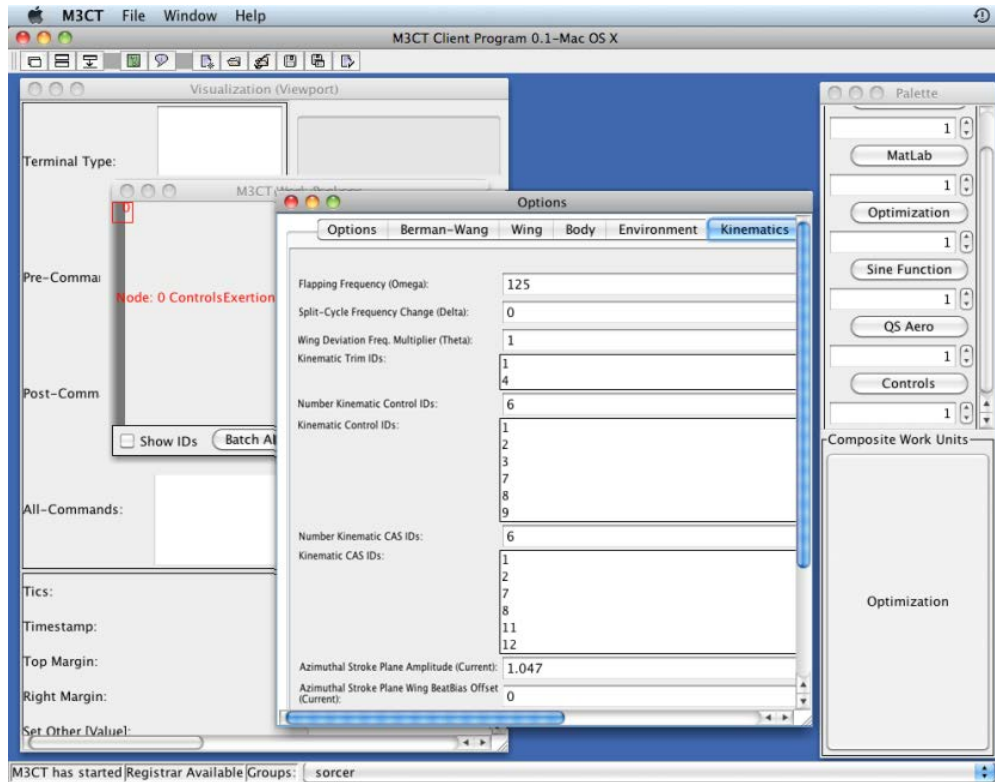


Figure 9: M³CT Service Provider Graphical User Interface

The pinned FWMAV model and CONMIN optimizer were previously integrated within the SORCER framework under Task Order 35 of this contract. This research was focused on developing a QTA capability integrated by M³CT in the SORCER environment using Fortran components. For the closed-loop model and MMA optimization method, a different integration approach was taken. As part of this research, a rapid method of utilizing existing Matlab source code and integrating that code into the SORCER framework via M³CT was considered. This was accomplished using the Matlab Compiler™. The FWMAV model for the closed-loop optimization include the quasi-steady blade element aerodynamics model, the LQR based controls, and the gust models, all originally developed by Bhatia et al.³ and Stanford et al.¹² These models were coupled under a single main routine, which provides design parameter management and methods for requesting gradient calculations. The MMA optimizer was compiled as a separate stand-alone executable utilizing the core routines presented by Svanberg⁸ with the addition of an integrated design parameter, objective, and constraint manager. These additions allowed the MMA provider to operate in the distributed framework in a more modular fashion.

The first step in integrating the existing Matlab version of the FWMAV and MMA algorithms into SORCER required a SORCER service provider application wrapper to be developed for each application. The SORCER development environment provides a set of utilities, such as a SORCER service provider/requestor template generator, for integrating arbitrary applications into SORCER. The FWMAV related models and MMA optimizer were kept in their original Matlab® implementation and integrated into SORCER and M³CT using the provided template generator. Data was exchanged between each provider object and its corresponding executable using an ad-hoc, file-based, context-management approach. In the

file-based approach, a provider object first writes to the executable input file, then invokes the executable, and finally reads from the corresponding output file. The file-based approach represents a crude, but effective proof-of-concept for representing methods of integrating existing models developed in programs like Matlab with little development time required by the originator.

A comparison to determine the difference in computational cost associated with running the FWMAV model and MMA in their native Matlab® script and as stand-alone executables was performed. The results indicated negligible differences in wall-clock time between the two executions; in most cases it was less than one percent in computation time, favoring back and forth. Additionally, the computational overhead associated with SORCER was also deemed negligible with 99% of the computational time for the study being consumed by the execution of the providers themselves, including the high overhead of the file-based, read-write methods embedded in the provider executables. Future versions of the service-provider implementation demonstrated here will utilize distributed shared memories, such as Sun JavaSpaces¹³, to improve efficiency while leveraging on the existing engineering methodology. The primary efficiency in utilizing the shared memory stems from the basic access speed associated with system memory rather than disc memory, which is inherently more costly. The shared memory approach also improves efficiency by accessing only changes in relevant parameters, rather than passing (through the network) large files, which rapidly grow over the course of simulation. Additional improvements may be made, such as facilitating the computation of finite-difference based sensitivities with parallel computing to obtain the gradients, or utilizing the adjoint method used by Beran et al.¹⁴ rather than the direct analytical method used in this research. It is the opinion of the authors that the overall benefit of collaborating in a the distributed computing environment and the wide range of problems that may be studied by leveraging the development of others far outweighs the existing cost associated with its use.

2.3 Open-Loop Results: Optimization using CONMIN of a Pinned MAV

The numerical methods described in the previous section were applied to obtain the following optimization results from the pinned Quasi-Steady (QS)-Aeroelastic model and the CONMIN gradient optimizer service providers in M³CT. The goal of this study is to minimize the cycle-averaged aerodynamic power required to produce 0.15 N of lift for a sustained flapping frequency of 125 rad/s. (The evaluated power is normalized by the specified lift.) The kinematic parameters, wing chord lengths and their respective thicknesses are prescribed for a wing composed of ten node elements. The initial kinematic and geometric design parameters and their final optimized solutions are reported in Table 1. The optimization is performed with a prescribed limit of 50 iterations; no other convergence criteria were prescribed in this analysis.

Table 1: QS-Aeroelastic Optimization with CONMIN: Case Study Results

Category	Description	Initial	Optimized
Objective	Cycle-Averaged Power (W/N)	3.672	0.787
Constraint	Lift (N)	0.241	0.151
	Peak Stress Indicator (kS)	1.412	3.989×10^{-3}
Kinematics	Azimuth Amplitude (ϕ_m)	1.047	0.776
	Azimuth Sharpness (K_ϕ)	0.010	0.010
	Elevation Amplitude (θ_m)	0.000	0.055
	Elevation Offset (θ_o)	0.000	-0.055
	Elevation Shift (θ_s)	0.000	0.011
	Pitch Amplitude (η_m)	0.785	0.945
	Pitch Offset (η_o)	1.571	1.571
	Pitch Phase (η_s)	-1.571	1.145
	Pitch Squareness (K_η)	0.100	0.102
Wing Geometry	From Root to Tip (m)	2.250×10^{-2}	2.331×10^{-2}
		3.238×10^{-2}	3.227×10^{-2}
		3.978×10^{-2}	3.956×10^{-2}
		4.472×10^{-2}	4.433×10^{-2}
		4.719×10^{-2}	4.664×10^{-2}
		4.719×10^{-2}	4.660×10^{-2}
		4.472×10^{-2}	4.430×10^{-2}
		3.978×10^{-2}	3.967×10^{-2}
		3.238×10^{-2}	3.267×10^{-2}
		2.250×10^{-2}	2.320×10^{-2}
	From Root to Tip (m)	4.300×10^{-4}	5.343×10^{-4}
		5.209×10^{-4}	3.970×10^{-4}
		5.890×10^{-4}	5.463×10^{-4}
		6.344×10^{-4}	5.967×10^{-4}
		6.572×10^{-4}	5.981×10^{-4}
		6.572×10^{-4}	5.770×10^{-4}
		6.344×10^{-4}	5.434×10^{-4}
		5.890×10^{-4}	4.991×10^{-4}
		5.209×10^{-4}	4.424×10^{-4}
		4.300×10^{-4}	3.715×10^{-4}

The initial and final designs are compared, in terms of wing chord and thickness distribution, in Figure 10.

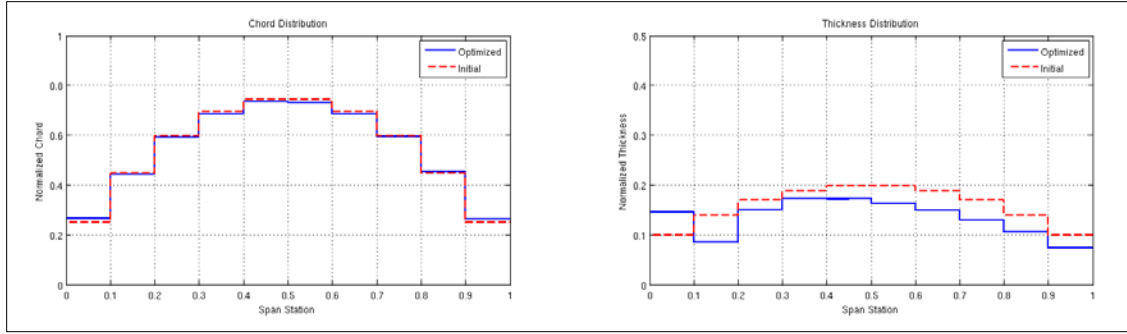


Figure 10: Wing Chord (left) and Thickness (right) Distribution Results

In this case study, the cycle-averaged power required (shown in Figure 11) was reduced 79% from the original required power. The reduction in cycle-averaged power was accomplished in part by reducing cycle-averaged lift 37% over the course of optimization to meet the lift constraint; the initial design generated more lift than necessary.

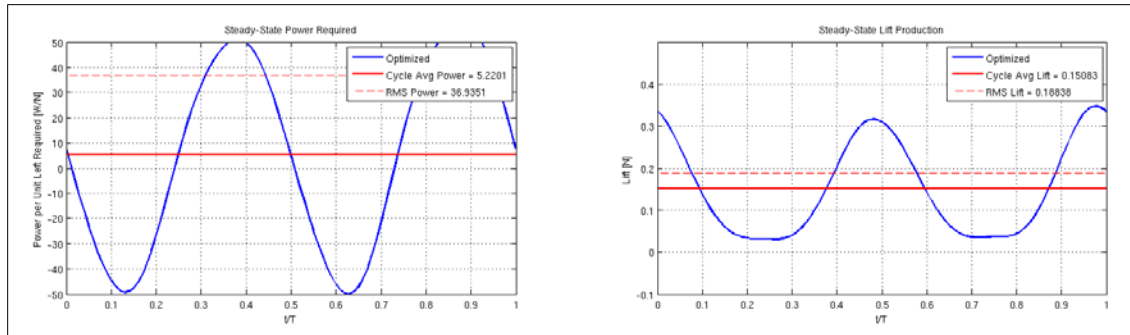


Figure 11: Optimized Steady-state Aerodynamic Power Required (left) and the Resultant Cycle-averaged Lift (right)

To minimize the resultant required power, the optimal kinematic motion (Figure 12) exploits the elastic response of the wing for the given flapping frequency, as shown in Figure 13.

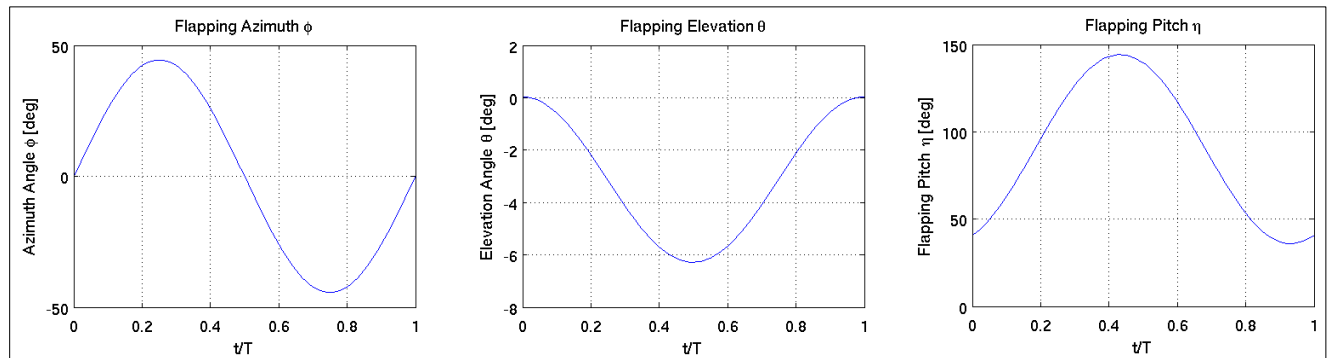


Figure 12: Optimized Output for Each Kinematic Motion

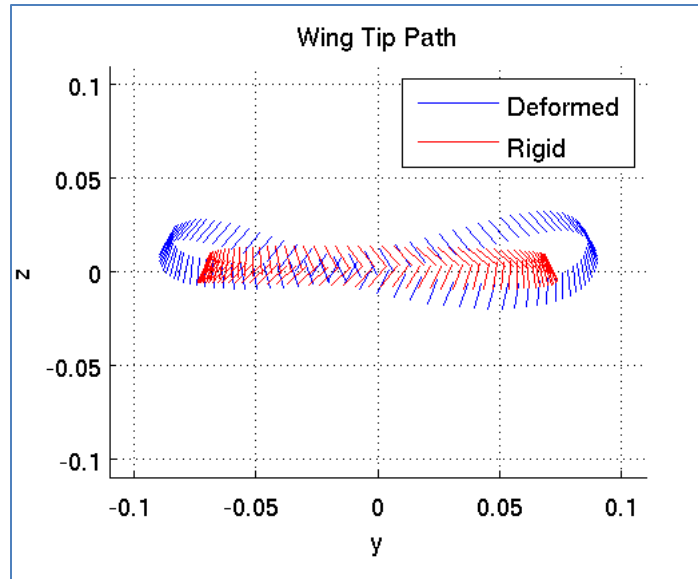


Figure 13: Optimized Wing Tip Trace for Aeroelastic Wing and Rigid Wing Model

2.4 Closed-Loop Results: Optimization using MMA of a Hovering MAV in Gust

For the portion of this research concerned with optimization of the closed-loop vehicle, the MAV was only considered for maintaining hovering flight with prescribed spherical constraints proportional to the gust frequencies and magnitudes. The trim calculation for the closed-loop control of the MAV balances physical forces, including gravity. As a result, the force inputs associated with maintaining orbital position, hover, are inherited in the control design. This eliminates the need to prescribe an additional design constraint related to lift required in a similar optimization study reported by Bryson et al.¹⁵ Five different optimization cases were considered for various gust scenarios and different sets of design variables.

For each optimization case, the vehicle mass body mass, wing thickness, and other relevant physical quantities were chosen to be consistent with the morphological parameters of the common hawkmoth (*Manduca sexta*), as described by Hedrick and Daniel¹⁶. Values are specified in Table 2, and were fixed for all the optimization cases. The wing thickness was distributed equally along each chord section and held constant throughout the optimization. The hawkmoth wing density was estimated from the wing planform area (989 mm^2), thickness, and mass. Unlike the hawkmoth, however, the wing chord distribution was reflected symmetrically about the un-swept mid-chord-line (wing symmetry is a limitation of the current FWMAV model).

Table 2: Prescribed Dimensions Based on the Morphology of Manduca Sexta

Component	Parameter	Value
Body	Mass (kg)	3.75×10^{-3}
	Length (m)	4.66×10^{-2}
	Radius (m)	6.00×10^{-3}
Wing	Radius (m)	5.30×10^{-2}
	Mean Thickness (m)	3.00×10^{-4}
	Planform Area (m ²)	9.89×10^{-4}
	Mass (kg)	4.60×10^{-5}
	Density (kg/m ³)	1.55×10^2
	Flapping Frequency (rad/s)	8.50×10^2

Two sets of optimizations are presented here; the first set focuses on evaluating optimization methods associated with single, laterally, and spherically distributed gust disturbances with prescribed kinematic parameters. The second set of studies considers the addition of kinematics as either design parameters or control parameters. For all cases, the gust disturbances occur at a constant frequency of 0.25 Hz over the first 340 of the total 680 flapping cycles. *All cases treat the wing geometric parameters and the linear quadratic controller cost function coefficient, ρ , as design variables.* Initial values of design variables and side constraints related to each case are presented in Table 3. Optimization results are summarized in Tables 4 and 5. The cases are as follows:

- Case 1A: Optimization of wing geometry and ρ for a single lateral gust disturbance.
- Case 1B: Optimization of wing geometry and ρ for multiple lateral gust disturbances.
- Case 1C: Optimization of wing geometry and ρ for multiple spherically distributed gust disturbances.
- Case 2A: Optimization of wing geometry, ρ , and the magnitudes, offsets, and phase shifts of the wing pitch and sweep for spherically distributed gust disturbances.
- Case 2B: Optimization of wing geometry, ρ , and the phase shift of the wing pitch and sweep under spherically distributed gust disturbances (the magnitudes and offsets for both the wing pitch and sweep are not optimized, but rather assigned to the controller).

In Case 1A, optimization was performed for a single gust to provide a baseline for Case 1B and Case 1C. In the second optimization study, we consider optimization under spherically distributed gust vectors with the addition of the kinematic design parameters θ_s and η_s . We also evaluate the effects of assigning θ_m , θ_o , η_m , η_o as either design parameters or control parameters. In both optimization studies, ϕ_m and ϕ_o are assigned as control parameters, while K_ϕ , K_η , and ω are held constant at 0.010, 0.010, and 85.0 rad/s (13.5 Hz), respectively.

Table 3: Design Parameter Initialization and Min/Max Side Constraints

Parameter	Description	Initial (Case 1)	Initial (Case 2)	Minimum Value	Maximum Value
Kinematics					
θ_m	Wing elevation amplitude (rad)	0	0	$-\pi/4$	$\pi/4$
θ_o	Wing elevation offset (rad)	0	0	$-\pi/4$	$\pi/4$
θ_s	Wing elevation phase shift (rad)	0	0	$-\pi/2$	$\pi/2$
η_m	Wing pitch amplitude (rad)	$\pi/4$	$\pi/4$	$-\pi/2$	$\pi/2$
η_o	Wing pitch offset (rad)	$\pi/2$	$\pi/2$	$-\pi/2$	$\pi/2$
η_s	Wing pitch phase shift (rad)	0	0	$-\pi/2$	$\pi/2$
Controller					
ρ	LQR coefficient (-)	0.001	0.001	10^{-10}	10
Geometry					
C_0	Root chord (m)	0.047	0.064	0.008	0.100
C_S	Span break chord (m)	0.047	0.026	0.005	0.100
C_N	Tip chord (m)	0.047	0.008	0.005	0.100
SBR	Span break ratio (%)	0.581	0.581	0.125	0.875

2.4.1 Convergence

For identifying the completion of each optimization, a simple convergence criterion was calculated based on the natural log of the absolute change between two consecutive design iterations. The design is considered converged when the solution is less than the prescribed tolerance; otherwise a visual check was used to classify a solution as converged. In many cases, the visual examination suffices when the convergence is oscillatory. Minor oscillations typically resulted from the sensitivities to the dynamic peak control power constraint. This constraint has a highly sensitive correlation with the spherical displacement and objective power.

In the event convergence is not met due to excessively oscillatory solutions, the optimization is terminated automatically by exceeding a predefined maximum number of iterations set by the user. In most instances of convergence failure, stiffness was evident in the opposing constraints of peak control power and spherical displacement. These constraints are difficult to balance while maintaining a feasible solution. A proposed method to address this issue is to implement a relaxation factor in the design constraint when oscillatory solutions are detected. Treating the convergence history as a signal waveform and analyzing the signal strength for a prescribed number of iterations may accomplish this relaxation. The side constraints may be relaxed when the signal strength exceeds some pre-determined threshold.

A second method for addressing design convergence requires determining an appropriate optimization step size. The MMA step size is a division of the normalized design parameters and may be adjusted to improve convergence. A correlation between the number of optimization design parameters and the ideal step size was prevalent in this study; however a specific quantification was never determined. It was found that for most cases, a step size of 1% was ideal; this typically resulted in a slow, but steady convergence. In optimization cases defining less than five design parameters, it was found that the step size could be increased to values

greater than 10%. In some instances, this increase in step size resulted in failure of the Floquet trim computation, and subsequent dynamics linearization. These instances usually occurred at the beginning of a study when arbitrary initial conditions were assigned to a set of design parameters that are tightly coupled. It was determined that beginning a study with a small step size for given set of arbitrary initial conditions and then increasing the step size as the solutions began to show signs of convergence helped speed up the optimization while maintaining a steady convergence trend. Subsequently, modifications to the Floquet solver were made so that the initial conditions for the Floquet solver in the trim and linearization were initialized based on converged Floquet solutions from the previous iteration, given that only very small changes in parameters are applied between iterations. In most cases, this drastically reduced the total number of iterations required to converge on a trim solution.

2.4.2 Results from Optimization Study 1: Analysis of Gust Vector Distributions

The purpose of the first study is to provide insight related to how various gust profiles acting on the MAV affect closed-loop vehicle performance. It is expected that a single gust vector will result in an optimal design for that particular scenario and that vehicle performance will decay as the gust direction vector deviates further away from the direction assumed in the optimization. Additionally, optimizations resulting from the statistical average of multiple gust vectors will generate more robust design solution across the gust spectrum. However, they are not expected to outperform the single-gust cases, which are highly tuned. Results from the three cases evaluated in this study are presented in Table 4.

Table 4: Summary of Results from Optimization Study 1

Case	1A	1B	1C
Cycle Avg. Power (W/kg)			
Initial Value	4702	4702	4729
Final Value	76.96	76.88	970.5
Change	-4625	-4625	-3758
Max Displacement (m)			
Initial Value	0.6495	0.6495	1.342
Final Value	0.6005	1.058	2.000
Change	-0.0490	0.4085	0.6580
Peak Control Cost (W/kg)			
Initial Value	20.47	36.64	70.35
Final Value	0.9689	5.984	51.27
Change	-19.50	-30.66	-19.08
Controller (final value)			
ρ	9.982	9.978	10.00
Geometry (final value)			
C_0 (m)	0.0075	0.0075	0.0155
C_S (m)	0.0122	0.0115	0.0235
C_N (m)	0.0117	0.0111	0.0365
SBR (%)	0.7842	0.7355	0.4719

In Cases 1A and 1B of this study, lateral response lacks the direct counteraction against lift, which is present in the longitudinal gust vectors. The negated lift and power required to compensate for such are in part opposing weights in the average calculation and therefore the

negated longitudinal effects are, to some degree, minimized. It may be advantageous to perform optimizations considering only the quarter sphere to minimize the positive lift, which inherently tends to contribute the lowest additional control cost. The third case (1C) final solution resulted in a much larger max displacement and cycle average power than the other two cases. This can be attributed to the combination of the additional longitudinal component of the gust distribution and possibly to some degree the dynamic nature of the peak power constraint. The failure to reduce the total objective power in this case resulted in a higher peak power constraint, which may have failed to stimulate the optimization. The convergence history data from the study related to the minimization of power design objective and the constraint behaviors for peak control power and spherical displacement are shown in Figure 14.

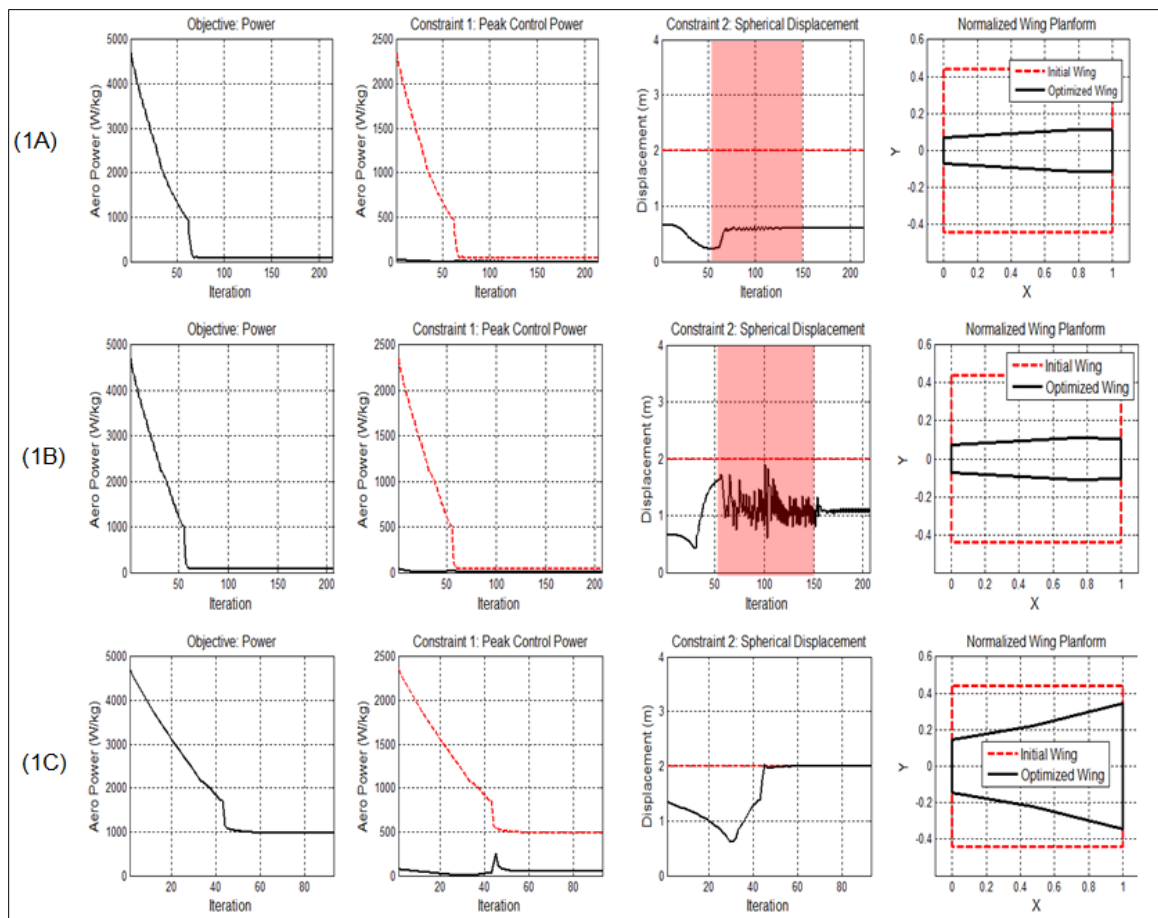


Figure 14: Convergence History of the Design Objective (left), Constraint Behaviors, and Resulting Wing Planform (right) for (from top) Cases 1A, 1B, and 1C

For all three cases in the first study, an initial optimization step size of 1% was assigned, along with a limit of 45 maximum iterations. The steps sizes were increased to 10% after all three cases failed to converge after the first 45 iterations; the cases were then allowed to continue for an additional 100 cycles. The third case (1C) was able to converge after 25 additional iterations, while the 10% increase for the first and second cases destabilized the convergence. The oscillatory convergence continued for the remainder of added iterations. For the third attempt, an additional 100 iterations were added and the optimization step sizes were reduced to 5%. The decrease in step size enabled both of the optimizations to converge on a steady solution

after an additional 25 iterations (175 total iterations). The oscillations related to the increase in step sizes are highlighted in red for the spherical displacement in Figure 14 and for the design parameters in Figure 15.

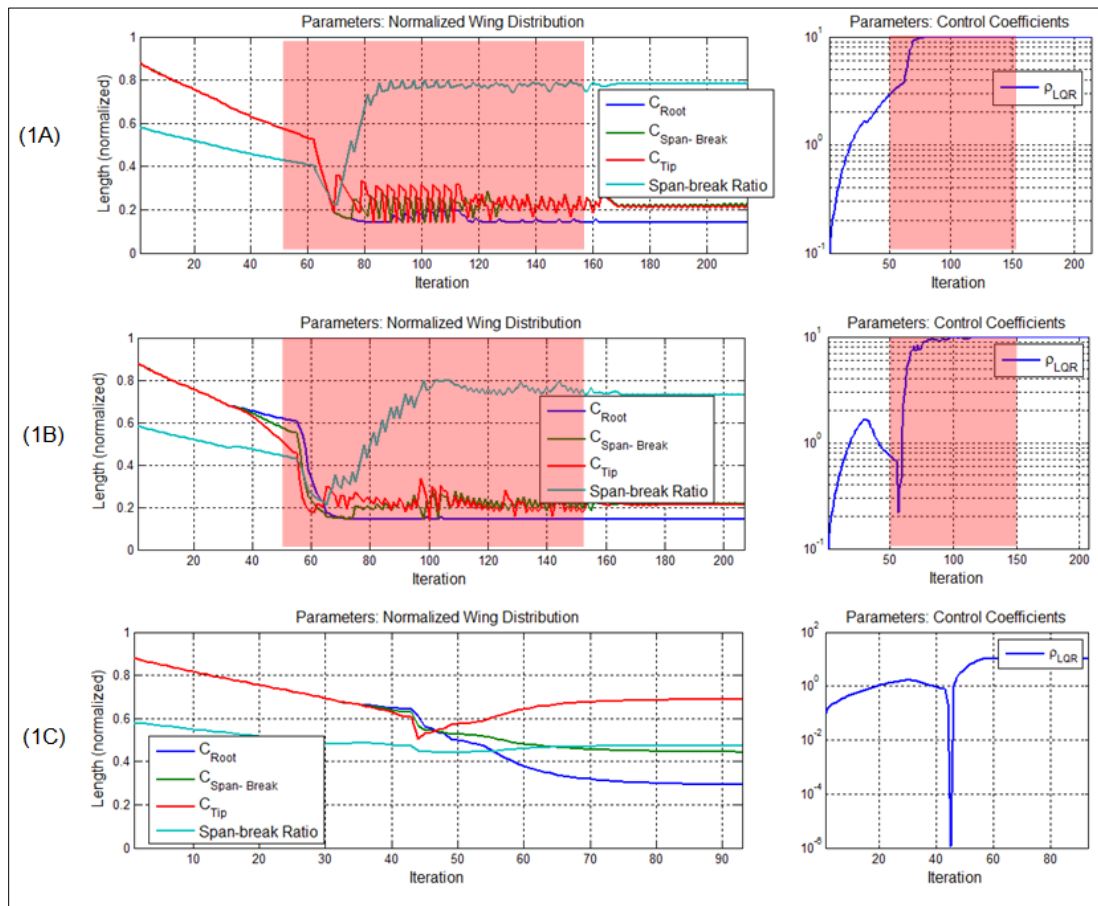


Figure 15: Comparison of the Design Parameters: Chord Distribution (left) and the LQR Coefficient (right) for (from top) Cases 1A, 1B, and 1C

The maximum absolute displacements of the single gust optimization (Case 1A) relative to the spherical gust optimization (Case 1C), defined over an entire spherical gust distribution, are presented in Figure 16. While not presented as a case in this study, a single negative longitudinal gust optimization is shown (right) along with the single lateral gust optimization used in Case 1A (left). The data is presented in the form of a three-dimensional spectrum of the spherical displacement performance where the gusts are described using polar coordinates. As a guide, the red arrows indicate the lateral and longitudinal gust directions along their respective axis.

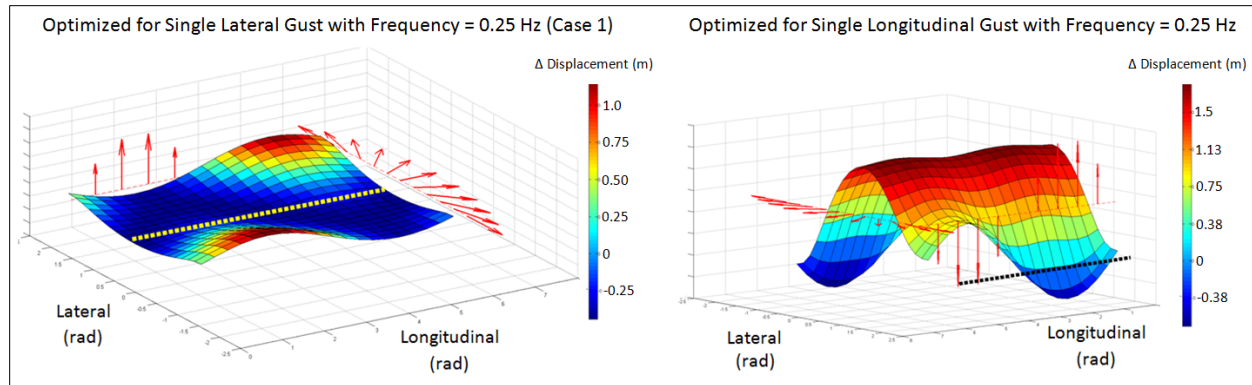


Figure 16: Gust Spectrum Change in Displacement Performance for Single Gust Vector Optimization and Spherical Gust Optimization

The results from the analysis were consistent with our previous expectations for performing optimization over multiple gust profiles. In each instance, the single gust optimizations exceeded the third case in performance for the single lateral or longitudinal gust in which it was optimized for as indicated by the dark blue regions in Figure 16; a yellow line is shown along the strongest lateral gust component on the left. Conversely, the performance of both single gust optimizations decreased relative to the third case as the gust directions diverged from the respective lateral or longitudinal direction (as indicated by the dark red regions). As indicated previously, we see that the displacement reaches a maximum when the longitudinal gust force is directed downwards in the left of Figure 16. In the right figure, we see that the lateral gust has a greater negative effect on the longitudinally optimized gust than the vertical gust has. Similar results can be found with regard to the control peak power and cycle average power, but are not presented here.

2.4.3 Results from Optimization Study 2: Analysis of Kinematics for Design and Control Parameters

The second study augments the previously developed MAV closed-loop controller presented by Bhatia et al. Here we experiment to provide additional insight into the kinematic optimization as it relates to control by comparing cases in which a single degree of freedom is allocated to the flapping wing controller (case 2A) or all three degrees of freedom are allocated to the controller (case 2B). In each instance, kinematics corresponding to the degrees of freedom are either allocated to the controller or identified as design parameters for optimization. The cases are representative of the control input sets 1 and 2 used by Bhatia et al.³, in their lateral and longitudinal gust disturbance analyses.

Table 5 summarizes the optimization results from the presented cases; the flapping stroke plane magnitude (ϕ_m) and offset (ϕ_o) are designated control kinematics and are never selected for optimization. For each of the presented cases, a spherical gust distribution containing 200 gust vectors was used. The cycle average power and maximum displacements were calculated as the mean average values over the entire 200 gusts. The peak control power was defined as the maximum of the peak powers obtained from each gust scenario. For each case an MMA step size of 1% was prescribed for each normalized parameter.

Table 5: Summary of Results from Optimization Study 2

Case	2A	2B
Cycle Avg. Power (W/kg)		
Initial Value	3638	3670
Final Value	55.27	30.75
Change	-3583	-3639
Max Displacement (m)		
Initial Value	1.087	0.0294
Final Value	1.909	0.5445
Change	0.822	0.5151
Peak Control Cost (W/kg)		
Initial Value	34.69	5329
Final Value	25.15	8.410
Change	-9.54	-5321
Kinematics (final value)		
θ_m (rad)	9.31×10^{-4}	Controller
θ_o (rad)	0.2584	Controller
θ_s (rad)	0.5868	0.0787
η_m (rad)	0.1256	Controller
η_o (rad)	1.371	Controller
η_s (rad)	0.3995	-0.9115
Controller (final value)		
ρ	6.328	8.193
Geometry (final value)		
C_0 (m)	0.0206	0.0080
C_S (m)	0.0352	0.0117
C_N (m)	0.0371	0.0139
SBR (%)	0.6348	0.7457

From the data above, assignment of kinematic design variables to the controller (Case 2B) decreased cycle-averaged power by 44% and reduced max displacement by 67% over the baseline spherical gust case (Case 2A). Equally as important is the reduction in peak control cost associated with the optimal solution for Case 2B, which is substantially lower than that of case 1C.

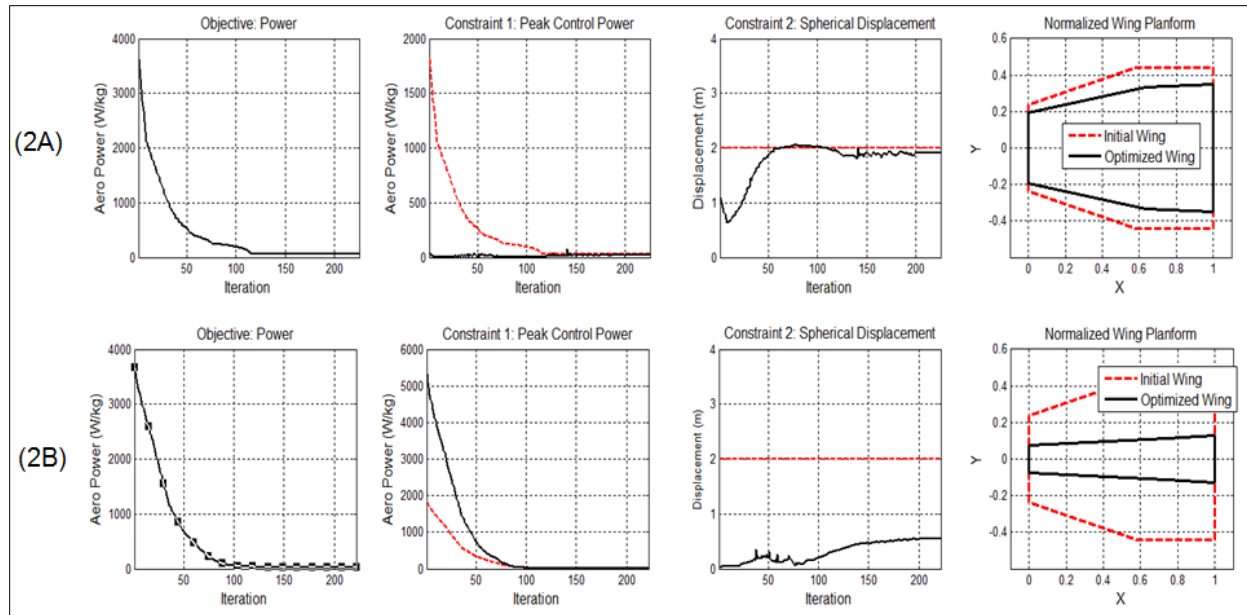


Figure 17: Convergence History of the Design Objective (left), Constraint Behaviors, and Resulting Wing Planform (right) for (from top) Cases 2A and 2B

The peak power constraint in Case 2B benefited the greatest from optimization, as shown in Figure 17. The large reduction in wing planform area between Cases 2A and 2B can be attributed to the need to reduce moments about the wing in order to assert higher control response. This geometry change likely contributed to the reduction in control power, but the linear quadratic cost coefficient, shown in the parameter convergence history in Figure 18, had the strongest influence on the control cost by sacrificing displacement, as previously discussed in this report.

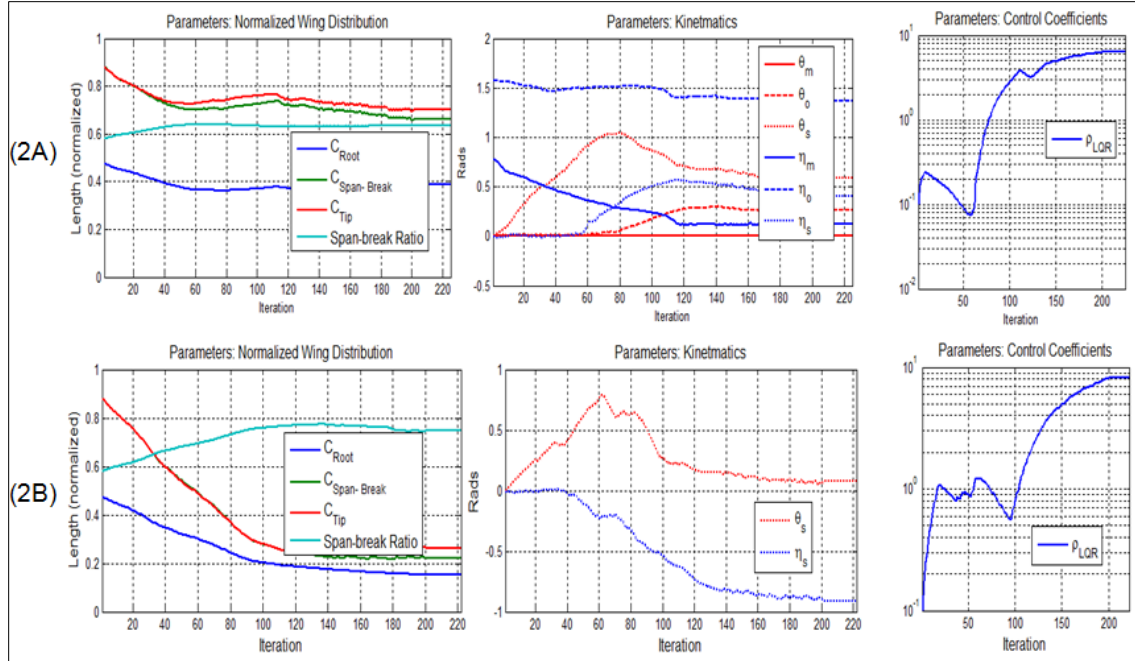


Figure 18: Comparison of the Design Parameters: (from left to right) Chord Distribution, Design Kinematics, and LQR Coefficient for (from top) Cases 2A and 2B

To provide a basis for direct comparison, Figure 19 shows the orbital paths (left) and kinematic perturbations (right) when Case 1 and 2 are subjected to a single discrete lateral gust disturbance of 1 m/s over a 4 second time period followed by 2 second period with no gust. It can be seen from the orbital path that utilizing the state displacement as a design constraint provides only the magnitude at which the vehicle has been disturbed from its initial point; it does not require that the vehicle return to that point. In case 1, we can see that the vehicle orbital path stochastically deviated further than that of case 2 and was unable to return to the point of origin in the allotted simulation time. The solution in case 2 generated a smooth orbital response path and was able to reasonably maintain its position at the point of origin in the allotted simulation time. The control authority distributions from both cases are shown on the right. The first case relied solely on the flapping stroke plane kinematic (ϕ) to maintain orbital stability requiring more mechanical work to return to its origin.

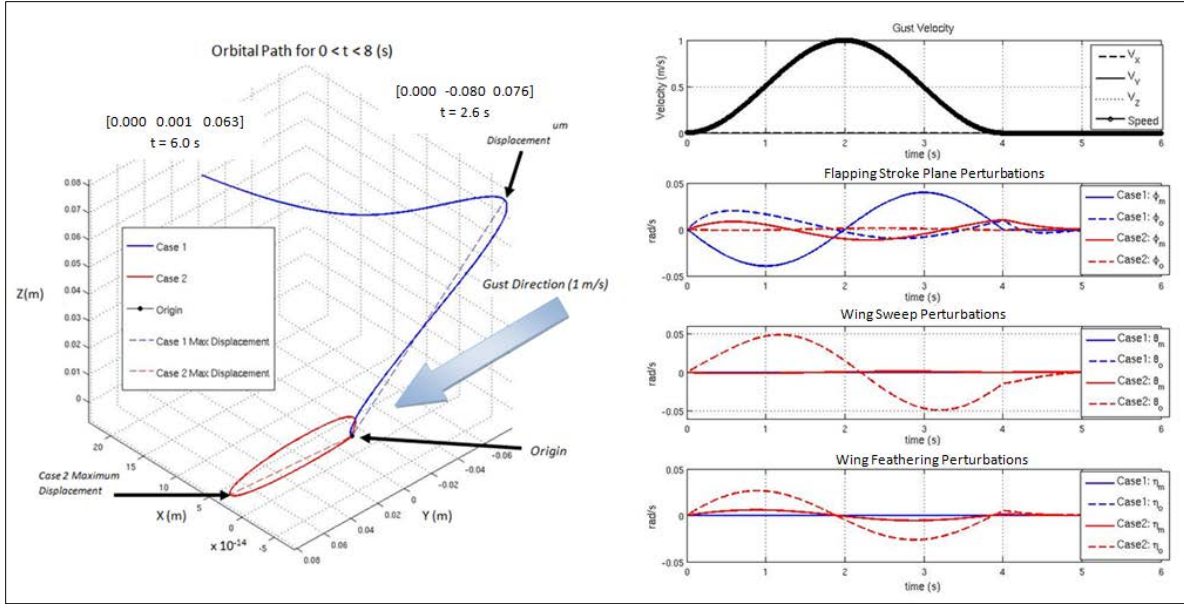


Figure 19: Orbital Path (left) and Kinematic Perturbations (right) from a Lateral Gust Disturbance Imposed on the Solutions to Case 1 and Case 2

2.5 Multidisciplinary Optimization Cost in the SORCER Environment

In comparing the computational overhead of executing test cases in SORCER versus stand-alone, there was nearly no measureable performance cost associated with the addition of the distributive computing component. The typical run time for an optimization utilizing the linearized gust equations with 8 design parameters and 200 gust vectors was roughly 4.5 hours, while the optimizations involving nonlinear gust equations (not presented here) with the same parameters required up to 72 hours. Typical relative costs associated with running the linear versus the nonlinear gust scenarios can be presented by the following equations which provide a rough time estimate in seconds, based on the number of gust vectors (N_G) presented, the number of design parameters (N_x), and optimization iterations (N_{IC}):

$$\text{Linear Optimization Time} = N_{IC}(40 + 0.001N_G)N_x \quad (13)$$

$$\text{NonLinear Optimization Time} = N_{IC}(40 + 2.5N_G)N_x \quad (14)$$

As can be seen in (13), little additional computational cost is associated with the number of gust vectors in the linearized gust model method. For example, to analyze 2500 linear gust vectors versus 250 vectors requires only 18 additional minutes. It was discovered, however, that little optimization advantage was gained through additional gust vectors, so long as the vectors are equally distributed.

Additionally, it was verified through additional testing that the linear and nonlinear gust simulations are in good agreement for gust speeds up to 1.5 meters/second. See Figure 20.

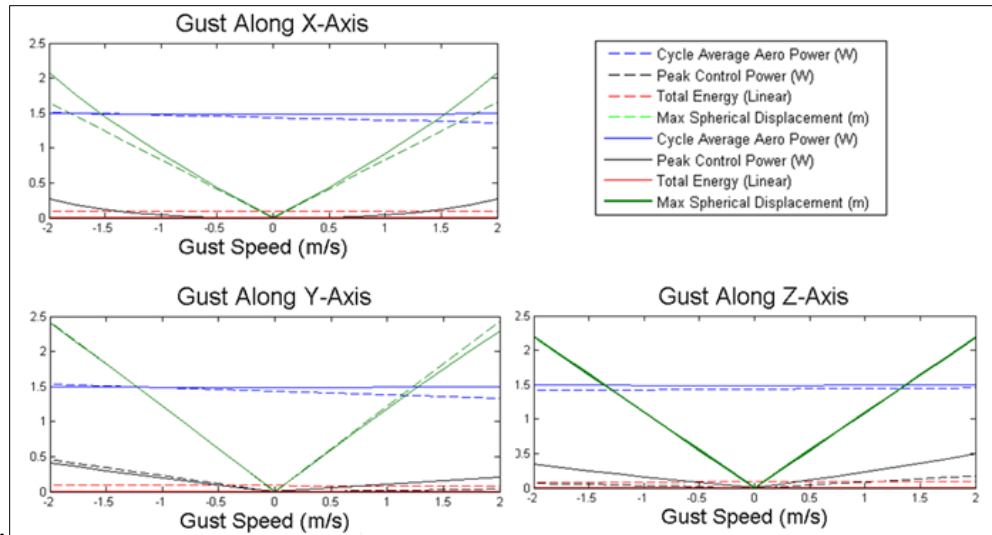


Figure 20: Linear Gust versus Nonlinear Gust for Small Disturbances

3.0 FLAPPING WING MAV SYSTEMS ENGINEERING DESCRIPTION AND MODELING USING FSI METHODS AND SIMULINK

Summary: This research effort utilizes Mathworks® Simulink® software and computational methods developed under the Flapping Science Integration (FSI) effort to generate a systems engineering description of the University of Arizona 25cm Ornithopter FWMAV. The FWMAV is implemented using multi-physics modeling techniques and experimentally obtained data related to flight performance and system dynamics. The calibrated model is used to develop a simulator of the ornithopter that can be used for FWMAV quantitative technical assessment. The systems engineering model encompasses aerodynamics, mechanics, and electronics while maintaining low computational overhead and real-time simulation for pilot in the loop testing. To reduce the computational overhead, the aerodynamic methods developed under the FSI effort are enhanced to incorporate prescribed surface deformation based on experimentally obtained aeroelastics. Mechanical, electrical, and kinematics (to include wing deformation) data collected by the University of Arizona are used to calibrate FWMAV descriptions. Real-time, pilot-in-the-loop simulation capability is provided for qualitative analysis. Simulink real-time synchronization and an open-source, flight-simulation package are used to generate the real-time experience and visualizations based on the system model states. The experience in developing the systems engineering description and modeling are used as comparative metrics in a quantitative technical assessment of the FSI effort.

3.1 Introduction

Physics-based models of FWMAV aircraft have started to become available, but these models generally ignore certain vehicle components and their integration at the system-level. To quantitatively assess MAV technology, a more detailed engineering FWMAV description is needed. Physics-based models of FWMAV aircraft have been developed as part of the MAV Hover Flight Sciences Project through a task entitled FSI. Another task “MAV MPP was funded to apply the FSI tools to FWMAV QTA. In this effort, the MPP activity is refocused to enable FWMAV QTA with realistic FWMAV engineering descriptions augmented by physical data. Calibrating the models with data obtained by ground or flight test increases the accuracy of these engineering descriptions.

A fairly unique source of system-level FWMAV data is the NATO AVT Task Group 184, “Characterization of Bio-Inspired Micro Air Vehicle Dynamics.” This Task Group is conducting a broad range of ground and flight tests on different micro air vehicles to characterize their behavior, develop international terms by which FWMAVs are described, and to refine the experimental techniques by which this data is collected. This portion of the research focuses on utilizing methods developed under TO-49 to generate a multi-physics engineering description model with Mathworks Simulink software.

The Simulink software was chosen based on its commercial availability and multi-physics simulation capabilities. Simulink and the computational interface methods (M³CT) developed for FSI provide two different approaches for generating multi-physics modeling. While Simulink provides a comprehensive simulation and modeling package, M³CT provides an interface to the distributed computing environment with the capability to accommodate a myriad of computer science languages, implementation methods, and utilization of distributed

computational resources. The purpose of performing this research using Simulink is to garner a better understanding of widely accepted methods for developing multi-physics modeling using a visual “black-box” system component approach. This research reviews ways to emulate the user experience and workflow that Simulink offers and apply that experience to the distributed computing approach. The intent of this future study is not to supplant Simulink or generate an alternative version, but to gain a different perspective on implementing systems modeling.

3.2 Methods, Assumptions, and Procedures

The Ornithopter model and systems engineering description is developed in Mathworks Simulink software. Simulink software provides a method of implementing the engineering description in a modular, self-documentation modeling style, which utilizes a visual interface resembling a flow chart. The Simulink model (shown in Figure 21) has been modularized into five main sub-models: Electromechanical, Aerodynamics, Control, Flight Data and Environment, and Flight Simulator. The Electromechanical description utilizes the Simulink Simscape software to develop a comprehensive multi-physics electrical and mechanical description of the ornithopter to include the battery behavior, motor and gear mechanics, wing and control surface actuation, along with the inertial and mass properties of the ornithopter mechanical assembly. The Aerodynamics module is implemented in native Simulink mathematical model form and calculates the aerodynamics forces, which are then coupled to the mechanical system. The Control module represents the internal and external (pilot) control interface for driving the vehicle motor and flight control systems. The environment module models the atmospheric characteristics of the model to include wind gusts. The flight data module models the six-degree-of-freedom movement of the vehicle based on the force and inertial description from the mechanical system and the flight profile from the environment system. The Flight Simulator module provides methods for displaying metrics from the simulation and provides an interface to the FlightGear flight simulation software for real-time, pilot-in-the-loop flight testing and visual feedback.

The FWMAV systems model is implemented in Simulink with the following Simulink blocksets: Aerospace, Simscape (to include the sub-blocksets SimPowerSystems, SimElectronics, and SimMechanics), and Digital Signal Processing. The implementation and modification of models using the sub-blocksets of Simscape require the appropriate license; however, only the Simscape license is required to “run” a model utilizing these sub-blocksets.

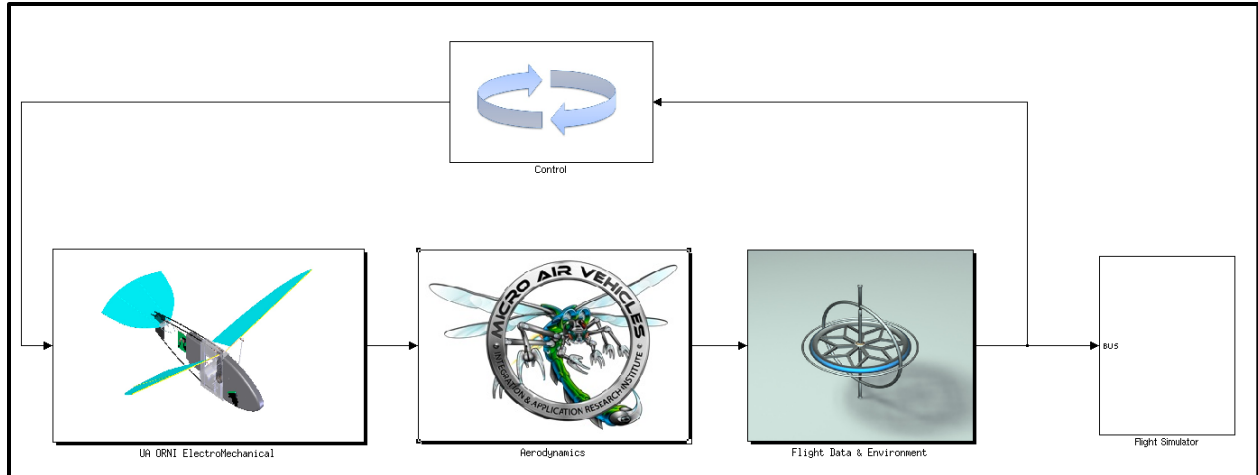


Figure 21: Simulink Ornithopter Model and Engineering Description

3.2.1 Coordinate System Definition

The Ornithopter wing and body movement is comprised of a fixed reference frame in the global coordinate system and the inertial frame of reference $I = \{I_X, I_Y, I_Z\}$. The vehicle body frame $R = \{R_X, R_Y, R_Z\}$, is obtained by rotating I with respect to the global frame. The positive direction of the body frame component R_X extends in what is considered the vehicle's normal forward-flight direction, while the positive R_Y coordinate extends out the left wing. The wing frame is rotated about R to obtain the flapping stroke (sweep), wing pitch (feather), and deviation as represented by the three Euler angles, ϕ , η , and θ , respectively. See Figure 22, which shows the stroke angle, the primary kinematic angle of interest when the wing is assumed rigid.

This model utilizes the quasi-steady aerodynamic model discussed in Section 2. The wing element model follows the coordinate system described in Figure 1, where the R_Y coordinate extends in the forward direction and R_X coordinate extends to the right. As a result, the simulation model frame is transformed to the aerodynamic frame to calculate the aerodynamic forces. The aerodynamic forces are then transformed to align with the global frame.

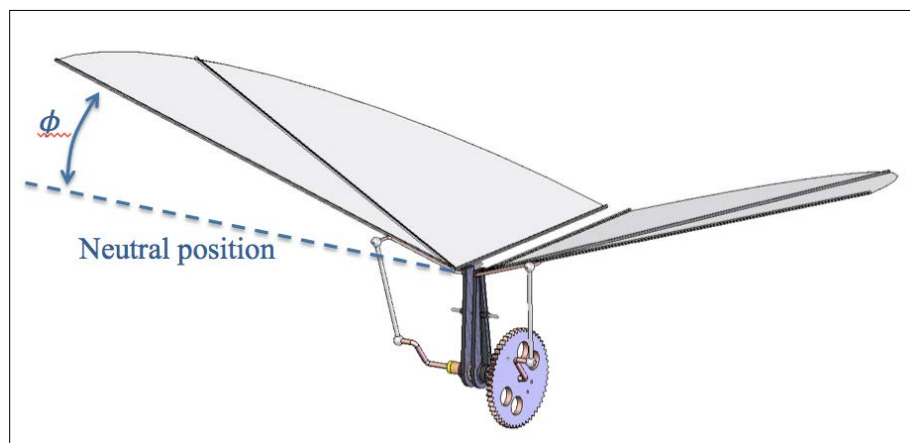


Figure 22: Ornithopter Coordinate System

3.2.2 Vehicle Aerodynamics

The aerodynamic modeling builds upon the quasi-steady, blade-element aerodynamics model developed under the FSI effort. The blade-element model was expanded to allow for prescribed deformation in the wing geometry, as described in the previous section. The wing aerodynamic forces are calculated with respect to the body-frame, center-of-geometry and the wing hinge attach point frame. The resultant aerodynamic forces and moments are then generated from the aerodynamics model with respect to the wing hinge frame. These forces and moments are then coupled with the mechanical portion of the model discussed in the following sections.

The flight-surface aerodynamic geometries are each represented as a discretized wing broken into rectangular chord sections, as shown in Figure 23. Each wing section is defined about a node point in the aerodynamic frame using three parameters: the section chord length, thickness, and relative angle. Each node point represents both the geometric center and center of mass for the rectangular wing section. The relative angle prescribed to each wing section is represented by the three Euler angles, ϕ , η , and θ respectively, and follows the descriptions in Section 3.2.1. Each wing section relative angle is with respect to the previous wing section moving in the spanwise direction from the wing root to the wing tip. The relative angles allow for prescribed deformations to be applied to the wing. The resultant position of each wing section is the accumulation of the wing kinematics and the proceeding wing section (inboard) Euler angles. The deformation model requires a pre-defined data set of wing node geometric behaviors correlating to the wing kinematics and their derivatives. The time varying wing deformation is interpolated from the prescribed values to determine the wing shape for the blade element aerodynamics model at each time step.

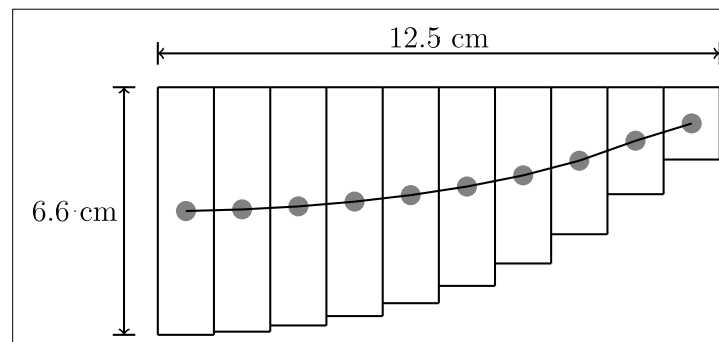


Figure 23: Wing Discretization and Node Points for Aerodynamic Calculations

The prescribed deformation can be used to duplicate experimentally obtained deformation data in lieu of using computationally intensive aeroelastic solvers. This data can be used in the model simulation to describe the wing shape for a given kinematic acceleration and velocity at the wing root. The prescribed deformation may also be used to describe stability and control surfaces as a single aerodynamic surface. The control surface deflection is then modeled as the relative angle at the node point, which coincides with the control surface hinge. This method was utilized to model the vertical and horizontal stabilizers along with the rudder and elevator, respectively, as shown for the vertical stabilizer and rudder in Figure 24. In the rudder aerodynamic model, the four leading (forward) elements remain rigid and aligned with the vertical stabilizer body, while the three aft elements are aligned with the rudder surface. Element 5 is defined as the deflection node and deflection angles are reflected in this node with respect to element 4. Elements 6 and 7 remain rigid with respect to element 5, creating the aerodynamic

deflection surface, as shown in Figure 24 (right). The horizontal stabilizer and elevator control surface are modeled using the same approach.

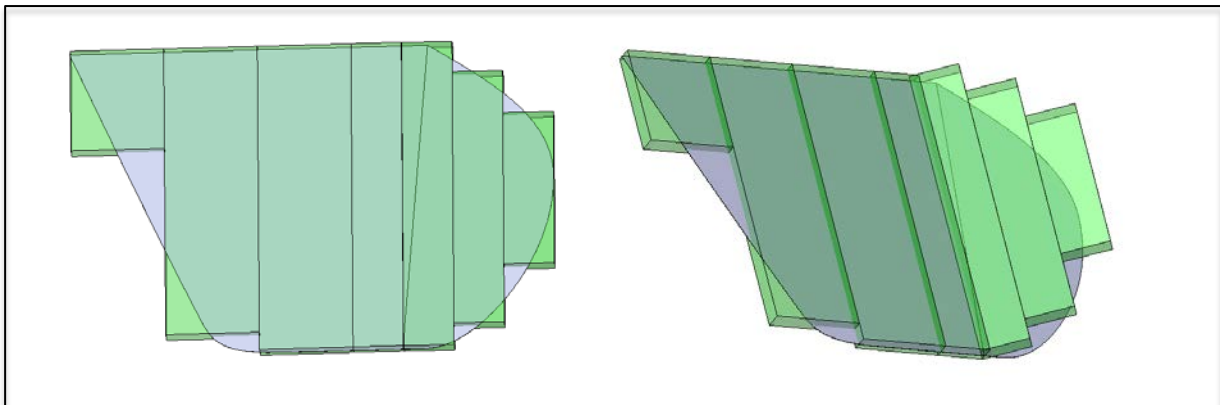


Figure 24: Vertical and Horizontal Stability and Control Surface Aerodynamic Geometry Description

The aerodynamics model was integrated into the Simulink Ornithopter using two different methods: the Matlab Function block and native Simulink blocks. The Matlab Function block utilizes Matlab code to generate embeddable C code for the Simulink coder. The Matlab code was also converted to native Simulink model by replacing the Matlab function calls with mathematically equivalent Simulink blocks from the Simulink library. The solver and time step in both approaches utilize the defined Simulink solver parameters.

3.2.2 Vehicle Mechanical and Electrical Engineering Description Modeling

The mechanical system is modeled utilizing Simulink SimMechanics first-generation blockset and SimElectronics. SimMechanics and SimElectronics are subsets of the Simulink Simscape blockset family. Simscape utilizes a multi-domain physical signal network rather than the numerical or mathematical operation signals utilized in the standard Simulink model sets. The physical signals allow the user to represent physical relationships between components directly; Simscape automatically constructs the system of equations that characterize the behavior of the system¹⁷. The ornithopter electromechanical model (shown in Figure 25) is comprised of the ornithopter fuselage model, brushed electrical motor, lithium cell battery, wing and corresponding mechanics, and the tail assembly.

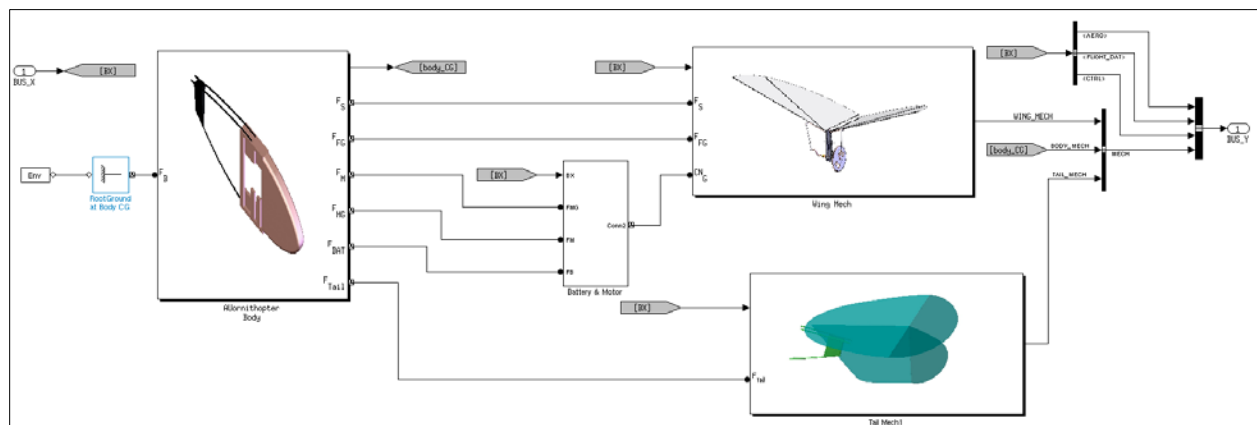


Figure 25: Simulink Ornithopter Electromechanical Model

The Ornithopter mechanical bodies such as the fuselage, batteries, gears, wings, etc...are modeled utilizing the SimMechanics Body block. The Body blocks represent rigid bodies described by their mass properties, including the moment of inertia tensors, center-of-gravity, the body coordinate frame, and attached body coordinate frames. The body's translational acceleration is influenced by the body's moment of inertia, which can be estimated from the body 3D models using CAD tools such as SolidWorks. Quantitative data related to the Body block such as the body translational information, forces, and moments can be accessed via the model Sensor block, as shown in the Ornithopter fuselage Simulink model in Figure 26 and the right-wing model in Figure 27.

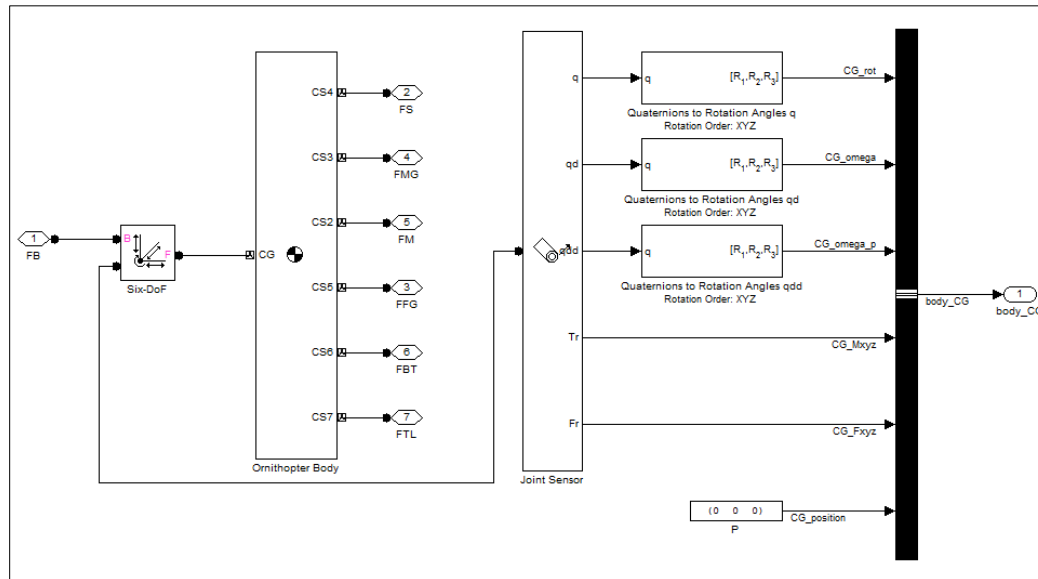


Figure 26: Simulink Mechanical Description of the Ornithopter Body

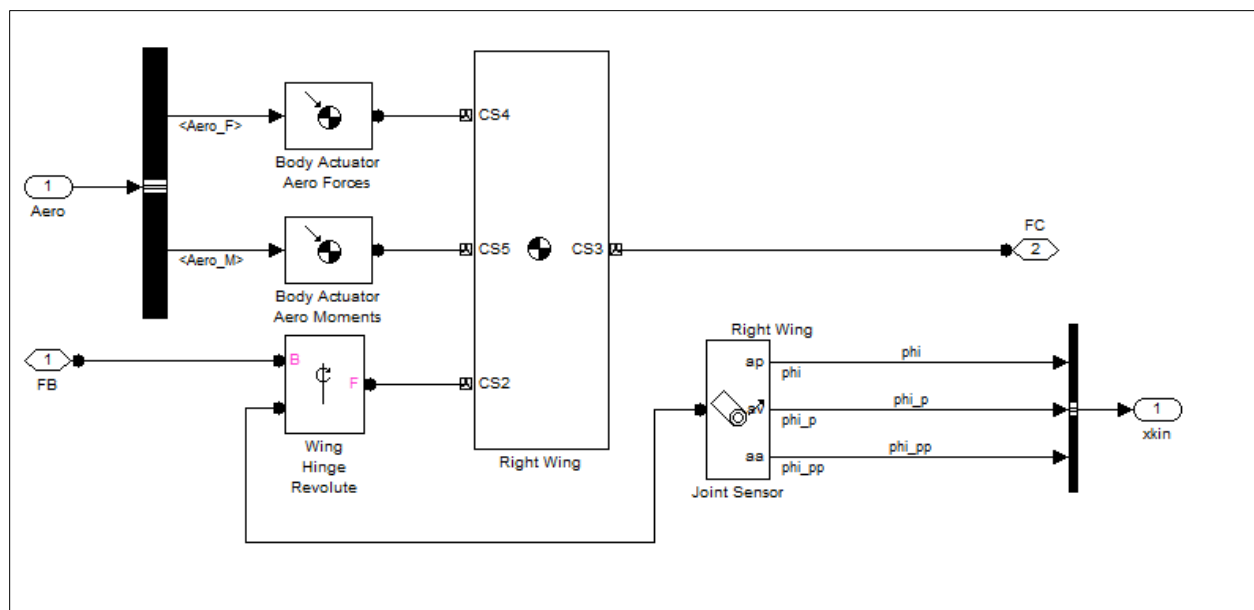


Figure 27: Simulink Mechanical Description of the Ornithopter Right Wing

The Ornithopter CAD models provided by the University of Arizona were converted into STL format and entered in to the SimMechanics Body block to generate a three-dimensional visual output of the model. The 3D visualization (shown in Figure 28) is continuously updated during the simulation to reflect the vehicle assembly, joint actuation, and related mechanical motion.

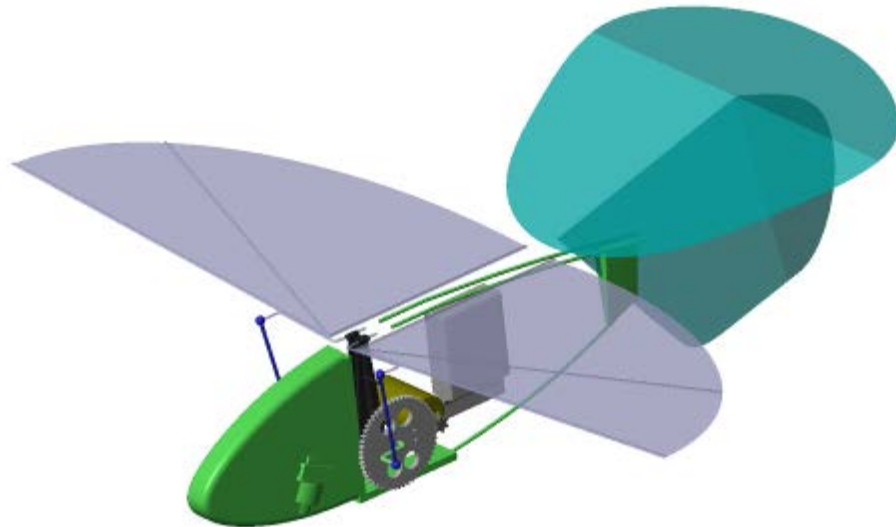


Figure 28: SimMechanics Visualization of the Ornithopter

The Ornithopter wing mechanical drive model is started at the follower gear (large gear), which is constrained by the motor gear. The SimMechanics implementation of the wing mechanical sub-system is shown in Figure 29. An electrical load is applied to the motor, which in turn produces a drive torque on the motor gear and constrained follow gear. The follow gear has an attached crank rod (Figure 30) with spherical joints connected to a left and right actuation rod, which in turn are connected to the left and right wing. The wings are constrained at the hinge joint attaching them to the fuselage, closing the mechanical loop. The wing kinematics, such as the Euler angles, velocity, and acceleration or measured at the wing hinges and provided to the aerodynamics model from, which the resultant aerodynamic loads are calculated. The aerodynamic loads are coupled to the mechanical model as actuation forces acting on the wing hinge. This coupling transmits the aerodynamic loads into mechanical torque actuating the wing and inertial forces acting on the vehicle body.

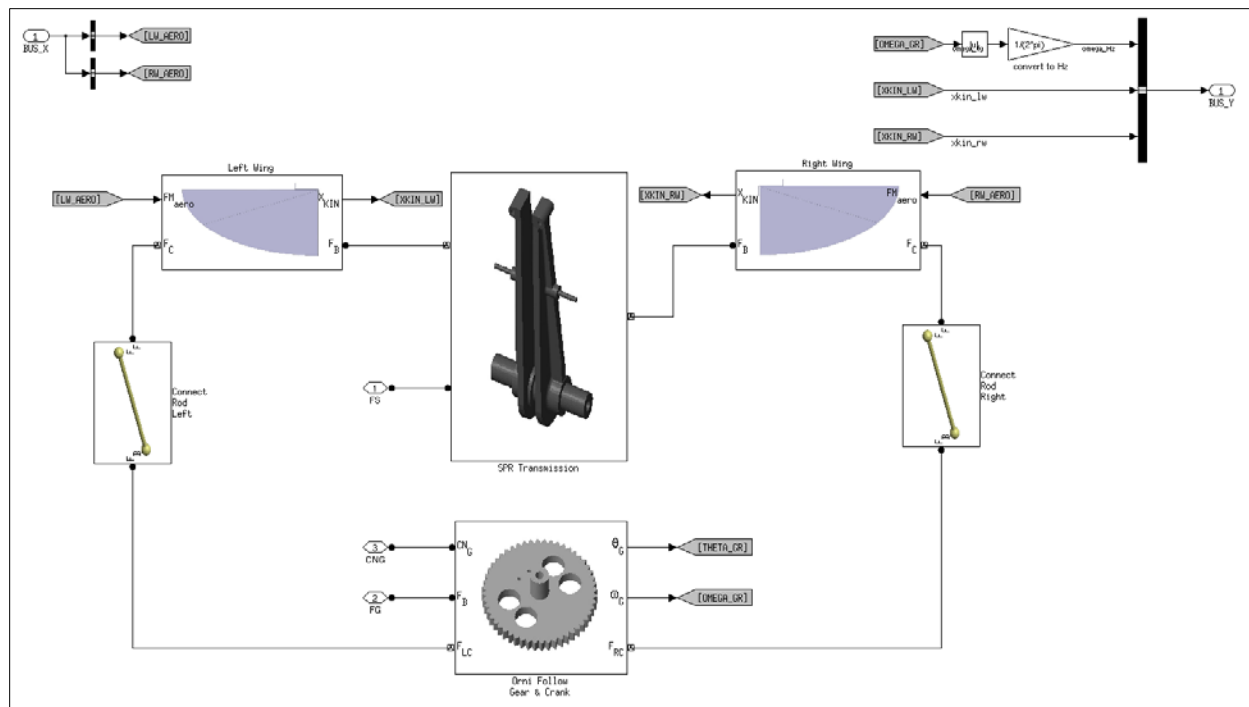


Figure 29: SimMechanics Implementation of the Ornithopter Wing Actuation

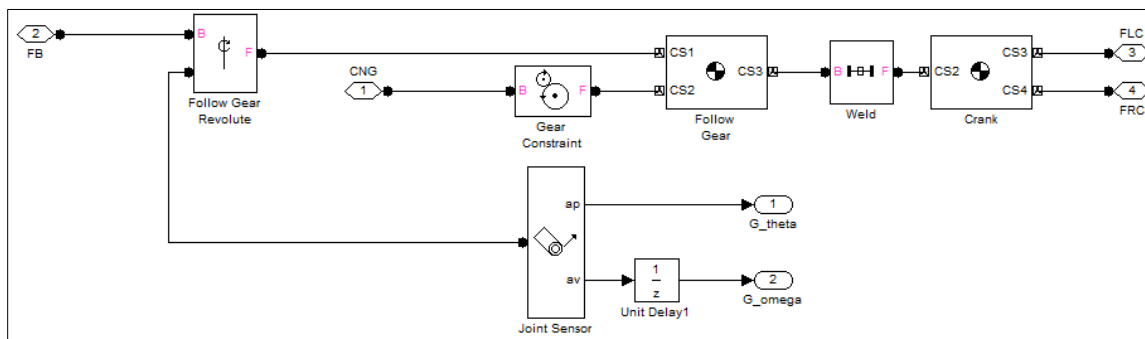


Figure 30: SimMechanics Model of the Follow Gear and Crank Rod Wing Drivers

Massless, time-varying distance drivers represent the wing connector rods, rather than Body models used in the other mechanical descriptions. This alleviates the restrictive spherical-to-spherical constraints that were found to be sensitive to small perturbations in the wing motion. The spherical joint sensitivity used in the Body joint model resulted in higher computational cost due to the small step sizes required by the solver. By replacing the spherical-to-spherical joint body with the distance driver model (Figure 31), the sensitivity was eliminated without sacrificing accuracy in realized kinematics motion. Mechanical loads resulting from the wing and aerodynamic effects are still translated to the follow gear through the distance driver as if it were a Body.

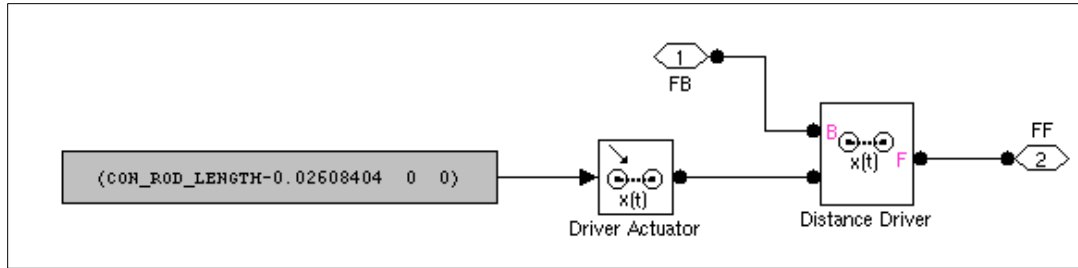


Figure 31: SimMechanics Model of the Wing Connector Rod

The ornithopter tail assembly shown in Figure 32 is comprised of four mechanical sub-assemblies: the vertical stabilizer surface, horizontal stabilizer surface, rudder flight control surface, and the elevator flight control surface. Each tail assembly component is modeled with the SimMechanics Body block.

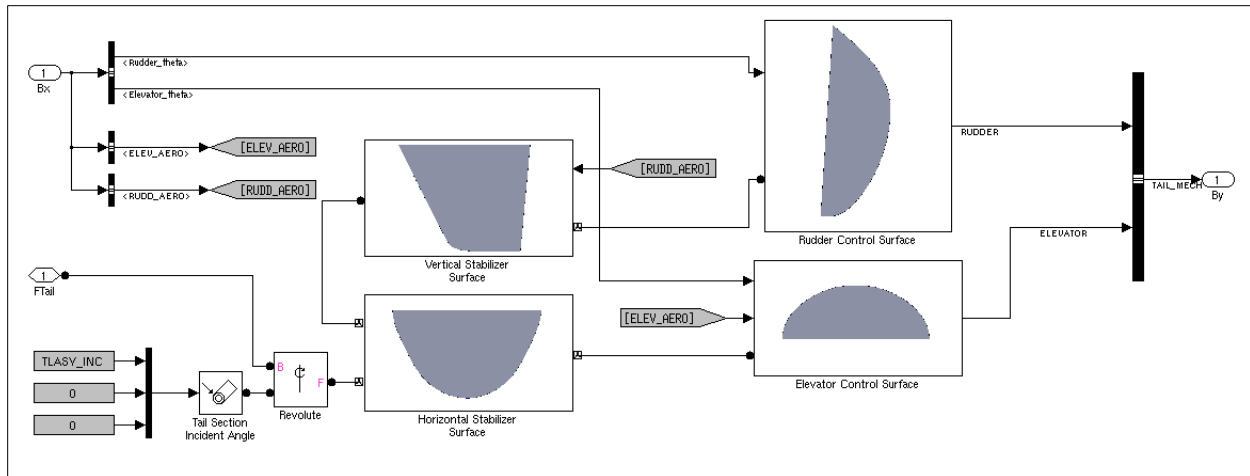


Figure 32: SimMechanics Model of the Ornithopter Tail Assembly

The flight control surfaces are connected to their corresponding stabilizer surfaces through a joint revolute block. The actuation of the flight control surfaces are restricted using a rate limiter, saturation, and transfer function to mimic the deflection rate, maximum deflection angles, and response, respectively. The joint sensor reports the control surface deflection Euler angle and its derivatives. These angles are conveyed to the aerodynamic model, which calculates the reactionary aerodynamic forces from the stabilizer and control surface combination. The reactionary forces and moments are then applied to the mechanical body at the hinge revolute frame. This method is applied to both the vertical and horizontal stabilizer and control surfaces, as shown in Figure 33.

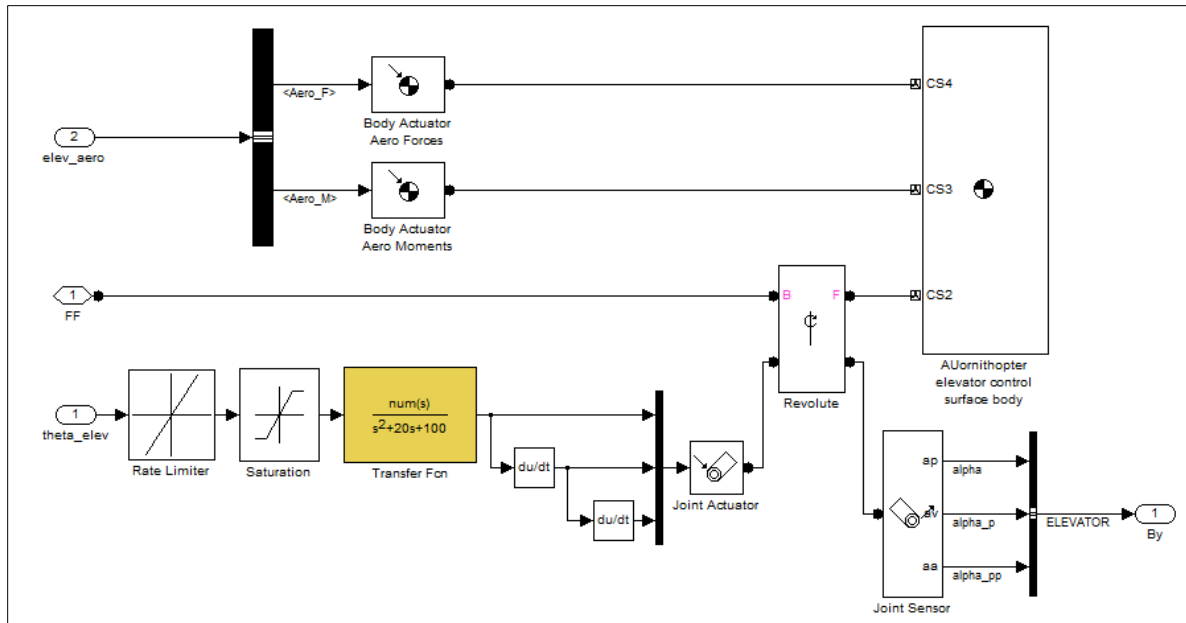


Figure 33: SimMechanics Model of the Ornithopter Elevator Control Surface

3.2.4 Pilot in the Loop Simulation

The Ornithopter Simulink model and engineering description is designed to allow for real-time, pilot-in-the-loop testing. Utilizing the Simulink Aerospace blockset Flight Simulator Interface for animation, the Ornithopter model can be visualized in a simulated, real-world environment using the open-source flight simulator package called FlightGear¹⁸. The FlightGear visualization provides no feedback to the Simulink model; all aerodynamic forces and motion are calculated utilizing the outputs from the models described and the Simulink Aerospace 6DOF blockset. Simulink is not inherently real-time; to adjust for this, a real-time delay based on the current CPU time is added to the simulation to artificially slow the modeling down. As a result of running the simulation in real-time mode, the user should be aware of the trade-off between decreasing the Simulink solver step-time so that the calculation time (or Simulink time) does not exceed real-time. Figure 34 and Figure 35 show the visualization of the Ornithopter using the FlightGear v2.8.

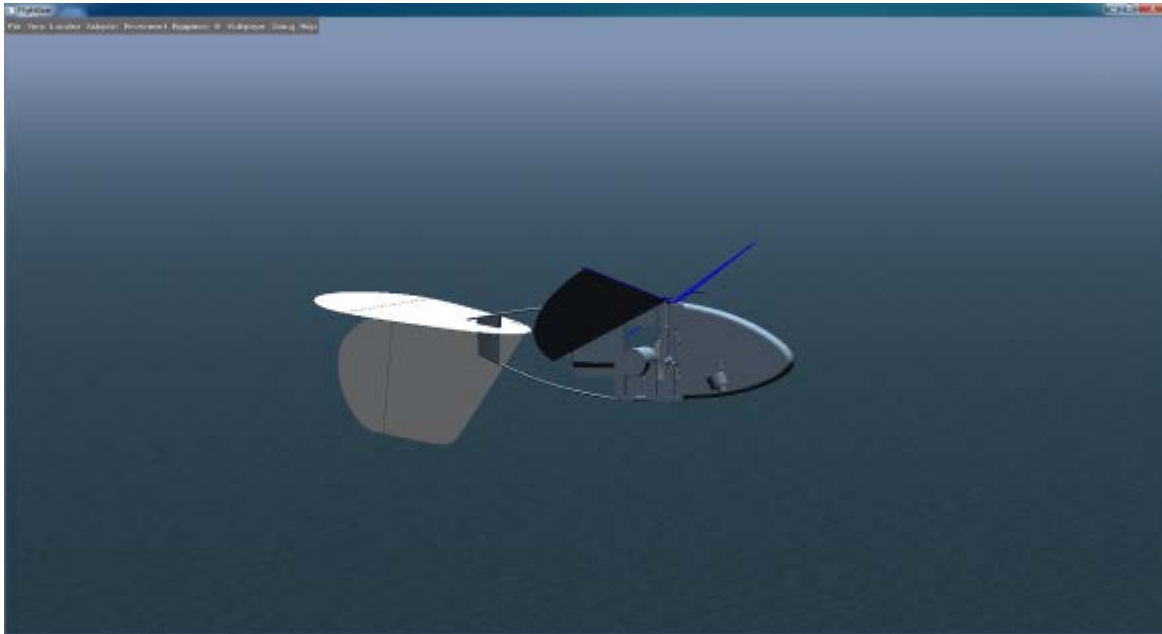


Figure 34: FlightGear Simulation of Ornithopter over Water

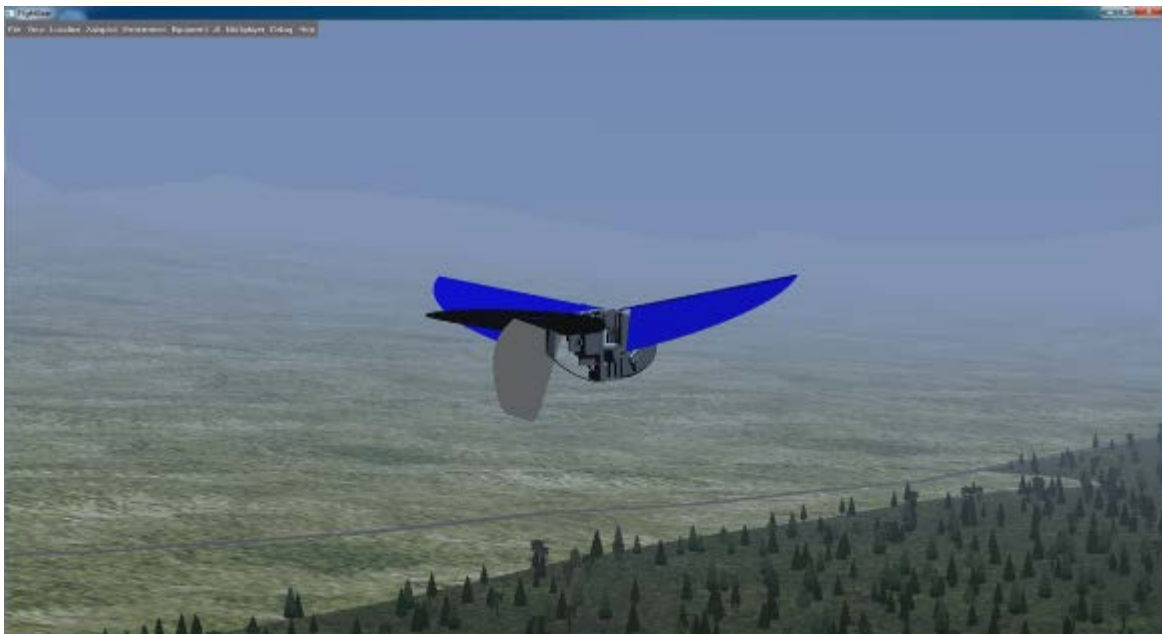


Figure 35: FlightGear Simulation of Ornithopter over Land

3.3 Application of Simulink Model

The model representation of the UA Ornithopter in Simulink was carried out as a conceptual investigation for performing quantitative technical assessment with FSI components. The descriptive Simulink engineering implementation helped determine best-practice methods, from an end-users perspective, for multi-physics modeling. The concepts and practices used in this development were used to identify shortcomings of the FSI M³CT distributed computing environment interface.

The configuration specifications of the UA Ornithopter are given in Table 6. Results obtained from the Simulink simulations of the UA Ornithopter include such quantities as vehicle position, kinematics, aerodynamic and mechanical power, and battery status. These will be reported in the in-house work unit related to this contractual effort.

Table 6: UA Ornithopter Specifications

Attribute	Value
Mass (g)	21.7-23.3
Speed range: min-max (cm/s)	300 - 500
Endurance (s)	240+
Wingspan (cm)	25
Planform area (cm ²)	137
Aspect ratio (-)	4.6
Dihedral angle (rad)	0.105
Flapping frequency: max (Hz)	25
Flapping amplitude (rad)	1.26

4.0 CONCLUSIONS

A multi-disciplinary optimization study for a closed-loop, flapping wing MAV was performed in a distributed computing environment using the service-oriented framework SORCER. As part of the integration, a graphical user interface implementation of a service requestor (M³CT) was coupled with SORCER. The M³CT facilitates the research process by providing a tool for accessing the distributed computing environment from a high-level perspective, while mitigating the tedious process of managing multiple optimization studies. This methodology promotes the use of the distributed framework by cloaking its implementation and inner workings. This approach encourages the efficient reuse of existing models/applications regardless of their native development environment.

The required applications for this research were originally written in C, FORTRAN, and Matlab®. The tools were coupled in the SORCER environment by first converting them to Matlab stand-alone executables and then deploying each as a separate service provider. The first service provider contains the quasi-steady, blade-element method, trim and linearization, LQR controls synthesis, and gust models, along with an algorithm for providing gradients upon request. The second executable converted to a service provider incorporates the method of moving asymptotes for performing optimization. Each service provider's sub components may be modularized and re-introduced as separate service providers within SORCER to provide more versatility to the optimization design and provide encouragement to others who wish to expand upon the research. This modification would help exploit the salient advantages of performing aerospace research and design in a distributed computing framework.

The test cases presented in this study evaluated the kinematics, control, and wing shape optimization for the FWMAV, with consideration for reduction in aerodynamic power. The optimization was performed under the constraints applied to both the orbital displacement caused by gust disturbances and the resultant peak control power in the case of the closed-loop optimization. Using the MMA optimization method, viable design variables were successfully demonstrated from the different disciplines: aerodynamics (geometry), kinematics, and controls. It was evident throughout analysis of the test cases that the wing distribution optimization was complimented by optimal design changes in the kinematic parameters. Additionally, the linear quadratic cost function coefficient was successfully used to help balance the dynamic pair of opposing control authority and spherical displacement constraints.

The test cases analyzed thus far have provided encouragement to explore other design parameters for optimization. Future cases will likely consider wing radius, baseline values related to controller assigned kinematics, and the split-cycle parameter (δ). Additional modifications in the future will include developing autonomous methods for identifying appropriate optimization step-sizes to balance computational overhead while avoiding oscillatory convergence. Future changes will also include exploring alternative statistical gust distributions for generating more representative performance values for a wide range of disturbances.

The FSI software was extended in Simulink to build an engineering description of the University of Arizona 25cm Ornithopter. The system-level engineering model encompasses vehicle aerodynamics, mechanics, and electronics, and qualitatively links these functions to FlightGear to provide a real-time, pilot-in-the-loop simulation capability. This effort was a

proof-of-concept study in support of the NATO AVT Task Group 184, “Characterization of Bio-Inspired Micro Air Vehicle Dynamics.”

5.0 REFERENCES

1. Sobolewski, M., "SORCER: Computing and Metacomputing Intergrid," *International Conference of Enterprise Information Systems*, Barcelona, Spain, June 12-16, 2008.
2. Berman, G., and Wang, Z., "Energy-Minimizing Kinematics in Hovering Insect Flight," *Journal of Fluid Mechanics*, Vol. 582, pp. 153-168, 2007.
3. Bhatia, M., Patil, M., Woolsey, C., Stanford, B., and Beran, P., "LQR Controller for Stabilization of Flapping Wing MAVs in Gust Environments," AIAA 2012-4867, August 2012.
4. Doman, D. B., Oppenheimer, M. W., and Sigthorsson, D.O., "Wingbeat Shape Modulation for Flapping-Wing Micro-Air-Vehicle Control During Hovering," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No.4, pp. 724-739, 2010.
5. Stanford, B., Kurdi, M., Beran, P., and McClung, A., "Shape, Structure, and Kinematic Parameterization of a Power-Optimal Hovering Wing," *AIAA Structures, Structural Dynamics and Materials Conference*, Orlando, FL, April 12-15, 2010.
6. Lancaster, P., and Rodman, L., *Algebraic Riccati Equations*, Oxford Science Publications, Oxford University Press, New York, 1995, Chapters 12-15.
7. Fidowski, K. J., Engelsen, F., Willcox, K. E., and Kroo, I. M., "Stochastic Gust Analysis Techniques for Aircraft Conceptual Design," *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia Canada, September 10-12, 2008.
8. Svanberg, K., "The Method of Moving Asymptotes – A New Method for Structural Optimization," *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 2, pp. 359-373, 1987.
9. Sun Microsystems, *Jini Architecture Specification*, v. 1.2, Sun Microsystems, Inc., Palo Alto, 2001.
10. Sobolewski, M., "Exerted Enterprise Computing: From Protocol-Oriented Networking to Exertion-Oriented Networking," *On the Move to Meaningful Internet Systems: OTM 2010 Workshops, Lecture Notes in Computer Science*, Vol. 6428/2010, pp. 182-201, 2010.
11. Sobolewski, M., "Metacomputing with Federated Method Invocation", *Advances in Computer Science and IT*, Edited by M. Akbar Hussain, In-Tech, intechweb.org, ISBN 978-953-7619-51-0, pp. 337-363, 2009.
12. Stanford, B., Beran, P., and Patil, M., "Optimal Flapping Wing Vehicle Dynamics via Floquet Multiplier Sensitivities," *Journal Guidance, Control, and Dynamics*, Vol. 36, No. 2, pp. 454-466, 2013.
13. Mamoud, Q.H., "Getting Started With JavaSpaces Technology; Beyond Conventional Distributed Programming Paradigms," *Developer Technical Articles and Tips* [online database], URL: <http://java.sun.com/developer/technicalArticles/tools/JavaSpaces/> [cited 01 June 2012].
14. Beran, P., Parker, G., Snyder, R., and Blair, M., "Design Analysis Strategies for Flapping Wing Micro Air Vehicles," *International Forum on Aeroelasticity and Structural Dynamics*, Stockholm, Sweden, June 18-20, 2007.
15. Bryson, D.E., Stanford, B.K., McClung, A.M., Sims, T.W., Miranda, J.L., Beran, P.S., and Parker, G.H., "Multidisciplinary Optimization of a Hovering Wing with a Service-Oriented Framework and Experimental Model Validation," AIAA 2011-1133, January 2011.

16. Hedrick, T. L., and Daniel, T. L., "Flight Control in the Hawkmoth *Manduca sexta*: The Inverse Problem of Hovering," *Journal of Experimental Biology*, Vol. 209, pp. 3114-3130, 2006.
17. MathWorks, *Simulink User's Guide*, The MathWorks, Inc., Natick, MA, 2012.
18. FlightGear.org. Dreyer, T., Olson, C., Talsma, D., 2012.
<http://www.flightgear.org/about/credits/>

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

<u>Symbol</u>	<u>Definition</u>
A	Discrete linear time-invariant system model system matrix
A_ϕ, A_θ	Amplitude scaling factors for wing stroke and wing sweep motion
B	Discrete linear time-invariant system model control coefficient matrix
$d\mathbf{x}^{T+t_0}$	State sensitivity vector at each flapping cycle
$f_i(x)$	Generic constraint function
$f_o(x)$	Generic objective function
f_G	Gust frequency (Hz)
I	Inertial coordinate frame
I_X, I_Y, I_Z	Cartesian X, Y, and Z components of the inertial coordinate frame
K	Controller gain matrix
k	Power constraint coefficient
K_ϕ	Euler angle function between sinusoidal ($K_\phi = 0$) and a rectangular waveform ($K_\phi = 1$)
K_η	Euler angle function between sinusoidal ($K_\eta = 0$) and a triangular waveform ($K_\eta = 1$)
L_i, U_i	Lower and upper asymptotes
m	Subscript for wing motion angle magnitude
N_θ	Wing sweep frequency to flapping frequency deviation factor
N_G	Number of gust vectors used in optimization
N_X	Number of design parameters used in optimization
N_{IC}	Number of design optimization iterations
o	Subscript for wing motion offset (rad)
P	Solution of algebraic Riccati equation
P	Cycle average aerodynamic power (W)
Q	Controller weight matrix for LQR synthesis
q	Vector containing the wing kinematic variables
R	Rotational body coordinate frame
R_X, R_Y, R_Z	Cartesian X, Y, and Z components of the rotating body coordinate frame
s	Subscript for wing motion phase shift (rad)
T	Superscript for the matrix transpose operation
W_n	Rotational wing coordinate frame where n represents the wing number
W_{nX}, W_{nY}, W_{nZ}	Cartesian X, Y, and Z components of the wing coordinate frame
x	Current position state vector
X_{kin}	Prescribed kinematic parameters
β	Time dependent frequency (rad/s)

List of Symbols, Abbreviations, and Acronyms (continued)

<u>Symbol</u>	<u>Definition</u>
δ	Split cycle control parameter
η	Wing feathering (pitch) angle (rad)
$\hat{\eta}_m, \hat{\eta}_o$	Time dependent wing feathering magnitude and offset coefficients
ω	Flapping frequency (rad/s)
ϕ	Wing stroke (flapping) angle (rad)
ρ	Linear quadratic controller cost function coefficient
θ	Wing sweep angle (rad)

<u>Abbreviation</u>	<u>Definition</u>
LQR	Linear Quadratic Regulator
SORCER	“Service-ORiented Computing EnviRonment”
QTA	Quantitative Technology Assessment
M ³ CT	Model Based Computational Tool”
MPP	Multi-physics Prototyping”
FSI	Flapping Sciences Integration”
FWMAV	Flapping Wing Micro Air Vehicle
SBR	Span-Break Ratio
MMA	Method Of Moving Asymptotes
MAV	Micro Air Vehicle