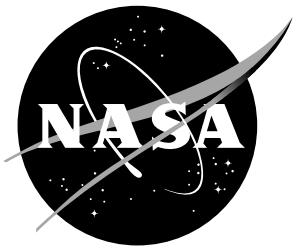
NASA/TM–20230009523



A Framework for Evaluating Distributed Electric Propulsion on the SUSAN Electrofan Aircraft

Nicholas C. Ogden
Armstrong Flight Research Center, Edwards, California
Andrew Patterson
Langley Research Center, Hampton, Virginia

July 2023

NASA STI Program Report Series

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

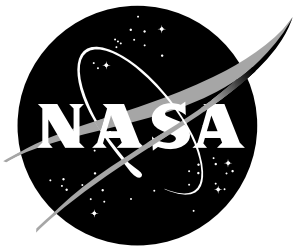
Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov
- Help desk contact information:

https://www.sti.nasa.gov/sti-contact-form/ and select the "General" help request type.

NASA/TM–20230009523



A Framework for Evaluating Distributed Electric Propulsion on the SUSAN Electrofan Aircraft

Nicholas C. Ogden
Armstrong Flight Research Center, Edwards, California
Andrew Patterson
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

July 2023

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Abstract

This work presents a framework for evaluating models and algorithms for Distributed Electric Propulsion (DEP) on the SUSAN Electrofan Aircraft. Throughout the development of the SUSAN aircraft, the performance of various configurations of the aircraft will need to be analyzed. However, the static behavior alone is not sufficient to describe the performance of these configurations. Therefore, simulation with fully integrated subsystem models is required. The proposed framework considers the vehicle aerodynamic, propulsion, and control subsystems. The presented framework automatically generates control laws for any vehicle configuration in response to changes in these subsystems. To compare these different vehicle configurations, various time and frequency domain performance metrics are compared.

Contents

List of Figures

Nomenclature

Abbreviations

DEP	=	Distributed Electric Propulsion
EMTAT	=	Electrical Modeling and Thermal Analysis Toolbox
PI	=	Proportional-Integral feedback controller
PID	=	Proportional-Integral-Derivative feedback controller
SISO	=	Single Input Single Output
SUSAN	=	SUbsonic Single Aft eNgine
TCM	=	Transport Class Model
T-MATS	=	Toolbox for Modeling and Analysis of Thermodynamic Systems

A	=	State matrix
B	=	Input matrix
b	=	Wing span
C	=	Aerodynamic coefficient
$\bar{c}$	=	Wing mean aerodynamic chord
d	=	Distance
$\boldsymbol{dT}$	=	Differential thrust vector
dT	=	Differential thrust
F	=	Force and moment vector
g	=	Earth's gravitational constant
h	=	Altitude
$\dot{h}$	=	Altitude rate
I	=	Moment of inertia
L	=	Reference length
M	=	Moment
(m, n)	=	Matrix dimensions
$\bar{q}$	=	Dynamic pressure
S	=	Reference surface area
$\boldsymbol{T}$	=	Thrust vector
T	=	Thrust
$\boldsymbol{u}$	=	Input vector
V	=	Airspeed
W	=	Weighting matrix
w	=	Weight
$\mathbf{X}$	=	State vector
$\dot{\mathbf{X}}$	=	State derivative vector
(p, q, r)	=	Body frame rotational rates x, y, and z axes respectively
$(\dot{p}, \dot{q}, \dot{r})$	=	Body frame rotational acceleration x, y, and z axes respectively
α	=	Angle of attack
β	=	Angle of sideslip
γ	=	Flight path angle
δ	=	Change
σ	=	Course angle
ω	=	Rotational velocity

(ϕ, θ, ψ) = Vehicle orientation given by roll, pitch, and yaw angles respectively

Subscripts

a	=	Adjustment
cmd	=	Command
des	=	Desired
err	=	Error
f	=	Failure
FF	=	Feed forward
m	=	Measured
min	=	Minimum limit on quantity
max	=	Maximum limit on quantity
PID	=	Proportional-integral-derivative
req	=	Required
s	=	Saturation
tf	=	Turbofan located at tail of aircraft
tot	=	Total
wf	=	Wingfans attached to electric motors
(xx,yy,zz)	=	Indicate a value is associated with the body frame x, y, and z axes respectively

Superscripts

$\top$	=	Transpose
$\dagger$	=	Pseudoinverse

1 Introduction

The Federal Aviation Administration (FAA) identifies jet fuel combustion as responsible for 97% of US aviation CO2 emissions [1]. NASA's SUbsonic Single Aft eNgine (SUSAN) Electrofan concept is being developed to reduce fuel usage. The target fuel consumption reduction is 50%. To achieve this goal, the SUSAN concept implements several novel technologies including a Distributed Electric Propulsion (DEP) system. The DEP system consumes electric power to produce thrust via an array of electric motors with ducted fans distributed across the span of the aircraft's wings. This system allows the turbofan to be sized for cruise, instead of takeoff conditions, and operate at peak efficiency throughout the flight envelope [2]. The electrically powered wingfans then use battery power to generate the extra thrust needed for takeoff.

The SUSAN project is also investigating reducing the size of the vertical tail to reduce drag and therefore reduce fuel consumption [3]. This increase in efficiency comes at the cost of reducing the aircraft's lateral stability. To improve lateral stability, the SUSAN project is considering using the distributed electric propulsion system to help actively stabilize the lateral dynamics of the aircraft. This document introduces a framework which enables integrated performance evaluation of various SUSAN system configurations as part of future trade studies.

The proposed framework is implemented in simulation with models for system kinematics, aerodynamics, control, propulsion, and electrical subsystems. The framework uses a series of algorithms that enable closed loop simulation by linearizing the system and then tuning a baseline control structure to user performance objectives. Therefore, different combinations of aircraft configurations can be compared with similar design procedures. This design approach allows the closed

loop performance of two different aircraft configurations to be compared directly. While there is no guarantee that the tuning method will produce the same performance, keeping the design procedures the same reduces the likelihood of over tuning the controller to a specific configuration. Over-design of the controller will bias the system comparison based on the amount of effort spent by the designer while the presented method is biased by the optimization procedure.

The framework will be validated with test cases where subsystems are modified, and integrated system performance is evaluated with the automatically tuned control parameters. These modifications could be changes to the subsystem parameters, algorithms, and dynamic models. Examples of subsystem changes would be updating the aerodynamic database to reflect the aircraft having a reduced vertical tail size or implementing an online optimization for control allocation. The framework can handle substantial changes to the subsystems provided they can still be linearized. We would expect the framework to perform poorly in subsystems that are modified to substantially rely on non-local information. For certain complex subsystems, designer-provided linear models may be required.

The specific evaluation cases for the framework are chosen to investigate possible benefits of DEP. The first evaluation case integrates the propulsion control system and flight control system to demonstrate the benefit of integrated flight control and propulsion control systems. The second evaluation case explores reducing the aircraft's vertical tail size to demonstrate the framework's ability to achieve comparable closed loop performance across different aircraft configurations. The third evaluation case implements two control allocation schemes to demonstrate the framework's ability to evaluate different methods of utilizing the actuator redundancy introduced by SUSAN's DEP design. This work applies the framework to these evaluation cases to demonstrate its capability.

In Section 2, background information on the baseline models used in the SUSAN simulation is covered. In Section 3, the architecture and algorithms used for the flight control system are covered. In Section 4, the propulsion system models and linearization procedures are described. In Section 5, the automatic tuning algorithm used by the control framework is shown. In Section 6, results from the evaluation cases are presented. In Section 7, the findings of this work are summarized.

2 Background

The presented framework is developed around a Simulink® model of the SUSAN aircraft. The SUSAN model is adapted from the Transport Class Model (TCM) developed by NASA Langley Research Center [4]. The baseline SUSAN model modifies the TCM to use its own aerodynamic databases, control architecture, and input structure. The aerodynamic database is produced in OpenVSP[5]. The input structure is updated to support SUSAN's DEP system.

SUSAN's propulsion system is a parallel hybrid electric system employing DEP. More specifically, a single turbofan produces both electrical power, to charge the system's batteries, and a portion of the system's thrust. The DEP system consists of electric propulsors distributed along the aircraft's wingspan which consume the electrical power stored in the batteries to produce thrust [6]. The exact number, design, and location of these propulsors is still being studied. This work is done using the SUSAN V3 configuration, shown in Fig. 1, which features 16 electric propulsors (eight per side) mounted on the underside of the wing [2].

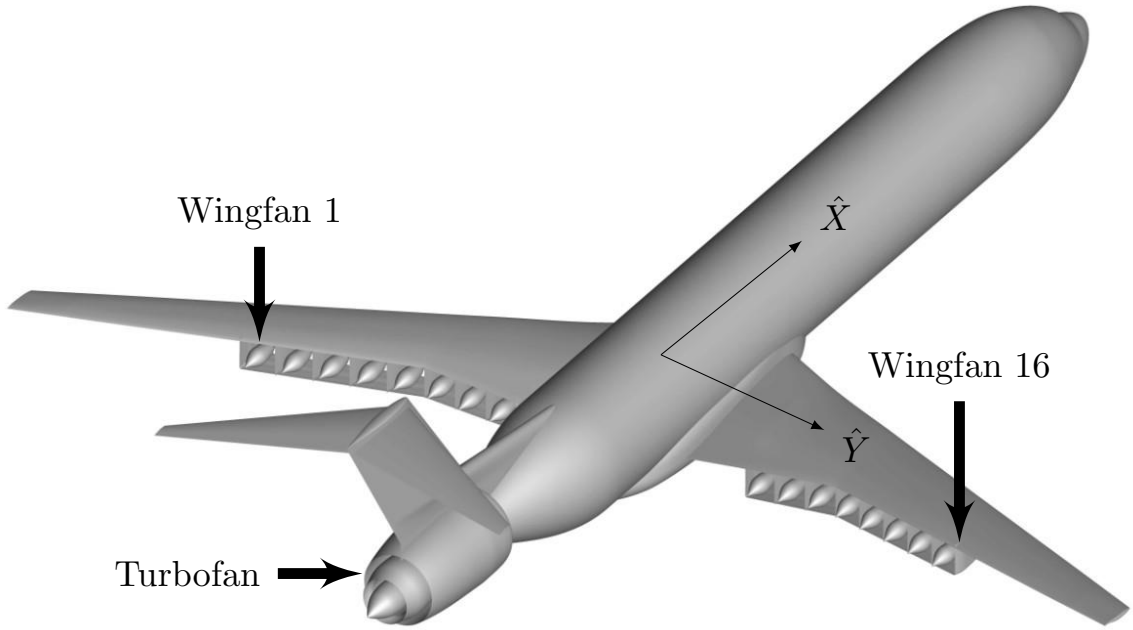


Figure 1: Model of the SUSAN V3 configuration from Ref. [3]. Labels have been added to match the notation in this work.

In addition to the aerodynamic update, the framework is tested with a modified propulsion model. This propulsion model is developed using NASA's Toolbox for Modeling and Analysis of Thermodynamic Systems (T-MATS) and Electrical Modeling and Thermal Analysis Toolbox (EMTAT) [7, 8]. These high-fidelity tools iteratively solve for the electrical and gas dynamics of each propulsor.

Lastly, the limitations of this work must be acknowledged. SUSAN and its models are still in development and many features are missing. For this reason, the framework is critical for testing new models and configurations. However, the specific performance results from this framework are only intended to inform further investigation. Major limitations to the currently implemented models are as follows. No flaps or other high-lift-devices are modeled at the time of this work. Additionally, the control surface actuators are represented with simple rate limits. The aerodynamic databases used in the SUSAN model are generated with OpenVSP, a relatively low fidelity tool [9]. The propulsion model is built from a high-fidelity tool set but still uses preliminary numbers for scaling and efficiency. In short, conclusions cannot be drawn from the exact performance of the SUSAN model, but it is still adequate for validating the framework presented in this work.

3 Baseline Flight Control Architecture

The flight control architecture, used as the baseline, allows the closed loop performance of the system to be compared between system configurations. In this work, a multi-loop control architecture, as shown in Fig. 2, is considered. The inner loop controls the aircraft's angular velocities and airspeed, the outer loop controls attitude, and the autopilot controls position and course. This framework is chosen to provide a standard control structure to enable analysis of the integrated closed loop system. To reduce clutter, feedforward, feedback, and auxiliary inputs are

omitted. The full interconnections are detailed in the equations presented in the following sections.

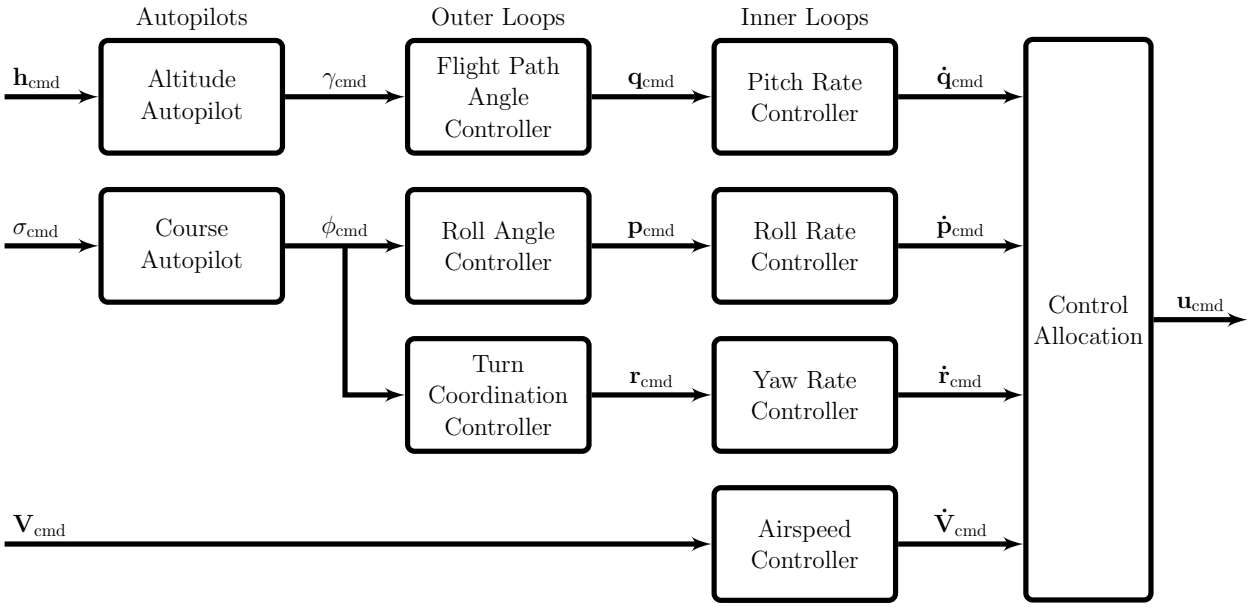


Figure 2: Control system block diagram.

3.1 Autopilot

The autopilot controller drives the system towards a user defined series of altitudes and course.

3.1.1 Altitude Autopilot

The altitude autopilot drives the aircraft towards a target altitude, h_{cmd}, by commanding a flight path angle, γ_{cmd}. To do this, the desired altitude rate, $\dot{h}_{\text{des}}$, is computed first. Under normal operation, $\dot{h}_{\text{des}}$ is dynamically generated based on the difference between the target altitude, h_{cmd}, and measured altitude, h,

$$\dot{h}_{\text{des}} = \text{PID}(h_{\text{cmd}}, h), \tag{1}$$

where PID is an instance of a proportional-integral-derivative feedback controller. To prevent integral windup during transients, the integral term is set to 0 when $|h_{\text{cmd}} - h| > 50$ ft. During terminal flight phases (takeoff or landing) guidance commands set $\dot{h}_{\text{des}}$ directly,

$$\dot{h}_{\text{des}} = \dot{h}_{\text{cmd}}, \tag{2}$$

where $\dot{h}_{\text{cmd}}$, is the commanded altitude rate received by the altitude autopilot.

Then, the flight path angle command, γ_{cmd}, is determined from $\dot{h}_{\text{des}}$ by leveraging the following kinematic relationship,

$$\dot{h} = V \sin(\gamma), \tag{3}$$

where V is the airspeed of the aircraft. Setting $\dot{h}$ to be $\dot{h}_{\text{des}}$ and rearranging the above equation to solve for γ_{cmd}, yields the governing equation for the altitude autopilot,

$$\gamma_{\text{cmd}} = \arcsin\left(\frac{\dot{h}_{\text{des}}}{V}\right). \tag{4}$$

This controller is only used when V is greater than a minimum speed, chosen to avoid commanding steep flight path angles which would stall the aircraft. Lastly, to prevent unachievable flight paths from being commanded by the autopilot, γ_{cmd} is subject to the limit

$$|\gamma_{\text{cmd}}| < 10^{\circ}, \tag{5}$$

chosen based on knowledge of the desired behavior of the aircraft. The flight path angle command is then sent to the flight path angle controller, serving as its target value.

3.1.2 Course Autopilot

The course autopilot drives the aircraft towards a target course angle, σ_{cmd}, by commanding a roll angle, ϕ_{cmd}. The roll angle, ϕ, can be related to the turn rate, $\dot{\sigma}$, by the kinematic approximation,

$$\dot{\sigma} \approx \phi \frac{g}{V}, \tag{6}$$

with g representing Earth's gravitational constant. The desired change in course, $\dot{\sigma}_{\text{des}}$, is determined dynamically from the vehicle's course, σ, and a PID feedback loop,

$$\dot{\sigma}_{\text{des}} = \text{PID}\left(\sigma_{\text{cmd}}, \sigma\right). \tag{7}$$

Rearranging Equation (6) to solve for ϕ and plugging Equation (7) in for σ provides the course autopilot's governing equation:

$$\phi_{\text{cmd}} = \text{PID}\left(\sigma_{\text{cmd}}, \sigma\right) \frac{V}{g}. \tag{8}$$

The roll angle commanded by the course autopilot goes on to drive the lateral dynamics of the aircraft through the roll angle controller. The resulting change in roll angle drives the turn coordination controller.

3.2 Outer Loops

The outer loop controllers drive the attitude of the system towards the autopilot targets. Additionally, the turn coordination controller counteracts lateral accelerations resulting from the current speed and attitude.

3.2.1 Flight Path Angle Controller

The flight path angle controller drives the aircraft towards a target flight path angle, γ_{cmd}, by commanding a pitch rate, q_{cmd}. This controller solely consists of the PID feedback loop,

$$q_{\text{cmd}} = \text{PID}\left(\gamma_{\text{cmd}}, \gamma\right). \tag{9}$$

The pitch rate commanded generated by this controller is sent to the pitch rate controller, which adjusts the pitch, θ, of the aircraft.

3.2.2 Roll Angle Controller

The roll angle controller drives the aircraft towards a target roll angle, ϕ_{cmd}, by commanding a roll rate, p_{cmd}. This controller solely consists of the PID feedback loop,

$$p_{\text{cmd}} = \text{PID}\left(\phi_{\text{cmd}}, \phi\right). \tag{10}$$

The roll rate command generated by this controller is sent to the roll rate controller.

3.2.3 Turn Coordination Controller

The turn coordination controller commands a yaw rate to balance the lateral force experienced during banked turning maneuvers. The equation,

$$r_{\text{req}} = \frac{g}{V} \frac{\tan(\phi)\cos(\alpha)}{\cos(\beta) + \tan(\phi)\sin(\alpha)\sin(\beta)}, \tag{11}$$

defines the yaw rate required, r_{req}, to achieve a coordinated turn, given a the vehicle angle of attack, α, and angle of sideslip, β. This coordinated turn equation is given in Ref. [10]. Achieving the yaw rate, r_{req}, minimizes both the angle of sideslip, and lateral acceleration experienced by the plane and its passengers. Therefore, the turn coordination controller command,

$$r_{\text{cmd}} = r_{\text{req}}, \tag{12}$$

drives the aircraft into a coordinated turn. The rate command, r_{cmd}, is then sent to its associated controller, that adjusts the yaw, ψ. Presently, the turn coordination controller operates as an open loop. Future work could make the controller more robust by incorporating feedback.

3.3 Inner Loops

The inner loops control the aircraft's rotational and translational velocity by commanding forces and moments to achieve the velocities prescribed by the outer loops.

3.3.1 Roll Rate Controller

The roll rate controller commands a moment about the longitudinal axis based on roll rate feedback. First, a PID feedback loop determines the desired roll acceleration,

$$\dot{p}_{\text{des}} = \text{PID}\left(p_{\text{cmd}}, p\right), \tag{13}$$

where $\dot{p}_{\text{des}}$ represents desired roll acceleration. Then, $\dot{p}_{\text{des}}$ is multiplied by the moment of inertia about the longitudinal axis, I_{xx}, to calculate the commanded rolling moment,

$$M_{xx_{\text{cmd}}} = \dot{p}_{\text{des}} I_{xx}. \tag{14}$$

The commanded rolling moment is then sent to the control allocation scheme which computes the effector commands needed to achieve this moment.

3.3.2 Pitch Rate Controller

The pitch rate controller commands a moment about the lateral axis based on pitch rate feedback. First, a PID feedback loop determines the desired pitch acceleration,

$$\dot{q}_{\text{des}} = \text{PID}\left(q_{\text{cmd}}, q\right), \tag{15}$$

where $\dot{q}_{\text{des}}$ represents desired pitch acceleration. Then, $\dot{q}_{\text{des}}$ is multiplied by the moment of inertia about the lateral axis, I_{yy}, to calculate the commanded pitching moment,

$$M_{yy_{\text{cmd}}} = \dot{q}_{\text{des}} I_{yy}. \tag{16}$$

The commanded pitching moment is then sent to the control allocation scheme which computes the effector commands needed to achieve this moment.

3.3.3 Yaw Rate Controller

The yaw rate controller commands a moment about the vertical axis based on yaw rate feedback. First, a PID feedback loop determines the desired yaw acceleration,

$$\dot{r}_{\text{des}} = \text{PID}\left(r_{\text{cmd}}, r\right), \tag{17}$$

where $\dot{r}_{\text{des}}$ represents desired yaw acceleration. Then, $\dot{r}_{\text{des}}$ is multiplied by the moment of inertia about the vertical axis, I_{zz}, to calculate the commanded yawing moment,

$$M_{zz_{\text{cmd}}} = \dot{r}_{\text{des}} I_{zz}. \tag{18}$$

The commanded yawing moment is then sent to the control allocation scheme which computes the effector commands needed to achieve this moment.

3.3.4 Airspeed Controller

The Airspeed Controller drives the aircraft towards a target speed, V_{cmd}, by commanding a change in thrust, δT_{cmd}. The commanded change in thrust is determined by summing two components: T_{PID}, and T_{FF}. The first component, T_{PID}, is the change in thrust commanded by a PID feedback loop on airspeed,

$$T_{\text{PID}} = \text{PID}\left(V_{\text{cmd}}, V\right). \tag{19}$$

The second component, T_{FF}, is a feed-forward term that accounts for the change in thrust required when the aircraft is climbing or descending. It is calculated by considering the commanded flight path angle, γ_{cmd}, and weight of the aircraft, w,

$$T_{\text{FF}} = w \sin\left(\gamma_{\text{cmd}}\right). \tag{20}$$

Summing these two terms yields the change in thrust to be commanded,

$$\delta T_{\text{cmd}} = T_{\text{PID}} + T_{\text{FF}}, \tag{21}$$

which is then sent to the control allocation scheme which computes the control effector commands needed to achieve this force.

3.4 Control Allocation

The control allocation receives the force and moment commands from the inner loop controllers and converts them into effector commands. The commanded forces and moments are grouped into the force/moment vector, F_{cmd}, where

$$F_{\text{cmd}} = \begin{bmatrix} M_{xx_{\text{cmd}}} & M_{yy_{\text{cmd}}} & M_{zz_{\text{cmd}}} & \delta \text{T}_{\text{tot}_{\text{cmd}}}, \end{bmatrix}^{\top}, \tag{22}$$

where M is the moment about the indicated axis, and the effector commands are grouped into the vector, u_{cmd}, where

$$u_{\text{cmd}} = \begin{bmatrix} \delta_{\text{aileron}} & \delta_{\text{elevator}} & \delta_{\text{rudder}} & \delta \text{T}_{\text{tot}} & \text{dT}_1 & \cdots & \text{dT}_8 \end{bmatrix}^{\top}. \tag{23}$$

The elements $\delta_{\text{aileron}}, \delta_{\text{elevator}}$, and δ_{rudder} represent commanded displacements from the trim condition of the aileron, elevator, and rudder respectively. The term, $\delta \text{T}_{\text{tot}}$, represents a change in commanded total thrust from the trim condition and $\text{dT}_1, \ldots, \text{dT}_8$ represent differential thrust commands for the eight pairs of wingfans. Currently, two different allocation methods are implemented for converting F_{cmd} into u_{cmd}.

3.4.1 Conventional Control Allocation

The conventional control allocation mimics how a pilot might control a simple aircraft. Rolling moments are achieved by aileron deflection, pitching moments by elevator deflection, and yawing moments by rudder deflection. Thrust commands are passed straight through to the propulsion system and are analogous to the pilot increasing the throttle.

The angle of deflection for each control surface is found by inverting the aerodynamic model of the aircraft. The moment, M, produced by control surfaces is modeled as

$$M = CSL\bar{q}, \tag{24}$$

where C is a unitless aerodynamic coefficient from a lookup table in the aerodynamic database, S represents a reference surface area, L is a reference length, and $\bar{q}$ is the dynamic pressure experienced by the aircraft. Rearranging the equation to solve for C yields the desired coefficient:

$$C_{\text{des}} = \frac{M_{\text{cmd}}}{SL\bar{q}}. \tag{25}$$

In the aerodynamic model, the value of C is derived from a lookup table across angle of attack, α, and the deflection of the control surface, δ. Therefore, for a given α and a desired aerodynamic coefficient, C_{des} from Equation (25), the lookup table can be inverted to solve for the required deflection:

$$\delta_{\text{req}} = f_{\text{lookup}}\left(\alpha, C_{\text{des}}\right), \tag{26}$$

where f_{lookup} is a function representing the aerodynamic lookup table. Inserting values specific to each of the three rotational axes into the general form yields the following equations for control surface commands:

$$\begin{aligned}
\delta_{\text{aileron}_{\text{cmd}}} &= f_{\text{aileron}_{\text{lookup}}}\left(\alpha, C_{\text{aileron}_{\text{des}}}\right), \\
C_{\text{aileron}_{\text{des}}} &= \frac{M_{\text{xx}_{\text{cmd}}}}{SL_{\text{aileron}}\bar{q}}, \\
\delta_{\text{elevator}_{\text{cmd}}} &= f_{\text{elevator}_{\text{lookup}}}\left(\alpha, C_{\text{elevator}_{\text{des}}}\right), \\
C_{\text{elevator}_{\text{des}}} &= \frac{M_{\text{yy}_{\text{cmd}}}}{SL_{\text{elevator}}\bar{q}}, \\
\delta_{\text{rudder}_{\text{cmd}}} &= f_{\text{rudder}_{\text{lookup}}}\left(\alpha, C_{\text{rudder}_{\text{des}}}\right), \\
C_{\text{rudder}_{\text{des}}} &= \frac{M_{\text{zz}_{\text{cmd}}}}{SL_{\text{rudder}}\bar{q}}.
\end{aligned} \tag{27}$$

In all cases, the reference area, S, is taken to be equal to the aircraft's wing area. The reference length for the aileron and rudder, L_{aileron} and L_{rudder} respectively, is the span of the wing, b. The reference length for the elevator, L_{elevator}, is the wing mean aerodynamic chord, $\bar{c}$. Finally, the thrust command, $\delta\text{T}_{\text{tot}_{\text{cmd}}}$, received from the speed controller is passed straight through to the propulsion system,

$$\delta\text{T}_{\text{tot}_{\text{cmd}}} = \delta\text{T}_{\text{tot}_{\text{cmd}}}. \tag{28}$$

The conventional allocation does not utilize differential thrust, so

$$\boldsymbol{u}_{\text{cmd}} = \begin{bmatrix} \delta_{\text{aileron}_{\text{cmd}}} & \delta_{\text{elevator}_{\text{cmd}}} & \delta_{\text{rudder}_{\text{cmd}}} & \delta\text{T}_{\text{tot}_{\text{cmd}}} & 0 & \cdots & 0 \end{bmatrix}^{\top}. \tag{29}$$

The conventional control allocation minimizes complexity but does not utilize DEP nor does it account for coupling of the control effectors. Ailerons and rudder are used to achieve commanded

rolling and yawing moments respectively, but also produce uncommanded yawing and rolling moments respectively. Similarly, commanding thrust from the propulsion system produces an uncommanded pitching moment. The unintended moments resulting from this coupling are accounted for by the integral terms in the inner loop PID controllers. Alternatively, the control allocation presented in the following section offers a direct method for managing actuator coupling.

3.5 Framework Evaluation Modification: Pseudoinverse Allocation

The control allocation method is modified and evaluated to demonstrate the framework's capabilities. Allocation modifications are useful to study in this class of vehicles where new actuators and control concepts are being developed. A pseudoinverse allocation is obtained by inverting the input matrix, B, of the linear state space model of the aircraft. The B matrix used for this inversion is found by interpolating between linearized operating points. The B matrix is of size $(m \times n)$ in which m is equal to the number of inputs and n is equal to the number of states in the state space model. For this SUSAN configuration there are 12 inputs $(\delta_{\text{aileron}}, \delta_{\text{elevator}}, \delta_{\text{rudder}}, \delta\text{T}_{\text{tot}}, \text{dT}_1, ..., \text{dT}_8)$ and 9 states $(V, \alpha, \beta, p, q, r, \phi, \theta, \psi)$. The pseudoinverse control allocation considers the four states controlled by the inner loops of the flight control system (p, q, r, V).

The state derivative of a state space model, $\dot{\boldsymbol{X}}$, is defined as

$$\dot{\boldsymbol{X}} = A\boldsymbol{X} + B\boldsymbol{u}, \tag{30}$$

In which A is the $n \times n$ state matrix, $\boldsymbol{X}$ is the $n \times 1$ state vector, and $\boldsymbol{u}$ is the $m \times 1$ input vector. For control allocation we consider the contributions from the control input to the state derivative by removing the state dynamics from Equation (30), this modified equation is denoted

$$\dot{\boldsymbol{X}}_{\text{cmd}} = B\boldsymbol{u}_{\text{req}}. \tag{31}$$

This equation is then inverted to find the required input vector to achieve a commanded state derivative,

$$\boldsymbol{u}_{\text{req}} = B^{-1}\dot{\boldsymbol{X}}_{\text{cmd}}. \tag{32}$$

However, B is non-square and therefore not invertible, so, the weighted pseudoinverse, $B^{\dagger}$, of B is used instead. The weighted pseudoinverse is calculated from the input matrix and a weighting matrix, W, where W is a square matrix of real numbers with dimensions equal to the number of control effectors ($W \in \mathbb{R}^{(m,m)}$).. With these two matrices, $B^{\dagger}$ is the minimum 2-norm solution to the control allocation formulated as

$$\begin{aligned} \arg\min_{\boldsymbol{u}} \quad & \frac{1}{2}\boldsymbol{u}^T W \boldsymbol{u} \\ \text{subject to} \quad & B\boldsymbol{u} = \dot{\boldsymbol{X}}_{\text{cmd}}. \end{aligned} \tag{33}$$

If the B matrix is full rank for the states being considered, the solution to the above optimization problem becomes

$$\boldsymbol{u} = W^{-1}B^T\left(BW^{-1}B^T\right)^{-1}\dot{\boldsymbol{X}} \tag{34}$$

and

$$B^{\dagger} = W^{-1}B^T\left(BW^{-1}B^T\right)^{-1}. \tag{35}$$

Details on this optimization can be found in Ref. [11]. The control allocation equation can then be written as

$$\boldsymbol{u}_{\text{cmd}} = B^{\dagger}\dot{\boldsymbol{X}}_{\text{cmd}}, \tag{36}$$

in which $B^\dagger$ is a matrix of real numbers with as many rows as there are states being controlled and as many columns as there are control effectors ($B^\dagger \in \mathbb{R}^{n\times m}$) and

$$\boldsymbol{u}_{\text{cmd}} = \begin{bmatrix} \delta_{\text{aileron}} & \delta_{\text{elevator}} & \delta_{\text{rudder}} & \delta\text{T}_{\text{tot}} & \text{dT}_1 & \cdots & \text{dT}_8 \end{bmatrix}^\top. \tag{37}$$

For this allocation to be compatible with the control architecture, $\dot{\boldsymbol{X}}_{\text{cmd}}$ must be defined from the force/moment vector, $\boldsymbol{F}_{\text{cmd}}$, multiplied by the inverse of each element's respective moment of inertia or mass. That is,

$$\dot{\boldsymbol{X}}_{\text{cmd}} = \begin{bmatrix} \dot{p}_{\text{cmd}} \\ \dot{q}_{\text{cmd}} \\ \dot{r}_{\text{cmd}} \\ \dot{V}_{\text{cmd}} \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{zz}} & 0 \\ 0 & 0 & 0 & \frac{1}{m} \end{bmatrix} \boldsymbol{F}_{\text{cmd}} = \begin{bmatrix} \frac{M_{\text{xx}_{\text{cmd}}}}{I_{xx}} \\ \frac{M_{\text{yy}_{\text{cmd}}}}{I_{yy}} \\ \frac{M_{\text{zz}_{\text{cmd}}}}{I_{zz}} \\ \frac{T_{\delta_{\text{cmd}}}}{m} \end{bmatrix}. \tag{38}$$

Further, this approach to control allocation allows for online modification and fine tuning via the weighting matrix, W. As a baseline, W is a diagonal matrix in which each element corresponds to one of the control effectors. Small entries in W incentivize the use of the corresponding effector while large entries do the opposite, that is

$$u\left(j\right) \propto \frac{1}{W\left(j,j\right)}, \tag{39}$$

where j represents the index of the effector. In the SUSAN implementation, each element in the weighting matrix is the product of three terms: a saturation term, W_{s}; failure term, W_{f}; and manual adjustment term, W_{a}.

The value of W_{s} is the only dynamic term and favors effectors that are furthest away from their physical limits. The value of W_{s} is computed with the following steps. First, the difference between the current effector usage and its minimum, $u_{\min}$, and maximum, $u_{\max}$, values is found and compared, with the smaller of the two being taken as the current distance from its physical limit d_s,

$$d_s\left(j\right) = \min \begin{Bmatrix} u\left(j\right) - u_{\min}\left(j\right) \\ u_{\max}\left(j\right) - u\left(j\right) \end{Bmatrix}. \tag{40}$$

Then, the saturation weighting term is set to be the inverse of the distance squared,

$$W_{\text{s}}\left(j,j\right) = \frac{1}{d_s\left(j\right)^2}. \tag{41}$$

As an effector approaches its physical limits, d_s approaches 0 and W_{s} approaches ∞, disincentivizing the use of that effector. Presently, no limits are placed on d_s to prevent dividing by zero when calculating W_s. Robustness is built into the inverse function which allows the pseudoinverse method to handle infinite elements in W_s. In future work, dividing by zero could be avoided by implementing a saturation condition, bounding d_s above some minimum value, and a condition number check.

The failure term, W_{f}, has each diagonal element set to 1 nominally and ∞ if failure is detected in the corresponding actuator. This term does not impact the weighting unless a failure is detected and in the event of a failure, removes the failed effector from the allocation scheme.

The manual adjustment term, W_{a}, allows the user to manually tune the control allocation. Like the failure term, setting an element of W_{a} to 1 will result in no change, while setting W_{a} to ∞ will remove the corresponding effector from the allocation. Additionally, setting $W_{\text{a}} < 1$ will

increase utilization of the corresponding effector, while setting $W_{\mathrm{a}} > 1$ will decrease utilization of the corresponding effector.

Combining the above terms produces a weighting matrix which automatically modifies the control allocation based on effector limits and failures while also allowing for user tuning of the allocation,

$$W = W_{\mathrm{s}} W_{\mathrm{f}} W_{\mathrm{a}}. \tag{42}$$

The process for forming W and implementing the pseudoinverse control allocation is depicted in Figure 3. Please note that the non-dynamic contributions to W, W_f, and W_a are omitted to improve readability.

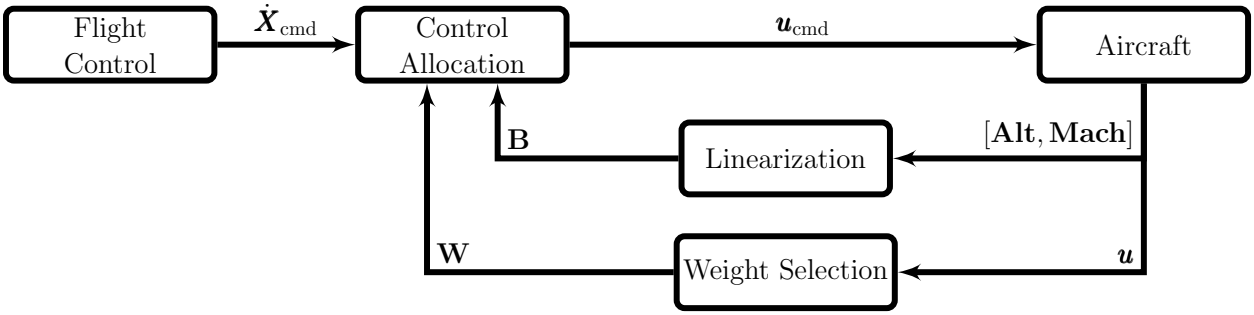


Figure 3: Pseudoinverse control allocation block diagram.

As an example, setting the rudder weight term to a large enough value, e.g., $W_{\mathrm{a}}(3,3) = 10^5$, will result in differential thrust being used to achieve yawing moments unless the wingfans are near their physical limits. The rudder's relatively large weighting term effectively removes it from the allocation until the saturation weighting term for the wingfans approaches a similar magnitude, bringing the rudder back into use. In practice, this means yawing moments would be produced by differential thrust during nominal conditions, but in scenarios like takeoff, where the wingfans are operating at max power and have no room for differential thrust, the rudder would be used for yaw control.

4 Propulsion System

The propulsion system model consists of four dynamic elements: control loops, wingfan actuators, a gas dynamics based turbofan/wingfan plant, and sensors. These four elements interact as shown in Figure 4. The signal $\mathrm{T}_{\mathrm{tot}_{\mathrm{cmd}}}$ is the input total thrust command, T_{tf} represents the thrust output from the turbofan, and the vector $\boldsymbol{T}_{\mathrm{wf}}$ represent the thrust produced by each wingfan. The subscripts wf and tf indicate wingfan or turbofan variables. Thus $\omega_{\square}$ and $\omega_{\square_m}$ represent the rotational velocity and measured rotational velocity, where the square symbol, $\square$, is replaced with either tf or wf to indicate the propulsor type. The system takes a thrust command as input, which is mapped to target rotational velocities. The controllers attempt to track the rotational velocity of the propulsors, allowing the flight controller to compensate for errors resulting from incorrect mappings from rotational velocity to thrust.

The baseline propulsion controller consists of proportional-integral (PI) feedback between the target and measured rotational velocity of each propulsor. These are the closed loops shown in Figure 4. The gains for these controllers are carefully selected to maintain the stability of the hybrid system and are not modified during design of the control system. Notice that the wingfan thrust reference is set based on the current rotational velocity of the turbofan. This locks the response of

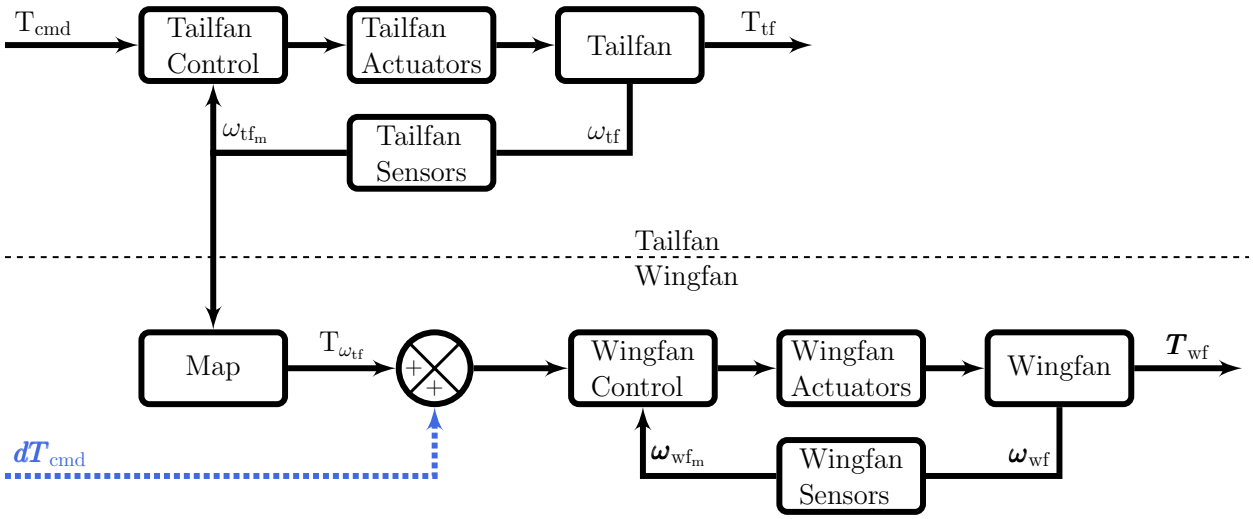


Figure 4: Propulsion system top level block diagram. Modifications to the baseline shown in blue with dashed lines. The $\boldsymbol{dT}_{\text{cmd}}$ signal is a feedforward term from the control law, added for framework evaluation.

the wingfans to the response of the turbofan to maintain electrical power balance. However, this artificially limits the performance of the wingfans, and limits some potential advantages of DEP.

The turbofan dynamics are produced by an EMTAT and T-MATS implementation of the gas and electrical dynamics for the propulsion system [7, 8]. This provides a detailed simulation of the system but introduces new states and non-linearities that make it difficult to analyze from a control standpoint. Consequently, linear approximations of the propulsion system will be used for all control design in this framework.

4.1 Evaluation Case Modification: Control and Propulsion Integration

The control and propulsion systems are integrated to demonstrate the framework's capabilities in analyzing modifications to complex interconnected dynamic simulations. System integration could be used to enable safe differential thrust control using the aircraft's wingfans by decoupling the wingfan and turbofan speeds for net-zero power commands.

A test case for the framework using differential thrust commands is implemented with two feedforward signals, as shown in Figure 5. The first signal is the differential thrust command, $\boldsymbol{dT}_{\text{cmd}}$. The turbofan rotational velocity still drives, and limits, the average thrust, $T_{\omega_{\text{tf}}}$, to be produced by the wingfans, ensuring the power split between the turbine and electrical system is balanced, while differential thrust commands, $\boldsymbol{dT}_{\text{cmd}}$, are added to this average. The differential thrust command is constructed to be net-zero power usage by symmetrically increasing and decreasing power usage on pairs of wingfans. This approach allows DEP to be utilized without destabilizing the electrical or propulsion systems because it does not alter the net thrust produced by the wingfans and therefore has a negligible impact on the total electrical power consumed by the system.

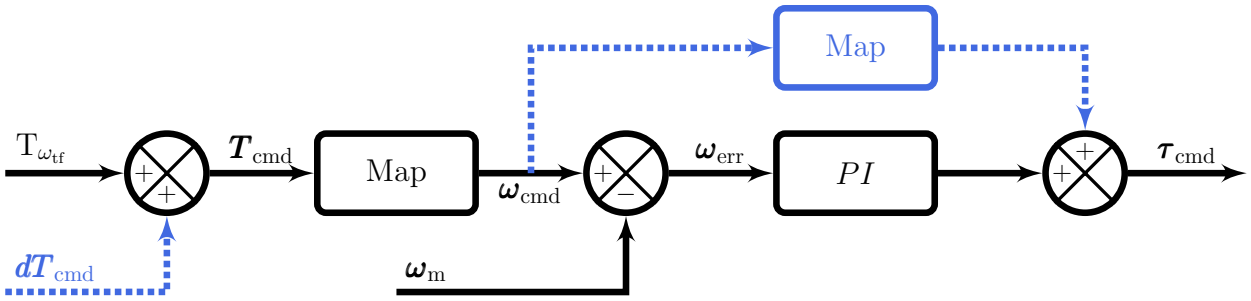


Figure 5: Wingfan controller block diagram. Modifications to the baseline shown in blue with dashed lines. The $\boldsymbol{dT}_{\text{cmd}}$ and the map parallel to the PI controller are both feedforward terms used to evaluate the framework for system integration cases.

The second feed-forward term is added to the propulsion controller to improve the wingfans speed controller response. This modification is shown in Figure 5. Data from the propulsion system operating at steady state across the flight envelope is used to map the steady state output of the controller to its rotational velocity. More specifically, a commanded rotational velocity, $\boldsymbol{\omega}_{\text{cmd}}$, is mapped to a commanded torque, $\boldsymbol{\tau}_{\text{cmd}}$, for the wingfan controller and to fuel flow for the turbofan controller. This mapping then provides each controller with an initial guess at what the steady state output would be. This improves the response of the system because the feedback PI controller only needs to correct for uncertainties in the mapping instead of converging to equilibrium from zero. The impact of these modifications is analyzed in Section 6.1 .

5 Tuning

With the previously stated flight control architecture in place, the performance of the aircraft can be altered by adjusting the gains in each controller. For this baseline flight control system, the autopilots and outer loops are hand tuned, while a gain schedule is algorithmically generated for the inner loops. To generate this gain schedule, single input single output (SISO) linear models are generated across the flight envelope for each of the four channels of interest (V, p, q, r). These linear models incorporate the dynamics of the airframe, control surface actuators, and propulsion system. The linearization process is described in detail in Appendix A. A tunable PID controller is then added as a feedback controller to the SISO model. Additionally, a static pseudoinverse control allocation is placed between the PID controller and plant to ensure that the PID controller is being tuned to output acceleration commands. The tunable PID controller is then given performance objectives. Currently, the user specifies minimum phase and gain margins as well as region in the complex plane that the poles must fall, as shown in Figure 6. The system is tuned towards a desired response using H_∞ optimization as implemented in MATLAB's® `systune` function [12]. The H_∞ method defines a cost function based on the user's performance requirements and then returns the controller which minimizes this cost function. This approach allows different configurations of the SUSAN aircraft to easily be tuned towards comparable closed loop performance for trade studies on the integrated system.

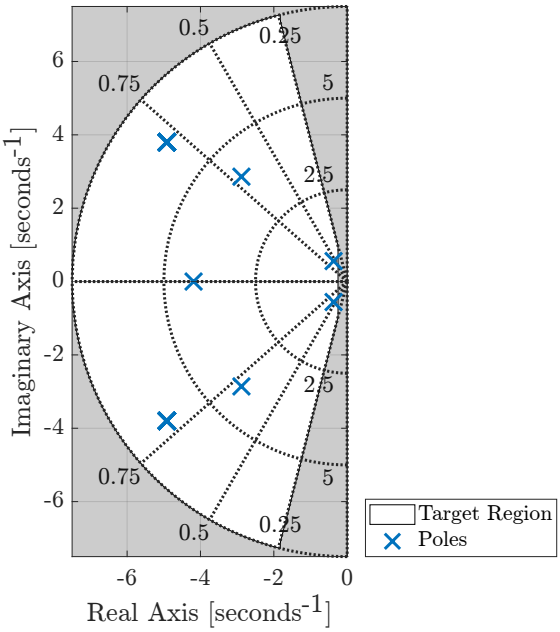


Figure 6: Pole placement tuning goal for yaw controller. Altitude = 37,000 ft, Mach number = 0.78.

The inner loops directly control the dynamics of the aircraft, and these dynamics change with flight condition; hence their selection for algorithmic tuning. In the future, this algorithmic tuning approach could be expanded to include the outer loops and autopilots.

6 Framework Evaluation Results

6.1 Evaluation Case: Control and Propulsion Integration

To design a control system that utilizes DEP, the dynamics and latency of the propulsion system must be understood. For example, using differential thrust for lateral control requires the propulsors to respond quickly to pilot commands. Changes to the propulsion system or control laws can adversely affect the response time and may lead to handling quality issues. The effect of different modifications on the propulsion control system can be evaluated using the framework presented in this work.

In this evaluation case, two different configurations are considered. The first configuration is the baseline configuration with no integration between the flight controller and the propulsion control system. The second configuration integrates the controllers to enable greater realized propulsion bandwidth and enable differential thrust. In this case, the performance objectives are the response to changes in thrust command from the flight control system through the propulsion system and to the thrust produced by the wingfans. This is a critical metric for enabling the use of DEP as a lateral control effector and affects forward acceleration tracking performance as well. Nominal requirements for the propulsion system are established using the baseline system configuration. These requirements must be met by controllers that use differential thrust for lateral control. First, a safety limit is established using the framework to evaluate the dutch roll mode across altitudes. If the bandwidth of the propulsion system is lower than the frequency of the dutch roll mode, the phase margin of the system is reduced. Second, a performance goal is established to maximize the

bandwidth of the integrated wing fan and propulsion systems.

From a flight control perspective, the response to two different scenarios is of interest. First, the response when increasing/decreasing the net thrust of the entire propulsion system. Second, the response to an increase/decrease in thrust for the electric wingfans only. The first scenario is dominated by the relatively slower dynamics of the turbofan and prioritizes maintaining stable combustion and battery charge over performance. The second scenario captures what the wingfans are capable of when used to produce differential thrust.

Four sets of models are created and analyzed in the presented framework to demonstrate suitability for evaluation of new control concepts. These models are created using:

1. the baseline propulsion system,
2. the propulsion control system with the differential thrust feedforward term,
3. the propulsion control system with the speed control feedforward term,
4. the propulsion control system with both feedforward terms.

Creating these linear models allows the design of the flight control system to incorporate the dynamics of the propulsion system. Further, having models for the four scenarios provides insight into the effectiveness of DEP as a control strategy and the impact of the feed-forward terms.

Figure 7 displays the bandwidth for each scenario and includes the frequency of the dutch roll mode as a reference point for the safety requirement. Bandwidth is plotted across altitude with each point on the plot being the average bandwidth across airspeed for each given altitude. A full plot of bandwidth across altitude and airspeed is provided by Figure B1 located in Appendix B.

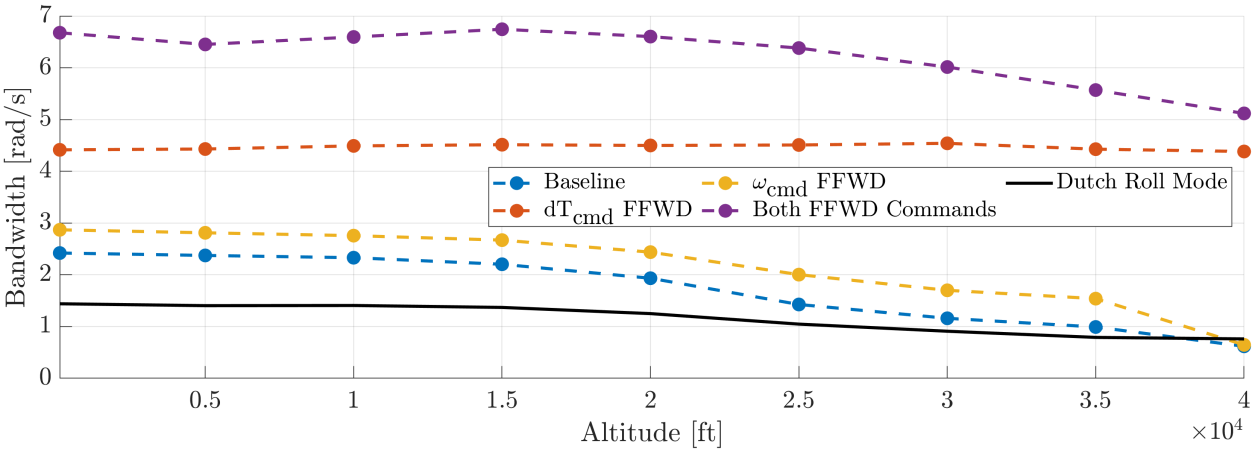


Figure 7: Bandwidth of the propulsion system across altitudes.

The wingfans by themselves have a higher bandwidth than the overall propulsion system. This difference in bandwidth may limit the performance of the vehicle when the response of the wingfans is linked to the response of the tailfan. Figure 7 demonstrates how different configurations change the realized bandwidth from thrust command to produced wingfan thrust. Further, the feed-forward term improves the response for both scenarios but has a much greater effect on the wingfans than the overall propulsion system. The main benefit of integration is that more of the bandwidth of the wingfans can be realized by feeding specific information forward and decoupling the response of specific control commands from the main engine. Commands that would change the operating point of the main engine, i.e., the net thrust, must be limited by the stability conditions of the engine. However, the differential component of the thrust command is constructed to require no net change in power. Therefore, the differential thrust command can be sent directly to the wingfans, without waiting on the main engine response.

Comparing the bandwidth of the propulsion system to the dutch roll mode reveals the need for the wingfans to be able to act independently of the overall propulsion system to compensate for the dutch roll mode. At low altitudes, the overall propulsion system's bandwidth is roughly twice the dutch roll mode's frequency. This ratio decreases with increasing altitude and at 40,000 ft, the dutch roll mode is faster than the propulsion system's bandwidth. In contrast, the wingfan's bandwidth is at least three times the dutch roll's frequency across the flight envelope and more than four times faster when the feed-forward term is implemented. In short, jointly designing controllers for propulsion subsystems increases aircraft performance by decoupling the wingfans from the performance of the turbofan, without reducing the efficiency or stability of the propulsion system.

6.2 Evaluation Case: Tail Sizing

An integrated framework is needed to evaluate and compare the performance and stability of aircraft configurations with different sized vertical tails. The framework is shown to enable these comparisons with the evaluation cases presented in this section. Aircraft configurations representing a full sized and 50% sized vertical tail are subjected to lateral disturbances and their response is analyzed. These evaluation cases are important because they demonstrate how DEP can be used to compensate for a reduced vertical tail.

To evaluate DEP controller compensation for disturbances, a 1 degree per second initial error in p and r is introduced to both the 100% and 50% vertical tail models. The open loop response to these disturbances is provided to demonstrate the lateral directional static stability of the aircraft.

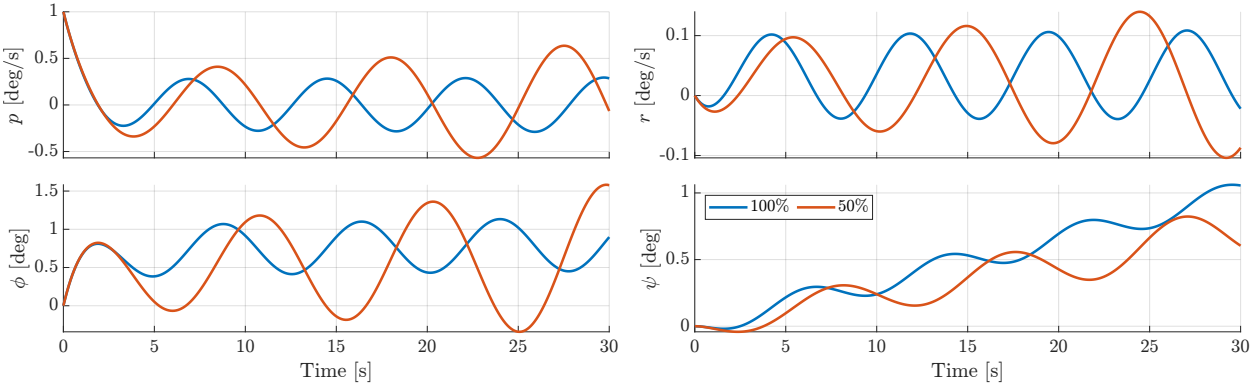


Figure 8: Open loop response to a roll rate disturbance of 1 degree per second.

When subjected to a p disturbance, both models experience oscillations in $p, r, \phi,$ and ψ as seen in Figure 8. The 100% tail size model exhibits oscillations which appear constant in amplitude, suggesting that it is neutrally statically stable. The oscillations exhibited by the 50% vertical tail model grew with time, indicating that the reduced tail model may be statically unstable in the lateral direction.

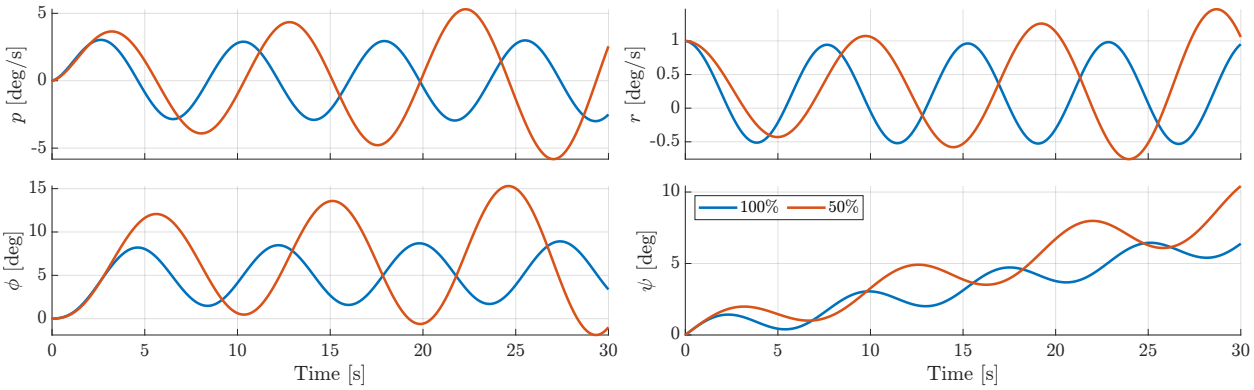


Figure 9: Open loop response to a yaw rate disturbance of 1 degree per second.

When subjected to an r initial error, the oscillations seen from both configurations in Figure 9 are qualitatively similar to those seen in Figure 8. The 100% model suggest neutral static stability while the 50% vertical tail model suggests static instability.

6.3 Evaluation Case: Control Allocation

The open loop response illustrates the difference in response resulting from the difference in tail size. The closed loop response will illustrate how the framework is able to account for these differences to provide similar closed loop performance across different aircraft configurations. Additionally, the closed response under both control allocation schemes, described in Section 3.4, is presented to demonstrate the framework's support for different control allocation algorithms as well as the potential benefit of the pseudoinverse allocation scheme.

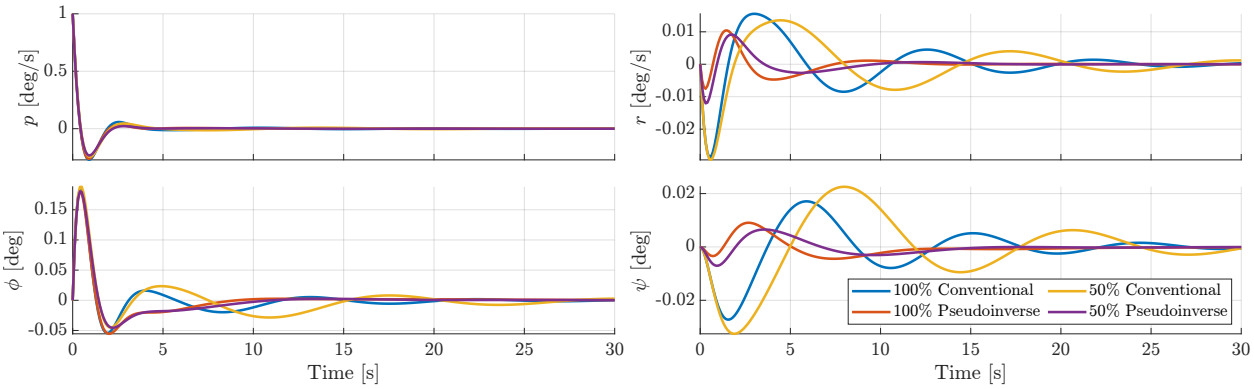


Figure 10: Closed loop response to a roll rate disturbance of 1 degree per second with conventional and pseudoinverse control allocation.

Figure 10 plots the closed loop response to a p disturbance and Figure 11 plots the closed loop response to an r disturbance. The plots of p and r show that all four combinations of tail size and control allocation reject the disturbance and exhibit similar responses. Table 1 contains the settling time and maximum overshoot for each combination of tail size and control allocation.

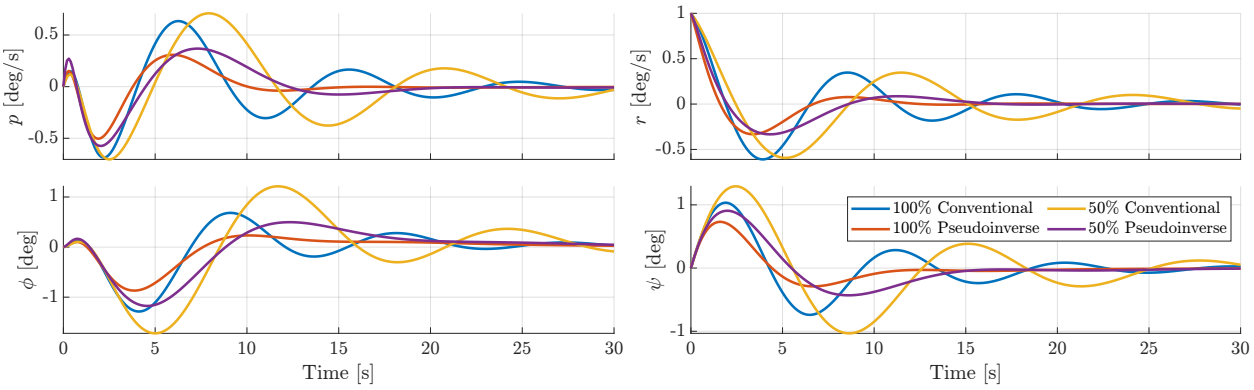


Figure 11: Closed loop response to a yaw rate disturbance of 1 degree per second with conventional and pseudoinverse control allocation.

When subjected to an r disturbance, the closed loop system again rejects the disturbance but not as quickly as it did the p disturbance. This is in line with the greater amplitudes seen during the open loop response. The performance metrics for the four combinations response to the r disturbance is also included in Table 1.

Table 1: Closed Loop Response to Lateral Error

Control Allocation	Error	2% Settling Time (s)	% Overshoot	Tail Size
Conventional	p	3.5	27.1%	100%
Pseudoinverse	p	3.2	26.3%	100%
Conventional	p	3.9	24.4%	50%
Pseudoinverse	p	3.1	23.3%	50%
Conventional	r	28.5	61.0%	100%
Pseudoinverse	r	11.4	33.2%	100%
Conventional	r	30.1	59.3%	50%
Pseudoinverse	r	15.4	33.3%	50%

For every combination of disturbance and tail size, the pseudoinverse control allocation provides superior performance (lower overshoot and faster settling time) compared to the conventional control allocation. Utilizing the pseudoinverse control allocation, the maximum overshoot and settling time differed by less than 3.5% between the two tail sizes when subjected to a p disturbance. When subjected to an r disturbance, the maximum overshoot differed by just 0.1% while the settling times differed 30%.

Due to the preliminary nature of the SUSAN models, the exact performance of each system has not been captured with this test. Instead, this evaluation case demonstrates that the framework provides comparable closed loop performance for aircraft configurations that feature different aerodynamic models. Additionally, it showcases that the framework can handle different control allocation algorithms and that the DEP utilizing pseudoinverse control allocation provides superior performance to the conventional control allocation.

6.4 Evaluation Case: DEP Fault Tolerance

To demonstrate the fault tolerance of the framework, two configurations of the SUSAN aircraft are subjected to wingfan failures. In these scenarios, the left most wingfan(s), as depicted in

Figure 1, will cease to produce thrust and the flight control system will attempt to stabilize the system without knowledge of the failure. Three specific failure cases will be analyzed:

1. the failure of the left most wingfan,
2. the failure of the four left most wingfans,
3. the failure of the entire left side of wingfans.

These cases are selected because they are the worst-case scenario for lateral controllability for a given number of wing fan failures. For convenience, the three failure cases will be referred to as Cases 1-3 respectively.

First, the open loop response to all three failure cases is provided, in Figure 12, as a baseline to compare the performance of the framework against.

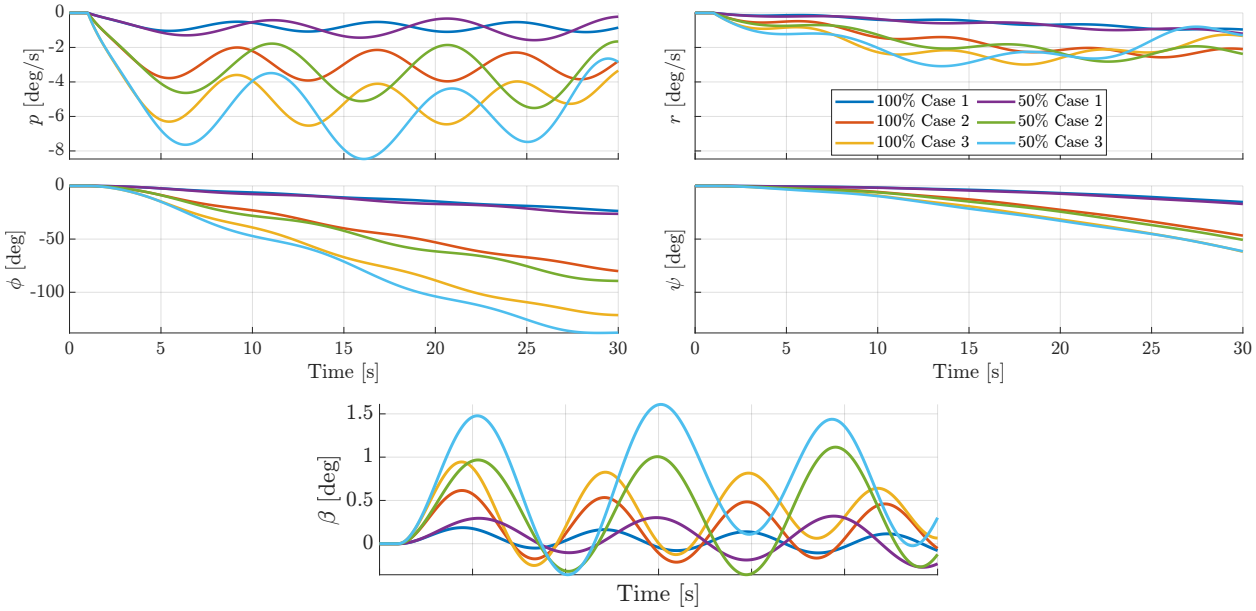


Figure 12: Open loop response to wingfan failure(s).

The open loop response indicates that wingfan failure(s) create an imbalance in the system, producing bounded disturbances in both p and r. The open loop response of the sideslip angle, β, is also bounded. While the system's angular rate response is bounded, the system experiences undesirable roll angles when wingfan failure(s) are left unchecked by automatic control or pilot action.

Next, the closed response to all three failure scenarios is shown in Figure 13. In all three failure cases, the closed loop framework counteracts the disturbances for both aircraft configurations. The sideslip angle, β, increases with tail size reductions and an increasing number of actuator failures to compensate for the thrust imbalance. The 50% tail aircraft experiences larger deviations than the 100% tail aircraft, but both converge at approximately the same speed.

Further, the longitudinal response to the three failure cases suggests that the physical limits of the propulsion system would be the limiting factor rather than the disturbances in the lateral axis.

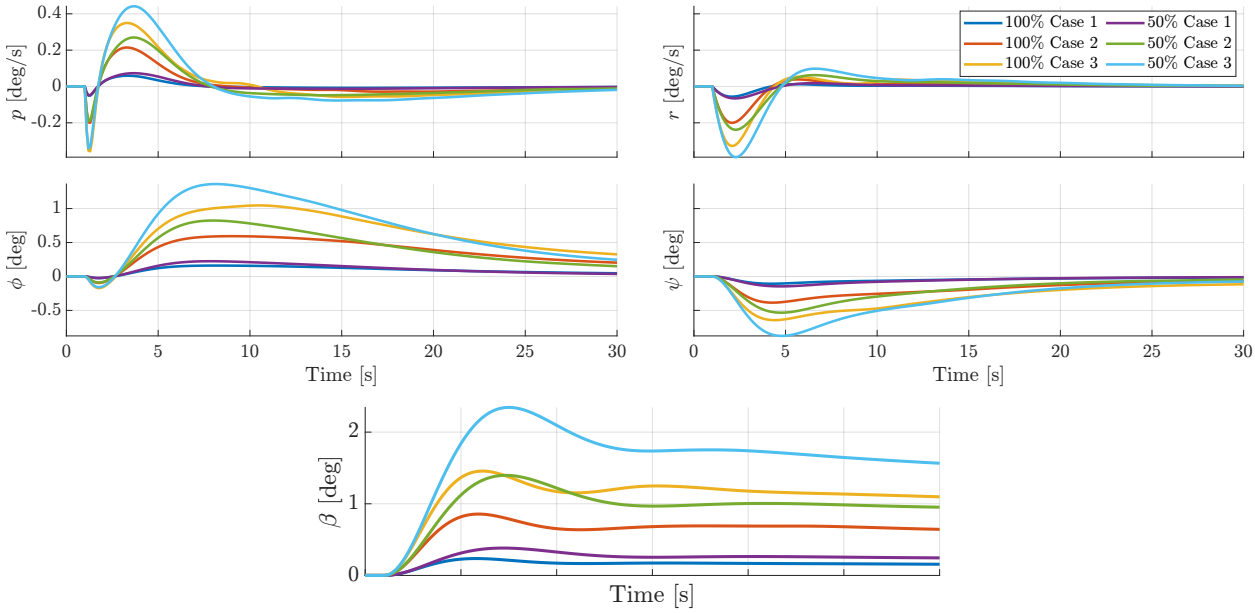


Figure 13: Closed loop response to wingfan failure(s).

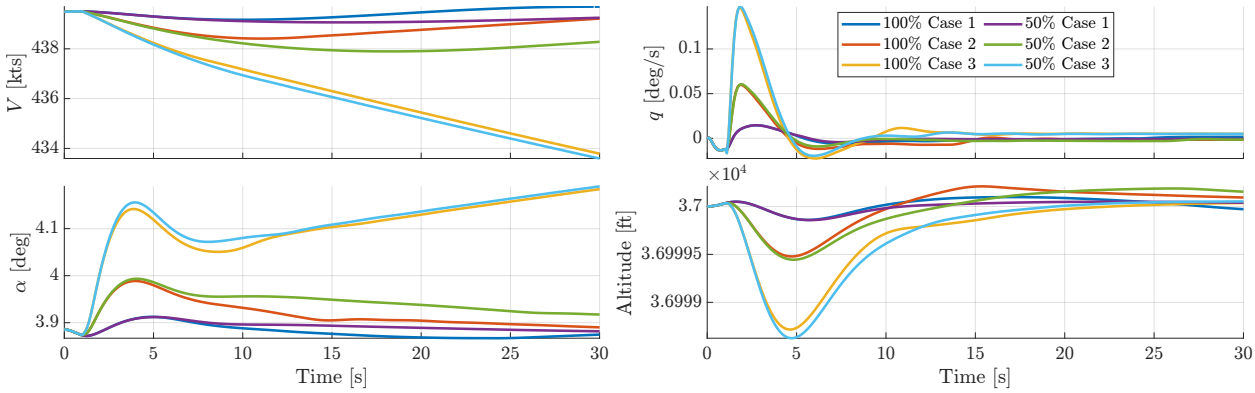


Figure 14: Longitudinal closed loop response to wingfan failures.

As Figure 14 illustrates, in the case where all eight of the left side wingfans fail, the system is no longer capable of producing enough thrust to maintain cruise conditions and begins pitching up and losing airspeed. When the full left side of wingfans fail, the framework is able to reject the lateral disturbance even though the aircraft is no longer capable of producing enough thrust to maintain steady level flight. The closed loop control effector usage during wingfan failures can be found in Figure B2 of Appendix B.

This evaluation case demonstrates that the closed loop framework is resilient to faults in the DEP system. This functionality is important to the SUSAN project because it will contribute to determining the reliability and failure modes of different configurations of the SUSAN aircraft.

7 Conclusions

In this work, a framework for evaluating distributed electric propulsion in the SUSAN Electrofan Aircraft is presented. This framework enables integrated performance analysis of different aircraft

configurations resulting from changes to aerodynamic models, propulsion systems, and control systems. The framework is validated with several evaluation cases that demonstrate its capability to handle new aircraft configurations. The results generated from this framework suggest that pseudoinverse allocation with feedforward integration between the control and propulsion systems may improve the performance for distributed electric propulsion system vehicles. The flexibility of the framework is useful for evaluating the performance of closed loop designs and will enable future trade studies for distributed electric propulsion in the SUSAN vehicle.

Future work will extend the framework to incorporate the dynamics of the electrical system. This addition will enable analyses of the impact of battery-aware control allocation on the performance of the closed loop system. Furthermore, individual subsystem models will be improved, for example improving models of high lift devices will allow analysis of closed loop performance during terminal flight phases. Under these flight conditions, control surfaces are less effective and wingfans are producing reduced thrust. These improvements create an opportunity for the framework to test algorithms which augment the control of the aircraft with DEP in different regions of the flight envelope.

Acknowledgments

This work is funded by Convergent Aeronautics Solutions with the support of the Transformational Tools and Technologies Project as part of the Transformative Aeronautics Concepts Program within the NASA Aeronautics Mission Directorate.

References

1. FAA, "United States 2021 Aviation Climate Action Plan," [Online; accessed June-2023], 2021. URL https://www.faa.gov/general/2021-united-states-aviation-climate-action-plan.

2. Jansen, R., Kiris, C. C., Chau, T., Machado, L. M., Duensing, J. C., Mirhashemi, A., Chapman, J., French, B. D., Miller, L., Litt, J. S., et al., "Subsonic Single Aft Engine (SUSAN) Transport Aircraft Concept and Trade Space Exploration," *AIAA SCITECH 2022 Forum*, 2022, pp. AIAA 2022–2179.

3. Chau, T., Kenway, G., and Kiris, C. C., "Conceptual Exploration of Aircraft Configurations for the SUSAN Electrofan," *AIAA SciTech 2022 Forum*, 2022, pp. AIAA 2022–2181.

4. Hueschen, R. M., "Development of the Transport Class Model (TCM) aircraft simulation from a subscale Generic Transport Model (GTM) simulation NASA/TM–2011-217169," Tech. rep., NASA, 2011. URL https://ntrs.nasa.gov/citations/20110014509.

5. McDonald, R. A., and Gloudemans, J. R., "Open Vehicle Sketch Pad: An Open Source Parametric Geometry and Analysis Tool for Conceptual Aircraft Design," *AIAA SCITECH 2022 Forum*, 2022, pp. AIAA 2022–0004.

6. Litt, J. S., Sowers, T. S., Buescher, H., and Jansen, R., "Implementation Approach for an Electrified Aircraft Concept Vehicle in a Research Flight Simulator," *AIAA SCITECH 2022 Forum*, 2022, pp. AIAA 2022–2306.

7. Chapman, J. W., Lavelle, T. M., May, R. D., Litt, J. S., and Guo, T.-H., "Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) User's Guide," Tech. rep., NASA, 2014.

8. Bell, M. E., and Litt, J. S., "Electrical Modeling and Thermal Analysis Toolbox (EMTAT) User's Guide NASA/TM–2018-0002976," Tech. rep., NASA, 2018. URL https://ntrs.nasa.gov/citations/20180002976.

9. Halila, G. L. O., Pedreiro, L. N., Molina, E. S., and Savio, M. C., "Efficient Aerodynamic Analysis for Propeller Aircraft Including Viscous Effects," *AIAA SCITECH 2022 Forum*, 2022, pp. AIAA 2022–0429.

10. Morelli, E. A., and Klein, V., *Aircraft system identification: Theory and practice*, Sunflyte Enterprises, 2016.

11. Levine, W. S., *The Control Handbook*, CRC press, 2018.

12. The MathWorks Inc., "Tune fixed-structure control systems modeled in MATLAB," [Online; accessed June-2023], 2023. URL https://www.mathworks.com/help/control/ref/dynamicsystem.systune.html.

Appendix A

Setup

The framework presented in this work is dependent on linear models for the airframe, propulsor, and actuators as well as steady state data on the propulsion system. These models and the steady state data are automatically generated through a series of setup scripts. This section outlines said scripts and the order in which they need to be executed. It should be noted that some of these scripts are computationally expensive but only need to be run when changes are made to the aircraft.

1. `SUSAN_AF_LinMod` creates linear models of the airframe across the flight envelope with the option to generate linear models for different variations of the aerodynamic model. The user specifies Mach number and altitude vectors in addition to a set of aerodynamic models. The exact logic for this script as well as all the following scripts is detailed in Appendix C.
2. `SUSAN_PropMap` then records steady state operating data for the propulsion system across the flight envelope. Throttle sweeps are performed at each operating point and the steady state values for thrust, rotational velocity, and actuator usage are recorded. This data is then used for the feed-forward terms described in Section 4 and to allow thrust commands to be mapped to rotational velocity commands.
3. `SUSAN_Prop_LinMod` creates linear models for the propulsion system across Mach number and altitude vectors. At each operating point, the system is subjected to a multi-sine excitation in both net thrust and differential thrust, and system identification is done with the response data. It is important that the previous script is run before this one so that the feed-forward terms have accurate data to work with.
4. `SUSAN_CleanConnect` connects the linear models generated by the previous scripts to create linear models of the entire system across the flight envelope.
5. `SUSAN_Autotune` Then generates the gain schedule to be used by the inner loop controller. The connected system is separated into single input single output models for each axis and tunable PID controllers are added to each of these models. The MATLAB function systune is then used to find the values for the gain schedule by optimizing to a user defined set of criteria.

Appendix B

Figures

This section of the appendix contains supplemental figures. Figure B1 depicts how the bandwidth of the propulsion system varies across both altitude and Mach number. Figure B2 illustrates the usage of control effectors during the three wingfan failure scenarios.

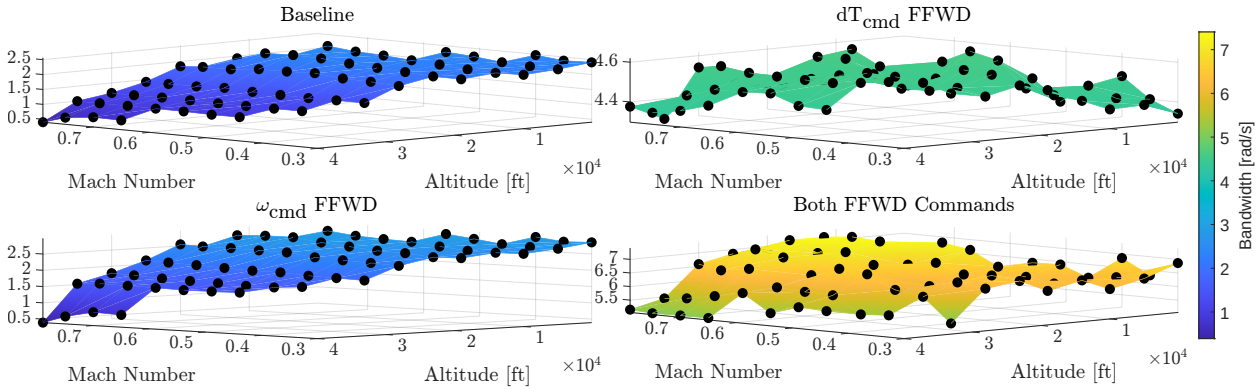


Figure B1: Surface plots of the propulsion system's bandwidth across altitude and Mach number

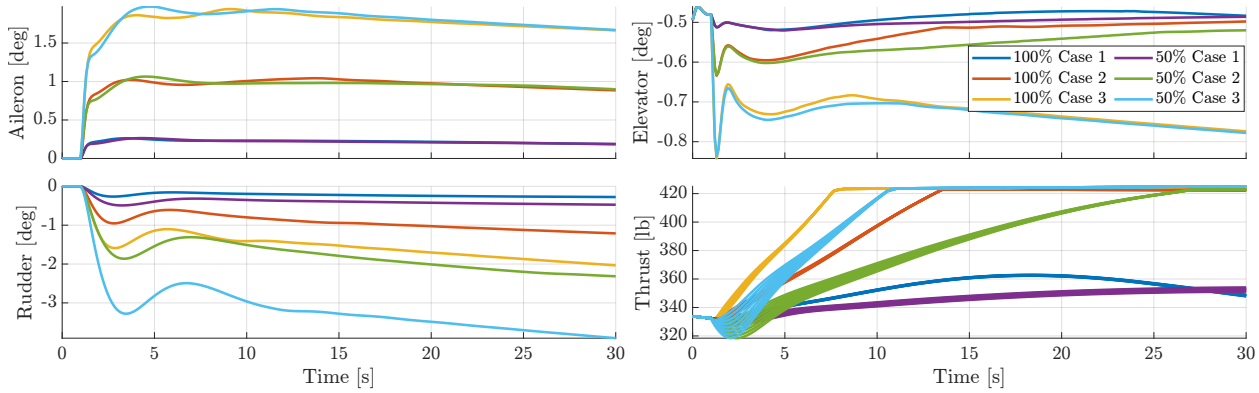


Figure B2: Effector usage during wingfan failures.

Appendix C

Scripts

The scripts presented in this section of the appendix are pseudo-code representations of the scripts described in Appendix A. These scripts are all written in MATLAB® and the pseudo-code reflects the syntax used by the MATLAB® language.

Algorithm 1: SUSAN_AF_LinMod.m

Input: $(ALT, MACH, adb\_num)$
For when changes are made to the airframe model

ALT: Vector of altitudes to linearize at [ft]
MACH: Vector of Mach numbers to linearize at
adb_num: Number of aerodynamic databases (adb) being linearized

for $ii = 1 : adb\_num$ **do**
  for $jj = 1 : Length(ALT)$ **do**
    for $kk = 1 : Length(MACH)$ **do**

      Iterates through permutations of altitude, Mach number, aerodynamic database

      if *ALT(jj),MACH(kk) are within flight envelope* **then**
        load(**ADB**(**ii**))
        trim_gtm(**Alt**(**jj**), **Mach**(**kk**)) [4]
        AF_LinMod(**ii**, **jj**, **kk**) = **SUSAN_Linearize**

return *AF_LinMod*

Algorithm 2: SUSAN_PropMap.m

Input: $(ALT, MACH)$
For when changes are made to the propulsion model

ALT: Vector of altitudes to record data from [ft]
MACH: Vector of Mach numbers to record data from

for $ii = 1 : Length(ALT)$ **do**
  for $jj = 1 : Length(MACH)$ **do**
    Iterates through permutations of altitude, and Mach number

    setupFS(**ALT**(**ii**), **MACH**(**jj**))
    Initializes propulsion model
    Prop_Data(**ii**, **jj**) = **sim**(**SUSAN_Prop**, **PLA_in**)
    simulates a throttle sweep, recording steady state data for each throttle setting
return *Prop_Data*

Algorithm 3: SUSAN_Prop_LinMod.m

Input: $(ALT, MACH)$
For when changes are made to the propulsion model

ALT: Vector of altitudes to linearize at [ft]
MACH: Vector of Mach numbers to linearize at

for $ii = 1 : Length(ALT)$ **do**
  for $jj = 1 : Length(MACH)$ **do**
    Iterates through permutations of altitude, and Mach number

    if *ALT(jj),MACH(kk) are within flight envelope* **then**
      load(**ADB**(**ii**))
      trim(**Alt**(**jj**), **Mach**(**kk**))
      Prop_Response(**ii**, **jj**) = **sim**(**SUSAN_Prop**, **Excite**)
      simulates a multi-sine excitation, recording time history of response
      Prop_LinMod = **sysid**(**Prop_Response**)
      creates a linear model based on response data
return *Prop_LinMod*

Algorithm 4: SUSAN_CleanConnect.m

Input: $(AF\_LinMod, Prop\_LinMod)$

Connects linear models of airframe, actuators, and propulsion system

AF_LinMod: Set of airframe linear models across altitude, Mach number, and aerodynamic database

Prop_LinMod: set of propulsion system linear models across altitude and Mach number

for $ii = 1 : adb\_num$ **do**
  for $jj = 1 : Length(ALT)$ **do**
    for $kk = 1 : Length(MACH)$ **do**

      Iterates through permutations of altitude, Mach number, aerodynamic database

      $\mathbf{SYS(ii, jj, kk) = connect(Prop\_LinMod(jj, kk), AF\_LinMod(ii, jj, kk))}$
      Connects outputs of propulsion model to inputs of airframe model
      $\mathbf{alloc = pseudoinverse(AF\_LinMod.B)}$
      Finds static control allocation at current operating point
      $\mathbf{SYS\_alloc(ii, jj, kk) = connect(alloc, Prop\_LinMod(jj, kk), AF\_LinMod(ii, jj, kk))}$
      Connects control allocation outputs to inputs of overall system

return *SYS,SYS_alloc*

Algorithm 5: SUSAN_Autotune.m

Input: $(SYS\_alloc, Tuning\_Goals)$

Generates gain schedule by tuning inner loop response to tuning objectives

SYS_alloc: Set of connected linear models with control allocation implemented

Tuning_Goals: Set of user defined criteria for inner loop response

for $ii = 1 : adb\_num$ **do**
  for $jj = 1 : Length(ALT)$ **do**
    for $kk = 1 : Length(MACH)$ **do**

      Iterates through permutations of altitude, Mach number, aerodynamic database
      for *each axis* **do**

        $\mathbf{C = tunablepid}$
        Creates tunable pid controller
        $\mathbf{Closed\_Loop = connect(C, SYS\_alloc(ii, jj, kk))}$
        creates closed loop system incorporating controller
        $\mathbf{PID(ii, jj, kk) = systune(CL, Tuning\_Goal)}$
        tunes closed response to tuning objectives

return *PID*

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704–0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-07-2023	Technical Memorandum	

4. TITLE AND SUBTITLE
A Framework for Evaluating Distributed Electric Propulsion on the SUSAN Electrofan Aircraft

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Nicholas C. Ogden
Armstrong Flight Research Center, Edwards, California

Andrew Patterson
Langley Research Center, Hampton, Virginia

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
533127.02.22.07.05

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, Virginia 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/TM–20230009523

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This work presents a framework for evaluating models and algorithms for Distributed Electric Propulsion (DEP) on the SUSAN Electrofan Aircraft. Throughout the development of the SUSAN aircraft, the performance of various configurations of the aircraft will need to be analyzed. However, the static behavior alone is not sufficient to describe the performance of these configurations. Therefore, simulation with fully integrated subsystem models is required. The proposed framework considers the vehicle aerodynamic, propulsion, and control subsystems. The presented framework automatically generates control laws for any vehicle configuration in response to changes in these subsystems. To compare these different vehicle configurations, various time and frequency domain performance metrics are compared.

15. SUBJECT TERMS
keywords

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			HQ-STI-infodesk@mail.nasa.gov
U	U	U	UU	35	19b. TELEPHONE NUMBER *(Include area code)* (757) 864-9658