Integrated Propulsion System Modeling for Advanced Aircraft Using Multidisciplinary Design Optimization

Anushka Tahiliani
MIT Gas Turbine Laboratory
Advisor: Prof. Masha Folk
Collaborators: Hannah Hajdik & Alejandra Uranga (Electra.aero), Andrew Lamkin (Supercritical)

Advanced Modeling & Simulation (AMS) Seminar Series

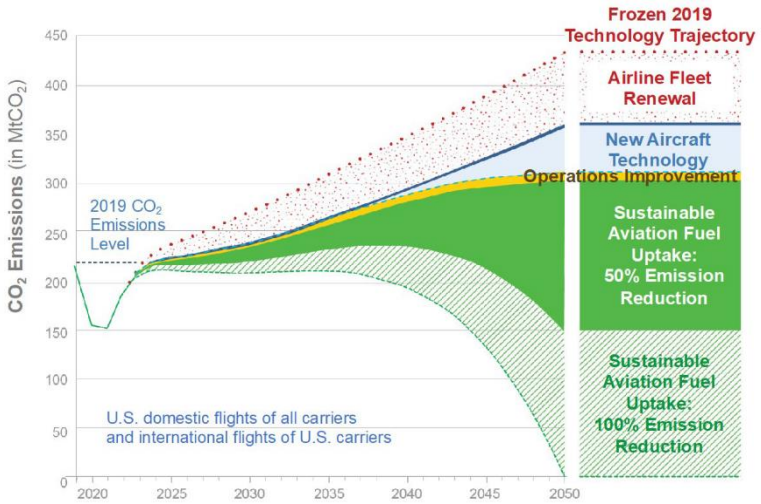
NASA Ames Research Center, August 6th, 2026

Outline

- **Background & Motivation**
- Research Contributions
  - On-design robustness
  - Off-design robustness
  - Integrated Framework
  - Gas-Turbine Modeling Fidelity for Aircraft MDO
- Demonstration & Results
- Conclusion

NASA's Advanced Aircraft Concepts for Environmental Substantiality (AACES) 2050 Program

Why Now



- Sustainability, noise, cost are **existential threats** to aviation
- US aviation faces **increasing competition**
- Conventional tube and wing aircraft are maxed out with **diminishing returns**

Why Electra



- Electra is the industry **leader in hybrid-electric aircraft**
- **Full-electrification is not viable** for 2050-era airliners
- **Supported by world-class team** including UM, MIT, UCI, American Airlines, Honeywell, Lockheed Martin, and Hinetics

What Was The Scope



- Conceptual design of **100+ passenger aircraft**
- Capable of achieving **conventional payload & range**
- Evaluation of the **opportunity for electrification and airframe integration** technologies

Integrated Propulsion Enables New Architectures and New Design Couplings

- **Propulsion system trend:** advanced concepts increasingly integrate propulsion with the airframe and onboard power systems, enabling distributed propulsion, blown lift, BLI, and decoupled thrust/power architectures.
- **Aircraft-level design implication:** engine-design intuition breaks down when propulsive-efficiency and integration synergies trade against losses, weight, TMS, and operability limits.

Takeaway: **aircraft, propulsion, and mission must be designed optimized together.**



D8 Double Bubble

BLI-enabled aft propulsion

~37% fuel-burn reduction reported



NASA SUSAN

DEP + aft BLI architecture

17–27% fuel-burn reduction studied

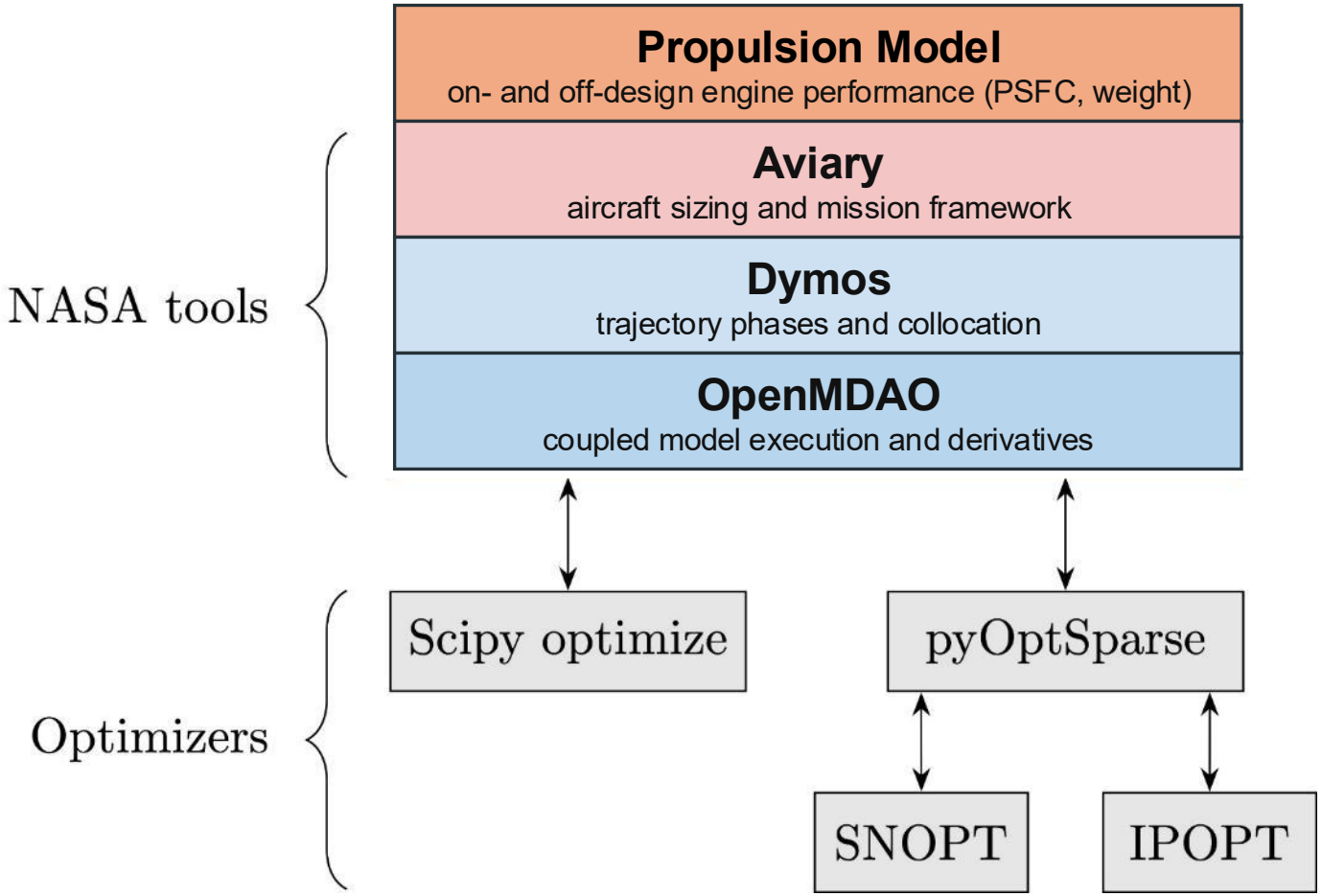


Electra EL2/9

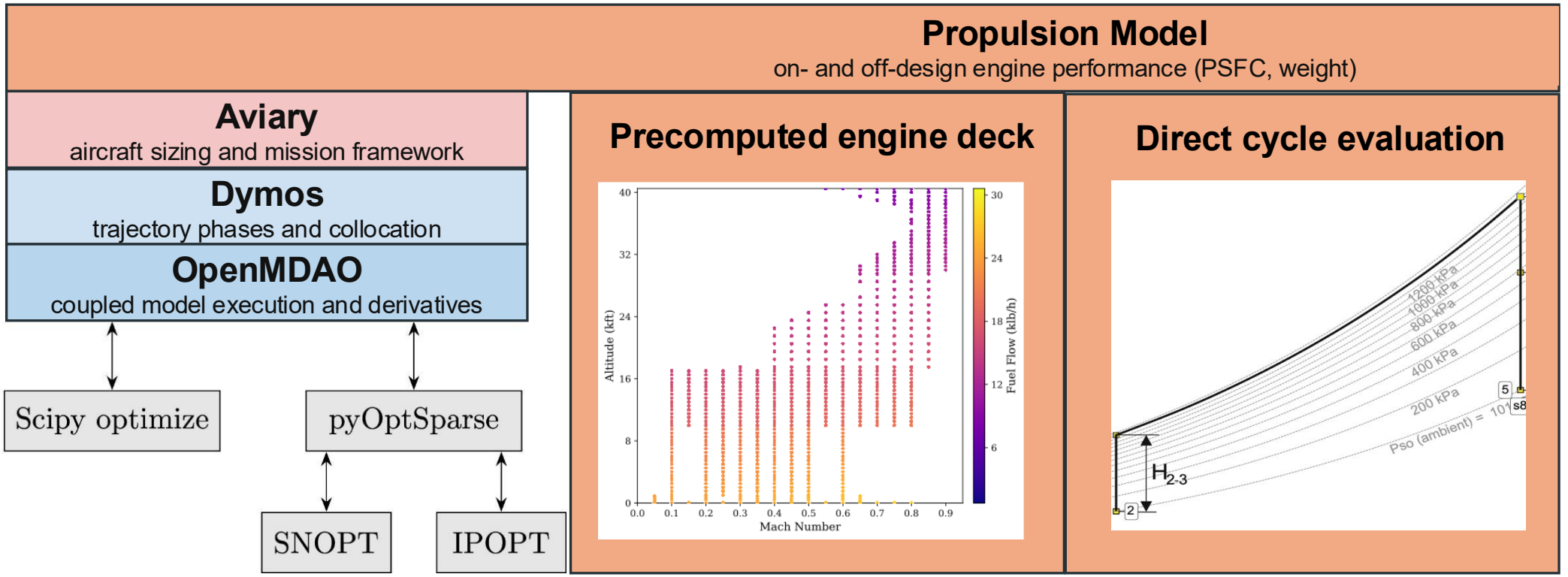
Blown lift + hybrid-electric propulsion

~40% lower fuel burn on 100-mi routes

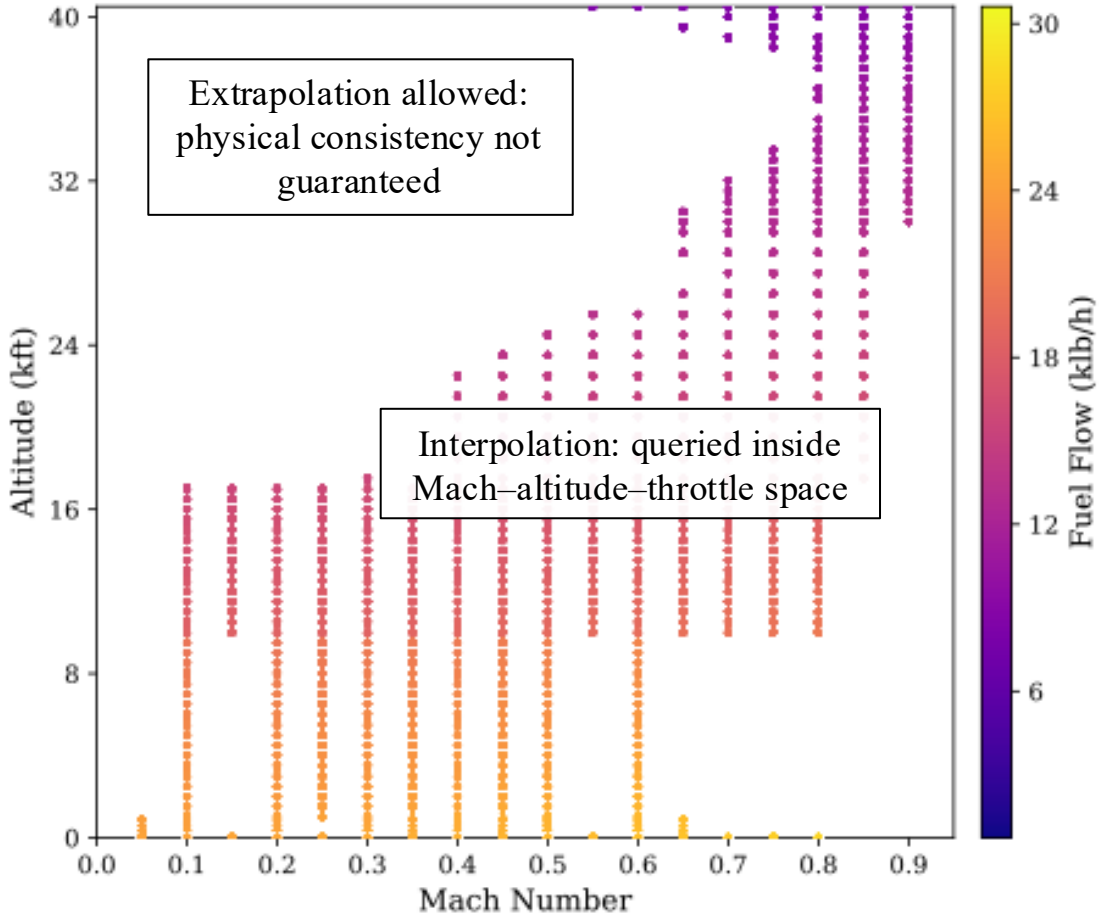
MDO framework is built on established modular open source tools for gradient-based optimization



MDO Framework Supports Multiple Propulsion-Modeling Fidelities



Precomputed Engine Deck is Efficient but Limits Design Space



Turbofan deck slice at a fixed throttle level showing predicted fuel flow across Mach–altitude space

+ **Efficient aircraft-level interface**
  - fast performance queries from precomputed engine data
- **Fixed cycle representation**
  - engine architecture and internal cycle variables are prescribed
- **Scaling-based design changes**
  - thrust/power scaling inherits deck PSFC trends, which breaks down when cycle variables or power extraction change
- **Limited MDO scalability**
  - added propulsion/electrification design variables require many new decks, causing combinatorial growth

Thermodynamic Cycle Model Expands Design Space but is not Robust

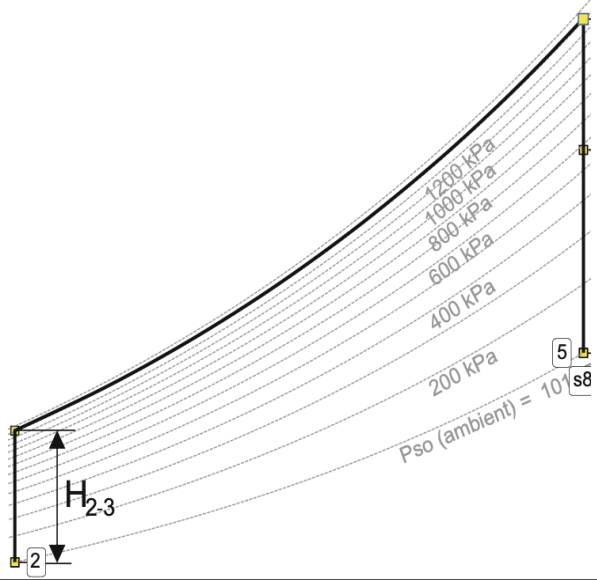
On-Design Inputs:
Power/Thrust
Freestream Conditions
Cycle Param. (BPR, OPR, etc.)

pyCycle

Cycle model with analytic derivatives

Built on OpenMDAO

Design & off-design capabilities

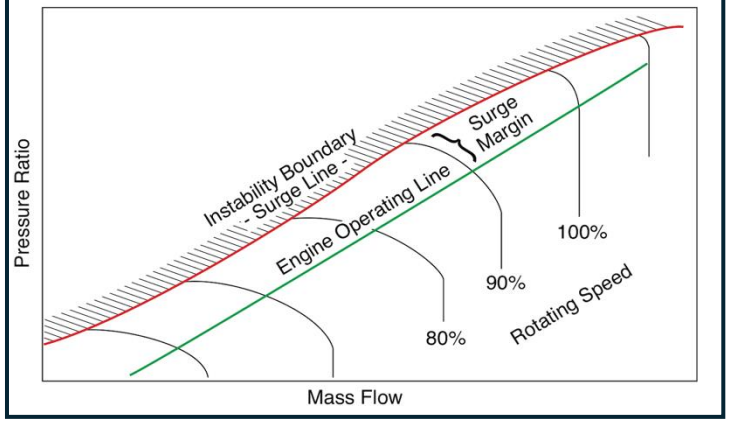


On-Design Outputs:
Map Scalars, Engine Areas

Thermodynamic Cycle Model Expands Design Space but is not Robust

On-Design Inputs:
Power/Thrust
Freestream Conditions
Cycle Param. (BPR, OPR, etc.)

Off-Design Inputs:
Freestream Conditions, Throttle
Aerodynamic Maps

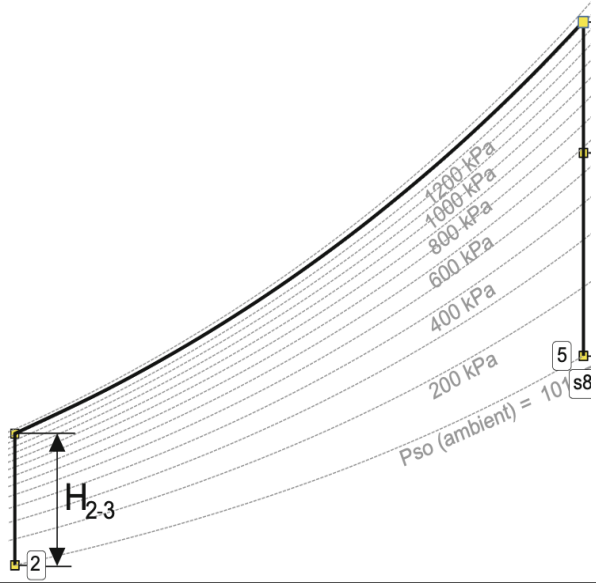


pyCycle

Cycle model with analytic derivatives

Built on OpenMDAO

Design & off-design capabilities



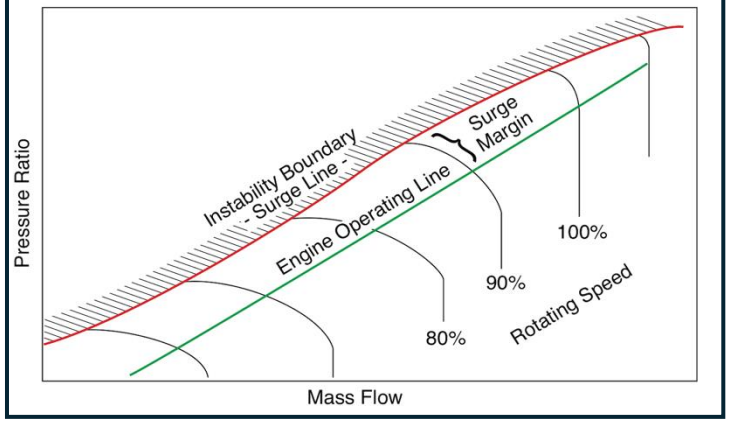
On-Design Outputs:
Engine Areas

Off-Design Outputs:
Fuel Flow, Shaft Power, Thrust

Thermodynamic Cycle Model Expands Design Space but is not Robust

On-Design Inputs:
Power/Thrust
Freestream Conditions
Cycle Param. (BPR, OPR, etc.)

Off-Design Inputs:
Freestream Conditions, Throttle
Aerodynamic Maps

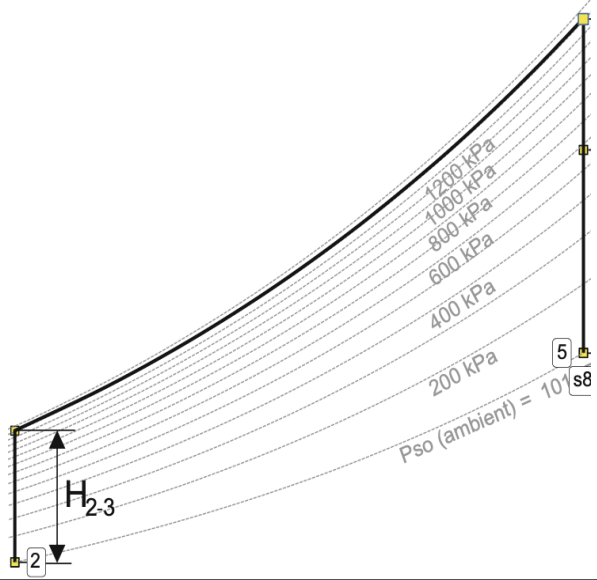


pyCycle

Cycle model with analytic derivatives

Built on OpenMDAO

Design & off-design capabilities



On-Design Outputs:
Engine Areas

Off-Design Outputs:
Fuel Flow, Shaft Power, Thrust

Goals:

1. *Robustly co-optimize the propulsion system, airframe, and mission within an aircraft MDO framework.*
2. *Define the gas-turbine modeling fidelity required to produce physically meaningful aircraft and mission outcomes.*

Outline

- Background & Motivation
- **Research Contributions**
  - On-design robustness
  - Off-design robustness
  - Integrated Framework
  - Gas-Turbine Modeling Fidelity for Aircraft MDO
- Demonstration & Results
- Conclusion

Key Contributions

1. **Enables broad on-design propulsion-system optimization**
   - **Optimizes thrust/power, BPR, OPR, power extraction, and electrification level**
   - **Engine sizing flight conditions treated as design variables for hybrid-electric operation**

2. Robust pyCycle convergence from full throttle down to 20% part power

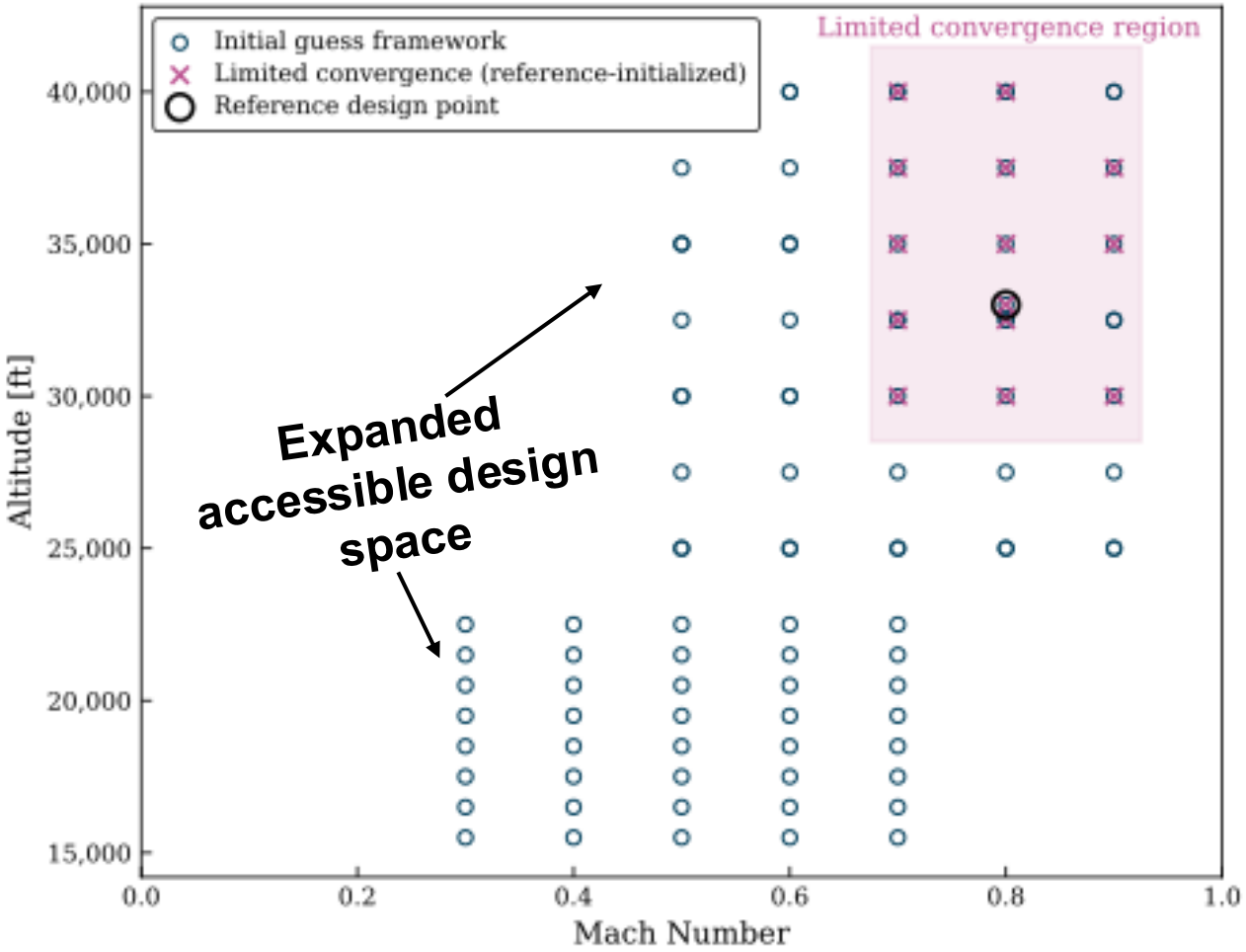
3. pyCycle integrated into aircraft-trajectory optimization framework

4. Establishes the gas-turbine modeling fidelity required for reliable aircraft- and mission-level optimization

Robust On-Design Initial Guess for pyCycle Eliminates Reliance on Lookup Tables

New reliable, physics-based initial guess model for pyCycle

- Simplified Brayton cycle model to determine mass flow and estimate engine states needed to meet target thrust/power
- Avoid need to prescribe difficult-to-estimate implicit states a priori
- Useful for engine analysis and optimization



Converged engine design optimizations across Mach-altitude space with and without initial guess

Key Contributions

1. Enables broad on-design propulsion-system optimization

2. **Robust pyCycle convergence from full throttle down to 20% part power**
   - **Support optimizer-driven electrification sweeps (all-engine → high electric-assist) operation across the mission**
   - **No need for manual initial guess matrices nor steering of optimizer**
   - **Black-box-like robustness unprecedented for an engine thermodynamic cycle models**

3. pyCycle integrated into aircraft-trajectory optimization framework

4. Establishes the gas-turbine modeling fidelity required for reliable aircraft- and mission-level optimization

Three Modeling Strategies Make Mission-Level Cycle Optimization Robust

MIT AEROASTRO

1 Off-Design Initialization

Initialize each mission point from physically scaled design-point cycle states.

Corrected-flow/speed scaling · throttle-based FAR and shaft-speed heuristics · failed-solve recovery

2 Physical Constraints

Prevent optimizer from exploring physically invalid engine states.

Positive core gross thrust · turbine/nozzle feasibility · SLS and low-Mach consistency · KS aggregation

3 Component Matching

Use bleed controls to maintain compressor operability during climb and part-power operation.

LPC operability bleed as a Dymos control · stall-margin and corrected-speed limits

Off-Design Initialization

- Improve off-design initial guesses using *guess_nonlinear*

- Use physics-based scaling laws for mass flow, BPR, FAR, and shaft speeds
  1. Use converged design-point states as the reference for each off-design solve
  2. Corrected-flow and -speed scaling for new freestream condition (Mach, alt)
  3. Throttle-based heuristic adjusts FAR, shaft speed, and BPR

- Use *restart_from_successful* for failed solve recovery from the last converged state

Physical Constraints

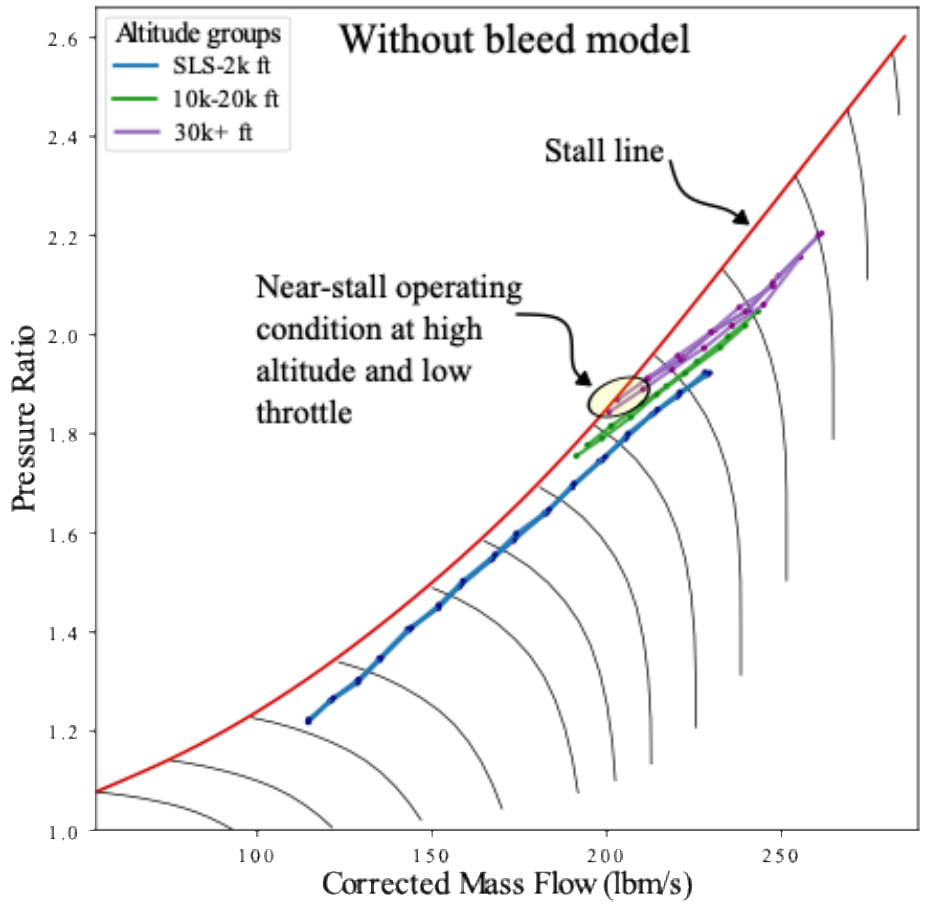
- KS aggregation $F_{core,\ gross} > 0$
  - Enforce physically valid core-nozzle operation

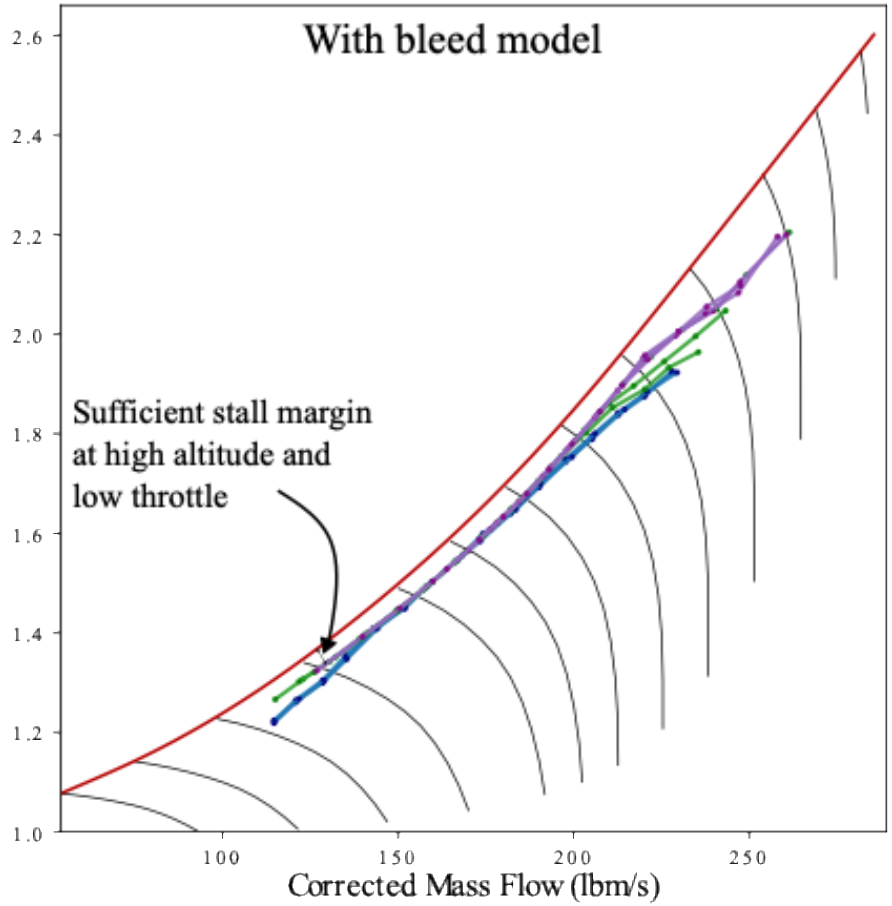
- KS aggregation Nc < 105%
  - Enforce corrected-speed limit on all compressor maps
  - Prevent over speeding

- $T_{t4max,SLS} / T_{t4max,TOC} > 1$ at alt < 10,000 ft
  - Single design-point sizing must remain feasible at SLS and low-Mach conditions for high BPR

LPC Bleed Extends Compressor Operability to 20% Part Power Across Mission Profile

- Implement LPC operability bleed using an IDF approach
  - Treat bleed as a Dymos control variable during climb
  - Enforce compressor map stall margin with a KS aggregation: $R_{line,lpc} \geq 1.15$





Key Contributions

1. Enables broad on-design propulsion-system optimization

2. Robust pyCycle convergence from full throttle down to 20% part power

3. **pyCycle integrated into aircraft-trajectory optimization framework**
   - **All mission nodes solved in parallel for computational efficiency**

4. Establishes the gas-turbine modeling fidelity required for reliable aircraft- and mission-level optimization

Framework Built on Aviary's Multi-Stage Design

PRE-MISSION

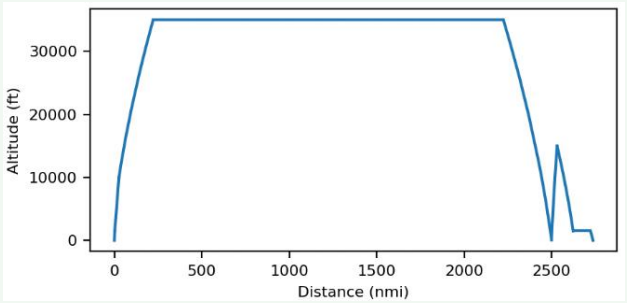
Determines Aircraft Sizing and Static Parameters



MISSION

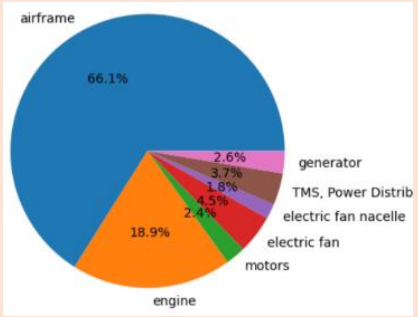
Evaluates the Aircraft Performance Over Mission Trajectory

Dymos-based mission analysis tracks fuel burn, battery state, energy use, and off-design engine behavior over time.

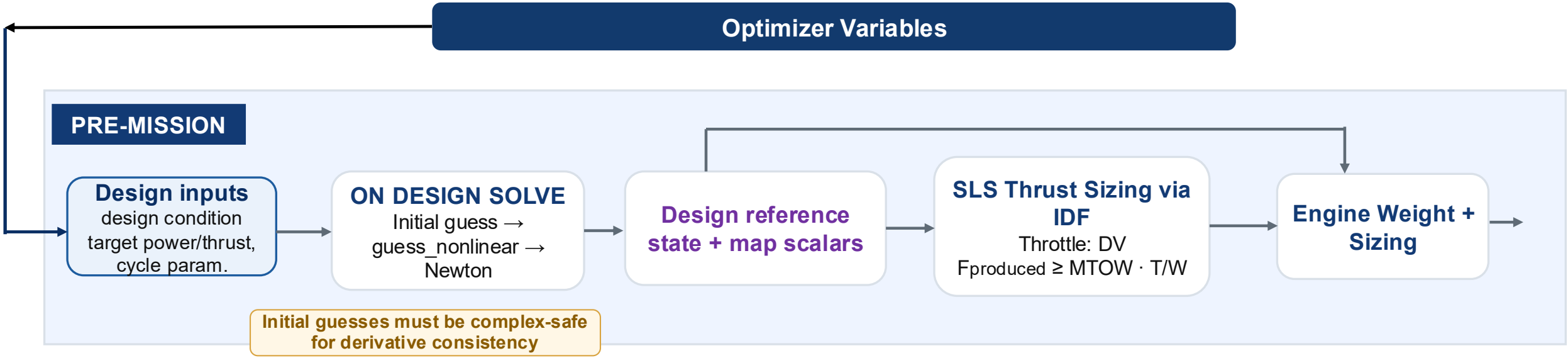


POST-MISSION

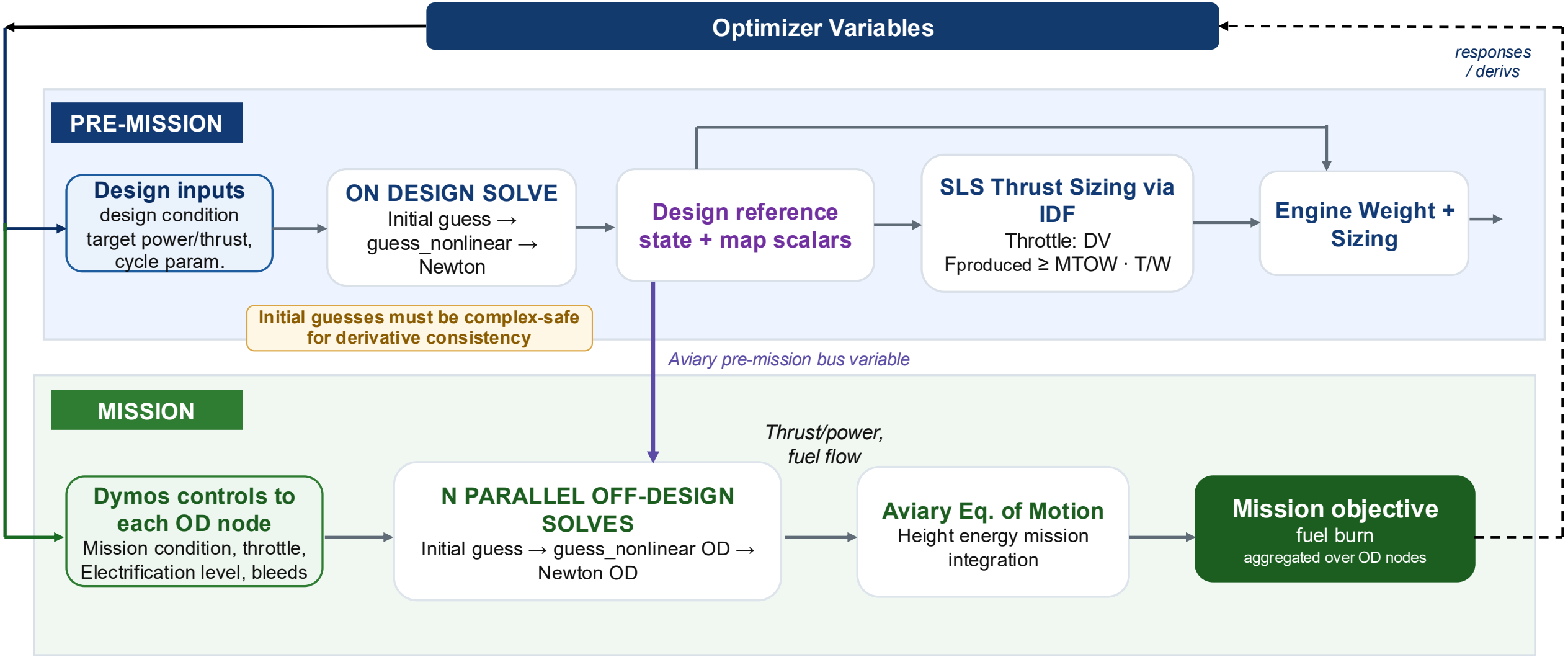
Calculates additional info. based on sizing and performance



Framework Built on Aviary's Multi-Stage Design



Framework Built on Aviary's Multi-Stage Design



Pre-mission sizes the engine; mission operates it across off-design nodes while the optimizer receives fuel-burn responses and derivatives.

Key Contributions

1. Enables broad on-design propulsion-system optimization
2. Robust pyCycle convergence from full throttle down to 20% part power
3. pyCycle integrated into aircraft-trajectory optimization framework

4. **Establishes the gas-turbine modeling fidelity required for reliable aircraft- and mission-level optimization**
   - **Some effects need to be captured:**
     - **Small core penalty**
     - **HPT cooling**
     - **Fan, LPC, LPT, nozzle weight**
   - **Other effects can be approximated:**
     - **Uniform PR can be assumed for fan core and bypass streams**
     - **Single compressor map can be scaled for use in any core size**
     - **Gas Generator weight**

Small Core Regime

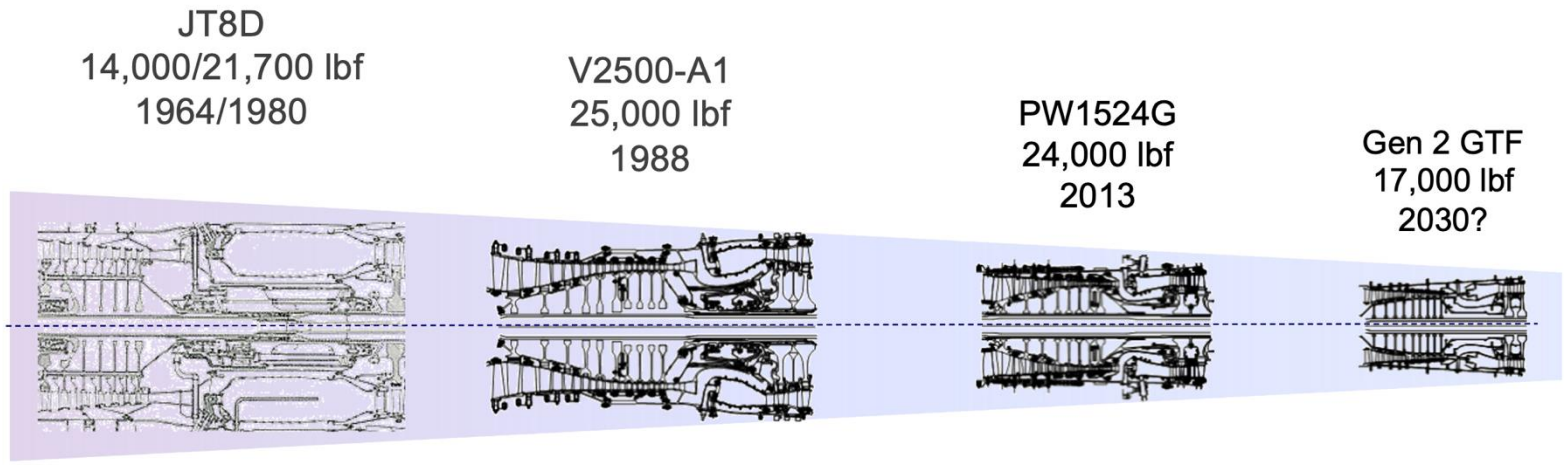
Trend towards small cores (< 5 lb/s HPC exit corrected flow relative to SLS reference)

- Push towards higher OPR and BPR to increase thermal efficiency ⇒ small last stages
- Ultra-efficient transports ⇒ less thrust and power ⇒ lower core mass flow
- Electrification with supplemental battery power ⇒ downsizing of engines
- Distributed power ⇒ several turbogenerators with lower power per engine

Implications:

- Degraded performance of all-axial compressors at these sizes needs to be captured
- Axi-centrifugal architectures may offer higher component efficiency

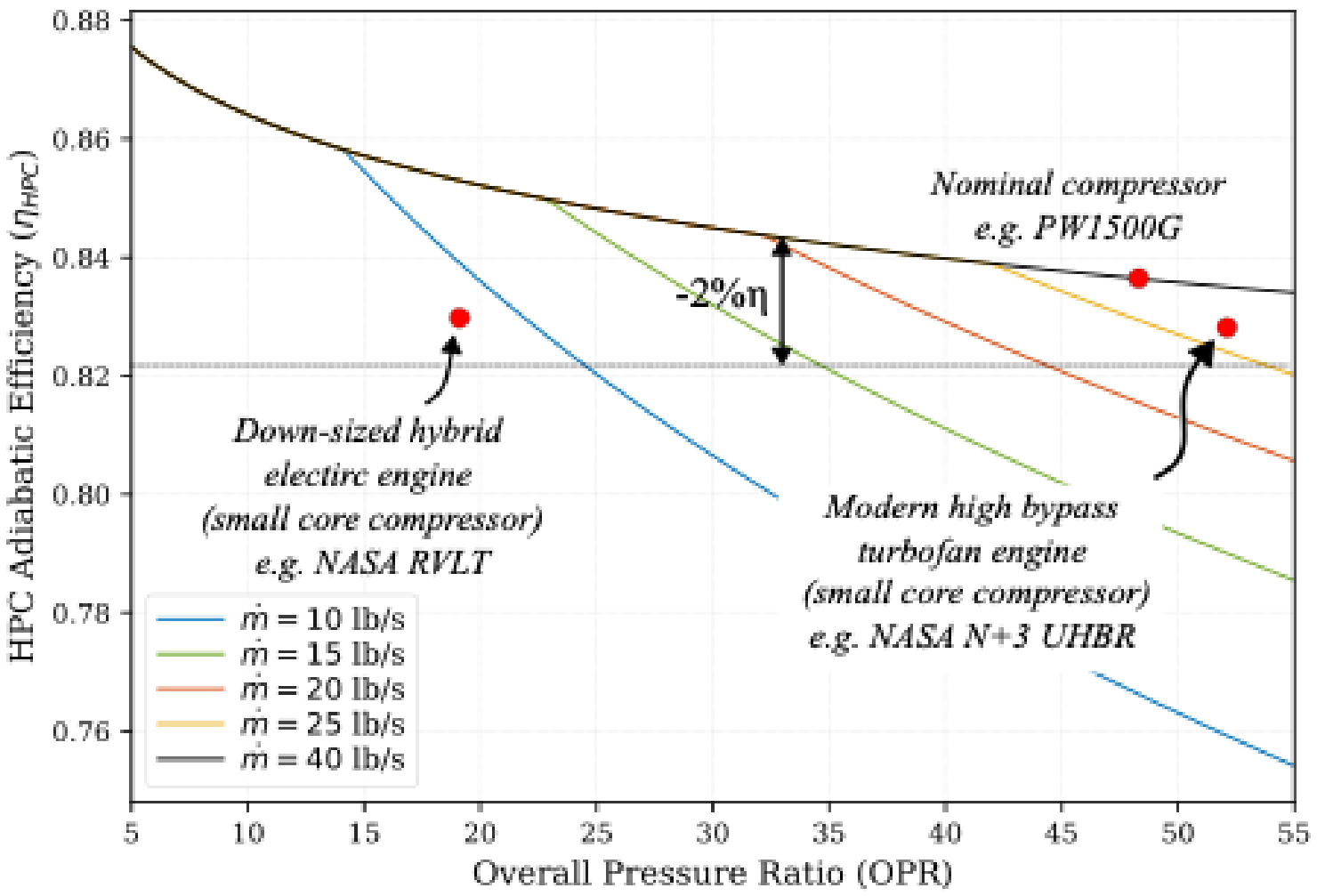
Trend towards lower thrust, smaller cores for given mission

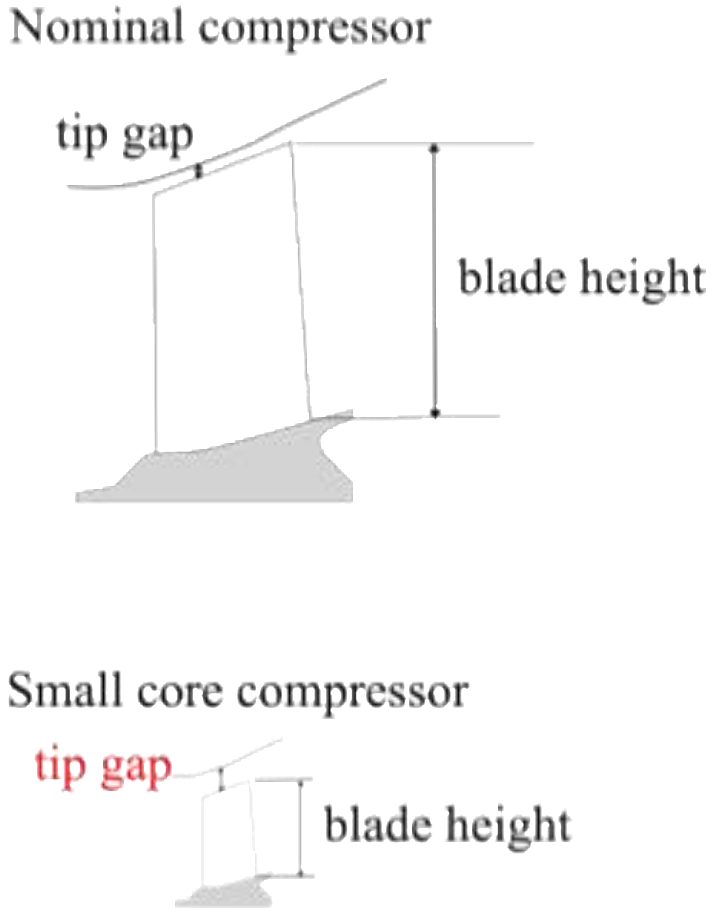


Source: Alan Epstein, Pratt & Whitney (modified in slide)

Tip-Clearance Penalties in Small-Core Compressors

Modeled as polytropic efficiency penalty applied to all HPC stages





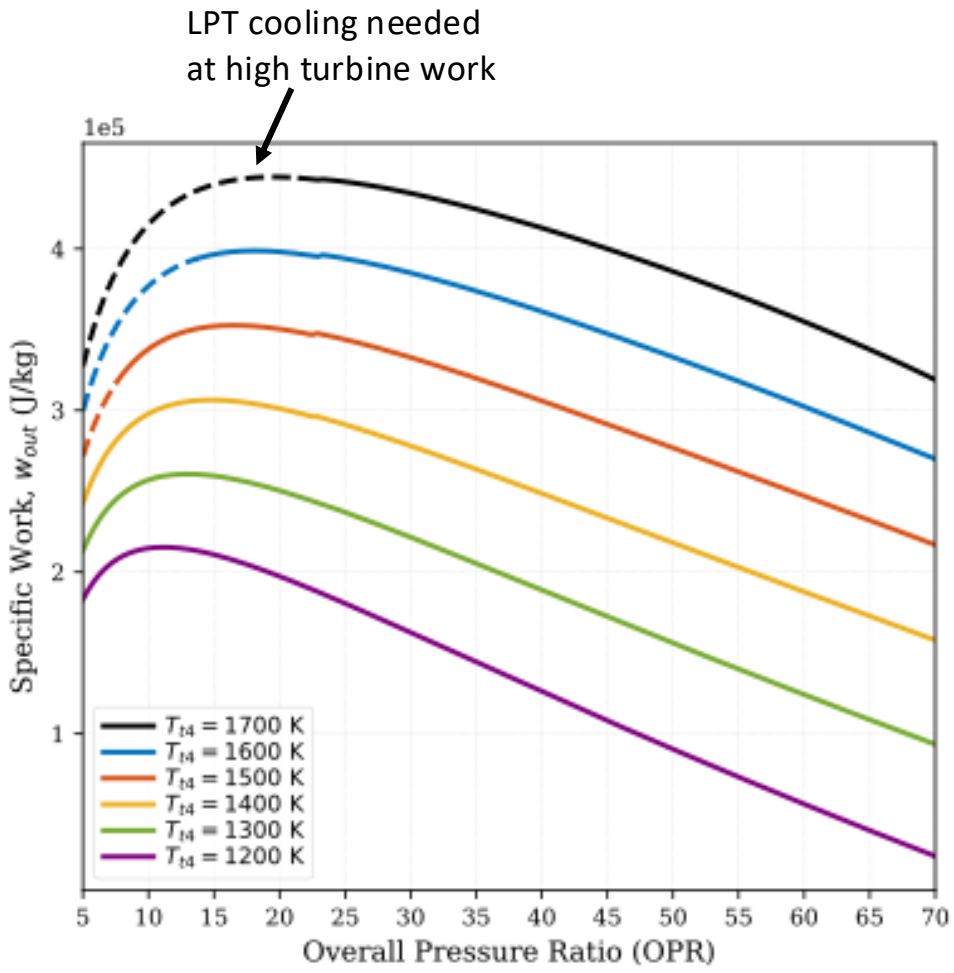
HP Turbine Cooling Model

- **Purpose:** accurately represent turbine cooling penalties and thermal feasibility at the high turbine inlet temperatures required for efficient, high-specific-work cycles

- **Scope**:
  - Model responsive to T_{t4}, OPR, coolant state, and off-design
  - only consider engines that avoid expensive cooling of LP Turbine

- Constraint and assumptions
  - LPT needs cooling when exit HPT > 1000 C (turbine gas temperature limit)
  - Film cooling in all stages (1 or 2)
  - Max HPT metal temperature ∝ technology level

Increasing Tt4 improves cycle efficiency and power, but gains are limited by uncooled LPT constraint

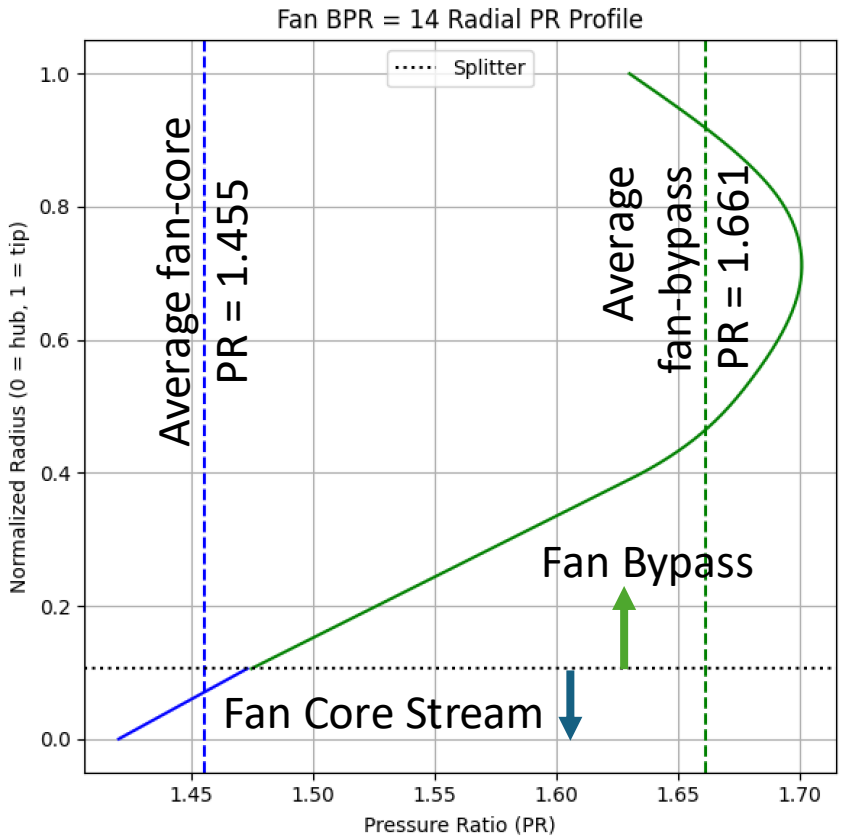


Fan Map Limitations & BPR Effects

Can we assume a uniform fan pressure ratio when modeling the fan core and bypass streams? ⇒ **YES**
Using different PR (split-PR fan model) versus uniform PR

- Changes aircraft fuel burn by less than 3%
- Core sizing and LPC/LPT impacts were secondary

Split-PR Fan Model



$$\frac{PR_{fan\ core}}{PR_{fan\ bypass}} = 0.876$$

Study details:

- Used radial profile of E³ fan map to inform a split-PR fan model
- Compared split-PR (1.661 PR for fan-bypass and 1.455 PR for fan-core) and uniform-PR (1.661 PR for both streams) results at constant thrust

Compressor Map Scaling and Aircraft-Level Impact

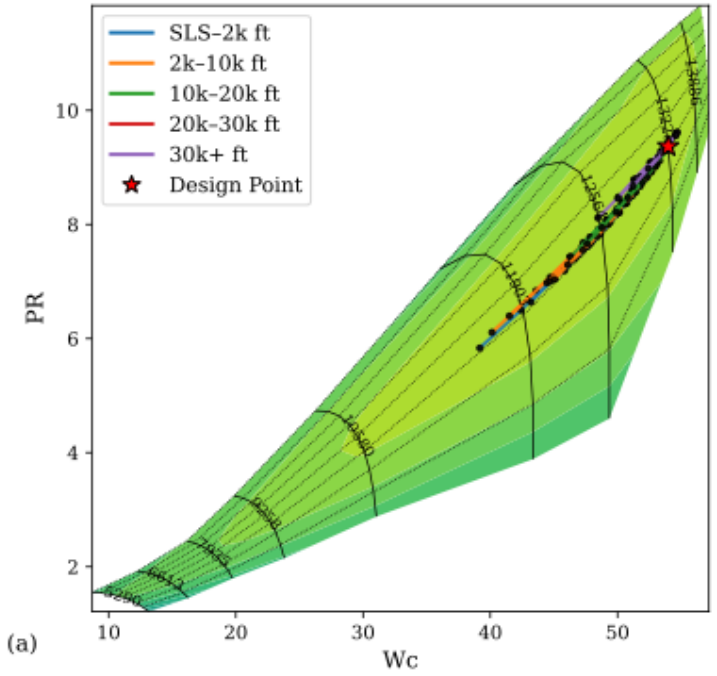
Can we scale compressor maps by large factors, in spite of the fact that blade size/shape changes alter map shape and efficiency? ⇒ **YES**

Using a scaled compressor map (1.8x, left) versus an appropriately-sized map (right) changes aircraft fuel burn by less than 0.5%.

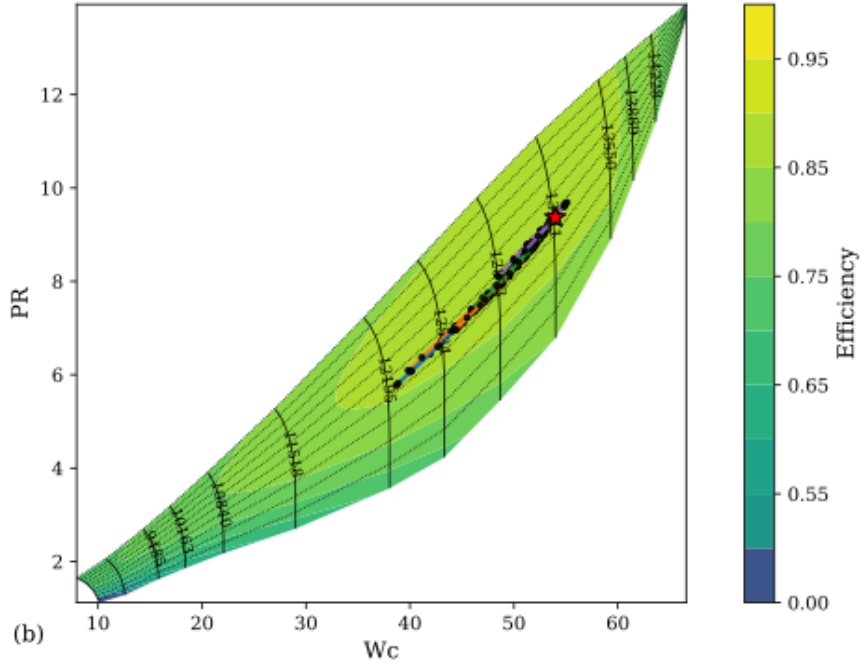
⇒ Scaling a single map is sufficient when considering aircraft-level effects

For the same target design PR = 9.37:

Scaled pyCycle AXI5 map
(1.8 scale factor on original: 1.2 < PR < 6, design PR = 5.2)



pyCycle CFM56 map
(no scaling, 1.1 < PR < 14, design PR = 9.37)

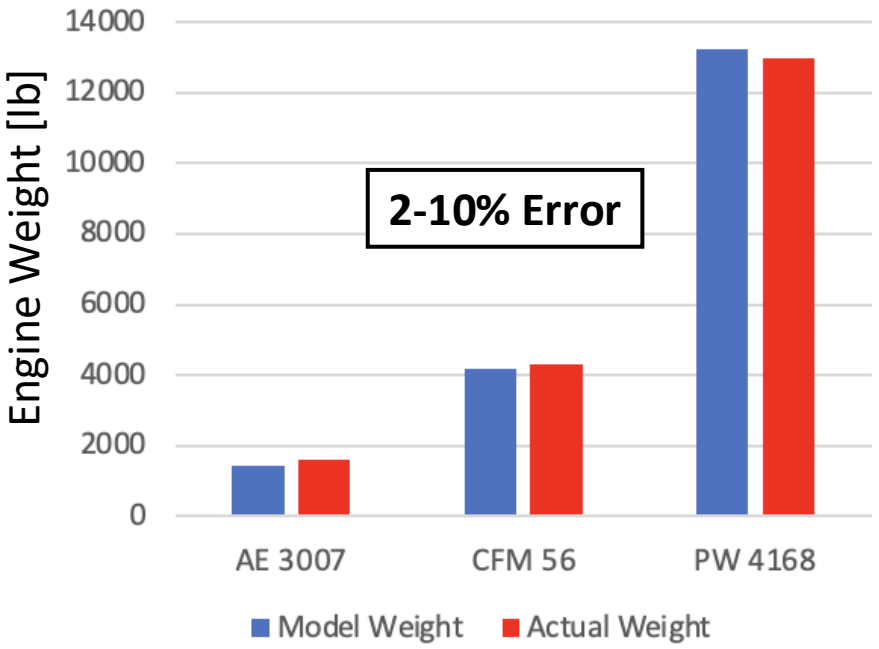


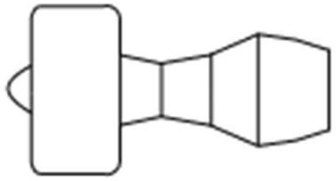
Challenge: Propulsion Weight Modeling

- No publicly accessible component-level engine weight model equivalent to WATE++.
- Thrust or Power-based regression models do not capture how propulsion architecture changes affect engine size and weight.
- Engine size must be penalized through geometry-sensitive terms, such as fan diameter, nacelle size, flow-path area, and component dimensions.
- Without these penalties, optimizations tend toward upper bounds: unrealistically high BPR and OPR designs.
- Nacelle drag provides some aircraft-level feedback, but it is insufficient for embedded or electrified turbogenerator architectures

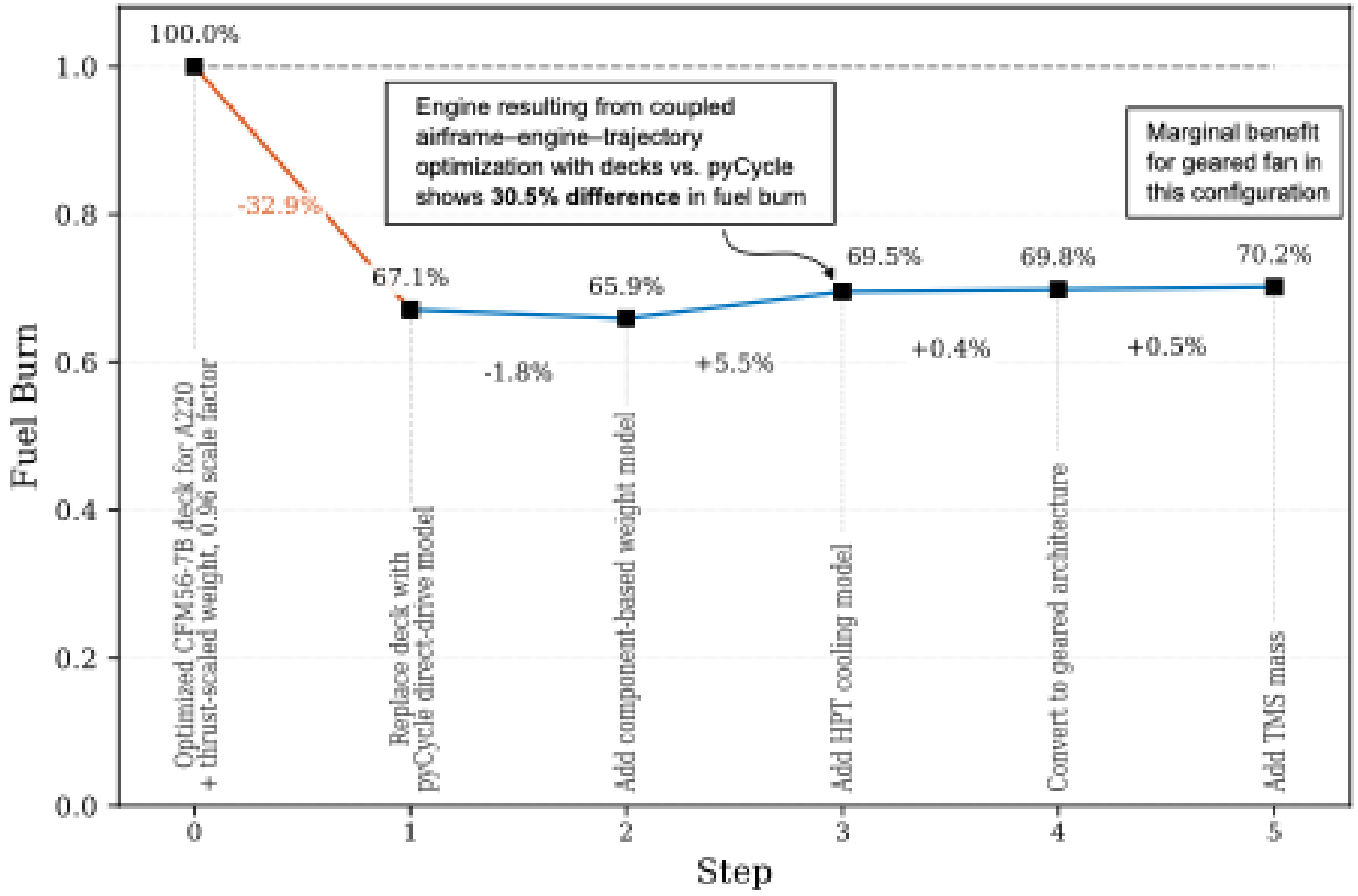
MDO-Compatible Engine Weight Model

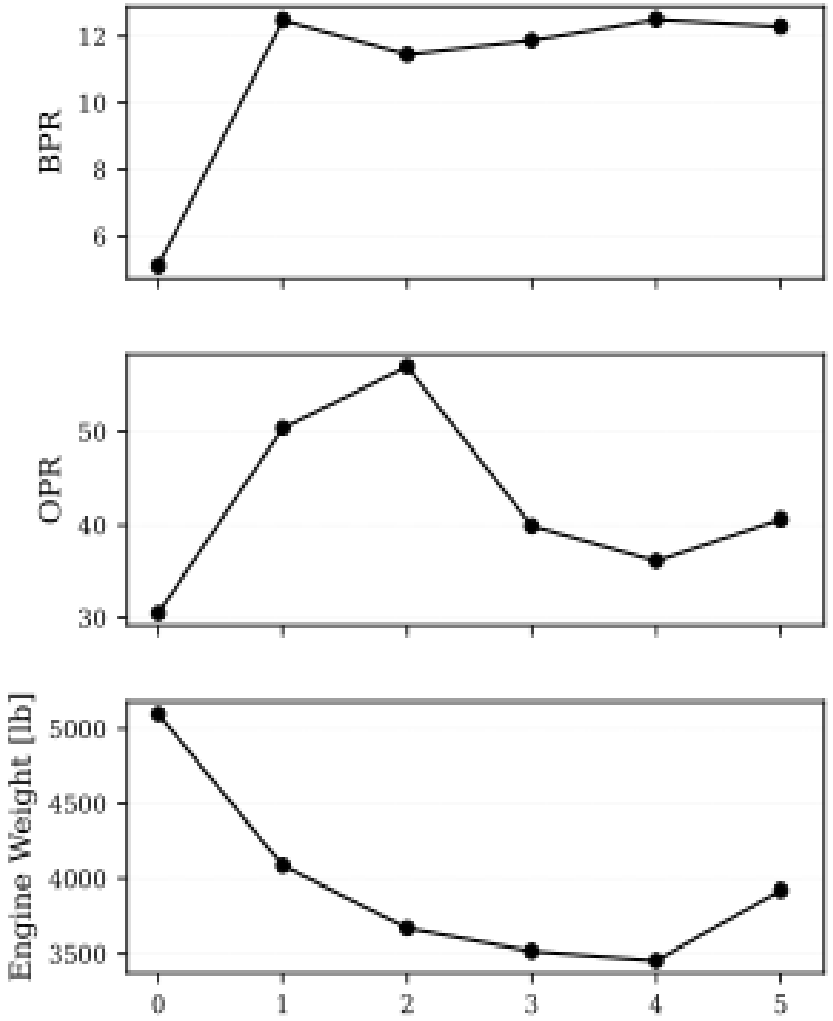
- Physics-informed, geometry-based weight buildup model used to estimate mass of LPC, LPT, fan, and nozzle (larger contributions)
  - Per-stage mass buildup includes rotor blades, stator, vanes, disks, outer casing, rotor drum, and containment (fan).
  - Max diameter + length estimated used in nacelle/fairing weight
- Gas generator and additional hardware (shaft, ducts, etc.)
  - Estimated via regression: 0.0777 kg/kW x maximum takeoff power





Fidelity Build-Up: Effects on the Optimized Cycle and Aircraft



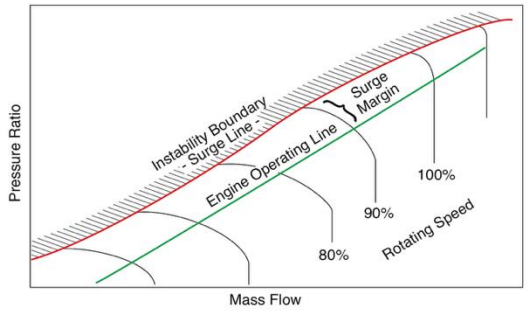


Engine Cycle Modeling for Aircraft MDAO Framework

On-Design Inputs:
Shaft Power/Thrust
Freestream Conditions
Cycle Param. (BPR, OPR, etc.)

Engine Layout:
Unified Propulsion System
Engine Architecture

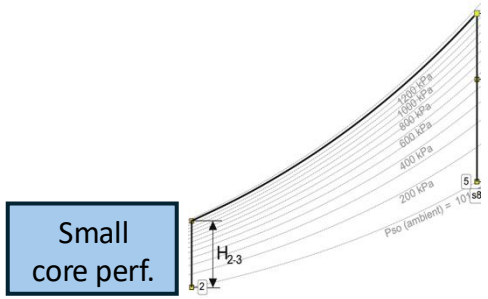
Off-Design Inputs:
Freestream Conditions, Throttle, Maps
Stall Margin Control, Constraints, T_{t4max} Ratio



Initial Guess

Initial Guess

pyCycle



HPT cooling

Power on/off take

Small core perf.

Integrated Generator Capability

On-Design Outputs:
Engine Areas

Off-Design Outputs:
Fuel Flow, Shaft Power,
Electrical Power

Weight & Geometry Model

Outputs:
Max Engine Diameter, Length
Weight, Number of Stages

Outline

- Background & Motivation
- Contributions
  - On-design robustness
  - Off-design robustness
  - Gas-Turbine Modeling Fidelity for Aircraft MDO
  - Integrated Framework
- **Demonstration & Results**
- Conclusion

Co-optimization Demonstration

Co-Optimize Airframe–Engine–Trajectory for Min Fuel Burn (2016 EIS assumptions)

- Aircraft: Initialized with CeRAS CSR-01, narrow-body, short-to-medium-range
- Trajectory: SLS, climb, top of climb, cruise segment (2,200 nmi)
- Objective: Minimize fuel burn

Co-optimization Demonstration

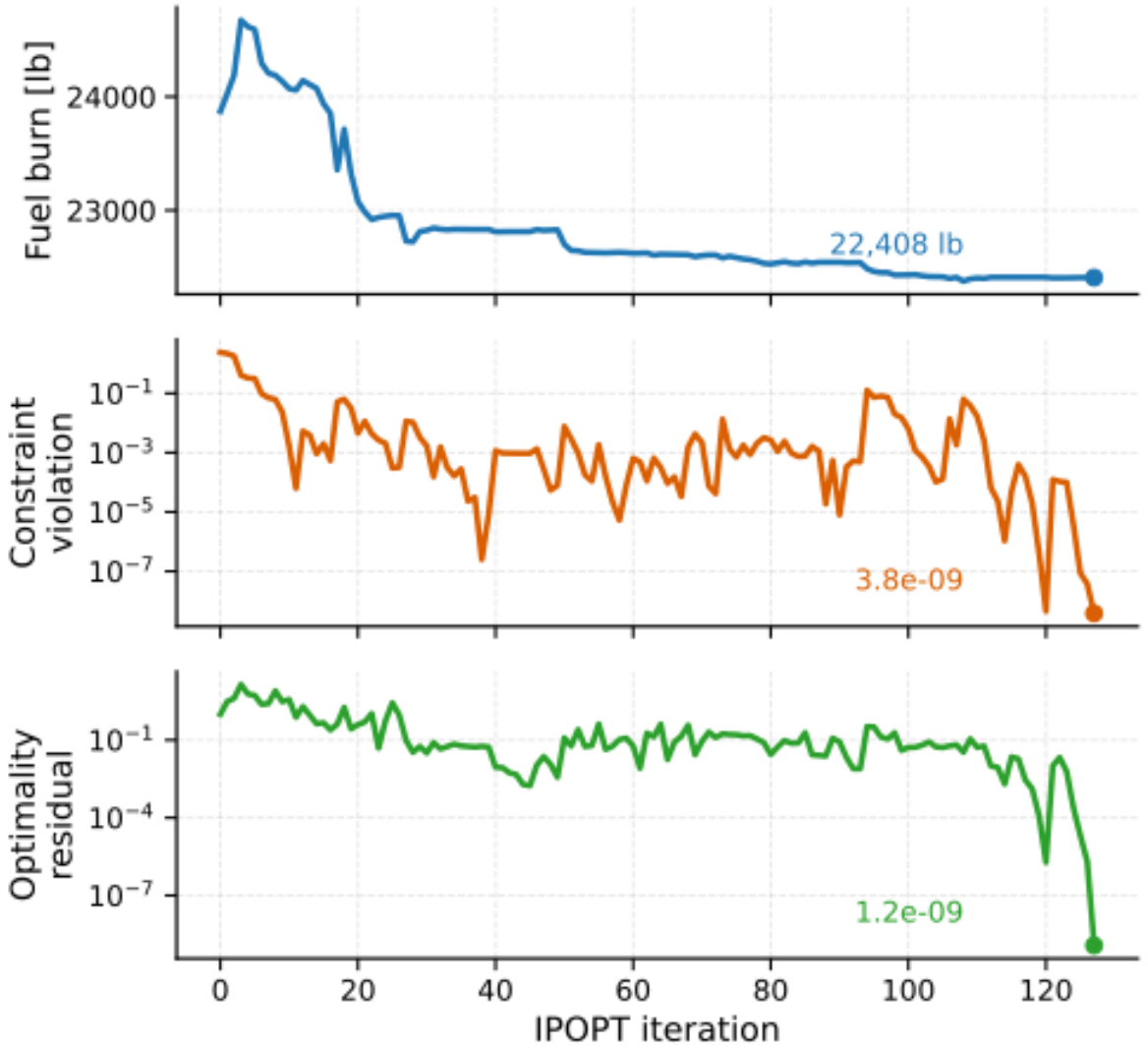
Co-Optimize Airframe–Engine–Trajectory for Min Fuel Burn (2016 EIS assumptions)

- Aircraft: Initialized with CeRAS CSR-01, narrow-body, short-to-medium-range
- Trajectory: SLS, climb, top of climb, cruise segment (2,200 nmi)
- Objective: Minimize fuel burn
  - 35 total design variables | 39 constraints

Engine DVs and bounds	Engine Constraints
5.0 ≤ BPR ≤ 12.5	Fan, LPC, HPC Nc ≤ 105%
1.4 ≤ FPR ≤ 1.7	LPC R-line ≥ 1.15
1.2 ≤ LPC PR ≤ 3.2	HPT exit temp ≤ 1273 K
4.0 ≤ HPC PR ≤ 12.2	Fan diameter ≤ 2.1 m
4000 ≤ TOC thrust ≤ 10000 lbf	SLS thrust ≥ required
0 ≤ W_bleed / W_core ≤ 0.1	Gross thrust ≥ 0

Hands-off pyCycle Convergence Enabled by Robustness Strategies

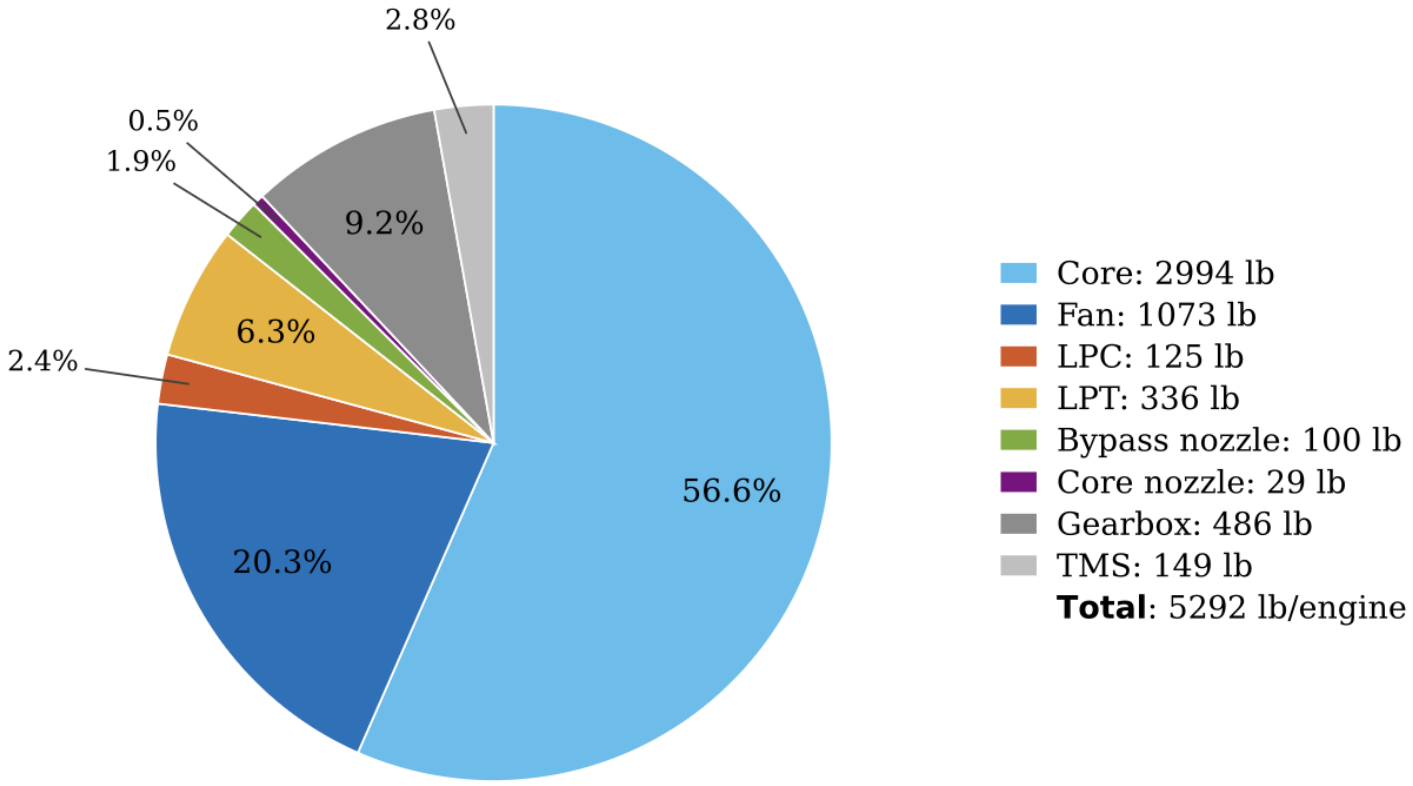
- Zero pyCycle analysis failures during optimization.
- No manual restarts, solver tuning, or intervention required.
- Avoids reliance on manual initialization strategies previously used in the literature (e.g., 22-variable guess matrices).
- Converged in 127 IPOPT major iterations with 221 recorded model evaluations.



Co-optimization Demonstration

Co-Optimize Airframe–Engine–Trajectory for Min Fuel Burn (2016 EIS assumptions)

Optimized Solution
BPR 12.2
OPR 49
Take-off Thrust 22 klbf
MTOW 145 klb
Engine wt 5292 lb



Converged Engine Weight Breakdown

Co-optimization Demonstration

Co-Optimize Airframe–Engine–Trajectory for Min Fuel Burn (2016 EIS assumptions)

Optimized Solution	Real Aircraft/Engine
BPR 12.2	PW1500G: 12.5
OPR 49	PW1500G: 50
Take-off Thrust 22 klbf	PW1500G: 19-25 klbf
MTOW 145 klb	A220-100: 145 klb
Engine wt 5292 lb	PW1500G: 4800 lb





Clean-sheet design optimization of A220-like mission recovers A220/PW1500G-Class Solution

Technology Sensitivities are Nonlinear

- **Re-optimizing after each technology change produces nonlinear fuel-burn responses**
  - T_{t4}: 1650 → 1680 K: −1.71% 1680 → 1720 K: −4.73%
  - T_{metal}: 1450 → 1460 K: −3.06% 1460 → 1470 K: −0.65%
  - $\eta_{poly,HPC}$: 0.89 → 0.90: −0.18%. 0.90 → 0.92: −4.57%

Technology Sensitivities are Nonlinear

- **Re-optimizing after each technology change produces nonlinear fuel-burn responses**
  - T_{t4}: 1650 → 1680 K: −1.71% 1680 → 1720 K: −4.73%
  - T_{metal}: 1450 → 1460 K: −3.06% 1460 → 1470 K: −0.65%
  - $\eta_{poly,HPC}$: 0.89 → 0.90: −0.18%. 0.90 → 0.92: −4.57%

- **Benefits came from coupled design shifts, not isolated engine improvements**
  - OPR ($\sim\eta_{th}$), BPR ($\sim\eta_{prop}$), pressure-ratio split, cooling flow, engine weight, required thrust, and installation drag all changed together.

- **Sensitivity impacts are largest when the assumption changes the feasible design space**
  - Thermal limits, diameter, off-design map efficiency, thrust, stall margin, and corrected-speed constraints.

Conclusions & Enabled Capabilities

- **Enables physics-based technology roadmapping by identifying which propulsion concepts materially improve aircraft performance and warrant further investment.**
- **Provides a unified capability to determine which engine technologies most strongly drive system-level performance through aircraft–propulsion–mission optimization, including:**
  - Propulsion system architectures
  - Number of spools, gearing, inclusion of power turbine
  - Peak-shaving
  - Small cores and compressor architectures
  - Improved materials and turbine temperature limits
  - Integrated generators
  - Intercooling and recuperation

Key Contributions

1. Enables broad on-design propulsion-system optimization
2. Robust pyCycle convergence from full throttle down to 20% part power
3. pyCycle integrated into aircraft-trajectory optimization framework
4. Establishes the gas-turbine modeling fidelity required for reliable aircraft- and mission-level optimization

Methods will be made publicly available in an open-source repository and integrated into pyCycle's codebase to support turbofan, turboshaft/turbogenerator, and ducted-fan architectures.

Learn More: Related AACES Research and Emerging Aviary Capabilities



Electra.aero AACES Concept Aircraft

Scan me!

Thank you for your time! Questions?

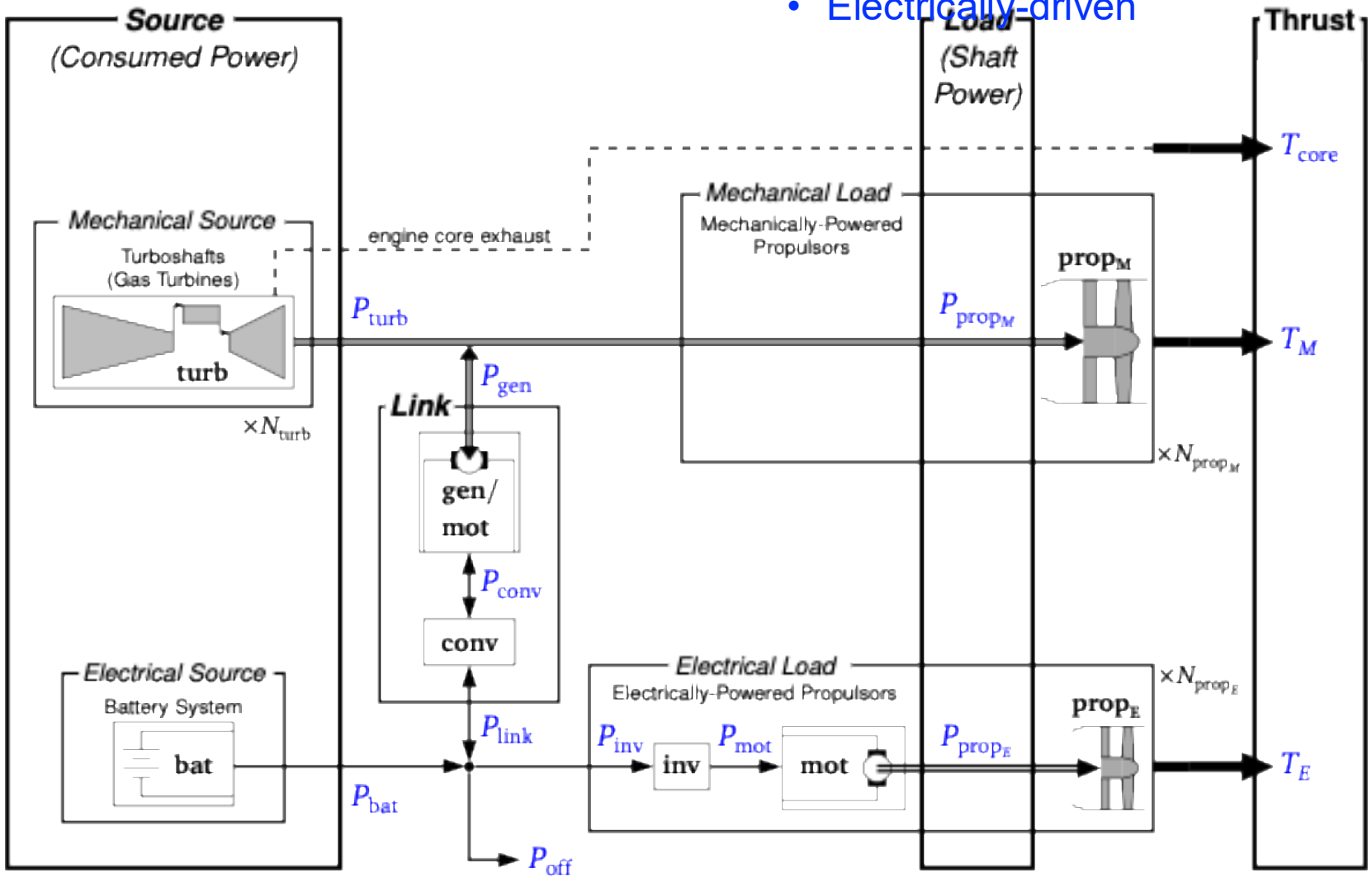
Contact: anushkat@mit.edu

Appendix

Unified Propulsion System

Power loads: propulsors
- Mechanical-driven
- Electrically-driven

Power sources
- Mechanical: gas turbines (turboshafts)
- Electrical: battery system



Useful output:
Thrust

Based on work by Kruger, M., Byahut, S., Uranga, A., *et al.*, AIAA 2018-4227, AIAA Aviation 2018

Unified Propulsion System

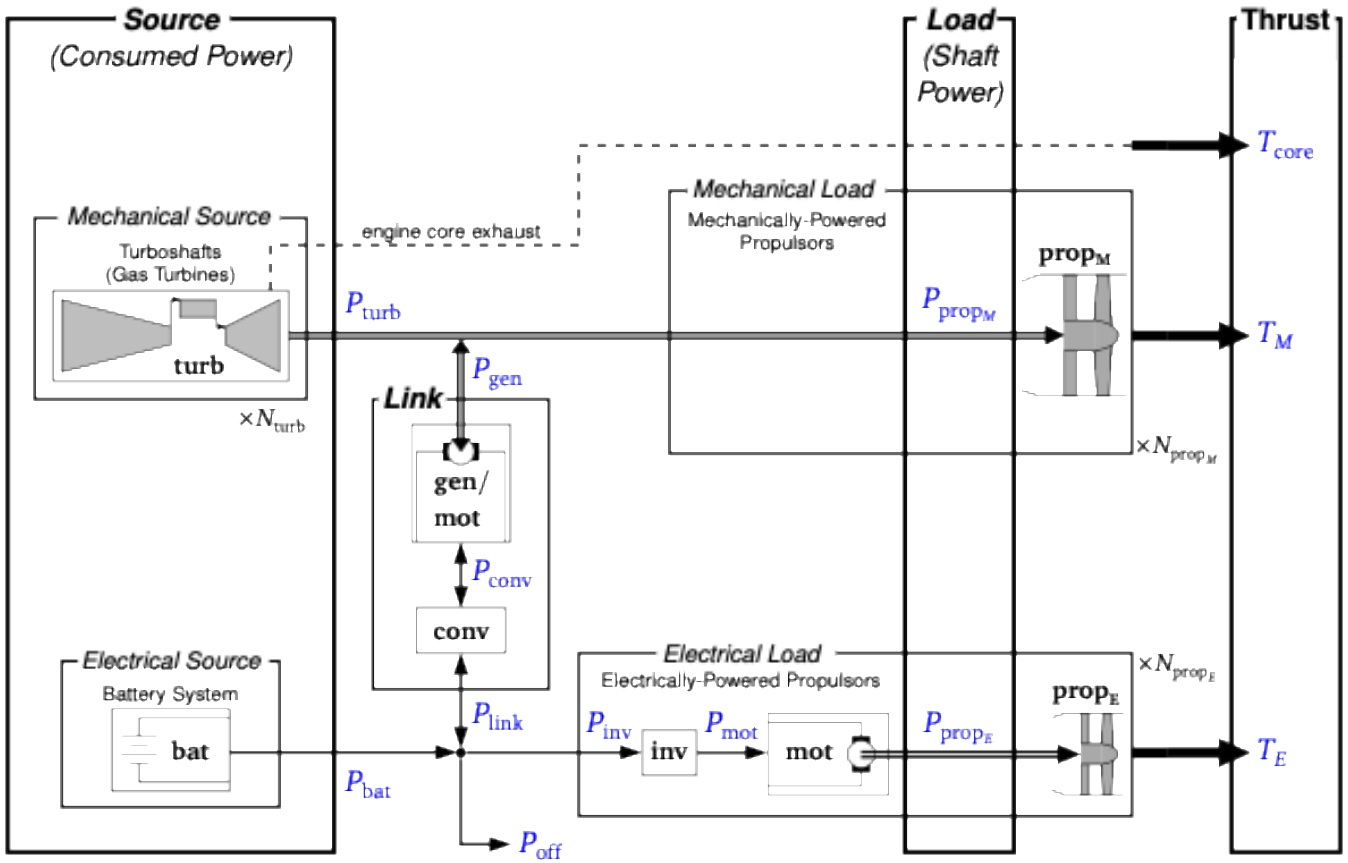
Level of electrification defined by two electrification factors:

Source electrification factor

$$f_S \equiv \frac{P_{\text{bat}}}{P_{\text{bat}} + P_{\text{turb}}}$$

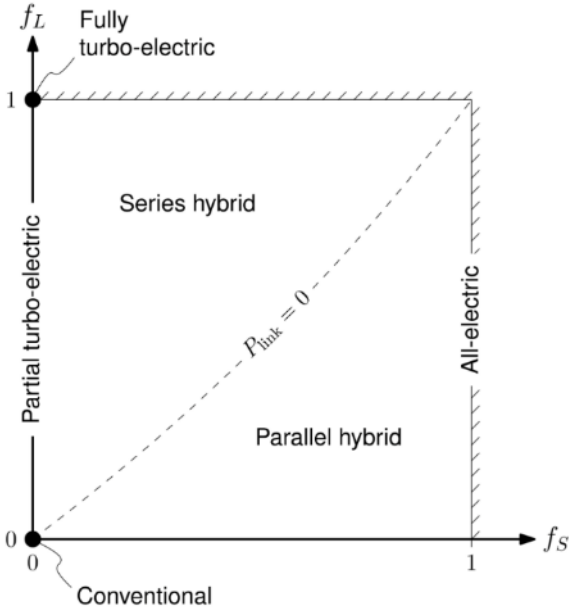
Load electrification factor

$$f_L \equiv \frac{P_{\text{prop, E}}}{P_{\text{prop, E}} + P_{\text{prop, M}}}$$



Additional factor:
Thrust electrification factor

$$f_T \equiv \frac{T_{\text{E}}}{T_{\text{E}} + T_{\text{M}} + T_{\text{core}}}$$



Kruger & Uranga, *J. Aircraft* 61(1), 2024

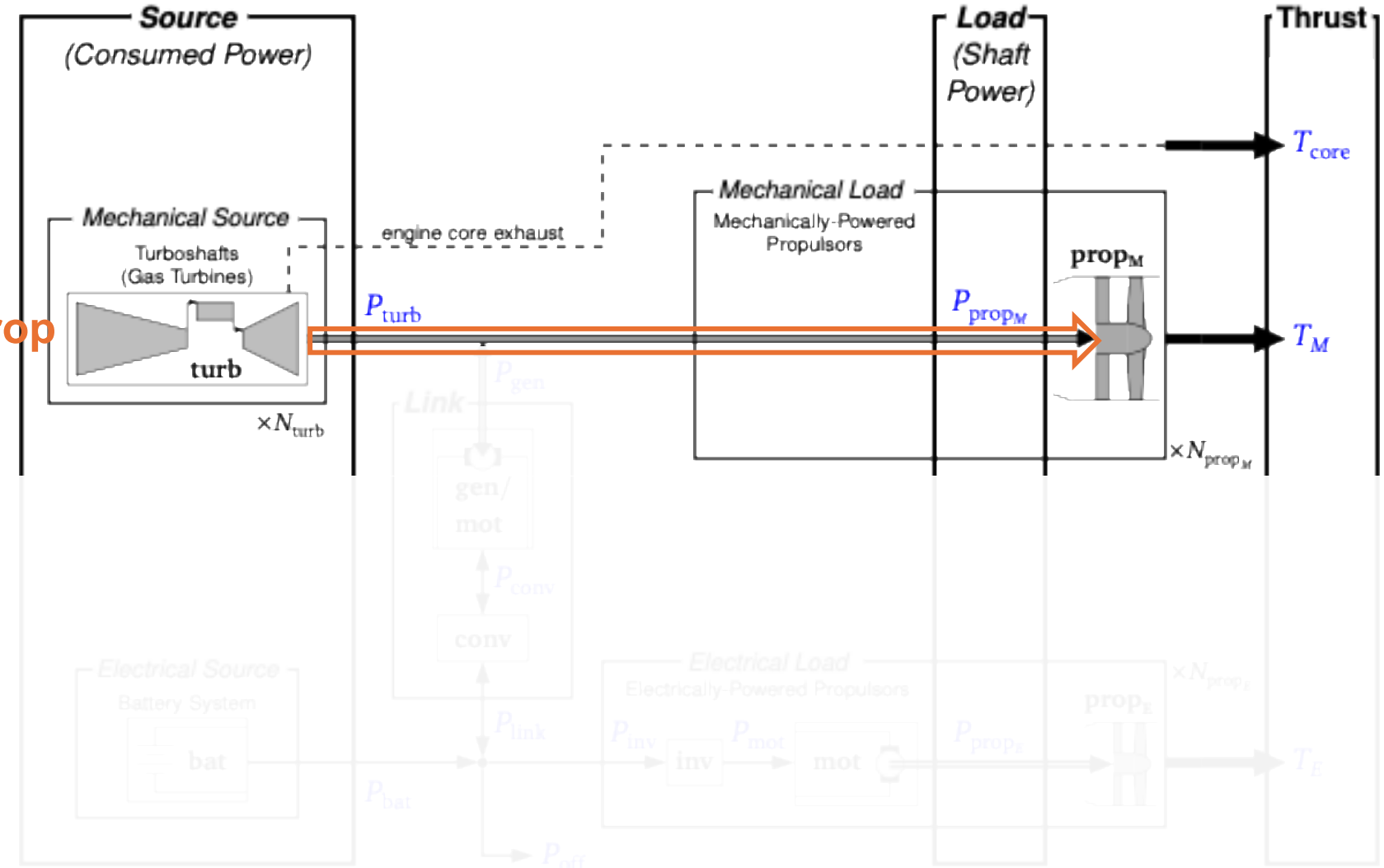
Based on work by Kruger, M., Byahut, S., Uranga, A., *et al.*, AIAA 2018-4227, AIAA Aviation 2018

Unified Propulsion System: Architectures

Conventional propulsion

$$f_s = 0$$
$$f_L = 0$$

Turbofan/Turboprop

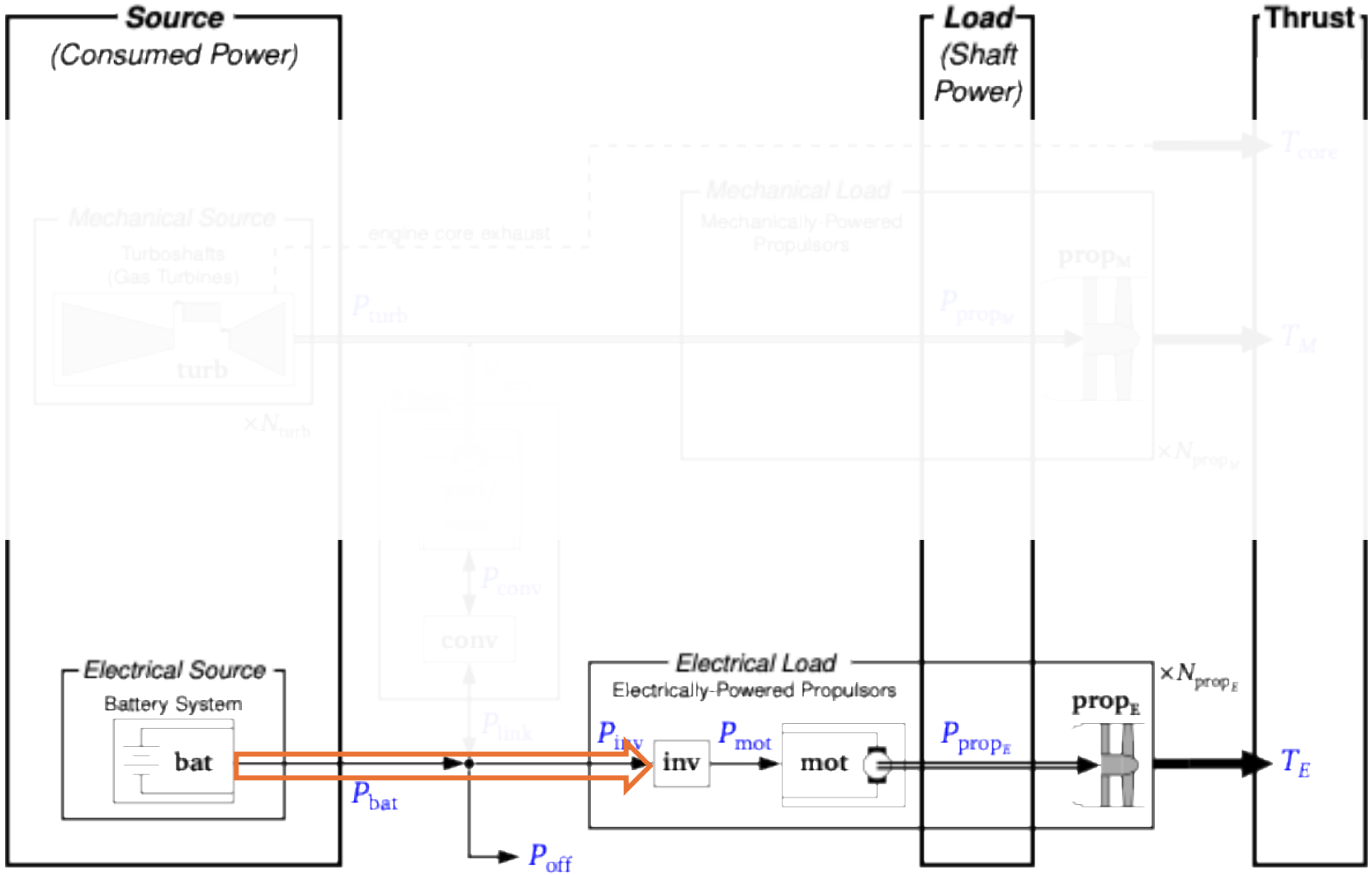


Based on work by Kruger, M., Byahut, S., Uranga, A., *et al.*, AIAA 2018-4227, AIAA Aviation 2018

Unified Propulsion System: Architectures

All-electric

$$f_s = 1$$
$$f_L = 1$$

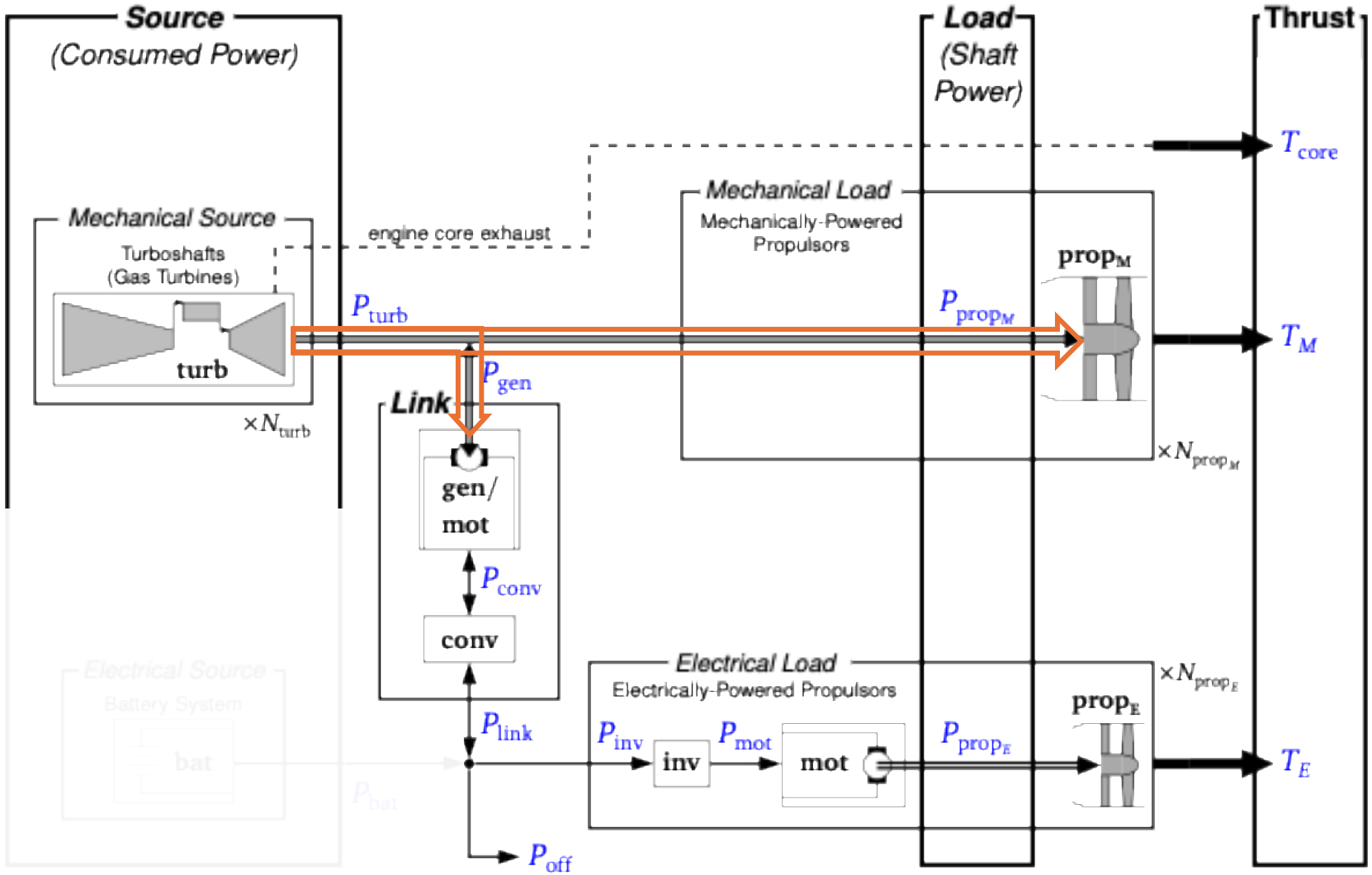


Based on work by Kruger, M., Byahut, S., Uranga, A., *et al.*, AIAA 2018-4227, AIAA Aviation 2018

Unified Propulsion System: Architectures

Partial turbo-electric

$$f_s = 0$$
$$0 < f_L < 1$$



Based on work by Kruger, M., Byahut, S., Uranga, A., *et al.*, AIAA 2018-4227, AIAA Aviation 2018

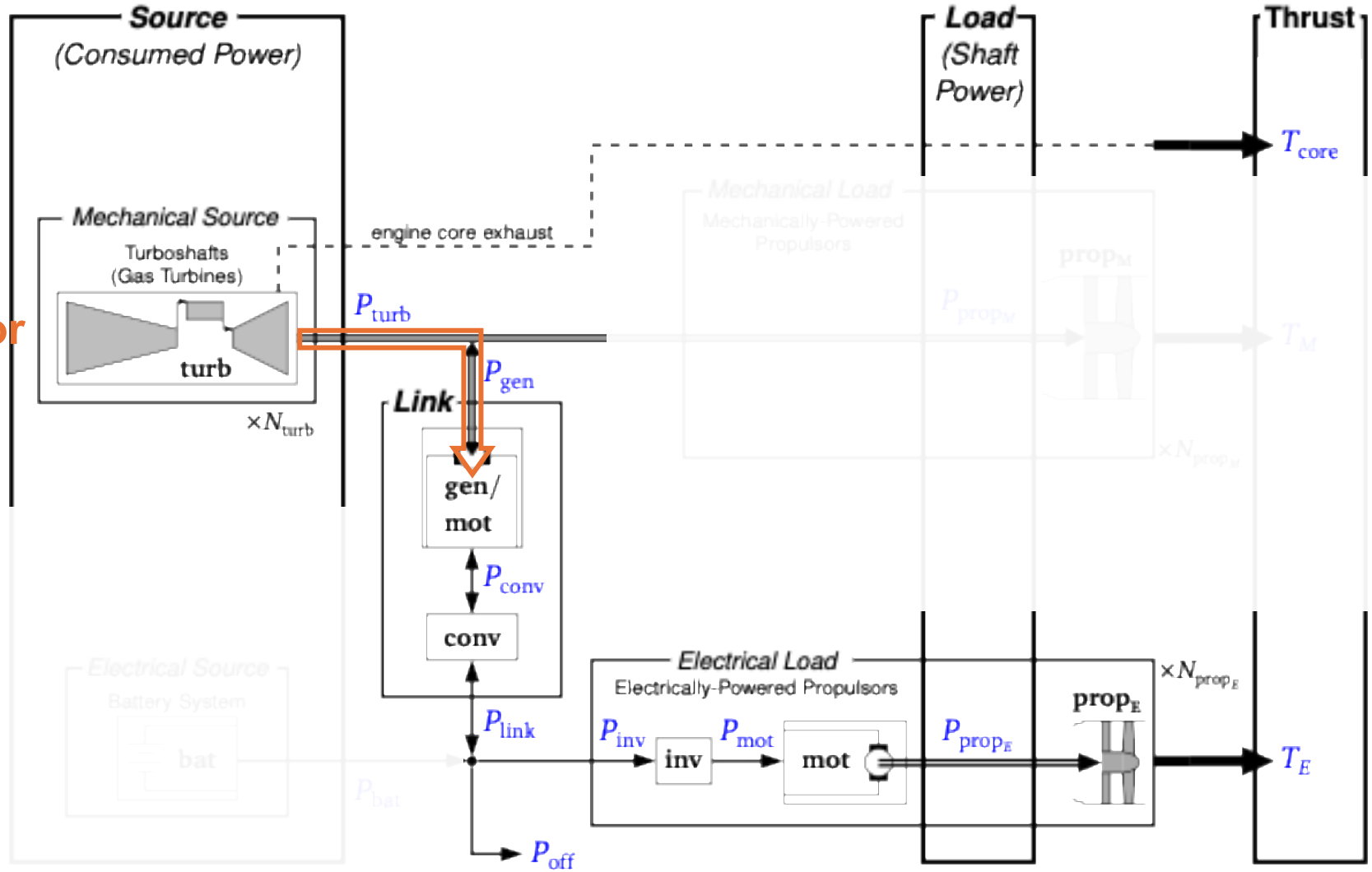
Unified Propulsion System: Architectures

Fully turbo-electric

$f_s = 0$

$f_L = 1$

Turbo-generator

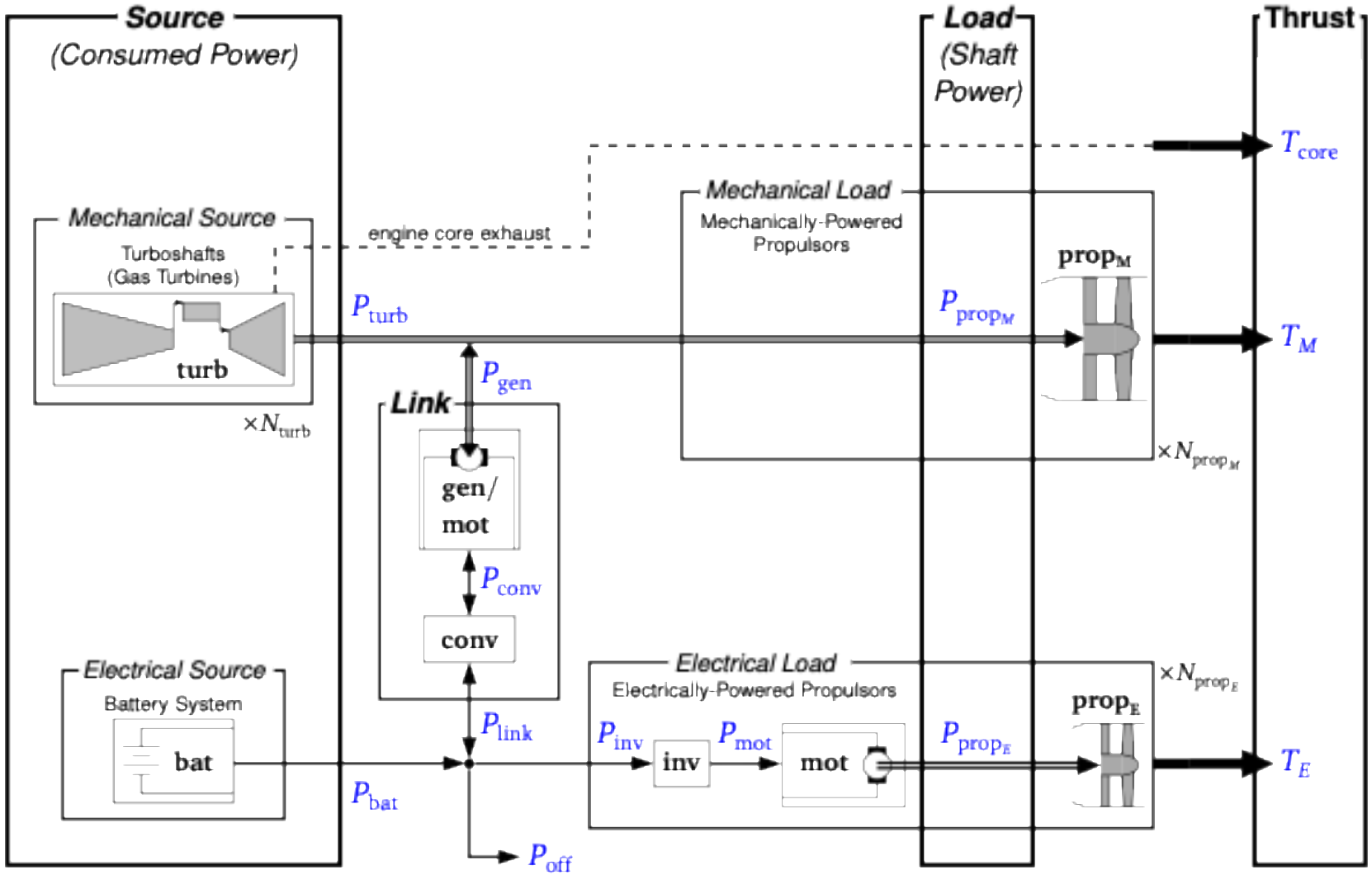


Based on work by Kruger, M., Byahut, S., Uranga, A., *et al.*, AIAA 2018-4227, AIAA Aviation 2018

Unified Propulsion System: Architectures

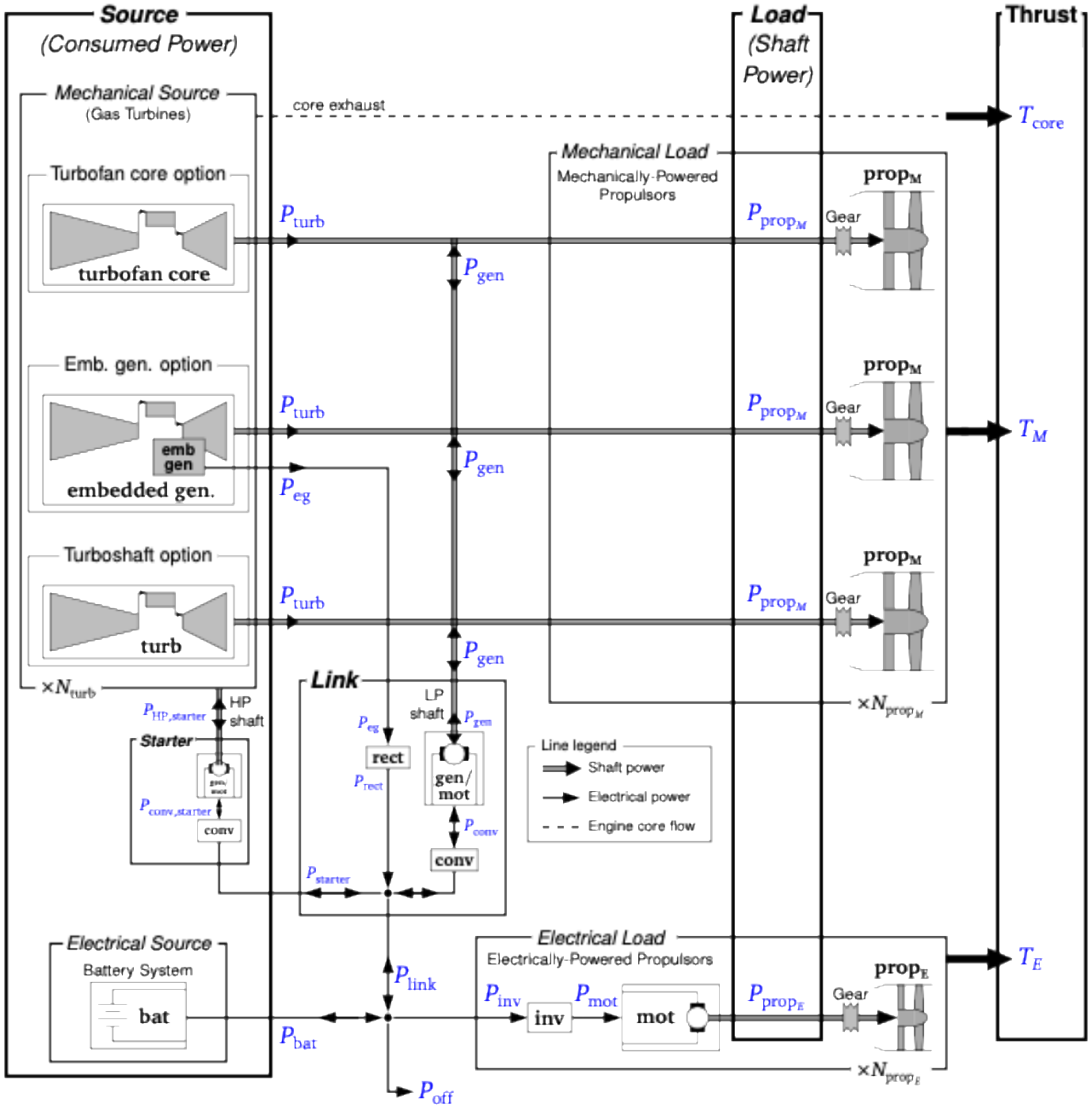
Hybrid

$$0 < f_s < 1$$
$$0 \leq f_L \leq 1$$

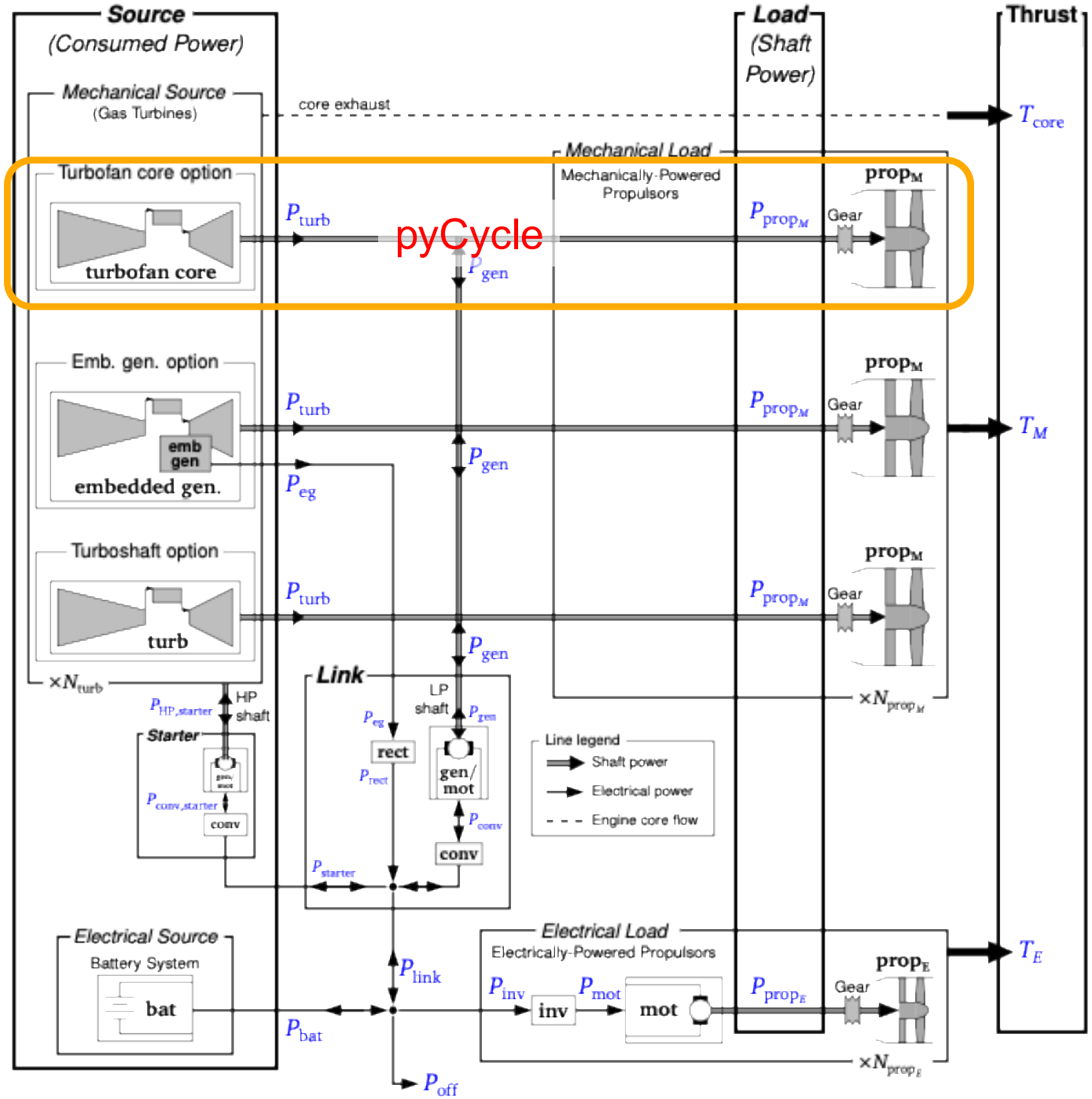


Based on work by Kruger, M., Byahut, S., Uranga, A., *et al.*, AIAA 2018-4227, AIAA Aviation 2018

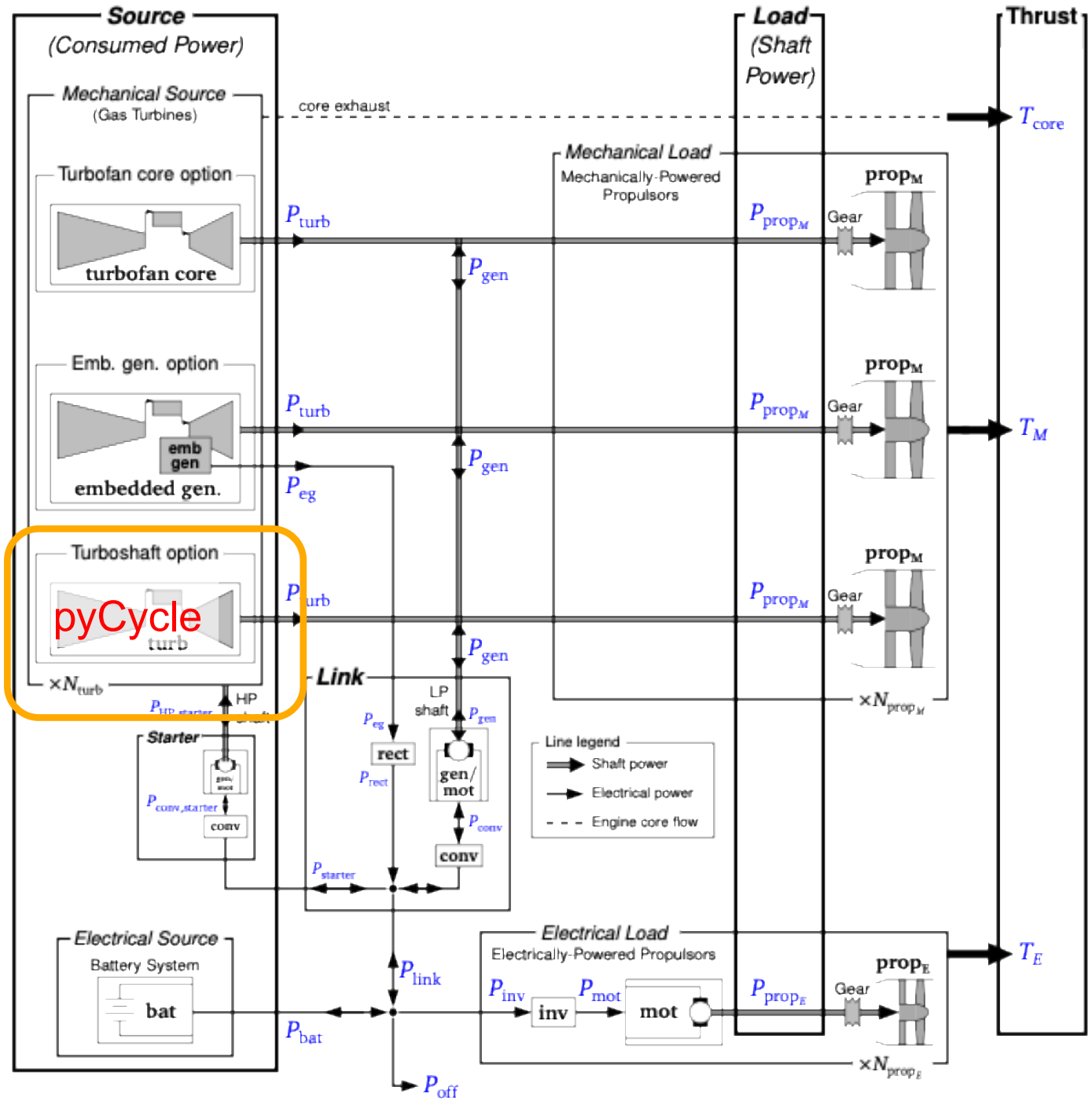
Unified Propulsion System



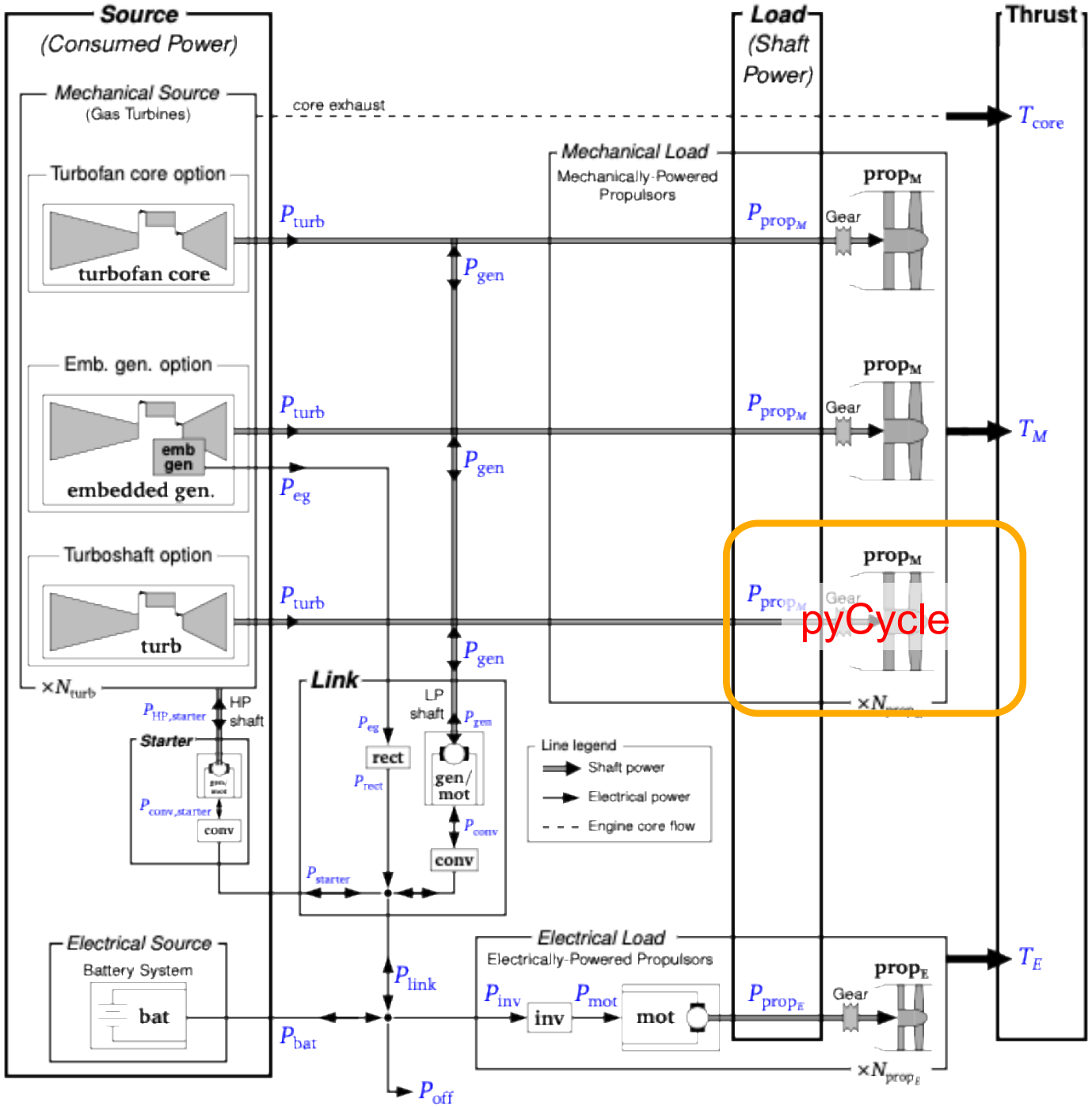
Unified Propulsion System



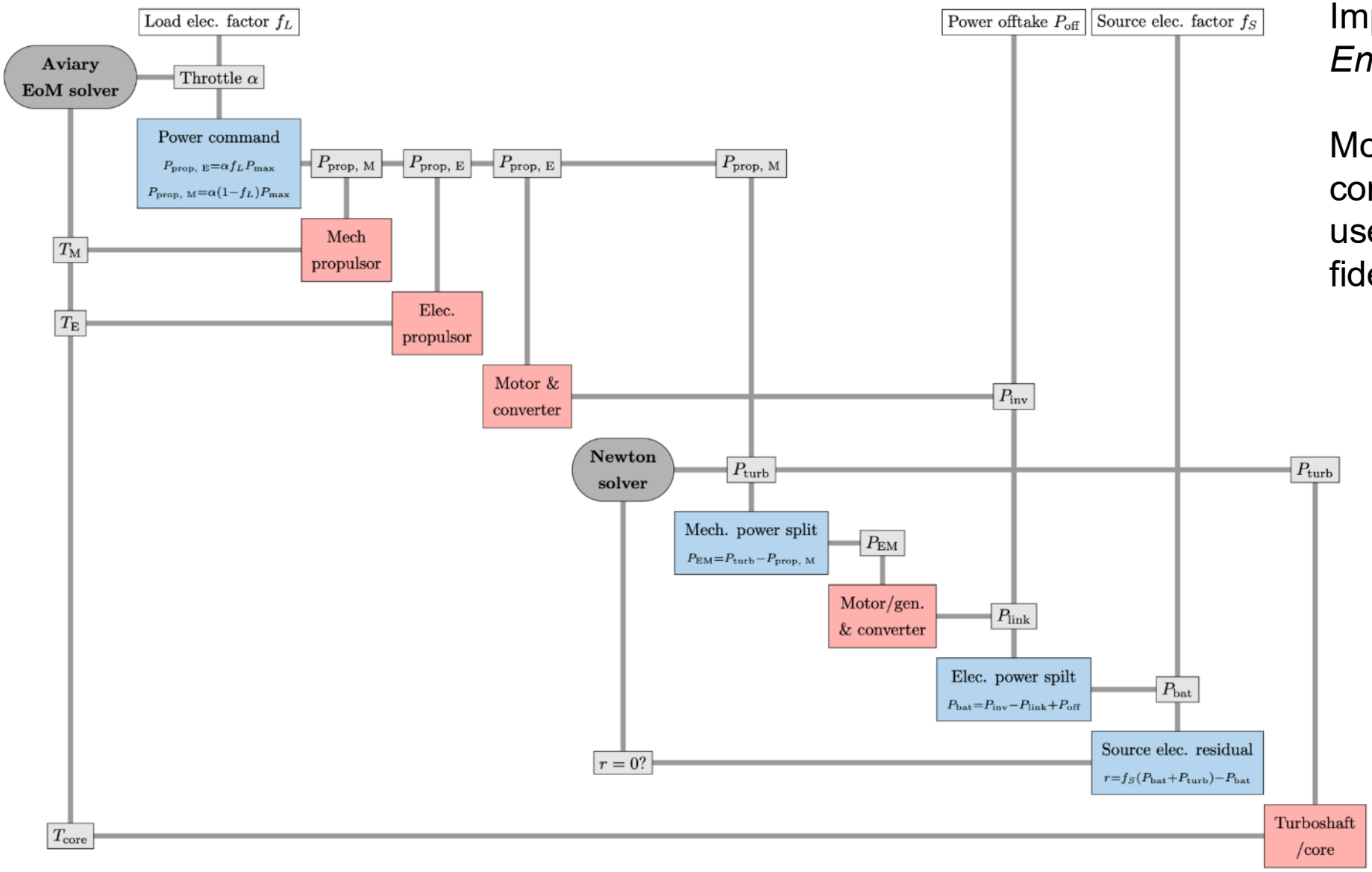
Unified Propulsion System



Unified Propulsion System



Implementation of Unified Propulsion System Model



Implemented using Aviary's *Engine Builder* user interface

Modular implementation with common power-based I/O: users can easily switch model fidelity for propulsion components