Comparing Hydrogen and Jet-A for an N+3 Turbofan with Water Recirculation using Gradient-Based Optimization

Peter N. Atma*, Andrew hbt!.R. Lamkin†, and Joaquim R. R. A. Martins‡
University of Michigan, Ann Arbor, MI, 48109

Advances in commercial propulsion technology led to the development of efficient high bypass ratio turbofan engines with larger overall pressure ratios and internal temperatures. Current trends suggest that geared ultra high bypass ratio turbofans are the next generation of commercial propulsion systems. Furthermore, the emphasis on decreasing emissions has driven the exploration of hydrogen-powered aircraft, adding to the already challenging design space. Carrying and burning hydrogen introduces complexity and weight penalties that we must offset using the fuel's thermodynamic and chemical properties. In this study, we create a closed-loop water recirculation system with a zero-dimensional thermodynamic model and compare the benefits between jet-A and hydrogen fuels. We perform a gradient-based optimization parameter sweep to explore the trade-offs between performance and emissions using both fuels with water recirculation. The results quantify the design space for next-generation propulsion concepts that can take advantage of hydrogen fuel's advantageous thermodynamic properties to reduce emissions and improve performance.

I. Introduction

The effects of climate change are pushing the aviation industry towards hydrogen-fueled propulsion systems as a solution to reduce emissions. N+3 technology estimates for turbofan engines that burn hydrocarbon fuels suggest that higher efficiency can be achieved by designing ultra high bypass ratio (UHBR) engines with small cores and high overall pressure ratios (OPR). Higher OPR and smaller cores challenge the limits of compressor and turbine design, placing an upper bound on potential performance and emissions improvements. Switching to hydrogen as the primary fuel source reduces carbon dioxide emissions immediately, but adds complexity and weight that offset the benefits. However, hydrogen is a versatile fuel with advantageous chemical and thermodynamic properties that can be exploited to increase the performance and reduce emissions. We introduce a closed-loop water recirculation model that demonstrates the possible efficiency gain when hydrogen is used for purposes other than combustion.

Water recirculation is the process of extracting water from the exhaust stream of a propulsion system and injecting it upstream of the combustor as finely atomized droplets. NASA, Boeing, and Rolls-Royce studied this concept and suggested that this technique reduces the NOx emissions as much as 47 percent [**?**]. Additionally, water recirculation improves fuel efficiency and thrust output with lower combustion temperatures that can improve the lifetime of turbine blades and reduce noise [**?**]. Traditional propulsion systems that burn hydrocarbon fuels would require external water storage on the aircraft because they do not produce enough in the exhaust for recirculation. The added weight of tanks, pumping, and ducting makes this concept infeasible for a conventional aircraft. The main product of hydrogen combustion is water vapor and can thus be recovered from the exhaust stream. Recirculating water vapor from the exhaust of hydrogen combustion reduces the requirement for storage tanks and allows for the creation of a closed loop system inside the propulsion cycle.

Zero-dimensional cycle modeling is an efficient tool for predicting the initial design, performance, and emissions of new propulsion concepts. Zero-dimensional analysis uses a first-principles approach with a chemical equilibrium analysis (CEA) thermodynamics solver [**?**] that considers the molecular species of different fuels. The industry standard for thermodynamic cycle analysis is the Numerical Propulsion System Simulation (NPSS) framework [**?**]. NPSS is a modular object-oriented library that models engine components as individual blocks with several thermodynamic solvers. **?**] created a new tool called pyCycle with the same functionality as NPSS with analytical derivatives for each engine component and thermodynamic solver [**?**]. pyCycle is built on top of the OpenMDAO framework [**?**] to enable gradient-based optimization and leverage hierarchical nonlinear solver structures for robustness.

[*] MSE Student, Department of Aerospace Engineering, AIAA Student Member
[†] Ph.D. Candidate, Department of Aerospace Engineering, AIAA Student Member
[‡] Professor, Department of Aerospace Engineering, AIAA Fellow

In this work, we analyze the thermodynamic benefits of a closed-loop water vapor recovery and water injection system in a high-bypass turbofan engine. We develop pyCycle components for water injection and vapor recovery to quantify the benefit of a closed loop recirculation system. We use gradient-based optimization to minimize fuel burn subject to performance requirements using both jet-A and hydrogen at a range of flight conditions. The optimized results show the trade-off between complexity, performance, and efficiency for jet-A and hydrogen fuels.

This work is organized as follows. First, in Section **??**, we introduce the turbofan model and explain the water injection and water recovery components. In section **??** the implementation of the multipoint optimization problem is discussed. Finally, we present the optimized results and discuss the design space in section **??**.

II. Methodology

A. Engine Model Overview

The UHB turbofan model is the NASA advanced technology "N+3" engine [**?**]. The N+3 reference cycle represents a UHB ratio geared turbofan that could be available in the 2030 to 2040 time frame. The flow path consists of an inlet that directs ambient air through a fan, followed by a duct that splits the flow into a core and a bypass stream, each ending in a core and bypass nozzle, respectively. The low pressure system is split into two mechanical subsystems. First, the fan is connected to the gearbox that reduces the shaft speed to decrease the fan tip speeds. Second, the gearbox attaches to the low-pressure shaft that connects to the low-pressure compressor (LPC) and low-pressure turbine (LPT). The high pressure compressor (HPC) is connected to the high pressure turbine (HPT) by the high pressure shaft.

We introduce the closed-loop water recovery system as a feedback loop that transports water from the exhaust to upstream of the compressors. The recovery system injects vaporized water into the core stream that reduces the combustion temperature due to heat absorption. The vapor recovery component is placed directly before the core nozzle to extract water from the exhaust and recycle it back to the injector.

In this section we explain the implementation and assumptions of the water recovery model. We present the full engine layout and provide details on the multipoint zero-dimensional modeling approach. The component flow interface and mechanical connections, including the water injector and water extractor, are depicted in Figure **??**.

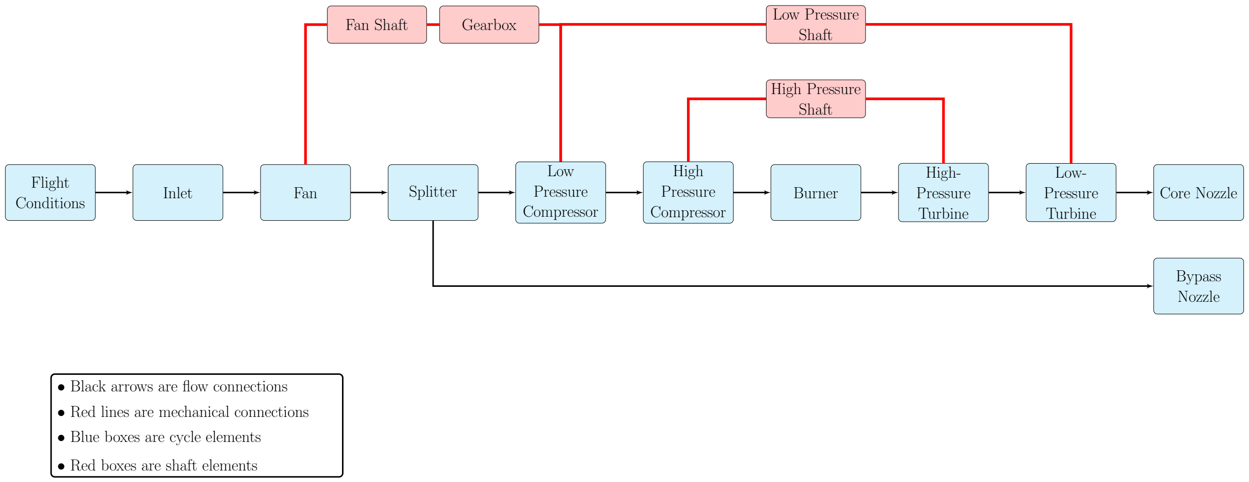

Fig. 1 N+3 engine cycle layout that shows how each of the pyCycle components are connected and how performance metrics are computed.

B. Water Recovery Loop Model

We implemented the closed-loop water recovery system as a feedback system that extracts water from the exhaust stream and injects it upstream of the HPC. We chose this injection location based on claims from a study by NASA, Boeing, and Rolls-Royce [**?**] that water injection directly into the combustor is unnecessary. The water vapor recovery component sits downstream of the LPT and extracts water from the flow before it exits the core nozzle. The component flow interface and mechanical connections, including the water injector and water extractor, are depicted in Figure **??**.

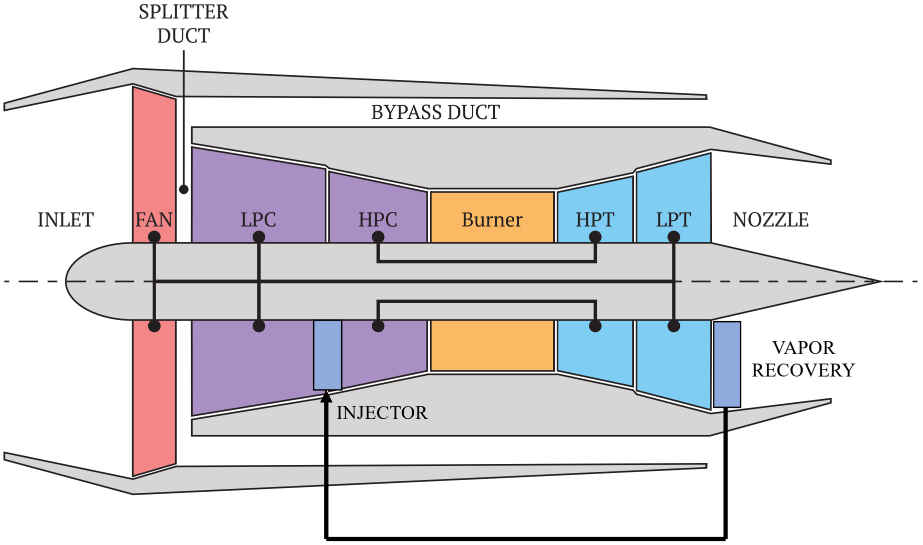

Fig. 2 The configuration of a high-bypass turbofan model with an integrated closed-loop water vapor recovery and injection system. The water vapor recovery system (extractor) extracts a fraction of the water in the core stream and reinjects it upstream of the high-pressure compressor. This diagram illustrates the feedback effect that this implementation has on the overall core flow.

Since water is being used in the sections of the engine upstream of the combustor, this model was designed to take the humidity of the air into account. To utilize water engine section infront of the combustor the N+3 engine model had to be modified to include H_2O as a species. This can be accomplished by using the wet_air dataset in pyCycle. The wet_air dataset simply uses the $Janaf$ dataset from the original model but removes a couple troublesome hydrocarbons from the product species as they can cause numerical issues. The FlightConditions() pyCycle subsystem can then have a reactant and mixture ratio specified when implemented. In this case, the reactant is be H_2O and the mixture ratio is the water-to-air (WAR) ratio of the water content in the air. The WAR value is simply the specific humidity (g water/kg total) of the air expressed in equal units and separating out the total mass (kg water/kg air). The humidity values used in this paper are from a NOAA dataset [**?**].

To model the complicated water recirculation loop in pyCycle new components had to be developed. The two new components that were developed are a water injector to add water to the flow and a water extractor to divert a fraction of the water in the flow away from the exhaust stream. The injector component is based on the combustor component already included in pyCycle which operates by injecting a given fuel-to-air ratio (FAR) or fuel flow rate to determine the mass of fuel added to the incoming flow. This new mixture combination is then used to compute the various chemical species present in the flow at the given thermodynamic state which is determined by the incoming flow. The new species composition and thermodynamic variables are then determined using a Gibbs Free Energy minimization approach called Chemical Equilibrium with Applications (CEA). Once the new composition is determined, the various thermodynamic variables are passed to the next component in the engine. The water injector component would inject water as a reactant instead of the fuel as the combustor would. The water injector can take either water-to-air ratio (WAR) or water mass flow rate to determine how much water is added to the flow. A simple schematic of the injector is shown in Figure **??** where Y_{H_2O} is the mole fraction of water molecules.

The water extractor component is based on an air-bleed component in pyCycle with the added complexity of extracting a fraction of a specific species within the flow. This extractor component works by first determining the water content of the incoming flow and extracting a given fraction of that water flow. The water content of the incoming flow is determined by solving a CEA analysis of the incoming flow. This water content in the incoming flow is then used with a given fraction of the water to extract to determine the how much water mass flow to remove from the stream. At the same time, the atomic mixture of the core stream is updated to represent the mixture that would be left if a number of hydrogen and oxygen atoms corresponding to the amount of water extracted were removed. The outputs of the extractor component are the flow rate of the core flow and the corresponding flow properties in addition to the

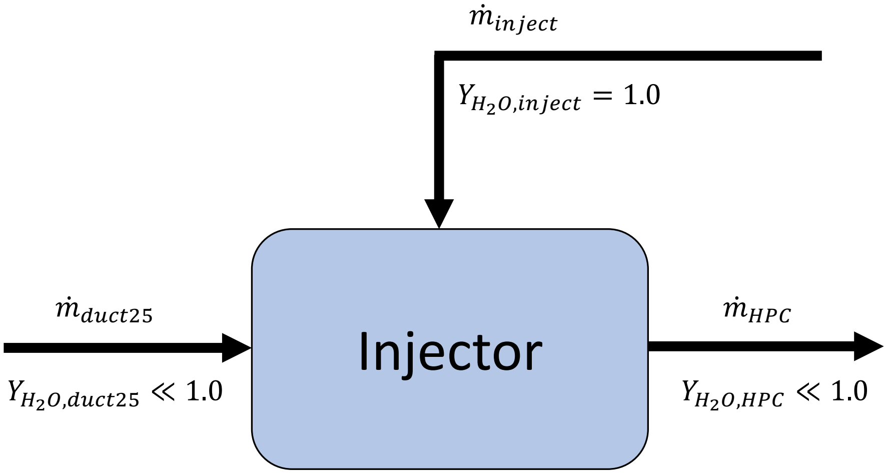

Fig. 3 Injector component schematic. Water from the extractor is simply injected into the core flow upstream of the high-pressure compressor.

extracted water mass flow rate. While this process certainly would have pressure loss penalties in terms of condensing water in a condenser, this project just looks at the effects of water recovery since there are no current models for nozzle exhaust condensers available. A simple schematic of the extractor is shown in Figure **??** where Y_{H_2O} is the mole fraction of water molecules and $X_{H_2O,k}$ is the fraction of water that is recovered from the core stream of operating condition, k.

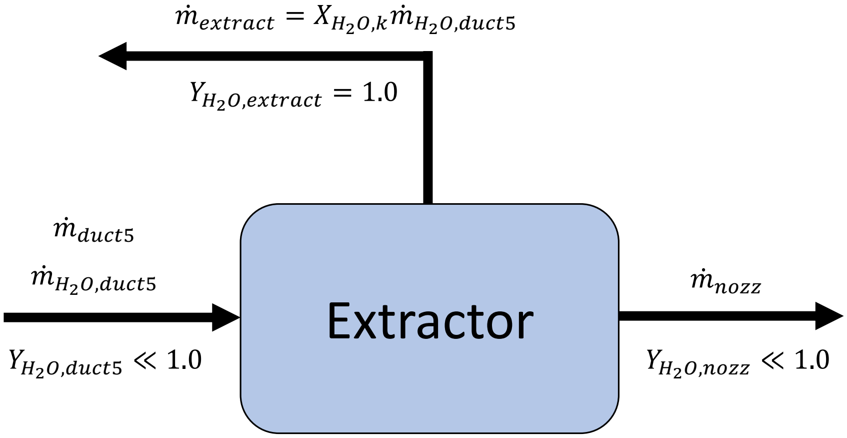

Fig. 4 Extractor component schematic. A certain mole fraction of the flow coming out of duct5 is comprised of water of which a certain specified fraction is extracted. The extracted water is routed back upstream to the injector and the rest is simply exhausted out the nozzle.

Unlike other flow streams in the N+3 model, the vapor recovery loop has a feedback effect since it is adding a mass flow rate at the injector and extracting a mass flow rate at the extractor. This is because the pyCycle model is solved sequentially component to component during each Newton iteration. So a Nonlinear Block Gauss-Seidel solver would be sufficient to solve this feedback loop. However, the N+3 model already has a nonlinear Newton solver at the Multipoint Design level, so the extractor mass flow rate and injector mass flow rate variables are connected for each design point and use this nonlinear solver to converge the water streams. These connections can be seen in Figure **??**.

In pyCycle, the Cycle block contains all of the governing thermodynamic equations needed to model an engine such as CEA analysis, compressor and turbine models, and equations of state. These equations allow any arbitrary engine design to be assembled using the modular cycle elements. The Cycle block itself may not result in a valid model of the

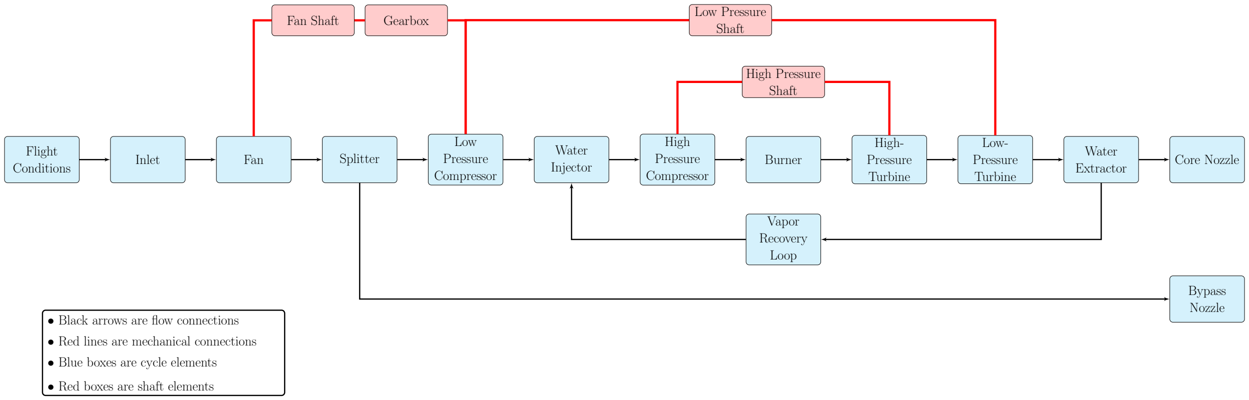

Fig. 5 N+3 engine cycle layout that shows how each of the pyCycle components are connected with the closed-loop vapor recovery system.

engine since some physical dependencies may not match. These physical dependencies are handled by the Balance block which solves a set of implicit state variables and nonlinear physical residual equations in the form:

$$\mathcal{R}_c = \sum_{i=1}^{n} x_i = 0 \tag{1}$$

where x_i is some physical variable such as a mass or energy flow rate. These residuals represent the physical conservation laws such as mass flow rate are converged to zero by the Solver. In pyCycle, a direct solver is used for linear equations and a Newton solver is used for nonlinear equations which utilizes the analytic partial derivatives provided by each OpenMDAO component. All residuals within the engine model are converged at the Cycle level for a given operation condition since all of the parameters downstream do not directly effect those upstream. However, with introducing the extractor and injector components, we add a feedback loop into the model where mass is removed downstream and added upstream. Therefore, the mass flow rates residuals of the closed-loop vapor recovery system shown below would need to be converged in the Multipoint Cycle explained in the next section.

$$\mathcal{R}_{c,\dot{m}} = \dot{m}_{extractor} - \dot{m}_{injector} = 0 \tag{2}$$

C. Multipoint Model

The N+3 engine model as it is implemented in pyCycle has many complex thermodynamic and performance connections between different operating conditions. Therefore, the N+3 engine model computes the engine performance at 4 different operating conditions in order to resolve the complexities between each operating condition on the final design. The operating conditions that considered are top-of-climb (TOC), rotating takeoff (RTO), sea-level static (SLS), and cruise (CRZ). The SLS condition represents the static performance requirements for the engine, the RTO condition represents the performance and cooling requirements at the point of rotation during takeoff, the TOC operating condition represents the performance requirements during a climb, and the CRZ operation condition represents the criuse performance of the engine. To account for each of these operating conditions, the N+3 model uses a technique called Multipoint Design Point (MDP) modeling to converge the model to an engine design that satisfies the requirements at each of these operating conditions. One of these operating conditions is specified as the design point, which in this case is TOC, and the other operating conditions are considered the off-design points. The design point for the model generally sets all of the engine sizing such as areas and pressure ratios across components. The off-design points are essentially the size of the engine at the design point but operating at the corresponding required thrust, cooling, and environmental conditions. The design point of the engine model is converged using a linear Newton solver which then passes the geometric sizing variables such as cross-sectional areas and map scalars to the off-design points and similarly converges those. Then, a nonlinear Newton solver is used to converge the overall model with respect to the variable connections between each operating condition. An XDSM diagram of the nominal N+3 engine model is shown in Figure **??** [**?**].

In the optimization problem described later, we set the mass flow rate of fuel, $\dot{m}_{fuel}$, as the objective function. Our comparison metric is $TSEC_{CRZ}$ but minimizing $\dot{m}_{fuel}$ is a better posed optimization problem since we will also have

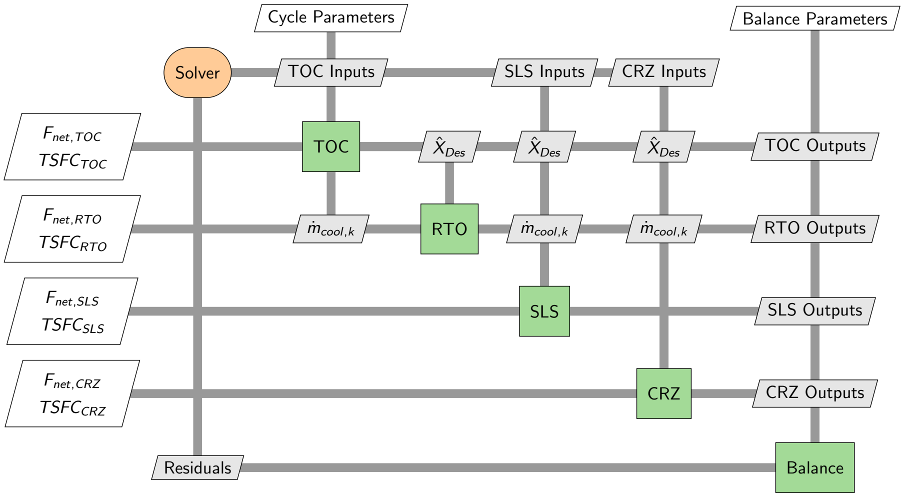

Fig. 6 N+3 engine multipoint setup XDSM diagram. This XDSM diagram shows how each of the operating conditions is coupled with each other and how the outputs are computed.

thrust constraints. Therefore, the water mass fraction that will most impact the objective function the most is at the CRZ operating condition and we thus want to size the area of the injector and extractor at CRZ instead of TOC like the rest of the areas are. Normally, the cross-sectional areas are set at the design point (TOC) and are connected to the off-design points. To size the areas of the injector and extractor at CRZ, we only connect the state variables across these components at the corresponding operating condition and set the areas of both components at the CRZ point. These connections are shown graphically in the XDSM diagram in Figure **??**.

In the N+3 model implementation presented in the pyCycle paper, the bypass ratio (BPR) at TOC is computed using a Balance component. The Balance component for BPR_{TOC} uses the exit velocity ratio of the core and bypass nozzles which is given as:

$$V_{ratio} = \frac{V_{core,ideal} C_{v,core}}{V_{bypass,ideal} C_{v,bypass}} \tag{3}$$

where V_{ideal} is the exit velocity of a nozzle and C_v is the velocity coefficient of the nozzle which represents non-ideal effects. This adds a constraint where the core exit velocity is higher than the bypass exit velocity.

D. Performance Metrics

Past implementations of engine cycles and example engine cycles in pyCycle generally have used either JetA or JP-7 as a fuel. However, in this work we are interested in comparing the performance of similar engines with different types of fuel, specifically hydrogen due to the many benefits it offers in terms of performance and thermal properties. Fortunately, pyCycle comes ready to use hydrogen with cataloged exhaust products for hydrogen fuel. The fuel type is simply changed from Jet-A to H2 in the pyCycle model.

For measuring the efficiency of jet engines, thrust-specific fuel consumption (TSFC) is generally used since it is a represents how low the fuel burn is for a given level of thrust. However, when comparing Jet-A and hydrogen fuels this is not such a good metric since a given mass flow rate of hydrogen has an energy content almost 3 times that of Jet-A. Therefore, a new metric for comparing the relative efficiencies of engine running on Jet-A versus hydrogen is thrust-specific energy consumption (TSEC) which multiplies TSFC by the lower heating value (LHV) of the fuel

$$TSFC = \frac{\dot{m}_{fuel}}{F_{thrust}} \tag{4}$$

$$TSEC = \frac{\dot{m}_{fuel} LHV}{F_{thrust}} = TSFC \times LHV \tag{5}$$

With all of the sub-models of the N3 engine presented, the complete engine cycle XDSM diagram is shown in Figure **??**.

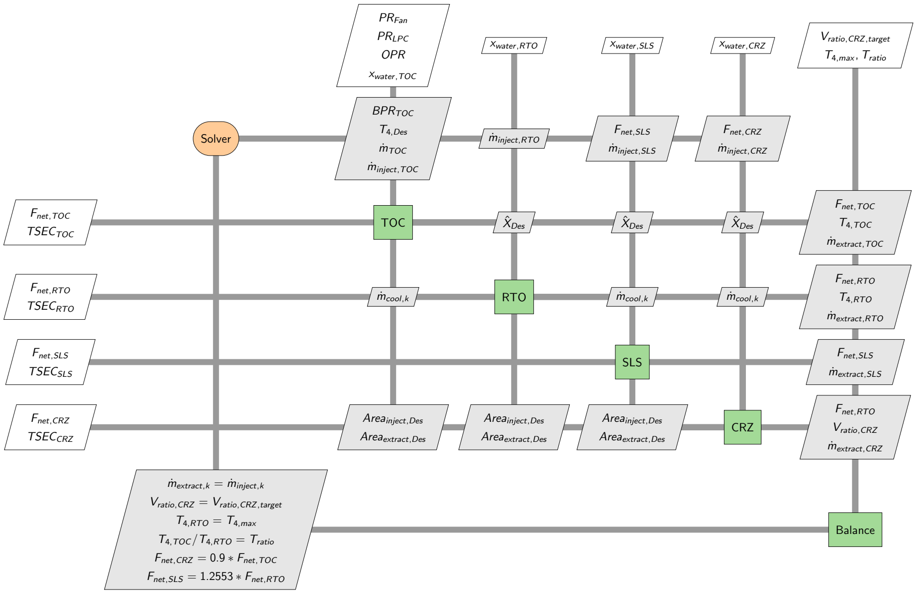

Fig. 7 Full model N+3 XDSM diagram. This XDSM diagram shows the multipoint coupling between the different operation conditions and shows how the water recovery fractions are used to solve for water mass flow rates.

E. Model Parametric Studies

To understand the coupling between the various operation conditions and design point, a few simple parametric studies of the water recovery fractions were run. The design parameters for the engine are shown in the table below. Additionally, the humidity conditions listed below are used in the flight conditions component in the N+3 engine to ensure that the humidity of the air entering the engine is equal to the humidity of the atmosphere at a given operating condition.

III. Optimization Problem

A. Problem Statment

We will perform gradient-based design optimization of the engine model with a multi-point architecture that considers several flight conditions experienced by commercial aircraft. An XDSM diagram of the optimization problem with the multipoint formulation is shown in Figure **??**. The objective is to minimize the TSEC at the cruise condition subject to constraints, thrust, max temperature, and engine diameter. We will perform a parameter sweep of the limiting constraint to understand the impact of different emissions and performance requirements on the design space. The optimization problem of the N3 engine and the vapor recovery loop are shown in Figure **??**.

This optimization problem was run for both Jet-A and H2 fuels using the parameters given in Table **??**.

B. Optimization Software

OpenMDAO supports the use of different optimization packages including Scipy and PyOptSparse. PyOptSparse is an object-oriented framework for formulating and solving nonlinear constrained optimization problems. PyOptSparse was chosen as the main optimization package for this problem due to its wide range of open-source gradient-based optimizers and integration with OpenMDAO.

SNOPT (Sparse Nonlinear OPTimizer) was the optimizer chosen for this work. SNOPT is a state-of-the-art optimizer

Table 1 N+3 engine model parameters. These parameters specify the engine design used for analysis in this report. Atmospheric parameters are presented that specify the flight conditions and humidity of the atmosphere.

Parameter	Value	Units	Comments
W_{TOC}	820.441	lbm/s	TOC total mass flow rate of air
BPR_{TOC}	23.945	–	TOC bypass Ratio
OPR_{TOC}	53.633	–	TOC overall pressure ratio
$PR_{LPC,TOC}$	3.00	–	TOC low-pressure compressor pressure ratio
$PR_{fan,TOC}$	1.30	–	TOC fan pressure ratio
$F_{net,SLS}$	28620.84	lbf	SLS net thrust
$F_{net,CRZ}$	5510.72	lbf	CRZ net thrust
$T_{4,TOC}/T_{4,RTO}$	0.926	–	TOC-to-RTO temperature ratio
$T_{4,RTO}$	3400.0	$^{\circ}R$	RTO combustor temperature
h_{TOC}	0.00017	kg_{water}/kg_{air}	humidity ratio at TOC [?]
h_{CRZ}	0.00017	kg_{water}/kg_{air}	humidity ratio at CRZ [?]
h_{RTO}	0.009	kg_{water}/kg_{air}	humidity ratio at RTO [?]
h_{SLS}	0.009	kg_{water}/kg_{air}	humidity ratio at SLS [?]
LHV_{JetA}	18564.0	BTU/lbm	Lower heating value of Jet-A [?]
LHV_{H2}	51591.0	BTU/lbm	Lower heating value of H2 [?]

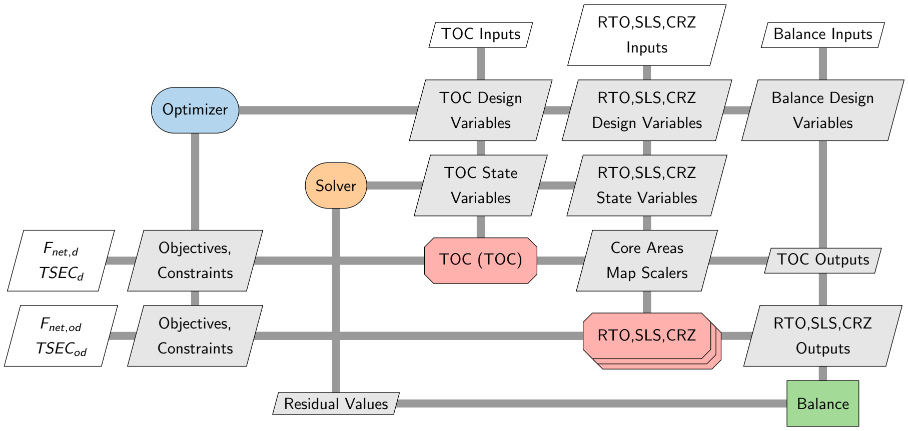

Fig. 8 XDSM diagram of the multipoint optimization problem with constraints. The XDSM diagram shows the variables and outputs within the optimization model and how each of these values is connected to the optimizer and design points.

used to solve large-scale optimization problems (linear and nonlinear). The software uses a sparse SQP (sequential quadratic programming) algorithm with a limited-memory quasi-Newton approximation to the Hessian of the lagrangian. SNOPT is particularly useful for nonlinear problems whose functions and derivatives are expensive compute. An augmented Lagrangian merit function ensures convergence from an arbitrary point. SNOPT allows the nonlinear constraints to be violated (if necessary) and minimizes the sum of these violations. Due to the robustness of the algorithm, SNOPT is a perfect optimizer to solve the highly complex Multipoint systek since the pyCycle engine cycle would occasionally reach a point that is non-physical and would thus cause the solver to fail. SNOPT is able to respond to these solver failures and backtrack to a region in the design space that is physical and continue the optimization.

IV. Results

Table 2 Multipoint optimization problem definition. The 4 design variables are the water recovery fractions of the core exhaust stream at each operation condition. The objective function is the fuel flow rate at the CRZ condition with a net thrust and fan diameter constraint at the design point, TOC.

	Variable/Function	Description	Units	Quantity
minimize	$\dot{m}_{fuel,CRZ}$	Fuel flow rate at CRZ	$\frac{lbm}{s}$	1
with respect to	$x_{H2O,TOC}$	Water recovery fraction at TOC	-	1
	$x_{H2O,RTO}$	Water recovery fraction at RTO	-	1
	$x_{H2O,SLS}$	Water recovery fraction at SLS	-	1
	$x_{H2O,CRZ}$	Water recovery fraction at CRZ	-	1
	$PR_{fan,TOC}$	TOC fan pressure ratio	-	1
	$PR_{LPC,TOC}$	TOC low-pressure compressor pressure ratio	-	1
	$PR_{OPR,TOC}$	TOC overall pressure ratio	-	1
	$T_{4,TOC}/T_{4,RTO}$	TOC-to-RTO temperature ratio	-	1
	$T_{4,RTO}$	RTO combustor temperature	°R	1
	$V_{ratio,CRZ}$	Velocity exit ratio at CRZ		1
		Total		4
subject to	$F_{net,TOC} \geq 5800$ lbf	Target net thrust at TOC	lbf	1
	$D_{Fan} \leq 100\ in^2$	Maximum Fan Diameter	in^2	1
		Total		1

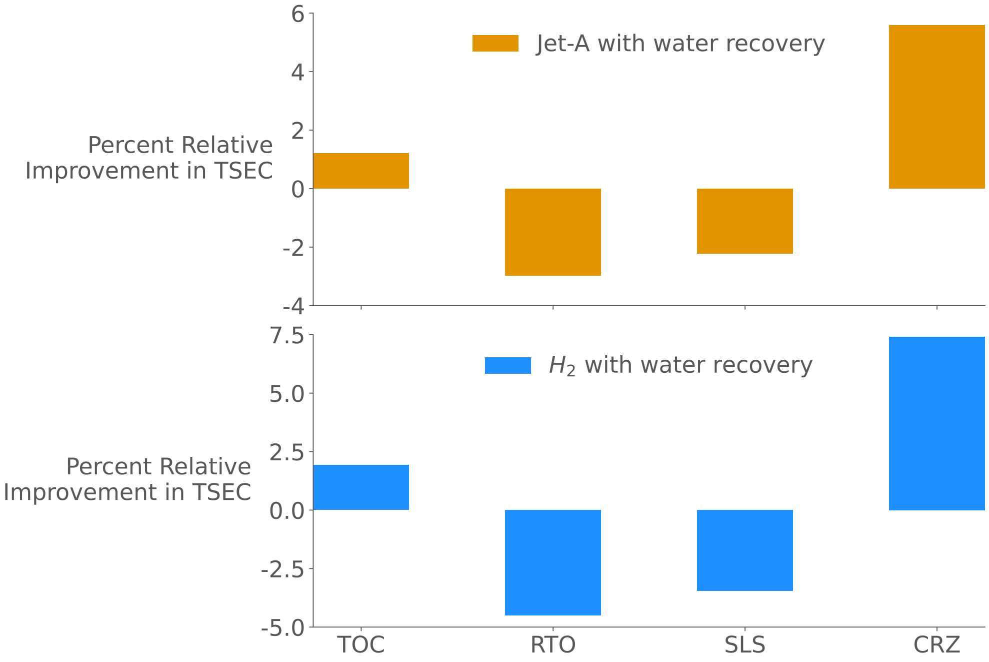

Fig. 9 Percent relative improvement in thrust specific energy consumption (TSEC) of an ultra-high bypass turbofan engine from water injection. Data was collected with and without water injected into the core stream of the engine for Jet-A and hydrogen fuels.

V. Conclusion

Table 3

	Variable/Function	Value	Lower	Upper	Units
objective	$\dot{m}_{fuel,CRZ}$		-	-	$\frac{lbm}{s}$
variables	$x_{H2O,TOC}$	0.053	0.0	-	-
	$x_{H2O,RTO}$	0.0	0.0	-	-
	$x_{H2O,SLS}$	0.0	0.0	-	-
	$x_{H2O,CRZ}$	0.3	0.0	0.3	-
	$PR_{fan,TOC}$			-	-
	$PR_{LPC,TOC}$			-	-
	$PR_{OPR,TOC}$			-	-
	$T_{4,TOC}/T_{4,RTO}$			-	-
	$T_{4,RTO}$			-	°R
	$V_{ratio,CRZ}$			-	
constraints	$F_{net,TOC}$	5800.0	5800.0	-	lbf
	D_{Fan}	100.0	100	-	in^2
		Total			

VI. Acknowledgements