

Control Co-Design Optimization of Spacecraft Trajectory and System for Interplanetary Missions

Gage W. Harris ^{*}, Ping He [†], and Ossama O. Abdelkhalik [‡]

Iowa State University, Ames, IA, 50011

This paper develops a control co-design (CCD) framework to simultaneously optimize the spacecraft's trajectory and onboard system (rocket engine) and quantify its benefit. An open-loop optimal control problem (two-finite burn Mars missions) is used as the benchmark, and the engine design considers the combustion equilibrium and nozzle geometry. The objective function is the fuel burn. The design variables are the trajectory control parameters (such as burn times, burn directions, and time of flight), initial fuel mass, and engine design parameters (such as throat area, mixture ratio, and chamber pressure). The constraints include final velocities and positions of spacecraft. Single-point optimizations are conducted for three departure dates in May, July, and September 2020. A multi-point optimization is also performed to balance the engine performance for these dates with 49 design variables and 20 constraints. It is found that the CCD optimizations exhibit 22% to 28% more fuel burn reduction than the trajectory-only optimization with fixed engine parameters and 16% to 20% more fuel burn reduction than the decoupled trajectory-engine optimization. The proposed CCD optimization framework can be extended to more spacecraft trajectory control parameters and onboard systems and has the potential to design more efficient spacecraft missions.

Nomenclature

A_e	=	Nozzle exit area, m ²
A_t	=	Nozzle throat area, m ²
α	=	Half angle, rad
d_n	=	Thrust directions
F	=	Thrust magnitude, N
γ	=	Specific heat ratio
I_{sp}	=	Specific impulse, s
L	=	Nozzle length, m
λ_b	=	Bell nozzle efficiency
λ_c	=	Conical nozzle efficiency
M_e	=	Exit Mach number
M_R	=	Mixture ratio
$\dot{m}$	=	Mass flow rate, kg/s
m_b	=	Fuel burn, kg
m^{engine}	=	Mass of engine, kg
m^{excess}	=	Excess fuel, kg
m^0	=	Fuel mass before flight, kg
P_c	=	Chamber pressure, MPa
P_e	=	Exit pressure, MPa
P_o	=	Atmospheric pressure, MPa
R	=	Universal gas constant, $\frac{J}{kg \cdot K}$
R_{ex}	=	Nozzle exit radius, m
R_{mag}	=	Center of Mars to spacecraft, km
r	=	Nozzle throat radius, m
T_b	=	Thruster burn time, s

^{*}PhD Student, Department of Aerospace Engineering. AIAA Student Member

[†]Assistant Professor, Department of Aerospace Engineering. AIAA Senior Member. Email: phe@iastate.edu.

[‡]Professor, Department of Aerospace Engineering. AIAA Senior Member

T_c	=	Chamber temperature, K
T_e	=	Exit temperature, K
TOF	=	Time of flight, s
$V_{x,y,z}$	=	Spacecraft relative velocity, km/s
v_e	=	Exhaust velocity, km/s
TMI	=	Trans-Mars injection
MOI	=	Mars orbit insertion
Superscripts		
Earth	=	Variables for the Trans-Mars injection
Mars	=	Variables for the Mars orbit insertion
May	=	Variables for the May departure date
Jul	=	Variables for the July departure date
Sep	=	Variables for the September departure date

I. Introduction

As we continue to advance our understanding of the universe, a scheme to optimize large-scale spacecraft missions efficiently is of growing importance. Cost is the most crucial factor when designing a spacecraft mission. Optimizing a spacecraft's trajectory and onboard systems helps minimize this cost, making space missions more feasible. Therefore, increasing research interests in spacecraft optimization have led to an increase in mission efficiency.

On-board system design aims to increase the efficiency of a specific onboard system. Various studies have been conducted to optimize spacecraft systems, such as the thermal design, as Galski et al. [1] did. His team developed a generalized external optimization algorithm similar to an evolutionary algorithm to find optimal radiator areas for a Brazilian multi-mission platform thermal control system. The stochastic algorithm considered two critical cases: operational hot and cold. Solutions to both cases resulted in new and more efficient design solutions. Frank et al. [2] used the rocket propulsion analysis (RPA) tool to rapidly evaluate chemical rocket engines' performance, weight, and size at a conceptual level. A surrogate model was trained from RPA solutions and used within their framework to develop a tool that can evaluate the performance of chemical rocket engines with an accuracy of 3%. The surrogate model also reduces the computational time by a factor of 10^5 compared with current physics-based models. There have also been studies that optimized two coupled systems (attitude and thermal control). Varatharajoo et al. [3] coupled the attitude control system with the thermal control system by utilizing an electrically conductive fluid that circulates in a closed loop to simultaneously serve as a heat conductor and momentum generator. A five-year mission was simulated with a circular orbit around Earth to test their system. They found the coupled system they created outperformed conventional uncoupled systems from the system volume point of view for spacecraft ranging from 100 to 500 kg. Hwang et al. [4] optimized a CubeSat's full spacecraft system to increase the data downloaded. Using the open-source multidisciplinary analysis design and optimization (OpenMDAO [5]) toolbox, the authors coupled energy generation, communications, attitude dynamics, and thermal control for a 3U CubeSat. With this implementation, they achieved a 40% improvement in the downloaded data.

Trajectory optimization, on the other hand, uses optimal control to minimize the spacecraft's energy consumption (often quantified as velocity change; ΔV) or flight time for a mission sequence. Flyby maneuvers or specific launch windows can reduce these objective functions. Genetic algorithm (GA) is a popular approach to optimize spacecraft trajectory and has been used in many studies [6–9]. The standard GA assumes the design variables of a solution as genes in a fixed-length chromosome. By applying the evolutionary operations of selection, mutation, and crossover, the population of these chromosomes converges to the global optimal solution [10]. Differential evolution is another heuristic method that aims to solve the trajectory optimization problem like genetic algorithms. Labroquère et al. [11] utilized a differential evolution technique and implemented four constraint-handling techniques to optimize a multi-gravity assist interplanetary mission to Jupiter. Three of the four techniques could find feasible solutions within a respectable time. Gradient-based optimization algorithms have also been applied to orbit trajectory problems. Ellithy et al. [12] derived analytical expressions for the gradients needed in optimizing N-impulse orbit transfers for two-body transfers. They used several test cases using different initial conditions to test their analytical gradients against typical finite difference methods. They found that the mission cost for both approaches gave the same values, but the analytical gradients were always better regarding function counts, reducing the computational cost.

Each of the above studies considered the spacecraft trajectory control and system design separately or in a decoupled manner. Optimizing one component while fixing the other component(s) limits the design freedom of the mission.

Control co-design (CCD) is a promising method to break the above limitation by simultaneously optimizing the control and physical systems (see Garcia-Sanz [13] for a detailed review). Recent studies have used CCD for optimizing aircraft [14–16], wind turbines [17–19], suspension systems [20], and hydro turbines [21], to list a few. CCD has also been used in space missions. For example, Chilan et al. [22] developed a CCD framework to optimize the spacecraft solar array structure and control for precision pointing and jitter reduction. Angeletti et al. [23] used the CCD approach to simultaneously optimize the spacecraft’s structure and dynamics to minimize its mass and maximize its agility. Ceccherini et al. [24] conducted a combined system-trajectory optimization for geostationary equatorial orbit transfer. Ricciardi et al. [25] performed a multi-objective (mass and downrange) co-design optimization considering re-entry trajectory and spacecraft mass models. Kluever and Pierson [26] solved a coupled electric spacecraft trajectory and sizing parameters for lunar missions. Koppel [27] found that “medium-high” specific impulse electric thrusters performed better than “very-high” thrusters when taking the spacecraft trajectory into consideration. Arya et al. [28] conducted a joint optimization of spacecraft trajectory and electric propulsion systems to maximize the payload. The propulsion system model was adopted from Petukhov and Wook [29]. Recently, Shimane et al. [30] carried out a multi-objective (mass and time of flight) co-design optimization for low-thrust trajectory and propulsion sizing with fly-bys. Isaji et al. [31] developed a multidisciplinary design optimization framework to couple space mission planning and vehicle design.

Despite the above research progress, existing CCD studies commonly used simple, empirical formulations to correlate system properties or performance with a handful of design parameters. For example, Shimane et al. [30] used a polynomial function to compute the thrust based on the input power for electric propulsion systems. The simplified system design may limit the CCD freedom because the detailed system parameters (e.g., engine nozzle geometry) and key physical processes (e.g., combustion) are not explicitly considered in the optimization. One challenge in physics-based system modeling within a CCD framework is the computational cost; one may need to solve the system’s governing equations in an iterative manner. In addition, as we consider more detailed system design parameters, the proper coupling (data transfer) between the system and trajectory designs requires special considerations. In a recent preliminary study [32], we built an optimization framework that can couple spacecraft trajectory and system design using NASA’s OpenMDAO framework [5]. This paper is a further step in this direction.

This paper’s objective is to outline a CCD framework that can simultaneously optimize spacecraft trajectory control and physics-based system design and quantify the benefit of coupled optimization. We will use an open-loop optimal control problem (two-finite burn Mars mission) as the benchmark, and the onboard system we consider is a rocket engine. Instead of using an empirical formulation to link the engine thrust with I_{sp} , we use an analytical engine model that considers the combustion equilibrium and nozzle geometry. This engine model allows computing engine thrust and I_{sp} based on design variables such as throat area, chamber pressure, and mixture ratio. To alleviate the high computational cost, we train a surrogate model to avoid solving combustion equilibrium equations in the optimization loop. The most original contribution of this paper is adopting physics-based system modeling instead of empirical formulations in the CCD framework. This salient feature is achieved by using gradient-based optimization algorithms for both trajectory and system components, such that we can use a large number of design variables to represent complex systems. This paper will elaborate on special numerical treatments (e.g., coupling variable formulations, explicit and implicit analysis conversion, surrogate modeling) that make the coupled trajectory and physics-based system design computationally efficient. To quantify the benefit of coupled optimization, we will compare its performance against trajectory-only optimization with a fixed system and a decoupled trajectory and system optimization. We will also perform a multi-point optimization to balance the engine performance for different departure dates.

The rest of the paper is organized as follows. The Method Section elaborates on the coupled spacecraft trajectory and engine optimization framework. The optimization results are presented in the Results and Discussion Section. Finally, we summarize our findings in the Conclusion Section.

II. Method

Our proposed CCD optimization framework is called the spacecraft mission optimization toolbox (SMOT). SMOT currently supports two components: interplanetary trajectory computation (control system) and rocket engine model (physical system). These two components are then coupled using OpenMDAO [5] to enable the simultaneous optimization of a spacecraft’s trajectory and system. In this section, we elaborate on our SMOT framework and its components. Following a similar approach, SMOT can be extended to other control parameters and onboard systems.

Table 1 Inputs and outputs of the trajectory computation component.

	Function/Variable	Description	Quantity
Inputs	F^{Earth}	Thrust magnitude (N) - TMI	1
	d_n^{Earth}	Thrust directions - TMI	3
	T_b^{Earth}	Burn time (s) - TMI	1
	I_{sp}^{Earth}	Specific Impulse (s) - TMI	1
	F^{Mars}	Thrust magnitude (N) - MOI	1
	d_n^{Mars}	Thrust directions - MOI	3
	T_b^{Mars}	Burn time (s) - MOI	1
	I_{sp}^{Mars}	Specific Impulse (s) - MOI	1
	TOF	Time of flight (day)	1
	m^0	Fuel mass before flight (kg)	1
	m^{engine}	Mass of engine (kg)	1
	Total Inputs		15
Outputs	R_{mag}	Center of Mars to spacecraft (km)	1
	$ V_{x,y,z} \leq 0.01$	Spacecraft relative velocity (km/s)	3
	m^{excess}	Excess fuel (kg)	1
	m_b	Fuel burn (kg)	1
	Total Outputs		6

A. Spacecraft Trajectory Computation and Control Using General Mission Analysis Tool (GMAT)

NASA's General Mission Analysis Tool (GMAT) [33] is an open-source mission design package. GMAT allows users to easily create spacecraft with a wide range of force models and uses various solvers to efficiently propagate the spacecraft. Users can then seamlessly add specific hardware components, such as fuel tanks and thrusters, to the spacecraft. Once the resources are configured, they are used in the mission sequence to simulate the spacecraft's motion.

A user can interact with GMAT in two ways: using a graphical user interface (GUI) or a scripting language such as Python and MATLAB. In this paper, we develop a Python interface in SMOT to interact with GMAT. This allows users to edit the GMAT scripting language directly through Python commands during optimization.

A fixed mission sequence must be created using the GUI or scripting language for the above Python API. This consists of creating spacecraft, propagators, hardware components, and a mission sequence. The mission sequence could be a Hohmann transfer orbit from LEO to GEO or an interplanetary mission from Earth to Mars. These sequences are "fixed" because the Python API cannot change the sequence itself but rather alter values within the sequence to change the result. Once the fixed mission sequence is created, a user can use the Python API to read the GMAT script file and change variable values in the script before running it. Changing the GMAT script through the Python API is how OpenMDAO couples with GMAT to run optimization. In this study, we use a two-finite burn interplanetary mission to Mars, so a GMAT script was created to perform such a mission. The inputs for this trajectory component include the control parameters for the two finite burns, i.e., thrust magnitude (F), thrust direction (d_n), burn time (T_b), and time of flight (TOF), and the spacecraft parameters such as engine mass (m^{engine}) and initial fuel mass (m^0). The outputs include the position (R_{mag}) and velocity (V_n) of the spacecraft relative to Mars and fuel burn (m_b) for the mission. All inputs and outputs of this mission can be seen in Table 1.

Note that our trajectory problem can also be computed by using a Lambert solver with impulsive burns. Lambert solvers are much faster than finite-burn solvers for computing trajectory problems. In addition, they can be coupled with an engine component by using one extra component to link the Lambert component's output ΔV and the engine component's output I_{sp} . However, we use a finite-burn propagation method instead because it can be easily extended for continuous low-thrust trajectory problems using electric propulsion systems, while solving continuous thrust problems using Lambert solvers is not straightforward. Nevertheless, we expect the finite-burn (propagation) and impulsive-burn (Lambert) methods will compute similar trajectories given the same design parameters, as will be verified in Sec. III.A.

Table 2 The inputs and outputs for the spacecraft propulsion system (engine) component.

	Function/Variable	Description	Quantity
Inputs	M_R	Mixture ratio	1
	P_c	Chamber pressure (MPa)	1
	A_t	Nozzle throat area (m ²)	1
	M_e	Exit Mach number	1
	Total Inputs		4
Outputs	F	Thrust magnitude (N)	1
	I_{sp}	Specific Impulse (s)	1
	A_e	Nozzle exit area (m ²)	1
	m^{engine}	Mass of engine (kg)	1
	Total Outputs		4

In the future, we will add the Lambert solver as an option for impulsive burn problems in our framework.

B. Spacecraft Propulsion System Design Using a Rocket Engine Model

As mentioned above, the spacecraft system considered in this paper is a chemical rocket engine. We implement a rocket engine model in Python to enable the interaction between the rocket engine and spacecraft trajectory.

To be more specific, the inputs of the engine model are the mixture ratio (M_R), combustion chamber pressure (P_c), nozzle throat area (A_t), and exit Mach number (M_e). The outputs are the engine thrust (F), specific impulse (I_{sp}), engine mass (m^{engine}), and exit area (A_e). All inputs and outputs of the engine component can be seen in Table 2. Note that our engine model uses the exit Mach number as an input rather than the exit area. This setting allows the engine model to be completely explicit. In other words, we can directly compute the engine outputs based on the inputs without any iteration needed. The drawback of this choice is that we cannot directly control the throat area of the engine during optimization. Instead, we control the exit Mach number and compute the exit area as the output. Therefore, in the optimization, we impose a constraint to equate the exit areas between both burns, as elaborated in the Results section.

To compute the component outputs, we first built a surrogate model to compute chamber temperature T_c , specific heat ratio γ , and the universal gas constant R based on P_c and M_R . We use the radial basis function surrogate from the surrogate model toolbox (SMT) [34]. We generated 500 sample points for $P_c \in [0.5 : 10.0]$ MPa and $M_R \in [1.0 : 10.0]$. Then, for each P_c and M_R in the sample, we ran the Rocket Propulsion Analysis tool (RPA version 1.2.9 lite) [35] to compute the corresponding outputs: T_c , γ , and R . The radial basis function surrogate model was trained with these outputs. We assume the fuel and oxidizer are liquid hydrogen and oxygen, respectively. The RPA tool solves the theoretical combustion equilibrium equation to compute the outputs.

Once the surrogate model is trained, we assume constant composition in the nozzle and use the following formulation to compute the final outputs of the engine component:

$$\dot{m} = \frac{A_t P_c}{\sqrt{T_c}} \sqrt{\frac{\gamma}{R}} \left(\frac{\gamma + 1}{2} \right)^{\frac{-\gamma-1}{2\gamma-2}} \quad (1)$$

$$\frac{A_e}{A_t} = \left(\frac{\gamma + 1}{2} \right)^{\frac{-\gamma-1}{2\gamma-2}} \left(1 + \frac{\gamma-1}{2} M_e^2 \right)^{\frac{\gamma+1}{2\gamma-2}} M_e^{-1} \quad (2)$$

$$\frac{T_e}{T_c} = \left(1 + \frac{\gamma-1}{2} M_e^2 \right)^{-1} \quad (3)$$

$$\frac{P_e}{P_c} = \left(1 + \frac{\gamma-1}{2} M_e^2 \right)^{\frac{-\gamma}{\gamma-1}} \quad (4)$$

$$v_e = M_e \sqrt{\gamma R T_e} \quad (5)$$

$$F = \dot{m}\lambda_b v_e + (P_e - P_o)A_e \quad (6)$$

To account for frictional losses, we define a nozzle efficiency (λ_b) for a bell nozzle in Eq. 6, following [36]. Initially, the half angle (α) for a conical nozzle with the same throat radius (r), length (L), and exit radius (R_{ex}) is computed using Eq. 7. The efficiency for a conical nozzle (λ_c) is then calculated with Eq. 8 and converted to a bell nozzle using Eq. 9. The coefficient used to convert the conical nozzle efficiency into a bell nozzle efficiency in Eq. 9 is found using the thrust efficiency plot at 100% fractional nozzle length from Figure 4–12 in reference [36].

$$\tan(\alpha) = \frac{R_{ex} - r}{L} \quad (7)$$

$$\lambda_c = \frac{1}{2}(1 + \cos\alpha) \quad (8)$$

$$\lambda_b = \left(\frac{0.992}{0.983}\right)\lambda_c \quad (9)$$

The engine's mass is computed using a formulation developed by Zandbergen [37], valid for chemical rocket engines with thrusts ranging from 15 kN to 8 MN. Since this paper focuses on a two-finite burn mission, two different masses could be computed. However, the same engine is used for both burns, so the burn with the larger thrust magnitude (Earth burn) will be used to calculate the engine mass. Note that the thrust magnitude of the Earth burn is within the above valid range (see Tables 5 to 7). This mass will be fed into the trajectory component to change the spacecraft's dry mass.

$$m^{\text{engine}} = 1.866 \times 10^{-10} F^2 + 0.00130 F + 77.4 \quad (10)$$

As a test, we ran our engine component with the inputs: $P_c = 5.0$ MPa, $M_R = 5.0$, and $M_e = 3.0$ to compute the specific impulse $I_{sp} = 413.17$ s. Then, we ran the RPA tool with the same inputs and the RPA's specific impulse $I_{sp} = 412.48$ s. This reasonably good agreement verifies our engine model implementation.

In addition to the above verification at a single point, we evaluated the surrogate's accuracy for a wide range of P_c and M_R . To do this, we generated another 500 random points for $P_c \in [0.2 : 8]$ MPa and $M_R \in [2 : 6]$. The RPA and our surrogate models were then run separately with the generated inputs to compute T_c , γ , and R . We plot the contours of T_c , R , and γ computed by the RPA model and our surrogate model, as shown in Fig. 1. We observe reasonably good agreement between the contour plots. Moreover, the L2 errors for T_c , γ , and R are 1.595, 0.0006058, and 0.6559, respectively. These results justify our surrogate model's accuracy for a wide range of working conditions.

In terms of runtime, the RPA model takes 0.16 s to run one prediction (i.e., compute T_c , R , and γ based on M_R and P_c), while the surrogate model takes only 0.015 s. The surrogate model achieves a speed-up factor of ten, which further justifies the benefit of using the surrogate model.

C. Open-source Multidisciplinary Analysis Design And Optimization Tool: OpenMDAO

NASA's OpenMDAO [5] is an open-source computing platform for systems analysis and multidisciplinary optimization written in Python. OpenMDAO allows users to decompose large-scale optimization problems into small components. Each component contains basic computation, making them easier to build and maintain. The benefit of OpenMDAO is its focus on gradient-based optimization with analytical derivatives, allowing one to explore large design spaces with thousands of design variables. This salient feature is achieved using the modular analysis and unified derivatives (MAUD) architecture proposed by Hwang and Martins [38].

As previously stated, OpenMDAO can decompose complex models into smaller disciplines or components. Each component is defined by its specific inputs and outputs. In a component, one must implement how to compute the outputs based on the inputs. This can be done with an analytical expression, as was done for the rocket engine model, or by external software such as GMAT. Lastly, one needs to define how the derivatives of the outputs with respect to the inputs will be calculated for each component. This, again, can be done with analytical expressions or by using finite differences. This study uses the finite-difference method to compute all the partial derivatives because it is easier to implement (analytical derivatives are not available in GMAT). Once all the individual components are set, one can link variables (i.e., inputs and outputs) across components to achieve a coupled model. Then, one can select any of the variables in the coupled model as the objective function, design variables (input variables only), and constraints. This feature greatly increases the flexibility of large-scale optimization. For example, we can easily add more complex constraints to the trajectory problem, such as the path constraint [39, 40]. Imagine one needs to ensure the spacecraft position is greater

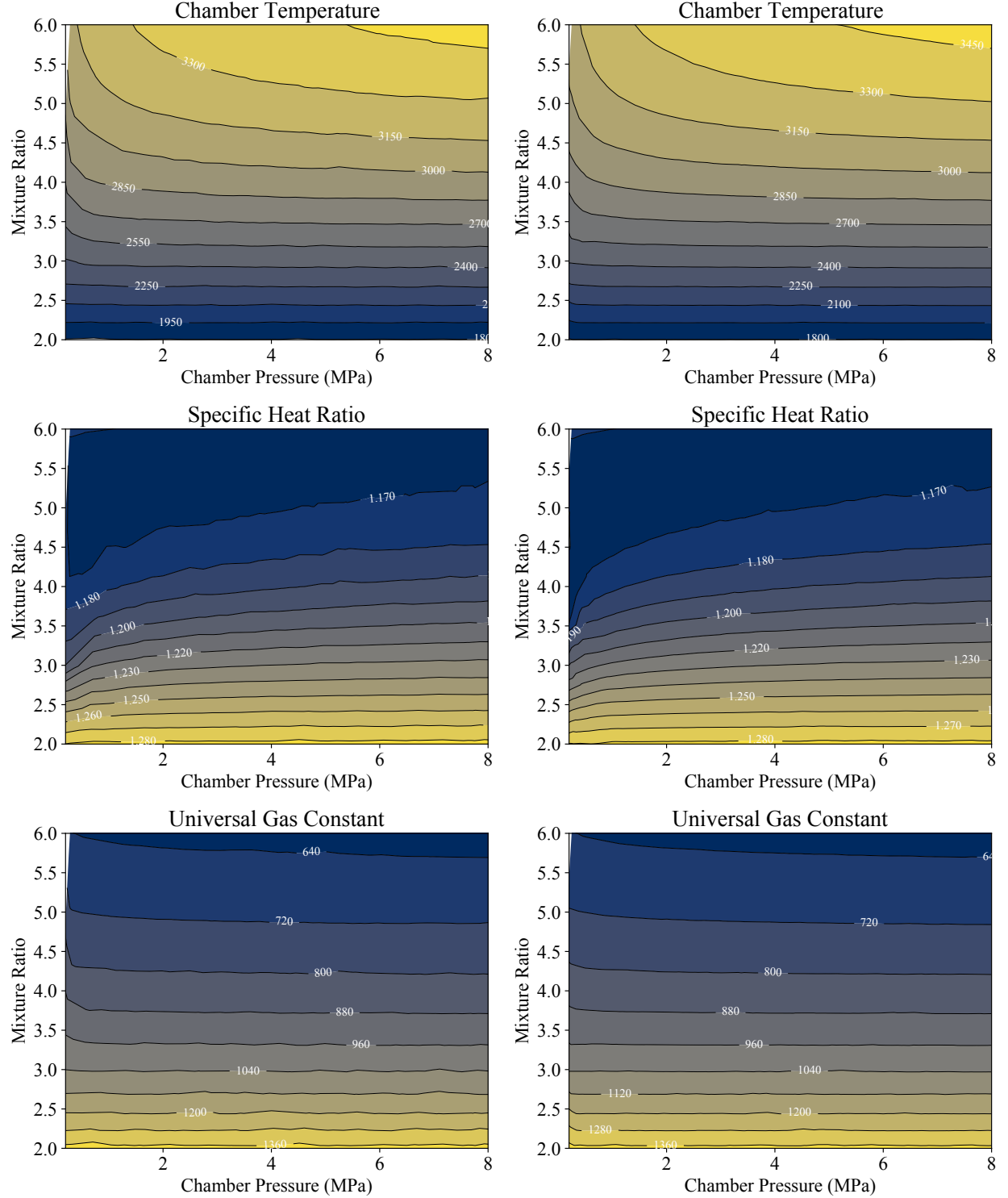


Fig. 1 Comparison of T_c , γ , and R contours between the RPA (left column) and surrogate (right column) models.

than d_{safe} from a planet to avoid a collision. The only required change is to compute the minimal distance d_{min} between the spacecraft and the planet and add it as an output for the trajectory component. Then, OpenMDAO could select d_{min}

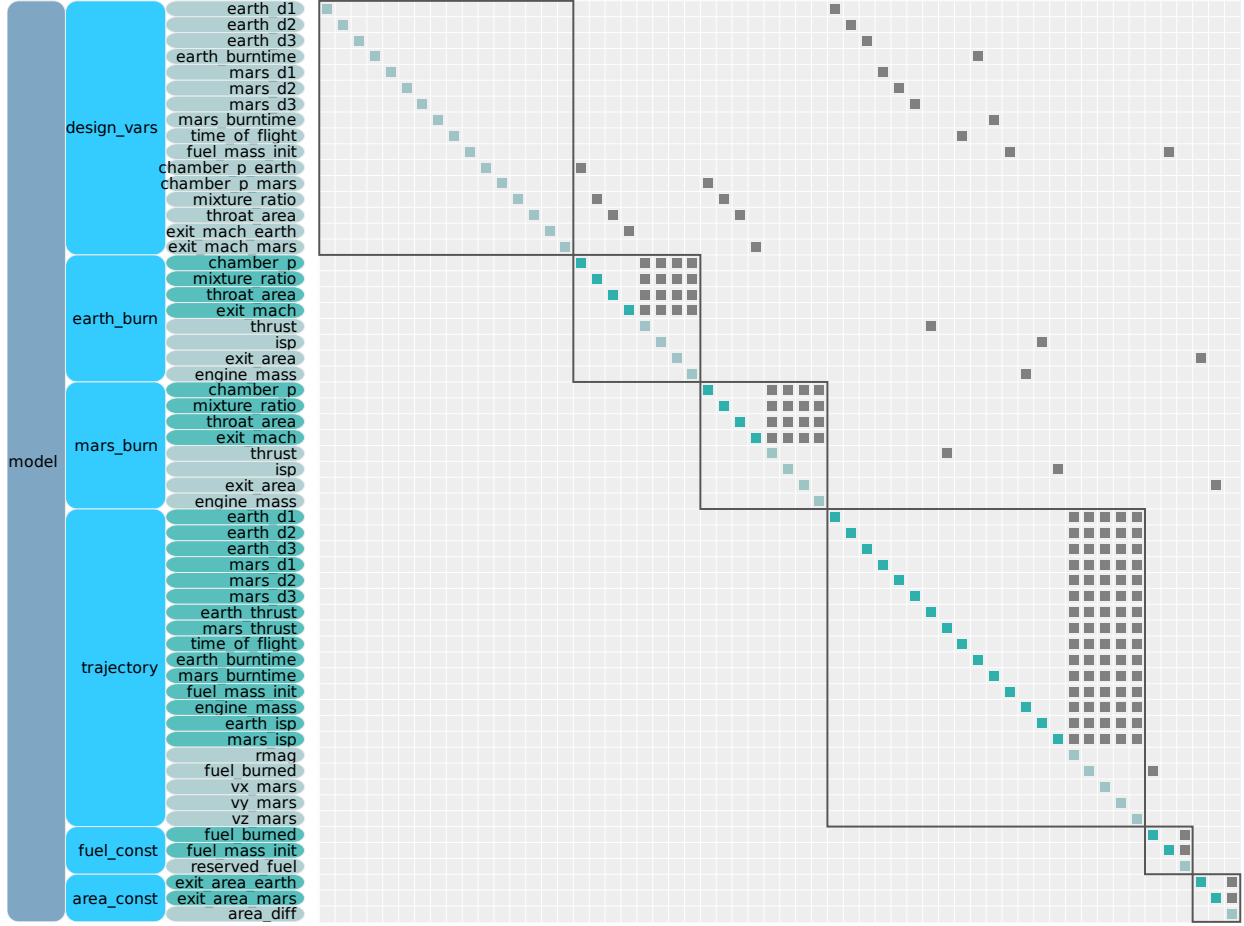


Fig. 2 OpenMDAO N2 diagram to illustrate the interaction between the trajectory and engine components during a CCD optimization.

as a constraint and enforce $d_{\min} > d_{\text{safe}}$ during optimization.

Figure 2 shows an example of the proposed spacecraft system and trajectory coupling through OpenMDAO's N2 diagram. The light blue blocks are the components, the dark green items denote the inputs of a component, and the light green represents the outputs. The off-diagonal grey blocks represent the data transfer between the components' input and output variables. The "fuel_const" and "area_const" components are for the excess fuel and nozzle exit area constraints (see Table 4 for details). As shown, in a coupled optimization, the design variables are the control and physical system parameters such as the direction of the burns, burn time, time of flight, mixture ratio, chamber pressure, throat area, and exit Mach number. The engine component will compute the thrust and I_{sp} ; both are then used in the GMAT component to compute the trajectory, subject to certain constraints. The objective function of the mission is the total fuel burn.

D. Control co-Design Optimization Using Spacecraft Mission Optimization Tool: SMOT

Space mission optimization tool (SMOT) is a Python interface we developed to combine GMAT, engine model, and OpenMDAO for control-code design optimization of spacecraft trajectory and system. OpenMDAO acts as the top-level driver and is where all the various components of the mission will be defined (e.g., spacecraft trajectory and rocket engine models). GMAT is used as the trajectory analysis tool, and the engine model is implemented as an OpenMDAO component.

A basic overview of the coupled model can be seen in Fig. 3. The diagonal blocks are the components, and the off-diagonal blocks denote the data transfer between components. A component receives the inputs from the vertical direction and outputs data in the horizontal direction. The gray blocks in Fig. 3 denote the data transfer between the

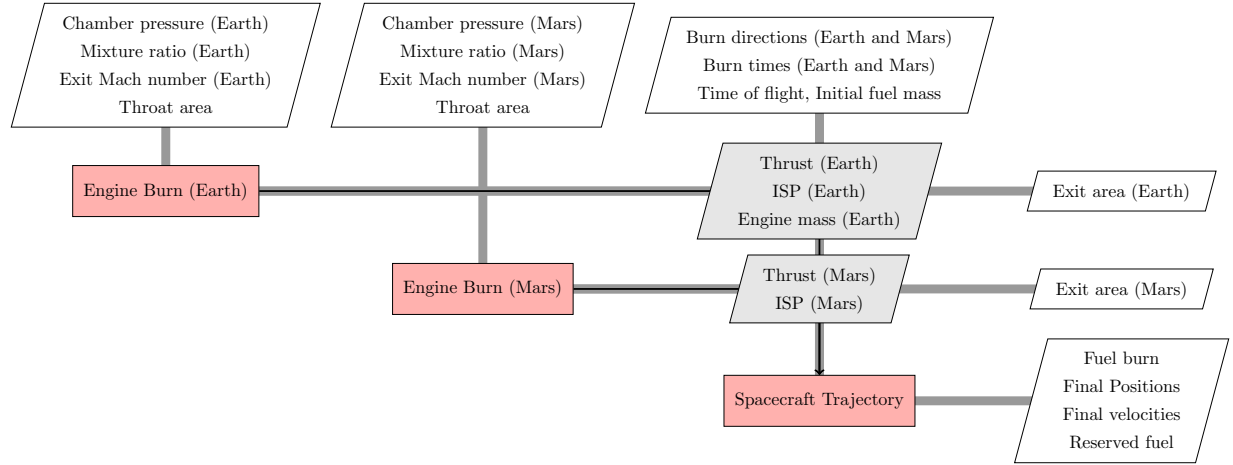


Fig. 3 CCD optimization of spacecraft trajectory and engine design for the two-burn Mars mission, illustrated using the extended design structure matrix (XDSM) diagram [41].

engine and trajectory components. To be more specific, the engine components (Earth and Mars burns) use the engine parameters (e.g., chamber pressures, mixture ratios, throat area) as inputs and output the thrusts, I_{sp} , and engine mass to the trajectory component. The trajectory component then uses them, along with trajectory control parameters (e.g., burn directions, burn magnitudes, and time of flight), as the inputs and output fuel burn, final positions and velocities, etc. There is no feedback from the trajectory to the engine component; the trajectory-engine components are one-way coupled.

Currently, only a two-finite burn interplanetary mission to Mars is set up in SMOT for optimization. The trajectory component will change entirely based on the optimized mission type. For example, the trajectory component for an interplanetary mission to Mars will look different from that of an interplanetary mission to Jupiter. This is because each mission has a fixed GMAT mission sequence assigned to it. In the future, we plan to have a variety of trajectory components defined in SMOT to optimize a wide range of missions. The beauty of SMOT is that we can neglect components we do not wish to use.

A Python run script is how a user will interact with SMOT. Users will decide which type of mission they want to optimize in the run script. This run script will tell SMOT which specific trajectory component to use and which GMAT script to read. In the future, we also plan to let the user choose which spacecraft systems components they want to be included in the optimization. The run script allows the user to set up the initial conditions and modify constraints to tailor the mission type to their requirements.

As previously stated, there is only one trajectory component in SMOT, a two-finite burn interplanetary mission to Mars. The mission can either be optimized by itself (e.g., trajectory only) or coupled with the engine component. More on this mission will be under the results section of this paper. In the future, we also plan to add more spacecraft system formulations, such as a solar electric propulsion engine model and a power generation model utilizing solar panel design.

E. Scalability of the SMOT Framework

This section discusses how the optimization computation cost scales when the framework is built upon further. We first discuss how additional design variables and constraints will impact the optimization speed. As mentioned in Section II.C, we use the finite-difference method to compute all the partial derivatives within OpenMDAO. Using the finite-difference method causes the computational time to scale linearly with each addition of the design variable. Therefore, as the framework is built up by adding design variables, the optimization computational time will increase linearly. To further improve the scalability, we will implement the adjoint method [42–44] to efficiently compute gradients with a large number of design variables. Note that the finite-difference computational cost is independent of the number of constraints (or competing objective functions), while the adjoint cost scales linearly with the number of constraints.

Next is how the optimization scales when more complicated spacecraft systems are added. As mentioned above,

OpenMDAO allows us to couple any number of components in a flexible way. However, as more subsystems are added to the framework, we expect the number of design variables will also increase. In addition, computing the system's output becomes more expensive because we need to solve/compute more components. Overall, we expect the optimization computation cost to increase linearly as more components are added.

This last paragraph will address the extension of the proposed framework to discrete optimization. Our current engine design method uses a continuous design space and may have practical challenges. Specifically, an optimized engine implies developing and fabricating a unique engine that works best for a specific mission. However, the industry might not deem the "best engine" because the cost and schedule additions of development and testing of a new engine potentially outweigh the delivered mass benefits of the customized engine. Instead, implementing discrete optimization, e.g., having a few distinct engines one could choose between instead of a continuous space of possible engine parameters, would appeal to industry standards. Even though the discrete optimization space might seem more practical from an industry perspective, we aimed to show the benefit of a fully coupled optimization for academic purposes. In the future, we will couple OpenMDAO with genetic algorithms, allowing us to consider more complex missions with discrete design variables, such as planet fly-by sequences, departure dates, and discrete engine parameters.

III. Results and Discussion

This section demonstrates the CCD optimization results for the interplanetary mission to Mars. Only two finite burns are allowed for these optimizations, and no fly-bys are being considered. Also, this mission is done with the Sun being the only gravitational body; forces from Earth and Mars are neglected. Three departure dates are used: May 27th, July 27th, and September 8th, 2020. We chose all of these departure dates using a pork chop plot from Conte et al.[45]. The objective is to compare the performance of a CCD trajectory-engine optimization to a trajectory-only optimization with fixed-engine parameters and a decoupled trajectory-engine optimization.

A. Comparison between CCD and trajectory-only optimizations

We first consider single-point optimization, i.e., optimization for a specific departure date. We compare optimization results from five different configurations, as shown in Table 3. The first configuration was trajectory-only optimization and used a fixed engine performance, namely I_{sp} [46]. As will be shown later, the trajectory-only and decoupled optimization results depend on the choice of engine I_{sp} . To make the optimization relevant to practical cases, we chose the RL-10A-4-2 engine based on NASA's MAVEN mission and used its parameters (e.g., throat area, chamber pressure, and mixture ratio) as initial guesses for all optimization cases. The second configuration was a trajectory optimization with fixed engine geometry (FEG). It used the geometry specifications of the RL-10A-4-2 engine but allowed the chamber pressure and mixture ratio to change to achieve the desired thrust. The mixture ratios between the two burns are the same. The third configuration was similar to the last but allowed different mixture ratios between the two burns. The last two configurations were CCD optimization that simultaneously changed the trajectory control parameters (e.g., burn magnitude, burn direction, and time of flight) and engine parameters (e.g., geometry, chamber pressure, and mixture ratio). The first CCD configuration required the mixture ratio of both burns to equal each other, while the second CCD configuration (CCD MR) allowed both mixture ratios to vary.

The optimization formulation for the most complicated CCD-MR configuration can be seen in Table 4. The optimization executes two finite burns: one to leave Earth and the second to rendezvous with Mars. Both burns use a constant thrust direction determined by the optimizer. The initial position and velocity of the spacecraft are the same as Earth's at the departure date. Essentially, the spacecraft is inside Earth at the start of the optimization. This is done to simplify the optimization, as is done for the development of most pork chop plots.

The optimizations aim to have the spacecraft reach the center of Mars within a 3000 km sphere. To complete the rendezvous, the spacecraft must have relative velocities within 10 m/s of Mars. We added the forced excess fuel constraint to ensure some future utility from the spacecraft. Finally, the exit area constraint ensures that both burns have the same nozzle geometry. The nozzle geometry must be the same because both burns use the same engine. All optimizations were run using the Sparse Nonlinear OPTimizer (SNOPT) package [47]. On average, the optimization converged in about 10 minutes on a Dell workstation.

Optimization convergence plots of the May 27th CCD configuration can be seen in Fig. 4. The optimization converged in 145 iterations with a significant reduction in the fuel burn (Fig. 4a). The time of flight increased in the optimized design (Fig. 4b), which corresponds to a decreased Earth and Mars thrust magnitude (Fig. 4c). Along with the thrust magnitudes, the engine throat area (Fig. 4d) decreased throughout the optimization.

All three departure dates were run using all five configurations, and their results can be seen in Tables 5, 6, and 7.

Table 3 Description of five optimization configurations

Configuration	Description
Trajectory	Trajectory-only optimization with fixed engine I_{sp} . It used a fixed RL-10A-4-2 engine performance, namely I_{sp} , and all other engine performance parameters were neglected.
FEG	Trajectory optimization with fixed engine geometry (FEG). It used the geometry specifications of RL-10A-4-2 but allowed the chamber pressure and mixture ratio to change for the desired thrust. The mixture ratios between the two burns are the same.
FEG-MR	Same as FEG except that we allowed different mixture ratios between the two burns.
CCD	Control co-design (CCD) optimization that simultaneously changes the trajectory control parameters (e.g., burn magnitude, burn direction, and time of flight) and engine parameters (e.g., engine geometry, chamber pressure, and mixture ratio). The mixture ratios between the two burns are the same.
CCD-MR	Same as CCD except that we allowed different mixture ratios between the two burns.

Table 4 Optimization formulation for the most complicated configuration (control co-design with variable mixture ratio; CCD-MR).

	Function/Variable	Description	Quantity
Minimize	m_b	Total fuel burned (kg)	1
w.r.t	M_R^{Earth}	Mixture ratio - TMI	1
	M_R^{Mars}	Mixture ratio - MOI	1
	p_c^{Earth}	Chamber pressure (MPa) - TMI	1
	p_c^{Mars}	Chamber pressure (MPa) - MOI	1
	A_t	Throat area (m ²)	1
	M_e^{Earth}	Exit Mach number - TMI	1
	M_e^{Mars}	Exit Mach number - MOI	1
	d_n^{Earth}	Thrust directions - TMI	3
	T_b^{Earth}	Burn time (s) - TMI	1
	d_n^{Mars}	Thrust directions - MOI	3
	T_b^{Mars}	Burn time (s) - MOI	1
	TOF	Time of flight (day)	1
	m^0	Fuel mass before flight (kg)	1
	Total Design Variables		17
Subject to	$R_{\text{mag}} \leq 3000$	Center of Mars to spacecraft (km)	1
	$ V_{x,y,z} \leq 0.01$	Spacecraft relative velocity (km/s)	3
	$m^{\text{excess}} \geq 200$	Forced excess fuel (kg)	1
	$A_e^{\text{Earth}} = A_e^{\text{Mars}}$	Earth-Mars exit areas equal	1
	Total Constraints		6

The general trend of the data is that the fuel burn reduction increases as we go from left to right columns, as expected. More design freedom results in better performance, with the CCD-MR optimizations having the best results. The CCD-MR optimizations obtain 22% to 28% more fuel burn reduction than the trajectory-only optimization with fixed

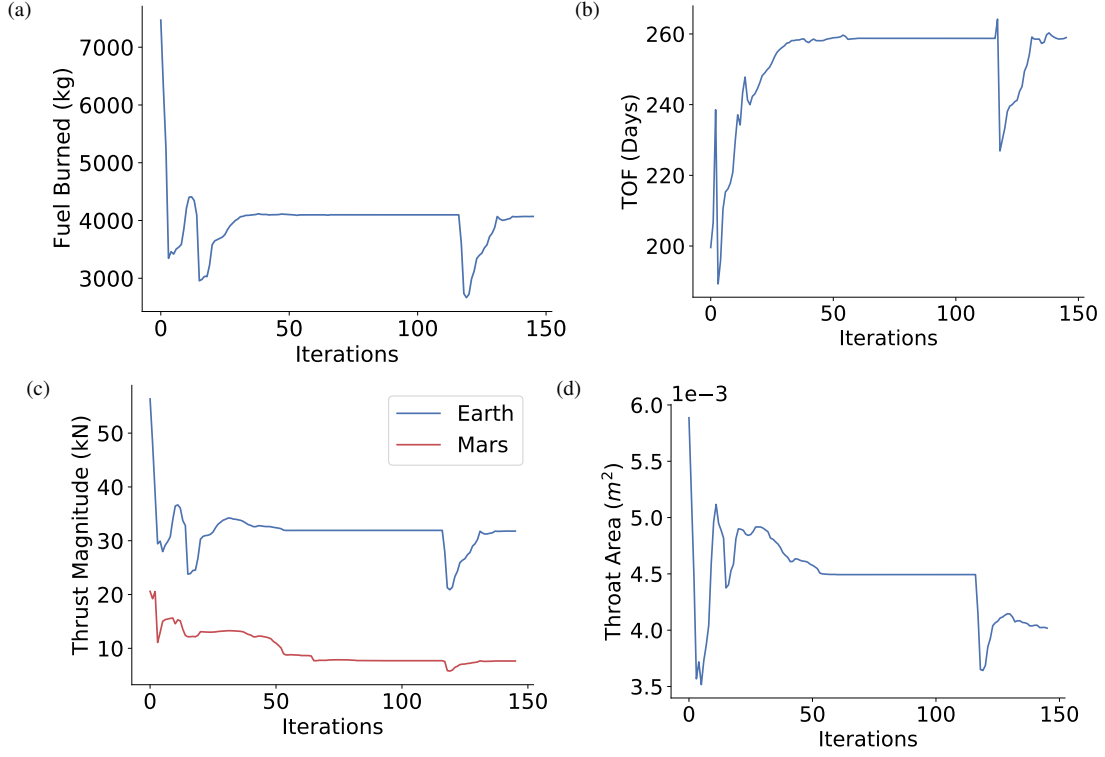


Fig. 4 Optimization convergence plots of various variables for the May 27th CCD configuration.

engine parameters. This reduction arises because the CCD optimizations can fully tailor the engine to the specific mission, changing the engine's geometry and weight to maximize I_{sp} .

A direct comparison of this reduction can be seen in Figure 5, which shows how the spacecraft's wet mass changes over the trajectory for the trajectory-only and CCD-MR optimizations during the May 27th departure date. The trajectory-only optimization starts and ends with a wet mass of 6160 kg and 1010 kg, respectively, while the CCD-MR optimization starts and ends with 4935 kg and 921 kg, respectively. Note that the trajectory-only optimization starts and ends with a higher wet mass than the CCD-MR optimization. This is caused by the trajectory-only optimization using a larger engine, the default RL-10A-4-2, compared with the optimized CCD-MR engine. A larger engine equates to a heavier dry mass and, in turn, requires larger amounts of fuel to propel the spacecraft.

In addition to a smaller engine, a higher I_{sp} will enable a more efficient mission. Figure 6 shows how the baseline (RL-10A-4-2) and optimized CCD-MR engines' I_{sp} will vary depending on the desired thrust. The engines for the May 27th departure date are shown, but all others follow the same pattern. In coupled optimization cases, the optimizer maximizes the engine's I_{sp} for the specific desired thrust and not a range of possible thrusts. By only looking at a specific thrust, the coupled optimization creates the best engine for a specific mission. This is why the CCD-MR engine has a larger I_{sp} than the baseline engine at the operating thrusts presented in Table 5. However, even though the optimizer only considers a single thrust, the CCD-MR engine is still more efficient for a much larger range of desired operating thrusts than the baseline engine.

One thing to point out is that the FEG single mixture ratio optimization is worse than the trajectory-only optimization. The reason for this is that the RL-10A-4-2's engine I_{sp} of 445.7 is at the desired operating thrust of about 100 kN, and the trajectory-only optimization assumes the engine maintains this I_{sp} at all thrust levels. The FEG single mixture ratio optimization uses the same engine but manually changes the chamber pressure to achieve the desired thrust. The RL-10A-4-2 engine is designed for much higher thrust values, so the actual I_{sp} will be less when operating at lower thrust values, as is shown in the FEG cases. The FEG-MR case does not suffer this problem because the mixture ratio can also be changed to maximize I_{sp} .

Another thing to point out is that the CCD optimization is only slightly worse than the CCD-MR optimization. Many design variables between the two are similar, some are even identical such as the Earth's burn duration, which meets the

Table 5 Comparison of May 27, 2020 optimization results among five configurations. CCD-MR is 24.8% more fuel efficient than trajectory-only.

	Function/Variable	Trajectory	FEG	FEG-MR	CCD	CCD-MR
Objective	m_b (kg)	5150	5250	4819	4071	4014
Constraints	R_{mag} (km)	3004	3025	3000	2807	2856
	$ V_{x,y,z} $ (km/s)	<0.01	<0.01	<0.01	<0.01	<0.01
	m^{excess} (kg)	200	200	200	200	200
	$ A_e^{\text{Earth}} - A_e^{\text{Mars}} $	$< 10^{-8}$	$< 10^{-8}$	$< 10^{-8}$	$< 10^{-8}$	$< 10^{-8}$
Variables	F_{Earth} (N)	51200	38580	36540	31800	31520
	F_{Mars} (N)	15730	38580	6974	7644	6399
	I_{sp} (s)	445.7	441.8	460.0	477.5	480.6
	A^e / A_t	61	61	61	126.9	153.4
	m^{engine} (kg)	206	206	206	118.9	118.6
	M_R^{Earth}	NA	5.5	4.066	4.086	4.266
	M_R^{Mars}	NA	5.5	3.578	4.086	3.976
	P_c^{Earth} (MPa)	NA	1.570	1.501	4.081	5.0
	P_c^{Mars} (MPa)	NA	1.570	0.2864	0.9732	1.012
	A_t (m ²)	NA	0.013	0.013	0.004018	0.003222
	M_e^{Earth}	NA	4.31	4.31	4.945	5.046
	M_e^{Mars}	NA	4.31	4.31	4.854	5.009
	d_{Earth}^1	0.5819	0.5819	0.5820	0.5820	0.5819
	d_{Earth}^2	-0.3229	-0.3223	-0.3228	-0.3207	-0.3221
	d_{Earth}^3	-0.7464	-0.7466	-0.7463	-0.7473	-0.7467
	T_b^{Earth} (s)	370.9	498.1	499.0	500.0	500.0
	d_{Mars}^1	0.9734	0.9735	0.9734	0.9739	0.9735
	d_{Mars}^2	-0.2288	-0.2280	-0.2287	-0.2258	-0.2279
	d_{Mars}^3	0.0151	0.0172	0.0151	0.0235	0.0179
	T_b^{Mars} (s)	224.0	91.59	500.0	411.9	490.9
	TOF (d)	258.4	258.6	258.4	259.0	258.6
	m^0 (kg)	5350	5450	5019	4271	4214

optimizer's upper bound of 500 seconds. There is only one extra design variable in the CCD-MR optimization, i.e., the varied mixture ratio, and it does not greatly impact the optimization result. One reason for this could be that during the CCD optimization, the engine already has a great deal of design freedom. The new design variable adds only marginal extra design freedom, but without it, the current engine design variables of the CCD formulation can still find a similar solution. Additionally, each optimization was very sensitive to specific design variables, namely the burn directions. If all other design variables were kept constant, small changes in the burn direction would result in large variations in the R_{mag} . That is why the burn directions look almost equal between optimization configurations.

Each departure date requires a larger thrust when leaving Earth compared with rendezvousing with Mars. That is why the chamber pressure for the Earth burn is always larger than the Mars burn. Both burns utilize the same engine geometry, so larger thrust magnitudes require larger chamber pressures. Engine sizing also depends on the thrust magnitudes required. The CCD formulations show that the lower the needed thrust, the smaller the throat area and the larger the expansion ratio. This allows I_{sp} to be maximized efficiently. The July departure date has the lowest thrust magnitudes and, thus: the lowest engine weight, smallest throat area, highest expansion ratio, and largest I_{sp} value.

Table 6 Comparison of July 27, 2020 optimization results among five configurations. CCD-MR is 27.8% more fuel efficient than trajectory-only.

	Function/Variable	Trajectory	FEG	FEG MR	CCD	CCD-MR
Objective	m_b (kg)	3258	3348	3076	2474	2462
Constraints	R_{mag} (km)	3200	3234	3120	3000	2940
	$ V_{x,y,z} $ (km/s)	<0.01	<0.01	<0.01	<0.01	<0.01
	m^{excess} (kg)	200	200	200	200	200
	$ A_e^{\text{Earth}} - A_e^{\text{Mars}} $	$< 10^{-8}$	$< 10^{-8}$	$< 10^{-8}$	$< 10^{-8}$	$< 10^{-8}$
Variables	F_{Earth} (N)	29860	23040	21750	17500	17450
	F_{Mars} (N)	15180	23040	38390	7014	7320
	I_{sp} (s)	445.7	439.4	459.6	487.3	488.6
	A^e/A_t	61	61	61	212.9	237.1
	m^{engine} (kg)	206	206	206	100.2	100.1
	M_R^{Earth}	NA	5.5	3.859	4.211	4.298
	M_R^{Mars}	NA	5.5	4.014	4.211	4.118
	P_c^{Earth} (MPa)	NA	0.9374	0.8942	4.562	5.0
	P_c^{Mars} (MPa)	NA	0.9374	1.578	1.819	2.095
	A_t (m ²)	NA	0.013	0.013	0.001932	0.001752
	M_e^{Earth}	NA	4.31	4.31	5.280	5.339
	M_e^{Mars}	NA	4.31	4.31	5.215	5.323
	d_{Earth}^1	0.9242	0.9251	0.9250	0.9250	0.9250
	d_{Earth}^2	0.3079	0.3046	0.3051	0.3050	0.3051
	d_{Earth}^3	0.2259	0.2266	0.2265	0.2265	0.2265
	T_b^{Earth} (s)	358.9	472.3	476.9	500.0	500.0
	d_{Mars}^1	0.7778	0.7760	0.7762	0.7762	0.7782
	d_{Mars}^2	0.4725	0.4692	0.4697	0.4697	0.4657
	d_{Mars}^3	-0.4144	-0.4215	-0.4206	-0.4205	-0.4213
	T_b^{Mars} (s)	232.4	154.1	91.15	438.4	419.7
	TOF (d)	205.2	204.7	204.8	204.8	204.8
	m^0 (kg)	3458	3548	3276	2674	2662

While the September departure date has the largest thrust magnitudes and, thus: the highest engine weight, largest throat area, smallest expansion ratio, and smallest I_{sp} value. From the data, it is clear that the July departure date is the best day to launch.

The comparison of spacecraft trajectory between the initial conditions and CCD optimizations for the May 27th case is shown in Figure 7. All optimization cases follow a similar trend, so only one is shown here. The initial conditions are feasible trajectories computed using GMAT, and the same initial conditions are used for each optimization framework within a departure date. The purpose of the reference initial conditions is to allow accurate comparisons between methods. Therefore, we did not include the initial conditions in Tables 5 to 7. For each departure date, the main difference that can be seen is the increase in flight time. This increase in TOF is one way the optimizer decreases the fuel burned. While all optimizations within the same departure date have different solutions, the variations cannot fully be seen in the plots due to the magnitude of the distances. The plots were plotted in Sun ecliptic X-Y coordinates because the orbital inclination change between Earth and Mars is minimal.

As mentioned above, we use the finite-burn propagation method to compute the spacecraft trajectory instead of the

Table 7 Comparison of September 8, 2020 optimization results among five configurations. CCD-MR is 21.8% more fuel efficient than trajectory-only.

	Function/Variable	Trajectory	FEG	FEG MR	CCD	CCD-MR
Objective	m_b (kg)	7103	7192	6626	5739	5705
Constraints	R_{mag} (km)	3000	3024	3003	2439	3000
	$ V_{x,y,z} $ (km/s)	<0.01	<0.01	<0.01	<0.01	<0.01
	m^{excess} (kg)	200	200	200	200	200
	$ A_e^{\text{Earth}} - A_e^{\text{Mars}} $	< 10^{-8}	< 10^{-8}	< 10^{-8}	< 10^{-8}	< 10^{-8}
Variables	F_{Earth} (N)	55240	55980	53880	45630	45450
	F_{Mars} (N)	13200	55980	56210	9083	7670
	I_{sp} (s)	445.7	443.4	459.0	473.9	475.3
	A^e/A_t	61	61	61	112.1	112.5
	m^{engine} (kg)	206	206	206	137.1	136.9
	M_R^{Earth}	NA	5.5	3.995	4.137	4.232
	M_R^{Mars}	NA	5.5	4.005	4.137	3.892
	P_c^{Earth} (MPa)	NA	2.279	2.220	4.715	5.0
	P_c^{Mars} (MPa)	NA	2.279	2.316	0.9302	0.8413
	A_t (m ²)	NA	0.013	0.013	0.005025	0.004701
	M_e^{Earth}	NA	4.31	4.31	4.856	4.841
	M_e^{Mars}	NA	4.31	4.31	4.761	4.808
	d_{Earth}^1	0.5331	0.5330	0.5321	0.5338	0.5338
	d_{Earth}^2	0.2554	0.2553	0.2531	0.2571	0.2571
	d_{Earth}^3	0.8066	0.8067	0.8080	0.8056	0.8056
	T_b^{Earth} (s)	486.0	483.4	476.7	500.0	500
	d_{Mars}^1	0.8055	0.8059	0.8087	0.8033	0.8034
	d_{Mars}^2	0.3367	0.3364	0.3349	0.3378	0.3378
	d_{Mars}^3	0.4877	0.4871	0.4836	0.4905	0.4903
	T_b^{Mars} (s)	319.0	75.34	73.93	422.9	500
	TOF (d)	242.2	242.1	241.4	242.7	242.6
	m^0 (kg)	7303	7392	6826	5939	5905

impulsive burn method with a Lambert solver. We expect that these two methods will compute similar trajectories for our case. To verify this point, we use the optimized design from the CCD-MR case as the benchmark and compute its trajectory using the finite-burn and impulsive-burn methods. The fuel burns computed from these two methods are summarized in Table. 8. We used the universal variable algorithm to solve the Lambert problem. The change in fuel mass for the Lambert approach was calculated using the rocket equation with the I_{sp} given from the CCD-MR optimization case. The fuel burns computed from these two methods are similar for all the departure dates, indicating that they compute similar trajectories for our case.

B. Comparison between CCD and trajectory-engine decoupled optimizations

To further quantify the benefit of coupled optimization, we compare the performance between the CCD and trajectory-engine decoupled optimizations in this subsection. The decoupled configuration ran the trajectory and engine optimizations separately. To be more specific, we first ran the trajectory-only optimization with one of the three departure dates. These trajectory-only optimizations are the same as presented in Tables 5, 6, and 7. The

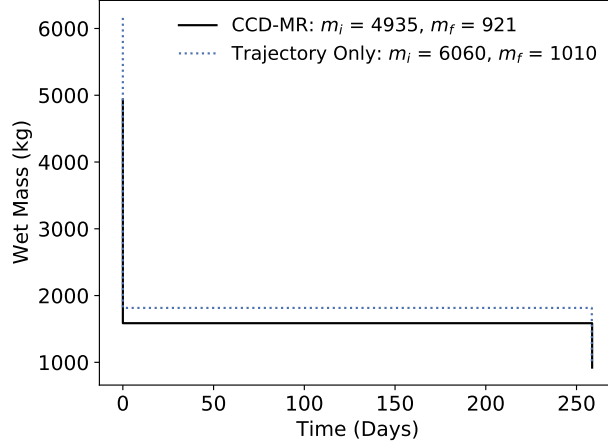


Fig. 5 Wet mass comparison for the trajectory only and CCD-MR optimizations during the May 27th case. Specific values for initial wet mass m_i and final wet mass m_f are given for both optimisations.

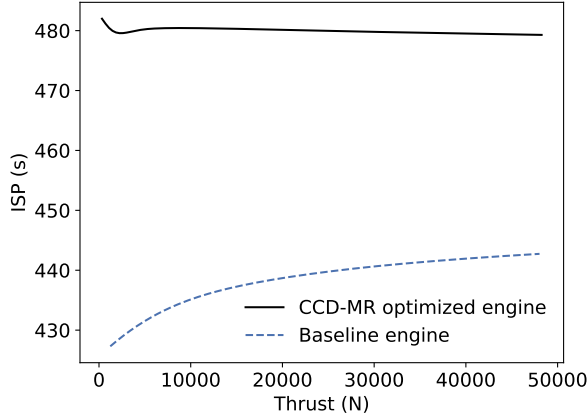


Fig. 6 Thrust vs. I_{sp} for the baseline (RL-10A-4-2) engine and the optimized CCD-MR engine during the May 27th case.

Table 8 Fuel burn (kg) comparison between the finite-burn (propagation) and impulsive-burn (Lambert) trajectory computation methods for each departure date.

Trajectory computation method	May	Jul.	Sep.
Finite burn (propagation)	4014	2462	5705
Impulsive burn (Lambert)	4029	2473	5700

trajectory-only optimization then outputted the required thrust magnitudes to the engine-only optimization. The engine-only optimization formulation can be seen in Table 9. The objective function is the average I_{sp} between two burns. The design variables are the engine parameters, such as chamber pressures, throat area, and mixture ratios. In terms of constraints, the thrust magnitudes are fixed to the values computed from the trajectory-only cases. We also require the exit areas to be equal between the Earth and Mars burns. The decoupled configuration ran the above trajectory-only and engine-only optimizations only once; there is no iteration between them.

The total fuel burn of the engine optimization was calculated using the mass flow rate for each burn using Eq. 11. The mass flow rate (see Eq. 1) was an intermediate value found during the engine optimization. The mass flow rate was then multiplied by the respective burn time, which was found during the trajectory-only optimization. Note that the burn

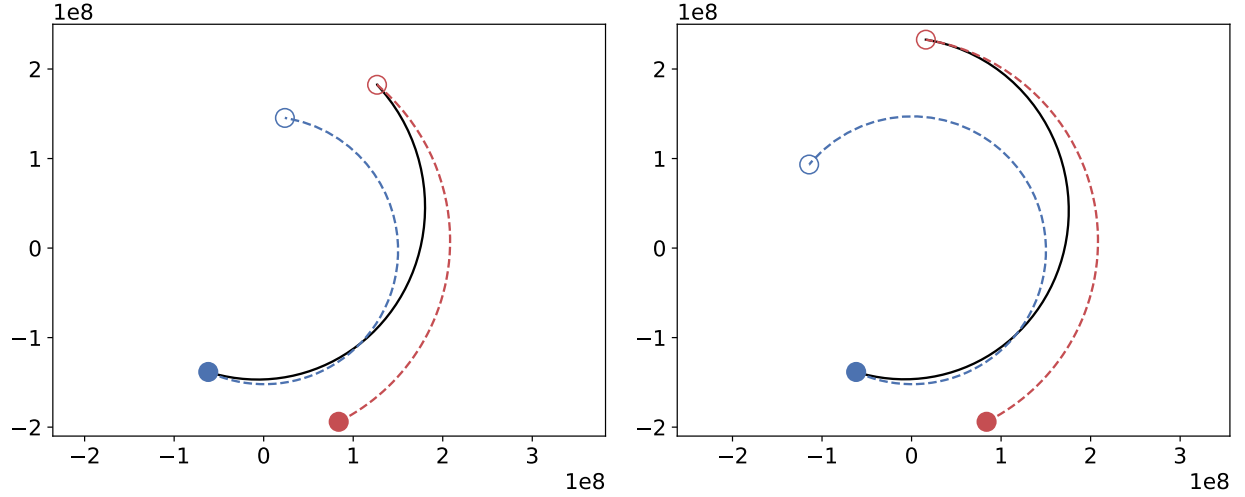


Fig. 7 Comparison of trajectory plots between the initial conditions (left column) and CCD-MR optimizations (right column) for the May 27th case.

Table 9 Engine-only optimization formulation. We maximize I_{sp} while fixing the thrust magnitudes (computed by the trajectory-only optimization).

	Function/Variable	Description	Quantity
Maximize	$\bar{I}_{sp}$	Average Specific Impulse (s)	1
w.r.t	M_R^{Earth}	Mixture ratio - TMI	1
	M_R^{Mars}	Mixture ratio - MOI	1
	P_c^{Earth}	Chamber pressure (MPa) - TMI	1
	P_c^{Mars}	Chamber pressure (MPa) - MOI	1
	A_t	Throat area (m ²)	1
	M_e^{Earth}	Exit Mach number - TMI	1
	M_e^{Mars}	Exit Mach number - MOI	1
	Total Design Variables		7
Subject to	$F_{Earth} = F_{Earth}^{trajectory}$	Thrust magnitude (N) - TMI	1
	$F_{Mars} = F_{Mars}^{trajectory}$	Thrust magnitude (N) - MOI	1
	$A_e^{Earth} = A_e^{Mars}$	TMI-MOI exit areas equal	1
	Total Constraints		3

Table 10 Comparison between the trajectory-and-engine decoupled and CCD optimization results. The CCD configuration outperforms the decoupled one.

Date	Decoupled (kg)	CCD-MR (kg)	Percent difference
May 27 th	4855	4014	19.0
July 27 th	3020	2462	20.4
September 8 th	6715	5705	16.3

Table 11 Engine-only optimization results for the three departure dates.

	Function/Variable	May 27 th	July 27 th	September 8 th
Objective	I_{sp} (s)	473.1	481.0	472.0
Constraints	F_{Earth} (N)	51200	29860	55240
	F_{Mars} (N)	15730	15180	13200
	$ A_e^{Earth} - A_e^{Mars} $	$< 10^{-8}$	$< 10^{-8}$	$< 10^{-8}$
Variables	M_R^{Earth}	4.225	4.267	4.219
	M_R^{Mars}	3.986	4.114	3.941
	P_c^{Earth} (MPa)	5.0	5.0	5.0
	P_c^{Mars} (MPa)	1.534	2.540	1.192
	A_t (m ²)	0.005318	0.003046	0.005752
	M_e^{Earth}	4.787	5.057	4.745
	M_e^{Mars}	4.770	5.049	4.727
	m^0 (kg)	5055	3230	6915
	A^e / A_t	103.6	156.0	98.0
	$\dot{m}_{Earth}$ (kg/s)	11.05	6.334	11.95
	$\dot{m}_{Mars}$ (kg/s)	3.385	3.215	2.846

times were fixed in the engine-only optimization.

$$m_b(kg) = \dot{m}_{Earth} T_b^{Earth} + \dot{m}_{Mars} T_b^{Mars} \quad (11)$$

The results of the decoupled optimizations can be seen in Table 10. The decoupled optimization results are better than the FEG case but slightly worse than the FEG-MR results previously shown for each date (see Tables 5, 6, and 7). Even though the FEG-MR case is not a full engine optimization, the coupling of the two components is more valuable than running the trajectory- and engine-only optimizations separately. As for the FEG cases, only the chamber pressures can be changed in the optimization, and more coupling is needed to overcome the two decoupled optimizations. Both CCD optimizations are far superior compared with decoupled optimizations. The final CCD-MR results are also shown in the table for comparison. The CCD-MR configuration is fully coupled and clearly benefits from the decoupled optimizations. Between the three dates, there is a fuel reduction between 16% to 20% by coupling the optimization using the CCD approach.

The main goal of this paper is to quantify the benefit of coupling the spacecraft trajectory control and onboard system design within the same optimization (i.e., CCD), and the results from Table 10 do this best. Before, the CCD optimizations were compared to a trajectory-only optimization that completely neglected the engine's design. The separate optimizations give a direct comparison because the design variables and constraints are the same between the CCD-MR and decoupled optimizations. The only difference is that the CCD configuration considers both trajectory and engine in the optimization, which leads to a clear benefit. In the decoupled optimization, the trajectory-only configuration did not know the required thrusts and engine size before the optimization started and had to use an oversized engine (RL-10A-4-2) with a relatively low I_{sp} . Although the engine parameters were then optimized in the engine-only configuration, the required thrusts and the corresponding engine fuel burns were much higher than the ones optimized by the CCD configuration. This is the main reason that the CCD configuration outperforms the decoupled one. To improve the performance of decoupled configuration, one could start the trajectory-only optimization by choosing a proper engine size with intuition or previous design experience. Alternatively, one would need to manually iterate between the trajectory- and engine-only optimizations. Both options are much more time-consuming than the automated CCD configuration.

Table 12 Control co-design multi-point optimization formulation.

	Function/Variable	Description	Quantity
Minimize	$(m_b^{\text{May}} + m_b^{\text{Jul}} + m_b^{\text{Sep}})/3$	Averaged fuel burned (kg) for the May, Jul, & Sep simulations	1
w.r.t.	M_R^{Earth}	Mixture ratio - TMI	3
	M_R^{Mars}	Mixture ratio - MOI	3
	P_c^{Earth}	Chamber pressure (MPa) - TMI	3
	P_c^{Mars}	Chamber pressure (MPa) - MOI	3
	A_t	Throat area (m ²)	1
	M_e^{Earth}	Exit Mach number - TMI	3
	M_e^{Mars}	Exit Mach number - MOI	3
	d_n^{Earth}	Thrust directions - TMI	9
	T_b^{Earth}	Burn time (s) - TMI	3
	d_n^{Mars}	Thrust directions - MOI	9
	T_b^{Mars}	Burn time (s) - MOI	3
	TOF	Time of flight (day)	3
	m^0	Fuel mass before flight (kg)	3
	Total Design Variables		49
Subject to	$R_{\text{mag}} \leq 3000$	Center of Mars to spacecraft (km)	3
	$ V_{x,y,z} \leq 0.01$	Spacecraft relative velocity (km/s)	9
	$m^{\text{excess}} \geq 200$	Forced excess fuel (kg)	3
	$A_e^{\text{Earth}} = A_e^{\text{Mars}}$	TMI-MOI exit areas equal	3
	$A_e^{\text{Earth}}(\text{May}) = A_e^{\text{Earth}}(\text{Jul})$	May-Jul exit areas equal	1
	$A_e^{\text{Earth}}(\text{May}) = A_e^{\text{Earth}}(\text{Sep})$	May-Sep exit areas equal	1
	Total Constraints		20

C. Control co-Design Multi-point Optimization

Each engine was optimized for a specific departure date in the previous optimizations. The optimized engine works well for the one departure date but may perform poorly for other dates. To alleviate this issue, we perform a multi-point optimization that combines the performance for all three departure dates. The goal is to create an engine that works well for various conditions.

The CCD multi-point optimization configuration is summarized in Table 12. There are many ways to formulate the multi-point optimization problem. We can use the average fuel burn from the May, July, and September flights as the objective function, letting the optimizer simultaneously minimize the fuel burn for all three departure dates. We can also use the maximal fuel burn among the three departure dates as the objective function, focusing on minimizing the maximal fuel burn and accommodating all possible departure dates. Note that the fuel burns from the three departure dates are significantly different, and September always has the largest fuel burn. Therefore, using the second option is equivalent to the single-point optimization (September). For this reason, we use the first option in this study. The design variables in Table 12 are similar to the CCD-MR configuration used in the single-point optimization cases (see Table 4), but each departure date now has its own set of design variables, e.g., mixtures ratios and chamber pressures. In other words, the design variables for the May case are independent of the ones for the July case. The exception is the engine throat area. All three departure dates share the same throat area because they will use the same engine geometry parameters. Like the design variables, all three departure dates must satisfy their constraints. To ensure that the exit areas are the same, we impose two additional exit area constraints to link the exit area between the May, July, and September cases. In addition, we use the engine mass computed from the September case's Earth burn as the engine mass for all three departure dates. We do this because the engine mass computed by the above burn is the largest. In total, we have 49 design variables and 20 constraints.

Table 13 Control co-design multi-point optimization results. The multi-point optimization creates an engine that balances the performance for the three departure dates.

	Function/Variable	May	July	September
Objective	m_b (kg)	4190	2692	5707
Constraints	R_{mag} (km)	3009	2873	3047
	$ V_{x,y,z} $ (km/s)	<0.01	<0.01	<0.01
	m^{excess} (kg)	200.0	200.0	200.0
	$ A_e^{\text{Earth}} - A_e^{\text{Mars}} $	< 10^{-8}	< 10^{-8}	< 10^{-8}
Variables	F_{Earth} (N)	35610	26602	45450
	F_{Mars} (N)	12071	9518	9563
	I_{sp} (s)	475.2	475.4	475.1
	A^e / A_t	113.8	113.8	113.8
	m^{engine} (kg)	136.9	136.9	136.9
	M_R^{Earth}	4.186	4.109	4.268
	M_R^{Mars}	3.972	3.931	3.932
	P_c^{Earth} (MPa)	3.919	2.927	5.000
	P_c^{Mars} (MPa)	1.326	1.045	1.050
	A_t (m ²)	0.004698	0.004698	0.004698
	M_e^{Earth}	4.844	4.844	4.841
	M_e^{Mars}	4.827	4.821	4.821
	d_{Earth}^1	0.5819	0.9242	0.5346
	d_{Earth}^2	-0.3243	0.3081	0.2591
	d_{Earth}^3	-0.7458	0.2258	0.8044
	T_b^{Earth} (s)	457.8	350.8	500.0
	d_{Mars}^1	0.9730	0.7779	0.8007
	d_{Mars}^2	-0.2308	0.4727	0.3393
	d_{Mars}^3	0.0093	-0.4139	0.4937
	T_b^{Mars} (s)	266.4	338.1	401.9
	TOF (d)	258.0	205.2	243.3
	m^0 (kg)	4390	2892	5907

The CCD multi-point optimization results are shown in Table 13. As expected, the results from the multi-point case are worse than the CCD-MR single-point optimizations for each date (please refer to the single-point fuel burn data in Tables 5, 6, and 7). Even though the results are worse when looking at a specific date, the overall engine performance is better. To justify this point, we perform one last optimization to evaluate the benefit of running multi-point optimization. As previously mentioned, the engine design for one departure date may perform poorly for another. This new verification optimization answers the problem by using the engine geometry optimized by one departure date and using it in the optimization of another date. For example, Table 5 shows the optimized engine throat area (A_t) and exit Mach numbers (M_e^{Planet}) for the May CCD-MR optimization. These three values were then fixed and used in a CCD optimization for the July and September departure dates. All other optimization setups are the same as the single-point optimization. We repeated the above for all departure date engine parameters.

The results from the multi-point verification optimizations can be seen in Table 14. The diagonal entries in the table are the CCD-MR fuel burns for the respective date. These entries are the most efficient because the engine was optimized specifically for that date. As previously discussed, every other entry used a fixed engine geometry from

Table 14 Multi-point verification optimization results for various engine designs. The single-point optimization engine designs may be infeasible for other departure dates.

Engine Design	Used in May 27 th	Used in July 27 th	Used in September 8 th
May 27 th	4014	2588	NA
July 27 th	NA	2462	NA
September 8 th	4208	2702	5734

another departure date. The main thing to point out is the NA terms in the table. These optimizations gave unfeasible solutions due to design variable limits being violated. The design variables being violated were the upper limit of the chamber pressures, namely P_c^{Earth} because the first burn in all optimizations tended to be larger than the second (P_c^{Mars}), requiring a higher chamber pressure. The upper limit of the chamber pressure was 5 (MPa) for each optimization. This value was chosen based on the design of the RL-10A-4-2 and other similar engines.

These high chamber pressures arise from the engine sizing. For example, only the September engine design was feasible using the September departure date. This is because the September departure date requires the largest thrust values out of any of the departure dates, and thus the single-point CCD-MR case created a relatively large engine. On the other end, the July departure date requires the smallest thrust values; thus, the single-point CCD-MR case created a relatively small engine. In order for the July engine to produce the thrust magnitudes required by the September departure date, the chamber pressures must be larger than allowed.

In essence, the single-point optimization made engines big enough for the specific date at hand and no bigger without violating design variable limits. That is why the September departure date could not use the smaller July or May optimized engines. The May departure date could use the bigger September engine but not the smaller July engine. Lastly, the July departure date could use every engine because the May and September engines were bigger than needed. These results show the need for multi-point optimization. Not only did some single-point optimized engines perform poorly for other dates, but some did not work at all. The multi-point optimization instead uses all three cases to balance the engine performance.

IV. Conclusion

This paper presents a control co-design (CCD) framework that simultaneously optimizes the spacecraft trajectory control and system design parameters for interplanetary missions. The CCD framework consists of an open-loop control component (interplanetary trajectory computation based on GMAT) and a physical system component (a rocket engine model). We then use OpenMDAO to couple the control and physical components for CCD optimization. The rocket engine component uses engine parameters, such as mixture ratio, chamber pressure, and throat area, as the inputs and computes the outputs, such as thrust and I_{sp} . These outputs are passed to the trajectory control component for the coupled optimization.

To quantify the benefit of the coupled optimization framework, we compare the results among trajectory-only, fixed engine geometry, decoupled, and CCD-coupled optimizations. We find that the coupled engine and trajectory (CCD) optimizations exhibit 22% to 28% more fuel burn reduction than the trajectory-only optimization with fixed engine parameters. We also observe that the CCD optimizations obtain 16% to 20% more fuel burn reduction than the traditional decoupled trajectory-engine optimizations. A multi-point optimization was also run to solve the potential drawbacks of single-point optimizations by creating an engine that balances the performance for multiple departure dates. Our results suggest that simultaneously optimizing the spacecraft engine and trajectory (CCD) has extra benefits, compared with fixed-engine and decoupled optimizations. This is because the CCD optimization can automatically find the best engine (minimal weight and maximal I_{sp}) tailored for the trajectory; no need to manually run trajectory and engine optimization multiple times.

In the future, we will further improve the CCD framework by including more spacecraft onboard systems. In addition, we will incorporate a global optimization algorithm into the CCD framework. This new feature will allow us to use discrete design variables (e.g., departure dates, fly-bys, and deep space maneuvers) and have a better chance of finding the global optimal solution.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Number 2223676.

References

- [1] Galski, R. L., De Sousa, F. L., Ramos, F. M., and Muraoka, I., "Spacecraft thermal design with the generalized extremal optimization algorithm," *Inverse Problems in Science and Engineering*, Vol. 15, No. 1, 2007, pp. 61–75. <https://doi.org/10.1080/17415970600573924>.
- [2] Frank, C. P., Pinon-Fischer, O. J., Mavris, D. N., and Tyl, C. M., "Design methodology for the performance, weight, and economic assessment of chemical rocket engines," *Journal of Aerospace Engineering*, Vol. 30, No. 1, 2017, p. 04016071. [https://doi.org/10.1061/\(ASCE\)AS.1943-5525.0000668](https://doi.org/10.1061/(ASCE)AS.1943-5525.0000668).
- [3] Varatharajoo, R., Kahle, R., and Fasoulas, S., "Approach for combining spacecraft attitude and thermal control systems," *Journal of Spacecraft and Rockets*, Vol. 40, No. 5, 2003, pp. 657–664. <https://doi.org/10.2514/2.6914>.
- [4] Hwang, J. T., Lee, D. Y., Cutler, J. W., and Martins, J. R., "Large-scale multidisciplinary optimization of a small satellite's design and operation," *Journal of Spacecraft and Rockets*, Vol. 51, No. 5, 2014, pp. 1648–1663. <https://doi.org/10.2514/1.A32751>.
- [5] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104. <https://doi.org/10.1007/s00158-019-02211-z>.
- [6] Darani, S. A., and Abdelkhalik, O., "Space trajectory optimization using hidden genes genetic algorithms," *Journal of Spacecraft and Rockets*, Vol. 55, No. 3, 2018, pp. 764–774. <https://doi.org/10.2514/1.A33994>.
- [7] Gad, A., and Abdelkhalik, O., "Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization," *Journal of Spacecraft and Rockets*, Vol. 48, No. 4, 2011, pp. 629–641. <https://doi.org/10.2514/1.52642>.
- [8] Izzo, D., Hennes, D., and Riccardi, A., "Constraint handling and multi-objective methods for the evolution of interplanetary trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2015, pp. 792–800. <https://doi.org/10.2514/1.G000619>.
- [9] Rauwolf, G. A., and Coverstone-Carroll, V. L., "Near-optimal low-thrust orbit transfers generated by a genetic algorithm," *Journal of Spacecraft and Rockets*, Vol. 33, No. 6, 1996, pp. 859–862. <https://doi.org/10.2514/3.26850>.
- [10] Goldberg, D. E., "Genetic algorithms in search, optimization, and machine learning," *Addison Wesley*, Vol. 1989, No. 102, 1989, p. 36.
- [11] Labroquère, J., Héritier, A., Riccardi, A., and Izzo, D., "Evolutionary Constrained Optimization for a Jupiter Capture," *International Conference on Parallel Problem Solving from Nature*, Springer, 2014, pp. 262–271. https://doi.org/10.1007/978-3-319-10762-2_26.
- [12] Ellithy, A., Abdelkhalik, O., and Englander, J., "Impact of Using Analytic Derivatives In Optimization For N-Impulse Orbit Transfer Problems," Springer Science and Business Media LLC, 2022, pp. 218–250. <https://doi.org/10.1007/s40295-022-00318-y>.
- [13] Garcia-Sanz, M., "Control Co-Design: an engineering game changer," *Advanced Control for Applications: Engineering and Industrial Systems*, Vol. 1, No. 1, 2019, p. e18. <https://doi.org/10.1002/ad2.18>.
- [14] Falck, R. D., Chin, J., Schnulo, S. L., Burt, J. M., and Gray, J. S., "Trajectory optimization of electric aircraft subject to subsystem thermal constraints," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017, p. 4002. <https://doi.org/10.2514/6.2017-4002>.
- [15] Hendricks, E. S., Falck, R. D., and Gray, J. S., "Simultaneous propulsion system and trajectory optimization," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017, p. 4435.
- [16] Cunis, T., Kolmanovsky, I. V., and Cesnik, C. E., "Control Co-Design Optimization: Integrating nonlinear controllability into a multidisciplinary design process," *AIAA Scitech 2022 Forum*, 2022, p. 2176. <https://doi.org/10.2514/6.2022-2176>.
- [17] Deshmukh, A. P., and Allison, J. T., "Unrestricted Wind Farm Layout Design With Optimal Control Considerations," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 58127, American Society of Mechanical Engineers, 2017, p. V02AT03A021. <https://doi.org/10.1115/DETC2017-67480>.

- [18] Pao, L. Y., Zalkind, D. S., Griffith, D. T., Chetan, M., Selig, M. S., Ananda, G. K., Bay, C. J., Stehly, T., and Loth, E., "Control co-design of 13 MW downwind two-bladed rotors to achieve 25% reduction in levelized cost of wind energy," *Annual Reviews in Control*, Vol. 51, 2021, pp. 331–343. <https://doi.org/10.1016/j.arcontrol.2021.02.001>.
- [19] Sundarrajan, A. K., Lee, Y. H., Allison, J. T., and Herber, D. R., "Open-loop control co-design of floating offshore wind turbines using linear parameter-varying models," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 85383, American Society of Mechanical Engineers, 2021, p. V03AT03A010. <https://doi.org/10.1115/DETC2021-67573>.
- [20] Allison, J. T., Guo, T., and Han, Z., "Co-design of an active suspension using simultaneous dynamic optimization," *Journal of Mechanical Design*, Vol. 136, No. 8, 2014. <https://doi.org/10.1115/1.4027335>.
- [21] Jiang, B., Amini, M. R., Liao, Y., Martins, J. R., and Sun, J., "Control Co-design of a Hydrokinetic Turbine with Open-loop Optimal Control," *arXiv preprint arXiv:2204.01134*, 2022. <https://doi.org/10.1115/OMAE2022-81006>.
- [22] Chilan, C. M., Herber, D. R., Nakka, Y. K., Chung, S.-J., Allison, J. T., Aldrich, J. B., and Alvarez-Salazar, O. S., "Co-design of strain-actuated solar arrays for spacecraft precision pointing and jitter reduction," *AIAA journal*, Vol. 55, No. 9, 2017, pp. 3180–3195. <https://doi.org/10.2514/1.J055748>.
- [23] Angeletti, F., Iannelli, P., and Gasbarri, P., "Automated nested Co-design framework for structural/control dynamics in flexible space systems," *Acta Astronautica*, 2022. <https://doi.org/10.1016/j.actaastro.2022.05.016>.
- [24] Ceccherini, S., Mani, K. V., and Topputo, F., "Combined System–Trajectory Design for Geostationary Orbit Platforms on Hybrid Transfer," *Journal of Spacecraft and Rockets*, Vol. 59, No. 2, 2022, pp. 448–466. <https://doi.org/10.2514/1.A35012>.
- [25] Ricciardi, L. A., Maddock, C. A., and Vasile, M., "Direct solution of multi-objective optimal control problems applied to spaceplane mission design," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 1, 2019, pp. 30–46. <https://doi.org/10.2514/1.G003839>.
- [26] Kluever, C. A., and Pierson, B. L., "Vehicle-and-trajectory optimization of nuclear electric spacecraft for lunar missions," *Journal of Spacecraft and Rockets*, Vol. 32, No. 1, 1995, pp. 126–132.
- [27] Koppel, R., "Optimal specific impulse of electric propulsion," *Second European Spacecraft Propulsion Conference, Proceedings of the conference held 27-29 May 1997 in Noordwijk, the Netherlands. Edited by Michael Perry. ESA SP-398. Paris: European Space Agency, 1997., p. 131*, Vol. 398, 1997, p. 131.
- [28] Arya, V., Taheri, E., and Junkins, J. L., "A composite framework for co-optimization of spacecraft trajectory and propulsion system," *Acta Astronautica*, Vol. 178, 2021, pp. 773–782. <https://doi.org/10.1016/j.actaastro.2020.10.007>.
- [29] Petukhov, V., and Wook, W. S., "Joint optimization of the trajectory and the main parameters of an electric propulsion system," *Procedia engineering*, Vol. 185, 2017, pp. 312–318. <https://doi.org/10.1016/j.proeng.2017.03.309>.
- [30] Shimane, Y., Preston, D., and Ho, K., "CONCURRENT OPTIMIZATION OF GRAVITY-ASSIST LOW-THRUST TRAJECTORY WITH POWER AND PROPULSION SUBSYSTEM SIZING," 2022.
- [31] Isaji, M., Takubo, Y., and Ho, K., "Multidisciplinary Design Optimization Approach to Integrated Space Mission Planning and Spacecraft Design," *Journal of Spacecraft and Rockets*, Vol. 59, No. 5, 2022, pp. 1660–1670. <https://doi.org/10.2514/1.A35284>.
- [32] Harris, G. W., He, P., and Abdelkhalike, O. O., "A Coupled Spacecraft System and Trajectory Optimization Framework using GMAT and OpenMDAO," *AAS/AIAA Astrodynamics Specialist Conference*, 2022.
- [33] Córdoba, I., Varando, G., Bielza, C., and Larrañaga, P., "On generating random Gaussian graphical models," *International Journal of Approximate Reasoning*, Vol. 125, 2020, pp. 240–250. <https://doi.org/10.1016/j.ijar.2020.07.007>.
- [34] Bouhlel, M. A., Hwang, J. T., Bartoli, N., Lafage, R., Morlier, J., and Martins, J. R. R. A., "A Python surrogate modeling framework with derivatives," *Advances in Engineering Software*, 2019, p. 102662. <https://doi.org/10.1016/j.advengsoft.2019.03.005>.
- [35] Ponomarenko, A., "RPA: Tool for Rocket Propulsion Analysis," *Space propulsion conference*, 2014.
- [36] Huzel, D. K., and Huang, D. H., "Design of liquid propellant rocket engines," Tech. rep., 1967.
- [37] Zandbergen, B., "Simple mass and size estimation relationships of pump fed rocket engines for launch vehicle conceptual design," *6th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2015.

- [38] Hwang, J. T., and Martins, J. R., "A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 44, No. 4, 2018, pp. 1–39. <https://doi.org/10.1145/3182393>.
- [39] Ranieri, C., "Path-constrained trajectory optimization for proximity operations," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008, p. 6275. <https://doi.org/10.2514/6.2008-6275>.
- [40] Weiss, A., Petersen, C., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Safe positively invariant sets for spacecraft obstacle avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2015, pp. 720–732. <https://doi.org/10.2514/1.G000115>.
- [41] Lambe, A. B., and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, 2012, pp. 273–284. <https://doi.org/10.1007/s00158-012-0763-y>.
- [42] Giles, M., and Pierce, N. A., "An introduction to the adjoint approach to design," 2000.
- [43] Peter, J. E., and Dwight, R. P., "Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches," *Computers & Fluids*, Vol. 39, No. 3, 2010, pp. 373–391.
- [44] Kenway, G. K., Mader, C. A., He, P., and Martins, J. R., "Effective adjoint approaches for computational fluid dynamics," *Progress in Aerospace Sciences*, Vol. 110, 2019, p. 100542.
- [45] Conte, D., and Spencer, D., "Targeting the Martian moons via direct insertion into Mars' orbit," *AIAA/AAS Astrodynamics Specialist Conference*, Vol. 156, 2015, pp. 2389–2406.
- [46] Santiago, J. R., "Evolution of Pratt and Whitney's cryogenic rocket engine RL-10," 1996. URL https://www.b14643.de/Spacerockets/Specials/P&W_RL10_engine/index.htm.
- [47] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM review*, Vol. 47, No. 1, 2005, pp. 99–131. <https://doi.org/10.1137/S1052623499350013>.