18th International Conference on Space Operations, Montreal, Canada, 26–30 May 2025.

SpaceOps-2025, ID # 163

Optimizing Low-Thrust Trajectories for Multiple Missed Thrust Events

Frank Laipert[1,*]

[1]*Nabla Zero Labs, South Pasadena, CA, USA, frank.laipert@nablazerolabs.com*
[*]Corresponding Author

Abstract

A method is presented to design trajectories accounting for unplanned thrust outages, often called missed thrust events. This problem lies at the intersection between trajectory optimization and mission operations. These thrust outages result from routine and unavoidable anomalies that send a spacecraft into safe mode. While missed thrust events cannot be predicted, we do have statistical models that describes how likely they are to occur, which indicate that missions of significant length are likely to experience multiple MTEs. The problem is formulated as an optimization problem with a probabilistic objective function. The objective function is transformed such that it can be estimated from a parametric missed thrust simulation considering multiple events combined with the statistical probability of each number of events occurring. The method is demonstrated on an example Earth-to-Mars low-thrust transfer, and result are presented showing a statistical trajectory metric as a function of a single reference trajectory constraint.

1. Introduction

Nabla Zero Labs has begun the development of a prototype of Missfit, a software tool for designing trajectories that accounts for unplanned thrust outages in low-thrust electric propulsion (EP) missions. EP is playing an increasingly important role in space exploration and commercial development. From scientific missions such as Psyche and the Earth Return Orbiter of the Mars Sample Return campaign to cargo delivery for humans on the Moon, EP can enable access to destinations requiring very high ΔV to reach and allow for very high delivered payload masses.

However, the trajectory design process for EP missions is more complex than for their chemical counterparts. EP mission operations are also more challenging because the propulsion system needs to operate for a large portion of the mission. Our proposed method will help with the design of trajectories that account for unplanned thrust outages, often called missed thrust events (MTE). This problem lies at the intersection between trajectory optimization and mission operations.

These thrust outages result from routine and unavoidable anomalies that send a spacecraft into safe mode. While missed thrust events cannot be predicted, we do have a statistical model that describes how likely they are to occur[2]. In contrast to other sources of uncertainty for a mission, like orbit determination error or maneuver execution error, the uncertainty stemming from missed thrust cannot be quantified in terms of a normal distribution. Instead, it is a random event-driven process, which requires a different approach to analyze. Our goal is to incorporate this model directly into the design process such that a given trajectory can be improved to meet a resiliency requirement for missed thrust.

Designing missed-thrust resilient trajectories is of critical importance for widespread adoption of low-thrust electric propulsion for cislunar and interplanetary missions. For scientific missions, which have a low risk tolerance, mission designers need to show that the spacecraft will reach its destination with a high degree of confidence. Chemical missions typically require enough propellant for a 99% worst case in navigation and maneuver execution error. We believe the same standard should apply to EP missions, and that the standard must include errors from missed thrust, which tend to dominate over navigation errors in these missions.

To account for MTEs, it's important to consider scenarios where more than one MTE occurs. Using the data-derived statistical model presented in [2], we can run a simple analysis to compute the probability that there will be more than N events in a mission as a function of the nominal mission time-of-flight (TOF). Figure 1 represents these probabilities for flight times up to eight years. We can see that for flight times of around 1 year, similar to the Earth-to-Mars transfer time of the Mars Sample Return Earth Return Orbiter, there is roughly a 50% chance that more than one MTE will occur, and about a 10% chance that there will be more than 3. For longer missions, it's in fact exceedingly unlikely that only one MTE will happen.

It's also worth noting that a four-day missed thrust event occurred on the Dawn mission during a critical point on its approach to Ceres[1]. Fortunately, the mission designers were able to re-design the capture sequence at Ceres, but

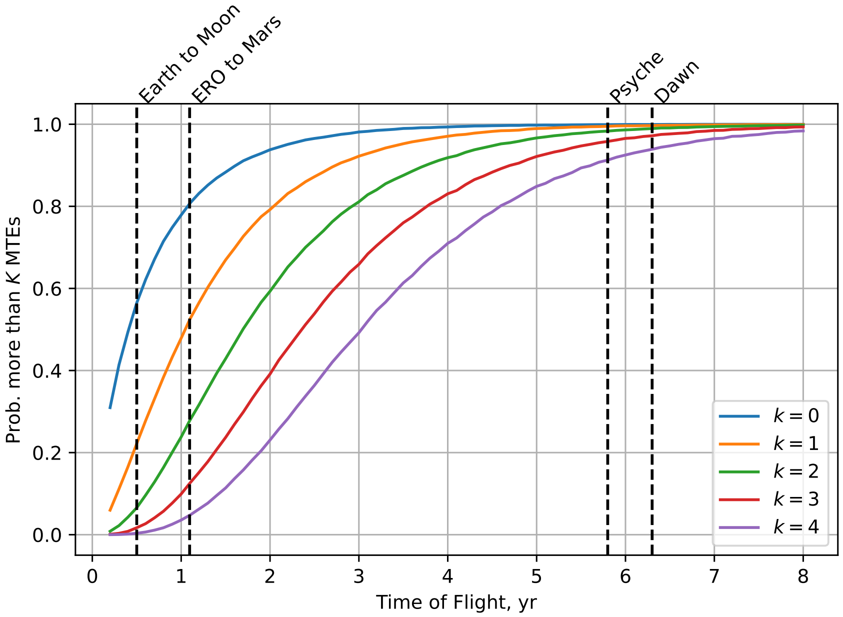

Figure 1: The probability that a mission will experience more than a given number of MTEs is computed for a range of nominal TOFs and numbers of MTEs.

not without a 26-day delay to arrival at the first science orbit. Because of the low relative velocity between Dawn and Ceres leading up to capture, the mission was not at high risk of failure. But for missions with trajectories that have higher velocity relative to their targets, which may be the case for a flyby or a hybrid mission with a high-thrust capture maneuver (as is the case for Mars Sample Return), it is possible that an ill-timed MTE could result in a mission failure.

2. Methods

We will develop a trajectory optimization application, called Missfit, that allows the user to add probabilistic missed thrust constraints to the mission design process. This will be the first low-thrust trajectory design tool to offer this capability. It will give a mission designer the ability to set a probability of success and see the cost to the mission in terms of propellant or time. Equivalently, they may optimize a probabilistic function, such as the 99th percentile propellant or time-of-flight. This approach directly targets mission risk—a topic of great importance to customers and review boards. Demonstrating that the risks to a mission are well-characterized and managed will help a project as it moves through its design phases, ultimately saving time and cost.

To accomplish our goal, we propose a separation that allows us to evaluate a probabilistic function of the trajectory with a series of deterministic optimizations combined with a weighting obtained from the statistics.

Let X be a random variable on a low-thrust trajectory as it is executed on a spacecraft prone to random shutdowns. As an example, X may be the final mass or the time-of-flight. We wish to incorporate a statistical operation of this variable in the optimization so that we may optimize or constrain it. Suppose our objective is to minimize the expected value of X:

$$\min J = E[X] \tag{1}$$

One way to compute $E[X]$ would be with a Monte Carlo simulation as outlined in [3], covering both the number of missed thrust events in a mission and their duration. However, a full Monte Carlo would be quite expensive to compute as part of a larger optimization algorithm. To help make the problem more tractable, we can re-write Equation 1 into a form that separates the probabilities of different combinations of MTEs. We will also make some simplifying assumptions:

1. Missed thrust events are equally likely at all times in the mission. The Weibull distribution in Imken et al.[2] is close to a pure exponential distribution, which supports this assumption.

2. All events are of a fixed duration.

Now, using K to represent the number of MTEs that occur during a mission,

$$E[X] = E[X|K=0]P(K=0) + E[X|K=1]P(K=1) + ... = \sum_{k=0}^{\infty} E[X]_k P_k \tag{2}$$

where $E[X]_k = E[X|K=k]$ is the expected value of X given k missed thrust events have occurred and $P_k = P(K=k)$ is the probability that k events have occurred.

In other words, we can separate the expectation of X into components corresponding to a specific number of missed thrust events occurring on the trajectory. Note that the first term in Equation 2, where no missed thrust events have occurred, is simply the value of that variable from the reference trajectory.

With this transformation in place, we can estimate each term in Equation 2 separately in a simple parametric fashion. For example, $E[X|K=1]$ can be estimated by simulating a single MTE at a regular time step along the reference trajectory, re-optimizing the trajectory after recovery from each event, and taking the average of X for each case. In the case of missed thrust events, P_1 can be computed ahead of time based only on the thruster on-time in the reference trajectory. For the next term, $E[X|K=2]$ can be estimated by repeating the process, treating each resulting trajectory from the first step as a reference trajectory in the next step. Figure 2 shows a sketch of the process, with only one event considered for clarity.

While it would take infinitely many terms to compute an exact value for $E[X]$, in general the value of P_k will decrease as k grows larger. The series can therefore be truncated after a few terms, letting the user decide how many events to consider in the computation and how fine a grid to use for each parametric run. In the case of one event, this method would be equivalent to the single-event methods developed by Venigalla et al.[5].

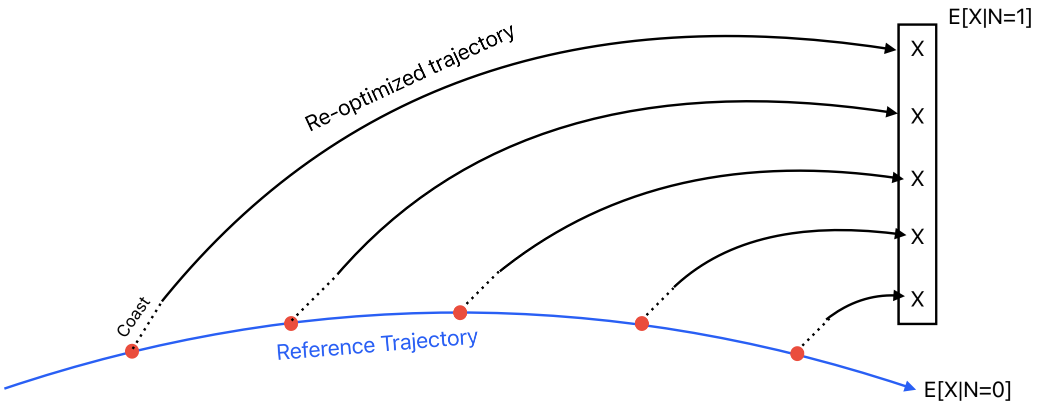

Figure 2: In this schematic, five missed thrust events are simulated evenly throughout the trajectory to compute $E[X|N \leq 1]$. Coasts are placed at regular intervals in the trajectory, and the trajectory is re-optimized for the remainder of the mission. This process can be repeated with each of the re-optimized trajectories serving as the reference trajectory for a second MTE, and so on.

Missfit will be structured as a two-stage algorithm with an inner loop and an outer loop. The goal of the inner loop will be to compute a probabilistic function as fast as possible, and the goal of the outer loop is to improve the reference with as few inner loop runs as possible.

2.1 Inner Loop

For the inner loop, the trajectory is divided into n segments. A missed thrust event is simulated on each segment by inserting a coast arc, and the trajectory is re-optimized from the end of the coast to the target. The process is repeated on each of the re-optimized trajectories for the second event, and so on for the remaining events. All missed thrust events will be of the same duration to simplify the problem. We can calculate the number of trajectory optimizations required to evaluate the effect of k missed thrust events on a trajectory divided into n segments with the binomial theorem.

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} \tag{3}$$

In Figure 3 we have plotted the number of trajectory optimizations for up to $k = 4$ events and $n = 50, 100, 150, 200$ trajectory segments. This calculation assumes that the segment duration is held constant for re-optimizations, rather than the number of segments. For example, if the original trajectory has 50 segments, the re-optimized trajectory following a missed thrust event on segment 10 would have 40 segments. Clearly, the number of re-optimizations increases quickly with n and k, and Missfit will need to perform this analysis several times over a run.

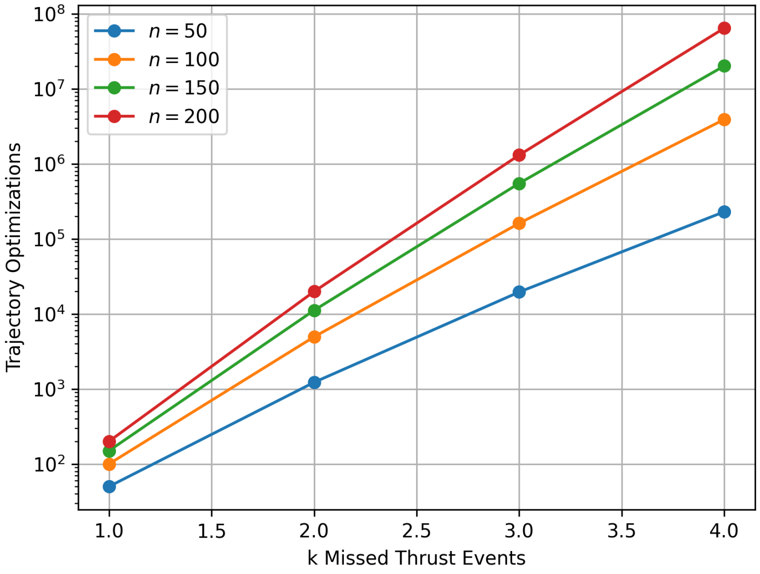

Figure 3: The number of trajectory optimizations scales up with trajectory segments and number of missed thrust events.

We can take advantage of parallelism to speed up the inner loop. In an example trajectory divided into 50 segments, all 50 of the initial re-optimizations can be run simultaneously, resulting in 1225 segments for the second missed thrust event. These second-event re-optimizations could also be run in parallel, yielding 19,600 segments for the third event, and 230,300 for the fourth event. Currently, Amazon Web Services offers a computing instance through their EC2 service with 128 cores (c6i.32xlarge), and several of these can be combined into a virtual computing cluster. If we combine four of them for 512 parallel processes, and we assume an average compute time of five seconds for each re-optimization (reasonable considering the post missed thrust state is close to optimal already), we can get through the inner loop in about forty minutes. This estimate is idealized in that a direct 1:1 speed up per parallel process is rarely achieved, but is a reasonable sanity check on the inner loop compute time.

A 512-core computing cluster is representative of the computing resources commonly found on-premises at large institutions or corporations. However, in the cloud computing paradigm, the choice of EC2 instances could easily be scaled up to the size of the problem being considered. The difference in problem size is not computing time, but computing cost in dollars. The current cost for the c6i instance type on AWS GovCloud is 5.1¢ per core per hour. For the case of four events on a trajectory divided into 100 segments, an AWS user under the default account limits could request 20 c6i.32xlarge instances with a total of 2560 cores. The function evaluation would take 2.22 hours costing a total of about $290. This financial cost is the same whether we use a lot of time with few cores or less time with many cores, so there is no reason to not use as many cores as possible. The default account limit of 20 EC2 instances can be lifted by the user upon request to AWS, so even very large problems can be solved in a reasonable amount of time. The user must only decide if it is worth the cost to the mission.

Table 1: Estimated computing cost for the inner loop function evaluation using AWS GovCloud c6i instances, assuming five seconds per trajectory re-optimization.

n	$k = 3$	$k = 4$	$k = 5$
25	\$0.54	\$1.45	\$5.26
50	\$1.81	\$18	\$168
100	\$12	\$290	\$5623

Table 1 contains an estimate of the computing cost with the above assumptions for a range of n trajectory segments and k missed thrust events. Very long trajectories will likely be expensive to thoroughly investigate as a whole because they will require both many segments and many events. However, this tool will still be valuable to mission designers in these cases. These long trajectories may be broken up into smaller phases at logical points, such as the Mars flyby in both the Psyche and Dawn trajectories, and the Vesta-Ceres transfer in the Dawn trajectory. We can see from 1 that analyzing a trajectory as two separate phases divided into 50 segments each will be much cheaper than analyzing it as 100 segments together.

It will be up to the user to determine an appropriate number of segments and outage duration to use for the analysis. A few preliminary runs considering only a single outage may help determine the resolution necessary to capture the dynamics in the trajectory before committing to a run with more events. A representative missed thrust duration can be determined considering the time between ground contacts, the time to implement a new thrust profile, and the time to diagnose the problem and bring the spacecraft out of safe mode.

2.2 Outer Loop

Because of the computational expense, we will limit the method to improving a subset of parameters on the reference trajectory rather than expose the entire set of trajectory design variables, which can reach into the hundreds. These parameters are often constraints on the reference which act as "resources" to help with recovery from missed thrust. As an example, consider a mission like the MSR Earth Return Orbiter. The Earth-to-Mars trajectory features a forced coast arc prior to the Mars orbit insertion burn. This forced coast period, which may be on the order of 60–90 days long, is a resource that improves resiliency to missed thrust because it may be incrementally shortened when re-optimizing the trajectory after a missed thrust. Other resources may be extra propellant, a duty cycle margin, or a duty cycle ramp-down profile prior to rendezvous. The outer loop optimization of Missfit will focus on these resources, using a method appropriate for a small number of design variables without derivatives. A secant method, for example, would require three function evaluations per iteration to approximate the first and second derivatives of a scalar function. On the inner loop, each trajectory re-optimization will be formulated to maximize the resource in question. In the ERO example, the re-optimizations would maximize the forced coast duration.

We expect Missfit to be of most value in the middle phases of mission development: after the initial formulation, but early enough that the spacecraft system design (which for EP missions is coupled with the trajectory design) can still be modified. For instance, starting from an initial trajectory resulting from a broad search of the design space, Missfit may be used to further refine parameters such as forced coast duration, propellant margin, or solar array power, and ensure that they are adequate to meet the mission risk tolerance. If it is found that the spacecraft needs more power to recover from missed thrust, or cannot deliver the intended payload mass, then discovering these issues sooner will save overall time and cost compared to discovering them after hardware has been procured or fabricated. Even if a run takes several days to complete, this method may be worth the computational effort.

2.3 Example Implementation

We consider an example trajectory similar to one for the Earth Return Orbiter of the Mars Sample Return campaign. The details of this trajectory are summarized in Table 2. More information about the ERO mission can be found in [4]. For the spacecraft power system, a $1/r^2$ model is used to vary power as a function of distance to the Sun, and a bounded linear relationship between thrust and available power is assumed at

$$T = 0.245 + 1.2 \times 10^{-5} P_a,$$

where P_a is power available in Watts and thrust is calculated in Newtons. P_a is bounded between 10 kW (below which thrust is zero) and 42 kW (above which thrust does not increase). Specific impulse is assumed to be a constant 4000 sec.

Table 2: Summary of trajectory and propulsion system for ERO example mission.

Earth Departure	30-SEP-2026
Initial Mass	6000 kg
Departure v_∞	2.2 km/sec
Mars Arrival	03-OCT-2027
Arrival v_∞	1.75 km/sec
Time of Flight	368 days
Power (1 AU)	41 kW
Thrust (1 AU)	0.737 N
Isp	4000 sec

The ERO trajectory has two coast constraints imposed on it. The first is a fixed 30-day checkout period following launch without thrust. The second is a forced coast prior to Mars arrival, called a terminal coast. One of the functions of the terminal coast is to serve as a resource that mission operators can use to recover following a missed thrust event. If a missed thrust occurs, the terminal coast may be shortened in a re-optimization of the trajectory-to-go following recovery from the event. While it increases robustness to missed thrust events, a longer terminal coast comes at a cost of reduced delivered mass to Mars. For this example trajectory, our goal is to compute a statistical trajectory metric as a function of this terminal allowing us to optimize this variable.

2.4 Missed Thrust Analysis

We can separate the missed thrust analysis into two parts: the simulation of an individual missed thrust event, and the collection of many simulations into a larger missed-thrust analysis.

2.4.1 Missed Thrust Event Simulation

These individual re-optimizations are formulated as follows:

1. At a given epoch, a child trajectory is generated which inherits the initial state from the parent trajectory.
2. A new optimization problem is constructed that fixes the inherited initial state (position, velocity, and mass) and has three legs: a fixed initial coast leg representing the missed thrust, a thrust leg, and a terminal coast leg. The final state at the end of the third leg is constrained to arrive at the target body with the same relative velocity magnitude. The arrival velocity direction and date of arrival may vary. Continuity constraints are enforced between legs.
3. The optimization problem is solved to maximize a dual objective function that combines the terminal coast duration and final mass.

The objective function for the missed thrust optimizations is

$$f(x) = \Delta\tau^2 + w\bar{m}^2, \tag{4}$$

where $\Delta\tau$ is the terminal coast duration normalized by the inherited terminal coast duration, $\bar{m}$ is the final mass normalized by the inherited final mass, and w is a user-specified weight. Maximizing Eqn. 4 balances the need to deliver sufficient initial mass while preserving the terminal coast for potential future missed thrust events.

2.4.2 Full Missed Thrust Analysis

To perform the full missed thrust analysis on a given reference trajectory, the individual missed-thrust event simulation described above is deployed systematically and recursively across the reference trajectory. The procedure for this is:

1. Starting with the reference trajectory, missed thrust simulations are performed at a user-defined cadence, e.g. every 20 days. We use the term *parent* for the reference trajectory, and *child* for the missed thrust simulation trajectories.
2. Each child trajectory is re-converged and saved. This completes a single-event analysis.
3. For the second event, we return to Step 1, with each child becoming a parent for the second-event trajectories.

4. Repeat the process for the number of missed thrust events to consider.
5. Once all events have been simulated, we group all trajectories according to the number of events they contain and compute some metric from each population, e.g. the 10th percentile low final mass.
6. We then combine the metric from each population, weighted by the probability of the number of events in that trajectory.

It is up to the user to decide the metric to compute. In our example, we use the 10th percentile low final mass, with the goal of finding the reference terminal coast duration which maximizes this metric. For a three-event simulation, the 10th percentile low final mass is computed with

$$m_{f10} = p_0 m_{f,0} + p_1 m_{f10,1} + p_2 m_{f10,2} + p_3 m_{f10,3} \tag{5}$$

where p_n is the probability of n missed thrust events, $m_{f10,n}$ is the 10th percentile low final mass given n missed thrust events, and $m_{f,0}$ is the reference final mass. This method could be applied to other metrics, and other variables could be solved for. Among the important parameters that effect the analysis are the time between events, the number of events to consider, and the weighting in Eqn. 4. Decreasing the time between events and increasing the number of events to consider will increase the fidelity at the cost of longer compute times. Selecting a weighting which favors mass over remaining coast duration will yield higher delivered mass but may result in trajectories with too little coasting prior to arrival.

Taken together, the process described here represents a single function evaluation, and many evaluations will generally be required to compute the optimal value for the design variable.

3. Results

For the example mission, we analyze reference terminal coast values ranging from 40 days to 50 days, in steps of one day. All missed thrust events in the simulation are 4 days in duration. Table 3 contains the probabilities that a given number of missed thrust events occurs on the example trajectory. Since we only analyze up to 3 events, the 10.1% chance of more than three events represents a risk in the analysis. The risk can be mitigated by considering more events.

Table 3: Probability associated with each number of missed thrust events for the example Earth-to-Mars trajectory.

Events	0	1	2	3	>3
Probability	0.222	0.296	0.247	0.134	0.101

Results are shown in Fig. 4, where the orange dots represent the final mass of the reference trajectory, the blue dots represent the median final mass with missed thrust taken into consideration, and the vertical bars showing the 90th percentile high and 10th percentile low final masses. The bottom of each vertical bar is the value we are interested in maximizing. The results are from a 3-event analysis with missed thrust events simulated every 20 days along the trajectory.

We can see that with a terminal coast duration of 40 days, the reference trajectory final mass is highest. If a mission designer were to only examine the reference trajectory, they may select that value. However, a value of 45 days maximizes the 10th percentile low final mass when missed thrust is included in the analysis. For values above 45 days, the 10th percentile final mass begins to decline again as the benefits of the longer terminal coast are overcome by the overall lower final mass in the reference. It may seem odd that the 90th percentile values are higher than the reference values, but this is possible because the missed thrust simulations may reduce the terminal coast, which can help deliver an increased final mass and make up for the penalty from the missed thrust event. Changing the weighting between final mass and coast duration in the objective function may lead to a different result.

In Fig. 5 we have results from the same analysis as Fig. 4, however only showing the data obtained from simulating a single missed thrust event. We can see in this case that the optimum coast is 40 days, rather than the 45 days obtained by considering three events. This contrast shows the value in considering multiple events in the trajectory design.

While the overall scale on the vertical axis of Figs. 4 and 5 may not be dramatic relative to the absolute final mass values, these results demonstrate the phenomenon that the optimal trajectory will differ when missed thrust is considered. The specific results will depend strongly on the particular trajectory.

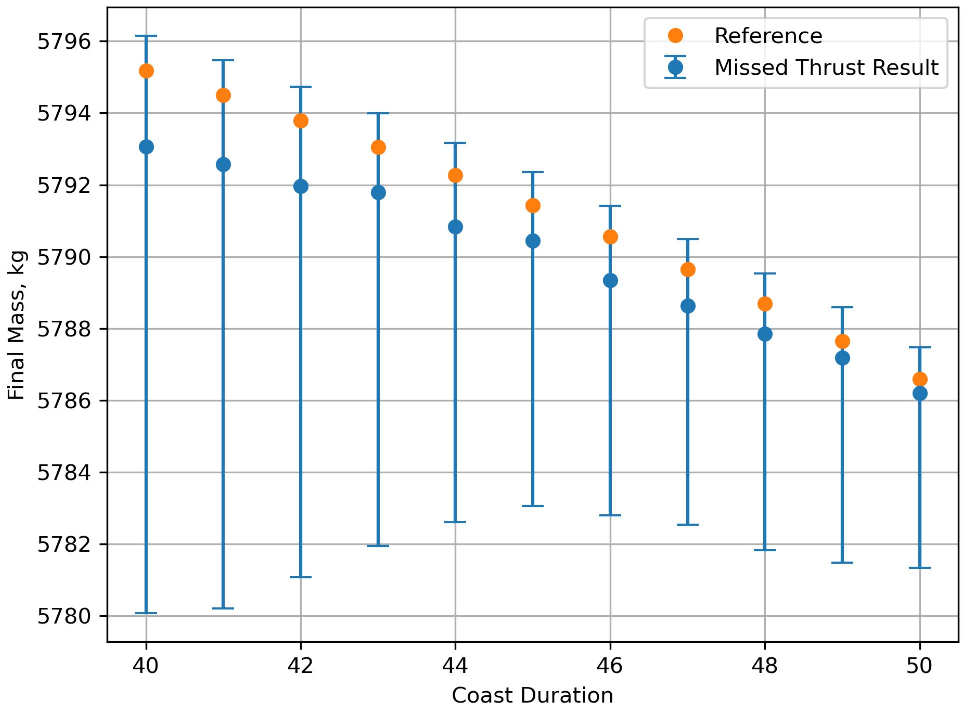

Figure 4: Final mass results for a 3-event analysis as a function of reference terminal coast duration. Vertical bars represent the 10%-90% probability, blue dots represent the median, and orange dots represent the reference value.

3.0.1 Computing Infrastructure

For the example trajectory, Missfit is configured to run with eight processes in parallel. Each stage is optimized in parallel before moving on to the next. So the reference trajectory is designed first, then all of the single-event simulations are run in parallel until all are complete, then the two-event trajectories are run in parallel, and so on. With this configuration, Missfit takes an average of 124 seconds for one objective function computation. The computer used for the analysis has an 8-core Intel Core i9-9900 CPU and 32 GB of RAM. Results are generated using MONTE version 168.1 and MColl uses IPOPT to solve the missed thrust event optimization problems.

3.1 Discussion

A significant challenge in the development of Missfit is the reliability of the inner-loop trajectory optimizer, which in this case is the MColl application included with MONTE. While MColl can perform well when guided by the user on an individual trajectory, its convergence can be highly sensitive and unpredictable, leading to problems when it is employed automatically over hundreds or thousands of initial conditions. In particular, MColl seemed to struggle with variable flight times, often crashing with confusing errors. It is important to allow arrival dates to vary to avoid over-constraining the problem, so this issue is difficult to work around.

For the results shown in Fig. 4, greater than 90% convergence is achieved at a minimum for each terminal coast duration analyzed, with most cases around 98%. However, these convergence values dropped significantly for terminal coasts outside the range presented. Because of these convergence issues, we cannot yet employ our algorithm in an outer-loop method to solve for the optimal terminal coast, and have instead chosen to perform a more controlled parametric optimization search. Still, the parametric search shows the overall value and feasibility of the approach. Future development will focus on creating an inner-loop optimization method which is tailored to the specific problem of re-optimizing trajectories after a missed thrust.

4. Conclusion

In this study, we have developed a framework for analyzing and optimizing a low-thrust trajectory accounting for multiple missed thrust events. The theoretical framework is implemented in a program called Missfit and demonstrated

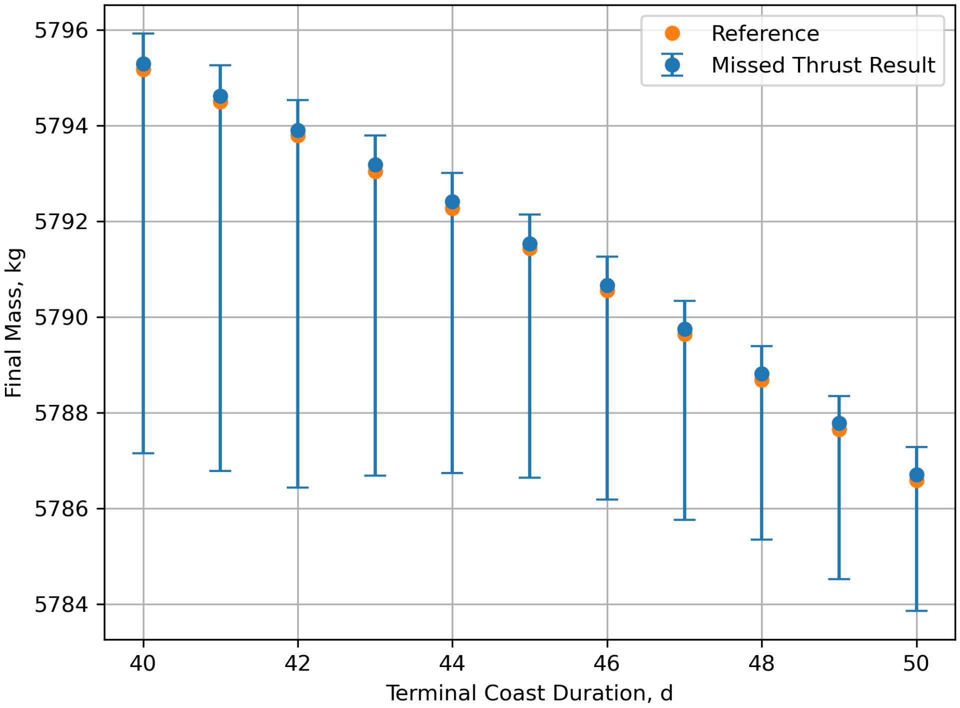

Figure 5: Final mass results for a 1-event analysis as a function of reference terminal coast duration. Vertical bars represent the 10%-90% probability, blue dots represent the median, and orange dots represent the reference value.

on an example Earth-to-Mars transfer. For the example trajectory, we have shown that the technique can be used to compute a probabilistic function—in this case the 10th percentile delivered mass at Mars arrival. To our knowledge, this represents the first-ever missed thrust optimization which considers more than a single missed thrust event occurring during a mission.

Future development will focus on improving the robustness of the inner-loop optimization method. Doing so will allow us to employ this technique with a simple scalar optimization for the outer loop. Better convergence can be achieved with an optimizer developed specifically for this type of problem. By their nature, missed thrust simulations start from an initial state which is close to optimal, so an indirect shooting method may be a good choice for this use case.

5. References

[1] Daniel Grebow, Gregory J Whiffen, Dongsuk Han, and Brian Kennedy. Dawn safing approach to Ceres re-design. In *AIAA/AAS Astrodynamics Specialist Conference*, page 5426, 2016.

[2] Travis Imken, Thomas Randolph, Michael DiNicola, and Austin Nicholas. Modeling spacecraft safe mode events. In *2018 IEEE Aerospace Conference*, pages 1–13, 2018.

[3] Frank Laipert and Travis Imken. A Monte Carlo approach to measuring trajectory performance subject to missed thrust. In *AIAA/AAS Space Flight Mechanics Meeting*, Kissimmee, FL, January 2018.

[4] O. Sutherland, S. Vijendran, J. Huesing, K. Geelen, D. Feili, J.M. Sanchez-Perez, A. Nicholas, and R. Lock. Mars Sample Return - Earth Return Orbiter: ESA's next Interplanetary Electric Propulsion Mission Concept. In *36th International Electric Propulsion Conference*, Vienna, Austria, September 2019. IEPC-2019-A-927.

[5] Chandrakanth Venigalla, Jacob A Englander, and Daniel J Scheeres. Multi-objective low-thrust trajectory optimization with robustness to missed thrust events. *Journal of Guidance, Control, and Dynamics*, 45(7):1–14, 2022.