Automation of POST Cases via External Optimizer and "Artificial p2" Calculation

Patrick D. Dees[1] and Mathew R. Zwack[2]

Jacobs ESSSA Group, Huntsville, AL 35806, United States

During early conceptual design of complex systems, speed and accuracy are often at odds with one another. While many characteristics of the design are fluctuating rapidly during this phase there is nonetheless a need to acquire accurate data from which to down-select designs [1] as these decisions will have a large impact upon program life-cycle cost [2]. Therefore enabling the conceptual designer to produce accurate data in a timely manner is tantamount to program viability.

For conceptual design of launch vehicles, trajectory analysis and optimization is a large hurdle. Tools such as the industry standard Program to Optimize Simulated Trajectories (POST) have traditionally required an expert in the loop for setting up inputs, running the program, and analyzing the output. The solution space for trajectory analysis is in general non-linear and multi-modal [3] requiring an experienced analyst to weed out sub-optimal designs in pursuit of the global optimum. While an experienced analyst presented with a vehicle similar to one which they have already worked on can likely produce optimal performance figures in a timely manner, as soon as the "experienced" or "similar" adjectives are invalid the process can become lengthy. In addition, an experienced analyst working on a similar vehicle may go into the analysis with preconceived ideas about what the vehicle's trajectory should look like which can result in sub-optimal performance being recorded. Thus, in any case but the ideal either time or accuracy can be sacrificed.

In the authors' previous work [4] a tool called multiPOST was created which captures the heuristics of a human analyst over the process of executing trajectory analysis with POST. However without the instincts of a human in the loop, this method relied upon Monte Carlo simultation to find successful trajectories. Overall the method has mixed results, and in the context of optimizing multiple vehicles it is inefficient in comparison to the method presented in [5].

POST's internal optimizer functions like any other gradient-based optimizer. It has a specified variable to optimize whose value is represented as *optval*, a set of dependent constraints to meet with associated forms and tolerances whose value is represented as *p2*, and a set of independent variables known as the *u-vector* to modify in pursuit of optimality. Each of these quantities are calculated or manipulated at a certain phase within the trajectory. The optimizer is further constrained by the requirement that the input u-vector must result in a trajectory which proceeds through each of the prescribed events in the input file. For example, if the input u-vector causes the vehicle to crash before it can achieve the orbital parameters required for a parking orbit, then the run will fail without engaging the optimizer, and a p2 value of exactly zero is returned. This poses a problem, as this "non-connecting" region of the u-vector space is far larger than the "connecting" region which returns a non-zero value of p2 and can be worked on by the internal optimizer. Finding this connecting region and more specifically the global optimum within this region has traditionally required the use of an expert analyst.

Several attempts have been made to mitigate the problem of finding the connecting space and the contained global optimum. Some have used external optimizers wrapped around POST, always with varying levels of success. The main issue with this approach is the large non-connecting space which

---

[1] Trajectory Engineer, Advanced Concepts Office, George C. Marshall Space Flight Center/ED04, Huntsville, AL, AIAA Member.

[2] Systems Engineer, Advanced Concepts Office, George C. Marshall Space Flight Center/ED04, Huntsville, AL, AIAA Member.

always returns a *p2* value of exactly zero. Any optimizer wrapped around POST needs an objective function value to work on, and so finding a non-zero starting point for an external optimizer returns to the initial problem of finding the connecting region which requires returning to Monte Carlo analysis. At that point, the internal optimizer will work faster than an external one.

In this work, a workaround to this problem is presented. At every point in the u-vector space, there is some value of *p2* which can be calculated using the dependent constraint variables, however not necessarily at its prescribed phase. By gathering the closest numbers to the native *p2* calculation, an "artificial" *p2* can be calculated by using whatever is available. In the example given above, if a crashing vehicle's insertion into a parking orbit is prescribed by an altitude and velocity, those quantities can be gathered from the final time step computed by POST and used as a stand-in for the native *p2*. In this manner, any arbitrary u-vector passed to POST can have a non-zero *p2* for use with an external optimizer.

The focus of this work is in the application of external optimizers to POST and best strategies for utilizing artificial *p2*. Several gradient-based and non-gradient optimizers will be compared along with strategies for utilizing POST's internal optimizer to speed up the process. It is expected that gradient-based optimizers will be less efficient than non-gradient optimizers due to the non-linear and multi-modal characteristics of the solution space. In addition, those gradient-based methods which rely on knowledge of previous iteration data will likely also be less efficient when POST's internal optimizer is also in use, as u-vector data returned after a connecting run will be different from that which was input.

Establishing a strategy for finding globally optimal performance for a single POST case can help in multiple ways. First, it can help guide an experienced analyst to more efficiently and effectively find a near global optimal solution for a given case. This is in part due to the removal of any predispositions the analyst may have based upon previous experience that could result in a sub-optimal case. Second, assuming an experienced analyst has provided an appropriate input file, a new analyst can use this strategy to effectively find a near global optimal solution without prior knowledge. Finally, by leveraging computer resources the task of optimizing trajectories can be done virtually non-stop, as computers can work more than 40 hours per week, do not take lunch breaks, and do not go home for the weekend.

[1] NASA Systems Engineering Handbook, National Aeronautics and Space Administration, December 2007, NASA/SP-2007-6105 Rev 1.

[2] Blair, J., Ryan, R., Schutzenhofer, L., and Humphries, W., "Launch Vehicle Design Process: Characterization, Technical Integration, and Lessons Learned," Tech. rep., National Aeronautics and Space Administration, May 2001.

[3] Brauer, G.L., Cornick, D.E., Habeger, A.R., Peterson, F.M., Stevenson, R., "Program to Optimize Simulated Trajectories (POST). Volume 1: Formulation Manual," NASA-CR-132689, 01 April 1975.

[4] Dees, P.D., Zwack, M.R., Steffens, M., Edwards, S., Diaz, M.J., Holt, J.B., "An Expert-System Driven Method for Parametric Trajectory Optimization During Conceptual Design," AIAA SPACE 2015 Conference and Exposition. 31 Aug. – 2 Sept. 2015, Pasadena, CA, United States.

[5] Dees, P.D., Zwack, M.R., Steffens, M., Edwards, S. "Augmenting Conceptual Design Trajectory Tradespace Exploration with Graph Theory," AIAA SPACE 2016 Conference and Exposition.13-16 Sept. 2016, Long Beach, CA, United States.