AAS 17-416

MULTI-TIERED APPROACH TO CONSTELLATION MANEUVER OPTIMIZATION FOR LOW-THRUST STATION-KEEPING

Andris D. Jaunzemis*, Christopher W.T. Roscoe†, Marcus J. Holzinger‡

This paper presents a multi-tiered approach to constellation-wide optimization of low-thrust station-keeping maneuvers. Starting from the general problem of constellation maneuver optimization, a tractable solution is presented for station-keeping. The approach utilizes a gradient-descent algorithm to efficiently drive each satellite toward its nominal orbit, encapsulating this trajectory optimization in an outer-loop genetic algorithm to optimize within discrete and non-differentiable constellation-level constraints. The output trajectories are validated and refined in a high-fidelity environment using NASA's General Mission Analysis Tool. A concrete example with operational constraints is presented, and limits of the computation-driven assumptions in the tractable solution are assessed.

INTRODUCTION

There is an ever-growing interest in satellite fractionation and constellations, which provide many advantages over large monolithic satellites including increased coverage and lower unit cost.[1] However, this increases system complexity and requires the development of new techniques for controlling these satellites through different phases of operation. During launch and deployment, a control scheme might be applied that allows long thrust- and coast-arcs since the execution time for these maneuvers is extended. However, during operation, time spent maneuvering typically impinges on time spent performing the mission, so here maneuvers should be over short time-scales and not disrupt the mission operations.

Since small satellites are often weight and power constrained, their propulsive capabilities are limited. Many trajectory design approaches assume impulsive maneuvering,[2] which is not applicable to low-thrust propulsion. Additional previous work has approached the control problem for a decentralized constellation, assessing stability and optimality of leader-follower and democratic maneuvers.[1,3] These studies focus on constellation maneuvering from a solely fuel-optimization perspective. While this is an important aspect of maneuvering, the operational constraints of the constellation should also be considered. For instance, an operator might be concerned with ensuring that satellites don't maneuver while in eclipse so as to ensure power limits are not exceeded. Alternately, regions-of-interest on the Earth might define areas where ground-observing satellites should not be maneuvering but should instead be focusing on their mission. These constraints are often non-differentiable and the decisions made are often discrete-valued actions, so typical optimization methods (e.g. gradient descent), which can be applied to the dynamics of maneuvering, are not readily applicable to the larger optimization within constellation constraints.

*Graduate Research Assistant, Aerospace Engineering, Georgia Institute of Technology, adjaunzemis@gatech.edu
†Aerospace Engineer, Applied Defense Solutions, Inc., croscoe@applieddefense.com
‡Assistant Professor, Aerospace Engineering, Georgia Institute of Technology, holzinger@ae.gatech.edu

This paper presents a general method for optimizing the maneuvers of a constellation of satellites subject to both individual trajectory and constellation-wide constraints. Starting with a development of the general problem without prescribing specific cost functions or constraints, assumptions are explicitly stated to arrive at a computationally tractable problem and related solution techique. The method incorporates both gradient-descent techniques to take advantage of the orbital dynamics as well as metaheuristics to optimize in the face of mixed-integer decision variables and non-differentiable constraints. The implementation of this method is discussed, and a trajectory refinement step is added to adjust the trajectories using higher-fidelity dynamics, allowing simplified dynamics and reduced computational complexity in the optimization routines. Sample results are then discussed, along with an analysis on the sensitivity of a set of simplifying assumptions employed in the implementation.

Constellation Trajectory Optimization Problem

Spacecraft trajectory design aims to drive the spacecraft state toward some nominal desired state. Extending this to a constellation-wide paradigm, the goal is to keep the minimize state error in every satellite of the constellation. Trajectory optimization attempts to design trajectories which minimize some parameters that may include maneuver time and/or control usage. Additional constraints may be imposed on the maneuvers to ensure that they do not impact mission objectives, such as avoiding maneuvers over observation areas. Therefore, constellation trajectory optimization is inherently is a high-dimensional, mixed-integer, multi-objective optimization problem with non-linear constraints on space object states and associated parameters.

Many existing satellite constellations are composed of relatively low numbers of monolithic satellites with powerful propulsion and excess fuel for executing station-keeping maneuvers. Therefore, maneuver constraints are easier to resolve simultaneously and near-impulsive maneuvers may be used. The recent surge in interest in large constellations of smaller satellites (e.g. cubesats) imposes a different set of constraints due to decreased propulsive capability and increased numbers of satellites to consider. This increases the difficulty of simultaneously satisfying all the constraints, and adds the difficulty of optimizing low-thrust trajectories.

This section intends to develop a general description of the constellation trajectory optimization problem. Importantly, the full constellation trajectory optimization problem is shown to be computationally intractable, so solutions are proposed by imposing additional assumptions to arrive at a computationally tractable sub-problem.

Constellation Geometry

For the general formulation, a satellite constellation is simply defined as a collection of $\mathcal{N}$ satellites. These satellites may be distributed throughout different orbit planes and regimes, each with a nominal orbit state defined at time t by an orbit state $\mathbf{x}_{i,nom} \in \mathbb{R}^6 \, \forall \, i \in 1, \ldots, \mathcal{N}$. The actual orbit state for satellite i at time t is given by $\mathbf{x}_i \in \mathbb{R}^6$. Therefore, the state error for satellite i at time t is given by $\delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{i,nom}$. The satellite state dynamics are expressed by:

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}_x(\mathbf{x}_i, t) \tag{1}$$

Satellite Trajectory Optimization

Consider the optimization of the trajectory of satellite i. This optimization may be subject to a number of constraints and design objectives (e.g. maneuver time and control usage). In addition to

the spacecraft state, a multitude of parameters may be required to express the design objective and constraints, including the satellite attitude, control authority, fuel remaining. These parameters are defined generically as $\mathbf{p}_i(\mathbf{x}_i, t) \in \mathcal{P}_i$, as these may also be state- and/or time-dependent parameters. These parameters may be partitioned into controlled parameters $\mathbf{p}_{i,c} \in \mathcal{P}_{i,c}$, over which the trajectory designer has direct control (e.g. control effort as a function of time, $\mathbf{u}_i(t)$), and uncontrolled parameters $\mathbf{p}_{i,u} \in \mathcal{P}_{i,u}$, such that $\mathcal{P}_{i,c} \times \mathcal{P}_{i,u} \in \mathcal{P}_i$.

A general design objective for the trajectory of satellite i may be expressed as:

$$f(\mathbf{x}_i, \mathbf{p}_i, t) \; : \; \mathbb{R}^6 \times \mathcal{P}_i \mapsto \mathbb{R} \tag{2}$$

Similarly, general constraints based on the states and associated parameters may be expressed as:

$$g_{i,j}(\mathbf{x}_i, \mathbf{p}_i, t) \geq 0 \quad \forall\, j \in 1, \ldots, \mathcal{J}_i \tag{3}$$

$$h_{i,j}(\mathbf{x}_i, \mathbf{p}_i, t) = 0 \quad \forall\, k \in 1, \ldots, \mathcal{K}_i \tag{4}$$

where $\mathcal{J}_i$ is the total number of inequality constraints imposed on and $\mathcal{K}_i$ is the total number of equality constraints.

Therefore, the individual satellite trajectory design problem may be written as an optimization problem as follows:

Problem 1 (satellite trajectory optimization): Designing a feasible trajectory for satellite i that minimizes the trajectory design objective over a time interval $\mathcal{T}$ is equivalent to the following optimization problem:

$$\min_{\mathbf{p}_{i,c}} \quad f(\mathbf{x}_i, \mathbf{p}_i, t)$$

subject to

$$\begin{aligned} &\dot{\mathbf{x}}_i(t) = \mathbf{f}_x(\mathbf{x}_i, t) \quad , \quad t \in \mathcal{T} \\ &g_{i,j}(\mathbf{x}_i, \mathbf{p}_i, t) \geq 0 \quad \forall\, j \in 1, \ldots, \mathcal{J}_i \\ &h_{i,k}(\mathbf{x}_i, \mathbf{p}_i, t) = 0 \quad \forall\, k \in 1, \ldots, \mathcal{K}_i \end{aligned}$$

Note that the design objective and constraints may be mixed-integer, non-linear, or non-differentiable, and no assumptions on their form have been made yet in this general derivation.

Constellation Trajectory Optimization

Optimizing the trajectories for all the satellites in the constellation builds upon the individual trajectory optimization developed above and adds constellation-specific constraints. For instance, it may be an operational constraint that at least one satellite in a region is non-maneuvering at any given time to avoid total coverage drop-out due to maneuvering. These types of constraints are often mixed-integer and non-differentiable, which makes them hard to solve using gradient-based methods.

Further complicating this problem is the fact that the design objectives for all $\mathcal{N}$ satellites may conflict with each other. Therefore, a general design objective for optimizing the maneuvers for each satellite in the constellation can be formulated as a multi-objective optimization as follows:

$$F(\mathbf{x}_1, \ldots, \mathbf{x}_\mathcal{N}, \mathbf{p}_1, \ldots, \mathbf{p}_\mathcal{N}, w_1, \ldots, w_\mathcal{N}, t) : (\mathbb{R}^6 \times \mathcal{P} \times \mathcal{W}) \times \mathcal{N} \mapsto \mathbb{R} \tag{5}$$

$$= \sum_{i=1}^{\mathcal{N}} w_i f(\mathbf{x}_i, \mathbf{p}_i, t) \tag{6}$$

where $w_1, \ldots, w_{\mathcal{N}} \in \mathcal{W} = [0, 1]$ are weights assigned to the individual objective functions for each satellite, such that $\sum_{i=1}^{\mathcal{N}} w_i = 1$.

Similarly, general constellation-wide constraints based on the states and associated parameters may be expressed as:

$$G_a(\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{N}}, \mathbf{p}_1, \ldots, \mathbf{p}_{\mathcal{N}}) \geq 0 \quad \forall\, a \in 1, \ldots, \mathcal{A} \tag{7}$$

$$H_b(\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{N}}, \mathbf{p}_1, \ldots, \mathbf{p}_{\mathcal{N}}) = 0 \quad \forall\, b \in 1, \ldots, \mathcal{B} \tag{8}$$

where $\mathcal{A}$ and $\mathcal{B}$ are the number of constellation-wide inequality and equality constraints, respectively.

Therefore, the individual satellite trajectory design problem may be written as an optimization problem as follows:

Problem 2 (constellation trajectory optimization): Designing feasible satellite trajectories that minimize the constellation trajectory design objective over a time interval $\mathcal{T}$ is equivalent to the following optimization problem:

$$\min_{\mathbf{p}_{1,c}, \ldots, \mathbf{p}_{\mathcal{N},c}, w_1, \ldots, w_{\mathcal{N}}} \quad F(\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{N}}, \mathbf{p}_1, \ldots, \mathbf{p}_{\mathcal{N}}, w_1, \ldots, w_{\mathcal{N}}, t)$$

subject to

$$\begin{aligned}
&\dot{\mathbf{x}}_i(t) = \mathbf{f}_x(\mathbf{x}_i, t) \quad \forall\, i \in 1, \ldots, \mathcal{N}, \quad t \in \mathcal{T} \\
&g_{i,j}(\mathbf{x}_i, \mathbf{p}_i, t) \geq 0 \quad \forall\, j \in 1, \ldots, \mathcal{J}_i,\ i \in 1, \ldots, \mathcal{N} \\
&h_{i,k}(\mathbf{x}_i, \mathbf{p}_i, t) = 0 \quad \forall\, k \in 1, \ldots, \mathcal{K}_i,\ i \in 1, \ldots, \mathcal{N} \\
&G_a(\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{N}}, \mathbf{p}_1, \ldots, \mathbf{p}_{\mathcal{N}}) \geq 0 \quad \forall\, a \in 1, \ldots, \mathcal{A} \\
&H_b(\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{N}}, \mathbf{p}_1, \ldots, \mathbf{p}_{\mathcal{N}}) = 0 \quad \forall\, b \in 1, \ldots, \mathcal{B} \\
&\sum_{i=1}^{\mathcal{N}} w_i = 1
\end{aligned}$$

In this most-general form, the problem has yet to be specified enough for solutions to be developed. Expressions for the trajectory and constellation objective functions (Eqns. (2) and (5) must be developed, as well as expressions for the associated constraints and dynamics. The following sections will begin to apply assumptions to reduce the problem to a tractable, solvable form to be analyzed through the rest of this paper. First, the specific dynamics used to model the spacecraft trajectories are developed. Next, the individual trajectory optimization approach and overall constellation optimization approach are developed. Finally, the method for validating and refining the trajectories is discussed.

Spacecraft Dynamics

The spacecraft dynamics are modeled identically to Roscoe et al.,[2] using a modified nearly non-singular osculating orbit element set. This set of orbit elements is preferred over a classical Keplerian orbit element set because it is defined over all orbit regimes except for equatorial orbits, eliminating nearly all the non-singularities typically associated with classical orbit elements. The assumption to exclude equatorial orbits is not terribly restrictive since even very small inclinations may be used; the restriction is for orbits where the inclination is exactly zero, which is not commonly encountered in operation.

The nearly non-singular orbit elements are defined as follows:

$$\tilde{\mathfrak{oe}} = \begin{bmatrix} a & \lambda & i & q_1 & q_2 & \Omega \end{bmatrix}^{\mathrm{T}} \tag{9}$$

where a is the semi-major axis, $\lambda = M + \omega$ is the mean argument of latitude, M is the mean anomaly, ω is the argument of perigee, i is the inclination, $q_1 = e\cos(\omega)$ and $q_2 = e\sin(\omega)$ are the components of the eccentricity vector in the orbital frame, e is the eccentricity, and Ω is the right ascension of ascending node. This orbit element set is modified from the nearly non-singular orbit element set used by Gim and Alfriend[4] by using mean argument of latitude, λ, instead of true argument of latitude, $\theta = f + \omega$ where f is the true anomaly.

Using mean orbit element theory, a transformation from osculating orbit elements $\tilde{\mathfrak{oe}}$ to mean orbit elements $\bar{\mathfrak{oe}}$ is given by:

$$\bar{\mathfrak{oe}} = \mathbf{g}\left(\tilde{\mathfrak{oe}}\right) \tag{10}$$

Here we use a first order approximation[5] to Brouwer's full transformation.[6]

The osculating orbit elements evolve according to

$$\dot{\tilde{\mathfrak{oe}}} = \mathbf{f}\left(\tilde{\mathfrak{oe}}\right) + \mathbf{g}\left(\mathbf{B}\left(\tilde{\mathfrak{oe}}\right)\mathbf{u}\right) \tag{11}$$

where $\mathbf{f}$ describes the unforced dynamics including the effects of J_2 gravitational perturbations, $\mathbf{B}$ represents a modified form (using modified nearly non-singular orbit elements) of Gauss' Variational Equations.[2]

The differential orbit element dynamics are derived similarly, following Roscoe et al.[2] Assuming the difference between the actual and nominal mean orbit elements are small, the dynamics of the differential mean orbit elements are found by linearizing about the nominal orbit:

$$\delta\dot{\mathfrak{oe}}(t) = \mathbf{A}\delta\mathfrak{oe}(t) + \mathbf{B}\mathbf{u}(t) \tag{12}$$

where $\mathbf{A}$ is the Jacobian of the dynamics $\mathbf{f}$ evaluated on the nominal orbit. The general solution to these linearized dynamics is well known:

$$\delta\mathfrak{oe}(t) = \mathbf{\Phi}(t, t_0)\delta\mathfrak{oe}(t_0) + \int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)\mathrm{d}\tau \tag{13}$$

where $\mathbf{\Phi}(t_2, t_1)$ is the state transition matrix (STM) of $\mathbf{A}$ from t_1 to t_2, which is derived by Gim and Alfriend.[4]

Station-Keeping Trajectory Optimization

In a constellation, there are a number of modes of operation which might require very different maneuver approaches. For instance, raising a satellite from a deployed parking orbit into its operational orbit (orbit insertion) or lowering a satellite from its operational orbit at end-of-life (orbit removal) both typically require significant propulsive effort due to the large state change. These maneuvers may be planned using near-impulsive techniques if the satellite has high-thrust capability, or otherwise may require long time-scale maneuvers, using long thrusting-arcs such as spiral trajectories. On the other end of the spectrum are station-keeping maneuvers, which are mid-life orbit corrections to maintain a nominal operational orbit. These typically require less propulsive effort

due to the smaller state change required, and may be accomplished using either high- or low-thrust propulsion in short time-scales.

To further specify the constellation trajectory optimization problem and develop a tractable solution, the following assumptions are made: First, all maneuvers considered in this study are station-keeping maneuvers, so larger orbit changes are not considered here. Additionally, the satellites in this constellation utilize low-thrust propulsion since this is a common design characteristic of proposed cubesat constellation architectures. The low-thrust propulsion is assumed to have a known maximum control authority limit, u_{lim}.

Since station-keeping maneuvers are typically small and the spacecraft is likely to already be near its nominal orbit, the dynamics can be linearized about the nominal orbit and a gradient-descent optimization algorithm can be applied to control toward the nominal condition. The trajectory is optimized in using a deterministic time-fixed two-point boundary-value optimization similar to the trajectory optimizer utilized in previous work by Jaunzemis et. al.[7] This will form the solution for a sub-problem of the greater constellation optimization problem.

For a given maneuver start time t_0, we know the estimated actual state $\text{œ}(t_0)$ and the desired nominal state $\text{œ}_{nom}(t_0)$; therefore, we know the initial differential orbit elements, $\delta\text{œ}(t_0) = \text{œ}(t_0) - \text{œ}_{nom}(t_0)$. At the maneuver end time t_f, we know the desired nominal state, $\text{œ}_{nom}(t_f)$, and we know that we want to attain that state, so $\delta\text{œ}(t_f) = \mathbf{0}$. These boundary conditions are fixed for this subproblem, and the job of the optimizer is to vary the control trajectory to find the trajectory that minimizes the cost function $f(\cdot)$ from (2):

$$\begin{aligned} \min_{\mathbf{u}(t)} \quad & f\left(\delta\text{œ}(t), \mathbf{u}(t)\right) \\ \text{subj. to} \quad & \delta\text{œ}(t_f) = \mathbf{0} \\ & \|\mathbf{u}(t)\|_2 \leq u_{lim} \end{aligned} \tag{14}$$

In the case of station-keeping optimization of a single spacecraft given fixed-time endpoints, it is safe to assume that a reasonable optimization criterion is fuel consumption, as reducing fuel consumption will extend operational life. For low-thrust spacecraft, typically one of two cost functions is assumed based on the type of propulsion. With variable specific impulse (VSI) engines, both the direction and magnitude of the thrust acceleration can be controlled (within control authority limits). For constant specific impulse (CSI) engines, only the direction of the thrust magnitude may be modified; the control magnitude is fixed. In this study, we proceed with the assumption of VSI engines to explore the entire direction-magnitude control space, though CSI engines could also be implemented using a slightly different cost function.

The cost functions for VSI and CSI trajectories can be found in Prussing and Conway.[8] For VSI propulsion, the trajectory cost function for satellite i is given in Eqn. (15):

$$J_{vsi}(\mathbf{u}_i(t)) = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}_i(\tau)^{\mathrm{T}}\mathbf{u}_i(\tau)\mathrm{d}\tau \tag{15}$$

where $\mathbf{u}(t)$ are the control thrust accelerations, direction and magnitude. Furthermore, a solution for the continuous-time optimal control trajectory $\mathbf{u}_i(t)$ can be approximated through discretization of the time-range t_0 to t_f. Using a discretization of N_T steps results in the following discrete-time

version of the VSI cost function:

$$J_{vsi}(\mathbf{u}_i(t)) = \frac{1}{2}\sum_{s=0}^{N_T-1} \mathbf{u}_i(t_s)^{\mathrm{T}}\mathbf{u}_i(t_s)\delta t_s \tag{16}$$

where δt_s is the time discretization step at time t_s, defined as $\delta t_s = t_{s+1} - t_s$. Therefore, using the VSI cost function as the trajectory optimization design objective and the nearly non-singular orbit element dynamics from Eqn. (11), we arrive at the following approximate solution to *Problem 1*:

Problem 1 Approximation (satellite trajectory optimization): Designing a feasible trajectory for satellite i that minimizes the VSI control cost function over a time interval $\mathcal{T}$ is equivalent to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{u}_i(t_0),\ldots,\mathbf{u}_i(t_f)} \quad & \frac{1}{2}\sum_{s=0}^{N_T-1} \mathbf{u}_i(t_s)^{\mathrm{T}}\mathbf{u}_i(t_s)\delta t_s \\ \text{subj. to} \quad & \dot{\tilde{\mathfrak{oe}}}_i(t) = \mathbf{f}(\tilde{\mathfrak{oe}}_i) + \mathbf{g}\left(\mathbf{B}(\tilde{\mathfrak{oe}}_i(t))\mathbf{u}_i(t)\right), \quad \forall\, t \in \mathcal{T} \\ & g_i = u_{lim,i} - \|\mathbf{u}_i(t)\|_2 \geq 0 \\ & \mathbf{h}_i = \delta\mathfrak{oe}_i(t_f) = \mathbf{0} \end{aligned} \tag{17}$$

Note that this form does not require a constant time-step discretization, but that for the results shown in this paper a constant time-step discretization is used for ease of application. Also note that the constraints are vector-valued constraints, of the same dimension as the state vector for $\mathbf{h}_i$ and of the same dimension as the control vector for $\mathbf{g}_i$. Since only one equality and one inequality constraint each are included in this formulation, the second subscript has been dropped for ease of notation.

The end result of this trajectory optimization is a discretized control thrust acceleration trajectory, from t_0 to t_f that satisfies the time-fixed two-point boundary value problem. The following section discusses the particular approach used in this paper to solve this problem.

Station-Keeping Trajectory Optimization via Gradient Descent

A benefit of the optimization formulation in Eqn. (17) is the ability to develop gradients of both the cost function and the constraints with respect to the decision variables $\mathbf{u}_i(t_j)\,\forall t_j \in \mathcal{T}$.[7] This allows a gradient descent algorithm to quickly navigate toward local minima, adjusting control trajectories based on the predicted effect on the constraints. Therefore, expressions must be developed for $\frac{\partial J_{vsi}(\mathbf{u}_i)}{\partial \mathbf{u}_i}$ and $\frac{\partial \mathbf{h}_i}{\partial \mathbf{u}_i}$ for all $t \in \mathcal{T}$.

Using Eqns. (15) - (16), the first expression is trivial:

$$\frac{\partial J_{vsi}(\mathbf{u}_i(t_j))}{\partial \mathbf{u}_i(t_j)} = \frac{\partial}{\partial \mathbf{u}_i(t_j)}\frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}_i(\tau)^{\mathrm{T}}\mathbf{u}_i(\tau)\mathrm{d}\tau \tag{18}$$

$$\simeq \frac{\partial}{\partial \mathbf{u}_i(t_j)}\frac{1}{2}\sum_{s=0}^{N_T-1} \mathbf{u}_i(t_s)^{\mathrm{T}}\mathbf{u}_i(t_s)\delta t_s \tag{19}$$

$$= \mathbf{u}(t_j) \tag{20}$$

where the approximation becomes equality if the control accelerations are constant over each step in the time discretization, δt_s.

Recall the linearization to differential orbit elements in Eqns. (11) - (13), and recall that the time range has been discretized to N_T time steps between t_0 and t_f. Using these, the second expression may be defined as follows:

$$\frac{\partial \mathbf{h}_i(t_j)}{\partial \mathbf{u}_i(t_j)} = \frac{\partial \delta\tilde{\mathfrak{oe}}_i(t_f)}{\partial \mathbf{u}_i(t_j)} \tag{21}$$

$$= \frac{\partial}{\partial \mathbf{u}_i(t_j)} \left[\mathbf{\Phi}_i(t_f, t_0)\delta\mathfrak{oe}_i(t_0) + \int_{t_0}^{t_f} \mathbf{\Phi}_i(t_f, \tau)\mathbf{B}(\tilde{\mathfrak{oe}}_i(\tau))\mathbf{u}_i(\tau)\mathrm{d}\tau \right] \tag{22}$$

$$= \frac{\partial}{\partial \mathbf{u}_i(t_j)} \int_{t_0}^{t_f} \mathbf{\Phi}_i(t_f, \tau)\mathbf{B}(\tilde{\mathfrak{oe}}_i(\tau))\mathbf{u}_i(\tau)\mathrm{d}\tau \tag{23}$$

$$\simeq \frac{\partial}{\partial \mathbf{u}_i(t_j)} \sum_{s=0}^{N_T-1} \int_{t_s}^{t_{s+1}} \mathbf{\Phi}_i(t_f, \tau)\mathbf{B}(\tilde{\mathfrak{oe}}_i(\tau))\mathbf{u}_i(\tau)\mathrm{d}\tau \tag{24}$$

$$= \frac{\partial}{\partial \mathbf{u}_i(t_j)} \sum_{s=0}^{N_T-1} \int_{t_s}^{t_{s+1}} \mathbf{\Phi}_i(t_f, \tau)\mathbf{B}(\tilde{\mathfrak{oe}}_i(\tau))\mathrm{d}\tau\mathbf{u}_i(t_s) \tag{25}$$

since $\mathbf{u}_i(\tau)$ is constant for $\tau \in [t_s, t_{s+1}]$.

Notice that, since $\mathbf{\Phi}_i$ and $\mathbf{B}(\mathfrak{oe}_i)$ are independent of $\mathbf{u}_i$, the partial derivative isolates the $t_s = t_j$ term from the summation:

$$= \int_{t_j}^{t_{j+1}} \mathbf{\Phi}_i(t_f, \tau)\mathbf{B}(\tilde{\mathfrak{oe}}_i(\tau))\mathrm{d}\tau \tag{26}$$

In the Gim and Alfriend STM formulation[4] and the modified form of Gauss' Variational Equations for nearly non-singular orbit elements,[2] the only time-varying terms are q_1 and q_2, which vary slowly with time. Therefore, under the assumption that the STM $\mathbf{\Phi}_i(t_f, \tau)$ is constant for $\tau \in [t_j, t_{j+1}]$:

$$= \mathbf{\Phi}_i(t_f, t_j) \int_{t_j}^{t_{j+1}} \mathbf{B}(\tilde{\mathfrak{oe}}_i(\tau))\mathrm{d}\tau \tag{27}$$

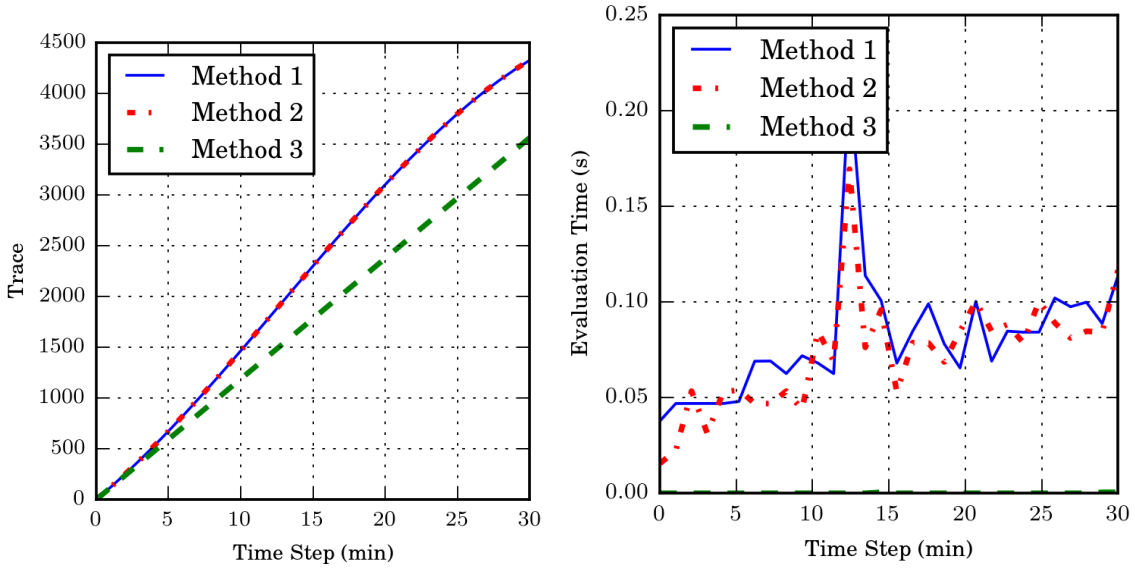
Furthermore, under the assumption that the expression for $\mathbf{B}(\mathfrak{oe}(\tau))$ is also constant for $\tau \in [t_j, t_{j+1}]$:

$$= \mathbf{\Phi}_i(t_f, t_j)\mathbf{B}(\tilde{\mathfrak{oe}}_i(t_j)) \int_{t_j}^{t_{j+1}} \mathrm{d}\tau \tag{28}$$

$$= \mathbf{\Phi}_i(t_f, t_j)\mathbf{B}(\tilde{\mathfrak{oe}}_i(t_j))\delta t_j \tag{29}$$

This final expression intuitively conveys the effect of this constraint gradient: $\mathbf{B}(\tilde{\mathfrak{oe}}_i(t_j))\delta t_j \mathbf{u}_i(t_j)$ is the state-change caused by control input $\mathbf{u}_i(t_j)$ over time interval δt_j, and $\Phi(t_f, t_j)$ propagates that state change to the final time t_f. Equations (26) - (29) therefore describe how a change in the control input at time t_j affects the resolution of the equality constraint at t_f to satisfy the two-point boundary value problem. Equation (29) is a more computationally efficient approximation of (27), which is in turn a more computationally efficient approximation of (26).

Since both q_1 and q_2 vary slowly, these assumptions are more valid for shorter time-steps δt_s. To assess the validity of the simplifying assumptions, Eqns. (26), (27), and (29) (labeled as Method 1, Method 2, and Method 3, respectively) are evaluated with increasing timesteps. Using a nominal, near-circular LEO orbit with parameters in Table 1, the time-step is varied from 1 second to 30



(a) Matrix trace

(b) Evaluation time

Figure 1. Analysis of simplifications for equality constraint gradient as a function of time step.

minutes ($t_s \in [1, 1800]$ seconds), and the matrix $\frac{\delta \mathbf{h}}{\delta \mathbf{u}} \in \mathbb{R}^{6\times 3}$ is evaluated with a final target time of $t_f = 90$ minutes, approximately one orbit. Figure 1 shows the trace of this matrix for each method. As expected all three methods agree well for small values of t_s. Methods 1 and 2 are nearly identical for across all values of t_s, whereas the extra assumption involving Gauss' variational equations for Method 3 causes larger deviations for large t_s. Since using the trace of the resultant matrix does not lend much intuition, Fig. 2 also shows the the effect in differential orbit elements of a control input on the final state at t_f, one orbit after the start of the control input. A sample control acceleration, $\mathbf{u}(t) = [1, 1, 1]\, m/s$, is applied for t_s seconds, using the three approximations to compute the resultant change in state. Once again, Methods 1 and 2 mostly agree across all values of t_s, whereas Method 3 deviates in almost all orbit elements as t_s increases.

These results confirm the expected result that these simplifying assumptions are less valid as t_s increases. However, the benefit of using the harshest assumptions is also seen in Fig. 1 by the evaluation time results. The line for Method 3 is barely perceptible above the x axis of the plot. The evaluation time of Method 3 is orders of magnitude faster than the other methods, independent of time step, since Method 3 should be evaluated in the same amount of time regardless of the time-step. The integral in Methods 1 and 2 significantly increases computational complexity. Therefore, if the time step is kept to a low value, the assumptions of Method 3 offer significant computational benefits while maintaining sufficient accuracy.

Station-Keeping Constellation Optimization

The previous section addressed optimizing the trajectory of one spacecraft to reach a desired state within a fixed time horizon. This section returns to the broader problem: optimizing the maneuvers of all the satellites in the constellation, represented by *Problem 2*, utilizing the trajectory optimization method described by Eqn. (17). At the trajectory optimization level, the routine developed earlier will solve for the fuel-optimal trajectory for a given actual state, desired final state, and time

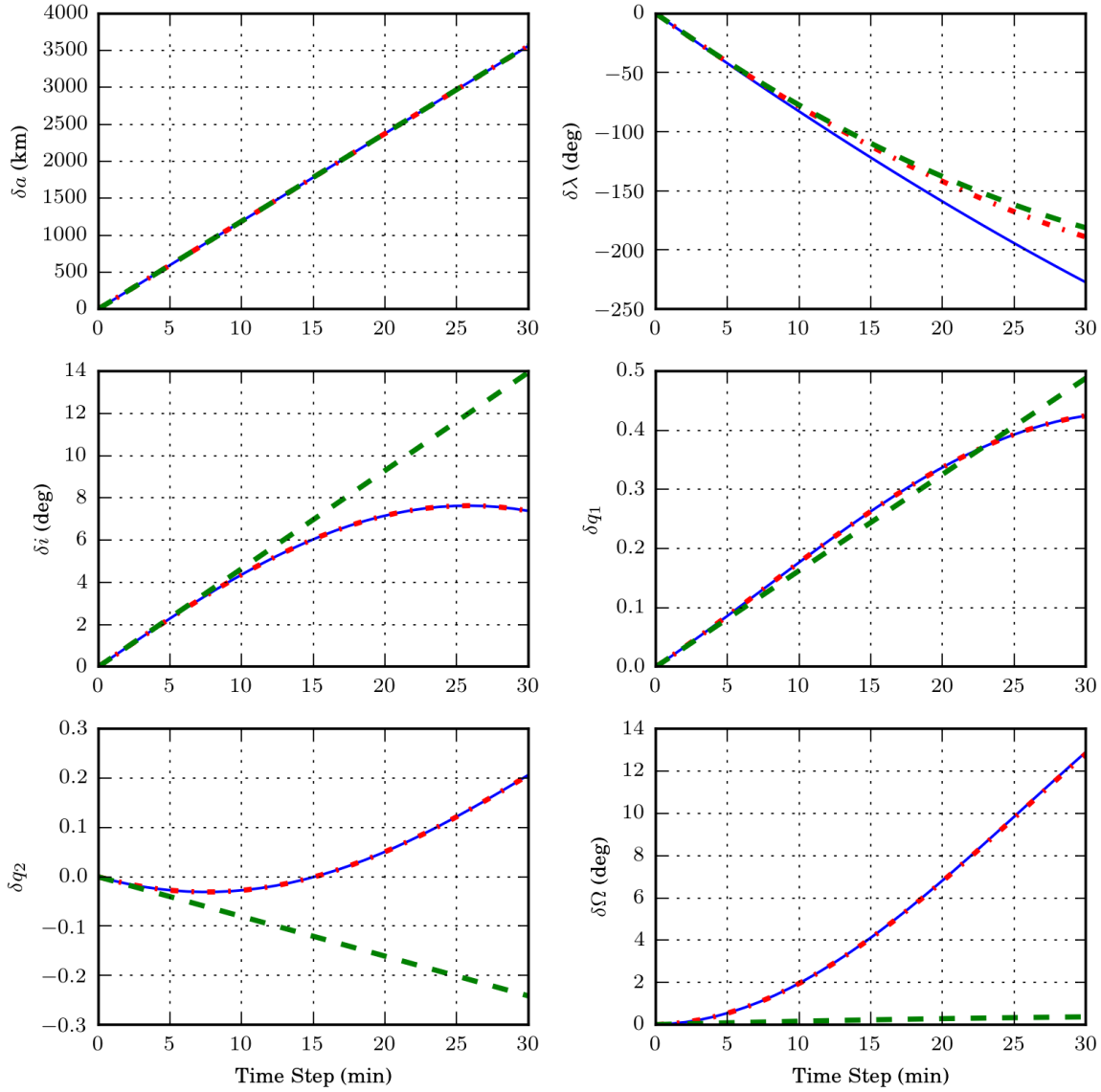


Figure 2. Differential orbit elements induced by sample control input after 1 orbit as a function of time step.

discretization. At the constellation optimization level, the decision variables are the maneuver start times for each satellite which are used to generate the fuel-optimal trajectory. Additionally, since the objective function for the constellation-wide problem is formulated as a weighted-sum of the individual trajectory objective function, those weighting factors are decision variables. Assigning the weighting factors emphasizes the relative importance of minimizing control effort for each satellite; this may appear in operation as attempts as changing weights to attempt to balance fuel-remaining among the satellites. Encoding an algorithmic process for determining these weights is beyond the scope of this paper; instead, the weights are assumed to be known and assigned prior to the beginning of the constellation trajectory optimization. For instance, the most simple set of weights one might consider is an equal weighting scheme: $w_i = \frac{1}{\mathcal{N}} \, \forall \, i \in 1, \ldots, \mathcal{N}$.

The prescription of constellation-wide constraints is difficult to generalize due to the wide variety of constraints. These constraints will often arise from operational requirements, especially for cubesat constellations since a cubesat is unlikely to be able to simultaneously maneuver and communicate or operate other instruments. Sample potential constraints include limited maneuvering over regions-of-interest to accomplish science objectives, limited simultaneous maneuvering to maintain coverage, limited maneuvering in eclipse to preserve battery charge, or even limitations to ensure passive safety of the constellation over an extended time period. These constraints are often integer-valued, meaning they are non-differentiable. Due to the wide variety of constellation constraints applicable, this formulation does not specify further the particular inequality constraints ($G_a \, \forall a \in 1, \ldots, \mathcal{A}$) or equality constraints ($H_b \, \forall b \in 1, \ldots, \mathcal{B}$). Instead, they are left in their general form here but specified later in the implementation section. The application of the VSI control cost function and associated constraints allows for the following approximate solution to *Problem 2*.

Problem 2 Approximation (constellation trajectory optimization): Designing feasible trajectories for all $\mathcal{N}$ satellites to minimize the weighted-sum of the VSI control cost function over a time interval $\mathcal{T}$ is equivalent to the following optimization problem:

$$
\begin{aligned}
\min_{t_{0,1},\ldots,t_{0,\mathcal{N}},t_{f,1},\ldots,t_{f,\mathcal{N}}} \quad & \sum_{i=1}^{\mathcal{N}} w_i \sum_{s=0}^{N_T-1} \mathbf{u}_i(t_s)^{\mathrm{T}} \mathbf{u}_i(t_s) \delta t_s && (30)\\
\text{subj. to} \quad & \dot{\tilde{\mathfrak{oe}}}_i(t) = \mathbf{f}(\tilde{\mathfrak{oe}}_i) + \mathbf{g}\left(\mathbf{B}(\tilde{\mathfrak{oe}}_i(t))\mathbf{u}_i(t)\right) \quad \forall i \in 1, \ldots, \mathcal{N}, \quad t \in \mathcal{T} \\
& g_i = u_{lim,i} - \|\mathbf{u}_i(t)\|_2 \geq 0 \quad \forall i \in 1, \ldots, \mathcal{N}, \quad t \in \mathcal{T} \\
& \mathbf{h}_i = \delta \mathfrak{oe}_i(t_f) = \mathbf{0} \quad \forall i \in 1, \ldots, \mathcal{N}, \quad t \in \mathcal{T} \\
& G_a(\mathfrak{oe}_1(t), \ldots, \mathfrak{oe}_{\mathcal{N}}(t), t_{0,1}, \ldots, t_{0,\mathcal{N}}, \mathbf{u}_1(t), \ldots, \mathbf{u}_{\mathcal{N}}(t)) \geq 0 \quad \forall \, a \in 1, \ldots, \mathcal{A} \\
& H_b(\mathfrak{oe}_1(t), \ldots, \mathfrak{oe}_{\mathcal{N}}(t), t_{0,1}, \ldots, t_{0,\mathcal{N}}, \mathbf{u}_1(t), \ldots, \mathbf{u}_{\mathcal{N}}(t)) = 0 \quad \forall \, b \in 1, \ldots, \mathcal{B} \\
& \sum_{i=1}^{\mathcal{N}} w_i = 1
\end{aligned}
$$

This formulation minimizes the weighted sum of VSI control costs for a set of station-keeping maneuvers for $\mathcal{N}$ satellites subject to constellation operational constraints. These constellation-wide constraints have yet to be specified since they are operation-specific. In the following section, the implementation of this algorithm will be discussed in detail, followed by some sample results that show the application of this approach to a constellation.

Station-Keeping Constellation Optimization via Genetic Algorithm

The constellation optimization task is a multi-objective, mixed-integer, non-linear programming problem, which places it in a difficult class of problems known as NP-complete. As in the well-known traveling-salesman problem, the worst-case running time for such an algorithm increases superpolynomially with the number of spacecraft. Additionally, many of these constraints are non-differentiable with respect to the decision variables; there is no analytical derivative for the fuel use with respect to the (integer-valued) number of maneuvering satellites at a given time, for instance. Therefore, a gradient-descent-type algorithm cannot be formulated to guide an optimizer toward local minima. However, a different class of optimization algorithms, known as metaheuristic optimization, is better-suited to address the discrete nature of the decision variables and the non-differentiability of potential cost functions used in constellation design. As the name implies, a metaheuristic algorithm employs heuristics ("rules of thumb"), as opposed to cost or constraint gradients, to guide the optimizer. Popular metaheuristic algorithms include genetic algorithms[9] and simulated annealing,[10] both of which are developed using physical phenomena as a background. A major advantage of this class of algorithms is they resist convergence, at the expense of more design point evaluations, to avoid getting stuck in local minima. Both algorithms contain stochastic mechanisms that enable exploration of the design space beyond just following gradients to the nearest optimum. The benefit then is the ability to apply these algorithms to non-convex objective spaces and avoid local minima; the draw-back is the lack of any guarantees as to the global optimality of the solution.

In this paper, constellation optimization is handled through a genetic algorithm. Below is a brief summary of the important aspects of a genetic algorithm.[9] Genetic algorithms, also sometimes referred to as evolutionary algorithms, emulate genetic processes involved in evolution. A set of design points under consideration are encoded as chromosomes, gray-code binary vectors. Each design point is represented by an allele, a subset of the chromosome. Given a set of chromosomes (a population or generation) that represent points in the design space, these chromosomes undergo a number of evolution-inspired processes. In the crossover or reproduction stage, a number of chromosomes are chosen and, if a random draw exceeds some crossover threshold, the chromosomes exchange sections of their binary string to generate a new pair of chromosomes. In the mutation stage, a chosen chromosome will flip bits if another random draw exceeds some mutation threshold, introducing additional variability in the generation. Typically the mutation threshold is much higher than the crossover threshold, as mutation occurs less frequently. In the selection or tournament stage, a number of chromosomes are chosen and compared, and the design point with the best fitness (see Eqn. (30)) is typically chosen to advance the next generation, subject to some tournament threshold. The selection process gives the genetic algorithm the "survival of the fittest" genetic analogy. Each of these processes occurs many times within a single generation, and each process can be performed in a myriad of ways, so the particular construction of the genetic algorithm is left to the algorithm designer. The stochastic nature of each step allows the design space to be explored and allows the algorithm to potentially escape local minima. For instance, the tournament threshold ensures that, with some (often low) probability, a less-optimal (higher cost) design point will be selected, encouraging diversity in the population. There are additional optional features of genetic algorithms, such as elitism which ensures the best chromosomes (as judged by their cost function) from a given generation will advance to the next generation, even if they lose in the tournament stage.

To solve Eqn. (30), the fitness of a given design point is evaluated using the weighted sum of the

individual optimal trajectories from Eqn. (17) which enforces the dynamics, control, and boundary-value constraints. An exterior penalty function is added to enforce constellation-level constraints G_a and H_b. Higher values of fitness indicate worse trajectories, either through more control effort or violation of constellation constraints, and will therefore be pruned out as the genetic algorithm advances in generations.

Trajectory Refinement

The optimization techniques thus far have been subject to a reduced dynamic model and a number of assumptions in order to reduce computational complexity. However, in operation, operators are typically also concerned with many other perturbations, such as high-order gravity effects (well above J_2 perturbations), solar radiation pressure, and 3rd body effects. Therefore, there is a need to refine the preliminary optimal trajectories generated by the gradient-descent and genetic algorithm combination in more realistic dynamics.

To address this, NASA's General Mission Analysis Tool (GMAT) is used to refine the trajectories. GMAT is a space mission design software system for the design and optimization of spacecraft missions.[11] The 2015a release includes both a Python interface and finite thrust components, both essential in application to this project. Each spacecraft trajectory received from the optimizer is loaded into a scenario in GMAT, and a differential corrector is used to refine the trajectory. Since the final target state is 6 states and the differential corrector is able to vary three components at a particular time step (direction), the initial and final control actions are chosen to be modified; there are 6 target states and 6 control states. Allowing GMAT to modify each individual discrete thrust input overwhelms the differential corrector. Therefore, the initial input is used since the propagation of that initial input over the whole orbit can have a significant effect, and the final input is used to aid in rejecting further disturbances during propagation.

IMPLEMENTATION

The constellation optimization procedure as defined above is implemented as shown in Fig. 3. Relevant constellation parameters are embedded in a configuration JSON-file, including the number of spacecraft, their nominal mean orbit states, their actual orbit states, and simulation timing parameters. Convergence tolerances for the trajectory optimization algorithm are prescribed in terms of LVLH parameters.[2] This allows an intuitive prescription of allowable drift and offset parameters, implemented as convergence constraints on the optimization.

The trajectory optimizer is configured for low-thrust, VSI engines. For constellation constraints, we impose that no two satellites may maneuver simultaneously. This restricts the maneuver start time decision variables for the genetic algorithm, and if this constraint is violated a penalty is applied to the overall cost.

The trajectory optimization routine is developed in Python 3.5, using the Spyder IDE bundled with the Anaconda distribution. The scientific computing (SciPy) and included numerical computing (NumPy) packages are leveraged, particularly the optimization toolbox in SciPy. In particular, SciPy's Sequential Least Squares Programming (SLSQP) provides a mechanism for gradient-descent optimization. Since, in station-keeping, the actual orbits are typically near the nominal orbits (small maneuvers), and with the addition of the cost- and constraint-gradients, the gradient-descent algorithm is quick.

The constellation optimization routine is also developed in Python 3.5. Here, a number of other

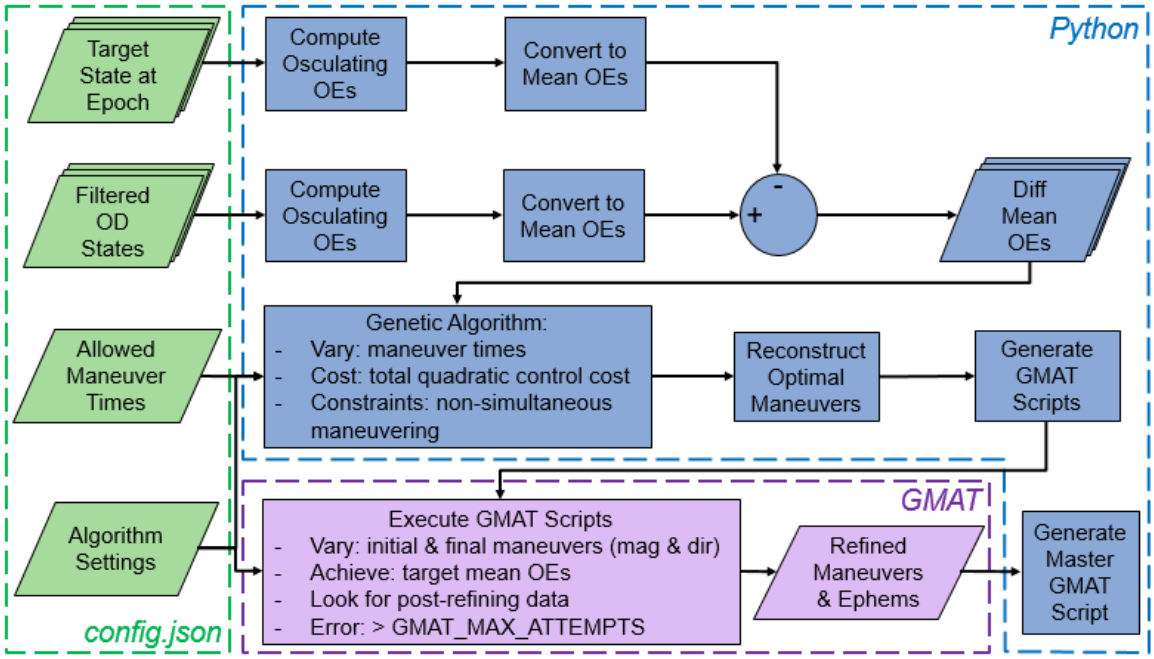


Figure 3. Constellation optimization algorithm flowchart.

packages are employed. The Distributed Evolutionary Algorithms for Python (DEAP) package provides the mechanisms for implementing a genetic algorithm.[12] The previous trajectory optimization function is wrapped inside the larger outer-loop as part of the fitness function, where the individual alleles (maneuver start times) must be evaluated. Additionally, as the name implies, DEAP supports distributed computing. Genetic algorithms are known as "embarrassingly parallel," which is to say each chromosome can be evaluated in parallel; they do not rely on each other for information. Parallelization is accomplished in DEAP through the use of the Scalable Concurrent Operations in Python (SCOOP) package, which enables the use of multiple processors or multiple cores on one processor.[13]

Finally, the trajectory is refined in higher-fidelity dynamics using GMAT. The GMAT-Python interface allows direct integration of user-generated Python functions to handle mean-osculating conversions. It also enables visualization in higher-dimensions than 2D plots. One drawback of GMAT, though, is spotty performance. The same scenario file can be run multiple times to no avail, but without changing it the analysis may then run successfully. To address this, a batch file was written to attempt execution of one file at a time, with a maximum number of re-submissions prescribed in settings. When the GMAT script is executed successfully, it generates files detailing the refined control trajectories. The final, refined trajectories (states and controls) are embedded in ephemeris files, and a final result GMAT script is generated to show the satellites performing their trajectory optimization.

SCENARIO - 6 SATELLITES, CO-PLANAR

To investigate the algorithm as developed, a sample scenario is presented to demonstrate the ability to generate a valid constellation maneuver set that minimizes the quadratic control cost function, satisfies end-point constraints, and also satisfies constellation constraints. In this scenario, the con-

Table 1. Nominal orbit parameters

Parameter	Value	Units
Semi-major Axis (a)	7300	km
Inclination (i)	50	deg
Eccentricity, Cosine Component (q_1)	0.001	-
Eccentricity, Sine Component (q_2)	-0.001	-
Right Ascension of Ascending Node (Ω)	230	deg

stellation consists of 6 satellites in one nearly-circular inclined LEO plane, with nominal initial orbit parameters as shown in Table 1. The mean arguments of latitude λ are initialized so that the satellites are distributed evenly through the orbit plane. Randomized differential orbit elements are generated as actual initial conditions. The trajectory optimization time horizon is set to one orbit period, discretized into 100 time steps, yielding a trajectory time step less than 100 seconds. This trajectory time step is well within the range of acceptable times to take advantage of the constraint gradient assumptions, as show in Figs. 1 and 2.

The constellation optimization time horizon is set to one sidereal day, discretized into 1000 equal time steps, roughly 86 seconds apart. Each time step in this horizon is a valid starting time for any satellite, and the constellation optimizer finds the optimal combination of starting times and maneuver trajectories to minimize the cost function. Additionally, the operational constraint imposed is that only one satellite may maneuver at a time to maintain coverage. For any solution that includes satellites maneuvering at the same time, a large penalty is applied to the fitness function. Each population in the genetic algorithm contains 60 individuals, and the crossover and mutation probabilities are set to 50% and 20%, respectively. 10 generations are evaluated to arrive at the final optimized solution. Finally, the resultant trajectories are refined in GMAT using a 70x70 geopotential gravity model, much higher-fidelity than the J_2 terms included in the nearly non-singular orbit dynamics.

Figure 4 shows the optimized trajectories of all 6 satellites in the LVLH frame, superimposed on one plot. The origins of the plots represents the respective nominal state for each spacecraft. The order of maneuvers can be implied from this plot by noting how many large ellipses a particular trajectory makes before proceeding toward the origin. By the end of the simulation time, all 6 satellites have reached their respective LVLH origins, meaning they have arrived at the nominal orbits. Similarly, Fig. 5 shows the LVLH errors for all 6 satellites, showing that they all settle to convergence tolerances within the allotted simulation time.

Figure 6 shows the convergence performance of the genetic algorithm for constellation optimization. The top-left plot shows the minimum and maximum fitness values in each generation. Notably, the first two generations do not contain any feasible solutions, yielding very high fitness values. Starting with generation 2, feasible solutions are found and improved. The bottom-left plot highlights this, showing an increasing percentage of each population with lower fitness values. This movement toward an optimal configuration occurs slowly, encouraging further exploration of the design space. Note that the bottom-left plot only shows the fitness range for the feasible solutions. While an increasing number of individuals in the population are feasible with each added generation, a few indiviuals still remain infeasible. This resistance to convergence to local minima is an advantage of metaheuristics, encouraging design space exploration while keeping track of local minima.

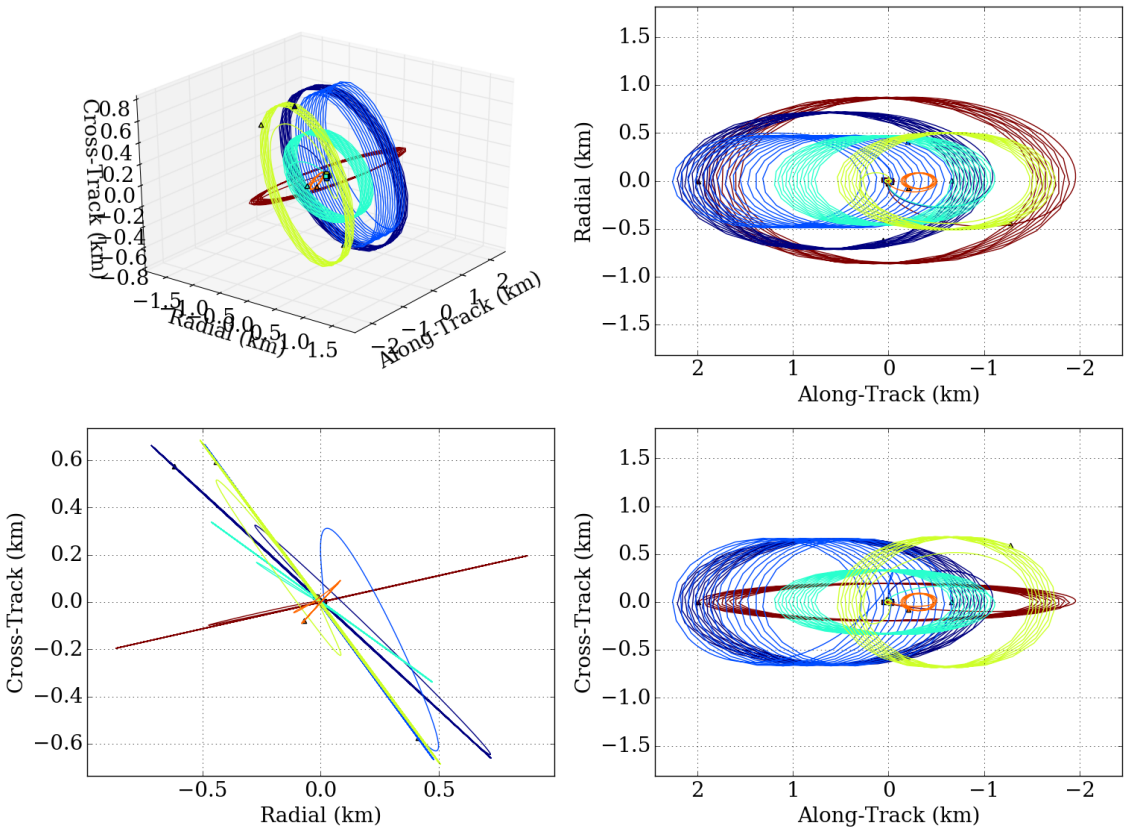


Figure 4. LVLH trajectories for all six satellites, after trajectory refining in GMAT.

Figure 7 shows the control data for the same scenario. Note that both pre-GMAT refining (dashed lines) and post-GMAT refining (solid lines) are represented to show the change in trajectory imposed by GMAT to reject the higher-order disturbances. The top three plots are the control thrust accelerations, and the bottom plot is a running total of delta-v used. The effect of the GMAT control refining, small adjustments in the initial and final thrust inputs, can be seen here.

This test case demonstrates the application of the solution to Problem 2 as expressed in Eqn. (30) to a synthetic constellation with imposed operational constraints. Alternate test cases may be constructed to apply various constellation-level constraints, as long as they can be formulated to impact the fitness function.

CONCLUSIONS

This paper develops a multi-tiered approach to constellation-wide optimization of maneuvers. The approach is first developed in a general manner consistent with optimization conventions, without prescribing specific constraints, to allow for future studies to build upon this formulation. A concrete example of constellation optimization, for low-thrust station-keeping, is developed to demonstrate application of the theoretical approach, explicitly stating the assumptions made in formulating the tractable problem and the solution. The approach utilizes a gradient-descent algorithm to efficiently drive each satellite toward its nominal orbit, encapsulating this individual trajectory op-

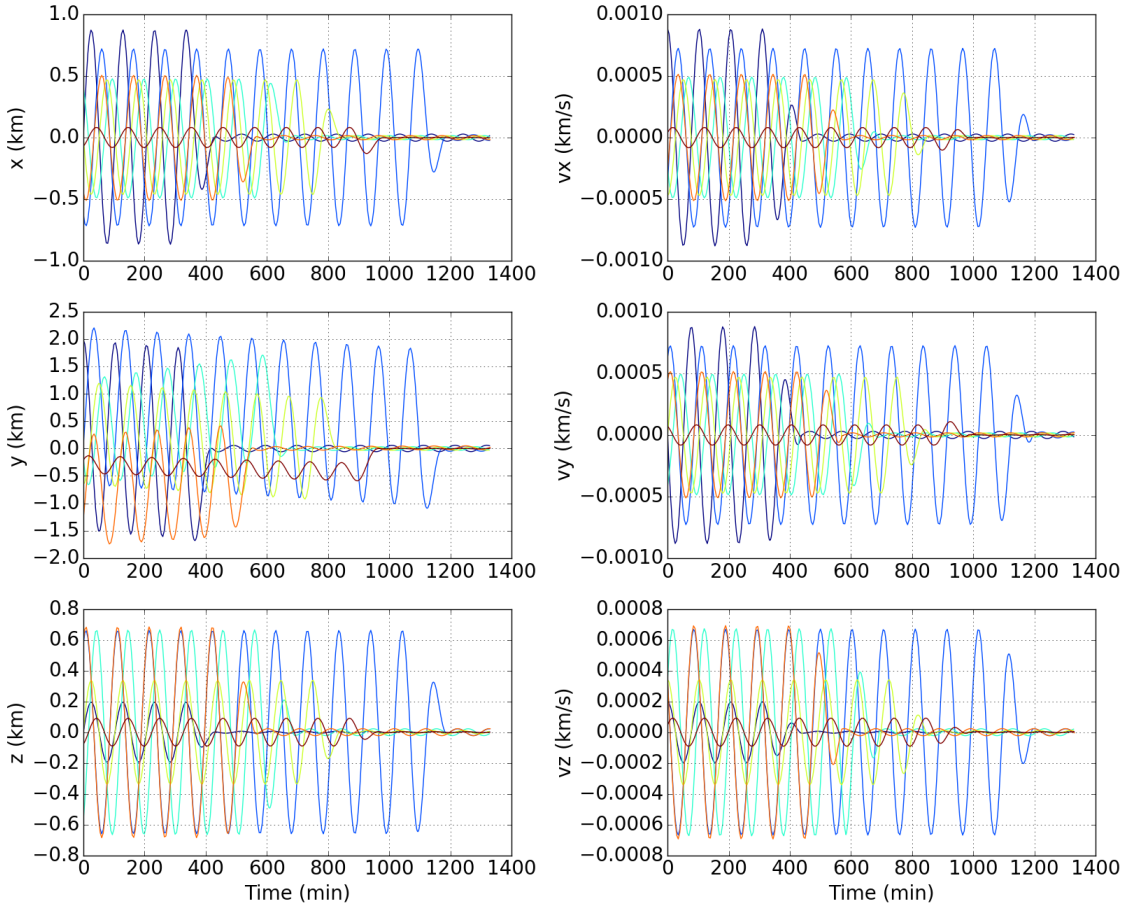


Figure 5. LVLH errors for all six satellites, after trajectory refining in GMAT.

timization in an outer-loop genetic algorithm to handle discrete and non-differentiable constellation constraints.

The effect of the simplifying assumptions in the gradient-descent method (for the constraint gradients) is presented to highlight their applicability in this formulation. The use of GMAT to refine the trajectories in high-fidelity dynamics reduces the effect of the assumptions made in favor of computational tractability, while further confirming that the effect of these assumptions is small on the overall outcome of the trajectories. Results are provided for a 6 satellite scenario, demonstrating the ability for this optimization approach to generate valid trajectories to meet mixed-integer constellation constraints alongside differentiable trajectory constraints. The trajectories generated through the hybrid genetic algorithm gradient-descent method allow a solution to be obtained in approximately one minute using a laptop computer. The final GMAT-refined trajectories show good agreement with the control trajectories generated in Python, with corrections applied at the endpoints to reject disturbances.

This work shows promising application of a hybrid optimization approach to solve this multi-objective, mixed-integer, non-linear programming problem by exploiting the dynamics in a gradient-descent method and exploring the non-convex constellation design space using metaheuristics. The

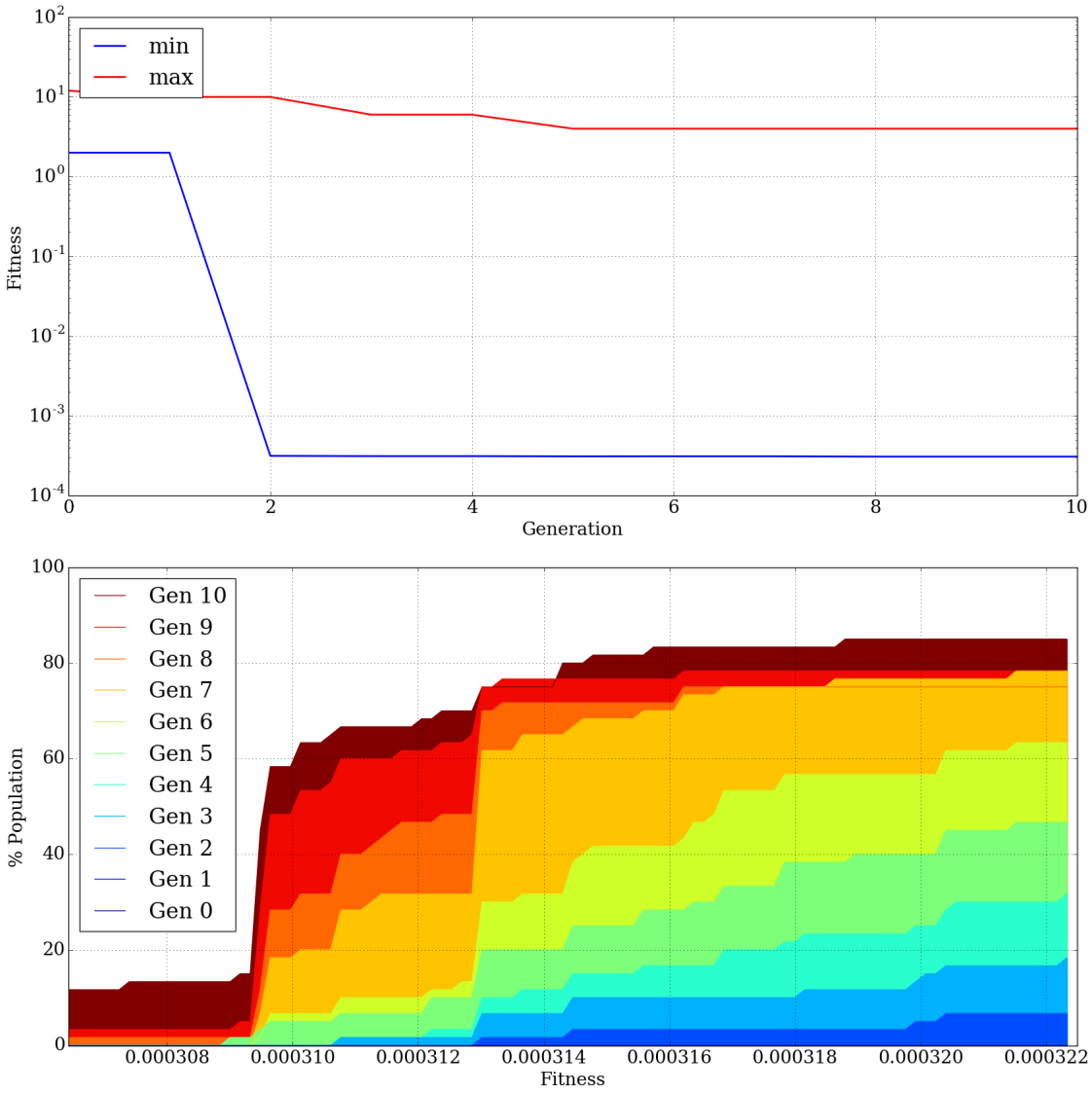


Figure 6. 6-satellite constellation, genetic algorithm convergence.

methodology presented is not limited to the specific constellation scenario presented; the general form in Eqn. (30) could be used to formulate solutions to arbitrary constellation maneuvering problems.

ACKNOWLEDGMENTS

The authors would like to thank Applied Defense Solutions, Inc. for funding this research, and Craig Nickel for guidance on this project.

REFERENCES

[1] E. Douglass, M. J. Holzinger, J. W. McMahon, and A. D. Jaunzemis, "Formation Control PProblem for Decentralized Spacecraft Systems," *AAS/AIAA Astrodynamics Specialist Conference*, 2013.

[2] C. W. Roscoe, J. J. Westphal, J. D. Griesbach, and H. Schaub, "Formation Establishment and Reconfiguration Using Differential Orbit Elements in J2-Perturbed Orbits," *Journal of Guidance, Control, and Dynamics*, 2015, doi:10.2514/1.G000999.

[3] M. J. Holzinger and J. W. McMahon, "Decentralized Mean Orbit Element Formation Stability for Uncoordinated Maneuvers," *American Control Conference (AAC)*, Washington, DC, USA, June 2013.

[4] D.-W. Gim and K. T. Alfriend, "State Transition Matrix of Relative Motion for the Perturbed Noncircular Reference Orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 956–971, doi:10.2514/2.6924.

[5] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*. AIAA Education Series, 3rd ed., 2014.

[6] D. Brouwer, "Solution of the Problem of Artificial Satellite Theory Without Drag," *Astronomical Journal*, Vol. 64, Nov. 1959, pp. 378–397.

[7] A. D. Jaunzemis, M. V. Mathew, and M. J. Holzinger, "Control Cost and Mahalanobis Distance Binary Hypothesis Testing for Spacecraft Maneuver Detection," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 9, 2016, pp. 2058–2072, doi:10.2514/1.G001616.

[8] J. E. Prussing and B. A. Conway, *Orbital Mechanics*. Oxford University Press, 2012.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, doi:10.1109/4235.996017.

[10] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 220, May 1983, doi:10.1126/science.220.4598.671.

[11] S. P. Hughes, R. H. Qureshi, D. S. Cooley, J. J. K. Parker, and T. G. Grubb, "Verification and Validation of the General Mission Analysis Tool (GMAT)," *AIAA/AAS Astrodynamics Specialist Conference* (S. Conferences and Exposition, eds.), San Diego, CA, August 2014.

[12] F.-A. Fortin, F.-M. D. Rainville, M.-A. Gardner, M. Parizeau, and C. Gagne, "DEAP: Evolutionary Algorithms Made Easy," *Journal of Machine Learning Research*, July 2012, pp. 2171–2175.

[13] Y. Hold-Geoffroy, O. Gagnon, and M. Parizeau, "Once you SCOOP, no need to fork," *XSEDE '14 Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*, No. 60, 2014.

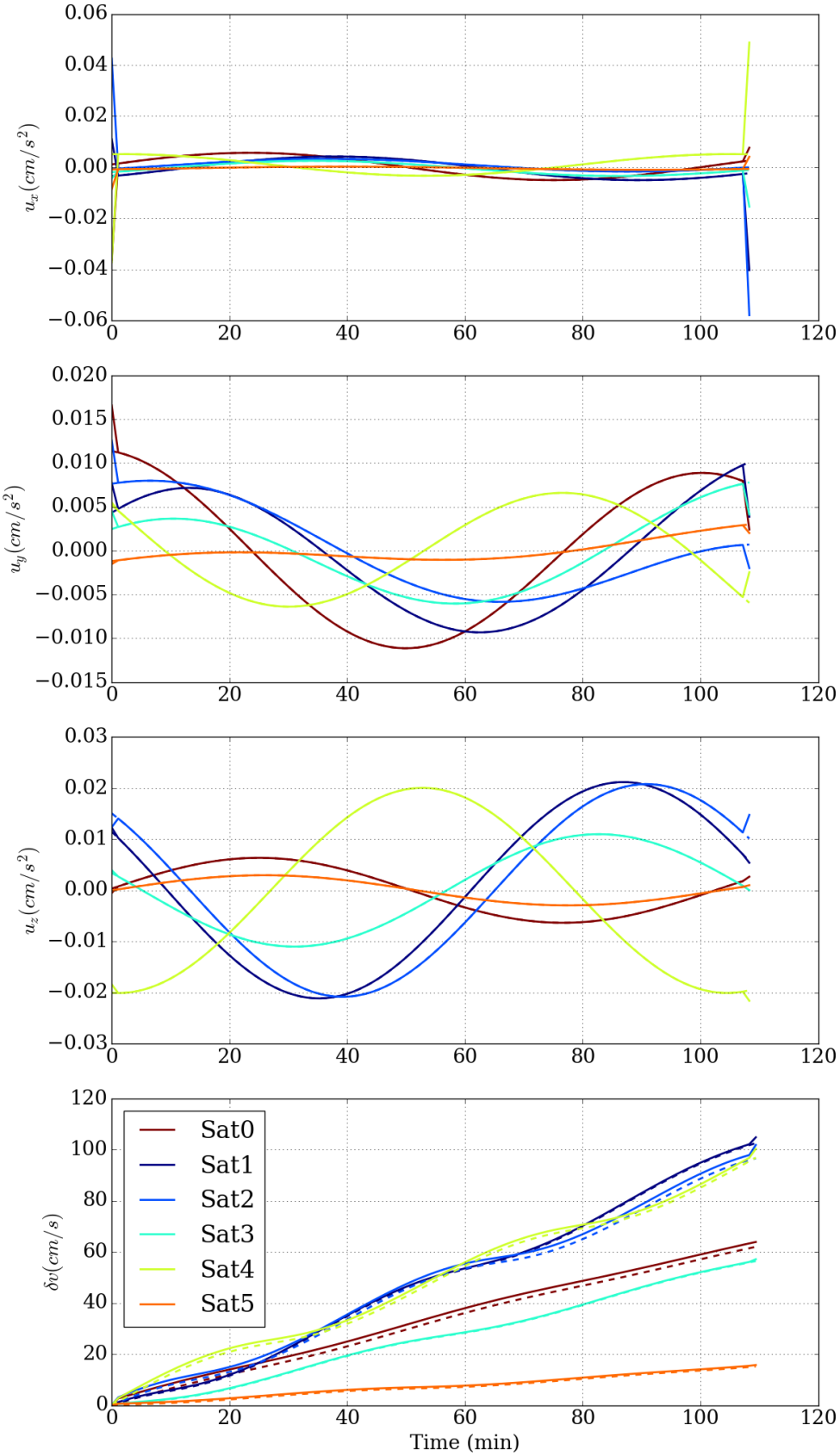


Figure 7. 6-satellite constellation, control thrust accelerations and delta-v.