Linear and Nonlinear Optimization of CubeSat Constellation Lifetime Using Differential Drag Control

Daniel Andenmatten

Russell Hawkins

Jonathan Wapman

December 12, 2018

Submitted towards fulfillment of the requirements for
MAE 298 - Optimal Control

Department of Mechanical Engineering
University of California Davis
Davis, CA 95616

1 ABSTRACT

CubeSat constellations have many uses for applications such as weather monitoring, communications, and earth imaging. CubeSats often have no onboard propulsion. However, to change velocity and altitude, CubeSats can control the amount of air resistance they experience by controlling the amount of surface area presented in their direction of motion. This causes the CubeSat to lose altitude while at the same time increasing their angular velocity. In this project, we first recreate previous work from Planet Labs and Sin et al. that implements CubeSat constellation separation using differential drag and linear programming. Sin's research linearizes nonlinear orbital dynamics in an effort to improve computational complexity. As a comparison, our project implements optimization using the real-world nonlinear dynamics and compares these results against the results of Sin's closed-loop linear programming-based method. Performance metrics include constellation altitude drop, computation complexity, and constraint satisfaction. In this project, we first successfully recreate the closed-loop linear programming optimization previously performed by Sin et al. From these experiments, we then find that linear, closed-loop optimization computation time scales approximately with N^3, where N is the number of satellites in the simulation. Finally, we show that although optimization using nonlinear constraints displays slightly better altitude drop performance for small numbers of satellites, attempts to use larger number of satellites results in scenarios where the problem's constraints cannot be met, and where results are heavily dependent on initial guesses for the optimizer.

2 OVERVIEW

Over the past decade, access to space and commercial opportunities have increased significantly. The decreasing cost of rocket launches and prevalence of modular and inexpensive "CubeSat" designs have made launching constellations of tens to hundreds of satellites feasible. Currently, companies such as SpaceX, Facebook, OneWeb, and others are developing satellite constellations to create a space-based internet designed to serve remote areas of the world which often lack communication infrastructure [1], [2]. Others, such as Planet Labs, have already deployed constellations of satellites for applications such as earth imaging [3]. There is no doubt that the number of small satellites deployed, and the range of applications they serve, will grow immensely in the near future.

The compact and modular design of small satellites has many economic advantages, however it generally imposes severe limitations on control. Often, small satellites - due to size and weight constraints - do not have any external thrusters that can be used for position and attitude control. In these cases, control of CubeSats is limited to internal reaction wheels that can control orientation only. This attitude control can be cleverly leveraged into control of the satellite's orbit by taking advantage of the attitude dependent atmospheric drag force acting on the satellite. Changing attitude changes the cross-sectional area of the satellite, and thus the drag it experiences. As the atmospheric drag increases, the satellite loses altitude while simultaneously gaining angular velocity. By intelligently controlling the drag force, the position in the constellation can be regulated.

When applied to a constellation of CubeSats, selectively changing the drag on each satellite individually can be used to create a constellation of equally-spaced satellites from an initially

concentrated distribution. This differential drag method of satellite positioning has been successfully demonstrated by Planet Labs's earth-imaging constellation [3]. There is a tradeoff, however, in that increasing the drag on a satellite necessarily decreases the service lifespan of the satellite. Drag inevitably leads to loss of altitude, eventually deorbiting the satellite. Therefore, any maneuver carried out via modifying drag forces should be optimized to minimize loss in altitude. This optimization procedure is the focus of this project. However, note that there are other optimizations for acquisition time, separation error or local area coverage due to other objectives. We will replicate the altitude optimization procedure recently presented in the literature, and then attempt to expand one aspect of that procedure.

3 LITERATURE REVIEW

The physics of satellite control using differential drag is a well-studied problem [4]–[6], and has been used in practice several times. In 2013, the Aerospace Corporation's three AeroCube-4 satellites utilized in-orbit differential drag to control satellite spacing by controlling their solar panels [7]. In a commercial application, Planet Labs has used differential drag to control spacing of an array of over 100 satellites [3]. Planet Labs' constellation uses "bang-bang" inputs, where each satellite is either in maximum drag or minimum drag [3], [8]. In our study, we consider the satellite's drag to be any value between the minimum and maximum values.

The majority of work in this proposal extends previous work done by Sin et al. in [9]. The authors present a linear-programming-based method of separating a constellation of satellites using differential drag with the goal of maximizing the radius of the lowest satellite of the constellation. This is equivalent to maximizing the operational lifetime of the constellation. Additionally, this linear programming method was analyzed both for open-loop control and closed-loop control (using Model Predictive Control). The work in this paper uses Planet Labs' constellation as its model (although it was never implemented in practice), but can be generalized to apply to any CubeSat constellation. Sin's work has been further extended in [10], again with a focus on Planet Labs' constellation implementation. However, in this paper, Blatner implements linear and quadratic programming algorithms to minimize non-uniform spacing between adjacent satellites, rather than maximizing the minimum radius of the constellation. Other portions of his work focus on optimal positioning of the satellites based on initial conditions when the satellites first separate.

4 ORBITAL DYNAMICS AND MODEL GENERATION

4.1 ORBITAL DYNAMICS

The physics of satellite control using differential drag is a well-studied problem [4]–[6]. By using a satellite's internal reaction wheels to reorient, the amount of atmospheric drag acting on a satellite can be controlled, thus allowing the operator some degree of control over the satellite's descent rate and angular velocity without the need for external actuators such as thrusters. The satellite's motion to be described by the following second-order ordinary differential equation:

$$\ddot{\vec{r}} = -\frac{\mu_e}{|\vec{r}|^3}\vec{r} + \vec{a}_{perturb}$$

where $\vec{r}$ is the position vector from the center of the earth pointing to the satellite and μ_e is the gravitational parameter of the Earth. The first term in the equation is the acceleration due to Earth's gravity, the second term $\vec{a}_{perturb}$ includes all accelerations due to other perturbing forces. These forces include perturbations due to the gravitational influence of the sun and moon, solar radiation pressure, and atmospheric drag. For our purposes we only consider atmospheric drag, whose acceleration is given by

$$\vec{a}_{atm} = -\frac{1}{2}\frac{C_d A}{m}\rho|\vec{v}_{rel}| * \vec{v}_{rel}$$

In this equation, C_D is the satellite's drag coefficient, A is the cross-sectional area of the satellite, m is the satellite mass, ρ is the atmospheric density, which is a function of altitude, and $\vec{v}_{rel}$ is the velocity of the satellite relative to the atmosphere.

For the simulations in this paper, we use the same constant values as Sin, who used the values from Li and Mason [8] for the satellite's drag coefficient, mass, maximum and minimum drag surface areas. For atmospheric density, we use the same simplified version of the Harris-Priester model as authors in [9]. In order to make the problem tractable, several assumptions are made. First, this model assumes near-circular orbits. Second, the relative velocity of the satellite with respect to the atmosphere is based on the assumption that the atmosphere rotates with that of the earth's rotation. Third, since atmospheric drag is opposite of the velocity vector, we ignore the radial acceleration component of the drag perturbation. Finally, we only consider the component of the earth's angular velocity that is about the normal axis of the orbital plane. We assume that both the angular velocity of the earth about its axis and the inclination of the orbit are constant. For a near-polar orbit, the relative speed of the satellite is essentially the tangential speed of the satellite. After a transformation into polar coordinates and applying these assumptions we are left with the following continuous model for the equations of motion for the satellite:

$$\ddot{r} = r\omega^2 - \frac{\mu_e}{r^2}$$

$$\ddot{\theta} = \frac{1}{r}\left(-2\dot{r}\omega - \frac{1}{2}\frac{C_d}{m}\rho(r)|\vec{v}_{rel}|^2 A\right)$$

This optimization problem requires the use of a discrete-time model to determine the radius, angular velocity, and angular position of each satellite i at time step k, which was introduced by Sin in [9]. These equations are given below:

$$r_i(k+1) = r_i(k) + \Delta t * S^R\{r_i(k), \omega_i(k)\} * u_i(k)$$

$$\omega_i(k+1) = \omega_i(k) + \Delta t * S^\Omega\{r_i(k), \omega_i(k)\} * u_i(k)$$

$$\theta_i(k+1) = \theta_i(k) + \omega_i(k) * \Delta t + \frac{1}{2}\Delta t * S^{\Omega}\{r_i(k), \omega_i(k)\} * u_i(k)$$

Note that these functions are inherently nonlinear. Each r, ω, and θ at each time step depend on their previous values. The functions S^R and S^{Ω} capture the effect of drag on the dynamics. $u_i(k)$ represents the exposed surface area of satellite i at each time step k, which is the variable used to control the system. S^R and S^{Ω} therefore determine how each input to the system affects the dynamics. These functions are derived by applying Gaussian variation of parameters to the equations of motion, and are used to approximate the rates of change of the time-varying elements in the solution for the unperturbed, two-body system due to small forces. Vallado showed how the average rate of change in the radius of an orbit and the angular speed of the satellite can be expressed in terms of the atmospheric drag perturbation [11]. These results can be expressed as follows:

$$S^R(r, \omega) = -\frac{C_d}{m}\rho(r)|\vec{v}_{rel}(r, \omega)|^2\sqrt{\frac{r^3}{\mu_e}}$$

$$S^{\Omega}(r, \omega) = \frac{3}{2}\frac{C_d}{m}\rho(r)|\vec{v}_{rel}(r, \omega)|^2\frac{1}{r}$$

We note that r(T), ω(T), and θ(T) do not depend linearly on the input u for our model.

4.2 LINEAR OPTIMIZATION PROBLEM

Our goal is to spread out an initial cluster of satellites in low Earth orbit. The objective of the optimization however can vary depending on the use of the CubeSat constellation. We are interested in maximizing the operational lifetime of the constellation by minimizing altitude loss during acquisition. Furthermore, we must complete this constellation formation maneuver in a fixed number of days subject to constraints on relative spacing and velocity between pairs of satellites, as well as minimum and maximum surface areas. Thus, the goal is to minimize the drop in altitude of the constellation, which is achieved by maximizing the altitude of the lowest satellite in the constellation at the final time step T of the optimization problem.

$$maximize \min_{i=1,\ldots,N} r_i(T)$$

Angular separations can be measured either between sequentially numbered satellites, or relative to a single reference satellite. Defining the angular position vector as $\theta = [\theta_1, \ldots, \theta_N]$, relative angular separations for all satellites can be expressed as $\Delta = D \cdot \theta$, and the angular separation error is $\Delta - \Delta des = D \cdot \theta - \Delta des$. The definitions for the separation matrices D and the desired angular separations Δdes are given below for both formulations. We chose to use sequential separations for our optimization.

Sequential Separations

$$D := \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ -1 & & & 1 \end{bmatrix} \in \mathbb{R}^{N \times N}$$

$$\Delta_{des} := \left[\frac{2\pi}{N}, \dots, \frac{2\pi}{N}, \frac{2\pi}{N}(N-1)\right]$$

Relative Separations

$$D := \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{bmatrix} \in \mathbb{R}^{N \times N}$$

$$\Delta_{des} := \left[0, \frac{2\pi}{N}, 2\frac{2\pi}{N}, \dots, (N-1)\frac{2\pi}{N}\right]$$

To achieve equal angular spacing of the satellites at the desired final time step T, we use the following inequality constraint in our optimization problem:

$$\|D \cdot \theta(T) - \Delta_{des}\| \leq \varepsilon_\theta$$

By constraining the maximum of the absolute values of these errors at time T to be less than or equal to some angular position error tolerance ε_θ, we can achieve approximately equal spacing. Similarly, by constraining the angular velocities of adjacent satellites to be approximately zero, we can ensure that the constellation will tend to remain equally spaced in the future.

$$\|D \cdot \omega(T)\| \leq \varepsilon_\omega$$

ε_ω is an angular velocity error tolerance close to zero. The last constraint missing is on the input, which we have to limit to the actual physical dimensions of the CubeSat. Planet Labs' constellation uses "bang-bang" inputs, where each satellite is either in maximum drag or minimum drag [3], [8]. In our proposal, we consider the satellite's drag to be any value between the minimum and maximum values.

$$U_{min} \leq U \leq U_{max}$$

To obtain a linear program, we must first precompute reference trajectories with the assumption that each satellite is under minimum drag input until final time step T. The reference trajectories are substituted in S^R and S^Ω so that the equations are linear but time-varying.

We estimate $r(T)$, $\omega(T)$ and $\theta(T)$ expressed in matrix form from:

$$r(T) = r(0) + \Delta t * \bar{S}^R * U$$
$$\omega(T) = \omega(0) + \Delta t * \bar{S}^\Omega * U$$
$$\theta(T) = \theta(0) + \Delta t * T * \omega(0) + \Delta t^2 * \bar{S}^\propto * U$$

$\bar{S}^R, \bar{S}^\Omega$ *and* $\bar{S}^\propto$ are matrices that are computed prior to solving carrying out the optimization. The methods for constructing these matrices can be found in [9]. The assumption of minimum drag for the open loop linear optimization results in the same reference trajectory for all satellites. For MPC with a decreasing horizon we recompute the reference trajectory for each individual satellite at each time step with the same assumption that the input will be under minimum drag until time step T. We considered implementing a reference trajectory computation in which only the first iteration uses the minimum drag assumption and all subsequent ones would use the previous iteration's solution. After gaining some experience with the system we quickly realized that our satellite drop

in altitude is small compared to the range of values we have for the atmospheric density and thus updating the reference trajectory with previous input commands will not have a profound effect on the solution.

By precomputing the reference trajectory, we can now state the problem in standard form:

$$\min_x f^T x$$

$$subject\ to\ \ Ax \leq b$$

Where $x = [U \quad t]^T$ and $f^T = [0^{1\times(N\cdot T)} \quad 1]$.

The matrices for the inequality constraints are:

$$A = \begin{bmatrix} -\Delta t * \bar{S}^R & -1^{N\times 1} \\ \Delta t^2 * D * \bar{S}^{\propto} & 0^{N\times 1} \\ -\Delta t^2 * D * \bar{S}^{\propto} & 0^{N\times 1} \\ \Delta t * D * \bar{S}^{\Omega} & 0^{N\times 1} \\ -\Delta t * D * \bar{S}^{\Omega} & 0^{N\times 1} \\ I^{(N*T)\times(N*T)} & 0^{N\times 1} \\ -I^{(N*T)\times(N*T)} & 0^{N\times 1} \end{bmatrix} \qquad b = \begin{bmatrix} r(0) \\ \varepsilon_\theta \cdot 1^{N\times 1} - D \cdot [\theta(0) + \Delta t \cdot T \cdot \omega(0)] + \Delta_{des} \\ \varepsilon_\theta \cdot 1^{N\times 1} + D \cdot [\theta(0) + \Delta t \cdot T \cdot \omega(0)] - \Delta_{des} \\ \varepsilon_\omega \cdot 1^{N\times 1} - D\omega(0) \\ \varepsilon_\omega \cdot 1^{N\times 1} + D\omega(0) \\ u_{max} \cdot 1^{(N\times T)\times 1} \\ -u_{min} \cdot 1^{(N\times T)\times 1} \end{bmatrix}$$

4.3 NONLINEAR OPTIMIZATION PROBLEM

In addition to replicating the results from [9], we wanted to carry out an original satellite optimization analysis of our own. After some consideration of the complexities of the orbital mechanics of the problem, we decided that an interesting and relatively straightforward extension would be to attempt the optimization without the linearization procedure. The objective is the same: to maximize the smallest orbital radius of the satellites at time T, (or to minimize the altitude loss of the satellites). The constraints are the same as well: that the angular separation of the satellites at time T be uniform to within a certain tolerance, and that the relative angular velocities be zero within a certain tolerance.

The problem can be formulated as

$$maximize \min_{i=1,\ldots,N} r_i(T)$$

$$subject\ to$$

$$\|D \cdot \theta(T) - \Delta_{des}\| \leq \varepsilon_\theta\,,$$

$$\|D \cdot \omega(T)\| \leq \varepsilon_\omega,$$

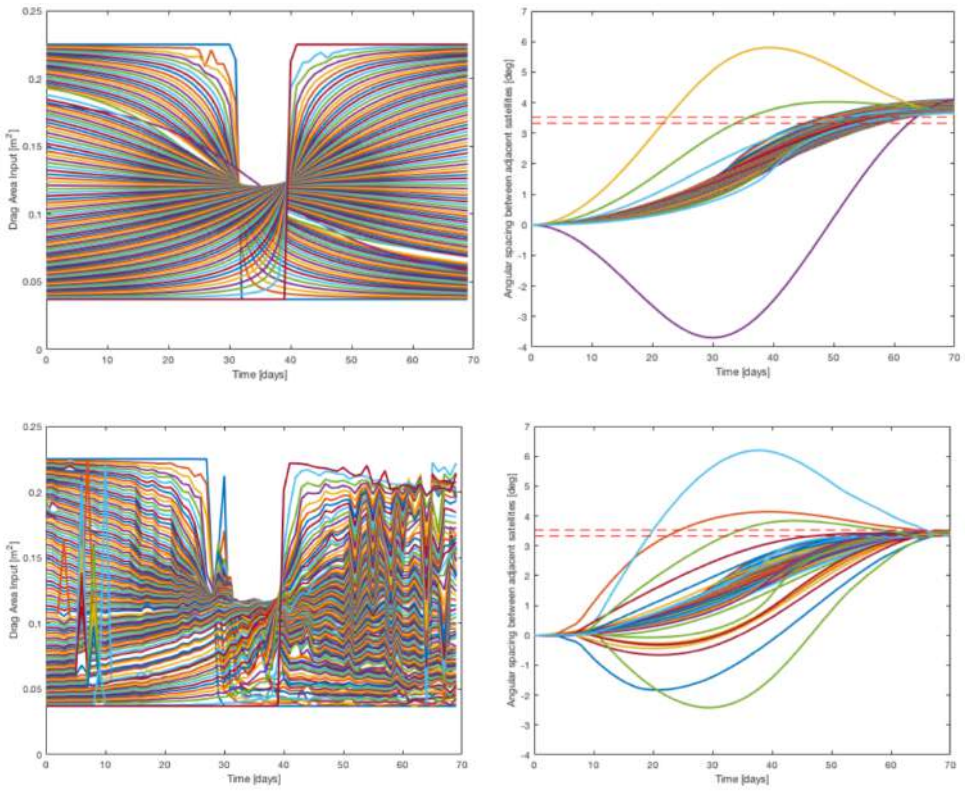
$$U_{min} \leq U \leq U_{max}$$

where now $r_i(T)$, $\theta(T)$, and $\omega(T)$ are nonlinear functions of the input U, computed directly from the discretization of the nonlinear dynamics. This nonlinearity eliminates guarantees of convexity and makes the computation substantially more expensive. The actual calculation was carried out in Matlab, using the fmincon optimization function with the same parameters used for the linear optimization.

5 RESULTS

5.1 ORIGINAL LINEAR PROGRAMMING RESULTS

The following figures display the original results initially determined in [9] and reprinted here for comparison. These simulations were run with N = 105 satellites and a horizon of T = 71 days, and will be used as a reference in the following sections.



From left to right, top to bottom:

Figure 1: Area Commands using Open-Loop Linear Programming

Figure 2: Satellite Spacing using Open-Loop Linear Programming

Figure 3: Area Commands using Closed-Loop Linear Programming

Figure 4: Satellite Spacing using Closed-Loop Linear Programming

5.2 OPEN-LOOP LINEAR PROGRAMMING RECREATION RESULTS

The first step in this project was to recreate the open-loop linear programming results of Sin's paper. This was completed successfully. Although there are some small differences in satellite spacing or area commands over time, overall our problem formulation delivers very similar results, with deviations likely resulting from imperfect information on the methods used by Sin in his original paper (such as approximations to the atmospheric density model).

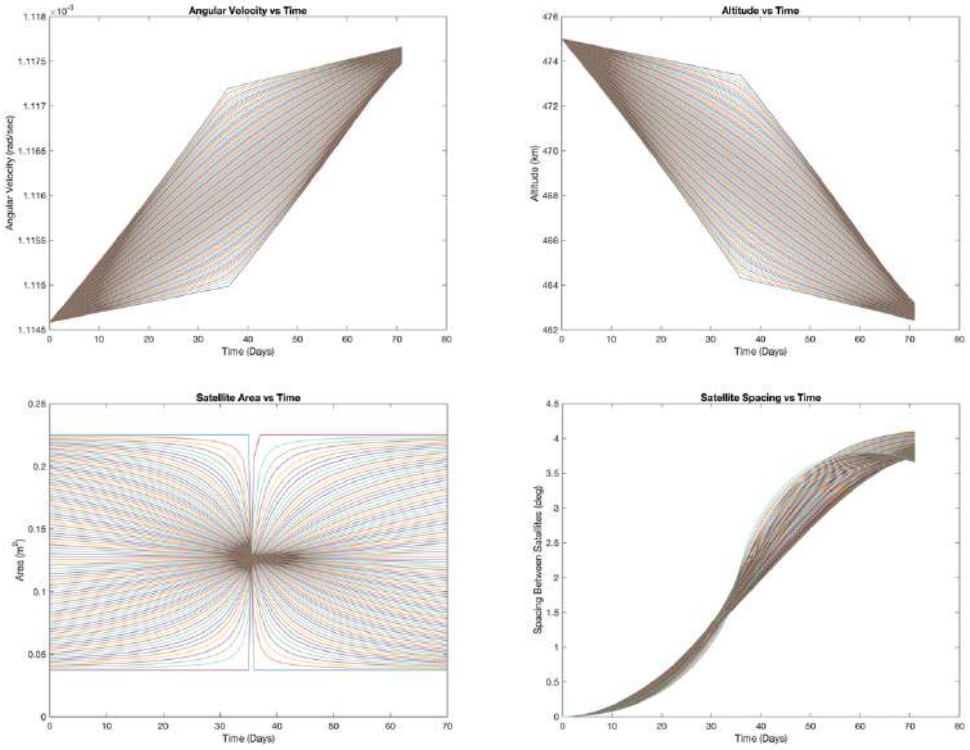


Figure 5: Replicated Open-Loop Linear Optimization

5.3 CLOSED-LOOP LINEAR OPTIMIZATION RECREATION RESULTS

After successfully recreating the open-loop linear optimization performed in [9], we then attempt to recreate this work's closed-loop linear optimization results. Overall, our algorithm displayed similar results. The altitude drop is reduced from 11.55 km to 11.16 km, and the final satellite spacing and velocities is within the constraints specified by the problem formulation. However, there are several significant differences. For example, **Error! Reference source not found.** shows the Sin's sequence of area commands, which display much more irregularities than the input commands computed by our algorithm, which are shown in Figure 6. Additionally, Sin's plots of satellite spacing over time (See Figure 2 and Figure 4) show that several pairs of satellites experience negative spacing. This behavior does not appear in our model. Finally, Sin's model-predictive control methods reduce the altitude drop from 11.64 km to 10.71 km (for an improvement of 0.93 km) vs our improvement of 11.55 km to 11.16 km (for an improvement of

0.39 km). It is not clear why these deviations exist. The authors failed to discuss their erratic behavior as well as the negative separation angles. Intuitively, one would not expect satellites to pass each other in the formation process. Thus the most reasonable explanation is that the authors of [9] made an undetected error in their algorithm. However we could have made an error in our algorithm, or that the original authors used additional information that was not made available in their paper.

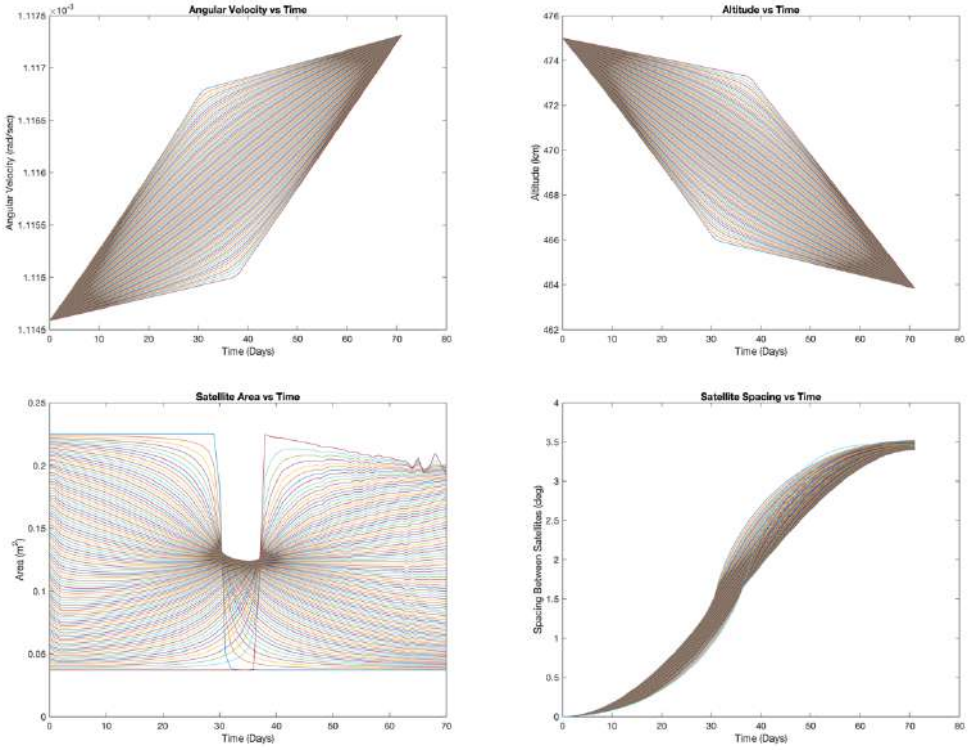
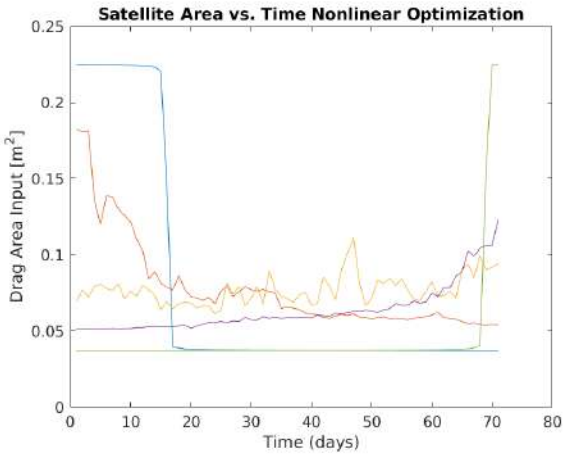
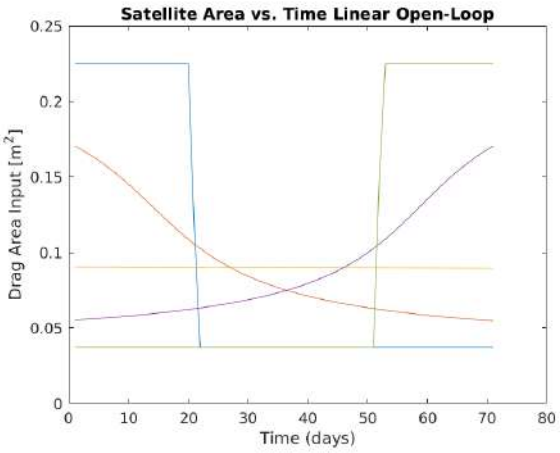


Figure 6: Replicated Closed-Loop Linear Optimization Results

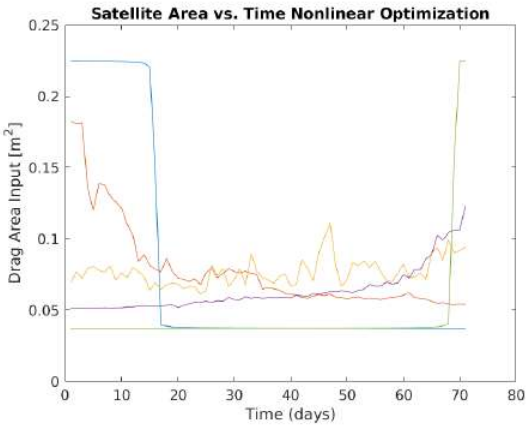
5.4 NONLINEAR OPTIMIZATION RESULTS

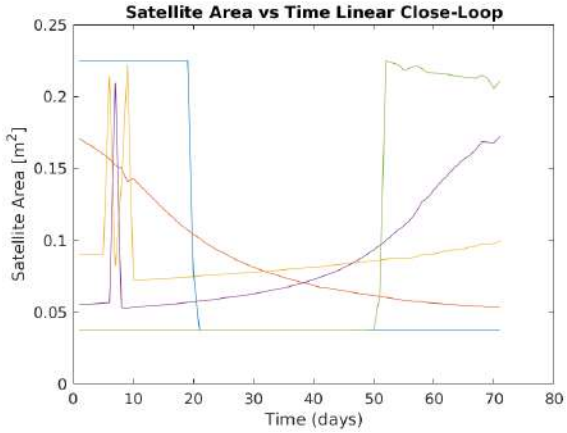
Without linearization, the optimization was much more computationally intensive. We found that for a timespan of T=71 days the largest number of satellites we could practically carry out the computation for was 5. Below is a comparison of the computed optimal commands for nonlinear optimization and the linear optimization:





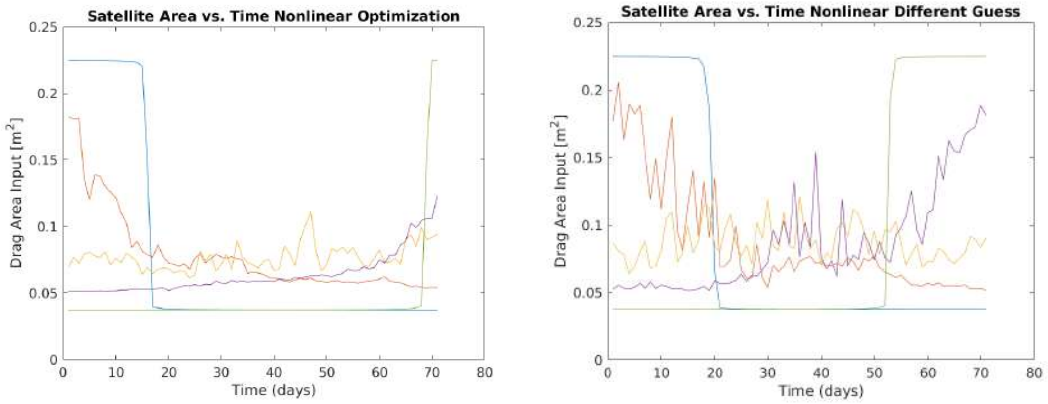
We can see some qualitative similarities, for example the fact that two of the satellites simply alternated swiftly from minimum area to the maximum (or vice versa) over the course of the trajectory, while the other three did something smoother, intermediate between maximum and minimum. Obviously, thought, the nonlinear optimization leads to a much noisier control input. A similar pattern can be seen in the comparison between the nonlinear optimization and the linear closed-loop optimization (MPC):





Again there is a qualitative similarity, but the nonlinear optimization gives more noise, however the linear closed-loop has some noisy fluctuations as well. It seems that when the nonlinearities of the problem are accounted for, the optimal control inputs become much messier and harder to interpret.

To get a sense of how unique the optimal solution is for the nonlinear case, we ran the optimization again but with a slightly different initial guess of the solution. The comparison is plotted below.



Both of these runs successfully satisfied the constraints, even though they are quite different in their details. The left run achieved an altitude drop about 1 km smaller than the right run. This suggests that the nonlinear problem does not in general have a unique solution, but that there are many local minima in the space of possible inputs. This also explains the spikiness seen in the linear closed-loop control.

Ultimately the most important comparison to make is how small of an altitude drop was achieved. In the above cases, the linear open-loop had a drop of 7.94 km, the linear closed-loop had a drop of 8.06 km, and the nonlinear had a drop of 7.13 km. Therefore the nonlinear optimization was able to achieve marginally better performance, by virtue of its more accurate model, at the cost of considerably more computation time.

5.5 COMPUTATION TIME COMPARISON

For this evaluation, we set T = 71 days and varied the number of satellites from N = 2 to N = 50 at increments of 5 satellites. After each iteration, we measured the elapsed time and altitude drop. The results of this experiment, which can be seen in Figure 7, show that the runtime complexity is approximately cubic, although it could be found to be a higher or lower-order polynomial if more data points were available.

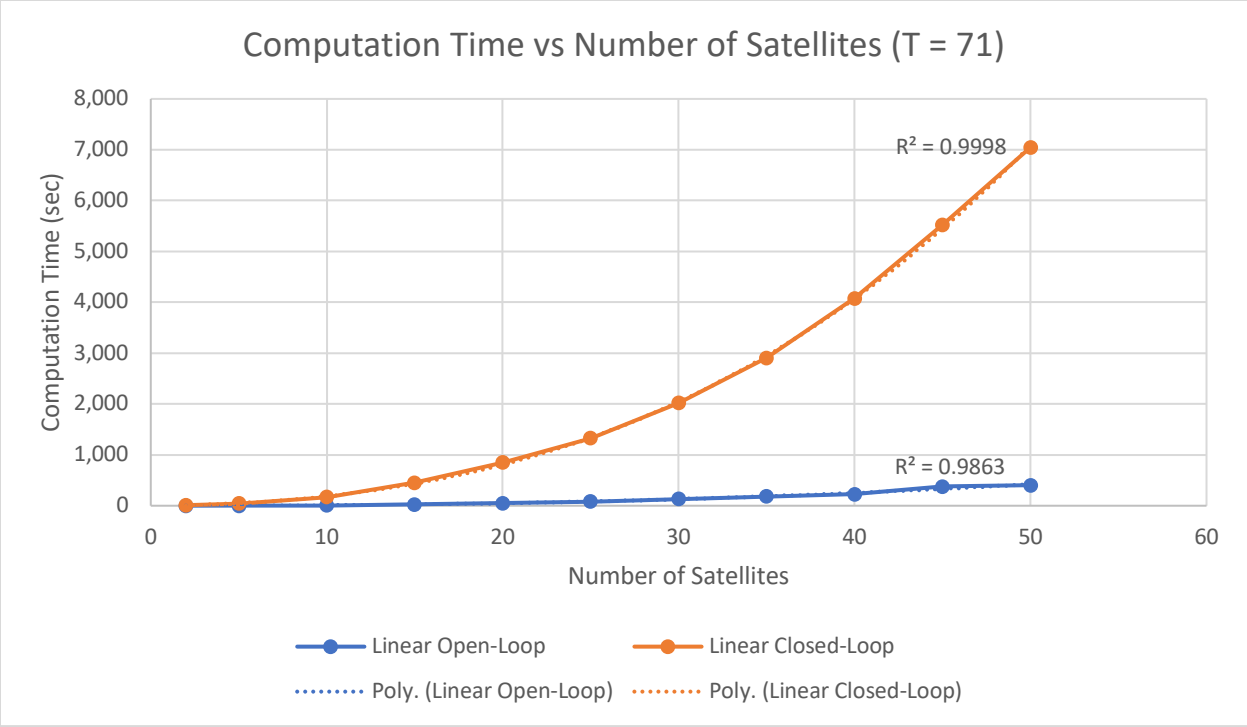


Figure 7: Computation Time vs Number of Satellites (T = 71)

6 CONCLUSION

We successfully recreated Sin's linear optimization as well as the feedback MPC with slight deviations from their results. Although we were not able to identify why our results differed from theirs, we suspect that this could be due to the atmospheric density model used in our optimization, which is poorly explained in their original paper. In addition to the density model, we assume that the reference trajectory is computed using the minimal drag assumption at each iteration and not the previously determined input commands. This project focuses on the optimization methods and procedures used for constellation separation, thus we chose to ignore the impacts of using different atmospheric models, time horizon, and reference trajectory computation. Although the orbital dynamics are nonlinear, we confirmed that the angular velocity of a satellite controlled by differential drag can be approximated as linear over the operating range of a day. Therefore, the solution from the linear program - even when applied in open-loop - provides a semi-equally spaced constellation. To improve the spacing of the constellation, a model predictive controller with the shrinking horizon approach was implemented. The MPC achieved an equally-spaced constellation that satisfied design tolerances. Finally, we found that the nonlinear program was able to converge to a solution, despite it being potentially problematic because of its nonconvexity. However, the computational cost was dramatically larger, preventing us from doing calculations for more than 5 satellites. For the N=5 case, the nonlinear optimization performed marginally better in terms of altitude drop, however the optimal control inputs were found to be sensitive to the initial guess of the optimization. This suggests that there are many local optima in the space of possible inputs.

7 CONTRIBUTIONS

Daniel and Jonathan originally began working together to find an optimization problem related to CubeSats. After Russell joined the group, he proposed a project related to Sin's paper, which we all agreed to. We then worked as a group to identify the goals of our project. Jonathan initially began work on this project by writing the dynamics and linear optimization code, including functions for performing the MPC optimization and generating the plots used in this paper. However, there were some initial bugs with this code, which Daniel and Russell helped resolve. After the linear programming was finished, Russell implemented optimization using nonlinear constraints. The majority of the final report (in particular the Orbital Dynamics and Model Generation section) and presentation were written by Daniel, with Jonathan and Russell contributing analysis for the results of their individual code contributions.

8 SOURCE CODE

The source code for this project can be found at:

https://github.com/jdwapman/SmallSatSeparation

9 REFERENCES

[1] L. Grush, "SpaceX wants to fly some internet satellites closer to Earth to cut down on space trash," *The Verge*, 09-Nov-2018. [Online]. Available: https://www.theverge.com/2018/11/9/18016962/spacex-internet-satellites-space-debris-trash-orbit-closer-earth-distance-atmosphere. [Accessed: 11-Nov-2018].

[2] M. Harris, "Tech giants race to build orbital internet [News]," *IEEE Spectr.*, vol. 55, no. 6, pp. 10–11, Jun. 2018.

[3] C. Foster, H. Hallam, and J. Mason, "Orbit Determination and Differential-drag Control of Planet Labs Cubesat Constellations," *ArXiv150903270 Astro-Ph Physicsphysics*, Sep. 2015.

[4] C. L. Leonard, W. M. Hollister, and E. V. Bergmann, "Orbital Formationkeeping with Differential Drag," *J. Guid. Control Dyn.*, vol. 12, no. 1, pp. 108–113, Jan. 1989.

[5] B. S. Kumar, A. Ng, K. Yoshihara, and A. D. Ruiter, "Differential Drag as a Means of Spacecraft Formation Control," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1125–1135, Apr. 2011.

[6] F. A. A. El-Salam and L. Sehnal, "Analytical second-order drag theory with application for lifetime determination of artificial satellites in low earth orbits," *Appl. Math. Comput.*, vol. 171, no. 2, pp. 948–971, Dec. 2005.

[7] J. Gangestad, B. Hardy, and D. Hinkley, "Operations, Orbit Determination, and Formation Control of the AeroCube-4 CubeSats," *AIAAUSU Conf. Small Satell.*, Aug. 2013.

[8] A. S. Li and J. Mason, "Optimal Utility of Satellite Constellation Separation with Differential Drag," in *AIAA/AAS Astrodynamics Specialist Conference*, 0 vols., American Institute of Aeronautics and Astronautics, 2014.

[9] E. Sin, M. Arcak, and A. Packard, "Small Satellite Constellation Separation using Linear Programming based Differential Drag Commands," *ArXiv171000104 Cs*, Sep. 2017.

[10] A. Blatner, "Optimal Differential Drag Control of Small Satellite Constellations," University of California, Berkeley, 2018.

[11] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*. California: Microcosm, 2013.