Optimal Differential Drag Control of Small Satellite Constellations

Andrew Blatner

August 10, 2018

Optimal Differential Drag Control of Small Satellite Constellations

by Andrew Blatner

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Murat Arcak
Research Advisor

(Date)

* * * * * * *

Professor Andrew Packard
Second Reader

(Date)

Abstract

Optimal Differential Drag Control of Small Satellite Constellations

by

Andrew Blatner

Master of Science, Plan II in Electrical Engineering and Computer Sciences

University of California, Berkeley

We analyze and improve two existing techniques for differential drag control of large constellations of small propulsion-less satellites. The system has two subproblems: determining the desired relative ordering of the satellites, referred to as their slotting, and generating an optimal sequence of control inputs to acquire the slotting. One technique, used in production by Earth-imaging company Planet, approximately solves both subproblems with simulated annealing with the objective of maximizing acquired imagery. The other technique, developed academically, ignores slot allocation and generates commands with linear programming, aiming to maximize constellation lifetime. First, we reconcile the practical details of both techniques. Then, we adapt the linear program to new models that better capture Planet's objective. Finally, we develop a fast method for slotting satellites deployed from a single launch. Though we find only small improvements in slot allocation, we provide assurances of optimality and significant improvements in solver time for command generation.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, my loving thanks to my parents for their constant support throughout my academic career, from pushing for opportunities for me in high school to helping me transition out of Berkeley, continually doing so even though I'm often too busy to take advantage of it. Thank you to Professor Murat Arcak for affording this opportunity, and to Emmanuel Sin for introducing me to this topic and mentoring me along the way. Thank you to Professor Andrew Packard for his course advice in undergrad before I ever seriously considered graduate school. Finally, I would like to thank everyone who has been part of EE 16B over the last 3 years, which kept me involved in the EECS community even when I felt tired of academics. Thank you to GSI's Nathan Mailoa, Emily Naviasky, and Saavan Patel for all their work in the first semester of the course, to Ana Shuler for her work as head lab TA before I took the role, and to Professors Michel Maharbiz and Michael Lustig. While there are too many other staff members and friends to list here, I would like to especially thank Rachel Zhang and Professors Elad Alon and Gireeja Ranade for getting me involved in the first place. Without EE 16B, I don't think I ever would have pursued graduate school.

Chapter 1

Introduction

As small satellites and low-Earth orbit launch capabilities are further commoditized, it is becoming more economical to launch large numbers of satellites at once. Constellations of many identical small satellites are well-suited for purposes such as communications and imaging, and they are often propulsion-less due to cost and engineering constraints. Planet Labs, Inc., an Earth-imaging company, has launched several constellations of such satellites, including Flock 3p, a constellation of 88 satellites launched in February 2017 [1].

Satellites launched together have similar initial orbits, so they require actuation to change their relative orbital phases. Differential drag utilizes the geometry and attitude control systems of satellites to control their aerodynamics. By varying the cross-sectional areas of the satellites, it controls the angular velocities and therefore the relative positions of the satellites. Differential drag is already used successfully by Planet and other organizations for satellite control.

Constellation acquisition can be divided into two components: slot allocation and command generation. The slot allocator first generates the desired ordering of satellites in their orbit by taking advantage of slight variations in initial trajectories, such as those produced by the deployment mechanism. Then, the command generator determines the drag inputs to optimally achieve the desired slotting. Most importantly, these two problems are tightly coupled, and an arbitrary slotting can add weeks or months to the acquisition time.

In [1], Planet solves both optimization problems using simulated annealing, an iterative method that probabilistically explores the search space of candidate slottings and inputs. Planet's command generator produces discrete high drag and low drag commands to minimize the L2-norm of the separation error, the deviation from the desired relative positions.

Another paper [2] formulates command generation as a linear program with continuous variables. As a result, instead of discrete commands, it uses any drag area between low and high drag. The LP minimizes the drop in altitude over the acquisition phase to maximize satellite lifetime. However, for the financial purpose of acquiring imagery, it may be more desirable to minimize acquisition time and total separation error.

This report seeks to reconcile and improve these approaches. First, we compare the discrete-time dynamics in [1] and [2] and verify that they are equivalent. We show that the continuous range of drag inputs can effectively be approximated as low drag and high drag inputs. Next, we define an ideal, though non-convex, objective function and approximate it with two new optimization models using the notation developed for the LP. We use this notation to formulate the L2-norm minimization as a quadratic program. Similarly, we formulate a new linear program to minimize the L1-norm of the separation errors, which more closely approximates the ideal objective function. Finally, we compare these developments with the system in [1] and develop a fast replacement for the slot allocator.

Chapter 2

Dynamics

2.1 Continuous-Time Dynamics

The orbit of each satellite is approximated as planar motion in polar coordinates because atmospheric drag predominately acts in-plane tangential to the orbit. The continuous-time dynamics for a single satellite are derived in [2] and are reproduced below.

$$\begin{aligned} \ddot{r} &= r\dot{\theta}^2 - \frac{\mu_E}{r^2} \\ \ddot{\theta} &= \frac{1}{r}\left(-2\dot{r}\dot{\theta} + (\vec{a}_{atmdrag})_\theta\right) \end{aligned} \tag{2.1}$$

2.2 Discrete-Time Dynamics

The discrete-time angular dynamics in the command generators are nearly equal. Though the LP formulation in [2] contains radial dynamics, they are not necessary with objective functions based on the angular dynamics. Both sets of dynamics use Euler's method for discretization, with the angular dynamics for each satellite given below in (2.2).

Parameters ρ and v are the atmospheric density and satellite velocity relative to the atmosphere, and a is the semimajor axis of the satellite's orbit. $\bar{\rho}$, $\bar{v}$, and $\bar{a}$ denote the corresponding predicted mean values. This report uses a simplified Harris-Priester atmospheric model [3], but it can easily be interchanged with Planet's high-fidelity models.

LP (Absolute Motion) [2]

$$\begin{aligned} \theta_{k+1} &= \theta_k + t_s\omega_k + 1/2 t_s^2 \bar{S}_k^{\Omega} A_k \\ \omega_{k+1} &= \omega_k + t_s \bar{S}_k^{\Omega} A_k \end{aligned}$$

$$\bar{S}^{\Omega}(r,\omega) \coloneqq \frac{3}{2}\frac{C_D}{m}\bar{\rho}(r)\bar{v}(r,\omega)^2\frac{1}{r}$$

$$\implies \ddot{\theta}_k = \bar{S}_k^{\Omega} A_k = \frac{3}{2}\frac{C_D}{m}\overline{\rho v^2}\frac{1}{r}A_k$$

Planet (Relative Motion) [1]

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}_k + \begin{bmatrix} 0 \\ B_k(u_k) \end{bmatrix} \tag{2.2}$$

$$B_k(u_k) \coloneqq \ddot{\theta}_{i,k} u_{i,k} - \ddot{\theta}_{\text{ref},k} u_{\text{ref},k}$$

$$u_{i,k} \coloneqq \begin{cases} 0 & \text{if } i \text{ is in low drag at time } k \\ 1 & \text{if } i \text{ is in high drag at time } k \end{cases}$$

$$\ddot{\theta}_k \coloneqq \frac{3}{2}\frac{C_D}{m}\overline{\rho v^2}\frac{1}{\bar{a}}(A_{\max} - A_{\min}) \tag{2.3}$$

For absolute motion, $\ddot{\theta}_k$ denotes absolute angular acceleration. For relative motion, $B(u)$ denotes angular acceleration relative to the reference satellite, and $\ddot{\theta}_k$ represents the available *control authority.* This is defined as the difference in angular acceleration between the maximum and minimum drag configurations. For absolute motion, this is computed as in (2.4). Because $r \approx \bar{a}$ for the nearly circular orbits of these satellites, the control authorities in both formulations are identical.

$$\ddot{\theta}_{\max} - \ddot{\theta}_{\min} = \frac{3}{2}\frac{C_D}{m}\rho v^2\frac{1}{r}(A_{\max} - A_{\min}) \tag{2.4}$$

However, there is a discrepancy in the dynamics obscured by these equations. In [1], initial conditions are calculated by fitting a line through a day of orbital data, effectively approximating the *mean motion* of the orbits. Mean motion is the orbit's mean angular velocity, $n = \frac{2\pi}{P}$, and is equal for all elliptical orbits with the same period [4, p. 131]. Most importantly, it determines the evolution of long-term relative angular separation. For example, two satellites with the same period may have different instantaneous angular velocities, but their average angular separation remains constant. In contrast, [2] assumes circular orbits and uses the instantaneous state, so MPC simulation fails when initialized with non-circular orbits. Even though the largest initial eccentricity for Planet's Flock 3P is only 0.0012 [4, p. 132, 5], the optimization problem becomes infeasible after a small number of time-steps.

Fortunately, $\bar{S}^{\Omega}$ is derived from the rate of change of an orbit's mean motion [4, pp. 561, 566], so resolution of this discrepancy is simple: for all discrete-time calculations, including simulation and command generation, use "circularized" orbits as the initial conditions. With optimization based only on angular separations, this amounts to replacing the instantaneous angular velocity ω with the mean motion n. Continuous-time simulations still use the original eccentric orbits. From an instantaneous position r and (non-angular) velocity v, such as those given in Planet's orbital ephemerides [5], mean motion is calculated by finding the semimajor axis with the vis-viva equation [4, pp. 109–110]:

$$n = \sqrt{\frac{\mu}{a^3}} \quad \text{where} \quad a = \frac{\mu r}{2\mu - v^2 r}$$

Figure 2.1 shows that this change is necessary. In open-loop simulation without circularization, the angular separations do not follow the model's predictions. In closed-loop MPC simulations, the problem quickly becomes infeasible without significantly extending the horizon.

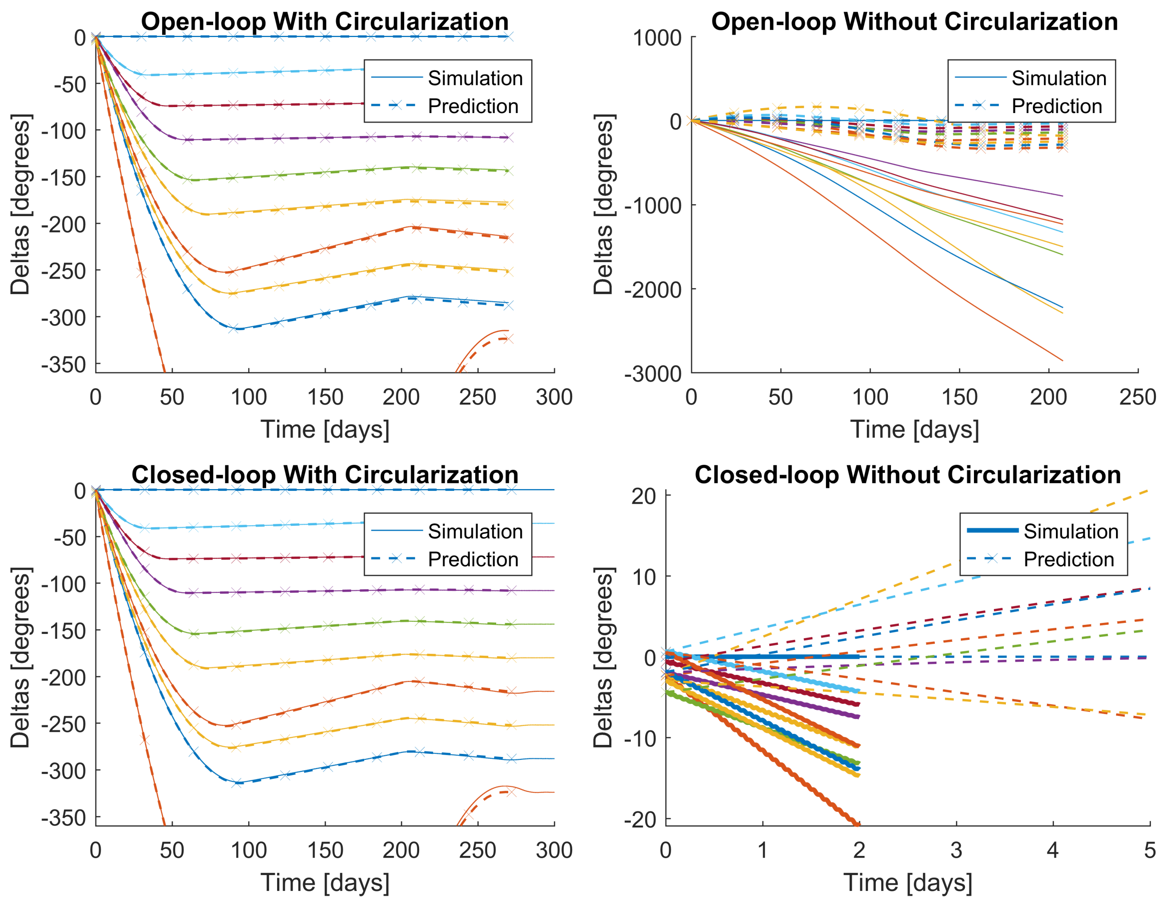


Figure 2.1: Impact of "circularizing" discrete-time orbits

2.3 Discretizing Input Commands

In practice, Planet uses binary control inputs of maximum and minimum drag areas, resulting in an intractable binary integer programming problem. Their methods in [1] circumvent this with simulated annealing at the cost of sub-optimality, while the LP formulation in [2] relaxes the control inputs to a continuous interval. In order to make the optimization-based approach applicable, each control input $A_{\text{des}} \in [A_{\min}, A_{\max}]$ must be approximated by binary control inputs $A_{\min}$ and $A_{\max}$.

Over a single time-step, A_{des} can be approximated by applying $A_{\max}$ for part of the time-step, and $A_{\min}$ for the remainder of each time-step. The length of each time-step, t_s, is typically 1 day. The transition time, $\tau \in [0, t_s]$, is chosen to impart an equal amount of mechanical impulse as is imparted by applying A_{des} for the entire time-step. The imparted force is linear in the drag area, so equivalently the amount of applied control is $u_{\text{des}} = A_{\text{des}} t_s$. This can be partitioned as in (2.5), and by solving, we find the required transition time τ as given in (2.6).

$$u_{\text{des}} = A_{\text{des}} t_s = A_{\max}\tau + A_{\min}(t_s - \tau) \tag{2.5}$$

$$\implies \tau = \frac{A_{\text{des}} - A_{\min}}{A_{\max} - A_{\min}} t_s \tag{2.6}$$

Evaluating the response of a satellite to this adapted input over a single time-step:

$$\omega(\tau) = \omega(0) + \tau \bar{S}^{\Omega} A_{\max} \qquad \theta(\tau) = \theta(0) + \tau\omega(0) + \frac{1}{2}\tau^2 \bar{S}^{\Omega} A_{\max}$$

$$\omega(t_s) = \omega(\tau) + (t_s - \tau)\bar{S}^{\Omega} A_{\min} \qquad \theta(t_s) = \theta(\tau) + (t_s - \tau)\omega(\tau) + \frac{1}{2}(t_s - \tau)^2 \bar{S}^{\Omega} A_{\min}$$

Making the appropriate substitutions:

$$\omega(t_s) = \omega(0) + t_s \bar{S}^{\Omega} A_{\text{des}} \qquad \theta(t_s) = \theta(0) + t_s\omega(0) + \frac{1}{2}t_s^2 \bar{S}^{\Omega} A_{\text{eff}} \tag{2.7}$$

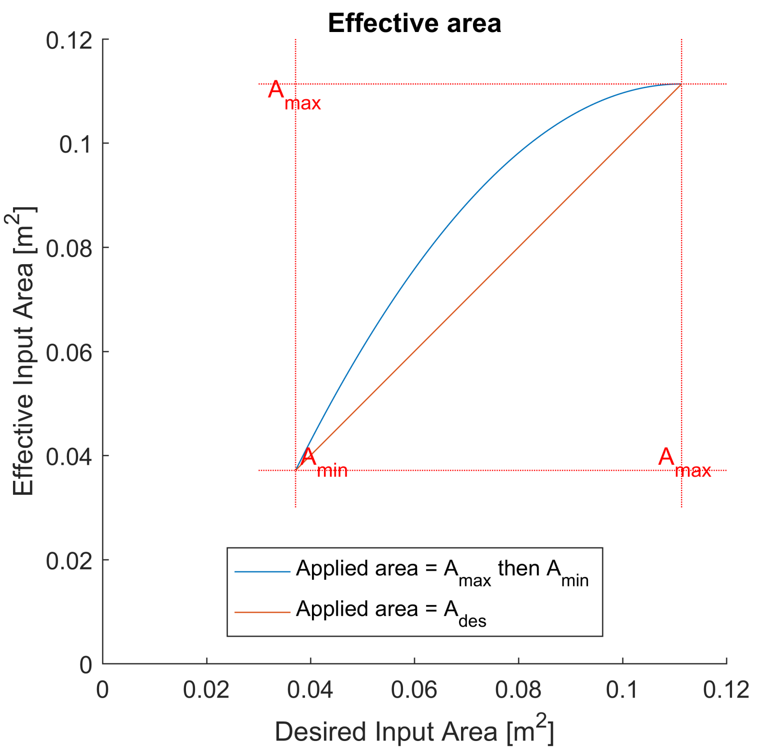
$$\text{where } A_{\text{eff}} = \frac{-(A_{\text{des}} - A_{\text{min}})^2}{A_{\text{max}} - A_{\text{min}}} + 2(A_{\text{des}} - A_{\text{min}}) + A_{\text{min}}$$

The nominal response to A_{des} is simply:

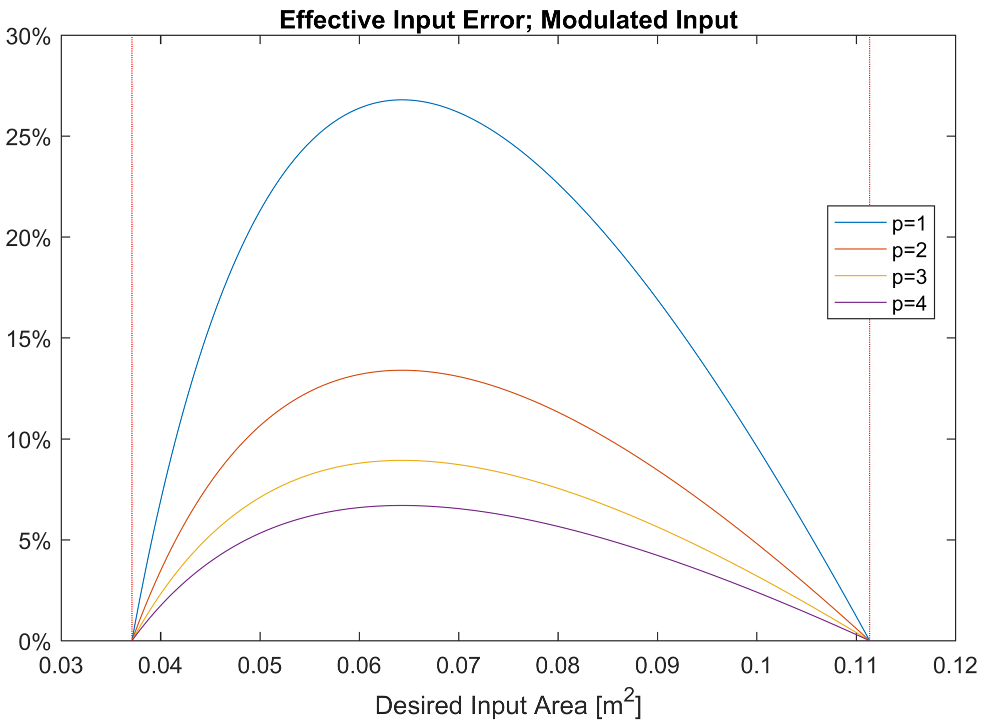
$$\bar{\omega}(t_s) = \omega(0) + t_s \bar{S}^{\Omega} A_{\text{des}} \qquad \bar{\theta}(t_s) = \theta(0) + t_s\omega(0) + \frac{1}{2}t_s^2 \bar{S}^{\Omega} A_{\text{des}} \tag{2.8}$$

Comparing (2.7) with (2.8), the angular velocity is identical because an equal impulse is applied to the satellite. The angular position response of the approximate input is equivalent to a constant input of A_{eff}, plotted in Figure 2.2a as a function of A_{des}.

Pulse-width modulating the input better approximates the desired input by dividing each time-step into p periods of length $\frac{t_s}{p}$. Then, high-drag is applied each period for a duration of $\frac{\tau}{p}$. The reduction of the effective input error with this method is shown in Figure 2.2b. Figure 2.3 illustrates that this approximation is accurate using two sets of initial conditions.

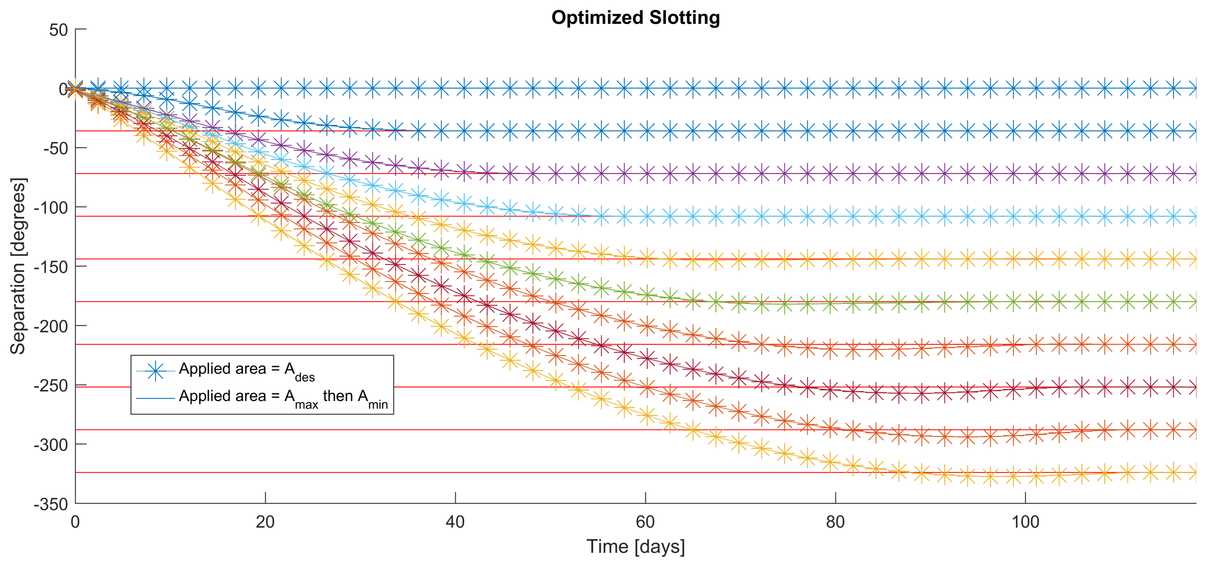


(a)



(b)

Figure 2.2: Approximating a desired area by modulating high-drag and low-drag



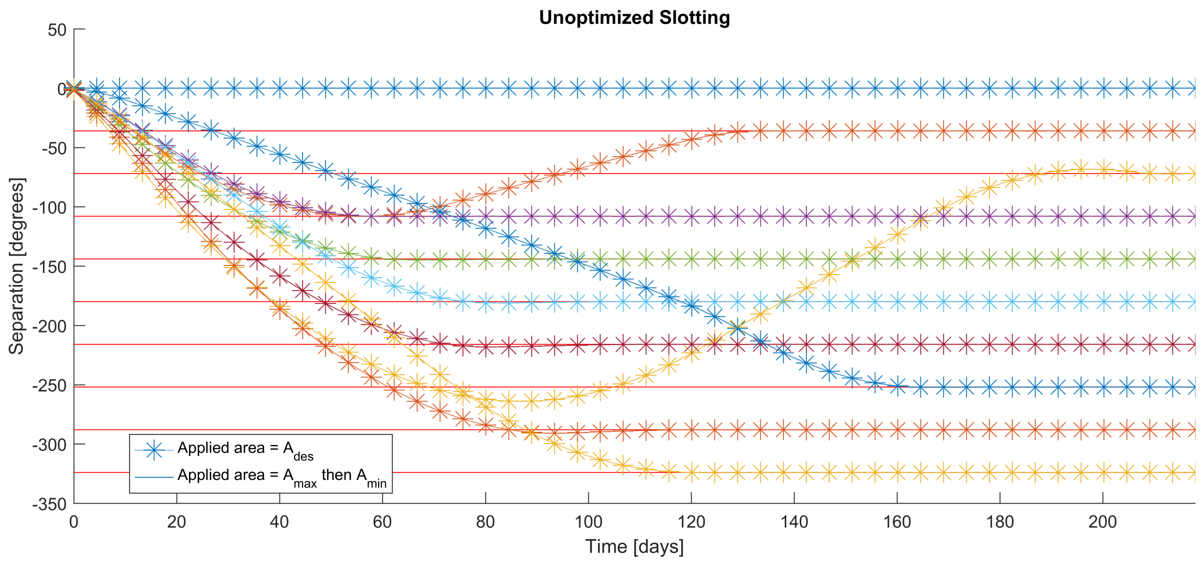


Figure 2.3: Simulated closed-loop performance with input approximation

Chapter 3

Optimization Models for Command Generation

The two methods for command generation have different objectives. For-profit companies, such as Planet, may be interested in maximizing the amount of imagery available for customer-facing services by minimizing acquisition time as in [1]. In contrast, the linear program in [2] maximizes the lifetime of the constellation by minimizing altitude loss during acquisition. While this could increase the amount of imagery a few years after deployment, acquiring full coverage of the Earth as quickly as possible may be more practically desirable.

In both formulations, there is a notion of desired angular separations between satellites. In [2], angular separations are measured between sequentially numbered satellites, while [1] computes separations relative to a single reference satellite, the choice of which is part of the slot allocation problem. Defining the angular position vector as $\theta = [\theta_1, \ldots, \theta_N]^\top$, angular separations can be expressed as $\Delta = \mathrm{D} \cdot \theta$, and the *angular separation error* is $\Delta - \Delta_{\text{des}} = \mathrm{D} \cdot \theta - \Delta_{\text{des}}$. The definitions for the separation matrices D and the desired angular separations Δ_{des} are given in (3.1), though [1] does not use this notation. The row of zeros in the relative separation matrix serves to keep the dimensions of D the same. The LP in [2] constrains the angular separation error to be near 0 at the final time-step T, while [1] minimizes its L2-norm over a horizon of T days.

$$\begin{array}{cc} \text{Sequential Separations} & \text{Relative Separations} \\ \mathrm{D} := \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ -1 & & & 1 \end{bmatrix} \in \mathbb{R}^{N \times N} & \mathrm{D} := \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ -1 & 1 & & \\ \vdots & & \ddots & \\ -1 & & & 1 \end{bmatrix} \in \mathbb{R}^{N \times N} \end{array} \tag{3.1a}$$

$$\Delta_{\text{des}} := [\frac{2\pi}{N}, \ldots, \frac{2\pi}{N}, -\frac{2\pi}{N}(N-1)] \qquad \Delta_{\text{des}} := [0, \frac{2\pi}{N}, 2 \cdot \frac{2\pi}{N} \ldots, (N-1) \cdot \frac{2\pi}{N}] \tag{3.1b}$$

First, we define an ideal objective function and formulate the L2-norm minimization as a quadratic program. Afterwards, we formulate a new linear program to approximate the ideal objective function. We experimentally show that using a single reference satellite performs better than calculating angular separations between sequential satellites, and lastly we explain some implementation details.

3.1 Primary Objective Function

A measure for the instantaneous quality of a constellation is the total angle spanned by (usually overlapping) intervals of width $\frac{2\pi}{N}$, centered on each of N satellites. While N

satellites may not be able to image the entire Earth without gaps, this measure is beneficial in that it is optimal for evenly spaced satellites and can be calculated quickly without simulation. With the angular position of satellite i at time-step $k \in 0 \ldots T$ denoted by $\theta_i(k)$, we can define the instantaneous *coverage error* as in (3.2a). We get an overall cost function by summing over time as in (3.2b). Assuming that a perfectly spaced constellation is capable of imaging the Earth once per day, we can interpret this as "total Earth surfaces not imaged." While this objective function can be quickly algorithmically calculated, it is non-convex in $\theta_i(k)$ and does not lend itself to optimization models such as linear or quadratic programs.

$$J_{\text{ideal}}(k) = 1 - \frac{1}{2\pi}\text{measure}\left\{\bigcup_{i=1}^{N}\left[\theta_i(k) - \frac{\pi}{N}, \theta_i(k) + \frac{\pi}{N}\right] \bmod 2\pi\right\} \tag{3.2a}$$

$$J_{\text{ideal}} = \sum_{k=0}^{T} J(k) \cdot t_s \tag{3.2b}$$

3.2 Quadratic Program Formulation

The L2-norm minimization in [1] naturally lends itself to formulation as a quadratic program after relaxing the inputs from binary to continuous variables. The angular positions at time k are given in (3.3) using the formulas and notation developed in [2]. Row vectors $\bar{S}_i^{\alpha}(1:k)$ represent the effect of the inputs on satellite i through time-step k, and the notation $\circ$ in

(3.5) is the element-wise (Hadamard) product.

$$\theta(k) = \theta(0) + t_s k\omega(0) + t_s^2 \bar{S}^\alpha(1:k) \cdot U \tag{3.3}$$

$$\text{where} \quad \bar{S}^\alpha(1:k) = \begin{bmatrix} \bar{S}_1^\alpha(1:k) & & \\ & \ddots & \\ & & \bar{S}_N^\alpha(1:k) \end{bmatrix} \in \mathbb{R}^{N\times NT} \tag{3.4}$$

$$\bar{S}_i^\alpha(1:k) = \left\{(k - \tfrac{1}{2}) \cdot \mathbf{1}^{1\times k} - [0,\ldots,(k-1)], \mathbf{0}^{1\times T-k}\right\} \circ \bar{S}_i^\Omega \in \mathbb{R}^{1\times T} \tag{3.5}$$

Then the angular separation errors at time k are:

$$\begin{aligned} \Delta(k) - \Delta_{\text{des}} &= D \cdot \theta(k) - \Delta_{\text{des}} = Q_k U + R_k \\ Q_k &= D \cdot t_s^2 \bar{S}^\alpha(1:k) \in \mathbb{R}^{N\times NT} \\ R_k &= D \cdot (\theta(0) + t_s k\omega(0)) - \Delta_{\text{des}} \in \mathbb{R}^{N\times 1} \end{aligned} \tag{3.6}$$

The L2-norm objective function in [1] can then be formulated as:

$$\begin{aligned} J_{\text{QP}}(U) &= \sum_{k=1}^{T} \|D \cdot \theta(k) - \Delta_{\text{des}}\|_2^2 \\ &= \sum_{k=1}^{T} \frac{1}{2} U^\top (2Q_k^\top Q_k) U + 2R_k^\top Q_k U + R_k^\top R_k \end{aligned} \tag{3.7}$$

Then, with Q_k, R_k as given in (3.6), the quadratic program can be expressed in standard form as in (3.8). The inequality constraints in (3.8c) are the same as the constraints in the LP formulation from [2] with the radius constraints and slack variables removed.

$$\begin{aligned} \underset{U \in \mathbb{R}^{NT \times 1}}{\text{minimize}} \quad & J_{\text{QP}}(U) = \frac{1}{2} U^\top H U + f^\top U + C \\ \text{subject to} \quad & A_{\text{QP}} \cdot U \leq b_{\text{QP}} \end{aligned} \tag{3.8a}$$

$$H = \sum_{k=1}^{T} 2 Q_k^\top Q_k \qquad f^\top = \sum_{k=1}^{T} 2 R_k^\top Q_k \qquad C = \sum_{k=1}^{T} R_k^\top R_k \tag{3.8b}$$

$$A_{\text{QP}} = \begin{bmatrix} Q_T \\ \text{-}Q_T \\ D \cdot t_s \bar{S}^{\Omega} \\ \text{-}D \cdot t_s \bar{S}^{\Omega} \\ \mathbb{I}_{NT \times NT} \\ \text{-}\mathbb{I}_{NT \times NT} \end{bmatrix} \qquad b_{\text{QP}} = \begin{bmatrix} \epsilon_\theta \cdot \mathbf{1}^{N \times 1} - R_T \\ \epsilon_\theta \cdot \mathbf{1}^{N \times 1} + R_T \\ \epsilon_\omega \cdot \mathbf{1}^{N \times 1} - D \cdot \omega(0) \\ \epsilon_\omega \cdot \mathbf{1}^{N \times 1} + D \cdot \omega(0) \\ A_{\max} \mathbf{1}^{NT \times 1} \\ \text{-}A_{\min} \mathbf{1}^{NT \times 1} \end{bmatrix} \tag{3.8c}$$

Quadratic Program Implementation

Formulation of the QP using the definitions in (3.8b) is notably slow due to constructing H with T large matrix additions and multiplications. Testing an implementation with a horizon of 160 days and time-step of 4 days ($T = 40$), the formulation time is typically over twice the solver runtime, with respective averages of 5.2 seconds and 2.4 seconds. The formulation time drastically decreases to only 0.5 seconds by constructing each of H and f with a single matrix multiplication as in equation (3.9), reducing the overall time from 7.6

to 2.9 seconds.

$$\begin{aligned} H &= 2Q^\top Q \\ f^\top &= 2R^\top Q \end{aligned} \qquad Q = \begin{bmatrix} Q_1 \\ \vdots \\ Q_T \end{bmatrix} \qquad R = \begin{bmatrix} R_1 \\ \vdots \\ R_T \end{bmatrix} \tag{3.9}$$

The simulations used for this report are constructed using units of kilometers. Though convenient in some ways, this choice results in scaling and numerical issues illustrated in table 3.1. Some elements of H are greater than 10^{15}, yet the decision variables are only on the order of 10^{-7}. Additionally, while H should be symmetric and positive-semidefinite, some of the zero-eigenvalues become negative due to numerical inaccuracies. The negative eigenvalues are significant enough for `quadprog` to determine the QP to be non-convex. These issues are resolved by scaling U and its bounds by $2^{20} (\approx 10^6)$ and inversely scaling $\bar{S}^\Omega$ by 10^6, thereby scaling H by 10^{-12} via Q_k and $\bar{S}^\alpha$ and converting the units to meters.

	U	$\max(\lvert H_{i,j}\rvert)$	$\min(\text{eig}(H))$	$\max(\text{eig}(H))$
Without scaling	1e−7	1e17	−293	1e18
With scaling	1e−1	1e5	−1e−10 ≈ 0	1e6

Table 3.1: Magnitudes of values in the quadratic programs

3.3 New Linear Program Formulation

Another potential optimization model is a new linear program to minimize the L1-norm of the angular separation error, now considered for a few reasons. First, in the expression for coverage error in (3.2), the cost of overlapping intervals is linear in the amount of overlap, just as the L1-norm is linear in the error. Additionally, L2-norm minimization harshly penalizes a constellation for a single satellite far out of position, even if all the other satellites are in the correct position. L1-norm minimization properly penalizes this the same as a constellation with each satellite slightly out of position. Lastly, a linear program may be faster to solve in practice. The initial objective function is:

$$J_{\text{LP}}(U) = \sum_{k=1}^{T} \|\Delta(k) - \Delta_{\text{des}}\|_1 = \sum_{k=1}^{T} \|Q_k U + R_k\|_1$$

We can bound the L1-norm with slack variables to convert it to an LP:

$$0 \leq \|Q_k U + R_k\|_1 \leq t_k \in \mathbb{R}^{N\times 1} \iff \begin{array}{r} Q_k U - \mathbb{I}_N t_k \leq \text{-}R_k \\ \text{-}Q_k U - \mathbb{I}_N t_k \leq \ \ R_k \end{array}$$

$$\implies J_{\text{LP}}(U) = \min_t J_{\text{LP}}(U,t) = \sum_{k=1}^{T} t_k \quad \text{such that} \quad \begin{array}{r} QU - \mathbb{I}_{NT} t \leq \text{-}R \\ \text{-}QU - \mathbb{I}_{NT} t \leq \ \ R \end{array} \tag{3.10}$$

Then, we can formulate the LP in standard form by augmenting the inequality constraints of the QP in (3.8c) with the new constraints in (3.10).

$$\begin{aligned} \underset{U,t\in\mathbb{R}^{NT\times 1}}{\text{minimize}} \quad & J_{\text{LP}}(U,t) = \mathbf{1}^{1\times NT} t \\ \text{subject to} \quad & A_{\text{LP}} \cdot x \leq b_{\text{LP}}, \quad x = \left[\begin{array}{c} U \\ \hdashline t \end{array}\right] \in \mathbb{R}^{2NT\times 1} \end{aligned} \tag{3.11a}$$

$$A_{\text{LP}} = \left[\begin{array}{c:c} A_{\text{QP}} & \mathbf{0}^{4N\times NT} \\ \hdashline Q & \text{-}\mathbb{I}_{NT} \\ \text{-}Q & \text{-}\mathbb{I}_{NT} \end{array}\right] \qquad b_{\text{LP}} = \left[\begin{array}{c} b_{\text{QP}} \\ \hdashline \text{-}R \\ R \end{array}\right] \tag{3.11b}$$

3.4 Simulation Details

Horizon Length

Before formulating either the new LP or QP, we must choose a horizon length. The feasibility of a given horizon length can be quickly determined using `linprog` and the minimal constraints ($A_{\mathrm{QP}}U \leq b_{\mathrm{QP}}$) without an objective function. Then the minimum horizon can be found with a binary search, assuming that the problem is feasible for all horizons greater than the minimum. This may not usefully compare slottings of similar minimum horizons, but in general horizon length correlates with total coverage error.

Time-Step

These optimization models must be quick to solve for the purposes of experimentation and simulation. Solver time is closely correlated with problem size, which is primarily determined by the number of satellites and the number of time-steps. The number of satellites is fixed, so the primary way to decrease the problem size is by increasing the time-step. The control period is mainly an issue in the closed-loop performance as the satellites approach the desired separations and overshoot is a possibility. However, this is not an issue because slotting is concerned with open-loop performance and MPC control is used in practice. Instead, the primary constraint is the maximum control authority for most of the acquisition maneuver when the satellites are still separating from each other.

3.5 Evaluation

Next, we compare each of the four combinations of optimization models and separation matrices using initial conditions similar to those of Flock 3p [5]. Figure 3.1 illustrates that computing separations from a reference satellite has significantly lower coverage error than sequentially computing separations. Coverage errors are expressed relative to reference-based slotting with the QP. The QP and new LP usually perform similarly for optimized slottings, but in some instances the coverage error with the new LP is 10% lower. For unoptimized slottings, the LP significantly outperforms the QP. Figure 3.2 clearly shows that the new LP is much faster to solve.

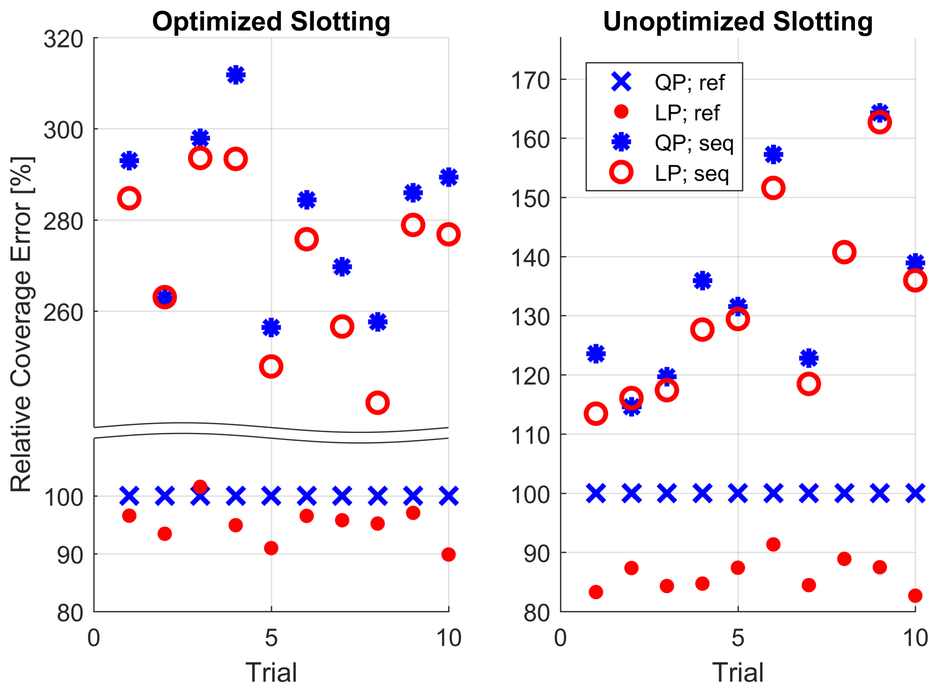


Figure 3.1: Coverage errors of each optimization model.
Plotted relative to the model most similar to [1]: the QP with reference-based separations.
Slottings optimized using method in section 4.2

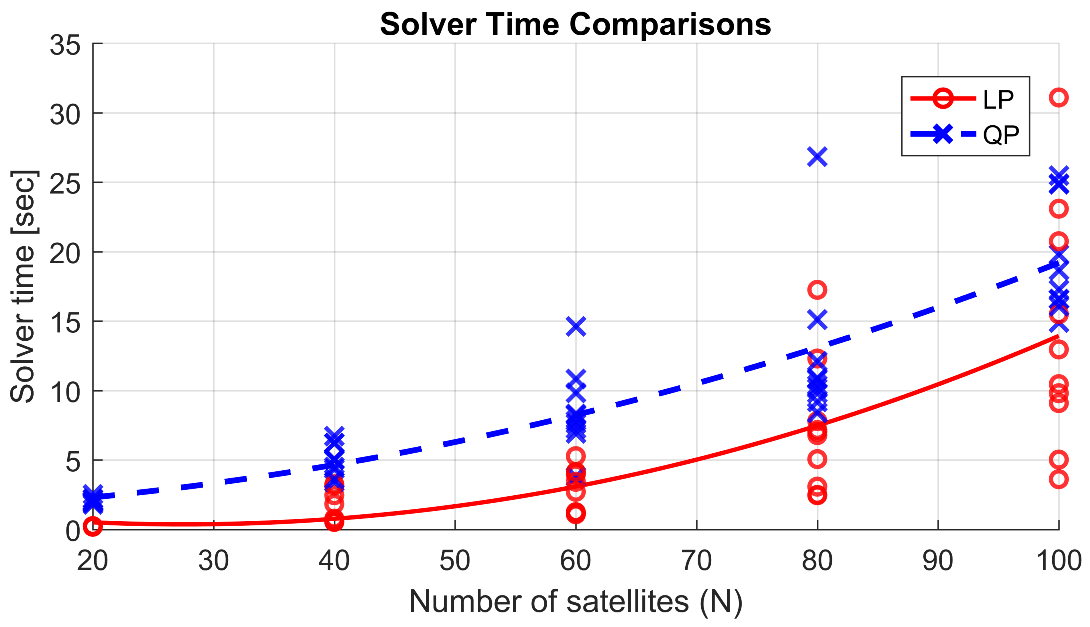


Figure 3.2: Runtimes of each solver

Chapter 4

Comparison with Planet's Methods

Planet publicly provides orbital data of their satellites each day. Throughout this chapter, simulations and evaluations of slot allocation and command generation are initialized using the states of Planet's 88 Flock 3p the day after their launch, February 15th, 2017 [5].

4.1 Planet's Slot Allocator

The slot allocator in [1] uses simulated annealing to iteratively minimize the phasing time of the constellation. At each iteration, the optimizer swaps the slots of two satellites and evaluates a heuristic for the phasing time. This heuristic calculates the maximum pairwise phasing time, notated $\Delta t = \max_i \Delta t_i$ of a reference satellite and satellites $i = 1 \ldots N - 1$. Assuming time-invariant control authority, Δt_i can be analytically computed with simple kinematics.

While this heuristic has low complexity and computational cost, it does not capture the likely conflict between pairwise-optimal inputs. For example, given satellites 1, 2, and 3, satellites 1 and 2 may phase optimally if 1 high-drags first, but satellites 1 and 3 may phase optimally if 1 high-drags second. Due to these conflicts, the true optimal global phasing time must be longer, perhaps significantly so, than the Δt returned by the slot allocator. It may seem that this heuristic is too restrictive, but it has the benefit of maximizing available time, beyond the maximum Δt, for resolving the control conflicts.

Table 4.1 shows the results of multiple trials of this slot allocator using a single initial condition and different random seeds. The results of each trial are nearly identical, indicating that a similar local minimum is reached with each run. As described in section 3.1, "coverage error" can be interpreted as Earth surface areas not imaged over the course of constellation acquisition.

Δt [days]	QP Coverage Error	LP Coverage Error	Min Horizon [days]
96.576	50.38	48.777	116
96.576	49.505	48.338	116
96.576	49.058	48.308	116
96.576	50.097	48.078	116
96.576	49.234	48.208	116

Table 4.1: Multiple trials of Planet's Slot Allocator. For each returned slotting: final Δt, coverage errors from the QP and new LP, and minimum horizon length

Figure 4.1 depicts the iterative performance of the slot allocator in the first trial. Notably, almost all of the progress is made in the first 2% of the 1e6 iterations, and none is made in the hatched area. This implies that slot allocation can be done with far fewer iterations and may be a simpler problem than is posed.

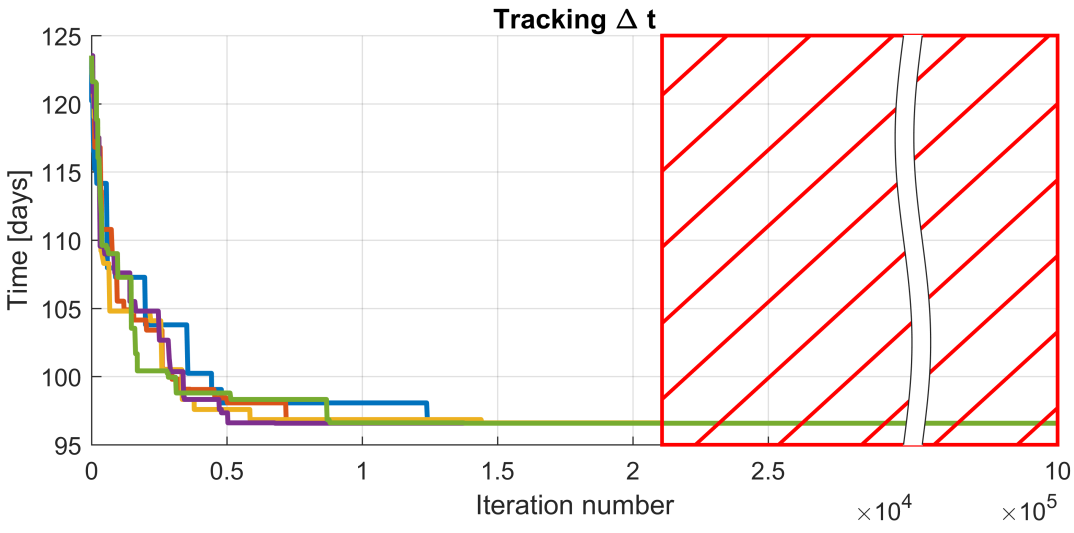


Figure 4.1: Iterative performance of Planet's slot allocator

4.2 Optimal Slotting for Single-Launch Constellations

Now we develop a quick method for computing the optimal slotting of satellites clustered from a single launch. Using the notion of pairwise phasing times, we can convert the two-dimensional state of angular position and mean motion into a one-dimensional measure of along-orbit forward progress.

There is a one-to-one mapping between semimajor axis length and mean motion so they are used interchangeably. Lower altitude satellites orbit more quickly, so given two satellites with the same angular position, the lower satellite is conceptually considered farther along the orbit. Comparisons are more complex for satellites with different angular positions: with an infinite horizon, the lower altitude satellite eventually passes the other, but in this context we have a limited horizon. Ultimately, this effect will be negligible for satellites from the same launch, such as Flock 3P.

Consider an imaginary reference satellite some angle (e.g. half an orbit) ahead of the initial cluster, with mean motion equal to the greatest mean motion of the constellation. First, this ensures consistency in control because each satellite must, counter-intuitively, high drag first to reach the reference moving forwards. This is due to the drag paradox, where drag forces increase angular velocity as the satellite decreases in altitude, essentially trading gravitational potential energy for kinetic energy. Additionally, this reference convention ensures the satellites will not naturally pass the reference.

For each satellite, compute Δt_i for each satellite with the imaginary reference. Consider *only* the cases where the satellite high drags first (and accelerates forwards) because we want to use each Δt_i as a measure of forward progress. Then sort the satellites ascendingly by Δt_i, and assign each satellite rank $r_i \in 0 \dots N-1$. Assign satellite 0 as the reference satellite for calculating separations, and calculate the desired separation for satellite i as $\Delta_{\text{des},i} = -r_i \frac{2\pi}{N}$.

Table 4.2 compares this slotting with the simulated annealing slotting. A lower bound on the coverage error for the methods in [1] is 49.6, computed by generating commands with the QP instead of simulated annealing. The slotting itself is only slightly improved when considered on its own, but using the newly developed methods, the coverage error improves

$\approx$ 9% to 45.2. The simulated annealing command generator in [1] is not guaranteed to be optimal, so it's possible the improvement is even better. This development does not completely preclude the use of simulated annealing for slot generation, as it may be preferable for more extreme initial conditions.

Slotting	Max Δt [days]	QP Coverage Error	LP Coverage Error	Min Horizon [days]
Planet	96.576	49.655	48.342	116
Δt	105.890	47.004	45.219	116

Table 4.2: Comparison of Planet's simulated annealing slotting and the new Δt slotting. Coverage error computed as in section 3.1

Chapter 5

Conclusion

We compared an existing in-orbit differential drag controller with another of academic focus. We established equivalences in their dynamics and control methods, and we showed that the academic linear programming methods can be applied to the more practical considerations of the other. Specifically, we optimally generate commands by reformulating Planet's objective function as a quadratic program, and we further improve it with a new linear program. Then, we evaluate Planet's slot allocator, and by developing a faster and marginally better slot allocator, we suggest that Planet's is already close to optimal. Together, these results form a differential drag controller with a faster runtime and strong assurances of optimality.

Bibliography

[1] C. Foster, J. Mason, V. Vittaldev, L. Leung, V. Beukelaers, L. Stepan, and R. Zimmerman, "Constellation phasing with differential drag on planet labs satellites," *Journal of Spacecraft and Rockets*, vol. 55, no. 2, 2018.

[2] E. Sin, M. Arcak, and A. Packard, "Small satellite constellation separation using linear programming based differential drag commands," 2017. eprint: `arXiv:1710.00104`.

[3] O. Montenbruck and E. Gill, *Satellite Orbits*. Springer, 2005, ch. 3.

[4] D. Vallado and W. McClain, *Fundamentals of Astrodynamics and Applications*, 1st ed. McGraw-Hill, 1997.

[5] Planet Labs, Inc., *Planet labs public orbital ephemerides*, Data retrieved from `http://ephemerides.planet-labs.com/planet_20170215.states`, 2017.