Numerical Optimization of Satellite Avoidance Maneuvers

Firas Sheikh[1]
Georgia Institute of Technology, Atlanta, GA 30332, USA

This paper investigates the numerical optimization of a collision avoidance maneuver and phase return for a satellite in a constellation network. The time until collision and return time are varied, and three impulse maneuvers are assumed. The total Delta-V is minimized while avoiding the collision by 1 km or greater and returning to the original satellite orbit phase. For this analysis, the Clohessy-Wiltshire (CW) solutions were utilized to linearize the system of equations for relative orbital motion without perturbation assumptions. Results demonstrated that a greater time until collision generally produces minimized Delta-V options but longer total maneuver times as a tradeoff.

Nomenclature

h = altitude $[km]$

n = mean motion $[rad/s]$

r_E = radius of Earth $[km]$

t = time $[s]$

t_0 = time of first impulse/maneuver start time $[0\ s]$

t_1 = time of collision after t_0 $[s]$

t_2 = time of second impulse after t_0 $[s]$

t_3 = time of third impulse after t_0 $[s]$

x = x-position $[km]$

y = y-position $[km]$

[1] Graduate Student, Aerospace Engineering

$\Delta\vec{V}_1$	=	First Impulse $[km/s]$
$\Delta\vec{V}_2$	=	Second Impulse $[km/s]$
$\Delta\vec{V}_3$	=	Third Impulse $[km/s]$
$\delta\vec{r}$	=	position vector $[km]$
$\delta\vec{s}$	=	state vector
$\delta\vec{v}$	=	velocity vector $[km/s]$
$\delta\vec{v}^-$	=	velocity before impulse $[km/s]$
$\delta\vec{v}^+$	=	velocity after impulse $[km/s]$
τ	=	time (used generally) $[s]$

I. Introduction

The state of the satellite constellation industry is currently growing at an exponential rate with more satellites getting placed into orbit every year. This is great for companies such as SpaceX and OneWeb, who are dominating this industry, as they can provide more global coverage for broad-band internet. However, this boom in the number of satellites launched into orbit means a dramatic increase in the probability of satellites colliding with space debris, which is defined as any object orbiting Earth such as fully functioning satellites, parts of satellites, or natural space debris such as asteroids. Because of this probability increase, the number of collisions will also begin to increase if safety precautions are not taken. In order to combat potential collisions, satellite avoidance maneuvers can be implemented and could be a key factor in maintaining the safety of constellation networks.

Because satellite maneuvers can be costly and the fuel on the satellite must last for its lifetime, it is essential to minimize the amount of fuel used for a maneuver should a collision become likely. This study aims to optimize the trajectory of a satellite on a collision course by minimizing the Delta-V required to avoid a collision by at least 1 km given a time until impact. Furthermore, because the satellite is part of a constellation, it is necessary to maneuver the satellite back to its original phase in the orbit so that the constellation coverage returns to its original state. Certain companies may want their satellite to return to the original phase in a specific amount of time to minimize the loss of total coverage. Thus, the return time of the satellite must also be given in order to determine the total optimal Delta-V necessary to initially avoid the collision and then arrive at the original satellite phase by the return time.

For this analysis, three impulse maneuvers were assumed to avoid the collision and only planar maneuvers are involved. For the trajectory, instead of the classic inertial frame two-body dynamics, the relative frame CW equation solutions are utilized. The time until collision and return time are varied within a numerical optimizer to output the optimal time for the second impulse and the minimized Delta-V. The total Delta-V is plotted as a function of the return time, and the impact of early or late decisions on the Delta-V cost is analyzed.

II. Approach

A. Parameter Assumptions/Notation

For all results discussed in this paper, the Earth radius, r_E, is equal to 6378 km. All motion is relative in the CW frame (discussed in Section B), and all perturbations are assumed to be negligible. The standard gravitational parameter of Earth is $3.986\text{e}5\ \text{km}^3/s^2$. All calculations assume a circular orbit with altitude of $h = 1000$ km because many of the current satellite constellation companies use altitudes at or near that value [2]. With $h = 1000$ km, the mean motion is then calculated as $n = 9.962\text{e} - 4\ \text{rad/s}$.

For the time notation, t_0 refers to the time at the start of the maneuver, which is 0 s; this is also when the first impulse is applied. Next, t_1 refers to the time that the collision is expected to occur after t_0. Lastly, t_2 refers to the time at which the second impulse is applied while t_3 refers to the return time and when the third impulse is applied.

B. Clohessy-Wiltshire Equations

In terms of computational formulas, the CW equations were utilized to calculate the relative orbital trajectory over time. The CW equations are particularly useful when the distance between two satellites is assumed to be small compared to the distance from the satellite to the center of the Earth. For this study, the distance between the satellite and space debris is very small, and it is acceptable to use the CW equations. The general derivation of the CW equations is shown below from the Prussing and Conway text [1].

Since no out-of-plane maneuvers are assumed, the z position and velocity are negligible. Thus, the CW equations of motion are as follows:

$$\ddot{x} = 3n^2x + 2n\dot{y} \tag{1}$$

$$\ddot{y} = -2n\dot{x} \tag{2}$$

Through integration of the coupled equations (1) and (2), the solution of the CW equations can be written in terms of the x and y positions and velocities. In equation (3), the general solution is conveniently expressed in terms of the state vector $\delta\vec{s}(\tau)$ of the vehicle as a function of time τ relative to the target state.

$$\delta\vec{s}(\tau) = \begin{bmatrix} \delta\vec{r}(\tau) \\ \delta\vec{v}(\tau) \end{bmatrix} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{3}$$

Equation (4) demonstrates the final state is equal to the state transition matrix $\Phi(\tau)$ multiplied by the initial state. The state transition matrix is obtained by assembling the coefficients of the CW solutions as shown in equation (5). Note that the $M(\tau), N(\tau), S(\tau),$ and $T(\tau)$ correspond to each 2x2 matrix within $\Phi(\tau)$ and that $\tau = \tau_2 - \tau_1$, where τ_1 is the initial time of the maneuver and τ_2 is the ending time of the maneuver. Also, $c = \cos(n\tau)$ and $s = \sin(n\tau)$.

$$\delta\vec{s}(\tau) = \Phi(\tau)\delta\vec{s}(\tau_1) \tag{4}$$

$$\Phi(\tau) = \left[\begin{array}{cc|cc} 4-3c & 0 & \frac{s}{n} & \frac{2}{n}(1-c) \\ 6(s-n\tau) & 1 & -\frac{2}{n}(1-c) & \frac{4s-3n\tau}{n} \\ \hline 3ns & 0 & c & 2s \\ -6n(1-c) & 0 & -2s & 4c-3 \end{array}\right] = \left[\begin{array}{c|c} M(\tau) & N(\tau) \\ \hline S(\tau) & T(\tau) \end{array}\right] \tag{5}$$

Equation (4) and (5) can then be split into equations (6) and (7), which calculate the final position and velocity after time τ, respectively. These equations can be used for any maneuver given the initial position and velocity before the maneuver. It is essential for the initial velocity to include the impulse Delta-V as well so that the maneuver can be performed.

$$\delta\vec{r}(\tau) = M(\tau)\delta\vec{r}(\tau_1) + N(\tau)\delta\vec{v}(\tau_1) \tag{6}$$

$$\delta\vec{v}(\tau) = S(\tau)\delta\vec{r}(\tau_1) + T(\tau)\delta\vec{v}(\tau_1) \tag{7}$$

Part of this orbit problem also uses the classic rendezvous process. When the satellite avoids the collision, it must return to the origin, which can be treated as the target. Thus, the final two Delta-V impulses can be analytically solved for with the following general process:

When $\delta\vec{r}(\tau) = 0$,

$$\delta\vec{v}^+(\tau_1) = -N(\tau)^{-1}M(\tau)\delta\vec{r}(\tau_1)$$

Then, $\delta\vec{v}^+(\tau_1)$ can be used to find $\delta\vec{v}^-(\tau)$,

$$\delta\vec{v}^-(\tau) = S(\tau)\delta\vec{r}(\tau_1) + T(\tau)\delta\vec{v}^+(\tau_1)$$

The two impulses can then be calculated as shown in equations (8) and (9).

$$\Delta v_1 = \delta\vec{v}^+(\tau_1) - \delta\vec{v}^-(\tau_1) \quad \textbf{(8)}$$

$$\Delta v_2 = \delta\vec{v}^+(\tau) - \delta\vec{v}^-(\tau) = -\delta\vec{v}^-(\tau) \quad \textbf{(9)}$$

C. Numerical Optimization

In terms of computational methods, a numerical optimizer was created in MATLAB using the function fmincon(). Refer to the Appendix to see the full MATLAB script. The numerical optimizer helped to determine the minimum Delta-V required to perform the collision avoidance maneuver as well as the optimal time, t_2, to perform the second Delta-V. The objective function, which is the function to be optimized, is shown in equation (10).

$$f_{objective} = |\Delta\vec{V}_1| + |\Delta\vec{V}_2| + |\Delta\vec{V}_3| \quad \textbf{(10)}$$

This equation is essentially the sum of the magnitudes of each of the three impulses, which must be minimized. The $\Delta\vec{V}_1$ is unknown, so an initial randomized guess is made for that variable, and fmincon() eventually solves for its optimal value. The important note about t_2 is that, alongside $\Delta\vec{V}_1$, it is also an initial randomized guess since its value is unknown, and fmincon() will eventually solve for this value. For each numerical iteration value of $\Delta\vec{V}_1$ and t_2, the code analytically solves for $\Delta\vec{V}_2$ and $\Delta\vec{V}_3$ using the rendezvous equations (8) and (9). When the objective function is finally minimized, the total minimum Delta-V is output.

It is noteworthy that the time t_1 and t_3 are input constants; however, multiple t_1 values are run in order to understand the impact of early or late decisions on the Delta-V cost. For each t_1 value, an array of t_3 values are run such that there is always a different combination of t_1 and t_3 values to calculate the Delta-V. For each combination of t_1 and t_3 values, a loop of 20 iterations is run each with a different randomized initial value for $\Delta\vec{V}_1$ and t_2, and the minimum Delta-V was collected as a data point for each case. It is important to note that t_2 can occur before or after t_1 depending on what the numerical optimizer decides is the best option. Also, t_3 is constrained such that it is always greater than t_1.

III. Results and Discussion

After the multiple iterations with various input values in the numerical optimizer, the data was collected, and the following plot in Figure 1 was created to display the total Delta-V as a function of the return time. Note that all velocity calculations were consistently done in units of km/s and then final Delta-V values are displayed in m/s in Figure 1. Each plot color represents a time before the potential collision, t_1, which was varied from 100 to 6000 seconds. The return time values, t_3, were varied from 200 to 14000 seconds, depending on the t_1 values that were used. In Figure 1, the general trend seen for each t_1 value is that as return time increases, the Delta-V decreases, but the Delta-V eventually levels off around a certain return time. Furthermore, as t_1, increases, the total Delta-V decreases dramatically, which makes sense because the more time that the satellite has until impact, the less fuel it will need to use in order to be greater than 1 km away from the collision.

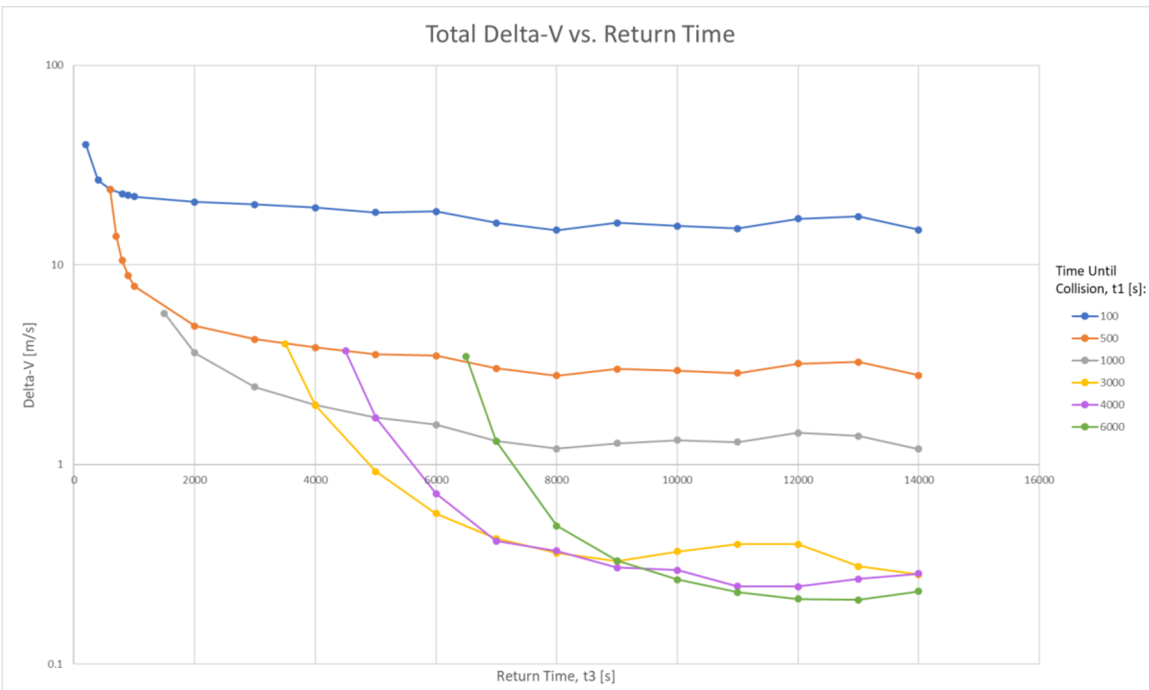

Figure 1 – Total Delta-V vs. Return Time

For $t_1 = 100\ s$, the Delta-V starts at approximately 40 m/s for a 200 s return time. This is the highest Delta-V on the plot and it makes sense because the satellite needs a high Delta-V to avoid the collision in 100 s and then return to its original phase in another 100 s, but this is not realistic. As the return time is increased to 800 s, the Delta-V can decrease to as much as 22.7 m/s, which is about half the previous Delta-V. Increasing the return time to 8000 s yields a Delta-V of less than 15 m/s, which is the lowest Delta-V for this t_1 case; any return time beyond this only results in greater Delta-V values, so this demonstrates that simply increasing return time will not always result in a minimum Delta-V. Therefore, there is a tradeoff between return time and Delta-V, and the satellite mission team should decide which is more valuable to them. For example, if they want to return earlier, they have to assume they will need to use a higher Delta-V. The reason plot for $t_1 = 100\ s$ levels off around the 15-18 m/s Delta-V range is because of the short amount of time until the collision; this ultimately limits the Delta-V from decreasing any lower because the satellite must achieve a distance greater than 1 km in 100 seconds, so the Delta-V will always be high to accommodate this constraint.

Similarly, for $t_1 = 500\ s$, the Delta-V starts out with a relatively high value around 24 m/s, but as the return time is increased, the necessary Delta-V falls below 3 m/s, which is an exceptional decrease to 12% of the starting Delta-V. In this case, there is a bit more flexibility with the Delta-V required compared to the case for $t_1 = 100\ s$, but there is still a limit on how low the Delta-V can go, and this can be seen in the plot since it levels off to 2-3 m/s with increasing return time.

As t_1 increases, the trends are similar to the previous cases. For $t_1 = 3000\ s$, an interesting case arises; at about a 9000 s return time, the Delta-V reaches a local minimum of 0.33 m/s, and for a few data points after that, the Delta-V increases but then drops back down to a minimum of 0.28 m/s by 14000 s. If the mission designers are attempting to minimize both Delta-V and the return time but need to compromise between the two, it would make much more sense to use the Delta-V of 0.33 m/s with a return time of 9000 s because, although that is not the absolute minimum Delta-V possible, its difference from the 0.28 m/s Delta-V is negligible, and it would be much more efficient to return in 9000 s as opposed to 14000 s. These are the types of tradeoffs that the plot below can provide valuable insight on and can help mission designers determine the best path forward.

Lastly, $t_1 = 6000\ s$ was used as the largest collision time since that is just before one full orbit for this altitude. This plot follows a continuously decreasing Delta-V trend as return time increases. The smallest overall Delta-V is 0.21 m/s at 13000 s, which is about two full orbital periods. Because there is a significant amount of time before the collision and there is a lot of time after the collision, the satellite does not have to use as much Delta-V to avoid the impact and return to phase. This demonstrates that the earlier the collision is known, the less Delta-V that could potentially be required to perform the maneuver, which could help satellite companies save heavily on fuel costs.

IV. Conclusion

A key takeaway from this analysis is that the more time the satellite has before potential impact, the lower the Delta-V can be. Moreover, the more time available after the impact to return to the original phase, the more likely the Delta-V will be lower as well. However, there is a point where increasing the return time will do little to nothing to lower the Delta-V and may even lead to diminishing returns. Overall, from Figure 1, it is apparent that having at least about half the orbital period until impact can lead to significantly low Delta-V's if given a respectable amount of time to return to the original satellite phase. In the best-case scenarios when the satellite has ample time to maneuver, the Delta-V can end up as low as 0.2 m/s, which is very small and would greatly contribute to a satellite avoiding a collision. However, in the unfortunate case when a satellite has little time to react before a potential impact, the Delta-V will be a much higher value since the satellite must avoid the debris by 1 km in a short amount of time.

For future analyses, the script could be updated so that it can account for N number of impulses for the maneuver. That way, the total Delta-V could be minimized for each N impulse case, and the effect of increasing the number of impulses and optimizing when they should occur could be analyzed as well. Since there would be more impulses, the effect of increasing the return time could be analyzed as well along with how this effects the total Delta-V. Another idea is to incorporate a probabilistic model that determines that likelihood of a collision given the time before impact. Depending on how far the satellite is from potential collision could impact how accurate the prediction is, and the closer the satellite gets to the point of impact, the more accurate the prediction becomes. This can be helpful because a collision maneuver could be avoided altogether if the probabilistic model determines that the likelihood of collision is very low. However, as the satellite gets closer to the potential impact and the probability goes up to a certain danger level, then a maneuver can be performed. The tradeoff is that the maneuver will require more Delta-V if it is performed later in the orbit, so that would be an interesting optimization and probability problem that could be analyzed.

Ultimately, these results demonstrate that satellite avoidance maneuvers do not have to be costly at all and can be performed provided there is a reasonable amount of time before and after the potential impact. Avoidance maneuvers are also an essential step forward for satellite constellation companies. As the LEO and GEO orbit spaces become more crowded with the increase in annual launches, the safety of the satellite networks has become uncertain. However, satellite avoidance maneuvers have the potential to safely maneuver satellites while saving companies money and fuel so that they can continue to provide their services to customers for years to come.

Appendix

MATLAB Code

```
%Satellite Optimization
%By: Firas Sheikh
%October 2022

close all
clearvars
clc
global t3
r0 = [0,0]; v0 = [0,0];
S0 = [r0,v0]';
t1 = t1v();
t3vec = [200, 400, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000,
9000, 10000, 11000, 12000, 13000, 14000];

% t2 = 150;
% t3 = 2100;
% dv0_1 = [.002,.008]; %initial delta-V1 guess
mat_final = [];
tic
for j = 1:length(t3vec)
    mat_all = [];
    t3 = t3vec(j);
    for i = 1:20
        t2 = rand()*t3;
        dv0_1 = [rand()/1000,rand()/1000]; %initial delta-V1 guess

        dv0 = [dv0_1, t2]'; %delta-V1 and t2

        %run fmincon to minimize the objective function given the dv0 guess

    %tic

        [dvout, fval] = fmincon(@objective, dv0, [],[],[],[], [-Inf,-Inf,0],
[Inf,Inf,t3], @constraint);

    %toc

    %%After the run
        t2 = dvout(3);

        Sout = state(dvout(1:2), t2, S0); %call state function to change the satellite
state
        delr_t2 = Sout(1:2); delv_t2m = Sout(3:4); %need the new position and
velocities before rendezvous
        [deltav2, deltav3] = rendezvous(delr_t2,delv_t2m,t3-t2);
        %save('C:\Users\e390937\Documents\GT\AE8900\dvout.mat', 'dvout')

        dVs = [dvout(1:2); deltav2; deltav3];
        %mag_dv = norm(dvout(1:2)) + norm(deltav2) + norm(deltav3)

        mat = [t1;t2;t3;fval;fval*1000;dVs]';
        mat_all = [mat_all; mat];
    end
    [minval, I] = min(mat_all(:,4));
    mat_min = mat_all(I,:);
    mat_final = [mat_final; mat_min];
end
```

```
toc
%TestRunNew

function out = t1v()
    out = 100; %t1 %time(s) until impact
end

% function out = t3v()
%     out = 800; %t1 %time(s) until impact
% end

function out = objective(dv) %objective function is total delta V (magnitude)
    r0 = [0,0]; v0 = [0,0]; S0 = [r0,v0]';

    t2 = dv(3);
    %t3 = t3v();
    global t3

    S2 = state(dv(1:2), t2, S0); %call state function to change the satellite state
    delr_t2 = S2(1:2); delv_t2m = S2(3:4); %need the new position and velocities
before rendezvous
    [dv2, dv3] = rendezvous(delr_t2,delv_t2m,t3-t2);
    %deltaV2 = [deltaV2 dv2];
    %deltaV3 = [deltaV3 dv3];
    out = norm(dv(1:2)) + norm(dv2) + norm(dv3); %sum of magnitude of delta-Vs
    %out = norm(dv(1:2)) + norm(dv2) + norm(dv3) + t2/1000 + t3/1000; %sum of
magnitude of delta-Vs

    %dv2 and dv3 are dependent on t2 and t3
end

function [c, ceq] = constraint(dv) %constraint function
    r0 = [0,0]; v0 = [0,0]; S0 = [r0,v0]';

    t1 = t1v(); t2 = dv(3);
    %t3 = t3v();
    global t3

    if t2 > t1
        S1 = state(dv(1:2), t1, S0); %call state function to change the satellite
state
        c1 = 1 - norm(S1(1:2)); %constraint to make sure that distance is > 1 km from
collision
        %c4 = 0;
    else
        S2 = state(dv(1:2), t2, S0); %call state function to change the satellite
state
        delr_t2 = S2(1:2); delv_t2m = S2(3:4); %need the new position and velocities
before rendezvous
        [dv2, ~] = rendezvous(delr_t2,delv_t2m,t3-t2);
        S1 = state(dv2, t1-t2, S2);
        c1 = 1 - norm(S1(1:2)); %constraint to make sure that distance is > 1 km from
collision
        %c4 = -t2;
    end
    %c2 = t2-t3;
    %c3 = t1-t3;

    %c = [c1, c2, c3, c4];
    c = c1;

    %ceq is equality constraint
    ceq = [];
end
```

```
function S2 = state(dv,t,S) %alters state of satellite using CW equations
    n = meanM();

    %initial position
    %r1x = S(1);
    %r1y = S(2);

    %initial velocity + delta-V
    S(3) = S(3) + dv(1);
    S(4) = S(4) + dv(2);

    %state transition matrix times position and velocity gives new position and
    %velocity
    %AKA: CW Equation Solutions
    [phi, Ma, Na, Sa, Ta]  = STMatrix(t,n);
    S2 = phi*S;
end

function [deltaVt2, deltaVt3] = rendezvous(drt2,dvt2_,t)
%rendezvous equations from class
    n = meanM();
    [~, M, N, S, T] = STMatrix(t,n);
    dvt2 = -inv(N)*M*drt2;
    dvt3_ = S*drt2 + T*dvt2;
    deltaVt2 = dvt2-dvt2_;
    deltaVt3 = -dvt3_;
end

function [phi, M, N, S, T] = STMatrix(t,n)

    c = cos(n*t); s = sin(n*t);
    M = [4-3*c      0;
        6*(s-n*t)   1];

    N = [s/n        2/n*(1-c);
        -2/n*(1-c)  (4*s-3*n*t)/n];

    S = [3*n*s      0;
        -6*n*(1-c)  0];

    T = [c          2*s;
        -2*s        (4*c-3)];

    phi = [M,N;
           S,T];

end

function n = meanM()
    %Grav Parameters
    G = 6.674e-11; %m^3/(kg*s^2)
    G = G./1000^3; %convert to km^3/(kg*s^2)
    M_e = 5.972e24; %kg %Mass of Earth
    R_e = 6378; %km %Earth radius
    mu = G*M_e; %mu value

    h = 1000; %km %orbit altitude
    a = R_e + h; %km %semi-major axis (circular)
    n = sqrt(mu/a^3); %mean motion
end
```

References

[1] Prussing, J. E. and Conway, B. A., *Orbital Mechanics*, 2nd ed., Oxford University Press, New York, 1993, pp. 182-191.

[2] Portillo, I., Cameron, B. G., and Crawley, E. F., "A Technical Comparison of Three Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband," Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2019, pp. 126.