ASE 372N

Extended Global Positioning System

Matthew Cullen Self

December 9, 2016

Abstract

The feasibility of using GPS signals at altitudes high above the constellation has been evaluated. A model was synthesized and evaluated to produce carrier-to-noise ratios of observations at various places in orbit around Earth, and at various times. It has been found that, given modern tracking and acquisition systems, a receiver in High Earth Orbit will be able to determine their location based solely off of GPS signals if the geometry is favorable (more likely than not). The impact of antenna and receiver design on the observability of a position has been discussed (more gain results in a higher likelihood of observability). Antenna gain patterns have been observed in carrier-to-noise maps, validating the implemented models. Confirmation of prior experiments further indicates the model is correct.

1 Introduction

Global Navigation Satellite System (GNSS) receivers work by very carefully beaming information from known locations to an unknown location and evaluating the resulting shifts and delays in the signals as they arrive. This commonly takes place as a mobile devices discover their locations in reference to the known positions of the GPS satellites. Usually these devices are at or near, the surface of the Earth - the use case for which GPS has been designed - thus GPS Satellite Vehicles (SVs) direct their information carrying signals towards the Earth. These signals are emitted by complex antenna systems, but the net result is effectively a hemisphere of radiation around the pointing direction (towards the center of the Earth). Because the GPS SVs are high in space ($\approx 20,200$ km altitude), the radiated signal does not fall only on Earth; some of the signal misses the Earth and is broadcast into an area surrounding the Earth. This overlap in signal coverage, forms the basis of the Space Service Volume (SSV), a region in space where GPS signals are available. The basis of this report is to examine the extent of the SSV and the viability of using GNSS signals in this region.

1.1 Motivation

Traditional methods of locating satellites rely on sporadic radar measurements to form a position and velocity estimate for a given SV. As more and more satellites are launched, the availability of radar measurements continues to decline (unless a SV is very important). To combat this, many satellites now have a GNSS receiver to provide navigational solutions formed not from ground based

observations, but instead formed continuously from GNSS signals. This trend has been accelerated by the increasing quality and availability of GNSS receivers, making GNSS tracking of satellites cheaper and more reliable.

Furthermore, it has been proven that GPS based navigation solutions are possible even from "above the constellation," or at altitudes greater than the orbit of GPS satellites. Starting in 1990, a classified geosynchronous satellite ($\approx 36,000$ km altitude) used GPS signals at least in part of its navigation solution [5]. The Radio Amateur Satellite Corporation (AMSAT) OSCAR-40 satellite was able to sporadically track satellites at an altitude of 59,000 km [6]. NASA's Manetospheric Multi-Scale (MMS) mission, with a recently developed high gain receiver, was able to continuously track at least four satellites (the number required for a navigation solution) at altitudes of 70,000 km [2]. These results form the baseline for additional studies into the feasibility of GPS navigation in High Earth Orbit.

1.2 Theory

The governing equations that determine whether or not a signal can be tracked or not are presented below in Equations 1 and 2 (from [6]).

$$\hat{P}_r = P_{out} + G_t + L_d + G_r \tag{1}$$

$$\widehat{C/N_0} = \hat{P}_r + N_T + 228.6 + L_{sys} \tag{2}$$

The estimated received power, $\hat{P}_r$, is a function of the emitted power, P_{out}, the transmitter gain, G_t, the receiver gain, G_r, and the free space attenuation $L_d < 0$. The attenuation follows the inverse-square law such that the decibel reduction is equal to

$$20\log(\frac{\lambda}{4\pi\rho}).$$

Once the received power has been estimated, an estimate of the carrier to noise ratio (C/N_0) can be formed based on the thermal noise, $N_T < 0$, and the system losses due to front end noise and conversion losses, $L_{sys} < 0$. The thermal noise can be modeled as

$$N_T = -10\log(T_{sys})$$

where T_sys is the equivalent system noise temperature. The resulting C/N_0 determines whether a satellite can be tracked. If the C/N_0 is above a threshold particular to the receiver, then the signal is deemed "observable."

Many of the parameters are dependent on the receiver (e.g. receiver gain, system losses, thermal noise, and C/N_0 threshold). These values can be systematically determined, and are particular to the antenna being used, the receiver hardware and software, and other implementation details.

The parameter of interest for this report is the transmitter gain, G_t. Lockheed Martin has released the antenna gain patterns of many GPS satellites, making it possible to model the antenna gain accurately as a function of transmitter and receiver locations [7]. Crucially, the gain is dependent on two angles: the offbore angle, θ, and the polar angle, φ. These two angles can be calculated based on the attitude of the transmitting satellite and the relative position vector. Then those angles can be used to determine the resulting directive gain of the transmission. Figure

1 shows an example of the gain pattern. Inside low values of θ, the gain is extremely high - this corresponds to the portion of the satellite's signal that is broadcast towards the Earth. There are additional "sidelobes" that extend outside this primary range, providing additional gain at some angles of φ.

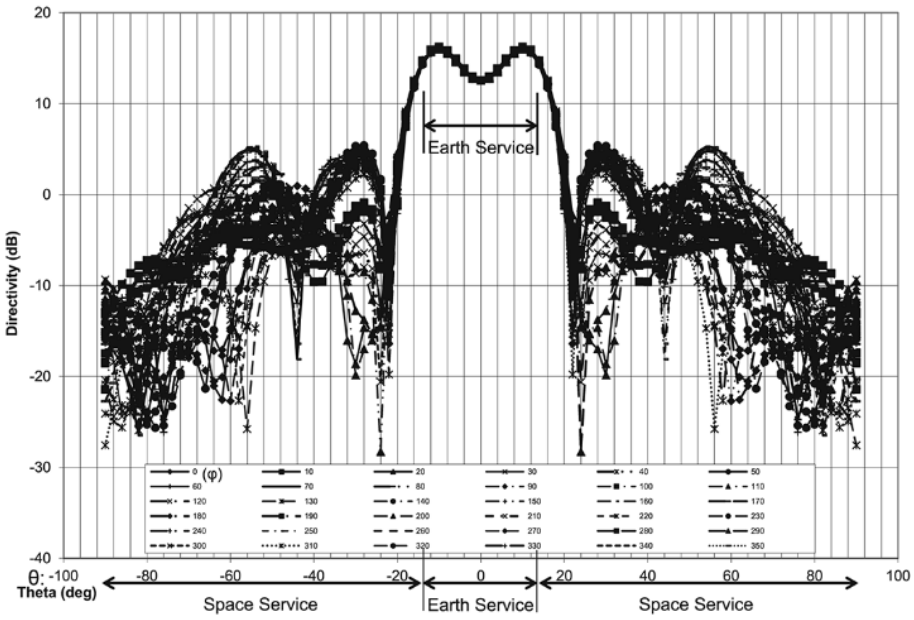

Figure 1: Typical L1 Gain Pattern

To calculate θ and φ, a yaw-steering attitude model was assumed [4]. This means that the satellite maintains a constant orientation towards the sun by varying a yaw angle around the primary pointing axis. This yaw angle, ψ can be found in terms of the β and μ, the angles between the sun and the orbital plane and the satellite and midnight (see Fig. 2).

$$\psi = \text{atan2}(-\tan(\beta), \sin(\mu)) \tag{3}$$

This yaw angle, combined with a local orbital frame defined by the radial, along-track, and cross-track directions, makes it possible to relate the satellite position in ECEF to the Lockheed Martin prescribed body fixed axis.

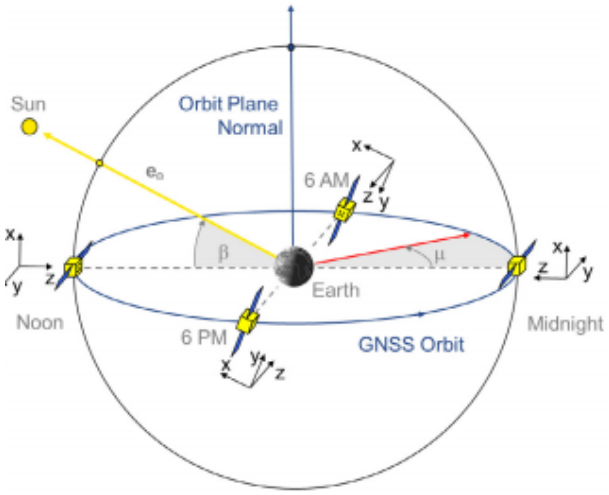

Figure 2: Sun - Earth - Satellite Geometry

1.3 Implementation

Taken together, the previous section describes a model for transmission and reception of a SV's signal. This was coded in MATLAB as a series of functions that can be called from a master script. Core to the process is the `findRotationAngle` function, which takes a structure containing a satellite's position, velocity, the relative position vector, and the time, and returns φ and θ. This is completed by first finding the position of the Sun in the ECEF frame (nontrivial, relies on `findSun` [8]), then relating that to the yaw angle, and finally using the yaw angle and position to find the directional cosine matrices necessary to find the body fixed axes in the ECEF frame.

A lookup function was implemented as `SatDirectivityGain`, which takes φ and θ as inputs, then performs the necessary bilinear interpolation. These were synthesized together in several experiments that tested the effects of various parameters on the number of tracked satellites.

2 Results and Analysis

2.1 Single Satellite

A single transmitting satellite was modeled. At a given instant in GPS time, the position of a satellite was determined, then the effects of changing several parameters were determined. The angles for ϕ and θ were calculated as a function of height above the North Pole, a region which was determined to have moderate visibility of the satellite. Figure 5 shows the two plots. Note how the offbore angle θ is negative and drops below $-90\,\text{deg}$. The drop below $-90\,\text{deg}$ reflects that, at a certain altitude, the receiver is now behind the transmitter, and is unable to acquire any signals.

Figure 6 shows the modeled unbiased carrier to noise ratio. The unbiased carrier to noise ratio is found by setting the net gain and attenuation of the receiver and antenna to 0 dB, modeling a "nuetral" receiver. The biased, or actual received C/N_0 can be found by adding back in the appropriate net gain to the unbiased C/N_0. At precisely the same altitude that $\theta = 90\,\text{deg}$, the C/N_0 drops to 0, indicating that the signal isn't observable under any circumstances. The C/N_0

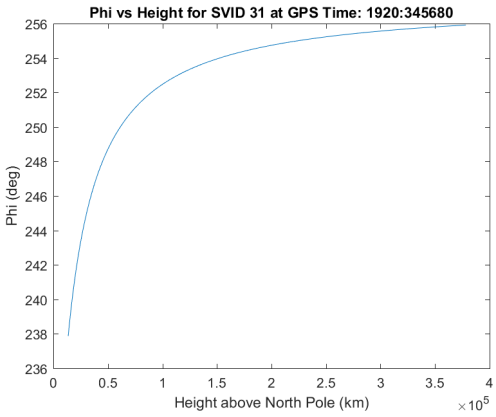

Figure 3: Phi

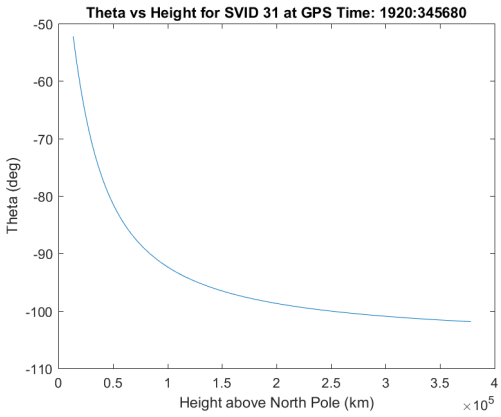

Figure 4: Theta

Figure 5: Angles vs Altitude

plot also shows the presence of two side lobes extending beyond the main center lobe.

To better visualize the space in which a signal is observable, a mesh of locations was generated along the equatorial axis, extending to ±100,000 km in both the ECEF x and ECEF y directions. The unbiased C/N_0 was then plotted at each individual location, yielding Figure 7. The plot has two main leaves, separated by the Earth's umbra, and bounded by the plane of the transmitting antenna. In each leaf, the side lobe gain patterns can be seen as certain lines of constant θ tend to have exceptionally high gains.

To get a idealized plot of the side lobes, the transmitting satellite was modeled as being on the ECEF x-axis instead of in its proper 3 dimensional space. This had the effect of constraining φ and θ to varying along the x-y plane that the receiver is modeled in. Figure 8 shows clearly the antenna gain pattern in the contour map on the floor of the plot. The highest C/N_0 is closest to the antenna boresight, right outside the Earth's shadow. In actual mission planning, care must be taken not to optimize C/N_0 so much that the received sigal actually passes through the Earth's atmosphere.

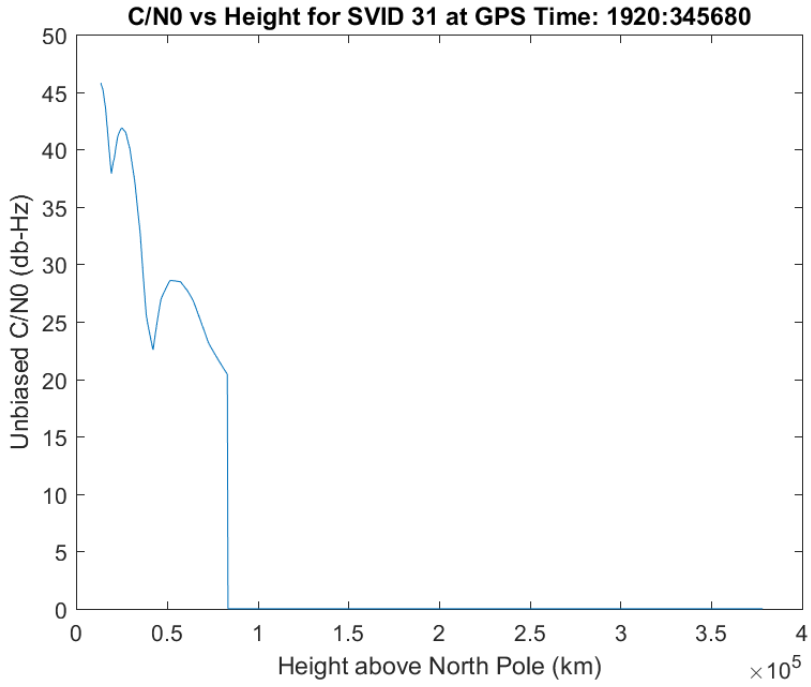

Figure 6: C/N_0 vs Altitude

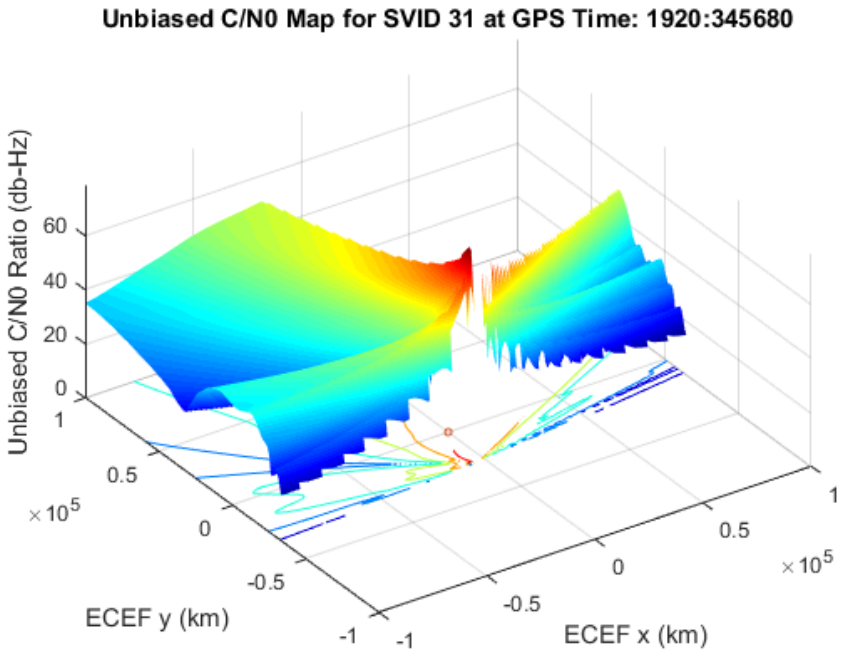

Figure 7: C/N_0 vs Position (Real Position)

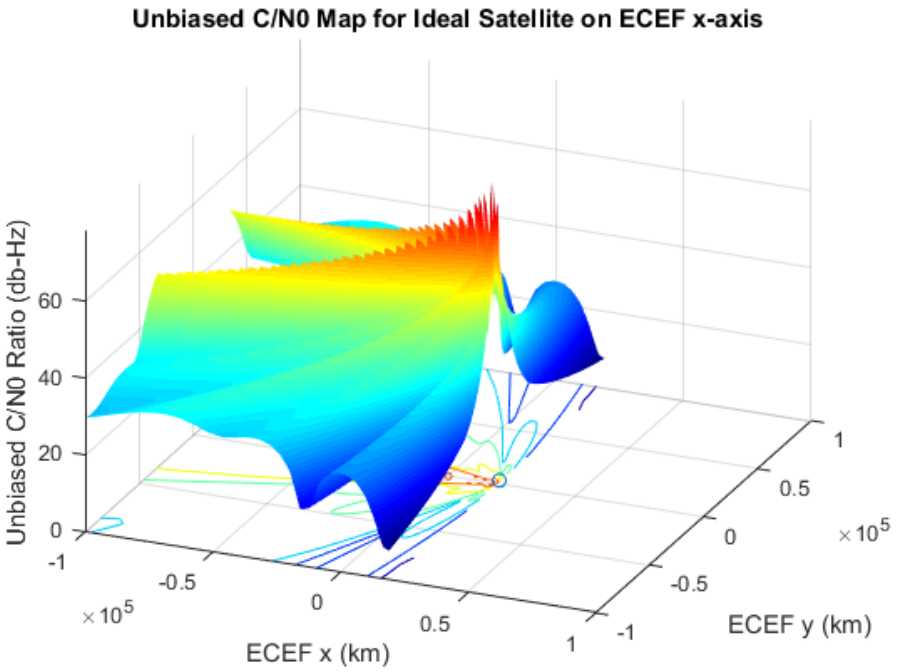

Figure 8: C/N_0 vs Position (Modeled Position)

2.2 Multiple Satellites

Using the same mesh of possible receiver locations, the $\hat{P}_r$ and unbiased C/N_0 for the L1 signal from all the GPS Block IIR and IIR-M satellites was calculated. Because extracting each individual gain matrix involves opening and processing by hand an Excel Spreadsheet contained in a PowerPoint presentation, the same gain characterization was assumed to be roughly valid for all satellites. By setting the tracking threshold to 25 dB-Hz (consistent with NASA's latest GNSS receiver [2]), the number of tracked satellites was mapped to the x-y plane in Figure 9. The resulting map is slightly misleading, as it includes results from inside the Earth reported as valid. The map is more than likely flawed in some way as the GPS constellation is generally symmetric, which should result in a largely symmetric map as well. Instead, this figure indicates that on one side of the Earth, 18 satellites are visible, and zero satellites are visible on the other.

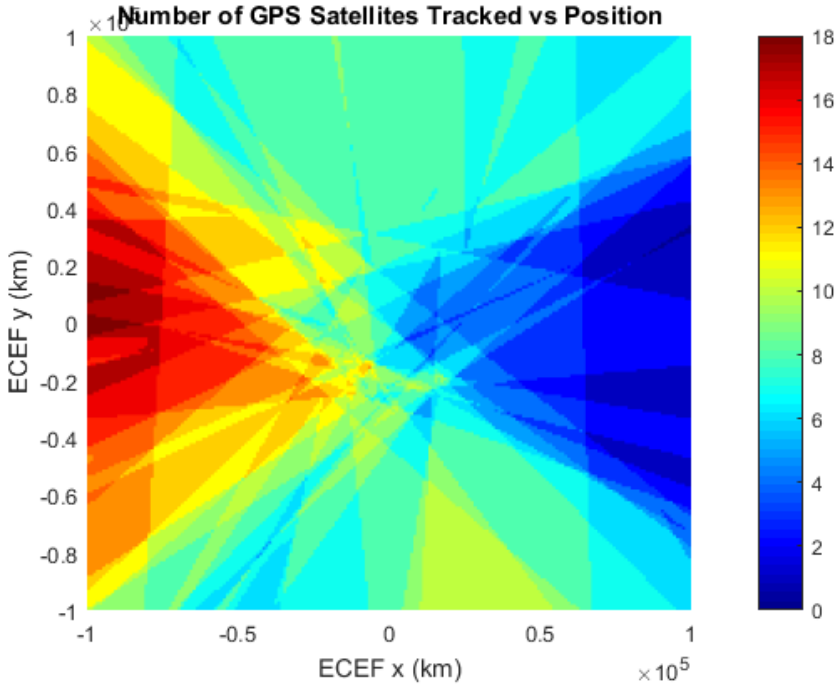

Figure 9: Tracked Satellites vs Position

2.3 AMSAT OSCAR-40 Replication

To coarsely verify that the results are sensible, an experiment was set up to replicate the results of AMSAT OSCAR-40's mission. The receiver parameters were set to be equal to those provided by Moreau et al. and placed at an altitude equal to the apogee of the actual AO40 satellite. The number of tracked satellites was calculated every 15 minutes during a simulated 48 hour interval, and the resulting quantity vs time plot is presented below (see Fig. 10). Note that, for the majority of the time, no satellites are tracked, with periodic bursts of one or two tracked satellites. This is consistent with the reported findings in [6].

In order to reflect a more modern version of AO40, the tracking threshold was decreased from 40 dB-Hz to 25 dB-Hz, while the net gains were assumed to remain the same. The changes reflect

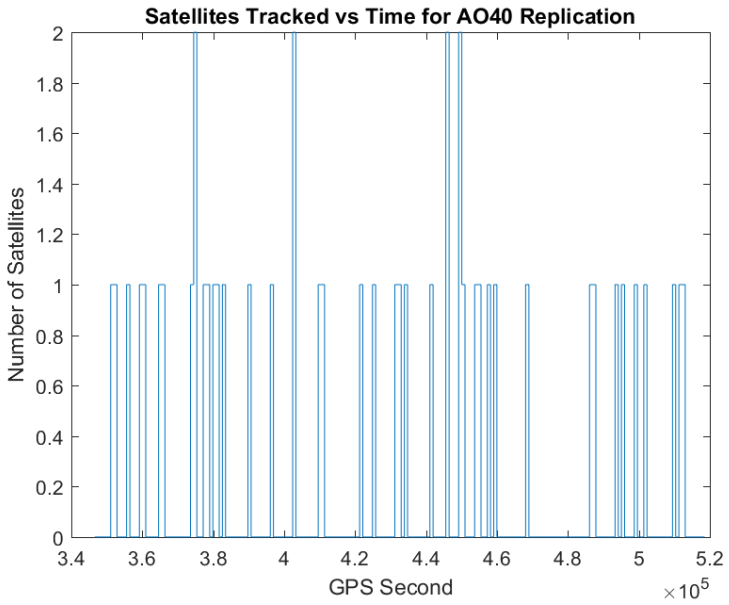

Figure 10: Tracked Satellites vs Time

an improvement in tracking systems, and a miniaturization of early 2000's components. Figure 11 shows the resulting number of tracked satellites, over the same time period. The drastic increase in number of tracked satellites shows that simply increasing the quality of the tracking and acquisition system can make the difference between being able to compute a navigational system, and being forced to rely on other external tracking systems. The mean number of satellites tracked is 2.5, compared to the old version's 0.25.

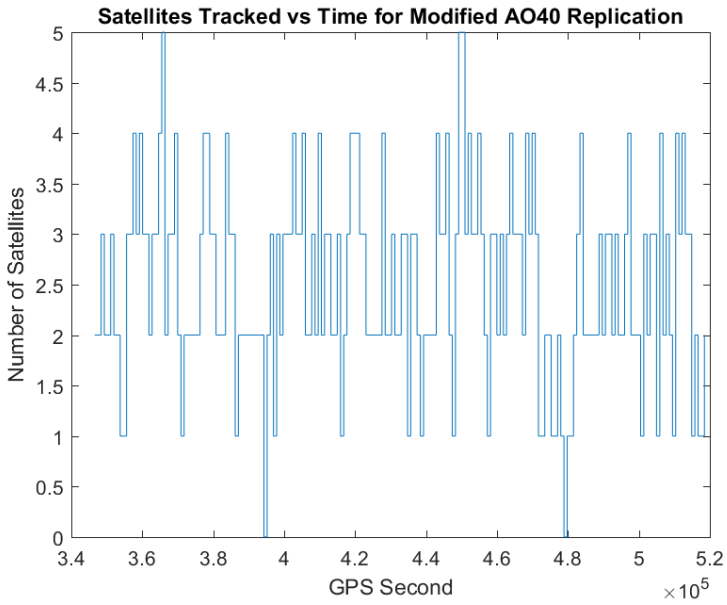

Figure 11: Tracked Satellites vs Time (Improved System)

3 Conclusions

GPS provides a system for determining navigational solutions not just on or near Earth's surface, but also at altitudes high above Earth's surface. Modern tracking and acquisition systems allow relatively weak signals to be captured, and it is possible to model the received power of GPS signals at a given location in space. The model relies on GPS gain patterns and attitude models, and varies in time and with receiver parameters. Increasing the receiver sensitivity greatly increases the number of satellites able to be tracked, which intuitively makes sense. A radio telescope pointed at Earth from Pluto would probably be able to determine a navigational solution, its just a matter of high enough gain and low enough noise. More so than providing an upper limit on how high GPS works, this report has illuminated the design considerations that must be accounted for when developing a high altitude GPS receiver.

More work could be done to implement more satellite's gain patterns, rather than just relying on one as a proxy for all of them. Additionally, Lockheed has provided the gain patterns for the signal broadcast at L2 - these could be included to augment the number of observable satellites. In order to actually model what a given satellite would receive, a better model of the receiver is needed. Manufacturer data sheets do not provide a map from reception angles to gain: this would need to be empirically determined. Additionally, testing would reveal better values for other RX parameters.

Finally, there is the potential to somehow incorporate the received signal strength into the measurement model of the navigational solution. This report has defined a model for the C/N_0 as a function of r. By comparing the estimated C/N_0 to the observed C/N_0 in terms of the partial derivative, the model could fit into the general navigational solution. For example, if a satellite receives signals from GPS satellites a and b and the receiver knows r_a and r_b, then that immediately constrains the possible solution space to those areas in front of satellites a and b.

Appendix A Selected Code

Listing 1: Single Satellite

```
clear; close all; clc;

addpath('../code');
addpath('../code/helpers');
addpath('../config');

gpsWeek = 1920;
gpsSec = 345680;
EarthCutoff = ceil(atan2d(6371+500,20200+6371));
config;

satdata = retrieveNavigationData(gpsWeek,gpsSec,0,'../NavFiles');

sat53 = satdata([satdata.SVID]==31);

[sv.r, sv.v] = satloc(gpsWeek,gpsSec,sat53);
sv.u = -sv.r/norm(sv.r);

for i = 1:1000
    d(i) = ((i-1)*(385000-20000)/1000+20000)*1000;
    sv.rel = sv.r - [0;0;d(i)];
    [phi(i), theta(i)] = findRotationAngle(sv, gpsWeek, gpsSec);
    if (abs(theta(i)) < EarthCutoff) || (abs(theta(i)) > 90)
        visible(i).power = NaN;
        visible(i).cn0 = 0;
    else
        visible(i).power = SatPowerOut(31) +
    SatDirectivityGain(dirGain,theta(i), phi(i)) + SatGainCF(31) +
    20*log10(lambdaGPSL1/(4*pi*norm(sv.rel))); % multiple things need to be
    checked out, log vs log10, need to make SatPowerOut matrix, SatDGain
    depends on another angle that I don't know, must make up RXGain, need to
    cite this formula
        visible(i).cn0 = visible(i).power - 10*log10(190) + 228.6; % RXNoise
    is negative
    end
end
figure(1);
plot(d./1000-6378,phi);
xlabel('Height above North Pole (km)');
ylabel('Phi (deg)');
title('Phi vs Height for SVID 31 at GPS Time: 1920:345680');
print('../images/phivsheight','-dpng');
figure(2);
plot(d./1000-6378,theta);
xlabel('Height above North Pole (km)');
ylabel('Theta (deg)');
title('Theta vs Height for SVID 31 at GPS Time: 1920:345680');
```

```
print('../images/thetavsheight','-dpng');
figure(3);
plot(d./1000-6378,[visible.cn0]);
xlabel('Height above North Pole (km)');
ylabel('Unbiased C/N0 (db-Hz)');
title('C/N0 vs Height for SVID 31 at GPS Time: 1920:345680');
print('../images/cn0vsheight','-dpng');

clearvars('phi','theta');
[X,Y] = meshgrid(-100000:1000:100000);
X = X*1000;
Y = Y*1000;
for i = 1:size(X,1)
    for j = 1:size(Y,1)
        sv.rel = sv.r - [X(1,i);Y(j,1);0];
        [phi(j,i), theta(j,i)] = findRotationAngle(sv, gpsWeek, gpsSec);
        if (abs(theta(j,i)) < EarthCutoff) || (abs(theta(j,i)) >= 90)
            cn0(j,i) = NaN;
            continue;
        end
        power = SatPowerOut(31) + SatDirectivityGain(dirGain,theta(j,i),
    phi(j,i)) + SatGainCF(31) + 20*log10(lambdaGPSL1/(4*pi*norm(sv.rel)));
        cn0(j,i) = power - 10*log10(190) + 228.6;
    end
end
figure(4)
surfc(X./1000,Y./1000,cn0);
hold on;
scatter3(sv.r(1)/1000,sv.r(2)/1000,0,1);
scatter3(0,0,0,10);
xlabel('ECEF x (km)');
ylabel('ECEF y (km)');
zlabel('Unbiased C/N0 Ratio (db-Hz)');
title('Unbiased C/N0 Map for SVID 31 at GPS Time: 1920:345680');

sv.r = [20000000;0;0];
sv.u = [-1;0;0];
sv.v = [0;sqrt(3.986004418e14/norm(sv.r));1];
clearvars('phi','theta');
[X,Y] = meshgrid(-100000:1000:100000);
X = X*1000;
Y = Y*1000;
for i = 1:size(X,1)
    for j = 1:size(Y,1)
        sv.rel = sv.r - [X(1,i);Y(j,1);0];
        [phi(j,i), theta(j,i)] = findRotationAngle(sv, gpsWeek, gpsSec);
        if (abs(theta(j,i)) < EarthCutoff) || (abs(theta(j,i)) >= 90)
            cn0(j,i) = NaN;
            continue;
        end
```

```
91          power = SatPowerOut(31) + SatDirectivityGain(dirGain,theta(j,i),
    phi(j,i)) + SatGainCF(31) + 20*log10(lambdaGPSL1/(4*pi*norm(sv.rel)));
            cn0(j,i) = power - 10*log10(190) + 228.6;
        end
    end
    figure(5)
96  surfc(X./1000,Y./1000,cn0);
    hold on;
    scatter3(sv.r(1)/1000,sv.r(2)/1000,1);
    scatter3(0,0,0,10);
    xlabel('ECEF x (km)');
101 ylabel('ECEF y (km)');
    zlabel('Unbiased C/N0 Ratio (db-Hz)');
    title('Unbiased C/N0 Map for Ideal Satellite on ECEF x-axis');
```

Listing 2: Many Satellites

```
    clear; close all; clc;
2
    addpath('../code');
    addpath('../code/helpers');
    addpath('../config');

7   gpsWeek = 1920;
    gpsSec = 345680;
    EarthCutoff = ceil(atan2d(6371+500,20200+6371));
    config;

12  satdata = retrieveNavigationData(gpsWeek,gpsSec,0,'../NavFiles');

    rx.Tsys = 1;
    rx.RXNoise = 0;
    rx.RXGain = 0;
17  thresh = 25;

    [X,Y] = meshgrid(-100000:1000:100000);
    X = X*1000;
    Y = Y*1000;
22  for i = 1:size(X,1)
        for j = 1:size(Y,1)
            rx.r = [X(1,i);Y(j,1);0];
            [ tracked, neededGain(j,i) ] = observe( rx , gpsWeek , gpsSec ,
    satdata , thresh );
            numSats(j,i) = numel(tracked);
27      end
    end
    surf(X./1000,Y./1000,numSats,'LineStyle','none')
    xlabel('ECEF x (km)')
    ylabel('ECEF y (km)')
32  title('Number of GPS Satellites Tracked vs Position')
    colorbar;
```

Listing 3: AO-40 Validation

```
clear; close all; clc;

addpath('../code');
addpath('../code/helpers');
addpath('../config');

rx.RXNoise = -4.5; % dB-Hz
rx.r = [60000,0,0]' * 1000; % m
rx.u = -rx.r/norm(rx.r); % unit vector representing rx antenna orientation
rx.Tsys = 190; % Kelvin ***double check this number
rx.RXGain = 4; % dB
gpsWeek = 1920;
gpsSec = 345680;
thresh = 40;

satdata = retrieveNavigationData(gpsWeek,gpsSec,0,'../NavFiles');
gpsSecVec = gpsSec+(1:48*4)*15*60;
for i = 1:(48*4)
    [ tracked, neededGain(i) ] = observe( rx , gpsWeek , gpsSecVec(i),
   retrieveNavigationData(gpsWeek,gpsSecVec(i),0,'../NavFiles') , thresh );
    numSats(i) = numel(tracked);
end

figure(1);
stairs(gpsSecVec,numSats);
xlabel('GPS Second');
ylabel('Number of Satellites');
title('Satellites Tracked vs Time for AO40 Replication');
print('../Images/ao40actual','-dpng');
mean(numSats)

% Lower the threshold
thresh = 25;

satdata = retrieveNavigationData(gpsWeek,gpsSec,0,'../NavFiles');
gpsSecVec = gpsSec+(1:48*4)*15*60;
for i = 1:(48*4)
    [ tracked, neededGain(i) ] = observe( rx , gpsWeek , gpsSecVec(i),
   retrieveNavigationData(gpsWeek,gpsSecVec(i),0,'../NavFiles') , thresh );
    numSats(i) = numel(tracked);
end

figure(2)
stairs(gpsSecVec,numSats);
xlabel('GPS Second');
ylabel('Number of Satellites');
title('Satellites Tracked vs Time for Modified AO40 Replication');
print('../Images/ao40better','-dpng');
mean(numSats)
```

Listing 4: Determine Angles

```
function [phiDeg, offboreDeg] = findRotationAngle( sat, gpsWeek, gpsSec )
% findRotationAngle - write this header
e_h = cross(sat.r,sat.v)/(norm(cross(sat.r,sat.v)));
% convert time to days since solstice
dnum = gps2utc( gpsWeek, gpsSec );
r_sun_ecef = findSun(dnum);
e_sun_ecef = r_sun_ecef/norm(r_sun_ecef);
% find elevation of sun above orbital plane
beta = acos(dot(e_sun_ecef,e_h)); % radians
% find midnight position
u_midnight_ecef = -e_sun_ecef - (dot(-e_sun_ecef,e_h)*e_h);
% find angle from midnight
mu = acos(dot(u_midnight_ecef,sat.r)/(norm(u_midnight_ecef) * norm(sat.r)));
    % radians
% find yaw angle, igs
yaw_igs = atan2(-tan(beta),sin(mu)); % radians
Q_rtn_bf = rotate(-yaw_igs,3);
Q_rtn_ecef = [ sat.u' ; (cross(e_h,sat.r/norm(sat.r)))' ; e_h' ]*[0 0 1;1 0
    0;0 1 0]';
% turn yaw angle to pointing angle
x_sv_ecef = Q_rtn_ecef' * Q_rtn_bf' * [1; 0; 0];
y_sv_ecef = Q_rtn_ecef' * Q_rtn_bf' * [0; 1; 0];
z_sv_ecef = Q_rtn_ecef' * Q_rtn_bf' * [0; 0; -1];
% change pointing angle to phi
rel_xy = sat.rel - dot(sat.rel,z_sv_ecef)*z_sv_ecef/(norm(z_sv_ecef)^2);
phiDeg = acosd(dot(x_sv_ecef,rel_xy)/(norm(x_sv_ecef) * norm(rel_xy)));
if dot(y_sv_ecef,rel_xy) > 0 % based on how Lockheed defines angles
    phiDeg = 360-phiDeg;
end
if norm(rel_xy) == 0
    phiDeg = 0;
end
rel_zy = sat.rel - dot(sat.rel,x_sv_ecef)*x_sv_ecef/(norm(x_sv_ecef)^2);
offboreDeg = acosd(dot(-z_sv_ecef,rel_zy)/(norm(z_sv_ecef) * norm(rel_zy)));
if dot(y_sv_ecef,rel_zy) > 0 % based on how Lockheed defines angles
    offboreDeg = -offboreDeg;
end
if norm(rel_zy) == 0
    offboreDeg = 180;
end
end
```

Listing 5: Find Sun

```
function [ r_ecef ] = findSun( dnum )
%findSun - provides the position vector of the sun in WGS-84 ECEF
% coordinates as a function of the current date
%
%   Inputs: dnum - Matlab datenum
%
```

```
%   Outputs: r_ecef - 3x1 vector (m)
%
%   BEWARE: Degrees everywhere, also: unsure of how to check
%
%   Reference:
%       Meeus, Jean, Astronomical Algorithms (2nd Ed.). Richmond:
%           Willmann-Bell, Inc., 1998, Ch. 25-26
T = (dnum-datenum(2000,1,1,12,0,0))/36525;
L = 280.46646 + 36000.76983 * T + 0.0003032 * T^2;
M = 357.52911 + 35999.05029 * T + 0.0001537 * T^2;
e = 0.016708634 - 0.000042037 * T - 0.0000001267 * T^2;
C = (1.914602 - 0.004817 * T -0.000014 * T^2) * sind(M) + ...
    (0.019993 - 0.000101 * T) * sind(2*M) + ...
    0.000289 * sind(3*M);
s = L + C;
v = M + C;
R = 149597870700*(1.000001018 * ( 1 - e^2 ))/(1 + e*cosd(v));
omega = 125.04 - 1934.136 * T;
long = s - 0.00569 - 0.00478 * sind(omega);
obliq = 23.0 + (26.0 + (21.448 - T * (46.8150 + T * ...
                            (0.00059 - 0.001813 * T))) / 60) / 60 + ...
                            0.00256 * cosd(omega);

r_ecliptic = R*[cosd(long);sind(long)*cosd(obliq);sind(long)*sind(obliq)];
r_ecef = eci2ecef(r_ecliptic,dnum);
end
```

Listing 6: Gain Lookup

```
(* ::Package:: *)

function [ gain ] = SatDirectivityGain( gainMat, offbore, phi  )
% SatDirectivityGain - use gainMat lookup table to find gain
% corresponding to angles offbore and phi. Bilinear interpolation
% dirGain(i,j) = bore(-90+2(i-1)) , phi(10*(j-1))

lowoff = rem(2*floor(offbore/2),90);
highoff = rem(2*ceil(offbore/2),90);
lowphi = rem(10*floor(phi/10),360);
highphi = rem(10*ceil(phi/10),360);

lowi = (lowoff + 90)/2 + 1;
highi = (highoff + 90)/2 + 1;
lowj = lowphi/10 + 1;
highj = highphi/10 + 1;

if (lowi == highi)
    if (lowj == highj)
        gain = gainMat(lowi,lowj);
    else
        gain = (phi-highphi)*gainMat (lowi,lowj)/10 + (phi-lowphi)*gainMat
   (lowi,highj)/10;
```

```
    end
elseif (lowj == highj)
    gain = abs(offbore-highoff)*gainMat (lowi,lowj)/2 +
    abs(offbore-lowoff)*gainMat (highi,lowj)/2;
else

    b = inv ([1 lowoff lowphi lowoff*lowphi; 1 lowoff highphi lowoff*highphi;
    1 highoff lowphi highoff*lowphi; 1 highoff highphi highoff*highphi])' ...
    * [1;offbore;phi;offbore*phi];

    gain = b(1)*gainMat(lowi,lowj) + b(2)*gainMat(lowi,highj) +
    b(3)*gainMat(highi,lowj) + b(4)*gainMat(highi,highj);
end
end
```

References

[1] P. Misra and P. Enge. *Global Positioning Systems*. Lincoln, Massachusetts: Ganga-Jumana Press, revised second ed., 2012.

[2] F. Bauer. *GPS Space Service Volume*. PNT Advisory Board Meeting, NASA. June 11, 2015.

[3] T. Ebinuma and M. Unwin. *GPS Receiver Demonstration on a Galileo Test Bed Satellite*. The Journal of Navigation, The Royal Institute of Navigation. Pages 349-362, Vol. 60, No. 3, September 2007.

[4] O. Montenbruck et al. *GNSS Satellite Geometry and Attitude Models*. Advances in Space Research. Pages 1015-1029, Vol 56, 2015.

[5] J. Kronman. *Experience Using GPS For Orbit Determination of a Geosynchronous Satellite*. GPS 2000, Institute of Navigation. Pages 1622-1626, September 2000.

[6] M. Moreau et al. *Results from the GPS Flight Experiment on the High Earth Orbit AMSAT Oscar-40 Spacecraft*. GPS 2002, Institute of Navigation. September 2002.

[7] W. Marquis and D. Reigh. *The GPS Block IIR and IIR-M Broadcast L-band Antenna Panel*. Navigation, Institute of Navigation. Pages 329-347, Vol. 62, Issue 4, Winter 2015.

[8] J. Meeus, *Astronomical Algorithms*. Richmond: Willmann-Bell Inc., second ed., 1998. Ch. 25-26