Lab 2

1. Using the RCVR_S1 data from the class website:

(a) Using the ephemeris, determine the SV positions as a function of GPS time. You can check your calculation using initial satellite positions provided. Provide a sky plot of the SV positions over the period of the data collection

(b) Using the pseudoranges and SV positions, calculate the GPS position. You can check your calculations using the initial GPS position provided.

(c) Convert the ECEF positions to LLA and plot in Google Earth, Google Maps, or GPS visualizer. Where was the data taken and how much error/wander does there appear to be over the static data set?

(d) Convert the ECEF positions from the Novatel to ENU (use Toomers Corner as the reference location). Plot the E,N,U positions vs. time and characterize their errors. Compare this to the DOP.

(e) Calculate and plot the velocity in ENU coordinates. What measurement did you use for velocity? What is the accuracy in each axis?

Solution:
Satellite positions were determined by using the calc_gps_sv_pos function given on the class website. This function takes in given ephemeris data, a GPS transmit time and a calculated transit time. The transit and transmit time was calculated with the following equations:

$$t_{transit} = \frac{\rho}{c}$$
$$t_{transmit} = t_{GPS} - t_{transit}$$

This results in the following skyplot:

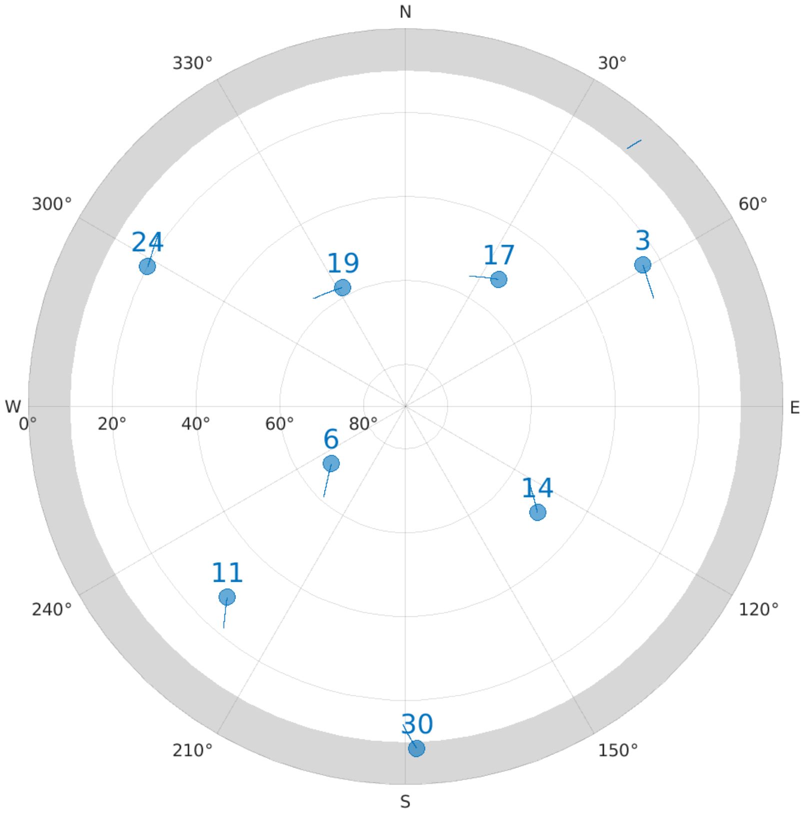

Figure 1: Static Receiver Skyplot.

The receiver position was calculated using a Gauss-Newton least squares approach shown in the following code.

```
function [x,b,P,DOP,i] = gnssPos(x_sv, psr, psr_var, x_hat)
% get number of measurements
N = size(x_sv, 1);
% determine if weighted or unweighted least squares
if length(psr_var) == 1
    W = eye(N);
    wgt = false;
else
    W = inv(diag(psr_var));
    wgt = true;
end
% initial conditions
if nargin < 4
    x_hat = [0;0;0;0];
end
error = Inf;
i = 0;
```

```
% least squares iterations
while (error > min(psr_var.^2)*1e-3) && (i < 10)
% while error > 1e-6
    i = i + 1;
    u = x_sv - x_hat(1:3)';
    r = sqrt(sum(u.^2,2));
    uv = u./r;

    H = [-uv, ones(N,1)];    % geometry matrix [-ux/r, -uy/r, -uz/r, 1]
    y = psr - (r + x_hat(4)); % meas. vector [rho - (r + b)]
    dx = inv(H'*W*H)*H'*W*y;

    x_hat = x_hat + dx;
    error = norm(dx);
end
x = x_hat(1:3);
b = x_hat(4);
% calculate covariance
if wgt
    P = inv(H'*W*H);
else
    P = psr_var^2 .* inv(H'*H);
end
DOP = inv(H'*H);
end
```

Using the code an initial receiver position was found to be:

$$x\,[\text{m}] = \begin{bmatrix} 422597.61 & -5362875.44 & 3415503.25 \end{bmatrix}$$
$$x\,[\circ, \circ, \text{m}] = \begin{bmatrix} 32.586296 & -85.494370 & 209.357 \end{bmatrix}$$

Plotting the position for the first 914 seconds of the static data:

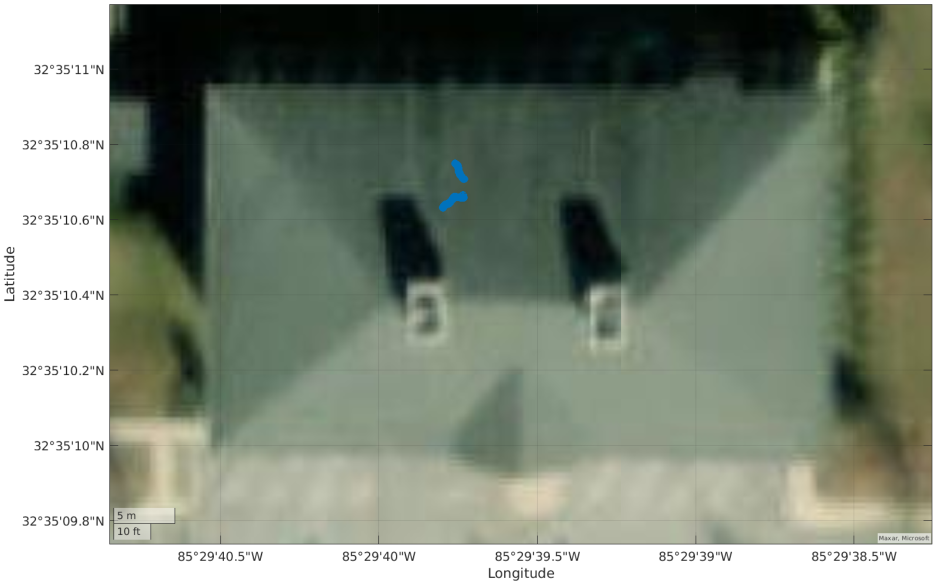

Figure 2: Positioning of the Static Receiver.

By looking at the geoplot it can be seen that the data was taken at the Auburn MRI building. There does seem to be some wander on the position measurements, as two line segments are formed by the position measurements drifting over time. However, this drift is limited to less than a few meters of error. Exact ENU positions were calculated with Toomer's Corner as a reference position.

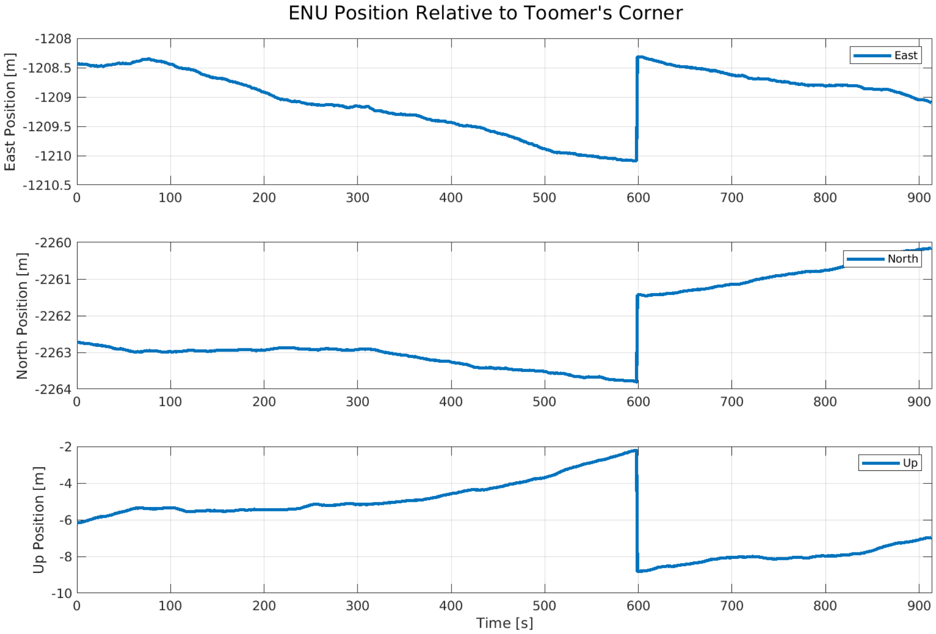

Figure 3: ENU Position Relative to Toomer's Corner.

Based on the ENU positions it can be seen that the positions errors remain less than a meters until a satellite is lost at 600 seconds. The plots of the DOP also show this jump in position error.

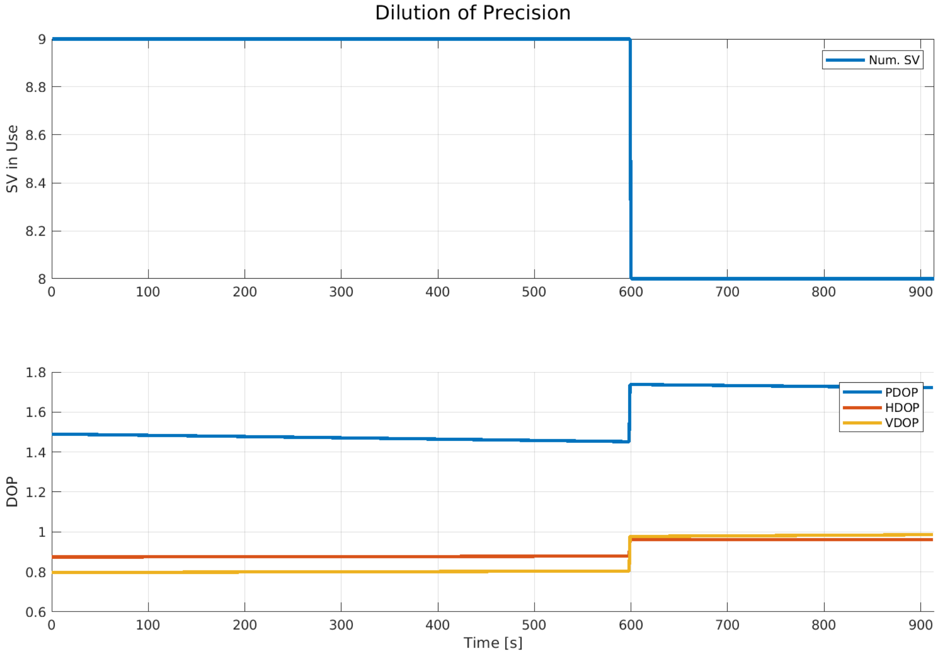

Figure 4: Static Receiver Dilution of Precision.

The PDOP shows that the position error should be within 1.5 meters and is held relatively constant. The ENU position drift also remains close to this 1.5 meter of error threshold.

The following is a plot of the ENU velocities:

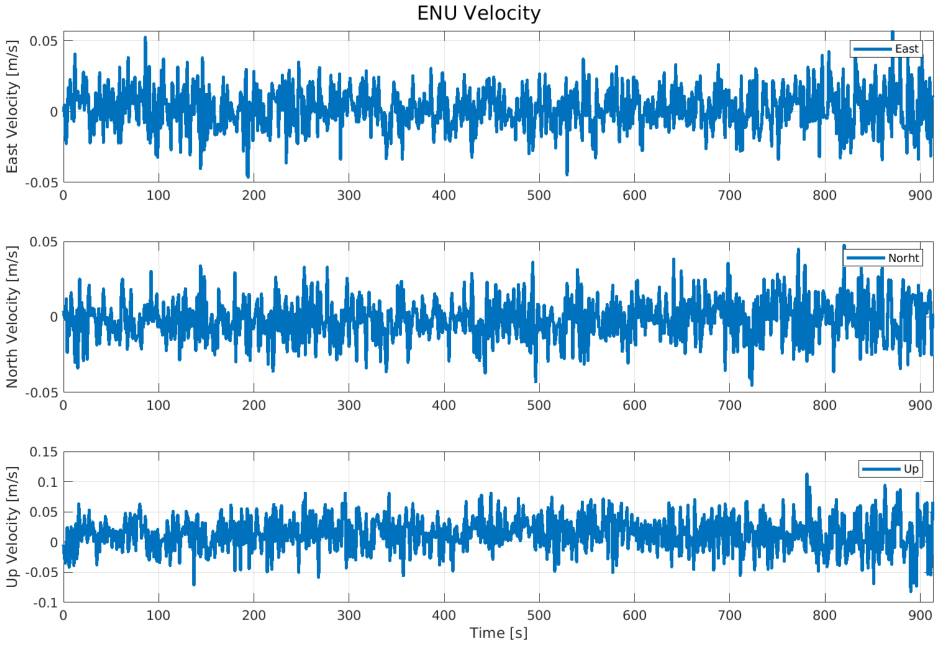

Figure 5: ENU Velocity.

The ECEF velocities were calculated using the doppler measurements and satellite velocities with a closed form least squares solution.

```
function [xDot, bDot] = gnssVel(x_sv, v_sv, x_user, dopp)
N = size(x_sv, 1);
% generate unit vector and range to satellite
u = x_sv - x_user';
r = sqrt(sum(u.^2,2));
uv = u./r;
% least squares parameters
H = [-uv, ones(N,1)];
y = dopp - sum(uv.*v_sv, 2);
x = inv(H'*H)*H'*y;
xDot = x(1:3);
bDot = x(4);
end
```

The ECEF velocities were converted to ENU using the MATLAB function *ecef2enuv*. The ENU velocities for each axis are approximately zero mean. For a majority of the run, the accuracy at each axis is below a magnitude of 0.05 meters per second.

2. Using the RCVR_D1 dynamic data from the class website:

(a) Using the L1 pseudoranges and SV positions, calculate the GPS positions for the dynamic data set. We have provide the first solution for you to verify your answer. What is your expected position error?

(b) Convert the ECEF positions LLA and plot in Google Earth, Google Maps, or GPS visualizer (where was the data taken and what was happening)?

(c) Calculate the moving platform's velocity. Plot the Speed and Course vs. time.

(d) Plot the clock bias and clock drift vs. time

Solution:

Using the same position and velocity functions described above, receiver position and velocity solutions were calculated for the dynamic data. This results in the position plot below.

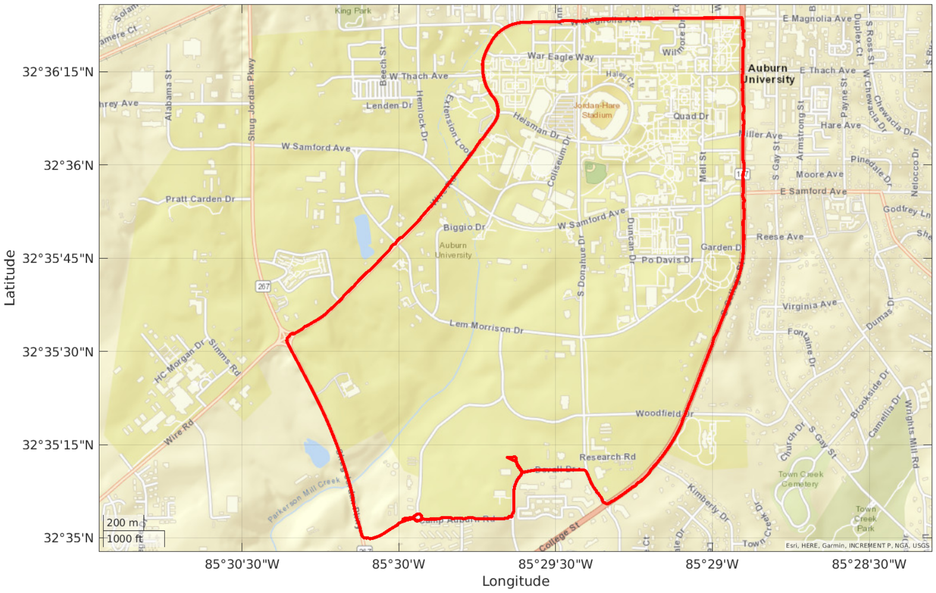

Figure 6: Dynamic Receiver Path.

Note the two timestamps had erroneous data where there were fewer than four satellites in view. These data points were removed from the solution. The data was taken on the streets around Auburn's campus. The path begins and ends at the Auburn MRI building parking lot.

Below is a plot of the DOPs for this dynamic data:

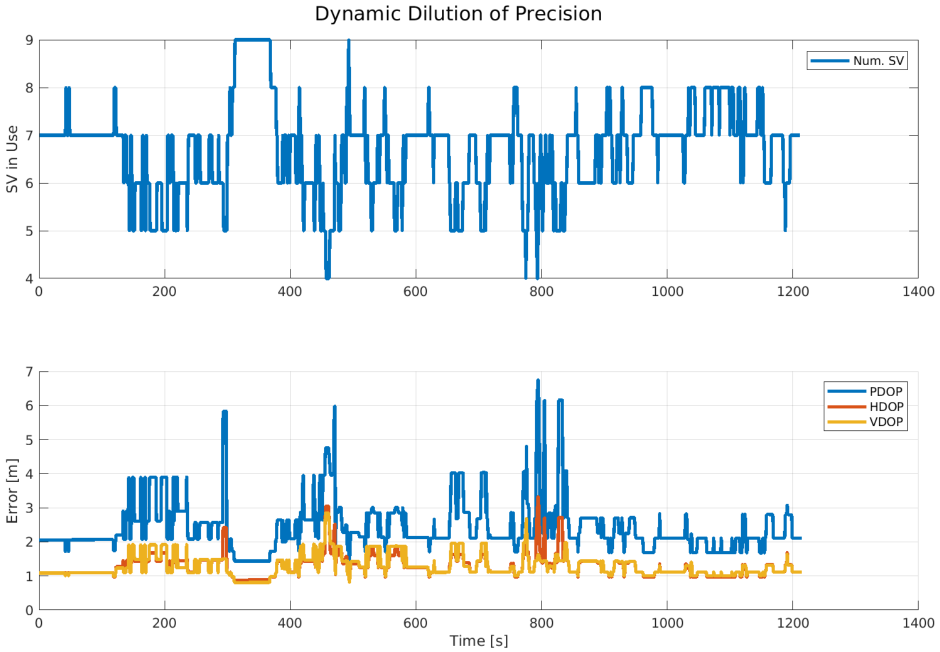

Figure 7: Dilution of Precision for the Dynamic Receiver.

The PDOP shows that the position error will remain below 7 meters at all time points and has a direct correlation to the number of available satellites.

The plots for speed and course is shown below:

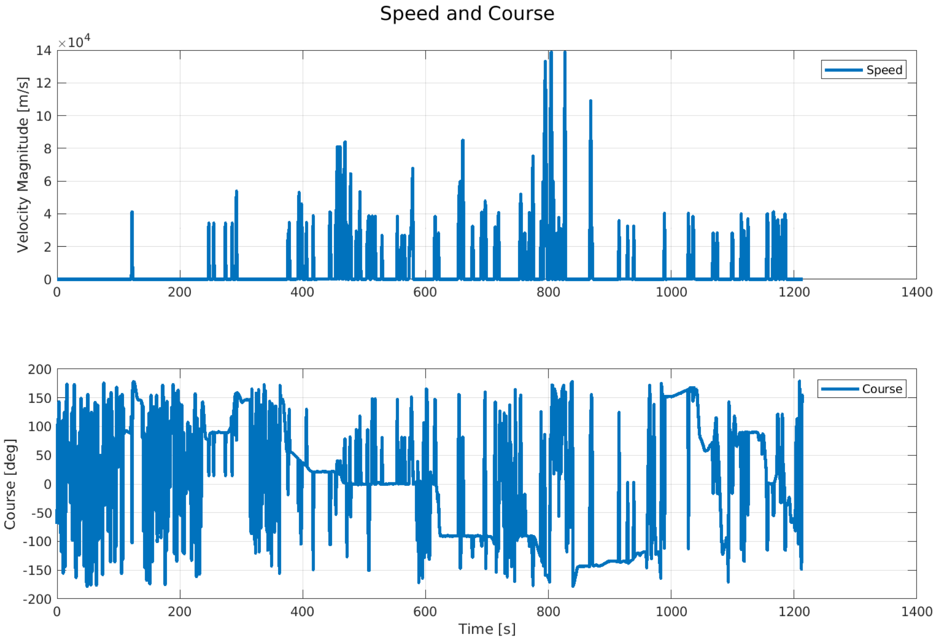

Figure 8: Speed and Course of Receiver.

As seen in the plot, the velocity does not remain smooth throughout the run. The velocity spikes to a unreasonable values multiple times. The general course can be followed, but is highly dependent on the velocity so it shows the same erroneous spikes.

The clock bias and drift is plotted below:

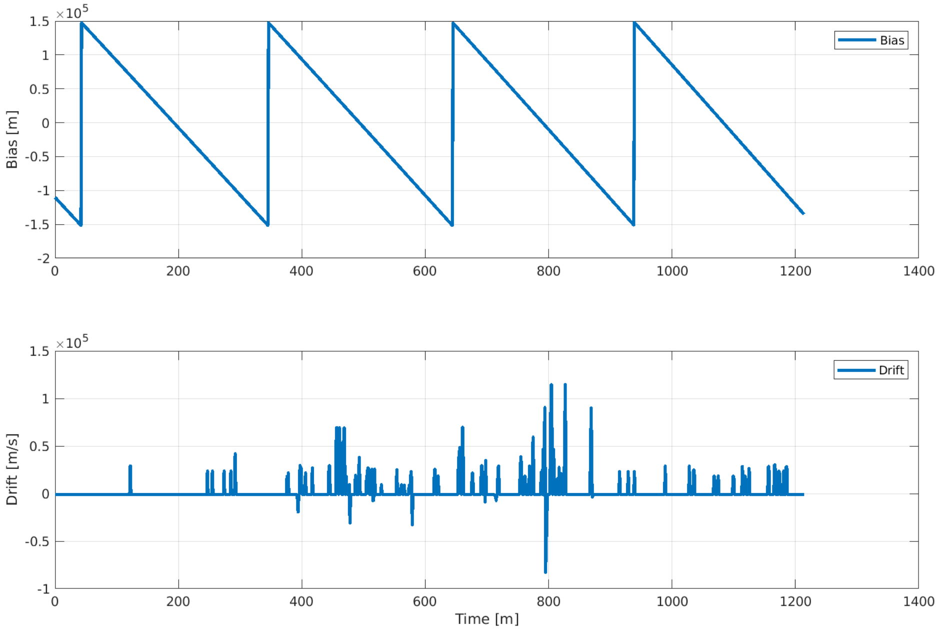

Figure 9: Clock Bias and Drift.

The clock bias constantly decreases to $-1.5(10^5)$ where is gets wrapped back around to $+1.5(10^5)$. If you were to plot the raw pseudoranges for this data set they follow this same pattern. The clock drift remains close to zero, but once again shows spikes that correspond to the spikes in velocity.

3. Bonus from LAB 1.

(a) Get Ephemerides Online. Using the RINEX files described in class, determine the navigation message file that will contain the appropriate parameters to calculate the satellite positions. Describe the parts of the filename that indicate this is the correct file. You may also be able to get the ephemeris from other websites in a more direct form.

(b) Sky View Plot. In terms of GPS week and seconds of week, when are the positions to be calculated? Generate the polar sky plot shown in Lab 1 – Geocaching and compare to the Sky Plot from Lab 1.

Solution:

A rinex file for January 31, 2023 at UTC time 17:21:27 was downloaded from `https://cddis.nasa.gov/archive/gnss/data/hourly/2023/031/17/`. A file from this link was chosen because it provided the closest ephemeris time to the actual time of the measurement. The ephemeris time provided in the rinex file was 2023/01/31 17:06:21, 15 minutes before our collection. The filename (AMC400USA_R_20230311600_01H_GN.rnx) provides an approximate location with the USA flag, the date given as a string of numbers, as well as GN at the end signifying it contains GPS navigation data. Below are skyplots from `http://gnssmissionplanning.com` as well as the one generated with this acquired ephemeris data.

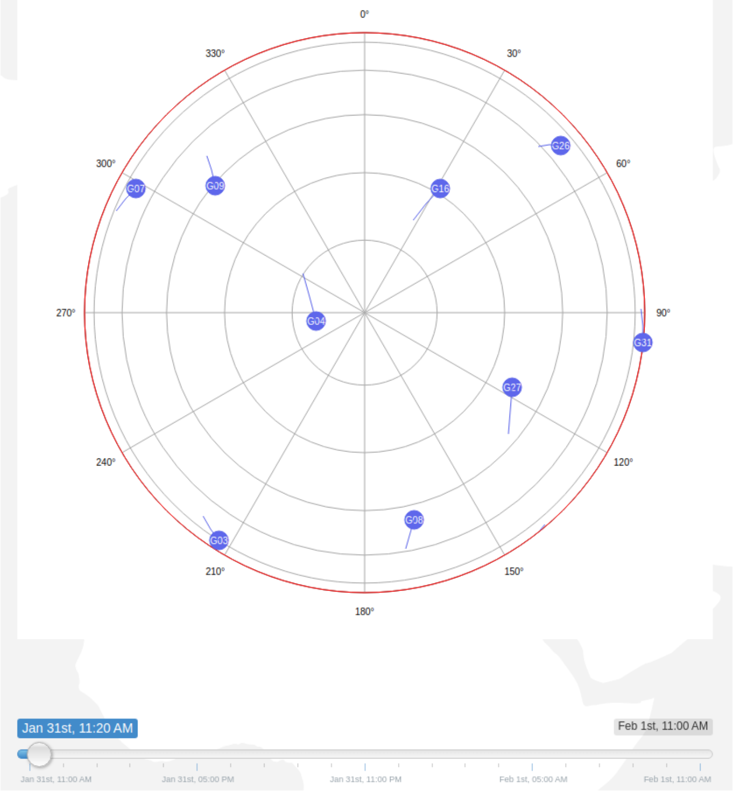

Figure 10: Skyplot from GNSSMissionPlanning

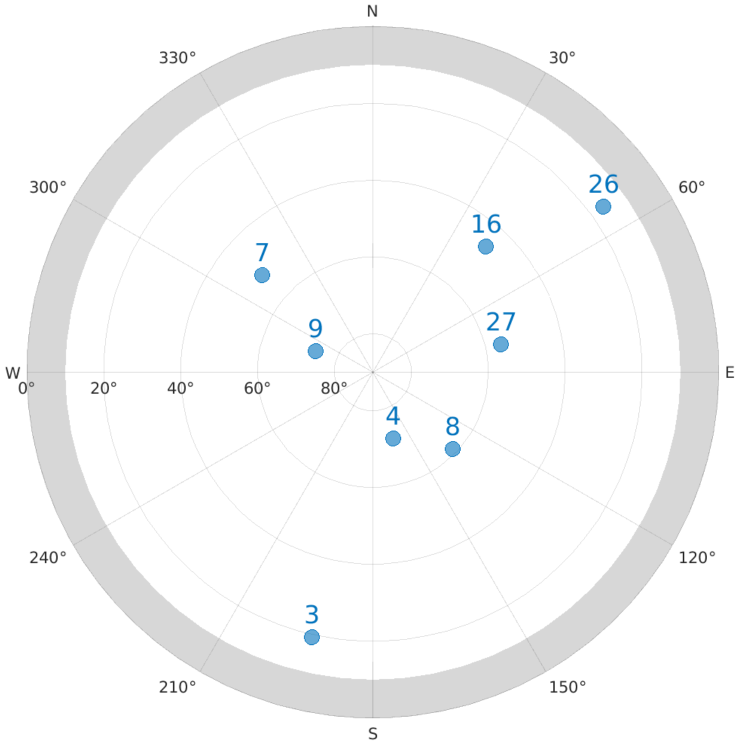

Figure 11: Recreated Skyplot.

The recreated skyplot was taken at GPS week 2247 and GPS seconds 148887. Both skyplots show the same satellites, however satellite 31 was at a 0 degree elevation angle in the skyplot from GNSSMissionPlanning. This same satellite does not appear in the recreated skyplot because the satellites are not in the same positions. The positions of all of the satellites are shifted in the recreated skyplot. This is likely due a discrepancy between the downloaded ephemeris and the GNSSMissionPlanning ephemeris.