The Georgia Institute of Technology

ECE 6390 Satellite Communication and Navigation Systems Fall 2011 Homework Assignment 3

Mason Nixon

25th November, 2011

Assignment

Below are the satellite subpoint coordinates and pseudo-ranges for the 4 GPS satellites whose signals are received by a positioning radio.

	Sat 1	Sat 2	Sat 3	Sat 4
Lon	-85.2234°	-62.3256°	-75.2009°	-95.0234°
Lat	35.5521°	43.2345°	29.0510°	40.7777°
PR/c	0.000000000s	0.001588238s	0.000341042s	0.000493039s

Note that all pseudo-ranges are given relative to the first satellite position (absolute time delay, of course, is unknown). Calculate the longitude, latitude, and altitude for the GPS receiver based on this data. Use 6380.20 km as the mean earth radius, 20,200.00 km as the GPS satellite altitude, and c = 299860.00 km/s in your calculations.

1 Solution

1.1 Process

To determine a location using the GPS satellites, we must know the precise location of three satellites and the precise amount of time it takes to receive a pulse from those satellites. Since we do not have accurate timing, we must be able to receive location information from an additional satellite to determine the fourth variable, clock error. Each equation defines a sphere and if we had two equations, i.e. two spheres, solving them would find the intersection of two spheres, which is a circle. By solving three equations, we are effectively determining the location of two points. This is because the intersection of a circle and a sphere occurs at two different points. By using this iterative method and by assuming that we may choose Atlanta as the initial point, we are able to eliminate one of the two possible solutions.

The method used was to 1) Calculate the Cartesian coordinates of the GPS satellites, 2) Select an initial position, 3) Use iterative equations to find the position, 4) Convert to spherical coordinates to view on Google Maps.

1.2 Simulation

```
% Name: Mason Nixon
% Date: Novemeber 29, 2011
% Description: The following program finds a location based on the location
% and pseudo-range of 4 satellites.

clc
clear

rE = 6380.20; % Radius of Earth (km)
gpsAlt = 20200.00; % Altitude of GPS satellite (km)
c = 299860.00; % Speed of light (km/s)

Lon = [-85.2234, -62.3256, -75.2009, -95.0234];
Lat = [35.5521, 43.2345, 29.0510, 40.7777];
T = [0, 0.001588238, 0.000341042, 0.000493039];

xi = inline('R*cos(Latitude)*cos(Longitude)');
yi = inline('R*cos(Latitude)*sin(Longitude)');
zi = inline('R*sin(Latitude)');

for a1 = 1:4
    x(a1) = xi(Lat(a1)*(pi/180),Lon(a1)*(pi/180),rE+gpsAlt); % Note: PR=T*c
    y(a1) = yi(Lat(a1)*(pi/180),Lon(a1)*(pi/180),rE+gpsAlt);
    z(a1) = zi(Lat(a1)*(pi/180),rE+gpsAlt);
end

% Need an initial value, use Atlanta
atl = [-84.4, 33.76, 6380];
xAtl = xi(atl(1)*(pi/180),atl(2)*(pi/180),atl(3)); % Note: PR=T*c
yAtl = yi(atl(1)*(pi/180),atl(2)*(pi/180),atl(3));
zAtl = zi(atl(1)*(pi/180),atl(3));

% (xi-ux)^2+(yi-uy)^2+(zi-uz)^2-(Ti*c)^2 = 0;
% Assume that we are at the first satellite then use the Jacobian to
% determine our actual location with respect to it.
equ = zeros(4,1);
terms = zeros(4,4);
x0 = [xAtl;yAtl;zAtl;0];
for a1 = 1:100
    for a2 = 1:4
        orig(a2) = (x(a2)-x0(1))^2 + (y(a2)-x0(2))^2 + (z(a2)-x0(3)).^2;
    end
    for a3 = 1:4
        terms(a3,1) = -(orig(a3)^(-1/2)/c)*(x(a3)-x0(1));
        terms(a3,2) = -(orig(a3)^(-1/2)/c)*(y(a3)-x0(2));
        terms(a3,3) = -(orig(a3)^(-1/2)/c)*(z(a3)-x0(3));
        terms(a3,4) = 1;
    end
    for a4 = 1:4
        equ(a4) = (1/c)*sqrt((x(a4)-x0(1))^2 + (y(a4)-x0(2))^2 + ...
            (z(a4)-x0(3))^2) + x0(4) - T(a4);
    end
    u = x0 - terms\equ;
    x0 = u;
end

finalLat = asin(u(3)/(rE));
finalLon = atan2(u(2),u(1));

finalLat = finalLat*(180/pi); % Convert from radians to degrees
finalLon = finalLon*(180/pi);

% Convert to degrees, minutes, seconds
degLon = floor(abs(finalLon));
minSecLon = abs(finalLon) - abs(degLon);
minLon = floor(minSecLon*60);
secLon = ((minSecLon*60) - minLon)*60;
if(finalLon/abs(finalLon) == -1)
    dirLon = 'W';
else
    dirLon = 'E';
end

degLat = floor(abs(finalLat));
minSecLat = abs(finalLat) - abs(degLat);
minLat = floor(minSecLat*60);
secLat = ((minSecLat*60) - minLat)*60;
if(finalLat/abs(finalLat) == -1)
    dirLat = 'S';
else
    dirLat = 'N';
end

fprintf('Latitude: %2.0f%c%2.0f''%2.0f" %s (%7.7f)\n',...
    abs(degLat),char(176),minLat,secLat,dirLat,finalLat)
fprintf('Longitude: %2.0f%c%2.0f''%2.0f" %s (%7.7f)\n',...
    abs(degLon),char(176),minLon,secLon,dirLon,finalLon)
fprintf('Altitude: %7.1f meters\n', sqrt(u(1)^2+u(2)^2+u(3)^2))
disp('Google Maps: ')
fprintf(...

'http://maps.google.com/maps?f=q&source=s_q&q=&vps=3&jsv=166d&sll=%5.9f,%5.9f\n',...
    finalLat,finalLon)
```

1.3 Simulation Output

The output of the program is as follows:

```
Latitude: 33°47'20" N (33.7888107)
Longitude: 84°23'60" W (-84.3999084)
Altitude:  6380.5 meters
Google Maps:
http://maps.google.com/maps?f=q&source=s_q&q=&vps=3&jsv=166d&sll=33.788810703,-
84.399908435
```

2 Results

When one types in the coordinates into Google Maps using the URL above, the address is 1168-1210 Mecaslin St NW, Atlanta, GA 30318 which seems to be around a baseball field and at a complex of townhouses.

Other potential sources of error in a GPS transmission were not considered such as: Selective Availability, Ephemeris Errors, Satellite Clock Error, Ionospheric Delay, Tropospheric Delay, Multipath, and Receiver Noise. It should be noted that multipath would be significant here due to the location being in Midtown Atlanta where tall buildings tend to create multiple propagation paths of the signal. If all of these errors were present, it would introduce an offset of about 22 m to the calculation which would not hinder one's ability to locate the appropriate destination.