HW4: GPS Positioning Accuracy

Contents

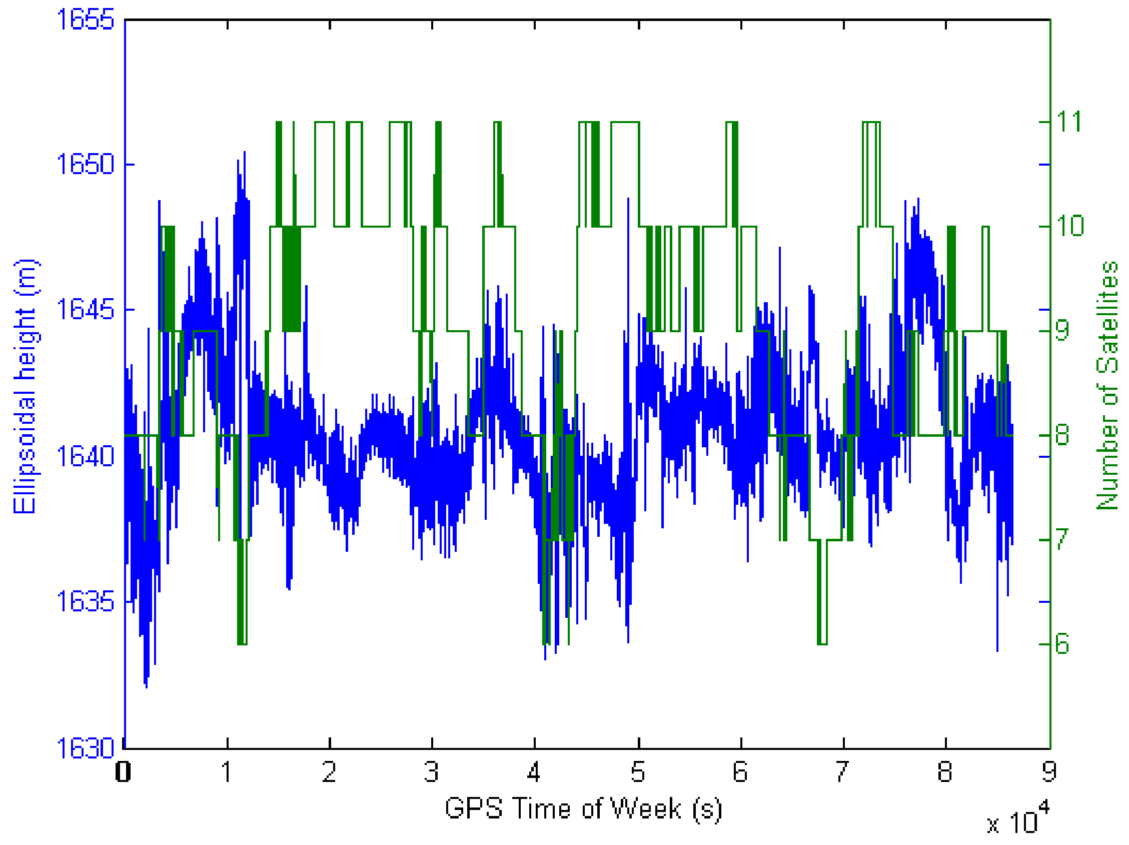
Initialize

```
fprintf('\n');
clearvars -except function_list pub_opt
close all
```

1) Ellipsoidal height vs time

The variation of ellipsoidal height seems to vary more when fewer satellites are used in the position solution.

```
load('GPS2009L1L2_data.mat');
time = data(:,1);
lat = data(:,2);
long = data(:,3);
ell_h = data(:,4);
num_sat = data(:,5);
[haxes, hline1, hline2] = plotyy(time, ell_h, time, num_sat);
ylim(haxes(2), [min(num_sat)-1, max(num_sat)+1]);
xlabel('GPS Time of Week (s)');
ylabel(haxes(1), 'Ellipsoidal height (m)');
ylabel(haxes(2), 'Number of Satellites');
```



2) Latitude/Longitude errors

The scatter looks more varied in the y direction (North/South) than the x direction.

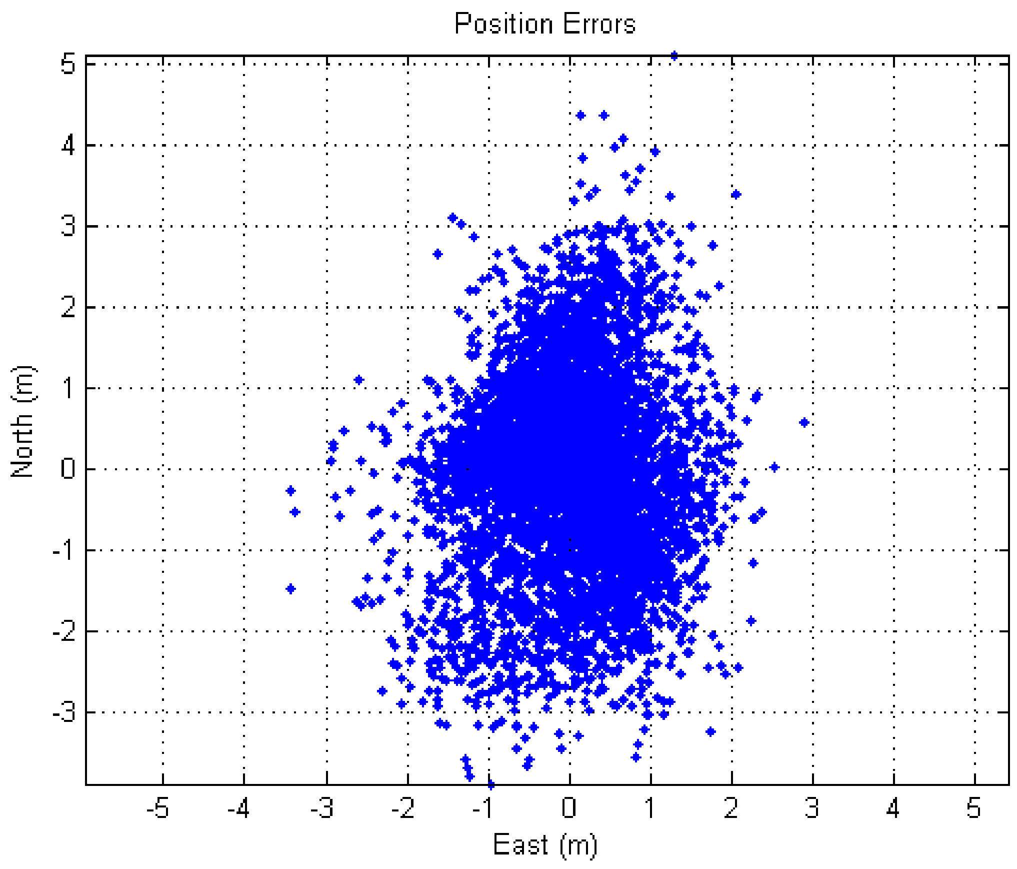
```
mean_lat = mean(lat);
lat_err = lat - mean_lat;
mean_long = mean(long);
long_err = long - mean_long;
nn = length(lat_err);
sig_lat = std(lat);
sig_mean_lat = sig_lat/sqrt(nn);
sig_long = std(long);
sig_mean_long = sig_long/sqrt(nn);
fprintf(sprintf('Mean latitude: %%.%df degrees\n', ...
    dec_places(sig_mean_lat)),mean_lat);
fprintf(sprintf('Mean longitude: %%.%df degrees\n', ...
    dec_places(sig_mean_long)),mean_long);

% Get the error arc-lengths
re = 6378e3; % m, assuming spherical
NS_err = 2*pi*re*lat_err/360;
EW_err = cos(lat*pi/180)*2*pi*re.*long_err/360;

figure
plot(EW_err, NS_err, '.')
title('Position Errors')
ylabel('North (m)')
xlabel('East (m)')
grid on
```

```
axis equal
hold on
```

```
Mean latitude: 40.0076470 degrees
Mean longitude: -105.2616708 degrees
```



3) Compute standard deviations, P, error ellipse, 50% CEP

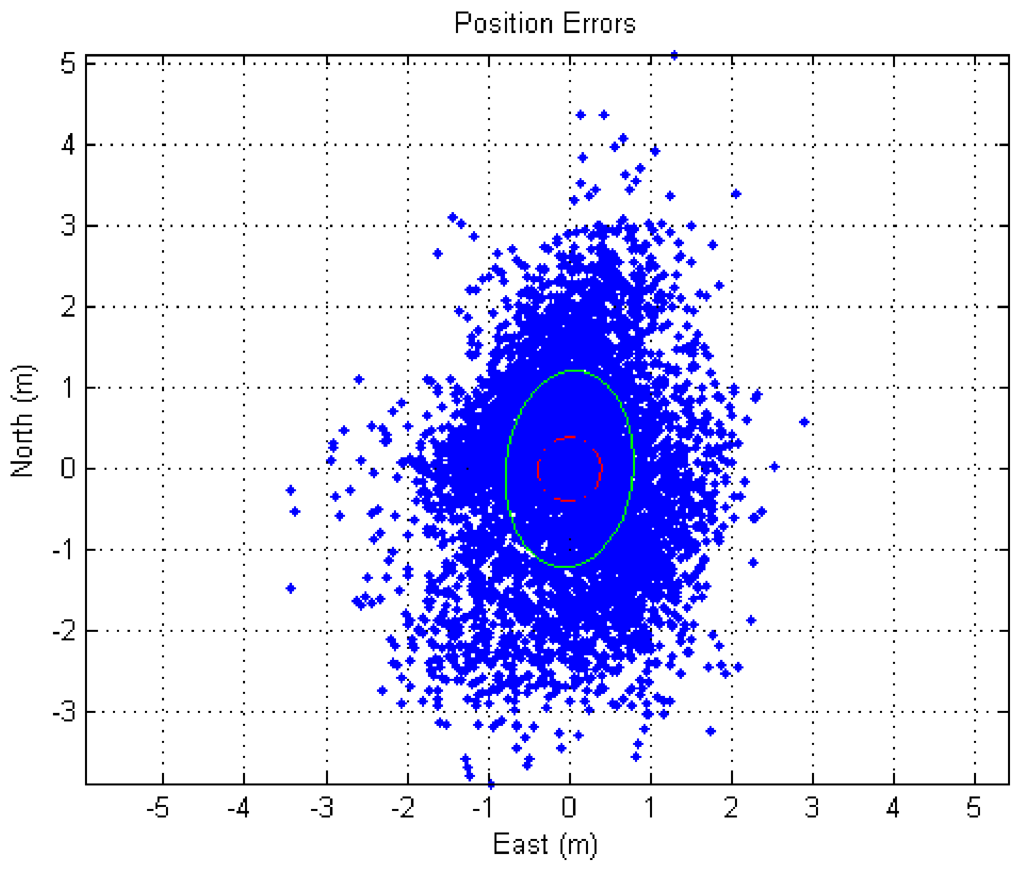
```
% Standard deviations:
sig_ell_h = std(ell_h);
fprintf('Latitude standard deviation = %.0e m\n', std(NS_err));
fprintf('Longitude standard deviation = %.0e m\n', std(EW_err));
fprintf('Ellipsoidal height standard deviation = %.0e m\n', sig_ell_h);

P = cov([EW_err NS_err]);
fprintf('Covariance matrix:\n');
for i = 1:2
    fprintf('\t')
    for j = 1:2
        if P(i,j) > 1
            fprintf('%4.3f\t', P(i,j));
        else
            fprintf(sprintf('%%.%df\t', dec_places(P(i,j))+3),P(i,j));
        end
    end
    fprintf('\n')
end
```

```
% Eigenvalues of the covariance matrix are the principle components
[evec,ev]=eig(P);
ell_a=sqrt(ev(1,1));%, evec(:,1)'
ell_b=sqrt(ev(2,2));%, evec(:,2)'
angle=atan2(evec(2,1),evec(1,1));
drawellipse(ell_a,ell_b,angle,0,0,'g-');
fprintf('Error Ellipse semimajor axis = %.3e m\n', ell_a);
fprintf('Error Ellipse semiminor axis = %.3e m\n', ell_b);

err_radii = sqrt((NS_err.*NS_err)+(EW_err.*EW_err));
CEP_50_radius = err_radii(round(nn/2));
fprintf('CEP 50%% radius = %.3e m\n', CEP_50_radius);
drawellipse(CEP_50_radius,CEP_50_radius,0,0,0,'r-.');
```

```
Latitude standard deviation = 1e+00 m
Longitude standard deviation = 8e-01 m
Ellipsoidal height standard deviation = 2e+00 m
Covariance matrix:
        0.6213  0.07308
        0.07308 1.464
Error Ellipse semimajor axis = 7.842e-01 m
Error Ellipse semiminor axis = 1.213e+00 m
CEP 50% radius = 3.907e-01 m
```



Published with MATLAB® R2013b

```
function abs_exponent = dec_places( input )
%dec_places Return the sci-no exponent of numbers less than 1. It's quick
%and dirty.
fcnPrintQueue(mfilename('fullpath')); % Add this code to code app

abs_exponent = 0;
comp_value = 1;
while input < comp_value
    abs_exponent = abs_exponent+1;
    comp_value = comp_value/10;
end
abs_exponent;
end
```

Published with MATLAB® R2013b

```
function C=drawellipse(a,b,th,xc,yc,lt)
%function C=drawellipse(a,b,th,xc,yc,lt)
%
% Input: a=semimajor axis length, b=semiminor axis length
%        th=angle between semimajor axis and positive x axis.
%        (xc,yc) are optional inputs specifying the ellipse center
%        lt is an optional input specifying the line type
% Output: rotation matrix
% Author: P. Axelrad, University of Colorado 2/94
fcnPrintQueue(mfilename('fullpath')); % Add this code to code app

E=0:pi/100:2*pi;
x=a*cos(E);
y=b*sin(E);

C=[cos(th) -sin(th); sin(th) cos(th)];

z=C*[x;y];

if (nargin == 4)
   ltt=xc;
elseif (nargin == 6)
   ltt=lt;
else
   ltt='g';
end

if (nargin > 4)
   z=[z(1,:)+xc; z(2,:)+yc];
end

plot(z(1,:),z(2,:),ltt)
axis('equal')
```

Published with MATLAB® R2013b

```
function fcnPrintQueue( filename )
global function_list;
if exist('function_list', 'var')
    file_in_list = 0;
    for idx = 1:length(function_list)
        if strcmp(function_list(idx), filename);
            file_in_list = 1;
            break
        end
    end
    if ~file_in_list
        fprintf('%s\n', filename);
        function_list = [function_list, filename];
    end
end
end
```

Published with MATLAB® R2013b