Navigation and Guidance Systems

Course 086759- Spring 2018

Assignment #10

(G.P.S)

Daniel Engelsman 300546173

1 Euphemeris message extraction

Table 1 shows the pseudo-ranges measured by a GPS receiver :

SV	PR [m]	X [m]	Y [m]	Z [m]
2	22228206.42	7766188.44	-21960535.34	12522838.56
26	24096139.11	-25922679.66	-6629461.28	31864.37
4	21729070.63	-5743774.02	-25828319.92	1692757.72
7	21259581.09	-2786005.69	-15900725.8	21302003.49

Table 1: Euphemeris location of an i-th satelite

(a) Using the iterative least-squares iteration, compute the position of the antenna :

In order to answer that question, I have used the following code :

```
%% ——— Antenna location via Euphemeirs extraction ——— %
clc; clear all; close all; set(0,'defaultfigurecolor',[1 1 1]);

n_itr = 1; xyz_err = 10;          % Initial Conditions
load('SV'); load('PR');           % Load preset files
X_hat = 0*[1 1 1 1]';             % Define Linearization Point
n_SV = length(PR);                % Number of SV
n_Parm = length(X_hat);           % Number of Parameters
while xyz_err > 1e-3              % Convergence Criterion
```

```matlab
    % ---------------- Hamiltonian ---------------- %
    for j = 1:n_SV
        for i = 1:n_Parm-1
            H(j,i) = ( X_hat(i,n_itr) - SV(i,j) ) /...
                norm( SV(:,j) - X_hat(1:n_Parm-1,n_itr) );
        end
        H(j,4) = 1;                  % Bias Derivative
        fx_0(j,1) = norm( SV(:,j) - X_hat(1:n_Parm-1,n_itr) );
    end
    dX(:,n_itr) = ( (H'*H)^(-1) )*(H')*(PR - fx_0);
    xyz_err(n_itr) = norm( dX(1:3,n_itr) );
    % -------- Update Estimated X_hat ---------- %
    X_hat(1:3, n_itr+1) = X_hat(1:3, n_itr) + dX(1:3, n_itr);
    X_hat(4, n_itr+1) = X_hat(4, n_itr) + dX(4, n_itr);
    n_itr = n_itr + 1;
end
```

Hereby is an $\hat{X}$ state vector table, showing convergence towards true location :

[#]	X^[m]	Y^[m]	Z^[m]	Bias [m]
1	0	0	0	0
2	-2977571.48	-5635278.16	4304234.51	1625239.8
3	-2451728.53	-4730878.46	3573997.52	1939310.53
4	-2430772.22	-4702375.8	3546603.87	2204060.24
5	-2430745.1	-4702345.11	3546568.71	2468751.37
6	-2430745.1	-4702345.11	3546568.71	2733442.5

Table 2: Calculated state vector vs. Iteration number

One can tell the sharp movement along iterations from the initial guessed linearization point towards the true one.

(b) Add a table with the error in each coordinate and the error of the overall position :

[#]	dR [m]	dX [m]	dY [m]	dZ [m]	Bias [m]
1	7034786.69	-2977571.48	-5635278.16	4304234.51	1625239.8
2	1170880.19	525842.94	904399.7	-730236.99	314070.73
3	41118.5585	20956.32	28502.66	-27393.65	264749.71
4	49.123280	27.12	30.69	-35.17	264691.13
5	0.00006891	4.06E-05	3.82E-05	-5.37E-05	264691.13

Table 3: Calculated ΔX vs. Iteration number

And, plotting it for a more intuitive presentation :

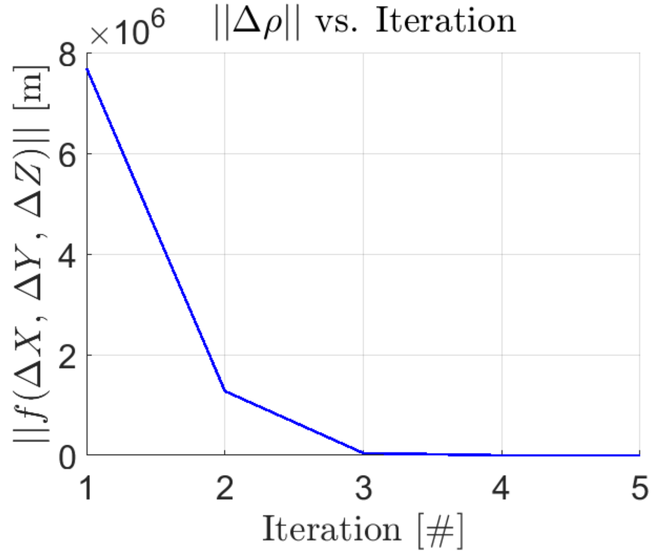

Figure 1: Overall position error vs. Iteration number

2 Additional PR measurement

Given an extra set of PR measurements, compute the length of the cable added :

	Pseudo-Range [m]			dX		
	PR_a [m]	PR_b [m]	delta	Iter. 5	Iter. 5	delta
X^[m]	22228206.42	22228209.42	3	-2430745.1	-2430745.1	0
Y^[m]	24096139.11	24096142.11	3	-4702345.11	-4702345.11	0
Z^[m]	21729070.63	21729073.63	3	3546568.71	3546568.71	0
Bias [m]	21259581.09	21259584.09	3	2733457.5	2733442.5	-15

Table 4: Re-calculation of ΔX relatively to section 1

The left side of the table shows the Pseudo-Range measurement in section 1 next to 2. The difference at each axis is 3 [m]. However, after running the algorithm, one can tell that new position is same as the previous. No $\Delta\rho$ exists, implying that no extra cable was added.

However, what is the cause for the Δ bias ?

The answer is probably the dynamic model or the initial guess combined within.

3 Altitude GPS errors

Altitude GPS errors are known to be almost twice as large as horizontal errors. Why ?

Each satellite is located with its own unique X,Y,Z coordinates. Since most of the satellites subjected to Ionospheric and Tropospheric errors, the altitude from which transmission can be made, is limited in a bounded range. As opposed, the horizontal distribution in the satellite's orbit is more diverse, at any angle, allowing to obtain smaller errors.

The accuracy of position is determined by the range errors and satellite configuration.

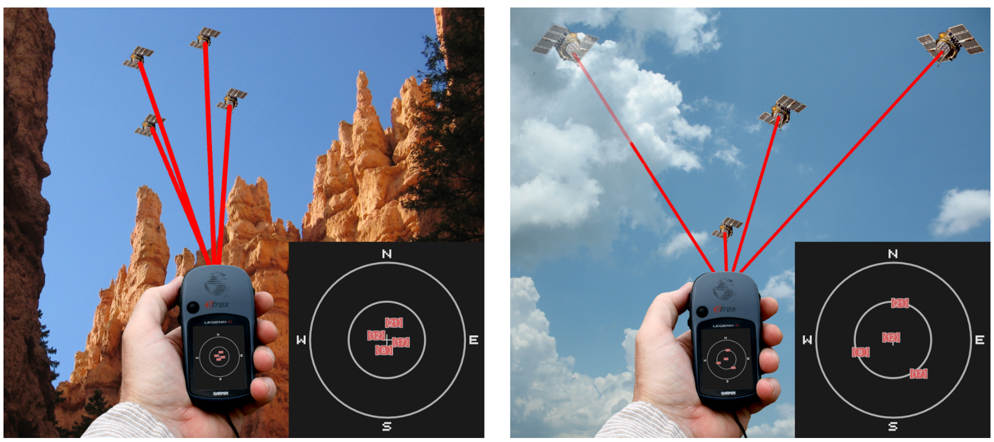

Figure 2: **Left :** Poor GDOP. **Right :** Good GDOP

4 GPS Computation Improvement

Given the Suppose you know the altitude at which a GPS antenna is located. How can you use that to improve the computation of a position solution?

(i) Initial Guess - Improving I.C. of the linearization point of $\hat{Z}_0$

(ii) Data fusion - combine it as an external source of information regarding our estimation

(iii) Observability - plug it in our dynamic model and improve observability by better knowing the state input.